International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

# AgroSense AI: A Deep Learning Framework for Automated Crop Disease Identification and Context-Aware Agricultural Advisory

Riya Patel
Department of Computer Science and Engineering
Indus Institute of Technology and Engineering, Indus University, Ahmedabad, Gujarat, India
Supervisor: Prof. Urvi Rabara, Assistant Professor

***Abstract*** **- Indian agriculture sustains over 58% of the rural workforce, yet it remains acutely vulnerable to crop diseases that account for yield losses of 20%–40% on a global scale. This paper introduces AgroSense AI — an integrated web-based platform combining convolutional neural network (CNN) inference, structured disease advisory, and a conversational AI assistant to address this challenge. Three leading CNN architectures — ResNet50, EfficientNetB3, and MobileNetV2 — were comparatively evaluated using transfer learning on the PlantVillage benchmark, which contains 54,305 annotated leaf images spanning 38 pathological categories across 14 crop genera. Experimental validation established that EfficientNetB3 attained the highest classification accuracy of 96.60%, with ResNet50 following at 95.88% and MobileNetV2 at 79.05%. The deployed system exposes a Flask-based REST API, a responsive browser-based user interface, and an advisory chatbot powered by Google Gemini, collectively supporting 62 disease categories across 15 crops with India-specific treatment, prevention, and yield-impact guidance.**

***Keywords*** **- Crop disease classification, convolutional neural network, transfer learning, ResNet50, EfficientNetB3, PlantVillage dataset, precision agriculture, computer vision, plant pathology**

## I. INTRODUCTION

Crop production underpins global food security and rural livelihoods, yet it is persistently threatened by fungal, bacterial, and viral pathogens. The Food and Agriculture Organisation of the United Nations estimates that plant diseases destroy between one-fifth and two-fifths of total agricultural output annually — losses translating to hundreds of billions of dollars in economic damage [18]. In countries such as India, where small-holder farming dominates, even localised outbreaks can cause widespread economic hardship.

Technological advances in computer vision have made automated disease identification increasingly viable. Convolutional neural networks excel at extracting discriminative visual features from leaf imagery — features that are difficult for traditional image-processing pipelines to capture consistently [1]. Pre-trained architectures including ResNet50 [2], EfficientNet [3], and MobileNetV2 [4] enable practitioners to apply knowledge learned from large-scale datasets to specialised agronomic tasks through transfer learning, substantially reducing the volume of domain-specific annotated data required.

Despite this technical maturity, a practical gap persists between laboratory research and field adoption. Many rural farmers still depend on expert opinion or heuristic observation for disease identification — an approach that is slow, subjective, and increasingly unsuitable at the scale of modern agriculture. Existing software solutions are typically narrow in scope, delivering either detection or advisory services, but seldom both within a single accessible platform [8].

This paper presents AgroSense AI, a holistic platform designed to bridge this gap. The core contributions of this work are: (i) a controlled comparative evaluation of three CNN architectures trained under consistent experimental protocols; (ii) an end-to-end deployment integrating CNN inference with a production-ready web interface; (iii) a structured recommendation engine covering 62 disease conditions, populated with India-specific chemical and organic treatment guidelines; and (iv) a suite of supplementary decision-support tools, including a fertilizer calculator, crop risk predictor, and conversational AI assistant.

## II. LITERATURE REVIEW

The application of deep learning to plant pathology diagnosis gained widespread attention following the foundational study by Mohanty et al. [1], who demonstrated that CNN classifiers trained on PlantVillage imagery could achieve accuracies exceeding

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

99% under controlled conditions. This work motivated extensive follow-on research aimed at improving architecture efficiency and real-world robustness.

Ferentinos [6] conducted a systematic comparison between deep learning models and classical machine learning approaches — including support vector machines and decision trees — for crop disease prediction, establishing that deep networks offered measurably superior accuracy across multiple plant species. Too et al. (2019) extended this by fine-tuning multiple pre-trained models (ResNet, VGG, DenseNet) for multi-class leaf disease classification, demonstrating that parameter initialisation from large vision datasets can substantially accelerate convergence without sacrificing per-class accuracy.

