OPEN ACCESS
VOL. 7, NO. 2, PP.203-212, MEI 2025

ISSN : 2622-8106 (ONLINE)
DOI: DOI 10.20895/INISTA.V7I2.1825

Journal of Informatics, Information System, Software Engineering and Applications (INISTA)

Application of MobileNetV2-Based Deep Learning in Detecting Diseases in Chili Plants

Nurseno Bayu Aji [*1], Tri Raharjo Yudantoro [2], Zulfa Safitri [3], Samuel Beta Kuntardjo[4], Mardiyono[5], Prayitno[6], Kuwat Santoso[7]

[1,2,3,4,5,6,7] *Department of Electrical Engineering, Politeknik Negeri Semarang*
[1,2,3,4,5,6,7] *Jl. Prof. H. Soedarto SH, Tembalang, Semarang, Central java, Postal Code 50275*

[1] bayu.nurseno@polines.ac.id, [2] tryudan@polines.ac.id, [3] zulfasafitri2063@gmail.com, [4] sambetak2@polines.ac.id, [5] mardiyono@polines.ac.id, [6] prayitno@polines.ac.id, [7] kuwatsantoso@polines.ac.id

Received on 28-04-2025, revised on 15-05-2025, accepted on 26-05-2025

Abstract

This study proposes a deep learning model based on MobileNetV2 architecture for the classification of chili leaf diseases using image data. The dataset was compiled from both public and private sources, covering six distinct categories of chili leaf conditions. MobileNetV2 was selected due to its efficiency and accuracy, making it ideal for real-time agricultural applications. The model was enhanced with additional layers to improve feature extraction and classification performance. Stratified 10-fold cross-validation was employed to ensure balanced evaluation across an imbalanced dataset. The experimental results showed an overall accuracy of 91.04% and an average F1-score of 0.906, indicating consistent and reliable classification performance across classes. Confusion matrix analysis highlighted strong predictive capability, particularly in detecting healthy leaves and severe disease symptoms, with minor misclassifications among visually similar categories. The findings confirm the potential of lightweight CNN architectures for practical, mobile-based agricultural diagnostics, contributing to advancements in precision farming and early disease management.

Keywords: MobileNetV2, Deep Learning, Chili Pepper Disease, Computer Vision

Corresponding Author:
*Nurseno Bayu Aji
Department of Electrical Engineering, Politeknik Negeri Semarang, Central Java, Indonesia
Jl. Prof. H. Soedarto SH, Tembalang, Semarang, Central java, Postal Code 50275
Email: bayu.nurseno@polines.ac.id

I. INTRODUCTION

Chili pepper (*Capsicum spp.*) is an economically significant horticultural commodity in both domestic and global markets [1]. However, its productivity is highly susceptible to environmental factors such as humidity, temperature, and water stress, which affect the plant's vegetative and generative phases [2]. Fluctuations in environmental conditions and disease outbreaks often cause yield instability and price volatility [1], [3].

Leaf-related diseases are a primary cause of reduced chili yields, commonly marked by visible symptoms such as yellowing, curling, or spotting [4]. Traditional disease identification relies heavily on manual observation and chemical treatments, both of which have limitations in scalability and sustainability [5]. Therefore, accurate and early detection systems are essential to support precision agriculture and reduce unnecessary agrochemical use.

Recent advances in artificial intelligence (AI) and computer vision have enabled the development of automated tools for monitoring plant health. These systems leverage deep learning algorithms to analyze captured images of plant leaves and detect disease symptoms in real time. For example, drones or stationary

cameras can collect visual data in the field, which is used to train models to support rapid and informed decision-making [6].

Convolutional Neural Networks (CNNs) have proven effective in extracting spatial patterns from images for various classification tasks [7], [8]. Among lightweight CNN architectures, MobileNetV2 is particularly advantageous for field applications due to its efficient structure, incorporating inverted residuals and linear bottlenecks, which significantly reduce computational complexity without compromising accuracy [9].

Recent studies have demonstrated the robustness of deep learning in plant disease detection across various crops. For instance, Too et al. compared several CNN architectures, including DenseNet and ResNet, for cassava and maize leaf diseases, achieving high accuracy through fine-tuning approaches. Similarly, Ferentinos employed deep networks such as AlexNet and VGG to classify 25 plant diseases, showing their effectiveness in large-scale diagnosis [10] ,[11]. However, most of these studies focus on controlled datasets and high-resource platforms, lacking considerations for mobile deployment or hybrid data variation.

