TABLE of Contents

Project Link:

https://drive.google.com/drive/folders/1FMI_em9X3K0VZkoz5V0H6J4ZQsgV9mMT?usp=sharing

Abstract

We have developed a website for farmers that can detect the plant name and identify the disease in the plant using an uploaded image of the plant's leaf. If the plant is healthy, the website will indicate that. For diseased plants, the website will suggest the pesticides that can be used for each disease. This project aims to provide farmers with an efficient and easy-to-use tool to identify plant diseases and take corrective measures quickly.

The proposed system uses image processing techniques to classify the uploaded image and suggest the plant name. And it will also show the disease of the plant, and will recommend the necessary pesticides to cure it.

This website is specifically designed for farmers who may not have easy access to plant disease experts or may not have the resources to diagnose plant diseases. By providing a simple and effective tool, farmers can take timely action to protect their crops, leading to higher crop yields and a more sustainable agricultural sector.

In addition, the website is designed to be user-friendly and accessible, requiring only an uploaded image of the plant's leaf to generate results. The system incorporates deep learning methods to accurately detect the plant and disease, providing reliable and consistent results. This eliminates the need for manual and subjective identification, which can be time-consuming and prone to errors.

Furthermore, the website's approach to suggesting pesticides for each disease takes into account the specific needs of the plant and the type of disease it is affected by. This ensures that the farmer is equipped with the most effective and appropriate pesticide for the given situation. By recommending the right pesticide at the right time, the website aims to minimize the use of harmful pesticides and promote environmentally sustainable agriculture practices.

Overall, the Plant Disease Diagnosis and Remediation System provides a valuable tool for farmers, enabling them to quickly and accurately identify plant diseases and take appropriate action. By leveraging the power of image processing and machine learning, the website offers an efficient and cost-effective solution to a common problem faced by farmers.

Introduction

Complete introduction about the field of study:

Crop diseases are a major problem for farmers and can have a significant impact on food production and global food security. Identifying the specific disease affecting a crop is the first step in controlling it, but this can be a challenge for farmers who may not have the training or expertise to identify diseases accurately. Farmers typically rely on manual inspection and experience to identify crop diseases, which can be time-consuming and often results in incorrect diagnoses. In many cases, farmers apply pesticides without knowing the type of disease affecting their crops, leading to the use of inappropriate chemicals, which can be ineffective and harmful to the living beings and environment. These can also result in further damage to the crops and increase the risk of resistance to the pesticide. Crop disease prediction using deep learning algorithms from images is a crucial aspect of modern agriculture. As the traditional method of detecting crop diseases is manual and time-consuming, and can lead to significant crop loss, the integration of deep learning algorithms with image analysis can help farmers make accurate predictions and take appropriate measures to prevent the spread of diseases.

Our project idea:

Here comes our project which aims to develop a system or a Website that can predict crop diseases from images taken of the crops and suggest the most appropriate pesticide for the farmers. The system will utilize deep learning algorithms such as Convolutional Neural Networks (CNNs), VGG, AlexNet, and other deep learning algorithms to identify the disease signatures from the images. The data for the model will be collected from various sources, including images of crops affected by various diseases, and the results of previous disease detection methods. Once the system has identified the crop disease, it will suggest the most appropriate pesticide for the farmers based on the type of disease, the stage of the disease, and other relevant factors. Our website will provide an easy-to-use and understandable interface for the farmers, allowing them to upload an image of their crops and receive a prediction of the disease and appropriate pesticide within minutes. We will also try to show the results in the local languages and use very minimal or no technical language in the conclusions, so that farmers can easily understand the functionality of the website and use it efficiently. The website has the potential to revolutionize the way farmers detect and prevent crop diseases. By using deep learning algorithms and image analysis, the system can provide fast and accurate predictions, helping farmers to save their crops and increase their yield. The project will also contribute to the development of the agriculture industry and support the sustainable food supply needed to feed the growing population. In addition, the agricultural department can also use the system to educate farmers about the importance of disease prevention and the use of appropriate pesticides. By providing access to this technology, the department can help farmers to make informed decisions about disease management and reduce the risk of crop loss.

Feasibility study of the project:

The feasibility of the project can be evaluated as high due to several factors. Firstly, the project is supported by a comprehensive dataset consisting of approximately 10,000 images, which provides a strong foundation for training deep learning algorithms. Additionally, information on the appropriate use of pesticides for specific diseases has been gathered from the web, further enhancing the accuracy of the predictions. Secondly, the key component of the project is the ability to accurately recognize plant diseases using powerful deep learning algorithms. This is a crucial aspect of the project and has the potential to revolutionize the way farmers detect and prevent crop diseases. Thirdly, the project is being developed using web programming languages such as ReactJS, which makes it possible to create a user-friendly website. The minimalist design of the website and the model makes the final product cost-effective and requires fewer resources to develop and maintain. In conclusion, the feasibility of the project is dependent on the availability of the right data, technology, and resources. With all these factors in place, the project has the potential to be a valuable tool for farmers and gardeners in their efforts to diagnose and treat plant diseases.

Objectives:

- To develop a user-friendly website that can accurately classify plant images and detect diseases in a timely manner.
- To provide farmers with an efficient and easy-to-use tool to identify plant diseases and take corrective measures quickly.
- To suggest farmers with the appropriate pesticides for the specific disease affecting their plant based on the analysis of the uploaded plant leaf image.
- To improve crop yield by allowing farmers to take timely actions to protect their crops.
- To reduce the need for costly and time-consuming manual inspections by plant disease experts.
- To expand the knowledge base of plant diseases and their remedies by collecting and analyzing data from user inputs.
- To increase accessibility to plant disease diagnosis tools and information, especially for farmers in remote or underserved areas.

