Plant Disease Prediction Using Machine Learning

Harsha Vardhan Reddy Manam

28-05-2023

1.Abstract

This project focuses on developing a machine learning-based system for accurately predicting and identifying plant diseases from images. The system leverages advanced image processing and machine learning techniques to diagnose plant diseases early, enabling timely and effective treatment. By providing an accessible and efficient diagnostic tool, this solution aims to support farmers, agronomists, gardening enthusiasts, and plant nurseries in mitigating crop losses and enhancing agricultural productivity. The proposed system addresses the critical need for innovative, technology-driven solutions in agriculture, contributing to improved crop health, increased yield, and sustainable farming practices.

2. Problem Statement

The goal is to develop a machine learning-based application that accurately predicts and identifies plant diseases from images. This will enable farmers to diagnose plant diseases early, ensuring timely treatment and potentially saving crop yields.

2. Assessment

Market Assessment:

- **Industry Size and Growth:**
  - The global agricultural market is expanding, driven by the increasing demand for food due to population growth.
  - The market for agricultural technology solutions is growing, with a particular focus on tools that improve yield and reduce losses.
- **Regulatory Environment:**
  - Various regions have agricultural regulations emphasizing crop health and disease management.
  - Compliance with agricultural standards and certifications can drive the adoption of advanced disease detection technologies.
- **Competitive Landscape:**
  - While traditional methods of disease detection are common, the integration of AI and ML represents an innovative approach.

    Competing products include manual inspection and traditional diagnostic tools, but this solution offers automated and precise disease identification using image analysis.

Customer Assessment:

- **Farmers:**
  - Farmers seek to minimize crop loss and improve yield by detecting diseases early.
- **Agronomists:**
  - Agronomists require accurate and timely disease identification to provide effective crop management advice.
- **Gardening Enthusiasts:**
  - Enthusiasts need easy-to-use tools to maintain the health of their plants.
- **Plant Nurseries:**
  - Nurseries aim to ensure the health of plants being sold and prevent disease spread.

Business Need Assessment:

- **Efficiency and Productivity:**
  - Early disease detection can significantly enhance crop productivity and reduce losses.
- **Cost Savings:**
  - Preventing disease outbreaks can save costs associated with crop replacement, treatment, and loss of yield.
- **Sustainability:**
  - Accurate disease detection supports sustainable farming practices by reducing the need for excessive pesticide use.
- **Market Demand:**
  - There is a growing demand for innovative agricultural technologies that support precision farming and increase overall efficiency.

3. Target Specifications

- **Image Acquisition:**
  - High-resolution image capture of plants for accurate analysis.
- **Machine Learning Models:**
  - Advanced models capable of identifying a wide range of plant diseases from images.
  - Continuous learning and updating of models with new data to improve accuracy.
- **User Interface:**
  - User-friendly interface for easy interaction and interpretation of results.
- **Reporting:**
  - Detailed reports on disease identification and recommended actions for treatment.
- **Integration:**
  - Capability to integrate with existing farm management systems for seamless operation.

4. Customer Characteristics:

- **Farmers:**
  - Looking for reliable and easy-to-use solutions to protect their crops from diseases.
- **Agronomists:**
  - Require detailed and accurate diagnostic tools to support their advisory roles.
- **Gardening Enthusiasts:**
  - Need intuitive and accessible tools for home use.
- **Plant Nurseries:**
  - Focused on maintaining plant health to ensure high-quality sales and prevent disease spread.

4. External Search

Online Information Sources/References:

- https://arxiv.org/abs/2209.07326v3
- https://arxiv.org/abs/1511.08060
- https://paperswithcode.com/paper/improving-plant-disease-classification-by

5. Bench Marking Alternate Products

Comparison with Existing Products/Services:

- Plantix: Mobile app for plant disease diagnosis.
- AgroAI: AI-based agricultural tools.
- CropSafe: Satellite-based crop health monitoring.

API's :

https://detect.roboflow.com/?model=plant-disease-detection-k6wnw&version=2&api_key='YOUR_API_KEY'

Using Code Snippet :

```
from inference_sdk import InferenceHTTPClient


CLIENT = InferenceHTTPClient(
    api_url="https://detect.roboflow.com",
    api_key="API_KEY"
)


result = CLIENT.infer(your_image.jpg, model_id="plant-disease-detection-k6wnw/2")
```

Comparison Criteria:

- Accuracy of disease detection.
- User-friendliness.
- Cost.
- Range of plant diseases covered.

