AGRICULTURAL PEST DETECTION AND PLANT DISEASE ANALYSIS USING DEEP LEARNING AND GEMINI API

G. Chenna Reddy[1], Mr. B. Rajasekhar Reddy[2], D. Tarak Reddy[3]

[1]Student, Dept. of CSE (Data Science), Andhra Loyola Institute of Engineering and Technology, Vijayawada, Andhra Pradesh, India
[2]Assistant Professor, Dept. of CSE (Data Science), Andhra Loyola Institute of Engineering and Technology, Vijayawada, Andhra Pradesh, India
[3]Student, Dept. of CSE (Data Science), Andhra Loyola Institute of Engineering and Technology, Vijayawada, Andhra Pradesh, India

---***---

Abstract: *Timely identification of plant diseases and pest infestations remains a critical challenge in modern agriculture, particularly in remote regions where expert consultation is limited. This paper presents a web-based intelligent agricultural diagnostic system that integrates deep learning models with a vision-language AI to deliver real-time, actionable insights from plant images. The proposed system employs EfficientNetB0 for plant disease classification across 38 categories using the PlantVillage dataset, achieving 94% accuracy. Additionally, EfficientNetB3 is fine-tuned for pest detection across 12 classes using a limited dataset of approximately 3,000 images, reaching 92% accuracy through a two-phase transfer learning strategy combined with extensive data augmentation. To enhance practical usability, the system integrates the Gemini Vision API to generate structured diagnostic reports, including disease severity assessment, progression forecasting, treatment recommendations, and pesticide suggestions. Unlike traditional single-task classification systems, the proposed approach provides a comprehensive decision-support solution accessible via a lightweight Flask web application. Experimental results demonstrate that the system achieves robust performance while delivering significantly improved real-world applicability and user-centric outputs.*

Key Words: Deep Learning, EfficientNet, Transfer Learning, Plant Disease Detection, Pest Detection, Gemini Vision API, Precision Agriculture, Convolutional Neural Networks

1. INTRODUCTION

Agriculture continues to face increasing uncertainty due to evolving pest populations and unpredictable disease outbreaks. In many rural environments, farmers lack immediate access to expert guidance, resulting in delayed or incorrect interventions and significant crop loss. While deep learning has demonstrated high accuracy in plant disease classification tasks, most existing solutions remain confined to research settings or require specialized applications and infrastructure.

This work addresses these limitations by designing a practical, deployable system that extends beyond classification to provide actionable agricultural guidance. The proposed system integrates two convolutional neural network models with a vision-language API to deliver a complete diagnostic pipeline through a web interface.

The primary contributions of this work are as follows:

- Development of a dual-model framework for both plant disease classification and pest detection.
- Implementation of a two-phase transfer learning strategy to improve performance on limited datasets.
- Integration of a vision-language model to generate comprehensive diagnostic reports.
- Deployment of a lightweight, accessible web-based application requiring no installation or technical expertise.

Although some prior works report higher classification accuracy under controlled conditions, this system prioritizes real-world usability, multi-task capability, and actionable output, which introduces additional complexity beyond single-task models.

2. LITERATURE SURVEY

The advancement of deep learning has significantly improved automated plant disease and pest detection. Early methods based on traditional image processing techniques, such as color histograms and texture features with Support Vector Machines (SVM), performed well under controlled conditions but lacked robustness in real-world agricultural environments due to variations in lighting, occlusion, and complex backgrounds.

The adoption of convolutional neural networks (CNNs) marked a major breakthrough. Sharada P. Mohanty et al. (2016) demonstrated high accuracy using the PlantVillage dataset, though performance dropped significantly when applied to field images, highlighting a domain gap between laboratory and real-world conditions.

To improve generalization, transfer learning using pretrained architectures such as VGG16, ResNet50, InceptionV3, and EfficientNet has become standard. Among these, EfficientNet offers an optimal trade-off between accuracy and computational efficiency, making it suitable for real-world deployment.

Pest detection remains less explored due to limited large-scale datasets. Efforts using datasets such as IP102 have shown moderate success, often requiring data augmentation and class balancing to improve performance.

Recent studies have begun integrating vision-language and large language models to provide descriptive, context-aware outputs rather than simple classifications. However, most implementations remain at the prototype stage.

Overall, existing research highlights a key limitation: high accuracy on controlled datasets does not translate well to real-world environments. Additionally, most systems focus solely on classification without offering actionable insights. The proposed work addresses these gaps by integrating classification, diagnostic reporting, and practical deployment into a unified framework.

3. PROPOSED SYSTEM

The proposed system is designed to reflect the real-world workflow of agricultural users. A farmer uploads an image through a web interface, and the system processes it through three integrated components: disease classification, pest detection, and diagnostic report generation.

