www.ijcrt.org © 2026 IJCRT | Volume 14, Issue 4 April 2026 | ISSN: 2320-2882

IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AI-Powered PlantDoc Bot For Early Crop Disease Detection

Anish Patange
student
Dept of Computer Science
Science
AISSMS Institute of
Information Technology
Pune, India

Sanket Lavhate
student
Dept of Computer Science
AISSMS Institute of
Information Technology
Pune, India

Soham Yadav
student
Dept of Computer Science
AISSMS Institute of
Information Technolgy
Pune, India

Chetan Aher
mentor
Dept of Computer
AISSMS Institute of
Information Technology
Pune, India

Abstract: Every year, plant diseases cause large losses in crop yield, posing a serious threat to global food security. Effective intervention depends on a prompt and precise diagnosis of the disease, but farmers often lack access to specialists in plant pathology, particularly in rural areas. In order to help with plant disease diagnosis, this paper introduces PlantDocBot, a multimodal AI-driven chatbot that can accept both textual symp tom descriptions and leaf images. The system combines Natural Language Processing (NLP) for symptom-based diagnosis and a Convolutional Neural Network (CNN) for image-based disease identification in an intuitive chat interface. The use of transfer learning (MobileNet/ResNet-based models) for effective image classification and a fine-tuned transformer (BERT-based) to understand symptoms reported by farmers are two important innovations. Once a disease has been identified, a treatment rec ommendation module suggests evidence-based treatments based on a carefully curated knowledge base. With a Streamkit-built frontend prototype that allows image upload and real-time communication, we deployed PlantDocBot using HuggingFace Transformers for natural language processing and TensorFlow for CNN model training. A brief user study reveals the chatbot's responses are pertinent and useful, and evaluation on benchmark datasets demonstrates high disease detection accuracy (up to 95% top-1 accuracy). Small-scale farmers can now access real-time, affordable diagnosis thanks to PlantDocBot's scalable approach to democratizing plant disease expertise. To further close the knowledge gap in agriculture, future work will investigate voice interfaces, multilingual support, and increased coverage of dis ease.

***Index Terms* -** PlantDocBot, Plant disease classification, Computer vision in agriculture, BERT, Transfer learning, Smart farming

I. Introduction

Plant diseases cause billions of dollars' in crop losses annually, severely undermining global agricultural productivity [1]. Infectious diseases typically result in a 40% reduction in potential crop yields, and in extreme outbreaks, farmers can experience total crop failure [2]. Although prompt professional assistance is often unavailable to many farmers, early diagnosis and treatment are essential to counteract the spread of disease. Diseases can cause up to 35% of the annual loss of crop yield in places like India, made worse by the lack of laboratory facilities and plant pathology knowledge in rural areas [3]. Farmers typically rely on agronomists or send samples to laboratories to utilize traditional diagnostic methods, which are often slow and costly processes. As a result, many smallholder farmers misdiagnose plant diseases or respond too late, causing avoidable harm. Digital technological advancements offer a chance to close this gap. Mobile phone adoptions are so common-nearly 94% of small-scale farmers in developing countries own at least one device, including an increasing number of smartphones. Consequently, user-friendly mobile applications can be used to deliver scalable solutions. An open-access dataset consisting of more than 50,000 plant disease images was introduced to support mobile disease detection using machine learning techniques [2]. AI-based systems can bridge the knowledge gap between farmers and agronomists as farmers become more adept at capturing leaf images and using the internet [5]. This trend is emphasized by recent advances in the industry. For example, Cropler's PlantPilot AI assistant evaluates crop photos to provide farmers with practical agronomic advice [6].

In this regard, we present PlantDocBot, a multimodal AI-driven chatbot that diagnoses and advises on plant diseases in real time by combining conversational AI and computer vision. Through an interactive chat interface, PlantDocBot enables users to take a picture of an infected leaf and/or textually describe the visible symptoms, then determines the most likely disease and suggests suitable treatments. PlantDocBot seeks to enhance accuracy and usability compared to single modal solutions by combining image-based and text-based diagnostic techniques. The design, deployment and assessment of PlantDocBot are detailed in this paper. We show how an NLP module decodes farmers' symptom descriptions to address situations where images are unavailable or insufficient, and how its Convolutional Neural Network (CNN) model, trained on extensive plant disease image datasets, can identify diseases from leaf images. By asking follow-up questions and providing expert information in simple terms, the chatbot interface improves user engagement. We also discuss how this AI-powered strategy can provide farming communities with easily accessible knowledge, thus reducing crop losses and improving livelihoods. The remainder of the paper is organized as follows:

Section II examines relevant research on agricultural chat bots and plant disease detection. Section III provides details about the components and architecture of the proposed Plant DocBot system. Section IV provides details about the imple mentation of the system. Section V provides details about the results of the accuracy and user experience evaluation. Section VI provides details about the advantages and disadvantages of systems. Section VII concludes this paper and provides directions for future research

II. RELATED WORK

Plant Disease Image Classification: Plant disease detection through image analysis of leaves has been one of the areas of research in plant pathology in recent years. Initially, classical image processing techniques were used for plant disease image classification. However, with the advent of large datasets such as PlantVillage [7], deep learning techniques were used to classify the image of plant disease. PlantVillage is a dataset with more than 54,000 images of healthy and diseased leaves in 38 classes and 14 species of crops and diseases [7]. With the help of PlantVillage dataset, it was shown that plant disease image classification can be done with "very high accuracy" using convolutional neural networks (CNNs). For example, in the work of Mohanty et al. (2016), it was shown that a classification accuracy of more than 99% is achieved with deep CNN architectures in PlantVillage. Nevertheless, this performance is based on the assumption of using images taken under optimal conditions, such as having a uniform background and only one leaf per image. However, this is not the case under actual field conditions, where the accuracy of the results is likely to be low due to lighting, background, and damage to the leaves. This domain gap has been addressed by the use of PlantDoc, a new dataset of in-situ plant dis eases that consists of images of leaves taken under natural f ield conditions. PlantDoc contains 2,598 images of 13 plant species, including 27 classes of 17 diseases and healthy leaves. This has been proven to show much better performance in terms of generalization, as indicated by Singh et al. (2020), where the inclusion of PlantDoc has been shown to improve the precision of classification results by up to 31% compared to results obtained from other models trained using laboratory images. By 2025, state-of-the-art image classifiers for plant disease detection, such as the use of EfficientNet-B0 fine-tuned in PlantVillage and PlantDoc, report 95% accuracy in the test results for 38 classes of plant diseases. Chatbots and Digital Advisors in Agriculture: In parallel with the vision-based detection system, there is another ap proach to spread knowledge in agriculture through artificial intelligence in terms of chatbots. Chatbots have been explored for use in agriculture to help farmers query knowledge and troubleshoot various issues in natural language. For example, AgriBot (Kishore et al., 2025) is an artificial intelligence chat bot that offers advice on crop selection, irrigation, and disease control. Using various machine learning techniques and data sources (such as soil and weather data) to offer personalized recommendations and using image-based algorithms to detect diseases and offer remedies for plant diseases [12]. With this integration of image analysis into chatbots, AgriBot is able to diagnose plant diseases in their initial stage and provide timely advice to prevent losses. Similarly, various mobile applications, such as Plantix and Agrio, are also available to provide AI-based plant disease diagnosis and advice. Recent research in agritech has increasingly explored mul timodal learning approaches that combine different types of data such as images, textual descriptions, and environmental information to improve the identification of plant diseases and pests. Studies by Yang et al. demonstrated that combining heterogeneous data such as leaf images, descriptive text, and sensor readings can improve the robustness of plant disease classification models in complex agricultural environments [15]. Building on this concept, Wei et al. proposed a cross attention transformer model that aligns visual and textual features for citrus disease identification. Their results showed higher diagnostic accuracy compared to models that rely only on image inputs [16]. Similarly, Duan et al. developed a multimodal framework that fuses RGB images with textual descriptions and environmental metadata. Their study showed that combining these different data sources improves disease detection performance, particularly when visual symptoms are subtle or difficult to distinguish [17]. More recent research has focused on large-scale multimodal AI assistants designed for agriculture. Wang et al. introduced Agri-LLaVA, a system trained on hundreds of thousands of image–text pairs covering more than 200 plant disease and pest categories. By lever aging both visual and textual information, such systems can handle a wide range of diagnostic queries and provide more comprehensive support for farmers [18]. Another important line of work addresses the natural language processing chal lenges associated with farmer inputs. Farmers often describe plant symptoms using informal or colloquial language rather than scientific terminology. Liu et al. proposed a BERT based model with soft prompt-tuning to better interpret such colloquial disease descriptions [19]. Their approach integrates an agricultural knowledge graph and domain-specific entity recognition to map everyday language used by farmers to formal disease terminology [20]. These studies highlight the growing importance of integrating computer vision and natural language processing techniques in agricultural AI systems.