Comparing different models built on dataset :

Rank	Model	Accuracy	PARAMS	F1	Paper	Code	Result	Year	Tags
1	**adaptive minimal ensembling**	100		100	Improving plant disease classification by adaptive minimal ensembling			2022	
2	**µ2Net+** (ViT-L/16)	99.89			A Continual Development Methodology for Large-scale Multitask Dynamic ML Systems			2022	
3	**Light-Chroma Inception V3**	99.48%	5M		Reliable Deep Learning Plant Leaf Disease Classification Based on Light-Chroma Separated Branches			2021	
4	**Inception V3 20%L + 80%AB**	99.48%	5M		Color-aware two-branch DCNN for efficient plant disease classification			2022	

For mode details : https://paperswithcode.com/sota/image-classification-on-plantvillage

6. Applicable Regulations

Government and Environmental Regulations:

- Compliance with agricultural data privacy laws.
- Adherence to environmental guidelines for pesticide recommendations.
- Conformity with digital health standards for agricultural tools.

7. Applicable Constraints

Constraints:

- **Space:** Minimal storage required for the mobile application.
- **Budget:** Limited initial funding; requires efficient resource management.
- **Expertise:** Need for expertise in machine learning, agriculture, and mobile app development.

8. Business Model

8.1 Subscription Fee

Charging a subscription fee involves offering users access to the app's features and functionalities in exchange for a monthly or yearly fee. This model ensures a steady stream of revenue while providing users with valuable, ongoing support for plant disease management.

Implementation:

1. **Subscription Tiers:**
    - **Basic Plan:**
        - **Cost:** $5 per month or $50 per year.
        - **Features:**
            - Access to disease identification for a limited number of plants.
            - General tips for plant care and disease prevention.
            - Basic reporting and analysis.
    - **Premium Plan:**
        - **Cost:** $15 per month or $150 per year.
        - **Features:**
            - Comprehensive disease identification covering a wide range of plants.
            - Detailed treatment plans and preventive measures.
            - Access to expert consultations and personalized advice.
            - Integration with existing farm management systems.
            - Advanced reporting and data analytics.
    - **Enterprise Plan:**
        - **Cost:** Custom pricing based on the number of users and features.
        - **Features:**
            - All Premium Plan features.
            - Customizable dashboard and reports.
            - Dedicated account manager and priority support.
            - On-site training and consultation services.

8.2 Affiliate Marketing

Affiliate marketing involves partnering with brands to recommend products related to plant care and disease treatment. The app can earn a commission on sales generated through these recommendations, creating a win-win situation for both the app and the partner brands.

8.3 Advertising

Integrate ads within the app. These can be in the form of banner ads, interstitial ads, or sponsored content. **Types of Ads:**

- Agriculture-related products and services.
- Farm equipment and supplies.
- Educational content on farming and plant care.

8.4 Data Analytics Services

Offer analytics and insights services to agribusinesses, research institutions, and governmental bodies.
Data Services:

- Aggregated and anonymized data on disease prevalence and spread.
- Trends and patterns in plant health and disease outbreaks.

8.5 Consulting and Expert Services

Provide consulting services and access to agricultural experts for personalized advice and support. **Service Packages:**

- One-on-one consultations with plant pathologists and agronomists.
- Customized crop management plans.
- On-site visits and assessments (for a higher premium).

8.6 Partnership with Insurance Companies

Partner with agricultural insurance companies to provide disease risk assessments and mitigation plans. Insurance Services**:**

- Integrate disease risk predictions into insurance plans.
- Offer discounted insurance premiums for app users who follow recommended

8.7 Crowdfunding and Grants

Seek funding from crowdfunding platforms and agricultural grants to support development and expansion. **Funding Opportunities:**

- Launch crowdfunding campaigns highlighting the benefits of the app.
- Apply for grants focused on agricultural innovation and technology.

10. Concept Generation

Personal Experience: The idea stemmed from witnessing the difficulties my father faced as a farmer. He struggled with identifying plant diseases and choosing the correct fertilizers and pesticides. An unidentified insect infestation caused substantial crop damage, and due to limited research, it took three years to find the right pesticide, leading to significant losses in harvest.