This study proposes a MobileNetV2-based deep learning model for the classification of chili leaf diseases across six categories. To improve classification robustness, the model is enhanced with additional dropout and dense layers. The dataset integrates both public images (via Roboflow) and privately collected samples from actual chili plantations, providing diverse real-world data for training. The evaluation uses stratified 10-fold cross-validation to ensure balanced performance assessment on the imbalanced dataset. Novelty of this study lies in its hybrid dataset integration, architectural enhancements to MobileNetV2, and its readiness for mobile deployment. The model is optimized for use in resource-constrained environments and can be embedded into Android-based diagnostic applications. This enables chili farmers or field technicians to perform offline disease detection directly from smartphones, supporting early response and reducing crop losses in remote or infrastructure-limited areas.

II. RESEARCH METHOD

1. Data

In this phase of the study, a comprehensive dataset was assembled by integrating both private and public sources. Specifically, 20% of the data were obtained from a local chili pepper plantation, ensuring that unique real-world variability in the field could be captured. The remaining 80% were sourced from the public database available on the Roboflow website. Furthermore, primary data were directly collected from chili fields located close to the author's residence, thereby enhancing the dataset's authenticity and representativeness.

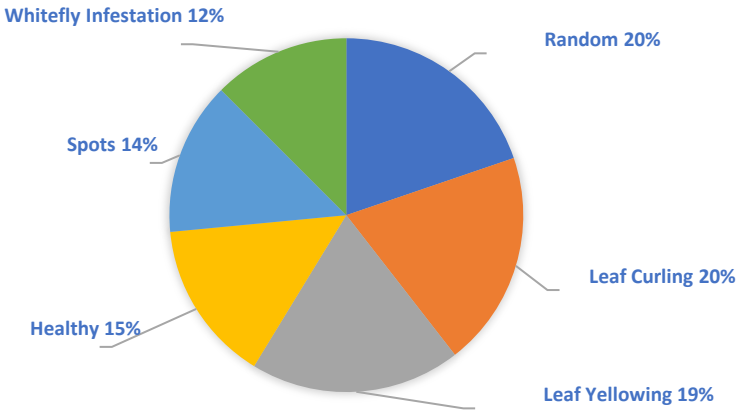

Fig. 1. Number of Label Distribution

The final dataset comprises a total of 2,120 image samples, which have been annotated into six distinct classes representing various conditions of chili leaves. The class distribution is as follows: the Random (Non-Chili Leaf) class contains 419 images of objects unrelated to chili leaves, such as leaves from other plants or various background elements, ensuring the model can distinguish chili leaves from irrelevant objects. The Leaf Curling class includes 418 images of chili leaves exhibiting symptoms of curling or twisting, often caused by viral infections, pests, or nutrient deficiencies. The Yellowing class, comprising

408 images, captures leaves undergoing chlorosis, characterized by a yellowing appearance typically linked to nutrient shortages or viral attacks. The Healthy Leaf class features 313 images of vibrant green, disease-free chili leaves, serving as a baseline for classification. The Spots class, with 297 images, illustrates leaves showing various spots due to fungal, bacterial, or viral infections. Lastly, the Whitefly class contains 265 images of chili leaves infested by whiteflies, which often lead to secondary symptoms such as yellowing and curling. Together, these classes provide a comprehensive representation of both healthy and diseased chili leaf conditions, essential for building a robust classification model. Figure 1 illustrates the diagram of per-class label distribution, providing a visual overview of the dataset composition. Figure 2 shows the image samples used in the data processing. These images were collected from public and private sources and show different conditions of chili leaves, both healthy and diseased. Some show symptoms like curling, yellowing, spots, and whitefly attacks.

