https://doi.org/10.17221/59/2024-RAE

A novel ensemble convolutional neural networks for rice disease identification

*Richard Alvin Pratama, Nabila Husna Shabrina**

Department of Computer Engineering, Universitas Multimedia Nusantara, Tangerang, Banten, Indonesia

**Corresponding author: nabila.husna@umn.ac.id*

Citation: Pratama R.A., Shabrina N.H. (2026): A novel ensemble convolutional neural networks for rice disease identification. Res. Agr. Eng., 72: 1–13.

Abstract: Rice is a crucial food commodity worldwide, particularly in Asian countries. However, various factors, such as drought, floods, and pest attacks, can lead to the emergence of diseases in rice plants. Accurately identifying these diseases poses a significant challenge for farmers, often leading to significant yield losses. Conventionally, farmers rely on manual methods based on their experience and visual inspections to identify rice diseases. However, this approach is highly ineffective, time-consuming, and prone to error. This study aimed to address this issue by proposing advanced deep learning techniques, an ensemble learning method, to automate and enhance the identification of rice plant diseases. The ensemble learning method was proposed by leveraging two state-of-the-art pre-trained models: EfficientNetV2B0 and MobileNetV3-Large. The proposed Average Ensemble method demonstrates superior performance compared with single models. The proposed Average Ensemble achieved superior performance with an average precision of 0.9339, a recall of 0.9330, an F1-score of 0.9328, and a test accuracy of 0.9330. The results of this study can be used to aid farmers and researchers in accurately identifying rice diseases, ultimately supporting better disease management practices, and enhancing the agricultural productivity.

Keywords: ensemble deep learning; precision agriculture; rice plant disease identification

Rice, scientifically referred to as *Oryza sativa*, is a staple food worldwide, especially in Asian countries. More than 90 percent of the world's rice is produced and consumed in Asia (Bandumula 2018), with six countries – China, India, Indonesia, Bangladesh, Vietnam, and Japan – leading the way. However, farmers face various obstacles, including plant diseases, which affect crop production and earnings. Moreover, recognising rice diseases is a crucial aspect of farming that can help farmers manage their fields. Farmers typically rely on their experience, knowledge, and visual abilities to diagnose these diseases. This process is inefficient, time-consuming, and prone to error. Furthermore, the lack of knowledge among farmers regarding how to identify diseases in rice plants exacerbates this problem (Manavalan 2020).

Advancements in technologies, particularly in artificial intelligence, have led to innovative and precise methods for identifying diseases in rice fields. One such method is deep learning, which uses artificial neural networks to analyse and extract features from data and can be used to diagnose diseases accurately. These techniques are increasingly considered viable alternatives for the continuous monitoring of plant diseases across diverse crops (Ramesh et al. 2018; Harakannanavar et al. 2022; Li et al. 2023; Shabrina and Brian 2023; Wang and Shabrina 2023; Wenjing et al. 2023; Shabrina et al. 2024). Several researchers have employed deep

https://doi.org/10.17221/59/2024-RAE

learning to diagnose rice plant diseases. Pascual et al. (2019) proposed an automated disease detection method using image processing, where green pixels were masked with blue or black pixels and fed into support vector machine (SVM) and random forest classifiers. The blue pixel and SVM combination achieved 82.41% accuracy, outperforming the other models. Haridasan et al. (2023) successfully utilised a combination of SVM classifiers and convolutional neural networks (CNNs) to accurately identify and classify various types of rice plant diseases. Their efforts yielded a validation accuracy of 0.9145. Lamba et al. (2023) proposed a hybrid approach that combined the capabilities of a CNN and an SVM to predict the rice disease type and intensity with a 98.43% accuracy rate and 41.25% loss rate. Khan et al. (2024) applied transfer learning to recognise ten various diseases using pre-trained architectures, including VGGNET, Inception V3, ResNet50, and InceptionResNetV2. Among these frameworks, ResNet50 attained the highest validation accuracy of 87.51%, with a precision of 90.33% and a recall of 99.80%.

Prior research has some limitations, such as a narrow focus on rice diseases and limited model performance. The goal of this study is to enhance the previous research by creating an ensemble-based deep learning method for identifying diseases in rice fields. Ensemble methods that combine several individual models have been demonstrated to outperform single architectures in terms of performance and are relatively straightforward to implement (Chompookham and Surinta 2021; Mohammed and Kora 2023). This study proposes an ensemble method that utilises two pre-trained CNN-based models, EfficientNetV2B0 (Tan and Le 2019, 2021) and MobileNetV3-Large (Howard et al. 2019), by employing two distinct ensemble approaches: average ensemble and concatenation ensemble. The key contribution of this study is the design of a novel architecture that demonstrates superior performance in identifying rice plant diseases.

MATERIAL AND METHODS

Research workflow

The workflow for identifying rice diseases using the proposed ensemble CNN method is illustrated in Figure 1. The process was divided into three primary stages: pre-processing, training, and model deployment and evaluation. The pre-processing stage involved applying data-balancing techniques and resizing the images. Four models were used in the training process: pretrained EfficientNet-V2B0, pretrained MobileNetV3Large, and the proposed average and concatenated ensembles. The performance of each model was evaluated to determine the best-performing model, which was then deployed in a progressive web application (PWA) system.

