

Lightweight Transfer Learning for Multi-Class Cabbage Leaf Disease Detection Using MobileNetV2

Ahmad Zidan Wirawan^{1*}, M. Syamsuddin Wisnubroto^{2*}, Fajri Farid^{3*}, Ade Lailani^{4*}, Rian Kurnia^{5*}

^{*}Data Science Study Program, Faculty of Science, Sumatera Institute of Technology, Lampung Selatan, Indonesia
ahmad.120450044@student.itera.ac.id¹, syamsuddin.wisnubroto@sd.itera.ac.id², fajri.farid@sd.itera.ac.id³, ade.lailani@sd.itera.ac.id⁴,
rian.kurnia@sd.itera.ac.id⁵

Article Info

Article history:

Received 2026-07-25

Revised 2026-09-19

Accepted 2026-10-03

Keyword:

*Cabbage Disease,
Convolutional Neural Network,
Deep Learning,
Image Classification,
MobileNetV2,
Smart Farming,
Transfer Learning.*

ABSTRACT

Accurate and deployable recognition of cabbage leaf diseases is important for reducing crop losses and supporting smart-farming decision support. This study develops a MobileNetV2-based transfer-learning framework for eight-class cabbage (*Brassica oleracea*) leaf condition classification using a public cabbage disease image dataset. The final experimental run used 1,201 training images and 299 evaluation images representing *Alternaria* leaf spot, bacterial spot rot, black rot, cabbage aphid colony, downy mildew, no disease, club root, and ring spot. All images were standardized to 224 x 224 pixels, normalized to a zero-to-one pixel range, and augmented during training using rotation, flipping, zoom variation, and brightness adjustment. MobileNetV2 with ImageNet initialization was used as a frozen feature extractor, followed by a compact softmax classification layer. The model contained 2,268,232 parameters, with only 10,248 trainable parameters and an estimated model footprint of 8.65 MB. Training used Adam optimization, categorical cross-entropy loss, a learning rate of 0.0001, mini-batches of 32 images, and 30 epochs. Evaluation produced a test accuracy of 89.30% and a test loss of 0.4959, while the confusion-matrix summary produced a diagonal accuracy of 87.63%. The class-wise analysis shows strong recognition of club root and ring spot, whereas bacterial spot rot remains the most challenging category. These results indicate that MobileNetV2 is a compact and promising baseline for cabbage leaf disease screening, although external field validation, measured inference time, memory benchmarking, and finalized browser deployment are still required before operational use.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. INTRODUCTION

Cabbage (*Brassica oleracea*) is a high-value horticultural commodity whose yield and marketability can be substantially affected by foliar diseases, insect-associated damage, and physiological disorders. In many production systems, disease recognition still relies on field inspection by farmers or extension officers. Although expert-based diagnosis remains indispensable, exclusive dependence on manual observation can delay treatment because early symptoms are often subtle, spatially heterogeneous, and visually confounded with nutrient stress or mechanical injury [1], [2], [3]. For small and medium farms, these delays may

translate into unnecessary pesticide application, reduced product quality, and avoidable economic losses.

Computer vision offers a non-destructive approach for converting leaf appearance into quantitative diagnostic evidence. Classical machine-learning methods generally depend on handcrafted color, texture, and shape descriptors; however, such features are sensitive to background clutter, illumination, camera distance, and symptom scale. Deep convolutional neural networks have improved this paradigm by learning hierarchical visual features directly from images and have demonstrated strong performance in image-based plant disease recognition[4], [5], [6], [7].

A persistent challenge is that high-performing deep networks are often computationally expensive. Architectures

such as VGG, ResNet, Inception, and larger EfficientNet variants may provide strong recognition capacity, but their model size and inference cost can be impractical for low-cost smartphones, browser-based tools, or rural edge devices. MobileNetV2 addresses this limitation through depthwise separable convolution, inverted residual connections, and linear bottlenecks, thereby reducing computational complexity while retaining useful representation capacity [8], [9].

Transfer learning further strengthens the feasibility of lightweight architectures. By initializing the feature extractor with ImageNet-pretrained weights, a model can exploit generic visual primitives such as edges, color transitions, lesion contours, and texture fragments before adapting them to crop-specific symptoms. This strategy is useful when agricultural datasets are relatively small or when class labels are expensive to obtain [10]. Data augmentation complements transfer learning by expanding the effective visual support of the training distribution; rotations, flips, zooming, and brightness perturbations encourage the model to learn disease-specific features rather than incidental acquisition conditions [11], [12].

