Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th February 2025. Vol. 4. No. 2; e-ISSN: 2808-4519

Performance Analysis of CNN Architecture in Rice Leaf Disease Image Classification

Hilmi Atha Dzahkwan[1], Robi Aziz Zuama[2]*, Arief Rama Syarif[3]

[1,2]Universitas Bina Sarana Informatika
[3]Universitas Nusa Mandiri
robi.rbz@bsi.ac.id [1]*

Abstract

The decline in rice yields in Indonesia is caused by the attacks of plant pests (OPT), including leaf diseases such as leaf blast, bacterial blight, and dwarfing. Early identification of rice diseases is a crucial step to improving productivity and food security. This study utilizes *Deep Learning* with *Convolutional Neural Network* (CNN) to detect rice leaf diseases based on digital images. The dataset consists of four classes of rice leaf diseases, totaling 5,932 images. The research process involves data collection, preprocessing, data augmentation, model training, and evaluation using accuracy, precision, recall, and F1-score metrics.
The results indicate that the CNN model achieved the highest accuracy of 99% on validation data with a training-to-validation data ratio of 80:20. The evaluation using a confusion matrix demonstrates excellent performance in distinguishing disease types. With high accuracy levels, this model has the potential to become an effective tool for early detection of rice leaf diseases, providing practical solutions for farmers in managing plant diseases.

Keywords: *Deep Learning, Convolutional Neural Network (CNN), Rice leaf diseases, Early detection; Food security;*

1. Introduction

Plant diseases are a factor that reduces the quality of crops and the quantity of agricultural production [1]. Diseases that attack rice plant leaves can decrease rice production, thereby affecting harvest yields. Statistical data on rice production in Indonesia highlights the importance of addressing this issue. Therefore, it is crucial for rice farmers to adopt appropriate cultivation methods to reduce the risk of diseases and pests, such as leaf blight, barnyard grass, rice blast, dwarf disease, and weeds. Typically, when rice plants are affected by diseases and pests, farmers directly apply pesticides or use other control methods that may not always match the specific disease or pest attacking the plants. Given the importance of rice production, the people of Indonesia are encouraged to innovate to ensure rice production increases or at least remains stable to maintain national food security [2]. Rice is a type of grain that ranks third as a staple food after wheat and corn, playing a crucial role in meeting daily needs. According to the Food and Agriculture Organization (FAO), Indonesia is the world's largest rice producer, with production reaching 54.65 million tons in 2020 [3]. However, reports from the National Food Agency (2023) indicate that, based on data from the Central Bureau of Statistics, the harvested area for rice decreased by 0.19% compared to 2019. One of the factors influencing rice production quality is the quality of fertilizers containing Phosphorus (P), Potassium (K), and Nitrogen (N). FAO also states that the primary cause of rice production decline is pest and disease attacks, which account for 20-40%. Rice crops are frequently attacked by plant pests (OPT), leading to potential crop failures. Limited knowledge and delays in diagnosing the types of diseases affecting rice plants worsen symptoms and result in crop failure [3]. According to the FAO, Indonesia, as the world's largest rice producer, achieved a total production of 54.65 million tons in 2020. However, based on data from the Central Bureau of Statistics, the harvested area for rice decreased by 0.19% in 2020 compared to 2019. FAO emphasizes that pest and disease attacks are responsible for 20-40% of the decrease in rice yields.

Studies have shown that advanced technology, such as machine learning and deep learning, can assist farmers in detecting and addressing pests and diseases in rice crops more effectively. For example: Research [4] titled "Implementation of Deep Learning in Rice Pest Classification Systems Using Convolutional Neural Network (CNN)" demonstrated the ability to classify and handle pests efficiently. The training accuracy reached 83.02%, 78.30%, and 81.13%, while the testing accuracy achieved 69.33%, 77.33%, and 76%. [5] conducted a study titled "Detection of Diseases on Rice Leaves Using Image Processing Based on Convolutional Neural Network (CNN)" to identify rice leaf diseases. The testing accuracy reached 93.75%, with a loss value of 0.3076. In their study [6] "Pattern Recognition of Rice and Weed Leaves Using Principal Components Analysis (PCA) and Extreme Learning Machine (ELM)," achieved a highest accuracy of 91.67% with 10, 15, 30, 35, and 40 hidden neurons, while the lowest accuracy was 58% with 5 hidden neurons. [7] in their research "Classification of Rice Leaf Diseases Using Random Forest and Color Histogram," achieved an accuracy of 99.65%.

