Pradeep, 2026, 14:6
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

LeafGuard: A Deep Learning System for Plant Leaf Disease Detection with Organic Treatment Recommendations

Pradeep[1], Dr.Mohit Sharma[2]

PhD, Research Scholar, Dept. of Computer Science and Engineering
NIILM university Kaithal, Haryana, India[1]

Assistant Professor, Dept. of Computer Science and Engineering, NIILM university Kaithal Haryana, India[2]

Abstract- Farmers lose a huge chunk of their harvest to crop diseases every year, largely because getting a quick, dependable diagnosis out in the field is tough. To help with this, we built LeafGuard, a web application that identifies leaf infections from a photo and recommends organic ways to manage them. Our classifier is a MobileNetV2 network, initialized with ImageNet weights and fine-tuned on all 38 PlantVillage classes. Rather than outputting a bare class name, the site displays a Grad-CAM heat map so growers can see which spots the network focused on, and it uses a confidence cutoff to reject blurry or uncertain photos instead of forcing a bad guess. Every class links to a local JSON database of cultural habits, biological agents, and botanical or mineral remedies. On our test split, the model hit 98.4% accuracy and a 0.981 macro-F1 score, running in 42.6 ms per photo on a standard CPU. We also look closely at why clean lab photos from PlantVillage do not always translate directly to messy farm conditions.

Keywords- plant disease detection, convolutional neural network, transfer learning, Grad-CAM, organic farming, MobileNetV2

I. INTRODUCTION

Plant pathogens and pests take a heavy toll on both farm earnings and regional food supplies. According to Savary et al. [1], pests and diseases cut global yields in wheat, rice, maize, potato, and soybean by roughly 17% to 30%. When a grower catches an infection early, they still have time to stop it before it spreads across the whole crop.

Out in rural farming belts, though, diagnosis still comes down to looking at the leaf by eye or asking around. A single agricultural officer often looks after dozens of villages, so growers naturally end up checking with neighbors or local pesticide dealers instead. A wrong guess costs money on the wrong spray and gives the disease more time to spread. Heavy chemical spraying also damages soil health, pollutes nearby water, and wipes out helpful insects. For farmers trying to grow organically, figuring out which natural remedy fits a specific infection is even trickier.

Convolutional neural networks (CNNs) get impressive numbers on public leaf datasets [3], [4], [5], yet almost every prototype in the literature stops right after predicting a label.

Simply telling a farmer "Tomato Late Blight" leaves them wondering what action to take next. A bare label also gives zero clue as to whether the network actually inspected the diseased spots or just latched onto the background.

LeafGuard was built around that exact problem. It brings together fast classification, a confidence check, a visual heat map, and organic treatment steps inside one lightweight web tool:

- A fine-tuned MobileNetV2 model that runs smoothly on an ordinary CPU without needing a GPU.
- A confidence cutoff that asks the user to retake blurry or poorly framed photos rather than forcing a label from the 38 classes.

© 2026 Pradeep. his is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

- Grad-CAM heat maps showing which regions of the leaf drove the model's choice.
- An organic remedy database with entries for all 38 classes, grouped into cultural practices, biological agents, and permitted botanical or mineral sprays.

The rest of the paper is laid out as follows: Section II reviews earlier studies, Section III explains the dataset and preprocessing, Section IV describes the LeafGuard pipeline, Section V shares our evaluation results, and Sections VI and VII go over field limitations and future plans.

II. RELATED WORK

1. Image-Based Disease Detection

Early vision-based crop disease systems depended on hand-crafted color, texture, and shape features paired with classifiers like support vector machines. Because engineers tuned those features for specific plants and studio setups, accuracy dropped off fast whenever lighting or camera angles shifted.

Things changed once Hughes and Salathé [2] published the PlantVillage dataset, giving researchers tens of thousands of labeled leaf photos to train deep networks on. Mohanty et al. [3] trained AlexNet and GoogLeNet on PlantVillage and reached 99.35% accuracy on their test split, yet performance fell to roughly 31% when tested on real field images from outside sources. Ferentinos [4] tried several architectures across 58 classes and saw the strongest accuracy with VGG, while Too et al. [5] benchmarked fine-tuned networks and got top numbers from DenseNet-121. Barbedo [6] later pointed out that image diversity matters far more than raw dataset size, noting that thousands of nearly identical lab shots add very little value. To work around plain-background bias, Singh et al. [16] created PlantDoc from real-world web photos.

