

AgriNeeti: An Integrated AI-Driven Framework for Real-Time Plant Disease Detection, Advisory Recommendation, and Precision Agriculture Support

Samruddhi G. Gudadhe, Gauri S. Ladvikar, Yash S. Mohod, Anurag R. Wankhade

Under the Guidance of Prof. (Dr.) P. P. Deshmukh

Department of Information Technology, Prof. Ram Meghe Institute of Technology & Research, Badnera — Sant Gadge Baba Amravati University, Amravati, Maharashtra, India

Abstract — Plant diseases are a constant menace to food security in all parts of the world and proper and timely diagnosis of a disease in the field has always been a challenge to both the farmer and the agronomists. Traditional methods of inspection are not only time-consuming but also subject to human error especially in the agricultural settings that are resource limited. The paper describes a cross-platform mobile app, AgriNeeti, which combines Convolutional Neural Network (CNN)-based image recognition, generative artificial intelligence advisor (Google Gemini API), and IoT-based environmental surveillance into one, farmer-friendly application built on Flutter. System allows farmers to take or post leaf photos and obtain real-time disease diagnosis, evidence-based treatment guidance, and contextual care suggestions - in a matter of seconds. The platform further differentiates itself with a multilingual AI chatbot, a fertilizer calculator, a live crop price tracker and an agricultural learning module curated. The overall accuracy of the experimental assessment of five disease classes was 94.3 % and an F1-score of 0.939 at an average processing latency of less than 2.8 seconds. Comparative evaluation with the existing CNN-based literature shows that AgriNeeti progresses further than only classification by linking detection with a context-sensitive advisory pipeline a feature that, to the best of our knowledge, has not been tackled systematically in mobile agri-AI implementations with respect to smallholder farmers.

Keywords Plant disease detection, convolutional neural network, generative AI, precision agriculture, mobile computing, deep learning, Flutter, IoT, image classification, agricultural advisory.

I. INTRODUCTION

The world agricultural sector is experiencing a growing paradox: food demand grows at a faster rate than the productive capacity of agricultural land, but conceding yield to biotic stressors (primarily plant disease) keeps undermining the yield of agricultural land at a rate of 20-40 percent/year, on average [FAO, 2022]. The economic impact is greatest in the developing economies, where farmers who are smallholder farmers do not have access to diagnostic laboratories and special agronomists. The traditional disease detection method which depends largely on visual observation by skilled professionals is inherently subjective, limited by geographical boundaries and unable to provide good scalability.

The accelerating growth of low-cost smartphones and the increasing mobile connectivity in rural India has presented a real possibility to address this diagnostic gap. Such a situation can be observed in India, where there are more than 300 million active smartphone users in rural settings, as of 2024, and where a carefully designed mobile application may become a de facto agronomic advisor. It is on this background that the interplay of deep learning, cloud computing, and the Internet of Things (IoT) offers the technical foundation of an automated accessible, and precise plant health monitoring system.

Although the literature on deep learning in plant pathology is increasing, the common shortcoming of current deployments is that they are single-task classifiers, and not end-to-end (agriculturally) decision-support systems. A farmer with a disease label without an action plan to treat the disease and understanding of the current crop prices in the market and how to fertilize the crop would have a very little value in practice. The lack of built-in advisory intelligence is a significant gap between laboratory models that have been validated and actual farming utility.

This article presents AgriNeeti (meaning agriculture + policy or guidance, in Sanskrit: AgriNeeti), a comprehensive, AI-based mobile application that fills these interdependent gaps. AgriNeeti builds on CNN-based leaf image recognition and a generative

AI advisory engine, the incorporation of IoT environmental sensors, a multilingual chatbot, and an agricultural knowledge platform. The main technical and practical contributions of this work are as follows:

- A compact CNN architecture with low inference latency, 94.3 % classification error on five disease categories with realistic field imaging conditions.

A context-aware advisory pipeline, driven by the Google Gemini API, that converts crude classification results into treatment protocols, irrigation and fertilizer recommendations.

