“Plant Disease Detection using Machine Learning ”

By

Rohan Safar

Date:08-01-2024

Abstract:

Plant diseases have a significant impact on agricultural productivity, Early and accurate detection of plant diseases is crucial for effective disease management and prevention. In this project, we propose a machine learning-based approach for plant disease detection. The objective of this project is to develop a robust and efficient system that can automatically detect and classify different types of plant diseases using machine learning algorithms. The proposed system utilizes image processing techniques to extract relevant features from plant images, such as leaves or fruits, and employs machine learning models for classification. A comprehensive dataset comprising healthy and diseased plant images is collected and annotated. Preprocessing techniques are applied to enhance image quality and remove noise. The models are trained on the extracted features and tested using a separate validation dataset to assess their performance in accurately identifying plant diseases.The results obtained from the experiments demonstrate the effectiveness of the proposed approach in detecting and classifying plant diseases. The developed system can assist farmers and agricultural experts in early disease diagnosis and timely intervention, thus mitigating the impact of plant diseases on crop yields and ensuring sustainable agriculture.

Problem Statement:

The aim of this model is to develop a machine learning model for plant disease detection. Plant diseases can cause significant damage to crops, Early detection of these diseases is crucial for timely intervention and effective management. However, manual detection of plant diseases is time-consuming and often requires specialized knowledge. Therefore, the objective of this project is to automate the process of plant disease detection using machine learning techniques.

To develop a sustainable business model centered around an innovative plant disease detection platform , the business model encompasses the following key elements , Automated Disease Detection Service , Freemium Features , Collaborations and Partnerships , Continuous Improvement and Innovation By aligning our business model with these principles, we aspire to create a lasting impact on the agriculture sector, empowering farmers, and agricultural experts with a powerful tool for disease management and prevention.

Market Overview:

Industry Growth:
The agriculture technology (AgTech) sector is expanding rapidly around the world, owing to increased awareness of precision farming and sustainable agriculture methods.

Partnerships and Collaborations:
To increase credibility and reach, we investigated prospective cooperation with agricultural research institutions, government agencies, and AgTech partners.

Farmer Needs:
Surveys and interviews were conducted with farmers to learn about their pain spots and issues in managing plant diseases.Preferences for technology adoption, convenience of use, and cost considerations were identified.

Agricultural Expert Requirements:
Engaged with agricultural specialists, agronomists, and extension services to learn about their specific disease diagnostic and management requirements. Recognized their expectations in terms of accuracy, data interpretation, and connection with existing agricultural methods.

Business Need Assessment:

Economic Impact:
Evaluated the potential economic impact of the plant disease detection system on farmers, considering reduced losses, increased productivity, and cost-effectiveness.

Regulatory and Policy Landscape:
Assessed regulatory requirements and policies related to agricultural technology and disease management. Understood any potential challenges or opportunities arising from regulatory compliance.

Business Viability:
The financial viability of the project, considering the development costs, subscription models, and potential revenue streams. Conducted a cost-benefit analysis to ensure a sustainable business model.

Target Specifications for Plant Disease Detection App:

1. Farmers:
Scale of Operation:Small to large-scale farmers will engage at the platform.
Digital Literacy: Varied levels of digital literacy; the website has an intuitive interface suitable for users with diverse technical backgrounds.
Crop Diversity: Farmers cultivating a variety of crops, necessitating a system capable of identifying diseases across different plant species.

2. Agricultural Experts:
Profession: Agronomists, extension service providers, and agricultural consultants.
Technical Proficiency: Higher level of technical expertise; expect them to utilize advanced features for in-depth analysis.
Data Integration: Desire for seamless integration with existing agricultural management systems for comprehensive insights.

Customer Characterization for Plant Disease Detection App:

Accessibility:
Device Compatibility: Ensured compatibility with a range of devices, including smartphones and tablets commonly used by farmers in the field.

Subscription Model Preferences:
Affordability: Pricing models that are affordable for individual farmers while offering scalable plans for larger agricultural enterprises.
Trial Periods: Trial periods or freemium features to encourage users to experience the app's benefits before committing to a subscription.

Environmental Impact Awareness:
Sustainability: Appeal to farmers and agricultural experts who prioritize sustainable practices by highlighting the model role in reducing pesticide use and promoting early disease intervention.

These target specifications and customer characteristics will guide the model development, ensuring it meets the diverse needs of farmers and agricultural experts while fostering widespread adoption and positive impact on sustainable agriculture.

External Research :

- Artificial Intelligence in Agriculture : Using Modern Day AI to Solve Traditional Farming Problems

  Pravar Jain — Updated On October 27th, 2023

- Artificial Intelligence (AI): The Ultimate Solution in Agriculture

  Sonia Singla — Published On December 11, 2021

- Machine learning in agriculture domain: A state-of-art survey

  Author links open overlay panel

  Vishal Meshram a, Kailas Patil a, Vidula Meshram a, Dinesh Hanchate b, S.D. Ramkteke c

  https://www.analyticsvidhya.com/blog/2021/12/artificial-intelligence-ai-the-ultimate-solution-in-agriculture/

- Siddhardhan(Machine Learning) _

  https://www.youtube.com/@Siddhardhan/featured

Benchmarking Alternate Products:

Benchmarking against existing products or services in the field of plant disease detection can provide valuable insights into market expectations, identify areas for improvement, and help refine your own product's features. Here's a general comparison with some existing plant disease detection solutions:

1. IBM Watson Decision Platform for Agriculture:
Strengths:
Utilizes AI and weather data for predictive analytics.
Offers insights into disease risk and optimal planting times.