The research gap addressed in this study is threefold. First, cabbage-specific multi-class disease recognition remains less explored than generic plant disease benchmarks or studies involving more common crops such as tomato. Second, many studies emphasize aggregate accuracy without providing sufficient class-wise analysis, model-size evidence, and deployment-oriented limitations. Third, lightweight deployment is often inferred from architecture selection alone, whereas practical evidence should include parameter count, model footprint, evaluation behavior, and a clear statement of what has and has not been validated in the deployment process.

The novelty of this study lies not in the isolated use of MobileNetV2, which is already established in visual classification, but in the integrated design of a cabbage-specific eight-class workflow. The contribution includes a transparent dataset description, reproducible preprocessing and augmentation settings, frozen-backbone transfer learning, quantitative compactness indicators, class-wise evaluation, error-pattern analysis, and a cautious discussion of web-oriented deployment readiness.

II. MATERIALS AND METHODS

Plant disease detection has become one of the most active applications of deep learning in precision agriculture. Reviews in this area emphasize that CNN-based approaches outperform classical pipelines when the imaging protocol, annotation quality, and class definitions are carefully controlled [6], [7]. However, these reviews also caution that inflated accuracy can occur when datasets are overly clean, when train-test leakage is not prevented, or when evaluation ignores field variability. Therefore, a rigorous plant disease classifier should be assessed not only by aggregate accuracy

but also by class-wise error behavior and deployment constraints.

MobileNet-family architectures represent a suitable compromise for agricultural applications because they replace standard convolutions with more economical operations. In a standard convolution, spatial and channel-wise filtering are computed jointly, whereas a depthwise separable convolution decomposes the operation into a depthwise spatial filter followed by a pointwise 1×1 projection. This factorization substantially reduces multiplication operations. MobileNetV2 improves the earlier MobileNet design by expanding low-dimensional features into a higher-dimensional space, applying depthwise convolution, and projecting the result through a linear bottleneck [8].

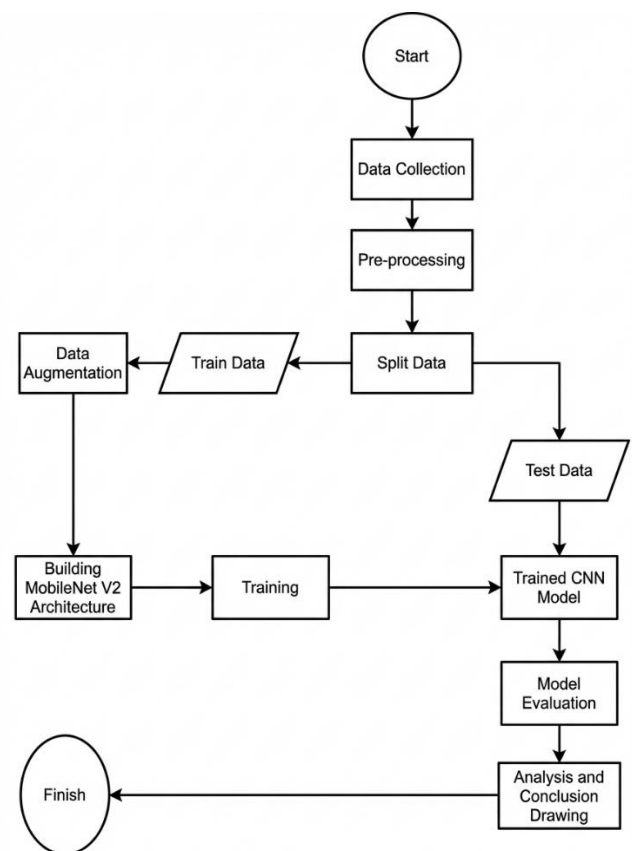


Figure 1. Research workflow for MobileNetV2-based cabbage leaf disease classification.

A. Research Design

This study employed a supervised deep-learning experiment. The workflow consisted of dataset acquisition, class identification, preprocessing, data partitioning, training-time augmentation, MobileNetV2-based transfer learning, model evaluation, lightweight-indicator analysis, and web-oriented deployment preparation. The final evaluation subset was not used to update the model weights, allowing the reported accuracy and class-wise metrics to reflect the behavior of the trained model on unseen images.

