JSCDM

JOURNAL OF SOFT COMPUTING AND DATA MINING
e-ISSN: 2716-621X

Vol. 7 No. 2 (2026) 17-29
https://publisher.uthm.edu.my/ojs/index.php/jscdm

Ensemble CNN Model for Detecting Rice Leaf Diseases by Integrating MobileNetV2 and EfficientNet-B1 Architectures

Kusworo Adi[1]*, Agus Setiadi[2], Catur Edi Widodo[1], Aris Puji Widodo[3], Resma Adi Nugroho[3], Aji Setiawan[4]

[1] *Diponegoro University, Department of Physics, Faculty of Science and Mathematics, Semarang, 50275, Central Java, INDONESIA*
[2] *Diponegoro University, Department of Animal, Faculty of Animal Husbandry and Agriculture, Semarang, 50275, Central Java, INDONESIA*
[3] *Diponegoro University, Department of Informatics, Faculty of Science and Mathematics, Semarang, 50275, Central Java, INDONESIA*
[4] *Darma Persada University, Department of Information Technology, Faculty of Engineering, East Jakarta, 139450, INDONESIA*

*Corresponding Author: kusworoadi@lecturer.undip.ac.id
DOI: https://doi.org/10.30880/jscdm.2026.07.02.002

Article Info

Received: 24 May 2025
Accepted: 1 January 2026
Available online: 26 June 2026

Keywords

Rice disease detection, Convolutional Neural Network (CNN), Hyperparameter Optimizer, ensemble model

Abstract

Rice diseases significantly impact agricultural productivity, requiring effective methods for early detection to prevent substantial damage. This paper introduces an ensemble Convolutional Neural Network (CNN) model that integrates MobileNetV2 and EfficientNet-B1 architectures for the identification of rice leaf diseases. The algorithm employs a dataset comprising 5,936 rice leaf photos from Mendeley Data to classify five disease categories: Bacterial blight, Blast, Brownspot, Tungro, and Healthy. The ensemble method employs hyperparameter optimization by Bayesian and Random techniques, integrating dropout, learning rate adjustment, and kernel regularization to achieve a compromise between model accuracy and generalization. The model attained a training accuracy of 99.99% and a validation accuracy of 99.90%, showcasing robustness and accuracy. The TensorFlow Lite format facilitates integration into mobile applications, providing farmers with a portable and effective solution. This system enhances agriculture by equipping stakeholders with advanced disease management tools, increasing yields, and promoting sustainable practices.

1. Introduction

Indonesia is an agriculturally abundant nation, rich in many natural resources due to its tropical position. Indonesia undergoes two distinct seasons: wet and dry, attributable to its tropical position [1]. The agricultural sector is vital for national progress in developing nations such as Indonesia, and advancements in agriculture foster social development, guaranteeing that inclusive growth includes those dependent on farming [2], [3]. The diversity of natural resources offers substantial opportunities for Indonesia's agricultural sector, contingent upon the management of activities from upstream to downstream processes. Notwithstanding the considerable potential of Indonesia's agricultural sector, a pervasive challenge among Indonesian farmers endures: the substandard quality of produced agricultural products, resulting in reduced selling prices, less consumer interest, and consequent financial losses for farmers. The inferior quality of the crop may stem from several factors, including rice types, nutritional and water levels, the presence or absence of diseases, and several other environmental conditions impacting the rice [4], [5].

The inferior quality of the crop contributes to Indonesia's ongoing importation of agricultural products to meet public demand for higher-quality goods at comparatively low prices. Indonesia continues to import agricultural products due to insufficient quantity and quality to satisfy the needs of its population [6]. Data from the Central Bureau of Statistics Indonesia (BPS) indicates a decline in lowland rice production, decreasing from 6.20 tonnes in 2020 to 5.78 tonnes in 2021. This transpired as a result of substandard rice varieties and the use of inappropriate fertilizers. The treatment of diseases in rice crops is similarly crucial. Daniya (2023) argued that the management of illnesses in rice plants is essential, as insufficient treatment could significantly damage the plant [7]. The management of rice affected by fungus will differ from that of rice influenced by Tungro.

Rice as a crucial food source for a substantial portion of the Indonesian population, is negatively impacted by diseases that reduce yield and threaten farmer income and national food security. Outbreaks of severe illnesses can lead to production declines of several percentage points, forcing farmers to replant, increase input costs, or rely on chemical control methods that are often applied reactively and ineffectively [8],[9]. In numerous agricultural sectors, disease identification continues to rely on manual field inspections by farmers or extension agents, a process that is labor-intensive, subjective, and often performed only after symptoms have escalated significantly. This scenario heightens the likelihood that sick plants would go unnoticed, facilitating the dissemination of diseases and resulting in additional economic detriment. Consequently, an effective and prompt disease identification system utilizing image analysis is crucial for facilitating rapid decision-making, minimizing wasteful pesticide use, and ensuring the stability of rice yield.

Prompt diagnosis of diseases in rice is crucial to detect problems before pests or pathogens spread and affect more rice plants. Therefore, a solution is necessary to harness the potential of science and technology in the agricultural sector to attain significant modernization and digitization [10]. Modern technological innovations, such as artificial intelligence, will significantly aid farmers in the early identification of rice diseases. A technology in artificial intelligence is the Convolutional Neural Network (CNN). A CNN is a method for image processing that enables the analysis of images and data processing [11]. CNN aims to identify viruses through rice plant leaves, allowing farmers to recognize plant diseases at an early stage. CNN will be used on mobile devices to improve accessibility for farmers. The CNN model will be constructed utilizing the ensemble model methodology by integrating MobileNetV2 and EfficientNet-B1 to get improved predictive accuracy.

