IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 6, December 2025, pp. 4971~4981
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i6.pp4971-4981

Performance evaluation of pre-trained deep learning model on garbage classification with data augmentation approach

I Komang Arya Ganda Wiguna[1,2], I Gusti Made Ngurah Desnanjaya[3], I Kadek Budi Sandika[2]
[1]Department of Informatics, Faculty of Mathematics and Natural Sciences, Udayana University, Bali, Indonesia
[2]Department of Informatics, Institute of Business and Technology Indonesia, Denpasar, Indonesia
[3]Department of Computer System Engineering, Institute of Business and Technology Indonesia, Denpasar, Indonesia

Article Info

Article history:

Received Nov 10, 2024
Revised Aug 23, 2025
Accepted Sep 7, 2025

Keywords:

Classification
Data augmentation
Deep learning
Garbage
Pre-trained model

ABSTRACT

Waste classification is one of the interesting topics for classifications in which data can be very varied and complex. This data diversity is a challenge to develop a model that is able to classify well. The purpose of this study is to analyze the performance of the pre-trained deep learning model using a data augmentation approach. There are three pre-training models used in this study, namely residual networks 50 (ResNet50), visual geometric group with 16 layers (VGG-16), and MobileNetV2. The results showed that the MobileNetV2 model received the highest accuracy value, reaching 84.45% for data without augmentation. With data augmentation there is a decrease of 2.73%. Conversely, VGG-16 shows performance stability with an increase in accuracy with augmentation data, reaching 75.84%. While ResNet50 gets the lowest results compared to both models. The application of data augmentation techniques with the aim of increasing data variations does not always have an impact on increasing the generalization of the model.

Corresponding Author:

I Komang Arya Ganda Wiguna
Department of Informatics, Faculty of Mathematics and Natural Sciences, Udayana University
Bali, Indonesia
Email: kmaryagw@gmail.com

1. INTRODUCTION

Waste management has become an increasingly pressing global issue, with significant impacts on the environment, human health, and the economy. According to a World Bank report, global waste production reaches more than 2 billion tons per year [1] and continues to increase along with urbanization and improved living standards. This accumulation of waste causes various environmental problems, including water, soil, and air pollution [2]. In addition, the cost of waste management, which includes collection, transportation, and final disposal, is a heavy burden for many countries, especially developing countries with inadequate waste management infrastructure.

One of the main solutions proposed to reduce the negative impact of waste is effective segregation. This sorting separates waste into categories of organic, inorganic, and recyclable materials [3], thereby reducing the amount of waste entering landfills and increasing the efficiency of the recycling process. However, manual sorting requires high awareness from the community and requires significant labor and time.

To overcome this obstacle, technology plays an important role in developing a more efficient waste management system. One promising solution is the development of smart waste bins equipped with an automatic sorting system based on microcontroller technology [4]. This smart trash can uses sensors such as proximity and infrared to automatically detect the type of waste [5]. While effective in basic sorting, these systems have limitations when dealing with more complex types of waste, which often leads to errors in classification.

To improve sorting accuracy, the use of digital images as additional input is proposed. Computer vision and deep learning technologies offer solutions to improve the accuracy of waste type classification [6]. Computer vision allows the system to process litter images visually, while deep learning is used to perform litter detection and classification with a high degree of accuracy [7]. This approach is expected to overcome sensor limitations and produce a more effective and efficient litter classification system [8]. However, the main problem in image-based litter classification is the diversity of litter shapes, colors, and conditions that often differ from one image to another. These variations can make it difficult for deep learning models to generalize the data, especially when faced with new data or environmental conditions that are different from the training data.

As one of the widely used approaches, transfer learning using pre-trained deep learning models provides significant advantages in accuracy and training time efficiency. Pre-trained models such as convolutional neural network (CNN) [9], visual geometric group with 16 layers (VGG-16) [10], residual networks 50 (ResNet50) [11], and MobileNetV2 [12] are able to utilize rich and advanced visual representation features. The resulting accuracy performance of these models reaches 81 to 95% [13]. However, challenges in model generalization remain, especially when the model is used for classification of out-of-distribution data, i.e. data that has a different distribution than the data trained on [14].

One approach that can improve the generalization ability of the model is to perform data augmentation. Data augmentation aims to expand the distribution of training data through synthetic variations such as rotation, flipping, zooming, and other transformations, so that the model becomes more adaptive to different image variations [9]. Although data augmentation has proven to be effective in some cases, the effect of data augmentation on generalization ability for various deep learning model architectures is still not fully understood [15].

This research focuses on developing a waste classification model with a pre-trained deep learning approach and incorporating data augmentation techniques to improve generalization capabilities [16]. The main objective is to evaluate and understand the effect of augmentation techniques on various deep learning architectures, namely ResNet50, VGG-16, and MobileNetV2, in the face of new data that has never been seen by the model. This approach is expected to contribute to the verification of model generalization by using an external dataset that is completely different from the training data as a validity test.

