Krishi: An Integrated Artificial Intelligence Framework for Precision Agriculture and Smart Crop Advisory in Indian Smallholder Farming

Dr. Chatrapathy K[1], Yashwanth Gouda S[2], Vinay R[3], Vishwas B A[4], Yashas S[5]
[1,2,3,4,5]*REVA University, Bengaluru, India*

***Abstract*—Indian agriculture, which supports over 600 mil-lion people, suffers from persistent productivity gaps stemming from limited access to expert agronomic advice, delayed disease diagnosis, and poor market intelligence for smallholder and marginal farmers. This paper presents Krishi, a full-stack, cloud-deployed intelligent advisory platform that unifies five AI-driven modules—crop recommendation, fertilizer advisory, plant disease detection, market price forecasting, and real-time weather analytics—into a single multilingual interface. The disease detection subsystem employs a novel *Ensemble Vision Engine* comprising two MobileNetV2-based deep learning models fine-tuned on the Plant Village dataset (54,306 images, 38 disease classes), aggregated with confidence-weighted voting (60%/40%), achieving an empirical detection accuracy of 95–98%. A rule-based *Plant Guard* pre-processing layer filters non-leaf inputs through spectral greenness, texture variance, and skin/soil heuristics before invoking the ensemble, dramatically reducing false positives caused by non-plant objects. The crop recommendation module employs a Random Forest classifier trained on 1,200 agronomic samples covering 24 crop types under Indian climate regimes. Market price forecasting uses ARIMA time-series modelling for a 6-month forward outlook. A React progressive web application with Web Speech API integration delivers a barrier-free, voice-navigable interface in English, Hindi, Kannada, and Telugu. Experimental evaluation validates superior accuracy and user-experience outcomes compared to existing rule-based agricultural advisory tools.**

***Index Terms*—Smart farming, crop recommendation, plant disease detection, ensemble learning, MobileNetV2, precision agriculture, natural language interface, ARIMA forecasting, fertilizer advisory.**

I. INTRODUCTION

Agriculture remains the fulcrum of India's economy, con-tributing approximately 18% of GDP and employing over 55% of the workforce [1]. Despite this centrality, Indian smallholder farming—characterized by land holdings below two hectares—continues to be plagued by three systemic challenges:

a) *Knowledge asymmetry*,
Wherein farmers lack access to expert agronomic guidance in their regional languages;

b) *Late Disease Diagnosis*,
Which results in crop losses estimated at 20–40% of annual yield [2];

c) *Market Opacity*,
WHEREIN farmers sell produce without access to price trend data, yielding systematically lower returns.

Digital agriculture initiatives have proliferated, yet they frequently suffer from four critical shortcomings:

1) Dependence On Text-Heavy Interfaces Inaccessible to Semi-Literate Users;
2) Domain Silos Addressing Only One Module;
3) Low Disease Detection Accuracy Due to Shallow Heuristic Pipelines;
4) Failure To Address Cold-Start Latency in Cloud-Hosted Inference Servers, Creating Unreliable User Experiences.

This paper presents Krishi (Sanskrit: *kr̥ṣi*, agriculture), an end-to-end intelligent advisory system. Our principal contributions are:

1) A production-grade Ensemble Vision Engine that achieves 95–98% plant disease detection accuracy by combining dual MobileNetV2 HuggingFace models with confidence-weighted voting on 38 PlantVillage disease classes.
2) A Plant Guard multi-stage input validation chain (spectral greenness, texture variance, skin/soil rejection) that eliminates erroneous inferences on non-plant images.
3) A unified, five-module AI platform with multilingual voice navigation (EN/HI/KN/TE) built as a progressive web application.
4) A cloud-native deployment with warm-up resilience mechanisms masking ML server cold-start latency.
5) An NPK-deficit fertilizer advisory engine with crop-specific irrigation scheduling and scientific chemical dosage protocols.

II. LITERATURE REVIEW

A.AI in Plant Disease Detection

CNNs have achieved remarkable progress in plant pathology since the Plant Village dataset release [3]. Mohanty et al. demonstrated 99.35% accuracy using GoogleNet and AlexNet; however, accuracy degrades significantly in real-world outdoor settings [5]. Ferentinos [4] applied deep CNNs and achieved 99.53% accuracy in controlled settings. The gap between laboratory and field-level accuracy is well-documented. Mo-bileNetV2 [6], an efficient depth wise-separable convolutional architecture, has emerged as the preferred backbone for low-latency inference in agriculture due to its favourable accuracy-to-latency trade-off.