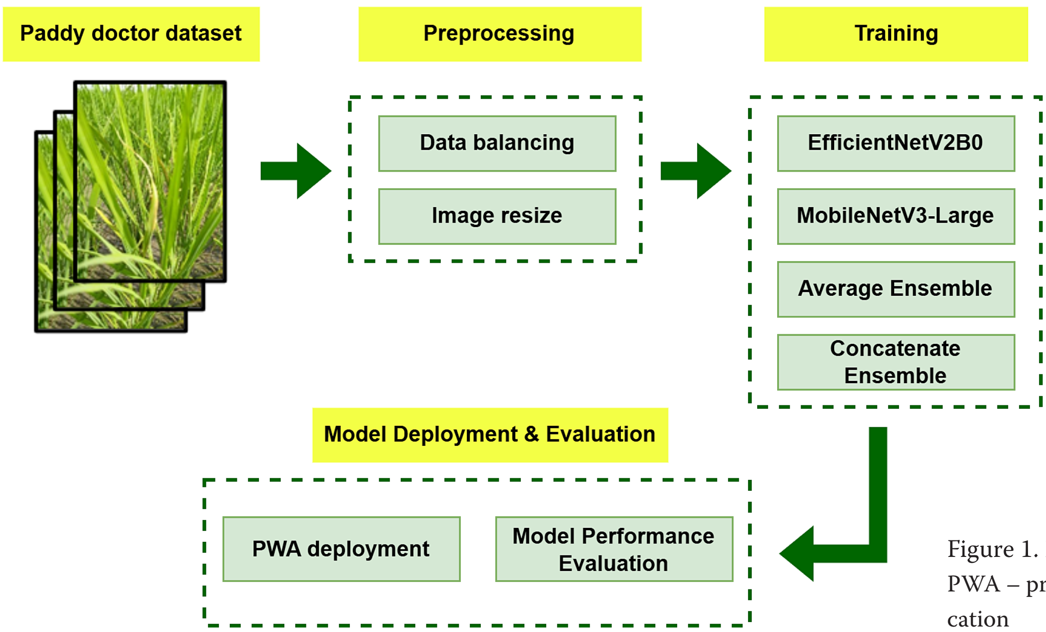

Figure 1. Research workflow
PWA – progressive web application

https://doi.org/10.17221/59/2024-RAE

Table 1. Class distribution in the Rice Doctor dataset

Class	Number of images
Tungro	1 088
Bacterial Leaf Streak	380
Dead Heart	1 442
Bacterial Leaf Blight	479
Hispa	1 594
Bacterial Panicle Blight	337
Normal	1 764
Brown Spot	965
Downy Mildew	620
Blast	1 738
Total	10 407

Datasets

This study used a subset of the Paddy Doctor Dataset (Petchiammal et al. 2022), the largest expert-annotated visual image dataset for rice disease classification. A subset of the original Rice Doctor was used due to the limited access to the original full version. The dataset was obtained from a rice field near Tirunelveli District, India, from February to April 2021. The subset of the Rice Doctor dataset contained 10 407 images which categorised into ten classes, namely Bacterial Leaf Blight, Bacterial Leaf Streak, Bacterial Panicle Blight, Blast, Brown Spot, Dead Heart, Downy Mildew, Hispa, and Tungro and Normal. The distribution for each class on the dataset is shown in Table 1 and the sample images of the dataset are presented in Figure 2. The dataset was divided into two sets: training and testing. The training set comprised 90% of the data, whereas the testing set comprised 10%. Furthermore, 10% of the training set was allocated for validation.

Data pre-processing

To address the issue of class imbalance in the dataset, two balancing techniques, undersampling and oversampling, were employed and compared to determine which performed better. Undersampling was applied to classes with more than 337 images by reducing the total number of images in these classes to 337. This involved trimming the excess images, ensuring that each class contained only 337 images. This resulted in an undersampled dataset containing 3 370 images.

For the oversampling, the dataset was first split by separating 100 images per class for testing purposes, resulting in 1 000 testing images. Oversampling was then performed on the training data in a two-step process. First, for classes with more than 1 000 images, the number of images was randomly reduced to 1 000. Subsequently, for classes with fewer than 1 000 images, data augmentation techniques were applied to increase the number of images to 1 000. This produced a training set containing 10 000 images. Therefore, the final oversampled dataset consisted of 11 000 images

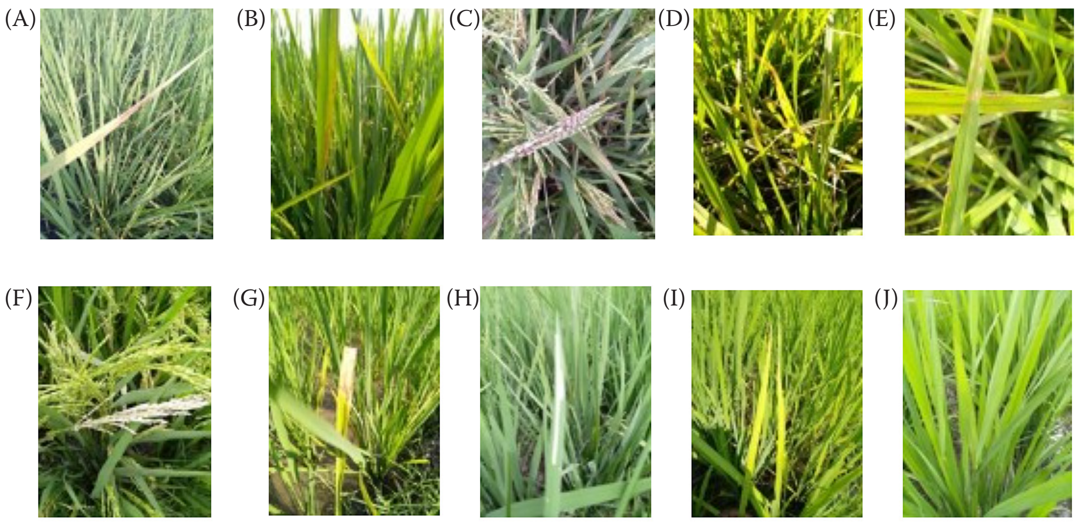

Figure 2. Sample of the Rice Doctor dataset: Bacterial Leaf Blight (A); Bacterial Leaf Streak (B); Bacterial Panicle Blight (C); Blast (D); Brown spot (E); Dead Hearth (F); Downey Mildew (G); Hispa (H); Tungro (I); Normal (J)

https://doi.org/10.17221/59/2024-RAE

in total, including 10 000 training images and 1 000 testing images.

To implement the oversampling dataset, data augmentation techniques were used to increase the number of images (Shorten and Khoshgoftaar 2019). Data augmentation included changing the brightness, flipping the image (both vertically and horizontally), blurring the image, and adjusting the zoom range. The details of the augmentation process applied to the datasets using the oversampling technique are as follows:
- Image flip: applied using random vertical and horizontal flips.
- Image brightness: applied with a random brightness, with values between 0.3 to 1.
- Image blurring: applied with a random kernel size varying from 1 to 3.
- Zoom: applied with parameters varying from 0.5 to 1.

In addition to balancing techniques, the data processing method also includes resizing the input images. The original image size was 480 × 640 pixels. The images were resized to 256 × 256 pixels to match the input requirements of the model. This size was chosen because it balances the computational efficiency with the optimal input size for the architecture.

Model architecture

EfficientNetV2B0 architecture. EfficientNetV2B0 is the newest version of the EfficientNet model, offering superior performance compared with its predecessor. EfficientNetV2 integrates both the MBConv and Fused-MBConv layers, with Fused-MBConv primarily utilised at the beginning of the network (Tan and Le 2019, 2021). Fused-MBConv enhances the performance of mobile and server accelerators. Moreover, EfficientNetV2 employs a training-aware neural architecture search (NAS) framework, which is more extensive than the original NAS framework and omits the final stride used in the original EfficientNet. In this study, EfficientNetV2 was selected due to its superior performance compared with the first version. Additionally, the choice of the B0 variant was influenced by its lightweight nature compared to the B1, B2, and other versions while still maintaining a comparable level of accuracy. Figure 3 illustrates the detailed architecture of EfficientNetV2B0.

