FarmTalk: A Real Time Platform for Plant Disease Identification and Management

Abhishek Chaudhari, Guruvasanth Amirthapandian Sureshkumar, Shesha Sai Kumar Reddy Sadu
NLP Nexus
chaud324@umn.edu, amirt003@umn.edu, sadu0004@umn.edu

Abstract

Leveraging LLMs for Real-Time Plant Disease Detection and Management addresses the critical challenges of plant disease detection in agriculture. Traditional methods are slow, error-prone, and lack scalability, threatening food security and crop yields.

FarmTalk combines YOLOv8 for precise disease detection with a Retrieval-Augmented Generation (RAG) framework using NVIDIAEmbeddings and FAISS for semantic search. Integrated with LLMs like GPT-3.5 Turbo, it delivers real-time, actionable remediation recommendations.

1 Introduction

Agriculture serves as the cornerstone of global food security, directly sustaining billions of lives and contributing significantly to the global economy. However, plant diseases remain one of the most persistent and devastating threats to agricultural productivity. These diseases reduce crop yields, degrade produce quality, and result in significant economic losses. According to research, plant diseases account for a staggering 20-30

Timely detection and management of plant diseases are crucial for minimizing their impact and ensuring sustainable farming practices. Early identification allows farmers to take precise actions to contain infections, reduce crop losses, and maintain yield quality. Traditional methods of plant disease detection, however, are labor-intensive, slow, and prone to errors. These methods typically rely on manual inspection by agricultural experts, which becomes impractical in large-scale farming operations and inaccessible for farmers in remote or resource-limited areas.

The advent of machine learning (ML) and artificial intelligence (AI) has opened new opportunities to address these challenges. Computer vision models like YOLOv8 have demonstrated their ability to analyze complex visual data, such as plant leaves, for early disease identification. Similarly, natural language processing (NLP) models, particularly large language models (LLMs), are capable of generating actionable insights and remediation strategies by processing contextually relevant information.

FarmTalk: Leveraging LLMs for Real-Time Plant Disease Detection and Management bridges these technological advancements into a unified platform. Combining YOLOv8's disease detection capabilities with the contextual power of LLMs like GPT-3.5 Turbo, FarmTalk uses a Retrieval-Augmented Generation (RAG) framework for delivering accurate and actionable remediation recommendations. With a user-friendly interface that supports image uploads and live camera feeds, FarmTalk empowers farmers, researchers, and policymakers to manage plant diseases effectively and sustainably.

This report outlines FarmTalk's development, methodology, and results, showcasing its potential to revolutionize plant disease management by reducing crop losses, improving decision-making, and contributing to a resilient agricultural system.

2 Limitations of Current Systems

Traditional and emerging methods of plant disease detection face significant limitations that hinder their effectiveness and scalability in modern agricultural settings:

- **LLM Hallucination:** While large language models (LLMs) like GPT-3.5 Turbo are capable of generating detailed recommendations, they can occasionally produce hallucinated or irrelevant information. These inaccuracies, if undetected, could mislead farmers and lead to inappropriate or ineffective disease management strategies.

- **Manual and Labor-Intensive Processes:** Disease detection through manual inspection by agricultural experts is time-consuming, resource-intensive, and impractical for large-scale farms. Farmers in remote or under-resourced areas often lack access to expert consultations, leaving them without reliable diagnostic tools.

- **Delayed Interventions:** The time required for disease identification and subsequent expert consultation often results in delayed responses, allowing diseases to spread and cause significant crop damage.
- **Limited Scalability of Traditional Methods:** Manual inspection methods struggle to scale in large agricultural operations, where monitoring vast fields and multiple crops is a daunting task.
- **Data Scarcity:** Many regions lack systematic disease tracking and annotated datasets, making it challenging to develop predictive models or provide accurate disease diagnostics.
- **Over-Reliance on Pesticides:** Without precise diagnostics, farmers often resort to excessive pesticide use as a preventive measure. This approach increases production costs, harms ecosystems, and contributes to environmental degradation.