Atila et al. [7] conducted a targeted assessment of EfficientNet variants on plant leaf disease benchmarks, reporting that the compound-scaled EfficientNet family achieves leading performance while requiring fewer trainable parameters than comparable architectures. Barbedo [10] surveyed image-based disease quantification techniques, cataloguing recurring challenges such as illumination inconsistency, overlapping symptom patterns, and background clutter that reduce model generalisation in outdoor settings.

At a broader scale, Kamilaris and Prenafeta-Boldu [8] reviewed forty deep learning applications across agricultural domains, noting that while CNNs dominate image-based tasks, most published evaluations are limited to controlled environments and that bridging the laboratory-to-field performance gap remains an open research problem. Singh et al. [9] subsequently released PlantDoc — a dataset curated from field conditions — and showed that models trained exclusively on laboratory images suffer substantial accuracy degradation under real-world imaging conditions, reinforcing the need for domain adaptation strategies.

## III. DATASET AND EXPLORATORY DATA ANALYSIS

### *A. PlantVillage Benchmark*

All experimental models in this study were trained and validated on the PlantVillage dataset [5], a publicly available collection of 54,305 RGB leaf images organised into 38 classes spanning 14 crop genera: Apple (4 pathology classes), Blueberry (1), Cherry (2), Corn/Maize (4), Grape (4), Orange (1), Peach (2), Bell Pepper (2), Potato (3), Raspberry (1), Soybean (1), Squash (1), Strawberry (2), and Tomato (10). Class-level sample counts vary considerably, from as few as 85 images for the Peach Healthy class to over 5,500 for Orange Huanglongbing, yielding an extreme class imbalance ratio of approximately 65:1.

### *B. Image Properties*

Images are uniformly sized at 256 × 256 pixels with a 1:1 aspect ratio, reflecting the standardised laboratory acquisition protocol. Analysis of mean pixel distributions across RGB channels confirmed that healthy foliage exhibits dominant green-channel intensity attributable to chlorophyll pigmentation, whereas diseased tissue displays characteristic channel-specific deviations dependent on pathogen type.

*Fig. 1. Uniform image dimension distribution across the PlantVillage dataset (all images 256 × 256 px).*

*Fig. 2. Per-channel RGB intensity histograms across the full dataset.*

*Fig. 3. Mean brightness per disease class (0–255 scale), illustrating inter-class luminance variation.*

*Fig. 4. Class-size distribution revealing significant long-tail imbalance across 38 categories.*

### *C. Preprocessing Pipeline*

A consistent four-stage preprocessing protocol was applied to all images prior to model ingestion: (1) spatial downsampling from 256 × 256 to 128 × 128 pixels via bilinear interpolation; (2) per-pixel normalisation scaling intensities from the integer range [0, 255] to the floating-point range [0, 1]; (3) mini-batch grouping with a batch size of 32 to enable efficient GPU memory utilisation; and (4) epoch-level training-set shuffling to prevent ordered-data memorisation.

### *D. Data Augmentation*

To promote model generalisation, online augmentation was applied stochastically during training using the Keras ImageDataGenerator API. Transformations included horizontal and vertical flipping, rotations within ±20°, translational shifts within ±10% of image dimensions, zoom perturbations within ±15%, and brightness scaling in the interval [0.8, 1.2]. Validation images were intentionally excluded from augmentation to maintain an unbiased assessment baseline.

## IV. METHODOLOGY

### *A. Transfer Learning Protocol*

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

A standardised transfer learning procedure was followed for all three architectures: (1) initialisation from ImageNet-pretrained weights with the classification head excluded; (2) addition of a custom head comprising global average pooling, a 128-unit dense layer with ReLU activation and 0.3 dropout, and a 38-class softmax output; (3) initial training with all backbone parameters frozen; (4) fine-tuning of the uppermost 20% of backbone layers at a reduced learning rate of $1 \times 10^{-5}$; (5) Adam optimiser with an initial learning rate of $1 \times 10^{-4}$; and (6) sparse categorical cross-entropy as the training objective.

### *B. ResNet50 Architecture*

ResNet50 [2], proposed by He et al., is a 50-layer residual network that introduced shortcut connections enabling gradient flow to bypass one or more convolutional blocks. This residual learning mechanism alleviates the vanishing gradient problem inherent in very deep networks, facilitating stable optimisation. The network contains approximately 25.6 million parameters and achieved a 3.57% top-5 error on ILSVRC 2015.

