Sainteks
ISSN: 2686-0546
Volume 20 No 2, Oktober 2023
DOI: 10.30595/sainteks.v20i2.18691 (107 – 118)

Comparison of EfficientNet B5-B6 for Detection of 29 Diseases of Fruit Plants

Vany Terisia[1], Widi Hastomo[2*], Adhitio Satyo Bayangkari Karno[3], Ellya Sestri[4], Diana Yusuf[5], Shevty Arbekti Arman[6], Nada Kamilia[7]

[1,2,4,5,6]*Department of Information Technology, Institut Teknologi dan Bisnis Ahmad Dahlan, Jakarta*
[3]*Department of Information System, Universitas Gunadarma*
[7]*Master of Information Technology, STMIK Jakarta STI&K*
[*]corr_author: widie.has@gmail.com

ABSTRACT

In initiatives to meet food needs and enhance the wellbeing of farmers and society at large, crop production performance is essential. For early attempts to be made for quick handling to prevent crop failure, farmers must be able to readily and quickly receive information in order to detect plant illnesses. In this study, two Convolutional Neural Network (CNN) architectures namely, EfficientNet versions B5 and B6 are used to develop a classification model for plant disease using Deep Learning (DL). The 66,556 visuals in the dataset, which is from Kaggle.com, are used. To create a model, the training method uses 57,067 images data and 3,170 image data for validation. The EfficientNet architecture versions B5 and B6 received very good accuracy scores for the total test results, namely 0.9905 and 0.9927. The model testing phase was carried out through testing phases utilising 3.171 images data. Future analysis can compare CNN architectures and try it with different datasets.

Keywords: Convolutional Neural Network, Deep Learning, EfficientNet

INTRODUCTION

Fruit plant diseases can cause significant losses in agricultural production and can threaten food security if not managed properly (Sankaran *et al.*, 2010). The use of Artificial Intelligence (AI) technology, especially deep learning, has shown great potential in supporting plant disease detection with a higher level of accuracy than conventional methods (Ferentinos, 2018; Xenakis *et al.*, 2020). One of the most recent developments in the deep learning field is EfficientNet, a series of Convolutional Neural Network (CNN) architectures proposed by (Tan and Le, 2019).

EfficientNet is designed using an efficient model optimization method (Shah *et al.*, 2022), which combines proportional adjustment of the model size (depth, width, and resolution) in one framework. These models have a high degree of efficiency and accuracy (Huang *et al.*, 2022), allowing them to be implemented in resource-constrained devices, such as mobile devices or embedded systems (Jeddi, Shafieezadeh and Nateghi, 2023). The application of EfficientNet in the detection of fruit plant diseases enables more complex pattern recognition and richer feature representation of infected leaf or fruit images. By using this model, the system can automatically recognize plant disease symptoms and classify disease types with high accuracy (Arun and Viknesh, 2022).

Research that has been carried out by (Atila *et al.*, 2021) uses a plant village dataset of 55,448 images with the EfficientNet B4-B5 architecture. Research by (Hanh, Van Manh and Nguyen, 2022) uses the EfficientNet B5, AlexNet, GoogleNet, VGGNet, ResNets, DenseNets, and MobileNets methods. The results achieved with the EfficientNet B7 method are more optimal compared to DenseNet (Srinidhi, Sahay and Deeba, 2021). The development of the EffiNet-TS model based on EfficientNetV2 is compared to the EffiNet-Teacher classifier, as a decoder, namely the EffiNet-Student classifier using the plantvillage dataset reaching 54,306 images (Gehlot and Gandhi, 2023).

Research with the development of the EfficientNet B4-Ranger model (Zhang, Yang and Li, 2020). Research that has been carried out by (Sun *et al.*, 2022) uses the EfficientNet method compared to ResNet50 and DenseNet169. Research by (Zhang, Yang and Li, 2020) uses the EfficientNet and Residual block methods. Research that has been carried out by (Hridoy and Tuli, 2021) uses a dataset totaling 138,980 images, using the EfficientNet B0-B7 model. Research by (Y *et al.*, 2022) uses the EfficientNet B0 model but has modified parameter sizes and flops.

This study, which intends to compare findings and test current architectural versions, uses the EfficientNet architecture versions (B5 and B6). A deep learning model with high accuracy weights that can be used to predict disease from images of plant leaves is the end result of the training and testing procedure (Atila *et al.*, 2021). One of the main advantages of EfficientNet is its ability to achieve high accuracy with a relatively low number of parameters. This makes it suitable for resource-constrained applications such as mobile devices or constrained computing environments (Tan, M., & Le, 2019).

