Plant Disease Identification Using EfficienNet

Widi Hastomo[1], Adhitio Satyo Bayangkari Karno[2], Nada Kamilia[3], Nia Yuningsih[4], Agita Tunjungsari[5]
[1]Department of Information Technology, Ahmad Dahlan Institute of Technology and Business, Jakarta, Indonesia
[2,4]Department of Information System, Faculty of Engineering, Gunadarma University, Depok, Indonesia
[3]Department of Information System, STMIK Jakarta, STI&K, Indonesia
[5]Department of Psychology, Faculty of Psychology Science, Gunadarma University, Depok
* Corresponding Author

ARTICLE INFO

Article history:

Received July 28, 2021
Revised August 28, 2021
Accepted September 28, 2021

Keywords:

EfficientNet;
Convolutional Neural Network;
Deep Learning.

ABSTRACT

Manually detecting plant diseases is time-consuming and prone to mistakes. Artificial intelligence (AI) and computer vision can be used to identify plant illnesses early, reducing their negative impacts while also overcoming some of the limitations of constant human monitoring. This experiment aims to utilize the Convolution Neural Network (CNN) algorithm to identify plant leaf diseases, to optimize four CNN algorithms to identify plant leaf diseases, to make it easier for users to identify plant leaf diseases quickly and accurately. We propose using a deep learning architecture based on a recent CNN algorithm called EfficientNetB3, B4 and B5 on 66,556 plant disease leaf images sourced from Kaggle.com. The training phase with 57,067 data train images and 3,170 validation data images produces a model. For model testing, the testing phase was carried out with 3,171 image data tests, the overall test results yielded excellent accuracy and f1-scores for the three architectures, namely EfficientNetB3 0.9890%, EfficientNetB4 0.9912%, EfficientNetB5 0.9905%. Agriculture is anticipated to be the backbone of Indonesia because it is connected to both the national economy and the wellbeing of the populace

Cite Article:

Widi Hastomo, Adhitio Satyo Bayangkari Karno, Nada Kamilia, Nia Yuningsih, Agita Tunjungsari, "Plant Disease Identification Using EfficienNet" *Indonesian Community on Optimization and Computer Application*, vol. 1, no. 1, pp. xx-xx, 2022, doi: .

1. INTRODUCTION

Farmers have traditionally used a variety of approaches to safeguard their crops from pests and diseases. To get rid of insects, people have created hardy plant kinds, poisoned plants with herbal ingredients, removed insects from plants, and rotated crops [1]. Nowadays, the majority of farmers, especially those in industrialized nations, use insecticides to manage pests. The term "pest" can refer to everything from insects to disease- carrying germs, viruses, and fungus, as well as animals like rats, birds, and weeds. Chemical use has significantly reduced crop losses and prices [2].

Farmers' understanding of field management has been considerably altered by digital integration thanks to cutting-edge technologies like intelligent computers, robotics, drones, or sensors on agricultural machinery [3]. With the use of this technology, data scientists and agronomists may more effectively manage fields and meet new difficulties (such as fungus attack detection, yield forecast, follow-up spraying, etc). [4]. In order to satisfy their demands and assist them in maximizing their yields using data and task automation, this new approach necessitates technical assistance from farmers [5].

Deep learning algorithms have shown success in a variety of fields thanks to improved computational performance and big data sets [6], [7]. They have been able to circumvent the drawbacks of conventional extraction approaches because of their expressive capabilities with data. This enables engines to carry out intricate processing on massive amounts of data, producing predictions with encouraging outcomes. To accomplish this, deep neural network methods have been applied in numerous agricultural activities in recent years [8],[9],[10]. As a result, agricultural researchers and specialists pay them more attention [11]. The ecological shift of farming systems is complicated by issues posed by precision agriculture [12], such as lowering energy usage, increasing production quality while decreasing intranet use, and retaining human decision-making.

Research related to the identification of diseases in rice plants uses the MobileNet-V2 and ImageNet methods which produce an accuracy of up to 98.48% [13]. Research on diseases of apple leaves uses the DenseNet121 and EfficientNetB7 methods with an accuracy rate of up to 96.25% [14]. Research on diseases on wheat leaves conducted by [15] used the VGG-16, Inception-v3, ResNet-50, DenseNet-121, EfficentNet-B6, ShuffleNet-v2 and MobileNetV3 algorithms with an accuracy rate of up to 92.5% . Identification of diseases on cucumber leaves uses the EfficientNet-B4-Ranger algorithm with an accuracy rate of up to 97% which has been carried out by [16]. Research on disease identification techniques on tomato leaves used genetic algorithms and VGG16, ResNet50, EfficientNetB0, EfficientNetB1, EfficientNetB2, EfficientNetB3 and EfficientNetB4 with an accuracy rate of up to 98.1% [17].

Identification of tomato leaf diseases uses the EfficientNetB0, EfficientNetB4 and EfficientNetB7 algorithms with an accuracy rate of 99.89% [10]. EfficientNetV1 and EfficientNetV2 algorithms for detecting diseases in fruit plants with an accuracy of up to 74%. Research on the identification of leaf diseases in cucumbers used the VGG16, ResNet50, ResNet101, Densenet201 algorithms with an accuracy rate of 98.4% [18]. Research on identifying diseases on plant leaves uses the ResNet50, DenseNet169, EfficientNet and FL-EfficientNet algorithms with an accuracy of up to 99.7% [19]. Research conducted by [15] used the SE-ResNet50, EfficientNet, AlexNet, VGG19, and Inception V3 algorithms with an accuracy rate of 97.24%. Research conducted by [20] identified leaf diseases in tomatoes and chilies using the CNN Deep Attention Dense algorithm with an accuracy rate of 97.33%.