MobileNetV3-Large architecture. MobileNetV3, the latest version of the MobileNet model, incorporates two advanced network search techniques: platform-aware NAS for block-wise searches and NetAdapt for layer-wise searches (Howard et al. 2019). The 1 × 1 convolution in MobileNetV2, which is used as the final layer, was repositioned to the layer following the final average pooling. Moreover, the bottleneck layer behind the 1 × 1 convolution is removed. The use of 32 filters in 3 × 3 convolution has been found to be non-essential and can be replaced with hard swish activation to improve the efficiency. In this study, MobileNetV3-Large was selected due to its superior capabilities. Figure 4 illustrates the detailed architecture of the MobileNetV3-Large.

Average ensemble. The average ensemble method combines the predicted results of multiple pretrained models by taking their average to produce a final ensemble prediction. This method

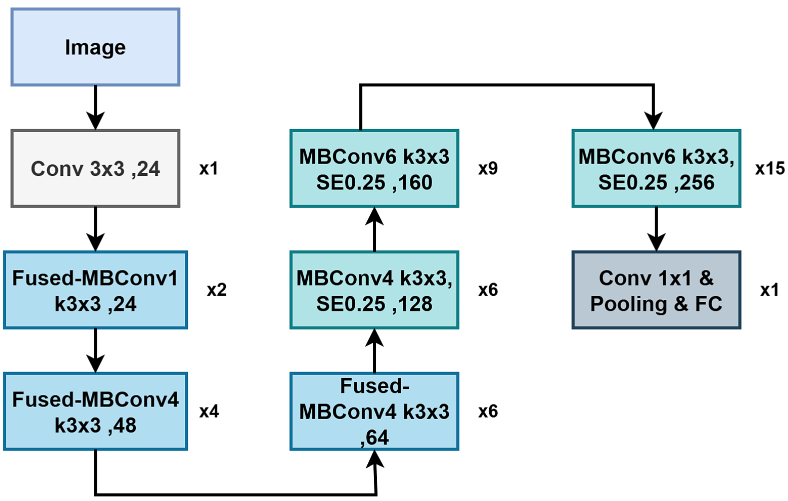

Figure 3. EfficientNetV2B0 architecture

https://doi.org/10.17221/59/2024-RAE

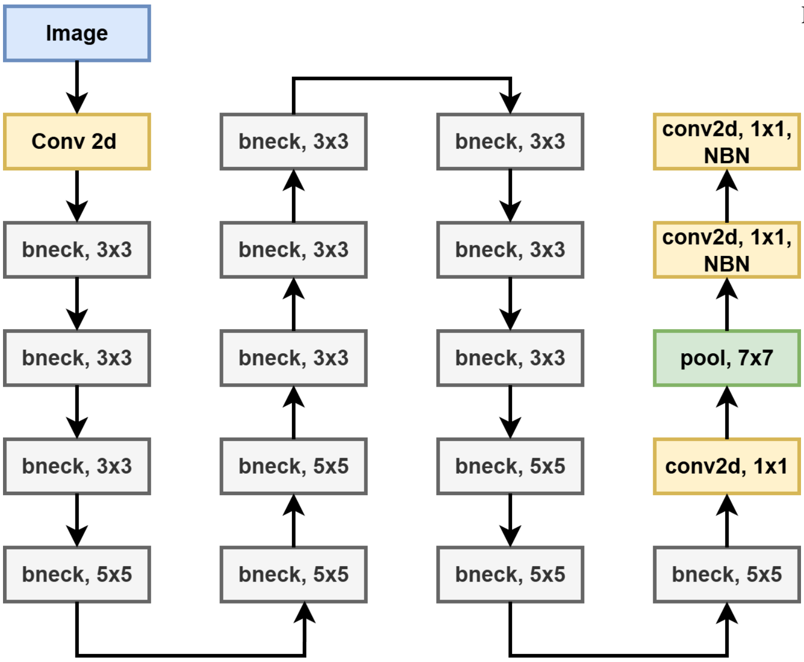

Figure 4. MobileNetV3-Large architecture

is straightforward and does not require complicated parameter configurations, as it relies on the average function of the model. One of the key advantages of the average ensemble method is its compatibility with various models, allowing for the combination of different architectures, such as EfficientNetV2B0 and MobileNetV3-Large, to enhance the predictive performance. Figure 5 illustrates the proposed average ensemble CNN.

Concatenate ensemble. The concatenation ensemble method combines the features from multiple models into a single comprehensive feature set, which is then used in the subsequent classification layer. The concatenation ensemble is highly compatible with various models, making it a versatile approach for enhancing the predictive performance. Figure 6 illustrates the proposed concatenation ensemble CNN.

Model implementation. During the training, several hyperparameter adjustments were performed. The first layer adjustment involved the addition of a flattened layer to convert the feature extraction results into a one-dimensional vector. Following this, a dense layer with 1 024 units and Rectified Linear Unit (ReLU) activation were incorporated to execute the fully connected layer operations, which are essential for the classification process. To prevent overfitting, a dropout layer with a rate of 0.2 was introduced, reducing the connections between the neurons. Finally,