Studies like [8] emphasize the potential of artificial intelligence, specifically deep learning, in identifying rice diseases. The CNN algorithm has proven highly effective, with experimental results showing 93.3% accuracy in classifying rice diseases and 92.68% validation accuracy.

Building on this foundation, this research will apply deep learning techniques using the CNN method to predict diseases in rice leaves, as explored in the study by [9] It is expected to contribute to early detection of diseases in rice crops and aid in improving agricultural outcomes [10].

2. Research Method

This section explains the research methods applied in this study. The dataset used consists of images of rice leaf diseases. The dataset is then classified using the CNN method to determine its accuracy. The following is the flow of the research methodology, which can be seen in the Fig. 1 below.

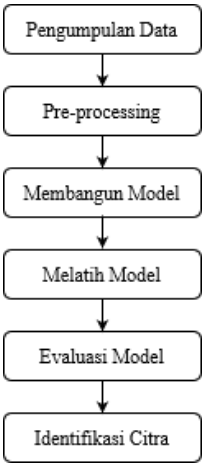

Fig. 1: Research Method

1. Data Collection
   The dataset used contains a collection of rice disease images based on leaf images. The dataset was obtained from the Kaggle website through the Rice Leafs Disease Dataset (https://www.kaggle.com/datasets/maimunulkjisan/rice-leaf-dataset-from-mendeley-data).
2. Pre-Processing
   Pre-processing is the stage of preparing data before it is used in the training of a CNN model. The purpose of pre-processing is to ensure the quality and consistency of the rice leaf disease dataset to be used in the model.
3. Model Development
   Building the model involves designing the architecture by determining convolutional layers, pooling layers, and fully connected layers as needed. Factors such as model depth, filter size, and activation functions. Afterward, the model is trained using the pre-processed dataset.
4. Model Training
   Training the Convolutional Neural Network (CNN) model involves teaching the network to recognize visual patterns in image data. During this phase, the CNN model is provided with a large set of labeled image data. The model processes these images iteratively, performing convolution, pooling, activation, and weight updates to optimize its performance. This process includes gradient calculations that adjust network weights based on the difference between the model's predictions and the correct labels. The model learns essential patterns and features in the images to perform the desired task.
5. Model Evaluation
   During the model evaluation phase, several steps are performed:
   a) Validation subset separation: A small portion of the dataset is set aside as a validation subset that is not used during training to objectively assess model performance.
   b) Performance measurement: Performance metrics such as accuracy, precision, recall, and F1-score are used to evaluate the model on the validation subset.
   c) Parameter tuning: Combinations of hyperparameters are adjusted to improve model performance. These hyperparameters include learning rate, the number of layers, the number of filters, and activation functions.
6. Image Identification
   Image identification is the process of recognizing or classifying objects, patterns, or features in image data. The goal is to identify or classify the content of images based on certain characteristics or attributes. In this research, the object to be identified is rice leaves. The ultimate goal of this study is to correctly identify diseases in rice leaves.

3. Result and Discussion

3.1. Data Collection

The data used consists of image data obtained from Kaggle, with a total of 5.932 images divided into 4 classes. The following Table 1 is the data collected from the mentioned sources.

Table 1: Data Collection

Disease Name	Number of Images	Image Size
Bacterial Blight	1.584	256 x 256
Blast	1.440	256 x 256

Brown Spot	1.600	256 x 256
Tungro	1.308	256 x 256

3.2. Pre-processing

1. The augmentation stage is used in machine learning to improve model performance and prevent overfitting. During this process, each image is modified so that the original image is transformed in shape and position. The following Table 2 is the augmentation that will be applied in this research process.