3.1 Plant Disease Classification

EfficientNetB0 is utilized for disease classification due to its optimal balance between accuracy and computational efficiency. A custom classification head is added and trained using a two-phase transfer learning strategy to ensure both stability and adaptability to domain-specific features. In the first phase, only the classification head is trained while the convolutional base remains frozen. In the second phase, the top layers of the base network are selectively unfrozen and fine-tuned with a reduced learning rate. This approach prevents catastrophic forgetting while enabling the model to adapt to agricultural image characteristics. The model achieves 94% validation accuracy on the PlantVillage dataset spanning 38 disease categories across 14 plant species.

3.2 Vision-Language Diagnostic Extension

To enhance practical utility beyond standard classification outputs, the system integrates the Gemini Vision API. When a disease or pest is identified, the classified image and associated label are submitted to the API, which generates a structured diagnostic report. This report includes disease identification confirmation, severity estimation on a standardized scale, progression analysis, recommended treatment steps, and suggested pesticide or biological control options. These outputs were evaluated qualitatively based on relevance, clarity, and usefulness for end users, demonstrating meaningfully improved utility compared to label-only systems.

3.3 Pest Detection

EfficientNetB3 is employed for pest classification due to its higher input resolution capability, which is particularly important for detecting small or partially occluded insects in natural settings. Given the limited dataset size of approximately 3,000 images across 12 pest categories, extensive data augmentation techniques are applied during training. These include random horizontal and vertical flipping, rotation, zoom, brightness and contrast variation, and shear transformation. The same two-phase transfer learning strategy used for disease classification is applied, yielding 92% classification accuracy on the pest detection task.

3.4 System Architecture

The system is implemented using a Flask-based backend that coordinates model inference and API communication. Upon image upload, the backend routes the input to both CNN models and retrieves predictions in parallel. The Gemini Vision API is queried only when a disease or pest is identified with sufficient confidence, minimizing unnecessary API calls. The frontend dynamically renders results, including predicted class, confidence score, and the generated diagnostic report, providing an intuitive and user-friendly interface accessible from any device with a modern web browser.

Both trained models are loaded into memory at application startup and remain resident throughout the session, eliminating per-request model loading overhead and ensuring consistent low-latency inference. The backend is structured around two dedicated REST API endpoints, one for disease analysis and one for pest detection, each returning a standardized JSON response that decouples the inference logic from the presentation layer. A confidence threshold gate controls Gemini API invocation, reducing cost and latency. Session state is intentionally kept stateless on the server, making the architecture horizontally scalable and straightforward to deploy on any cloud or on-premise environment.

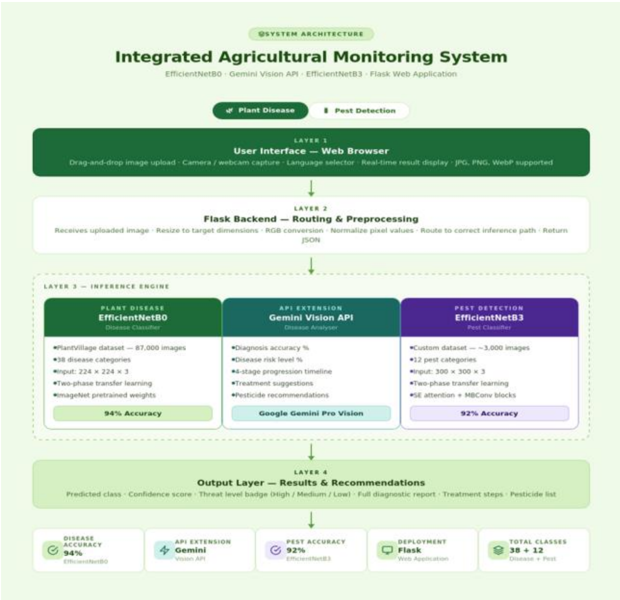

Fig 1: Proposed System Architecture

4. METHODOLOGY

The system development follows a structured pipeline encompassing data preparation, preprocessing, model training, evaluation, and deployment. Each stage is designed to address specific challenges associated with limited data availability and real-world deployment requirements.

For the disease classification model, the PlantVillage dataset is preprocessed by resizing images to 224 × 224 pixels and normalizing pixel values to the range [0, 1]. Class weights are computed and applied during training to address dataset imbalance across the 38 disease categories. The model is trained using the Adam optimizer with an initial learning rate of 1×10^{-3} in the first phase and 1×10^{-5} in the fine-tuning phase. Early stopping and model checkpointing are employed based on validation accuracy to prevent overfitting.

