E-ISSN: 2962-6110
P-ISSN: 2964-7843
JSRET
jsret.knpub.com
Journal of Scientific Research, Education, and Technology

Impact of Data Augmentation Techniques on the Implementation of a Combination Model of Convolutional Neural Network (CNN) and Multilayer Perceptron (MLP) for the Detection of Diseases in Rice Plants.

Mohamad Firdaus, Kusrini, M. Rudyanto Arief
Universitas Amikom Yogyakarta
Email:mohamad.1221@students.amikom.ac.id, kusrini@amikom.ac.id, rudy@amikom.ac.id

ABSTRACT

Detection of disease in rice plants is important to avoid damage and reduce yields. In this study, the influence of data augmentation techniques on the application of a combination model of Convolutional Neural Network (CNN) and Multilayer Perceptron (MLP) was carried out to detect diseases in rice plants. This study uses a dataset of rice images for disease detection which contains 6034 images of rice with five categories, namely Bacterial Leaf Bligh, Blast, Brown Spot, normal, and tungro. This dataset is divided into three parts, namely training data, validation data and test data. The data augmentation techniques used are rotation, brightness and zoom on rice images. The combination model of CNN and MLP is built using the Python programming language and the TensorFlow deep learning framework. In measuring the success rate of the built model, it can be measured using the accuracy, precision and recall values obtained in the model test. Several scenarios were carried out to produce the best model, namely the use of data augmentation techniques, the number of layers and the number of iterations (epochs). From the experiments that have been carried out which have been tested with data as many as 25 digital images, the best model is obtained with an testing accuracy of 92%, 94% precision and 92% recall. This model applies a random zoom augmentation technique with a value of 0.5 – 1.0, CNN+MLP with 3 layers and a dataset ratio of 80:20 and an epoch early stop, This result has increased by more than 10%.

Keywords: Data augmentation, Convolutional Neural Network, Multilayer Perceptron, disease detection.

INTRODUCTION

Indonesia is the world's third-largest rice producer after China and India. This fact is in accordance with the publication of the Food and Agriculture Organization on the Rice Market Monitor Volume XXI Issue No.1 April 2018 (Nations, 2018), which shows that Indonesia ranks third in the world as the largest producer of rice with a production volume of 74.5 million tons per year and has an average rice consumption rate of 38.41 million tons per year with a consumption growth rate of 0.3% per year (Nations, 2018). Based on this data, the production of rice in Indonesia is crucial to maintain high-quality rice with a high level of productivity, ensuring the availability of food commodities domestically. To achieve

this goal, the cultivation of rice crops must be managed carefully by considering various factors such as land processing, seed selection, seeding, maintenance, and pest and disease control. Pest and disease control in rice crops is a major issue due to the potential significant impact they may cause, resulting in significant losses.

The method of detecting rice leaf diseases using digital image input has been widely conducted by researchers, from the use of machine learning to deep learning. One such study conducted by (Islam, et al., 2021) researched the detection of rice leaf diseases using the Deep Convolutional Neural Network (DNN) algorithm, classifying five categories of rice leaf diseases: Brown Spot, Leaf Blast, Leaf Blight, Leaf Smut, and Healthy Leaf, using four models: VGG-19 with an accuracy rate of 0.900, ResNet-101 with an accuracy rate of 0.928, Xception with an accuracy rate of 0.945, and Inception-ResNet-V2 with an accuracy rate of 0.953%.

Another study conducted by (Alidrus, Aziz, & Putra, 2021) developed a model for detecting rice leaf diseases using the Convolutional Neural Network (CNN) algorithm with four convolutional layers and four max pooling layers for feature extraction to generate feature maps. Each convolutional layer used a filter with a dimension of 3 x 3, and max pooling layers with a dimension of 2 x 2. The image then underwent a flattening process, which aims to convert the multidimensional feature map array into a vector that can be used as input for the fully connected layer, which functions to process data for classification. The output of the rice leaf disease category was divided into three categories: Blast, Brown Spot, and Hispa. The literature used for this study was RiceLeafs datasets obtained from Kaggle, consisting of a total of 900 samples of diseased rice plant leaf images with a dimension of 1566 x 1566 pixels. The study found an accuracy rate of 92% for rice leaf disease classification.