RESEARCH METHOD

There are 4 stages carried out in this research; (1) dataset preprocessing, (2) training, (3) testing, (4) results and evaluation (Figure 1). This dataset was obtained from kaggle.com, this raw data consists of 66,556 images located in 58 plant disease folders (figure 3). The operating environment used for this experiment is Intel Core i5-10500, NVIDIA GeForce GTX 1060, 16GB DDR4 RAM, NVMe SSD storage, Windows operating system, deep learning framework: with TensorFlow, development environment using Anaconda with Jupyter Notebook editor.

Preprocessing is done to make sure that data can be used correctly and in line with the process that will be carried out. The outputs of the machine learning process will be validated and tested using test data (Straub, 2021). The machine will use data as a learning process (training). If the error rate in the test results remains above the predetermined threshold, the training procedure will be repeated with improved methods (an updated weight matrix).

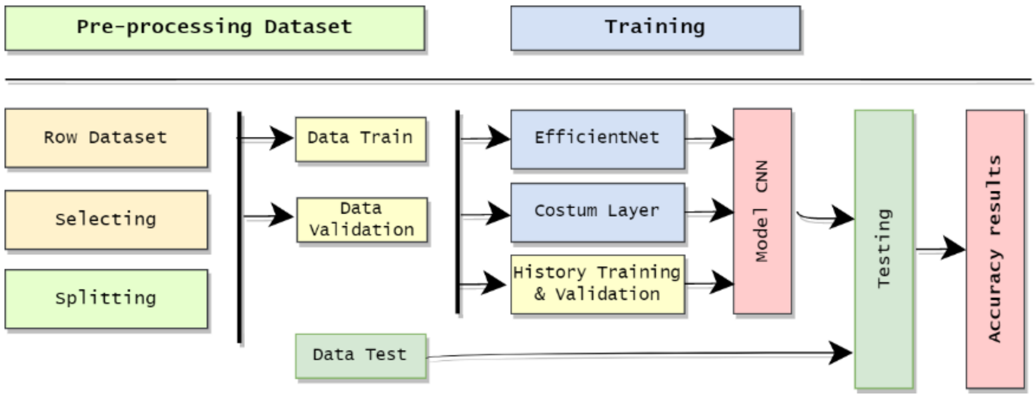

Figure 1. Phases of research

Sainteks
ISSN: 2686-0546
Volume 20 No 2, Oktober 2023
DOI: 10.30595/sainteks.v20i2.18691 (107 – 118)

The subsequent iteration will employ this update weight value until a high percentage accuracy and low loss are attained. The training outcome model is the final weight value with the highest accuracy value and the lowest loss. Additionally, the procedure for testing employing test data will utilise the training outcomes model till the end result is reached in the form of a confusion matrix.

Evaluation of model performance on classification tasks using a confusion matrix (Riehl, Neunteufel and Hemberg, 2023), consisting of four components, namely, True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). From these components, several evaluation metrics such as accuracy, precision, recall, and F1-score can be calculated;

- Accuracy: measures the extent to which the model successfully predicts correctly among all the predictions made. Accuracy is calculated as (1)

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

- Precision: measures the extent to which positive predictions made by the model are correct. Precision is calculated as (2)

$$Prec. = \frac{TP}{TP+FP} \tag{2}$$

- Recall or Sensitivity: measures the extent to which the model can correctly detect positive data. Recall is calculated as (3)

$$Rec. = \frac{TP}{TP+FN} \tag{3}$$

- F1-score: the harmonic average between precision and recall. This is useful when you want to strike a balance between precision and recall. F1-score is calculated as (4)

$$F1 - Score = \frac{2.(Prec.recall)}{Prec+recall} \tag{4}$$

RESULTS AND DISCUSSION

1. Pre-processing Dataset

Figure 2 shows folders with leaf photos of various plants organised by the type of plant disease. A selection method is used with folders containing a minimum of 500 pieces of data since there is an imbalance in the amount of data in each folder. The data is reduced to 63,408 images in 29 files that represent various illness kinds as a consequence of the sorting process, in the initial dataset there were 58 types of leaf diseases. Additionally, the 63,408 images in the data were divided into 57,067 (90%) training images, 3,170 (5%), validation images, and 3,171 (5%), testing images (Figure 3).