Based on the explanation above, this study aims to identify plant diseases, in which Plants' growth is influenced by environmental, climatic, and soil variables, and diseases have endangered them, substantially lowering agricultural output and quality. The CNN method is a common and highly optimal method used in classifying images, for comparison, the architecture, EfficientNet (B3, B4, and B5) is used. The results of the training and testing process are high-accuracy matrix weight models that can be used to predict disease through plant leaf images.

2. METHODS

Broadly speaking, this research consists of 4 stages, namely: dataset preprocessing, training, testing and results to be evaluated [21]. The dataset preprocessing stage consists of several subprocesses that aim to prepare the data so that it can be used in a format according to the CNN architecture in the training stage. Training is the stage where the machine will carry out the learning process using train data and evaluate learning outcomes using validation data to update the previous weight values (updated weights) . This update weight value will be used in the next iteration process, until a high percentage accuracy and low loss is achieved [22]. The final weight value with the maximum accuracy value and minimum loss is referred to as the training result model. Furthermore, the training result model will be used in the testing process using test data, until the final result is obtained in the form of a confusion matrix , can be shown in Figure 1.

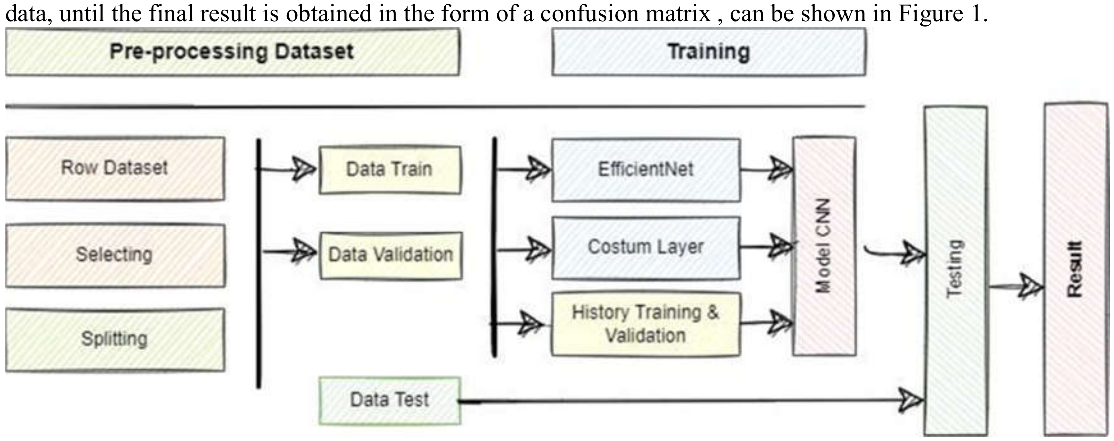


Fig. 1. Propose method

3. RESULTS AND DISCUSSION

3.1. Preprocessing Dataset

The dataset was obtained from www.kaggle.com [23], containing leaf images from various plants (Figure 2) which are grouped in folders according to the type of plant disease. Dataset preprocessing is the initial process for preparing data so that it can be used properly and is suitable for the next stage (training and testing stages). The raw dataset consists of 66,556 images in 58 plant disease folders (Figure 3). This raw dataset seems to have an unbalanced number of images in each folder. To overcome the data imbalance, the process of selecting folders is carried out, only folders that have more than 500 images. The raw data which was originally 66,556 images in 58 folders due to the selecting process was reduced to 63,408 images in 29 disease folders. Then the 63,408 image data was splitted into 57,067 (90%) training data, 3,170 (5%) validation data and 3171 (5%) testing data (Figure 3).