B. Dataset Source, Class Definition, and Image Conditions

The dataset was obtained from the public Balanced Cabbage Disease Dataset on Kaggle [14]. The images were not collected directly by the authors; therefore, the dataset is treated as a secondary public-image source. The final experimental folder contained eight categories: Alternaria leaf spot, bacterial spot rot, black rot, cabbage aphid colony, downy mildew, no disease, club root, and ring spot. The final run detected 1,201 training images and 299 evaluation images. Because the original public dataset does not provide complete metadata for individual leaf identity, camera type, geographic origin, or acquisition device, this study acknowledges the possibility of source bias and recommends external field validation before operational deployment.

TABLE I
VISUAL CHARACTERISTICS OF CABBAGE LEAF DISEASE CLASSES.

Class	Visual characteristics	Possible confusion
No disease	Apparently healthy leaf tissue without dominant lesion or colony pattern.	Early or mild infection may appear visually healthy.
Club root	Abnormal plant/leaf condition associated with root-related disease symptoms.	Above-ground appearance may be less specific when symptoms are weak.
Downy mildew	Pale discoloration, yellowing, and mildew-like surface appearance.	Can resemble nutrient stress or early leaf-spot symptoms.
Cabbage aphid colony	Clustered insect or colony pattern on the leaf surface.	Small colonies may resemble spot-like texture.
Ring spot	Circular or ring-like lesions with necrotic or discolored borders.	Overlaps with Alternaria leaf spot and other lesion diseases.
Black rot	Dark necrotic areas and vein-associated discoloration.	Can resemble severe bacterial or spot-rot patterns.
Bacterial spot rot	Small spots or localized tissue damage with rot-like appearance.	May resemble mild disease or weak healthy variation.
Alternaria leaf spot	Circular to irregular necrotic spots, often with visible lesion margins.	Frequently overlaps with ring spot and other spot-like symptoms.

The representative images show diverse visual conditions, including variation in leaf orientation, symptom extent, background, and lighting. However, the available metadata are not sufficient to confirm that all images fully represent uncontrolled field conditions. Consequently, the dataset is suitable for developing and testing a compact baseline model, but it should not be considered a complete substitute for real-farm validation.

TABLE II
EVALUATION DISTRIBUTION AND CORRECT PREDICTIONS DERIVED FROM THE RECORDED CONFUSION MATRIX.

Class	Declared public dataset	Evaluation support	Correct / Recall
Alternaria leaf spot	200	40	33 / 82.50%
Bacterial spot rot	200	22	15 / 68.18%
Black rot	200	40	36 / 90.00%
Cabbage aphid colony	200	40	33 / 82.50%
Downy mildew	200	39	32 / 82.05%
No disease	200	40	37 / 92.50%
Club root	200	38	38 / 100.00%
Ring spot	200	40	38 / 95.00%
Total	1,600	299	262 / 87.63%

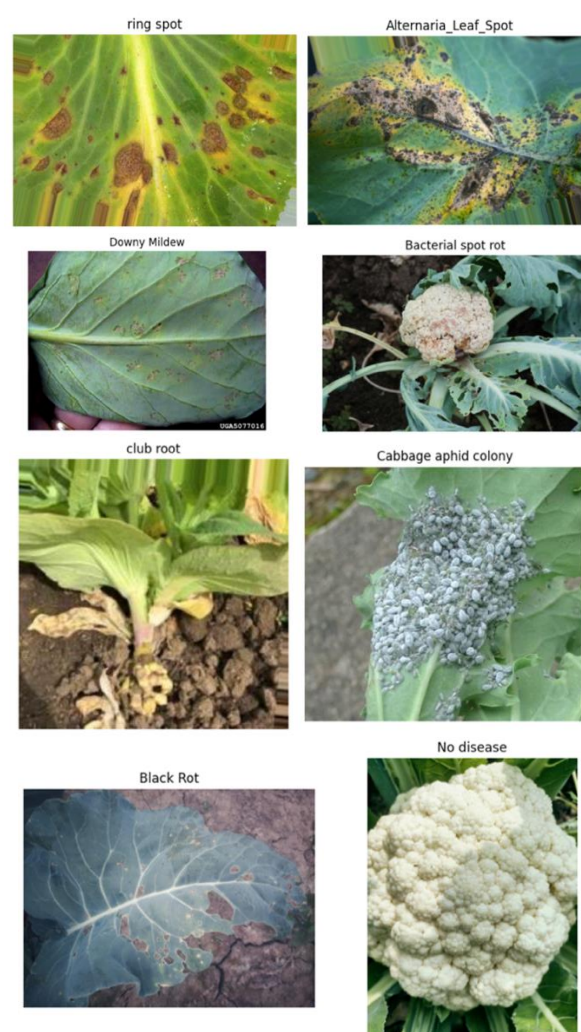


Figure 2. Representative cabbage leaf images across disease and healthy categories.

