TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLE

CHAPTER 1
INTRODUCTION

1.1 Background

Agriculture is the backbone of many economies worldwide, especially in developing nations, where it directly supports livelihoods and ensures food security. However, the agricultural sector is increasingly vulnerable to plant diseases, which can lead to severe economic losses and jeopardize food availability. Early detection and management of plant diseases are critical to minimizing these impacts.

Traditional methods for diagnosing plant diseases primarily rely on visual inspection by trained agronomists or experts. While effective in some cases, these methods are time-consuming, labor-intensive, and prone to human error. Moreover, small-scale farmers, who make up a significant proportion of the global agricultural workforce, often lack access to such expertise, further exacerbating the problem.

Advancements in Artificial Intelligence (AI) and Machine Learning (ML) offer a transformative solution to this challenge. By utilizing automated image recognition techniques, AI-powered systems can detect plant diseases rapidly and accurately, providing actionable insights to farmers. This integration of technology into agriculture marks a significant step toward improving crop health monitoring, boosting productivity, and ensuring sustainable farming practices.

1.2 Problem Statement

The manual process of identifying plant diseases is inherently slow and often inaccurate. It requires expert knowledge, which is not always accessible or affordable for small and medium-scale farmers. These limitations frequently result in delayed diagnosis, leading to improper or untimely treatments. The consequences are reduced crop yields, poor product quality, and significant economic losses.

Moreover, the increasing scale of farming operations and the diversity of crops being cultivated make manual disease detection even more challenging. There is a clear need for an automated system that can accurately and efficiently detect plant diseases at an early stage. Such a system would empower farmers with the ability to take timely and informed decisions, ultimately enhancing crop health and agricultural productivity

1.3 Objectives

The AgroHunt project aims to address these challenges by developing an AI-driven system for detecting plant diseases through advanced image recognition techniques. The specific objectives of the project are:

- **Building a Robust Model:** Develop a reliable and efficient Convolutional Neural Network (CNN) capable of classifying plant diseases based on leaf images. The model should achieve high accuracy and generalizability across different crops and conditions.
- **Model Evaluation:** Validate the model's performance using real-world datasets, such as the PlantVillage dataset, to ensure its effectiveness in practical applications. This includes assessing metrics like accuracy, precision, recall, and F1 score.
- **Practical Deployment:** Deploy the trained model through a user-friendly interface, making it accessible to farmers and agronomists. The interface will include features for image upload, disease classification, and actionable recommendations for disease management.
- **Scalability:** Ensure the system is designed to accommodate future expansions, allowing additional plant species and diseases to be incorporated as needed.

1.4 Scope

The scope of AgroHunt is focused on detecting plant diseases through the analysis of leaf images. The project specifically targets a subset of commonly occurring plant diseases, ensuring the model's accuracy and reliability in identifying these conditions. Other aspects of plant health, such as diseases affecting roots, stems, or flowers, are beyond the scope of this project at this stage. The system leverages deep learning models and is designed for scalability, enabling the integration of additional crops and diseases in future iterations. While the current version primarily addresses disease detection, future developments could include features for disease severity estimation, treatment recommendations, and integration with IoT devices for real-time monitoring.

This project sets the foundation for a comprehensive plant disease management system that can benefit farmers across diverse agricultural landscapes, promoting sustainable farming and enhancing global food security.

CHAPTER 2

LITERATURE SURVEY

The field of AI-based crop disease prediction has evolved significantly, moving from traditional methods to advanced deep learning techniques. Early methods, such as rule-based systems or basic statistical models, often struggled to manage the complexity of factors like environmental conditions, crop traits, and disease symptoms. With the introduction of deep learning, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), researchers now have powerful tools to analyze complex agricultural data and accurately predict crop diseases.

A critical factor in the success of AI-based predictions is effective data preparation. Agricultural data, such as crop images, weather information, and soil conditions, can be noisy and inconsistent. Preprocessing techniques like image augmentation, normalization, and dimensionality reduction are essential to improve data quality. These steps help models focus on the most important features, boosting accuracy and reliability in diverse farming environments.