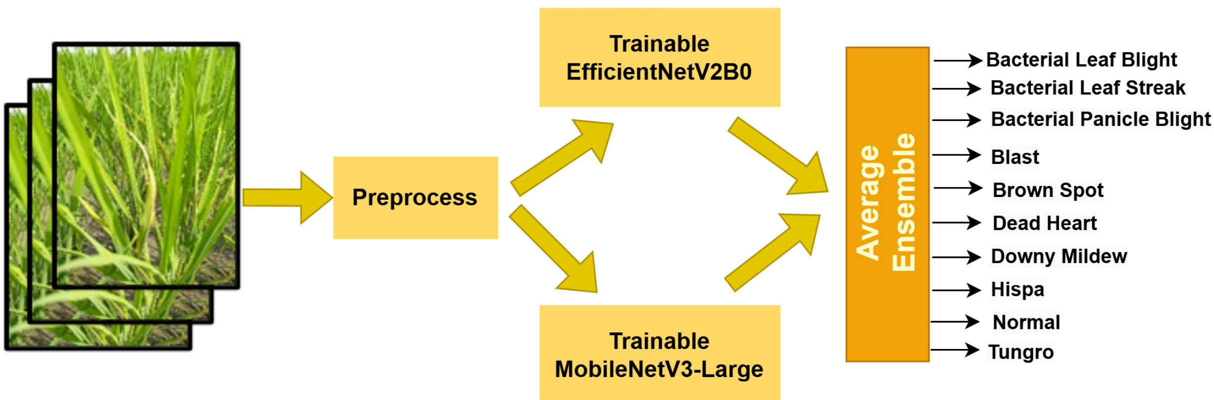

Figure 5. Average ensemble

https://doi.org/10.17221/59/2024-RAE

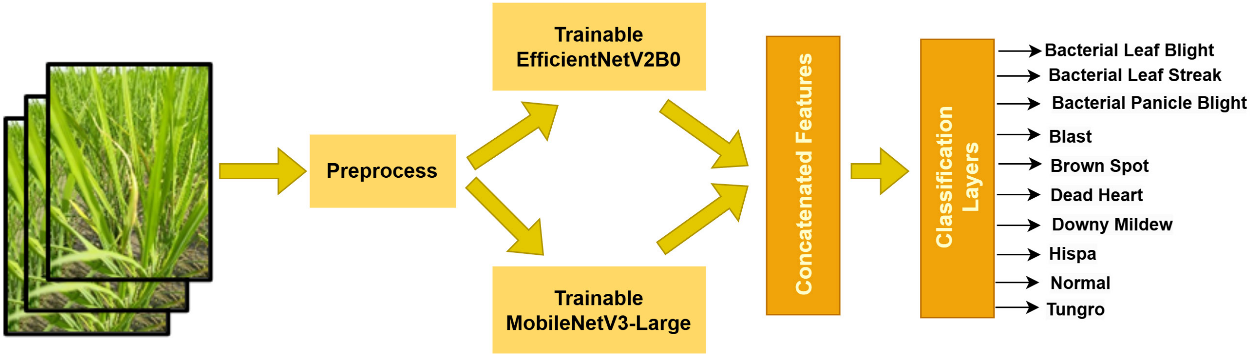

Figure 6. Concatenate ensemble

a dense layer with 10 units and softmax activation was included to produce the classification results across 10 classes. These adjustments collectively optimise the model's performance and generalisation capabilities during training. An Adaptive Moment Estimation (adam) optimiser with a learning rate of 0.0001 was used to find the local minimum of the loss function and adjust the weights during training. A categorical cross-entropy loss function was employed to measure the difference between the predicted and expected outputs. The model was trained with a batch size of 32 and over ten epochs. To prevent overfitting, EarlyStopping was used as a callback during the training.

The model is trained using two different system specifications. The first system was a computer equipped with an Intel® Core™ i7-8550U CPU @ 1.80 GHz 1.99 GHz, NVIDIA GeForce MX130 GPU, and 16 GB of memory. The second system was Google Colab Pro, which featured an Intel Xeon Processor @ 2.3 GHz, an A100 Nvidia GPU with 40 GB of RAM, and 83.5 GB of memory.

Metric evaluation

The evaluation of model metrics plays a crucial role in analysing the performance of the utilised models. It serves multiple purposes, such as measuring the performance, comparing different models, optimising models, and aiding in drawing conclusions from model tests. The evaluation metrics used were the accuracy, precision, recall, F1-score, and support. The accuracy assesses how well the trained model classifies rice plant diseases, essentially measuring the proportion of correct predictions among the total samples. The precision measures the accuracy of the model's predictions against the requested data. The recall evaluates a model's success in retrieving relevant information. The F1-score provides a balanced view by considering both precision and recall. Equations (1)–(4) depict the formulas for the accuracy, precision, recall, and F1-score, respectively.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{F1} - \text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

where: TP – true positive (correctly identified positive cases); TN – true negative (correctly identified negative cases); FP – false positive (negative cases that were incorrectly identified as positive); FN – false negative (positive cases that were incorrectly identified as negative).

RESULTS AND DISCUSSION

Model performance results. In this section, we assess the proposed ensemble methods using key metrics, such as the test accuracy, precision, recall, and F1-score, to evaluate their performance in identifying rice diseases. As shown in Table 2, the proposed Average Ensemble method, which combines the outputs of EfficientNetV2B0 and MobileNetV3-Large, demonstrates superior performance compared to individual models under both balancing techniques. The Average Ensemble achieved the highest performance, with an average

https://doi.org/10.17221/59/2024-RAE

Table 2. Model performance results

Model	Data balancing techniques	Average precision	Average recall	Average F1-score	Test accuracy
EfficientNetV2B0	under-sampling	0.8286	0.8147	0.8137	0.8147
	oversampling	0.9105	0.9050	0.9059	0.9050
MobileNetV3-Large	under-sampling	0.8407	0.8353	0.8359	0.8353
	oversampling	0.8900	0.8760	0.8776	0.8760
Average ensemble	under-sampling	0.8189	0.8000	0.8028	0.8000
	oversampling	0.9339	0.9330	0.9328	0.9330
Concatenation ensemble	under-sampling	0.8606	0.8559	0.8537	0.8559
	oversampling	0.9108	0.9040	0.9046	0.9040

precision of 0.9339, a recall of 0.9330, an F1-score of 0.9328, and a test accuracy of 0.9330. Similarly, the Concatenation Ensemble method also showed notable performance improvements, achieving an average precision of 0.9108, a recall of 0.9040, an F1-score of 0.9046, and a test accuracy of 0.9040. These results indicate that the ensemble methods consistently outperformed the individual models across both balancing scenarios. Furthermore, the results confirm that all the models exhibited performance variations depending on the balancing technique, but the ensemble methods consistently provided more robust and higher performance than single models.

The impact of data balancing techniques on the model performance is evident across all the models. EfficientNetV2B0 showed significant performance improvement with the oversampling compared to the undersampling. MobileNetV3-Large also benefits from the oversampling, with improved metrics compared to the undersampling. Both the Average and Concatenation Ensemble methods exhibit notable performance differences depending on the data-balancing technique used. These results demonstrate that the oversampling significantly enhances the models' ability to generalise from the data, leading to improved performance across all the measured metrics. These findings highlight the critical role of data balancing techniques in optimising the model performance. The use of oversampling proved to be highly effective in enhancing the predictive capabilities of both the individual models and ensemble methods. These results highlight the potential of the proposed ensemble methods to achieve superior accuracy in identifying rice diseases when appropriate preprocessing techniques are applied.