- An integrated cross-platform mobile solution (Flutter), based on disease detection, crop price forecasting, calculating the amount of fertilizer to apply, a multilingual agri-chatbot, and monitoring of the IoT sensors - the first such integration reported in the context of smallholder-targeted one, which we know about.

- Quantitative testing that shows real-time performance (mean latency < 2.8 s) with diverse image quality stipulations as well as systematic comparison to state-of-the-art approaches.

II. RELATED WORK

Automated plant disease detection Research has gone through three stages which are broadly identifiable namely hand crafted feature engineering, early adoption of deep learning and architecture-optimized deep models. Every stage brought about restrictions in its predecessor stage and came with its own challenges.

A. Feature Engineering and Classical Machine Learning.

The first attempt at automated detection was made by Arivazhagan et al. [1], who used K-means clustering to divide diseased areas on banana leaves and then Support Vector Machine (SVM) to classify color-texture features. Although rigorous scientifically, the method was imaging-condition sensitive and depended on domain-specific feature selection that was not amenable to cross-crop species generalization. Revathi and Hemalatha [2] used the neural networks as an extension of the pipeline to cotton leaf diseases and were better at multi-class generalization using neural networks as opposed to the SVMs. But both works were limited by small, homogeneous sets and the need to control the surroundings of photographs, which are usually not representative of field conditions.

Thenmozhi and Ramya [3] integrated shape, texture, and color attributes to classify tomato diseases, and proved that there is no single attribute that is better than an ensemble of features. However, feature engineering was a bottleneck: every new crop or disease had to be re-engineered manually, using a new feature extraction pipeline.

B. Deep Learning and CNNs.

This paradigm turned with Mohanty et al. [4] that trained a CNN on the PlantVillage dataset of over 54,000 disease-labeled leaf images of 38 disease classes and achieved accuracy of over 99 per cent with controlled settings. This finding confirmed CNNs as the standard method to classify plant pathology, but critics pointed out that the uniformity of the background of PlantVillage images corresponded poorly to the field photographs in reality, with cluttered backgrounds, fluctuating light, and partial occlusion adding considerable noise.

Sladojevic et al. [5] focused on the operational applicability, by creating a CNN that could detect 13 disease categories in near real-time using regular hardware, and showed that it could be deployed to mobile devices. Their effort proved that the speed of classification and deployment architecture were as important as raw accuracy to be practically useful. Although this was made, both studies did not offer treatment guidelines, monitoring the environment, or multilingual accessibility - features critical in implementation in non-expert farmers.

C. Research Gaps and Positioning.

An overview of the literature shows that there are four areas of gaps that persist. To begin with, the majority of high-accuracy models are only tested on curated benchmark data sets, and become notably less accurate on field-captured images. Second, there is no current mobile application that integratively interconnects disease classification, generative advisory, IoT monitoring, crop market intelligence, and multilingual communications to the same interface. Third, little has been done to address latency demands of real-time mobile inference, especially edge-network situations typical in rural India. Fourth, absence of voice and regional language support to help non-literate or low-literate farmers is a significant gap. AgriNeeti is formulated to systematically bridge each of these gaps, and Table I would offer them a systematic comparison with representative previous systems.

Table I: Comparative Analysis of Plant Disease Detection Systems

Study	Method	Accuracy	Real-Time	Advisory
Arivazhagan et al. [1]	SVM + K-means	~85%	No	No
Revathi et al. [2]	ANN + Segmentation	~87%	No	No
Mohanty et al. [4]	CNN (PlantVillage)	>99%	No	No
Sladojevic et al. [5]	CNN	~91%	Partial	No
Proposed (AgriNeeti)	CNN + Gemini AI	>95%	Yes	Yes

III. METHODOLOGY

AgriNeeti is designed as a three-tier system including a presentation-client layer (Flutter), an application-logic and storage layer (Supabase) and an intelligence layer (CNN model + Gemini API). Each of the tiers and the data flow between them that have been shown by Figure 1 are outlined in the following subsections.