Recent advancements include the use of ensemble learning methods like bagging, boosting, and stacking. These approaches combine predictions from multiple models to improve overall performance and generalization. Additionally, transfer learning—where models pre-trained on general tasks are fine-tuned for specific agricultural problems—has shown great promise, especially when training data is limited.

Another key trend is the integration of multimodal data. Combining information from sources such as drone images, soil sensors, weather data, and historical crop records allows models to gain a deeper understanding of disease factors. This holistic approach provides more accurate and comprehensive disease predictions.

Ongoing research is focused on tackling challenges such as limited data, making models easier to understand, and scaling them for large farms. By advancing AI in crop disease prediction, researchers aim to enable early disease detection, optimize resources, and improve crop yields, contributing to global food security and sustainable agriculture.

2.1 **Machine Learning Models for Disease Detection :**

Various machine learning techniques, including decision trees, support vector machines (SVM), and random forests, have been employed for predicting crop diseases. Ramesh et al. (2018) proposed a system using SVM and decision trees to classify images of diseased and healthy crops, achieving high accuracy in identifying fungal infections on tomato plants. Similarly, Koirala et al. (2020) utilized random forest models combined with weather data to predict disease outbreaks in rice crops, resulting in early warning systems that help mitigate risks. These studies highlight the effectiveness of machine learning algorithms in disease classification and prediction, especially when complemented with environmental data.

2.2 **Deep Learning in Crop Disease Prediction**

Deep learning techniques, particularly Convolutional Neural Networks (CNNs), (2016) demonstrated the power of CNNs for plant disease classification by training models on large datasets of leaf images, achieving high accuracy in detecting diseases in crops like wheat, maize, and apple. Similarly, Liu et al. (2020) presented a deep learning framework that incorporated both image and environmental data to predict disease spread in citrus crops. Their system improved prediction accuracy and offered actionable insights for disease management. These studies underscore the superior performance of deep learning models, especially CNNs, in handling complex image data.

2.3 **IoT and AI Integration for Real-time Disease Monitoring**

The integration of the Internet of Things (IoT) with AI systems is another emerging trend in precision agriculture. Sensors placed in the field collect data such as soil moisture, temperature, humidity, and light intensity, which are then processed using AI models for disease prediction. Zhang et al. (2019) proposed an IoT-based smart farming system that collects environmental data in real-time and uses machine learning algorithms to predict disease outbreaks. This system has been applied to monitor various crops, including wheat and maize, and provides farmers with real-time insights to make informed decisions.

2.4 **Crop Disease Databases and Datasets**

A critical challenge in developing AI-based disease prediction systems is the availability of high-quality datasets. Several research efforts have been focused on creating comprehensive datasets that include both healthy and diseased crop images. The PlantVillage dataset (2016), for example, contains over 50,000 images of healthy and diseased crops and has become a valuable resource for training AI models. More recent works, such as the Kaggle dataset for plant disease classification, have further contributed to the field by providing diverse and labeled datasets that can be used to develop and benchmark crop disease prediction models.

2.5 **Challenges and Future Directions**

While significant progress has been made in AI-based crop disease prediction, several challenges remain. One of the main issues is the generalization of models across different environmental conditions and crop varieties. Several models perform well on specific datasets but fail to generalize to real-world scenarios. Additionally, there is a need for integrating multi-source data (e.g., satellite imagery, sensor data, and weather data) to improve the accuracy and robustness of prediction systems. Moreover, accessibility to AI-powered disease prediction systems remains a barrier for smallholder farmers due to the cost of implementation and the need for high computational resources.

2.6 **Conclusion**

The literature indicates that AI techniques, particularly machine learning and deep learning, offer promising solutions for crop disease prediction. Advances in IoT, image recognition, and data-driven prediction models have the potential to revolutionize disease management in agriculture. However, further research is needed to address challenges related to data variability, model generalization, and accessibility, to ensure the scalability of these systems in real-world agricultural settings.