3 Literature Survey

3.1 YOLOv8 for Object Detection

In the paper "Leaf-based disease detection in bell pepper plant using YOLO v5" by Mathew and Mahesh, 2021, the use of YOLOv5 for pepper plant disease detection with high accuracy as shown in Figure 1. The use of YOLO models in plant disease identification has shown great promise in achieving real-time performance across various datasets. Terven et al., 2023 show that YOLOv8 builds upon these foundations, offering enhanced detection capabilities and faster inference times, making it suitable for real-time disease identification.

3.2 Retrieval-Augmented Generation (RAG)

In the study "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks" conducted by Lewis et al., 2021 introduces the concept of Retrieval-Augmented Generation (RAG) to enhance Large Language Models (LLM) by fetching context-specific data from external databases.

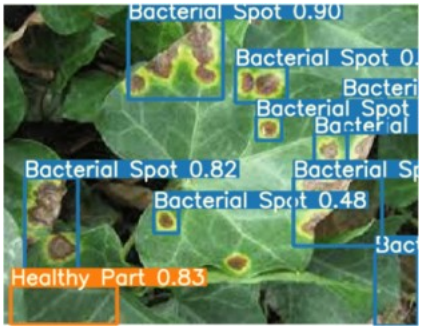

Figure 1: Bell Pepper Disease Detection from Mathew and Mahesh, 2021

This approach significantly reduces hallucination issues in generated responses, ensuring that LLM outputs are grounded in accurate, up-to-date information. RAG was tested across various knowledge-intensive NLP tasks, proving its ability to seamlessly integrate external data with model outputs, which is critical for reliable remediation in agricultural applications.

3.3 Artificial Intelligence Plant Doctor: Plant Disease Diagnosis Using GPT-4 Vision

This study by Hue et al., 2024 presents a GPT-4-based system for diagnosing plant diseases by combining image recognition and natural language processing. Using a detailed knowledge base from official Korean plant disease and pesticide registries, the AI Plant Doctor offers interactive advice on diagnosis, control methods, and pesticide use. The system, trained on 1,420 host plants, 2,462 pathogens, and 37,467 pesticide instances, demonstrates how GPT-4 can be utilized for precise and interactive disease management.

Similarly, Singh, 2024 highlights the integration of the Retrieval-Augmented Generation (RAG) framework with vision-guided AI agents for risk prevention and root cause analysis. This approach, leveraging NVIDIA and LangChain tools, emphasizes the importance of combining retrieval mechanisms with LLMs to enhance reliability and context-specific outputs.

Building on these ideas, our project integrates the RAG framework with YOLOv8 and GPT-3.5 Turbo to address limitations identified in both studies, such as biased data or subjective visual observations. This integration aims to enhance accuracy, minimize hallucinations in LLM-generated recommendations, and provide a scalable, robust solution for plant disease management.

4 Problem Definition

To address the challenges of slow, labor-intensive, and inaccurate plant disease identification methods, we introduce **FarmTalk**, a hybrid solution. FarmTalk combines YOLOv8, a state-of-the-art object detection model, for real-time plant disease identification using leaf images, and integrates GPT-3.5 with Retrieval-Augmented Generation (RAG) to provide reliable, context-specific remediation.

Our goal is to enable farmers to use a live camera feed to quickly identify plant diseases and receive actionable solutions. By leveraging RAG, we aim to overcome the hallucination problem commonly associated with LLMs, ensuring more accurate and relevant recommendations for disease treatment.

5 Broader Impact

FarmTalk has the potential to make a transformative impact across multiple sectors:

- **Farmers and Agricultural Workers:** By providing real-time, accurate disease detection and actionable remediation strategies, FarmTalk empowers farmers to make informed decisions, reduce crop losses, and optimize resource usage.
- **Food Security Organizations:** By minimizing losses due to plant diseases, FarmTalk contributes to enhancing global food security, ensuring a more stable food supply.
- **Environmental Impact:** The platform promotes sustainable farming practices by reducing the overuse of pesticides, which in turn helps protect ecosystems and reduce environmental degradation.