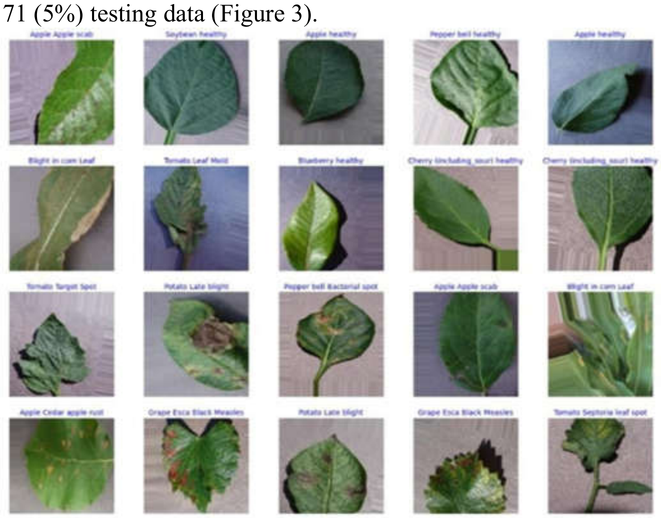


Fig. 2. Image Dataset

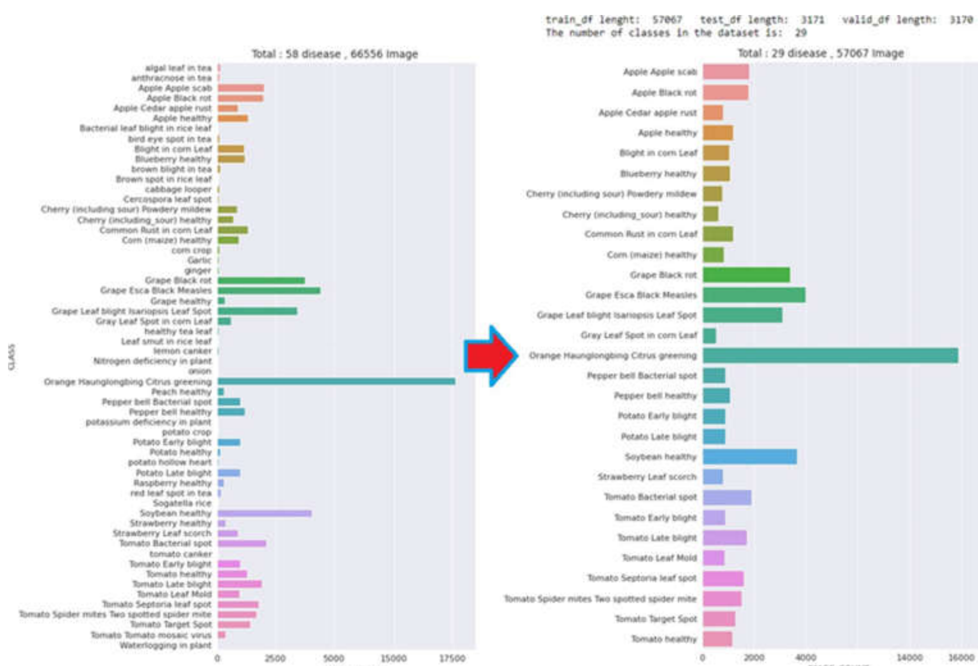


Fig. 3. Row dataset (left) and preprocessing results for training data (right)

3.2. Training and Testing

In the training phase, the training process is carried out using data training and the validation process uses data validation. In each epoch the validation process is carried out after the training process, comparing with the target value to obtain the deviation value as the basis for measuring the weight matrix to update. The latest weight matrix (after updating) will be used for the next epoch training process. This process is repeated until it reaches the desired epoch value. This training and validation process uses the base model 3 of the EfficientNet architecture (B3, B4, B5) which is in the tensorflow-keras library and the custom layer architecture. During the iteration process, accuracy and loss values will be recorded in the form of history. This process runs in the desired epoch and achieves high accuracy and low loss. The results in the form of a weight matrix are expressed as a model of training results. This model is used for the next process, namely the process of testing or testing using test data.

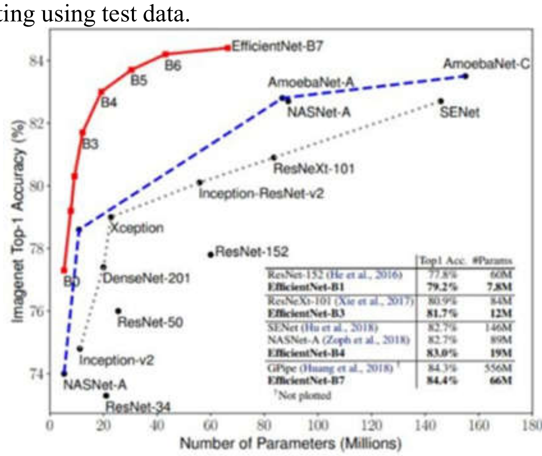


Fig. 4. Model Size vs ImageNet Accuracy

Many CNN architectures are designed to increase high accuracy scores by adding thick layers that require higher computational requirements. The EfficientNet architecture was built not solely by adding layers to increase accuracy, but also considering the minimal level of efficiency. As can be seen from the graph (Figure 4) compared to other architectures, EfficientNet is able to achieve high accuracy scores using a minimal number of parameters compared to other architectures. Taking into account the existing computer capabilities, this study chose EfficietNet with 3 versions (B3, B4, B5) as a comparison. In addition, this architecture was chosen because EfficientNet has demonstrated strong capabilities in image classification for remote sensing [2]. In a study conducted by Mingxing Tan [1], EfficientNet versions B3, B4, and B5 using ImageNet training data were able to achieve an accuracy of 81.6%, 82.9%, 83.6% respectively with a total of 12M, 19M, and 30M computational parameters. (Figure 5)

Model	Top-1 Acc.	Top-5 Acc.	#Params	Ratio-to-EfficientNet	#FLOPs	Ratio-to-EfficientNet
EfficientNet-B0	**77.1%**	**93.3%**	**5.3M**	**1x**	**0.39B**	**1x**
ResNet-50 (He et al., 2016)	76.0%	93.0%	26M	4.9x	4.1B	11x
DenseNet-169 (Huang et al., 2017)	76.2%	93.2%	14M	2.6x	3.5B	8.9x
EfficientNet-B1	**79.1%**	**94.4%**	**7.8M**	**1x**	**0.70B**	**1x**
ResNet-152 (He et al., 2016)	77.8%	93.8%	60M	7.6x	11B	16x
DenseNet-264 (Huang et al., 2017)	77.9%	93.9%	34M	4.3x	6.0B	8.6x
Inception-v3 (Szegedy et al., 2016)	78.8%	94.4%	24M	3.0x	5.7B	8.1x
Xception (Chollet, 2017)	79.0%	94.5%	23M	3.0x	8.4B	12x
EfficientNet-B2	**80.1%**	**94.9%**	**9.2M**	**1x**	**1.0B**	**1x**
Inception-v4 (Szegedy et al., 2017)	80.0%	95.0%	48M	5.2x	13B	13x
Inception-resnet-v2 (Szegedy et al., 2017)	80.1%	95.1%	56M	6.1x	13B	13x
EfficientNet-B3	**81.6%**	**95.7%**	**12M**	**1x**	**1.8B**	**1x**
ResNeXt-101 (Xie et al., 2017)	80.9%	95.6%	84M	7.0x	32B	18x
PolyNet (Zhang et al., 2017)	81.3%	95.8%	92M	7.7x	35B	19x
EfficientNet-B4	**82.9%**	**96.4%**	**19M**	**1x**	**4.2B**	**1x**
SENet (Hu et al., 2018)	82.7%	96.2%	146M	7.7x	42B	10x
NASNet-A (Zoph et al., 2018)	82.7%	96.2%	89M	4.7x	24B	5.7x
AmoebaNet-A (Real et al., 2019)	82.8%	96.1%	87M	4.6x	23B	5.5x
PNASNet (Liu et al., 2018)	82.9%	96.2%	86M	4.5x	23B	6.0x
EfficientNet-B5	**83.6%**	**96.7%**	**30M**	**1x**	**9.9B**	**1x**
AmoebaNet-C (Cubuk et al., 2019)	83.5%	96.5%	155M	5.2x	41B	4.1x
EfficientNet-B6	**84.0%**	**96.8%**	**43M**	**1x**	**19B**	**1x**
EfficientNet-B7	**84.3%**	**97.0%**	**66M**	**1x**	**37B**	**1x**
GPipe (Huang et al., 2018)	84.3%	97.0%	557M	8.4x	-	-

