Jurnal Teknik Informatika (JUTIF) Vol. 6, No. 3, Juni 2025, Page. 1147-1158
P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.3.4417

An Efficient Model for Waste Image Classification Using EfficientNet-B0

Teofilus Kurniawan[1], Khadijah*[2], Retno Kusumaningrum[3]

[1,2,3]Informatics, Universitas Diponegoro, Indonesia

Email: [2]khadijah@live.undip.ac.id

Received : Feb 6, 2025; Revised : Apr 24, 2025; Accepted : May 5, 2025; Published : Jun 10, 2025

Abstract

Waste management remains a significant challenge, particularly in developing countries. To address this issue, artificial intelligence can be leveraged to develop a waste image classifier that facilitates automatic waste sorting. Previous studies have explored the use of Convolutional Neural Networks (CNNs) for waste image classification. However, CNNs typically require a large number of parameters, leading to increased computational time. For practical applications, a waste image classifier must not only achieve high accuracy but also operate efficiently. Therefore, this study aims to develop an accurate and computationally efficient waste image classification model using EfficientNet-B0. EfficientNet-B0 is a CNN architecture designed to achieve high accuracy while maintaining an efficient number of parameters. This study utilized the publicly available TrashNet dataset and investigated the impact of image augmentation in addressing imbalance data issues. The highest performance was achieved by the model trained on the unbalanced dataset with the addition of a Dense(32) layer, a dropout rate of 0.3, and a learning rate of 1e-4. This configuration achieved an accuracy of 0.885 and an F1-score of 0.87. These results indicate that the inclusion of a Dense(32) layer prior to the output layer consistently improves model performance, whereas image augmentation does not yield a significant enhancement. Furthermore, our proposed model achieved the highest accuracy while maintaining a significantly lower number of parameters compared to other CNN architectures with comparable accuracy, such as ResNet-50 and Xception. The resulting waste classification model can then be further implemented to build an automatic waste sorter.

Keywords : *Classification, CNN, Efficientnet, Waste Image, Waste Sorting.*

1. INTRODUCTION

Waste management remains a significant challenge, particularly in developing countries. Population growth and industrial sector expansion contribute to the increasing volume of waste generation. This trend aligns with shifts in consumption patterns and lifestyles [1]. In 2020, the generated municipal solid waste globally exceeded two billion tonnes, and this figure is projected to rise to 3.8 billion tonnes by 2025. A proper and integrated waste management strategy is strongly needed to minimize the risk of uncontrolled waste to the planetary crisis, including climate changes, biodiversity loss, and pollution [2].

Municipal waste management consists of upstream actions to minimize waste generation and downstream actions to mitigate its environmental impact. Among downstream strategies, waste recycling plays a crucial role in increasing waste value as a resource, preventing environmental leakage, and reducing pollution. However, recycling rates vary significantly across countries and regions, with high-income nations achieving rates above 50%, while some regions recycle less than 5%. Waste sorting is essential to support recycling efforts. This process aims to categorize and separate waste based on type, quantity, and characteristics [2].

The development of artificial intelligence technology, especially machine learning, can be utilized to build a model that is capable of sorting waste automatically by using classification algorithm [3].

Vol. 6, No. 3, Juni 2025, Page. 1147-1158
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.3.4417

Several studies related to waste image classification have been carried out. Generally, there are two approaches to perform image classification. The first approach applies manual features extraction followed by conventional machine learning classification algorithm. Khadijah et al. [4] performed solid waste image classification by using combined features from color, shape, and texture of images followed by SVM as classification algorithm. To prepare the image for feature extraction, PSPNet segmentation was applied. This research reached the best classification accuracy of 76.49%.

The second approach applies deep learning method which do not need to extract features manually. Convolutional Neural Network (CNN) is recognized as a popular deep learning method and is proven to provide good performance, especially for image classification in various domains [5], [6], [7]. CNN is able to overcome the difficulties in manual feature extraction. CNN carries out automatic feature extraction as well as the prediction process. Automatic feature extraction in CNN is performed by the convolution layer at the beginning of the network before fully-connected layers which is responsible for image label prediction [8].

Several previous studies have also implemented CNN for waste image classification. Sidharth et al. [9] applied CNN with six layers and was able to achieve the best accuracy of 76.19. Altikat et al. [10] achieved an accuracy of 70% by employing a five-layer convolutional neural network (CNN) to classify four categories of waste. Faria et al. [11] also compared several types of CNN architectures to classify four categories of waste images. Their study demonstrated that the VGG-16 architecture achieved the highest accuracy of 88.42%, compared to VGG-19, Inception-V3, and ResNet-50. Gaurav et al. [12] also employed the VGG-16 architecture as a feature extractor, followed by a Random Forest classifier, to classify six categories of garbage images, achieving an accuracy of 85%. Similarly, Gyawali et al. [13] carried out classification of trash images by comparing several CNN architectures. The results show that the ResNet-18 was able to achieve the best accuracy of 87.8% compared to other architectures. Adedeji & Wang [14] achieved the best accuracy of 87% in Trashnet dataset classification by utilizing pretrained ResNet-50 as a feature extractor and the SVM as a classifier. Rismiyati et al. [15] conducted garbage classification using the TrashNet dataset and demonstrated that the Xception architecture outperformed both VGG-16 and ResNet-50. Another study related to waste classification using the TrashNet dataset was conducted by Fan et al. [16], who employed a deep learning model known as LMNet (Lightweight Multiscale CNN). The proposed LMNet achieved the highest accuracy of 85.45%, outperforming AlexNet, VGG-11, MobileNet, and LeNet-5, respectively.

