PETIR: Jurnal Pengkajian dan Penerapan Teknik Informatika
Vol. 17, No. 2, September 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i2.2416

Performance Comparison of VGG16, Mobilenet, And Xception Model Architecture in Rice Plant Leaf Identification

Riki Ruli A. Siregar[1,4)]; Nurin Masyitha Safiera [2]; Abdul Haris[3,4]*

1, 2, 3. Informatika, Institut Teknologi PLN, Menara PLN, Jl. Lingkar Luar Barat, Duri Kosambi, Cengkareng, Jakarta Barat, DKI Jakarta 11750 Indonesia

4. Department of Computer Science, IPB University, Jawa Barat 16680, Jl. Raya Darmaga Kampus IPB, Babakan, Kec. Dramaga, Kabupaten Bogor, Jawa Barat 16680 Indonesia

[*)]*Email: rulirikisiregar@apps.ipb.ac.id*

Received: 16 Maret 2024 | Accepted:14 November | Published: 10 Januari 2025

ABSTRACT

Rice is one of the world's most essential staple foods, particularly in Indonesia. Proper nutrition is crucial for the growth and development of rice plants, as nutrient deficiencies can impact both growth processes and the quality of the crop at harvest. This study utilizes a dataset from Kaggle, comprising 1,190 images of rice leaves, categorized into two classes: Sufficient and Deficient. The data were divided with an 80% training set, 10% testing set, and 10% validation set. Three model architectures—VGG16, MobileNet, and Xception—were evaluated using Jupyter and Google Colab as the primary tools. Experiments were conducted with 10 epochs and batch sizes of 32 and 64. The highest accuracy achieved was 78.15% and 76.47% for VGG16, 82.69% and 86.55% for MobileNet, and 82.33% and 88.24% for Xception. Overall, the Xception model performed best, with an accuracy of 88.24% using a batch size of 32 on Jupyter.

Keywords: *Convolutional Neural Network, nutrition, CNN model architecture, VGG16, MobileNet Xception, rice*

Vol. 17, No. 2, September 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i2.2416

1. INTRODUCTION

Urban agricultural land is rapidly decreasing, primarily due to conversion into industrial and residential zones driven by economic, social, and population growth factors, as well as land limitations [1][2]. Over the past decade, this trend has significantly affected agricultural spaces in cities. The hydroponic method has emerged as a viable solution to overcome limitations in available agricultural land [3]. Rice, one of the world's most important staple crops, is particularly vital in Indonesia. However, while population growth drives up demand for rice, production struggles to keep pace due to challenges such as adverse weather, temperature fluctuations, and plant diseases. This imbalance highlights the need for strategies that promote healthy rice crop growth and disease management [4][5]. Adequate nutrition plays a crucial role in plant productivity and growth rate, as nutrient deficiencies can disrupt these processes, leading to reduced yields and visible abnormalities in plants [6]. One essential nutrient for rice, especially during the vegetative growth phase, requires monitoring to optimize plant health [7][8]. The Leaf Color Chart (LCC) is commonly used by farmers to visually assess plant fertility, yet this manual approach is time-intensive and can lead to inconsistent results due to subjective perceptions among users [9][10][11]. Convolutional Neural Networks (CNNs), as a form of deep learning, have gained popularity for their advancements in fields like pattern recognition, including image and object classification [12][13][14]. Previous studies in plant health monitoring applied CNNs to detect diseases in crops such as apples, using a dataset of 3,151 leaf images from PlantVillage. The CNN model, based on the LeNet-5 architecture, demonstrated high accuracy and decreasing loss across increasing epochs, indicating good model performance. For example, at 50, 75, and 100 epochs, the training model achieved an average accuracy of 99.2% with a 0.063 loss, while testing accuracy averaged 94.9% with a 0.2777 validation loss. Among the apple classes, Black Rot achieved the highest accuracy at 100%, while Apple Scab was the lowest at 74.4%, resulting in an average accuracy of 89.62% [15]. Additional research has used CNNs to develop systems for detecting plant conditions in lettuce [16] and rice leaves [17]. These studies included processes such as image capture, processing, and classification [18][19][20]. A partition test with 90% training data and 10% testing data showed the highest accuracy at 97.6% based on a confusion matrix. Further testing with 250 epochs and a batch size of 15 yielded an impressive 99.3% accuracy and 99.2% accuracy, respectively [16]. This research aims to identify nutrient deficiencies in rice leaves using CNNs, comparing the performance of the VGG16, MobileNet, and Xception model architectures. By analyzing their accuracy, this study seeks to determine the most effective model for providing accurate nutrient recommendations for rice crops.