Fig. 5. Results of EfficienNet capabilities

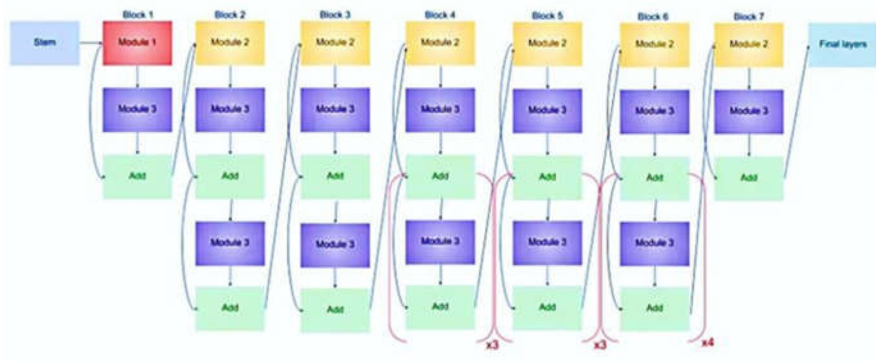


Fig. 6. Architecture for EfficientNet-B3

By looking at the EfficienNet architecture, it can be seen that this architecture is a thick stack of Figure 6 module layers. Training data and validation data will be fed to the EfficienNet architecture. The tensorflow hardware library application is used to simplify the use of EfficienNet-B3 and ImageNet to make it more efficient.

Adding a custom layer after EfficienNet is done to obtain output that is as expected. The custom layer design contains 4 convolutional layers, and 3 FC (fully connected) layers. Using the plot model library, a custom layer model chart can be displayed in png format. The custom layers consist of: input, conv2D, max_pooling, flatten and dense. Each layer has its own function in image classification.

3.3. Conv2D

The output of the previous layer with a pixel size of 256 x 256 x 3 will be the input for the next layer, namely Conv2D (convolution for 2 dimensions). In this Conv2D (layer 2), 16 nodes are placed with a mask size of 3x3, stride = 1, using ReLU type activation, which is applied to each pixel to replace all negative pixel values to zero. To facilitate understanding of the convolution technique, an image with a 6x6 matrix is multiplied by dots with a 3x3 filter matrix to obtain a 3x3 output matrix (Figure 7).

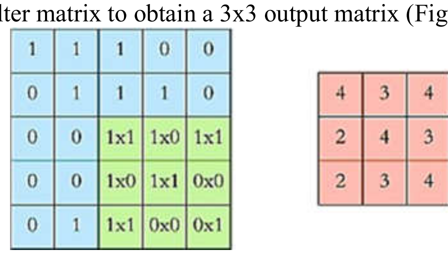


Fig. 7. Convolution

3.4. Max Pooling

Aims to reduce the size of the dimensions (downsampling), so that the calculation process becomes faster and can improve model performance (Lee, Gallagher, & Tu, 2015). This pooling layer generally comes after the conv layer. In Lapsan 3, pooling reduces the dimensions from 256x256x16 to 128x128x16. From smaller 4x4 dimensions to 2x2 dimensions, max pooling can be illustrated in Figure 8.

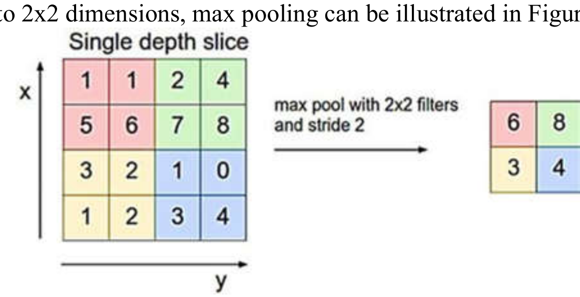


Fig. 8. Max Pooling

3.5. Flatten

This flatten layer is generally placed after the pooling layer in CNN. This layer is the process of changing the matrix into a single vector form. The formation of this single vector is useful as input for the next process, namely classification (dense) (Figure 9).

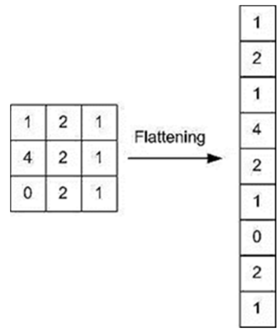


Fig. 9. Flattening

3.6. Dense

Dense or fully connected is part of the classification process, a dense layer is placed after flattening with an output vector format to make it possible to process ordinary neural networks with a number of nodes in each hidden layer (Figure 10).