The research conducted by Oktaviana, Hendrawan, Annas, & Wicaksono (2021) developed a classification model for Rice Leaf Diseases using a Resnet 101 model with Convolutional Neural Network (CNN) algorithm. The datasets used were obtained from the UCI Machine Learning Repository titled "Rice Leaf Diseases Data Set". The data consisted of three classes, namely Bacterial leaf blight, Brown spot, and Leaf smut, each with 40 data, totaling 120 data for all three classes. The model architecture built consisted of a Dense Layer 512 with 'relu' activation, Batch Normalization layer, Dropout layer of 0.7, Dropout layer of 0.7, Dense Layer 64 with 'relu' activation, Batch Normalization layer, Dropout layer of 0.3, Dropout layer of 0.3, and ending with a Dense layer with output consisting of 3 classes with 'softmax' activation. The model produced an accuracy rate of 100%.

From the description presented, the researchers conducted a study entitled "The Effect of Data Augmentation Techniques on the Application of a Combination Model of Convolutional Neural Network (CNN) and Multilayer Perceptron (MLP) for Plant Leaf Disease Detection". It is hoped that this study will produce a model that can classify rice plant images well through several scenarios about the effects of augmentation techniques and the number of layers in the CNN and MLP models to achieve high test accuracy values.

A. Data Augmentation

Data augmentation is performed in this study with the aim of achieving optimal performance and good accuracy, as well as preventing overfitting (good performance during the training process but poor performance when recognizing new data) in the built deep learning model. Data augmentation involves increasing the variety of data by manipulating the data without losing the essence of the original data.

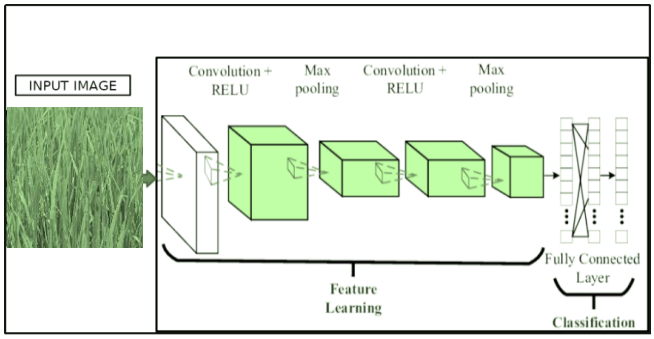
B. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN or ConvNet) is one of the algorithms included in Deep Learning Algorithm that can receive input in the form of images or visuals. In the input image, CNN will map and determine objects or aspects, and then the data is used for the machine to learn to recognize images and distinguish between one image and another (Awangga, Andarsyah, & Putro, 2020). CNN is a development of the Standard Neural Network, and its working principle is inspired by the mammalian visual cortex. CNN has neurons arranged in three dimensions, with length, width, and height corresponding to the image in the input. This is what makes CNN very effective and efficient in analyzing images or visuals.

C. Convolutional Neural Network Architecture

The architecture of Convolutional Neural Network can be seen in Figure 1.

Figure 1. CNN Architecture



1. Feature Learning

In the Feature Learning section, there are layers that function to directly receive input images or visuals in the initial process and process them to produce multidimensional array data as output. Within the Feature Learning section, there are processes that occur, namely the Convolution Layer and Pooling Layer, in which each layer will produce feature maps that are numbers representing the input image and then forwarded to the classification layer.

2. Classification Layer

The Classification Layer consists of layers that have neurons that are all connected to other layers. The Classification Layer's task is to receive input from the output layer of the Feature Learning section, which is then processed in Flatten by adding several hidden layers to the classification layer to generate an output in the form of accuracy values for each class classification.

D. Multilayer Perceptron (MLP)

Feature extraction produces feature maps as output, but they are still in the form of a multidimensional array, so they need to be flattened or reshaped into a vector in order to be used as input for the MLP. If we apply the "Softmax" function to the MLP, we know it as the Fully Connected Layer (FCL).

In the MLP layer, the process of determining which features are related or have correlation with a certain class will take place. The MLP layer also functions to connect all nodes into one dimension. The MLP layer is placed at the end of the architecture process, where it performs a simple matrix multiplication with bias vector addition and applies a non-linear function.

