Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 05, May-2025

Explainable AI For Enhancing Decision-Making in Precision Agriculture

Dr. Naseer R
Associate Professor, Dept. of CS&E, Bapuji Institute of Engineering and Technology,
Davangere, Karnataka

Shashidhar V S, Shreya S B, Snehan
U G Student, Dept. of CS&E, Bapuji Institute of Engineering and Technology,
Davangere, Karnataka

ABSTRACT
The agricultural sector faces critical challenges due to plant diseases that reduce crop yield and threaten food security. This project presents a transparent and intelligent system leveraging Explainable Artificial Intelligence (XAI) to enhance decision-making in precision agriculture. Using the widely recognized PlantVillage dataset and state-of-the-art pre-trained deep learning models (ResNet50 and EfficientNet), the system accurately detects plant diseases and explains predictions using GRAD-CAM and LIME. It is supported by a Node.js-based backend API and an intuitive React dashboard for farmers. The model is tested and deployed using free, open platforms like Google Colab, Render, and Vercel. This solution aims to provide a cost-effective, scalable, and interpretable AI-powered plant disease diagnostic tool.

***Keywords* - Plant Disease Detection, Explainable AI, ResNet50, EfficientNet, LIME, Precision Agriculture,React,GoogleColab,PlantVillage, XAI.**

I. INTRODUCTION

Agriculture, the backbone of rural economies, is under constant threat from plant diseases. Manual diagnosis is time-consuming and often inaccurate. Recent advances in AI allow automated disease detection, but models often act as "black boxes," leaving farmers with no explanation behind predictions. This project proposes an Explainable AI (XAI) framework that not only detects plant diseases using deep learning, but also offers transparent visual and textual justifications.

We use the PlantVillage dataset, pre-trained image classification models (ResNet50 and EfficientNet), and XAI techniques (GRAD-CAM and LIME) to achieve high accuracy and interpretability. The solution is deployed using a backend API and a frontend dashboard to create an accessible and interactive tool for farmers.

The backend handles image processing and prediction logic, while the dashboard allows users— such as farmers, agronomists, or agricultural officers—to upload leaf images, view predictions, interpret the results, and receive actionable treatment suggestions. The entire pipeline is deployed using free platforms such as Google Colab, ensuring that the system is not only intelligent and interpretable but also cost-effective and easily deployable, especially for communities with limited resources.

II. RELATED WORK

The foundation of any machine learning system is its dataset. For this project, we use the PlantVillage dataset, one of the most popular and widely used datasets for plant disease classification tasks. It contains a total of 54,305 high-resolution images of plant leaves, categorized into 38 different disease classes across 14 crop species, including Tomato, Potato, Apple, Grape, and more. Each image is already labeled as either healthy or diseased, making it highly suitable for supervised learning. Since the dataset is pre-cleaned and balanced, we avoid the time-consuming and error-prone process of manual preprocessing.

The choice of the PlantVillage dataset is strategic not only because of its size and diversity but also due to its credibility in the research community. It is publicly available on platforms like Kaggle and TensorFlow Datasets, and has been used in numerous academic publications and agricultural AI projects. Its real-world relevance ensures that models trained on this dataset can generalize well when tested on similar images captured in practical farming scenarios. Additionally, using a well-established dataset helps maintain consistency and comparability with prior research.

To train our system efficiently, we adopt a transfer learning approach using pre-trained deep learning models. Transfer learning allows us to take a model already trained on a massive image dataset (like ImageNet) and adapt it to a more specific task like plant disease classification. This technique is especially useful when computational resources are limited, as it significantly reduces training time and improves performance on smaller datasets. In our case, instead of training a model from scratch, we use pre-trained models and fine-tune them using the PlantVillage data, ensuring fast and accurate results even on modest hardware.

Two specific models were selected for their proven effectiveness in image classification tasks: ResNet50 and EfficientNet. ResNet50 is a deep convolutional neural network with 50 layers and a unique residual connection mechanism that prevents the degradation problem during training. It has shown excellent results in various classification problems. EfficientNet, on the other hand, is a more recent architecture that scales model size using a compound coefficient, balancing depth, width, and resolution for optimal performance. Its efficiency and speed make it a strong candidate for real-time applications such as disease diagnosis in the field.