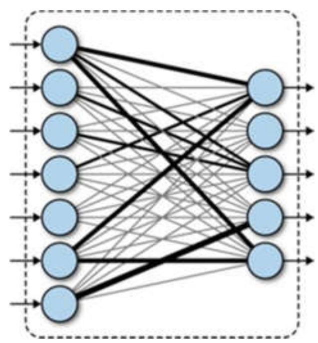


Fig. 10. Flattening

3.7. Accuracy and Loss

From the model that has been formed, training is carried out with epoch 50 (optimization), the results are obtained in the form of history for the 3 architectures. The results of history and training graphs (figure 11,12,13) and summarized in table 1, it can be seen that the larger the version used, the longer the training time will be due to the thicker layer. For the highest accuracy in version B4 (0.9978), accuracy validation is in version B3 (0.9858), the lowest loss is in version B4 and loss validation is in version (0.1946).

Table 1. Table of training results for 3 EfficientNet architectures

EfficientNet	Accuracy	Val_accuracy	Loss	Val_loss	Time
B3	0.9972	**0.9858**	0.1588	**0.1946**	1:24:28
B4	**0.9978**	0.9943	**0.1585**	0.1543	1:34:18
B5	0.9967	0.9924	0.1700	0.1679	2:04:45

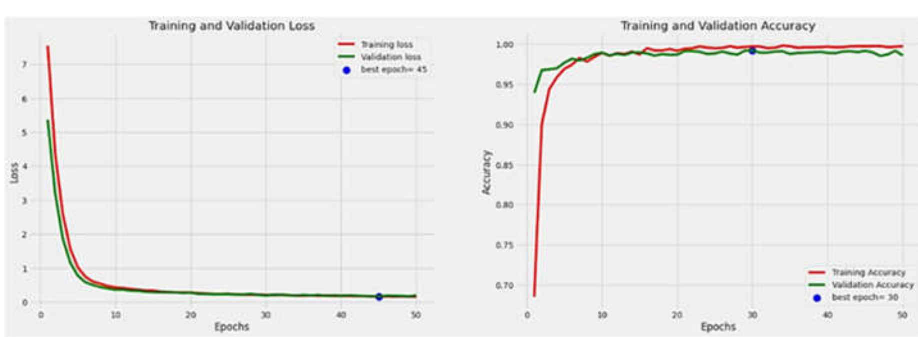


Fig. 11. Chart history training EfficientNetB3

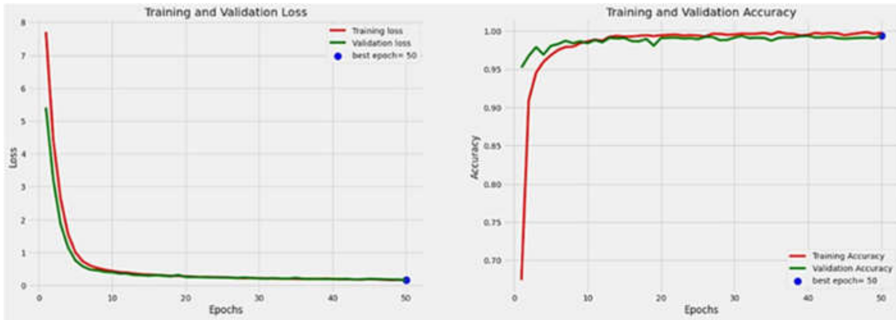


Fig. 12. Chart history training EfficientNetB4

Fig. 13. Chart history training EfficientNetB5

3.8. Confusion Matrix and Classification Report

To calculate the accuracy value of the deep learning training process in classifying data into the desired label, a program is created using the Seaborn Heatmap library in Python from the results of the predicted value and the actual value, resulting in a confusion matrix (Figure 14,15,16).

