

EfficientNet Model for Multiclass Classification of The Correctness of Wearing Face Mask

Khadijah¹, Retno Kusumaningrum¹, Rismiyati¹, Nur Sabilly¹

¹Department of Informatics, Universitas Diponegoro, Indonesia

Article Info

Article history:

Received Oct 26, 2023

Revised Dec 19, 2024

Accepted Jan 11, 2025

Keywords:

CNN

Classification

EfficientNet

Masked-face image

Wearing face mask

ABSTRACT

A face mask is essential for protecting individuals from the entry of infectious or hazardous materials through the nose or mouth in specific situations. To optimize its protective function, it must be worn correctly. This research aims to develop a multiclass classification model, rather than a binary one, to assess the correctness of wearing face mask. The proposed model is designed to achieve high accuracy while maintaining efficiency, with a low number of model parameters. To this end, a deep convolutional neural network (CNN), specifically EfficientNet, is utilized. Experiments are conducted on the public MaskedFace-Net image dataset, which consists of four categories (correctly masked, uncovered chin, uncovered nose, and uncovered nose and mouth), using 3,000 randomly selected images from each category. The experiments test several EfficientNet models (B0-B3) and network hyperparameters (learning rate and dropout). The best accuracy of 0.99 is achieved by EfficientNet-B0 with a learning rate of 0.01 and a dropout rate of 0.2. The EfficientNet-B0 model outperforms other benchmark CNN models, including MobileNet-V3 and Inception-V3, despite having a slightly higher number of parameters than MobileNet-V3. This result demonstrates that the EfficientNet model is both accurate and efficient for multiclass classification of the correctness of wearing face mask.

Copyright © 2025 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Khadijah,

Department of Informatics,

Universitas Diponegoro,

Prof. Soedarto, S.H. Street, Tembalang, Semarang 50275, Indonesia,

Email: khadijah@live.undip.ac.id

1. INTRODUCTION

The World Health Organization (WHO) strongly recommended that people wear face masks [1] after the declaration of the COVID-19 pandemic in 2020 [2]. Even though we have now entered the post-pandemic era, the recommendation to use face masks still continues. These recommendations are specifically aimed at people who have recently been exposed to COVID-19, individuals who suspect they are suffering from COVID-19, those at high risk of exposure, and even healthy people in crowded areas or poorly ventilated rooms [3]. In these situations, a face mask is used to prevent the entry or exit of droplets, which are the primary means of transmission for COVID-19, from or into the mouth or nose. Therefore, a face mask must be worn correctly (covering the nose and mouth) to optimize its protective function [1], [4].

In addition to the COVID-19 pandemic, wearing a face mask is also essential in the medical field, particularly for frontline healthcare workers and surgeons during surgery. The use of face masks is also crucial for workers in the mining and construction industries, where they serve as protective equipment against the inhalation of toxic dust in these environments [5].

Computer vision can be utilized to ensure whether a person is wearing a face mask correctly or not by developing a supervised model to predict its wearing condition. Currently, Convolutional Neural Network (CNN) is type a deep network with the convolutional layers in the beginning of network architecture. The convolutional layers of CNN are very useful for feature map extraction of an input image. Hence, the researchers do not need to perform feature extraction separately [6]. Therefore, many studies utilized CNN as

main algorithm to solve various tasks in computer vision, including image classification. The accuracy achieved by CNN in image classification is higher than the accuracy of other conventional machine learning algorithms, such as Random Forest, Naive Bayes Classifier, and Support Vector Machine [7].

Previous studies have been conducted on masked-face image classification. Kong et al. developed a model framework that receives input from real-time video, detects the face area using Inception, and predicts it using MobileNet-V3 [8]. Sanjaya et al. also applied MobileNetV2 for masked-face image classification, achieving an accuracy of 96.85% [9]. Umer et al. performed face mask classification using a dataset of 750 real images and a customized CNN as the classifier algorithm. Their experimental results showed that the CNN achieved the highest accuracy of 97.5%, compared to other classical machine learning algorithms [10]. Thai et al. performed head pose estimation and masked face classification using deep learning. In the classification task, ResNet152 achieved the highest accuracy on the MAFA (Masked Face) dataset. However, the MAFA dataset has an unbalanced ratio of masked face images to normal face images [11]. Another study by Basha et al. applied ResNet50 to the Real-world Masked Face Recognition Dataset (RMFRD), which contains 90,000 normal (non-masked) face images and 5,000 masked face images. ResNet50 achieved an accuracy of 97.8% [12].

Previous studies applying CNNs generally demonstrated good performance in terms of accuracy. However, some studies were conducted using limited datasets or imbalanced data. Training a model with limited data increases the risk of overfitting, and the complexity of the classifier may further exacerbate this risk [13]. The use of imbalanced training data can negatively impact the classifier's performance on minority classes [14]. Additionally, previous studies have primarily focused on predicting whether individuals use a face mask or not (a binary classification problem). To better optimize the protective function of face masks, it is crucial to classify the accuracy of face mask usage into more precise categories.

Koklu et al. performed classification of face mask wearing condition into four categories. The combination of Transfer Learning VGG-16 and Bidirectional Long Short Term Memory (BiLSTM) were applied as feature extractor and predictor, respectively. However, their experiment involved only 2000 images and took long training time [15]. The resulting final model consists of two deep learning networks that involve a large number of parameters and high computational time. The prediction of wearing mask conditions, especially in public locations must be performed quickly to ensure the efficiency of the automatic prediction model.

In view of these shortcomings, this research aims to develop a masked-face image classification model that focuses on the following three aspects: i) multiclass classification of the correctness of wearing a face mask, rather than binary classification, to ensure that a person is wearing the face mask correctly; ii) high accuracy of the model; and iii) model efficiency, with a low number of parameters to ensure fast predictions. To achieve these objectives, this research utilizes a customized CNN, namely EfficientNet, to develop the classification model.