E. Confusion Matrix

Confusion matrix is a technique for measuring the performance of a classification learning model, whether its output is two classes or more. To measure the performance of a classification model, methods for finding precision, recall, and accuracy can be used. In measuring the performance of a classification model, we use the term positive tuple, which is the focus of discussion, while negative tuple is any tuple other than the focus. The confusion matrix is a table with 4 combinations of different predicted and actual values.

Table 1. Confusion Matrix

		Predicted Class	
		Positive	Negative
True Class	Positive	TP	FN
	Negative	FP	FN

From the above Table 1 about the Confusion Matrix, true positive (TP) is the number when the model predicts positive and it is correct, true negative (TN) is the number when the model predicts negative and it is correct, false positive (FP) is the number when the model predicts positive and it is incorrect, while false negative (FN) is the number when the model predicts negative and it is incorrect. P' is the sum of TP and FP, while N' is the sum of FN and TN.

1. Accuracy

Accuracy is the percentage value of the number of test data that are classified into the correct class. Accuracy can be expressed in the following equation (1).

$$Accuracy = \frac{TP}{n} \quad (1)$$

2. Precision

Precision is the accuracy value between the requested information and the answers provided by the model. Precision can be expressed in the following equation (2).

$$Precision = \frac{TP}{(TP+FP)} \quad (2)$$

3. Recall

Recall is a value that indicates how many images are successfully classified by the model. Recall can be expressed in the following equation (3).

$$Recall = \frac{TP}{(TP+FN)} \quad (3)$$

METHOD

This research uses an experimental research method, which involves a series of actions to search for the best data augmentation, features, and architecture to achieve the accuracy level of the CNN + MLP algorithm in the classification model of rice leaf diseases. In this study, a dataset consisting of 5 classes, including Bacterial Leaft Blight, Brown Spot, Blast, Tungro, and Normal, was used. The dataset was obtained from the public data source https://www.kaggle.com/paddy-disease-classification (https://www.kaggle.com/, 2023), which contained 6,034 images divided into three parts: training data, validation data, and testing data. The research flow is shown in Figure 2.

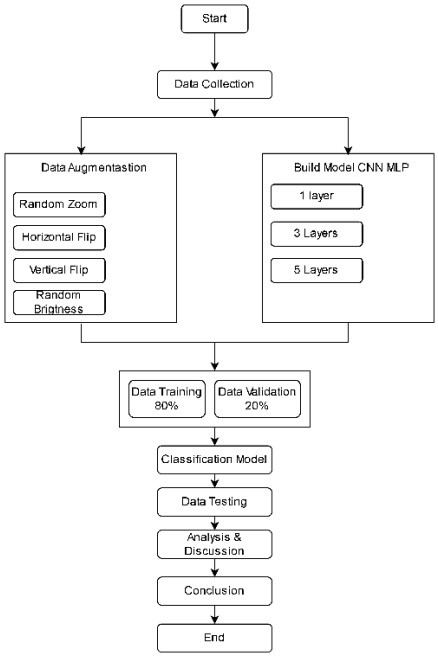


Figure 2. Research flow

The steps carried out in this study are shown in Figure 2.

1. Collection of digital image data of rice plants

The dataset used in this study was obtained from the public data repository kaggle public https://www.kaggle.com/paddy-disease-classification (https://www.kaggle.com/, 2023) totaling 6034, consisting of 5 classes namely Bacterial Leaft Blight, Brown Spot, Blast, Tungro and Normal. The dataset will be used in the training, validation and testing process.

2. Data Augmentation

Experimenting with adding data augmentation techniques, namely Random Zoom, Random brightness, Horizontal flip, and Vertical flip, as well as all combinations.

3. Building the CNN - MLP Model

The stage of building several scenarios with the first using 1 CNN layer, the second using 3 CNN layers, and the third using 5 CNN layers.

4. Training and Validation Data

In this step, each experiment is trained and validated.

5. Classification Model

From the previous step, the training and validation data process will result in a model that will be used to recognize new rice plant image data that has not been introduced to the model before.

6. Data Testing

In this step, the model is tested in recognizing new rice plant image data that has not been introduced before. In this study, there are 25 testing data.