The implementation of deep learning method for waste image classification in the previous studies is able to enhance the accuracy of the resulting models compared to conventional machine learning method. However deep learning suffers from the large number of model parameters. For instance, the VGG-16 and ResNet-50 models involve a large number of parameters, exceeding 138 million and 25 million, respectively, which leads to increased computational time. In practice, a waste classification model designed for general public use must demonstrate not only high accuracy but also ability to operate with low computational time.

Tan & Le proposed a specialized type of CNN, namely EfficientNet, which is not only intended to achieve good performance in prediction, but also an efficient model. An efficient model is characterized by a reduced number of network parameters, leading to lower computational time. This efficiency is achieved by uniformly scaling the depth, width, and resolution within the EfficientNet architecture. These three dimensions are related to each other; for instance, when the resolution size increases, a wider filter size can be used for convolution operations. In contrast, other CNN architectures scale these dimensions independently, often resulting in less efficient architectures and suboptimal model performance [17].

Baik et al. confirmed that EfficientNet outperforms other architectures, such as Inception-ResNet-V2, in image classification tasks [18]. EfficientNet-B0 also achieves high accuracy with fewer

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.3.4417

parameters compared to ResNet-50 and Inception-V3 in masked face image classification [19]. The implementation of EfficientNet for leaf plant diseases is also able to achieve higher accuracy than ResNet and AlexNet and competitive accuracy with VGG16 and Inception-V3 [20]. In medical image classification, EfficientNet outperformed other benchmark CNN architectures, including ResNet50 and MobileNet [21]. Chung et al. demonstrated that the EfficientNet-B5 model significantly outperformed ResNet152 and VGG19 in classifying knee MRI images [22]. The previous studies highlight the success of EfficientNet in achieving better performance compared to other CNN architectures.

In addition to the classification algorithm, the used training data affect the performance of the classification model. When the model is trained using imbalance dataset, usually the performance of model for predicting data from the minority class is also poor [23]. However, Batista argued that the performance of model was not systematically affected by the problem of imbalance datasets [24]. Therefore, this research performed the experiment using the original unbalanced dataset and augmented balanced dataset. To deal with the imbalanced data problem, image augmentation was utilized since the dataset is in the form of image. Image augmentation is cost-effective strategies to enrich the image dataset which allows the learning algorithm to improve its generalization performance [25].

This study aims to develop an accurate and computationally efficient waste image classification model by utilizing CNN algorithm with the EfficientNet-B0 architecture. The EfficientNet-B0 version was chosen because it has the lowest input resolution and the fewest parameters, making it particularly suitable for predicting waste images where fast computational time is required during inference. In addition, this research also tries to address the imbalance data problem by utilizing image augmentation. Therefore, this research compared the model training by using original imbalanced dataset and the augmented balanced dataset. It is expected that the resulting classification model can then be implemented to support the waste sorting process.

2. METHOD

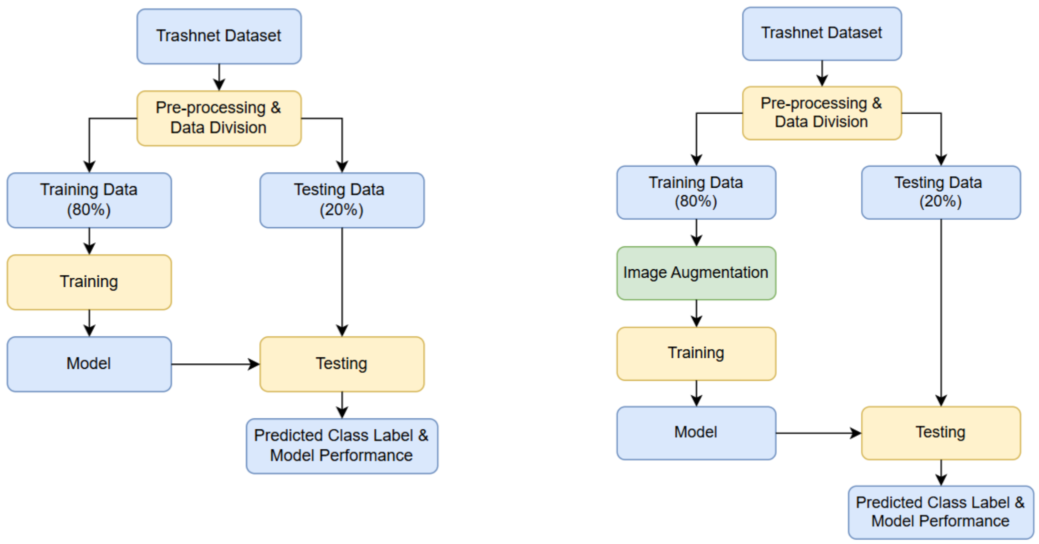
This research utilized a public dataset, namely Trashnet, which contains 2,527 of waste images with dimensions of 512 x 384. The image dataset consists of six classes or trash categories, namely cardboard (403 images), glass (501 images), paper (594 images), plastic (482 images), metal (410 images), and trash (137 images) [26]. Figure 1 shows a sample of image in each category.