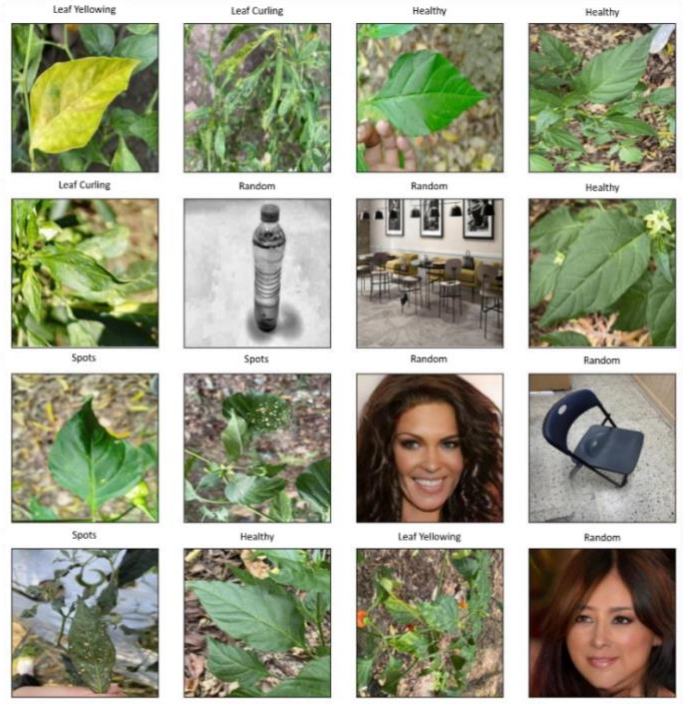

Fig. 2. Sample images from the dataset showing various chili leaf conditions

2. CNN

Convolutional Neural Networks (CNNs) are deep learning models designed for visual data processing, widely used in tasks such as image classification and object detection [12]. Inspired by the structure of the human visual cortex, CNNs apply multiple layers to extract features hierarchically from simple edges to complex shapes [11],[13]. Each convolutional layer uses filters to capture spatial patterns, enabling robust and translation-invariant recognition. Techniques like pooling, normalization, and dropout further enhance generalization and reduce overfitting. Due to their efficiency and accuracy, CNNs are well-suited for detecting visual symptoms in plant disease diagnosis, as applied in this study. Figure 3 shows the basic CNN architecture.

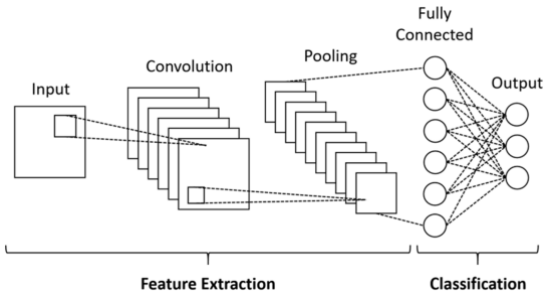

Fig. 3. Basic CNN Architecture

3. *MobileNet V2*

This study employs the MobileNetV2 architecture to construct a deep learning model aimed at detecting diseases in chili leaves. MobileNetV2 was selected due to its notable efficiency and accuracy, making it particularly advantageous for real-time operations on devices with limited computational resources [14], [15]. In this study, MobileNetV2 serves as the core feature extractor, enhanced with additional layers to refine feature representations and improve classification performance.

As illustrated in Table I, the model architecture begins by accepting an input image with dimensions of 224×224×3, representing an RGB image with three color channels. A 3×3 convolutional layer with a stride of 2 is then applied, reducing the spatial dimensions to 112×112×32 while capturing low-level features such as edges and textures. This operation is crucial for early-stage feature extraction, enabling the model to identify basic visual patterns. A Batch Normalization layer is used to stabilize the learning process and accelerate convergence by maintaining consistent activation distributions. Subsequently, a ReLU activation function introduces non-linearity, allowing the model to learn complex representations beyond simple linear mappings.

TABLE I. MOBILENET V2 ARCHITECTURE

Layer Type	Output Shape	Parameters
Input	224x224x3	0
Conv2D (k3x3, s2)	112x112x32	864
BatchNorm	112x112x32	128
ReLU	112x112x32	0
Bottleneck 1	112x112x16	896
Bottleneck 2-14	…….	…….
Bottleneck 15	7x7x320	1,293,120
Conv2D (k1x1)	7x7x1280	409,600
BatchNorm	7x7x1280	5,120
ReLU	7x7x1280	0
GlobalAvgPool	1x1x1280	0
ReLU	1x1x256	327,936
Dropout	1x1x256	0
ReLU	1x1x128	65,792
Dropout	1x1x128	0
Softmax	1x1x6	1,542
Total		2,653,254

The feature maps are then processed through a series of 15 bottleneck residual blocks, the core components of the MobileNetV2 architecture. Each bottleneck block performs three main operations: expansion to higher dimensions, lightweight depthwise convolution to extract spatial features efficiently, and projection back to a lower-dimensional space. These operations, combined with optional residual connections, significantly reduce computational complexity while preserving representational power. As a result, the spatial resolution of the feature maps progressively reduces to 7×7×320, while the depth of feature channels increases, allowing the network to capture increasingly abstract patterns relevant to disease symptoms.

