Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 9, No. 3, September 2023, pp. 522-534
ISSN: 2338-3070, DOI: 10.26555/jiteki.v9i3.26341

Strawberry Plant Diseases Classification Using CNN Based on MobileNetV3-Large and EfficientNet-B0 Architecture

Dyah Ajeng Pramudhita, Fatima Azzahra, Ikrar Khaera Arfat, Rita Magdalena, Sofia Saidah
Department of Telecommunication Engineering, Faculty of Electrical Engineering, Telkom University, Bandung, Indonesia

ARTICLE INFO

Article history:

Received June 05, 2023
Revised July 05, 2023
Published July 11, 2023

Keywords:

Strawberry diseases;
Convolutional Neural Network;
MobileNetV3-Large;
EfficientNet-B0;
Image classification

ABSTRACT

Strawberry is a plant that has many benefits and a high risk of being attacked by pests and diseases. Diseases in strawberry plants can cause a decrease in the quality of fruit production and can even cause crop failure. Therefore, a method is needed to assist farmers in identifying the types of diseases in strawberry plants. Currently, the most popular method for identifying types of disease in strawberry plants automatically is using Convolutional Neural Network (CNN). This study proposed a system to be able to detect strawberry plant diseases by classifying the disease based on leaf images with high accuracy using CNN. The problem with using CNN in the previous studies is the heavyweight architectures that are not suitable for deployment on restricted resource devices. The research contribution is implementing the method with lightweight architectures. The proposed system is a CNN algorithm using MobileNetV3-Large and EfficientNet-B0 models to train pre-processed four-class classification datasets, such as healthy leaves, spider mites pest leaves, caterpillars pest leaves, and powdery mildew leaves. Using those architectures helps the parameters and model size keep in the small condition. The result of this study shows that the MobileNetV3-Large model outperforms EfficientNet-B0. The results obtained the best accuracy reaching 92.14% using the MobileNetV3-Large architecture with the hyperparameter optimizer RMSProp, epochs 70, and learning rate 0.0001. The percentage of the evaluation model using MobileNetV3-Large for precision, recall, and F1-Score achieved 92.81%, 92.14%, and 92.25%. Overall, it presents fairly good results and is felicitous to be deployed on low-power and low-storage devices. Furthermore, in future work, it needs to obtain higher accuracy by generating more datasets with different lighting conditions, trying other augmentation techniques, such as lighting transformation, and proposing a better model.

Corresponding Author:

Sofia Saidah, Department of Telecommunication Engineering, Faculty of Electrical Engineering, Telkom University, Bandung, Indonesia
Email: sofiasaidahsfi@telkomuniversity.ac.id

1. INTRODUCTION

Strawberries, or *Fragaria vesca*, are subtropical fruit that is high in nutrients and antioxidants, making them beneficial for reducing the risk of cancer, bad cholesterol, and heart disease [1]. Strawberries have a wide market opportunity [2], so they have high economic value both in terms of agribusiness and in terms of agro-tourism. The multitude of benefits offered by strawberries has increased their production rate in Indonesia. Data from the Central Statistics Agency (BPS) shows that from 2019 to 2021, strawberry production in Indonesia has consistently risen. In 2019, Indonesia produced 7,501 tons of strawberries, which increased to 8,350 tons in 2020 and further rise to 9,860 tons in 2021. When accumulated, this indicates a 31.4% increase in strawberry production in Indonesia. However, upon closer examination, it is evident that the strawberry commodity exhibits fluctuation when assessing its productivity levels from 2019 to 2021, with respective values of 13.8, 12.24, and 14.45 tons/ha, considering the harvested land area that did not witness significant growth.

Several factors contribute to the inconsistent productivity levels of strawberry plants, including attacks by plant pests and human resource limitations [3], [4]. Plant pests or Plant Disturbing Organisms (OPT) are organisms that interfere with plant growth and have the potential to cause damage, such as pests, diseases, and weeds. It can lead to decreased fruit quality and even crop failure. Common diseases of strawberry plants typically exhibit physical symptoms on the plants, especially on the leaves [4]. Although the physical characteristics of diseases exist, strawberry farmers still face difficulty in identifying the disease attacking their crops and determining the appropriate treatment steps due to their lack of knowledge, especially for inexperienced farmers. If not addressed promptly, this can result in poor quality harvest yields, strawberry production losses, and can also have an impact on the farmers' economic losses [5], [6].

Based on the issues that have been outlined, a solution is needed to facilitate strawberry farmers in classifying strawberry plant diseases. Utilizing deep learning and image processing techniques can provide an answer to the problem. Until now, there have been several automated methods to help farmers in identifying the types of diseases that infect strawberry plants. One of the most popular research projects in classifying plant disease is using a deep learning approach, particularly Convolutional Neural Network (CNN), because of its advantages that are time-saving and cost-effective if used [7]. Convolutional Neural Networks also have been widely used for image classification due to their excellent ability to classify images and achieve high accuracy results [8].

There are several previous studies about strawberry plant disease classification. Ramdani *et al.* [9] used CNN to identify diseases in strawberry plants based on images of leaves and compared the accuracy from three different architectures, such as VGG-16, ResNet-50, and G-Net, then present the highest accuracy while implementing the ResNet-50 as an architecture. Xiao *et al.* [10] proposed a model to detect strawberry diseases, such as gray mold, leaf blight, and powdery mildew using ResNet-50 as the architecture. On the other hand, several architectures have also been implemented to achieve optimal accuracy for detecting diseases in different kinds of plants. Hassan *et al.* [7] used CNN to identify plant disease using various plants from the PlantVillage dataset and implement InceptionV3, InceptionResNetV2, MobileNetV2, and EfficientNetB0 architectures. Elfatimi *et al.* [11] present their study using MobileNetV2 architecture for the classification of diseases from beans leaf. Then, Bi *et al.* [12] introduced the improved architecture of MobileNetV3, that was CD-MobileNetV3 to identify corn leaf. According to this previous research, they only use heavyweight architectures which is not suitable for deployment on restricted resource devices due to the large computing power requirement. However, there has been no research that uses lightweight CNN architectures to classify diseases in strawberry plants to make it easy and suitable to be implemented to be deployed in low-power and low-storage devices.

Based on the information provided, it is crucial to accurately identify the types of diseases that affect strawberry plants as early as possible. This is important to ensure that diseases are handled promptly and accurately, preventing them from spreading to other plants in the vicinity and reducing the risk of crop failure. Therefore, we aim to enhance the concept of digital image processing by utilizing Convolutional Neural Networks and comparing the performance between the lightweight architectures, such as MobileNetV3 and the EfficientNetB0 architectures for detecting diseases in strawberry plants.