2. METHOD

This research proposes the construction of a waste classification model using three architectures, namely ResNet50, VGG-16, and MobileNetV2. The research stage consists of several steps, starting with data collection (dataset), then dividing the data into three parts: training data, validation data, and testing data [17], [18]. The training data will go through an augmentation process to enrich the data variation. After that, the augmented data will be used to train the deep learning model, while the validation data will be directly used in the training process. Once the training is complete, the model will be evaluated to determine the best performance. Figure 1 provides a general illustration of the research stages in building a waste classification model.

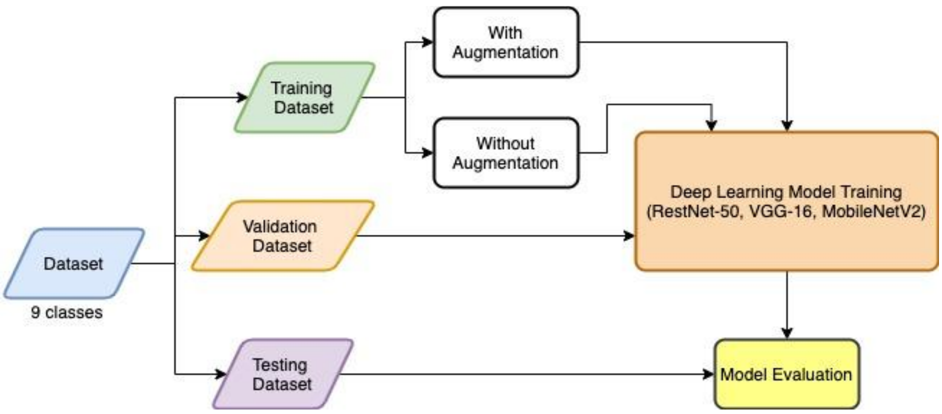


Figure 1. Research stages

2.1. Dataset

The dataset used in the research is an open dataset from the Kaggle website. RealWaste dataset is a collection of waste classification data grouped into 9 types where the data is collected from landfills [19]. Figure 2 shows a dataset sample, providing a visual representation of the various types of waste included. Each image has a resolution of 224×224 in order to distinguish object features more clearly and get better classification accuracy [20]. The data is divided into 3 parts, 80% for training data, 10% for validation data, and 10% as testing data [21]. The class labels and number of datasets are shown in Table 1.

Figure 2. Example dataset

Table 1. Labels and number of datasets

Label	Total data
Cardboard	461
Food organics	411
Glass	420
Metal	790
Miscellaneous trash	495
Paper	500
Plastic	921
Textile trash	318
Vegetation	436

2.2. Augmentation data

Augmentation methods are applied to the training dataset with the aim of getting better performance. The augmentation methods applied are flip, rotation, and zoom [22]. Table 2 shows the details of the parameters used. The determination of augmentation techniques is done randomly. Each augmentation results in different data variations for each sample, allowing the model to see features in various contexts [23].

Table 2. Augmentation parameters

Parameters	Value
Rescale	1./255
Rotation_range	30
Shear_range	0.1
Zoom_range	0.2
Horizontal_flip	True
Vertical_flip	True
Width_shift_range	0.2
Height_shift_range	0.2

In the code, the `ImageDataGenerator` object is used for image augmentation with various transformations. The parameter `rescale=1./255.` is used to normalize the image pixel values by dividing them by 255, so that they are in the range [0, 1]. `rotation_range=30` allows rotation of the image up to 30 degrees randomly. `shear_range=0.1` gives the image a shear effect of 10%. `zoom_range=0.2` allows zooming in and out of the image within a 20% range. `horizontal_flip=True` and `vertical_flip=True` enable horizontal and vertical flipping of the image. `width_shift_range=0.2` and `height_shift_range=0.2` enable horizontal and vertical shifting of the image up to 20% of the image width or height respectively. All these parameters are designed to enrich the variety of training data by randomly modifying the images, thus improving the generalization of the model. Figure 3 shows the result of the augmentation process applied to the dataset sample shown previously in Figure 2.

Figure 3. Data augmentation result

2.3. MobileNetV2

MobileNetV2 is a model proposed by Google in 2017 [24]. This architecture was released in early 2018 as a development of the previous model, MobileNetV1, which was proposed in 2015 [12], [25]. The development of this model is seen in the emergence of linear bottlenecks between layers and fast connections between bottlenecks [25]. Bottlenecks in MobileNetV2 help to package the model from low-level concepts to high-level descriptors so that the model can be trained faster and produce high accuracy even though the model uses fewer parameters [26]. Figure 4 illustrates the overall architecture of MobileNetV2, highlighting the use of inverted residual blocks with linear bottlenecks that contribute to its efficiency and performance in image classification tasks.