C. Preprocessing and Augmentation

All images were converted to RGB format, resized to 224 x 224 pixels, and normalized before model training. Pixel normalization used the following transformation for 8-bit RGB images:

$$x' = \frac{x}{255}, \quad x \in \{0,1,\dots,255\}, \quad x' \in \{0,1\}. \quad (1)$$

This preprocessing strategy was used consistently throughout the experiment. Because MobileNetV2 was initialized from ImageNet weights, future work may compare this simple zero-to-one normalization with the official MobileNetV2 preprocessing routine to determine whether a different input scaling strategy improves performance.

Augmentation was applied only to the training subset. The transformations included random rotation up to 30 degrees, zoom variation up to 20%, horizontal and vertical flipping, brightness adjustment between 0.8 and 1.2 times the original intensity, and nearest-neighbor filling for newly created boundary regions. These transformations were selected because disease symptoms should remain in the same biological class under moderate changes in orientation, scale, and illumination.

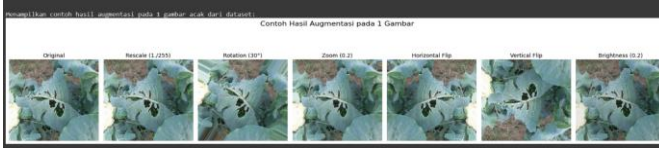


Figure 3. Example of training image augmentation under geometric and photometric transformations.

Training images were augmented using random rotations up to 30 degrees, horizontal and vertical flips, zoom perturbation up to 20%, and brightness adjustment. These transformations were applied only to the training subset. Their purpose was to increase the empirical support of the training distribution without altering the semantic class label. Figure 3 shows an example of the augmentation process.

D. MobileNetV2 Architecture and Transfer Learning Strategy

MobileNetV2 was adopted as the feature-extraction backbone using ImageNet initialization. The original top classifier was removed and replaced with a global average pooling layer followed by an eight-class softmax output layer. During the reported experiment, the MobileNetV2 backbone was frozen, and only the newly added classification layer was trained. No additional unfreezing stage was performed in the final experimental run; therefore, this manuscript describes the method as feature-extraction transfer learning rather than two-stage fine-tuning.

The ReLU6 activation used in MobileNetV2 can be expressed as:

$$\text{ReLU6}(x) = \min(\max(0, x), 6). \quad (2)$$

For a class logit z_k , the softmax probability for class k is given by

$$p_k = \frac{\exp(z_k)}{\sum_j \exp(z_j)}. \quad (3)$$

The softmax function transforms the final class scores into a probability distribution across the eight cabbage leaf conditions. Model training minimized categorical cross-entropy:

$$L = - \sum_k y_k \log p_k. \quad (4)$$

TABLE III
ADAPTED MOBILENETV2 CLASSIFICATION ARCHITECTURE.

Component	Description
Input	224 x 224 x 3 RGB image
Backbone	MobileNetV2 with ImageNet initialization
Backbone status	Frozen during the reported training run
Pooling	Global average pooling
Classifier	Eight-node softmax output layer
Transfer-learning strategy	Feature extraction only
Total parameters	2,268,232
Trainable / non-trainable parameters	10,248 / 2,257,984

E. Training Configuration, Hyperparameter Rationale, and Evaluation Metrics

The model was trained using Adam optimization with a learning rate of 0.0001, mini-batches of 32 images, 30 training epochs, and categorical cross-entropy loss. These values were selected as a conservative transfer-learning configuration rather than as the outcome of a full hyperparameter-search experiment. The learning rate was kept small to reduce abrupt updates to the newly initialized classifier, while the batch size was selected to balance optimization stability and available computational resources. The final training log reported training accuracy of 93.75%, training loss of 0.5888, validation accuracy of 89.93%, and validation loss of 0.4532.

TABLE IV
TRAINING CONFIGURATION.

Parameter	Value
Optimizer	Adam
Learning rate	0.0001
Batch size	32
Training epochs	30
Loss function	Categorical cross-entropy
Training images	1,201
Evaluation images	299
Training augmentation	Rotation, zoom, flipping, brightness adjustment

Model performance was assessed using accuracy, class-wise precision, recall, F1-score, macro average, weighted average, and confusion-matrix analysis. Accuracy is informative because the original public dataset was designed to be balanced, while class-wise metrics remain necessary

because visually similar diseases can produce asymmetric false-positive and false-negative behavior.