Accuracy and loss curve. Figures 7 and 8 illustrate the loss and accuracy curves, respectively, for the training set. The loss curve in Figure 7 shows a gradual decrease approaching nearly zero, indicating that the model effectively minimised the error during training. In Figure 8, the accuracy curve demonstrates a consistent increase, reaching nearly 100%, which signifies that the model successfully learns features from the data. Moreover, the training accuracy graphs revealed the differences between the undersampling and oversampling techniques. The oversampling technique dataset exhibits a slightly higher and more stable training accuracy curve compared to the undersampling technique dataset. This suggests that the oversampling allows the model to achieve better learning during training, as reflected in both the faster convergence and higher final accuracy values. This observation aligns with the previous model performance results, where utilising more data through oversampling enhanced the overall model performance.

Figures 9 and 10 show the loss and accuracy curves for the validation set, respectively. In Figure 9, it is evident that oversampling improves the validation loss, with each model achieving a validation loss below 0.5, whereas the models trained with undersampling show validation losses above 0.5. Unlike the training set, the loss curve for the validation set did not converge as effectively. This discrepancy is likely due to the challenging patterns present in the images, making it difficult for the models to generalise and causing them to rely more heavily on the training data. Among the models trained with the oversampling technique, EfficientNetV2B0 and the ensemble methods exhibited lower and more stable validation loss curves. While Mo-