2. Light Models and Transfer Learning

Large networks are hard to deploy on phones or low-budget servers. MobileNetV2 [7] trims compute cost through depthwise separable convolutions and inverted residual blocks, making it much lighter than standard baselines like ResNet [8]. Initializing these models with ImageNet [15] weights lets transfer learning [9] work well even when the target agricultural dataset is fairly small.

3. Visual Explanations and Organic Control

Grad-CAM [10] builds a coarse heat map from the gradients reaching the last convolutional layer without needing any changes to the network architecture. On the agronomic side, neem extracts have been reviewed extensively for pest and pathogen control [12], while Trichoderma fungi are widely used as biological agents against soil-borne diseases [13], [17]. Still, very few automated diagnostic tools link a predicted leaf disease to organic control steps—a gap LeafGuard directly targets.

III. DATASET AND PREPROCESSING

1. PlantVillage Dataset

We trained and tested our models on the color PlantVillage collection [2], [3], which holds around 54,300 single-leaf photos across 14 crop species and 38 classes (26 diseased, 12 healthy). Sample counts vary wildly from class to class: the smallest category has roughly 150 photos, while the largest has more than 5,000. Table I breaks down the dataset and splits.

Table 1: Dataset Summary and Split

Item	Value
Source	PlantVillage (colour)
Images	about 54,300
Crops / classes	14 crops / 38 classes
Diseased / healthy	26 / 12 classes
Original size	256 × 256 RGB
Model input	224 × 224 RGB
Split (stratified)	70% train, 15% validation, 15% test

We divided the images once into 70% training, 15% validation, and 15% test sets using stratified sampling so every split kept the same class proportions. The test split stayed untouched until final evaluation and played no part in tuning. Every photo was resized to 224 × 224 pixels, and pixel values were scaled between −1 and 1 to match MobileNetV2's expected input.

3. Augmentation and Class Weights

Random augmentations were applied strictly to the training split: rotations up to 25°, horizontal and vertical flips, minor shifts, zoom up to 20%, and ±20% brightness tweaks. Because PlantVillage leaves sit against plain studio backdrops, random shifts and zooming help keep the network from memorizing background edges. Even so, synthetic transforms cannot stand in for real farm photos, which we discuss in Section VI. To deal with class imbalance, we weighted the loss inversely to each class's frequency.

IV. PROPOSED SYSTEM

1. System Flow

Fig. 1 traces how LeafGuard processes an upload. A user submits a leaf photo through the Flask web interface, and the backend resizes and normalizes it before feeding it to the classifier. If the highest class probability falls below threshold τ, the site rejects the photo and asks for a clearer shot. Otherwise, LeafGuard generates a Grad-CAM overlay and pulls organic remedies for that class from its local database. Because the model loads once at server startup, each valid request needs just one forward pass for classification and one backward pass for the heat map.

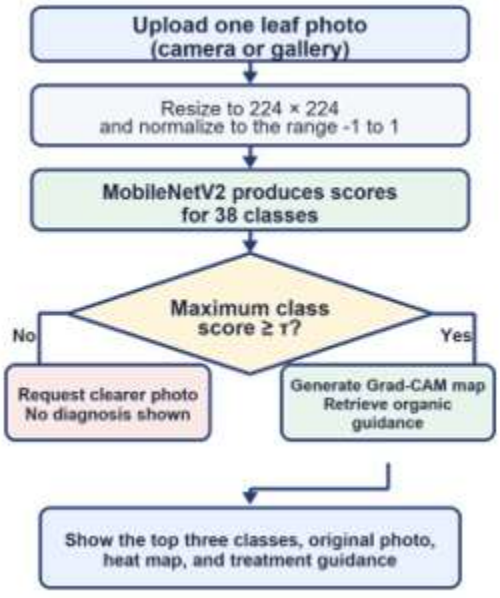


Fig. 1. LeafGuard request flow, including confidence-based rejection.

