IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511135

AI-Based Plant Disease Detection System for Chili Crops Using Deep Transfer Learning: Comparison of MobileNetV2 and EfficientNetB0 with a Web-Based Advisory Deployment

Priyanka D. Lanjewar[1], Jinendra G. Gavit[2], Sudarshan B. Patil[3]*, Uday P. Patil[4], Ganesh K. Pawar[5]

Assistant Professor, R. C. Patel Institute of Technology, Shirpur, Dist. Dhule, Maharashtra, India – 425405[1]

UG Students, Department of Computer Science and Engineering (Data Science)

R. C. Patel Institute of Technology, Shirpur, Dist. Dhule, Maharashtra, India – 425405[2-5]

*Corresponding Author

Abstract: Smallholder chili growers often struggle to tell leaf diseases from nutrient disorders, and misdiagnosis costs them both yield and fruit quality. This paper presents an AI-based system that sorts leaf images into six categories (Bacterial Spot, Cercospora Leaf Spot, Leaf Curl Virus, Healthy Leaf, Nutrient Deficiency and White Spot) and recognises five fruit growth stages. We compared two lightweight networks, MobileNetV2 and EfficientNetB0, on a balanced dataset of 12,000 images, using a stratified 70/15/15 split, on-the-fly augmentation and a two-phase training schedule. On the 1,800-image test set, MobileNetV2 reached 98.33% accuracy with a macro F1-score of 0.983 and processed an image in 0.42 seconds, while EfficientNetB0 reached 89.42%. The growth-stage model achieved 95.3% accuracy on its own test set. Both models are served through a FastAPI inference service connected to a Spring Boot API, with JWT authentication, a MySQL database and a React portal that returns the diagnosis, a confidence score and treatment guidance. Under the protocol used here, MobileNetV2 appears to be the more suitable choice for deployment among farmers, although the system needs further validation before it is used in the field.

Keywords: chili disease detection, deep learning, transfer learning, MobileNetV2, EfficientNetB0, Grad-CAM, precision agriculture, web-based decision support

I. INTRODUCTION

Chili (Capsicum annuum L.) is both a spice and a vegetable crop, grown by millions of smallholders worldwide; India is among its leading producers, consumers and exporters. The crop is under constant pressure from bacteria, fungi, viruses and pests, and the first signs usually appear on the leaves as spots, curling, yellowing or powdery patches. Spotting these symptoms early allows treatment to be targeted, which protects both yield and fruit quality. When they are misread or missed, farmers tend to fall back on broad-spectrum pesticides, which raises costs and leaves more chemical residue in the environment.

In many rural areas, diagnosis depends on what the grower, a dealer or an occasional extension worker can judge by eye. Such judgements vary from person to person, and help often arrives late. Similar symptoms make matters worse: nutrient deficiencies and viral infections can both turn leaves yellow, and different leaf-spot diseases may leave lesions that look almost identical. Automated tools that work from a smartphone photograph could shorten the path from noticing a symptom to acting on it.

Convolutional neural networks (CNNs) now dominate image recognition, a shift that followed the influential ImageNet studies [2]. Training them from scratch, however, requires large labelled datasets and more computing power than most agricultural projects can afford. Transfer learning [3] offers a way around this by reusing features already learned from a larger dataset. For tools aimed at farmers, lightweight architectures such as MobileNetV2 and EfficientNetB0 are especially attractive because they respond quickly even on modest hardware.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511135

This work builds and evaluates such a service for chili. Its main contributions are:

- a six-class chili leaf classifier based on transfer learning, evaluated on an independent, class-balanced test set of 1,800 images with per-class precision, recall and F1-score analysis;
- a controlled comparison of MobileNetV2 and EfficientNetB0 trained with identical data handling, augmentation and optimiser settings;
- a companion classifier for five fruit growth stages that uses the same serving pipeline; and
- an end-to-end farmer portal (React, Spring Boot, FastAPI, MySQL) that returns the predicted class, a confidence score and structured treatment advice, along with a discussion of the checks needed before field release.

