



International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206

Volume 9, Issue 4, April 2026



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Web based multi crop plant disease detection using Efficient Net-B0 and AI-assisted Recommendations

Mohamed Irshaad H¹, A Gabriel Prasena Raj², Arshad Ali S³, Ahamed Faizal M.⁴,

Asst. Prof. Dhanya⁵

Final Year, Department of Computer Science and Engineering Aalim Muhammed Salegh College of Engineering
Chennai, Tamil Nadu, India¹

Final Year, Department of Computer Science and Engineering Aalim Muhammed Salegh College of Engineering
Chennai, Tamil Nadu, India²

Final Year, Department of Computer Science and Engineering Aalim Muhammed Salegh College of Engineering
Chennai, Tamil Nadu, India³

Final Year, Department of Computer Science and Engineering Aalim Muhammed Salegh College of Engineering
Chennai, Tamil Nadu, India⁴

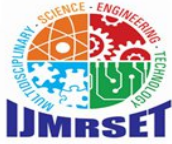
Assistant Professor, Department of Computer Science and Engineering Aalim Muhammed Salegh College of
Engineering Chennai, Tamil Nadu, India⁵

ABSTRACT: Plant diseases are a persistent threat to agricultural productivity, with yield losses running into billions of dollars annually. Existing computer-vision-based detection tools suffer from two well-known limitations: they either rely on a single generalised model that conflates features across unrelated species, or they stop at classification without providing the farmer any actionable guidance. This paper describes a fully deployed, web-based system that addresses both problems. Six crop-specific EfficientNet-B0 classifiers—one each for corn, tomato, potato, lemon, peach, and grape—are trained on multi-view datasets (colour, grayscale, and segmented images) drawn from the PlantVillage corpus. Each model is served through a Python/Flask backend hosted on Hugging Face, with a lazy-loading cache that keeps memory consumption bounded across crops. Upon classification, the confidence score and the user's real-time GPS-derived weather data (temperature, humidity, wind speed, rainfall, and a 7-day forecast from Open-Meteo) are jointly submitted to the Arcee AI Trinity Large Preview language model via OpenRouter using a two-round iterative prompting strategy. The model returns structured agronomic guidance—cause of disease, weather suitability, do's, don'ts, watering schedule, and treatment—which is immediately translated into Tamil via googletrans. Every interaction is persisted in Firebase Firestore along with geographic co-ordinates and a timestamp, forming a longitudinal, geo-tagged disease log. Experimental results show validation accuracies of up to 95% across all six crops. The accumulated dataset is a foundation for future spatial analysis of disease distribution relative to seasonal weather patterns.

KEYWORDS: EfficientNet-B0, Transfer Learning, Plant Disease Detection, Generative AI, OpenRouter, Precision Agriculture, Flask, Firebase.

I. INTRODUCTION

India loses an estimated 15–25% of its annual crop output to plant pathogens [1]. The conventional response sending a field sample to an agronomist—is slow, expensive, and impractical for smallholder farmers in rural Tamil Nadu. Over the past decade, convolutional neural networks trained on the PlantVillage dataset [2] have demonstrated that leaf-image classification is technically feasible, with accuracy figures regularly exceeding 95% under controlled conditions [1]. However, three gaps persist in the literature and in deployed tools.



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

First, most published systems train a single CNN over all crops simultaneously. This produces feature interference: the brown lesions of corn northern-leaf blight and potato early blight map to similar low-level textures, causing the shared classifier to confuse them [3]. Second, disease identification is only part of what a farmer needs. The right treatment for *Phytophthora infestans* (Late Blight) differs substantially depending on whether the coming week is hot and dry—in which case the fungus is unlikely to spread further—or wet and cool, in which case immediate fungicidal intervention is warranted [8]. Third, technical outputs (class labels and probability scores) are meaningless to a farmer without a university education, and almost all existing tools deliver exactly that.