The EfficientNet architecture is optimized to achieve high accuracy with a low number of network parameters and computational time. EfficientNet is designed to achieve those objectives by using the principle of compound scaling, rather than arbitrary scaling, for depth (the number of convolutional layers), width (the number of filters), and resolution (the size of the input image) of the network. The depth, width, and resolution influence each other, for example when the resolution of image is bigger, the network requires more convolutional layers (depth) to expand the receptive fields of network and more filters (width) to extract more fine-grained features of the image. Additionally, the main building block of the base EfficientNet model, namely MBConv (mobile inverted bottleneck convolution), combines expansion convolution, depthwise convolution, and projection convolution sequentially. As a result, MBConv is more efficient than standard convolution in extracting complex features [16]. The advantages of EfficientNet align with the face mask classification model which requires high accuracy and low computational cost, since the model will likely be implemented on edge or mobile device.

EfficientNet-B7 achieved the highest accuracy of 84.3% in ImageNet classification by using more than eight times fewer network parameters compared to the other competitors [16]. In the binary classification of masked face images, EfficientNet outperforms ResNet-50 and Inception [17]. Some modifications of EfficientNet also achieve the highest accuracy in rock image classification, outperforming other CNN architectures such as AlexNet, GoogleNet, VGG16, and ResNet34 [18]. The success of EfficientNet is also seen in other image classification tasks, including brain tumor classification from magnetic resonance images [19], skin disease classification [20], classification of plant leaf disease [21], and heartbeat sound classification based on spectrogram image [22].

Therefore, this research employs EfficientNet to obtain an accurate and efficient model for predicting the correctness of wearing face mask, which is divided into four categories: correctly masked, uncovered chin, uncovered nose, and also uncovered nose and mouth. To achieve this, the research utilizes a public masked-face image dataset, MaskedFace-Net [23]. The focus of this research is on image classification, where a face

image is taken as input and classified into the specific conditions of face mask usage. The main contributions of this paper are: i) investigating the performance of several EfficientNet families and the effect of hyperparameters (learning rate and dropout) in the context of multiclass masked-face image classification; and ii) developing a model that can classify or predict the correctness of face mask usage into the four categories.

2. METHOD

The flowchart of this research methodology is shown in Figure 1. The dataset was collected from an online public source. First, preprocessing was performed to prepare the dataset for classification (training and testing). This research applied EfficientNet as the classification model. Subsequently, the preprocessed data were randomly divided into training and testing sets. The training data were used to perform cross-validation with various hyperparameter values. Cross-validation was employed to find the best combination of hyperparameters. Then, the final model was trained using the training data and the best combination of hyperparameter values. In the final stage, the resulting model was evaluated using the testing data, and the evaluation results were reported.

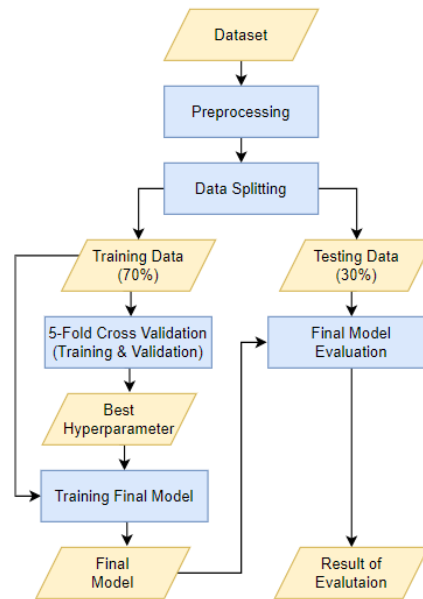


Figure 1. The flowchart of research methodology

2.1. Dataset

This research used the MaskedFace-Net dataset, which consists of the Correctly Masked Face Dataset (CMFD) and the Incorrectly Masked Face Dataset (IMFD). All images in both the CMFD and IMFD were created artificially from the Flickr-Faces-HQ (FFHQ) dataset. Each image in both datasets is an RGB image with a resolution of 1024 x 1024 pixels. The CMFD contains images of correctly masked faces, while the IMFD contains images of incorrectly masked faces, which are grouped into three categories/labels: 'uncovered chin,' 'uncovered nose,' and 'uncovered nose and mouth' [23]. Due to computational limitations and to balance the number of images in each category, this research selected 3,000 random images from the CMFD, labeled as 'correctly masked,' and 3,000 random images from each category of the IMFD. Therefore, the total number of images used in this research is 12,000. Examples of images in each category are shown in Figure 2.

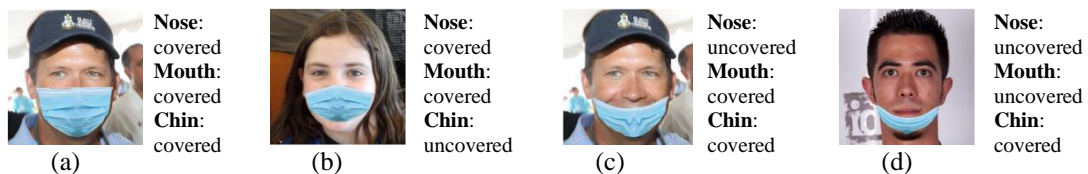


Figure 2. The example of an image in each category (a) correctly masked, (b) uncovered chin, (c) uncovered nose, and (d) uncovered nose and mouth [23]

2.2. Preprocessing

The preprocessing step aims to prepare the images before they are classified by the classification algorithm. In this stage, each image was resized to the resolution required by the classification algorithm. Different versions of EfficientNet require different image resolutions as input. Therefore, each image was resized to 224 x 224 for EfficientNet-B0, 240 x 240 for EfficientNet-B1, 260 x 260 for EfficientNet-B2, and 300 x 300 for EfficientNet-B3.