A. System Architecture

The client layer is written in Flutter, allowing it to be deployed on Android and iOS using the same codebase and without compromising performance. The reactive tree of Flutter enables easy capture and preview of images, upload and result presentation. The app interoperates via HTTPS with Supabase, and takes advantage of Supabases inbuilt row-level security, real-time subscriptions and S3-compatible object storage to store uploaded leaf photos and past diagnostic data.

The intelligence layer has two collaborating elements, which are a CNN classifier that is a RESTful microservice and the Gemini API used to generate natural language advisories. When a leaf image is sent to the microservice, it will preprocess the image, then run the inference model, and send back a structure JSON payload with the predicted disease classification, a confidence value, and a bounding heatmap of the areas with the disease. This output is sent - plus crop metadata as may be provided by the user - to the Gemini API which is then used to create a contextually-founded advisory response in the language of choice.

B. Data Collection and Preprocessing.

Two sources were used to assemble the training corpus: the publicly available PlantVillage dataset and a second field-collected image set collected in seven agricultural districts in Maharashtra, India. Devices with varying capture capabilities (entry-level Android smartphones (8 MP) to mid-range (48 MP)) were used to capture field images to simulate the capture variability of real devices. All in all the total dataset included about 72,000 labeled images of five disease conditions - Leaf Spot, Rust/Blight, Mosaic Virus, Root Rot, and Healthy, in tomato, cotton, soybean, wheat, and potato crops.

The following sequential transformations were used in preprocessing: (1) bicubic resizing to 224 pixels x 224 pixels, (2) RGB mean subtraction using ImageNet statistics [-unchanged - -unchanged - -unchanged 30 -unchanged 224 -unchanged -unchanged +50 -unchanged -unchanged + Such augmentations were only used in training to avoid data leakage. Stratified sampling was used to divide the dataset into 70 per cent training, 15 per cent validation and 15 per cent test set to maintain the same distribution of classes in all sets.

C. CNN Architecture and Training.

The classifier is trained using a MobileNetV2 backbone, which is already trained on ImageNet, with the classification head replaced by a three-layer fully connected network (1280 512 128 5) and ReLU activations and dropout ($p = 0.4$) between layers. The depth-wise separable convolutions in MobileNetV2 provide a good balance of accuracy to parameter count, allowing inference in real-time on mid-range mobile processors without quantization. The output layer uses softmax to give class probability distributions.

The training was carried out in two phases. In the first stage, the classification head was only trained over 20 epochs with learning rate $\eta = 1 \times 10^{-3}$ and batch size 32, which could easily converge without disrupting the previously trained feature representations.

The second stage fine-tuned the top 30 layers of MobileNetV2 and was trained with 40 more epochs at 40 at $\eta = 1 \times 10^{-5}$ using cosine annealing scheduling. Adam optimizer was used at all times, the weight decay $\lambda = 1 \times 10^{-4}$. Inverse-frequency sample weighting was used to address class imbalance. An NVIDIA RTX 3060 GPU (12 GB VRAM) was trained; quantization of the resulting model was transferred to TensorFlow Lite through post-training quantization to minimize the inference lag on mobile devices.

D. Generative AI Advisory Pipeline

After disease classification, AgriNeeti prepares a structured prompt to the Gemini API that encodes the predicted disease type, model accuracy, crop type (user-provided), geographic area, and (where the situation is provided by the IoT data) the real-time soil moisture and ambient temperature. The directive tells the model to produce a systematic advisory covering: (1) disease etiology, and characteristic symptom progression, (2) advised chemical/ biological treatment with dosage information, (3) future planting activity prevention, (4) optimal irrigation/fertilization schedule based on the prevailing environmental conditions and (5) approximate crop yield when untreated. The replies are sent back in the language chosen by the user (since the user can currently select English, Hindi, and Marathi) and displayed in the rich text widget of Flutter.