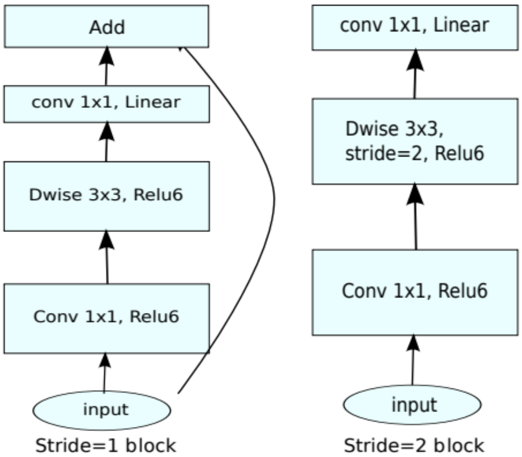


Figure 4. MobileNetV2 Architecture

2.4. Visual geometric group with 16 layers

VGG-16 is one of the CNN architectures that has 16 layers in the form of 13 convolutional layers and 3 fully-connected layers [27], [28]. The advantage of this architecture is that it has an architecture consisting of 3×3 convolutional layers and 2×2 pooling [29], [30]. which is considered more accurate when compared to previous CNN architectures. The accuracy of this model is taken from the state-of-the-art CNN architecture, namely the benchmark for the accuracy of this image is taken from the three models [31].

VGG-16 also has good generalization capabilities and is often used as a model base for various image recognition tasks. In addition, this architecture is known to be easier to optimize due to its simple yet effective design, although it requires considerable computational resources [10]. Figure 5 shows the architectural structure of VGG-16, illustrating its sequential and uniform layer configuration.

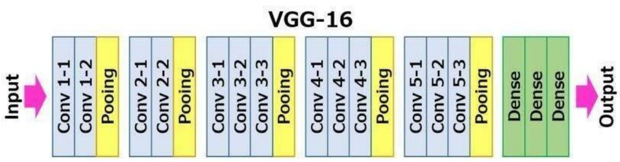


Figure 5. VGG-16 architecture

2.5. ResNet50

ResNets are deep convolutional networks whose basic idea is to bypass the block process of the convolutional layer by using shortcut connections. ResNet has a basic block called bottleneck block which has two rules in its simple design [32]. The first rule is that for the same output feature map size, each layer has the same number of filters. The second rule is that if the size of the output feature map is divided into two, the number of filters for each layer is doubled. Sampling is done directly by the convolution layer which has the step of converting each byteplot into an RGB image and batch normalization which is done right after each convolution and before the ReLU activation process. Converting each byteplot into an RGB image resizes it to 224×224 dimensions and subtracts the average RGB dataset data from each pixel [33]. Figure 6 illustrates the architecture of ResNet50, highlighting the shortcut connections and bottleneck design that make the network deeper and more efficient.

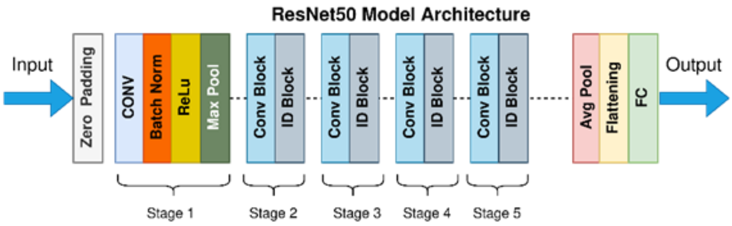


Figure 6. ResNet50 Architecture

3. RESULTS AND DISCUSSION

3.1. Training results

In model training, various important parameters are needed to support the training process optimally. The detailed configuration of these parameters is summarized in Table 3, which presents the values applied uniformly across all models used in this study. As shown in Table 4, ResNet50 model exhibits a train loss of 1.3397 and a validation loss of 1.4833, indicating that the model struggles to effectively learn from the data during both training and validation. This is further supported by its train accuracy of 0.5101 and validation accuracy of 0.4568, suggesting that the model is experiencing underfitting, where it fails to capture the underlying patterns in the training data, thus leading to poor generalization. In contrast, VGG-16, as presented in Table 4, demonstrates improved learning capability with a lower train loss of 0.2647 and validation loss of 0.7211. Its train accuracy of 0.9146 reflects a strong ability to recognize patterns in the training set. However, the gap between training and validation accuracy (validation accuracy of 0.7579) indicates a tendency toward overfitting, though the model still maintains a relatively good level of generalization. Among the three models in Table 4, MobileNetV2 achieves the best overall performance. It records a remarkably low train loss of 0.0015 and validation loss of 0.7561, alongside a perfect train accuracy of 1.0000. Notably, its validation accuracy reaches 0.8232, suggesting excellent generalization capabilities and an effective learning process without any significant overfitting. These results confirm that MobileNetV2 is the most efficient model in learning the data in the absence of augmentation.