https://doi.org/10.17221/59/2024-RAE

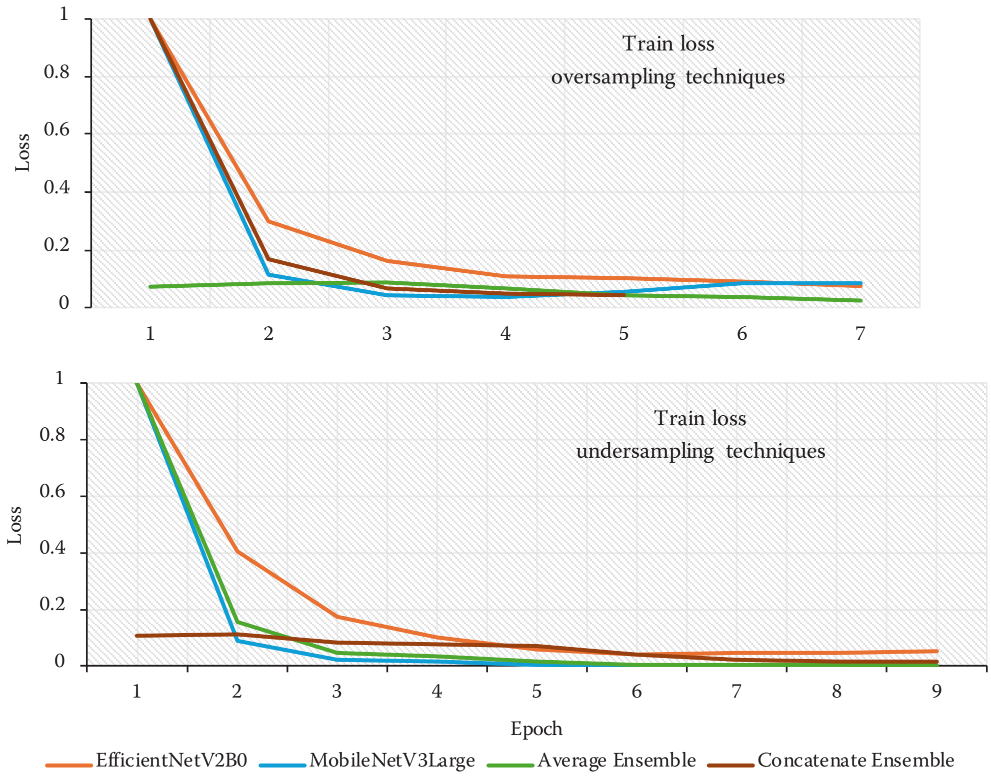

Figure 7. Loss curve for the training set

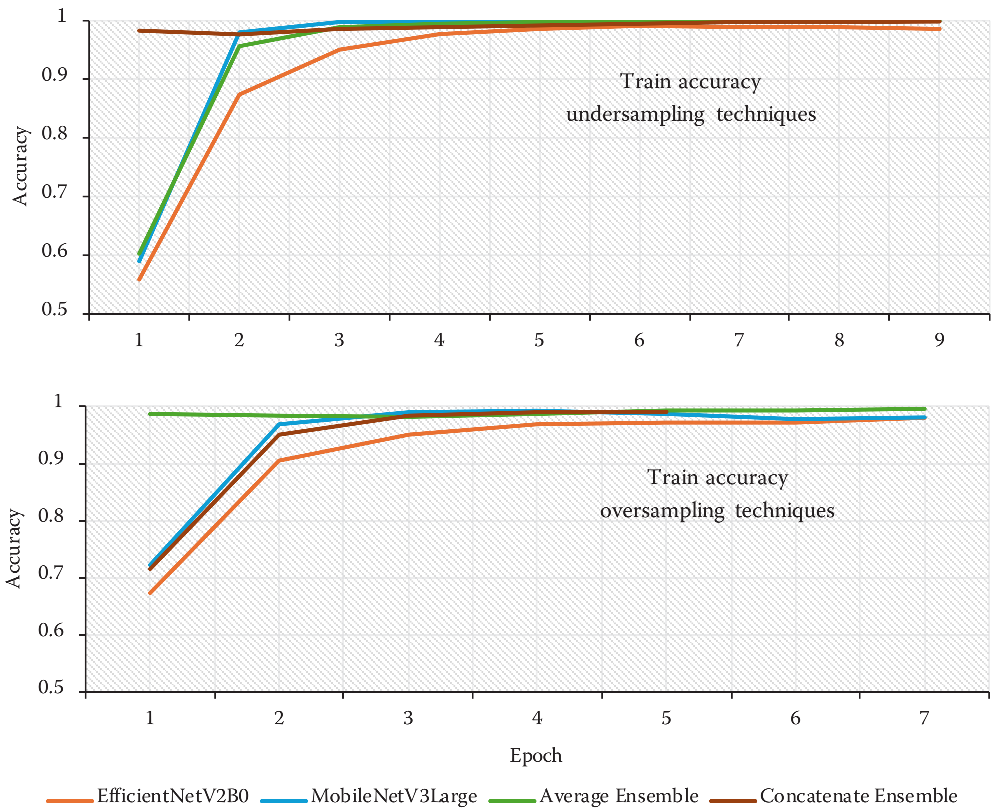

Figure 8. Accuracy curve for the training set

https://doi.org/10.17221/59/2024-RAE

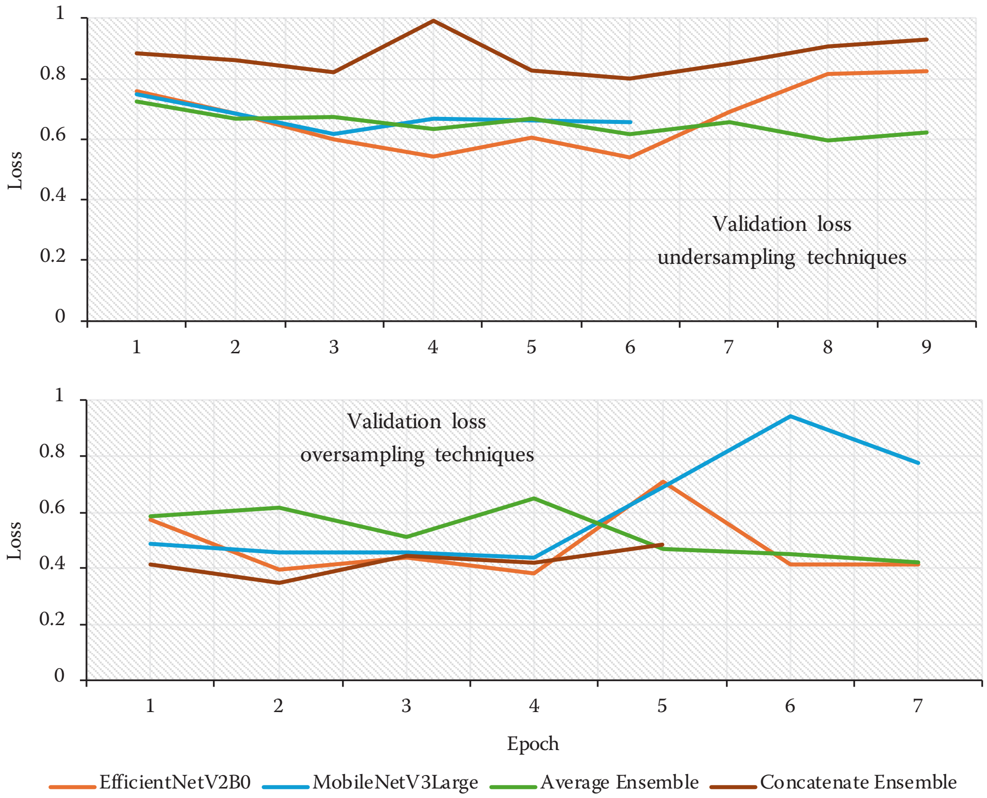

Figure 9. Loss curve for the validation set

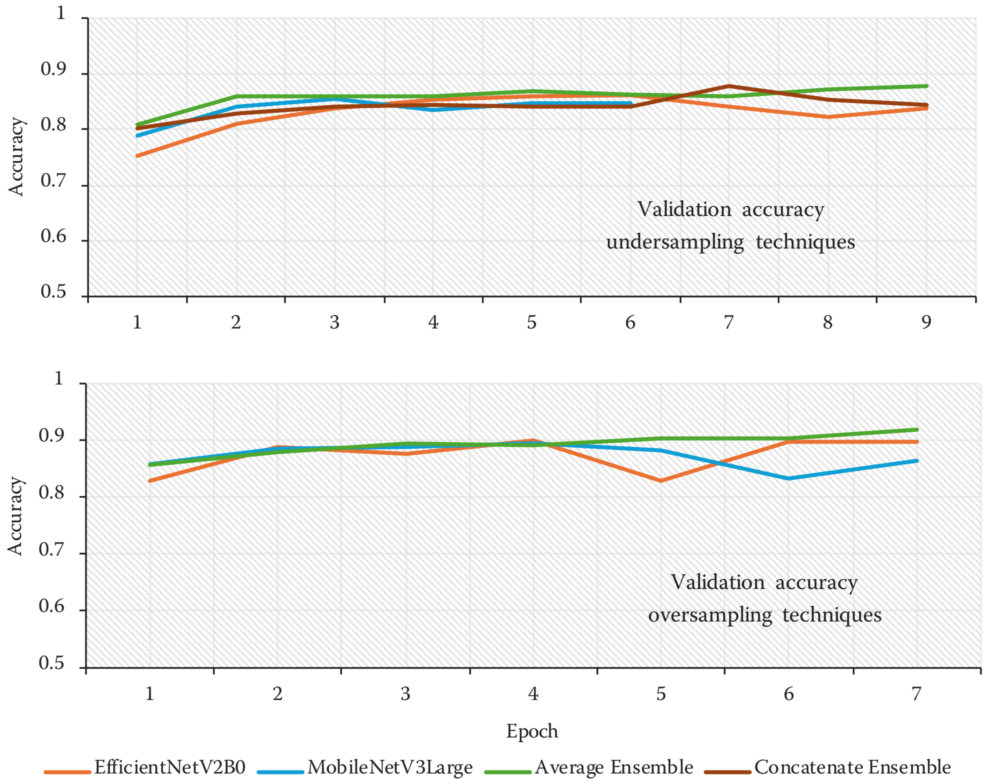

Figure 10. Accuracy curve for the validation set

https://doi.org/10.17221/59/2024-RAE

bileNetV3-Large did not achieve a validation loss below 0.5 in the final epoch, it still demonstrated improvement compared to its performance under undersampling. Additionally, as shown in Figure 9, some models stop training earlier due to the early stopping mechanism, which terminates the training when the validation performance no longer improves after a set number of epochs, helping to prevent overfitting.

Regarding the validation accuracy results, Figure 10 shows that the validation accuracy of the concatenation ensemble consistently increased, approaching 88%. Conversely, the average ensemble achieved a validation accuracy close to 88% at the sixth epoch, but then experienced a decline. This disparity arises from the differing methodologies of the two ensembles; the concatenation ensemble does not utilise the training results of the two pre-trained models, whereas the average ensemble is retrained based on these results. This difference in the approach makes the average ensemble less effective for datasets when using the undersampling technique.

Deployment to PWA system. The design of the PWA system uses the NuxtJs 2 framework to create user-friendly interfaces and functions for uploading and viewing images of the rice plants. NuxtJs 2 was selected for its superior performance and its Nuxt-PWA module, which facilitates PWA development. The setup process involved installing NuxtJs 2 and the Nuxt-PWA module, followed by their configuration in the Nuxt configuration file. The display will be designed for ease of access and user-friendliness. The next step involves connecting NuxtJs 2 to the FastAPI backend. The system was tested and debugged using a network waterfall to ensure seamless integration. As illustrated in Figure 11, the PWA display features a home view with various utilities, including camera access and image upload capabilities, for predicting the rice plant conditions. Additionally, it includes an information page where users can read about different rice diseases. This design ensures that users have easy access to essential features and information directly from their home view, enhancing their overall experience. The PWA system and the trained model are publicly available at https://github.com/RichardAlvin/PWA_Paddyist.

