IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 4, August 2026, pp. 3318~3325
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i4.pp3318-3325

Android-based tomato leaf disease classification using a lightweight MobileNetV2 convolutional neural network

Andi Riansyah[1], Irfan Eka Mahdy[1], Mochamad Abdul Basir[2], Noorminshah A. Iahad[3]
[1]Department of Informatics Engineering, Faculty of Industrial Technology, Universitas Islam Sultan Agung, Semarang, Indonesia
[2]Mathematics Education Study Program, Faculty of Teacher Training and Education, Universitas Islam Sultan Agung, Semarang, Indonesia
[3]Department of Applied Computing and Artificial Intelligence, Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, Malaysia

Article Info

Article history:

Received Jan 25, 2025
Revised Jun 5, 2026
Accepted Jul 9, 2026

Keywords:

Android application
Convolutional neural network
MobileNetV2
Plant disease classification
Tomato leaf disease

ABSTRACT

Early screening of tomato leaf diseases is important because foliar symptoms can reduce plant vigor and delay appropriate crop management. This study develops an Android-based tomato leaf disease classification system using MobileNetV2 as a lightweight convolutional neural network (CNN) architecture. The contribution of this work is the integration of model training, independent testing, and on-device Android deployment that supports camera and gallery inputs without relying on server-side computation. The dataset consisted of 1,200 balanced tomato leaf images from five classes: bacterial spot, late blight, target spot, tomato yellow leaf curl virus, and healthy leaf. Images were resized, normalized, augmented for training, and divided into training, validation, and independent testing subsets. The model obtained 94.12% training accuracy, 93.00% validation accuracy, and 89.00% independent test accuracy. The confusion matrix showed that tomato yellow leaf curl virus was classified without error, whereas bacterial spot, late blight, target spot, and healthy leaves produced several misclassifications because of similar lesion and discoloration patterns. The results show that MobileNetV2 is suitable for lightweight mobile disease screening, although larger field datasets, cross-validation, model comparison, and explainability analysis are still needed for broader deployment.

Corresponding Author:

Andi Riansyah
Department of Informatics Engineering, Faculty of Industrial Technology, Universitas Islam Sultan Agung
St. Raya Kaligawe Km. 4, Semarang 50112, Indonesia
Email: andi@unissula.ac.id

1. INTRODUCTION

Tomato has strategic value as a horticultural crop because it is consumed fresh, processed into food products, and cultivated widely in many agricultural areas. Indonesian production data show that tomato farming covered approximately 63,369 hectares and produced about 1,168,744 tons in 2022 [1]–[3]. This production potential needs to be protected because consumer demand continues to grow, while leaf diseases can reduce photosynthetic capacity, weaken plant development, and lower harvest quality. Bacterial, fungal, and viral symptoms on leaves often appear before severe yield loss occurs. Consequently, fast and accessible disease screening has become an important requirement in precision agriculture and smart crop management [4]–[7].

The research problem addressed in this work is the automatic classification of tomato leaf images into disease categories. The input is a leaf image, while the output is one of five labels: bacterial spot, late blight, target spot, tomato yellow leaf curl virus, or healthy leaf. These categories are visually represented by differences in lesion shape, spot distribution, color degradation, and texture. A convolutional neural network (CNN) is appropriate for this task because it can learn discriminative visual patterns directly from image pixels. MobileNetV2 was adopted as core model because its depthwise separable convolutions and inverted residual structure reduce computational complexity while preserving feature extraction capability [8]–[11].

In an ideal field situation, farmers and extension officers should be able to identify leaf diseases quickly, consistently, and near the cultivation site. However, identification is commonly based on manual observation, which is influenced by experience, image conditions, disease stage, and similarity among symptoms. Spot-like lesions, yellowing, necrotic areas, and uneven lighting may cause incorrect preliminary diagnosis. Laboratory examination and expert consultation are more reliable, but they may be costly, slow, or unavailable in small farming communities. This condition creates a practical gap for a smartphone-based system that can support early screening before further expert confirmation is performed.