FarmTalk exemplifies the application of AI to address critical challenges in agriculture, offering a scalable and accessible solution that benefits diverse stakeholders while supporting global goals of food security and sustainability.

6 Approach

6.1 Dataset

The dataset for this project consists of over 46,000 labeled images sourced from Kaggle, encompassing 25 plant disease categories, including Tomato Mosaic Virus and Northern Corn Leaf Blight. The images were preprocessed to a uniform size of 256x256 pixels, ensuring compatibility with YOLOv8. Additional annotations were used to enhance semantic retrieval, supporting context-specific recommendations.

6.2 Overall Approach

The architecture integrates:

- **YOLOv8 for Disease Detection:** Identifies diseases in plant leaves with bounding box annotations.
- **RAG Framework:** Combines FAISS for semantic search and NVIDIAEmbeddings for vector representation of research documents.
- **LLM Integration:** Utilizes GPT-3.5 Turbo to generate remediation strategies based on retrieved context.

An architecture diagram illustrating the data flow from input (image upload/live feed) to detection, retrieval, and response generation is included (see Figure 2).

6.3 Detailed Methodology

6.3.1 YOLOv8

The dataset used to finetune the YOLOv8n (nano) model was the New Plant Disease Dataset from Kaggle which itself was imported from Mohanty et al., 2016, which contains 87K rgb images of healthy and diseased crop leaves which is categorized into 38 different classes. Out of these 38 classes, the model for this project was trained for 25 classes.
Since, the dataset is not of the YOLO specific structure, we need to pre-process the Kaggle dataset into a YOLO compatible dataset. That includes creating a respective annotation text file for each of the image containing the ground truth label and the bounding box. Due to the nature of the images in the Kaggle dataset, we can define the bounding box as the entire image, as it is essentially already segmented. The associated code can be modified to include/exclude specific classes from the dataset. After this we can train the YOLOv8n via YOLO using the hyperparameters mentioned in Table. 1,

RAG Implementation: Semantic search was powered by FAISS and NVIDIAEmbeddings, enabling the retrieval of context-aware data from indexed research papers.

Interface Development: A Streamlit-based user interface was designed to support image uploads, live camera feeds, and natural language queries.

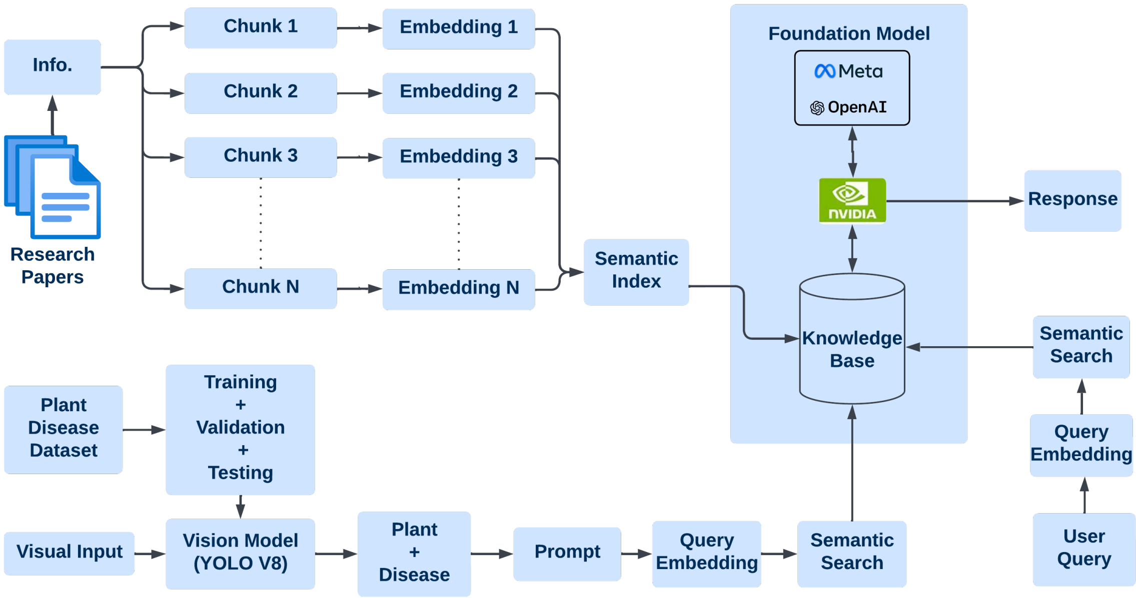

Figure 2: System Architecture for FarmTalk: Data flow from input to output.