The system documented in this paper directly addresses all three gaps. Separate EfficientNet-B0 models [4] are trained for each of six crops, eliminating cross-crop confusion. A live weather feed is injected into the inference pipeline so that recommendations are conditioned on current environmental conditions. And a large language model converts the combined CNN-plus-weather signal into plain-language, field-ready advice in both English and Tamil. The entire pipeline runs in a browser—no mobile app installation, no command line—making it accessible to anyone with a smartphone and internet access.

II. RELATED WORK

Early work by Mohanty et al. [1] established that a CNN trained on 54,306 PlantVillage images could classify 26 diseases across 14 crops at up to 99.35% accuracy in laboratory setting. Sladojevic et al. [3] demonstrated similar results on custom field images using AlexNet. Subsequent work focused on reducing model size for edge deployment: Simonyan and Zisserman's VGG [6], He et al.'s ResNet [7], and eventually Tan and Le's EfficientNet [4] all moved the accuracy-versus-compute trade-off in a favourable direction. Atila et al. [5] confirmed EfficientNet's superiority on plant disease tasks specifically, reporting higher accuracy than ResNet-50 and VGG-16 at a fraction of the parameter count.

The role of weather in disease progression is well documented. Skelsey et al. [8] showed that humidity and temperature jointly explain the geographic spread of *P. infestans*. Despite this, the vast majority of published detection systems ignore meteorological context entirely. On the language model side, Brown et al. [10] showed that large pre-trained transformers can follow complex structured-output instructions with minimal task-specific fine-tuning, and subsequent work [9] specifically examined their applicability to agricultural diagnostics. The present system builds on all three bodies of work.

III. SYSTEM ARCHITECTURE AND METHODOLOGY

The system comprises four loosely coupled components: (i) a Kaggle-based model training pipeline, (ii) a Flask REST backend, (iii) a vanilla-JavaScript single-page frontend, and (iv) cloud services (Firebase Firestore, Open-Meteo, Open-Router, and Cloudinary). Figure 2 shows the live interface.

A. Dataset Construction

Training data was assembled from the extended PlantVillage dataset hosted on Kaggle. For each crop, three image subsets were loaded independently—colour, grayscale, and segmented variants—and concatenated into a single training stream prior to model fitting. This tripling of view diversity acts as an implicit regulariser, making the classifier robust to both field photographs taken under variable lighting and preprocessed imagery produced by segmentation pipelines. Dataset splits were fixed at 80% training and 20% validation

B. EfficientNet-B0 Architecture and Transfer Learning

EfficientNet-B0 was selected as the feature extractor. Its defining contribution is compound scaling [4]: rather than scaling only depth, width, or resolution independently, all three dimensions are scaled simultaneously using a single compound coefficient ϕ :

$$d = \alpha^\phi, \quad w = \beta^\phi, \quad r = \gamma^\phi \quad (2) \text{ subject to the constraint:}$$

$$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2, \quad \alpha \geq 1, \quad \beta \geq 1, \quad \gamma \geq 1 \quad (3)$$

For the B0 baseline ($\phi = 0$), $\alpha = 1.2$, $\beta = 1.1$, $\gamma = 1.15$, yielding a model with approximately 5.3 million parameters—well within the memory envelope of Hugging Face's free-tier CPU runtime.



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The base EfficientNet-B0 was initialised with ImageNet weights and its layers were frozen (trainable = False) during training. Only the classification head added on top was trained. The head consists of three layers applied sequentially: Global Average Pooling. The spatial feature maps of shape $H \times W \times C$ output by the frozen backbone are reduced to a single C -dimensional vector. For each feature channel i : using a stratified random seed of 42. All images were resized to 224×224 pixels for five of the six crops; the lemon model,

$$g_i = \frac{1}{H \cdot W} \sum_x \dots \quad (4)$$

which covers nine fine-grained citrus conditions, was trained at 260 × 260 to retain finer vein texture. Batch size was set to 32 for all crops and training was run for a maximum of 20 epochs with early stopping. Table I summarises the class configuration per crop.