F. Quantitative Lightweight Indicators and Architecture-Level Comparison

The lightweight claim is supported by the reported model summary. The classifier contains 2,268,232 total parameters, with 10,248 trainable parameters and 2,257,984 frozen parameters. The estimated footprint is 8.65 MB. These indicators support the claim that the model is compact relative to larger CNN backbones. However, measured inference time, peak memory usage, and browser-device benchmarking were not available in the final experimental record and should be added in future validation.

TABLE V
QUANTITATIVE INDICATORS SUPPORTING THE LIGHTWEIGHT CLAIM.

Indicator	Value	Interpretation
Total parameters	2,268,232	Compact relative to larger CNN backbones
Trainable parameters	10,248	Only the classification head was trained
Non-trainable parameters	2,257,984	Frozen MobileNetV2 feature extractor
Estimated model footprint	8.65 MB	Supports compact model-size claim
Browser export status	Prepared but not fully benchmarked	Operational deployment still requires validation
Inference time and memory	Not measured	Must be evaluated on the target device or browser

TABLE VI
ARCHITECTURE-LEVEL COMPLEXITY COMPARISON.

Backbone	Approx. parameters	Approx. compute at 224 x 224	Deployment implication
MobileNetV2	~2.3M	~300M MACs	Selected compact architecture
MobileNetV1	~4.2M	~569M MACs	Lightweight but typically heavier than MobileNetV2
EfficientNetB0	~5.3M	~390M MACs	Compact but larger than MobileNetV2
ResNet50	~25.6M	~4.1B MACs	Powerful but substantially heavier
VGG16	~138M	~15.5B MACs	Generally unsuitable for low-resource web/mobile deployment

III. RESULTS AND DISCUSSION

A. Effect of Preprocessing and Augmentation

Preprocessing produced a uniform numerical representation for all images, while augmentation simulated plausible variations in field acquisition. These transformations regularized the feature extractor by discouraging memorization of fixed viewpoints and by introducing moderate photometric diversity. Because cabbage leaf diseases may appear under different orientations, scales, and lighting conditions, the applied transformations helped the model focus on disease-related visual structures rather than incidental acquisition conditions.

B. Classification Performance and Class-Wise Reporting

TABLE VII
SUMMARY OF REPORTED CLASSIFICATION PERFORMANCE.

Metric	Value
Test loss	0.4959
Test accuracy	0.8930 (89.30%)
Confusion-matrix diagonal accuracy	0.8763 (87.63%)
Macro precision	0.8796
Macro recall	0.8659
Macro F1-score	0.8698
Weighted F1-score	0.8754

The model achieved a test accuracy of 89.30% and a test loss of 0.4959 on the evaluation subset. The confusion-matrix summary produced 262 correct predictions from 299 samples, corresponding to a diagonal accuracy of 87.63%. The small difference between these two accuracy summaries should be interpreted conservatively because class-wise values depend on the final evaluation order used during prediction. For this reason, the confusion matrix is used primarily to analyze difficult classes and error directions, while the reported test accuracy is used as the main aggregate score.

TABLE VIII
CLASS-WISE PERFORMANCE DERIVED FROM THE RECORDED CONFUSION MATRIX.

Class	Precision	Recall	F1-score	Support
Alternaria leaf spot	0.7674	0.8250	0.7952	40
Bacterial spot rot	0.8824	0.6818	0.7692	22
Black rot	0.9231	0.9000	0.9114	40
Cabbage aphid colony	0.9706	0.8250	0.8919	40
Downy mildew	0.8205	0.8205	0.8205	39
No disease	0.8409	0.9250	0.8810	40
Club root	0.9268	1.0000	0.9620	38
Ring spot	0.9048	0.9500	0.9268	40
Macro average	0.8796	0.8659	0.8698	299
Weighted average	0.8793	0.8763	0.8754	299

C. Confusion-Matrix Interpretation and Error Mechanisms

The confusion matrix indicates that most predictions are concentrated along the principal diagonal, although several biologically interpretable confusions remain. Bacterial spot rot has the lowest recall because several images were assigned to the no-disease class, suggesting that mild spot-rot symptoms may visually resemble healthy or weakly affected leaves. *Alternaria* leaf spot is sometimes confused with downy mildew or club root, and downy mildew is sometimes confused with *Alternaria* leaf spot or ring spot. These patterns show that visually similar symptoms require careful false-positive and false-negative analysis.