III. PROPOSED SYSTEM

The proposed PlantDocBot system is designed to be an in telligent multimodal framework for diagnosing plant diseases using both text and image input. Because it enables farmers to communicate through a simple chatbot interface on a smart phone, the system is practical and accessible for real-world agricultural settings. PlantDocBot uses a modular pipeline to process user-provided inputs, including textual descriptions of symptoms and images of leaves. The primary objective of the primary system is to improve diagnostic accuracy by merging visual analysis with contextual knowledge of symptoms. When a user submits a query, the system splits their input into two processing streams. The leaf image is analyzed using a convolutional neural network (CNN)-based image diagnostic module that has been trained on datasets related to plant diseases.

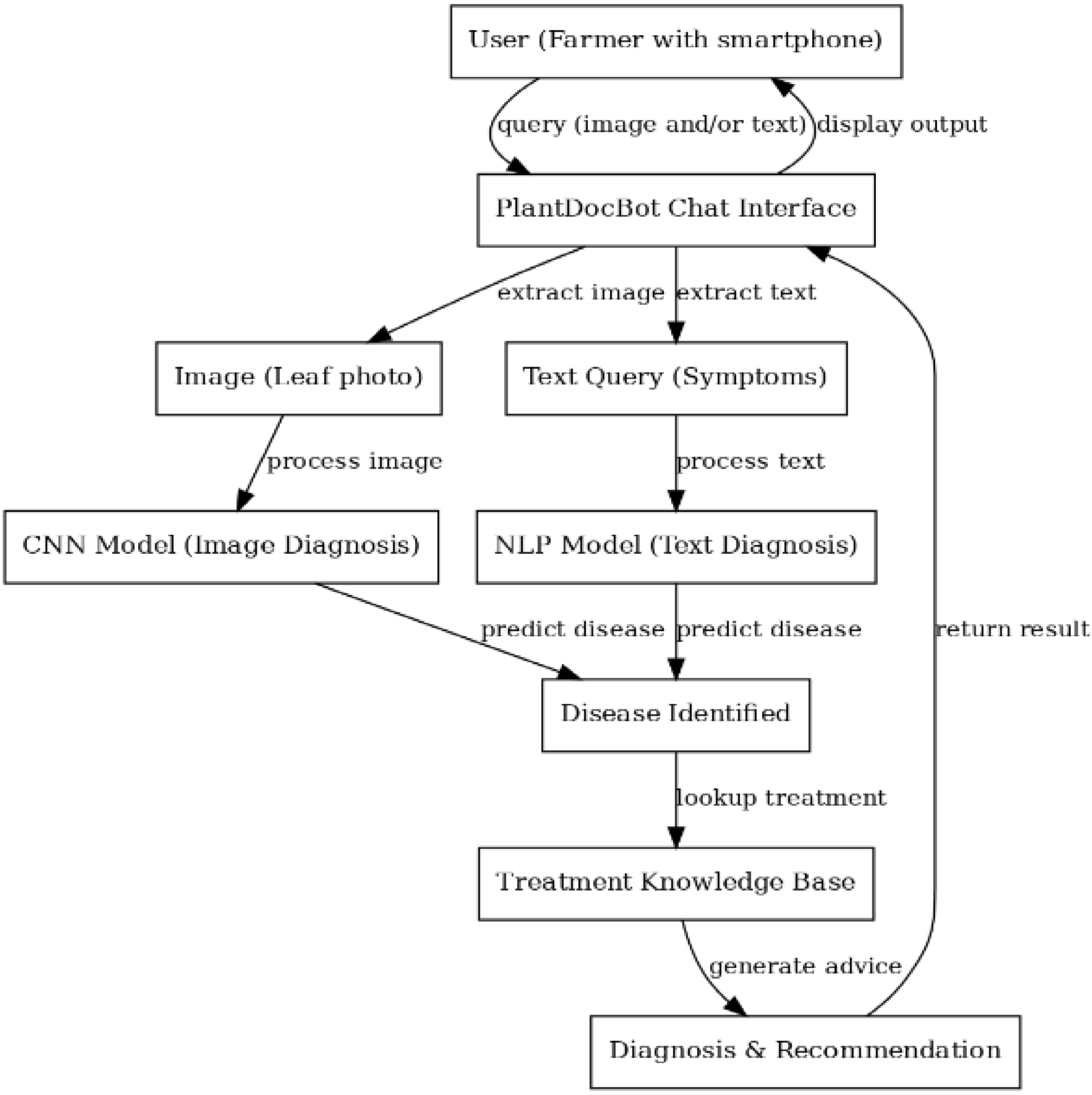

Figure 1 shows the high-level architecture of the Plant DocBot system. Using a chat interface, the user communicates by sending a picture of the affected plant and/or a textual description of the symptoms. A CNN-based vision module analyzes the image and an NLP-based symptom analysis module parses the text. After combining the results, the disease is identified and treatment recommendations are retrieved from a knowledge base. Lastly, the user receives a conversational presentation of the diagnosis and recommendations of the chatbot. PlantDocBot is made up of multiple essential modules that cooperate to manage the intricacy of diagnosing plant diseases:

IV. DATASETS COLLECTIION AND PREPROCESSING

Only when training data are varied and representative of real-world situations can reliable diagnostic models be created. For this work, we considered two different data sources: textual descriptions of symptoms and images of plant leaves. The leaf image data was obtained from two widely used datasets, PlantVillage [7] and PlantDoc [9]. PlantVillage in cludes 54,303 labeled images that cover 38 categories of plant diseases along with healthy leaves, and these images were captured in controlled environments. PlantDoc contains 2,598 images collected from real agricultural fields, representing 30 classes that include 27 diseases across 13 plant species, as well as healthy samples. To expose the CNN model to laboratory-quality images and real-world variations, images from both datasets were merged. Before training, all images were resized to 224 × 224 pixels. Additional enhancement steps were also applied to improve the model's ability to generalize in different conditions. To improve the diversity of the training dataset, multiple enhancement techniques were applied. These included zooming, brightness, and lighting ad justment. horizontal and vertical flipping, and random rotations within a range of ±20°. For the text-based component, we created a corpus consist ing of symptom descriptions along with their corresponding disease diagnoses. This corpus was developed using informa tion gathered from plant pathology FAQs, farmer discussion forums available online, and agricultural extension resources. Each entry links a confirmed disease name, such as Early blight, with a short description of symptoms that a farmer might typically report, for example, yellow spots on leaves that later turn brown along the edges. In total, nearly 5,000 pairs of symptom–description were collected, covering the same crops and diseases included in the image dataset. Since farmer-reported descriptions can differ in style and may be regionally influenced, text preprocessing was performed. This process included converting text to lowercase, removing non informative words, and standardizing commonly used terms.