2. RESEARCH METHODS

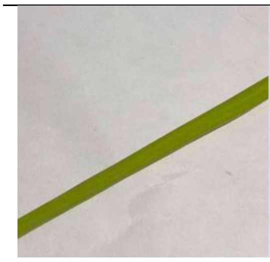
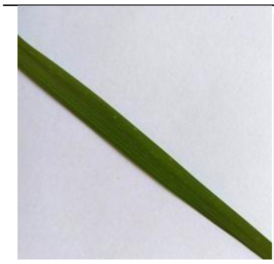
The research begins with problem identification, followed by a literature review aimed at building on existing studies. Next, data collection is conducted based on a needs analysis. The research method includes the stages of data preprocessing, architecture design, model training and evaluation, and concludes with documentation and report writing. These stages are detailed as follows:

2.1. Problem identification

In the initial stage of this research, the problems to be addressed are identified. The focus of this study is the provision of nutrients to hydroponic rice plants. To address this, the research proposes a system utilizing various CNN architectures, which will be compared to determine the most effective mode.

Vol. 17, No. 2, September 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i2.2416

Table 1. Nutrient Classification of Rice Leaf Images Dataset

image	class
 	Less
 	Normal

This Table 1 presents examples from a dataset used to classify rice leaves based on nutrient levels, divided into two classes: "Less" and "Normal." The "Image" column contains sample images of rice leaves categorized by nutrient status, with each image showing visual differences in leaf coloration and texture that correlate with nutrient levels. The "Class" column indicates the nutrient classification of each leaf image, with two categories: "Less" represents rice leaves with nutrient deficiencies, typically appearing lighter in color due to a lack of essential nutrients, while "Normal" represents rice leaves with sufficient nutrients, which appear healthier and greener, indicating balanced nutrient levels. This dataset is intended to train and evaluate models for nutrient deficiency detection in rice plants using image classification techniques.

2.2. Data collection method

This stage involves collecting data on healthy rice plant leaves to create a dataset. The dataset used in this study was obtained from the Kaggle website in June 2022. The data consists of images in .jpg format. The images are divided into two classes: the "Sufficient" class, containing images of rice leaves with adequate nutrients, and the "Less" class, containing images of rice leaves with nutrient deficiencies. Each class contains 595 images, resulting in a total of 1,190 images for this dataset.

2.3. Pre-Processing

At this stage the data that has been obtained is pre-processed first with the aim that the existing image data has a uniform size between one another and can speed up the training process. There are 2 (three) steps involved in data pre-processing, namely:

1. Equalizing the size of images in the dataset, so that images that previously had a larger or smaller size can become the same size.
2. The process of converting images into arrays, this is done so that existing data can be normalized before entering the next process.

Vol. 17, No. 2, September 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i2.2416

2.4. Architecture Model Design

In this section, we will do some training and testing on several model architectures, the model architectures are VGG16, MobileNet, and Xception. The dataset is divided into 80% for training data and 20% for testing and validation data, each of which is divided into 828 training data, 104 testing data, and 104 validation data.