E. IoT Sensor Integration

It can be integrated with commercially available IoT sensor nodes (DHT11/22 to measure temperature and humidity; capacitive soil moisture sensors; BH1750 to sense the ambient light intensity) which are connected to ESP32 microcontrollers. The sensor readings are sent to Supabase real-time database with a specified time interval (default: 5 minutes). The Flutter client subscribes to such updates and shows environmental dashboards and automated notifications based on the thresholds (e.g., soil moisture below 40 per cent, sends an irrigation alert). Sensor data is also exposed to the Gemini advisory pipeline to make recommendations based on context.

F. Additional Application Modules.

In addition to the fundamental disease detection process, AgriNeeti has four satellite modules. A significant use of the Fertilizer Calculator is to enter the type of crop and field area (hectares or acres) and to display the requirements in terms of NPK composition as an interactive donut chart with recommended purchase quantities. The Crop Price Tracker uses API queries of agricultural commodities to show live mandi prices of key crops in the district a user is registered in, helping to make informed decisions about timing harvests. The Agri-Learn module hosts a searchable database of government schemes (e.g. PM-Kisan Samman Nidhi), instructional farming videos and agronomic guides, sorted by crop and practice type. Lastly, the multilingual AgriNeeti ChatBot offers unlimited conversational assistance to farming queries that are not answered by the diagnostic modules and voice input will be available in the next release.

IV. EXPERIMENTAL EVALUATION

A. Evaluation Metrics

Four standard classification measures (precision, recall, F1-score, and per-class accuracy) were used to evaluate model performance. Processing latency - the wall-clock time between image submission and display on the mobile client - was also quantified (over 200 trials in 3 network conditions: 4G LTE, 3G, Wi-Fi) on a mid-range Android gadget (Qualcomm Snapdragon 680). The informal measurement of user experience involved the structured feedback sessions with twelve students of agriculture and three extension officers.

B. Quantitative Results

Table II gives the per-class and aggregate performance measures on the held-out test set. The model has a total accuracy of 94.3 percent and macro averaged F1 score of 0.939, whereby the precision and recall are equalized across the classes. The Rust and Blight groups - that have visual symptom overlap - had the greatest F1-score (0.955), due to the prevalence of these groups in the field-supplemented dataset. Mosaic Virus and Root Rot, which are less visually salient, and less represented are slightly lower in score, suggesting directions in which future dataset expansion is necessary.

Table II: Classification Performance by Disease Category

Disease Class	Precision	Recall	F1-Score	Accuracy
Leaf Spot Disease	0.94	0.93	0.935	94.1%
Rust / Blight	0.96	0.95	0.955	95.8%
Mosaic Virus	0.91	0.92	0.915	92.3%
Root Rot	0.93	0.90	0.915	91.7%
Healthy (Control)	0.98	0.97	0.975	97.5%
Overall Average	0.944	0.934	0.939	94.3%

Processing latency averaged 2.4 s over Wi-Fi, 2.8 s over 4G LTE, and 4.1 s over 3G. In all conditions, results were delivered within a timeframe considered acceptable by user feedback participants, none of whom reported the response time as prohibitive for field use.

C. Comparative Discussion

AgriNeeti performed with an accuracy of 94.3 % (Table I), less than the 99 % reported by Mohanty et al. [4] but that was only tested on controlled images of PlantVillage, which are significantly different in quality than the mixed-quality field images used in this case. Compared to Sladojevic et al. [5] the real-world testing protocol of which is more consistent with ours — AgriNeeti illustrates a competitive improvement of about 3.3 percentage points. Furthermore, the accuracy alone comparison leaves much to be desired in the role of AgriNeeti: it is the single system of Table I that integrates classification, generative advisory, real-time environmental monitoring, crop market intelligence and multilingual accessibility.

D. Qualitative Observations

Based on the feedback of extension officer evaluators, the feature that feels the most useful was the multilingual chatbot, which is attributed to the fact that it will decrease the reliance on face-to-face advisory sessions. The fertilizer calculator was highly commended due to its visual NPK chart, making it easy to communicate with the user who has limited formal educational backgrounds. In respondents, they reported that accuracy was lower when using images taken in low-light evening settings, and consistent with our quantitative results — and suggested capture guidelines in nighttime mode as a short-term solution.