2.3. Data Splitting

The research dataset was randomly divided into 70% training data (8,400 images) and 30% testing data (3,600 images), with a balanced ratio in each category. Model training typically requires the adjustment of several hyperparameters. Therefore, 5-fold stratified cross-validation was applied to the 70% training data to find the best hyperparameters for training the final model. The 5-fold cross-validation was used instead of 10-fold cross-validation because the dataset is quite large (8,400 images). With 5-fold cross-validation, the number of training samples in each iteration is 6,720, which is sufficient for training a deep network. Additionally, the number of iterations in each validation is smaller, resulting in lower computational time. After performing cross-validation, the original 70% training data was retrained using the best combination of hyperparameters, and the 30% testing data was used to assess the performance of the final model.

2.4. Classification Using EfficientNet

EfficientNet is a type of convolution neural network (CNN) with efficient architectural model proposed by Tan and Le [24]. CNN is known as a type of deep learning network that has been successfully applied in research related to computer vision, such as image classification. The special layer of CNN, namely convolutional layer, is used for feature map extraction of an image. Therefore, by using CNN the features of an image can be extracted automatically, without having to specify the handcrafted features of an image manually [6].

The performance of CNN in extracting feature map depends on the architecture in the convolutional layer including the number of convolutional layers (depth), the number of filters (width) used in each convolutional layer, and the size of the input image (resolution). CNN usually uses more than one convolutional layer. The beginning convolutional layers are used to capture simpler features, while the deeper layers are used to capture more complex features of an image. Using the depth network is good to capture complex features, but may increase the problem of vanishing gradient in the training process. Subsequently, each convolutional layer has some number of filters or channels. The higher number of channels (wide network), the more fine-grained features can be captured. However, higher level features are difficult to be captured when the network is not deep enough, although the network is wide. Using the higher resolution of input image allow to capture more fine-grained pattern, but the experiment showed that using very high resolution may decrease the accuracy of network [25]. Researchers usually manually tune the depth, width, and resolution to find the best architecture that give the best performance to solve the specific case [24].

Individual scaling of depth, width, or resolution of a network may improve network performance. However, after reach the certain number of scaling, the network performance tends to saturate. Those parameters are related to each other, for example using the high input resolution usually requires the high number of filters and the deeper network to capture the more fine-grained features and the complex features. Therefore, Tan and Le proposed the principles of compound scaling to scale the depth, width, and the resolution of a network uniformly as shown by equation (1). The principle uses the compound scaling coefficient ϕ where d , w , r refers to depth, width, and resolution of network, respectively. The scaling coefficient ϕ is user-specified that is used to control available resources for model scaling. Then the parameter α , β , and γ specify the scaling up of the depth, width, and resolution of a network, respectively, according to the available resources specified by ϕ . A small grid search can be performed to determine the value of α , β , and γ [24].

$$\begin{aligned} d &= \alpha^\phi, w = \beta^\phi, r = \gamma^\phi \\ \text{s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 &\approx 2 \text{ and } \alpha \geq 1, \beta \geq 1, \gamma \geq 1 \end{aligned} \quad (1)$$

The baseline network of EfficientNet families is called EfficientNet-B0. Figure 3(a) shows the architecture of EfficientNet-B0. This baseline network uses mobile inverted bottleneck convolution (MBConv) added by squeeze-and-excitation (SE) optimization as its main building block [24]. MBconv block consist of several operations, namely 1x1 pointwise convolution, depthwise separable convolution, SE module, followed by 1x1 pointwise convolution and dropout as shown in Figure 3(b) [18]. The first 1x1 pointwise convolution is used to expand the feature map. Different from standard convolution operator, depthwise convolution perform single convolutional kernel per input channel [26]. The SE module calculates the weight of each channel in the input feature map based on its importance. The last 1x1 pointwise convolution is used to narrow

back the size of feature map. Then, the resulting feature map are added to the initial input feature map. Batch normalization is added to standardized the output values so that the model can reach convergency more quickly. Swish activation function is used to add non-linearity [18].