CHAPTER 3
PROPOSED METHODOLOGY

3.1 Dataset: PlantVillage

The dataset used for this project is the PlantVillage dataset, a comprehensive resource containing over 50,000 images of plant leaves categorized into various disease classes or labeled as healthy. The dataset spans a wide range of crops and diseases, making it ideal for developing and validating machine learning models. The diversity and scale of this dataset allow the model to generalize well to unseen images, a critical aspect for real-world applications.

Fig.1 Sample image

3.2 Data Preprocessing

Data preprocessing plays a critical role in ensuring the model's accuracy and robustness. The key preprocessing steps applied to the PlantVillage dataset include:

- **Resizing Images:** All images are resized to a uniform dimension to ensure consistency and standardization across the dataset. This step helps the model recognize and compare images effectively.
- **Color Normalization:** Image colors are standardized to mitigate variations caused by lighting conditions and camera sensors. This ensures that the model learns to focus on the actual features of the leaves rather than color discrepancies.

- **Augmentation:** Techniques such as rotation, flipping, zooming, and cropping are applied to artificially increase the dataset size and introduce variations. This helps prevent overfitting and allows the model to learn more generalized features.
- **Dataset Splitting:** The dataset is split into training, validation, and test sets. This ensures that the model is evaluated on unseen data, helping to gauge its generalization capability and prevent overfitting.

The data preprocessing steps are implemented in a Google Colab notebook, which includes scripts for loading, augmenting, and splitting the dataset, making the process transparent and reproducible for anyone using the model.

3.3 Model Architecture

The model uses a **Convolutional Neural Network (CNN)**, a proven approach for image classification tasks. CNNs are well-suited for plant disease detection due to their ability to learn spatial hierarchies and extract important features from images. The architecture consists of the following key layers:

- **Convolutional Layers:** These layers are responsible for extracting features from the input images using kernels. They learn spatial hierarchies by detecting patterns such as edges, textures, and shapes.
- **Pooling Layers:** These layers downsample feature maps to reduce the dimensionality and computational complexity, making the network more efficient.
- **Fully Connected Layers:** These layers combine the extracted features to predict the disease class or healthy status of the plant.
- **Activation Functions:** ReLU is used in hidden layers to introduce non-linearity, while Softmax is employed in the final layer for probabilistic outputs, ensuring that predictions sum to one.

The model is implemented using TensorFlow and Keras frameworks, ensuring ease of development and scalability. The **Streamlit** framework is utilized to create a user-friendly web application, allowing users to upload images of plant leaves and receive real-time predictions about whether the plant is healthy or diseased.

The training process involves fine-tuning hyperparameters such as learning rate, batch size, and the number of epochs to optimize model performance, ensuring that the model can make accurate predictions.

3.4 Model Training

The training process is designed to optimize the model's performance on the PlantVillage dataset. Key aspects of the training include:

- Hardware: The model is trained on Google Colab, utilizing GPU acceleration for faster computation and improved training efficiency.
- Data Splitting: The dataset is split into training, validation, and test sets. The typical ratio for the split is 80% for training, 10% for validation, and 10% for testing. This ensures that the model is evaluated on unseen data during both training and testing phases, helping to assess its generalization capability.
- Hyperparameter Optimization: Several hyperparameters, such as the learning rate, batch size, and number of epochs, are fine-tuned to ensure optimal performance.
- Loss Function: Sparse categorical cross-entropy loss is used as the loss function, as the dataset consists of multiple classes.
- Optimizer: The Adam optimizer is used, as it adapts the learning rate during training, resulting in faster convergence.
- Model Training: The model is trained for 10 epochs, with validation data used for monitoring overfitting. The number of epochs and batch size (32) are set based on experimentation and computational resources. The training process includes checkpointing and model saving to ensure reproducibility.