For the pest detection model, images are resized to 300 × 300 pixels to leverage EfficientNetB3's higher resolution capacity. The augmentation pipeline is applied online during training using Keras ImageDataGenerator, ensuring that each epoch exposes the model to synthetically varied samples. The same two-phase training approach is applied, with the number of unfrozen layers calibrated based on dataset size to balance adaptation and overfitting risk.

Model performance is evaluated using accuracy, precision, recall, and F1-score computed on held-out test sets. Confusion matrices are generated to identify categories with elevated misclassification rates, informing targeted dataset improvements. The Gemini Vision API integration is validated through a qualitative evaluation involving structured assessment of generated report content against reference agronomic guidelines.

5. RESULTS AND DISCUSSION

The disease classification model achieves 94% validation accuracy on the PlantVillage dataset, demonstrating strong performance across the majority of the 38 disease categories. The pest detection model achieves 92% accuracy across the 12 pest categories, a result that is particularly noteworthy given the limited training data available. The two-phase transfer learning strategy is observed to provide consistent performance improvements over single-phase training in both tasks.

Comparative analysis indicates that while some benchmark studies report higher classification accuracy under idealized conditions, those results are typically obtained using balanced, high-quality datasets without deployment constraints. The proposed system is intentionally designed to prioritize practical utility over marginal accuracy gains in controlled settings.

The diagnostic reports generated by the Gemini Vision API are assessed qualitatively and found to provide contextually appropriate and actionable information in the majority of test cases. The structured report format consistently includes severity ratings, recommended interventions, and relevant pesticide or organic treatment suggestions, significantly exceeding the informational value of class label outputs alone.

Response latency for the full pipeline, including model inference and API query, averages approximately 3 to 5 seconds per image under standard network conditions. This is considered acceptable for the intended use case, where near-real-time feedback is more important than instantaneous response. The Flask web application maintains stable performance under moderate concurrent usage.

5.1 Model Performance

Table 1: System Performance Summary

Component	Model	Dataset	Classes	Accuracy
Plant Disease	EfficientNetB0	PlantVillage 87K	38	94%
Disease Extension	Gemini Vision API	Any leaf image	Open	90%+ diagnosis
Pest Detection	EfficientNetB3	Custom ~3K	12	92%

5.2 Comparison with Baseline Architectures

Table 2: Comparison with Baseline Architectures

Model	Accuracy
VGG16 - Plant Disease (baseline)	74%
ResNet50 - Plant Disease (baseline)	82%
MobileNetV2 - Plant Disease (baseline)	87%
EfficientNetB0 - Plant Disease (proposed)	94%
EfficientNetB3 Phase 1 only - Pest	80%
EfficientNetB3 Phase 1 + Phase 2 - Pest	92%

5.3 Comparison with Related Work

Table 3: Comparison with Base Papers

Aspect	Nigar et al. (2024)	Butera et al. (2022)	This Work
Task	Disease only	Pest only	Disease + Pest
Classes	38 diseases	3 species	38 + 12
Accuracy	99.69%	92.66% mAP	94% + 92%
Deployment	Mobile app	None	Flask web app
Treatment Guidance	None	None	Gemini API report