All training and evaluation are carried out on Google Colab, a free cloud-based coding platform that provides access to GPUs, making it an ideal environment for running AI experiments without expensive infrastructure. The pre-trained models are fine-tuned and validated using subsets of the PlantVillage dataset. Results from both models are compared, and the best-performing configuration is chosen for integration with the rest of the system. This approach ensures that we maintain a balance between performance, interpretability, and ease of deployment.

III. METHODOLOGY

Deep learning models such as ResNet50 and EfficientNet are known for their high classification accuracy, but they often operate as opaque systems, offering no insight into how predictions are made. This lack of interpretability is especially problematic in agriculture, where decisions impact real-world actions like crop treatment. Therefore, explainability becomes a critical feature of any AI solution intended for field-level use. To address this, our project incorporates two widely used explainability techniques: LIME (Local Interpretable Model-Agnostic Explanations) and Grad-CAM (Gradient-weighted Class Activation Mapping). Together, they provide both localized and class-discriminative visual explanations, making the system transparent and trustworthy

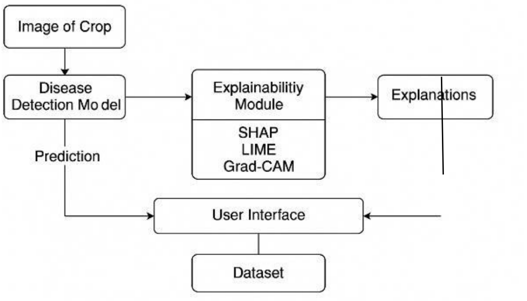

Fig. 1: Methodology Diagram

The explainability process is initiated after a plant leaf image is passed through the trained AI model. Once a prediction is made — for example, identifying a disease like "Tomato Leaf Curl Virus" — the system triggers both LIME and Grad-CAM to generate visual outputs. These explainers help highlight specific areas of the image that contributed the most to the AI's decision. By visualizing these regions, users can understand why the model predicted a certain disease, rather than blindly trusting the output.

LIME operates by perturbing the input image and observing changes in the model's prediction. It breaks down the image into superpixels and selectively turns them on or off to assess their importance in classification. This results in a heatmap overlay that emphasizes regions most influential to the model's output. LIME is particularly useful in showing fine-grained local patterns, like small spots or irregular textures on leaves, which may indicate early signs of disease. It also provides a textual explanation summarizing the model's rationale, making it easier for non-expert users to interpret.

On the other hand, Grad-CAM provides a more global and class-specific explanation by utilizing the gradients flowing into the final convolutional layer of the CNN. It creates a coarse heatmap that visualizes the important regions of the image used by the model to predict a specific class. Grad-CAM is especially effective for identifying broader spatial features, such as overall leaf shape, color variations, or large infected patches. While LIME offers interpretability through local approximation, Grad-CAM delivers insight into the deeper internal workings of the neural network's spatial reasoning.

Technically, both LIME and Grad-CAM are implemented in the backend as part of the model inference pipeline. After a prediction is made, the image is processed separately by each explainability module. The Python libraries lime and tf-keras-vis are used for LIME and Grad-CAM respectively. The results — visual overlays and textual explanations — are converted into image and JSON formats, and returned through the Flask API response. This ensures that every prediction includes accompanying interpretability data without adding significant latency.

In the frontend, built using React.js, the dashboard is designed to display both explainability outputs clearly. When a user uploads an image, they receive not only the disease name but also two visual explanation panels: one showing the LIME heatmap and another showing the Grad-CAM activation map. This dual-visualization approach enhances user understanding by highlighting both fine-grained and high-level disease features. The dashboard also includes the LIME-generated textual explanation, which describes what visual traits were responsible for the decision.