### *C. Training Hyperparameters*

*TABLE I: Training Hyperparameter Configuration (Uniform Across All Architectures)*

| Hyperparameter | Setting | Rationale |
|---|---|---|
| Input Resolution | 128 × 128 px | Memory-efficient while preserving disease-relevant texture |
| Batch Size | 32 | Compatible with GPU memory constraints for all models |
| Training Epochs | 15 | Sufficient for convergence under transfer learning |
| Optimiser | Adam | Adaptive gradient scaling; standard for fine-tuning |
| Initial Learning Rate | $1 \times 10^{-4}$ | Prevents destructive updates to pretrained weights |
| Loss Function | Sparse Cat. Cross-Entropy | Appropriate for multi-class integer-label targets |
| Train / Val Split | 80% / 20% | ~43,444 training / ~10,861 validation samples |

## V. SYSTEM DESIGN AND ARCHITECTURE

### *A. Platform Overview*

AgroSense AI follows a modular client-server design. The system comprises five integrated components: a browser-based responsive frontend, a Python/Flask REST API backend, a TensorFlow/Keras inference engine, a structured disease recommendation database, and a Google Gemini-powered conversational advisory module. This architecture allows each component to be updated or scaled independently, supporting long-term maintainability.

*TABLE II: AgroSense AI System Component Overview*

| Component | Technology Stack | Functional Role |
|---|---|---|
| Frontend | HTML5 / CSS3 / JavaScript | User interface, image upload, result rendering |
| API Backend | Python 3.11 / Flask 3.x | Request routing, validation, response assembly |
| Inference Engine | TensorFlow 2.x / Keras | CNN preprocessing and forward-pass execution |
| ML Backbone | ResNet50 (production) | Feature extraction from leaf images |
| Advisory Engine | Python dictionary (62 entries) | Disease-specific treatment and prevention lookup |
| AI Chatbot | Google Gemini Flash API | Open-ended agricultural question answering |
| Leaf Validator | OpenCV / HSV analysis | Rejection of non-leaf or irrelevant inputs |

*Fig. 5. DFD Level 0 — System context diagram depicting data flows between Farmer, Web Application, ML Models, Disease Database, and Gemini API.*

*Fig. 6. DFD Level 1 — Detailed internal processing modules within AgroSense AI.*

*Fig. 7. Use Case Diagram — Farmer interactions with the AgroSense AI platform.*

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

*Fig. 8. Activity Diagram — End-to-end workflow for disease detection, risk prediction, tools, and chatbot pathways.*

### *B. Inference Pipeline*

Each incoming prediction request traverses a sequential ten-stage pipeline: (1) multipart file reception and temporary storage; (2) image loading and conversion to RGB colour space; (3) leaf authenticity screening via green-pixel ratio analysis in the HSV colour space; (4) spatial downsampling to 128 × 128 pixels; (5) array conversion and batch-dimension insertion, yielding a tensor of shape (1, 128, 128, 3); (6) model inference generating a 38-element probability vector; (7) argmax class selection and confidence score extraction; (8) confidence thresholding — predictions below 60% confidence are returned as 'Uncertain'; (9) recommendation lookup by class label; and (10) JSON response serialisation and dispatch.

### *C. REST API Endpoints*

The Flask backend exposes three primary API routes: (i) POST /predict — receives a multipart image, returns a diagnosis JSON containing the disease label, confidence score, and full advisory entry; (ii) POST /api/fertilizer — accepts crop type, area, and growth stage parameters, and returns a computed NPK requirement with product recommendations; and (iii) POST /api/chat — forwards a user message to Gemini Flash and returns the model's natural-language response.

### *D. Disease Recommendation Engine*

The recommendation engine maintains structured advisory records for all 62 supported disease conditions. Each record includes the pathogen aetiology, disease severity classification, a brief diagnostic description, immediate remediation steps, chemical treatment options with dosage specifications, preventive cultural practices, organic intervention alternatives, and an estimated yield impact estimate. Advisory content was compiled based on guidelines issued by the Indian Council of Agricultural Research (ICAR) [16], the National Horticultural Research and Development Foundation (NHRDF), and the Directorate of Plant Protection, Quarantine and Storage (DPPQS) [17].