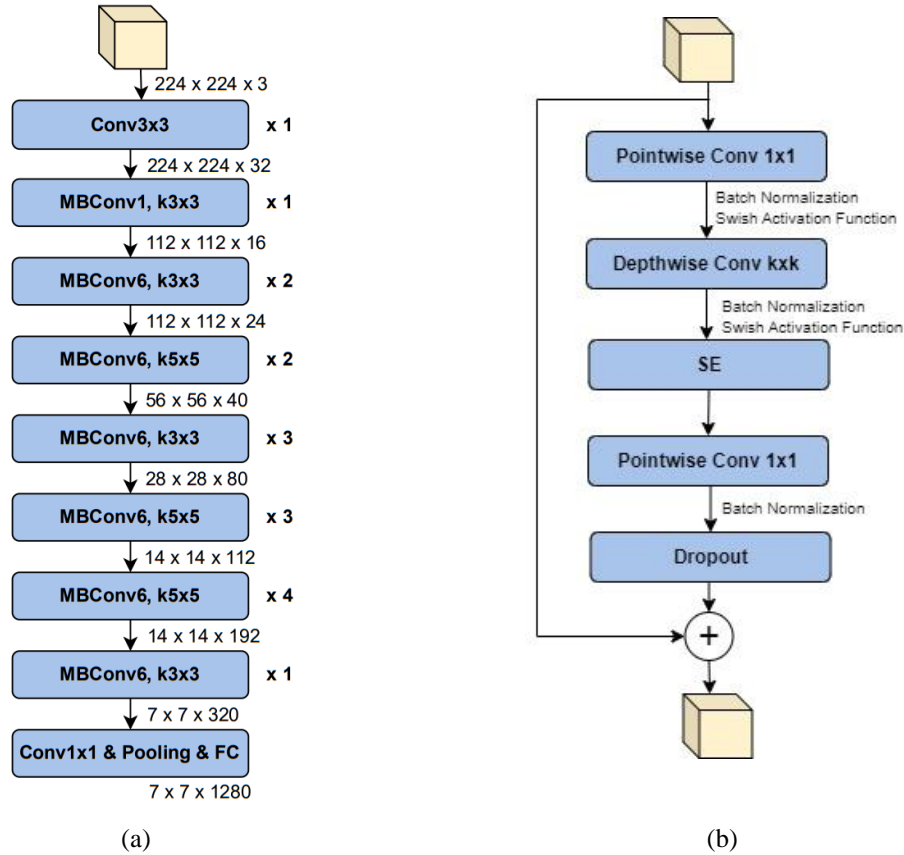


Figure 3. (a) Architecture of baseline EfficientNet-B0 [16] and (b) Structure of MBConv module[18]

By using the principle of compound scaling as (1), the baseline network is scaled up from EfficientNet-B0 until EfficientNet-B7. Using the value $\phi = 1$, EfficientNet-B0 found the best value of $\alpha = 1.2$, $\beta = 1.1$, and $\gamma = 1.15$ through the small grid search. Then, the version of EfficientNet-B1 to B7 are obtained by using the constant value of those α , β , and γ with different value of compound scaling ϕ . Each model of EfficientNet requires the different input size or resolution. The higher version of EfficientNet have a deeper and wider network architecture and requires the higher input resolution. EfficientNetB0 until EfficientNet-B7 uses the input resolution of 224, 240, 260, 300, 380, 456, 528 respectively [24].

The architecture of the classification model is shown in Figure 4. The classification model for predicting the label of masked face image in this research was built by using EfficientNet with the weight initialization from the pre-trained EfficientNet model on ImageNet dataset. Therefore, it allows to use the lower epoch in the model training because the weight initialization is not carried out randomly. After the last layer of pretrained EfficientNet, the GlobalAveragePooling2D was added for dimensionality reduction of the spatial feature map. Then, batch normalization was added to standardized the output values of the preceding layer in order to avoid the explosion of vanishing gradient problem. Subsequently, dropout was also added to reduce the number of nodes randomly in order to avoid overfitting. The last layer is output layer consisting of 4 nodes with softmax activation function because there are four classes in this classification problem. The outputs of softmax layer represent the probabilities of an input data belonging to each class label. The equation of softmax function is shown by equation (2) where z_i is the weighted sum of i -th output node and m is the number of classes [25].

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{i=1}^m \exp(z_i)} \quad (2)$$

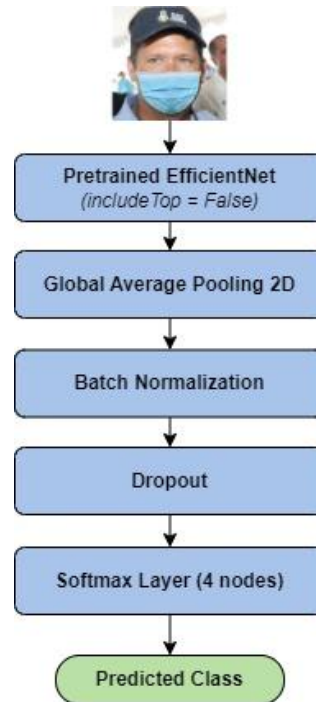


Figure 4. Architecture of EfficientNet for multiclass classification of the correctness of wearing face mask

2.5. Evaluation Metrics

This study utilized accuracy as an evaluation metrics to measure the performance of the resulting classifier model since the number of samples in each class is equal. Accuracy is obtained from the number of correctly predicted data divided by the total number of data [27]. In addition to accuracy, the loss is also reported. Since the problem is multiclass classification, the categorical cross entropy loss function was used. The loss value measures the difference between the predicted probability and the actual category of each samples.

3. RESULTS AND DISCUSSION

These experiments involve two phases: model development and model evaluation. The goal of model development is to select the best combination of hyperparameters by performing stratified 5-fold cross-validation on the training data. In the model evaluation phase, the performance of the final model is assessed. The best hyperparameter values from the model development phase are used to train the final model using the training data. Then, the resulting final model is evaluated using the testing data.

3.1. Model Development

These experiments involve some hyperparameters tuning to find the best combination of them which provide the model with the highest accuracy. The combination of hyperparameters for EfficientNet model in this research are:

- The versions of EfficientNet: B0, B1, B2, and B3. Each version of EfficientNet has different network layer architecture and requires different size of input data. The higher version of EfficientNet model have deeper and wider network architecture resulting in larger number of model parameters or more complex model. A complex model may have better capacity to handle more complex case, but also has higher risk of overfitting. In this research, we limit to EfficientNet-B3 because the higher version needs higher size of data input which requires higher computational time.
- Learning rate: 0.1, 0.01, and 0.001. Learning rate affects the speed of training process. A small value of learning rate will result in a small weight update. However, if the learning rate is too large, it also allows for divergence of the network.
- Dropout rate: 0.2 and 0.4. Dropout reduce the number of nodes in a certain layer randomly to avoid overfitting. The higher dropout rate may reduce the higher number of nodes in a layer which may reduce the learning capacity of network. Therefore, to reduce the risk of overfitting, while maintaining the performance of network, the dropout value needs to be fine-tuned.