Weaknesses:
Might be complex for smaller farmers with limited technical expertise.
Subscription costs may be higher compared to simpler solutions.

2.CABI Plantwise Knowledge Bank:
Strengths:
Provides a knowledge repository for plant health information.
Offers advisory services for farmers based on local data.

Weaknesses:
May not have a strong emphasis on real-time disease detection.
Requires collaboration with local agricultural extension services.

3. Deepfield Robotics' Bonirob:
Strengths:
Utilizes robotic technology for plant health monitoring.
Offers real-time, in-field data collection.

Weaknesses:
Hardware-intensive and may not be cost-effective for smaller farmers.
Limited to specific crops and may not cover a broad range of diseases.

Comparison for Our Plant Disease Detection :

1. Accurate Diagnosis:
Machine learning algorithms can analyze large amounts of plant image data and learn to recognize patterns and features associated with different diseases. This enables accurate disease diagnosis, often surpassing human capabilities in terms of consistency and objectivity.

2. Early Detection:
Machine learning-based systems can detect diseases at early stages before visible symptoms manifest. Early detection allows for timely intervention and preventive measures, minimizing crop damage and improving overall crop health.

3. Speed and Efficiency:
Automated disease detection using machine learning can rapidly process and analyze large volumes of plant images, providing quick results. This allows for timely decision-making and enables farmers to take immediate action to mitigate disease spread.

4. Scalability:
Machine learning models can be trained on diverse datasets, making them adaptable to different plant species and diseases. This scalability allows the system to be deployed in various agricultural settings and cater to the specific needs of different crops.

5. Cost-effective:
Once the initial development and training stages are completed, machine learning-based disease detection systems can be cost- effective. They eliminate the need for manual inspection and reduce dependence on expert knowledge, making disease monitoring more accessible and affordable for farmers.

Monetization Idea for the Plant Disease Detection Business Model:

1. Free Basic Plan:
Offer a free basic plan with essential disease detection features.
Targeted primarily at smaller-scale farmers to encourage widespread adoption.

2. Tiered Subscription Plans:
Create tiered subscription plans catering to different user segments:
Basic Plan:Free with essential features for small-scale farmers.
Standard Plan:Affordable subscription for mid-sized farms with additional features.
Premium Plan:Comprehensive subscription with advanced analytics, unlimited image storage, and priority customer support.

3.Enterprise Solutions:
Introduce enterprise-level plans for large-scale agricultural enterprises and agricultural consultancy services.
Customized solutions with integration options, dedicated support, and advanced analytics for extensive farm management.

4. Pay-Per-Use Model:
Implement a pay-per-use model for occasional users or those with smaller farms.
Users can purchase credits for specific disease detection analyses without committing to a subscription.

5. Partnership Revenue:
Explore partnerships with AgTech companies, seed suppliers, or agricultural insurance providers.
Generate revenue through collaboration, data sharing, or integrations that add value to their services.

6. Government Subsidies and Grants:
Collaborate with government agricultural departments to make the app accessible to farmers through subsidized plans or grants.
Enhances social impact and widens the user base.

7.Sponsored Content and Partnerships:
Partner with agricultural input suppliers, pesticide companies, or seed manufacturers for sponsored content within the app.
Display targeted ads or educational content, creating an additional revenue stream.

Concept generation

Here's a structured process facilitated for generation of this creative concepts:

1. Defined the Problem:
Clearly articulated the problem aims to solve: timely and accurate plant disease detection in agriculture.
Understood the challenges faced by farmers and agricultural experts in disease management.

2. Market Research:
Conducted thorough research on existing plant disease detection solutions.
Identified gaps, limitations, and opportunities for improvement in current offerings.

3. Brainstorming Sessions:
Organized brainstorming sessions with a diverse team, including experts in agriculture, technology, and machine learning.
Encouraged free-flowing idea generation without immediate judgment.

4. Idea Combining:
Combined different ideas to create hybrid concepts that may offer unique solutions.
Explored how the strengths of one idea can complement the weaknesses of another.

5. Constraints and Resources:
Considered any constraints, such as budget limitations, technological constraints, or regulatory requirements.
Identified available resources that can be leveraged for concept development.