Table 2: Augmentation Type

Augmentation Type	Value
Rotation range	40
Width shift range	0.2
Height shift range	0.2
Shear range	0.2
Zoom range	0.2
Horizontal flip	True
Fill mode	Nearest

Table Explanation: the parameters used are Rotation Range = 40 The image can be rotated between -40 to +40 degrees. Width Shift Range = 0.2 The image can be shifted horizontally up to 20% of its width. Height Shift Range = 0.2 The image can be shifted vertically up to 20% of its height. Shear Range = 0.2 The shear angle can vary up to 20%. Zoom Range = 0.2 The image can be enlarged or reduced up to 20%. Flip Horizontal = True The image can be flipped horizontally. Fill Mode = 'nearest' Empty pixels are filled with the nearest pixels from the image.

2. **The distribution of training and validation data** is an important step in training a model for image classification. The data will be split into two groups: training data for the model training process, and validation data for testing the trained model. The data will be divided three times with three different scenarios. In Scenario 1, the data is split 70% for training data and 30% for validation data. In Scenario 2, the data is split 80% for training data and 20% for validation data. In Scenario 3, the training data uses 60% of the total data, and the validation data uses 40% of the total data. You can see the data breakdown and the number of images for each scenario in the Table 3 below.

Table 3: Split Data

Scenario	Data Split	Training Data	Validation Data
1	70:30	4.153	1.779
2	80:20	4.747	1.185
3	60:40	3.560	2.372

3. **Building the CNN Model**
Building a CNN (Convolutional Neural Network) model is the process of developing an algorithm or system capable of processing and analyzing visual data, particularly image data. The purpose of creating the CNN model is to produce a model that is effective and accurate in performing tasks.
The architecture used is CNN. The CNN architecture is a neural network designed specifically for image processing and other spatial data processing tasks. This architecture consists of several layers, each with the specific function of extracting important features from the image step by step. The following Table 4 is a general description of the CNN architecture in this study.

Table 4: CNN Architecture

Layer	Pixel Size
Input Shape	100*100*3
Convolution 1	3*3*32
maxpooling	2*2
Convolution 2	3*3*64
Maxpooling	2*2
Dense	128
Activation	ReLU
Dense	4
Activation	Softmax

The input shape of the image is 100x100 pixels with 3 color channels (RGB). The first Convolution layer has 32 filters with a size of 3x3, which helps in extracting features from the image. The Maxpooling layer has a pooling size of 2x2, which selects the maximum value from each 2x2 window to reduce the spatial dimensions of the feature map. The second Convolution layer follows with 64 filters of size 3x3, further refining the extracted features. After that, a Dense layer with 128 units is applied, which connects all neurons and processes the features. The activation function used in this layer is ReLU (Rectified Linear Unit), which helps in introducing non-linearity to the model. The final Dense layer has 4 units, representing the classes that the model will predict. The Softmax activation function is used in this final layer to produce prediction probabilities for each class, enabling the model to make a decision on which class the image belongs to.

3.3. Augmentation Results

The following Fig. 2 is the result of the augmentation that was done previously.

Fig. 2: Augmentation Results

3.4. Architectur Model Result

The process of creating the model has been discussed in the previous section. The following Table 5 is the result of the model that has been created.

Table 5: Architecture CNN Model Result

Layer (Type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 98, 98, 32)	896
max_pooling2d_1 (MaxPooling2D)	(None, 49, 49, 32)	0
conv2d_2 (Conv2D)	(None, 47, 47, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 23, 23, 64)	0
flatten_1 (Flatten)	(None, 33856)	0
Dense_1 (Dense)	(None, 128)	4333696
dense_2 (Dense)	(None, 4)	516
Total params: 4353604 (16.61 MB)		
Trainable params: 4353604 (16.61 MB)		
Non-trainable params: 0 (0,00 Byte)		

The model consists of multiple layers, starting with a Conv2D (Convolutional Layer) that uses 32 filters and produces an output of size (None, 98, 98, 32) with 896 trainable parameters. This is followed by a MaxPooling2D (Max Pooling Layer) that reduces the spatial dimensions, resulting in an output size of (None, 49, 49, 32). The next layer is another Conv2D (Convolutional Layer) with 64 filters, yielding an output of size (None, 47, 47, 64) and containing 18,496 trainable parameters. Another MaxPooling2D (Max Pooling Layer) is applied, reducing the output size to (None, 23, 23, 64). A Flatten Layer then reshapes the data from (None, 23, 23, 64) to a 1D vector of size (None, 33,856). This is followed by a Dense Layer with 128 neurons, producing an output size of (None, 128) and 4,333,696 trainable parameters. Finally, the model includes another Dense Layer with 4 neurons for the output, resulting in an output size of (None, 4) and 516 parameters. In total, the model contains 4,353,604 trainable parameters, which are optimized during the training process.

