Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 6 No. 5 (2022) 710 - 718 ISSN Media Electronic: 2580-0760

Disease Detection in Banana Leaf Plants using DenseNet and Inception Method

Andreanov Ridhovan[1], Aries Suharso[2], Chaerur Rozikin[3]
[1,2,3]Department of Informatics Engineering, Faculty of Computer Science, University of Singaperbangsa Karawang
[1]andreanov.ridhovan18011@student.unsika.ac.id, [2]aries.suharso@unsika.ac.id, [3]chaerur.rozikin@staff.unsika.ac.id

Abstract

Diseases that attack banana plants can affect the growth and productivity of the fruit produced. The disease can be identified by looking at changes in the pattern and color of the leaves. Infected leaves will experience an increased transpiration process and the photosynthesis process is almost non-existent. Furthermore, disease on banana leaves can cause yield losses of up to 50%. Therefore, early detection is needed so that diseases on banana leaves can be overcome as soon as possible by using deep learning. This study aims to compare the performance of DenseNet and Inception methods in detecting disease on banana leaves. DenseNet is a transfer learning architecture model with fewer parameters and computations to achieve good performance. Inception, on the other hand, is a transfer learning architectural model that applies cross-channel correlation, executes at lower resolution inputs, and avoids spatial dimensions. In conducting the test, this study uses several data handling schemes to test the two methods, namely without data handling, under-sampling, and oversampling. Furthermore, the data is separated into training data and test data with a ratio of 80:20. The result is that the model using the DenseNet method with an oversampling scheme is superior to other models with a percentage value of 84.73% accuracy, 84.80% precision, 84.73% recall, and 84.62% f1 score. In addition, the machine learning model using the DenseNet method in all schemes is also superior to the machine learning model using the Inception method.

Keywords: *Deep learning, Disease Detection, DenseNet, Inception*

1. Introduction

Bananas are a staple food consumed worldwide by more than millions of people. Due to its high demand and consumption throughout the year, bananas have an impact on the country's economy. Healthy banana plants with better resistance to pests and diseases can determine higher banana productivity [1]. Generally, banana plants are very susceptible to disease or pests if not addressed early on so it can affect growth and poor fruit yields. The occurrence of plant diseases hurts agricultural production and if plant diseases are not detected early it can lead to increased food insecurity [2].

Diseases in plants can be identified in various ways, one of which is by looking at changes in the pattern and color of the leaves [3]. The leaf surface plays an important role in assessing the level of disease attack at the leaf level and extrapolation because it affects the photosynthesis process [4]. The physiology and metabolism of disease-infected leaves change. The process of transpiration is increased in infected leaves, whereas photosynthesis is almost non-existent [5]. Recent studies on foliar diseases show how they can harm plant growth. In banana plants, some diseases infect the leaves such as Sigatoka, Cordana, and Pestalotiopsis. When the disease on the leaves gets worse it will drastically reduce the yield of fruit production. It is estimated that plant-infecting pathogens can cause annual yield losses of up to 16% globally [6].

Early warning and forecasting are the basis for effective plant disease prevention and control. It plays an important role in management and decision-making for agricultural production. Until now, visual observation of experienced producers is still the main approach for the detection of plant diseases in rural areas of developing countries. Of course, it requires continuous monitoring of experts and may cost a lot of money to pay for these experts [2].

Based on the description above, the approach that can be used to detect disease in banana leaves is a deep learning approach using transfer learning. Then the

Accepted: 30-06-2022 | Received in revised: 10-09-2022 | Published: 01-10-2022

transfer learning methods that will be used are DenseNet and Inception. At the end of the study, the best method for detecting disease in banana leaves will be known based on the evaluation of the resulting model.

2. Research Methods

This research begins with a literature study related to banana leaf disease and then continues with data collection that will be used to train machine learning models. Then the data will go through a data pre-processing stage which consists of several data handling schemes in testing the two methods and the data will be separated into training data and test data. After that, image augmentation is used to reproduce the image with several techniques. The next stage is model creation and training. In the last stage, an evaluation of the model that has been previously trained using a confusion matrix is carried out. The confusion matrix is a matrix that shows how accurate the model is in making predictions in handling classification cases.

2.1. Data collection

This study uses data in the form of images of banana leaves, namely images of healthy banana leaves, banana leaves infected with Cordana, banana leaves infected with Sigatoka, and banana leaves infected with Pestalotiopsis. The data used is public and obtained from the Kaggle website. The data is 4 gigabytes in size and contains 936 images of healthy banana leaves and disease-infected banana leaves which are divided into several categories [7].