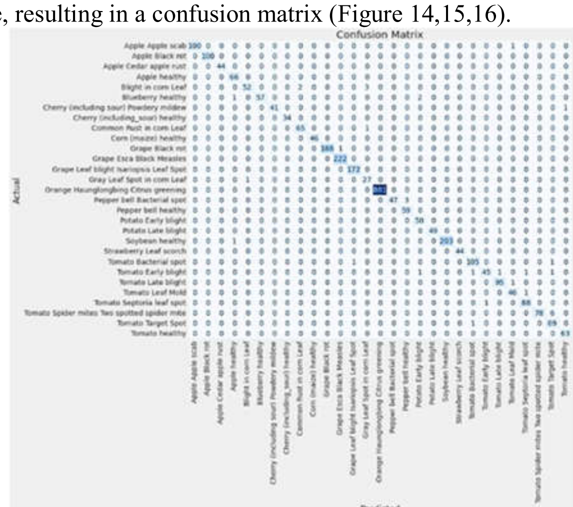


Fig. 14. Confusion matrix EfficientNetB3

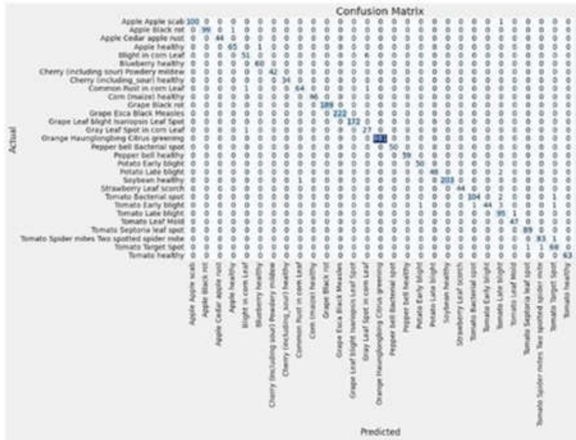


Fig. 15. Confusion matrix EfficientNetB4

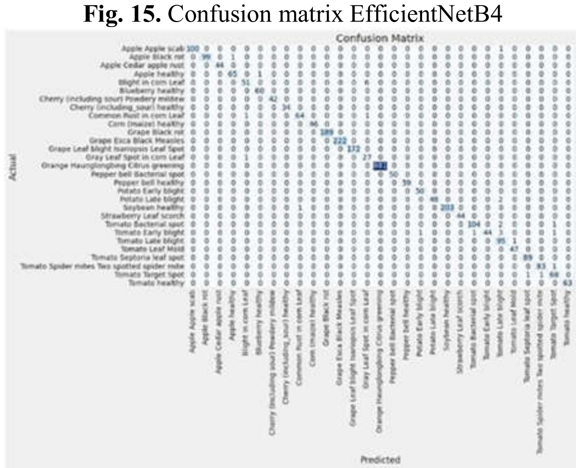
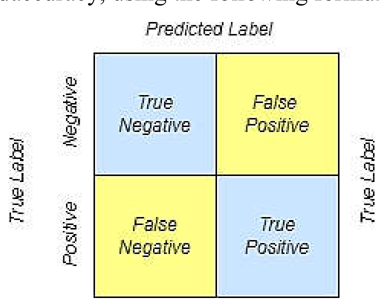


Fig. 16. Confusion matrix EfficientNetB5

The results of the confusion matrix can be computed to determine the value of precision, recall, f1-score andaccuracy, using the following formula (Figure 17).



$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

$$\text{F1 Score} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3}$$

$$\text{Accuracy} = \frac{TN + TP}{TN + FP + TP + FN} \tag{4}$$

Fig. 17. Computation for classification report

The results of the programming that has been done to determine the values of precision, recall, f1-score and accuracy, are summarized in the form of a classification report for 29 plant diseases (Figure 18,19,20).

Classification Report:
----

	precision	recall	f1-score	support
Apple Apple scab	1.0000	0.9901	0.9950	101
Apple Black rot	1.0000	1.0000	1.0000	100
Apple Cedar apple rust	1.0000	1.0000	1.0000	44
Apple healthy	0.9706	1.0000	0.9851	66
Blight in corn Leaf	0.9811	0.9123	0.9455	57
Blueberry healthy	1.0000	0.9500	0.9744	60
Cherry (including sour) Powdery mildew	1.0000	0.9762	0.9880	42
Cherry (including_sour) healthy	1.0000	1.0000	1.0000	34
Common Rust in corn Leaf	0.9701	0.9848	0.9774	66
Corn (maize) healthy	1.0000	1.0000	1.0000	46
Grape Black rot	1.0000	0.9947	0.9973	189
Grape Esca Black Measles	0.9955	1.0000	0.9978	222
Grape Leaf blight Isariopsis Leaf Spot	0.9942	1.0000	0.9971	172
Gray Leaf Spot in corn Leaf	0.8710	0.9643	0.9153	28
Orange Haunglongbing Citrus greening	1.0000	1.0000	1.0000	881
Pepper bell Bacterial spot	1.0000	0.9400	0.9691	50
Pepper bell healthy	0.9516	1.0000	0.9752	59
Potato Early blight	0.9434	1.0000	0.9709	50
Potato Late blight	1.0000	0.9800	0.9899	50
Soybean healthy	1.0000	0.9951	0.9975	204
Strawberry Leaf scorch	1.0000	1.0000	1.0000	44
Tomato Bacterial spot	0.9813	0.9813	0.9813	107
Tomato Early blight	0.9783	0.9000	0.9375	50
Tomato Late blight	0.9794	0.9896	0.9845	96
Tomato Leaf Mold	0.9583	0.9787	0.9684	47
Tomato Septoria leaf spot	0.9778	0.9888	0.9832	89
Tomato Spider mites Two spotted spider mite	1.0000	0.9286	0.9630	84
Tomato Target Spot	0.8961	0.9857	0.9388	70
Tomato healthy	0.9844	1.0000	0.9921	63
accuracy			0.9890	3171
macro avg	0.9805	0.9807	0.9801	3171
weighted avg	0.9894	0.9890	0.9890	3171

Fig. 18. Classification Report EfficientNetB3

Classification Report:
----

	precision	recall	f1-score	support
Apple Apple scab	1.0000	0.9901	0.9950	101
Apple Black rot	1.0000	0.9900	0.9950	100
Apple Cedar apple rust	1.0000	1.0000	1.0000	44
Apple healthy	0.9848	0.9848	0.9848	66
Blight in corn Leaf	0.9623	0.8947	0.9273	57
Blueberry healthy	0.9836	1.0000	0.9917	60
Cherry (including sour) Powdery mildew	1.0000	1.0000	1.0000	42
Cherry (including_sour) healthy	1.0000	1.0000	1.0000	34
Common Rust in corn Leaf	0.9846	0.9697	0.9771	66
Corn (maize) healthy	1.0000	1.0000	1.0000	46
Grape Black rot	1.0000	1.0000	1.0000	189
Grape Esca Black Measles	1.0000	1.0000	1.0000	222
Grape Leaf blight Isariopsis Leaf Spot	1.0000	1.0000	1.0000	172
Gray Leaf Spot in corn Leaf	0.7941	0.9643	0.8710	28
Orange Haunglongbing Citrus greening	1.0000	1.0000	1.0000	881
Pepper bell Bacterial spot	1.0000	1.0000	1.0000	50
Pepper bell healthy	1.0000	1.0000	1.0000	59
Potato Early blight	0.9804	1.0000	0.9901	50
Potato Late blight	1.0000	0.9600	0.9796	50
Soybean healthy	1.0000	0.9951	0.9975	204
Strawberry Leaf scorch	1.0000	1.0000	1.0000	44
Tomato Bacterial spot	0.9905	0.9720	0.9811	107
Tomato Early blight	1.0000	0.8800	0.9362	50
Tomato Late blight	0.9223	0.9896	0.9548	96
Tomato Leaf Mold	0.9792	1.0000	0.9895	47
Tomato Septoria leaf spot	0.9889	1.0000	0.9944	89
Tomato Spider mites Two spotted spider mite	0.9881	0.9881	0.9881	84
Tomato Target Spot	0.9577	0.9714	0.9645	70
Tomato healthy	1.0000	1.0000	1.0000	63
accuracy			0.9912	3171
macro avg	0.9833	0.9845	0.9834	3171
weighted avg	0.9917	0.9912	0.9912	3171