2.5. Evaluation of architectural design results

The results of the evaluation process of the model that has been trained, in this case the confusion matrix will be used to obtain accuracy, recall, precision, and F1-Score values. Then, the evaluation results obtained will be used as a basis for whether the model that has been made is good enough or still needs more training. The following formulas are used in the evaluation process using confusion matrix:

$$Accuracy = \frac{\sum_{i=1}^{N} TP(i=j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} C(i,j)}\ x\ 100\% \quad (1)$$

$$Precision(Ci) = \frac{TP(i=j)}{TP(i=j)+\sum_{j=1}^{N} C(j,i)}$$
$$(2)$$

$$Recall(Ci) = \frac{TP(i=j)}{TP(i=j)+\sum_{i=1}^{N} C(i,j)}$$
$$(3)$$

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

3. RESULTS AND DISCUSSION

The results obtained in this study are the accuracy level of several models as an identification of hydroponic rice plant nutrition obtained in the design of several model architectures using the CNN method, the accuracy results of each model architecture used in this study, which have been carried out 3 times test data with batch size values of 32 and 64 conducted in Jupyter and Google Collaboratory. Training results and model comparison using a dataset of 952 images as training data and 119 images as test and validation data. The following data are the accuracy results obtained:

3.1. VGG16.

The following are the accuracy results during the research that has been carried out using the VGG16 model architecture:

Table 2. VGG16 accuracy result batch size = 32

Jupyter	Google Colaboratory
68,91%	73,95%
72,27%	76,47%
74,79%	78,15%

PETIR: Jurnal Pengkajian dan Penerapan Teknik Informatika
Vol. 17, No. 2, September 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i2.2416

Table 3. VGG16 accuracy result batch size = 64

Jupyter	Google Colaboratory
68,91%	69,75%
75,63%	75,63%
76,47%	73,95%

Table 2 and Table 3 show the accuracy results on the VGG16 model with batch size = 32. In results 1 to 3 there is a difference, although only a few percent, which shows that each training of a model with the same data and parameters does not necessarily produce the same accuracy.

In this case, the accuracy obtained on Google Collaboratory is relatively higher than the accuracy obtained on Jupyter.

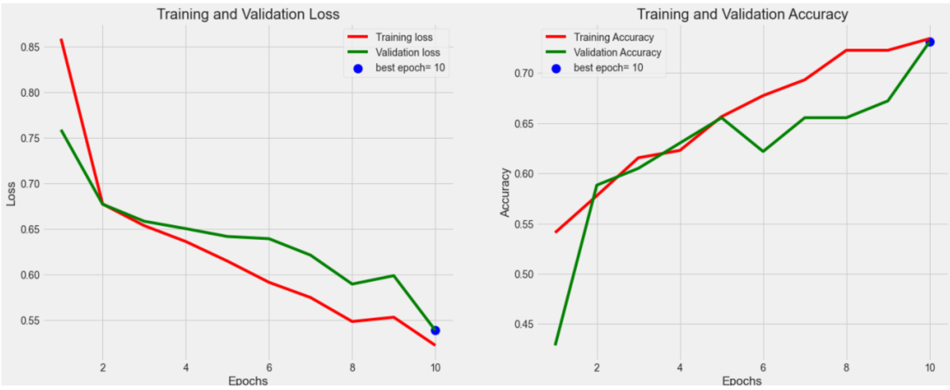


Fig 1. VGG16 Architecture Training Chart on Jupyter

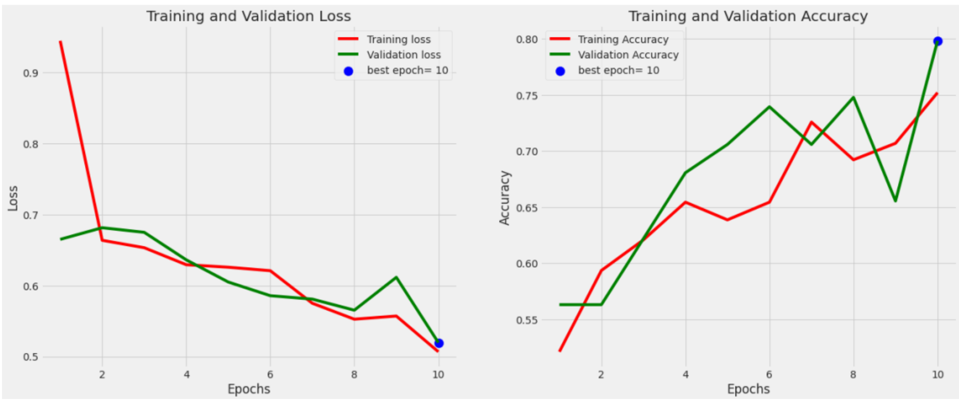


Fig 2. VGG16 Architecture Training Chart on Google Colaboratory

In Figure 1, the best epoch or process is at the 10th epoch in one of the model trainings conducted on Jupyter using the VGG16 model architecture. In Figure 2, the training process graph shows that the best process or the best epoch is at the 10th epoch in one of the model trainings using the VGG16 model architecture on Google Collaboratory.