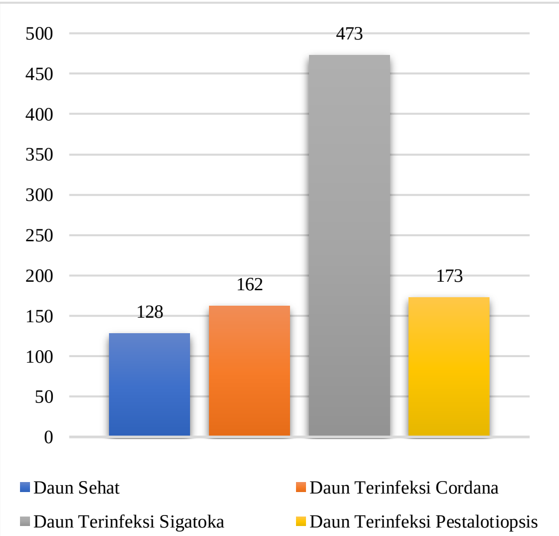


Figure 1. Composition of Initial Data

The data was created by GIS & Remote Sensing Lab, Sheikh Mujibur Rahman Bangabandhu Agricultural University, Bangladesh using a smartphone camera under adequate light conditions by one operator during the first two-week period of March 2020. Then labeled as one of the four classes by one of the plant pathologists [7].

Figure 2. Sample Research Data

2.2. Data Preprocessing

Furthermore, this research will use initial data and two unbalanced data handling schemes, namely oversampling and under-sampling. The unequal distribution of data between categories needs to be overcome by carrying out an under-sampling or oversampling process to equalize the number of images in each category following the category that has the amount of data.

Under-sampling is a technique to balance uneven datasets by storing all data in the minority class and reducing the size of the majority class [8]. While oversampling is a technique to balance uneven datasets by generalizing the class size to the majority class size [9]. The purpose of the process is so that the model does not tend to predict categories that have a large amount of data (the majority) to reduce bias in the model.

Furthermore, this study carried out several schemes for handling image data. The first scheme is the data used purely without the data handling process, the second scheme is under-sampling the data, and the third scheme is over-sampling the data. The results of handling image data in each category can be seen in Table 1.

Table 1. Schema of Image Data Handling Process

Category	Schema 1	Schema 2	Schema 3
Cordana	162 pictures	128 pictures	473 pictures
Sehat	128 pictures	128 pictures	473 pictures
Pestaloti-opsis	173 pictures	128 pictures	473 pictures
Sigatoka	473 pictures	128 pictures	473 pictures

Furthermore, the image data will be separated into training data to train machine learning models and test data to test machine learning models that have been trained previously using certain comparison sizes [10]. The ideal ratio of comparison between training data and test data is 80:20.

Table 2. Table of Data Separation Process

Schema	Total Training Data	Total Testing Data
1	747	189
2	408	104
3	1512	380

DOI: https://doi.org/10.29207/resti.v6i5.4202
Creative Commons Attribution 4.0 International License (CC BY 4.0)

The next process in this research is image augmentation. Image augmentation is used to increase the number of images through a modification process in the image so that it can produce a better model. Image augmentation helps to improve dataset performance [11].

Table 3. Table of Image Augmentation Parameter

Parameter	Value
rescale	1/255
Rotation_range	40
vertical_flip	True
shear_range	0,2
zoom_range	0,2
fill_mode	nearest

2.3. Model Building and Training

This study uses DenseNet and Inception methods which are expected to produce good model accuracy and evaluation. DenseNet introduces a direct connection between two layers with the same feature map size. DenseNet naturally scales to hundreds of layers, exhibiting no optimization difficulties, and tends to produce consistent accuracy improvements as the number of parameters increases, with no signs of performance degradation or overfitting. In addition, DenseNet requires much fewer parameters and less computation to achieve advanced performance [12]. DenseNet also uses reuse and merge features where each layer receives information and input from all previous layers and transfers the combined knowledge via feature maps to all subsequent layers. DenseNet201 architecture has 201 convolutional layers. In Figure 3, DenseNet architecture has several layers in it such as a convolutional layer, pooling layer, transition layer, classification layer, and DenseBlock layer.

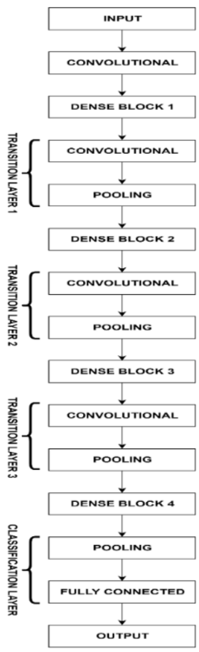


Figure 3. DenseNet Arsitektur Architecture

While Inception is a development model of the Convolutional Neural Network (CNN) which was first introduced by Szegedy, et al., in 2014 in a journal entitled Going Deeper with Convolutions. The Inception architecture can produce very good performance with relatively low computing. The Inception model is very well tested on image classification. The initial structure implements cross-channel correlation, and executes at lower resolution inputs, avoiding spatial dimensions [13]. There are several popular versions of Inception according to the Keras library such as InceptionV1, InceptionV2, and InceptionV3. Each version is the result of improvements from the previous version so that the latest version has advantages over the previous version. However, each version can be used properly according to the data used so a lower version might work better in certain cases. In the latest version of Inception, InceptionV3 has a total of 42 layers.