Finally, there are 24 combinations of hyperparameters in this scenario. The training process for all scenarios was carried out using a batch size of 64 and 20 epochs. The results of these experiments are presented

in Table 1. It is shown that most models achieve a lower validation loss when using a smaller learning rate. A lower learning rate allows for smoother weight updates by taking smaller steps in the direction of minimizing the loss function. The validation accuracies vary depending on the EfficientNet version and the hyperparameters used. However, each model is able to achieve an accuracy of more than 97%. This result confirms that EfficientNet models perform excellently in this classification task. The baseline EfficientNet-B0 outperforms the other versions, as shown by its accuracy value of more than 98% in all combinations of learning rate and dropout. To observe the effect of each hyperparameter on classification performance, the results are aggregated based on the hyperparameter values.

Table 1. The validation loss and accuracy of each model in model development

Model Number	Version	Learning Rate	Dropout	Validation Loss	Validation Accuracy (%)
1	EfficientNet-B0	0,100	0.2	3,619	98,107
2	EfficientNet-B0	0,100	0.4	3,619	98,000
3	EfficientNet-B0	0,010	0.2	0,083	98,976
4	EfficientNet-B0	0,010	0.4	0,156	98,357
5	EfficientNet-B0	0,001	0.2	0,156	98,357
6	EfficientNet-B0	0,001	0.4	0,047	98,619
7	EfficientNet-B1	0,100	0.2	4,651	97,560
8	EfficientNet-B1	0,100	0.4	4,894	97,643
9	EfficientNet-B1	0,010	0.2	0,137	97,952
10	EfficientNet-B1	0,010	0.4	0,156	98,357
11	EfficientNet-B1	0,001	0.2	0,051	98,488
12	EfficientNet-B1	0,001	0.4	0,053	98,405
13	EfficientNet-B2	0,100	0.2	4,357	97,786
14	EfficientNet-B2	0,100	0.4	5,770	97,524
15	EfficientNet-B2	0,010	0.2	0,141	98,393
16	EfficientNet-B2	0,010	0.4	0,155	98,226
17	EfficientNet-B2	0,001	0.2	0,044	98,810
18	EfficientNet-B2	0,001	0.4	0,044	98,655
19	EfficientNet-B3	0,100	0.2	5,862	97,321
20	EfficientNet-B3	0,100	0.4	7,130	97,440
21	EfficientNet-B3	0,010	0.2	0,177	98,071
22	EfficientNet-B3	0,010	0.4	0,173	97,845
23	EfficientNet-B3	0,001	0.2	0,046	98,631
24	EfficientNet-B3	0,001	0.4	0,055	98,393

The comparison of average accuracy across different versions of EfficientNet with varying learning rates is shown in Figure 5. The results indicate that most models (EfficientNet-B1, B2, and B3) achieve higher accuracy with smaller learning rates, except for EfficientNet-B0, where the average accuracy is higher with a learning rate of 0.010 compared to 0.001. This suggests that lower learning rates tend to perform better for most versions of the EfficientNet network. When the learning rate is higher, the network weights are updated with the higher value, which can lead to instability in the training process and cause the model to miss the optimal result.

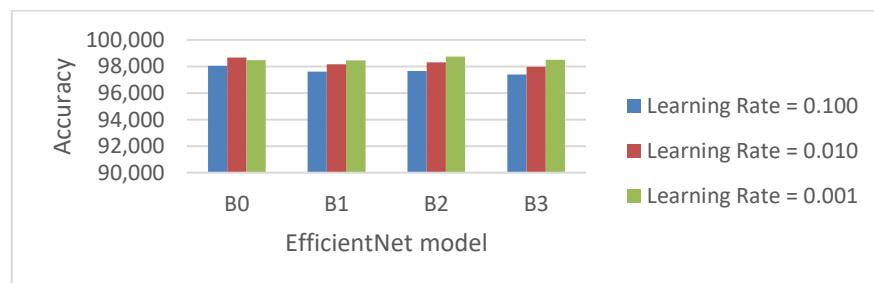


Figure 5. Comparison of the average accuracy across different versions of EfficientNet with varying learning rates

Figure 6 shows the comparison of average accuracy across different versions of EfficientNet with varying dropout values. Most experiments achieve higher accuracy with the smaller dropout value of 0.2,

compared to a dropout value of 0.4. This trend is observed in EfficientNet-B0, B2, and B3, while in EfficientNet-B1, most models perform better with the higher dropout value. Therefore, it can be concluded that the effectiveness of the dropout value is dependent on the architecture of the network model. However, in most cases, the smaller dropout value results in higher accuracy.

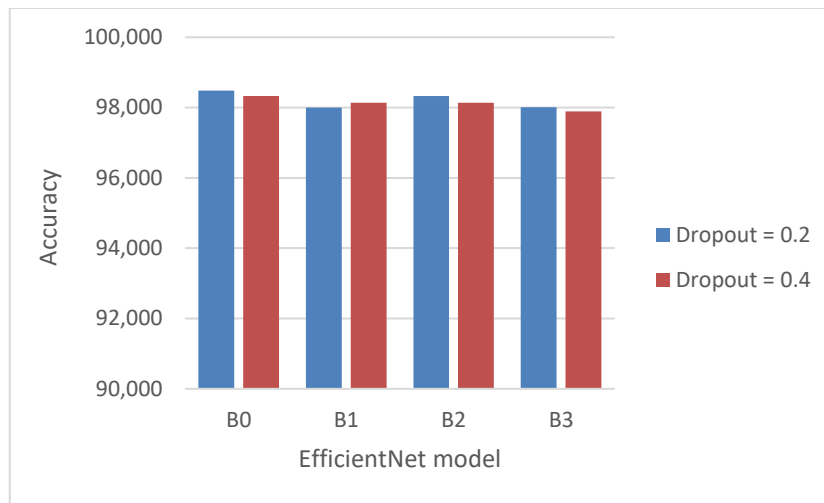


Figure 6. Comparison of the average accuracy across different versions of EfficientNet with varying dropout rate

The comparison of average accuracy across different versions of EfficientNet is shown in Figure 7. Based on the results, the average accuracy of the EfficientNet-B0 model is the highest among the other EfficientNet models. The average accuracy achieved by EfficientNet-B1 is lower than that of EfficientNet-B2. However, the most advanced version of EfficientNet in this experiment, EfficientNet-B3, achieves the lowest average accuracy compared to the other versions. This suggests that, in this case, a simpler architecture is sufficient to achieve the best result, while more complex architectures, with a larger number of model parameters, may increase the risk of overfitting.