Deep learning approaches have been widely explored for plant disease recognition, including CNN variants and lightweight models for mobile or low-resource environments [12]–[17]. Although these studies demonstrate the capability of neural networks in recognizing visual disease features, several limitations remain. Many works emphasize benchmark accuracy but provide less detail about data preparation, augmentation strategy, hyperparameter choices, and the reproducibility of the experimental pipeline. In addition, some studies stop at model evaluation and do not explain the transition from trained model to Android-based use. The remaining gap is therefore a compact framework that links MobileNetV2 training, independent testing, mobile integration, and error analysis for tomato leaf disease recognition.

The gap is important because mobile deployment changes the value of a classification model from laboratory evaluation to practical decision support. From an academic perspective, the study clarifies how a lightweight CNN can be prepared, tested, and embedded in an Android workflow. From a practical perspective, the application can assist farmers, students, and agricultural field officers in obtaining an initial disease indication from camera or gallery images without server-side processing. Without this type of implementation, image-based plant disease recognition may remain difficult to use in locations with limited access to experts, unstable connectivity, or low computational resources.

Accordingly, this study aims to develop and evaluate an Android-based tomato leaf disease classification system using MobileNetV2. The objects analyzed were tomato leaf images representing four disease categories and one healthy category. The study examined the relationship between image input, preprocessing configuration, model training, and classification performance. The expected contribution is a lightweight classification model that can be embedded in an Android application and used for preliminary disease screening. The study also provides a basis for future development involving larger field datasets, k-fold validation, explainable artificial intelligence, and comparison with alternative lightweight architectures.

2. METHOD

2.1. Dataset

This study used the segmented version of a publicly available Kaggle tomato leaf disease image dataset. From the ten categories in the original dataset, five classes were selected for the experiment: bacterial spot, late blight, target spot, tomato yellow leaf curl virus, and healthy leaf. Each class contributed 240 images, so the total dataset contained 1,200 images. The images were arranged in a balanced composition consisting of 900 training images, 200 validation images, and 100 independent testing images. The testing set was not involved in training or model selection and was used only for final evaluation through the confusion matrix.

All images were resized to 224×224 pixels before being processed by MobileNetV2. Pixel intensities were scaled into the range of 0 to 1 to make the input distribution more stable during optimization. Augmentation was applied only to the training subset through rotation, horizontal flipping, zooming, brightness variation, and small spatial shifts. The validation and testing subsets received only resizing and normalization to avoid artificially changing the evaluation data. A fixed random seed was used during data splitting so that the experiment can be repeated with the same subset composition.

2.2. Convolutional neural network

Figure 1 presents the proposed CNN-based classification pipeline. The process begins with a tomato leaf image, followed by preprocessing, feature extraction, and class prediction. CNN layers learn visual representations progressively, beginning from local features such as edges, color boundaries, and texture patterns and moving toward higher-level disease characteristics [13]–[15]. MobileNetV2 was chosen because its lightweight architecture is suitable for Android devices while still maintaining sufficient representation capacity for leaf disease images [16], [17].

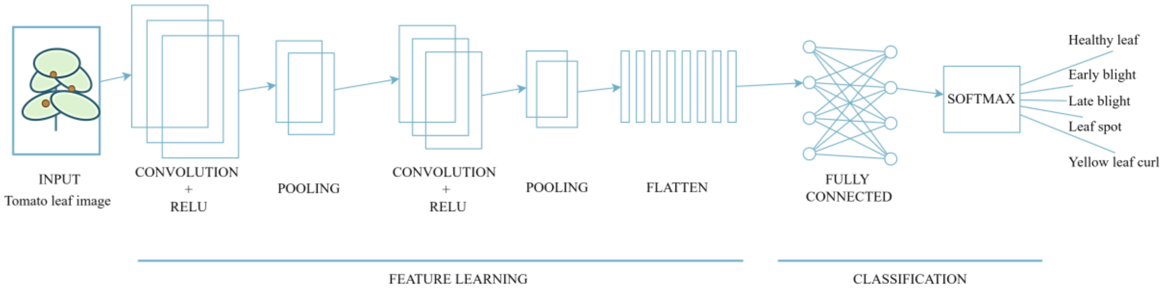

Figure 1. Process flow of the CNN method

During feature extraction, the MobileNetV2 backbone transforms the preprocessed image into compact feature maps. These feature maps are then processed by the classification head, which produces probability scores for all target classes through the softmax layer [18], [19]. The label with the largest probability is selected as the prediction result. This architecture was selected because it supports efficient inference and can be converted for use inside an Android application.