Fig. 19. Classification Report EfficientNetB4

Classification Report:
----

	precision	recall	f1-score	support
Apple Apple scab	1.0000	0.9901	0.9950	101
Apple Black rot	0.9901	1.0000	0.9950	100
Apple Cedar apple rust	1.0000	1.0000	1.0000	44
Apple healthy	1.0000	1.0000	1.0000	66
Blight in corn Leaf	0.9623	0.8947	0.9273	57
Blueberry healthy	1.0000	1.0000	1.0000	60
Cherry (including sour) Powdery mildew	1.0000	0.9762	0.9880	42
Cherry (including_sour) healthy	1.0000	1.0000	1.0000	34
Common Rust in corn Leaf	0.9701	0.9848	0.9774	66
Corn (maize) healthy	1.0000	1.0000	1.0000	46
Grape Black rot	0.9947	1.0000	0.9974	189
Grape Esca Black Measles	1.0000	0.9955	0.9977	222
Grape Leaf blight Isariopsis Leaf Spot	1.0000	1.0000	1.0000	172
Gray Leaf Spot in corn Leaf	0.8065	0.8929	0.8475	28
Orange Haunglongbing Citrus greening	1.0000	1.0000	1.0000	881
Pepper bell Bacterial spot	1.0000	1.0000	1.0000	50
Pepper bell healthy	0.9833	1.0000	0.9916	59
Potato Early blight	1.0000	1.0000	1.0000	50
Potato Late blight	0.9608	0.9800	0.9703	50
Soybean healthy	1.0000	1.0000	1.0000	204
Strawberry Leaf scorch	1.0000	1.0000	1.0000	44
Tomato Bacterial spot	0.9904	0.9626	0.9763	107
Tomato Early blight	1.0000	0.9200	0.9583	50
Tomato Late blight	0.9592	0.9792	0.9691	96
Tomato Leaf Mold	0.9400	1.0000	0.9691	47
Tomato Septoria leaf spot	0.9886	0.9775	0.9831	89
Tomato Spider mites Two spotted spider mite	0.9643	0.9643	0.9643	84
Tomato Target Spot	0.9583	0.9857	0.9718	70
Tomato healthy	1.0000	1.0000	1.0000	63
accuracy			0.9905	3171
macro avg	0.9817	0.9829	0.9820	3171
weighted avg	0.9908	0.9905	0.9906	3171

Fig. 20. Classification Report EfficientNetB5

4. CONCLUSION

Based on the Confusion Matrix and Classification Report, the following information is obtained, EfficientNetB3. The results of testing using test data resulted in the most wrong predictions, namely the type of plant disease "Tomato Spider mites Two spotted spider mites". A total of 84 images "Tomato Spider mites Two spotted spider mites", with the highest correct prediction was 78 and the most incorrect was 6 with the prediction of the type of disease "Tomato Target Spot. EfficientNetB4. The results of testing using test data resulted in the most wrong predictions, namely the type of plant disease "Blight in corn leaf". A total of 57 images "Blight in corn leaf", with the highest correct prediction was 51 and the most wrong was 6 with the prediction of the type of disease "Gray leaf Spot in corn Leaf". EfficientNetB5. The results of testing using test data resulted in the most wrong predictions, namely the type of plant disease "Blight in corn leaf". A total of 57 images "Blight in corn leaf", with the highest correct prediction was 51 and the most wrong was 5 with the prediction of the type of disease "Gray leaf Spot in corn Leaf.

After testing with 3,171 image data tests, using a matrix model resulting from the training process with 57,067 image data trains and 3,170 image data validations, overall results in accuracy and f1-score for EfficientNet versions B3, B4, B5 are very good at 0.9890%, 0.9912%, 0.9905%. Prediction errors that occur because there are 2 images of leaves with certain types of diseases that have a similar shape or color ("Blight in corn leaf" is similar to "Gray leaf Spot in corn Leaf" and "Tomato Spider mites Two spotted spider mite" is similar to “Tomato Target Spot“).

DECLARATION

Author Contribution

Broadly speaking, this research consists of 4 stages, namely: dataset preprocessing, training, testing and results to be evaluated.

Funding

Thanks are addressed to LP3M ITB Ahmad Dahlan Jakarta for their support, so that this research went well.

Acknowledgments

Thanks are addressed to LP3M ITB Ahmad Dahlan Jakarta for their support, so that this research went well.

Conflict of Interest

Declare conflicts of interest or state “The authors declare no conflict of interest.”