2. Related Work

Zhang et al. (2021) conducted a study employing the MobileNetV2 architecture, augmented with channel pruning techniques, to detect diseases in rice plants [12]. This model achieved a 50% reduction in size, a 74% decrease in memory consumption, and a 49% reduction in FLOPS, while attaining an accuracy of 90.84%. A work by Shovon et al. (2023) introduced PlantDet, an ensemble model utilizing weighted averages that combines the InceptionResNetV2, EfficientNetV2L, and Xception models for detecting illnesses in rice plants and betel leaves [13]. The model demonstrated exceptional performance, achieving an accuracy of 98.53% for the rice plant dataset and 96.02% for the betel leaf plant dataset. The amalgamation of these models enhances precision and mitigates the challenges of underfitting and overfitting. A recent study by Atila et al. (2024) investigated the EfficientNet architecture for classifying plant illnesses [14]. The EfficientNet B5 and B4 models demonstrated exceptional performance, achieving accuracies of 99.97% on the enhanced dataset, while the EfficientNet B1 model reached an accuracy of 99.33%, offering an efficient option for resource-limited applications.

Prior research employing CNN, especially in model ensembles, has shown effectiveness in attaining high-accuracy plant disease diagnosis. However, the model's effectiveness depends on the quality and quantity of the data used, as well as the technological setup applied during training [15][16]. This study employs a comparable methodology by integrating MobileNetV2 and EfficientNetB1 in a model ensemble to identify plant diseases, aiming to create an efficient and precise model. This method enhances precision and facilitates deployment on devices with constrained computational resources, rendering it a versatile option for many practical applications.

3. Methods

Developing an ensemble model for the identification of diseases in rice plant leaves involves multiple stages. The process will commence with the gathering and analysis of datasets, culminating in the evaluation of the model. The stages are depicted in the flowchart in Figure 1.

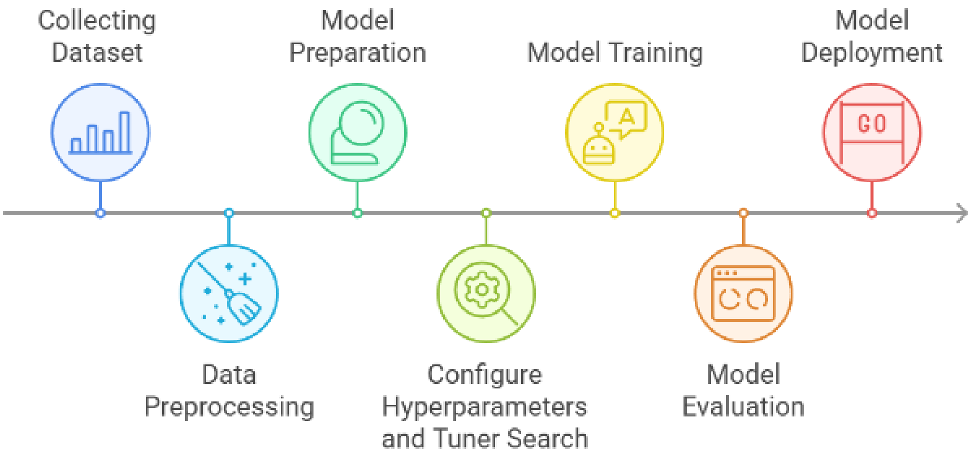
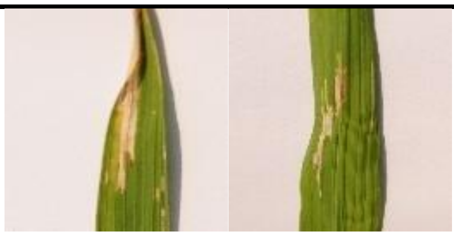


Fig. 1 *Model design flow diagram*

3.1 Collecting Dataset

The collection comprises photos of rice plant leaves afflicted by diseases or pests. The data was gathered via the open-source platform Mendeley Data, consisting of around 5.936 photos classified into five disease categories: Bacterial blight, Blast, Brown spot, Tungro, and Healthy condition [17][18] with detail as shown in Table 1.

Table 1 *Class of dataset*

Class	Images	Label	Characteristic
0		Bacterial Blight	Water-soaked, long, narrow lesions that start off yellow and later become brown are found along the leaf margins.
1		Blast	lesions that resemble diamonds or spindles, with brown edges with gray or white cores.
2		Brown spot	Small, round to oval lesions with gray cores and brown or reddish-brown borders.

3		Healthy	Leaves that are vibrant green and devoid of any imperfections or discolorations
4		Tungro	Leaf yellowing that begins at the tip and moves downward, frequently with an orange tint

3.2 Data Preprocessing

The collected data will be divided into training data and validation data. The data is allocated in a 70% to 30% proportion. The data must be modified to align with the model's input requirements. This phase focuses on altering the data to improve and optimize the model's image recognition abilities [19]. The image size will be modified, and the details of image augmentation preparation are illustrated in Figure 2.