3.2. MobileNet

The following are the accuracy results during the research that has been carried out using the

Vol. 17, No. 2, September 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i2.2416

MobileNet model architecture:

Table 4. MobileNet accuracy result batch size = 32

Jupyter	Google Colaboratory
79,83%	78,85%
76,47%	82,69%
86,55%	81,73%

Table 5. MobileNet accuracy result batch size = 64

Jupyter	Google Colaboratory
80,67%	79,83%
84,03%	81,73%
81,51%	79,81%

From the accuracy results in Table 4 and Table 5, it can be seen that the resulting accuracy has a different value even though the parameters entered in the process are the same. The best model accuracy of the MobileNet model architecture in this study is 86.55% with batch size = 32 and run on Jupyter.

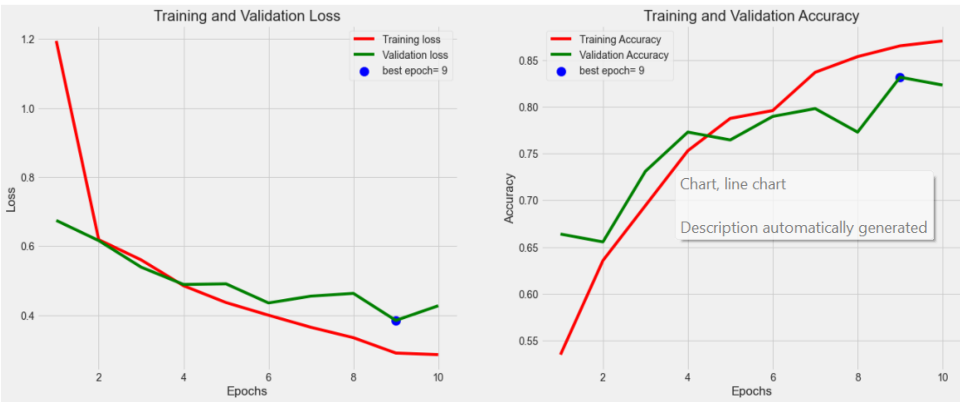


Fig 3. MobileNet Architecture Training Chart on Jupyter

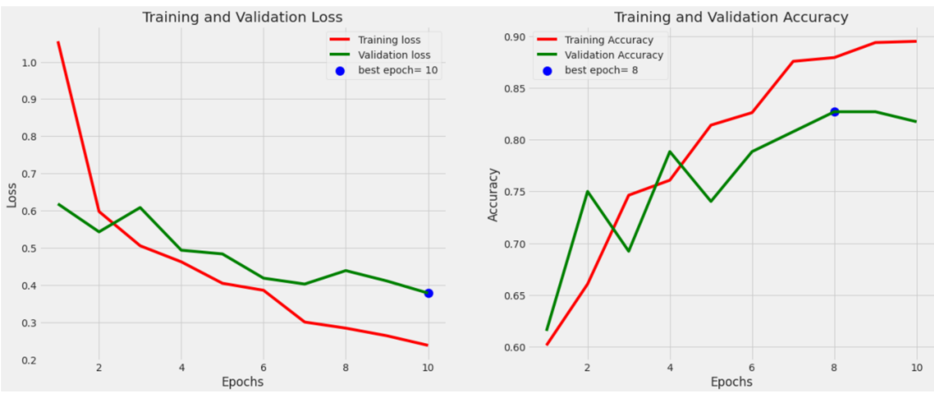


Fig 4. MobileNet Architecture Training Chart on Google Colaboratory

Vol. 17, No. 2, September 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i2.2416

Figure 3 explains that the best epoch or the best training process in the graph is in the 9th epoch in one of the model trainings carried out on Jupyter with the MobileNet architecture model. Figure 4 shows that the best epoch found in one of the trainings in Google Collaboratory with the MobileNet architecture model is in the 10th process or epoch.

