North South University

Department of Electrical & Computer Engineering

Project Proposal

Project Name: Plant Disease Detection

Course: CSE 299 **Section:** 05

Semester: Spring2025

Under Guidance of Mrs. Tanjila Farah [TnF]

Member 1 :**Zunan Nashid Tahiya** **ID: 2112014642**

Member 2 : **Pyal Saha** **ID: 2111344642**

Member 3 :**Sabrina Afrin Jaya** **ID: 2112153742**

Git Repository: https://github.com/ZunanHiya/Plant-Disease-Detection-project

Date Prepared: 25/01/2025

INTRODUCTION

The proposed research focuses on the development of a comprehensive web application for **plant disease detection**, tailored to support farmers in managing crop health more effectively. Our web application aims to combine advanced technological tools with user-centric features to provide a seamless and interactive experience. Key functionalities will include a disease detection system, a growth stage analysis module for real-time monitoring of crop development.

Additionally, the platform will serve as a community hub where farmers can share opinions, discuss challenges, and seek advice. To ensure practical utility, the app will also offer actionable treatment plans, including do-it-yourself (DIY) recipes for sustainable and cost-effective remedies. Also, provide a faster and more affordable alternative to traditional laboratory testing and expert consultations. By integrating these features, our research seeks to empower farmers with the knowledge and tools to enhance productivity, reduce losses, and foster collaboration within the agricultural community.

ABSTRACT

- The goal of this project is to develop a machine learning model that can identify diseases in plants by analyzing leaf images.
- The system aims to achieve high accuracy while being scalable and user-friendly for agricultural stakeholders.
- This project aims to empower farmers with an accessible, data-driven tool to enhance crop health, reduce dependency on chemical pesticides, and promote sustainable farming practices.
- By combining AI-driven disease detection with farmer engagement and knowledge sharing, the proposed web application aspires to bridge the gap between technology and agriculture, ultimately improving productivity and food security.
- To ensure early disease detection, accurate diagnosis, reduce manual inspection efforts by automating disease detection and robustness across different lighting, weather conditions, and plant growth stages for real-world usability.
- Also, the key point of the goal is that it is user-friendly, cost-effective, and time-efficient.

The ultimate objective is to reduce plant disease losses in order to increase agricultural output and food security.

METHODOLOGY

1. Data Collection & Preprocessing :

- Will an existing dataset called **“PlantVillage-Dataset”** of plant disease images.
- Data augmentation (rotation, zoom, flip) to improve generalization.
- ImageDataGenerator to preprocess images (rescaling pixel values).

2. Model Selection & Training :

Model Selection

- A **Convolutional Neural Network (CNN)** was chosen for its ability to effectively extract spatial features from images. It’s a machine learning algorithm for machines to understand the features of the image with foresight and remember the features to guess whether the name of the new image is fed to the machine.
- Chosen **EfficientNetB0** as the pre-trained base model for feature extraction.

  The architecture of the model includes:
    - Conv2D: 3x3 convolution with stride 2.
    - Batch Normalization.
    - Swish activation (a type of smooth, non-linear activation function).
- **Will freeze the base model and add custom classification layers (GlobalAveragePooling, Dropout, and Dense).**
- Input layer: Processes 256 x 256 images.
- Convolutional layers: Extract spatial features using filters.
- Pooling layers: Reduce the spatial dimensions to avoid overfitting.
- Fully connected layers: Perform classification.
- Output layer: Uses softmax activation for multi-class classification.
- Training the model on the dataset using categorical cross-entropy loss and the Adam optimizer.

3. Model Evaluation :

- Data was split into 80% training & 20% validation.
- Will evaluate performance using accuracy metrics.
-

4. Dynamic Data Expansion :

- A function will be introduced, **add_new_data(image_path, label)** to automatically add new images to the dataset.
- Will organize new data into appropriate disease categories.

5. Automated Model Retraining :

- After adding new data, will trigger **retrain_model()** to update the model with the expanded dataset.
- Will retrain the model with fewer epochs to prevent overfitting.

6. Model Deployment & Logging :

- At last will save the trained model as "plant_disease_model.h4".
- Will use a logging system to track new data additions & retraining events.

7. Tools and Libraries :

- **Programming Language:** Python
- **Libraries:** TensorFlow, Keras, NumPy, Pandas, Matplotlib, OpenCV, random, matplotlib, tqdm.