Cardboard Glass Metal Paper Plastic Trash

Figure 1. A Sample Of Image In Each Category

The development of waste image classifier in this research involved several stages as shown in Figure 2. First, the images form Trashnet dataset was pre-processed and divided into training and testing data. The number of images in the Trashnet dataset varies for each category. Therefore, this research carried out two main scenarios. In the first scenario, model training was carried out using original trash images from Trashnet with the unbalanced data distribution as shown in Figure 2(a), while in the second scenario, data augmentation was added on the training data which aims to balance the amount of data in each category (balanced data) as shown in Figure 2(b). Subsequently, the training data was employed to train the model by using EfficientNet-B0 algorithm. The trained model was tested by using the testing data. The testing process output a predicted class label of input image which is compared to the actual class using certain metrics to obtain a measure of model performance.

Vol. 6, No. 3, Juni 2025, Page. 1147-1158
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.3.4417



(a) (b)

Figure 2. Stages In The Waste Image Classification (A) Unbalanced Data Scenario; (B) Balanced Data Scenario

2.1. Data Pre-processing and Data Division

Each image of Trashnet dataset was resized according to the size required by EfficientNet-B0 architecture, as implemented in this study. The input image size on the EfficientNet-B0 architecture is 224 x 224 pixel. Therefore, each image of Trashnet dataset was resized to 224 x 224 pixel. Subsequently, the data was divided into training data and testing data with a ratio of 80%:20%. Data division was carried out randomly while maintaining the original ratio of data in each class. The training data was used to train the model, while the testing data was used to evaluate the resulting trained model [27].

2.2. Image Augmentation

Image augmentation aims to increase the number of images by creating new images from the original images through several digital image processing techniques. In this research, image augmentation was utilized to equalize or balance the number of trash image samples in each category. This image augmentation is only applied in the second scenario (balanced data). Image augmentation is applied only to training data so that during the training process the model gets an equivalent learning experience in each class [28]. Image augmentation carried out in this research including: 1) random rotation with a maximum degree of 20 degrees; 2) random width shift with a maximum stretch to the right and left 20% of the image size; 3) random height shift with maximum range up and down by 20% of the image size; 4) random shear with the maximum perception angle shifted by 20% of image size; 5) random zoom with a maximum image enlargement of 20% of the image; 6) horizontal flip.

2.3. Model Training with EfficientNet-B0

The model for classifying waste images in this research was built using the CNN algorithm with the EfficientNet-B0 architecture. CNN organizes input data in the form of a three-dimensional tensor that describes width, height, and depth. The main contribution of CNN lies in its ability to perform feature extraction automatically from the input tensor. In general, CNN is composed of several types of layers, namely convolutional layers, pooling layers, and fully-connected layers [8]. The convolutional layer is responsible for extracting important feature map from an input image. The convolution operation uses a filter or kernel matrix for feature extraction. Subsequently, the resulting feature map is passed through a pooling layer, which reduces its dimensions while preserving important information [29]. The output from the pooling operation is flattened and forwarded to the fully-connected layer. The last fully-connected layer is responsible for predicting the class label [30].

CNN architectures generally vary in depth (number of filters), width (size of kernel), and resolution (size of input). These three parameters influence each other, for example, when the resolution size increases, a wider filter size can be used for convolution operations. Tan & Le [17] proposed an EfficientNet architecture that performs uniform scaling of depth (d), width (w), and resolution (r). This uniformity can be achieved by scaling each of these dimensions with compound scaling coefficient ϕ so that $d = \alpha^{\phi}$, $w = \beta^{\phi}$, and $r = \gamma^{\phi}$ such that $\alpha.\beta^2.\gamma^2 \approx 2$ and $\alpha \geq 1, \beta \geq 1$. The parameters α, β, γ determine how to allocate the magnitude scaling of the depth (d), width (w), and resolution (r) network, respectively. The resulting architecture has been proven capable of producing an efficient number of network parameters with high accuracy. The most basic version of EfficientNet is EfficientNet-B0. The EfficientNet-B0 architecture receives the input of 224x224 pixels and consists of 9 blocks as shown in Table 1. This network uses kernel size of 3x3 or 5x5 in most of its layers, except for the last layers which uses kernel size of 1x1. The number of channels typically increases in deeper layers, resulting in a proportional increase in the depth of the output size. The final feature map generated by the network has dimensions of 7×7x1280 [17].