The training process was conducted for 30 epochs using stochastic gradient descent (SGD), a learning rate of 0.0001, and a batch size of 16. This small learning rate was selected to keep parameter updates gradual, while the batch size was chosen to balance memory usage and gradient stability. The configuration was determined by monitoring the training and validation curves and selecting the setting that produced stable convergence with low validation loss. The preprocessing procedure, split ratio, epoch number, optimizer, learning rate, and batch size were kept consistent across experiments to improve reproducibility.

2.3. Mobile application workflow

The overall workflow of mobile application used in this study is illustrated in Figure 2 [20]–[22]. The flowchart describes the sequence of processes starting from user interaction to image classification output. Users can capture images using the camera or select images from the gallery, after which the system automatically performs disease classification and displays the results [23].

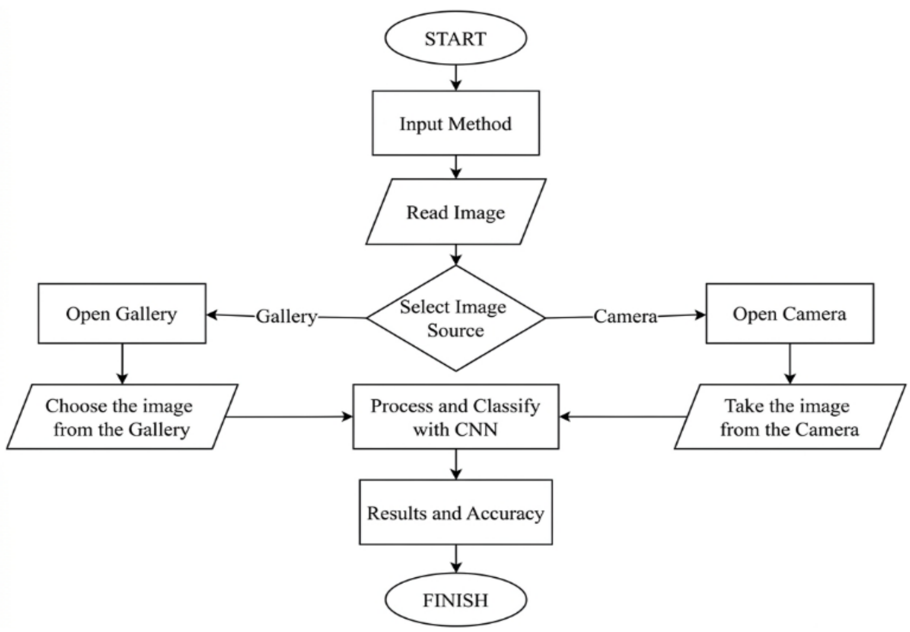

Figure 2. Mobile application flowchart

In the implemented workflow, the application first directs users to the main page, where image input can be selected. Once an image is provided, the system converts it into the required input size and format before running on-device classification. The predicted class is then shown on the application interface together with the classification information. The application is intended for preliminary screening in agricultural environments; therefore, expert confirmation remains necessary for critical treatment decisions.

3. RESULTS AND DISCUSSION

3.1. Training and validation

The experiment used a balanced split of 1,200 images. A total of 900 images were used for training, 200 images for validation, and 100 images for independent testing. Each class had the same number of samples in every subset to reduce the possibility of dominance by a particular disease category. The validation data were used to observe learning stability during training, whereas the testing data were reserved for the final performance assessment. Figure 3 displays the accuracy trend over 30 epochs. The training accuracy increased gradually, and the validation accuracy followed a similar pattern. The closeness between the two curves indicates that the model learned relevant disease features without showing a strong overfitting pattern.