Motivation: These challenges highlighted a significant gap in the availability of tools for early disease detection and management in agriculture. The goal is to create a solution that prevents such prolonged and costly struggles for farmers, thereby enhancing crop productivity and sustainability.

Solution: The proposed solution is a mobile application leveraging advanced image processing and machine learning techniques. This app will enable farmers to diagnose plant diseases early, receive treatment recommendations, and make informed decisions about fertilizers and pesticides, ultimately supporting better crop health and higher yields.

11. Concept Development

The concept development phase aims to transform the initial idea of the plant disease prediction app into a viable and functional product. The following steps outline the process to achieve this:

11.1. Technical Feasibility

Feasibility Study:

- Conduct a feasibility study to assess the technical viability of developing a mobile application that integrates machine learning and image processing technologies for plant disease identification.
- Evaluate existing technologies, patents, and academic research related to plant disease detection using machine learning models.

Technology Assessment:

- Identify the most suitable machine learning models and image processing techniques for accurate disease detection.
- Assess the compatibility of the app with various mobile devices and operating systems to ensure widespread accessibility.

11.2. System Design

Architecture Design:

- Design the system architecture, including the integration of various components such as the user interface, image processing unit, machine learning model, and database.
- Select appropriate cloud services for data storage, processing, and machine learning model deployment.

User Interface Design:

- Develop a user-friendly interface that allows farmers to easily capture and upload images of their plants.
- Design the interface to provide clear and actionable disease identification results, along with treatment recommendations.

Data Processing and Analytics:

- Develop a central data processing unit capable of handling large volumes of image data and performing real-time analysis.
- Ensure the system can handle concurrent users and provide quick response times

11.3. Machine Learning Model Development

Data Collection:

- Collect a large dataset of plant images representing various diseases and healthy plants.
- Source data from agricultural research institutions, online databases, and field data collection.

Model Training:

- Train machine learning models using advanced algorithms in deep learning, such as Convolutional Neural Networks (CNNs), to accurately identify plant diseases.
- Implement techniques like data augmentation to enhance the model's robustness and accuracy.

Model Validation:

- Conduct extensive testing and validation of the models using a separate dataset to ensure accuracy and reliability.
- Perform cross-validation to assess the model's performance and prevent overfitting.

11.4. System Integration

Component Integration:

- Integrate the image processing unit, machine learning model, and user interface into a cohesive system.
- Ensure seamless communication between the app components and backend services.

User Experience Optimization:

- Develop features for real-time feedback and recommendations based on the model's output.
- Implement notifications and alerts for disease detection and treatment reminders.

11.5. Testing and Validation

Simulated Testing:

- Conduct extensive testing in simulated environments to evaluate the system's performance under various conditions.
- Test the app with a diverse set of plant images to ensure robustness and accuracy.

Field Testing:

- Deploy the app in real-world agricultural settings and gather feedback from farmers.
- Validate the app's effectiveness in identifying plant diseases and providing useful recommendations.

Compliance and Standards:

- Ensure the app meets all relevant agricultural and data privacy regulations.
- Conduct security audits to protect user data and ensure compliance with standards.

11.6. Commercialization strategy

Business Model Development:

- Develop a comprehensive business model, including subscription fees, freemium plans, and affiliate marketing strategies.
- Outline pricing structures for different user tiers and services.

Patenting and Intellectual Property:

- Plan for patenting the technology to protect intellectual property and prevent competitors from copying the system.
- Engage with legal experts to navigate the patent application process.

12. Final Product Prototype

Abstract :

The final product prototype of the plant disease prediction app is designed to provide a seamless and efficient user experience for identifying plant diseases and recommending treatment solutions. The app allows users to capture and upload images of plant leaves, which are then analyzed using advanced machine learning models to identify potential diseases. The system retrieves relevant treatment information from a database and provides users with recommendations, including details about pesticides and nearby shops for purchase.

Key Components of the System :

1. **User Interface (UI):**
    - Mobile app interface for users to capture and upload images of plant leaves.
    - Display area for showing disease identification results and treatment recommendations.
2. **Web Server:**
    - Manages requests between the mobile app and the application server.
    - Ensures secure and efficient data transmission.
3. **Application Server:**
    - Handles image preprocessing and interacts with the machine learning model.
    - Processes the image to prepare it for analysis by the machine learning model.
4. **Machine Learning Model:**
    - Trained to identify various plant diseases from leaf images.
    - Provides predictions based on the input image.