II. RELATED WORK

Mohanty *et al.* [4] helped popularise deep learning for plant disease recognition. They trained AlexNet and GoogLeNet on the PlantVillage images [5] and reported 99.35% accuracy across 26 disease–crop classes, but also found that accuracy dropped sharply on photographs taken outside the laboratory. Ferentinos [6] later reported success rates above 99% on a much larger collection of leaf images spanning 58 classes. Barbedo [7] examined how dataset size and variety affect transfer-learning results and concluded that diverse, representative images matter at least as much as the choice of network. These findings explain why this study pays close attention to augmentation and is candid about what its test set does and does not represent.

Work on chili specifically is more limited; Table I lists the studies most relevant to this paper. Classical pipelines built on handcrafted texture descriptors, such as the grey-level co-occurrence matrix [8], require manual feature design and cope poorly with cluttered backgrounds. The region-based network of Raja *et al.* [9] reached 96.75% accuracy but was reported to be slow. Vasavi *et al.* [1] fine-tuned several ImageNet models and found EfficientNetB0 the most accurate, although their data were laboratory images covering only two classes. The closest prior work is that of Keerthana and Khan [10], who classified six chili conditions with a MobileNetV2 model, added Grad-CAM explanations and exposed the model through a FastAPI web application.

TABLE I. RELATED STUDIES ON CHILI LEAF DISEASE DETECTION

Ref.	Data	Method	Reported result
[9]	Chili leaf images	GLCM texture features + region-based CNN	96.75% accuracy; long processing time
[1]	PlantVillage pepper subset, 2,475 images, 2 classes	Transfer learning: MobileNet, Inception-ResNetV2, EfficientNetB0, DenseNet	0.995 accuracy (EfficientNetB0)
[10]	12,000 chili leaf images, 6 classes	MobileNetV2 + Grad-CAM, FastAPI web app	98.56% test accuracy; 0.42 s per image
This work	12,000 chili leaf images, 6 classes; 5 growth stages	MobileNetV2 vs EfficientNetB0, two-phase training, authenticated full-stack portal	See Section VI

This study differs from [10] in four ways. It compares two backbones under a single protocol rather than presenting one model; it reports class-level precision, recall and F1-score derived from the full confusion matrix; it adds a growth-stage model; and it deploys the models inside an authenticated portal that stores prediction history and returns structured treatment guidance.

III. MATERIALS AND METHODS

A. Dataset

The experiments use 12,000 RGB images of chili leaves, organised into six folders: Bacterial Spot (BS), Cercospora Leaf Spot (CLS), Leaf Curl Virus (CV), Healthy Leaf (HL), Nutrient Deficiency (ND) and White Spot (WS). Each class contains exactly 2,000 images, so the dataset is perfectly balanced. The source photographs were captured at 1000×1000 pixels. M. A. S. Nirob, A. K. M. F. K. Siam, P. Bishshash, and M. Assaduzzaman, "Chili Plant Leaf Disease and Growth Stage Dataset from Bangladesh," Mendeley Data, V1, 2025. DOI: 10.17632/w9mr3vf56s.1. Before any further processing, an automated script opened every file and removed unreadable, corrupted or blank images. A separate collection of fruit images labelled Dry, Flower, Green, Red and Rotten chili was used for the growth-stage model.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511135

B. Pre-processing and Data Split

Each image was converted to RGB, resized to 224×224 pixels by bilinear interpolation to match the input layer of both backbones, stored as float32 and rescaled to [0, 1]. Labels were one-hot encoded. The data were split by stratified sampling so that every subset keeps the class proportions (Table II). The validation set guides early stopping, and the test set is used only for the final evaluation.

TABLE II. DATASET PARTITION

Subset	Images	Share	Per class
Training	8,394	69.95%	≈1,399
Validation	1,806	15.05%	≈301
Test	1,800	15.00%	300
Total	12,000	100%	2,000

C. Data Augmentation

To reduce over-fitting and mimic the variation found in the field, augmentation was applied on the fly to training batches only. It consisted of random horizontal flips, rotations of up to ±15°, random zoom and brightness changes. Validation and test images were never augmented, so the reported scores remain comparable between runs.