3.3. Xception

The following are the accuracy results during the research that has been carried out using the Xception model architecture:

Table 6. Xception accuracy result batch size = 32

Jupyter	Google Colaboratory
87,39%	78,99%
84,03%	78,15%
88,24%	82,35%

Table 7. Xception accuracy result batch size = 64

Jupyter	Google Colaboratory
79,83%	74,79%
68,91%	78,15%
80,67%	79,83%

The accuracy obtained during the experiment can be seen in Table 6, which shows that at an input batch size of 32, the accuracy results obtained on the Xception architecture using Jupyter tend to be higher than using Google Collaboratory. At batch size 64, Table 7, there is no significant change from the other models. Then the best model from the Xception architecture test results in this study is 88.24% using Jupyter and an input batch size of 32.

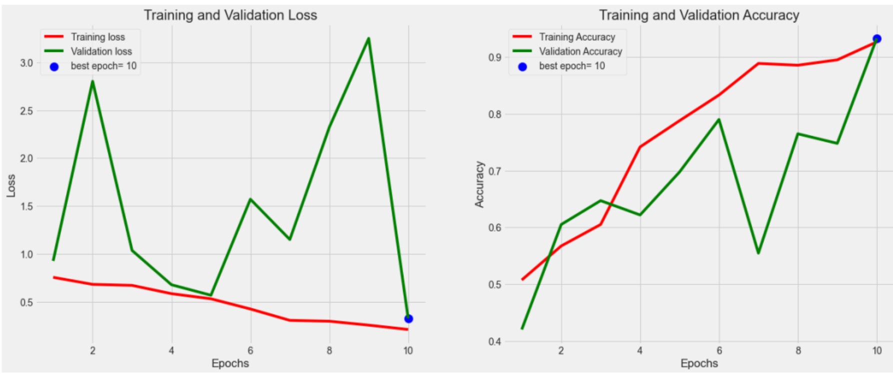


Fig 5. Xception Architecture Training Chart on Jupyter

Vol. 17, No. 2, September 2024, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v17i2.2416

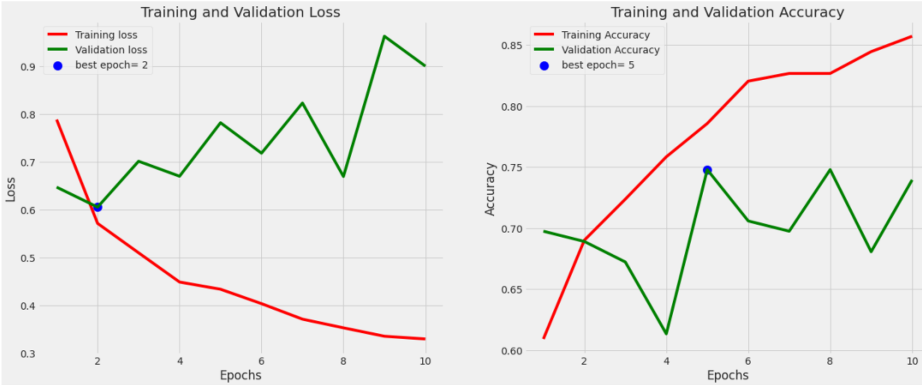


Fig 6. Xception Architecture Training Chart on Google Colaboratory

Figure 6 shows that the best epoch in one of the Xception models trained in Google Collaboratory is at epoch 5. When running the training process, there are some differences in each processing on a model and the tools used, these differences are mainly in the computation time and the accuracy generated at the epoch entered in this study, which is 10.

Overall, the computation time is shorter when the training process is carried out on Google Collaboratory, and of the three model architectures tested during the research, the shortest training process is obtained on the Mobilenet model architecture. The slowest training process occurs in the Jupyter and Xception model architectures.