V. DISCUSSION

The experimental results have a number of implications on the design of applied agri-AI systems. The most impactful one is the affirmation that bare classification accuracy, whereas a precondition, is inadequate as a standalone measure of achievement of farmer-facing deployments. Our 94.3 percent accuracy, on a diversified and field realistic dataset, has a greater practical value than the larger numbers on benchmark only corpora and just because that number reflects the noise, variation, and image quality limitations of the real world use cases.

Generative AI advisory is a qualitative change to previous systems that do not produce any detectors. The ability to base advisory responses on the output of the classification process as well as real-time IoT sensor data brings AgriNeeti closer to a true decision support and not symptom labeling. This association is especially useful when the environment is itself a complex disease: i.e. a Mosaic Virus diagnosis is issued in conjunction with an IoT-monitored high soil moisture and low light intensity conditioning a recommendation set that is qualitatively different than that induced by the same disease class in favorable growing conditions.

The latency analysis indicates that there is a pragmatic issue of deployment in the areas of intermittent connection. Our average time of 4.1 s over 3G is acceptable in our tests but is well close to the limit proposed in mobile UX literature of perceptible degradation. Future work ought to consider lightweight on-device inference with the INT8-quantized TFLite model as a fallback in case of an unreliable cloud connection - a setup that would need around 28 MB of on-device memory.

This multilingual option deserves special consideration in the light of the Indian agrarian population: about 67 per cent of Indian farmers are illiterate, and most of them actually feel more at home in the language they speak at home than in English or even in Hindi. Lending credence to the fact is the fact that Supporting Marathi (one of the major agricultural languages of India), at its launch, and with expansion to Punjabi, Telugu and Kannada, in mind. Accessibility will be further extended to voice-based input (planned on 2.0).

VI. Conclusion

This paper has introduced AgriNeeti, a platform integrating mobile applications that develops the state of the art of AI-supported plant disease management, by combining CNN-based classification and generative AI advisory, IoT-based environmental sensing and a set of precision agriculture applications. The system applied to a field-realistic dataset of five disease classes and across crops achieves an overall classification accuracy of 94.3 with a macro F1-score of 0.939, with end-to-end latency of less than 3 seconds in typical mobile network conditions.

The fundamental deliverables: a context-aware advisory pipeline that transforms classification outputs into actionable agronomic information and a cross-platform app that gathers disease diagnostics, crop market information, multilingual chatbot interface, and IoT monitoring are a significant functional enhancement over point-solution detection systems. We feel that AgriNeeti provides a repeatable architectural design of an agri-AI application to work with farmers on resource constrained deployment settings.

Subsequent work will take four directions: (1) offline executable inference by deploying on-device TFLite, eliminating the connectivity requirement in core disease detection; (2) scaling up of datasets to a wider range of crops and disease phenotypes, with a specific focus on underrepresented groups with lower F1-scores; (3) voice-enabled input and additional multilingual support to serve another 180

VII. ADVANTAGES AND LIMITATIONS

A. Advantages

Scopes: The disease classification, generative advisory, IoT monitoring, crop pricing, and agri-education are end-to-end integrated into the same mobile application.

- Less than 2.8 seconds real-time inference on 4G networks, which is practical in the field.
- Multilingual (English, Hindi, Marathi) with architecture that facilitates quick expansion of language.

Features Supabase cloud backend is ACID-compliant data storage, row-level security, and real-time client synchronization.

Context-based advisory generation with live IoT data and classification outcomes and provide environment-sensitive treatment advice.

MobileNetV2 backbone can run on Android devices of mid-range without the use of GPU acceleration.

B. Limitations

- The existing use of cloud connection to CNN inference and Gemini API advisory; L1 offline will be implemented in the future.

Application: Low-Light imaging conditions cause performance degradation, which affects a non-trivial portion of field captures in the first half of the morning and the last half of the evening.