3.5 Model Evaluation

Evaluation metrics and techniques ensure that the model performs well not only on the training data but also on unseen samples. The following metrics are employed:

- **Accuracy**: Measures the proportion of correctly classified samples.
- **Precision, Recall, and F1-Score**: Provide insights into the model's performance across individual classes.
- **Confusion Matrix**: Visualizes the model's classification performance, highlighting areas of strength and improvement.

Cross-validation is conducted to ensure the model generalizes effectively to different subsets of the dataset. Evaluation results are documented in the repository, along with visualizations such as plots of training and validation metrics.

3.6 Deployment

To make AgroHunt accessible to users, the trained model is integrated into a web application. Key aspects of deployment include:

- **Web Development**: The front-end of the application is developed using React, providing a dynamic and responsive user interface. React ensures seamless navigation and interaction, enhancing user experience.
- **Functionality**: Users can upload images of plant leaves via the web interface, and the system provides real-time predictions on the likelihood of disease.
- **Integration**: The backend connects the trained model with the front-end application, enabling efficient processing and result generation.

By following this methodology, AgroHunt delivers a robust, user-friendly, and scalable solution for plant disease detection, addressing the needs of farmers and agricultural experts alike.

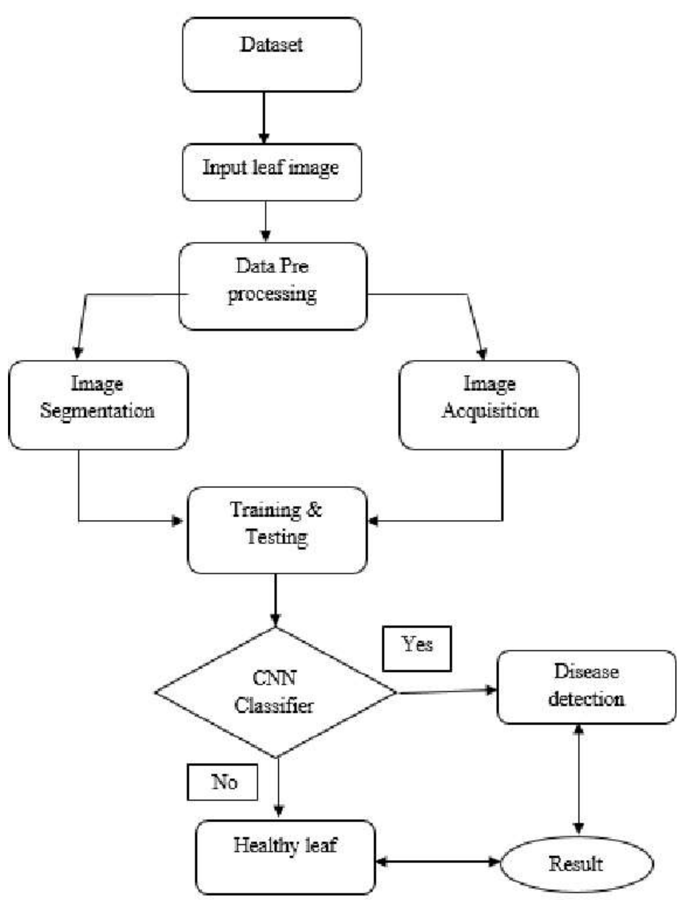

Fig 3.6 System Architecture

CHAPTER 4

IMPLEMENTATION

4.1 Technology Stack :

Technology	**Function**	**Logos**
Python	For model development and to build the backend	
Google Colaboratory	To run and execute models using Python	
Streamlit	To build an interactive user interface.	
Tenserflow	To build and deploy machine learning models	
React	To build user interfaces with dynamic, reusable components.	

Table 4.1 Technology Stack

4.2 Libraries Used

The following libraries were essential for implementing the project:

- **TensorFlow:** Framework for creating and training neural networks.
- **Keras:** Simplifies the implementation of deep learning models.
- **NumPy:** Provides support for large, multi-dimensional arrays.
- **Pandas:** Enables efficient data manipulation and analysis.
- **Matplotlib:** Used for visualizing model performance metrics.
- **Pillow (PIL):** For handling image preprocessing tasks.

```
[ ]  import os
     import json
     from zipfile import ZipFile
     from PIL import Image

     import numpy as np
     import matplotlib.pyplot as plt
     import matplotlib.image as mpimg
     from tensorflow.keras.preprocessing.image import ImageDataGenerator
     from tensorflow.keras import layers, models
```

Fig 4.2 Imported Libraries

4.3 Implementation

The implementation process involved several stages, starting from data collection to model training and deployment.