### *E. Supplementary Tools*

Beyond disease detection, AgroSense AI incorporates six agricultural decision-support modules: (1) Fertilizer Calculator — derives NPK requirements from crop species, field area, and phenological stage; (2) Pesticide Calculator — computes safe application dosage and dilution factors; (3) Profit Calculator — projects expected yield, revenue, and return on investment; (4) Irrigation Planner — generates stage-specific watering schedules; (5) Soil Health Analyser — interprets pH and macronutrient levels to indicate soil condition; and (6) Crop Risk Predictor — applies a machine learning model trained on over 100,000 field observations to estimate pathogen outbreak probability from soil chemistry inputs.

*Fig. 9. AgroSense AI main interface — disease detection upload panel and platform performance metrics.*

*Fig. 10. Smart Farm Tools Dashboard — six integrated agricultural calculators and planners.*

*Fig. 11. Soil-Based Disease Risk Predictor — ML-driven form accepting pH, N, P, K, organic matter, and EC readings.*

### *F. Gemini AI Chatbot Integration*

The conversational assistant is built on Google Gemini Flash, configured via a domain-specific system prompt that establishes the model's persona as an agricultural domain expert with knowledge of crop diseases, soil management, irrigation, pest control, and government subsidy schemes applicable to Indian farming. The chatbot complements CNN-based detection by handling free-form queries on subjects the structured database does not cover — such as market prices, crop selection, and climate-adaptive practices.

*Fig. 12. Chatbot welcome interaction demonstrating greeting and scope introduction.*

*Fig. 13. Chatbot response to a multi-crop yield query relevant to Indian farming contexts.*

## VI. EXPERIMENTAL RESULTS

### *A. ResNet50 Training Dynamics*

ResNet50 demonstrated characteristic rapid early-epoch accuracy gains attributable to the discriminative power of ImageNet-pretrained features. At the conclusion of the initial epoch, validation accuracy reached 91.75% — significantly above the 69.76% training accuracy recorded in the same epoch. This inversion is a well-documented transfer learning artefact in which frozen backbone features immediately confer strong generalisation before training-set memorisation occurs. Convergence continued steadily through epoch 15, at which point validation accuracy reached 95.88% with a validation loss of 0.1511.

*Fig. 14. ResNet50 training versus validation accuracy curves across 15 epochs, demonstrating consistent convergence.*

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

*TABLE III: ResNet50 Epoch-Level Training Metrics (Representative Epochs)*

| Epoch | Train Accuracy | Validation Accuracy | Validation Loss |
|---|---|---|---|
| 1 | 69.76% | 91.75% | 0.2543 |
| 3 | 88.14% | 93.94% | 0.1840 |
| 7 | 91.41% | 95.18% | 0.1531 |
| 10 | 92.50% | 95.42% | 0.1605 |
| 13 | 93.43% | 95.68% | 0.1510 |
| 14 | 93.46% | 96.04% | 0.1462 |
| 15 | 93.91% | 95.88% | 0.1511 |

### *B. Comparative Model Performance*

*TABLE IV: Comparative Validation Performance Across Three CNN Architectures*

| Architecture | Val. Accuracy (%) | Val. Loss | Parameters | Relative Training Cost |
|---|---|---|---|---|
| EfficientNetB3 | 96.60% | 0.1188 | ~12M | High |
| ResNet50 | 95.88% | 0.1511 | ~25.6M | Moderate |
| MobileNetV2 | 79.05% | — | ~3.4M | Low |

EfficientNetB3 achieved the top validation accuracy of 96.60%, establishing it as the strongest single-model option for server-deployed inference. ResNet50 achieved a comparable 95.88%, representing a margin of only 0.72 percentage points. MobileNetV2 recorded 79.05% — a gap of approximately 17 percentage points relative to EfficientNetB3 — consistent with the known accuracy-efficiency trade-off of lightweight architectures. Notably, MobileNetV2's 3.4 million parameters represent approximately one-seventh of ResNet50's parameter count, positioning it as the preferred option for on-device or bandwidth-constrained deployments.

### *C. Per-Class Classification Analysis — ResNet50*

The macro-averaged F1 score for ResNet50 reached 0.94 and the weighted-average F1 score reached 0.96 on the 10,861-image validation partition, consistent with the headline accuracy of 95.88%. Per-class analysis identified Potato Healthy (F1: 0.80, support: 25) and Grape Healthy (F1: 0.86, support: 85) as the lowest-performing categories, directly reflecting the class imbalance present in the training data. High-frequency classes such as Orange Huanglongbing and Tomato Yellow Leaf Curl Virus attained near-perfect recall and precision.