7. Result Evaluation

In the Data testing stage, the model will output the results tested, and in this evaluation process, the confusion matrix method will be used with a total of 25 test data. The evaluation parameters are accuracy, precision, recall, and F1-score.

8. Conclusion

In this stage, all the results in the evaluation stage are summarized in the conclusion based on the values obtained from accuracy, precision, recall, and f1-score to determine the best scenario experiment with the fastest computing time.

FINDING AND DISCUSSION

1. Training and Validation Results

This study used a dataset consisting of 6034 digital images of rice plants with 5 classes: Bacterial Leaf Blight, Brown Spot, Blast, Tungro, and Normal. The dataset was divided into 3 parts: training data, validation data, and testing data, which will be used in the CNN-MLP classification model experiments according to the scenarios to be executed. The purpose of the training and validation data is to make the model recognize rice plant images repeatedly according to the epoch value entered into the command. After obtaining the maximum validation value, the model will be given the test dataset to determine the values of accuracy, precision, recall, and f1-score, and the results of each scenario will be known. Table 2 shows the scenarios carried out in this study.

Table 2. Experiment scenario

Scenario Experiment	Alias
CNN 1 Layer + MLP + RGB + Epoch(Early Stop) + Data 80:20	S1
CNN MLP 1 Layer + Epoch(Early Stop) + Data Augmentation (Random zoom [0.5,1.0])	S2
CNN MLP 1 Layer + Epoch(Early Stop) + Data Augmentation (Random brightness [0.2, 1.0])	S3
CNN MLP 1 Layer + Epoch(Early Stop) + Data Augmentation (vertical flip)	S4
CNN MLP 1 Layer + Epoch(Early Stop) + Data Augmentation (horizontal flip)	S5
CNN MLP 1 Layer + Epoch(Early Stop) + Data Augmentation (1. Random zoom 2. Random brightness 3. Vertical flip Horizontal flip	S6
CNN 3 Layers + MLP + RGB + Epoch(Early Stop) + Data 80:20	S7
CNN 3 Layers + MLP + RGB + Epoch(Early Stop) + Data 80:20 + Data Augmentation (Random zoom [0.5,1.0])	S8
CNN 3 Layers + MLP + RGB + Epoch(Early Stop) + Data 80:20 + Data Augmentation (Random brightness [0.2, 1.0])	S9
CNN-MLP 3 Layers + Epoch(Early Stop) + Data Augmentation (vertical flip)	S10
CNN-MLP 3 Layers + Epoch(Early Stop) + Data Augmentation (Horizontal flip)	S11
CNN MLP 3 Layers + Epoch(Early Stop) + Data Augmentation (1. Random zoom 2. Random brightness 3. Vertical flip 4. Horizontal flip	S12
CNN 5 Layers + MLP + RGB + Epoch(Early Stop) + Data 80:20	S13
CNN 5 Layers + MLP + RGB + Epoch(Early Stop) + Data 80:20 + Data Augmentation (Random zoom [0.5,1.0])	S14
CNN 5 Layers + MLP + RGB + Epoch (Early Stop) + Data 80:20 + Data Augmentation (Random Brightness [0.2, 1.0])	S15
CNN 5 Layers + MLP + RGB + Epoch (Early Stop) + Data 80:20 + Data Augmentation (Vertical Flip)	S16
CNN 5 Layers + MLP + RGB + Epoch (Early Stop) + Data 80:20 + Data Augmentation (Horizontal Flip)	S17
CNN MLP 5 Layers + Epoch (Early Stop) + Data Augmentation (1. Random zoom 2. Random brightness 3. Vertical flip 4. Horizontal flip)	S18

The table 2 above shows the experimental scenarios conducted in this research using the same dataset of digital rice plant images, which were preprocessed by resizing all digital image data to 224x224 size for use in training, validation, and testing processes of the model. In this study, there were 3 CNN-MLP models built, which were 1 layer, 3 layers, and 5 layers.

a. 1 layer Model

- Conv2D layer with 64 filters and kernel size (3, 3), using ReLU activation function, and input_shape (224, 224, 3)