8. Training and Validation :

- The dataset was split into 80% for training and 20% for validation.
- Cross-entropy loss was used as the loss function.
- Early stopping was implemented to avoid overfitting.

Extend our Dataset Dynamically :

- Implemention of a user feedback system (allow users to report misclassifications).
- Enable to active learning (identify uncertain predictions and request human labeling).
- Use of semi-supervised learning (train on a mix of labeled and unlabeled data).
- Development of a crowdsourcing platform (allow farmers or researchers to upload new disease images).

This methodology ensures that our model continuously improves as new plant diseases are discovered.

LITERATURE REVIEW

1. Conventional Methods of Plant Disease Detection

Historically, plant disease detection relied on visual observation and laboratory based techniques:

- Visual Inspection: Farmers and agronomists identify diseases based on symptom characteristics such as leaf spots, discoloration, and wilting. However, this method is subjective and requires expertise.
- Microscopy and Pathogen Isolation: Laboratory techniques involve culturing pathogens on selective media, polymerase chain reaction (PCR), and enzyme-linked immunosorbent assay (ELISA) for precise identification. These methods are accurate but costly and time-consuming.
- Spectroscopy-Based Methods: Hyperspectral and multispectral imaging allow the identification of diseases based on unique spectral signatures of infected plants. While effective, these methods require expensive equipment.

2. Machine Learning Approaches

Machine learning techniques have been widely applied in plant disease detection due to their ability to analyze large datasets and recognize complex patterns.

- Support Vector Machines (SVMs): SVMs have been used for classifying plant diseases based on image features such as color, texture, and shape. For example, Pantazi et al. (2016) used SVMs to detect diseases in wheat crops with high accuracy.
- Random Forest (RF) and Decision Trees: RF classifiers have been employed to classify healthy and diseased plants based on spectral and morphological features (Picon et al., 2019).
- K-Nearest Neighbors (KNN): KNN is used for plant disease classification based on feature similarity (Singh et al., 2017). However, its performance declines with large datasets.

3. Challenges and Future Directions

Despite advancements, several challenges remain:

- Data Quality and Availability: Large, well-annotated datasets are needed for robust model training.
- Generalization Across Environments: Models trained on specific conditions may not generalize to diverse environments.
- Computational Requirements: Deep learning models require significant computational resources, which may limit deployment in resource-constrained settings.
- Integration with Precision Agriculture: Future research should focus on integrating disease detection with automated spraying and decision support systems.

TECHNOLOGY

Backend (API & Model Serving) :

- **FastAPI (Python) or Flask** to serve the ML model.
- **TensorFlow/Keras or PyTorch** for deep learning.
- **OpenCV** for image preprocessing.
- **MongoDB/PostgreSQL** (if storing user data).
- **Cloud Storage (AWS S3, Firebase, Google)** for storing images.
- **TensorFlow** just to use the tensorboard to compare the loss and adam curve our result data or obtained log.
- **numpy** to process the image matrices
- **random** to shuffle the data to overcome the biasing
- **matplotlib** to display the result of our predictive outcome.
- **tqdm** instantly make loops show a smart progress meter, just for simple design's sake

Frontend (User Interface) :

- **React.js** (with Next.js for better performance) or Vue.js for UI.
- **Tailwind CSS** for styling.
- **React Dropzone** (or simple file input) for image upload.
- **Axios** for API calls.

Machine Learning Model :

- Pre-trained **CNN** models like **ResNet, MobileNet,** or **EfficientNet.**
- **Fine-tune** on **PlantVillage** dataset
- Convert the model to **TensorFlow** Lite or **ONNX** for faster inference.

Deployment :

- **Frontend:** Vercel, Netlify, or Firebase Hosting.
- **Backend:** AWS Lambda (serverless), Render, or DigitalOcean.
- **ML Model Hosting:** Google Cloud AI, AWS SageMaker, or a custom FastAPI server.

DATASET

Data Collection :

- The dataset used for this project is the **PlantVillage dataset**, which contains over 87,000 labeled images of healthy and diseased plant leaves. The images are categorized by plant species and type of disease.
- It is categorized into 38 different classes.
- The total dataset is divided into 80/20 ratio of training and validation set preserving the directory structure.
- A new directory containing 33 test images is created later for prediction purpose.

Data Preprocessing :

- Image Resizing: All images were resized to 224x224x3 pixels for uniformity.
- Data Augmentation: Techniques such as rotation, flipping, and zooming were applied to increase the diversity of the dataset and prevent overfitting.
- Normalization: Pixel values were normalized to a range of [0, 1].