Following the bottleneck stage, a 1×1 convolutional layer is applied to expand the feature maps to 7×7×1280, enriching the semantic information without changing the spatial size. Another Batch Normalization and ReLU activation are introduced to maintain activation stability and further refine the extracted features. A Global Average Pooling layer then condenses each 7×7 feature map into a single value, producing a 1×1×1280 feature vector. This pooling strategy reduces the risk of overfitting by summarizing spatial information globally, which is beneficial when the number of training samples is limited.

To further refine the features, the model introduces fully connected layers interleaved with ReLU activations and Dropout layers. The feature vector is first reduced to 1×1×256 using ReLU, followed by a Dropout layer to prevent overfitting by randomly deactivating neurons during training. A second reduction to 1×1×128 is performed, again followed by Dropout regularization. This design improves the model's ability to generalize, especially critical for preventing overfitting to particular visual patterns in an imbalanced chili leaf dataset. Finally, a Softmax activation layer projects the final feature representation to six output classes, providing probability scores for each category of chili leaf condition. The inclusion of Dropout and dimensionality reduction in the dense layers was intentionally designed to enhance the model’s robustness and to achieve better generalization across different disease variations. We used the Adam optimizer with a learning rate of 0.0001 and batch size of 32, chosen based on performance on validation

folds. Overall, this architecture successfully balances computational efficiency, feature extraction depth, and overfitting mitigation, making it highly suitable for real-time, mobile-based agricultural diagnostic systems.

4. Confusion Matrix

A confusion matrix serves as a fundamental instrument for assessing the performance of classification algorithms by aligning predicted labels with true labels in a table, where rows denote actual classes and columns denote predicted classes [16]. This setup facilitates a detailed examination of classification errors and enables the computation of key metrics such as accuracy, precision, recall, and F1. The confusion matrix can be seen in Figure 4. Equations (1), (2), and (3) present the formulas for F1 score, precision, and recall. In binary classification tasks, a basic 2×2 confusion matrix differentiates between true positives, false positives, true negatives, and false negatives. However, when addressing multiclass problems, the confusion matrix expands to an N×N format, requiring more nuanced analysis to uncover patterns of misclassification among multiple categories. For multilabel classifications, the complexity intensifies, prompting modifications such as adjusted multilabel confusion matrices to better handle partial predictions and uncertainties[17].

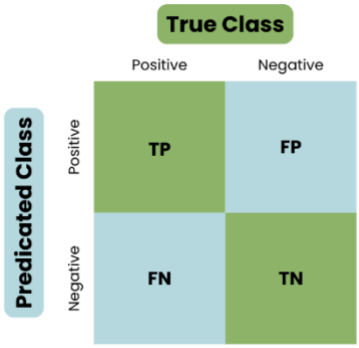

Fig. 4. Confusion Matrix

Recent theoretical advancements, such as the probabilistic confusion matrix, further enhance evaluation by incorporating probabilistic information to account for prediction uncertainties. Additionally, newer visualization methods like confusion stars and confusion gears have been introduced to improve interpretability by visually highlighting performance variations across different classes. Together, these innovations both in theory and visualization demonstrate ongoing efforts to optimize the evaluation of classification models across increasingly complex scenarios, spanning binary, multiclass, and multilabel tasks[18].

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

5. K-fold Cross Validation

K-fold cross-validation is a widely used evaluation technique in machine learning that improves model generalization by partitioning the dataset into k disjoint subsets. In each iteration, the model is trained on k–1 subsets and tested on the remaining one, ensuring that every subset serves once as the validation set. This approach minimizes the risks of overfitting and underfitting by fully utilizing the data for both training and validation, resulting in a more comprehensive assessment of predictive performance [19]. K-fold cross-validation also plays a critical role in hyperparameter tuning, such as optimizing SVM configurations or validating deep learning models through transfer learning techniques [20].

Additionally, k-fold cross-validation facilitates fair comparisons among classification algorithms like Naïve Bayes, K-Nearest Neighbors (KNN), CART, and logistic regression by averaging their performance metrics across folds to reduce sensitivity to specific train-test splits. The choice of k affects the bias-variance trade-off, with k = 10 often used as a practical compromise. Overall, k-fold cross-validation remains an

essential and evolving method in machine learning, providing a robust framework for model evaluation and hyperparameter optimization across a broad range of applications, from traditional statistical approaches to modern deep learning systems.