2. METHODS

This research begins with the method used shown in Fig. 1 which explains the process starting from data collection. Then preprocessing is done by cropping images, resizing images, and performing data augmentation. Next, we train the model that has been created based on the MobileNetV3-Large and EfficientNet-B0 architectures. The models from each architecture are then compared based on the model evaluation results.

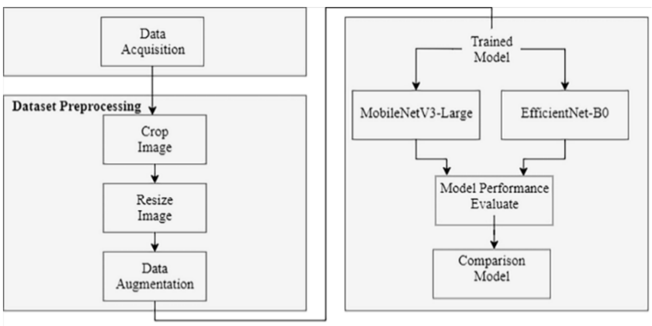

Fig. 1. Proposed method for strawberry plant diseases Classification

2.1. Data Acquisition

The input image used in this study is a dataset in the form of an image of a strawberry plant leaf. The dataset is collected directly from CV. Bumi Agro Technology, West Java, Indonesia. Images were taken during the day with bright lighting conditions, and the shooting distance between the object and the camera is 15 cm. The collected dataset was captured by using a Canon EOS M3 camera with size 6000×3376 pixels and resolution with a depth of 180 dpi. The image used is an RGB color image in JPG format. The dataset was collected based on healthy leaves and leaves affected by pests such as spider mites, caterpillars, and powdery mildew as shown in Fig. 2. Based on the collection of images performed, a total of 1336 images were obtained which were divided into 4 classes, where each class consisted of 334 images. The dataset will then be divided into three parts according to its use, namely training data, validation data, and test data with a ratio of 80%, 10%, and 10% of the total data.

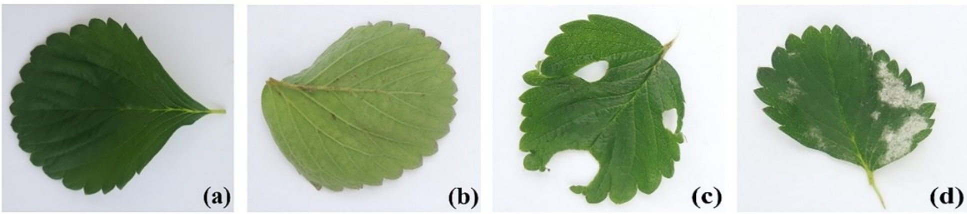

Fig. 2. Sample dataset image of strawberry plant leaves. (a) Healthy; (b) Spider Mite Pests; (c) Caterpillar Pests; (d) Powdery Mildew.

2.2. Data Preprocessing

Preprocessing is a process carried out before entering the stages of the training model. In this process, digital images are processed to obtain better image quality and optimal results when the data is processed during the training model [13]. The preprocessing process is divided into 2 steps, resizing the images and data augmentation: rotation, horizontal flip, width shift, and height shift.

2.2.1. Preprocessing Resize Images

In the preprocessing stage, to be efficient resizing the image becomes a relatively small spatial resolution [16]. In this case, the image size is set to 224 × 224 pixels to get a smaller pixel size and get an ideal image, thus facilitating image processing. Before resizing, the image is cropped first to trim certain areas of the image, in this case trimming some of the existing background so that the strawberry leaf feature can become the dominant object. Fig. 3 shows the original image, cropped image, and resized image.

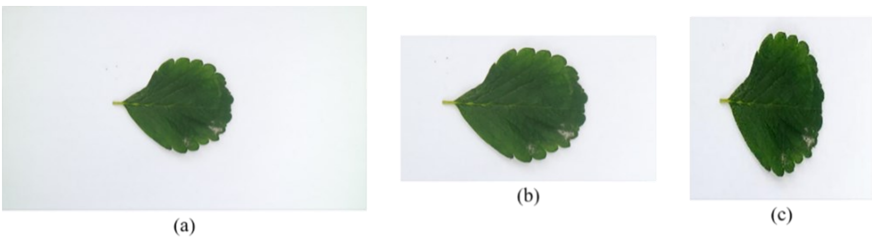

Fig. 3. Preprocessing image. (a) Original image; (b) Cropping image; (c) Resizing image

2.2.2. Data Augmentation

Data augmentation was used as an effort to reduce overfitting. Overfitting is a condition where performance is different between the training and validation or test stages [14]. Overfitting occurs because there are limited training data, contains many distractions [15], and the algorithm is too complicated and requires too many parameters [14], [16]. Overfitting will cause a system to only recognize data trained [17]. If there is new input data, the system will have difficulty classifying the data resulting in low system performance [18]. Several data augmentation techniques are flipping, cropping, rotation, brightness, contrast, colors, saturation, shear, translation, noise injection, etc. [14], [19]. In this study, the augmentation was done using the framework with several techniques including, rotation, flipping, and shifting. Fig. 4 is an example of the results of the data augmentation performed. These three augmentation techniques are basic geometry operation methods. These techniques are used due to their ease of application. In addition, the use of the three techniques can allow the model to learn features from different angles or positions.

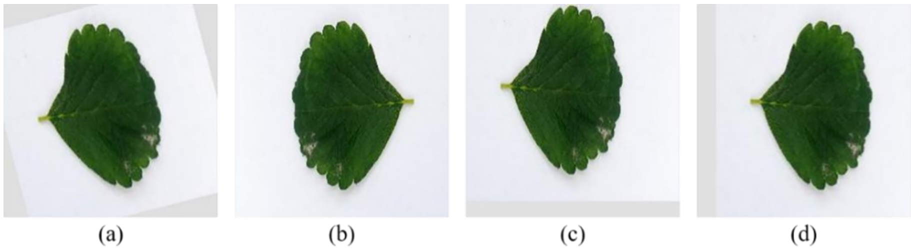

Fig. 4. Augmentation Data. (a) Rotation; (b) Horizontal flip; (c) Width shift; (d) Height shift

2.3. Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep learning algorithm frequently used for image classification [20]. CNN is widely used in image classification because of the many advantages it offers, including reduced computation compared to original neural networks, convolution can simplify calculations significantly without losing the essence of the data, and efficiency in terms of memory and complexity. However, from the advantages offered by CNN, there are also some disadvantages. One of the biggest disadvantages of CNN is the large data set it has to provide. If the dataset is small, an overfitting phenomenon will likely occur which will cause the system to be unable to classify the images correctly.