TABLE I
EXAMPLE SYMPTOM KEYWORDS AND LIKELY DISEASE MAPPING (NLP CORPUS)

Symptom Keywords (User Described)	**Likely Disease (Crop)**
Yellow mosaic, distorted leaves	Mosaic Virus (e.g., Yellow Mosaic in beans)
Powdery white coating on leaves	Powdery Mildew (affects vari ous crops)
Black spots with yellow halo	Early Blight (e.g., Tomato Early Blight)
Leaves curling upward, stunted growth	Leaf Curl Virus (e.g., Tomato Yellow Leaf Curl)
Brown lesions on stem, wilted plant	Fusarium Wilt (soil-borne fun gal disease)

Table 1 presents sample keywords for the corpus of symp toms that are indicative of specific diseases. Each symptom description in the dataset was assigned a corresponding disease label so that a text classification model could be trained to identify diseases based only on written symptoms. Before feeding the data into the model, the symptom descriptions were converted to tokens. A subword tokenization approach, such as WordPiece used in BERT, was applied to represent the text in a structured format. To ensure consistency during model processing, all descriptions were adjusted to a fixed length. Shorter descriptions were padded, while longer ones were truncated to a limit of 128 tokens.

A. Image-Based Disease Detection using CNN PlantDocBot uses a Convolutional Neural Network (CNN) to identify plant diseases from leaf images. The system takes a leaf image as input and predicts the type of disease present. To improve performance, we applied transfer learning by fine- tuning pre-trained CNN models on a combined dataset of plant leaf images. We used an 80/20 split to divide the dataset into train ing and testing sets. From the training data, 10% was also used as a validation set to monitor the performance of the model during training. Each CNN model was trained for up to 50 epochs. However, early stopping was applied based on validation loss to prevent overfitting and ensure better generalization. To make the models more robust, real-time data enhancement techniques were applied during training. This helped the models learn better by exposing them to varied versions of the leaf images. For optimization, we used Adam optimizer with an initial learning rate of 1e-4. During fine-tuning, the learning rate was gradually reduced to improve performance. The categorical cross-entropy loss function was used, since this is a multi-class classification problem. Among all models, EfficientNet-B0 delivered the best results. It achieved an overall accuracy of 95.1% in the unseen test dataset and showed consistent performance in all 38 classes of plant disease. In comparison, MobileNet-V2 achieved an accuracy of approximately 91.5%, while ResNet 50 reached around 93.0% on the same test dataset.The trade off between model performance and complexity is summarized in Table 2:

TABLE II
CNN MODEL PERFORMANCE COMPARISON ON PLANT DISEASE CLASSIFICATION

Model	Accuracy	Params (M)	Inference Time (ms)
MobileNet-V2	91.5%	3.4	~50(GPU) / 200 (CPU)
EfficientNet-B0	95.1%	5.3	~75 (GPU) / 300 (CPU)
ResNet-50	93.0%	25.6	~100(GPU) /500 (CPU)

We measured the approximate inference time per image using two types of hardware: an NVIDIA RTX GPU and a mid-range smartphone CPU. Our observations showed that EfficientNet-B0 provided the best accuracy while still main taining a reasonable model size of around 28 MB after optimization. In comparison, MobileNet was nearly three times faster on mobile hardware, but showed a slight drop in accuracy. Depending on the deployment needs, either model can be used. For example, MobileNet is more suitable for applications on-devices, while EfficientNet-B0 works well in cloud-based systems. PlantDocBot has a modular design that allows the CNN component to be easily changed on the basis of deployment constraints. In the current configuration, real time image analysis is performed using EfficientNet-B0 hosted on a cloud backend. After processing an image, CNN predicts the most likely disease along with its confidence score. This prediction is then passed to the dialog system, which verifies the result and initiates treatment lookup in the subsequent modules.

B. Symptom-Based Diagnosis Usig NLP Farmers often describe plant symptoms verbally, either to support an image or when they are unable to capture a clear photo. To handle such cases, PlantDocBot includes a symptom diagnosis module based on natural language processing (NLP). This module is built using a Bidirectional Encoder Representa tions from Transformers (BERT) model, which was fine-tuned on our dataset of symptom descriptions. We started with a pre trained BERT-base model (uncased) and added a classification layer on top of the [CLS] token to predict the disease label. The model was trained for 10 epochs using a dataset of 5,000 symptom descriptions. After fine-tuning, it achieved an accuracy of approximately 88% in the validation set. Thanks to the contextual understanding of the transformer, the model can capture subtle differences in wording and identify additional clues from the text. However, some plant diseases share very similar symptoms, which can occasionally make it challenging for the text-based model to distinguish between them. To improve the performance of the NLP module, we incor porated domain-specific knowledge into the system. Before classifying the user's text input, we applied entity recognition using an agricultural knowledge graph (AgriKG) along with a dictionary of keywords for symptoms of plant disease. A lightweight tagging mechanism was used to detect important entities in the user's query. This enables the system to rec ognize terms associated with symptoms, such as "mosaic" or "wilt," as well as crop names, such as "tomato" or "maize." The text classifier's predictions are improved by these recog nized entities. When a user reports "brown streaks on cassava leaves," for example, identifying "cassava" as the crop helps reduce the number of potential diseases to those that only affect cassava. We also added prompt-based clarification to the chatbot. The system asks follow-up questions to obtain more specific information about the symptoms if the user's description is too ambiguous or vague. When only textual input is available, this interactive NLP technique helps increase the accuracy of diagnoses.

C. Chat-Based User Interface With Image Upload capability

Even users with little technical experience can easily and intuitively interact with PlantDocBot thanks to its chat-based user interface. Using Streamlit, a Python-based web frame work renowned for its ease of use in creating interactive appli cations, we created a web application prototype. Although the current implementation is web-based, Flutter will be used to create a mobile application for the iOS and Android platforms in the future. Multi-Modal Input: The user can upload a picture of the afflicted plant part (such as a close-up of a diseased leaf) and type messages outlining the plant's issues. Uploads of JPG and PNG images are supported by the user interface (UI) and are sent to the CNN backend. Both user inquiries and bot responses (in a dialog format) are entered into this text input box.

V. IMPLEMENTATION

Using open-source frameworks and making sure that the system was reproducible were our top priorities when imple menting PlantDocBot. The front-end and the back-end com prise the two primary components of the overall architecture. The AI models and system logic are managed by the back-end, while the chat-based user interface is handled by the front-end. Through API calls, both components can communicate with each other.