Table 1: YOLO Hyperparameters

Hyperparameters	
No of Epochs	10
Image Size	256
Batch Size	64

Prompt Engineering: Optimized prompts were crafted to align user queries with the semantic retrieval pipeline, ensuring accurate and actionable LLM outputs.

7 Challenges Encountered and Novel Approach

7.1 Challenges Anticipated and Encountered

- **Hardware Setup:** Integrating a live camera feed with the YOLOv8 detection model and Streamlit interface posed synchronization and calibration challenges, particularly ensuring real-time performance in diverse environments.

- **Query Embedding:** Generating embeddings for diverse user queries using NVIDIAEmbeddings required handling variations in natural language while ensuring relevance and contextual alignment.

- **RAG Framework Testing:** Testing the Retrieval-Augmented Generation (RAG) framework highlighted challenges with hallucination in LLM responses, requiring fine-tuning and grounding outputs in reliable data.

- **Dataset Preparation:** Addressing missing annotations, class imbalances, and poor-quality images was critical to improving model generalization and robustness.

- **Performance Optimization:** Achieving a balance between detection accuracy and response latency, especially in low-resource setups, required iterative profiling and optimization.

The initial attempts often fell short, particularly in ensuring accurate embeddings and reducing hallucination in LLM outputs. These insights guided the iterative refinement of both the architecture and workflows.

7.2 Scientific Novelty of the Approach

- **Integrated Vision-Language System:** The combination of YOLOv8 with the RAG framework represents a novel synergy, leveraging computer vision for detection and LLMs for actionable insights, ensuring a holistic solution to plant disease management.

- **Context-Aware Remediation:** The use of semantic search via FAISS and NVIDIAEmbeddings ensures that remediation strategies are not only accurate but also contextually relevant, minimizing hallucinations commonly associated with LLMs.

- **User-Friendly Accessibility:** A Streamlit-based interface democratizes access, enabling real-time disease detection and interaction for farmers without requiring technical expertise.

- **Grounded LLM Responses:** By integrating RAG, FarmTalk overcomes the typical pitfalls of standalone LLMs, ensuring that recommendations are both precise and grounded in factual data.

This combination of robust detection, contextual retrieval, and user-centric design underscores FarmTalk's novel contribution to agricultural AI.

8 Experimental Setup

The experimental setup for **FarmTalk: Leveraging LLMs for Real-Time Plant Disease Detection and Management** is illustrated in the accompanying figure. This setup was designed to replicate a real-world scenario of plant disease detection and user interaction, demonstrating the platform's practicality and effectiveness.

8.1 Components of the Setup

- **Web Camera:** Positioned to capture high-quality images of the tomato plant leaves, the camera served as the primary data acquisition device. It enabled real-time disease detection through live feed integration.

- **Potted Tomato Plant:** A tomato plant with mosaic virus in a pot was used to simulate an agricultural setting. The plant provided a test case for evaluating the YOLOv8 model's accuracy in detecting potential diseases.

- **Laptop Running FarmTalk:** The laptop hosted the FarmTalk platform, featuring a user-friendly Streamlit interface. It facilitated interaction with the system, allowing users to upload images or use the live feed for disease detection. Users could also query the system for remediation recommendations or additional insights via prompts.

8.2 Workflow Demonstrated in the Setup

The workflow of the FarmTalk platform is as follows:

1. The web camera captured real-time images of the tomato plant.

2. The YOLOv8 model processed the images to identify signs of diseases, such as *"Tomato Mosaic Virus."*

3. Users interacted with the Streamlit interface by asking questions related to detected diseases.

4. The integrated Retrieval-Augmented Generation (RAG) framework utilized GPT-3.5 Turbo to provide detailed, context-specific responses, ensuring actionable insights for disease management.