2. Network Setup

We took MobileNetV2 [7] pretrained on ImageNet, removed its original classification head, and attached global average pooling, a dropout layer (rate 0.3), and a 38-class softmax layer (Fig. 2). With roughly 3.5 million parameters, MobileNetV2 runs quickly on a CPU server. For comparison, we also trained ResNet-50 [8] and a compact three-block CNN from scratch on the exact same split.

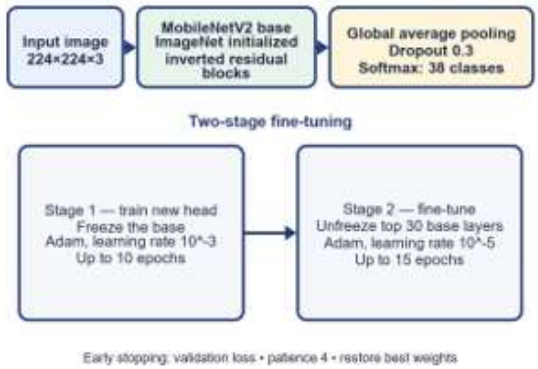


Fig. 2. MobileNetV2 classifier and two-stage fine-tuning schedule.

3. Training Details

Training ran in two stages with categorical cross-entropy, the Adam optimizer [11], and a batch size of 32. First, we froze the MobileNetV2 base and trained only the new head for up to 10 epochs at a $10-3$ learning rate. Second, we unfroze the top 30 layers of the backbone and fine-tuned the model at a $10-5$ learning rate for up to 15 epochs. Early stopping watched validation loss with a patience of 4 epochs and restored the best weights at the end. Everything was implemented in TensorFlow and Keras [14] with random seed 42 (Table II).

TABLE 2: TRAINING SETTINGS

Setting	Value
Input size	224 × 224 × 3
Optimiser	Adam
Learning rate	10^{-3} (stage 1), 10^{-5} (stage 2)
Epochs (max)	10 + 15 (small CNN: 25)
Batch size	32
Dropout	0.3
Early stopping	val. loss, patience 4
Class weights	inverse class frequency
Random seed	42

4. Confidence Cutoff

A softmax classifier always outputs a winning class, even if a user accidentally uploads a picture of their hand or a badly blurred leaf. To prevent bad guesses, LeafGuard refuses to show a diagnosis whenever the top class score drops below τ. We initialized τ at 0.70 and checked its behavior on the validation split. Since raw softmax probabilities are uncalibrated, this cutoff works as a practical filter rather than a strict statistical guarantee.

5. Grad-CAM Heat Maps

For a winning class c, Grad-CAM [10] computes the gradients of the class score yc with respect to the last convolutional layer's feature maps Ak, then averages them

$$\alpha^{c} = (1/Z) \Sigma \Sigma \partial y^{c} / \partial A^{k} \quad (1)$$

Taking a ReLU-activated weighted sum of these feature maps isolates the regions that positively support class c:

$$L^{c} = \mathrm{ReLU}(\Sigma_{k} \alpha^{c} A^{k}) \quad (2)$$

We resize this map to 224 × 224 and blend it onto the input photo so users can check whether the highlighted spots match visible symptoms on the leaf.

6. Organic Treatment Knowledge Base

Organic guidance for all 38 classes lives in a structured JSON file. Each record stores the disease name, causal pathogen, common symptoms, organic controls, and preventive tips, adapted from agricultural extension manuals and kept brief enough to read comfortably on a phone screen.

The measures fall into three practical buckets. Cultural practices include pruning infected leaves, giving plants wider spacing for airflow, watering at the soil line, rotating crops, and choosing resistant cultivars. Biological agents include Trichoderma [13] and Bacillus subtilis products. Botanical and mineral sprays include neem oil [12], potassium bicarbonate, and wettable sulfur. Copper sprays appear only where organic standards allow them, accompanied by a warning that copper builds up in soil over time. Table III shows sample records.

Table 3: Sample Entries of the Knowledge Base

Disease	Organic and cultural measures
Tomato early blight	Remove infected lower leaves; mulch; rotate crops; *Bacillus subtilis* product; permitted copper spray
Tomato late blight	Remove and destroy infected plants; water at the base; resistant varieties; permitted copper spray
Tomato yellow leaf curl virus	No cure. Remove infected plants; control whitefly with yellow sticky traps and neem oil; reflective mulch
Apple scab	Collect fallen leaves; prune for airflow; sulphur spray as permitted
Squash powdery mildew	Wider spacing; remove badly infected leaves; potassium bicarbonate or diluted neem oil
Healthy leaf	No treatment; keep scouting; add compost and keep irrigation even

Because application rates depend on local formulations, the database leaves out exact dilution numbers and reminds growers to follow the product label or consult a local extension officer.