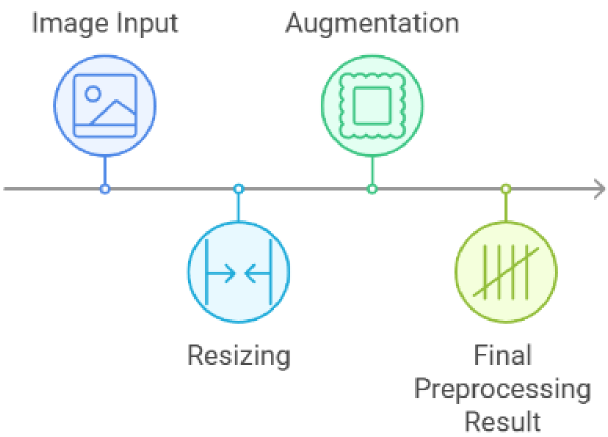


Fig. 2 *Data flow diagram preprocessing*

3.3 Model Preparation

An ensemble model amalgamates various models to improve accuracy and performance compared to singular models. This strategy integrates the predictive outcomes of many models through methods like weighted averaging, leading to enhanced and more resilient forecasts that mitigate underfitting and overfitting issues [20]. Prior to employing the model for training within the ensemble framework, the foundational model must be meticulously developed. The chosen models for implementation are MobileNetV2 and EfficientNet-B1. The input dimensions for each model will be standardized to (224, 224) to facilitate the future training process; see the ensemble model schematic in Figure 3.

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 17-29

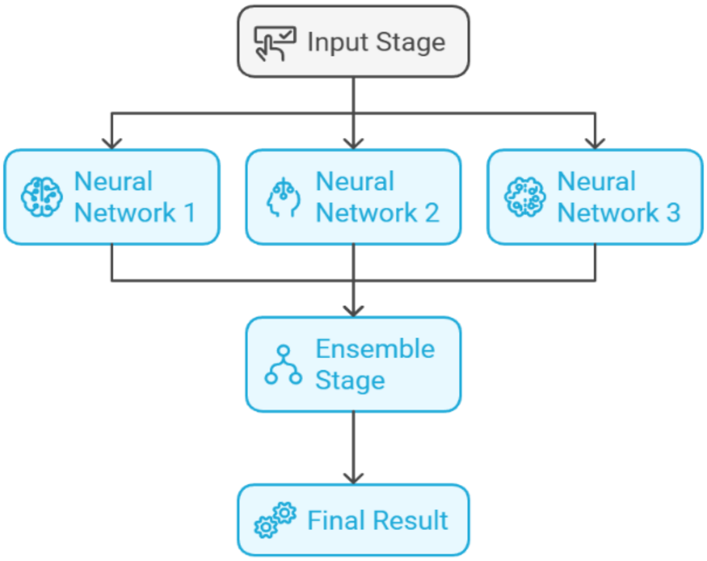


Fig. 3 *Ensemble model diagram*

3.4 Configure Hyperparameter and Tuner Search

The proposed ensemble incorporates two pre-trained CNN backbones, resulting in a high-dimensional and non-convex hyperparameter space. Manual tuning or grid search would incur significant computing costs and be susceptible to suboptimal designs. Consequently, we utilize Bayesian Optimization as a sample-efficient method to investigate potential areas of the search space, supplemented with Random Search as a baseline to evaluate the resilience of the chosen configuration. This combination enables the acquisition of near-optimal hyperparameters at a reasonable computational expense.

The hyperparameter tuning method entails defining essential hyperparameters using the Bayesian Optimization technique. This method use a probabilistic model to systematically seek optimal parameter values by balancing exploration and exploitation. The Bayesian Optimization process is governed by Equation 1 [21].

$$\hat{y} = \arg\max_{x \in X}((\mu(x) + k\sigma(x)) \tag{1}$$

where $\mu(x)$ is the predicted mean of the Gaussian Process, σ(x) represents the standard deviation, and κ controls the exploration-exploitation trade-off. The activation function is selected from ReLU, ReLU6, or SiLU to improve model adaptability to non-linear inputs, while the learning rate is set at 0.001, 0.0001, or 0.00001 to control the learning process. Furthermore, dropout is configured at 0.3, 0.4, or 0.5 to mitigate overfitting, while the kernel regularizer utilizes values of 0.0001, 0.001, or 0.01 to enhance model generalization. The merging layer technique utilizes two options: concat (combining layer outputs without reducing dimensionality) or avg (calculating the average output), with the merged layer implemented through Concatenate() or Average() depending on hyperparameter choice. The selected ranges are based on prevalent settings in CNN literature, offering a sensible balance among convergence speed, stability, and overfitting management for medium-scale image datasets [22]. This Bayesian Optimization-driven approach enables the identification of the optimal combination to enhance model performance on validation data while mitigating the risks of overfitting and underfitting [23].

The Random Search optimizer was utilized with three trials to evaluate the efficacy of hyperparameter optimization strategies through the analysis of parameter combinations. The Random Search technique is defined by the stochastic sampling of hyperparameters, as specified in Equation 2 [24].

$$\theta_i = random(a, b) \tag{2}$$

where θi denotes the hyperparameter values, and a and b, define the sampling range. The Random Search optimizer was chosen for this assessment because of its capacity to investigate a broad spectrum of hyperparameter combinations without necessitating significant computer resources [25].

3.5 Model Training

The training phase of a CNN is an iterative process aimed at achieving a precise model. Image data is converted into tensor format, extracting features through convolutional and pooling layers. These attributes are categorized with a completely interconnected layer, yielding class predictions. The backpropagation algorithm modifies neuron weights based on the difference between the expected output and the actual class (loss). The training phase is affected by the quantity of training data, model design, training methodology, and hyperparameters [26], as illustrated in Figure 4 of the CNN Training Process.