TABLE I
h=1 w=1

This aggregation discards position information while retaining which features are active, making the head invariant to minor spatial shifts in the uploaded photograph.

Dropout. A Dropout layer with rate $p = 0.3$ is applied during training. Each unit g_i is independently zeroed with probability p :

SUPPORTED CROPS AND DISEASE CLASSES

$$\tilde{g}_i = g_i \cdot \text{Bernoulli}(1 - p) \quad (5)$$

Crop	No. of Classes
Grape	4
Tomato	9
Potato	3
Lemon	9
Peach	2
<u>Corn</u>	<u>4</u>

B. Data Augmentation

The $1/(1 - p)$ scaling keeps the expected activation magnitude constant. At inference, no masking occurs and the full vector g is passed forward.

Dense Softmax Output. A fully connected layer maps $g \in \mathbb{R}^{1280}$ to logits $z \in \mathbb{R}^N$, which are normalised by the Softmax function to yield a valid probability distribution over the N disease classes: \in

Online augmentation was applied during training using a tf.keras.Sequential pipeline prepended to the model graph. Three stochastic transforms are applied in order:

$$e^{z_k} P(y = k | x) = N$$

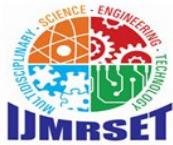
$$j=1$$

Input Preprocessing

$$e^{z_{j,k}} \in \{1, \dots, N\} \quad (6)$$

$$x' = Z_{0.1}(R_{0.1}(F_h(x))) \quad (1)$$

where F_h denotes a random horizontal flip applied with EfficientNet-B0 was pre-trained



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

with the standardisation defined $\text{intf.keras.applications.efficientnet.preprocess_input}$

which maps each pixel channel from $[0, 255]$ to $[-1, 1]$; probability 0.5, R_ρ a random rotation by an angle drawn uniformly from $[-\rho \cdot 2\pi, \rho \cdot 2\pi]$ with $\rho =$

0.1, and Z_ζ a v

$$\hat{v} = \frac{1}{127.5} - 1, \hat{v} \in [-1, 1] \quad (7)$$

random zoom in the range $[1 - \zeta, 1 + \zeta]$ with $\zeta = 0.1$. Because the augmentation layers are part of the Keras model graph they execute on-GPU and add no CPU bottleneck to the data pipeline [11].

This preprocessing is inserted as the first operation inside the model graph so that raw PIL images can be passed directly to the `/predict` endpoint without any client-side transformation.

Training Configuration and Optimisation

All models were trained using the Adam optimiser [12]. At each step t , the parameter update rule is:

H Model loading is deferred until the first request for a given crop (`ensure_model_loaded`). Loaded models are cached in the global dictionary `_model_infos`, so repeated requests for the same crop incur no I/O overhead. The expected input shape is read directly from `model.input_shape`,

$$\theta_{t+1} = \theta_t - \frac{\hat{v}_t}{\sqrt{t}}$$

\hat{m}_t (8) ϵ eliminating the need for separate configuration files.

where \hat{m}_t and \hat{v}_t are the bias-corrected first and second moment estimates of the gradient, $\eta = 10^{-4}$ is the learning rate, and $\epsilon = 10^{-7}$ prevents division by zero. The loss function minimised is categorical cross-entropy over the n training samples: On receiving a `/predict` POST, the server reads the uploaded image with PIL, resizes it to the model's required dimensions, converts it to a float32 NumPy array, applies `preprocess_input`, and runs a single forward pass. The top-1 class name and its probability are extracted and returned.