5. **Database:**

    - Stores information about plant diseases, treatment options, and recommended pesticides.
    - Contains details about nearby shops that offer the recommended products.

6. **Backend Services:**

    - Support the application server in preprocessing images and managing data.
    - Ensure the smooth operation of machine learning model inference and data retrieval.

Interaction Flow

1. **Image Upload:**

    - The user captures an image of a plant leaf using the mobile app.
    - The app sends the image to the web server.

2. **Request Handling:**

    - The web server forwards the image to the application server.

3. **Image Preprocessing:**

    - The application server preprocesses the image (e.g., resizing, normalization).

4. **Disease Prediction:**

    - The pre-processed image is fed into the machine learning model.
    - The model predicts the plant disease based on the input image.

5. **Information Retrieval:**

    - The application server searches the database for information related to the predicted disease.
    - Retrieves details about recommended pesticides and nearby shops.

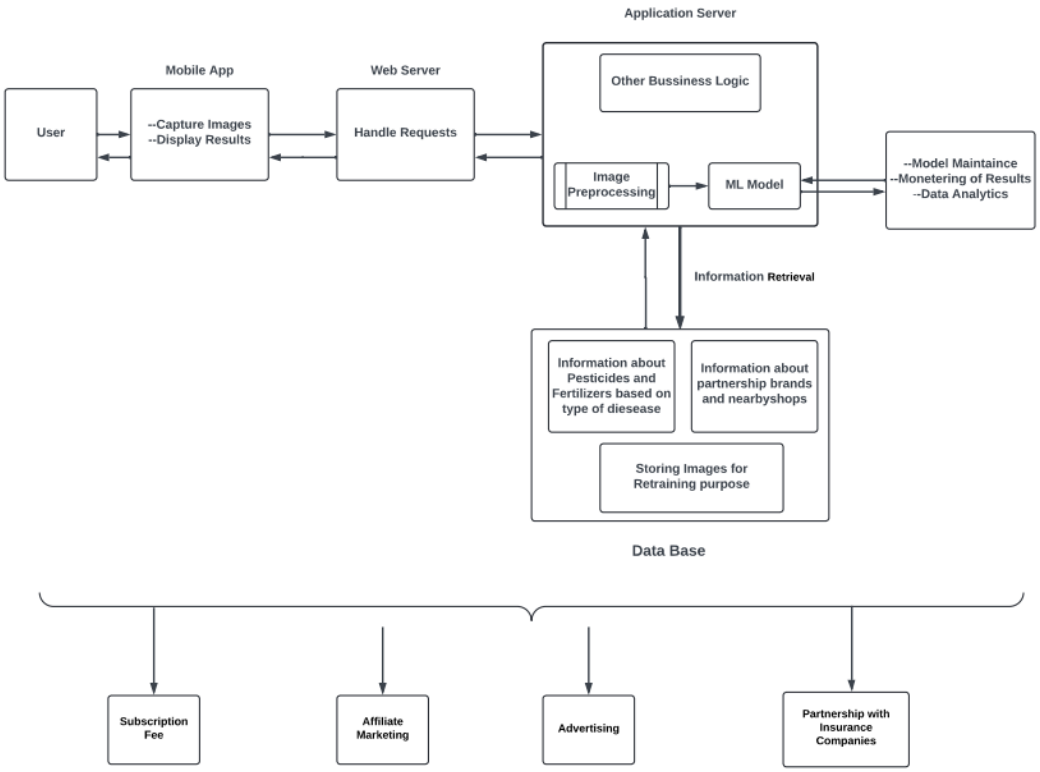
6. **Response Generation:**

    - The application server compiles the prediction results and treatment recommendations.
    - Sends the information back to the web server.

7. **Result Display:**

    - The web server forwards the results to the mobile app.
    - The app displays the disease identification and treatment recommendations to the user.

Schematic Diagram(Prototype Design)



13. Product Details

How does it work?

- The mobile application allows users to take or upload images of their plants.
- The images are processed using a machine learning model trained on a large dataset of plant images.
- The model analyzes the image, identifies any present diseases, and provides a diagnosis.
- Based on the diagnosis, the app suggests treatment options and preventive measures.

Data Sources

- Publicly available datasets of plant images with annotations for different diseases (e.g., PlantVillage dataset).
- User-submitted images to further train and improve the model.