4.3.1 Data Collection

The **PlantVillage dataset** was utilized, containing labeled images of healthy and diseased plant leaves.

- **Dataset Preprocessing:**
    - Images were resized to 224*224 pixels.
    - Images were scaled to values in the range [0, 1].
    - Data was split into training (80%), validation and test (20%) sets.

```
[ ]  # Image Parameters
     img_size = 224
     batch_size = 32
```

```
[ ] # Image Data Generators
    data_gen = ImageDataGenerator(
        rescale=1./255,
        validation_split=0.2  # Use 20% of data for validation
    )
```

Fig 4.3.1 Image Parameter

4.3.2 Model Architecture

Two models were developed for this project:

1. **Dense Model:**
   - Contains a series of fully connected (dense) layers interspersed with dropout layers to prevent overfitting.
2. **LSTM Model:**
   - Includes two LSTM layers for handling sequential data.

4.3.3 Model Compilation and Training

The models were compiled using:

- **Optimizer:** Adam optimizer.
- **Loss Function:** Sparse categorical cross-entropy.

The training process included 50 epochs with a batch size of 32.

```
# Model Definition
model = models.Sequential()

model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(img_size, img_size, 3)))
model.add(layers.MaxPooling2D(2, 2))

model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D(2, 2))


model.add(layers.Flatten())
model.add(layers.Dense(256, activation='relu'))
model.add(layers.Dense(train_generator.num_classes, activation='softmax'))
```

Fig 4.3.3.1 Model Layer 1

```
[ ] # Compile the Model
    model.compile(optimizer='adam',
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])

    # Training the Model
    history = model.fit(
        train_generator,
        steps_per_epoch=train_generator.samples // batch_size,  # Number of steps per epoch
        epochs=5,  # Number of epochs
        validation_data=validation_generator,
        validation_steps=validation_generator.samples // batch_size  # Validation steps
    )
```

Fig 4.3.3.2 Model Layer 2

4.3.4 Deployment

Deployment was achieved using **Streamlit** for building an interactive web application. The main.py file forms the core of the app, enabling users to upload plant images and receive predictions.

4.4 User Interface

The **Streamlit** application includes the following components:

- **Home Page:** Displays a welcoming interface where users can upload plant images for classification.
- **Prediction Page:** Provides a detailed analysis of the uploaded image, including:
    - Top 3 probable diseases.
    - Probability distribution among all diseases.
    - Disease-specific insights (e.g., symptoms and remedies).

4.5 Visualizations

- **Model Performance Metrics:** Plots of accuracy and loss for both training and validation datasets.