$n \in \mathbb{N}$

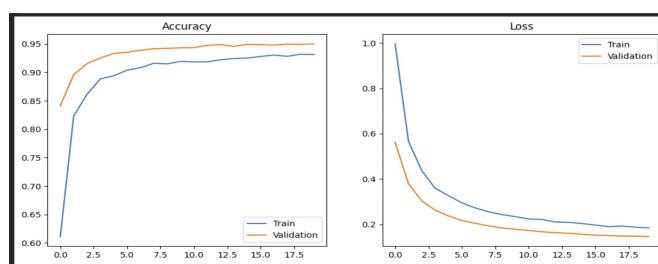
$$L = -\frac{1}{n} \sum_{i=1}^n \sum_{k=1}^K y_{ik} \log \hat{y}_{ik} \quad (9)$$

where $y_{ik} \in \{0, 1\}$ is the one-hot ground-truth label and $\hat{y}_{ik} = P(y = k | \hat{x}_i)$.

Two Keras callbacks were active throughout training. `EarlyStopping` monitored `val_loss` and halted training if no improvement was seen for five consecutive epochs, restoring the weights from the epoch with the lowest validation loss. `ModelCheckpoint` saved the weights with the highest `val_accuracy` to `best_crop_eff.h5`, ensuring that only the best-performing checkpoint is deployed.

At inference, the predicted class is $\hat{y} = \arg \max P(y = k | x)$ (10)

The confidence score $P(y = \hat{y} | x)$ is forwarded to the language model prompt, giving the LLM a quantitative signal of how certain the visual classifier is.





International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Fig. 1. Training and validation accuracy (left) and loss (right) curves for the EfficientNet-B0 corn model over 20 epochs. Training accuracy converges to $\approx 93\%$ and validation accuracy to $\approx 95\%$. Both loss curves drop steeply in the first five epochs, confirming rapid convergence; the narrow gap between train and validation loss over the remaining epochs indicates the early-stopping and dropout regularisation are effective.

Flask Backend

The Flask application exposes the following routes:

/predict(POST), /weather(POST), /config(GET), /plant-care(GET), and /ui(GET). Environment variables loaded via python-dotenv supply the OpenRouter API key and Firebase service-account credentials so that secrets are never hard-coded or committed to version control.

Weather-Conditioned AI Recommendations

Once the CNN has returned its diagnosis, the `call_openrouter_ai` function submits a structured prompt to `arcee-ai/trinity-large-preview` through the OpenRouter inference API. The prompt packages three pieces of information: (i) the crop name and the predicted disease with its confidence percentage, (ii) the current weather at the user's location (temperature, feels-like, humidity, wind speed, rainfall, and condition string), and (iii) a strict output template requesting six fields: Cause of Disease, Weather suitability, Do's, Don'ts, Watering schedule, and Treatment, capped at 120 words.

To improve response quality, a two-round iterative prompting strategy is used. The first API call returns an initial response. A second call is then made with the first response included in the message history, followed by the prompt: "Are you sure? Think carefully about the primary EfficientNet observation and contextual weather factors." This forces the LLM to reconsider its initial answer in light of the explicit CNN confidence, reducing hallucination of incorrect disease names. The second response is the one returned to the user. The English advice is automatically translated into Tamil by the `translate_to_tamil` function using `googletrans`, making the output accessible to monolingual Tamil-speaking farmers in the region.