Table 1. EfficientNet-B0 Architecture [17]

Block	Operator	Kernel Size	#Channels	#Layers	Output Size
1	Conv	3x3	32	1	224 x 224 x 32
2	MBConv1	3x3	16	1	112 x 112 x 16
3	MBConv6	3x3	24	2	112 x 112 x 24
4	MBConv6	5x5	40	2	56 x56 x 40
5	MBConv6	3x3	80	3	28 x 28 x 80
6	MBConv6	5x5	112	3	14 x 14 x 112
7	MBConv6	5x5	192	4	14 x 14 x 192
8	MBConv6	3x3	320	1	7 x 7 x 320
9	Conv & Pooling & FC	1x1	1280	1	7 x 7 x 1280

This research utilized EfficientNet-B0 to build a waste image classification model as shown in Figure 3. The model was developed using the pre-trained EfficientNet-B0 on ImageNet dataset (include top = false) by adding a flatten layer, a fully-connected layer consisting of 32 units (Dense32), a dropout layer, and an output layer consisting of 6 units (Dense6) as the number of classes in the Trashnet dataset. Several research experiments in particular scenarios ignored the use of the Dense32 layer. The input image size to EfficientNet-B0 is 224x224x3, then the final output from the feature extraction is 7x7x1280. The output is then flattened so that it becomes a vector of size 62,720 and forwarded to the Dense32 layer so that the output becomes a vector of size 32. Dropout layer was added in the training process to avoid overfitting by pruning a number of units in the previous layer randomly. Applying a dropout layer does not change the output dimensions. Finally, the output was passed to the output layer which provides the final output in the form of a vector of size 6. The output layer applied the softmax function, so that the value for each unit in the output layer represents the probability value of the input image being classified into each class represented by each unit of output layer.

All experiments in this research were run on Google Colaboratory. The implementation of method was written in Python programming language version 3.9.16. Tensorflow 2.9.2 was also utilized to construct the deep learning model based on EfficientNet architecture.

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871

Vol. 6, No. 3, Juni 2025, Page. 1147-1158
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.3.4417

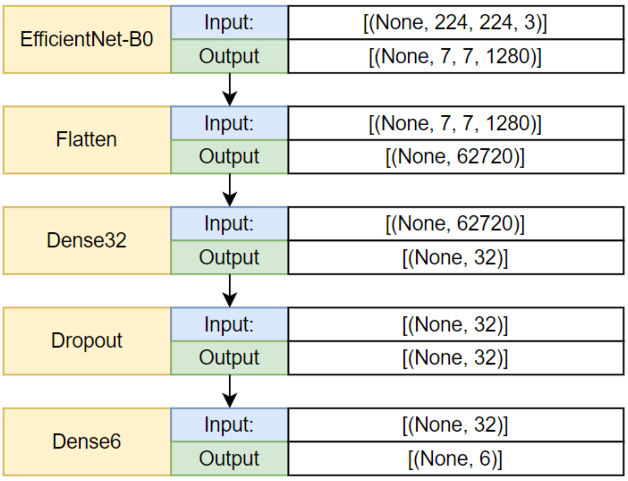


Figure 3. Network Architecture For Waste Image Classification Using Efficientnet-B0

2.4. Model Testing

This stage aims to evaluate performance of the resulting model from the training process by using testing data. To evaluate model performance, a confusion matrix was utilized. Confusion matrix is a matrix that can explain the performance of the classification model in each class. The matrix has two dimensions. The first dimension represents the predicted class label generated by the trained model, while the other dimension represents the actual class. From the confusion matrix, various metrics can then be calculated such as accuracy, precision, recall, and F1-score. In this study, the metrics reported are accuracy and F1-score [31].

3. RESULT

3.1. Data Pre-processing and Data Division

After the pre-processing, each image in the Trashnet dataset was resized to 224x224. Subsequently, the dataset was divided into 80% training data and 20% testing data. Therefore, the ratio of training and testing samples in each category are carboard (332:71), glass (400:101), paper (475:119), plastic (385:97), metal (328:82), and trash (109:28).

3.2. Image Augmentation

Image augmentation was carried out only in training data. The paper class contained the largest number of training samples (475 images). Consequently, the number of training samples in other classes was augmented to 475 images to achieve a balanced training set across all classes.

3.3. Model Training

To identify the best model for classifying trash images, several experimental scenarios were designed, as summarized in Table 2. These scenarios were structured based on the dataset type, distinguishing between the use of the original (imbalanced) dataset and an augmented (balanced) version. For each dataset type, the network architecture was further refined by evaluating the inclusion or exclusion of an additional Dense(32) layer after the Flatten layer. Furthermore, in each scenario, training was also carried out by tuning some combination of hyperparameter values, namely dropout and learning rate. Therefore, each scenario consists of 16 experiments from the combination of dropout and learning rate values. In each scenario, training process was carried out using the Adam optimization algorithm and a batch size of 16 over a maximum of 50 epochs with early stopping of patience 10. If there is no increase in the accuracy value for 10 epochs, the training will be stopped and the best model during training will be saved to be tested against the test data.