Figure 3: Experimental setup demonstrating the FarmTalk platform, including a web camera, potted tomato plant, and laptop running the Streamlit interface.

The figures below provide a comprehensive view of the hardware arrangement and the working Streamlit interface, emphasizing the seamless interaction between the various components of the FarmTalk platform.

The hardware setup includes a web camera, strategically positioned to capture high-quality real-time images of the potted tomato plant. This plant, infected with Tomato Mosaic Virus, serves as a controlled test subject to demonstrate the accuracy and efficiency of the YOLOv8 model in identifying plant diseases. The laptop, running the Streamlit interface, acts as the central processing unit where the captured images are analyzed in real-time.

Figure 3 illustrates the physical setup of the FarmTalk platform, showcasing the integration of the camera, the potted tomato plant, and the laptop, which collectively form a robust and accessible system. This setup ensures that even users in field conditions can seamlessly interact with the platform to detect plant diseases effectively.

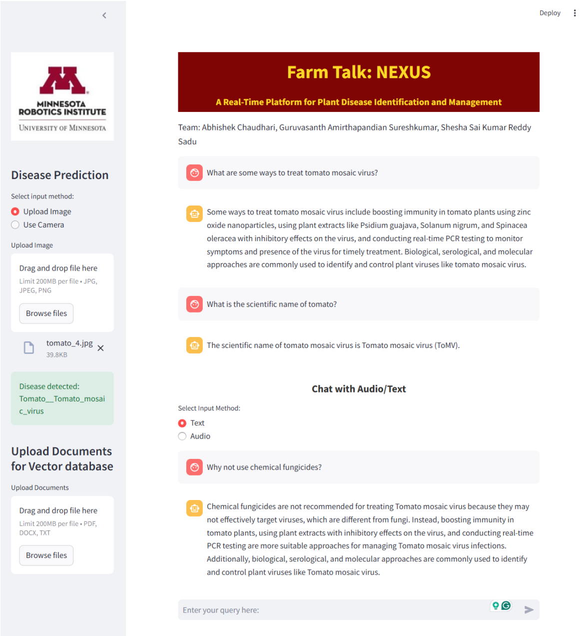

Figure 4: Streamlit user interface showcasing the real-time disease detection results, interactive query response system powered by the RAG framework, and annotated YOLOv8 bounding boxes.

Figure 4 highlights the Streamlit user interface, which provides an intuitive and user-friendly experience. The interface features options to upload images or use a live camera feed, with the YOLOv8 model processing the input to detect diseases like Tomato Mosaic Virus. The detected disease is displayed along with actionable insights generated through the Retrieval-Augmented Generation (RAG) framework powered by GPT-3.5 Turbo. Users can interact further by asking disease-related queries through text or audio inputs, and the platform delivers detailed, context-specific responses with high fluency and relevance.

Together, these figures underscore the accessibility, scalability, and real-world applicability of the FarmTalk platform, demonstrating its potential to revolutionize agricultural disease management by combining state-of-the-art AI models with a practical and interactive user interface.

9 Results and Discussion

9.1 Pre-Evaluation

Before model deployment, exploratory data analysis (EDA) was conducted to understand class distributions and detect potential biases in the dataset. Missing annotations and imbalanced classes were identified and addressed during preprocessing.

9.2 YOLO Evaluation

Quantitative Metrics: The results from training the YOLO on the dataset are as shown in Figure. 5. The precision, recall and mAP50 after 10 epochs were 0.993, 0.994, and 0.994 respectively. The rest of the graphs including the confusion matrix are included in the Appendix as Figure. 7 and 8 respectively.