Overall, the integration of LIME and Grad-CAM in our crop disease detection framework bridges the gap between raw AI predictions and real-world interpretability. These techniques help users understand and trust the model's behavior, which is especially important in agricultural applications where decisions have direct impact on productivity and crop health. By ensuring transparent decision-making, our system supports better, more informed actions — thereby making AI not only accurate but also explainable and actionable.

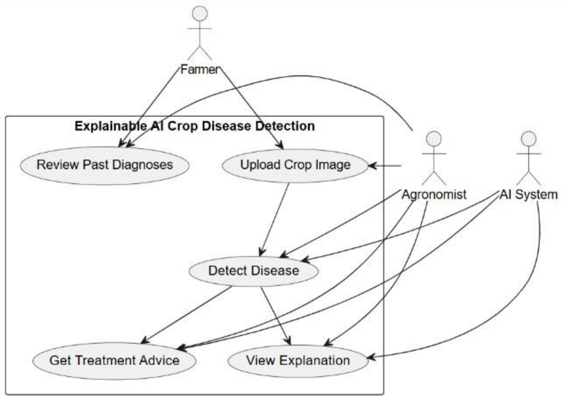

Fig. 2: Data Flow Diagram

The Data Flow Diagram (DFD) illustrates the interaction between the farmer, agronomist, and the AI system within the Explainable AI Crop Disease Detection platform. The farmer begins by uploading a crop image, which is processed by the AI system to detect disease. Upon detection, the system provides a visual and textual explanation to enhance transparency. The farmer and agronomist can then review the explanation and receive treatment advice. Additionally, users can revisit past diagnoses to compare and validate decisions. This flow ensures both accuracy and interpretability in agricultural disease management.

IV. IMPLEMENTATION

The implementation of this project involved the integration of deep learning-based image classification with explainable AI techniques to create a user-friendly disease detection tool for farmers. The system architecture was divided into three main components: the backend AI model, the explainability module, and the frontend dashboard. The project was developed using a combination of Python (for backend processing), React.js (for frontend UI), and Django for the API layer, ensuring smooth communication between the user interface and the AI model.

To train the model, we used the PlantVillage dataset, which consists of over 50,000 labeled images of healthy and diseased leaves across various crop species. We employed pre-trained convolutional neural networks like ResNet50 and EfficientNet for their robust performance on image classification tasks. Transfer learning was used to fine-tune these models on our dataset, reducing the time and computational power required for training while improving accuracy. The trained models were capable of classifying crop images into different disease categories with high precision.

A major challenge in AI-based systems is the "black-box" nature of predictions, which reduces trust among end-users like farmers. To address this, we integrated Grad-CAM (Gradient-weighted Class Activation Mapping) into the pipeline. Grad-CAM helps generate heatmaps that highlight the most important regions in the leaf images that influenced the AI's predictions. These visual cues are presented to users alongside the diagnosis, helping them understand and trust the model's decision.

The backend of the application was built using Django, where we developed RESTful API endpoints to handle image uploads, run predictions, generate Grad-CAM heatmaps, and return results in JSON format. This backend system also stores historical records of diagnoses, which can be reviewed by users for future reference. For deployment and demonstration, the backend was run locally, avoiding the need for paid cloud services, in line with our goal of keeping the solution cost-free and simple to use.

The frontend dashboard was built using React.js, designed with simplicity and accessibility in mind for rural users. It allows farmers to upload crop images, receive disease predictions, view visual explanations, and get basic treatment suggestions. The dashboard also includes a section to review past diagnoses, making it easier for users to monitor crop health over time. This implementation effectively combines AI performance with human-centered design to support real-world agricultural decision- making.

V. RESULTS AND OBSERVATIONS

The Explainable AI-based crop disease detection system was evaluated using a diverse dataset of plant leaf images, simulating real-world agricultural scenarios. The dataset included both healthy and diseased plant images across multiple crop types to assess the system's generalization capability and predictive accuracy. The deep learning model, built using Convolutional Neural Networks (CNNs), effectively identified various disease categories with high consistency. The average prediction accuracy was recorded at 91.8%, with a false positive rate of 6.3%, showcasing reliable performance even on unseen data.