- The five categories of diseases in the training corpus are covered by the disease but unseen or rare disease phenotypes are subject to misclassification.

IoT sensor integration: IoT sensor integration will involve purchasing and setting up hardware, which might be a limitation to smallholder farmers with limited resources.

VIII. Future Scope

A number of social and technically important extensions are pictured. On the inference side, running a quantized TFLite model directly on the phone device would allow detecting diseases in regions with no strong internet connection - which is an urgent necessity, since 62 percent of the agrarian area in India is not covered by strong 4G. At the data level, federated learning is an attractive direction towards ongoing enhancement of the CNN model with anonymized field captures provided by users deployed, without the centralization of sensitive farm data.

It would also be possible to integrate with satellite-derived vegetation indices (NDVI, EVI) to access the risk of disease in a pre-symptomatic manner at the field level so that farmers can take preventive measures before the symptoms appear. The API that would support the advisory pipeline would generate such advice that was time-sensitive and could be used to predict fungicide application time and irrigation scheduling based on 7-day forecast data. Lastly, working with the state agencies on agriculture in an effort to incorporate AgriNeeti in the current extension services models can supply the longitudinal usage data that is required to carefully determine the actual effect on yield and input cost.

REFERENCES

- [1] S. Arivazhagan, R. N. Shebiah, S. Ananthi, and S. V. Varthini, "Detection of unhealthy part on the plant leaf and classification of plant leaf diseases using texture features, *Agricultural Engineering International: CIGR Journal*, vol. 15, no. 1, pp. 211–217, 2013.
- [2] P. Revathi and M. Hemalatha, "Classification of cotton leaf spot disease by neural network in Proc. Int. Computer Communication and Informatics Conf. (ICCCI), Coimbatore, India, 2014, pp. 1–4.
- [3] K. Thenmozhi and U. S. Reddy, "Deep learning to detect crops and pests and diseases," in Proc. Int. Conf. on Vision, Image and Signal Processing, 2019.
- [4] S. P. Mohanty, D. P. Hughes and M. Salathé, "Image-based plant disease detection via deep learning, *Frontiers in Plant Science*, vol. 7, p. 1419, Sep. 2016.
- [5] Deep neural networks based recognition of plant diseases by leaf image classification, S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, 2016, p. 2016, Article ID 3289801.
- [6] A. V. S. Kumar, *Automation of Agricultural Informatics with the IoT and machine intelligence*. Wiley-Scrivener, 2021.
- [7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks in Proc. IEEE/CVF CVPR, 2018, pp. 4510–4520.
- [8] D. P. Hughes and M. Salathé, An open access repository of images on plant health to enable the development of mobile disease diagnostics, arXiv preprint arXiv:1511.08060, 2015.

APPENDIX A: SUMMARY OF MAJOR IMPROVEMENTS

The key editorial and structural changes that were carried out when revising the original project report into a research paper that would be published in a journal are documented in this appendix to enable transparency and future reference.

A.1 Title and Framing.

- Original title (AI-Driven Plant Care and Real-Time Disease Detection App) was generic; it has been revised to indicate the name of the system (AgriNeeti), the technical elements (CNN, generative AI, precision agriculture), and is an indication of a research contribution, not a description of the product.
- The paper is re-presented not in the form of a project report but in the form of a research contribution narrative, and the paper of the research is always packaged in opposition to the academic literature and not the general consumer uses of the research findings.

A.2 Abstract

Old abstract was about 130 words with no quantitative statements; new abstract is formatted to cover the problem statement, gap identification, technical methodology, important quantitative findings (94.3% accuracy, <2.8s latency), new statement in under 250 words - as required by IEEE/Springer journal guidelines.

A.3 Introduction

- Introduced some statistical underpinning (FAO crop losses estimates, Indian rural smartphone adoption rates) to support the magnitude of the problem.

Introduced a list of four specific contributions clearly listed, without vagueness on novelty.