III. RESULTS AND DISCUSSION

The base model selected for developing this application is MobileNetV2. This architecture was chosen because of its lightweight nature, making it highly suitable for implementation on Android devices without consuming extensive resources. In this study, where the dataset is imbalanced, the Stratified K-Fold Cross Validation method was employed to prevent bias during model evaluation. This method partitions the dataset into multiple folds, ensuring balanced representation across classes. Using 10 folds, the model undergoes training and validation 10 times, with each fold serving as a validation set exactly once. The averaged results provide a more stable and less biased performance estimate compared to a single train-test split approach. This section discusses the outcomes of the training graphs, confusion matrix, and classification report, focusing on the average accuracy results.

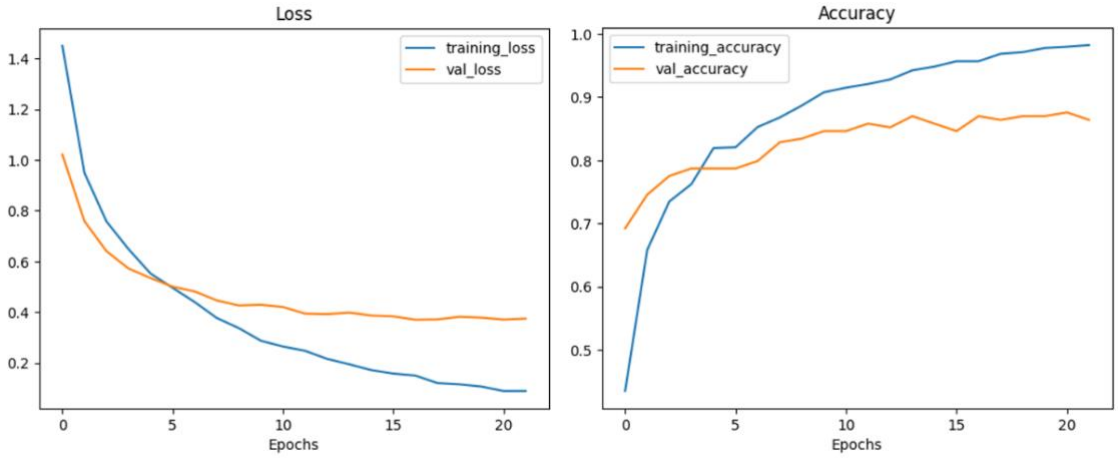

Fig. 5. Training Result

The training was conducted over 100 epochs with a batch size of 32. With a total of 1,526 training samples divided by the batch size, 48 batches were generated, and training generally converged around epoch 20 out of the scheduled 100 epochs.

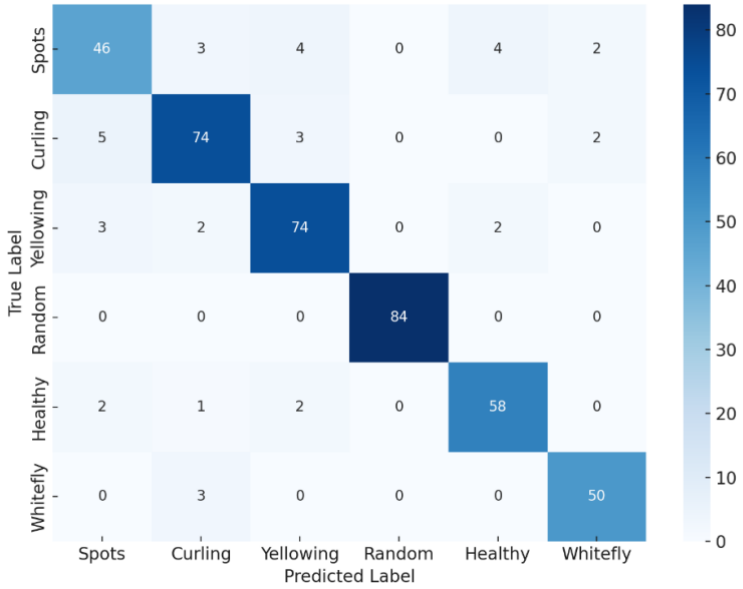

Fig.6 Confusion Matrix Testing Result

Figure 5 shows that the initial loss values for both training and validation were relatively high. As the number of epochs increased, the loss values progressively decreased and reached lower levels, indicating that the model was learning and improving its ability to predict the data accurately. Meanwhile, the accuracy graph shows that training accuracy was initially low; however, it gradually increased with each epoch, demonstrating that the model was effectively learning and enhancing its predictive performance.