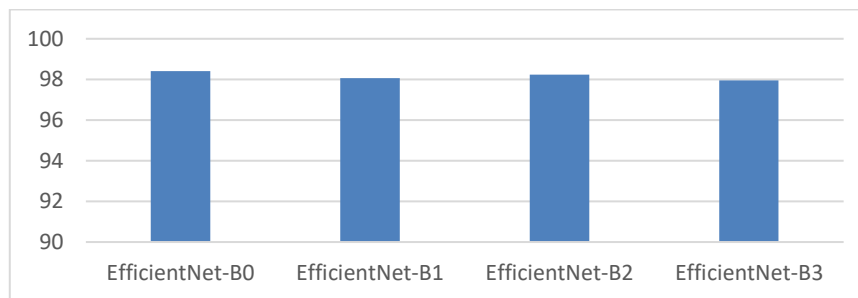


Figure 7. Comparison of average accuracy between versions of EfficientNet

3.2. Model Assessment

Based on the result in model development, it is found that the best hyperparameter values providing the best accuracy is:

- Model = EfficientNet-B0
- Learning rate = 0.010
- Dropout = 0.2.

After training the final model by using the best hyperparameter values on 70% training data and testing the model on 30% testing data, the performance of the final model is obtained. Figure 8 shows the performance of the final model on training and testing. It is shown that the model is able to reach good accuracy in early epoch, then the accuracy fluctuates slightly in the next iteration. The best accuracy reached is close to 0.99. The line of training and testing are lies to each other which shown that the network has good generalization performance and is not overfit.

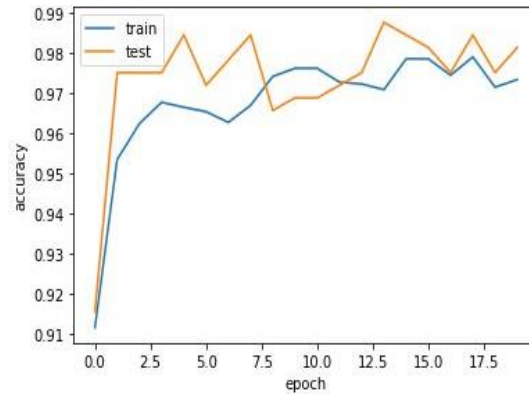


Figure 8. The accuracy of EfficientNet model in training and testing

In addition, this research also compares the performance of the best resulting model with other CNN benchmark architecture, such as MobileNet-V3 [28] and Inception-V3 [29]. Figure 9(a) and 9(b) show the performance of MobileNet-V3 and Inception-V3, respectively. It is shown that MobileNet-V3 and Inception-V3 require slightly more epochs to achieve good accuracy than EfficientNet-B0. Beside that, the best accuracy achieved by MobileNet-V3 is up to 0.95 which is lower than the best accuracy of EfficientNet-B0. The InceptionV3 model is only able to achieve the highest accuracy up to 0.72 which is much lower than EfficientNet-B0 or MobileNet-V3.

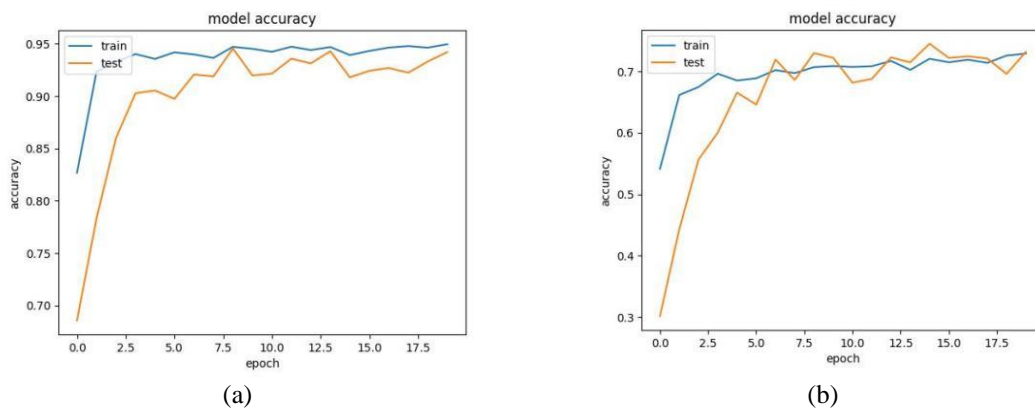


Figure 9. The accuracy of other CNN benchmark model in training and testing (a) MobileNet-V3 and (b) Inception-V3

In spite of accuracy, the number of model parameters also affects the performance of the model in terms of efficiency or speed of execution. It can be seen from Table 2 that MobileNet-V3 is the model with the lowest number of parameters, followed by EfficientNet-B0 and Inception-V3. The number of EfficientNet-B0 parameters is one-fifth lower than the number of Inception-V3 parameters. When compared to MobileNet-V3, the number of EfficientNet-B0 parameters is higher, but the difference is less than 25%.