3.5. Training the Model

Table 6: Experiment Results

Experiment	Dataset Ratio	Epoch	Train Accuracy	Validation Accuracy	Training Loss	Validation Loss
1	**70:30**	**10**	**0.99**	**0.99**	**0.08**	**0.06**
1	80:20	10	0.99	0.99	0.06	0.07
1	60:40	10	0.93	0.93	0.69	0.81
2	70:30	15	0.98	0.98	0.12	0.27
2	80:20	15	0.98	0.99	0.11	0.20
2	60:40	15	0.97	0.96	0.23	0.20
3	70:30	20	0.96	0.97	0.98	0.87
3	80:20	20	0.97	0.97	0.94	0.69
3	60:40	20	0.98	0.99	0.10	0.16

In the process above Table 6, the model is trained in three different scenarios. Here is the explanation of the results:

1. **Experiment 1** with a dataset ratio of 70:30 using 10 epochs achieved a train accuracy of **0.99** and a validation accuracy of **0.99**, with a training loss of **0.08** and a validation loss of **0.06**.
2. **Experiment 1** with a dataset ratio of 80:20 using 10 epochs achieved a train accuracy of **0.99** and a validation accuracy of **0.99**, with a training loss of **0.06** and a validation loss of **0.07**.
3. **Experiment 1** with a dataset ratio of 60:40 using 10 epochs achieved a train accuracy of **0.93** and a validation accuracy of **0.93**, with a training loss of **0.69** and a validation loss of **0.81**.
4. **Experiment 2** with a dataset ratio of 70:30 using 15 epochs achieved a train accuracy of **0.98** and a validation accuracy of **0.98**, with a training loss of **0.12** and a validation loss of **0.27**.
5. **Experiment 2** with a dataset ratio of 80:20 using 15 epochs achieved a train accuracy of **0.98** and a validation accuracy of **0.99**, with a training loss of **0.11** and a validation loss of **0.20**.
6. **Experiment 2** with a dataset ratio of 60:40 using 15 epochs achieved a train accuracy of **0.97** and a validation accuracy of **0.96**, with a training loss of **0.23** and a validation loss of **0.20**.
7. **Experiment 3** with a dataset ratio of 70:30 using 20 epochs achieved a train accuracy of **0.96** and a validation accuracy of **0.97**, with a training loss of **0.98** and a validation loss of **0.87**.
8. **Experiment 3** with a dataset ratio of 80:20 using 20 epochs achieved a train accuracy of **0.97** and a validation accuracy of **0.97**, with a training loss of **0.94** and a validation loss of **0.69**.
9. **Experiment 3** with a dataset ratio of 60:40 using 20 epochs achieved a train accuracy of **0.98** and a validation accuracy of **0.99**, with a training loss of **0.10** and a validation loss of **0.16**.

The best training model results are in Scenario 1, the amount of data used consists of 4,153 samples for training data and 1,779 samples for data validation, with 4 classes in each subset. In this study, the training process was carried out for 10 epochs. The results of Scenario 1 show the highest performance of **99% for training data** and **99% for validation data**, with a **validation loss value of 0,06**. This indicates that the model performs well in recognizing patterns in the data used for training and is also capable of generalizing its results to unseen data. Below is a graphical representation of the training process for Scenario 1.

1. Accuracy Graph
   The graph below Fig. 3 shows that the accuracy of the training and validation data has increased significantly at each iteration..