Reference

@article{Mohanty_Hughes_Salathé_2016,
title={Using deep learning for image-based plant disease detection},
volume={7},
DOI={10.3389/fpls.2016.01419},
journal={Frontiers in Plant Science},
author={Mohanty, Sharada P. and Hughes, David P. and Salathé, Marcel},
year={2016},
month={Sep}}

REFERENCES

Research Papers and Articles

- Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). "Using Deep Learning for Image-Based Plant Disease Detection." Frontiers in Plant Science.
  - DOI: 10.3389/fpls.2016.01419
- Ferentinos, K. P. (2018). "Deep learning models for plant disease detection and diagnosis." Computers and Electronics in Agriculture.
  - DOI: 10.1016/j.compag.2018.01.009
- Wallelign, S., Farooq, M. S., & Wan, L. (2022). "Plant Disease Identification Using Convolutional Neural Networks." IEEE Access.
  - DOI: 10.1109/ACCESS.2022.3153261
- https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset/data

- Chen, J., Liu, Y., & Fang, Y. (2021). Attention-based CNN for plant disease recognition. Computers and Electronics in Agriculture, 185, 106178.
- Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture, 145, 311–318.
- Mahlein, A. K., Oerke, E. C., Steiner, U., & Dehne, H. W. (2018). Hyperspectral imaging for small-scale analysis of symptoms caused by different sugar beet diseases. Plant Methods, 14, 36.
- Pantazi, X. E., Moshou, D., & Tamouridou, A. A. (2016). Automated leaf disease detection in different crop species through shape analysis and SVM classifiers. Computers and Electronics in Agriculture, 127, 312–320.
- Picon, A., et al. (2019). Machine learning approaches for plant disease detection from hyperspectral data. Remote Sensing, 11(19), 2210.
- Rajaraman, S., et al. (2019). Hybrid deep learning models for plant disease detection. IEEE Access, 7, 103032–103045.
- Ramcharan, A., et al. (2019). Mobile applications for plant disease detection using cloud computing. Computers and Electronics in Agriculture, 161, 346–358.
- Singh, V., et al. (2017). Machine learning techniques for plant disease detection: A review. Computers and Electronics in Agriculture, 142, 52–65.

STATEMENT OF LIMITATIONS

a. **Model Accuracy and Generalization**

- The accuracy of the disease detection model depends on the quality and diversity of training data. If the dataset does not cover certain plant diseases or variations in environmental conditions, the model may struggle to generalize well.
- False positives or false negatives in disease detection could lead to misdiagnosis, resulting in unnecessary treatments or overlooked diseases.

b. **Dependence on Image Quality**

- The system relies on high-quality images for accurate disease detection. Poor lighting, blurred images, or partial leaves in uploaded photos can reduce the model's accuracy.
- Farmers using low-resolution cameras or outdated devices may face challenges in obtaining reliable results.

c. **Computational Requirements**

- Running deep learning models for disease detection in real time requires significant computational resources. If hosted on cloud servers, response time might be affected by internet speed and server load.
- Processing large datasets for growth stage analysis and alerts may require high storage and processing power, increasing operational costs.

d. **Connectivity and Internet Dependence**

- Many rural areas, where farmers are the primary users, may have limited internet access, making real-time usage difficult.
- The web-based approach requires consistent connectivity, and offline functionality is limited.

CONCLUSION

The plant disease detection model developed in this project successfully identifies diseases in plants based on leaf images. By leveraging machine learning techniques, particularly deep learning models like CNNs, the system achieves high accuracy in detecting and classifying plant diseases.

Key outcomes of the project include:

- Effective Disease Detection: The model accurately classifies different plant diseases, helping farmers and agricultural experts take timely action.
- Scalability and Extendability: The system is designed to incorporate new disease data, allowing for continuous improvement and adaptation to emerging plant health issues.
- Automation and Efficiency: Compared to manual inspection, the model offers a fast and reliable alternative for disease diagnosis, reducing labor costs and increasing productivity.
- Challenges and Future Scope: Although the model performs well, challenges such as data imbalance, environmental variations, and the need for higher-quality datasets remain. Future improvements could involve transfer learning, real-time implementation in mobile applications, and integration with IoT devices for precision agriculture.

Overall, this project demonstrates the potential of AI in modern agriculture, enhancing crop health monitoring and contributing to sustainable farming practices.

THANK YOU