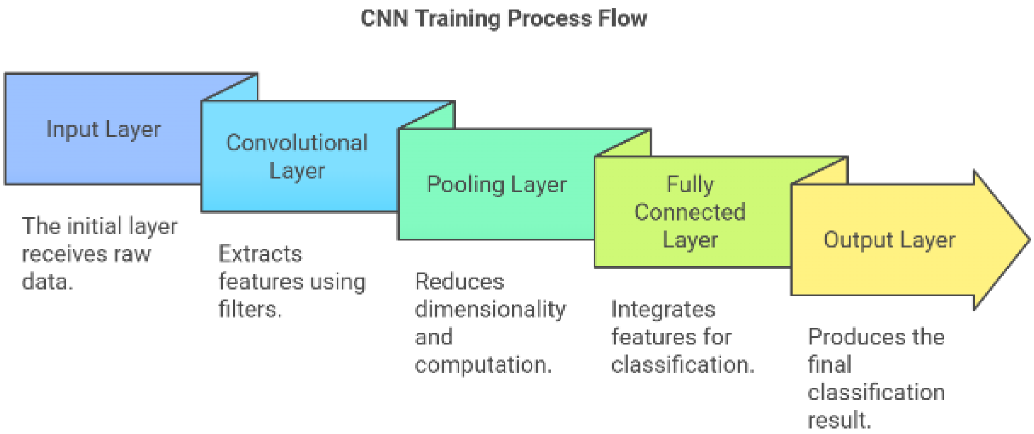


Fig. 4 CNN Training Process

3.6 Model Evaluation

Model evaluation use accuracy metrics and loss to assess performance during training and validation. These metrics aim to evaluate the model's ability to recognize patterns in training data and to generalize these patterns to new data. The accuracy metric assesses the model's ability to make right predictions, whereas the loss quantifies the extent of prediction errors incurred by the model. The confusion matrix was essential to determine that the generated model can address classification problems [27]. Figure 5 illustrates the model assessment utilizing F1 score values, recall, precision, and accuracy.

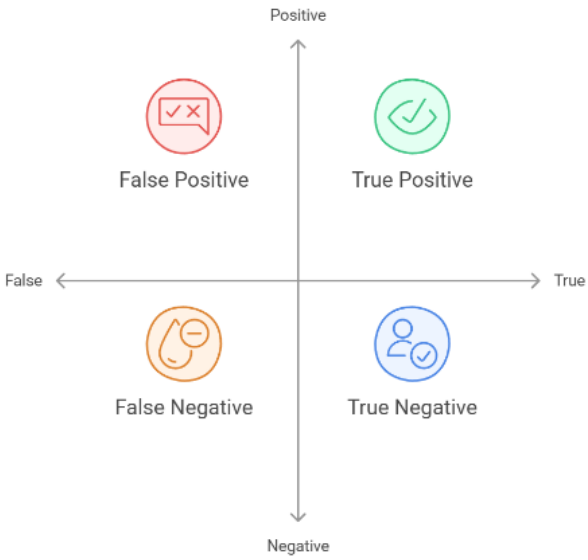


Fig. 5 *Confusion matrix to classify*

In classification evaluation, the model's performance is often assessed using the elements of a confusion matrix, which provide detailed insights into predictive accuracy. A True Positive (TP) refers to instances where the model correctly detects the existence of a condition or violation, aligning with its real manifestation. A False Positive (FP) occurs when the model mistakenly predicts a condition or violation that does not exist, leading to overprediction. A False Negative (FN) refers to cases where the model fails to recognize the presence of a condition or violation, despite its actual occurrence, resulting in underprediction. A True Negative (TN) signifies accurate predictions where the model accurately identifies the absence of a condition or violation, hence representing the genuine negative state. These parameters collectively serve as critical indicators for evaluating model reliability and its ability to balance sensitivity and specificity in practical applications. The calculation of the confusion matrix for the indices is delineated by Equations 3 to 6.

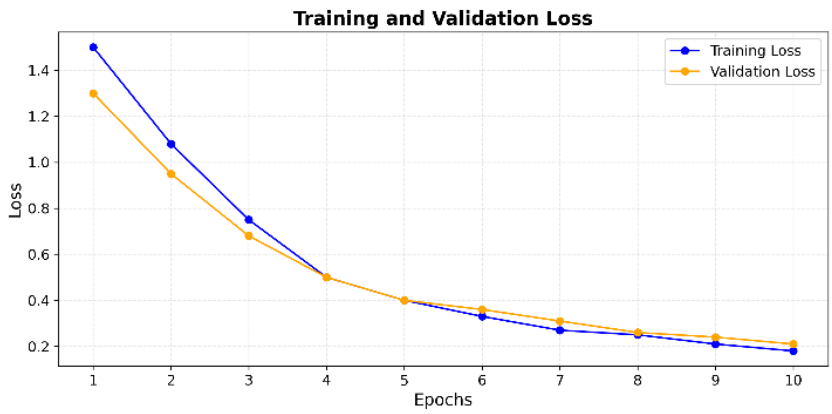
$$accuracy = \frac{TP+TN}{TP+FP+TN+FN} * 100\% \tag{3}$$

$$precision = \frac{TP}{TP+FP} * 100\% \tag{4}$$

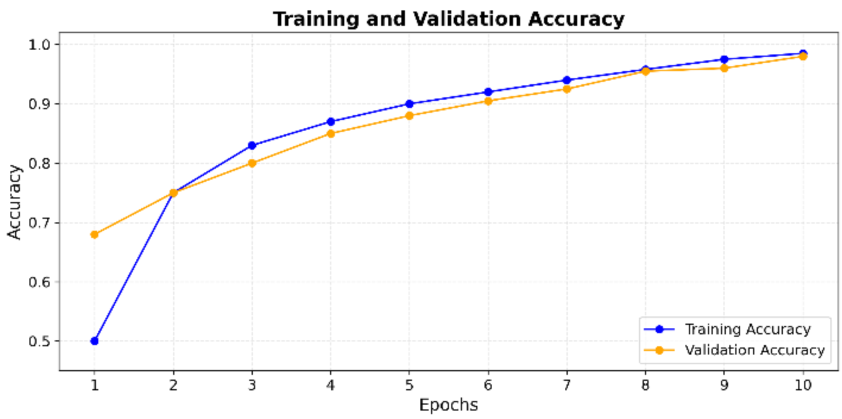
$$recall = \frac{TP}{TP+FN} * 100\% \tag{5}$$