- MaxPooling2D layer with pooling size (2, 2)
- Flatten layer to flatten the output of the previous layer into a 1D vector
- Dropout layer with a probability of 0.5, used as a regularization technique to prevent overfitting
- Dense MLP layer with 512 units and using ReLU activation function
- Dropout layer with a probability of 0.5, as a regularization technique
- Last dense layer with the same number of units as the number of classes in the dataset, using softmax activation function

In this model, the image data will pass through convolution and pooling layers to extract important features from the images, then followed by dense layers to perform classification on the corresponding classes. Dropout layers are used to prevent overfitting in the model.

b. 3 Layers Model

- Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 3)): 2D convolutional layer with 32 filters of size 3x3, ReLU activation function, and input_shape of 224x224 pixels with 3 channels (RGB).
- MaxPooling2D((2, 2)): 2D max pooling layer with filter size of 2x2.
- Conv2D(64, (3, 3), activation='relu'): 2D convolutional layer with 64 filters of size 3x3 and ReLU activation function.
- MaxPooling2D((2, 2)): 2D max pooling layer with filter size of 2x2.
- Conv2D(128, (3, 3), activation='relu'): 2D convolutional layer with 128 filters of size 3x3 and ReLU activation function.
- MaxPooling2D((2, 2)): 2D max pooling layer with filter size of 2x2.
- Flatten(): reshape the output of the previous layer into a 1D vector.
- Dropout(0.5): dropout layer with a rate of 0.5, used to prevent overfitting.
- Dense MLP (512, activation='relu'): fully connected layer with 512 units and ReLU activation function.
- Dropout(0.5): dropout layer with a rate of 0.5.
- Dense(train_generator.num_classes, activation='softmax'): output layer with the number of units equal to the number of classes in the dataset, and softmax activation function to generate output probabilities.

c. 5 Layers Model

- Conv2D layer with 32 filters, 3x3 kernel size, and ReLU activation function as the input layer with input_shape (224, 224, 3)
- MaxPooling2D layer with 2x2 pool size
- Conv2D layer with 64 filters, 3x3 kernel size, and ReLU activation function
- MaxPooling2D layer with 2x2 pool size
- Conv2D layer with 128 filters, 3x3 kernel size, and ReLU activation function

- MaxPooling2D layer with 2x2 pool size
- Conv2D layer with 256 filters, 3x3 kernel size, and ReLU activation function
- MaxPooling2D layer with 2x2 pool size
- Conv2D layer with 512 filters, 3x3 kernel size, and ReLU activation function
- MaxPooling2D layer with 2x2 pool size
- Flatten layer to flatten the output from the previous layer as input for the Dense layer
- Dropout layer to prevent overfitting with a dropout rate of 0.5
- Dense layer with 1024 neurons and ReLU activation function
- Dropout layer to prevent overfitting with a dropout rate of 0.5
- MLP Dense layer with 512 neurons and ReLU activation function
- Dropout layer to prevent overfitting with a dropout rate of 0.5
- Dense layer with output neurons corresponding to the number of classes in the dataset, and using the softmax activation function as the output layer.

In table 3 below are the results of the validation accuracy and computation time for each scenario experiment in the training and validation process.

Table 3. Validation accuracy

Alias	Time / Step	Training
		Val. Accuracy
S1	318ms	0.77
S2	685ms	0.90
S3	359ms	0.75
S4	311ms	0.79
S5	367ms	0.75
S6	765ms	0.67
S7	305ms	0.88
S8	706ms	0.93
S9	356ms	0.89
S10	319ms	0.89
S11	342ms	0.89
S12	884ms	0.87
S13	342ms	0.85
S14	783ms	0.92
S15	368ms	0.92
S16	330ms	0.90
S17	335ms	0.91
S18	847ms	0.74

In table 3, the results of the training and validation process with an 80:20 dataset ratio are presented. The highest validation accuracy value was obtained in scenario 8 (s8) with a validation accuracy of 93%. This scenario consists of a CNN 3-layer - MLP composition with the addition of data augmentation techniques, specifically Random zoom [0.5 - 1.0]. This result shows an improvement compared to scenario 7 (S7) which had the same CNN 3-

layer - MLP model but without data augmentation techniques, with a validation accuracy value of 88%, an increase of 5%. The results also show that not all data augmentation techniques can improve validation accuracy, such as in scenario 1 (S1) which obtained a validation accuracy of 77%, but when combined with a combination of data augmentation techniques in scenario number 6 (S6), the validation accuracy level decreased by 10% to 67%. The following figure shows the comparison of validation accuracy values before and after data preprocessing augmentation for each model.

a. Comparison of 1-layer Model Validation Accuracy

The following figure 3 shows the comparison of validation accuracy values during the training and validation processes in a 1-layer model.