*TABLE V: ResNet50 Per-Class Classification Metrics (Selected Categories)*

| Disease Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Apple Scab | 0.97 | 0.96 | 0.97 | 126 |
| Apple Black Rot | 0.95 | 0.94 | 0.95 | 132 |
| Tomato Late Blight | 0.94 | 0.93 | 0.94 | 376 |
| Corn Common Rust | 0.98 | 0.97 | 0.97 | 219 |
| Tomato YLC Virus | 0.97 | 0.96 | 0.96 | 1,044 |
| Orange Huanglongbing | 0.99 | 0.98 | 0.98 | 1,106 |
| Potato Healthy | 0.82 | 0.78 | 0.80 | 25 |
| Grape Healthy | 0.88 | 0.85 | 0.86 | 85 |
| Macro Average | 0.94 | 0.94 | 0.94 | 10,861 |
| Weighted Average | 0.96 | 0.96 | 0.96 | 10,861 |

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

### *D. Confusion Matrix Analysis*

Visual inspection of the 38-class confusion matrix for ResNet50 revealed four principal patterns: (1) strong diagonal dominance corresponding to the overall 95.88% accuracy; (2) intra-species confusion between Tomato Early Blight and Tomato Late Blight, whose chlorotic and necrotic lesion patterns share morphological similarity; (3) elevated misclassification rates for minority classes with fewer than 100 training samples; and (4) near-perfect discrimination for Orange Huanglongbing and Tomato Yellow Leaf Curl Virus, both well-represented in the dataset.

*Fig. 15. Confusion matrix for ResNet50 on the 38-class PlantVillage validation set (10,861 images), showing strong diagonal dominance.*

## VII. SYSTEM IMPLEMENTATION

### *A. Technology Stack*

*TABLE VI: Full Technology Stack Summary*

| Layer | Technology | Purpose |
|---|---|---|
| API Framework | Flask 3.x | HTTP routing and middleware |
| Deep Learning Runtime | TensorFlow 2.x / Keras | Model loading and inference |
| Image Processing | OpenCV 4.x / Pillow 10.x | Pre- and post-processing |
| Numerical Backend | NumPy 1.26.x | Tensor manipulation |
| Cross-Origin Handling | Flask-CORS 4.x | Browser security compliance |
| Conversational AI | Google Gemini Flash | Agricultural advisory chatbot |
| Frontend | HTML5 / CSS3 / JavaScript | Responsive user interface |
| Execution Environment | Python 3.11 | Runtime platform |

### *B. Backend Implementation*

The Flask application is organised into five logical sections: model initialisation at startup (using tf.keras.models.load_model() with a global model reference to eliminate per-request loading latency), the disease recommendation dictionary, the /predict inference handler, the /api/fertilizer calculator endpoint, and the /api/chat Gemini relay. Storing the model as a global Python object reduces average inference latency by an order of magnitude compared with per-request loading, which would otherwise make real-time operation impractical.

### *C. Frontend Implementation*

The user interface is implemented as a single-page application with JavaScript managing five primary functions: uploadAndPredict() — constructs a multipart FormData payload and dispatches it to /predict; renderResults() — dynamically builds HTML to display the disease label, confidence bar, severity badge, and full advisory record; computeFertiliser() — collects crop-type and area inputs and invokes /api/fertilizer; sendChat() — manages the chatbot turn-taking loop with /api/chat; and validateLeafImage() — enforces file-type constraints client-side before upload.

### *D. Deployment Recommendations*

A production-grade deployment should incorporate: a WSGI server (Gunicorn or uWSGI) for concurrent request handling; an Nginx reverse proxy providing SSL termination, HTTP/2 support, and load balancing; cloud compute infrastructure (AWS EC2, Google Cloud Run) with GPU acceleration; model compression via TensorFlow Lite with INT8 quantisation for edge-device variants; and persistent storage (PostgreSQL or SQLite) for advisory content management, replacing the current in-memory dictionary.