In the results of the research done, the model is built and trained using 1190 image data, which is divided into 592 data as training data and 119 data, each of which is used as test and validation data. In this training process, there is a loss or training loss which shows the loss function value of the training and prediction data on the trained model, then there is training accuracy which shows the calculation of the accuracy value in the training process and prediction data on the model, then validation loss which is the calculation of the loss function of the validation and prediction data on the trained model, and there is validation accuracy which shows the calculation of the accuracy value of the validation and prediction data. In this study, it was found that differences in batch size and also the location of the training execution on multiple models being trained can affect the level of accuracy that will be generated as well as the length of time the computation runs when training data. The computation will be faster if done on Google Collaboratory, but requires a stable network for the computation to run smoothly. The computation is slower when done on Jupyter, but using Jupyter does not require a signal and the computation will continue as long as the computer is not turned off. The highest accuracy result in this study is shown by the Xception model architecture with batch size = 32 inputs and data training performed in Jupyter, where the accuracy is 88.24%.

After model training, each model was tested. Testing is done using the Iconfusion matrix by determining the True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) values in order to calculate the accuracy, precision, recall and F1 score. In the model with the best accuracy, the TP value is 52, TN 53, FP 10 and FN 4. Based on the confusion matrix calculations, the resulting accuracy is 88%, which corresponds to the accuracy during model training. Then the precision result is 83%, the recall is 92%, the F1 score is 88%.

4. Conclusions And Suggestions

Based on the research findings and discussion, several conclusions can be drawn. The development of a model for identifying the nutritional needs of rice plants using rice leaf images involves multiple stages, including data collection and class division, image preprocessing, model architecture selection (VGG16, MobileNet, Xception), convolution, max pooling, flattening, fully connected layers, and model training with 10 epochs. From the model training process in this study, the best accuracy results for each architecture were as follows: VGG16 achieved 78.15% accuracy with Google Colab and 76.47% with Jupyter; MobileNet achieved 82.69% accuracy with Google Colab and 86.55% with Jupyter; and Xception achieved 82.35% accuracy with Google Colab and 88.24% with Jupyter. Overall, the accuracy results obtained are quite satisfactory. Model testing using the confusion matrix yielded an accuracy of 88%, precision of 83%, recall of 92%, and an F1 score of 88%. Future research could develop this model into an application or tool for real-time identification by implementing preprocessing without augmentation, and later adding image augmentation at the data preprocessing stage. Increasing the dataset size for model training is also recommended to enhance accuracy and improve model performance further.

REFERENCES

[1] A. D. Yonida, “Cara Budidaya Padi untuk Memperoleh Hasil Optimal,” *Farming.Id*, 2018.

[2] P. Wardhana *et al.*, “Panduan Teknologi Budidaya Padi System of Rice Intensification (SRI),” *Badan Penelit. dan Pengemb. Pertan.*, pp. 1–20, 2015.

[3] “Mengenal Budidaya Padi Hidroponik dan Keuntungannya Halaman all - Kompas.com.” [Online]. Available: https://agri.kompas.com/read/2023/02/12/233624284/mengenal-budidaya-padi-hidroponik-dan-keuntungannya?page=all#page2. [Accessed: 23-Sep-2023].

[4] R. R. A. Siregar, K. B. Seminar, S. Wahjuni, and E. Santosa, “Vertical Farming Perspectives in Support of Precision Agriculture Using Artificial Intelligence: A Review,” *Computers*, vol. 11, no. 9, 2022, doi: 10.3390/computers11090135.

[5] D. D. Avgoustaki and G. Xydis, *How energy innovation in indoor vertical farming can improve food security, sustainability, and food safety?*, 1st ed., vol. 5. Elsevier Inc., 2020.

[6] L. Wang *et al.*, “Estimation of paddy rice nitrogen content and accumulation both at leaf and plant levels from uav hyperspectral imagery,” *Remote Sens.*, vol. 13, no. 15, 2021, doi: 10.3390/rs13152956.

[7] J. Yang *et al.*, “Application of fluorescence spectrum to precisely inverse paddy rice nitrogen content,” *Plant, Soil Environ.*, vol. 61, no. 4, pp. 182–188, 2015, doi: 10.17221/7/2015-PSE.