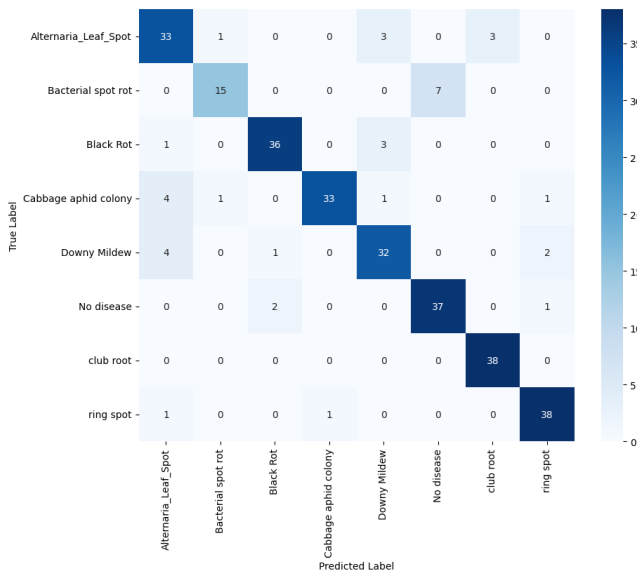


Figure 4. Numerically annotated confusion matrix of the MobileNetV2 cabbage leaf classifier.

TABLE IX
FALSE-POSITIVE AND FALSE-NEGATIVE MECHANISMS AND MITIGATION STRATEGIES.

Observed error pattern	Likely mechanism	Practical risk	Recommended mitigation
Bacterial spot rot predicted as no disease	Mild or localized symptoms resemble healthy tissue	False negative may delay treatment	Use confidence thresholds and collect more early-stage examples
Alternaria leaf spot confused with downy mildew or club root	Shared discoloration or lesion-like patterns	Incorrect follow-up recommendation	Add explainability maps, lesion segmentation, or top-k predictions
Downy mildew confused with Alternaria or ring spot	Spot-like and discoloration symptoms overlap	Cross-disease confusion under similar appearance	Use higher-resolution crops or patch-based classification

Observed error pattern	Likely mechanism	Practical risk	Recommended mitigation
Lighting-sensitive confusion	Brightness changes alter apparent symptom color	Model may learn illumination artifacts	Expand field-data augmentation and perform external validation

These observations indicate that the remaining errors are not random failures but biologically interpretable confusions. Such a pattern is typical in plant-image classification: labels are assigned at the image level, while symptoms may occupy only a small region of the leaf. If the diseased region is small, blurred, overexposed, or partially occluded, the global image representation may be dominated by healthy tissue. Future work could therefore combine MobileNetV2 with attention maps, lesion segmentation, or Grad-CAM-based interpretability to verify whether the model focuses on agronomically meaningful regions [15], [16].

D. Training Dynamics and Generalization Behavior

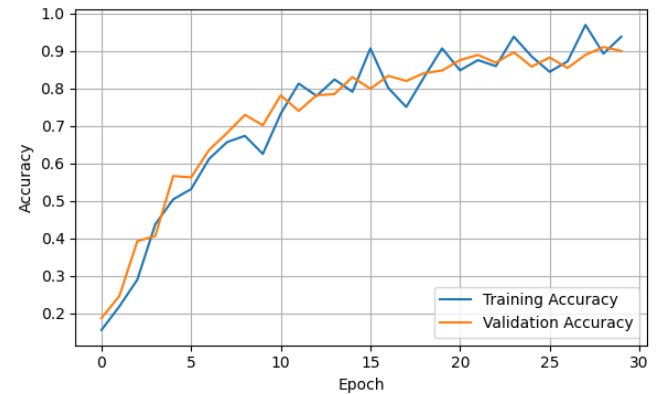


Figure 5. Training and validation accuracy curves.

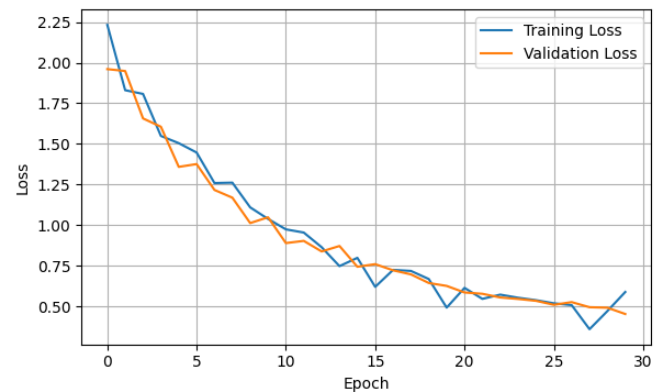


Figure 6. Training and validation loss curves.