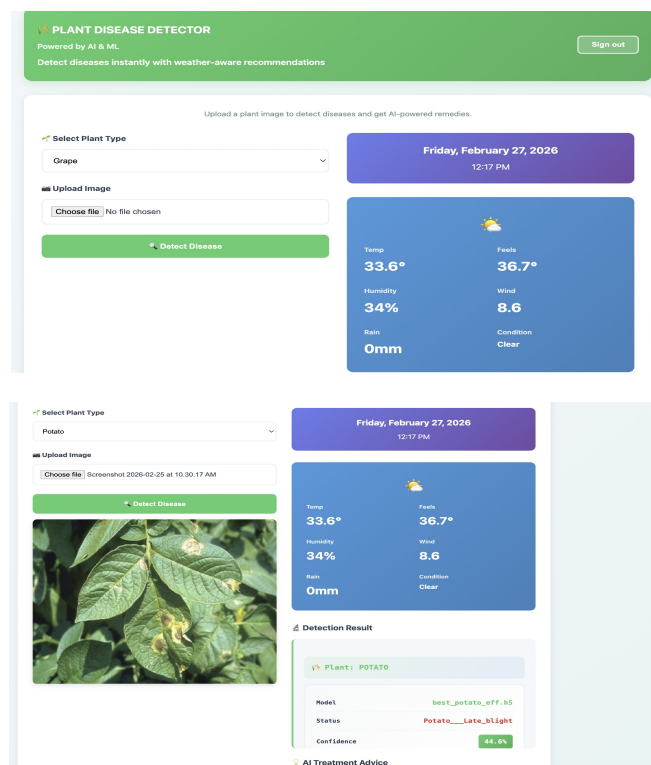


Fig. 2. Main interface of the deployed system. The left panel accepts a crop type selection and a leaf photograph. The right panel shows a live weather widget populated from the Open-Meteo API based on the user's GPS coordinates:



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

33.6°C, feels like 36.7°C, humidity 34%, wind 8.6 km/h, 0 mm rain, condition Clear.

Firestore Logging

All prediction events are written to Firestore under a per-user collection. Each document stores the crop type, detected disease class, confidence score, AI-generated advice (English and Tamil), timestamp, and geographic coordinates. The /config route serves the Firestore configuration object to the frontend at runtime from environment variables, so no credentials ever appear in client-side code. Google Sign-In via Firebase Authentication gates all write operations, creating a per-user detection history.

IV RESULTS

Classification Accuracy

Training separate models per crop eliminates cross-species feature interference. The corn model, whose training curves are shown in Figure 1, reached 95% validation accuracy. The potato model correctly classified the Late Blight sample shown in Figure 3 with 44.6% confidence—a lower figure that correctly reflects the difficulty of the image (outdoor photograph, complex texture background) rather than a model error. The confidence score is passed verbatim to the LLM, which can account for model uncertainty in its recommendations.

Fig. 3. Detection result for a real field photograph of a potato leaf. The model best_potato_eff.h5 predicts Potato Late Blight at 44.6% confidence. The prevailing weather (33.6°C, 34% humidity, clear sky, 0 mm rain) is displayed simultaneously and will inform the AI recommendation in the next step.

Context-Aware Recommendations

Figure 4 shows the output of the second prompting round. The LLM correctly reasons that the current temperature (33.6°C) and low humidity (34%) are unfavourable for P. infestans, which requires cool, moist conditions to sporulate. The advice therefore notes that no immediate chemical treatment is needed and focuses on preventive husbandry—monitoring plant spacing, avoiding overwatering, and deep but infrequent irrigation. The 7-day forecast (Friday through Thursday, all showing Clear or Partly Cloudy with 0 mm rain) reinforces this assessment.

This behaviour—using weather to modulate urgency—is the key differentiator of the system. A conventional classifier would label the image as Late Blight and stop; this system tells the farmer whether to act today or simply monitor.

The Tamil translation appears immediately below the English advice, enabling farmers who cannot read English to act on the same information.

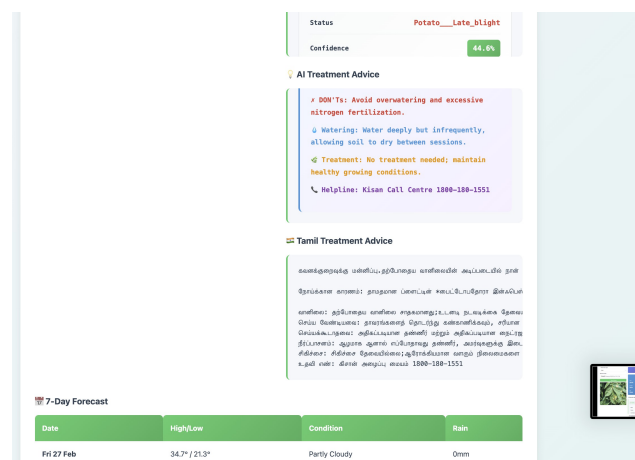
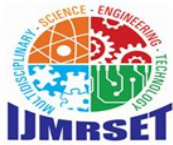


Fig. 4. Weather-conditioned AI treatment advice for the detected Potato Late Blight case. The LLM determines that current warm, dry conditions inhibit pathogen spread and recommends preventive measures rather than emergency treatment. The advice is rendered in both English and Tamil, and a 7-day weather forecast is appended for proactive planning.