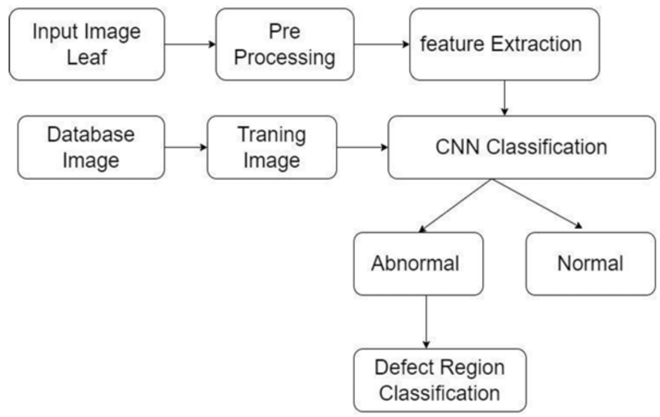
6. Technology Integration:
Explored how emerging technologies (machine learning) can be integrated for enhanced disease detection capabilities.
Considered the feasibility and impact of adopting cutting-edge technologies.

7. Final Concept Selection:
Select the final concept that aligns best with the project's objectives, user requirements, and market opportunities.

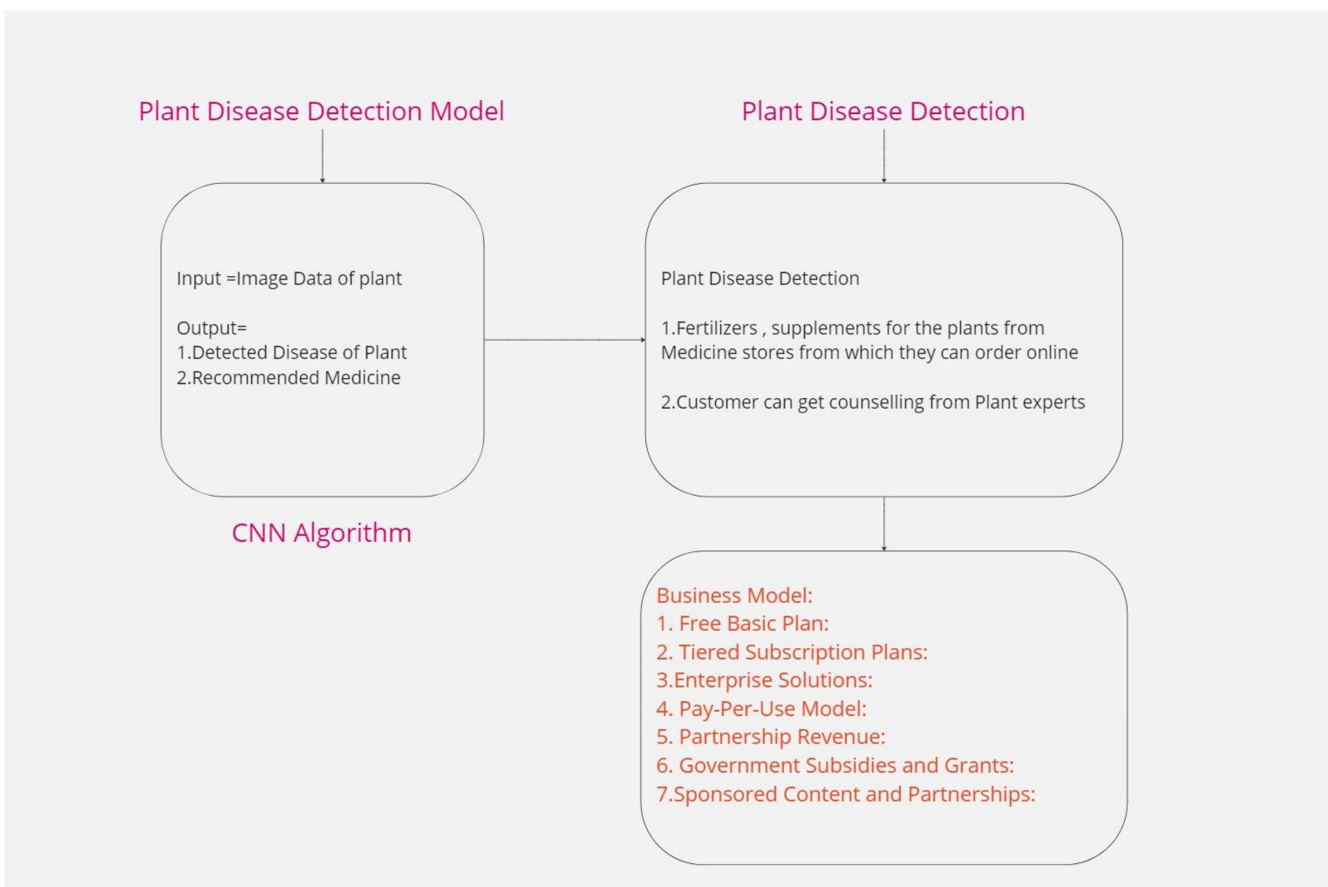
By following this systematic process, we generated and refined innovative concepts for our plant disease detection app, ensuring that the chosen idea addresses real-world challenges effectively.

Final Product Prototype :

Working Flow of Project Model:



Business Model Prototype:



Product Details:

Data Source:

In this data-set, 39 different classes of plant leaf and background images are available. The data-set contains 61,486 images. We used six different augmentation techniques for increasing the data-set size. The techniques are image flipping, Gamma correction, noise injection, PCA color augmentation, rotation, and Scaling.