$$F1score = \frac{2*precision*recall}{precision*recall} \tag{6}$$

Monitoring accuracy and loss trends enables the swift identification of potential issues, such as overfitting and underfitting, hence allowing for timely optimization measures. The significance of this technique is in its capacity to provide a comprehensive study of overall model performance, encompassing accuracy and loss comparisons for both training and validation datasets, as depicted in Figure 6.



(a)



(b)

Fig. 6 *Graph of (a) Accuracy; and (b) Loss Comparison*

3.7 Model Deployment

The conversion to TF Lite format enables the adoption of pre-trained models in mobile app development. Utilizing file formats helps enhance mobile application development. TF Lite is an improved and expedited variant of the TensorFlow framework utilized in machine learning modeling for the integration of models into mobile applications. Thus, mobile applications can identify medical conditions using photographs captured using advanced models. Algorithm 1 demonstrates the model integration code within the application.

Algorithm 1. Pseudocode Model Integration Code in Application

```
Require: TensorFlow Lite model
Ensure: Classification results
1: Import necessary libraries
2: Procedure Initialize
3:   Load TensorFlow Lite Model
4:   Set Number of Threads for
        Optimization
5: End Procedure
6: Procedure Predict(image)
7:  Resize image to 240 x 240
8:  Initialize inputBytes as
      Float32List with size (1 x
240 x
      240 x 3)
9:  For each pixel in image do
10:      Normalize RGB values of
the
           pixel (divide by 255)
11:      Append normalized values
to
           inputBytes
12:   End For
13:   Reshape inputBytes to input
tensor
        format (1 x 240 x 240 x 3)
14:   Initialize output as
Float32List
        with size (1 x Number of
Classes)
15:   Run model with inputBytes as
input
        and output as output
16:   Find index of maximum value in
        output list
17:   Return corresponding class
label
18: End Procedure
```

4. Result and Discussion

Testing was performed on apps developed utilizing the ensemble model technique, which integrates the EfficientNetB1 and MobileNetV2 architectures. This model employs hyperparameters refined by Bayesian and Random Optimization to achieve the optimal configuration, encompassing activation function (ReLU, ReLU6, or SiLU), learning rate (0.001, 0.0001, 0.00001), dropout rate (0.3, 0.4, 0.5), and kernel regularization (0.0001, 0.001, 0.01). The optimal accuracy for determining the best parameters was achieved by three trials done on the dataset, as shown in Table 2.

Table 2 *Hyperparameter trial parameter Bayesian Optimizer*

Trial	Activation	Learning Rate	Dropout	Kernel	Merge Layer	Accuracy
0	Relu6	0.001	0.5	0.01	Avg	0.9325
1	Relu	0.0001	0.3	0.001	Avg	0.9908
2	Silu	0.00001	0.4	0.0001	Avg	0.9929

Table 2 encapsulates the outcomes of three trials with diverse hyperparameters derived from the Bayesian Optimizer. In Trial 0, the model utilized the ReLU6 activation function, a learning rate of 0.001, a dropout rate of 0.5, and a kernel regularization of 0.01. The employed merge layer was average pooling (avg), with a final validation accuracy of 0.9325. This configuration serves as a benchmark for comparison with subsequent attempts. In Trial 1, the activation function was altered to ReLU, the learning rate was reduced to 0.0001, and the dropout rate was decreased to 0.3. The kernel regularization was established at 0.001, and the average pooling merge layer was retained. The enhancements led to a significant improvement in the model's performance, with a final validation accuracy of 0.9908.

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 17-29

In Trial 2, the activation function was altered to SiLU (Sigmoid Linear Unit), along with a diminished learning rate of 0.00001. The dropout rate rose to 0.4, although the kernel regularization was reduced to 0.0001. The average pooling merge layer was utilized again, attaining the maximum performance in all trials, resulting in a final validation accuracy of 0.9929. The statistics demonstrate that careful tweaking of hyperparameters, particularly learning rates and activation functions, can significantly enhance model accuracy, as evidenced by the optimal configuration in Trial 2.