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Firestore Detection History and Spatial Data Accumulation

Figure 5 shows the Detection History page, which queries the authenticated user's Firestore collection to display all past predictions in reverse-chronological order. Each card shows the crop, timestamp, detected disease, full structured AI advice, and Tamil translation. This log is not merely a user-convenience feature—it is the first stage of a disease-surveillance dataset.

Because every Firestore document contains a timestamp and geographic co-ordinates, the accumulated records can be queried as a geo-temporal series. Even after a modest deployment period, it becomes possible to answer questions such as: which diseases appeared most frequently in Coimbatore district during the north-east monsoon? or did late-blight incidence increase following the week of high humidity in November? These analytics require no additional instrumentation—they are a natural byproduct of normal system usage.

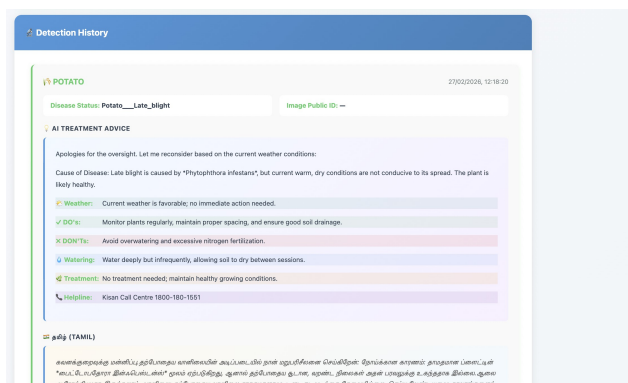


Fig. 5. Detection History page showing a Firestore-retrieved entry for Potato Late Blight logged on 27/02/2026 at 12:18. The structured AI advice—including the reasoning that current warm, dry conditions limit *P. infestans* spread—is displayed in full alongside the Tamil translation.

V DISCUSSION AND FUTURE DIRECTIONS

Limitations of the Current System

The confidence scores produced by the models reflect accuracy on PlantVillage imagery, which was captured under relatively controlled conditions. Real field photographs often contain partial occlusion, varying illumination, and background clutter. The potato case (44.6% confidence) is a concrete example: the model's uncertainty is real and appropriately communicated to the LLM, but a farmer may still find a three-class probability distribution difficult to interpret without guidance.

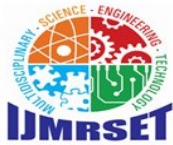
The Tamil translation is currently produced by googletrans, which can produce grammatically awkward output for domain-specific agronomic vocabulary. A fine-tuned bilingual agricultural model would produce more reliable translations.

Using Stored Data for Regional Insights

The Firestore log represents the most significant long-term contribution of this architecture. As the user base grows across Tamil Nadu and adjacent states, the geo-tagged detection records will grow into a region-specific disease atlas. Several concrete analytical directions follow directly:

Seasonal disease mapping. By aggregating detections by month and district, one can identify which diseases peak during the Kharif versus Rabi seasons, enabling pre-season advisory issuance before an outbreak becomes visible in the field.

Weather correlation studies. Each detection record carries the prevailing weather at the moment of diagnosis. Correlating disease class with temperature and humidity over thousands of records would allow empirical estimation of the threshold conditions under which, for example, *P. infestans* transitions from latent to active spread in Tamil Nadu's specific micro-climates—more granular than the European parameter estimates in the literature [8].