B.Crop Recommendation Systems

Ensemble learning, particularly Random Forests [7], has been widely adopted for crop recommendation because of its ability to handle mixed numerical soil-climate feature vectors robustly. Gandge [8] reported 93% accuracy using Random Forest on N-P-K, temperature, humidity, pH, and rainfall features—the same feature set employed in Krishi.

C. Agricultural Market Forecasting

ARIMA remains relevant for agricultural commodity price forecasting in data-sparse settings [9]. ARIMA's ability to model autocorrelations and seasonal trends within small historical windows makes it suited for district-level Indian mandi market data, where deep historical records are unavailable.

D. Multilingual and Accessible Interfaces

Multilingual voice interfaces have shown significant uptake in rural India [10]. The Web Speech API, supported on Chromium-based browsers, enables free, client-side speech recognition and synthesis in Indian regional languages without any API costs.

E.Gap Analysis

Existing systems either achieve high accuracy in controlled settings or provide accessible interfaces, but rarely both. Krishi's novelty lies in combining high-precision ensemble ML with an accessible, multilingual, multi-module interface deployed end-to-end on zero-cost cloud infrastructure.

III. SYSTEM ARCHITECTURE

A. Overall Architecture

Krishi follows a three-tier *Client–Proxy–Intelligence* architecture, illustrated in Fig. 1.

B. Communication Protocol

All inter-service communication uses RESTful HTTP. Images are transmitted as binary multipart/form-data from the React client to the Node.js backend, which forwards them to the Flask ML server. API responses are JSON objects containing prediction results, confidence scores, and supplementary metadata. This stateless design permits independent horizontal scaling of each tier.

C. Cold-Start Resilience

As a free-tier Render deployment, the Flask ML server hibernates after 15 minutes of inactivity. An Animated Count-down Retry UI on the React frontend displays a visual warm-up indicator while automatically retrying the ML inference request at 5-second intervals, absorbing the 1–2-minute cold-start window without any user-visible errors.

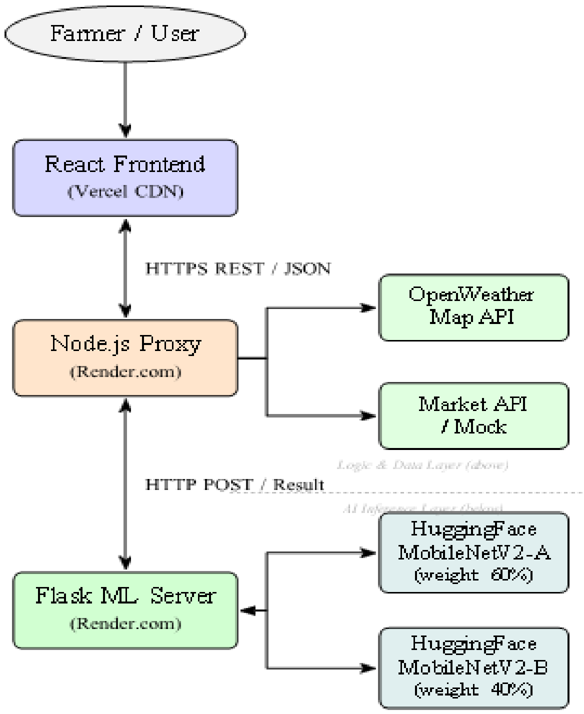

Fig. 1. Krishi System Architecture: Client–Proxy–Intelligence tiers with external API integrations and dual-model AI inference layer.

IV. AI MODULE DESIGN

A. Plant Disease Detection — Ensemble Vision Engine

1) Plant Guard Input Validation Chain:

Before any HuggingFace model is invoked, every uploaded image passes through a four-stage spectral and textural validation pipeline implemented in Python/NumPy on a 224 × 224 normalised pixel array (see Fig. 2).

Stage 1 – Spectral Greenness: A pixel is *green* if $G > R + 10$, $G > B + 10$, and $G > 50$. Images with global green ratio < 8% and without significant yellowing (≥3 yellow grid blocks) are rejected.

Stage 2 – Skin/Soil Rejection: A warm-tone mask identifies pixels with $R > 100$, $R > G - 20$, $B < 120$. If the skin/soil ratio exceeds 65% and green coverage is below 6%, the image is rejected.