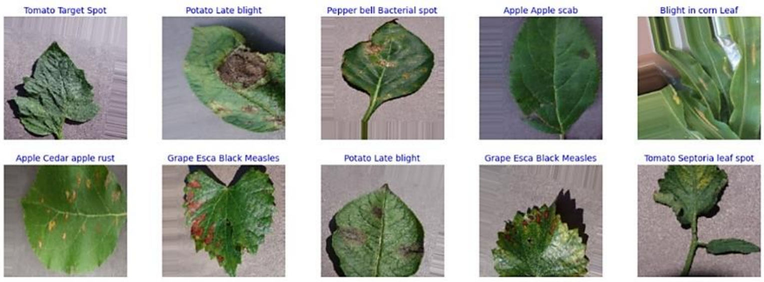

Figure 2. Image dataset

Sainteks
ISSN: 2686-0546
Volume 20 No 2, Oktober 2023
DOI: 10.30595/sainteks.v20i2.18691 (107 – 118)

(a) **(b)**

Figure 3. (a) Row dataset, (b) pre-processing results for training data

2. Training and Testing

The training stage concludes by the validation stage using validation data and the validation stage utilising the training stage using data train. After the training process in each epoch, the validation process is carried out, comparing the result with the goal value to determine the deviation value, which serves as the foundation for updating the weight matrix. The training process for the following epoch will use the most recent weight matrix (after updating). Until the desired epoch value is reached, this process is repeated. The tensorflow-keras library's basic model 2 of the EfficientNet architecture (B5 and B6) as well as the custom layer architecture are used in the training and validation process.

Accuracy and loss numbers will be stored as history during the iteration process. This method achieves great accuracy and little loss while operating in the specified epoch. As a model of training results, the outcomes are expressed as a weight matrix. This model is utilised for the following procedure, which is testing or testing with test data. The EfficientNet architecture will be fed both train and validation data. EfficientNet and ImageNet are utilised with the tensorflow hardware library application to make their use simpler and more effective.

3. The EfficientNet Model

The hundreds of layers that make up the EfficientNet architecture's complexity. Therefore, seven modules will be used to form each of these levels. Data preparation for the following module is done in the stem module and consists of rescaling, normalisation, zero padding, conv2d, batch normalisation, and activation. Block-1's starting point for sub-blocks is module-1. Blocks 2 through 7 employ Module 2 as the starting point for their respective sub-blocks. The Final Module, which serves as a connecting link to all of the sub-blocks, is the last module to emerge from the training process.

Sainteks
ISSN: 2686-0546
Volume 20 No 2, Oktober 2023
DOI: 10.30595/sainteks.v20i2.18691 (107 – 118)

After EfficientNet has been completed, a custom layer should be added to ensure the output is what is desired. Three FC (fully connected) layers and four convolutional layers make up the unique layer design. A custom layer model chart can be presented in png format by using the plot model library. Input, conv2D, max_pooling, flatten, and dense are the custom layers. Each layer in images classification serves a different purpose.

4. Accuracy and Loss

From the model that has been formed, training is carried out with epoch 50, the results are obtained in the form of history for the two architectures (Figure 4 – 7).

```
history=model.fit(x=train_gen,  epochs=epochs, verbose=1, callbacks=callbacks,  validation_data=valid_gen,
               validation_steps=None,  shuffle=False,  initial_epoch=0)

 validation loss of  0.1666 is below lowest loss, saving weights from epoch 48  as best weights
Epoch 49/50
290/290 [==============================] - 144s 498ms/step - loss: 0.1699 - accuracy: 0.9966 - val_loss: 0.1851 - val_accuracy: 0.9921

 validation loss of  0.1851 is above lowest loss of  0.1666 keeping weights from epoch 48 as best weights
Epoch 50/50
290/290 [==============================] - 145s 500ms/step - loss: 0.1700 - accuracy: 0.9967 - val_loss: 0.1679 - val_accuracy: 0.9924

 validation loss of  0.1679 is above lowest loss of  0.1666 keeping weights from epoch 48 as best weights
loading model with weights from epoch  48
training elapsed time was 2.0 hours,  4.0 minutes, 45.29 seconds)
```

Figure 4. History trainning EfficientNet B5

```
history=model.fit(x=train_gen,  epochs=epochs, verbose=1, callbacks=callbacks,  validation_data=valid_gen,
               validation_steps=None,  shuffle=False,  initial_epoch=0)

 validation loss of  0.2060 is above lowest loss of  0.1887 keeping weights from epoch 45 as best weights
Epoch 49/50
290/290 [==============================] - 170s 585ms/step - loss: 0.1750 - accuracy: 0.9972 - val_loss: 0.1823 - val_accurac
y: 0.9921

 validation loss of  0.1823 is below lowest loss, saving weights from epoch 49  as best weights
Epoch 50/50
290/290 [==============================] - 170s 585ms/step - loss: 0.1685 - accuracy: 0.9978 - val_loss: 0.1833 - val_accurac
y: 0.9902

 validation loss of  0.1833 is above lowest loss of  0.1823 keeping weights from epoch 49 as best weights
loading model with weights from epoch  49
training elapsed time was 2.0 hours, 23.0 minutes, 57.95 seconds)
```