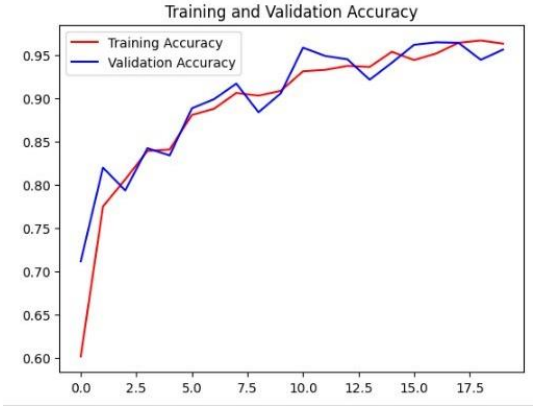

Fig. 3: Accuracy Graph

2. Loss Graph
   The graph below Fig. 4 shows that the loss on the training and validation data is relatively low, indicating that the model can easily learn patterns in the training and validation data.

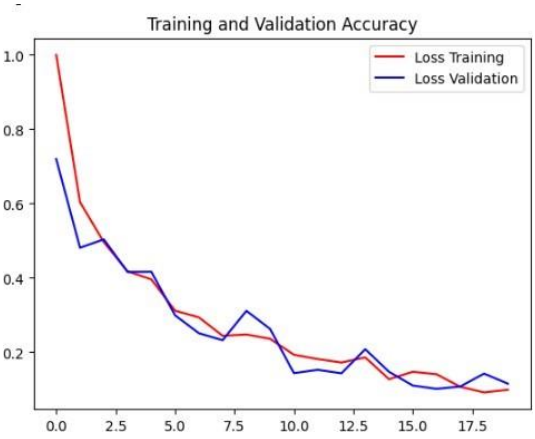

Fig. 4: Loss Graph

3.6. Model Evaluation

This process Table 7 presents the evaluation results of each scenario described above. The model evaluation results are shown in the table below:

Table 7: Evaluation Results

Scenario	Precision	Recall	F1-score	Classification Result
1	95%	94%	94%	94%
2	97%	97%	97%	97%
3	94%	94%	94%	94%

1. **Result Of Scenario 1**
   The evaluation process in Scenario 1 was carried out using a validation dataset comprising 30% of the total dataset. Based on the model training results above, the evaluation results are as follows.

```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.86      0.92        14
           1       0.86      1.00      0.92         6
           2       0.83      1.00      0.91         5
           3       1.00      1.00      1.00         7

    accuracy                           0.94        32
   macro avg       0.92      0.96      0.94        32
weighted avg       0.95      0.94      0.94        32
```

2. **Result Of Scenario 2**
   The evaluation process in Scenario 2 was conducted using a validation dataset comprising 20% of the total dataset. Based on the model training results above, the evaluation results are as follows.

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4
1	0.92	1.00	0.96	12
2	1.00	1.00	1.00	8
3	1.00	0.88	0.93	8
accuracy			0.97	32
macro avg	0.98	0.97	0.97	32
weighted avg	0.97	0.97	0.97	32

3. **Result Of Scenario 3**
   The evaluation process in Scenario 3 was conducted using a validation dataset comprising 40% of the total dataset. Based on the model training results above, the evaluation results are as follows.

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.83	0.91	6
1	0.88	0.88	0.88	8
2	0.90	1.00	0.95	9
3	1.00	1.00	1.00	9
accuracy			0.94	32
macro avg	0.94	0.93	0.93	32
weighted avg	0.94	0.94	0.94	32

4. Connclusion

Based on the results of the research that has been conducted, the following conclusions can be drawn:

1. Based on the experiments conducted, we have developed a CNN model to detect diseases in rice leaves. In the first experiment, the accuracy reached 99% for both training data and validation data, which was found in scenario 3 with an 80:20 split. In the second experiment, the accuracy also reached 99% for both training data and validation data, which occurred in scenario 3 with an 80:20 split. However, the best result was obtained in the third experiment, where the accuracy reached 99% for both training data and validation data, which occurred in scenario 3 with an 80:20 split. The best result in the third experiment shows that the CNN model successfully identified patterns and characteristics of rice leaf diseases very well.
2. Based on the classification results using the CNN model, it was found that the model was able to classify rice leaf diseases with high accuracy, reaching 99%. The evaluation results using the confusion matrix showed excellent performance in classifying various disease classes in rice leaves. The high precision, recall, and F1-score for each class also indicate the model's ability to recognize and distinguish diseases well. Thus, this CNN model has the potential to be an effective tool in detecting diseases in rice plants, which can provide significant benefits in supporting agriculture and the welfare of farmers.
3. Through this research, a Convolutional Neural Network (CNN) model was successfully developed to detect diseases in rice leaves. The developed CNN model has proven its potential as an effective tool in recognizing and detecting diseases that affect rice plants.

