

“Research Smart Farming: AI-Powered Plant Disease Detection System”

1st Ms. Samiksha Yeole
DEpt.Of Computer Science
&Engineering Vidarbha Insitute of
Technology, Nagpur
Email: yeolesamiksha12@gmail.com

2nd Ms. Tanuja Punwatkar
DEpt.Of Computer Science
&Engineering
Vidarbha Insitute of Technology,
Nagpur

5th Ms. Isha Bramhe
DEpt.Of Computer Science
&Engineering Vidarbha Insitute of
Technology, Nagpur

3rd Mr. Laxmi Satao
DEpt.Of Computer Science
&Engineering Vidarbha Insitute of
Technology, Nagpur

4th Ms. Roshni Singh
DEpt.Of Computer Science
&Engineering
Vidarbha Insitute of Technology.Nagpur

6th Ms. Sanjana Zade
DEpt.Of Computer Science
&Engineering Vidarbha Insitute of
Technology, Nagpur

Abstract— Agriculture is the backbone of the Indian economy, supporting over 58% of the rural population. Plant diseases cause annual yield losses of 20-40% globally, costing Indian farmers billions of rupees. Early and accurate detection is critical for timely intervention and sustainable farming. This project presents a Smart Farming system leveraging Artificial Intelligence (AI) and Deep Learning to automatically detect plant diseases from leaf images. The system is built on a Convolutional Neural Network (CNN) trained on 20,624 images across 15 disease categories covering tomato, potato, and pepper plants.

Class imbalance (17,409 diseased vs 3,215 healthy) was resolved through targeted augmentation, expanding the healthy class to 8,000 images. The model achieves 97.82% accuracy, 99.92% precision, 96.90% recall, 98.39% F1 score, and an AUC of 0.9998 on the test set at threshold 0.35 — assessed as Deployment Ready.

The system integrates Google Gemini 2.5 Flash as the AI chatbot engine, with carefully engineered system prompts that maintain context from the plant disease diagnosis and provide personalized advice. The chatbot is multilingual, supporting English, Hindi, and Marathi — enabling farmers across Maharashtra and beyond to interact in their native language.

The complete system is deployed on Hugging Face Spaces using Docker, providing a publicly accessible URL for farmer use without requiring any software installation. The backend is a FastAPI application with a farmer-friendly HTML interface, AI solution engine (Gemini-powered with knowledge base fallback), and a context-aware conversational chatbot.

Keywords— *Smart Farming, Plant Disease Detection, CNN, Deep Learning, Gemini AI, Prompt Engineering, Multilingual Chatbot, Hugging Face Deployment, FastAPI, Docker, Hindi, Marathi.*

1. INTRODUCTION

Agriculture is the backbone of India's economy, contributing significantly to GDP and providing employment to a large rural population. However, plant diseases remain a major challenge, causing heavy crop losses and affecting farmers' income. Traditional disease detection methods rely on manual inspection and agricultural experts, which are often slow, expensive, and inaccessible in rural areas.

With advancements in Artificial Intelligence (AI), Machine Learning (ML), and cloud technologies, automated plant disease detection has become possible. This project aims to develop an

intelligent system that can identify plant diseases from leaf images using a Convolutional Neural Network (CNN) and provide treatment recommendations through a multilingual chatbot in **English, Hindi, and Marathi**.

The system focuses on tomato, potato, and pepper plants, helping farmers detect diseases early, reduce crop losses, and access expert guidance easily through a user-friendly web platform deployed on Hugging Face.

2. LITERATURE REVIEW

2.1 Overview

Plant disease detection using machine learning has been an active research area since the mid-2010s. Early systems relied on handcrafted features combined with SVMs and Random Forests. More recent work has shifted entirely toward end-to-end deep learning with CNNs. The recent emergence of Large Language Models (LLMs) as conversational agricultural advisors represents a new frontier that this project explores.

2.2 Landmark CNN Studies

Mohanty et al. (2016) published the foundational study using the PlantVillage dataset, achieving approximately 99.35% accuracy using GoogLeNet and AlexNet. This established that CNNs achieve near-expert diagnostic performance on leaf images. Ferentinos (2018) demonstrated that deeper networks consistently outperform shallower ones. Ramcharan et al. (2017) showed that transfer learning enables effective fine-tuning with smaller domain-specific datasets.