D. Network Architectures

Both backbones start from ImageNet weights with their original classification layers removed. **MobileNetV2** [11] is built from inverted residual blocks with linear bottlenecks and depthwise separable convolutions, which reduce the number of operations and parameters [12]. **EfficientNetB0** [13] is the baseline model of a family that scales depth, width and resolution together through a fixed compound coefficient; its MBConv blocks also include squeeze-and-excitation attention. To isolate the effect of the backbone, each feature extractor is followed by the same head: global average pooling, a 128-unit dense layer with ReLU, dropout, and a 6-unit softmax layer.

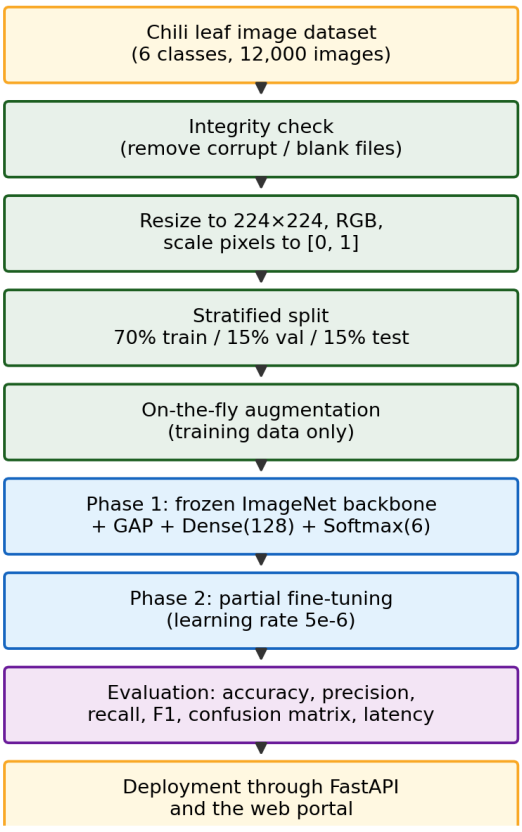


Fig. 1. Overall methodology, from data preparation to deployment.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511135

E. Two-Phase Training

In Phase 1 the backbone was frozen and only the new head was trained, using the Adam optimiser [14] at a learning rate of 1×10^{-4}. In Phase 2 the upper convolutional blocks were unfrozen and the network was fine-tuned at a much lower learning rate of 5×10^{-6}, which adapts the pretrained features to chili lesions gradually instead of overwriting them. Training used categorical cross-entropy, a batch size of 32, a maximum of 50 epochs, learning-rate reduction on plateau, and early stopping with a patience of five epochs. Class weighting was unnecessary because the classes are balanced, but the per-class loss was still monitored to confirm that each class contributed evenly to the gradients.

F. Explainability

Gradient-weighted class activation mapping (Grad-CAM) [15] produces a heat map of the leaf regions that drove a prediction. The map is blended with the input photograph, so a user or an agronomist can check whether the model attended to the lesion rather than to the background.

G. Evaluation Metrics

Performance is reported as overall accuracy, per-class precision, recall and F1-score, macro averages, a confusion matrix and per-image processing time, where precision, recall and F1-score are defined as:

$$Precision = TP / (TP + FP), \quad Recall = TP / (TP + FN),$$

$$F1 = 2 \times Precision \times Recall / (Precision + Recall).$$

IV. SYSTEM IMPLEMENTATION

The trained models are integrated into a three-tier web application (Fig. 2). Through the **React** front end, a farmer can register, sign in, upload a leaf or fruit photograph, view the result and browse earlier predictions. The **Spring Boot** REST API (Java 21) manages users, image upload and validation, and storage. It protects its endpoints with JSON Web Token authentication, hashes passwords with BCrypt and separates farmer and administrator roles. Users, image records and prediction history are stored in a **MySQL** database. The API forwards each validated image to a **FastAPI** inference service. That service applies the same 224×224, [0, 1] pre-processing used in training, runs the TensorFlow/Keras disease and growth-stage models, generates the Grad-CAM map and returns the class labels with their confidence scores.