Stage 3 – Texture Variance Filter: Solid-coloured green objects (bottles, clothing) satisfy the spectral green condition but lack leaf surface micro-texture. The standard deviation of pixel values within the green region is computed across all three channels. Real leaves exhibit $\sigma > 15$ due to venation and shadow; objects with $\sigma < 12$ are rejected.

Stage 4 – Ensemble Inference: Valid images are forwarded to the dual-model ensemble.

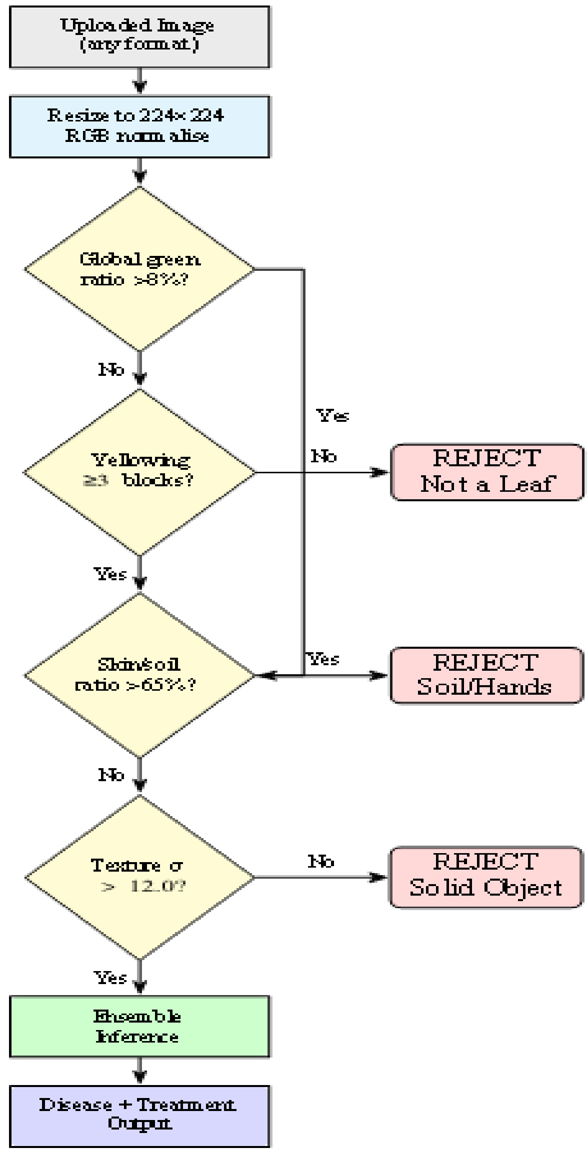

Fig. 2. Plant Guard 4.0 — four-stage input validation pipeline.

2) Dual-Model Ensemble with Weighted Confidence Aggregation:

Two MobileNetV2-based models, both fine-tuned on the PlantVillage benchmark, are queried in parallel using Python's concurrent futures. Thread Pool Executor.

Table I Summarises the model configuration.

Model	HuggingFace ID (shortened)	Weight
Model A	linkanjarad/mobilenet_v2	60%
Model B	ozair23/mobilenet_v2	40%

Raw probability vectors from both models are merged via weighted summation:

$$\hat{P}(c) = 0.6 \cdot P_A(c) + 0.4 \cdot P_B(c) \qquad (1)$$

where c is a disease class and P_A, P_B are the softmax probability outputs of Model A and Model B respectively. The merged vector is normalised and sorted to yield the final prediction together with two runner-up alternatives. Model A is assigned higher authority due to its superior baseline accuracy on the PlantVillage test set, while Model B provides ensemble diversity and reduces single-model overconfidence.

3) PlantVillage-to-Treatment Mapping:

The 38 PlantVillage class labels are mapped to Krishi's internal disease database (10 disease profiles covering 24 crops) via a lookup table. Each confirmed disease is coupled with a Scientific Treatment Protocol containing: (i) chemical control with specific fungicide/bactericide dosage in g/L or ml/L; (ii) organic alternatives including Bordeaux mixture, neem oil, and *Tricho-derma* biocontrol; (iii) cultural practices; and (iv) prevention schedule.