Figure 5. History training EfficientNet B6

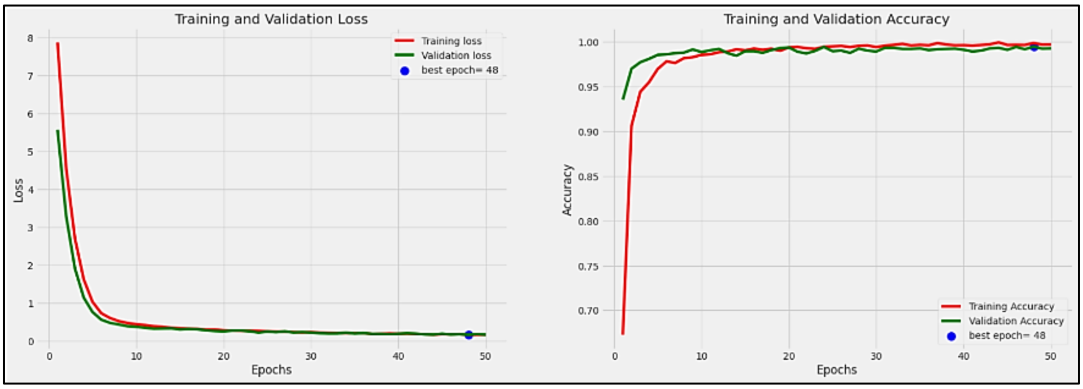

Figure 6. EfficientNetB5 training history graph

Sainteks
ISSN: 2686-0546
Volume 20 No 2, Oktober 2023
DOI: 10.30595/sainteks.v20i2.18691 (107 – 118)

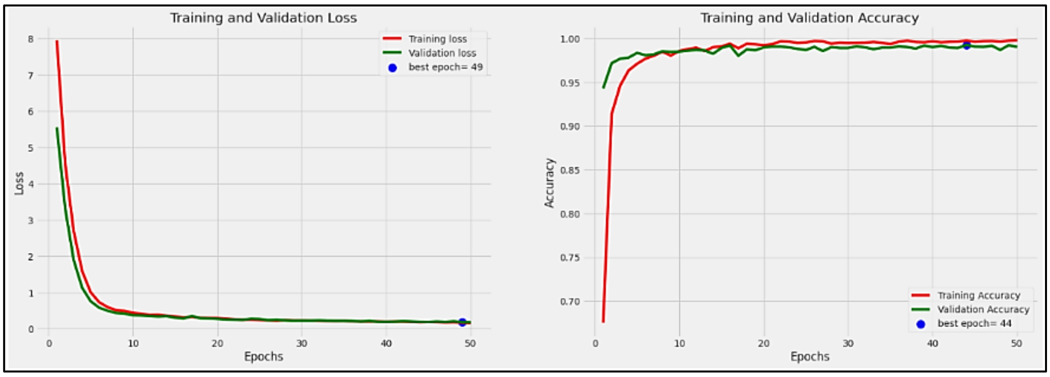

Figure 7. EfficientNetB6 training history graph

Table 1 summarises the history and training graph results. It is clear that larger versions tend to require more training time due to their thicker layers. In a comparison of the two architectures, version B6 obtains the best accuracy with a score of 0.9978, version B6 also achieves the lowest loss with a score of 0.1685, while version B5 has the most validation loss with a score of 0.1679. EfficientNet version B5 has the lowest processing time when using the same data.

Accuracy validation of 0.9902 indicates that the EfficientNet model has been able to recognize and classify leaf images with a very high level of accuracy. This means that most of the leaf disease images have been identified correctly by the model. However, it is always important to understand that no model is perfect, and there are various other aspects that also need to be considered in actual implementations.

The loss value of 0.1685 is a relatively low value, indicating that the model has achieved a good fit with the training data. The lower the loss value, the better the model adapts to the training data. It is important to understand that loss values are only one aspect of model evaluation, and must be analyzed along with other evaluation metrics to get a complete picture of the performance of a foliar disease classification model.