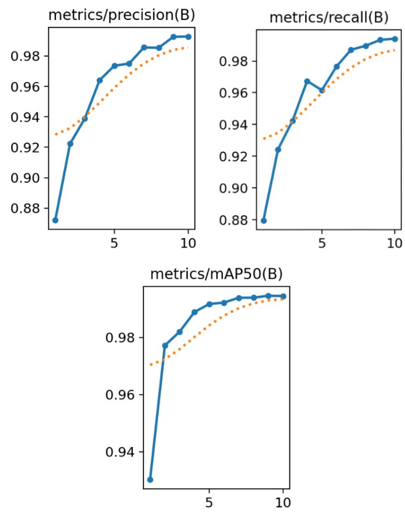

Figure 5: YOLO Results

Qualitative Analysis: Accurate detection of diseases with bounding box annotations, especially in clear, well-lit images as shown in Figure. 6

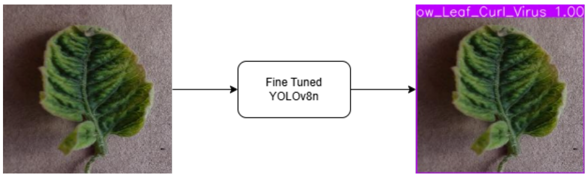

Figure 6: YOLO Inference

9.3 Error Analysis

Failures:

- Misclassifications due to overlapping leaves in dense images.
- Reduced accuracy in poor lighting conditions or non-standard image sizes.

Insights:

- These errors highlighted the importance of image quality and dataset diversity.
- Future work will include multi-modal data integration to address these challenges.

9.4 Human Evaluation RAG Framework

To assess the performance and practicality of our Retrieval-Augmented Generation (RAG) framework integrated with Large Language Models (LLMs), we conducted a detailed human evaluation using the following criteria:

- **Relevance:** Measures how well the model's responses align with the query's intent.
- **Fluency:** Evaluates the grammatical correctness and readability of the responses.
- **Factuality:** Assesses the accuracy of the information provided in the response.
- **Satisfaction:** Reflects overall user satisfaction with the response's quality.

9.4.1 Evaluation Process

We utilized a diverse set of queries relevant to plant diseases, along with irrelevant or out-of-context queries, to test the robustness of the system. A total of 50 queries spanning crops like tomato, corn, apple, grape, and potato were evaluated. Responses were rated on a scale from 1 (poor) to 5 (excellent) for each of the metrics. When faced with irrelevant queries, such as *"What is the weather in Minneapolis?"* or *"Who is the president of India?"*, the model appropriately responded with *"I don't know,"* highlighting its ability to avoid hallucination.

9.4.2 Results

The average scores achieved during the evaluation were:

- **Relevance:** 4.50
- **Fluency:** 4.52
- **Factuality:** 4.44
- **Satisfaction:** 4.52

These results indicate the model's strong performance in delivering precise, coherent, and factually accurate responses. The evaluation data sheet is available in the linked GitHub.

9.4.3 Analysis

Strengths: The system excelled in relevance and fluency, showcasing its ability to generate contextually appropriate and grammatically correct responses. It also maintained high factual accuracy, critical for practical applications in agriculture.

Challenges: Some responses to specific queries, such as detailed chemical compositions or obscure disease remedies, lacked depth. Additionally, for queries not aligned with the system's scope, the response *"I don't know"* ensured factual integrity but highlighted areas for expanding the knowledge base.

9.4.4 Illustration

An example of the evaluation:

- **Query:** *"What are the benefits of crop rotation for reducing early blight in potatoes?"*
- **Model Response:** *"Crop rotation helps reduce early blight by disrupting pathogen cycles in the soil. Rotating potatoes with non-host crops decreases Alternaria spp., the causative agent."*
- **Scores:** Relevance: 5, Fluency: 5, Factuality: 4, Satisfaction: 5

The evaluation metrics demonstrate the effectiveness of the RAG framework in generating accurate and user-satisfactory responses for plant disease management. The system's ability to refrain from guessing irrelevant answers further strengthens its reliability, making it a robust tool for agricultural applications. Future enhancements will focus on enriching the knowledge base and fine-tuning factual accuracy for edge cases.

This analysis validates the system's potential to aid farmers and agricultural experts in real-world scenarios.

9.5 Prompt Engineering

Prompt engineering was critical for aligning user queries with retrieved content. Techniques such as chain-of-thought prompting and context-specific templates improved LLM accuracy. Challenges included handling ambiguous queries, which required additional contextual clarification.