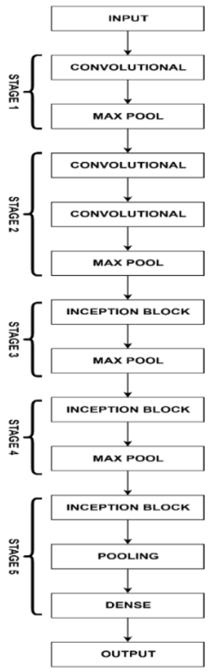


Figure 4. Architecture of Inception

In Figure 4 it can be seen that the Inception network has 5 stages. The network starts with an image size of 224x224x3. It then passes through a 1x1 convolutional layer, a 3x3 MaxPool, a 1x1 convolutional layer, a 3x3 convolutional layer, and a 3x3 MaxPool, and produces a 192x28x28 image. Stage 3 has two Inception blocks and is terminated by the Max Pool layer. Stages 4 and 5 are very similar to stage 3. Stage 4 also has two initial blocks followed by a MaxPool layer, and stage 5 has 2 initial blocks followed by a GlobalAveragePool (Pooling) and a Dense layer.

Therefore, the parameter configuration at the layer that will be used to train the data is then carried out. The first step is to import the method that will be used. Then use

DOI: https://doi.org/10.29207/resti.v6i5.4202
Creative Commons Attribution 4.0 International License (CC BY 4.0)

the DenseNet and Inception methods on the first layer with weights, namely imagenet, include_top is false because it will create a new classification, and the input values are 150, 150, 3. The second layer uses a flattened layer which is used in the transition from the convolution layer to full connected layers. The last layer uses a dense layer with an input value of 4 because the data used consists of 4 categories.

Table 4. Table of DenseNet Model Details

Layer (type)	*Output Shape*	Parameter
Densenet201	None, 4, 4, 1920	18321984
Flatten	None, 30720	0
Dense_2	None, 4	1028

Table 5. Table of Inception Model Details

Layer (type)	*Output Shape*	Parameter
Densenet201	None, 3, 3, 2048	21802784
Flatten	None, 30720	0
Dense_2	None, 4	1028

Furthermore, the model that has been made will be frozen so that the value of the weight does not change and the model will not be retrained [14]. After that, the model will go through a compiling process with the Adam optimizer and the learning rate value is 0.0001. In addition, the metric for measuring machine learning models uses loss which has a categorical_crossentropy value because in this study there are 4 categories or labels. Then the model is trained with epochs or many iterations in training 50 models to produce a good enough model and the verbose used is worth 1 so that the model training process can be seen.

2.4. Model Evaluation

After training the model, the next phase is to evaluate the model to determine if it will do a good job of predicting targets on new and future data. The process of evaluating the model uses a confusion matrix. The confusion matrix is a very popular measure used when solving classification problems. It can be applied to binary classification as well as to multiclass classification problems. The confusion matrix represents the sum of the predicted and actual values [15]. After getting the confusion matrix, then the value of accuracy, precision, recall, and f1 score can be calculated.

The ratio between the number of correctly classified samples and the total number of samples is called accuracy. Precision is used to measure the model's performance in measuring the number of true positives from all positive predictions made. The recall is used to measure the performance of the model in terms of measuring the number of true positives in the right way from all the true positive values. While the f1 score is a machine learning model performance metric that gives equal weight to precision and recall to measure its performance in terms of accuracy and makes it an alternative to accuracy metrics.

$$\text{Accuracy} = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + True\ Positive + True\ Negative} \quad (1)$$

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (2)$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (3)$$

$$\text{F1}\ score = \frac{2\ x\ Recall\ x\ Precision}{Recall + Precision} \quad (4)$$

3. Results and Discussions

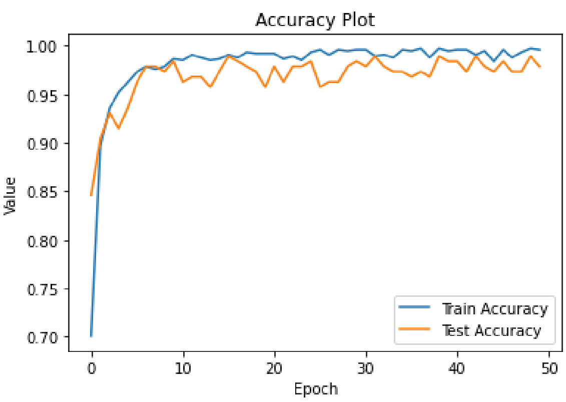
3.1. Model Training

The process of training machine learning models on the entire scheme takes a long time. The time is expressed in seconds. In addition, there are also variables val_loss and val_accuracy which are used to validate the model during training. This variable is also a reference for the model to experience overfitting or underfitting if there is a very large difference in value. The duration of the model training process is 50 epochs or iterations can be seen in Table 6. These results are obtained from the summary in Google Collab when the model training process is complete.

Table 6. Table Details of Model Training Time

Schema	Methods	Total Time (seconds)	Average Time / Epochs (seconds)
1	DenseNet	13800	276
	Inception	11640	233
2	DenseNet	6540	131
	Inception	5280	106
3	DenseNet	21720	435
	Inception	19980	400

After the model training process is complete, the accuracy and loss metrics are visualized in the form of a line chart to show whether the model is good, overfitting, or underfitting. In Figure 5, the machine learning model that has been trained does not experience an underfitting or overfitting process. This is indicated by the movement of the training accuracy and test accuracy parameters which have values that are not much different. In addition, the movement of the training loss and test loss parameters also does not have a very large difference in value.