Table 1. Efficientnet Architecture Training Results

EfficientNet	Accuracy	Val_accuracy	Loss	Val_loss	Time
B5	0.9967	0.9924	0.1700	**0.1679**	**2:04:45**
B6	**0.9978**	**0.9902**	**0.1685**	0.1833	2:23:58

5. Confusion Matrix and Classification Report

A programme is then written in Python using the seaborn heatmap module from the results of the predicted value and the actual value, resulting in a confusion matrix (Figure 8–9). The accuracy value of the training process is calculated in order to classify data into the appropriate label as in Table 2.

With an accuracy of 99.91% and 99.97%, research by (Atila *et al.*, 2021) employs a plant village dataset of 55,448 photos and the EfficientNet B4-B5 architecture. The study's findings (Hanh, Van Manh and Nguyen, 2022) demonstrate that the EfficientNet B5 approach (99.997%) is superior to AlexNet, GoogleNet, VGGNet, ResNets, DenseNets, and MobileNets in terms of performance. subject in contrast to DenseNet, the results obtained using the EfficientNet B7 approach are more ideal, coming up at 99.8% and 99.75% respectively (Srinidhi, Sahay and Deeba, 2021). When compared to the EffiNet-Teacher classifier as a decoder, the EffiNet-Student classifier was developed based on EfficientNetV2, and when utilising the plantvillage dataset, which contains 54,306 images, the accuracy reached 99% and the validation loss was 4.5% (Gehlot and Gandhi, 2023).

ISSN: 2686-0546
Volume 20 No 2, Oktober 2023
DOI: 10.30595/sainteks.v20i2.18691 (107 – 118)

Table 2. Performance Study of Present Models

Reference	Methods	Results
(Atila *et al.*, 2021)	EfficientNet B4-B5	accuracy reach 99.91% and 99.97%
(Hanh, Van Manh and Nguyen, 2022)	EfficientNet B5, AlexNet, GoogleNet, VGGNet, ResNets, DenseNets, and MobileNets	accuracy EfficientNet B5 99.997%
(Srinidhi, Sahay and Deeba, 2021)	EfficientNet B7 and DenseNet	accuracy 99.8% and 99.75%
(Gehlot and Gandhi, 2023)	EffiNet-TS based on EfficientNetV2 and EffiNet-Teacher classifier	accuracy EffiNet-Student classifier 99%
(Zhang, Yang and Li, 2020)	EfficientNet B4-Ranger	accuracy 97%
(Sun *et al.*, 2022)	EfficientNet, ResNet50 and DenseNet169	accuracy EfficientNet 99.72%
(Zhang, Yang and Li, 2020)	EfficientNet and Residual block	accuracy 98.87%
(Hridoy and Tuli, 2021)	EfficientNet B0-B7	the accuracy of the B5 version reaches 98.13%
(Y *et al.*, 2022)	EfficientNet B0 with FLOPS	accuracy 92.6%
Proposed models	EfficientNet B5-B6	accuracy 99.05% and 99.27%.

Research that led to the creation of the 97% accurate EfficientNet B4-Ranger model (Zhang, Yang and Li, 2020). Comparing the EfficientNet approach to ResNet50 and DenseNet169, research by (Sun *et al.*, 2022) found that EfficientNet achieved the highest accuracy of 99.72%. Using the EfficientNet and Residual block approaches, research by (Zhang, Yang and Li, 2020) obtains an accuracy of 98.87%. A dataset of 138,980 images has been utilised in research by (Hridoy and Tuli, 2021), and utilising the EfficientNet B0-B7 model, version B5 produced the best performance (98.13%). Using the EfficientNet B0 model with customised parameter sizes and flops, research by (Y *et al.*, 2022) achieves an accuracy of 92.6%.

ISSN: 2686-0546
Volume 20 No 2, Oktober 2023
DOI: 10.30595/sainteks.v20i2.18691 (107 – 118)