There are a total of 39 Classes that we have to predict using the CNN Model.

https://data.mendeley.com/datasets/tywbtsjrjv/1

Prerequisites :

PyTorch

Convolutional Neural Networks

Working:

The working of a plant disease detection project:

1. Data Collection: Gather a dataset of images or samples of healthy and diseased plants. This dataset should cover a variety of plant species and different types of diseases.

2. Preprocessing: Clean and preprocess the collected data to enhance its quality and remove any irrelevant information. This may involve resizing, cropping, or filtering the images to focus on the relevant parts.

3. Feature Extraction: Extract meaningful features from the preprocessed data. These features could include color, texture, shape, or other characteristics that can help differentiate between healthy and diseased plants.

4. Training: Use machine learning or deep learning techniques to train a model using the extracted features and the corresponding labels (healthy or diseased). The model learns the patterns and characteristics of different diseases and their effects on plants.

5. Validation: Evaluate the performance of the trained model using a separate validation dataset. This step helps assess the model's accuracy, precision, recall, and other performance metrics.

6. Testing: Apply the trained model to new, unseen plant samples to detect diseases. This involves inputting the features of the test samples into the model and obtaining predictions about their health status.

7. Disease Classification: Based on the predictions of the model, classify the plant samples as healthy or diseased. If a disease is detected, the specific type of disease can also be identified based on the trained model's output.

8. Decision Support: Provide recommendations or suggestions for disease management based on the detected disease. This could include suggesting appropriate treatments, recommending specific actions to mitigate the spread of the disease, or advising on the optimal time for intervention.

<u>Algorithms , frameworks Used:</u>

Convolutional Neural Network:
Once preprocessing is done, then CNN is used for training purposes and after that we get a trained model. That CNN method is written with the help of tensor flow. By using this model, we classify the image that the system is getting after pre- processing of the testing image. Then we get a particular disease name or status of a healthy leaf if there is no disease on that leaf. With the help of that disease name, we get the remedies which will help the farmer to take action so as to eradicate or decrease the effect of disease. Training and Testing Algorithm Input: providing an image of leaves localization Output: classification of a review into healthy or diseased, it is diseased provides the remedies for overcoming the deficiency.

Step1: Start
Step2: prepare a database (healthy or diseased)
Step 3: preprocessing normalization
Step4: Train CNN Step5: real images from Google or dataset
Step6: preprocessing
Step7: test network
Step8: if the probability of healthy > probability of unhealthy display a healthy leaf, otherwise display a diseased leaf.
Step9: go to the fourth step
Step10: stop

Code Implementation :

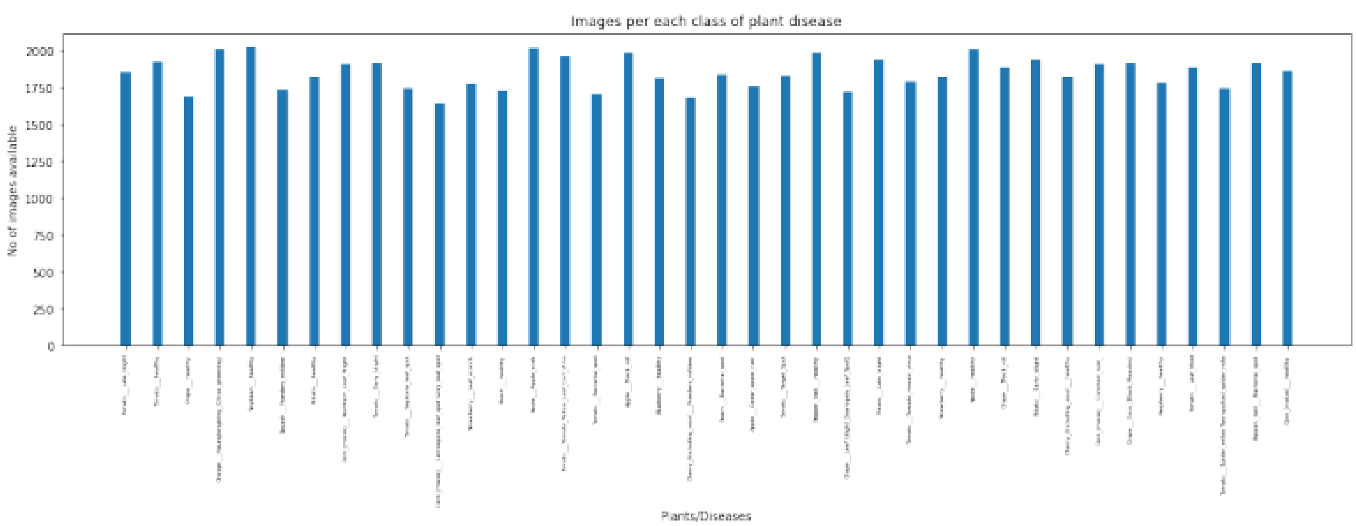
Github Repository link :
https://github.com/RohanIncantato/Plant-Disease-Detection-Business-Model..git

Import Dependencies:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
import torch
from torchvision import datasets, transforms, models  # datsets  , transforms
from torch.utils.data.sampler import SubsetRandomSampler
import torch.nn as nn
import torch.nn.functional as F
from datetime import datetime
```

Visualizing the above information on a graph:



Transforms are used for Data Augmentation like cropping the image, resizing the image, converting the image to tensor, rotate the image, and many more. Transforms work as a filter for all images. We are using the following code to transform the images.