Figure 4 presents the loss curves for training and validation. Both curves decreased during training, showing that the optimization process reduced classification error. The stable validation loss also suggests that the selected configuration produced acceptable generalization on unseen validation images. The final configuration used SGD, a learning rate of 0.0001, a batch size of 16 and 30 epochs. This configuration was retained because it produced consistent convergence in the accuracy and loss curves. Even so, the evaluation remains constrained by the limited dataset size and the use of a single train-validation-test split. Future experiments should apply k-fold cross-validation and include larger images collected directly from farms to increase the reliability of the findings.

At the end of training, the model obtained a training accuracy of 94.12% with a loss value of 0.1503. The validation accuracy reached 93.00%, while the validation loss was 0.0797. These results indicate that MobileNetV2 was able to extract useful disease-related visual patterns from the selected dataset. However, these values should not be interpreted alone because validation images may not fully represent field variation; therefore, the independent test result provides a more realistic performance indication [24]–[27].

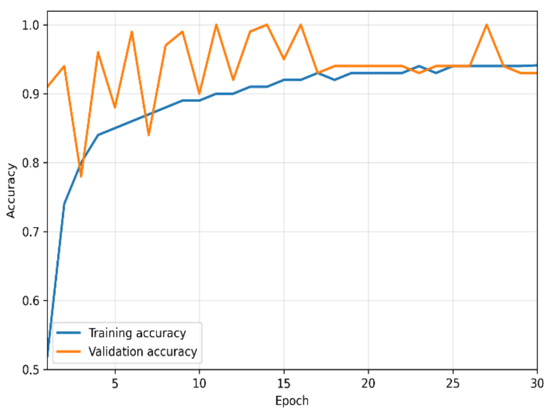

Figure 3. Training and validation accuracy curves

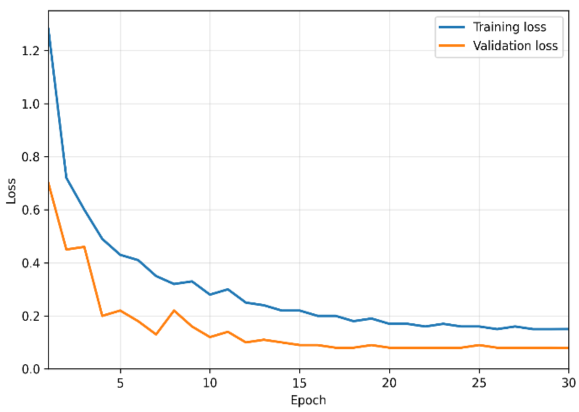

Figure 4. Training and validation loss curves

3.2. Testing

Figure 5 shows the confusion matrix obtained from 100 independent test images, consisting of 20 images for each class. The matrix was used to derive accuracy, precision, recall, and F1-score, as well as to observe which categories were confused by the classifier [18], [28]. This evaluation was separated from training and validation to provide an independent measurement of classification performance.

The confusion matrix indicates that all tomato yellow leaf curl virus images were classified correctly. The remaining errors mainly involved bacterial spot, healthy leaf, late blight, and target spot. This result suggests that symptom overlap, especially spot-like lesions and color degradation, still affects model decisions. The independent test produced an overall accuracy of 89.00%, and the detailed class-level results are presented in Table 1.