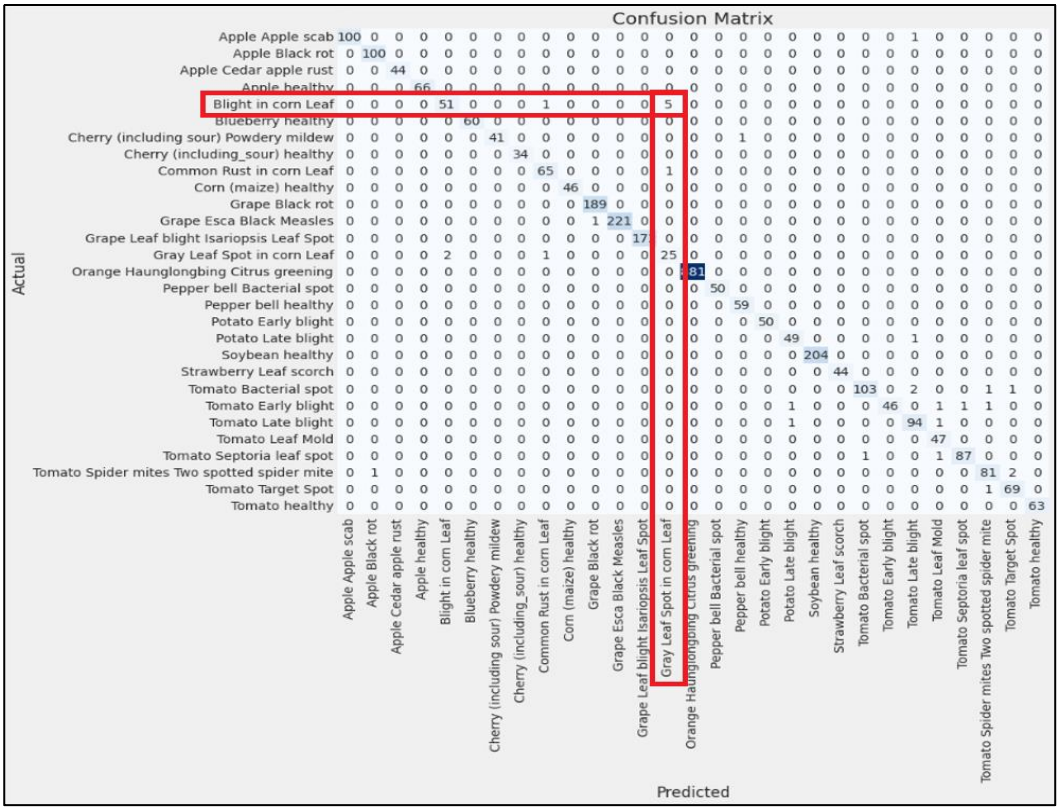

Figure 8. Confusion matrix EfficientNetB5

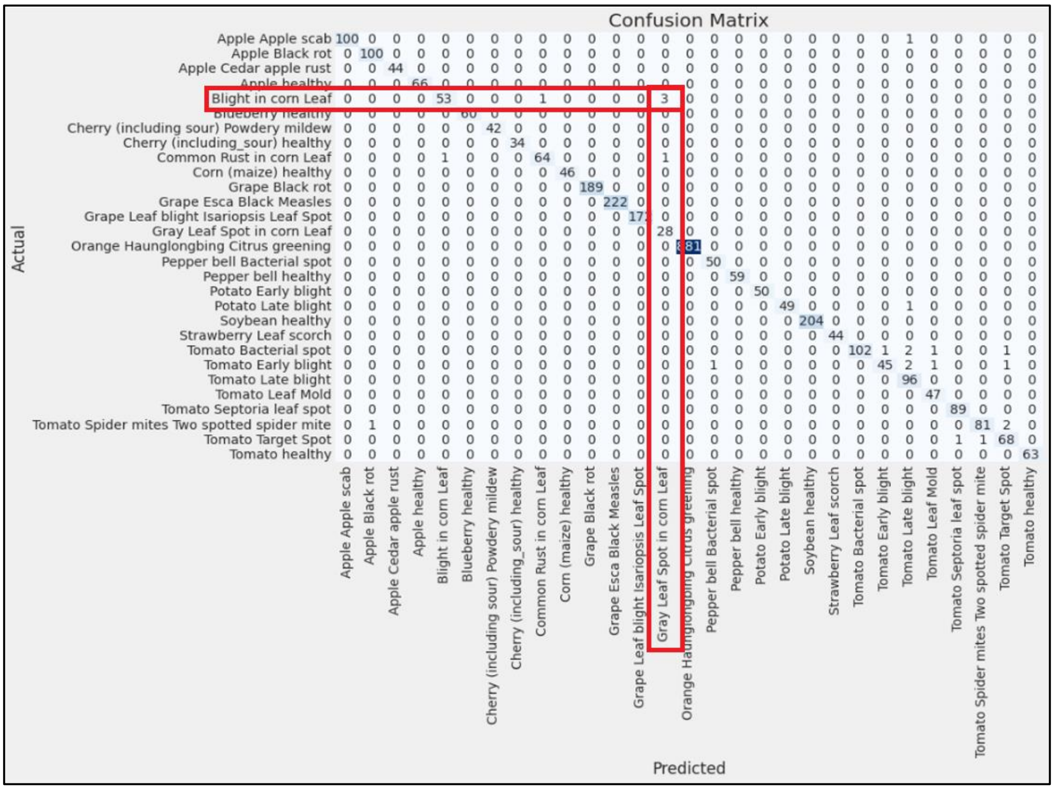

Figure 9. Confusion matrix EfficientNetB6

Computing can be used to calculate the precision, recall, f1-score, and accuracy values for the confusion matrix. Figure 10-11, which is a classification report for 29 plant diseases, presents the results of the programming used to calculate the precision, recall, f1-score, and accuracy values.