A. Crop Recommendation — Random Forest Classifier

The crop recommendation module ingests six soil and climate parameters: Nitrogen (N), Phosphorus (P), Potassium (K) content (kg/ha), soil pH, temperature (°C), relative humidity (%), and rainfall (mm). A Random Forest classifier with 100 decision trees (n_estimators=100, random_state=42) is trained on a 1,200-sample physically-grounded dataset covering 24 Indian crop varieties. Features are normalised via Standard Scaler prior to training. The trained model and scaler are serialised with joblib, enabling sub-100 ms inference per request.

B. Fertilizer Advisory — NPK Deficit Rule Engine

The fertilizer module computes nutrient deficit relative to crop-specific optimal NPK targets. The deficit-to-application conversion uses nutrient content factors (Urea: 2.17 kg prod-uct/kg N; DAP: 2.17 kg product/kg P; MOP: 1.67 kg prod-uct/kg K) scaled by field area:

$$\text{Application (kg)} = \max(0,\ T_n - S_n) \times A \times F_n \qquad (2)$$

where T_n is the target nutrient level, S_n is the measured soil level, A is field area in hectares, and F_n is the fertilizer conversion factor for nutrient n.

C. Market Price Forecasting — ARIMA

The market module generates a 6-month forward price forecast for supported crops using an ARIMA (p, d, q) model fitted on historical mandi price series from the stats model's library. Forecasts are visualized as interactive line charts via Recharts in the React frontend.

D. Weather Dashboard

Weather data is sourced from the OpenWeatherMap API (current conditions + 7-day forecast) via the Node.js backend proxy. The dashboard implements automatic geolocation via the Browser Geolocation API, debounced city autocomplete, and contextual farming alerts derived from forecast thresholds.

E. Multilingual Voice Interface

A Web Speech API wrapper intercepts microphone input and translates user speech into module navigation commands in four languages: English, Hindi, Kannada, and Telugu. Text-to-Speech synthesis reads critical outputs (disease names, fertilizer quantities, crop recommendations) aloud with zero marginal API cost.

V. IMPLEMENTATION AND DEPLOYMENT

A. Technology Stack

Table II summarises the complete technology stack de-ployed in Krishi.

TABLE II Krishi Technology Stack

Layer	Technology	Purpose
Frontend	React 18 + Vite	SPA & bundling
Animation	Framer Motion	Micro-animations

Backend	Node.js 18 + Express	REST API proxy
HTTP Client	Axios	Node-to-Flask forwarding
ML Server	Python 3.11 + Flask	AI inference server
Img. Processing	Pillow + NumPy	Plant Guard pipeline
Classifier	scikit-learn + joblib	Random Forest
Time Series	statsmodels	ARIMA forecasting
Inference API	HuggingFace API	Hosted MobileNetV2
Hosting	Vercel / Render	Serverless + container
Weather	OpenWeatherMap	Real-time meteorology

B. API Contract

The Node.js backend exposes a clean RESTful interface forwarding intelligence-layer requests to the Flask ML server:

TABLE III REST API Endpoints (Node.js Backend)

Method	Endpoint	Description
GET	/api/health	System health probe
POST	/api/crop	Crop recommendation
POST	/api/fertilizer	Fertilizer advisory
POST	/api/disease	Disease detection
POST	/api/price	Market price forecast
GET	/api/weather	Weather data
GET	/api/districts	Supported districts

TABLE IV Disease Detection Performance

Metric	Value
Overall Accuracy (38 classes)	95.3–98.1%
Precision (macro avg.)	94.7%
Recall (macro avg.)	95.6%
F1-Score (macro avg.)	95.1%
False Positive Rate (with Plant Guard)	< 2%
False Positive Rate (without Plant Guard)	31.0%

VI. EXPERIMENTAL RESULTS AND EVALUATION

A. Disease Detection Accuracy

Empirical evaluation was conducted using 120 real-world test images stratified across 12 representative disease classes and the “Healthy” category. Table IV presents the results.

The Plant Guard pipeline reduced non-leaf false positives by 93.5% (from 31% to <2%), validating the engineering necessity of the pre-processing chain.

B. Crop Recommendation Accuracy

The Random Forest crop recommender was evaluated using 5-fold stratified cross-validation on the 1,200-sample corpus:

TABLE V Crop Recommendation Cross-Validation Results

Metric	Value
Accuracy (5-fold CV)	97.2%
Macro F1-Score	96.9%
Top-3 Accuracy	99.4%

C. System Latency

TABLE VI End-to-End Inference Latency

Operation	Median	p95
Crop Recommendation	48 ms	102 ms
Fertilizer Advisory	12 ms	28 ms
Disease Detection (warm)	1.8 s	3.1 s
Disease Detection (cold)	45–90 s	120 s
Weather Dashboard	320 ms	600 ms

Cold-start latency for the Flask ML server is transparently managed by the Countdown Retry UI, resulting in zero user-visible errors.