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871

Vol. 6, No. 3, Juni 2025, Page. 1147-1158
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.3.4417

Table 2. Testing Scenarios

Scenario	Dataset	Additional Layer	*Dropout*	*Learning rate*
1	Unbalanced dataset	*None*	0.1, 0.2, 0.3, 0.4	10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}
2	Unbalanced dataset	*Dense*32 *Layer*	0.1, 0.2, 0.3, 0.4	10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}
3	Balanced dataset	*None*	0.1, 0.2, 0.3, 0.4	10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}
4	Balanced dataset	*Dense*32 *Layer*	0.1, 0.2, 0.3, 0.4	10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}

3.4. Model Testing

Table 3 presents the comparative results of four scenarios in this research experiment as described in Table 2. In scenario 1, no linear relationship is observed between the dropout and learning rate values, for example when using lower dropout value of 0.1 the best result was achieved when using the higher learning rate value of 10^{-1}. Conversely, when using the higher dropout value of 0.3, the best result was achieved when using the lower learning value of 10^{-4}. However, it can be seen that most of experiments generally provides better accuracy when using the lower learning rate value. The best experiment in this scenario was achieved by experiment used dropout value of 0.3, and learning rate value of 10^{-4} with the highest F1-score of 0.85 and the highest accuracy of 0.874.

Table 3. Comparison Results of Each Scenarios for Waste Image Classification

Dropout	Learning rate	Scenario 1		Scenario 2		Scenario 3		Scenario 4	
		F1-score	Acc	F1-score	Acc	F1-score	Acc	F1-score	Acc
0.1	10^{-1}	0.84	0.850	0.29	0.358	0.79	0.812	0.20	0.314
0.1	10^{-2}	0.81	0.840	0.58	0.641	0.79	0.816	0.43	0.610
0.1	10^{-3}	0.82	0.842	0.84	0.864	0.78	0.820	0.83	0.874
0.1	10^{-4}	0.83	0.846	0.83	0.856	0.82	0.850	0.84	0.866
0.2	10^{-1}	0.79	0.818	0.19	0.312	0.78	0.822	0.12	0.255
0.2	10^{-2}	0.82	0.850	0.37	0.456	0.79	0.814	0.41	0.625
0.2	10^{-3}	0.86	0.868	0.82	0.844	0.80	0.828	0.84	0.860
0.2	10^{-4}	0.84	0.854	0.86	0.872	0.83	0.844	0.84	0.870
0.3	10^{-1}	0.83	0.848	0.24	0.346	0.80	0.816	0.18	0.228
0.3	10^{-2}	0.81	0.836	0.39	0.460	0.82	0.836	0.61	0.549
0.3	10^{-3}	0.83	0.852	0.85	0.862	0.85	0.860	0.84	0.830
0.3	10^{-4}	**0.85**	**0.874**	**0.87**	**0.885**	0.82	0.828	0.87	0.872
0.4	10^{-1}	0.80	0.834	0.24	0.316	0.81	0.836	0.17	0.137
0.4	10^{-2}	0.82	0.840	0.36	0.435	0.80	0.822	0.48	0.677
0.4	10^{-3}	0.81	0.870	0.86	0.870	**0.85**	**0.864**	0.83	0.864
0.4	10^{-4}	0.82	0.856	0.86	0.879	0.83	0.842	**0.85**	**0.881**

The results in scenario 2 indicate that the accuracy and F1-score obtained in each experiment are higher when using lower learning rate values. At each variation of dropout value, the F1-score and accuracy are higher when using lower learning rate value. When using higher learning rate value, both accuracy and F1-score decrease significantly compared to the first scenario. This condition implies that when additional Dense32 layer is used, the network need slower update in the learning process by using the lower learning rate. The test results for scenario 2 show that experiment with the dropout value of 0.3, and learning rate value of 10^{-4} obtained the highest F1-score value of 0.87 and the highest accuracy value of 0.885. The addition of an additional layer in the form of the Dense32 layer also contributes to the finding hidden features or patterns so that it can increase the accuracy of model.

The results in scenario 3 indicate that there is no linear relationship between the value of dropout and learning rate. The results of experiment vary depend on the combination of dropout and learning values. The best results in scenario 3 is able to achieve an F1-score of 0.85 and an accuracy of 0.864

Vol. 6, No. 3, Juni 2025, Page. 1147-1158
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.3.4417

when using dropout 0.4 and learning rate 10^{-3}. The best accuracy achieved in this scenario is lower than the highest accuracy obtained in Scenario 1 which utilized unbalanced dataset. These results indicate that the augmentation has not been able to improve the model's accuracy.

In scenario 4, it can be seen that the accuracy and F1-score obtained in each experiment are higher when using lower learning rate values. At each variation of dropout value, the F1-score and accuracy are higher when using lower learning rate value. Experiments with lower learning rate values show significant decrease in the F1-score and accuracy compared to the experiments in scenario 3. This condition implies that when additional Dense32 layer is used, the network is suitable to be trained by using low learning rate value which allow slower weight updates in the learning process. The best experiment in scenario 4 is able to reach F1-score of 0.85 and the accuracy of 0.881 by using dropout 0.4, and learning rate 10^{-4}. In this case, dropout plays an important role in training the model to prevent overfitting, while the addition of the Dense32 layer is responsible to determine the hidden patterns or features so that the model may achieve better accuracy.