EfficientNetB5 The type of plant disease "Blight in maize leaf" was one of the most inaccurate forecasts made by testing utilising test data. A total of 57 photos of "Blight in maize leaf" were used, with the highest right prediction standing at 51 and the lowest at 5, which was the disease type "Grey Leaf Spot in maize Leaf". EfficientNetB6 The type of plant disease "Blight in maize leaf" was one of the most inaccurate predictions made by testing utilising test data. A total of 57 photos of "Blight in maize leaf" were used, with the highest right prediction standing at 53 and the lowest at 3, which was the disease type "Grey Leaf Spot in maize Leaf."

This application has great potential in the world of agriculture. With the help of this technology, farmers can detect diseases in their crops more quickly and accurately. This allows them to take timely preventive or treatment measures, which in turn is expected to increase crop yields and reduce crop losses.

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

Figure 10. Classification report EfficientNetB5

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	0.9815	0.9298	0.9550	57
Blueberry healthy	1.0000	1.0000	1.0000	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	0.8750	1.0000	0.9333	28
Orange Haunglongbing Citrus greening	1.0000	1.0000	1.0000	881
Pepper bell Bacterial spot	0.9804	1.0000	0.9901	50
Pepper bell healthy	1.0000	1.0000	1.0000	59
Potato Early blight	1.0000	1.0000	1.0000	50
Potato Late blight	1.0000	0.9800	0.9899	50
Soybean healthy	1.0000	1.0000	1.0000	204
Strawberry Leaf scorch	1.0000	1.0000	1.0000	44
Tomato Bacterial spot	1.0000	0.9533	0.9761	107
Tomato Early blight	0.9783	0.9000	0.9375	50
Tomato Late blight	0.9412	1.0000	0.9697	96
Tomato Leaf Mold	0.9592	1.0000	0.9792	47
Tomato Septoria leaf spot	0.9889	1.0000	0.9944	89
Tomato Spider mites Two spotted spider mite	0.9878	0.9643	0.9759	84
Tomato Target Spot	0.9444	0.9714	0.9577	70
Tomato healthy	1.0000	1.0000	1.0000	63
accuracy			0.9927	3171
macro avg	0.9866	0.9882	0.9871	3171
weighted avg	0.9930	0.9927	0.9927	3171

Figure 11. Classification Report EfficientNetB6

CONCLUSION

By implementing the EfficientNet algorithm, we succeeded in building a model that is very efficient and accurate in recognizing 29 types of leaf diseases. An accuracy of 0.9927 indicates that the model is very good at classifying leaf images into different disease categories. The main advantage of EfficientNet is its ability to achieve a high level of accuracy with a relatively lower number of parameters compared to traditional CNN architectures. These conclusions demonstrate the great potential of EfficientNet in addressing image classification problems, especially in the context of detection of leaf diseases. This high level of accuracy can be beneficial in practical applications such as crop monitoring and digital farming, where recognizing diseases at an early stage can help increase crop yields and reduce crop losses. It is important to note that success can also depend on the quality and quantity of training data used, as well as a careful model tuning process. In addition, it is necessary to consider the reliability of the model in real-world situations that may involve variations in lighting, image resolution and different conditions.

ACKNOWLEDGEMENT

Thank you to LP3M ITB Ahmad Dahlan Jakarta support for the experiment.

REFERENCES

Arun, Y. and Viknesh, G.S. (2022) 'Leaf Classification for Plant Recognition Using EfficientNet Architecture', in *2022 IEEE Fourth International Conference on Advances in Electronics, Computers and Communications (ICAECC)*, pp. 1–5. Available at: https://doi.org/10.1109/ICAECC54045.2022.9716637.

Atila, Ü. *et al.* (2021) 'Plant leaf disease classification using EfficientNet deep learning model', *Ecological Informatics*, 61, p. 101182. Available at: https://doi.org/https://doi.org/10.1016/j.ecoinf.2020.101182.

Ferentinos, K.P. (2018) 'Deep learning models for plant disease detection and diagnosis', *Computers and Electronics in Agriculture*, 145, pp. 311–318. Available at: https://doi.org/https://doi.org/10.1016/j.compag.2018.01.009.