DOI: https://doi.org/10.29207/resti.v6i5.4202
Creative Commons Attribution 4.0 International License (CC BY 4.0)

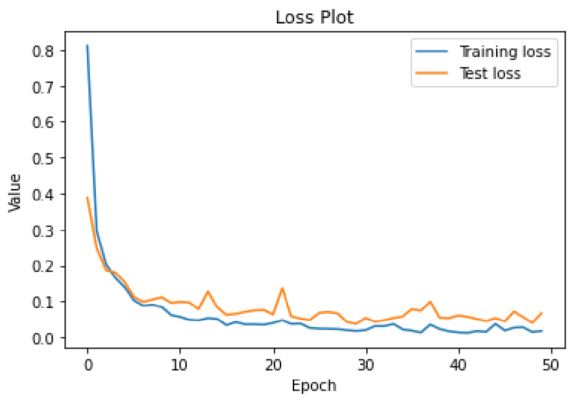


Figure 5. Visualization of Accuracy and Loss Schema 1 DenseNet Method

Based on the loss visualization in Figure 6, the graph shows that the machine learning model is overfitting where the movement of the training loss and train loss shows a large difference in values. Likewise with the movement of the training accuracy and test accuracy parameters where the graph shows a fairly large difference in values.

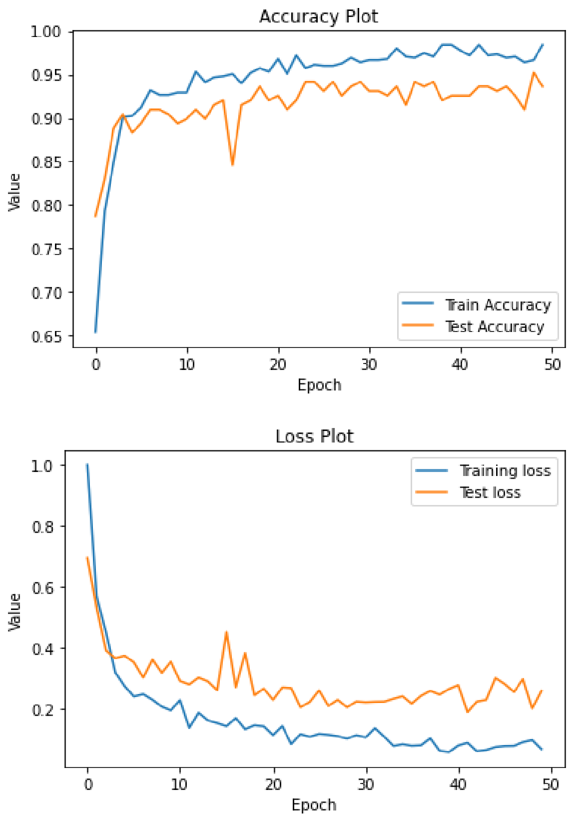


Figure 6. Visualization of Accuracy and Loss Schema 1 Inception Method

In Figure 7, the machine learning model that has been trained does not experience an underfitting or overfitting process. This is indicated by the movement of the training accuracy and test accuracy parameters which have values that are not much different. In addition, the movement of the training loss and test loss parameters also does not have a very large difference in value.

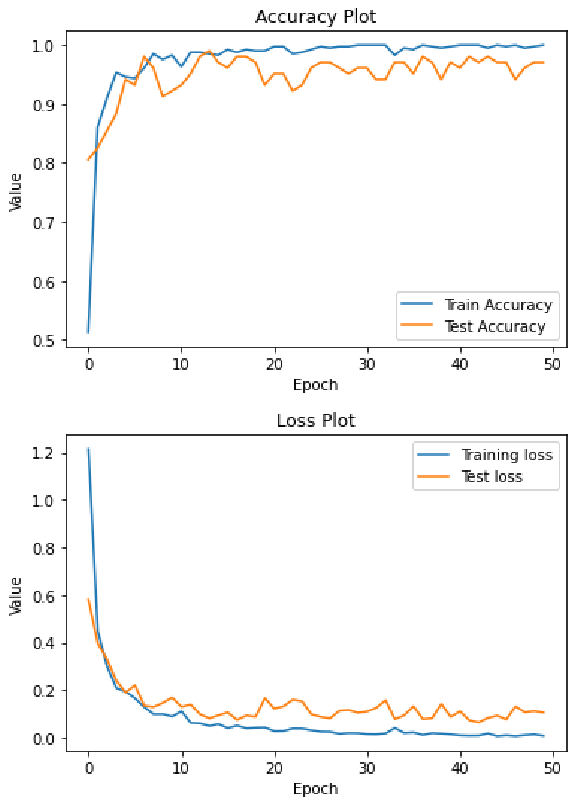
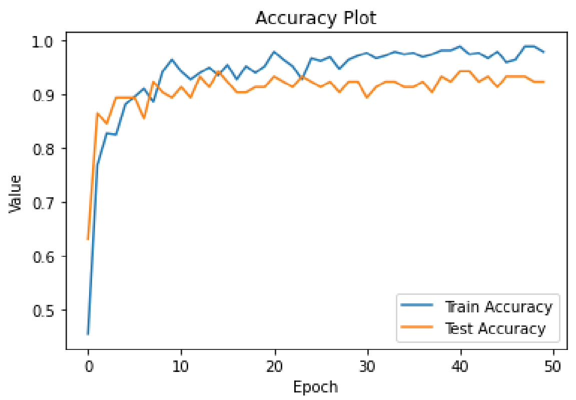