The work of this algorithm was inspired by neural connection in brain function to learn the features of representation data using the convolution operation [10], [21]. CNN consists of several layers that start from the input layer in the form of an image that has a certain pixel size and end at the output layer which goes through stages in several hidden layers [22]. The hierarchical process in the hidden layer contains a combination of the convolutional layer, pooling layer, and fully connected layer [21], [23]. The illustration of the convolutional neural network for image classification is shown in Fig. 5.

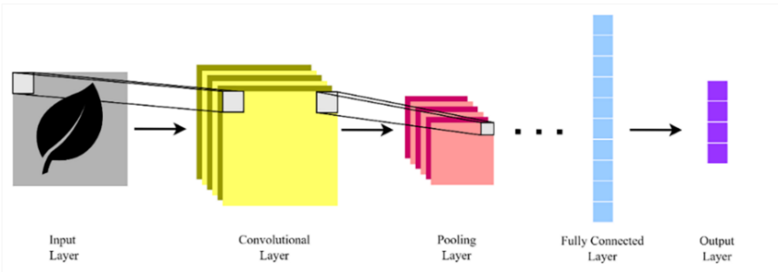

Fig. 5. Block diagram of CNN architecture.

The convolutional layer is an essential layer of CNN architecture [24]. In this layer, a combination of linear and nonlinear operations occurs to create a feature map [25]. The linear operation which is called convolution works by capturing information from input images using a kernel [23]. The output from the convolution operation is then mapped through the activation function (nonlinearity) to decide an output value based on the input value and the type of activation function, such as ReLU, softmax, sigmoid, tanh, etc [20], [23], [26].

The next component of CNN architecture is the pooling layer. This layer simplifies the large-size feature maps into smaller ones [23]. The use of a pooling layer is also beneficial to reduce overfitting [27]. Before heading to the final layer, the output feature maps from the pooling layer should modify it into a one-dimensional vector by implementing a flattening operation [26], [28]. A fully connected layer means that each node is attached to all nodes from the previous layer [29]. The dense layer has a function as a classifier to produce the output by using the activation function to decide the probability of every suitable class [23], [30].

2.3.1. MobileNetV3-Large

The MobileNetV3 architecture is designed for mobile CPUs [31]. MobileNetV3-Large obtains parameters through a network architecture search (NAS) application and is equipped with the NetAdapt algorithm to acquire the optimal number of kernels [32], [33]. Fig. 6 shows the structural architecture of MobileNetV3-Large. Like MobileNetV2, the layer of MobileNetV3 also has a bottleneck block. Each bottleneck block consists of 3 layers that perform 1×1, 3×3, and 1×1 convolution operations [34]. The 1×1 convolution will reduce, then restore the channel dimension of the input image [35]. Then, the 3×3 depthwise convolution will produce a smaller image size with no effect on the channel dimension [34], [36]. However, in

MobileNetV3, this bottleneck concept is combined with Squeeze and Excite [33]. In MobileNetV3, the layer is improved by modifying the use of the ReLU nonlinearity activation function to become h-swish nonlinearity to increase the accuracy of the neural network [12].

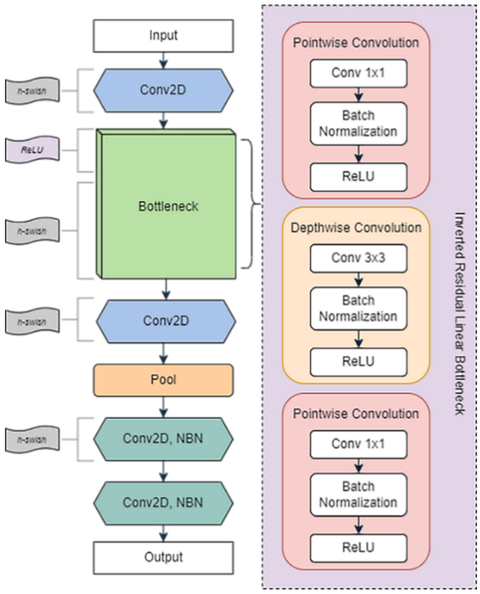

Fig. 6. MobileNetV3-Large architecture.

2.3.2. EfficientNet-B0

EfficientNet is one of the CNN architectures that use the compound scaling method to improve accuracy [37]. The compound scaling method is a method that performs scaling uniformly on all width, depth, and resolution dimensions using a compound coefficient [38]. There are several models on EfficientNet, including EfficientNet B0 to B7. The EfficientNet-B0 architecture is based on the components contained in MobileNetV2, namely the Mobile Inverted Bottleneck Conv (MBConv) with the addition of Squeeze and Excitation (SE) optimization [39], [40]. The use of the MBConv and SE blocks has been shown to increase accuracy with a minimum number of parameters [41], allowing the application to mobile devices [42]. Fig. 7 is an illustration of the layer arrangement in EfficientNet-B0.

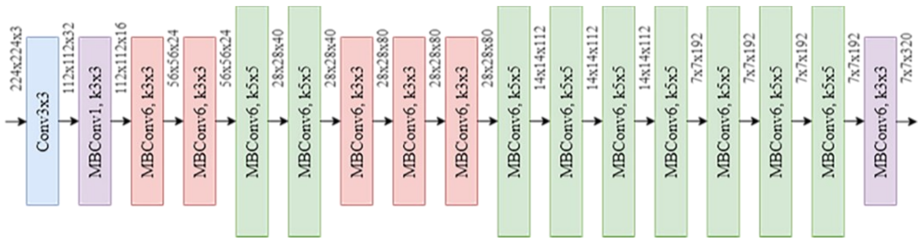

Fig. 7. EfficientNet-B0 architecture.