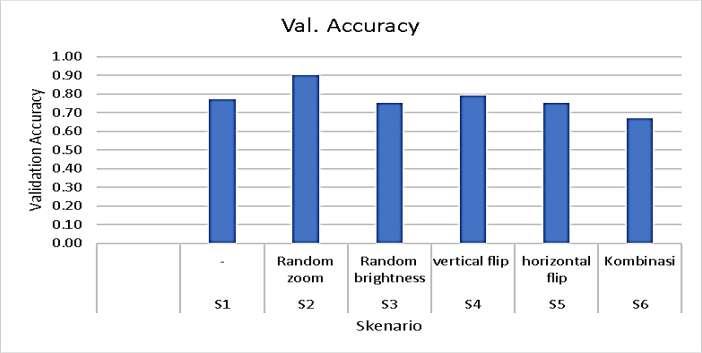


Figure 3. Graphic of 1 layer Model val. accuracy

In Figure 3, the highest validation accuracy is achieved in scenario 2, which utilizes the data augmentation technique of random zoom and obtains a validation accuracy of 90%.

b. Comparison of 3-layer Model Validation Accuracy

The following figure 4 shows the comparison of validation accuracy values during the training and validation processes in a 3-layer model.

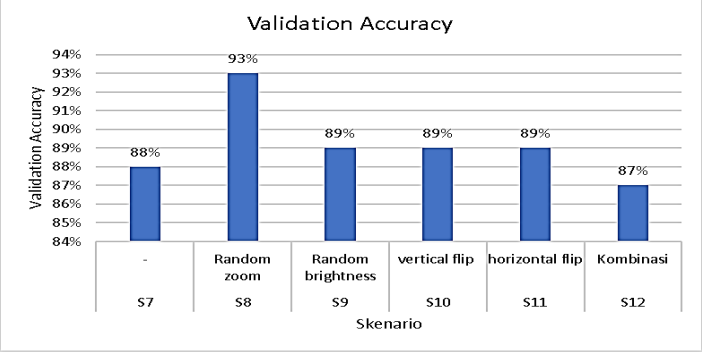


Figure 4. Graphic of 3 layers Model val. accuracy

In figure 4, the highest validation accuracy value is shown in scenario 8, which is achieved by adding data augmentation technique of random zoom with a value of 93%.

c. Comparison of 5-layer Model Validation Accuracy

The following figure 5 shows the comparison of validation accuracy values during the training and validation processes in a 5-layer model.

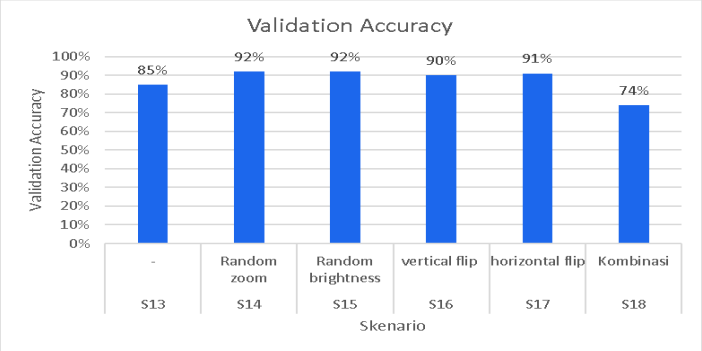


Figure 5. Graphic of 5 layers Model val. accuracy

In figure 4, the highest validation accuracy value is shown in scenario 14, which is achieved by adding data augmentation technique of random zoom with a value of 92%.

2. Testing Results

After the model was trained and validated, the next step was to test the model using 25 digital images of rice plants. Here are the results of the model testing.