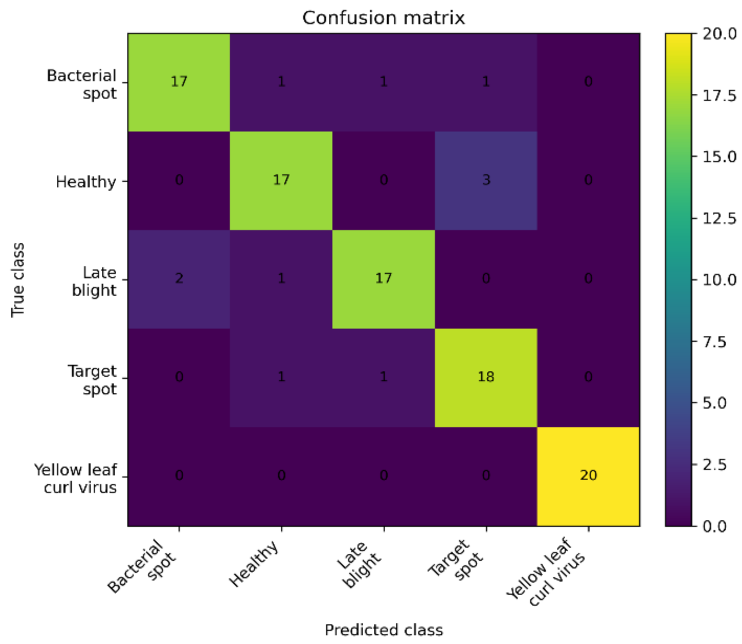

Figure 5. Confusion matrix for the independent test subset

Table 1. Classification performance on the independent test subset

Class	Precision	Recall	F1-score
Bacterial spot	0.89	0.85	0.87
Healthy leaf	0.85	0.85	0.85
Late blight	0.89	0.85	0.87
Target spot	0.82	0.90	0.86
Yellow leaf curl virus	1.00	1.00	1.00
Macro	0.89	0.89	0.89
Overall accuracy			0.89

Table 1 summarizes the class-level test performance. The macro precision, recall, and F1-score were all 0.89, indicating relatively balanced performance across classes. The yellow leaf curl virus class achieved the best result, with precision, recall, and F1-score all equal to 1.00. Bacterial spot and late blight obtained similar F1-scores of 0.87, while the healthy class reached an F1-score of 0.85. Target spot achieved a recall of 0.90 but a lower precision of 0.82, meaning that several images from other classes were assigned to this category. These findings show that MobileNetV2 performed well, although symptom similarity still caused classification ambiguity. Figure 6 presents the Android interface developed for tomato leaf disease classification. The interface demonstrates the integration of the trained MobileNetV2 model into a mobile application. Users can provide an input image from the camera or gallery, and the application returns the predicted disease category directly on the device.

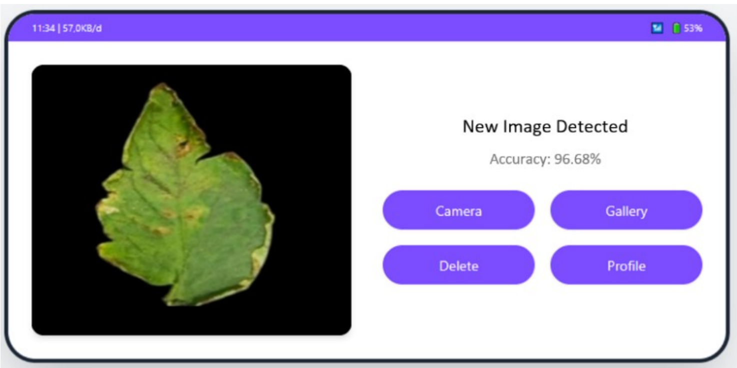

Figure 6. Android application interface for tomato leaf disease classification

The Android implementation extends the contribution of this work beyond model training. By embedding the classifier into a mobile workflow, the system can support preliminary disease recognition without continuous internet access or server-side computation. This implementation offers a practical path for use by farmers, agricultural extension officers, students, and crop-monitoring teams. It also demonstrates how a lightweight CNN can be connected to a simple user interface for field-oriented screening.

MobileNetV2 is not introduced as a new architecture in this study; rather, it is used because its computational efficiency is suitable for mobile agricultural applications. The main contribution lies in the integration of a lightweight classifier, independent testing, and Android-based deployment for tomato leaf disease screening. Nevertheless, the current work does not yet compare MobileNetV2 with EfficientNet, DenseNet, or transformer-based vision models. Such comparisons are needed to determine whether MobileNetV2 provides the most appropriate accuracy-efficiency trade-off for this application.

For wider deployment, the system should be tested under more diverse field conditions. Lighting variation, background clutter, leaf orientation, disease stage, and camera quality can influence classification results in real farms. The application may also be expanded by combining image classification with environmental data from internet of things (IoT) sensors, such as humidity, temperature, and soil moisture. Future versions should include farm-acquired datasets, explainability methods such as gradient-weighted class activation mapping (Grad-CAM) or saliency maps, and architecture comparisons to improve transparency and scalability.