2.3.3. Model Training

Table 1 shows the proposed models made in deep learning systems by adding a layer after the MobileNetV3-Large and EfficientNet-B0 architectural blocks. The output images from the MobileNetV3-Large and EfficientNet-B0 architecture blocks were then convolved with a 3 × 3 filter and added the ReLU activation function. Then batch normalization is applied to normalize input values at each network layer to increase training accuracy [43]. Then add global average pooling and flatten that will change the image in matrix form to a fully connected layer. Afterward, dropout is used to avoid overfitting by randomly removing some nodes from a fully connected layer [43]. Before the output layer is added, there is a dense layer with a value of 4, where four is the total classification class. The activation function (non-linearity) used in the output layer is softmax. The activation function is used for mapping the input into output by giving the decision to activate or not activate a neuron based on input weight calculations [23].

After that, from the proposed model, a training process was run out to determine the model performance of the MobileNetV3-Large and EfficientNet-B0 architectures. The hyperparameter used in the training process includes batch size (32), epochs (30, 50, 70), learning rate (0.0001, 0.0003, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3), and commonly used optimizer for image classification (Adam, Stochastic Gradient Descent (SGD), Root Mean Square Propagation (RMSProp)) [44].

Table 1. The proposed deep learning model for the classification of strawberry disease.

Layer	Output		Parameter	
	MobileNetV3-Large	**EfficientNet-B0**	**MobileNetV3-Large**	**EfficientNet-B0**
Pre-trained Model	960	1280	2996352	4049571
Conv 2D	32	32	276512	368672
ReLU	32	32	0	0
Batch Normalization	32	32	128	128
Global Average Pooling 2D	32	32	0	0
Flatten	32	32	0	0
Dropout	32	32	0	0
Dense	4	4	132	132
Softmax	4	4	0	0

2.3.4. Model Performance Evaluate

The use of evaluation metrics is necessary to determine the classification ability of the model that has been trained. The most widely used evaluation metric for classification is accuracy [45]. However, it is precise to implement other metrics to optimize the classification evaluation [12]. The evaluation metrics used in this study are accuracy, precision, recall or sensitivity, and F1-Score. Accuracy is the ratio of the correct prediction class over the total number of data evaluated (1). Precision is the accuracy of positive predictions that are true to all positive predictions (2). Recall or Sensitivity is the accuracy in making positive predictions that are true to the entire correct prediction (3). F1-Score is the average value between precision and recall (4). The following are the equations:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1_{Score} = 2 \times \frac{presisi \times recall}{presisi + recall} \tag{4}$$

where TP represents True Positive, the condition when predictions and actual data are mutually positive. TN is a True Negative that indicates the amount of actual data and predictions are negative which means that the classification is correct. The next is FP symbolizes the False Positive, which is a condition when the predicted data is positive, but the actual data is negative. Then, FN represents False Negative which is a condition opposite to FP means that the predicted data is negative, but the actual data is positive.

3. RESULTS AND DISCUSSION

This section presents the results and discussion of this research, including the results of model training, model evaluation, model comparison of MobileNetV3-Large and EfficientNet-B0 architectures, and performance comparison between the results of this research and other research.

3.1. Result

3.1.1. Result Using MobileNetV3-Large

The classification accuracy using the MobileNetV3-Large architecture can be discovered in Table 2. It shows that for the optimizer parameter using Adam, the highest accuracy rate for 30 epochs was in learning rate of 0.01 (91.43%), for 50 epochs in learning rate of 0.03 and 0.1 (91.43%), and for 70 epochs in learning rate of 0.001 (92.14%). Moreover, implementing the SGD parameter as an optimizer gives the highest performance for 30 and 70 epochs with a learning rate of 0.03 (90.00% and 90.71%) and for 50 epochs with a learning rate of 0.01 (90.00%). On the other hand, the usage of the RMSProp parameter optimizer provides the greatest performance for 30 epochs with a learning rate of 0.0001 and 0.01 (91.43%), then for 50 epochs with a learning rate of 0.001 (92.14%), and 70 epochs in learning rate 0.0001, 0.0003, 0.001, and 0.03 (92.14%).

Table 2. Accuracy percentage comparison considering the hyperparameter scenarios for learning rate, epoch, and optimizer, using 32 batch size and MobileNetV3-Large as an architecture.

Parameters \ **Learning Rate**	0.0001	0.0003	0.001	0.003	0.01	0.03	0.1	0.3
Epochs	30	30	30	30	30	30	30	30
	50	50	50	50	50	50	50	50
	70	70	70	70	70	70	70	70
Batch = 32, Optimizer = Adam								
Accuracy (%)	90.71	90.71	90.71	89.29	91.43	84.29	88.57	83.57
	90.00	88.57	90.00	90.71	90.00	91.43	91.43	79.29
	91.43	90.00	92.14	91.43	90.00	87.86	89.93	91.43
Batch = 32, Optimizer = SGD								
Accuracy (%)	76.43	76.43	82.66	85.00	88.57	90.00	88.50	85.71
	75.00	82.14	85.00	89.29	90.00	89.29	85.00	87.86
	80.00	84.29	85.00	90.00	87.14	90.71	80.00	88.57
Batch = 32, Optimizer = RMSProp								
Accuracy (%)	91.43	90.71	90.71	89.29	91.43	84.29	88.57	83.57
	89.29	88.57	92.14	89.29	90.71	90.00	88.57	70.71
	92.14	92.14	92.14	91.43	90.00	92.14	89.29	79.29