Figure 7. Visualization of Accuracy and Loss Schema 2 DenseNet Method

In Figure 8, the graph shows that the machine learning model is overfitting where the movement of training loss and train loss show a fairly large difference in value. However, based on the movement of the training accuracy and test accuracy parameters, it shows that the machine learning model does not experience overfitting because it does not have a significant difference in value.



Creative Commons Attribution 4.0 International License (CC BY 4.0)

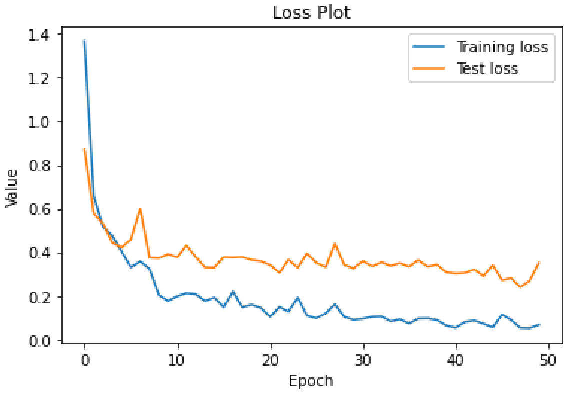


Figure 8. Visualization of Accuracy and Loss Schema 2 Inception Method

In Figure 9 it can be seen that the machine learning model that has been trained does not experience underfitting or overfitting processes. This is indicated by the movement of the training accuracy and test accuracy parameters which have values that are not much different. In addition, the movement of the training loss and test loss parameters also does not have a very large difference in value.

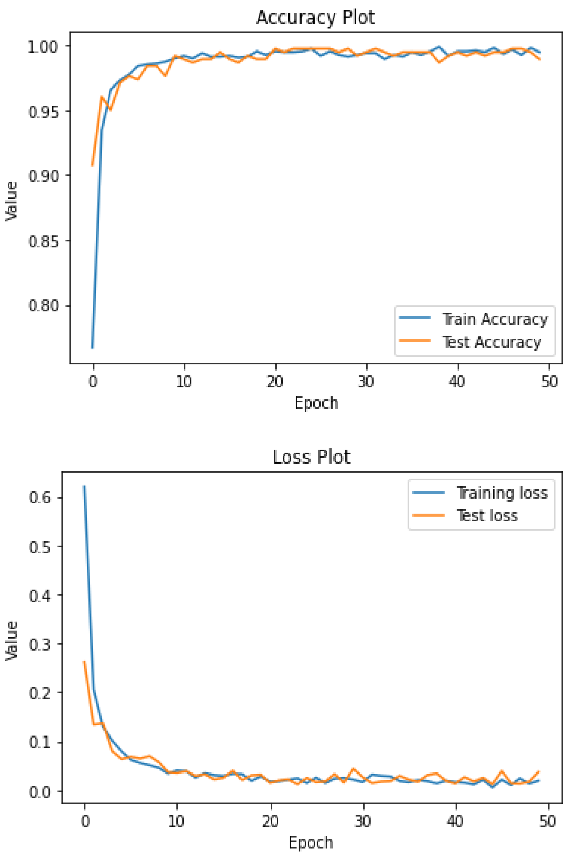


Figure 9. Visualization of Accuracy and Loss Schema 3 DenseNet Method

In Figure 10, the machine learning model that has been trained does not experience an underfitting or overfitting process. This is indicated by the movement of the training accuracy and test accuracy parameters which have values that are not much different. In addition, the movement of the training loss and test loss parameters also does not have a very large difference in value.