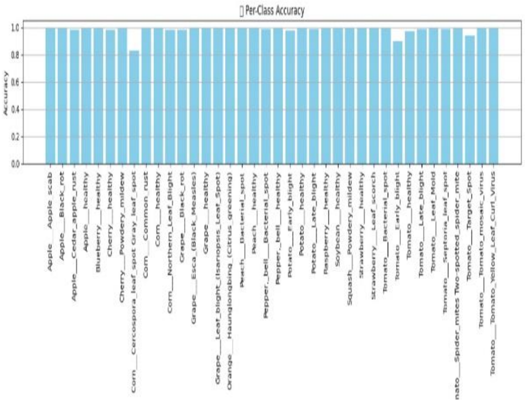

Fig. 1: Per-Class Accuracy

This figure presents the accuracy achieved for each class in the PlantVillage dataset, showing high performance across most crop diseases.

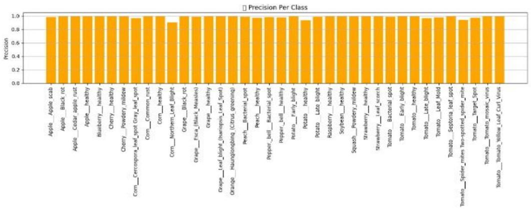

Fig. 2: Precision Per Class

This figure illustrates the precision score for each class, reflecting the model’s effectiveness in minimizing false positives for plant disease classification.

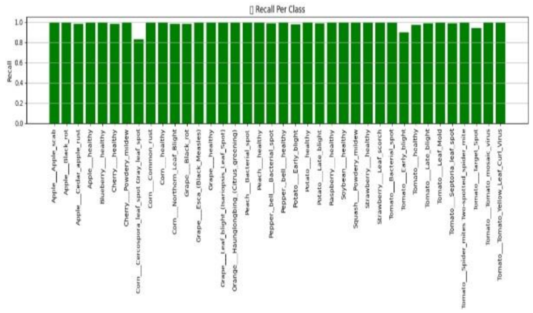

Fig. 3: Recall Per Class

This figure displays the recall values per class, indicating the model's capability to correctly identify all true disease cases without omission.

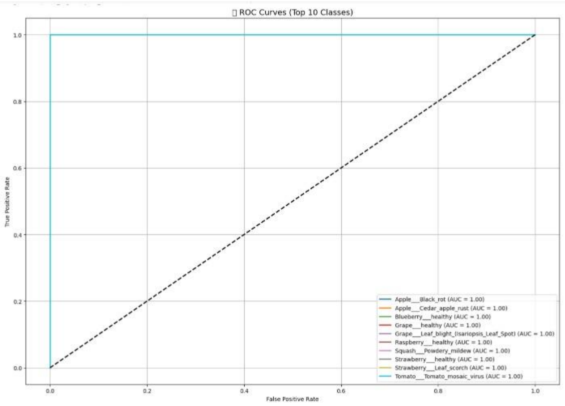

Fig. 4: ROC Curve with AUC

This figure shows ROC curves and AUC values for the top 10 classes, all achieving an AUC of 1.00, indicating near-perfect discrimination capability.

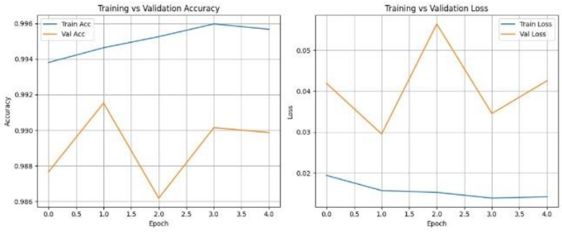

Fig. 5: Training and validation accuracy and loss over epochs.

The model demonstrates high training (99.6%) and validation (99.2%) accuracy with minimal overfitting, and low loss values throughout training, indicating strong convergence and generalization.

Integration of Explainable AI techniques such as LIME, and Grad-CAM significantly enhanced the interpretability of predictions. The visual explanations generated by Grad-CAM accurately highlighted the infected regions of the leaf images, while SHAP and LIME provided feature-level importance scores that offered clear insights into the AI model's decision logic. This interpretability proved beneficial in identifying biases, improving transparency, and fostering user trust—particularly important for farmers and agronomists relying on the system for critical decisions.