## VIII. DISCUSSION

### *A. Architecture Selection Guidance*

Practitioners selecting an architecture should balance accuracy requirements against deployment constraints. EfficientNetB3 is optimal for server-hosted deployments where GPU resources are available, offering the highest accuracy with a moderate parameter footprint. ResNet50 is a robust second choice, delivering accuracy within 0.72 percentage points of EfficientNetB3 with

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

the benefit of a well-documented and extensively tested codebase. MobileNetV2 is most appropriate for mobile or edge scenarios, where its parameter count of 3.4 million enables efficient on-device inference despite its lower classification accuracy.

### *B. Effects of Class Imbalance*

The class imbalance inherent in PlantVillage had a measurable impact on minority-class performance. Potato Healthy (25 validation samples, F1: 0.80) and Peach Healthy (81 samples, F1: 0.86) showed the most pronounced degradation, underscoring that aggregate accuracy is an insufficient evaluation criterion for imbalanced detection tasks. Remediation strategies to explore in future iterations include class-conditional sample weights during training and synthetic minority oversampling via SMOTE or generative augmentation.

### *C. Efficacy of Transfer Learning*

The observation that validation accuracy exceeded training accuracy throughout the frozen-backbone phase — most strikingly for ResNet50 at epoch 1 (validation: 91.75%, training: 69.76%) — confirms that ImageNet-derived features are highly transferable to plant pathology imagery. This finding validates the transfer learning strategy as practical and resource-efficient for agricultural computer vision applications where large domain-specific annotated datasets are scarce.

## IX. LIMITATIONS

### *A. Laboratory Dataset Bias*

All training images were acquired under controlled illumination on uniform backgrounds, which does not reflect the variability encountered in field conditions — including natural lighting, occlusion by adjacent foliage, and soil contamination of leaf surfaces. Published studies on field-collected datasets such as PlantDoc [9] indicate substantial accuracy degradation when laboratory-trained models are tested on real-world imagery.

### *B. Limited Crop Coverage*

The current platform supports 14 crop genera, which excludes several economically significant Indian crops including groundnut, mustard, chilli, onion, and various legume species. Extending coverage to these crops would require sourcing or collecting annotated field image datasets specific to these species.

### *C. Static Advisory Content*

Treatment and prevention guidance is stored in a static Python dictionary that requires code modification and redeployment for any content update. Integration with a content management system or relational database would enable dynamic updates by agronomists without requiring software engineering involvement.

### *D. Single-Image Diagnostic Paradigm*

The system analyses individual leaf photographs, which may not adequately represent the spatial heterogeneity of symptom distribution across a plant or across phenological stages. Longitudinal multi-image analysis or attention-based region proposal networks may provide richer diagnostic context.

### *E. Absence of Environmental Context*

The platform does not incorporate real-time environmental telemetry — such as temperature, relative humidity, or cumulative rainfall — despite these variables being critical modulating factors for pathogen development. Their absence limits the platform's capacity for proactive outbreak risk forecasting.

## X. FUTURE WORK

### *A. Domain Adaptation for Field Imagery*

Priority future work includes fine-tuning models on field-collected images from representative Indian agricultural regions, using domain adaptation techniques — such as adversarial domain alignment or feature distribution matching — to reduce the distributional shift between laboratory and field imaging conditions.

### *B. Extended Species Coverage*

Collaboration with ICAR, state agricultural universities, and public image repositories will facilitate the acquisition of annotated datasets for additional Indian crop species, broadening the platform's practical utility to a wider range of farmers.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

### *C. Mobile Edge Deployment*

Converting MobileNetV2 to TensorFlow Lite format with INT8 quantisation will enable offline inference on Android smartphones, addressing connectivity barriers in rural areas lacking reliable internet access.

### *D. Advanced Augmentation Strategies*

Training with Mixup, CutMix, or AugMix augmentation has demonstrated consistent generalisation improvements on standard visual benchmarks. Evaluating these techniques on PlantVillage and field-collected data represents a tractable near-term improvement.

### *E. Ensemble and Explainability Methods*

Combining ResNet50 and EfficientNetB3 predictions through probability-averaging ensembles may exceed the accuracy of either individual model. Additionally, implementing Grad-CAM saliency visualisations will provide interpretable evidence of which leaf regions influence each diagnosis — supporting expert validation and increasing farmer confidence in system outputs.