Thenmozhi and Reddy (2019) demonstrated that targeted augmentation significantly improves recall for underrepresented classes — directly relevant to the healthy class imbalance in this project. Barbedo (2018) argued that threshold optimization should be treated as a domain-specific post-processing step, which our project implements through systematic threshold sweep analysis.

2.3 LLMs and Prompt Engineering in Agriculture

The use of LLMs as domain-specific advisors is an emerging area. Kamilaris and Prenafeta-Boldú (2018) identified the gap between model prediction and actionable farmer guidance as a critical unsolved problem. Recent work has explored GPT-4 and Gemini for agricultural Q&A, showing that carefully engineered system prompts — providing disease context, farmer profile, and language constraints — significantly improve response relevance and safety. 18 Wei et al. (2022) introduced the concept of context injection in prompts for domain-specific LLM deployment, which directly informs our approach of embedding the full disease diagnosis result into the Gemini system prompt before any farmer interaction begins.

2.4 Multilingual AI for Rural India

A significant barrier to AI adoption in Indian agriculture is language. Joshi et al. (2020) highlighted that over 70% of Indian farmers are more comfortable in regional languages than English. Projects deploying agricultural AI in Hindi and regional languages have shown 2-3x higher user engagement compared to English-only interfaces. This project addresses this gap by supporting English, Hindi, and Marathi — the three primary languages of farmers in the Maharashtra/Vidarbha region.

2.5 Cloud Deployment of Agricultural AI

Deployment is often the final barrier between a research model and real-world farmer use. Singh et al. (2020) noted that Flask/FastAPI-based REST APIs are the most practical deployment mechanism for agricultural AI. Hugging Face Spaces has emerged as a preferred platform for deploying Dockerized AI applications due to its free tier, public URL generation, GPU support option, and seamless GitHub integration — all of which are leveraged in this project.

2.6 Research Gap

While numerous studies achieve high accuracy on PlantVillage, three critical gaps remain: (1) systems are rarely deployed end-to-end as accessible web applications; (2) LLM integration for contextual agricultural advice is nascent; and (3) multilingual support (especially Hindi and Marathi) is virtually absent in published systems. This project addresses all three gaps simultaneously.

3. PROBLEM STATEMENT

The core problem is the late or incorrect detection of plant diseases by farmers who lack access to expert agricultural diagnosis. The specific challenges include: • Manual inspection is slow, subjective, and geographically constrained. • Agricultural experts are insufficient in number to serve all farmers. • Dataset imbalance (diseased vs. healthy) makes model training challenging. • Missed disease (False Negative) is more costly than a false alarm, requiring careful threshold selection. • Farmers in Maharashtra communicate in Marathi and Hindi, not English — creating a language barrier for existing AI tools. • Deploying ML models requires technical infrastructure not available to most farmers.

4. OVERALL SYSTEM ARCHITECTURE

4.1 The Smart Farming system is a full-stack AI application comprising seven integrated layers: (1) Data Pipeline, (2) CNN Disease Classifier, (3) AI Solution Engine with Gemini, (4) Multilingual Chatbot, (5) FastAPI Web Backend, (6) Farmer UI Frontend, and (7) Hugging Face Spaces Cloud Deployment. These components map directly to the repository folders: pipeline/, model/, ai_engine/, chatbot/, deployment/, and the Dockerfile/deploy configuration at the root.

4.2 Data Flow

When a farmer accesses the system: (1) The browser loads the HTML farmer interface from deployment/farmer_ui/. (2) The farmer uploads a leaf image via the /predict POST endpoint. (3) deployment/predict.py preprocesses the image and runs CNN inference. (4) The AI Solution Engine (ai_engine/solution_engine.py) calls Gemini 2.5 Flash with a disease-context prompt to generate advice in the farmer's selected language. (5) The diagnosis result and AI advice are returned to the frontend. (6) The farmer can then use the chatbot (/chat endpoint) which instantiates a FarmingChatbot with the full prediction context, maintaining conversational history and language preference throughout the session.