Table 3 *Hyperparameter trial parameter Random Optimizer*

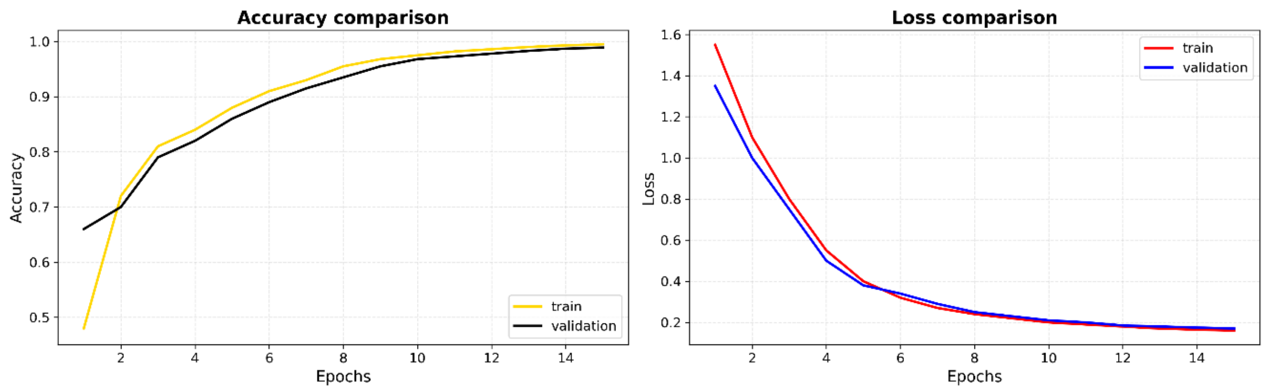
Trial	Activation	Learning Rate	Dropout	Kernel	Merge Layer	Accuracy
0	Relu	0.0001	0.4	0.01	Avg	0.9471
1	Silu	0.0001	0.4	0.0001	Concat	0.9904
2	Relu6	0.001	0.4	0.0001	Avg	0.9337

The outcomes of the trials for the Random Optimizer are displayed in Table 3: Trial 0 achieved a definitive validation accuracy of 94.71% with the following hyperparameters: {activation: 'relu', learning_rate: 0.0001, drop_out: 0.4, kernel_reg: 0.01, merge_layer: 'avg'}. Trial 1 attained the peak validation accuracy of 99.04% with the parameters {activation: 'silu', learning_rate: 0.0001, drop_out: 0.4, kernel_reg: 0.0001, merge_layer: 'concat'}. Ultimately, Trial 2 attained an accuracy of 93.37% using the settings {activation: 'relu6', learning_rate: 0.001, drop_out: 0.3, kernel_reg: 0.0001, merge_layer: 'concat'}.

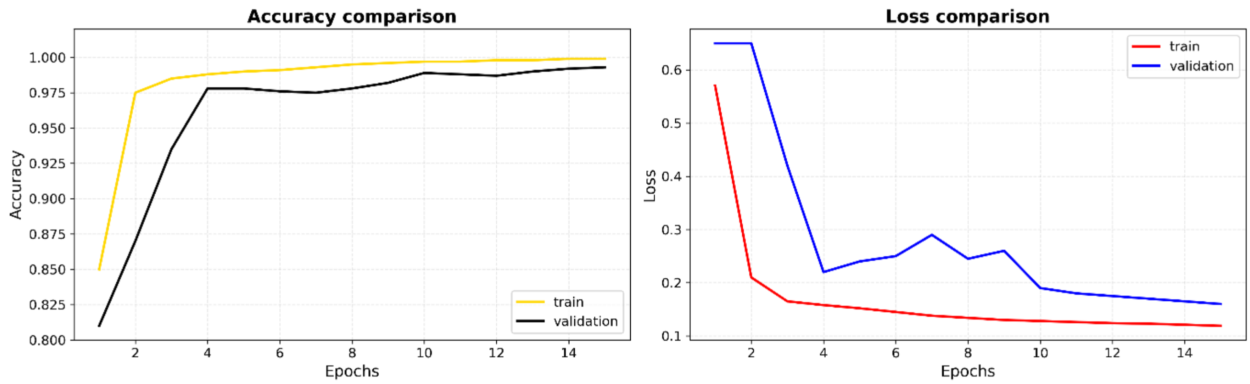
Among the three trials, Trial 1 demonstrated the highest efficacy, achieving a validation accuracy of 99.04%. This experiment utilized the SiLU activation function, which likely enhanced the model's adaptability to non-linear inputs, in conjunction with a learning rate of 0.0001 to ensure stable training. A dropout rate of 0.4 and a kernel regularizer of 0.0001 effectively reduced overfitting, while the concatenation merge layer preserved more robust feature representations.

The outputs from both designs are integrated using the merge layer technique, specifically by concatenation or averaging, contingent upon the hyperparameter value. The model architecture consists of convolutional layers derived from EfficientNetB1 and MobileNetV2, each generating output through GlobalAveragePooling2D. The outputs of these two models are combined using the chosen approach (concatenate or average) and subsequently processed through multiple Dense layers for classification. The Dense layer employs an activation function determined by the selected hyperparameters and incorporates L2 kernel regularization to mitigate overfitting. The model was created with the RMSprop optimizer with a dynamically modified learning rate, a sparse categorical cross-entropy loss function for multiclass classification, and accuracy as the assessment metric. The training procedure employed EarlyStopping and ReduceLROnPlateau callbacks to monitor validation loss and adaptively modify the learning rate. Upon completion of 16 epochs, the model attained a training accuracy of 99.99% and a validation accuracy of 99.90%, accompanied by a consistent validation loss of 0.1302.

The model's accuracy suggests that utilizing an ensemble method with EfficientNetB1 and MobileNetV2 can enhance precision in plant disease classification. The amalgamation of a lightweight and efficient CNN architecture with appropriate regularization methods enables the model to identify patterns in the data while effectively mitigating overfitting. Bayesian Optimization enhances the effectiveness of hyperparameter adjustment, yielding an accurate model for multiclass classification [28]. Nonetheless, the challenges encountered in training this model involve potentially distorted accuracy values that require more examination, in addition to the accuracy and loss graph depicted in Figure 7.