The confusion matrix table was used to evaluate the performance of the classification model in the machine learning process. This table displays the number of correct and incorrect predictions made by the model for each category. It includes both true labels and predicted labels, each representing the names of chili leaf diseases. The testing results of the MobileNetV2 and CNN models produced a confusion matrix with the highest recorded value of 84 and the lowest value of 0. Some misclassifications occurred, such as yellow-diseased leaves being incorrectly predicted as healthy, curled, or spotted leaves.

Figure 6. presents the confusion matrix of the classification results obtained from the chili leaf disease detection model. The highest number of correct predictions was achieved in the "Random" class with 84 instances, followed by the "Yellowing" and "Curling" classes. Some misclassifications were observed, particularly among the "Spots," "Yellowing," and "Healthy" classes, where instances of leaf diseases were sometimes confused with one another.

Fig 7. Prediction Result of The Model.

The precision, recall, and F1-score analyses provide deeper insights into the model's performance. The "Random" class achieved perfect scores, with a precision, recall, and F1-score of 1.000, indicating flawless classification for this category. The "Whitefly" and "Healthy" classes also demonstrated strong results, with F1-scores of 0.935 and 0.913, respectively. "Yellowing" and "Curling" followed closely with F1-scores of 0.902 and 0.886. The "Spots" class showed the lowest performance among the categories, with an F1-score of 0.800, reflecting some challenges in distinguishing it from similar classes. The overall average F1-score across all classes was 0.906, indicating consistent and reliable classification performance. Figure 7 presents a comparison between the model's predictions and the ground truth labels.

To further assess the effectiveness of the proposed MobileNetV2 model, we conducted comparative experiments using the same dataset with two alternative deep learning architectures: VGG16 and a custom-implemented AlexNet. The VGG16 model achieved an accuracy of 51.58%, while the AlexNet-based model reached 64.59%. In contrast, the MobileNetV2 model significantly outperformed both, achieving an accuracy of 91.04%. These results highlight the model's robustness, particularly in identifying healthy and severely affected leaves, while also indicating areas for potential improvement in differentiating between visually similar symptoms.

Table II describes the 10-fold cross-validation process, where the model demonstrated consistent performance across different subsets of the data. The validation loss ranged from a minimum of 0.1206 to a maximum of 0.3700, while the validation accuracy varied between 86.98% and 95.88%. The highest validation accuracy was achieved in Fold 4 with 95.88%, accompanied by the lowest validation loss of 0.1206, indicating optimal model learning in this fold. In contrast, Folds 9 and 10 recorded the lowest validation accuracies at 86.98%. The average validation loss across all folds was 0.2660, and the average validation accuracy reached 90.21%. These results indicate that the model maintains stable and reliable performance, with minimal variance across different validation sets.

TABLE II. K-FOLD CROSS VALIDATION RESULT

Fold	Validation Loss	Validation Accuracy
Fold 1	0.36245569586753845	87.64705657958984
Fold 2	0.280963659286499	87.64705657958984
Fold 3	0.23816584050655365	91.17646813392639
Fold 4	0.12058541923761368	95.88235020637512
Fold 5	0.2174811214208603	90.58823585510254
Fold 6	0.2489173263311386	92.9411768913269
Fold 7	0.31963616609573364	88.75739574432373
Fold 8	0.1773861050605774	93.49112510681152
Fold 9	0.3248450458049774	86.98225021362305
Fold 10	0.3700486123561859	86.98225021362305
Average	0.2660484991967678	90.2095365524292