```
transform = transforms.Compose(
    [transforms.Resize(255), transforms.CenterCrop(224), transforms.ToTensor()]
)
```

We use the dataset for making our own dataset from images and for that we are using the ImageFolder method from datasets.
The folder where I stored all images is named ‘Dataset’. We also apply our transform to all images.

```
dataset = datasets.ImageFolder("Dataset", transform=transform)
```

Train Test Split :

```
train_indices, validation_indices, test_indices = (
    indices[:validation],
    indices[validation:split],
    indices[split:],
)
```

```
train_sampler = SubsetRandomSampler(train_indices)
validation_sampler = SubsetRandomSampler(validation_indices)
test_sampler = SubsetRandomSampler(test_indices)
```

```
targets_size = len(dataset.class_to_idx)
```

Here in the above code we first get indices and then split the data into train , test and validation data. Total 36584 for train , 15679 for validation and remaining images for testing.

```
train_sampler = SubsetRandomSampler(train_indices)
validation_sampler = SubsetRandomSampler(validation_indices)
test_sampler = SubsetRandomSampler(test_indices)
```

SubsetRandomSampler is used to sample our data. Here we are creating an object of the SubsetRandomSampler Object and later we will use this sampler in the train data loader and test data loader.

```
batch_size = 64
train_loader = torch.utils.data.DataLoader(
    dataset, batch_size=batch_size, sampler=train_sampler
)
test_loader = torch.utils.data.DataLoader(
    dataset, batch_size=batch_size, sampler=test_sampler
)
validation_loader = torch.utils.data.DataLoader(
    dataset, batch_size=batch_size, sampler=validation_sampler
)
```

As we discussed we use train_sampler for train-loader and vice-versa. Now our dataset is ready for training and testing.

Model Creation : We use a convolutional neural network for model creation. We create model layers as mentioned below the image. We also specified filter size for the Conv layer and Pool layer and the shape on each layer. shape = (channels , height , width) .

In PyTorch shape is not automatically calculated we manually have to take care of shape on each layer. At the First Fully connected layer we have to mention output size as per the shape of the convolutional layer. This calculation is also called Convolutional Arithmetic. Here is Equation for Convolutional Arithmetic:

Shape:

- Input: $(N, C_{in}, H_{in}, W_{in})$
- Output: $(N, C_{out}, H_{out}, W_{out})$ where

$$H_{out} = \left\lfloor \frac{H_{in} + 2 \times \text{padding}[0] - \text{dilation}[0] \times (\text{kernel\_size}[0] - 1) - 1}{\text{stride}[0]} + 1 \right\rfloor$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2 \times \text{padding}[1] - \text{dilation}[1] \times (\text{kernel\_size}[1] - 1) - 1}{\text{stride}[1]} + 1 \right\rfloor$$

Here for this project dilation = 0.

```
model = CNN(targets_size) # targets_size = 39
```

Here we have to classify the images into 39 Categories so that’s why we used categorical cross-entropy as loss and adam optimizer. In the Model, we are using ReLU as activation but for the last layer, we have to use Softmax activation. In PyTorch, we have a cross-entropy loss which is a mixture of softmax and category cross-entropy loss.

```
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters())
```

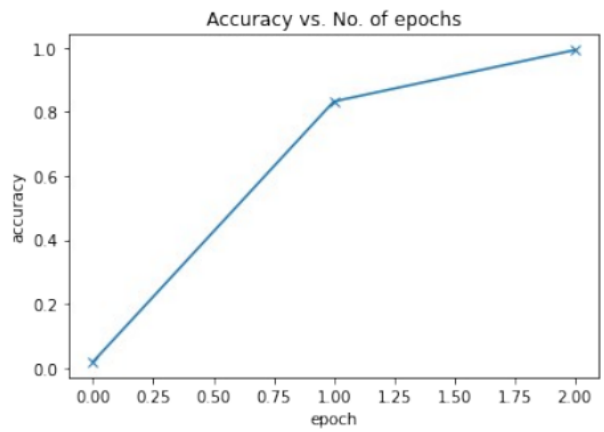
Batch Gradient Descent:

```
def batch_gd(model, criterion, train_loader, test_laoder, epochs):
    train_losses = np.zeros(epochs)
    validation_losses = np.zeros(epochs)

    for e in range(epochs):
        t0 = datetime.now()
        train_loss = []
        for inputs, targets in train_loader:
            inputs, targets = inputs.to(device), targets.to(device)

            optimizer.zero_grad()

            output = model(inputs)

            loss = criterion(output, targets)

            train_loss.append(loss.item())  # torch to numpy world

            loss.backward()
            optimizer.step()

        train_loss = np.mean(train_loss)

        validation_loss = []

        for inputs, targets in validation_loader:

            inputs, targets = inputs.to(device), targets.to(device)

            output = model(inputs)

            loss = criterion(output, targets)

            validation_loss.append(loss.item())  # torch to numpy world

        validation_loss = np.mean(validation_loss)

        train_losses[e] = train_loss
        validation_losses[e] = validation_loss

        dt = datetime.now() - t0

        print(
            f"Epoch : {e+1}/{epochs} Train_loss:{train_loss:.3f} Test_loss:{valic
        )

    return train_losses, validation_losses
```

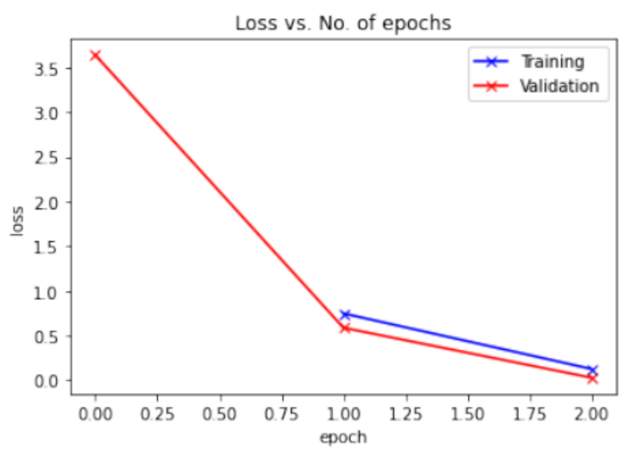
As you can see above, the function is used for Batch Gradient Descent. batch_gd() is the function where all the learning happened.

Validation Accuracy:

```
plot_accuracies(history)
```



Validation loss:

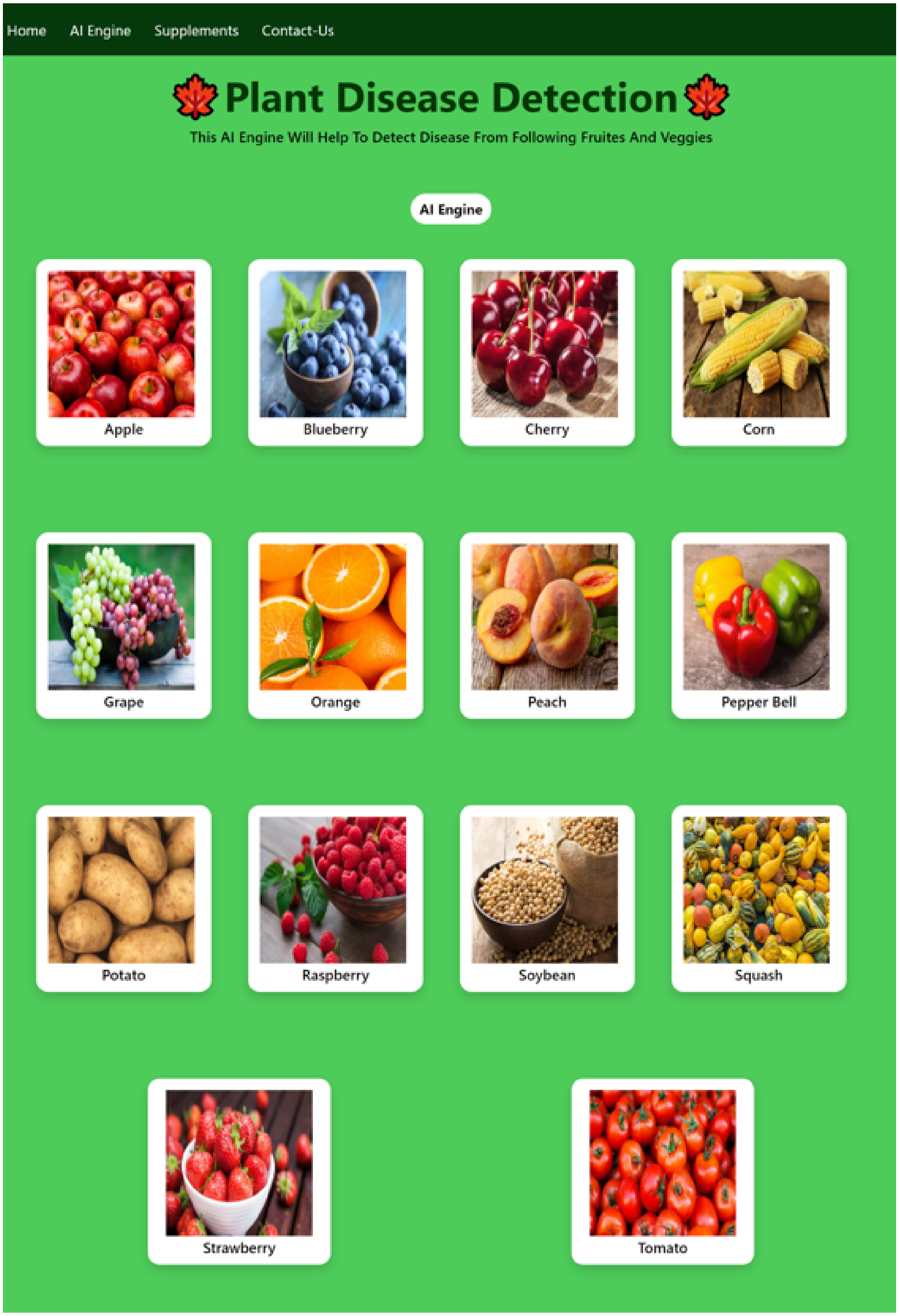


Accuracy:

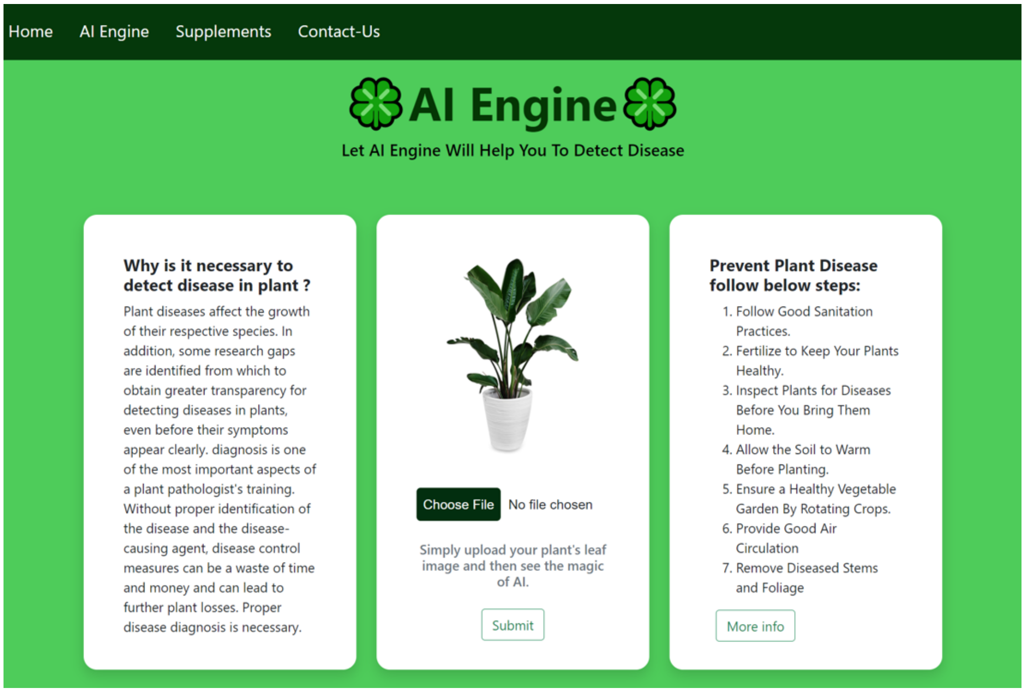
```
def accuracy(loader):
    n_correct = 0
    n_total = 0