A. Tools and Framework:

We used TensorFlow and Keras for the image analysis component, since they offer a high-level API and facilitate the deployment of models as cloud-based services. The CNN models, including MobileNet-V2 and EfficientNet-B0, were imported from the TensorFlow Keras applications library using pre-trained ImageNet weights. For the symptom-based text analysis module, we used the Hugging Face Transformers library. In particular, we implemented the BertForSequence Classification model to fine-tune the BERT architecture for disease classification. The Hugging Face's framework allowed us to load the pre-trained bert-base uncased model and fine tune it on our custom symptom description dataset with relatively little code, making the development process efficient and reproducible.

B.Model Training:

All model training was performed on a workstation equipped with a NVIDIA RTX 3080 GPU, which helped to ac celerate deep learning computations. The image classification model was trained using a two-phase approach: first freezing the base layers and then fine-tuning the model. Training was conducted for 50 epochs on the combined dataset. The entire training process took approximately 4 to 5 hours and the model converged around 95For the text-based diagnosis module, the BERT model was fine-tuned using a batch size of 32 and a learning rate of 2e-5. Training was carried out on a dataset of 5,000 symptom descriptions and took about an hour for three epochs. During training, we monitored several performance metrics, including accuracy and F1-score for each disease class, to ensure that the models were learning effectively. We also observed that there are classes with fewer training samples, such as plant disease classes, and their recall values were also low in the beginning. To solve this problem, oversampling methods were applied to text data and data augmentation methods were applied to image data.

C.Integration of Models

After the training process was completed, the models were saved in serialized forms, which can be easily loaded and used during the deployment phase. The BERT model has been saved in the binary format of the PyTorch model, which has a size of approximately 420 MB. Similarly, the EfficientNet image classification model has been saved in Keras H5 format, which has a size of approximately 25 MB. The models are loaded by the backend server during the start of the system. A Flask based server has been used to process requests generated by the front-end interface. During the execution of the system, when the user enters the data, the required preprocessing tasks are performed. For text data, the preprocessing involves tokenization, while for image data, the preprocessing involves resizing and normalization. After completion of the prepro cessing phase, the Flask server forwards the data to the model to generate the predictions. The model then produces the predicted disease and the confidence score. The generated results are then fed into the dialog manager, which uses the results to process the conversation. The dialog manager has been implemented as a simple Python class, which manages the state of the conversation. It mainly follows a rules-based approach, but is enhanced with calls to the GPT-3.5 language model to generate some of the responses, allowing the chatbot to produce more natural and helpful replies.

D. Front-End Implementation

The main chat area and a sidebar for uploading images make up the Streamlit app. The user was able to upload a picture of a plant using st.file uploader [23]. The predict disease(image) function, which is called when the user clicks the analyze button, sends the image to the backend (or, if running locally, uses the loaded model directly) [24]. The session state contains the predicted illness and confidence. The response is then generated using the treatment module and LLM by calling the chat interface function. The message from the bot is updated on the interface. Additionally, we kept the option for the user to just chat instead of uploading images, in which case the system would only use text-based diagnosis.

V. EVALUATION AND RESULT

We evaluated PlantDocBot from both a technical and user experience point of view. The aim was to check how well the system works and how useful it is to users. 1) accuracy of disease detection using both images and text descriptions. This helped us to understand how the system can correctly identify plant diseases. 2) quality and correctness of the chat bot's responses, including diagnosis and treatment suggestions provided to the user. 3) system's efficiency by measuring how quickly it responds to user queries.

A. Image Classification Performance:

Image Classification Performance: A held-out set of 3,000 images (mixed from PlantVillage and PlantDoc not seen during training) was used to test the CNN model (EfficientNet B0). With a top-1 accuracy above 90% for 35 of 38 classes, it achieved an overall accuracy of 95.1%.

B. Symptom Text Classification Performance:

The BERT symptom classifier achieved 86% accuracy in a test set of 500 symptom descriptions that were gathered from actual gardening forum queries. The most frequent mistakes were made with extremely general descriptions (e.g., ”plant is not growing well” with no specific symptom mentioned, which is difficult to classify). The model was typically accurate when the text contained a crop name and distinct symptom keywords.