Table 3. Hyperparameters used for all models

Parameters	Value
Input image	224×224
Optimizers	Adam
Learning rate	0.001
Epoch	100
Batch size	32
Activation function FC	ReLu
Function in the model	Avg

Table 4. Training results of model without augmentation

Deep learning model	Train loss	Valid loss	Train accuracy	Valid accuracy
ResNet50	1.3397	1.4833	0.5101	0.4568
VGG-16	0.2647	0.7211	0.9146	0.7579
MobileNetV2	0.0015	0.7561	1.0000	0.8232

Table 5 shows the results of testing models with data augmentation. It can be seen that none of the three models have improved performance compared to without augmentation. In ResNet50, the train loss of 1.6345 and valid loss of 1.6292 show that even though this model uses augmentation, its performance in understanding patterns from training data remains low. This is also reflected in the train accuracy which only reaches 0.3916 and valid accuracy of 0.4379. The ResNet50 model still has difficulty in generalizing patterns from training data to validation data, and its performance is still low. On VGG-16, data augmentation helped the model learn better. Train loss of 0.5155 and valid loss of 0.7485 show that the model is able to reduce errors in both training data and validation data. Train accuracy of 0.8167 and valid accuracy of 0.7411 show that the model is more stable in learning data with augmentation and is able to maintain good generalization, although there is a slight gap between the performance on training data and validation data. Meanwhile, MobileNetV2 again showed the best performance. With very low train loss (0.0972) and valid loss (0.6279), the model showed excellent learning ability from the training data and was able to maintain generalization on the validation data. Train accuracy of 0.9690 and valid accuracy of 0.8189 show that with augmentation there is a decrease in performance.

Table 5. Training results of model with augmentation

Deep learning model	Train loss	Valid loss	Train accuracy	Valid accuracy
ResNet50	1.6345	1.6292	0.3916	0.4379
VGG-16	0.5155	0.7485	0.8167	0.7411
MobileNetV2	0.0972	0.6279	0.9690	0.8189

3.2. Testing results

The test results using test data from several models, both those using augmentation and those without augmentation, will be used to select the best model among the three models evaluated. Table 6 presents a comparison of the overall test performance. A comparison was made to see the performance of three deep learning models, namely ResNet50, VGG-16, and MobileNetV2 with and without the use of data augmentation techniques. Data augmentation aims to improve the generalization of the model to the test data. In the ResNet50 model, the results showed that when augmentation was used, recall was 38.90%, precision 51.26%, F1-score 44.23%, and accuracy 41.60%. However, when augmentation was removed, there was an increase in recall to 46.04% and accuracy to 46.43%, although precision decreased slightly to 48.21%. This shows that ResNet50 gains a slight advantage without augmentation, but its overall performance remains low compared to other models. Meanwhile, the VGG-16 model shows a fairly stable performance both with and without augmentation. With the application of augmentation, the accuracy value increased by 1.89%. This shows that augmentation helps in maintaining performance stability, although the difference is not very significant. The MobileNetV2 model gave the best results among the three. With augmentation, recall reached 81.89%, precision 80.44%, F1-score 81.16%, and accuracy 81.72%. Without augmentation, the performance actually improved, with recall 84.35%, precision 85.22%, F1-score 84.78%, and accuracy 84.45%. This shows that MobileNetV2 has an excellent ability to generalize data, even without augmentation. Overall, augmentation has a varying impact on each model, but MobileNetV2 still excels in waste type classification, both with and without augmentation. The comparison of the test accuracy between the three models is shown in Figure 7.

In many cases, data augmentation helps deep learning models avoid overfitting on training data and improves their performance on unseen test data. By introducing controlled variation, augmentation enhances the model's ability to generalize to new inputs. However, there are situations where augmentation may reduce accuracy. This typically occurs when the variations introduced do not align with the natural characteristics of the original data. For instance, in some classification tasks, transformations such as 90-degree rotations or flips may generate images that are irrelevant or unnatural for the class being recognized. As a result, the model may learn from misleading patterns, leading to incorrect predictions. Moreover, if a model already demonstrates strong pattern recognition on the original dataset, additional variation can introduce unnecessary noise, reducing its ability to focus on key features. This phenomenon was observed in MobileNetV2, where accuracy improved when augmentation was not applied. Research comparing three models—ResNet50, VGG-16, and MobileNetV2—further emphasizes that the effect of augmentation depends on model architecture. ResNet50 performed poorly overall, with or without augmentation, while VGG-16 showed stable but moderate results. MobileNetV2 consistently outperformed both, even without augmentation, proving to be the most effective model for waste type classification.