References

[1] A. Jinan, B. H. Hayadi, and U. P. Utama, "Klasifikasi Penyakit Tanaman Padi Mengunakan Metode Convolutional Neural Network Melalui Citra Daun (Multilayer Perceptron)," *J. Comput. Eng. Sci.*, vol. 1, no. 2, pp. 37–44, 2022.

[2] M. D. Pratama, R. Gustriansyah, and E. Purnamasari, "Klasifikasi Penyakit Daun Pisang menggunakan Convolutional Neural Network (CNN)," *J. Teknol. Terpadu*, vol. 10, no. 1, pp. 1–6, 2024, doi: 10.54914/jtt.v10i1.1167.

[3] D. Putri Ayuni, Jasril, M. Irsyad, F. Yanto, and S. Sanjaya, "Augmentasi Data Pada Implementasi Convolutional Neural Network Arsitektur Efficientnet-B3 Untuk Klasifikasi Penyakit Daun Padi," *Zo. J. Sist. Inf.*, vol. 5, no. 2, pp. 239–249, 2023, doi: 10.31849/zn.v5i2.13874.

[4] S. Yuliany, Aradea, and Andi Nur Rachman, "Implementasi Deep Learning pada Sistem Klasifikasi Hama Tanaman Padi Menggunakan Metode Convolutional Neural Network (CNN)," *J. Buana Inform.*, vol. 13, no. 1, pp. 54–65, 2022, doi: 10.24002/jbi.v13i1.5022.

[5] S. Sheila, I. Permata Sari, A. Bagas Saputra, M. Kharil Anwar, and F. Restu Pujianto, "Deteksi Penyakit Pada Daun Padi Berbasis Pengolahan Citra Menggunakan Metode Convolutional Neural Network (CNN)," *Multinetics*, vol. 9, no. 1, pp. 27–34, 2023, doi: 10.32722/multinetics.v9i1.5255.

[6] Ahmad Izzuddin and M. Rizal Wahyudi, "Pengenalan Pola Daun untuk Membedakan Tanaman Padi dan Gulma Menggunakan Metode Principal Components Analysis (PCA) dan Extreme Learning Machine (ELM)," *ALINIER J. Artif. Intell. Appl.*, vol. 1, no. 1, pp. 44–51, 2020, doi: 10.36040/alinier.v1i1.2521.

[7] S. K. Wildah, A. Latif, A. Mustopa, S. Suharyanto, M. S. Maulana, and A. Sasongko, "Klasifikasi Penyakit Daun Kopi Menggunakan Kombinasi Haralick, Color Histogram dan Random Forest," *J. Sist. dan Teknol. Inf.*, vol. 11, no. 1, p. 35, 2023, doi: 10.26418/justin.v11i1.60985.

[8] A. Faizin, A. Tri Arsanto, Moch. Lutfi, and A. Rochim Musa, "Deep Pre-Trained Model Menggunakan Arsitektur Densenet Untuk Identifikasi Penyakit Daun Padi," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 6, no. 2, pp. 615–621, 2022, doi: 10.36040/jati.v6i2.5475.

[9] B. Julianto, I. N. Farida, and M. A. D. W. Dara, "Implementasi Metode CNN Pada Aplikasi Android Untuk Deteksi Penyakit Pada Daun Padi Penulis Korespondensi," *Inotek*, vol. 7, pp. 2549–7952, 2023, [Online]. Available: https://proceeding.unpkediri.ac.id/index.php/inotek/

[10] I. Irmawati, H. Hermanto, E. H. Juningsih, S. Rahmatullah, and F. Aziz, "Prediksi Lama Tinggal Pasien Rawat Inap Di Rumah Sakit Pada Masa Pandemi Covid-19 Menggunakan Metode Ensemble Learning Dan Decission Tree," *J. Inform. Kaputama*, vol. 5, no. 2, pp. 391–397, 2021, doi: 10.59697/jik.v5i2.276.