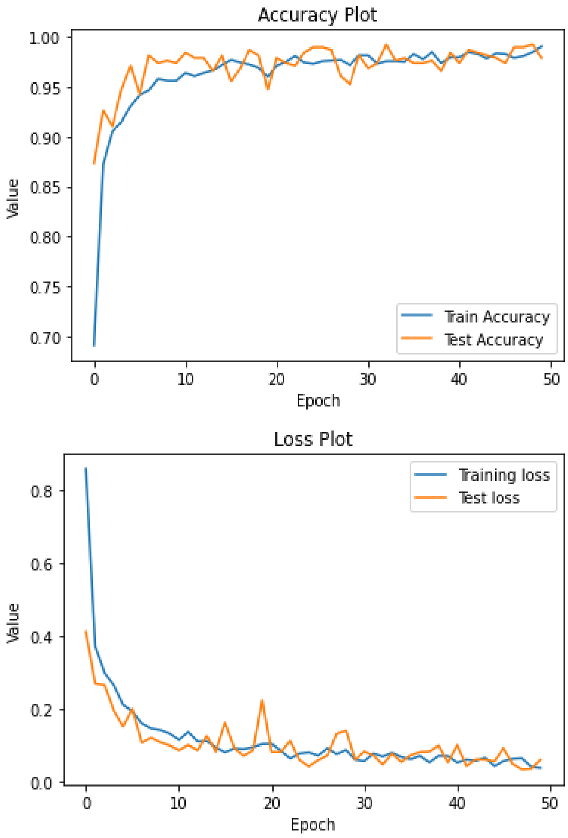


Figure 10. Visualization of Accuracy and Loss Schema 3 Inception Method

3.2. Model Evaluation

After training the model using DenseNet and Inception methods, then the machine learning model will be evaluated using a confusion matrix. This evaluation was conducted to determine how well the quality of the machine learning model produced in detecting diseases in banana leaves. This test uses test data that has been previously separated into the content/dataset/test directory so that the number of values in the confusion matrix matches the number of test data.

In Figure 11 the results of the confusion matrix show that the model tends to predict categories that have a large amount of data. This happens because in Scheme 1 the data does not go through the process of handling unbalanced data (imbalanced data) so the machine learning model that is generated tends to predict categories that have large data (major).

DOI: https://doi.org/10.29207/resti.v6i5.4202
Creative Commons Attribution 4.0 International License (CC BY 4.0)

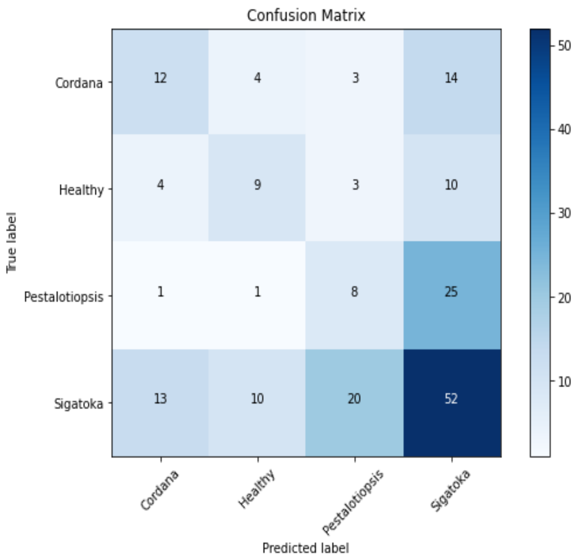


Figure 11. Confusion matrix Schema 1 DenseNet method

In Figure 12 also the results of the confusion matrix show that the model tends to predict categories that have a large amount of data. This happens because in Scheme 1 the data does not go through the process of handling unbalanced data (imbalanced data).

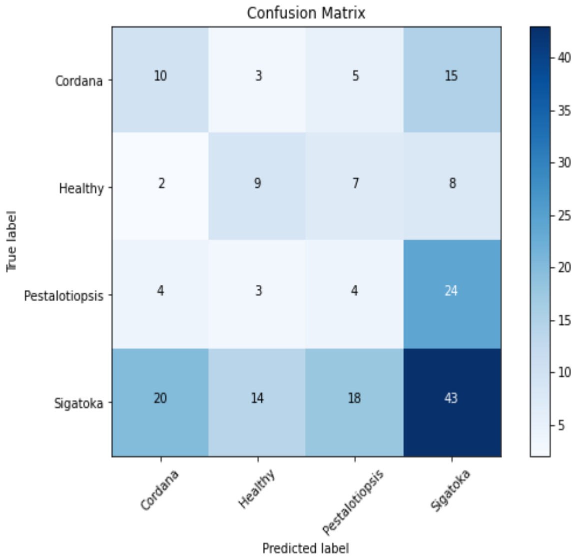


Figure 12. Confusion matrix Schema 1 Inception Method

Furthermore, Figure 13 shows that the resulting machine learning model can predict correctly. This is indicated by the high number of true and predicted labels.

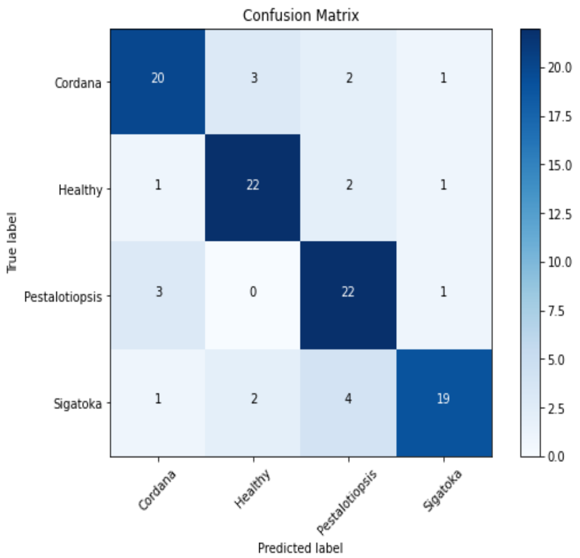


Figure 13. Confusion matrix Schema 2 DenseNet method

Then Figure 14 also shows that the machine learning model produced can predict correctly. This is indicated by the high number of true and predicted labels.