Table 6. Comparison of best model results

Pre-trained model	Using augmentation				No augmentation			
	Recall	Precision	F1-score	Accuracy	Recall	Precision	F1-score	Accuracy
ResNet50	38.90	51.26	44.23	41.60	46.04	48.21	47.10	46.43
VGG-16	73.75	78.20	75.91	75.84	73.65	76.85	75.21	73.95
MobileNetV2	81.89	80.44	81.16	81.72	84.35	85.22	84.78	84.45

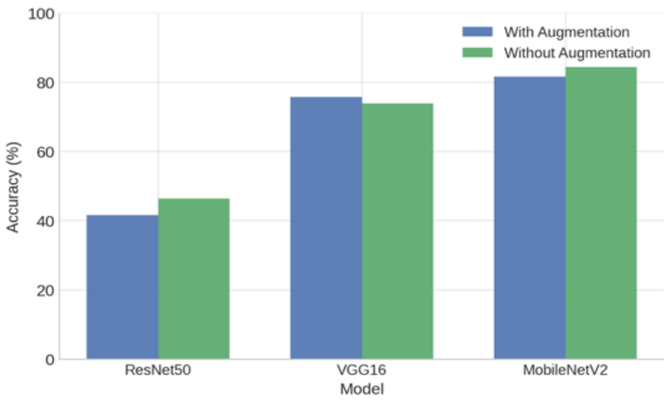
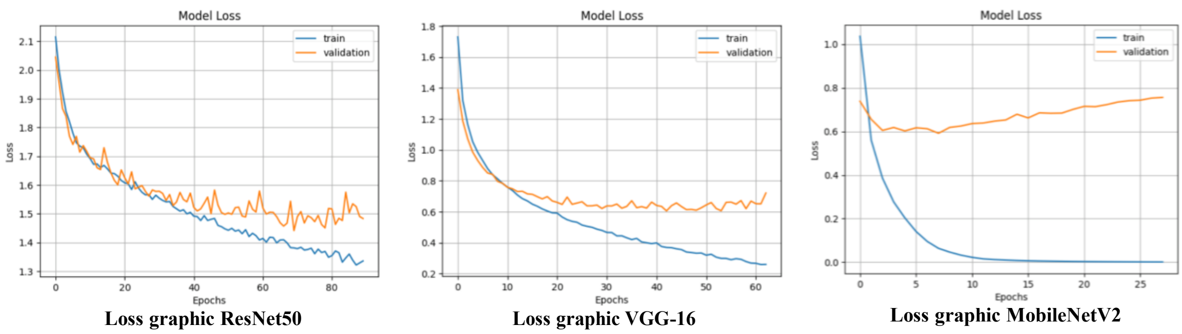


Figure 7. Comparison of test accuracy among ResNet50, VGG-16, and MobileNetV2 with and without data augmentation

3.3. Best model

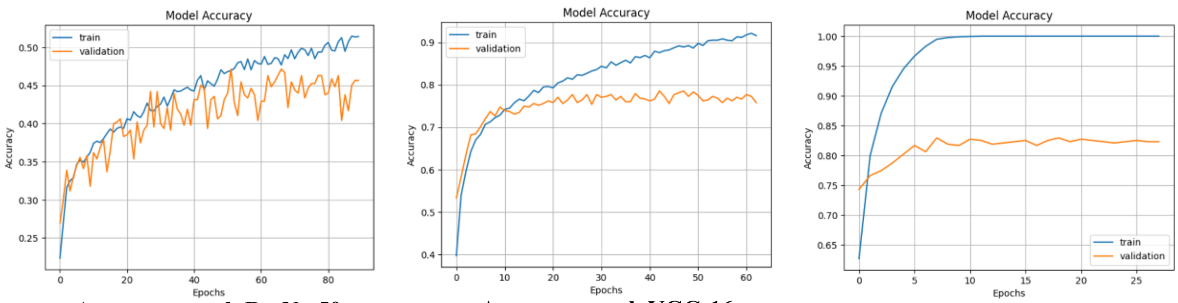
The model loss graph depicts the loss progression during the training process for both training and validation data. Figure 8 illustrates this loss behavior across the three models tested, providing a visual comparison of how each model responds over the course of training. From the three graphs shown, it can be seen that ResNet50 (Figure 8(a)) and VGG-16 (Figure 8(b)) models show a less significant decrease in train loss. In terms of validation loss. In contrast, MobileNetV2 model (Figure 8(c)) shows a drastic decrease in train loss, almost approaching zero as the number of epochs increases, indicating that it is able to learn the patterns in the training data well, the MobileNetV2 model decreased at the beginning, but then stabilized and started to increase after about five epochs. ResNet50 model showed a significant increase in validation loss at the 10th epoch, while VGG-16 experienced an increase at the 20th epoch, where both still showed instability. This increase in validation loss is an indication of overfitting, where the model fits the training data too well but does not generalize well to the validation data. This shows that while MobileNetV2 model performs very well on the training data, it cannot perform optimally on the validation data. The model memorizes the training data rather than capturing more general patterns, making it less effective when faced with new data. In terms of epochs, it also shows that the MobileNetV2 model converges faster, requiring only 25 epochs to reach stability. This is due to MobileNetV2's more lightweight and parameter-efficient architecture, and is designed to perform optimally in resource-constrained environments.