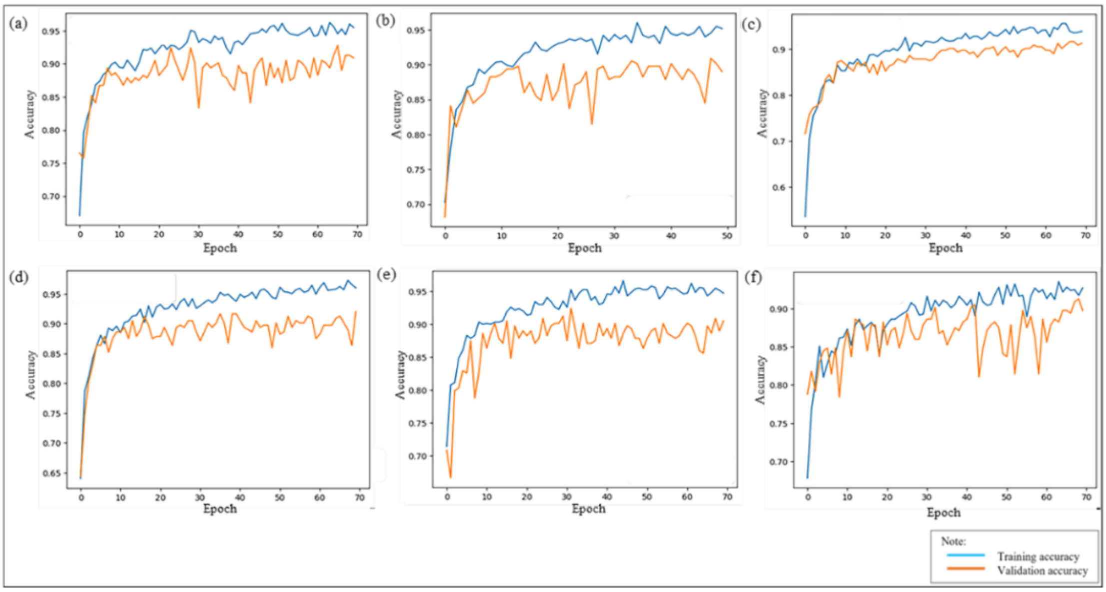

Fig. 8. Graph comparison of the highest accuracy using MobileNetV3-Large. (a) Optimizer = Adam, Epoch = 70, Learning rate = 0.001; (b) Optimizer = RMSProp, Epoch = 50, Learning rate = 0.001; (c) Optimizer = RMSProp, Epoch = 70, Learning rate = 0.0001; (d) Optimizer = RMSprop, Epoch = 70, Learning rate = 0.0003; (e) Optimizer = RMSProp, Epoch = 70, Learning rate = 0.001; (f) Optimizer = RMSProp, Epoch = 70, Learning Rate = 0.03.

Fig. 9 illustrates the evaluation of model performance for Fig. 8(c) using a confusion matrix and the result model evaluation with the classification report for each class. It shows that the highest error prediction of the actual class is indicated by the healthy class which is mistaken for the spider mite pests' class with five images. Whereas the lowest error prediction is indicated by the spider mite pests' class which is mistaken for the healthy class with only one image. Based on this confusion matrix, the highest and lowest recall results are obtained in class spider mite pests (0.97) and healthy (0.86), respectively. Then, for precision, the highest value is achieved by the powdery mildew and caterpillar pests' class (1.0). Nevertheless, the F1-score achieves the highest score in class caterpillar pests (0.97). Globally, the percentage of the evaluation model for precision, recall, and F1-Score achieved 92.81%, 92.14%, and 92.25% respectively.

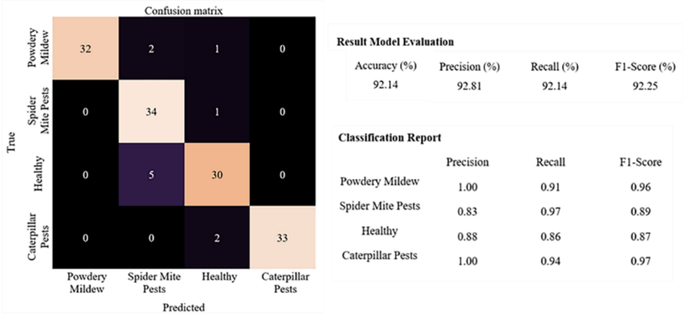

Fig. 9. Confusion and evaluation matrix of the best model performance evaluation using MobileNetV3-Large.

3.1.2. Result Using EfficientNet-B0

Table 3 shows the results of the accuracy percentage using the EfficientNet-B0 architecture. In the optimizer parameter using Adam, the highest accuracy result for 30 epochs was in learning rate of 0.0001 (90.00%), for 50 epochs with a learning rate of 0.001 (88.57%), and for 70 epochs, the highest accuracy was in learning rate of 0.03 (90.71%). While, for using SGD as an optimizer, the highest accuracy for 30 epochs was in a learning rate of 0.003 (83.57%), and the best accuracy for 50 epochs and 70 epochs was in the same learning rate of 0.1 (87.86% and 87.14%). On the other hand, using RMSProp as an optimizer, the highest accuracy result for 30 epochs was in learning rate 0f 0.003 (86.43%), for 50 epochs in learning rate of 0.0003 (84.29%), and for 70 epochs in learning rate of 0.01 (88.57%). From the results of the percentage accuracy using the EfficientNet-B0 architecture, the best accuracy was reached at 90.71% by using Adam as an optimizer for 70 epochs with a learning rate of 0.03. The graph of the best model accuracy is shown in Fig. 10(a).

Table 3. Accuracy percentage comparison considering the hyperparameter scenarios for learning rate, epoch, and optimizer, using 32 batch size and MobileNetV3-Large as an architecture.

Learning Rate / **Parameters**	0.0001	0.0003	0.001	0.003	0.01	0.03	0.1	0.3
Epochs	30 50 70	30 50 70	30 50 70	30 50 70	30 50 70	30 50 70	30 50 70	30 50 70
Batch = 32, Optimizer = Adam								
Accuracy (%)	90.00 83.57 82.86	77.14 85.71 85.71	77.86 88.57 82.14	82.14 86.43 85.00	88.57 84.29 86.43	87.14 78.57 90.71	76.43 78.57 86.43	76.43 85.00 75.00
Batch = 32, Optimizer = SGD								
Accuracy (%)	73.57 70.00 75.00	61.76 77.14 77.14	77.86 80.71 82.14	83.57 81.43 82.14	82.86 82.14 82.86	81.43 82.14 85.71	72.86 87.86 87.14	82.14 85.71 82.86
Batch = 32, Optimizer = RMSProp								
Accuracy (%)	82.14 83.57 85.00	84.29 87.14 86.43	82.14 77.86 85.71	86.43 84.29 83.57	80.00 82.14 88.57	81.43 86.43 85.71	77.14 86.43 80.71	58.33 76.43 83.57

Analyzing the confusion matrix with the classification report of each class, Fig. 10(b) shows that of the errors between actual and predicted classes, the healthy class has the highest error rate. The healthy class was mistakenly divided into the powdery mildew class with two images, further divided into the spider mites' pests' class with four images, and further divided into the caterpillar pests class with four images. While the class that has the highest success rate in classifying images is the caterpillar pests class. From the results of the confusion matrix analysis, the caterpillar class has the highest recall (1.00), and the lowest recall is found in the healthy class (0.71). In terms of precision, the highest precision value is found in the healthy class (0.94). Meanwhile, in terms of the F1-score, the powdery mildew class has the highest F1-score (0.96). Overall, the percentage of model evaluation for precision, recall, and F1-score reached 92.65%, 90.00%, and 90.37%.