REFERENCES

[1] F. Chaddad, “Chapter 2 - Enabling Conditions,” F. B. T.-T. E. and O. of B. A. Chaddad, Ed. San Diego: Academic Press, 2016, pp. 19–44. doi: https://doi.org/10.1016/B978-0-12-801695-4.00002-1.
[2] M. W. Aktar, D. Sengupta, and A. Chowdhury, “Impact of pesticides use in agriculture: their benefits and hazards.,” Interdiscip. Toxicol., vol. 2, no. 1, pp. 1–12, Mar. 2009, doi: 10.2478/v10102-009-0001-7.
[3] F. Caffaro, M. Micheletti Cremasco, M. Roccato, and E. Cavallo, “Drivers of farmers’ intention to adopt technological innovations in Italy: The role of information sources, perceived usefulness, and perceived ease of use,” J. Rural Stud., vol. 76, pp. 264–271, 2020, doi: https://doi.org/10.1016/j.jrurstud.2020.04.028.
[4] R. Akhter and S. A. Sofi, “Precision agriculture using IoT data analytics and machine learning,” J. King Saud Univ. - Comput. Inf. Sci., 2021, doi: https://doi.org/10.1016/j.jksuci.2021.05.013.
[5] S. Wolfert, L. Ge, C. Verdouw, and M.-J. Bogaardt, “Big Data in Smart Farming – A review,” Agric. Syst., vol. 153, pp. 69–80, 2017, doi: https://doi.org/10.1016/j.agsy.2017.01.023.
[6] E. S. Hastomo, W., Karno, A. S. B., Sudjiran, S., Arif, D., & Moreta, “Exloratory Data Analysis Untuk Analisa Data Belanja Pelanggan dan Pendapatan Bisnis Retail Fashion Dengan Bahasa Python,” Infotekmesin, vol. 13, no. 2, 2022, [Online]. Available: https://www.ejournal.pnc.ac.id/index.php/infotekmesin/article/view/1319
[7] W. Hastomo, A. Satyo, B. Karno, and N. Kalbuana, “Characteristic Parameters of Epoch Deep Learning to Predict Covid-19 Data in Indonesia,” 2021, doi: 10.1088/1742-6596/1933/1/012050.
[8] C. LeCun, Y., Kavukcuoglu, K., & Farabet, “Convolutional networks and applications in vision,” Inf. Sci. Eng. 2018 IEEE Int. Symp. circuits Syst., pp. 253–256, 2018.
[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, and A. Anguelov, D., ... & Rabinovich, “Going deeper with convolutions,” Proc. IEEE Conf. Comput. Vis. pattern Recognit., pp. 1–9, 2019.
[10] M. E. H. Chowdhury et al., “Automatic and Reliable Leaf Disease Detection Using Deep Learning Techniques,”
[11] AgriEngineering, vol. 3, no. 2, pp. 294–312, 2021, doi: 10.3390/agriengineering3020020.
[12] K. G. Liakos, P. Busato, D. Moshou, & Pearson, S., and D. Bochtis, “Machine learning in agriculture: A review,”
[13] Sensors, vol. 18, no. 8, p. 2674, 2018.
[14] W. Hastomo, N. Aini, A. Satyo, B. Karno, and L. M. R. Rere, “Metode Pembelajaran Mesin untuk Memprediksi Emisi Manure Management,” vol. 11, no. 2, pp. 131–139, 2022.
[15] J. Chen, D. Zhang, A. Zeb, and Y. A. Nanehkaran, “Identification of rice plant diseases using lightweight attention networks,” Expert Syst. Appl., vol. 169, p. 114514, 2021, doi: https://doi.org/10.1016/j.eswa.2020.114514.
[16] P. Bansal, R. Kumar, and S. Kumar, “Disease Detection in Apple Leaves Using Deep Convolutional Neural Network,” Agriculture, vol. 11, no. 7, 2021, doi: 10.3390/agriculture11070617.
[17] [15] J. Jiang et al., “Evaluation of Diverse Convolutional Neural Networks and Training Strategies for Wheat Leaf Disease Identification with Field-Acquired Photographs,” Remote Sens., vol. 14, no. 14, 2022, doi: 10.3390/rs14143446.
[18] P. Zhang, L. Yang, and D. Li, “EfficientNet-B4-Ranger: A novel method for greenhouse cucumber disease recognition under natural complex environment,” Comput. Electron. Agric., vol. 176, p. 105652, 2020, doi: https://doi.org/10.1016/j.compag.2020.105652.
[19]
[20] M. Moussafir, H. Chaibi, R. Saadane, A. Chehri, A. El Rharras, and G. Jeon, “Design of efficient techniques for tomato leaf disease detection using genetic algorithm-based and deep neural networks,” Plant Soil, 2022, doi: 10.1007/s11104-022-05513-2.
[21] M. A. Khan et al., “Cucumber Leaf Diseases Recognition Using Multi Level Deep Entropy-ELM Feature Selection,” Applied Sciences , vol. 12, no. 2. 2022. doi: 10.3390/app12020593.
[22] X. Sun, G. Li, P. Qu, X. Xie, X. Pan, and W. Zhang, “Research on plant disease identification based on CNN,”
[23] Cogn. Robot., vol. 2, pp. 155–163, 2022, doi: https://doi.org/10.1016/j.cogr.2022.07.001.
[24] A. Pandey and K. Jain, “A robust deep attention dense convolutional neural network for plant leaf disease identification and classification from smart phone captured real world images,” Ecol. Inform., vol. 70, p. 101725, 2022, doi: https://doi.org/10.1016/j.ecoinf.2022.101725.
[25] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, “Plant leaf disease classification using EfficientNet deep learning model,” Ecol. Inform., vol. 61, p. 101182, 2021, doi: https://doi.org/10.1016/j.ecoinf.2020.101182.
[26] W. Hastomo, A. S. Karno, S. Sutarno, D. Arif, & Moreta, E., and S. Sudjiran, “Mengatasi Ketimpangan Data Deep Neural Network dengan Pelipatan Fitur Data Klasifikasi Spektroskopi Darah,” J. Ilm., vol. 8, no. 2, pp. 579–591, 2022, doi: https://doi.org/10.35326/pencerah.v8i2.2251.
[27] S. M. Sakib, “Plant Disease Expert,” kaggle.com, 2022. https://www.kaggle.com/datasets/sadmansakibmahi/plant-disease-expert
[28]
[29]
[30]
[31]
[32]

[33]
[34]