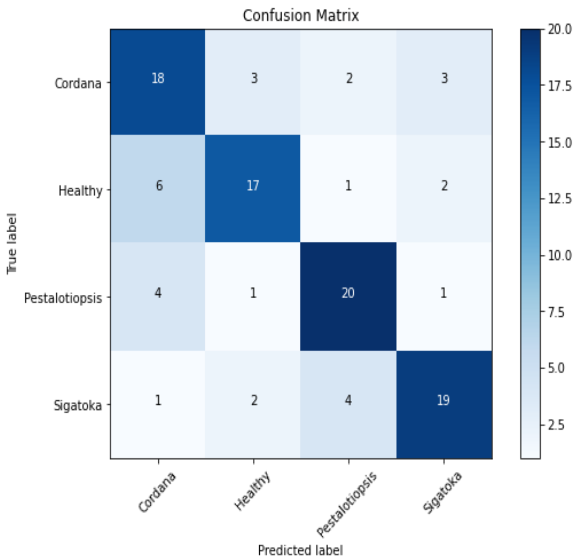


Figure 14. Confusion matrix Schema 2 Inception Method

Figure 15 also shows that machine learning models are getting better at predicting correctly. This is indicated by the high number of true and predicted labels.

DOI: https://doi.org/10.29207/resti.v6i5.4202
Creative Commons Attribution 4.0 International License (CC BY 4.0)

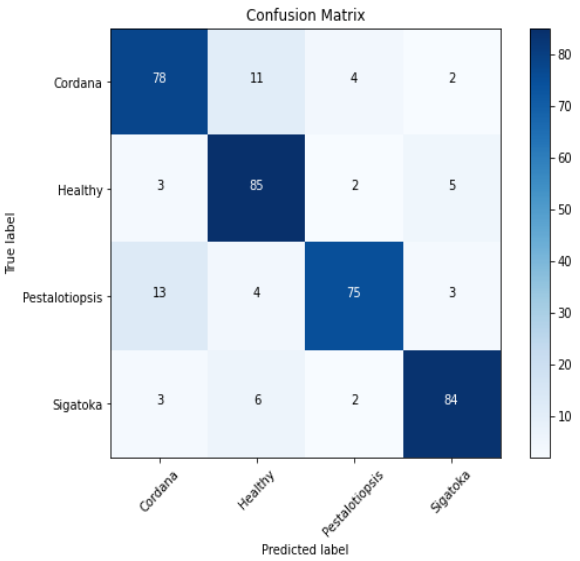


Figure 15. Confusion matrix Schema 3 DenseNet method

Then Figure 16 also shows that the machine learning model produced can predict correctly. This is indicated by the high number of true and predicted labels.

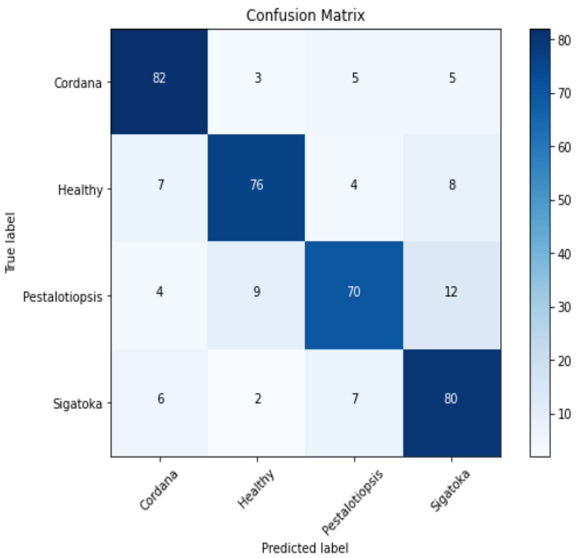


Figure 16. Confusion matrix Schema 3 Inception Method

Based on some of the confusion matrix images above, it can be seen that the confusion matrix has a True label and Predicted label axes. Numbers on the labels that cross and match each other indicate how much the model made a correct prediction. While numbers other than that indicate how much the model produces incorrect predictions.

To better understand the value of the confusion matrix in the previous image, see the table below. In the table, the values are presented in percentage form to make it easier to analyze the performance of the model using the DenseNet method and the model using the Inception method in several schemes.

Table 7. Table of Confusion Matrix Results for All Schemes

Sche-ma	Methods	Accu-racy	Precision	Recall	F1-score
1	DenseNet	42,85%	38,12%	37,13%	37,58%
	Inception	34,92%	29,58%	30,39%	29,94%
2	DenseNet	79,80%	80,29%	79,80%	79,79%
	Inception	71.15%	71,15%	71,15%	71,20%
3	DenseNet	84,73%	84,80%	84,73%	84,62%
	Inception	81,05%	81,08%	82,05%	81,00%

Based on the results of the confusion matrix in Table 7, it can be concluded that the machine learning model generated using the DenseNet method in Scheme 3 is superior to other machine learning models based on the average value of the parameters of accuracy, precision, recall, and f1 score. In addition, the machine learning model using the DenseNet method in all schemes is also superior to the machine learning model using the Inception method.