Loss graphic ResNet50 (a) **Loss graphic VGG-16** (b) **Loss graphic MobileNetV2** (c)

Figure 8. Loss graphic for (a) ResNet50, (b) VGG-16, and (c) MobileNetV2

The graph in Figure 9 displays the accuracy graphs for the three models. The ResNet50 model also showed varying fluctuations from the start of training until it reached 50% accuracy (Figure 9(a)), but the validation accuracy showed lower values. VGG-16 showed a steadier increase in accuracy up to a lift of 90% (Figure 9(b)). The validation accuracy also improved more significantly than ResNet50, indicating better generalization ability. MobileNetV2 quickly achieves high accuracy on the training data, approaching 100% within about the first 7 epochs (Figure 9(c)). However, the accuracy on the validation data only reached about 82%, with small fluctuations throughout the training process. The discrepancy between the very high training accuracy and the lower validation accuracy indicates that the model may suffer from overfitting, where the model memorizes its training data too much and does not generalize well to new data.



Accuracy graph ResNet50 **Accuracy graph VGG-16** **Accuracy graph MobileNetV2**

(a) (b) (c)

Figure 9. Accuracy graph of training and validation results across of (a) ResNet50, (b) VGG-16, and (c) MobileNetV2

The confusion matrix illustrates the performance of the MobileNetV2 model in classifying 9 categories of waste, including cardboard, food organics, glass, metal, miscellaneous trash, paper, plastic, textile trash, and vegetation. From the matrix in Figure 10, it can be seen that the model has a fairly high accuracy in some classes, such as metal with 77 correct predictions out of a total of 85, plastic with 80 correct predictions out of 89, and vegetation with 48 correct predictions out of 50. However, some classes experience a fairly high level of confusion, especially in miscellaneous trash and food organics, where the model often misclassifies them as other classes. This shows that while the model works well on some categories, there is still room for improvement on classes that are more ambiguous or difficult to recognize.

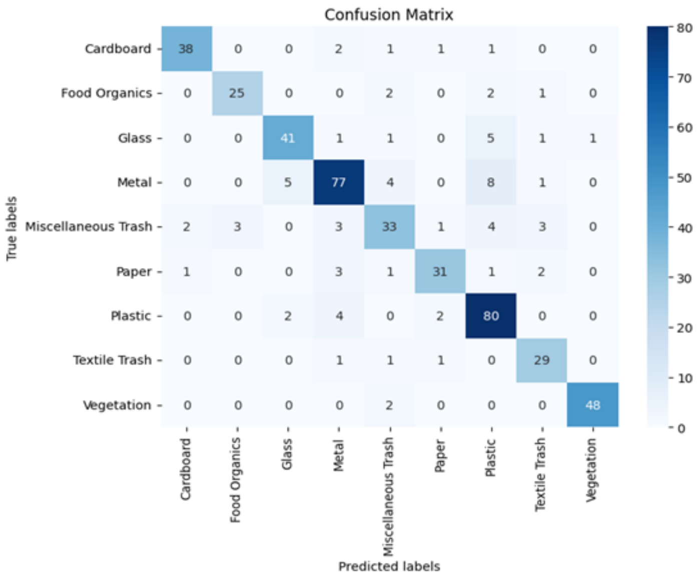


Figure 10. MobileNetV2 confusion matrix

4. CONCLUSION

Based on the results, MobileNetV2 showed the best performance in waste type classification compared to ResNet50 and VGG-16. With data augmentation techniques, MobileNetV2 achieved an accuracy of 81.72%, while without augmentation the accuracy increased to 84.45%, showing that the model is highly efficient even without excessive augmentation. VGG-16 also shows stable performance with an accuracy of 75.84% when using augmentation and 73.95% without augmentation, indicating that augmentation provides a slight, but not significant improvement. In contrast, ResNet50 had the lowest performance with an accuracy of only 41.60% using augmentation and slightly improved to 46.43% without augmentation, indicating that this model is less than optimal for litter classification on the dataset used. Overall, data augmentation has a different effect on each model, with MobileNetV2 proving to be the most efficient and optimal model in this task, both with and without augmentation.

ACKNOWLEDGEMENTS

The authors would like to thank the Ministry of Education, Culture, Research and Technology of the Republic of Indonesia (Kemendikbudristek) for the full support given in the implementation of this research with number 110/E5/ PG.02.00.PL/2024. Assistance in the form of funding through the fundamental research grant program has contributed significantly to the achievement of the results of this research.

FUNDING INFORMATION

This research received funding from the Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia (Kemendikbudristek) under grant number 110/E5/PG.02.00.PL/2024.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
I Komang Arya Ganda Wiguna	✓	✓	✓	✓	✓	✓	✓		✓	✓		✓		✓
I Gusti Made Ngurah Desnanjaya	✓		✓			✓	✓	✓	✓	✓	✓		✓	
I Kadek Budi Sandika		✓		✓	✓			✓		✓	✓		✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors declare no conflict of interest.

INFORMED CONSENT

We have obtained informed consent from all individuals included in this study.