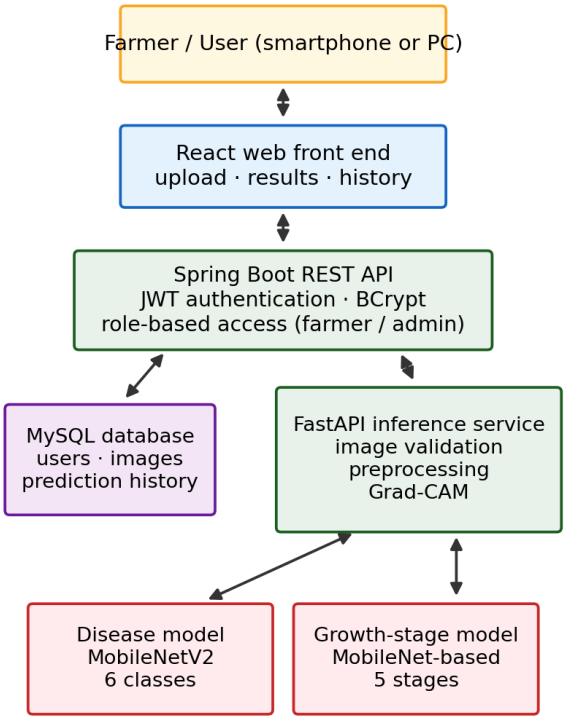


Fig. 2. Architecture of the farmer portal and inference pipeline.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511135

The result page (Fig. 3) displays the uploaded photograph, the predicted disease with its confidence, the predicted growth stage and an advisory card. The card describes the condition, its cause and symptoms, and lists chemical, organic and preventive measures. The advisory text is a fixed knowledge base keyed to the predicted class.