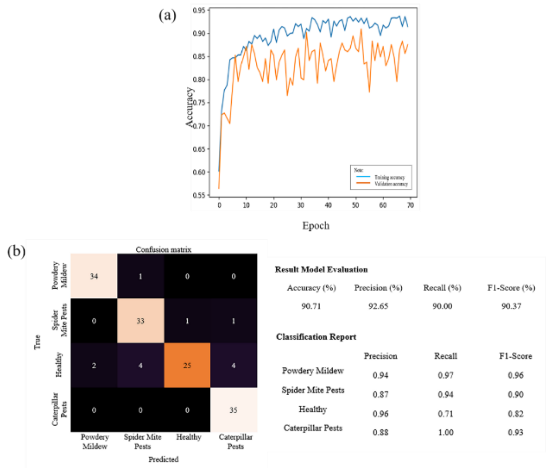

Fig. 10. (a) Graph of the best model accuracy; (b) Confusion and evaluation matrix of the best model performance evaluation using EfficientNet-B0.

3.2. Discussion

In general, the highest accuracy score obtained for each MobileNetV3-Large and EfficientNet-B0 architecture is 92.14% and 90.17% respectively. From these results, we can notice that the MobileNetV3-Large model outperforms EfficientNet-B0 in strawberry disease classification. The MobileNetV3-Large architecture reaches better classification accuracy by utilizing 1×1 convolution from bottleneck residual block to reduced parameter usage while maximizing the performance of the depth wise and pointwise convolutional. Conversely, the EfficientNet-B0 operates MBConv which allows a larger number of channels and parameters by expansion convolution to achieve higher accuracy from depth wise convolution. From the results of this study, the best model was reached by using MobileNetV3-Large architecture, which contains RMSProp as an optimizer for 70 epochs with a learning rate of 0.0001. RMSprop works better in learning because it uses the sum of squared gradients with a decay rate to reduce training time. Using a low learning rate, it will make the model more precise in learning. The larger the epoch used, meaning that the model performs more steps of training processes so that it can recognize the characteristics of each classification class better. These results have achieved fairly high accuracy and can predict well as seen from the evaluation of the confusion matrix shown in Fig. 9. Table 4 is a comparison of the model performance of several studies related to plant disease detection. According to the results, not all references insert the parameter size of the model and the model size in the flite file.

Table 4. Performance comparison of related study on plant-disease detection.

Reference	Plant Species	Dataset (Number of Images)	CNN Architectures	Best Accuracy	Params	Model Size (MB)
This Study	Strawberry	1336	MobileNetV3-Large and EfficientNet-B0	92.14% in MobileNetV3-Large	3.2 M	12.4
Ramdani *et al* [9]	Strawberry	3600	ResNet-50, VGG-16, and GoogLeNet	98.09% in ResNet-50	N/A	N/A
Xiao *et al* [10]	Strawberry	792 for original images and 1306 for feature images	ResNet-50, VGG-16, and GoogLeNet	99.60% in ResNet-50 using feature images	N/A	N/A
Hassan *et al* [7]	14 Kinds	54305	InceptionV3, InceptionResNetV2, MobileNetV2, and EfficientNet-B0	99.56% in EfficientNet-B0	5 M	N/A
Elfatimi *et al* [11]	Beans	1296	MobileNet	92%	N/A	N/A
Bi *et al* [12]	Corn	25167	CD-MobileNetV3 based on MobileNetV3	98.23%	4.36 M	N/A

This study is distinguished from previous studies when comparing the fewer in number size of the model parameters and the smaller model file size. So, it is felicitous to be deployed on low-power and low-storage devices. Nevertheless, it shows that the best performance of this study is still lacking when compared to the accuracy results of other studies. This can be caused by several factors, including the proposed model that is not good enough to carry out the training process. In addition, this can happen because the image data used is still lacking, so it is necessary to try other augmentation techniques to get higher classification results. Since the sample images were only captured under bright lighting conditions, when classifying an image with different lighting, it will affect the result of the classification, because the model does not recognize image features properly. Based on experiments conducted in deep learning applications, several solutions that can improve performance include increasing training time, adding regularization, and increasing the dataset by trying other augmentation techniques, such as lighting transformations to provide various lighting conditions by adjusting the brightness, contrast, saturation, etc. [23].

4. CONCLUSION

This paper presents an automatic model using the deep learning CNN algorithm to identify the strawberry leaf plant disease, such as powdery mildew leaf, spider mite pests leaf, healthy leaf, and caterpillar pests leaf by implementing MobileNetV3-Large and EfficientNet-B0 as an architecture. The collected dataset from the field with a total of 1336 images were then processed in augmentation. After that, the training was executed and several different hyperparameter scenarios enforce to achieve high performance in classification. Eventually, the highest accuracy reached 92.14% using MobileNetV3-Large with the hyperparameter optimizer RMSProp, epochs 70, and learning rate 0.0001. For this model, the precision, recall, and F1 scores reached 92.81%, 92.14%, and 92.25%. The MobileNetV3-Large model outperforms EfficientNet-B0 in strawberry disease classification due to the help of depth wise and pointwise convolutional layers.

Overall, it presents fairly good results in classifying strawberry leaf plant disease and implementing due to felicitous to be deployed on low-power and low-storage devices. The confusion matrix results on the MobileNetV3-Large architecture show better results than the confusion matrix on the EfficientNet-B0 architecture. This can be seen from image testing on the MobileNetV3-Large model that classifies more strawberry leaf images correctly according to the labels given. However, it still has deficiencies in terms of feature identification when using images with different or darker light conditions. Furthermore, in future work, we need to obtain higher accuracy by generating more datasets with different environments and lighting conditions and also enhancing another class of strawberry plant disease. Then, to propose a better model, we need to try other augmentation techniques, such as lighting transformations to provide various lighting conditions.

Acknowledgments

The authors would like to acknowledge all support from the Department of Telecommunication Engineering, Telkom University, and all parties involved in completing this paper.

REFERENCES

[1] X. Chen *et al.*, "Metabolomic and transcriptomic analysis reveals the molecular mechanism by which blue light promotes lutein synthesis in strawberry," *J Integr Agric*, 2023, https://doi.org/10.1016/j.jia.2023.04.002.

[2] E. Elhariri, N. El-Bendary, and S. M. Saleh, "Strawberry-DS: Dataset of annotated strawberry fruits images with various developmental stages," *Data Brief*, vol. 48, 2023, https://doi.org/10.1016/j.dib.2023.109165.