Algorithms, Frameworks, Software Needed

- Convolutional Neural Networks (CNN) for image classification.
- TensorFlow or PyTorch for model development.
- OpenCV for image preprocessing.
- A mobile development framework like React Native for cross-platform app development.

Team Required to Develop

- Data Scientists and Machine Learning Engineers to develop and train the model.
- Software Developers with expertise in mobile app development.
- Agricultural Experts for validating the disease identification and treatment recommendations.
- UI/UX Designers to ensure the app is user-friendly.

What does it cost?

- Development Costs: Initial costs for hiring a development team, purchasing necessary software and hardware, and acquiring data.
- Maintenance Costs: Ongoing costs for app updates, server maintenance, and customer support.
- Marketing Costs: Budget for promoting the app to the target audience.

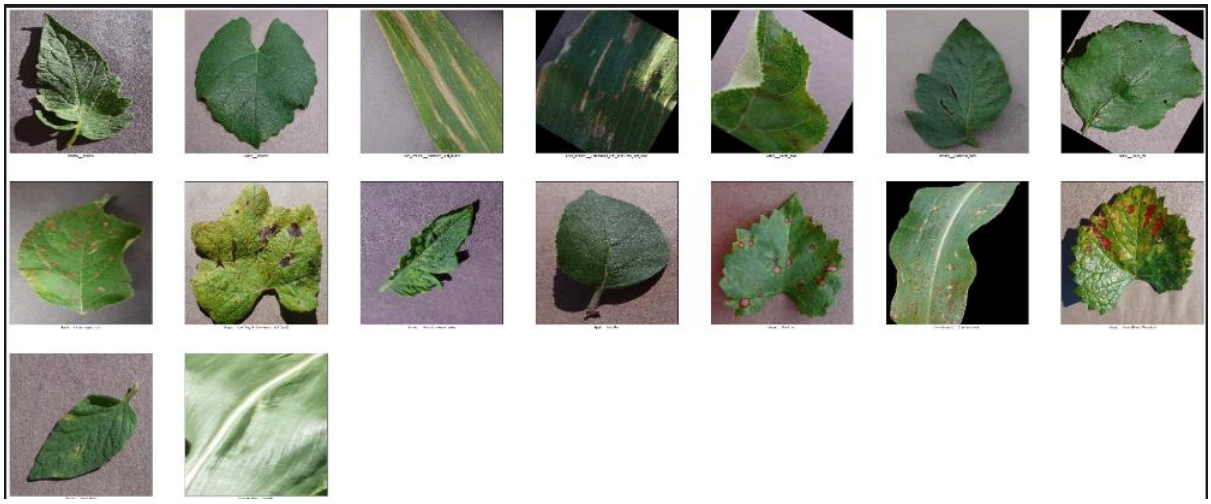
14. Code Implementation

IMPORTING NECESSARY LIBRARIES FOR TRAINING OF MODEL

```
import tensorflow
from tensorflow import keras
from keras.models import Sequential,load_model,Model
from keras.layers import Conv2D,MaxPool2D,AveragePooling2D,Dense,Flatten,ZeroPadding2D,BatchNormalization,Activation,Add,Inp
from keras.optimizers import SGD
from keras.initializers import glorot_uniform
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import ModelCheckpoint,EarlyStopping,ReduceLROnPlateau
```

Loading Dataset and Visualizing :

```
path='/kaggle/input/lacaydata/train'
plt.figure(figsize=(70,70))
count=0
plant_names=[]
total_images=0
for i in os.listdir(path):
  count+=1
  plant_names.append(i)
  plt.subplot(7,7,count)

  images_path=os.listdir(path+"/"+i)
  print("Number of images of "+i+":",len(images_path),"||",end=" ")
  total_images+=len(images_path)

  image_show=plt.imread(path+"/"+i+"/"+images_path[0])

  plt.imshow(image_show)
  plt.xlabel(i)

  plt.xticks([])
  plt.yticks([])

print("Total number of images we have",total_images)
```