ETHICAL APPROVAL

This study does not involve human participants or animals, therefore ethical approval is not required.

DATA AVAILABILITY

The secondary data used in this study are publicly available on Kaggle at https://www.kaggle.com/datasets/joebeachcapital/realwaste/data.

REFERENCES

[1] H. Kaur and P. Kaur, "Factors determining household waste segregation behaviour: an Indian case study," *International Journal of Experimental Research and Review*, vol. 41, pp. 83–95, 2024, doi: 10.52756/ijerr.2024.v41spl.007.

[2] A. Siddiqua, J. N. Hahladakis, and W. A. K. A. Al-Attiya, "An overview of the environmental pollution and health effects associated with waste landfilling and open dumping," *Environmental Science and Pollution Research*, vol. 29, no. 39, pp. 58514–58536, 2022, doi: 10.1007/s11356-022-21578-z.

[3] A. Pramudianto, "The role of international law and national law in handling marine plastic litter," *Lampung Journal of International Law*, vol. 1, no. 2, pp. 43–54, 2020, doi: 10.25041/lajil.v1i2.2024.

[4] P. Wiriwithya, S. Rungnarongruck, S. Pongamphai, S. Puapattanakul, and R. Chancharoen, "Smart trash classification machine," in *2023 9th International Conference on Mechatronics and Robotics Engineering*, 2023, pp. 146–150, doi: 10.1109/ICMRE56789.2023.10106603.

[5] V. T. Widyaningrum, A. S. Romadhon, and R. Safitri, "Automatic waste sorter machine using proximity sensor," in *Proceedings of the International Conference on Health Informatics, Medical, Biological Engineering, and Pharmaceutical*, 2020, pp. 264–270, doi: 10.5220/0010331102640270.

[6] A. G. Kurbis, B. Laschowski, and A. Mihailidis, "Stair recognition for robotic exoskeleton control using computer vision and deep learning," in *2022 International Conference on Rehabilitation Robotics (ICORR)*, 2022, pp. 1–6, doi: 10.1109/ICORR55369.2022.9896501.

[7] H. Abdu and M. H. M. Noor, "A survey on waste detection and classification using deep learning," *IEEE Access*, vol. 10, pp. 128151–128165, 2022, doi: 10.1109/ACCESS.2022.3226682.

[8] N. Buduma, N. Buduma, and J. Papa, *Fundamentals of deep learning*. Sebastopol, United States: O'Reilly Media, Inc, 2022.

[9] M. I. B. Ahmed *et al.*, "Deep learning approach to recyclable products classification: towards sustainable waste management," *Sustainability*, vol. 15, no. 14, 2023, doi: 10.3390/su151411138.

[10] R. Puspita and C. Rahayu, "Pneumonia prediction on chest x-ray images using deep learning approach," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 1, pp. 467–474, 2024, doi: 10.11591/ijai.v13.i1.pp467-474.

[11] S. R. Shah, S. Qadri, H. Bibi, S. M. W. Shah, M. I. Sharif, and F. Marinello, "Comparing inception V3, VGG 16, VGG 19, CNN, and ResNet 50: a case study on early detection of a rice disease," *Agronomy*, vol. 13, no. 6, 2023, doi: 10.3390/agronomy13061633.

[12] L. Yong, L. Ma, D. Sun, and L. Du, "Application of MobileNetV2 to waste classification," *PLoS ONE*, vol. 18, pp. 1–16, 2023, doi: 10.1371/journal.pone.0282336.

[13] M. Talaat, X. Si, and J. Xi, "Multi-level training and testing of CNN models in diagnosing multi-center COVID-19 and pneumonia X-ray images," *Applied Sciences*, vol. 13, no. 18, 2023, doi: 10.3390/app131810270.

[14] Z. Chen, Z. Ding, X. Zhang, X. Zhang, and T. Qin, "Improving out-of-distribution generalization in SAR image scene classification with limited training samples," *Remote Sensing*, vol. 15, no. 24, 2023, doi: 10.3390/rs15245761.

[15] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, "Image data augmentation for deep learning: a survey," *arXiv:2204.08610*, 2022.

[16] M. Satvilkar and P. N. Cosgrave, "Image based trash classification using machine learning algorithms for recyclability status," *M.Sc. Research Project*, School of Computing, National College of Ireland, Dublin, Ireland, 2018.

[17] C. Albon, *Machine learning with Python cookbook : practical solutions from preprocessing to deep learning*. Sebastopol, United States: O'Reilly Media, 2018.

[18] S. A. Kamran *et al.*, "SANS-CNN: an automated machine learning technique for spaceflight associated neuro-ocular syndrome with astronaut imaging data," *npj Microgravity*, vol. 10, no. 1, pp. 1–7, 2024, doi: 10.1038/s41526-024-00364-w.

[19] S. Single, S. Iranmanesh, and R. Raad, "RealWaste: a novel real-life data set for landfill waste classification using deep learning," *Information*, vol. 14, no. 12, 2023, doi: 10.3390/info14120633.