Table 2. The Comparison of accuracy and number of parameters between CNN models

Model	Total Parameters	Trainable Parameters
EfficientNet-B0	4,059,815	7,684
MobileNet-V3	3,237,060	6,148
Inception-V3	21,819,172	12,292

Finally, the results are also compared to other research that utilized the MaskedFace-Net dataset. The proposed model outperforms the Vision Transformer method, which achieved a validation accuracy of 0.960 and a testing accuracy of 0.953 [30]. The proposed model also demonstrates competitive performance compared to the models developed by Azouji et al., who used CNN as a feature extractor and Large Margin Piecewise Linear (LMPL) as a classifier. Their research developed two models: the first model classifies face mask usage into three categories (correctly worn, incorrectly worn, and not worn) with an accuracy of 0.9953, while the second model classifies incorrectly worn face masks into three categories (uncovered chin, uncovered nose, and uncovered nose and mouth) with an accuracy of 0.9964 [31].

4. CONCLUSION

This research performs a multiclass classification of the correctness of face mask usage into four categories: correctly masked, uncovered nose, uncovered chin, and uncovered nose and mouth. The classification model is built using the EfficientNet pretrained model on the ImageNet classification dataset. The results of the experiment show that, in most scenarios, using lower values for the learning rate and dropout results in better performance. The best results are also achieved with the simplest version of the EfficientNet family, namely EfficientNet-B0. In the model assessment, the final EfficientNet-B0 achieves an accuracy close to 0.99. A comparison with other benchmark CNN architectures, namely MobileNet-V3 and Inception-V3, shows that the accuracy of EfficientNet-B0 is the highest among the others. In terms of the number of model parameters, the number of parameters in EfficientNet-B0 is much lower than that of Inception-V3, while slightly higher than that of MobileNet-V3. These results demonstrate that EfficientNet-B0 excels both in accuracy and efficiency (in terms of the number of model parameters).

The future research may combine the dataset of face image that do not use a face mask, therefore the resulting model may also differentiate between people that do not face mask, use face mask correctly, and do not use face mask correctly (uncovered nose, uncovered chin, and uncovered nose and mouth).

ACKNOWLEDGEMENTS

The authors would like to thank the Faculty of Science and Mathematics, Universitas Diponegoro, for the financial support of this research.




REFERENCES

- [1] World Health Organization (WHO), "Mask use in the context of COVID-19," Dec. 2020. Accessed: Aug. 12, 2023. [Online]. Available: WHO/2019-nCoV/IPC_Masks/2020.5
- [2] World Health Organization (WHO), "WHO Director-General's opening remarks at the media briefing on COVID-19," 11 March 2020, Accessed: Aug. 09, 2023. [Online]. Available: <https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020>
- [3] World Health Organization, "WHO updates COVID-19 guidelines on masks, treatments and patient care," <https://www.who.int/news/item/13-01-2023-who-updates-covid-19-guidelines-on-masks--treatments-and-patient-care>.
- [4] C. Matuschek *et al.*, "Face masks: Benefits and risks during the COVID-19 crisis," Aug. 12, 2020, *BioMed Central Ltd.* doi: 10.1186/s40001-020-00430-5.
- [5] Y. Goh, B. Y. Q. Tan, C. Bhartendu, J. J. Y. Ong, and V. K. Sharma, "The face mask: How a real protection becomes a psychological symbol during Covid-19?," Aug. 01, 2020, *Academic Press Inc.* doi: 10.1016/j.bbi.2020.05.060.
- [6] Y. Lecun, K. Kavukcuoglu, and C. Farabet, "Convolutional Networks and Applications in Vision." [Online]. Available: <http://www.cs.nyu.edu/>
- [7] S. Sagar and J. Singh, "An experimental study of tomato viral leaf diseases detection using machine learning classification techniques," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 1, pp. 451–461, Feb. 2023, doi: 10.11591/eei.v12i1.4385.
- [8] X. Kong *et al.*, "Real-Time Mask Identification for COVID-19: An Edge-Computing-Based Deep Learning Framework," *IEEE Internet Things J.*, vol. 8, no. 21, pp. 15929–15938, Nov. 2021, doi: 10.1109/JIOT.2021.3051844.
- [9] S. A. Sanjaya and S. A. Rakhmawan, "Face Mask Detection Using MobileNetV2 in The Era of COVID-19 Pandemic," in *International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, Sakheer: IEEE, Oct. 2020.
- [10] M. Umer *et al.*, "Face mask detection using deep convolutional neural network and multi-stage image processing," *Image Vis Comput*, vol. 133, May 2023, doi: 10.1016/j.imavis.2023.104657.
- [11] C. Thai, V. Tran, M. Bui, D. Nguyen, H. Ninh, and H. Tran, "Real-time masked face classification and head pose estimation for RGB facial image via knowledge distillation," *Inf Sci (N Y)*, vol. 616, pp. 330–347, Nov. 2022, doi: 10.1016/j.ins.2022.10.074.
- [12] C. Z. Basha, B. N. L. Pravallika, and E. B. Shankar, "An efficient face mask detector with pytorch and deep learning," *EAI Endorsed Trans Pervasive Health Technol*, vol. 7, no. 25, pp. 1–8, 2021, doi: 10.4108/eai.8-1-2021.167843.
- [13] X. Ying, "An Overview of Overfitting and its Solutions," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Mar. 2019. doi: 10.1088/1742-6596/1168/2/022022.
- [14] Y. Sun, A. K. C. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *Intern J Pattern Recognit Artif Intell*, vol. 23, no. 4, pp. 687–719, Jun. 2009, doi: 10.1142/S0218001409007326.