The learning curves show improving optimization across 30 epochs. Training and validation accuracy increased rapidly in the early epochs and then stabilized, while the loss curves generally decreased. The final epoch reported training accuracy of 93.75% and validation accuracy of 89.93%, with validation loss of 0.4532. This behavior indicates that ImageNet initialization supported efficient feature learning,

while augmentation helped reduce overfitting. Nevertheless, the remaining gap between training and validation performance suggests that additional field data and external validation are still necessary.

E. Deployment-Oriented Analysis

The deployment-oriented component is presented as a prototype pathway rather than as a fully validated operational system. The trained model can be saved and prepared for a web-compatible representation, after which a web interface can receive a cabbage leaf image, apply the required preprocessing, produce a class-probability vector, and display the most likely class with its confidence value. This formulation avoids treating the web prototype as a completed field-ready system while still explaining its practical direction. The web-oriented prototype was prepared using a Keras-trained model intended for TensorFlow.js-compatible browser inference; however, browser-side benchmarking was not completed in the present experiment.

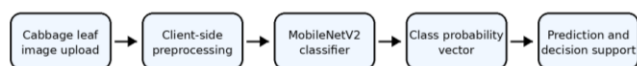


Figure 7. Proposed lightweight web-oriented inference workflow.

The proposed interface should display not only the top predicted class but also the associated probability and an uncertainty warning when the confidence is low. Such design would reduce the risk of over-trusting the classifier and would position the system as decision support rather than an automated prescription mechanism.

Klasifikasi Penyakit Tanaman Menggunakan MobileNet

Upload gambar daun untuk mengetahui hasil prediksi

Pilih gambar daun...

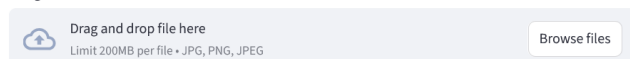


Figure 8. Web-based cabbage leaf image upload interface.



Gambar yang diupload

Hasil Prediksi:

Ring Spot (15.16%)

Figure 9. Example of prediction output generated by the web prototype.

The web screenshots document the intended path from offline model preparation to lightweight inference. They should be interpreted as workflow demonstrations rather than as evidence of verified deployment reliability. Before claiming operational web deployment, browser-side inference time, memory use, prediction consistency, and cross-device performance must be measured.

F. Comparative Positioning with Prior Work

The result aligns with prior evidence that CNN-based approaches can automate plant disease [4], [5], [6], [7]. This study differs by focusing on cabbage-specific categories, using a compact frozen MobileNetV2 backbone, and connecting model evaluation with deployment preparation. A larger model may improve accuracy under some conditions, but that improvement may not justify additional computational cost for farmer-facing systems. The current evidence supports compactness through parameter count and model footprint, while empirical baseline accuracy and device-level inference benchmarks remain future work.

For farmers and extension workers, the practical value of the proposed system lies in rapid triage. A recommended workflow is to capture the image under adequate lighting, submit it to the classifier, inspect the predicted class and confidence score, compare the symptoms with agronomic guidance, and seek expert confirmation for low-confidence or high-risk cases. This protocol reduces the risk of treating model output as a final diagnosis while still benefiting from automated screening.

G. Limitations and Threats to Validity

Several limitations must be considered. First, the final experimental folder contained 1,201 training images and 299 evaluation images rather than the nominal 1,600-image balance stated by the dataset title. Second, public-dataset

images may differ from actual farm images captured with low-cost mobile phones. Third, the metadata do not fully identify individual leaves, plants, locations, or camera devices; therefore, source bias and visually related samples across subsets cannot be fully excluded. Fourth, the confusion matrix should be regenerated in a final publication run using a fixed evaluation order and a non-augmented evaluation subset. Fifth, the browser-oriented export has not yet been validated through measured browser inference time, memory use, or cross-device consistency. Sixth, empirical comparison with MobileNetV1, EfficientNet-Lite, MobileNetV3, ShuffleNet, ResNet, and a simple CNN baseline is still required.