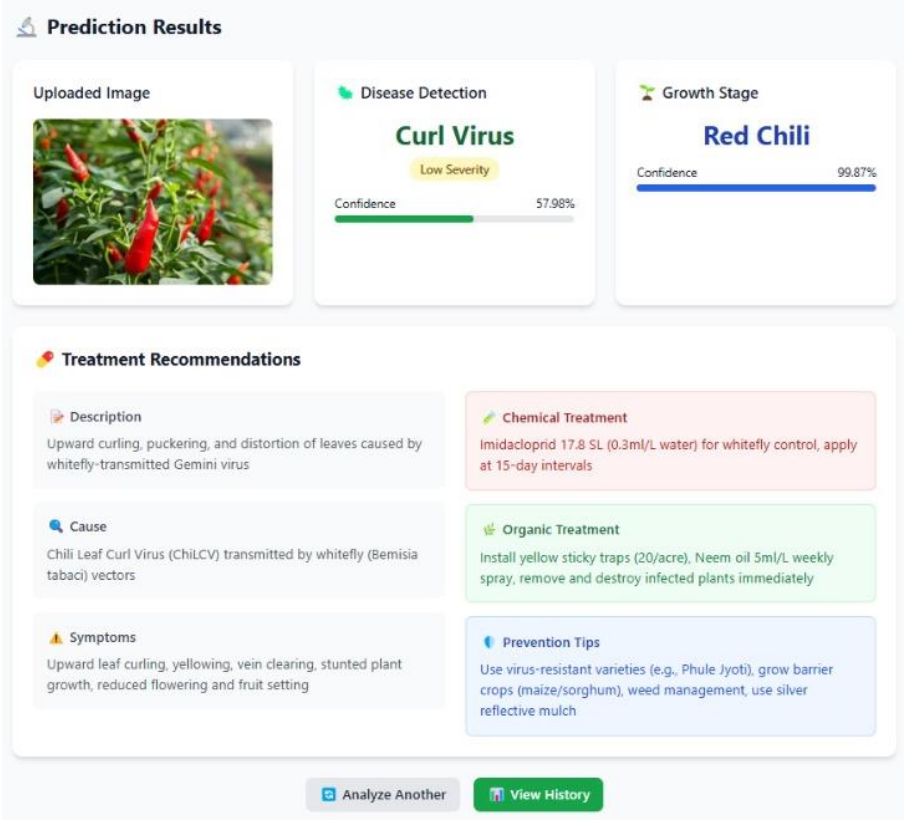


Fig. 3. Result page of the portal for a sample photograph.

V. EXPERIMENTAL SETUP

All models were implemented in Python 3.10 with TensorFlow 2.15 [16], Keras, scikit-learn and OpenCV, and trained in a GPU-accelerated cloud notebook, with checkpoints saved to cloud storage. Both backbones received identical data partitions, pre-processing, augmentation and optimiser settings (Table III).

TABLE III. TRAINING CONFIGURATION

Setting	Value
Input size	224 × 224 × 3
Batch size	32
Optimiser	Adam
Learning rate, Phase 1	1×10^{-4} (frozen backbone)
Learning rate, Phase 2	5×10^{-6} (fine-tuning)
Maximum epochs	50
Early stopping	Patience of 5 epochs
Loss	Categorical cross-entropy
Head	GAP, Dense(128, ReLU), Dropout, Softmax(6)

VI. RESULTS AND DISCUSSION

A. Disease Classification with MobileNetV2

Early stopping ended the fine-tuned MobileNetV2 run after 23 epochs, with a validation accuracy of 98.39%. On the 1,800 held-out test images the model classified 1,770 correctly, an overall accuracy of 98.33%. Table IV gives the class-

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511135

wise results computed from the confusion matrix in Fig. 4. Precision, recall and F1-score are at least 0.97 for every class, and the macro-averaged values are 0.983, 0.983 and 0.983.

TABLE IV. PER-CLASS RESULTS ON THE TEST SET (MOBILENETV2, 300 IMAGES PER CLASS)

Class	Precision	Recall	F1
Bacterial Spot	0.980	0.980	0.980
Cercospora Leaf Spot	0.980	0.983	0.982
Leaf Curl Virus	0.983	0.970	0.977
Healthy Leaf	0.990	0.993	0.992
Nutrient Deficiency	0.977	0.977	0.977
White Spot	0.990	0.997	0.993
Macro average	0.983	0.983	0.983

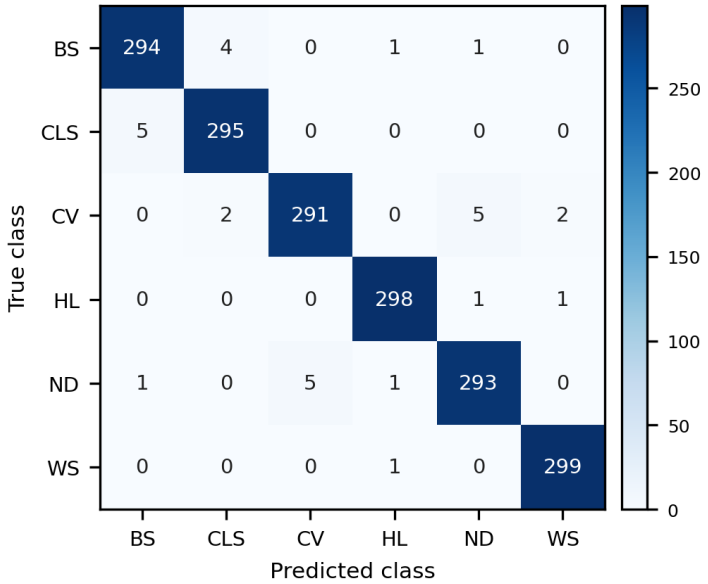


Fig. 4. Confusion matrix of MobileNetV2 on the 1,800-image test set (BS: Bacterial Spot, CLS: Cercospora Leaf Spot, CV: Leaf Curl Virus, HL: Healthy Leaf, ND: Nutrient Deficiency, WS: White Spot).