4. CONCLUSION

This study designed an Android-based tomato leaf disease classification system using a lightweight MobileNetV2 CNN. The model achieved 94.12% training accuracy, 93.00% validation accuracy, and 89.00% accuracy on the independent test subset. These results indicate that MobileNetV2 can provide a reasonable balance between classification performance and mobile deployment efficiency. The Android implementation allows users to classify tomato leaf images through camera or gallery input, making the framework useful as an early screening tool in agricultural settings. However, the study is still limited by the dataset size, the use of a single split, the absence of architecture comparison, and the lack of visual explainability. Future research should use larger field datasets, apply k-fold validation, compare MobileNetV2 with other lightweight and transformer-based models, integrate Grad-CAM or saliency maps, and combine image-based predictions with IoT-based environmental monitoring.

ACKNOWLEDGMENTS

The authors express their sincere gratitude to the Center for Industrial Intelligence Studies, Universitas Islam Sultan Agung, for the academic support, facilities, and research environment provided during the completion of this study. The authors also appreciate the constructive input received during the development of the Android-based tomato leaf disease classification system and the preparation of this manuscript. This support contributed to the successful completion of the research.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Andi Riansyah	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓
Irfan Eka Mahdy		✓	✓			✓	✓	✓	✓	✓	✓	✓		
Mochamad Abdul Basir	✓			✓	✓		✓			✓	✓		✓	✓
Noorminshah A. Iahad	✓			✓	✓		✓			✓	✓		✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that there are no financial, personal, professional, or non-financial conflicts of interest that could have influenced the research process, results, interpretation, or manuscript preparation. The authors also confirm that no competing interests are associated with this work.

DATA AVAILABILITY

The data used in this study were derived from a publicly available tomato leaf disease image dataset. The processed subset, experimental configuration, and evaluation results are retained by the authors and may be made available from the corresponding author, [AR], upon reasonable request for academic and reproducibility purposes.

REFERENCES

[1] A. Riansyah, S. Mulyono, and M. Roichani, "Applying fuzzy proportional integral derivative on internet of things for figs greenhouse," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 3, pp. 536–544, Sep. 2021, doi: 10.11591/ijai.v10.i3.pp536-544.

[2] L. Xu, Y. Dong, Y. Zhang, P. Xu, Z. Cai, and J. Zhao, "MLSA: a multi-head latent and self-attention deep learning network for tomato leaf disease identification," *Smart Agricultural Technology*, vol. 12, Dec. 2025, doi: 10.1016/j.atech.2025.101604.

[3] A. Marwanto, A. Riansyah, and M. K. Anwar, "An improvement of apple leaf diseases detection using convolutional neural network methods based on mobile systems," in *2024 11th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, Sep. 2024, pp. 440–447. doi: 10.1109/EECSI63442.2024.10776345.

[4] M. Gupta, S. Garg, N. Deb, and P. Pandey, "CS-D: a dataset for disease in tea leaf (*Camellia sinensis*) from Assam," *Data in Brief*, vol. 63, Dec. 2025, doi: 10.1016/j.dib.2025.112251.

[5] A. Arora and V. Gautam, "Plant leaves disease detection in complex background using automatic segmentation and improved multi-scale feature fusion network," *Systems and Soft Computing*, vol. 8, Jun. 2026, doi: 10.1016/j.sasc.2025.200434.

[6] D. K. Saha, M. R. Ahmed, T. D. Nath, R. I. Boby, M. J. Hossen, and M. F. Mridha, "Fusing explainable deep learning ensembles and LLM recommendations for real-time plant leaf disease diagnosis," *Intelligent Systems with Applications*, vol. 28, Dec. 2025, doi: 10.1016/j.iswa.2025.200596.

[7] M. M. Billah, S. Sagor, and M. S. Uddin, "A comprehensive combined dataset on hibiscus and tea plant leaf disease images for classifications," *Data in Brief*, vol. 64, Feb. 2026, doi: 10.1016/j.dib.2025.112357.