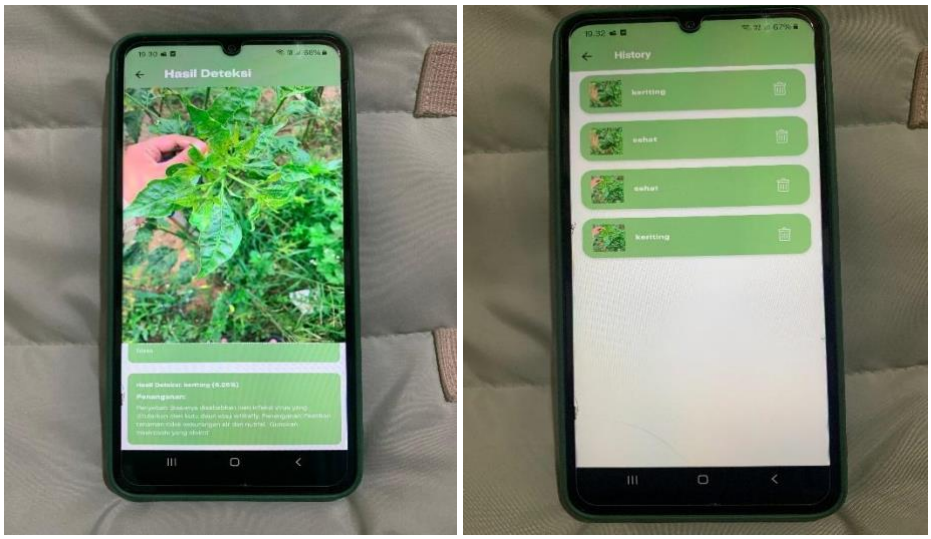

Fig 8. The Application Result.

To demonstrate real-world applicability, the proposed MobileNetV2-based model has been successfully integrated into an Android mobile application using the TensorFlow Lite framework. The app allows users, such as farmers or agricultural technicians, to capture leaf images using their smartphone cameras and receive immediate feedback on the disease classification result. Inference is performed locally on the device, making the system fully operational even in offline rural environments. This mobile-based implementation ensures that farmers can conduct on-site diagnoses without requiring specialized equipment or internet access. A limited user trial involving 32 respondents, consisting of farmers and field practitioners, yielded a positive response, with 84% of users expressing satisfaction with the application's usability, speed, and relevance to their field needs. Figure 8 illustrates the interface of the application, showing the image input, predicted class output, and a brief description of the identified disease. This integration confirms the feasibility of deploying the proposed model as a practical decision-support tool in precision agriculture.

IV. CONCLUSION

This study successfully developed a deep learning model based on the MobileNetV2 architecture for the classification of chili leaf diseases using image data. The model demonstrated strong performance with an overall accuracy of 91.04% and an average F1-score of 0.906, highlighting its robustness across multiple disease categories. Through the application of Stratified K-Fold Cross Validation, the evaluation showed consistent results with minimal variance across the folds, indicating that the model generalizes well to unseen data.

The integration of MobileNetV2, with its lightweight yet powerful design, proved particularly effective in maintaining a balance between computational efficiency and predictive accuracy, making it highly suitable for deployment in mobile and embedded agricultural diagnostic systems. Furthermore, the results support the validity of deep learning-based approaches for plant disease detection, reinforcing findings from previous studies while advancing the application of compact CNN architectures in precision agriculture.

Future research may explore further optimizations or the use of hybrid models to enhance classification performance, particularly for visually similar symptoms.

REFERENCES

[1] S. Mardiyati and M. Natsir, "Fluctuations and Trends in the Prices of Red Chilies and Cayenne Peppers in the Traditional Markets of Makassar City," *Iop Conf. Ser. Earth Environ. Sci.*, vol. 1302, no. 1, p. 12124, 2024, doi: 10.1088/1755-1315/1302/1/012124.

[2] F. Ahmad, K. Kusumiyati, M. A. Soleh, M. O. Khan, and R. S. Sundari, "Microclimates Growing and Watering Volumes Influences the Physiological Traits of Chili Pepper Cultivars in Combating Abiotic Stress," 2024, doi: 10.21203/rs.3.rs-4916999/v1.

[3] E. K. Asamani, B. K. Maalekuu, P. Kumah, and F. Appiah, "Effect of Varieties, Solar Drying and Zeer Pot Refrigeration Lining Media on Physico – Chemical Characteristics of Chili Pepper Fruits," *Eur. J. Nutr. Food Saf.*, vol. 16, no. 4, pp. 139–150, 2024, doi: 10.9734/ejnfs/2024/v16i41416.

[4] A. P. W. Wibowo, "Penerapan Teknik Computer Vision Pada Bidang Fitopatologi Untuk Diteksi Penyakit Dan Hama Tanaman Cabai," *J. Inform. J. Pengemb. It*, vol. 2, no. 2, pp. 102–108, 2017, doi: 10.30591/jpit.v2i2.528.

[5] E. A. Ukpoju, A. Adefemi, A. O. Adegbite, O. D. Balogun, B. O. Obaedo, and A. Abatan, "A Review of Sustainable Environmental Practices and Their Impact on U. S. Economic Sustainability," *World J. Adv. Res. Rev.*, vol. 21, no. 1, pp. 384–392, 2024, doi: 10.30574/wjarr.2024.21.1.2717.