Model inference time. The model speed prediction results were retrieved through the network waterfall page on the PWA system, which was tested locally. This local setup minimised the network overhead, ensuring that the recorded response time primarily reflected the model's inference time with minimal system-level delay. In contrast, if the application programming interface (API) endpoint were deployed remotely, such as on a cloud server, internet access would be required, and the total response time could be significantly affected by the network speed and stability. By conducting all the tests locally, external network factors were eliminated, allowing for a more accurate and consistent measurement of the inference performance. The test was performed five times, and the results are presented in Table 3. In this evaluation, undersampling and oversampling were tested as separate pre-processing approaches. Each model was evaluated under both techniques independently to compare their impact on inference time. The concatenate ensemble was not tested owing to limited computational resources.

The inference times for EfficientNetV2B0, MobileNetV3Large, and their ensemble methods across the five tests provided insight into the efficiency of each approach. EfficientNetV2B0 showed a relatively broad range of inference times, from 341 ms to 495 ms, averaging 381 ms. This indicates that although EfficientNetV2B0 can perform ef-

Table 3. Model inference times (in ms)

Test	EfficientNetV2B0		MobileNetV3-Large		Average ensemble	
	under-sampling	over-sampling	under-sampling	over-sampling	under-sampling	over-sampling
1	336	350	304	256	304	383
2	329	369	307	229	307	382
3	347	341	301	233	301	398
4	482	495	310	220	310	423
5	348	350	300	223	300	402
Average	368	381	304	232	304	398

https://doi.org/10.17221/59/2024-RAE

ficiently, its performance is less predictable and can be significantly slower under certain conditions. MobileNetV3Large, on the other hand, consistently outperformed EfficientNetV2B0, with inference times ranging from 220 ms to 256 ms, resulting in a much lower average of 232 ms. This demonstrates that MobileNetV3Large is not only faster on average, but also more consistent in its performance.

The Average Ensemble method, which combines the outputs of both models, displays a range of inference times from 382 to 423 ms, averaging at 398 ms. This approach results in slower inference times compared with both the individual models, indicating that the overhead of combining the model outputs might outweigh the potential benefits of ensemble learning in this scenario. Owing to the computational limitations, the results

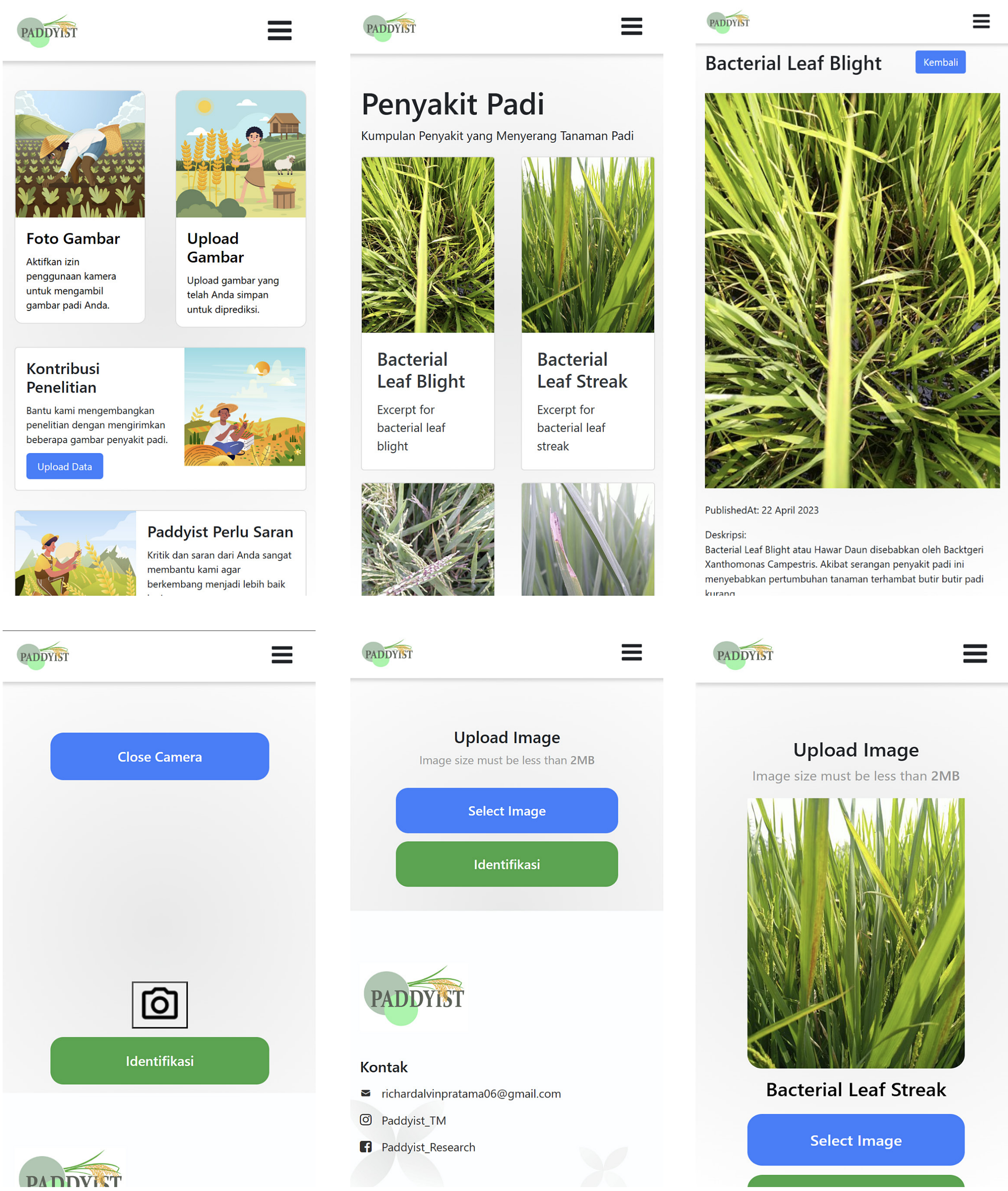

Figure 11. The progressive web application (PWA) system

https://doi.org/10.17221/59/2024-RAE

for the concatenated ensemble method are not available. Although the ensemble model is approximately 4.5% slower than the oversampled EfficientNetV2B0, the trade-off is justified by its improved accuracy, precision, and robustness, especially in non-real time use cases such as mobile agricultural advisory tools. In conclusion, while ensemble methods aim to leverage the strengths of multiple models, they do not provide a clear advantage in terms of the inference time efficiency.