Future work should validate the model on independently collected field images, compare several compact backbones under the same data split, measure inference time on smartphones and browsers, add Grad-CAM or segmentation-based explainability, complete browser deployment validation, and integrate the classifier with smartphone cameras or IoT-based farm monitoring workflows.

IV. CONCLUSION

This study presented a lightweight transfer-learning framework for multi-class cabbage leaf disease detection using MobileNetV2. The revised manuscript clarifies the research gap, dataset source, disease categories, preprocessing, augmentation parameters, ImageNet initialization, model architecture, class-wise reporting, and deployment limitations. The final experimental run reported a test loss of 0.4959 and a test accuracy of 89.30% on 299 evaluation images. The model summary reported 2,268,232 total parameters and an estimated footprint of 8.65 MB, with only 10,248 trainable parameters because the MobileNetV2 backbone was frozen.

The main implication is that MobileNetV2 can provide a compact and reasonably accurate baseline for smart-farming screening. Nevertheless, the model should be treated as an assistive tool rather than a replacement for expert agronomic diagnosis. Future research should use a cleaner three-way split, fixed-order evaluation, non-augmented test assessment, external field validation, measured inference time and memory use, explainable AI, and validated browser deployment.

ACKNOWLEDGMENT

The authors would like to thank the Data Science Study Program, Faculty of Science, Sumatera Institute of Technology, for academic support during the preparation of this study. The authors also acknowledge the public dataset provider and open-source machine-learning communities that made the experimental work reproducible.

REFERENCES

- [1] Direktorat Jenderal Hortikultura, Pusat Data dan Sistem Informasi Pertanian, and H. dan P. B. P. S. Direktorat Statistik Tanaman Pangan, *Pedoman Statistik Pertanian Hortikultura (SPH)*, Revisi I. Jakarta: Direktorat Jenderal Hortikultura, Kementerian Pertanian, 2022. Accessed: Mar. 16, 2026. [Online]. Available: <https://hortikultura.pertanian.go.id/publikasi/>
- [2] G. N. Agrios, *Plant pathology*. Elsevier, 2005.
- [3] J. G. A. Barbedo, "A review on the main challenges in automatic plant disease identification based on visible range images," *Biosyst. Eng.*, vol. 144, pp. 52–60, 2016.
- [4] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Front. Plant Sci.*, vol. 7, p. 1419, 2016.
- [5] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, pp. 311–318, 2018.
- [6] A. Ahmad, D. Saraswat, and A. El Gamal, "A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools," *Smart Agricultural Technology*, vol. 3, p. 100083, 2023.
- [7] C. K. Sunil, C. D. Jaidhar, and N. Patil, "Systematic study on deep learning-based plant disease detection or classification.," *Artif. Intell. Rev.*, vol. 56, no. 12, p. 14955, 2023.
- [8] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.
- [9] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
- [10] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2009.
- [11] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.
- [12] R. Takahashi, T. Matsubara, and K. Uehara, "Ricap: Random image cropping and patching data augmentation for deep cnns," in *Asian conference on machine learning*, PMLR, 2018, pp. 786–798.
- [13] M. C. Untoro, N. Triawati, Y. A. Effendi, H. Natalia, and M. S. Wisnubroto, "Comparison of K-means and K-medoids algorithm in grouping dengue fever patient data (Case study: Kaliasin health center)," in *Proceedings Of The International Conference On Advanced Technology And Multidiscipline (ICATAM) 2021: "Advanced Technology and Multidisciplinary Prospective Towards Bright Future" Faculty of Advanced Technology and Multidiscipline*, AIP Publishing LLC, 2023, p. 020009.
- [14] Kaggle and Mubbassir, "Balanced Cabbage Disease Dataset (200 each)," Kaggle. Accessed: Jan. 16, 2026. [Online]. Available: <https://www.kaggle.com/datasets/mubbassir/balanced-cabbage-dataset-200-each>
- [15] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: visual explanations from deep networks via gradient-based localization," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 336–359, 2020.
- [16] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4, no. 4. Springer, 2006.
- [17] D. Smilkov *et al.*, "Tensorflow. js: Machine learning for the web and beyond," *Proceedings of machine learning and systems*, vol. 1, pp. 309–321, 2019.
- [18] F. Chollet and F. Chollet, *Deep learning with Python*. simon and schuster, 2021.
- [19] H.-A. Goh, C.-K. Ho, and F. S. Abas, "Front-end deep learning web apps development and deployment: a review," *Applied intelligence*, vol. 53, no. 12, pp. 15923–15945, 2023.