Healthy leaves were recognised with 99.33% recall and white spot with 99.67%. The result for healthy leaves matters in practice, since a healthy leaf misread as diseased leads to unnecessary spraying. Leaf Curl Virus had the lowest recall (97.0%): five of its 300 test images were labelled Nutrient Deficiency and two White Spot. The reverse confusion also occurs, with five Nutrient Deficiency images predicted as Leaf Curl Virus. Both conditions can cause yellowing and distortion, which makes this pair the main source of error. The only other notable confusion is between the two leaf-spot diseases: four Bacterial Spot images were read as Cercospora and five Cercospora images as Bacterial Spot, which fits their similar lesion shapes.

B. Comparison of the Two Backbones

Table V compares the two networks under the shared protocol. MobileNetV2 outperformed EfficientNetB0 by roughly nine percentage points, but this gap should be read as specific to the protocol rather than as a general ranking of the architectures. The Keras EfficientNet implementation includes its own input normalisation and expects raw pixel values, whereas both networks here received images scaled to [0, 1]. A mismatch of this kind can prevent the pretrained EfficientNet weights from being fully exploited and is a likely contributor to the slow convergence. Because the aim was a strictly identical input pipeline, no architecture-specific adjustment was made; re-running both networks with their native preprocessing is the fair next test.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511135

TABLE V. BACKBONE COMPARISON (IDENTICAL PROTOCOL)

Model	Test accuracy	Time per image
MobileNetV2	98.33%	0.42 s
EfficientNetB0	89.42%	0.61 s

C. Processing Time

The end-to-end time for one MobileNetV2 prediction was about 0.42 s: 0.08 s for pre-processing, 0.25 s for model inference and 0.09 s for Grad-CAM rendering. Training and latency benchmarks were evaluated on a local ASUS TUF Gaming A15 laptop with an AMD Ryzen 7 processor, 16 GB RAM and an NVIDIA GeForce RTX GPU, running Python 3.10 and TensorFlow. A sub-second response is adequate for interactive use in the portal.

D. Growth-Stage Classification

The companion model separates five fruit stages. On its test partition of 341 images it classified 325 correctly (95.3%). Dry chili, flower and red chili were recognised without error, whereas green chili (58 of 65) and rotten chili (66 of 75) were weaker. Seven rotten fruits were labelled green and five green fruits flower, which is consistent with the colour overlap between those stages (Fig. 5).

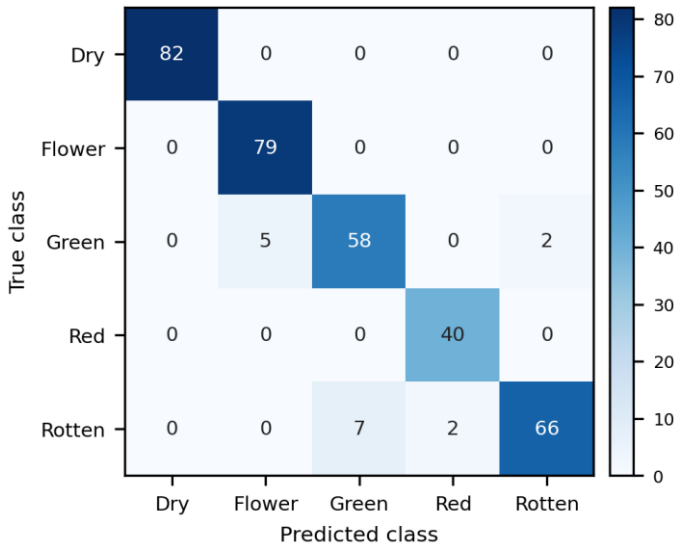


Fig. 5. Confusion matrix of the growth-stage model (341 test images).