[6] A. O. Adewusi, O. F. Asuzu, T. Olorunsogo, E. M. Adaga, and D. O. Daraojimba, "AI in Precision Agriculture: A Review of Technologies for Sustainable Farming Practices," *World J. Adv. Res. Rev.*, no. 1, pp. 2276–2285, 2024, doi: 10.30574/wjarr.2024.21.1.0314.

[7] N. B. Aji, Kurnianingsih, N. Masuyama, and Y. Nojima, "CNN-LSTM for Heartbeat Sound Classification," *Int. J. Informatics Vis.*, vol. 8, no. 2, pp. 735–741, 2024, doi: 10.62527/joiv.8.2.2115.

[8] N. B. Aji, L. Triyono, Kurnianingsih, W. Caesarendra, Mardiyono, and T. R. Yudantoro, "Health Protocol System: Face Mask Detection Using Deep Transfer Learning," *J. Eng. Sci. Technol.*, vol. 18, no. 4, pp. 17–30, 2023.

[9] D. Indrajaya, H. A. Parhusip, S. Trihandaru, and D. Hartanto, "MobileNetV2-D and Multiple Cameras for Swiftlet Nest Classification Based on Feather Intensity," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 34, no. 2, p. 1144, 2024, doi: 10.11591/ijeecs.v34.i2.pp1144-1158.

[10] E. C. Too, Y. Li, S. Njuki, and Y. Liu, "A Comparative Study of Fine-Tuning Deep Learning Models for Plant Disease Identification," *Comput. Electron. Agric.*, vol. 161, pp. 272–279, 2019, doi: 10.1016/j.compag.2018.03.032.

[11] K. P. Ferentinos, "Deep Learning Models for Plant Disease Detection and Diagnosis," *Comput. Electron. Agric.*, vol. 145, pp. 311–318, 2018, doi: 10.1016/j.compag.2018.01.009.

[12] W. Hu, Y. Huang, L. Wei, F. Zhang, and H.-C. Li, "Deep Convolutional Neural Networks for Hyperspectral Image Classification," *J. Sensors*, vol. 2015, pp. 1–12, 2015, doi: 10.1155/2015/258619.

[13] Y. Cui, C. Zhang, K. Qiao, L. Wang, B. Yan, and L. Tong, "Study on Representation Invariances of CNNs and Human Visual Information Processing Based on Data Augmentation," *Brain Sci.*, vol. 10, no. 9, p. 602, 2020, doi: 10.3390/brainsci10090602.

[14] F. Yang, "Enhancing Concrete Crack Image Detection Using MobileNetV2 and Transfer Learning," *Front. Sci. Eng.*, vol. 4, no. 4, pp. 110–119, 2024, doi: 10.54691/ynk4nf60.

[15] R. Gambheer and M. S. Bhat, "Optimized Compressed Sensing for IoT: Advanced Algorithms for Efficient Sparse Signal Reconstruction in Edge Devices," *Ieee Access*, vol. 12, pp. 63610–63617,

2024, doi: 10.1109/access.2024.3396494.

[16] I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis, "Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem," *Technologies*, vol. 9, no. 4, p. 81, 2021, doi: 10.3390/technologies9040081.

[17] I. Markoulidakis and G. Markoulidakis, "Probabilistic Confusion Matrix: A Novel Method for Machine Learning Algorithm Generalized Performance Analysis," *Technologies*, vol. 12, no. 7, p. 113, 2024, doi: 10.3390/technologies12070113.

[18] J. I. Aizpurua *et al.*, "Probabilistic Forecasting Informed Failure Prognostics Framework for Improved RUL Prediction Under Uncertainty: A Transformer Case Study," *Reliab. Eng. Syst. Saf.*, vol. 226, p. 108676, 2022, doi: 10.1016/j.ress.2022.108676.

[19] F. Shehzad, M. Islam, M. I. Omar, S. I. H. Shah, R. Ahmed, and N. Sohail, "Optimizations of Modified Machine Learning Algorithms Using K-Fold Cross Validations for Wheat Productivity: A Hyper Parametric Approach," *Sarhad J. Agric.*, vol. 38, no. 5, 2022, doi: 10.17582/journal.sja/2022/38.5.271.278.

[20] D. S. Soper, "Greed Is Good: Rapid Hyperparameter Optimization and Model Selection Using Greedy K-Fold Cross Validation," *Electronics*, vol. 10, no. 16, p. 1973, 2021, doi: 10.3390/electronics10161973.