[3] S. Lahiri, H. A. Smith, M. Gireesh, G. Kaur, and J. D. Montemayor, "Arthropod Pest Management in Strawberry," *Insects*, vol. 13, no. 5, 2022, https://doi.org/10.3390/insects13050475.

[4] D. E. Kusumandari, M. Adzkia, S. P. Gultom, M. Turnip, and A. Turnip, "Detection of Strawberry Plant Disease Based on Leaf Spot Using Color Segmentation," in *Journal of Physics: Conference Series*, vol. 1230, no. 1, p. 012092, 2019, https://doi.org/10.1088/1742-6596/1230/1/012092.

[5] H. Kim and D. Kim, "Deep-Learning-Based Strawberry Leaf Pest Classification for Sustainable Smart Farms," *Sustainability (Switzerland)*, vol. 15, no. 10, 2023, https://doi.org/10.3390/su15107931.

[6] Y. Guo-feng, Y. Yong, H. Zi-kang, Z. Xin-yu, and H. Yong, "A rapid, low-cost deep learning system to classify strawberry disease based on cloud service," *Journa of Integrative Agriculture*, vol. 21, no. 2, pp. 460–473, 2022, https://doi.org/10.1016/S2095-3119(21)63604-3.

[7] S. M. Hassan, A. K. Maji, M. Jasiński, Z. Leonowicz, and E. Jasińska, "Identification of plant-leaf diseases using CNN and transfer-learning approach," *Electronics (Switzerland)*, vol. 10, no. 12, Jun. 2021, https://doi.org/10.3390/electronics10121388.

[8] M. Xin and Y. Wang, "Research on image classification model based on deep convolution neural network," *EURASIP J Image Video Process*, vol. 2019, no. 1, 2019, https://doi.org/10.1186/s13640-019-0417-8.

[9] A. Ramdani and S. Suyanto, “Strawberry Diseases Identification from Its Leaf Images Using Convolutional Neural Network,” in *IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, pp. 186–190, 2021, https://doi.org/10.1109/IAICT52856.2021.9532573.
[10] J. R. Xiao, P. C. Chung, H. Y. Wu, Q. H. Phan, J. L. A. Yeh, and M. T. K. Hou, “Detection of strawberry diseases using a convolutional neural network,” *Plants*, vol. 10, no. 1, pp. 1–14, 2021, https://doi.org/10.3390/plants10010031.
[11] E. Elfatimi, R. Eryigit, and L. Elfatimi, “Beans Leaf Diseases Classification Using MobileNet Models,” *IEEE Access*, vol. 10, pp. 9471–9482, 2022, https://doi.org/10.1109/ACCESS.2022.3142817.
[12] C. Bi, S. Xu, N. Hu, S. Zhang, Z. Zhu, and H. Yu, “Identification Method of Corn Leaf Disease Based on Improved Mobilenetv3 Model,” *Agronomy*, vol. 13, no. 2, 2023, https://doi.org/10.3390/agronomy13020300.
[13] T. Singh, K. Kumar, and S. S. Bedi, “A review on artificial intelligence techniques for disease recognition in plants,” in *IOP Conference Series: Materials Science and Engineering*, 2021, https://doi.org/10.1088/1757-899X/1022/1/012032.
[14] K. Alomar, H. I. Aysel, and X. Cai, “Data Augmentation in Classification and Segmentation: A Survey and New Strategies,” *J Imaging*, vol. 9, no. 2, 2023, https://doi.org/10.3390/jimaging9020046.
[15] I. Betto, R. Hatano, and H. Nishiyama, “Distraction detection of lectures in e-learning using machine learning based on human facial features and postural information,” *Artif Life Robot*, vol. 28, no. 1, pp. 166–174, 2023, https://doi.org/10.1007/s10015-022-00809-z.
[16] X. Ying, “An Overview of Overfitting and its Solutions,” in *Journal of Physics: Conference Series*, vol. 1168, p. 022022, 2019, https://doi.org/10.1088/1742-6596/1168/2/022022.
[17] P. Rattan, D. D. Penrice, and D. A. Simonetto, “Artificial Intelligence and Machine Learning: What You Always Wanted to Know but Were Afraid to Ask,” *Gastro Hep Advances*, vol. 1, no. 1, pp. 70–78, 2022, https://doi.org/10.1016/j.gastha.2021.11.001.
[18] S. Pavlitskaya, J. Oswald, and J. M. Zöllner, “Measuring Overfitting in Convolutional Neural Networks using Adversarial Perturbations and Label Noise,” in *IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1551–1559, 2022, https://doi.org/10.1109/SSCI51031.2022.10022094.
[19] A. Mumuni and F. Mumuni, “Data augmentation: A comprehensive survey of modern approaches,” *Array*, vol. 16, 2022, https://doi.org/10.1016/j.array.2022.100258.
[20] I. Kandel, M. Castelli, and A. Popovič, “Comparative Study of First Order Optimizers for Image Classification Using Convolutional Neural Networks on Histopathology Images,” *J Imaging*, vol. 6, no. 9, 2020, https://doi.org/10.3390/JIMAGING6090092.
[21] S. Sa’idah, I. P. Y. N. Suparta, and S. R. Fauziah, “Efficient Scaling of Convolutional Neural Network for Detecting dan Classifying Pneumonia Disease,” in *4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pp. 581-586, 2021, https://doi.org/10.1109/ISRITI54043.2021.9702779.
[22] Y. H. Liu, “Feature Extraction and Image Recognition with Convolutional Neural Networks,” in *Journal of Physics: Conference Series*, vol. 1087, p. 062032, 2018, https://doi.org/10.1088/1742-6596/1087/6/062032.
[23] L. Alzubaidi *et al.*, “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” *J Big Data*, vol. 8, no. 1, 2021, https://doi.org/10.1186/s40537-021-00444-8.
[24] A. Ajit, K. Acharya, and A. Samanta, “A Review of Convolutional Neural Networks,” in *International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, pp. 1–5, 2020, https://doi.org/10.1109/ic-ETITE47903.2020.049.
[25] A. Raju and S. Thirunavukkarasu, “Convolutional Neural Network Demystified for a Comprehensive Learning with Industrial Application,” in *Dynamic Data Assimilation - Beating the Uncertainties*, pp/ 1-17, 2020, https://doi.org/10.5772/intechopen.92091.
[26] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, 2018, https://doi.org/10.1007/s13244-018-0639-9.
[27] R. Sangeetha and M. Mary Shanthi Rani, “Tomato leaf disease prediction using convolutional neural network,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 1, pp. 1348–1352, 2019, https://doi.org/10.35940/ijitee.L3776.119119.
[28] A. Zafar *et al.*, “A Comparison of Pooling Methods for Convolutional Neural Networks,” *Applied Sciences (Switzerland)*, vol. 12, no. 17, 2022, https://doi.org/10.3390/app12178643.
[29] A. Soni, R. Koner, and V. G. K. Villuri, “Fusion of Dual-Scale Convolution Neural Network for Urban Building Footprints,” *Ain Shams Engineering Journal*, vol. 13, no. 3, 2022, https://doi.org/10.1016/j.asej.2021.10.017.
[30] W. Tasya, S. Sa’idah, F. Nurfajar, and B. Hidayat, “Breast Cancer Detection Using Convolutional Neural Network with EfficientNet Architecture,” in *IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, pp. 1–6, 2022, https://doi.org/10.1109/APWiMob56856.2022.10014095.
[31] L. Zhao and L. Wang, “A new lightweight network based on MobileNetV3,” *KSII Transactions on Internet and Information Systems*, vol. 16, no. 1, pp. 1–15, 2022, https://doi.org/10.3837/tiis.2022.01.001.
[32] J. Chai, H. Zeng, A. Li, and E. W. T. Ngai, “Deep learning in computer vision: A critical review of emerging techniques and application scenarios,” *Machine Learning with Applications*, vol. 6, p. 100134, 2021, https://doi.org/10.1016/j.mlwa.2021.100134.
[33] A. Howard *et al.*, “Searching for MobileNetV3,” in *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1314–1324, 2019, https://doi.org/10.1109/ICCV.2019.00140.