[8] N. Deb and T. Rahman, "Lightweight CNN and MobileNetV2 with feature selection and explainable AI for efficient insect detection in potato plants," *Journal of Agriculture and Food Research*, vol. 25, Feb. 2026, doi: 10.1016/j.jafr.2025.102604.

[9] V. P. Raj, K. Radha, S. H. Gopalan, C. Venkataramanan, R. Dhanapal, and A. Manikandan, "COA-SSD: a novel chaotic optimization algorithm for enhancement of real-time video surveillance images using single shot detector with MobileNetV2," *Systems and Soft Computing*, vol. 7, Dec. 2025, doi: 10.1016/j.sasc.2025.200417.

[10] M. S. A. M. Al-Gaashani, A. S. B. Mahel, and A. Muthanna, "Truncated MobileNetV2 sparse vision graph attention model for explainable monkeypox disease classification," *Knowledge-Based Systems*, vol. 330, 2025, doi: 10.1016/j.knosys.2025.114503.

[11] M. Chilakarao, S. K. Behera, K. P. Raju, P. P. Dash, P. K. Sethy, and A. Nanthaamornphong, "Domain-aligned feature extraction with Mobilenetv2-GRL and KNN for wheat leaf disease detection," *Results in Engineering*, vol. 28, Dec. 2025, doi: 10.1016/j.rineng.2025.108118.

[12] P. Yuan, Z. Chen, Q. Jin, Y. Xu, and H. Xu, "Lightweight lotus phenotype recognition based on MobileNetV2-SE with reliable pseudo-labels," *Computers and Electronics in Agriculture*, vol. 232, May 2025, doi: 10.1016/j.compag.2025.110080.

[13] Y.-C. Chen, J.-C. Wang, M.-H. Lee, A.-C. Liu, and J.-A. Jiang, "Enhanced detection of mango leaf diseases in field environments using MSMP-CNN and transfer learning," *Computers and Electronics in Agriculture*, vol. 227, Dec. 2024, doi: 10.1016/j.compag.2024.109636.

[14] W. Jiao, C. Persello, and G. Vosselman, "PolyR-CNN: R-CNN for end-to-end polygonal building outline extraction," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 218, pp. 33–43, Dec. 2024, doi: 10.1016/j.isprsjprs.2024.10.006.

[15] U. Athiyah, S. Rustad, M. A. Soeleman, M. Muljono, M. Akrom, and R. A. Konoralma, "Data enhancement to improve CNN accuracy for early detection of colorectal cancer," *Array*, vol. 28, Dec. 2025, doi: 10.1016/j.array.2025.100581.

[16] L. Nanni and S. Brahnam, "Vector to matrix representation for CNN networks for classifying astronomical data," *Astronomy and Computing*, vol. 49, Oct. 2024, doi: 10.1016/j.ascom.2024.100864.

[17] R. T. A. Guna, O. K. Sikha, and R. Benitez, "Interpreting CNN predictions using conditional generative adversarial networks," *Knowledge-Based Systems*, vol. 302, Oct. 2024, doi: 10.1016/j.knosys.2024.112340.

[18] B. B. Gupta, S.-H. Pan, A. Gaurav, V. Arya, W. Alhalabi, and K. T. Chui, "Optimized hybrid CNN-LSTM-GRU model for cyber attack detection using adaptive equilibrium optimization," *ICT Express*, vol. 12, no. 2, pp. 324–329, Apr. 2026, doi: 10.1016/j.icte.2025.09.006.

[19] S. Liu, W. Wang, L. Deng, and H. Xu, "CNN-trans model: a parallel dual-branch network for fundus image classification," *Biomedical Signal Processing and Control*, vol. 96, Oct. 2024, doi: 10.1016/j.bspc.2024.106621.

[20] I. Grijalva, H. B. Adams, and B. McCornack, "A fine-tuned deep learning model for detecting Japanese beetles in soybeans using unmanned aircraft systems (UAS) and mobile imaging," *Machine Learning with Applications*, vol. 21, Sep. 2025, doi: 10.1016/j.mlwa.2025.100711.