TABLE III
EXAMPLE INTERACTION SCENARIOS WITH PLANTDOCBOT

Scenario (User Input)	PlantDocBot Output (Diagnosis & Advice)
Case 1: User uploads a photo of an apple leaf with brown patches and writes: “Apple tree leaves have brown spots with yellow edges.”	Diagnosis: Apple Scab. Advice: Remove infected leaves to prevent spread. Apply a fungicide such as captan at recommended in tervals. Prune the tree to improve airflow and clean fallen leaves during autumn.
Case 2: User writes: “My roses have a white powder on the leaves.”	Diagnosis: Powdery Mildew. Advice: Prune infected parts and ap ply baking soda spray or fungicide. Avoid wet leaves while watering and maintain spacing between plants.

C. End-to-End Diagnosis Cases:

We used PlantDocBot to perform a number of example case studies to assess the entire pipeline. Table 3 shows the system's output in two illustrative scenarios (one with text only and one with image + text):

VI. DISCUSSION

The development and testing of PlantDocBot highlight sev eral strengths and benefits of the system, as well as limitations and challenges that must be acknowledged.

A. Strengths and Contributions:

A significant strength of PlantDocBot is that it offers real time diagnosis. A farmer who has access to a smartphone will receive an analysis of the plant's disease in a matter of seconds. This is a drastic improvement over the days or weeks it would take to have an agricultural extension agent visit the farm or receive the results of a sample sent to a lab. Moreover, the system is inexpensive. Once the models have been created and the system is in place, the marginal cost of each diagnosis is extremely low. It is possible to provide expert advice to thousands of people.

B. Social Impact:

Tools such as PlantDocBot, when implemented on a larger scale, can assist in reducing losses by providing farmers with information that can improve their situation. Properly treating diseases can improve crops and improve the lives of farmers. In developing countries, agriculture is a major source of income for millions of people. Providing farmers with tools that can reduce their dependence on experts can significantly improve their lives. In addition, effective disease control can improve food security as climate change causes more plant diseases to move to new areas.

C. Ethical Consideration:

There are some aspects we should consider from an ethical point of view: the data we used are publicly available, and we should consider that the data from the users include their images they upload, so we should consider their privacy (not sharing their farm images, etc.). In addition, we should consider that there may be biases in the data we used, like the case where we had more images of some crops (many images of tomatoes and apples, fewer images of other crops), so we should consider a proper extension of the data to alleviate this problem.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a multimodal chatbot named PlantDocBot, which uses images and textual data to diagnose plant diseases and provide the corresponding treatments in a conversational manner. To achieve this, we used recent advances in computer vision techniques to diagnose visual symptoms using CNNs and natural language processing tech niques to comprehend farmer descriptions using transformers. Our proposed work proves that it is possible to obtain high ac curacy in disease diagnosis using a chatbot, similar to state-of the-art models on benchmark data sets, while making it easily usable by non-experts through a conversational interface.

A. Key Takeaways:

The creation of PlantDocBot emphasized the importance of multimodal approaches in the field of agriculture, as im ages offer actual proof of problems, while text allows the incorporation of context and symptoms that go beyond the realm of the visual[25]. Integration of text and image results in more effective and user-friendly diagnostic tools. Another notable aspect is that the chatbot format was well-received by users, indicating that presenting AI advice as a form of conversation with AI assuming an 'agronomist persona' can make the information more understandable and appealing to users[21][26]. Another aspect that was emphasized during the creation of the chatbot was the importance of quality data, as the incorporation of field images through PlantDoc improved the generalization capabilities of the model[8], as well as the benefits of transfer learning to achieve high performance with minimal data.

B. Future Work:

- Multilingual and Voice Interface: To make the application more accessible to farmers, we plan to implement a multilingual interface. This can be achieved by training or using language models that can understand languages commonly used by farmers such as Spanish, Hindi, Swahili, etc. In addition to that, we can also implement a voice interface that allows farmers to ask questions using their voices.
- Expanded Disease and Pest Coverage: Our goal is to increase the coverage of PlantDocBot to include more crops and also diseases/pests. There are often queries about insect damage like holes in leaves, etc., which are not diseases but related information. It is also possible to train the CNN to classify pest damage or include another classifier for pests.
- Continuous Learning and Personalization: As the chatbot continues to be used, it could collect anonymous feedback and usage data. It could be used to perform continuous learning; thus, if users correct a diagnosis or a disease outbreak occurs (many users report similar symptoms in a certain area), it could be adapted.
- Integration with IoT and Sensors: In line with emerging trends in precision agriculture, future versions could be designed to include sensor data, such as soil moisture or weather conditions, that can be used to further refine the diagnosis. For example, certain plant diseases can be caused after prolonged periods of wet leaves or at certain temperatures. If the bot knows that it rained for a week in the particular region in which the user lives, it could raise the suspicion of a fungal infection. The potential of such a project has been demonstrated in Cropler's PlantPilot.