Comparison to previous work. In comparison to previous studies, Petchiammal et al. (2022) used a more detailed version of the Paddy Doctor dataset, comprising 16 225 images across 12 classes. Their work employed a self-developed CNN architecture, VGG16, and MobileNet architectures, achieving test accuracies of 88.84%, 92.42%, and 93.19%, respectively. Using the same dataset with a smaller number of total images (due to the availability on the open repository), our proposed approach achieved a test accuracy of 93.30%, with an average precision of 0.9339, a recall of 0.9330, and an F1-score of 0.9328.

Despite working with a reduced dataset size, our model achieved slightly higher test accuracy compared to MobileNet and VGG16 reported in the previous work. Moreover, the high and balanced values of the precision, recall, and F1-score across all the classes indicate that our approach not only maintains high accuracy, but also ensures robust and reliable classification performance. This demonstrates that our ensemble-based method effectively addresses class imbalance and data limitations. To the best of our knowledge, no other published study besides the work of Petchiammal et al. (2022) has utilised the Paddy Doctor dataset for rice disease classification, and therefore, the comparison is limited to their research.

CONCLUSION

This study proposed an ensemble CNN approach for identifying rice diseases by utilising both average and concatenation ensemble techniques. The ensemble method combines two pre-trained models: EfficientNetV2B0 and MobileNetV3-Large. This study utilised a subset of the Paddy Doctor dataset, containing 10 407 images across ten disease classes. Data pre-processing involves undersampling and oversampling techniques to address the class imbalance. The proposed approach achieved an average precision of 0.9339, a recall of 0.9330, an F1-score of 0.9328, and a test accuracy of 0.9330, thereby demonstrating its effectiveness in accurately classifying rice plant diseases. The results of this study can be used to aid farmers and researchers in accurately identifying rice diseases, ultimately supporting better disease management practices, and enhancing the agricultural productivity.

Acknowledgement

The authors would like to thank Universitas Multimedia Nusantara for the support provided during this study.

REFERENCES

Bandumula N. (2018): Rice production in Asia: Key to global food security. Proceedings of the National Academy of Sciences, India Section B: Biological Sciences, 88: 1323–1328.

Chompookham T., Surinta O. (2021): Ensemble methods with deep convolutional neural networks for plant leaf recognition. ICIC Express Letters, 15: 553–665.

Harakannanavar S.S., Rudagi J.M., Puranikmath V.I., Siddiqua A., Pramodhini R. (2022): Plant leaf disease detection using computer vision and machine learning algorithms. Global Transitions Proceedings, 3: 305–310.

Haridasan A., Thomas J., Raj E.D. (2023): Deep learning system for paddy plant disease detection and classification. Environmental Monitoring and Assessment, 195: 120.

Howard A., Sandler M., Chen B., Wang W., Chen L.C., Tan M., Chu G., Vasudevan V., Zhu Y., Pang R., Adam H., Le Q. (2019): Searching for mobilenetv3. In: Proc. 2019 IEEE/CVF Int. Conf. on Computer Vision (ICCV), Seoul, Oct 27–Nov 2, 2019: 1314–1324.

Khan I., Sohail S.S., Madsen D.Ø., Khare B.K. (2024): Deep transfer learning for fine-grained maize leaf disease classification. Journal of Agriculture and Food Research, 16: 101148.

Lamba S., Kukreja V., Rashid J., Gadekallu T.R., Kim J., Baliyan A., Gupta D., Saini S. (2023): A novel fine-tuned deep-learning-based multi-class classifier for severity of paddy leaf diseases. Frontiers in Plant Science, 14: 1234067.

Li J., Zhu Z., Liu H., Su Y., Deng L. (2023): Strawberry R-CNN: Recognition and counting model of strawberry based on improved faster R-CNN. Ecological Informatics, 77: 102210.

Manavalan R. (2020): Automatic identification of diseases in grains crops through computational approaches: A review. Computers and Electronics in Agriculture, 178: 105802.

https://doi.org/10.17221/59/2024-RAE

Mohammed A., Kora R. (2023): A comprehensive review on ensemble deep learning: Opportunities and challenges. Journal of King Saud University – Computer and Information Sciences, 35: 757–774.

Pascual E.J.A.V., Plaza J.M.J., Tesorero J.L.L., De Goma J.C. (2019): Disease detection of asian rice (*Oryza sativa*) in the Philippines using image processing. In: Proc. 2nd Int. Conf. on Computing and Big Data, Taichung, Oct 18–20, 2019: 131–135.

Petchiammal A, Kiruba B., Murugan D, Arjunan P. (2022): Paddy doctor: A visual image dataset for automated paddy disease classification and benchmarking. IEEE DataPort. Available at https://ieee-dataport.org/documents/paddy-doctor-visual-image-dataset-automated-paddy-disease-classification-and-benchmarking

Ramesh S., Hebbar R., Nivedhita M., Pooja M., Bhad P., Shashank N., Vinod P.V. (2018): Plant disease detection using machine learning. In: Proc. 2018 Int. Conf. on Design Innovations for 3Cs Compute Communicate Control (ICDI3C), Bangalore, April 25–28, 2018: 41–45.

Shabrina N.H., Brian A. (2023): A comparative analysis of pre-trained deep neural networks for mango leaves pests and diseases identification. ICIC Express Letters, Part B : Applications, 14: 1207–1215.

Shabrina N.H., Indarti S., Maharani R., Kristiyanti D.A., Irmawati, Prastomo N., Adilah M.T. (2024): A novel dataset of potato leaf disease in uncontrolled environment. Data in Brief, 52: 109955.

Shorten C., Khoshgoftaar T.M. (2019): A survey on image data augmentation for deep learning. Journal of Big Data, 6: 60.

Tan M., Le Q.V. (2019): EfficientNet: Rethinking model scaling for convolutional neural networks. In: Proc. 36th Int. Conf. on Machine Learning, Long Beach, June 9–15, 2019: 6105–6114.

Tan M., Le Q.V. (2021): EfficientNetV2: Smaller models and faster training. In: Proc. 38th Int. Conf. on Machine Learning, Virtual, July 18–24, 2021: 10096–10106.

Wang A.R., Shabrina N.H. (2023): A deep learning-based mobile app system for visual identification of tomato plant disease. International Journal of Electrical and Computer Engineering, 13: 6992–7004.

Wenjing Z., Zezhong J., Lifeng Q. (2023): Identifying multiple apple leaf diseases based on the improved cbam-resnet18 model under weak supervision. Smart Agriculture, 5: 111–121.

Received: July 10, 2024
Accepted: October 31, 2025
Published online: January 19, 2026