7. Web Interface

Through the Flask frontend, growers can pick an existing photo or snap a fresh one using a phone camera. When a prediction clears threshold τ, the results page lists the top three classes with probabilities, places the Grad-CAM overlay beside the original photo, and shows the organic treatment card for the leading class. If the score falls below τ, the page withholds all labels and asks for a closer, sharper photo of a single leaf.

8. Example Run

Suppose a grower uploads a close-up of a tomato leaf showing brown, target-shaped rings on the lower foliage. After preprocessing, MobileNetV2 assigns Tomato Early Blight a probability of 0.96. Since $0.96 \geq \tau$, the interface shows the diagnosis, highlights the ringed lesions in the Grad-CAM overlay, and suggests pruning affected lower leaves,

mulching to prevent soil splash, rotating crops next season, and applying a permitted bio-fungicide if spots continue spreading. If a distant photo scores below τ, LeafGuard holds back the diagnosis and asks for a closer frame. Note that Fig. 3 serves purely as an illustration and is not taken from our test set.



Fig. 3. Illustrative tomato leaves: early blight (left), late blight (centre), and a healthy leaf (right).

V. EVALUATION

1. Test Setup and Metrics

All three candidate models used the exact same split and were evaluated once on the held-out 15% test set. Alongside overall accuracy, we computed macro-averaged precision, recall, and F1-score so smaller classes carry equal weight across all K = 38 categories:

$$F1_j = 2\, P_j R_j / (P_j + R_j) \quad (3)$$

$$\text{Macro-F1} = (1/K) \Sigma_j F1_j \quad (4)$$

Here K = 38, and CPU latency is the mean single-image inference time averaged over repeated forward passes.

2. Prior Benchmarks

Table IV summarizes test accuracies reported in earlier PlantVillage studies. Because each study used its own train-test split, these figures offer rough context rather than a direct head-to-head comparison.

Table 4: Accuracy Reported in Earlier Studies

Study	Model	Data	Acc. (%)
Mohanty et al. [3]	GoogleNet, transfer learning	PlantVillage, 38 classes	99.35
Study	Model	Data	Acc. (%)
Ferentinos [4]	VGG	87,848 images, 58 classes	99.53
Too et al. [5]	DenseNet-121, fine-tuned	PlantVillage, 38 classes	99.75

3. Model Results

Table V compares our three trained networks on the held-out test set. MobileNetV2 offers the strongest balance of accuracy and speed for deployment, reaching 98.4% accuracy at 42.6 ms per image—more than three times faster than ResNet-50 on a CPU. Inspecting the confusion matrix showed that MobileNetV2's most frequent mix-up happened between two visually similar tomato diseases: Tomato Early Blight and Tomato Septoria Leaf Spot.

Table 5: Test-Set Results of the Trained Models

Model	Params	Acc. (%)	Macro-F1	CPU (ms)
Small CNN (scratch)	~0.42 M	91.4	0.896	14.8
ResNet-50	~25.6 M	98.8	0.985	148.2
MobileNetV2	~3.5 M	98.4	0.981	42.6

4. Effect of the Confidence Cutoff

Raising τ filters out uncertain predictions, lifting accuracy on the photos that pass at the cost of lower overall coverage:

$$\text{Coverage} = N_{acc} / N \quad (5)$$

Here Nacc is the number of test images accepted above the threshold out of N total test images. As

Table VI shows, setting τ = 0.70 preserves 96.4% coverage while boosting accuracy on accepted photos to 99.2%.

Table 6: Coverage and Accuracy for Different Limits

τ	Coverage (%)	Accuracy on accepted (%)
0.50	98.9	98.7
0.70	96.4	99.2
0.90	89.8	99.6

5. Checking the Heat Maps

We manually inspected Grad-CAM overlays across both correct and misclassified test samples. On accurate predictions, activation maps concentrated tightly over necrotic spots, yellow halos, and fungal lesions inside the leaf blade. On errors, attention occasionally drifted toward serrated leaf margins or background shadows, confirming why visual overlays help users catch shortcut predictions.