10 Failure Cases and Potential Solutions

10.1 YOLOv8 Failure Cases

Poor Lighting Conditions: YOLOv8 struggles to detect diseases in images with inadequate lighting or high contrast. For instance, plant leaf discoloration under low light can be mistaken for disease symptoms, leading to false positives or missed detections.

Potential Solution: Employ image preprocessing techniques such as histogram equalization to normalize lighting conditions or augment the dataset with diverse lighting scenarios.

Overlapping Leaves in Dense Foliage: YOLOv8 can misclassify or fail to detect diseases when multiple leaves overlap, obscuring visible disease markers.

Potential Solution: Enhance training data with more examples of overlapping foliage and incorporate higher-resolution images to improve object separation.

10.2 RAG Framework Challenges

Unsatisfactory Responses to Ambiguous Queries: While the framework performs well for direct questions, ambiguous or poorly phrased queries can lead to unsatisfactory responses. For example:

> **Query:** "Are there any resistant potato varieties?"
> **Model Response:** "I don't know."

This response reflects the model's inability to retrieve relevant information, even when it exists in the knowledge base.

Potential Solution: Implement query rephrasing mechanisms or use chain-of-thought prompting to guide the model in clarifying ambiguous inputs.

Limitations in Knowledge Base Grounding: The RAG framework sometimes retrieves information that partially answers a query but lacks specificity or depth. This is particularly evident in queries requiring precise scientific details or local context.

Potential Solution: Expand and regularly update the knowledge base with verified, domain-specific literature. Additionally, enhance the embedding model's semantic understanding with domain-tuned embeddings.

11 Conclusion

FarmTalk successfully demonstrates the potential of integrating vision models and LLMs for precision agriculture. YOLOv8 provided reliable plant disease detection, while the RAG framework enhanced context-aware remediation. The platform's user-friendly interface bridges the gap between AI capabilities and real-world agricultural needs, empowering stakeholders to make informed decisions. This project serves as a proof-of-concept for the broader application of AI in agriculture.

12 Further Discussion

12.1 Limitations

- Sensitivity to image quality and environmental conditions.
- Dependency on annotated datasets for model training.
- Occasional hallucinations in LLM outputs despite semantic retrieval.

12.2 Future Work

- Expand the dataset to include more plant species and disease categories.
- Integrate multi-modal data (e.g., hyperspectral and thermal imaging) for robust detection.
- Transition the system to edge devices for scalability in remote agricultural areas.

12.3 Replicability

FarmTalk: NEXUS ensures replicability through its open-source code and detailed documentation, accessible on GitHub:

FarmTalk-Nexus Repository

The repository provides implementation details, trained model configurations, and setup instructions for easy reproduction and further research. The website for this project can be found here.

12.4 Datasets

The platform utilizes pre-trained models and publicly available datasets, such as the Kaggle plant disease dataset, to ensure robustness and reliability without requiring custom dataset creation.

12.5 Ethics

FarmTalk addresses ethical concerns, including:

- **Bias:** Expanding datasets is essential to mitigate detection inaccuracies for underrepresented plant species.
- **Dependence:** FarmTalk is designed as a supportive tool to enhance, not replace, agricultural expertise.
- **Privacy:** Secure handling of user-uploaded images and queries ensures data protection and minimizes risks of misuse.

References

Yoeguang Hue, Jea Kim, Gang Lee, Byungheon Choi, Hyun Sim, Jongbum Jeon, Mun-Il Ahn, Yong Han, and Ki-Tae Kim. 2024. Artificial intelligence plant doctor: Plant disease diagnosis using gpt4-vision. *Research in Plant Disease*, 30:99–102.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Midhun P. Mathew and Therese Yamuna Mahesh. 2021. Leaf-based disease detection in bell pepper plant using yolo v5. *Signal, Image and Video Processing*, 16:841 – 847.

Sharada P. Mohanty, David P. Hughes, and Marcel Salathé. 2016. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7.

Dherya Singh. 2024. Vision-guided rag agent for risk prevention and root cause analysis with nvidia and langchain. Accessed: 2024-12-10.