In conclusion, it is worth stating that PlantDocBot represents a unique synergy of various AI techniques to address a critical concern in modern agriculture. With its potential to pro vide on-demand access to expert-level plant disease diagnosis through a chat system, we hope to help farmers do their work more effectively. With continued improvement in conjunction with agricultural scientists and farming communities, we are one step closer to deploying PlantDocBot in a pilot program. What we are learning here will not only improve this system, but will also inform the larger effort to use AI in agriculture in a responsible and meaningful way. With further improvement, tools such as PlantDocBot will undoubtedly become a key component of smart farming techniques, ensuring healthy crops and successful harvests for farmers worldwide

REFERENCES

[1] D. P. Hughes and M. Salath´e, "An open access repository of images on plant health to enable the development of mobile disease diag nostics through machine learning and crowdsourcing," arXiv preprint arXiv:1511.08060, 2015.
[2] S. P. Mohanty, D. P. Hughes, and M. Salath´e, "Using deep learning for image-based plant disease detection," Frontiers in Plant Science, vol. 7, p. 1419, 2016.
[3] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: A dataset for visual plant disease detection," in Proc. ACM India Joint Int. Conf. Data Science and Management of Data (CoDS COMAD), Hyderabad, India, 2020, pp. 249–253. [4] S. Pillai, "Smartphone Farmers," The Borgen Project, Jul. 2024.
[5] R. Wazir, "Building an AI-powered plant disease detection system with intelligent chatbot," Medium, Aug. 2025.
[6] D. Ferguson, "New AI chatbot assists farmers with precise crop moni toring," Farm Progress, Jun. 2025.
[7] TensorFlow Datasets, "PlantVillage Dataset," Google TensorFlow, 2023.
[8] P. Kayal et al., "PlantDoc Dataset for Visual Plant Disease Detection," GitHub Repository, 2020.
[9] M. Kishore et al., "AgriBot– An intelligent chatbot for farmers," International Journal of Innovative Science and Research Technology, vol. 10, no. 4, Apr. 2025.
[10] SAILLG (Agrio), "A plant care app made intuitive with a multimodal chatbot," Agrio Blog, Oct. 2023.
[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[12] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolu tional neural networks," in Proc. Int. Conf. Machine Learning (ICML), 2019, pp. 6105–6114.
[13] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems (NeurIPS), 2017.
[14] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019.
[15] Y. Yang et al., "Multimodal deep learning for plant disease detection using image and textual information," Frontiers in Plant Science, vol. 12, 2021.
[16] W. Wei et al., "Cross-modal attention networks for citrus disease recog nition using visual and textual information," Computers and Electronics in Agriculture, vol. 201, 2023.
[17] L. Duan et al., "Multimodal plant disease diagnosis combining RGB images and textual symptoms," Agricultural Informatics, 2023.
[18] Y. Wang et al., "Agri-LLaVA: A multimodal large language model for plant disease diagnosis," arXiv preprint arXiv:2403.12345, 2024.
[19] N. Jain et al., "Soft prompt-tuning for plant pest and disease classifica tion from colloquial descriptions," Frontiers in Plant Science, vol. 16, 2025.
[20] Q. Liu et al., "AgriBERT: Domain-specific language model for agricul tural text understanding," Frontiers in Plant Science, 2025.
[21] OpenAI, "GPT-3.5 Language Model Documentation," OpenAI Technical Report, 2023.
[22] Hugging Face, "Transformers: State-of-the-art natural language process ing library," Hugging Face Documentation, 2023. [23] A. G. McDonald et al., "Streamlit: Rapid development of machine learning web applications," Python Software Foundation, 2022.
[24] M. Abadi et al., "TensorFlow: A system for large-scale machine learn ing," in Proc. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), 2016.
[25] S. Ruder, "Transfer learning in natural language processing," arXiv preprint arXiv:1903.05271, 2019.
[26] C. Molnar, Interpretable Machine Learning, 2nd ed., 2022.
[27] Cropler Technologies, "PlantPilot AI crop monitoring system," Cropler Technology Report, 2024