Loading Resnest50 base model and Attaching Custom Classification head :

```
from tensorflow.keras.applications import DenseNet121
from tensorflow.keras.applications.resnet50 import preprocess_input
```

```
base_model_tf=DenseNet121(include_top=False,weights='imagenet',input_shape=(224,224,3),classes=num_classes)
```

```
base_model_tf.trainable = False
pt = Input(shape=(224, 224, 3))
x = base_model_tf(pt, training=False)
x = GlobalAveragePooling2D()(x)
x = Dropout(0.5)(x)
x = Dense(128, activation='relu')(x)
x = Dropout(0.5)(x)
x = Dense(64, activation='relu')(x)
x = Dense(num_classes, activation='softmax')(x)


model_main = Model(inputs=pt, outputs=x)
model_main.summary()
```

Data Augmentation :

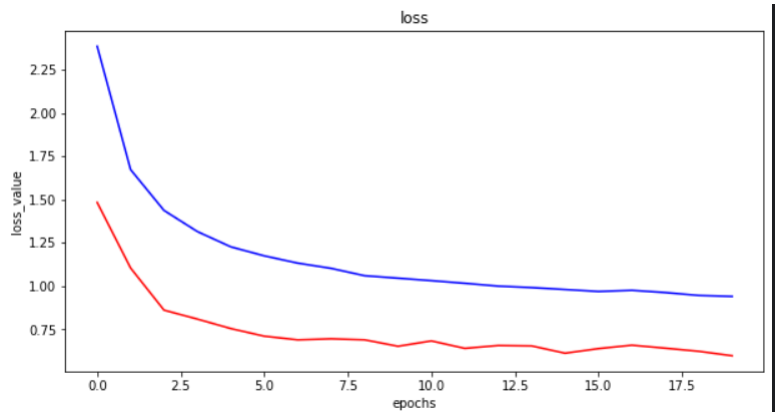
```
#Image augmentation
train_datagen= ImageDataGenerator(zoom_range=0.2,width_shift_range=0.2,height_shift_range=0.2,
                                                                    shear_range=0.2, horizontal_flip=True,fill_mode='nearest

val_datagen=ImageDataGenerator()

path_train='/kaggle/input/lacaydata/train'

path_valid='/kaggle/input/lacaydata/valid'

train= train_datagen.flow_from_directory(directory=path_train,batch_size=32,target_size=(224,224),
                                         color_mode='rgb',class_mode='categorical',seed=42,subset="training", shuffle = True

valid=train_datagen.flow_from_directory(directory=path_train,batch_size=32,target_size=(224,224),
                                        color_mode='rgb',class_mode='categorical',subset="validation", shuffle = True)

test= val_datagen.flow_from_directory(directory=path_valid,batch_size=32,target_size=(224,224),
                                        color_mode='rgb',class_mode='categorical', shuffle = False)
```

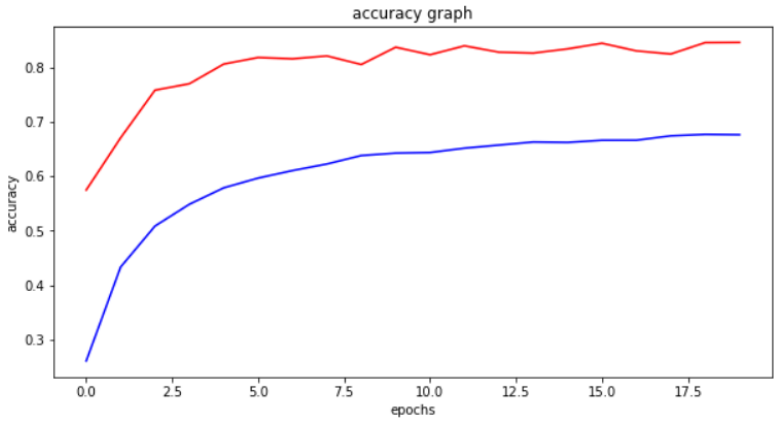
Model Training :

```
model_main.compile(optimizer='Adam',loss='categorical_crossentropy',metrics=['accuracy', 'recall', 'precision', 'f1'])
```

```
history = model_main.fit(train,validation_data=valid,epochs=20,verbose=1,callbacks=[es])
```

Model Evaluation :





GitHub Link: https://github.com/HarshaManam49/Plant-Disease-Prediction-/tree/main

15. Conclusion

This project aims to leverage machine learning to address a critical need in agriculture: early detection and treatment of plant diseases. By developing an accessible mobile application, we can empower farmers, gardeners, and nurseries to maintain healthy plants and improve crop yields. The proposed business model ensures sustainability and scalability, making it a viable solution for the agricultural community.