[34] W. Glegoła, A. Karpus, and A. Przybyłek, "MobileNet family tailored for Raspberry Pi," in *Procedia Computer Science*, Elsevier B.V., pp. 2249–2258, 2021, https://doi.org/10.1016/j.procs.2021.08.238.
[35] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, "Review of image classification algorithms based on convolutional neural networks," *Remote Sensing*, vol. 13, no. 22, 2021, https://doi.org/10.3390/rs13224712.
[36] J. Yun *et al.*, "Real-Time Target Detection Method Based on Lightweight Convolutional Neural Network," *Front Bioeng Biotechnol*, vol. 10, 2022, https://doi.org/10.3389/fbioe.2022.861286.
[37] H. Alhichri, A. S. Alswayed, Y. Bazi, N. Ammour, and N. A. Alajlan, "Classification of Remote Sensing Images Using EfficientNet-B3 CNN Model with Attention," *IEEE Access*, vol. 9, pp. 14078–14094, 2021, https://doi.org/10.1109/ACCESS.2021.3051085.
[38] I. Papoutsis, N. I. Bountos, A. Zavras, D. Michail, and C. Tryfonopoulos, "Benchmarking and scaling of deep learning models for land cover image classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 195, pp. 250–268, 2023, https://doi.org/10.1016/j.isprsjprs.2022.11.012.
[39] M. Tan and Q. V Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *36th International Conference on Machine Learning*, pp. 6105–6114, 2019, https://doi.org/10.48550/arXiv.1905.11946.
[40] J. S. Suri *et al.*, "UNet Deep Learning Architecture for Segmentation of Vascular and Non-Vascular Images: A Microscopic Look at UNet Components Buffered with Pruning, Explainable Artificial Intelligence, and Bias," *IEEE Access*, vol. 11, pp. 595–645, 2023, https://doi.org/10.1109/ACCESS.2022.3232561.
[41] D. Stamoulis *et al.*, "Single-Path Mobile AutoML: Efficient ConvNet Design and NAS Hyperparameter Optimization," *IEEE J Sel Top Signal Process*, vol. 14, no. 4, pp. 609–622, 2020, https://doi.org/10.1109/JSTSP.2020.2971421.
[42] B. Kuriakose, R. Shrestha, and F. E. Sandnes, "SceneRecog: A Deep Learning Scene Recognition Model for Assisting Blind and Visually Impaired Navigate using Smartphones," in *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, pp. 2464–2470, 2021, https://doi.org/10.1109/SMC52423.2021.9658913.
[43] C. Garbin, X. Zhu, and O. Marques, "Dropout vs. batch normalization: an empirical study of their impact to deep learning," *Multimed Tools Appl*, vol. 79, no. 19–20, pp. 12777–12815, 2020, https://doi.org/10.1007/s11042-019-08453-9.
[44] E. Hassan, M. Y. Shams, N. A. Hikal, and S. Elmougy, "The effect of choosing optimizer algorithms to improve computer vision tasks: a comparative study," *Multimed Tools Appl*, vol. 82, no. 11, pp. 16591-16633, 2022, https://doi.org/10.1007/s11042-022-13820-0.
[45] S. A. Hicks *et al.*, "On evaluation metrics for medical applications of artificial intelligence," *Sci Rep*, vol. 12, no. 1, 2022, https://doi.org/10.1038/s41598-022-09954-8.

BIOGRAPHY OF AUTHORS

Dyah Ajeng Pramudhita is a final-year Telecommunication Engineering student at the Faculty of Electrical Engineering at Telkom University, Bandung, Indonesia. Her current research interests are image processing and back-end web development. Email: dyahajengp@student.telkomuniversity.ac.id.

S

Fatima Azzahra is a final-year student at the Department of Telecommunication Engineering, Telkom University, Bandung. She is interested in researching image processing, data communication, and telecommunication transmission. Email: fatimaazzahra@student.telkomuniversity.ac.id.

Ikrar Khaera Arfat is currently a final-year Telecommunication Engineering student at Telkom University. He has a passion for design and information technology including mobile development and UI/UX especially in Android development. Email: ikrarkhaeraa@student.telkomuniversity.ac.id.

Rita Magdalena is currently a lecturer in School of Electrical Engineering, Telkom University. Her research interest is information signal processing, especially in image processing, audio processing, and biomedical engineering. Email: ritamagdalena@telkomuniversity.ac.id.

Sofia Saidah received the B.S. and M.S. degree from Telecommunication Engineering, Telkom Institute of Technology, Bandung, Indonesia in 2012 and 2014 respectively. She is currently a lecturer in School of Electrical Engineering Telkom University. Her research interest includes image processing, audio processing, biomedical engineering, steganography, and watermarking. Email: sofiasaidahsfi@telkomuniversity.ac.id.