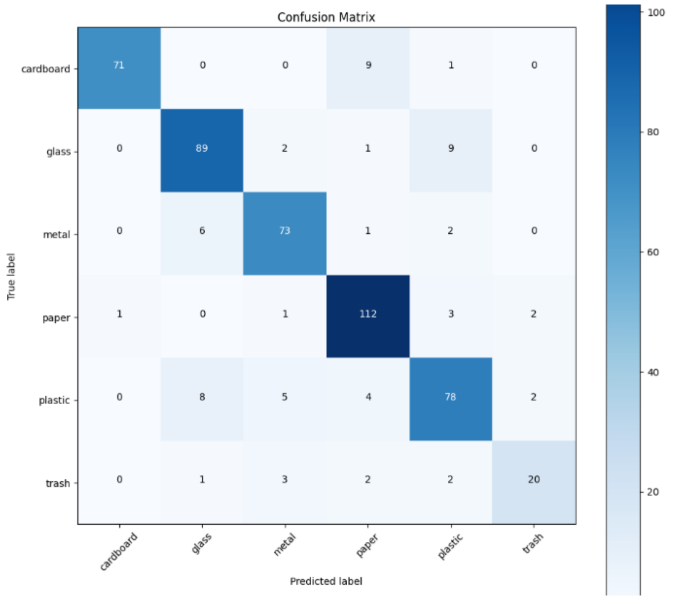


Figure 4. Confusion Matrix Of The Best Model

The prediction performance of the best model in each class is presented by the confusion matrix in Figure 4. The confusion matrix shows that most of the predictions made by the model were accurate as seen with darker color in the diagonal area compared to the surrounding area of the confusion matrix. The highest sensitivity of all class was achieved by paper class, while the lowest sensitivity of all class was achieved by the trash class. This can happen because the paper class has the largest number of samples, while the plastic class has the smallest number of samples compared to the other classes. In addition, it can be seen that the incorrectly predicted images in the cardboard category were mainly predicted to class paper. This can be due to the similar shape of cardboard and paper. The incorrectly predicted images in the glass category were mainly predicted to plastic, and vice versa. This can happen because glass and plastic are both transparent or shiny color. The trash class in this case has the lowest true positive value compared to other classes. This could be due to the more varied shapes and colors in the sample images in this class compared to other classes.

4. DISCUSSIONS

Figure 5 presents comparison of accuracy and f1-score for the best models in each scenario. The highest performance is achieved by the model trained on the unbalanced dataset with the addition of a

Vol. 6, No. 3, Juni 2025, Page. 1147-1158
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.3.4417

Dense(32) layer, a dropout rate of 0.3, and a learning rate of 1e-4, yielding an accuracy of 0.885 and an F1-score of 0.87. These results indicate that incorporating the Dense(32) layer before the output layer consistently enhances model performance (F1-score and accuracy) regardless of whether the dataset is balanced or unbalanced. The inclusion of this additional hidden layer increases the network's capacity to learn more complex patterns from the training data. Interestingly, the use of data augmentation to balance the dataset by increasing the number of minority class samples did not lead to improved performance. This is evidenced by the comparatively lower accuracy and F1-score obtained from models trained on the balanced dataset, with or without the Dense(32) layer. Therefore, it can be concluded that, in this study, the addition of the Dense(32) layer has a more significant impact on model performance than balancing the dataset through augmentation techniques.

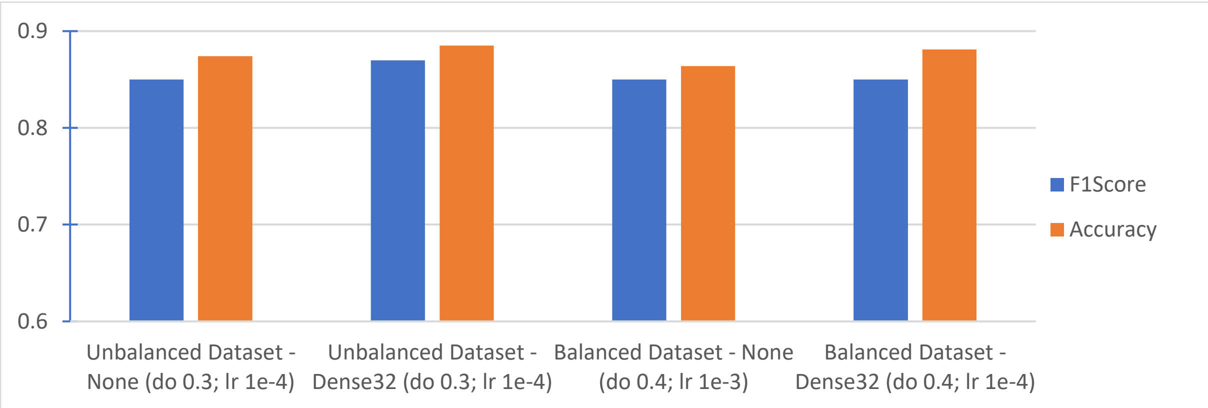


Figure 5. Comparison of Accuracy and F1-Score for The Best Models in Each Scenario

The best results in each scenario are also generally achieved when using higher dropout values and lower learning rate value. This implies that the use of dropout is able to minimize the risk of overfitting by deactivating a number of neurons randomly with the aim that each neuron has the same learning opportunity. In this case, the use of lower learning rate values also plays a good role. The low learning rate value allows the network to learn slowly by updating the weights of network with smaller values in the model training, so that network performance is more stable and able to achieve learning convergence.