[20] M. P. Salas and P. L. D. Geus, "Deep learning applied to imbalanced malware datasets classification," *Journal of Internet Services and Applications*, vol. 15, no. 1, pp. 342–359, 2024, doi: 10.5753/jisa.2024.3907.

[21] M. Harahap, V. Damar, S. Yek, M. Michael, and M. R. Putra, "Static and dynamic human activity recognition with VGG-16 pre-trained CNN model," *Jurnal Infotel*, vol. 15, no. 2, pp. 44–48, 2023, doi: 10.20895/infotel.v15i2.916.

[22] E. Ayan and H. M. Ünver, "Data augmentation importance for classification of skin lesions via deep learning," in *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting*, 2018, pp. 1–4, doi: 10.1109/EBBT.2018.8391469.

[23] E. Tasci, C. Uluturk, and A. Ugur, "A voting-based ensemble deep learning method focusing on image augmentation and preprocessing variations for tuberculosis detection," *Neural Computing and Applications*, vol. 33, no. 22, pp. 15541–15555, 2021, doi: 10.1007/s00521-021-06177-2.

[24] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[25] M. Akay *et al.*, "Deep learning classification of systemic sclerosis skin using the MobileNetV2 model," *IEEE Open Journal of Engineering in Medicine and Biology*, vol. 2, pp. 104–110, 2021, doi: 10.1109/OJEMB.2021.3066097.

[26] K. Dong, C. Zhou, Y. Ruan, and Y. Li, "MobileNetV2 model for image classification," in *2020 2nd International Conference on Information Technology and Computer Application*, Dec. 2020, pp. 476–480, doi: 10.1109/ITCA52113.2020.00106.

[27] A. V. Samsonovich, "Universal learner as an embryo of computational consciousness," in *AAAI Fall Symposium - Technical Report*, 2007, pp. 129–134.

[28] S. A. Althubiti, F. Alenezi, S. Shitharth, S. K., and C. V. S. Reddy, "Circuit manufacturing defect detection using VGG16 convolutional neural networks," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, 2022, doi: 10.1155/2022/1070405.

[29] O. N. Belaid and M. Loudini, "Classification of brain tumor by combination of pre-trained VGG16 CNN," *Journal of Information Technology Management*, vol. 12, no. 2, pp. 13–25, 2020, doi: 10.22059/JITM.2020.75788.

[30] D. Theckedath and R. R. Sedamkar, "Detecting affect states using VGG16, ResNet50 and SE-ResNet50 networks," *SN Computer Science*, vol. 1, no. 2, pp. 1–7, 2020, doi: 10.1007/s42979-020-0114-9.

[31] P. Bhat *et al.*, "Brain tumor detection using CNN," in *Communications in Computer and Information Science*, 2024, pp. 18–28, doi: 10.1007/978-3-031-70001-9_2.

[32] N. Behar and M. Shrivastava, "ResNet50-based effective model for breast cancer classification using histopathology images," *CMES-Computer Modeling in Engineering & Sciences*, vol. 130, no. 2, 2022.

[33] E. Sitompul, V. L. Setiawan, H. J. Tarigan, and M. Galina, "Image classification of fabric defects using ResNet50 deep transfer learning in FastAI," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 5, pp. 3255–3267, 2024.

BIOGRAPHIES OF AUTHORS

I Komang Arya Ganda Wiguna received a Bachelor's degree in Informatics Engineering from Udayana University in 2014 and a Master's degree in Department of Computer Science from Universitas Gadjah Mada in 2017. Currently a lecturer at Department of Informatics, Faculty of Mathematics and Natural Sciences, Udayana University, Bali, Indonesia. His research interest includes information system, machine learning, deep learning, and decision support system. He can be contacted at email: kmaryagw@gmail.com.

I Gusti Made Ngurah Desnanjaya is a lecturer and innovation researcher at the Institute of Business and Technology Indonesia (INSTIKI). He has been a lecturer since 2015 and is actively involved in academic innovation at INSTIKI. He obtained his Bachelor's degree in 2011, Master's degree in 2013, and Professional Engineer degree in 2021 from Udayana University, Bali. He is also actively researching and reviewing several journals. His research interests include embedded systems, AI, control systems, the internet of things, and renewable energy. He can be contacted at email: ngurah.desnanjaya@gmail.com.

I Kadek Budi Sandika received his bachelor’s degree in civil engineering from Udayana University in 2008, master’s degree in educational administration from Universitas Pendidikan Ganesha in 2012, and doctoral degree in technology and vocational education from Universitas Negeri Yogyakarta in 2016. He is currently working as a lecturer at Department of Informatics, Faculty of Technology and Informatics, Institute of Business and Technology Indonesia. His area of research includes foundational literacies development, waste management, IT governance and auditing, and partnership in vocational education. He can be contacted at email: ikbsandika@instiki.ac.id.