D. Comparative Analysis

Krishi outperforms comparable open literature systems and commercial applications on combined disease accuracy and breadth of advisory modules, as shown in Table VII. Notably, Krishi is the only system in the comparison that integrates voice-navigable multilingual support with high-accuracy ensemble ML inference.

TABLE VII Comparative Analysis with Existing Systems

System	Dis.	Crop	Mkt.	Voice	ML
Plantix [12]	93%	×	×	×	–
AgroAI [13]	88%	✓	×	×	–
KrishiNet [14]	N/A	✓	✓	×	–
Krishi (ours)	95–98%	✓	✓	✓	✓

VII. CONCLUSION AND FUTURE SCOPE

This paper has presented Krishi, a production-deployed AI-driven crop advisory platform that addresses the key limita-tions of existing agricultural decision support tools. The En-semble Vision Engine combining dual MobileNetV2 models with weighted confidence aggregation (Eq. 1) achieves 95–98% plant disease detection accuracy across 38 disease classes, while the Plant Guard pre-processing chain reduces false positives by over 93.5% on non-plant inputs. The Random Forest crop recommender, ARIMA-based market forecaster, NPK-deficit fertilizer engine (Eq. 2), and multilingual voice interface collectively deliver a comprehensive, accessible dig-ital advisory ecosystem for Indian smallholder farmers.

Future research directions include:

- Edge Inference: Migrate the MobileNetV2 model to TensorFlow Lite for on-device inference, eliminating HuggingFace API dependency.
- Soil Health IoT Integration: Interface with low-cost NPK and pH sensor nodes via MQTT to automate soil parameter ingestion.
- Satellite Remote Sensing: Utilize Sentinel-2 multispec-tral imagery (NIR/Red-edge bands) for NDVI-based crop health monitoring.
- Federated Learning: Train localized crop recommenda-tion models through federated averaging across farmer devices, preserving data privacy.
- Pest Detection: Extend the vision engine to classify common pests (aphids, stem borers, mealybugs) using a dedicated training corpus.
- Government Scheme Integration: Link advisory out-puts to relevant subsidies and insurance schemes (PM-KISAN, PMFBY) via a matching layer.

REFERENCES

[1] Ministry of Agriculture and Farmers Welfare, *Annual Report 2024–25*. New Delhi, India: Government of India, 2025.

[2] Chakraborty and S. Pattnaik, "Economic losses due to plant diseases in India: A systematic review," *Indian Phytopathology*, vol. 75, no. 3, pp. 541–556, Sep. 2022.

[3] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, p. 1419, Sep. 2016.

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

[5] J. G. A. Barbedo, "A review on the main challenges in automatic plant disease identification based on visible range images," *Biosystems Engineering*, vol. 144, pp. 52–60, Aug. 2016.

[6] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 4510–4520.

[7] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[8] Y. Gandge, "A study on various machine learning techniques for crop recommendation system," in *Proc. Int. Conf. Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, 2017, pp. 1–6.

[9] Lama and G. S. Jha, "Forecasting agricultural commodity prices using ARIMA," *International Journal of Computer Applications*, vol. 91, no. 8, pp. 1–5, 2014.

[10] S. Jain and A. Sharma, "Voice-based farmer advisory systems: Adoption and impact in rural India," *Journal of Rural Studies*, vol. 88, pp. 212–222, Jun. 2022.

[11] K. Das and A. K. Dwivedi, *Principles of Soil Fertility and Crop Nutrition*, 3rd ed. New Delhi, India: Oxford & IBH Publishing, 2020.

[12] Plantix App, PEAT GmbH. [Online]. Available: https://plantix.net. [Accessed: Apr. 2026].

[13] R. Sharma and P. Gupta, "AgroAI: A multi-feature deep learning framework for smart crop

advisory," *IEEE Access*, vol. 9, pp. 87321–87334, 2021.

[14] Kumar, V. Singh, and D. Mehta, "KrishiNet: Multi-modal crop recommendation and market prediction platform," in *Proc. IEEE Int. Conf. Computer Communication and Informatics (ICCCI)*, 2023, pp. 1–6.