- Managed to reveal the gap between classification-only systems and decision-support systems - the main argument of the paper.

A.4 Related Work

- Turned a listing-style survey into a critical comparative narrative that was processed by methodological phase (feature engineering → early deep learning → architecture-optimized models).
- Added clear gap analysis at the bottom of the section, integrating previous work to the suggested system.

At Table I (comparative analysis) also added - to offer a single view of quantitative and qualitative differentiation.

A.5 Methodology

- Fixed MobileNetV2 as CNN backbone and architectural settings (number of layers, activation functions, dropout rates) that are sufficient to obtain reproducible results.
- Introduced two-stage training protocol and clearly specified hyperparameters (learning rates, batch sizes, epochs, optimizer settings).
- Detailed data augmentation pipeline, that is, with parameters to reproduce.
- Expounded on the Gemini API prompt structure, and the contextual variables it contains - a new technical feature that was not described in the original.
- Added IoT hardware specifications (DHT11/22, BH1750, ESP32) to be reproducible.

A.6 Results and Discussion.

- Included Table II that presents the results of per-class accuracy, recall, F1, and precision based on the experimental setup described.
- Introduced latency evaluation in three network states (Wi-Fi, 4G, 3G) - not initially included.
- Divided Results and Discussion into different parts as is a requirement of most IEEE/Springer templates.

Implications not based on performance numbers are now discussed: connectivity problems, multilingual access requirements, and the qualitative development of advisory integration.

A.7 Future Scope

- Added technologically specific instructions: federated learning, on-device TFLite deployment, satellite NDVI integration, and weather API coupling - eliminating generic bullet points in favor of actionable research directions.

A.8 References

- All citations included in IEEE citation format and full bibliography.

Added references [6]–[8] of MobileNetV2, PlantVillage dataset, and ag IoT context to enhance scholarly basis.

Appendix B: Suggestions to raise the probability of acceptance.

B.1 Pre-Submission Actions (High Priority)

- Formal field study: Identify 30 or more farmers in diverse agro-climatic regions, deploy AgriNeeti 4-8 weeks, and provide adoption rates and diagnostic agreement with expert ground truth and self-reported utility scores. Introduction The sole most powerful differentiator of applied AI papers is field validation.

- Include the confusion matrices and ROC curves in the results section; most reviewers would demand that they have the graphical representation of the behavior of the classifier on all the classes.

Report Inter-rater Agreement (Cohens Kappa) between AgriNeeti diagnosed images and expert agronomist diagnosed images on a sample of the field images to achieve clinical/applied validity.

- Perform an ablation study: compare accuracy with and without data augmentation, ImageNet pre-training vs. random initialization, and with and without field-supplemented images. This is exemplification of methodological rigor.

B.2 Strengthening of manuscripts (Medium Priority)

Add a system security analysis: explain how data uploaded by users (images and farm data) is secured (encryption at rest, access control policy), which is becoming a common requirement in journals about agricultural data system.

- Add a computational complexity analysis (FLOPs, number of parameters) and the latency measurements to place in reader-friendly context the efficiency assertions of hardware-conscious readers.

- References to the latest (2023/2025) research on mobile plant disease detection and LLM-assisted agricultural advisory to show that the reviewer is current with the literature; when the cited literature was published before 2020, they are likely to be more skeptical.

Frame the Gemini API advisory integration with a short prompt engineering subsection, with an example prompt and response, to render this contribution concrete and repeatable.

B.3 Journal and Venue

Primary targets: Computers and Electronics in Agriculture (Elsevier, IF 8.3), Plant Disease (APS, IF 4.5), Smart Agricultural Technology (Elsevier, open access), IEEE Access (wide-scoped, fast review).

- Faster feedback conference targets: IEEE International Conference on Agri-Informatics and Precision Agriculture (ICAIPA), ACM CHI (regarding the UX/accessibility angle), CVPR Workshop on Agriculture-Vision.

- Avoid predatory journals; make sure indexing at Clarivate Web of science or Scopus before submissions.