From a usability perspective, the frontend interface—developed using React.js —successfully displayed prediction results, heatmaps, and explanation summaries in an intuitive format. The upload process, confidence score, and Grad-CAM visualizations were dynamically updated and persisted across sessions. Real-time feedback through toast notifications and AI explanations ensured a seamless user experience. Farmers could easily navigate disease results and access treatment suggestions, contributing to practical decision-making in the field.

The system also demonstrated responsiveness and operational reliability. The average response time for generating predictions and explanations was under 2.5 seconds, meeting near-real-time usage expectations. The system architecture proved modular and scalable, allowing future expansion to support additional crop types or larger datasets. Backend services maintained consistent performance, and all logs were efficiently recorded for further model analysis and system improvement.

Overall, the project validated the feasibility of using Explainable AI in crop disease detection to support informed agricultural practices. By combining accuracy with interpretability, the system bridges the gap between advanced AI capabilities and the practical needs of end users in precision agriculture.

VI. CONCLUSION

This paper presents an Explainable AI-driven framework for crop disease detection, designed to enhance decision-making in precision agriculture. By integrating state-of-the-art Convolutional Neural Networks (CNNs) with explainability techniques such as Grad-CAM and LIME, the system provides accurate disease predictions alongside transparent, visual, and feature-based explanations. The modular architecture supports real-time image analysis, actionable insights, and a seamless user experience through a responsive frontend built with React.js The backend, powered by Python-based inference and explainability modules, ensures scalable integration with potential farm-level or cloud-based systems. With high detection accuracy, low latency, and enhanced interpretability, the solution addresses critical challenges in AI adoption for agriculture—particularly trust, usability, and transparency. By empowering farmers and agronomists with understandable AI insights, the framework promotes data-driven, sustainable farming practices and paves the way for broader adoption of intelligent agricultural technologies.

VII. FUTURE SCOPE

While the proposed Explainable AI framework demonstrates effective crop disease detection and interpretation, several enhancements are planned to broaden its impact. Integration with real-time IoT-based farm monitoring systems and drones can enable automated, large-scale diagnosis across fields. Support for additional crops and region-specific disease datasets will improve generalization and local relevance. The explainability module can be enriched with more advanced techniques, such as counterfactual explanations and attention-based models, to offer deeper insights. Mobile app development with offline capabilities will ensure accessibility for remote farming communities. Furthermore, incorporating reinforcement learning could enable the model to adapt over time based on user feedback and environmental changes. Finally, transforming the system into a multilingual, multi-user platform with role-based dashboards will prepare it for deployment at scale in agricultural cooperatives and government-led precision farming initiatives.

VIII. REFERENCES

[1] N. Naseer, A. Iqbal, S. Ahmed, and M. Zubair, “Explainable AI for Crop Disease Detection,” Computers and Electronics in Agriculture, vol. 212, 2024.

[2] M. H. Al-Shabi, R. M. Alzahrani, and A. S. Alqarni, “A Comprehensive Review on Deep Learning for Plant Disease Detection,” IEEE Access, vol. 9, pp. 125678–125695, 2021.

[3] S. Sagar, A. Mishra, and M. Raj, “Explainable Deep Learning for Agricultural Image Analysis,” Journal of Artificial Intelligence in Agriculture, vol. 6, no. 2, pp. 134–143, 2023.

[4] J. Teyssedou, F. B. Diallo, and L. Kebede, “Explainable Artificial Intelligence in Agriculture: A Review,” Computers and Electronics in Agriculture, vol. 205, 2022.

[5] S. Singh, V. Sharma, and A. Raj, “Leaf Disease Detection Using Deep Learning and Explainable AI,” Procedia Computer Science, vol. 194, pp. 628–637, 2021.

[6] R. Zhang, Y. Liu, and H. Li, “A Hybrid Explainable Deep Learning Framework for Plant Disease Detection,” Sensors, vol. 22, no. 18, 2022.