Juan Terven, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. 2023. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, 5(4):1680–1716.

Ultralytics YOLO. Ultralytics docs.

A Appendix

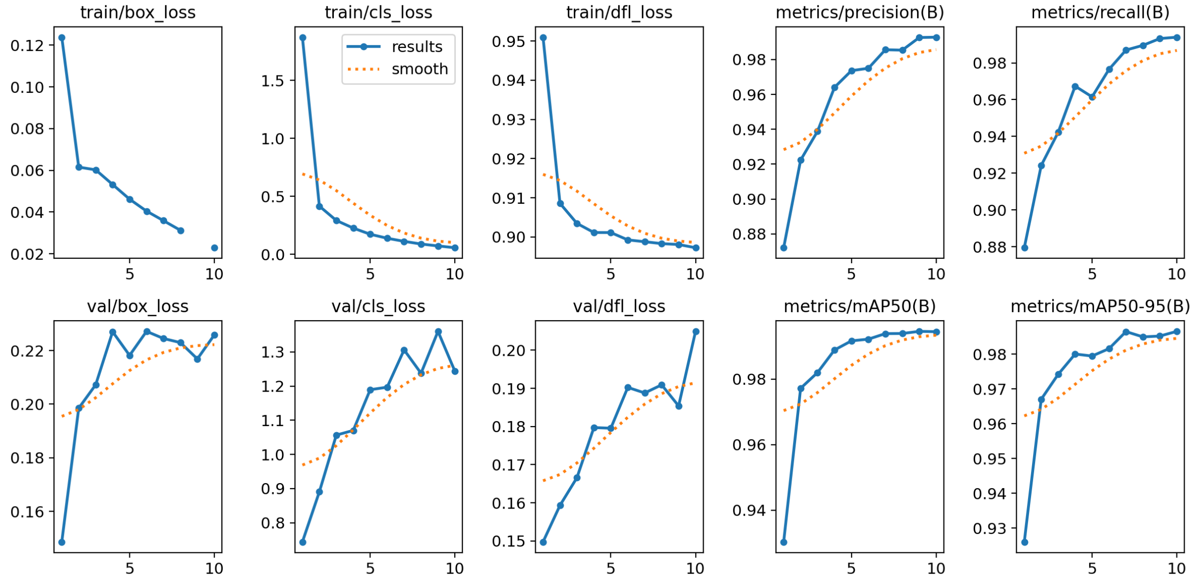

Figure 7: YOLO All Results

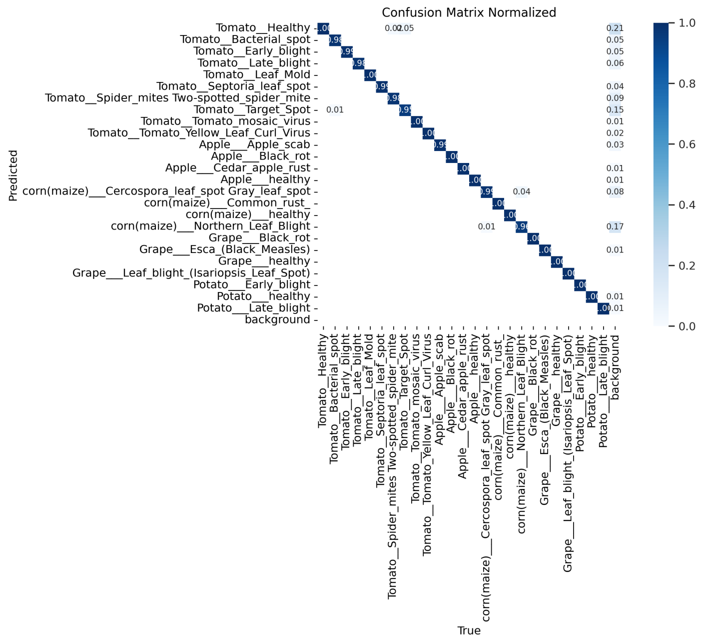

Figure 8: YOLO Confusion Matrix