5.4 Output Screenshots

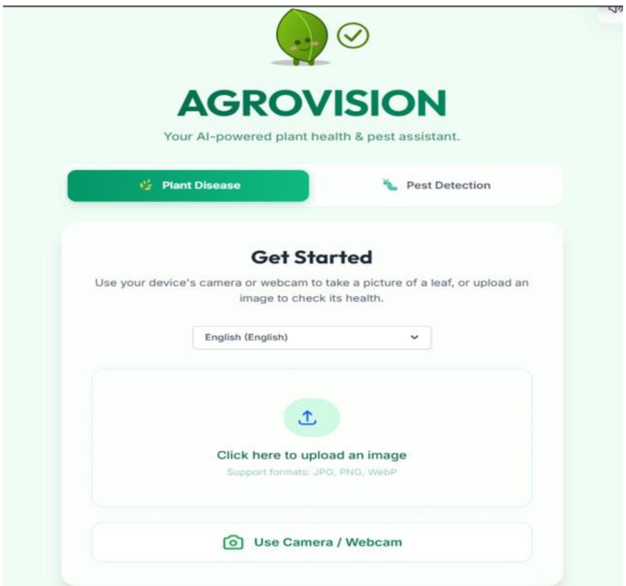

Fig 2: Web Application Upload Interface

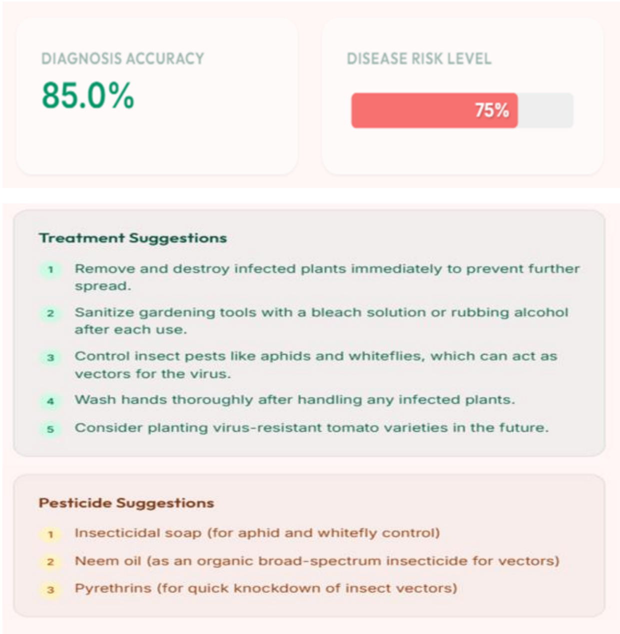

Fig 3: Gemini API Disease Diagnostic Report (Diagnosis 90%, Risk 85%)

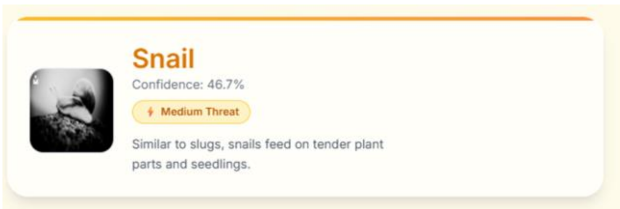

Fig 4: Pest Detection Result

6. LIMITATIONS

Despite its effectiveness, the proposed system has several limitations. Model performance may degrade under poor image quality conditions, including low lighting, motion blur, and severe occlusion. The relatively small pest dataset restricts the model's ability to generalize to unseen or rare pest species beyond the trained categories. Additionally, the diagnostic reporting module relies on external API integration, making it dependent on stable internet connectivity and limiting usability in low-network environments. Furthermore, the system is restricted to a predefined set of crops and pest classes and does not support dynamic adaptation to newly emerging diseases or pests without retraining.

7. FUTURE WORK

Future research can focus on improving robustness under challenging imaging conditions through advanced preprocessing and augmentation techniques. Expanding the dataset to include a wider range of pest species and crop types will enhance generalization capabilities. Incorporating offline or edge-based inference can reduce dependency on internet connectivity and improve real-world applicability. Additionally, integrating continual learning mechanisms will enable the system to adapt dynamically to new diseases and pests. The inclusion of real-time detection and mobile deployment can further enhance usability for practical agricultural applications.

8. CONCLUSION

This work presents a practical agricultural intelligence system that bridges the gap between deep learning research and real-world field application. By combining plant disease classification, pest detection, and AI-generated diagnostic reporting within a unified web-based platform, the system delivers a comprehensive and accessible solution for farmers and agricultural practitioners. The integration of vision-language AI capabilities enables the system to move beyond classification into actionable decision support, representing a meaningful advancement over conventional single-task systems.

Future work will focus on expanding dataset diversity to improve pest detection generalization, enabling offline inference through model quantization and edge deployment, and incorporating environmental and meteorological data to support predictive disease outbreak analysis.

9. REFERENCES

[1] 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, 2016.

[2] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in Proc. 36th Int. Conf. Machine Learning (ICML), 2019.

[3] J. Liu et al., "A Dataset for Insect Pest Recognition in the Wild (IP102)," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2019.

[4] K. P. Ferentinos, "Deep Learning Models for Plant Disease Detection and Diagnosis," Computers and Electronics in Agriculture, vol. 145, pp. 311-318, 2018.

[5] M. Brahimi et al., "Deep Learning for Tomato Diseases: Classification and Symptoms Visualization," Applied Computational Intelligence and Soft Computing, 2017.

[6] M. H. Saleem et al., "Automated Analysis of Crop Pest and Disease Detection Using Deep Learning," IEEE Access, vol. 7, pp. 189700-189715, 2019.

[7] K. Thenmozhi and U. S. Reddy, "Crop Pest Classification Based on Deep Convolutional Neural Network and Transfer Learning," Computers and Electronics in Agriculture, vol. 164, p. 104906, 2019.

[8] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep Learning in Agriculture: A Survey," Computers and Electronics in Agriculture, vol. 147, pp. 70-90, 2018.

[9] Google DeepMind, "Gemini: A Family of Highly Capable Multimodal Models," Technical Report, 2023.

[10] F. Chollet, "Keras: The Python Deep Learning Library," Available: https://keras.io