E. Behaviour in the Deployed Portal

The sample in Fig. 3 shows how the two outputs appear together. The growth stage was returned with a confidence of 99.87%, whereas the disease prediction (Leaf Curl Virus) carried only 57.98%. A score this modest should not be treated as a firm diagnosis. The portal therefore displays the confidence next to every result, and the recommended safeguard is a minimum-confidence threshold below which users are advised to consult an agronomist before applying any chemical.

VII. LIMITATIONS

- **Evaluation data.** The reported scores come from a single stratified split and a single training run, with no confidence intervals. If augmented copies of the same original photograph were assigned to different subsets before splitting, the test accuracy would be optimistic. The split should therefore be verified at the level of original photographs and, if necessary, repeated.
- **Field conditions.** No separate set of unseen, farm-captured photographs has been used so far. Strong shadows, mixed backgrounds, partial occlusion and different phone cameras may lower accuracy, as Mohanty *et al.* [4] and Barbedo [7] observed for other crops.
- **Baseline fairness.** The EfficientNetB0 result reflects the shared [0, 1] scaling discussed in Section VI-B.
- **Advisory content.** The treatment texts are static and have not been reviewed by a plant-protection specialist. Dosages must be validated locally before the system is released.
- **Scope.** Only six leaf conditions and five fruit stages are covered. Co-occurring diseases and other crops lie outside the current model.

DOI: 10.17148/IJARCCE.2026.1511135

VIII. CONCLUSION

This paper presented an AI-based plant disease detection system for chili crops. A MobileNetV2 model fine-tuned with a two-phase transfer-learning strategy reached 98.33% accuracy and a macro F1-score of 0.983 on a balanced six-class test set, at 0.42 s per image, and outperformed EfficientNetB0 under the same input pipeline. A companion model recognised five growth stages with 95.3% accuracy. Combined with an authenticated web portal that offers confidence scores, Grad-CAM explanations and structured treatment advice, these models form a practical decision-support tool for growers.

Future work will (i) collect and expert-label farm photographs taken under natural lighting and test the models on them, (ii) repeat the backbone comparison with each network's native preprocessing and with several random seeds, (iii) add attention modules and calibrated confidence thresholds, (iv) have agronomists review the advisory content, and (v) provide an offline mobile build and low-bandwidth access for farmers without reliable internet.

ACKNOWLEDGMENT

The authors are grateful to Prof. P. D. Lanjewar and the faculty of R. C. Patel Institute of Technology for their steady support and guidance throughout this project. They also thank the team members and technical staff whose work was essential to developing and testing the system.

REFERENCES

[1]. P. Vasavi, A. Punitha, and T. V. N. Rao, "Image based chili crop disease prediction using deep transfer learning," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 11, no. 10, pp. 1145–1149, 2023.

[2]. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst.*, vol. 25, 2012.

[3]. S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.

[4]. S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Front. Plant Sci.*, vol. 7, art. 1419, 2016.

[5]. D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," arXiv:1511.08060, 2015.

[6]. K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, pp. 311–318, 2018.

[7]. J. G. A. Barbedo, "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification," *Comput. Electron. Agric.*, vol. 153, pp. 46–53, 2018.

[8]. R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, 1973.

[9]. K. Raja, J. Shiny Duela, M. Gopichand, K. Kannan, and M. G. Sathish, "Chilli leaf diseases detection with different features of original chilli using region based convolutional neural network," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 3s, pp. 298–305, 2024.

[10]. B. Keerthana and Z. A. Khan N., "Deep learning-based CNN approach for disease detection in chili plant leaves," *Int. J. Eng. Res. Technol.*, vol. 14, no. 5, paper IJERTV14IS050145, May 2025.

[11]. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

[12]. A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv:1704.04861, 2017.

[13]. M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[14]. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015.

[15]. R. R. Selvaraju *et al.*, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 618–626.

[16]. M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. USENIX Symp. Oper. Syst. Des. Implement.*, 2016, pp. 265–283.

 This work is licensed under a Creative Commons Attribution 4.0 International License