The comparison of accuracy and the number of model parameters between EfficientNet-B0, as implemented in this study, and other CNN architectures from previous studies is presented in Table 4. All of these studies were conducted by using Trashnet dataset. In terms of accuracy, the EfficientNet-B0 model achieved the highest accuracy of 0.885, followed by Xception and ResNet-50, respectively. In terms of the number of parameters, MobileNet has the lowest number of parameters, followed by EfficientNet-B0. The other model has larger number of parameters exceeding 10 million and the VGG-16 is the largest model with more than 100 million parameters. However, the accuracy of MobileNet is relatively low, 0.644. With the trade-off between accuracy and the number of parameters, it can be concluded that EfficientNet-B0 is the best model, achieving the highest accuracy while maintaining a significantly lower number of parameters compared to other models with similar accuracy.

Table 4. Comparison of EfficientNet-B0 and Other CNN Architectures

Model	Accuracy	Number of parameters (million)
ResNet-50 [14]	0.870	23.60
VGG-16 [15]	0.846	134.28
ResNet-50 [15]	0.855	23.6
Xception [15]	0.879	14.59

Model	Accuracy	Number of parameters (million)
MobileNet [16]	0.644	**4.23**
AlexNet [16]	0.778	14.59
LMNet [16]	0.854	12.04
EfficientNet-B0	**0.885**	6.06

5. CONCLUSION

This research built an efficient model for waste image classification to support waste sorting by utilizing Convolutional Neural Network (CNN) with EfficientNet-B0 architecture. In addition, this study also compared the use of additional hidden layer before the output layer, as well as some combination of hyperparameter values, namely dropout and learning rate. The experimental results show that the image augmentation implemented in this case to balance the dataset has not been able to increase the accuracy or F1-score achieved by the trained model. On the other hand, the addition of a hidden layer in the form of Dense32 successfully increase the accuracy and F1-score of the model whether the model was trained using a balanced dataset or an unbalanced dataset. The best results in each scenario are also generally achieved when using higher dropout and lower learning rate values. The best model was achieved when the model was trained on an unbalanced dataset and using the additional hidden layer Dense32. The confusion matrix of the best model shows that the model has good sensitivity in each class prediction. Furthermore, our proposed model achieved the highest accuracy while maintaining a significantly lower number of parameters compared to other CNN architectures with comparable accuracy, such as ResNet-50 and Xception. Future research can be extended to explore data augmentation methods and utilize more diverse datasets to improve the performance of model. The resulting waste classification model can then be further implemented to build an automatic waste sorter.

REFERENCES

[1] Z. Zhang *et al.*, "Municipal solid waste management challenges in developing regions: A comprehensive review and future perspectives for Asia and Africa," Jun. 20, 2024, *Elsevier B.V.* doi: 10.1016/j.scitotenv.2024.172794.

[2] United Nations Environment Programme, *Global Waste management Outlook 2024: Beyond an age of waste - Turning rubbish into resource*. Nairobi: United Nations Environment Programme, 2024. doi: https://doi.org/10.59117/20.500.11822/44939.

[3] D. V. Yevle and P. S. Mann, "Artificial intelligence based classification for waste management: A survey based on taxonomy, classification & future direction," May 01, 2025, *Elsevier Ireland Ltd*. doi: 10.1016/j.cosrev.2024.100723.

[4] Khadijah, S. N. Endah, R. Kusumaningrum, Rismiyati, P. S. Sasongko, and I. Z. Nisa, "Solid waste classification using pyramid scene parsing network segmentation and combined features," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 19, no. 6, pp. 1902–1912, 2021, doi: 10.12928/TELKOMNIKA.v19i6.18402.

[5] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, "Review of image classification algorithms based on convolutional neural networks," Nov. 01, 2021, *MDPI*. doi: 10.3390/rs13224712.

[6] R. Hanieh, S. Elham, and S. B. Mehdi, "Enhancing breast cancer detection in thermographic images using deep hybrid networks," *Imaging and Radiation Research*, vol. 7, no. 1, p. 6195, Jul. 2024, doi: 10.24294/irr6195.

[7] A. Peryanto, A. Yudhana, and R. Umar, "Convolutional Neural Network and Support Vector Machine in Classification of Flower Images," *Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika*, vol. 8, no. 1, pp. 1–7, 2022, doi: 10.23917/khif.v8i1.15531.

[8] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539ï.

[9] R. Sidharth, P. Rohit, S. Vishagan, R. Karthika, and M. Ganesan, "Deep Learning based Smart Garbage Classifier for Effective Waste Management," in *2020 5th International Conference on*

Communication and Electronics Systems (ICCES), Coimbatore, India: IEEE, Jun. 2020, pp. 1086–1089. doi: 10.1109/ICCES48766.2020.9137938.

[10] A. Altikat, A. Gulbe, and S. Altikat, "Intelligent solid waste classification using deep convolutional neural networks," *International Journal of Environmental Science and Technology*, vol. 19, no. 3, pp. 1285–1292, Mar. 2022, doi: 10.1007/s13762-021-03179-4.