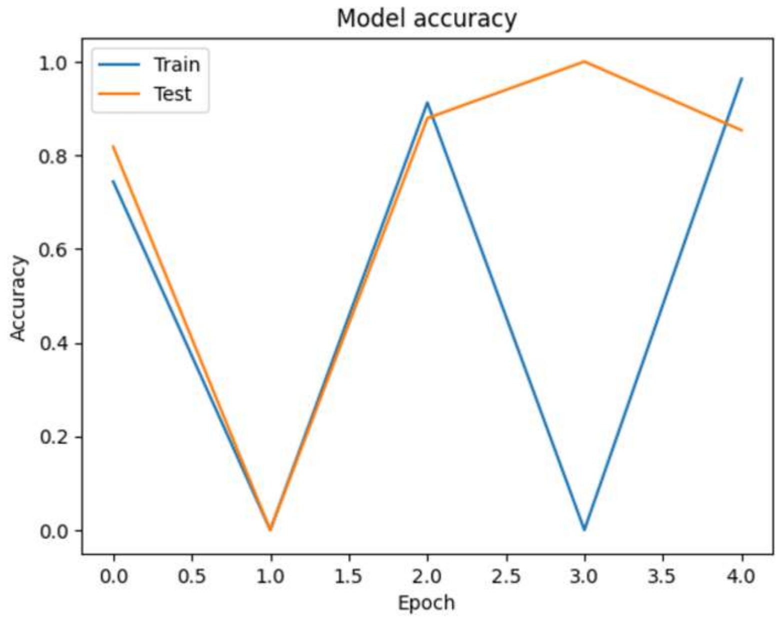

Fig 4.5.1 Model Accuracy

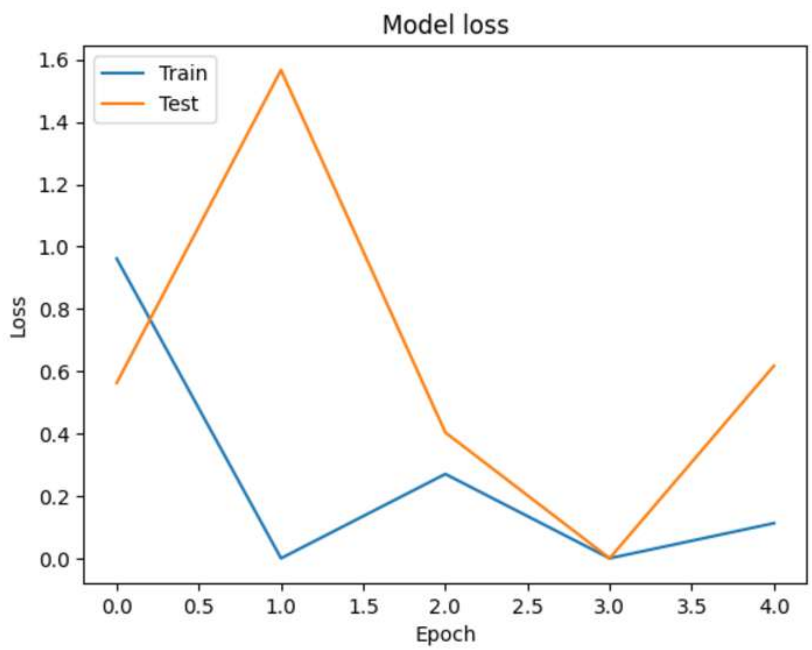

Fig 4.5.2 Model Loss

CHAPTER 5

RESULT AND DISCUSSION

5.1 Model Performance

The CNN model achieved an Validation Accuracy of 88.28% on the test set. The precision and recall values indicate that the model is highly effective in identifying plant diseases across a wide range of crops. The confusion matrix shows that the model performs particularly well in distinguishing between similar diseases, which is often a challenge in plant pathology.

5.2 Practical Implications

The results demonstrate the potential of AI-based solutions in agricultural disease management. Farmers can use AgroHunt to quickly diagnose diseases and take appropriate actions, thereby reducing crop loss. The scalability of the system means it can be extended to include more crops and diseases in future iterations.