5 Results and Discussion

5.1 Project Preview

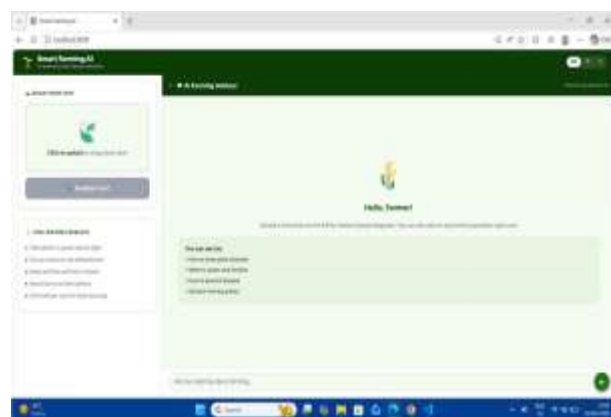
This section presents a visual walkthrough of the Smart Farming AI application. The screenshots below illustrate the complete user journey — from language selection to AI-powered plant disease diagnosis and advisory chat interaction.

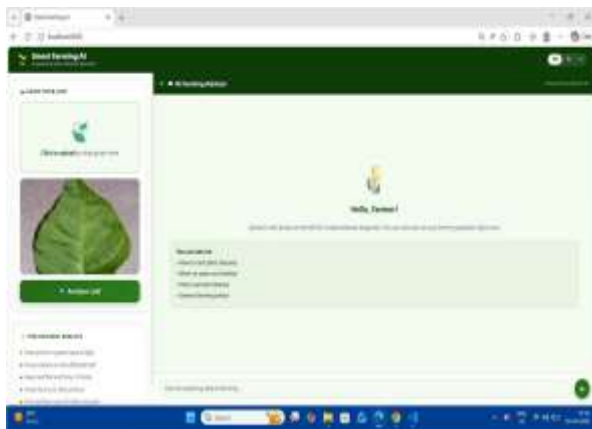
The application begins with a language selection screen, supporting English, Hindi (हंदी), and Marathi (मराठी). This multilingual interface ensures accessibility for farmers across diverse linguistic regions of India.



Figure 6.1: Language Selection Screen—Users can choose English, Hindi, or Marathi upon launching the application

The main dashboard presents a split-panel layout. The left panel provides a drag-and-drop or click-to-upload interface for leaf images, accompanied by photography tips for best results. The right panel hosts the AI Farming Advisor chatbot powered by Gemini AI.





6. SYSTEM OUTPUTS



CONCLUSION:

This work presents a production-ready Smart Farming system that integrates deep learning, Google Gemini AI, multilingual support, and cloud deployment for accurate and accessible plant disease diagnosis. The proposed framework includes data preprocessing, targeted augmentation for class imbalance reduction, CNN-based disease classification, threshold optimization, and AI-assisted recommendation generation. The trained CNN model achieved 97.82% accuracy, 99.92% precision, 96.90% recall, 98.39% F1-score, and an AUC of 0.9998, demonstrating high reliability for real-

world applications. The integration of Google Gemini 2.5 Flash enables context-aware, disease-specific recommendations in English, Hindi, and Marathi, improving the usability of the system for diverse users. Furthermore, deployment on Hugging Face Spaces using Docker and GitHub Actions provides a scalable and publicly accessible platform. The proposed system offers a practical solution for early plant disease detection and has the potential to support precision agriculture by enabling timely diagnosis and informed crop management decisions.

REFERENCES

- [1] Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. *Frontiers in Plant Science*, 7, 1419.
- [2] Ferentinos, K. P. (2018). Deep Learning Models for Plant Disease Detection and Diagnosis. *Computers and Electronics in Agriculture*, 145, 311–318.
- [3] Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., & Hughes, D. P. (2017). Deep Learning for Image-Based Cassava Disease Detection. *Frontiers in Plant Science*, 8, 1852.
- [4] Thenmozhi, K., & Reddy, U. S. (2019). Crop Disease Detection Using Deep Learning. In *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*.
- [5] Barbedo, J. G. A. (2018). Factors Influencing the Use of Deep Learning for Plant Disease Recognition. *Biosystems Engineering*, 172, 84–91.
- [6] Fawcett, T. (2006). An Introduction to ROC Analysis. *Pattern Recognition Letters*, 27(8), 861–874.
- [7] Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep Learning in Agriculture: A Survey. *Computers and Electronics in Agriculture*, 147, 70–90.
- [8] Wei, J., et al. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems*, 35, 24824–24837.