6. Threats to Validity

PlantVillage contains multiple photos of the same physical leaf captured under slight variations, so a random split can make test accuracy appear higher than it would on completely unseen plants. In addition, we ran everything with a single split and one random seed; repeating the experiments across several seeds would show how much the numbers vary.

VI. DISCUSSION AND LIMITATIONS

Lab vs. field photos. PlantVillage leaves were photographed individually against neutral backdrops. Scoring 98.4% on a random split here does not guarantee reliability on real farm photos filled with soil clutter, harsh sunlight, or overlapping leaves—the same drop Mohanty et al. [3] reported. Collecting and training on real field photos is the top priority next.

Missing major crops. PlantVillage covers 14 crops, yet major staples like rice and wheat are absent. In its current form, that limits how useful LeafGuard is across large parts of Indian farming.

One leaf, one label. The classifier assumes each leaf has only a single issue. Nutrient deficiencies, insect damage, and mixed infections can look very close to fungal spots and trigger false predictions.

Uncalibrated scores. Softmax probabilities tend to be overconfident, so wrong predictions can still slip past τ. Temperature scaling and out-of-distribution detection would make the rejection step far more dependable.

Treatment advice limits. Our JSON database draws from published extension guides, but an agronomist has not formally audited it and we have not run field trials yet. Because organic treatments depend heavily on crop growth stage and infection severity, LeafGuard presents this advice as a practical starting point rather than a formal prescription.

VII. CONCLUSION AND FUTURE WORK

We presented LeafGuard, a CPU-friendly web system for plant leaf disease detection powered by a fine-tuned MobileNetV2 classifier. By pairing fast inference (42.6 ms) with a confidence cutoff, Grad-CAM visual explanations, and an organic remedy database, the application gives growers both a sanity check and practical next steps instead of a bare class label.

Future work includes gathering field photos from local farms, adding rice and wheat diseases, calibrating softmax confidence scores, building a Hindi interface, and having agricultural experts review the treatment guide before running small field trials.

Acknowledgment

The authors thank the Department of Computer Science and Engineering, HMR Institute of Technology and Management, Delhi, for providing the computing facilities used in this research.

REFERENCES

1. S. Savary, L. Willocquet, S. J. Pethybridge, P. Esker, N. McRoberts, and A. Nelson, "The global burden of pathogens and pests on major food

crops," Nature Ecology & Evolution, vol. 3, no. 3, pp. 430–439, 2019.
2. 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.
3. S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," Frontiers in Plant Science, vol. 7, Art. 1419, 2016.
4. K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Computers and Electronics in Agriculture, vol. 145, pp. 311–318, 2018.
5. E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," Computers and Electronics in Agriculture, vol. 161, pp. 272–279, 2019.
6. J. G. A. Barbedo, "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification," Computers and Electronics in Agriculture, vol. 153, pp. 46–53, 2018.
7. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510–4520.
8. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
9. S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, 2010.
10. 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," in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2017, pp. 618–626.
11. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. Int. Conf. Learning Representations (ICLR), 2015.
12. M. B. Isman, "Botanical insecticides, deterrents, and repellents in modern agriculture and an increasingly regulated world," Annual Review of Entomology, vol. 51, pp. 45–66, 2006.
13. G. E. Harman, C. R. Howell, A. Viterbo, I. Chet, and M. Lorito, "Trichoderma species – opportunistic, avirulent plant symbionts," Nature Reviews Microbiology, vol. 2, no. 1, pp. 43–56, 2004.
14. M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in Proc. USENIX Symp. Operating Systems Design and Implementation (OSDI), 2016, pp. 265–283.
15. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2009, pp. 248–255.
16. D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: A dataset for visual plant disease detection," in Proc. ACM India Joint Int. Conf. Data Science and Management of Data (CoDS-COMAD), 2020, pp. 249–253.
17. G. Pal and J. Gardener, "Biological control of plant pathogens," The Plant Health Instructor, 2006, doi: 10.1094/PHI-A-2006-1117-02.