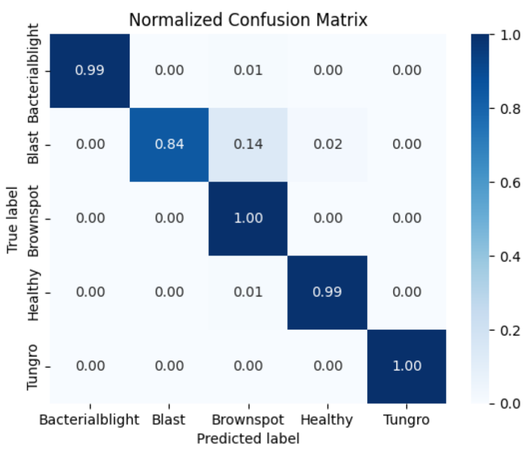
(a)



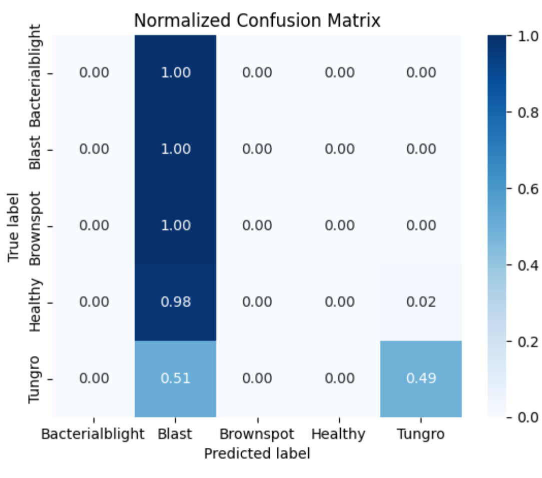
(b)

Fig. 7 *Accuracy and loss graphs based on the (a) Bayesian Optimizer; and (b) Random Optimizer*

The confusion matrix is encapsulated using Bayesian optimization and a random optimizer with three parameter trials, with the optimal outcome illustrated in Figure 8.



(a)



(b)

Fig. 8 *Confusion matrix with (a) Bayesian Optimizer; and (b) Random Optimizer*

Figure 8 depicts the classification efficacy of models refined using Bayesian and Random Search optimizers, evaluated by the classification report and confusion matrix. The model employing the Bayesian optimizer attained an overall accuracy of 96%, demonstrating notable precision and recall metrics across the classes. Bacterial blight, Blast, and Tungro demonstrated perfect precision and recall, signifying outstanding achievement in these categories. Brown spot and Healthy exhibited strong results, achieving F1-scores of 0.93 and 0.99, respectively. The overall average F1-score for the Bayesian optimizer model was 0.96, indicating consistent performance across all classes.

In contrast, the model optimized with the Random Search optimizer demonstrated inferior results. The overall accuracy was only 30%, facing significant challenges in predicting most classes. Bacterial blight, brown spot, and healthy categories exhibited 0% precision and recall, yielding an F1-score of 0.00 for these classifications. The model faced challenges in generating accurate predictions, particularly for the Bacterial blight and Brown spot classes, which demonstrated inadequate recall values. Despite this, Tungro achieved a rather high F1-score of 0.65, signifying a measure of effectiveness in a certain category. The overall average F1-score for the Random Search model was just 0.20, indicating a significant performance gap compared to the Bayesian-optimized model. These findings highlight the Bayesian optimizer's superior effectiveness in achieving higher classification accuracy and more balanced predictions among the classes.

5. Conclusion

An excellent model for plant disease identification employing CNN technology through an ensemble approach has been developed, achieving high accuracy. This system employs the EfficientNetB1 and MobileNetV2 designs, which are integrated using concatenation methods (concatenate or average) to enhance model performance. The collection originates from the open-source Mendeley platform, which offers plant leaf picture data categorized by several disease classifications. The model was trained with the RMSprop optimizer, which utilizes a variable learning rate and the sparse categorical cross-entropy loss function. The test outcomes indicated a training accuracy of 99.99% and a validation accuracy of 99.90%, accompanied by a stable validation loss. Callbacks such as EarlyStopping and ReduceLROnPlateau have effectively alleviated overfitting and improved the training process. This system aims to provide an effective and accurate solution for identifying plant diseases, while enhancing agricultural productivity through the use of advanced technologies.

Acknowledgement

This research received financial support from Diponegoro University, Semarang, Indonesia, under Grant Number 609-84/UN7D2/PP/VIII/2023.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of the paper.

Author Contribution

The authors are responsible for the study conception, research design, data collection, data analysis, result interpretation and manuscript drafting.

References

[1] W. Jearanaiwongkul, C. Anutariya, and F. Andres, "A Semantic-Based Framework for Rice Plant Disease Management: Identification, Early Warning, and Treatment Recommendation Using Multiple Observations," *New Generation Computing*, vol. 37, no. 4, 2019, doi: 10.1007/s00354-019-00072-0.

[2] W. S. Merritt *et al.*, "Reflecting on an Integrated Approach to Understanding Pathways for Socially Inclusive Agricultural Intensification," *Journal of Development Studies*, vol. 58, no. 8, 2022, doi: 10.1080/00220388.2022.2029418.

[3] J. Guo, S. Hu, and Y. Guan, "Regime shifts of the wet and dry seasons in the tropics under global warming," *Environmental Research Letters*, vol. 17, no. 10, 2022, doi: 10.1088/1748-9326/ac9328.

[4] M. R. Romadhon *et al.*, "Assessing the Effect of Rice Management System on Soil and Rice Quality Index in Girimarto, Wonogiri, Indonesia," *Journal of Ecological Engineering*, vol. 25, no. 2, 2024, doi: 10.12911/22998993/176772.