[11] R. Faria, F. Ahmed, A. Das, and A. Dey, "Classification of Organic and Solid Waste Using Deep Convolutional Neural Networks," in *2021 IEEE 9th Region 10 Humanitarian Technology Conference (R10-HTC)*, Bangalore, India: IEEE, Sep. 2021, pp. 01–06. doi: 10.1109/R10-HTC53172.2021.9641560.

[12] A. Gaurav *et al.*, "Smart waste classification in IoT-enabled smart cities using VGG16 and Cat Swarm Optimized random forest," *PLoS One*, vol. 20, no. 2 February, Feb. 2025, doi: 10.1371/journal.pone.0316930.

[13] D. Gyawali, A. Regmi, A. Shakya, A. Gautam, and S. Shrestha, "Comparative Analysis of Multiple Deep CNN Models for Waste Classification," Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.02168

[14] O. Adedeji and Z. Wang, "Intelligent waste classification system using deep learning convolutional neural network," in *Procedia Manufacturing*, Elsevier B.V., 2019, pp. 607–612. doi: 10.1016/j.promfg.2019.05.086.

[15] Rismiyati, S. N. Endah, Khadijah, and I. N. Shiddiq, "Xception Architecture Transfer Learning for Garbage Classification," in *The 4th International Conference on Informatics and Computational Sciences (ICICoS 2020)*, IEEE, 2020. doi: 10.1109/ICICoS51170.2020.9299017.

[16] M. Fan, K. Zuo, J. Wang, and J. Zhu, "A lightweight multiscale convolutional neural network for garbage sorting," *Systems and Soft Computing*, vol. 5, Dec. 2023, doi: 10.1016/j.sasc.2023.200059.

[17] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *International Conference on Machine Learning*, Long Beach, USA, Jun. 2019, pp. 6105–6114. Accessed: May 14, 2024. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html

[18] S. M. Baik, K. S. Hong, and D. J. Park, "Deep learning approach for early prediction of COVID-19 mortality using chest X-ray and electronic health records," *BMC Bioinformatics*, vol. 24, no. 1, Dec. 2023, doi: 10.1186/s12859-023-05321-0.

[19] Khadijah, R. Kusumaningrum, Rismiyati, and A. Mujadidurrahman, "An Efficient Masked Face Classifier Using EfficientNet," in *2021 5th International Conference on Informatics and Computational Sciences (ICICoS)*, Semarang: IEEE, Nov. 2021, pp. 277–281. doi: 10.1109/ICICoS53627.2021.9651883.

[20] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using EfficientNet deep learning model," *Ecol Inform*, vol. 61, Mar. 2021, doi: 10.1016/j.ecoinf.2020.101182.

[21] D. R. Nayak, N. Padhy, P. K. Mallick, M. Zymbler, and S. Kumar, "Brain Tumor Classification Using Dense Efficient-Net," *Axioms*, vol. 11, no. 1, Jan. 2022, doi: 10.3390/axioms11010034.

[22] K. M. Chung *et al.*, "Deep Learning-Based Knee MRI Classification for Common Peroneal Nerve Palsy with Foot Drop," *Biomedicines*, vol. 11, no. 12, Dec. 2023, doi: 10.3390/biomedicines11123171.

[23] V. H. Barella, L. P. F. Garcia, M. C. P. de Souto, A. C. Lorena, and A. C. P. L. F. de Carvalho, "Assessing the data complexity of imbalanced datasets," *Inf Sci (N Y)*, vol. 553, pp. 83–109, Apr. 2021, doi: 10.1016/j.ins.2020.12.006.

[24] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, Jun. 2004, doi: 10.1145/1007730.1007735.

[25] N. E. Khalifa, M. Loey, and S. Mirjalili, "A comprehensive survey of recent trends in deep learning for digital images augmentation," *Artif Intell Rev*, vol. 55, no. 3, pp. 2351–2377, Mar. 2022, doi: 10.1007/s10462-021-10066-4.

[26] M. Yang and G. Thung, "Classification of Trash for Recyclability Status," 2016. Accessed: May 13, 2024. [Online]. Available: https://cs229.stanford.edu/proj2016/report/ThungYang-ClassificationOfTrashForRecyclabilityStatus-report.pdf

Vol. 6, No. 3, Juni 2025, Page. 1147-1158
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.3.4417

[27] M. Yani, B. Irawan, and C. Setiningsih, “Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry’s Nail,” in *Journal of Physics: Conference Series*, Institute of Physics Publishing, May 2019. doi: 10.1088/1742-6596/1201/1/012052.

[28] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[29] S. Ravichandiran, *Hands-On Deep Learning Algorithms with Python*. Packt Publishing Ltd, 2019.

[30] T. Ganegedara, *Natural Language Processing with TensorFlow*. Birmingham: Packt Publishing Ltd, 2018.

[31] X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, “An improved method to construct basic probability assignment based on the confusion matrix for classification problem,” *Inf Sci (N Y)*, vol. 340–341, pp. 250–261, May 2016, doi: 10.1016/j.ins.2016.01.033.

Vol. 6, No. 3, Juni 2025, Page. 1147-1158
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.3.4417