[8] J. Yang, L. Du, W. Gong, J. Sun, S. Shi, and B. Chen, “Application of the chlorophyll fluorescence ratio in evaluation of paddy rice nitrogen status,” *Plant, Soil Environ.*, vol. 63, no. 9, pp. 396–401, 2017, doi: 10.17221/460/2017-PSE.

[9] Y. Sari, M. Alkaff, and M. Maulida, “Classification of Rice Leaf using Fuzzy Logic and Hue Saturation Value (HSV) to Determine Fertilizer Dosage,” *2020 5th International Conference on Informatics and Computing, ICIC 2020*. 2020, doi: 10.1109/ICIC50835.2020.9288585.

[10] G. B. Dela Cruz, “Nitrogen deficiency mobile application for rice plant through image processing techniques,” *Int. J. Eng. Adv. Technol.*, vol. 8, no. 6, pp. 2950–2955, 2019, doi: 10.35940/ijeat.F8721.088619.

[11] J. W. Orillo, G. J. Emperador, M. G. Gasgonia, M. Parpan, and J. Yang, “Rice plant nitrogen level assessment through image processing using artificial neural network,” *2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, HNICEM 2014 - 7th HNICEM 2014 Joint with 6th International Symposium on Computational Intelligence and Intelligent In*. 2014, doi: 10.1109/HNICEM.2014.7016187.

[12] K. Liang, Y. Wang, L. Sun, D. Xin, and Z. W. Chang, “A Lightweight-Improved CNN Based on VGG16 for Identification and Classification of Rice Diseases and Pests,” *Lecture Notes in*

Electrical Engineering, vol. 813. pp. 195–207, 2022, doi: 10.1007/978-981-16-6963-7_18.

[13] R. Sharma and A. Singh, "Big bang–big crunch-CNN: an optimized approach towards rice crop protection and disease detection," *Arch. Phytopathol. Plant Prot.*, vol. 55, no. 2, pp. 143–161, 2022, doi: 10.1080/03235408.2021.2003054.

[14] A. S. Reddy and J. Thomas, "Leaf Disease Identification in Rice Plants Using CNN Model," *Lecture Notes on Data Engineering and Communications Technologies*, vol. 114. pp. 17–31, 2022, doi: 10.1007/978-981-16-9416-5_2.

[15] G. Wicaksono, S. Andryana, and B. -, "Aplikasi Pendeteksi Penyakit Pada Daun Tanaman Apel Dengan Metode Convolutional Neural Network," *JOINTECS (Journal Inf. Technol. Comput. Sci.*, vol. 5, no. 1, p. 9, 2020, doi: 10.31328/jointecs.v5i1.1221.

[16] M. F. Reza, Y. Purwanto, and C. Setianingsih, "Deteksi Kondisi Tanaman Selada Berdasarkan Citra Menggunakan Metode Convolutional Neural Network (Cnn) Lettuce Plant Condition Detection Based on Image Using Convolutional Neural Network (Cnn) Method," *Proceeding Eng. J.*, vol. 7, no. 3, pp. 9330–9338, 2020.

[17] Teuku Rifki Dhulul Fata, R. R. A. Siregar, and Indrianto, "Detection of brown spot and leaf blight in rice with the You Only Look Once (YOLO) algorithm," *Petir*, vol. 16, no. 1, pp. 117–130, 2023, doi: 10.33322/petir.v16i1.2003.

[18] T. Y. Lee, I. A. Lin, J. Y. Yu, J. M. Yang, and Y. C. Chang, "High Efficiency Disease Detection for Potato Leaf with Convolutional Neural Network," *SN Comput. Sci.*, vol. 2, no. 4, 2021, doi: 10.1007/s42979-021-00691-9.

[19] C. H. Son, "Leaf spot attention networks based on spot feature encoding for leaf disease identification and detection," *Appl. Sci.*, vol. 11, no. 17, 2021, doi: 10.3390/app11177960.

[20] A. K. M. S. Hosain *et al.*, "Rice Leaf Disease Detection with Transfer Learning Approach," *4th IEEE International Conference on Artificial Intelligence in Engineering and Technology, IICAIET 2022*. 2022, doi: 10.1109/IICAIET55139.2022.9936780.