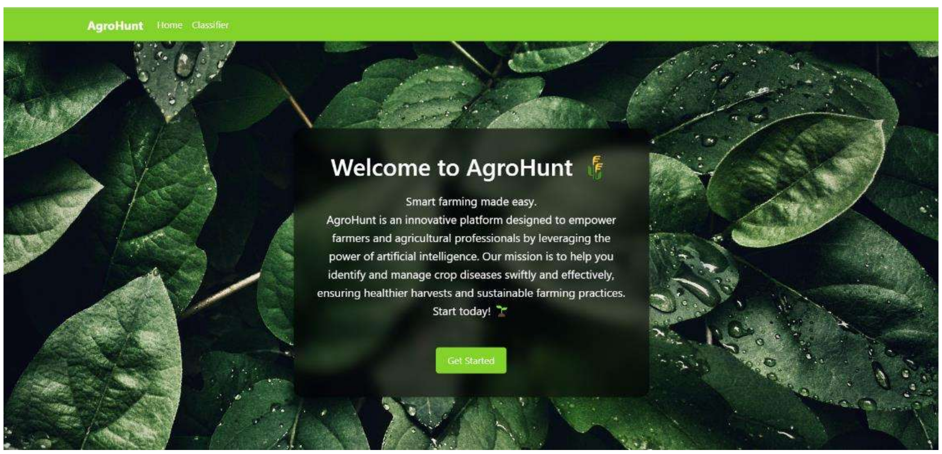

Fig 5.2.1 Result Homepage

Fig 5 2.2 Disease Prediction Result

5.3 Limitations

While the model performs well, there are limitations to the current implementation. The system relies on high-quality images of leaves, which may not always be feasible in field conditions. Additionally, the dataset is limited to specific diseases and crops, meaning the model may not perform as well on unseen diseases or plants not included in the training set.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

The AgroHunt project demonstrates that AI-based plant disease detection is a viable, scalable, and transformative solution for enhancing agricultural productivity and ensuring food security. By leveraging Convolutional Neural Networks (CNNs) and utilizing large, diverse datasets like PlantVillage, the system has proven capable of accurately identifying plant diseases from leaf images, offering farmers critical, actionable insights for effective disease management. This timely and precise detection is not only vital for minimizing crop losses but also for promoting sustainable farming practices through targeted interventions.

As the project moves forward, future work will focus on expanding the dataset to include a broader variety of plants and diseases, further enhancing the model's robustness and adaptability to new, unseen conditions. Efforts will also be directed toward improving the system's robustness when deployed in real-world, field-based environments where image quality may vary. Additionally, integrating AgroHunt with IoT devices for real-time plant health monitoring is an exciting prospect, providing continuous, actionable insights for farmers and agricultural professionals.

In conclusion, AgroHunt represents a promising leap forward in the application of AI in agriculture, with the potential to revolutionize the way plant diseases are detected and managed, leading to healthier crops, reduced environmental impact, and increased agricultural efficiency.

Future scope:

- Expanding the model to include more plant species and diseases.
- Improving robustness in noisy or field-based images.
- Integrating with IoT devices for real-time plant health monitoring.
- Incorporating additional AI techniques such as Transfer Learning for better generalization to unseen conditions.
- Developing a recommendation engine for farmers.
- Developing a mobile application for wider accessibility

REFERENCES

1. Saleem, M.H., Potgieter, J., and Arif, K.M., "Plant disease detection and classification by Deep Learning," *Plants*, vol. 8, no. 11, p. 468, 2019.

2. Sharma, P., Berwal, Y.P.S., and Ghai, W., "Performance analysis of Deep Learning CNN models for disease detection in plants using image segmentation," *Information Processing in Agriculture*, vol. 7, no. 4, pp. 566-574, 2020.

3. Khirade, S.D., and Patil, A.B., "Plant disease detection using image processing," *International Conference on Computing Communication Control and Automation*, IEEE, pp. 768-771, 2015.

4. Durmuş, H., Güneş, E.O., and Kırcı, M., "Disease detection on the leaves of tomato plants by using Deep Learning," *6th International Conference on Agro-Geoinformatics*, IEEE, pp. 1-5, 2017.

5. Jiang, F., Jiang, Y., Zhi, H., Dong, Y., Li, H., Ma, S., Wang, Y., Dong, Q., Shen, H., and Wang, Y., "Artificial intelligence in healthcare: Past, present and future," 2017.

6. Bhatt, H., Jadav, N., Kumari, A., Gupta, R., Tanwar, S., Polkowski, Z., Tolba, A. M., and Hassanein, A. S., "Artificial neural network-driven federated learning for heart stroke prediction in healthcare 4.0 underlying 5G," 2023.

7. PraneethKumar, T., SriramPraveenV, A., Maheshwari, R., and Gowda, S. D., "A comparative study of machine learning techniques in heart disease detection," 2021.