[5] D. L. Mapiemfu *et al.*, "Physical rice grain quality as affected by biophysical factors and pre-harvest practices," *International Journal of Plant Production*, vol. 11, no. 4, 2017, doi: 10.22069/ijpp.2017.3718.

[6] M. H. Montolalu, M. Ekananda, T. Dartanto, D. Widyawati, and M. Panennungi, "The Analysis of Trade

Liberalization and Nutrition Intake for Improving Food Security across Districts in Indonesia," *Sustainability (Switzerland)*, vol. 14, no. 6, 2022, doi: 10.3390/su14063291.

[7] T. Daniya and S. Vigneshwari, "Rice Plant Leaf Disease Detection and Classification Using Optimization Enabled Deep Learning," *Journal of Environmental Informatics*, vol. 42, no. 1, 2023, doi: 10.3808/jei.202300492.

[8] R. Li *et al.*, "Predicting rice diseases using advanced technologies at different scales: present status and future perspectives," 2023. doi: 10.1007/s42994-023-00126-4.

[9] M. Jeger *et al.*, "Global challenges facing plant pathology: multidisciplinary approaches to meet the food security and environmental challenges in the mid-twenty-first century," 2021. doi: 10.1186/s43170-021-00042-x.

[10] R. Abbasi, P. Martinez, and R. Ahmad, "The digitization of agricultural industry – a systematic literature review on agriculture 4.0," 2022. doi: 10.1016/j.atech.2022.100042.

[11] A. Setiawan, K. Adi, and C. E. Widodo, "Comparative Analysis of Deep Convolutional Neural Network for Accurate Identification of Foreign Objects in Rice Grains," *Engineering Letters*, vol. 32, no. 2, 2024.

[12] Z. Zhang, Y. Gu, and Q. Hong, "Rice Disease Identification System Using Lightweight MobileNetV2," 2021. doi: 10.12792/icisip2021.007.

[13] M. S. H. Shovon, S. J. Mozumder, O. K. Pal, M. F. Mridha, N. Asai, and J. Shin, "PlantDet: A Robust Multi-Model Ensemble Method Based on Deep Learning For Plant Disease Detection," *IEEE Access*, vol. 11, 2023, doi: 10.1109/ACCESS.2023.3264835.

[14] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using EfficientNet deep learning model," *Ecological Informatics*, vol. 61, 2021, doi: 10.1016/j.ecoinf.2020.101182.

[15] S. E. Whang, Y. Roh, H. Song, and J. G. Lee, "Data collection and quality challenges in deep learning: a data-centric AI perspective," *VLDB Journal*, vol. 32, no. 4, 2023, doi: 10.1007/s00778-022-00775-9.

[16] M. C. Wu, G. F. Lin, and H. Y. Lin, "The effect of data quality on model performance with application to daily evaporation estimation," *Stochastic Environmental Research and Risk Assessment*, vol. 27, no. 7, 2013, doi: 10.1007/s00477-013-0703-4.

[17] R. Sharma *et al.*, "Plant disease diagnosis and image classification using deep learning," *Computers, Materials and Continua*, vol. 71, no. 2, 2022, doi: 10.32604/cmc.2022.020017.

[18] K. Kumar K and E. Kannan, "Detection of rice plant disease using AdaBoostSVM classifier," *Agronomy Journal*, vol. 114, no. 4, 2022, doi: 10.1002/agj2.21070.

[19] R. E. Morrison, C. M. Bryant, G. Terejanu, S. Prudhomme, and K. Miki, "Data partition methodology for validation of predictive models," in *Computers and Mathematics with Applications*, 2013. doi: 10.1016/j.camwa.2013.09.006.

[20] A. Choubineh, J. Chen, D. A. Wood, F. Coenen, and F. Ma, "Deep ensemble learning for high-dimensional subsurface fluid flow modeling," *Engineering Applications of Artificial Intelligence*, vol. 126, 2023, doi: 10.1016/j.engappai.2023.106968.

[21] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," 2016. doi: 10.1109/JPROC.2015.2494218.

[22] C. A. C. F. da Silva, D. C. Rosa, P. B. C. Miranda, T. Si, R. Cerri, and M. P. Basgalupp, "Automated CNN optimization using multi-objective grammatical evolution," *Applied Soft Computing*, vol. 151, 2024, doi: 10.1016/j.asoc.2023.111124.

[23] G. C. Cawley and N. L. C. Talbot, "Preventing over-fitting during model selection via bayesian regularisation of the hyper-parameters," *Journal of Machine Learning Research*, vol. 8, 2007.

[24] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, 2012.

[25] C. Zhu, J. Xu, C. H. Chen, L. H. Lee, and J. Q. Hu, "Balancing Search and Estimation in Random Search Based Stochastic Simulation Optimization," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, 2016, doi: 10.1109/TAC.2016.2522094.

[26] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, “Fundamental concepts of convolutional neural network,” in *Intelligent Systems Reference Library*, vol. 172, 2019. doi: 10.1007/978-3-030-32644-9_36.

[27] K. Riehl, M. Neunteufel, and M. Hemberg, “Hierarchical confusion matrix for classification performance evaluation,” *Journal of the Royal Statistical Society. Series C: Applied Statistics*, vol. 72, no. 5, 2023, doi: 10.1093/jrsssc/qlad057.

[28] R. J. Borgli, H. Kvale Stensland, M. A. Riegler, and P. Halvorsen, “Automatic Hyperparameter Optimization for Transfer Learning on Medical Image Datasets Using Bayesian Optimization,” in *International Symposium on Medical Information and Communication Technology, ISMICT*, 2019. doi: 10.1109/ISMICT.2019.8743779.