Table 4. Model testing results

Alias	Acc	Precision	Recall	F1-Score
S1	0.72	0.79	0.72	0.71
S2	0.88	0.91	0.88	0.88
S3	0.72	0.78	0.72	0.71
S4	0.72	0.84	0.72	0.74
S5	0.68	0.75	0.68	0.68
S6	0.52	0.49	0.52	0.52
S7	0.84	0.88	0.84	0.83
S8	0.92	0.94	0.92	0.92
S9	0.92	0.93	0.92	0.91
S10	0.88	0.89	0.88	0.87
S11	0.80	0.82	0.80	0.79
S12	0.84	0.89	0.84	0.84
S13	0.84	0.89	0.84	0.83
S14	0.92	0.93	0.92	0.92
S15	0.84	0.86	0.84	0.84
S16	0.92	0.93	0.92	0.92
S17	0.88	0.89	0.88	0.88
S18	0.80	0.86	0.80	0.81

From table 4 regarding the results of testing the model with 25 digital images of rice plants, conducted in 18 scenario experiments, the best accuracy, recall, precision, and f1-score values were obtained in scenario number 8, with a testing accuracy of 93%, precision value of 92%, recall value of 94%, and f1-score value of 92%. Figure 6 shows the confusion matrix of the testing results in scenario number 8.

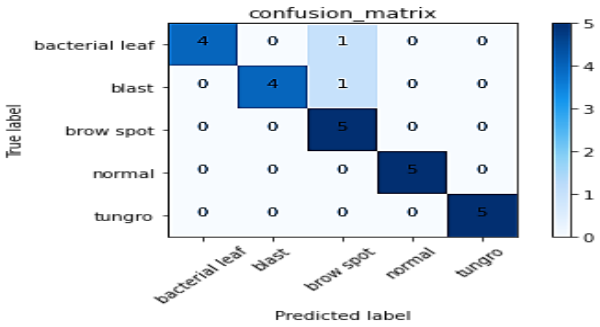


Figure 6. Scenario 8's confusion matrix

Based on the results of the confusion matrix shown in figure 6, it can be concluded that the correct prediction rate for tungro class is 100%, for normal class is 100%, for brown spot class is 100%, for blast class is 80%, and for bacterial leaf blight class is 80%. From the 25 test data, the model was able to correctly predict 23 of them, while 2 data were incorrectly predicted by the model.

CONCLUSION

Based on the analysis of the experimental scenarios, which involved a total of 18 experiments, it can be concluded that the best scenario is scenario 8 with a validation accuracy of 93%, testing accuracy of 92%, precision value of 94%, recall value of 92%, and f1-score of 92%. Out of the 25 test data, this model successfully predicted 23 of them correctly, while the remaining 2 were predicted incorrectly. Scenario 8 uses the same model as scenario 7, but without data augmentation, which had a testing accuracy of 84%. Therefore, the use of data augmentation, in this case, the random zoom technique with a parameter value of 0.5-1.0, increases the testing accuracy by more than 10%.

REFERENCES

Awangga, R. M., Andarsyah, R., & Putro, E. C. (2020). *Object Detection With Faster Region-Based Convolutional Neural Network (Faster R-CNN).* Bandung: Kreatif Industri Nusantara.

Alidrus, S. A., Aziz, M., & Putra, O. V. (2021). Deteksi Penyakit Pada Daun Tanaman Padi Menggunakan Metode Convolutional Neural Network. *SENAMIKA*, 103-109.

Food and Agriculture Organization of the United Nations. (2018). *Rice Market Monitor.* London: fao.

Gazali, W., Soparno, H., & Ohliati, J. (212). Penerapan Metode Konvolusi Dalam Pengolahan Citra Digital. *Matstat*, 103 - 113.

Islam, M. A., Shuvo, M. N., Shamsojjaman, M., Hasan, S., Hossain, M. S., & Khatun, T. (2021). An Automated Convolutional Neural Network Based Approach for Paddy Leaf Disease Detection. *(IJACSA) International Journal of Advanced Computer Science and Applications*, 280-288.

Oktaviana, U. N., Hendrawan, R., Annas, A. D., & Wicaksono, G. W. (2021). Klasifikasi Penyakit Padi berdasarkan Citra Daun Menggunakan Model . *JURNAL RESTI*, 1216-1222.