- [15] M. Koklu, I. Cinar, and Y. S. Taspınar, “CNN-based bi-directional and directional long-short term memory network for determination of face mask,” *Biomed Signal Process Control*, vol. 71, Jan. 2022, doi: 10.1016/j.bspc.2021.103216.
- [16] 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>
- [17] Khadijah, R. Kusumaningrum, Rismiyati, and A. Mujadidurrahman, “An Efficient Masked Face Classifier Using EfficientNet,” in *5th International Conference on Informatics and Computational Sciences (ICICoS)*, Semarang: IEEE, Nov. 2021.
- [18] Z. Huang, L. Su, J. Wu, and Y. Chen, “Rock Image Classification Based on EfficientNet and Triplet Attention Mechanism,” *Applied Sciences (Switzerland)*, vol. 13, no. 5, Mar. 2023, doi: 10.3390/app13053180.
- [19] F. Zulfiqar, U. Ijaz Bajwa, and Y. Mehmood, “Multi-class classification of brain tumor types from MR images using EfficientNets,” *Biomed Signal Process Control*, vol. 84, Jul. 2023, doi: 10.1016/j.bspc.2023.104777.
- [20] A. Rafay and W. Hussain, “EfficientSkinDis: An EfficientNet-based classification model for a large manually curated dataset of 31 skin diseases,” *Biomed Signal Process Control*, vol. 85, Aug. 2023, doi: 10.1016/j.bspc.2023.104869.
- [21] 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.
- [22] H. F. D. Ul Haq *et al.*, “EfficientNet Optimization on Heartbeats Sound Classification,” in *Proceedings - International Conference on Informatics and Computational Sciences*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 216–221. doi: 10.1109/ICICoS53627.2021.9651818.
- [23] A. Cabani, K. Hammoudi, H. Benhabiles, and M. Melkemi, “MaskedFace-Net – A dataset of correctly/incorrectly masked face images in the context of COVID-19,” *Smart Health*, vol. 19, Mar. 2021, doi: 10.1016/j.smhl.2020.100144.
- [24] M. Tan and Q. V. Le, “Rethinking model scaling for convolutional neural networks,” in *Proceedings of the International Conference on Machine Learning*, Long Beach, CA, USA, May 2019, pp. 6105–6114. doi: <https://doi.org/10.48550/arXiv.1905.11946>.
- [25] J. Krohn, G. Beylerveld, and A. Bassens, *Deep Learning Illustrated: A Visual, Interactive Guide to Artificial Intelligence (Addison-Wesley Data & Analytics Series) 1st Edition*, 1st ed. Boston: Pearson Addison-Wesley, 2020.
- [26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” Jan. 2018, [Online]. Available: <http://arxiv.org/abs/1801.04381>
- [27] J. Han and M. Kamber, *Data Mining: Concepts and Techniques Second Edition*. San Farnsisco: Elsevier Inc., 2006.
- [28] A. Howard *et al.*, “Searching for MobileNetV3,” May 2019, Accessed: Oct. 26, 2023. [Online]. Available: <https://arxiv.org/abs/1905.02244>
- [29] C. Szegedy, V. Vanhoucke, S. Ioffe, and J. Shlens, “Rethinking the Inception Architecture for Computer Vision.” Accessed: Oct. 23, 2023. [Online]. Available: <https://arxiv.org/abs/1512.00567>
- [30] H. D. Jahja, N. Yudistira, and Sutrisno, “Mask Usage Recognition using Vision Transformer with Transfer Learning and Data Augmentation,” Mar. 2022, [Online]. Available: <http://arxiv.org/abs/2203.11542>
- [31] N. Azouji, A. Sami, and M. Taheri, “EfficientMask-Net for face authentication in the era of COVID-19 pandemic,” *Signal Image Video Process*, vol. 16, no. 7, pp. 1991–1999, Oct. 2022, doi: 10.1007/s11760-022-02160-z.

BIOGRAPHIES OF AUTHORS






Khadijah    received her Bachelor of Informatics Engineering (S.Kom) from the Universitas Diponegoro, Indonesia in 2011 and the Master of Computer Science (M.Cs) from the Universitas Gadjah Mada, Indonesia in 2014. She has been a lecturer with the Department of Informatics, Universitas Diponegoro since 2014. Her main research interests are artificial intelligence and machine learning. She can be contacted at email: khadijah@live.undip.ac.id.



Retno Kusumaningrum    received her Bachelor of Science (S.Si) degree in mathematics from Universitas Diponegoro, Semarang, Indonesia, in 2003, and her Master of Computer Science (M.Kom) and Ph.D. degrees from Universitas Indonesia, Depok, Indonesia in 2010 and 2014, respectively. She became a member of IEEE in 2016. She is currently a lecturer at the Department of Informatics, Faculty of Science and Mathematics, Universitas Diponegoro. Furthermore, she is currently participating in the Laboratory of Intelligent Systems. Her research interests include machine learning, natural language processing, computer vision, pattern recognition, and topic modelling. She is a member of the IEEE computational intelligence society, IEEE computer society, and ACM. Her awards and honors include the sandwich-like scholarship award from the Directorate General of Higher Education of Indonesia for visiting the School of System Engineering, University of Reading, Reading, U.K. as a student visitor in 2012, the best paper of the Second International Conference on Informatics and Computational Sciences in 2018, first place for outstanding lecturer - Universitas Diponegoro for the science and technology category in 2019, and second place for the best paper award of the Third International Symposium on Advanced Intelligent Informatics in 2020. She can be contacted at email: retno@live.undip.ac.id.



Rismiyati    received a Bachelor of Engineering (B.Eng) from the School of Electrical and Electronics Engineering, Nanyang Technological University (NTU), in 2007 and a Master of Computer Science (MCs) from the Universitas Gadjah Mada, Indonesia, in 2016. She has been a lecturer with the Department of Informatics, Universitas Diponegoro, since 2017. Her main research interests are artificial intelligence, image processing and machine learning. She can be contacted at email: rismiyati@live.undip.ac.id



Nur Sabilly received his Bachelor of Informatics (S.Kom) from the Department of Informatics, Universitas Diponegoro, Indonesia in 2023. He has been a research assistant with the Laboratorium of Intelligent System in Department of Informatics, Universitas Diponegoro since 2021. His main research interests are machine learning and data science. He can be contacted at email: nursabilly@alumni.undip.ac.id.