    for inputs, targets in loader:
        inputs, targets = inputs.to(device), targets.to(device)

        outputs = model(inputs)

        _, predictions = torch.max(outputs, 1)

        n_correct += (predictions == targets).sum().item()
        n_total += targets.shape[0]

    acc = n_correct / n_total
    return acc
```

By Using this Current Model we are getting accuracy 87 % on Train data, 84 % On Validation data and 83% on Test data. For this practical we use test data as completely new data. We did not use test data on any phase; it was made only for final testing.

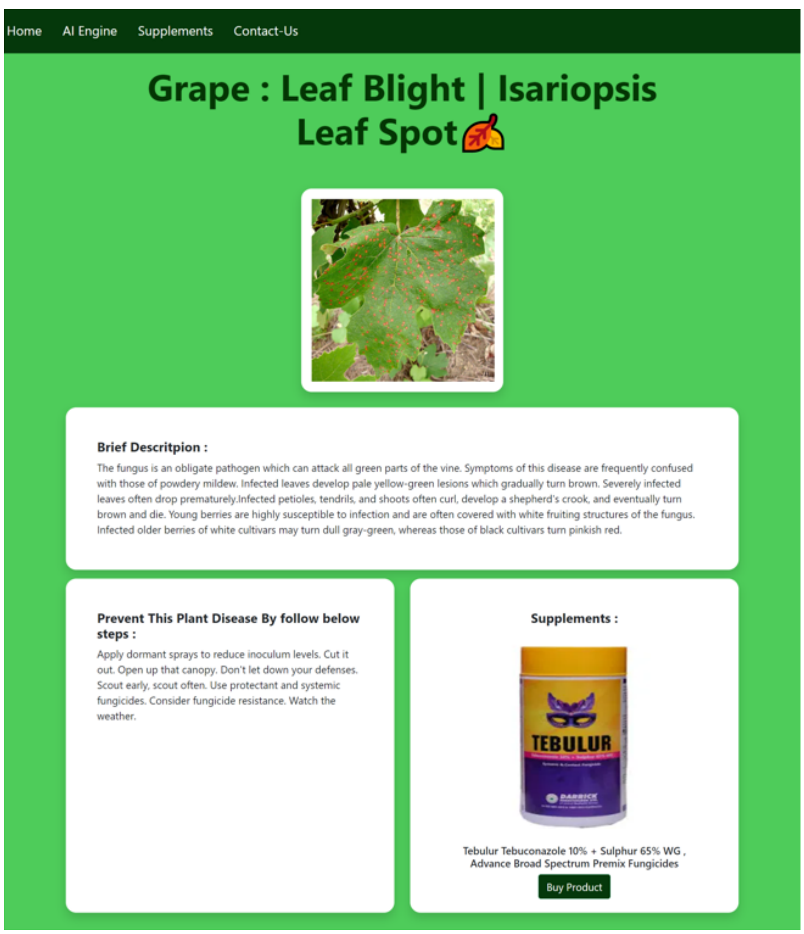
Snippet of web app:



AI Engine:


Home
AI Engine
Supplements
Contact-Us
AI Engine
Let AI Engine Will Help You To Detect Disease
Why is it necessary to detect disease in plant ?
Plant diseases affect the growth of their respective species. In addition, some research gaps are identified from which to obtain greater transparency for detecting diseases in plants, even before their symptoms appear clearly. diagnosis is one of the most important aspects of a plant pathologist's training. Without proper identification of the disease and the disease-causing agent, disease control measures can be a waste of time and money and can lead to further plant losses. Proper disease diagnosis is necessary.
Choose File
No file chosen
Simply upload your plant's leaf image and then see the magic of AI.
Submit
Prevent Plant Disease follow below steps:
1. Follow Good Sanitation Practices.
2. Fertilize to Keep Your Plants Healthy.
3. Inspect Plants for Diseases Before You Bring Them Home.
4. Allow the Soil to Warm Before Planting.
5. Ensure a Healthy Vegetable Garden By Rotating Crops.
6. Provide Good Air Circulation
7. Remove Diseased Stems and Foliage
More info

Result Page:


Home
AI Engine
Supplements
Contact-Us
Grape : Leaf Blight | Isariopsis Leaf Spot
Brief Descritpion :
The fungus is an obligate pathogen which can attack all green parts of the vine. Symptoms of this disease are frequently confused with those of powdery mildew. Infected leaves develop pale yellow-green lesions which gradually turn brown. Severely infected leaves often drop prematurely.Infected petioles, tendrils, and shoots often curl, develop a shepherd's crook, and eventually turn brown and die. Young berries are highly susceptible to infection and are often covered with white fruiting structures of the fungus. Infected older berries of white cultivars may turn dull gray-green, whereas those of black cultivars turn pinkish red.
Prevent This Plant Disease By follow below steps :
Apply dormant sprays to reduce inoculum levels. Cut it out. Open up that canopy. Don't let down your defenses. Scout early, scout often. Use protectant and systemic fungicides. Consider fungicide resistance. Watch the weather.
Supplements :
TEBULUR
Tebulur Tebuconazole 10% + Sulphur 65% WG , Advance Broad Spectrum Premix Fungicides
Buy Product

Future scope :

1. Enhanced Accuracy:
Continued research and advancements in machine learning algorithms can lead to further improvements in disease detection accuracy. By refining existing models and exploring novel techniques, the goal is to achieve higher precision and recall rates, minimizing false positives and false negatives in disease identification. This can be accomplished through the incorporation of more complex neural network architectures,advanced feature extraction methods, and ensemble learning techniques.

2. Transferability and Generalization:
Future projects can focus on developing models that are more transferable and can generalize well across different plant species and diseases. The ability to detect diseases in various crops, even with limited labeled data, would greatly enhance the practicality and adoption of the disease detection systems. Techniques like domain adaptation and transfer learning can be employed to leverage knowledge learned from one plant species to improve detection in others.

3. Multimodal Data Fusion:
Integrating multiple sources of data, such as images, sensor data, and spectral information, can enhance the accuracy and robustness of plant disease detection. By combining different modalities, including visual and non-visual data, the system can gain a more comprehensive understanding of plant health conditions and disease symptoms. Techniques like fusion models, deep learning architectures, and multimodal learning can be explored to leverage the complementary nature of different data sources.

4. Real-Time Monitoring and Decision Support:
The future scope of plant disease detection projects involves real-time monitoring and decision support systems. By leveraging edge computing and IoT devices, plant disease detection systems can continuously monitor plant health parameters, analyze data on-site, and provide immediate feedback and recommendations to farmers. This can enable proactive disease management strategies and timely interventions, minimizing crop losses and optimizing resource utilization.

Conclusion:

This paper presents the survey on different disease classification techniques used for plant leaf disease detection and an algorithm for image segmentation technique that can be used for automatic detection as well as classification of plant leaf diseases later. Bananas, beans, jackfruit, lemon, mango, potato, tomato, and sapota and many more of those species on which the proposed algorithm is tested. Therefore, related diseases for these plants were taken for identification. With very less computational efforts the optimum results were obtained, which also shows the efficiency of the proposed algorithm in recognition and classification of the leaf diseases. The proposed business model, which is based on a subscription-based service and several monetization mechanisms, exemplifies a balanced approach to sustainability and accessibility. The plant disease detection app intends to change disease management procedures for farmers and agricultural specialists by emphasizing early detection, accuracy, and user-friendly interfaces.