4. Conclusion

Based on the results of the research that has been done, it can be concluded that the application of deep learning with the DenseNet and Inception methods can detect diseases in banana leaves based on images in bright conditions. In addition, the resulting model can classify banana leaves into the categories of healthy banana leaves, Cordana infected banana leaves, Sigatoka infected banana leaves, and Pestalotiopsis infected banana leaves. Furthermore, based on the results of the performance evaluation of the DenseNet and Inception methods in Table 7, it can be concluded that the machine learning model with the DenseNet method in the oversampling scheme is superior to machine learning models in other schemes with a percentage value of 84.73% accuracy, precision 84.80 %, recall is 84.73%, and f1 score is 84.62%. The comparison results are obtained from the accuracy, recall, precision, and f1 scores of the two methods obtained from the confusion matrix.

Reference

[1] S. Deenan, S. Janakiraman, and S. Nagachandrabose, “Image Segmentation Algorithms for Banana Leaf Disease Diagnosis,” J. Inst. Eng. Ser. C, vol. 101, no. 5, pp. 807–820, 2020, DOI: 10.1007/s40032-020-00592-5.

[2] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, “Using deep transfer learning for image-based plant disease identification,” Comput. Electron. Agric., vol. 173, no. March, p. 105393, 2020, DOI: 10.1016/j.compag.2020.105393.

[3] M. Sardogan, A. Tuncer, and Y. Ozen, “Plant Leaf Disease Detection and Classification Based on CNN with LVQ Algorithm,” UBMK 2018 - 3rd Int. Conf. Comput. Sci. Eng., pp. 382–385, 2018, DOI: 10.1109/UBMK.2018.8566635.

[4] R. Drienovsky, C. Rujescu, F. Sala, and P. Nutrition, “Scan Sick & Healthy Leaf – A SOFTWARE APPLICATION FOR THE DETERMINATION OF THE DEGREE OF THE LEAVES ATTACK,” vol. 49, no. 4, pp. 225–233, 2017.

[5] S. Moscatello et al., “Peach leaf curl disease shifts sugar metabolism in severely infected leaves from source to sink,” Plant Physiol. Biochem., vol. 112, pp. 9–18, 2017, DOI: 10.1016/j.plaphy.2016.12.001.

DOI: https://doi.org/10.29207/resti.v6i5.4202
Creative Commons Attribution 4.0 International License (CC BY 4.0)

[6] J. Amara, B. Bouaziz, and A. Algergawy, “A *deep learning*-based approach for banana leaf diseases classification,” Lect. Notes Informatics (LNI), Proc. - Ser. Gesellschaft fur Inform., vol. 266, pp. 79–88, 2017.

[7] K. Al Mahmud, "Banana Leaf *Dataset*", Kaggle.com, 2022. [Online]. Available: https://www.kaggle.com/kaiesalmahmud/banana-leaf-*dataset*. [Accessed: 20- Feb- 2022].

[8] F. Rayhan, S. Ahmed, A. Mahbub, R. Jani, S. Shatabda, and D. M. Farid, “CUSBoost: Cluster-Based Under-Sampling with Boosting for Imbalanced Classification,” 2nd Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solut. CSITSS 2017, pp. 1–5, 2018, DOI: 10.1109/CSITSS.2017.8447534.

[9] R. Mohammed, J. Rawashdeh and M. Abdullah, "*Machine learning* with Oversampling and Undersampling Techniques: Overview Study and Experimental Results," 2020 11th International Conference on Information and Communication Systems (ICICS), 2020, pp. 243-248, DOI: 10.1109/ICICS49469.2020.239556.

[10] B. Eagan, M. Misfeldt, and A. Siebert-Evenstone, Advances in Quantitative Ethnography, vol. 1. 2019.

[11] H. Salehinejad, S. Valaee, T. Dowdell and J. Barfett, "Image Augmentation Using Radial Transform for *Train*ing Deep Neural Networks," 2018 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2018, pp. 3016-3020, DOI: 10.1109/ICASSP.2018.8462241.

[12] G. Huang and K. Q. Weinberger, “Densely Connected Convolutional Networks.”

[13] M. Hasan, B. Tanawala, and K. J. Patel, “*Deep learning Precision* Farming: Tomato Leaf Disease Detection by Transfer Learning,” SSRN Electron. J., pp. 1–5, 2019, DOI: 10.2139/ssrn.3349597.

[14] "Transfer Learning dan Fine Tuning – 3 – SkillPlus", Skillplus.web.id, 2022. [Online]. Available: https://skillplus.web.id/transfer-learning-dan-fine-tuning-3/. [Accessed: 20- Feb- 2022].

[15] J. Xu, Y. Zhang, and D. Miao, “Three-way *confusion matrix* for classification: A measure driven view,” Inf. Sci. (NY)., vol. 507, pp. 772–794, 2020, DOI: 10.1016/j.ins.2019.06.064

DOI: https://doi.org/10.29207/resti.v6i5.4202
Creative Commons Attribution 4.0 International License (CC BY 4.0)