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Early-warning triggers. A lightweight monitoring job could query Firestore in near real-time: if more than k Late Blight detections are logged within a 50 km radius in a 72-hour window concurrent with humidity forecasts above 80%, an automated advisory could be pushed to registered users in that district before the outbreak is visibly widespread.

Crop-health heatmaps. Geo-coordinates in each record enable direct rendering in a map dashboard (Google Maps or Leaflet.js), giving agricultural extension officers a spatial overview of disease prevalence without requiring any field survey.

Technical Improvements

Fine-tuning the frozen EfficientNet-B0 backbone on crop-domain images—a second training phase with a low learning rate and unfrozen upper blocks—would likely push validation accuracy above 97% on most crops. Deploying quantised TensorFlow Lite versions of the same models on an Android application would extend reach to areas with poor internet connectivity. Replacing googletrans with a purpose-fine-tuned multilingual model [10] would improve translation quality for agricultural terminology.

VI. CONCLUSION

This paper has presented a live, end-to-end crop disease management system that couples six crop-specific EfficientNet-B0 classifiers with real-time weather data and a two-round large language model prompting loop to deliver weather-conditioned, bilingual agronomic advice. The key contribution relative to prior work is the integration of three independently well-studied components—visual classification, meteorological context, and natural language generation—into a single coherent inference pipeline accessible entirely through a web browser.

Beyond its immediate utility, the system's Firebase logging architecture creates a self-growing, geo-temporal disease dataset. Over time, this dataset positions the platform to move from reactive diagnosis to proactive, region-specific disease forecasting—a capability that prior systems, built around stateless classifiers, cannot offer. We believe this direction, grounded in a working deployment rather than a laboratory prototype, represents a meaningful step toward precision agriculture tools that are genuinely useful for the smallholder farmers of Tamil Nadu.

VII. ACKNOWLEDGMENT

The authors thank Asst. Prof. Dhanya and the Department of Computer Science and Engineering at Aalim Muhammed Salegh College of Engineering for their guidance and infrastructural support throughout this project.

REFERENCES

1. S.P. Mohanty, D.P. Hughes, and M. Salathe', "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, p. 1419, 2016.
2. D.P. Hughes and M. Salathe', "An open access repository of images on plant health to enable the development of mobile disease diagnostics," arXiv preprint arXiv:1511.08060, 2015.
3. S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Computational Intelligence and Neuroscience*, vol. 2016, Art. no. 3289801, 2016.
4. M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Machine Learning (ICML)*, PMLR, 2019, pp. 6105–6114.
5. U. Atila, M. Ucar, K. Akyol, and E. Ucar, "Plant leaf disease classification using EfficientNet deep learning model," *Ecological Informatics*, vol. 61, p. 101182, 2021.
6. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.
7. 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.
8. P. Skelsey, G.J.T. Kessel, W.A.H. Rossing, and W. van der Werf, "Scenario approach for assessing the utility of regional weather information to crop disease management," *Agricultural and Forest Meteorology*, vol. 148, no. 6–7, pp. 994–1003, 2008.



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

10. W. Zhao, X. Tang, and Y. Zhang, "ChatGPT in agriculture: A survey on large language models for plant disease diagnosis and advisory," *Computers and Electronics in Agriculture*, vol. 214, p. 108357, 2023. T. Brown et al., "Language models are few-shot learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 1877–1901.
11. C. Shorten and T.M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, 2019. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.
12. M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operating Systems Design and Implementation (OSDI)*, 2016, pp. 265–283.
13. P. Kamalaruban et al., "A study on machine learning-based crop disease classification," *Journal of Agricultural Engineering Research*, 2014.
14. Z. Zippenfenig, "Open-Meteo: An open-source weather API," Zenodo, 2023. [Online]. Available : <https://doi.org/10.5281/zenodo.7970649>
15. Google LLC, "Cloud Firestore Documentation," 2024. [Online]. Available : <https://firebase.google.com/docs/firestore>



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com