[21] C. S. -López, "Maze solving mobile robot based on image processing and standard nodal analysis," *Franklin Open*, vol. 12, Sep. 2025, doi: 10.1016/j.fraope.2025.100357.

[22] J. Wen *et al.*, "Accuracy and clinical application prospects of mobile low-field strength magnetic resonance imaging in the diagnosis of hydrocephalus: a retrospective cohort study," *Brain Research Bulletin*, vol. 232, Nov. 2025, doi: 10.1016/j.brainresbull.2025.111585.

[23] M. Huang and J. Zhu, "A fast image recognition method of mobile point radioactive source based on sequential bayesian analysis," *Radiation Physics and Chemistry*, vol. 237, Dec. 2025, doi: 10.1016/j.radphyschem.2025.113060.

[24] M. M. Yapici, “A novel image based approach for mobile android malware detection and classification,” *Knowledge-Based Systems*, vol. 323, Jul. 2025, doi: 10.1016/j.knosys.2025.113855.
[25] D. K. N. Ho *et al.*, “Mitigating errors in mobile-based dietary assessments: effects of a data modification process on the validity of an image-assisted food and nutrition app,” *Nutrition*, vol. 116, Dec. 2023, doi: 10.1016/j.nut.2023.112212.
[26] S. Titouni *et al.*, “Hybrid CNN-autoencoder model for accurate and efficient fault diagnosis in grid-connected photovoltaic systems,” *International Journal of Electrical Power & Energy Systems*, vol. 173, Dec. 2025, doi: 10.1016/j.ijepes.2025.111454.
[27] T. P. T. Nguyen, T. T. Nguyen, H. Q. Nguyen, T. D. Nguyen, C. K. Nguyen, and N. G. Cu, “An enhanced image classification model based on graph classification and superpixel-derived CNN features for agricultural datasets,” *Computers, Materials & Continua*, vol. 85, no. 3, pp. 4899–4920, 2025, doi: 10.32604/cmc.2025.067707.
[28] J. S. ThangaPurni and M. Braveen, “Unified ARP-ViT-CNN system: hybrid deep learning approach for segmenting and classifying multiple skin cancer lesions,” *Array*, vol. 28, Dec. 2025, doi: 10.1016/j.array.2025.100515.

BIOGRAPHIES OF AUTHORS

Andi Riansyah is a lecturer in Department of Informatics Engineering, Faculty of Industrial Technology, Universitas Islam Sultan Agung, Semarang, Indonesia. He primary expertise is in Informatics, particularly artificial intelligence, machine learning, and computer vision for plant disease detection and medical diagnosis. He also develops artificial intelligence-based chatbots and analyzes public opinion through natural language processing. His work includes IoT, fuzzy logic, publications, and patents benefiting healthcare, agriculture, and SMEs. He can be contacted at email: andi@unissula.ac.id.

Irfan Eka Mahdy is a data and machine learning enthusiast from Semarang and a graduate of Department of Informatics Engineering, Faculty of Industrial Technology, Universitas Islam Sultan Agung, Semarang, Indonesia. His interests include data analysis, machine learning, Android applications, and web technologies. He develops data-driven web platforms to create intelligent, scalable solutions for real-world business and social challenges. He can be contacted at email: irfanekam@gmail.com.

Mochamad Abdul Basir is a lecturer in the Mathematics Education Study Program, Faculty of Teacher Training and Education, Universitas Islam Sultan Agung, Semarang. His expertise focuses on mathematics education, numeracy literacy, algebraic reasoning, and mathematics anxiety in learning. He is also active in developing learning innovations, writing scientific publications and textbooks, and serving as a peer reviewer in several academic fields. He can be contacted at email: abdulbasir@unissula.ac.id.

Noorminshah A. Iahad is an academic at the Department of Applied Computing and Artificial Intelligence, Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, Malaysia. She is a recognized digital leader and was one of 330 participants selected nationally to complete the Digital Leadership Experience Program organized by the Malaysia Communications and Multimedia Commission and Huawei. She currently serves as president of the Association for Information Systems. She can be contacted at email: minshah@utm.my.