Gehlot, M. and Gandhi, G.C. (2023) '"EffiNet-TS": A deep interpretable architecture using EfficientNet for plant disease detection and visualization', *Journal of Plant Diseases and Protection*, 130(2), pp. 413–430. Available at: https://doi.org/10.1007/s41348-023-00707-x.

Hanh, B.T., Van Manh, H. and Nguyen, N.-V. (2022) 'Enhancing the performance of transferred efficientnet models in leaf image-based plant disease classification', *Journal of Plant Diseases and Protection*, 129(3), pp. 623–634. Available at: https://doi.org/10.1007/s41348-022-00601-y.

Hridoy, R.H. and Tuli, M.R.A. (2021) 'A Deep Ensemble Approach for Recognition of Papaya Diseases using EfficientNet Models', in *2021 5th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, pp. 1–6. Available at: https://doi.org/10.1109/ICEEICT53905.2021.9667825.

Huang, C. *et al.* (2022) 'Tuberculosis Diagnosis using Deep Transferred EfficientNet', *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pp. 1–9. Available at: https://doi.org/10.1109/TCBB.2022.3199572.

Jeddi, A.B., Shafieezadeh, A. and Nateghi, R. (2023) 'PDP-CNN: A Deep Learning Model for Post-Hurricane Reconnaissance of Electricity Infrastructure on Resource-Constrained Embedded Systems at the Edge', *IEEE Transactions on Instrumentation and Measurement*, 72, pp. 1–9. Available at: https://doi.org/10.1109/TIM.2023.3236321.

Riehl, K., Neunteufel, M. and Hemberg, M. (2023) 'Hierarchical confusion matrix for classification performance evaluation', *Journal of the Royal Statistical Society Series C: Applied Statistics* [Preprint]. Available at: https://doi.org/10.1093/jrsssc/qlad057.

Sankaran, S. *et al.* (2010) 'A review of advanced techniques for detecting plant diseases', *Computers and Electronics in Agriculture*, 72(1), pp. 1–13. Available at: https://doi.org/https://doi.org/10.1016/j.compag.2010.02.007.

Shah, H.A. *et al.* (2022) 'A Robust Approach for Brain Tumor Detection in Magnetic Resonance Images Using Finetuned EfficientNet', *IEEE Access*, 10, pp. 65426–65438. Available at: https://doi.org/10.1109/ACCESS.2022.3184113.

Srinidhi, V. V, Sahay, A. and Deeba, K. (2021) 'Plant Pathology Disease Detection in Apple Leaves Using Deep Convolutional Neural Networks : Apple Leaves Disease Detection using EfficientNet and DenseNet', in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 1119–1127. Available at: https://doi.org/10.1109/ICCMC51019.2021.9418268.

Straub, J. (2021) 'Machine learning performance validation and training using a "perfect" expert system', *MethodsX*, 8, p. 101477. Available at: https://doi.org/https://doi.org/10.1016/j.mex.2021.101477.

Sun, X. *et al.* (2022) 'Research on plant disease identification based on CNN', *Cognitive Robotics*, 2, pp. 155–163. Available at: https://doi.org/https://doi.org/10.1016/j.cogr.2022.07.001.

Tan, M., & Le, Q. (2019) 'Efficientnet: Rethinking model scaling for convolutional neural networks', *International conference on machine learning (PMLR)*, pp. 6105–6114.

Tan, M. and Le, Q. (2019) 'EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks', in K. Chaudhuri and R. Salakhutdinov (eds) *Proceedings of the 36th International Conference on Machine Learning*. PMLR (Proceedings of Machine Learning Research), pp. 6105–6114. Available at: https://proceedings.mlr.press/v97/tan19a.html.

Xenakis, A. *et al.* (2020) 'Applying a Convolutional Neural Network in an IoT Robotic System for Plant Disease Diagnosis', in *2020 11th International Conference on Information, Intelligence, Systems and Applications (IISA*, pp. 1–8. Available at: https://doi.org/10.1109/IISA50023.2020.9284356.

Y, V. *et al.* (2022) 'Early Detection of Casava Plant Leaf Diseases using EfficientNet-B0', in *2022 IEEE Delhi Section Conference (DELCON)*, pp. 1–5. Available at: https://doi.org/10.1109/DELCON54057.2022.9753210.

Zhang, P., Yang, L. and Li, D. (2020) 'EfficientNet-B4-Ranger: A novel method for greenhouse cucumber disease recognition under natural complex environment', *Computers and Electronics in Agriculture*, 176, p. 105652. Available at: https://doi.org/https://doi.org/10.1016/j.compag.2020.105652.