### *F. Multilingual Interface and Weather Integration*

Expanding the interface to support Hindi, Gujarati, Marathi, Telugu, and Tamil will substantially improve accessibility for non-English-speaking farming communities. Coupling this with real-time weather API integration — linking local temperature and humidity data to disease risk thresholds — will enable context-aware, proactive advisory outputs.

## XI. CONCLUSION

This paper has presented AgroSense AI — a production-oriented crop disease detection and advisory platform designed specifically for the requirements and constraints of Indian smallholder agriculture. Controlled comparative evaluation of three CNN architectures — ResNet50, EfficientNetB3, and MobileNetV2 — trained under identical conditions on the PlantVillage benchmark established that EfficientNetB3 achieves the highest classification accuracy at 96.60%, with ResNet50 performing within a 0.72 percentage-point margin at 95.88%. MobileNetV2, with a compact parameter footprint of 3.4 million, offers a viable path to edge and mobile deployment despite its 79.05% accuracy.

The AgroSense AI platform distinguishes itself from prior work by delivering a fully integrated solution: CNN-based disease detection, a 62-class structured advisory engine with India-specific treatment protocols, six agricultural decision-support tools, and a conversational Gemini AI assistant — all accessible through a standard web browser without requiring specialist hardware or software. The platform's deployment as a functional web application demonstrates that high-accuracy deep learning research can be translated into accessible tools with direct utility for field-level agricultural management.

Future development will prioritise field-image domain adaptation to close the laboratory-to-farm accuracy gap, expansion of crop coverage to include additional Indian species, offline mobile deployment via TensorFlow Lite quantisation, and the integration of real-time environmental sensor data for proactive disease risk forecasting.

## REFERENCES

[1] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Deep learning for image-based plant disease detection," Frontiers in Plant Science, vol. 7, article 1419, Sep. 2016.

[2] 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), Las Vegas, NV, USA, 2016, pp. 770–778.

[3] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. 36th Int. Conf. Machine Learning (ICML), PMLR vol. 97, 2019, pp. 6105–6114.

[4] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE CVPR, Salt Lake City, UT, USA, 2018, pp. 4510–4520.

[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 preprint arXiv:1511.08060, Nov. 2015.

[6] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Computers and Electronics in Agriculture, vol. 145, pp. 311–318, Feb. 2018.

[7] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using EfficientNet deep learning model," Ecological Informatics, vol. 61, art. 101182, Jan. 2021.

[8] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," Computers and Electronics in Agriculture, vol. 147, pp. 70–90, Apr. 2018.

[9] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: A dataset for visual plant disease detection," in Proc. 7th ACM IKDD CoDS and 25th COMAD, Hyderabad, India, 2020, pp. 249–253.

[10] J. G. A. Barbedo, "Digital image processing techniques for detecting, quantifying and classifying plant diseases," SpringerPlus, vol. 2, no. 1, art. 660, Dec. 2013.

[11] M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep learning for tomato diseases: Classification and symptoms visualization," Applied AI, vol. 31, no. 4, pp. 299–315, 2017.

[12] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A comprehensive survey of image augmentation techniques for deep learning," Pattern Recognition, vol. 137, art. 109347, May 2023.

[13] P. Tm, T. Pranathi, K. SaiAshritha, N. B. Chittaragi, and S. G. Koolagudi, "Tomato leaf disease detection using convolutional neural networks," in Proc. 2018 Eleventh Int. Conf. Contemporary Computing (IC3), Noida, India, pp. 1–5. IEEE, 2018.

[14] S. Arivazhagan, R. N. Shebiah, S. Ananthi, and S. V. Varthini, "Detection of unhealthy region of plant leaves using texture features," Agricultural Engineering International: CIGR Journal, vol. 15, no. 1, pp. 211–217, 2013.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[16] Indian Council of Agricultural Research (ICAR), Package of Practices for Major Crops. New Delhi: ICAR Publications, 2023.

[17] Directorate of Plant Protection, Quarantine and Storage (DPPQS), List of Registered Pesticides in India. New Delhi: Ministry of Agriculture and Farmers Welfare, Government of India, 2024.

[18] Food and Agriculture Organisation of the United Nations (FAO), The State of Food and Agriculture: Making Agrifood Systems More Resilient. Rome: FAO, 2021.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, Sep. 2014.

[20] 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.