Literature Survey:

PAPAER 1: Plant Disease Detection Using Deep Learning

Reference:
https://rspsciencehub.com/article_9834_e9c3d09ae4406e80dc2981886d728cd3.pdf

In this paper, the proposed solution aims to develop a deep learning-based method for detecting plant leaf diseases. The method uses a combination of convolutional neural network (CNN) and a deep neural network to accurately identify and recognize the symptoms of crop diseases. The study focuses on using image processing techniques like segmentation, feature extraction, and classification to detect plant diseases. The images of leaves from various plants are taken with a digital camera and the images are used to classify the affected parts. Here the authors aim to improve the accuracy of plant disease detection by using deep learning techniques and to address the limitations of the existing system where only humans are capable of predicting diseases and the process is slow and expensive. The proposed methodology has the potential to improve agricultural productivity by detecting diseases early and reducing the negative impact on small-scale farmers.

PAPER 2: Identification of Plant Disease using Image Processing Technique

Reference:https://e-tarjome.com/storage/panel/fileuploads/2019-08
22/1566463161_E12705-e-tarjome.pdf

In this paper they discussed about developing a software solution for detecting and classifying plant diseases in agriculture. Here the authors explain that agriculture is a crucial industry for many countries and that a large portion of the population depends on it for sustenance. However, diseases can cause significant losses in crop yield and quality, making disease detection a key factor in preventing these losses. The authors aim to develop a system that can detect diseases in plants automatically, using image processing techniques.

The system starts with capturing high-resolution images of healthy and unhealthy plants. The images are then pre-processed for improvement and segmented using k-means clustering. Features are extracted and the Random Forest Classifier is used for training and classification. Finally, the system is able to recognize diseases in the plants. The authors also provided a brief overview of the different types of plant diseases.

PAPER 3: Plant Disease Detection using Deep Transfer Learning

Reference:
https://journalppw.com/index.php/jpsp/article/download/5753/3778/6597#:~:text=%5B6%5D%20studied%20the%20transfer%20learning,even%20under%20complex%20background%20conditions.

In this paper they categorized the majority of plant diseases into three categories: bacterial disease, viral disease, and fungal disease. The manual process of identifying plant diseases is time-consuming and inefficient, so here the authors suggest using deep learning to reduce the effort required. In this paper they introduced Deep Learning and Deep Transfer Learning and explained how they can be used to classify data. The authors also provided a review of related work in the field of detecting plant diseases using deep learning methods. The authors proposed using the VGGNet-19 model, which is pre-trained using the ImageNet dataset, for detecting plant diseases. They freeze the top layers and use transfer learning by adding a few layers to the model to improve its performance and accuracy. The results showed an accuracy of 97.52% for apple leaves and 95.75% for grape leaves after 20 epochs. In conclusion, the authors highlighted the potential of using deep learning methods, especially transfer learning, for detecting plant diseases and reducing the effort required.

PAPER 4: Comparison of Artificial Intelligence Algorithms in Plant Disease Prediction

Reference:https://www.researchgate.net/profile/Sumit-Kumar-4/publication/360564285_Comparison_of_Artificial_Intelligence_Algorithms_in_Plant_Disease_Prediction/links/6281df143a23744a7283f587/Comparison-of-Artificial-Intelligence-Algorithms-in-Plant-Disease-Prediction.pdf

This research paper discusses the use of artificial intelligence algorithms in predicting plant diseases. It highlights the potential benefits of using machine learning and deep learning techniques in the agriculture industry. The paper compares five different machine learning algorithms, including Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Support Vector Machine (SVM), and K-Nearest Neighbor (KNN), for disease forecasting based on weather conditions such as temperature, soil moisture, and humidity. The results show that ANN had the highest accuracy of 90.79%. The paper also mentions the various factors that need to be considered when developing a disease forecasting model and mentions some recent online disease forecasting models used in different countries.

PAPER 5: Improving prediction of plant disease using k-efficient clustering and classification algorithms

Reference:https://www.researchgate.net/profile/Refed-Adnan-Jaleel/publication/361903013_Improving_prediction_of_plant_disease_using_k-efficient_clustering_and_classification_algorithms/links/62cbe342d7bd92231faa3808/Improving-prediction-of-plant-disease-using-k-efficient-clustering-and-classification-algorithms.pdf

In this research paper, the authors have applied the k-NN algorithm with clustering techniques to diagnose plant diseases in a soybean dataset. The data consisted of 18 different types of soybean diseases and a healthy class. The authors compared seven experiments, including k-means, k-medoids, k-efficient clustering, k-NN, k-means with k-NN, k-NN with k-medoids, and k-NN with k-efficient clustering. The results showed that k-NN with k-efficient clustering had the best performance with 100% accuracy, precision, and recall, while k-NN without clustering had the least performance. The authors concluded that the proposed model, which combines k-NN with k-efficient clustering, could be used by farmers to diagnose soybean and other plant diseases, decreasing the monitoring effort and improving agricultural progress.

PAPER 6: Plant Disease Detection using Machine Learning

Reference: https://www.irjet.net/archives/V7/i7/IRJET-V7I7511.pdf

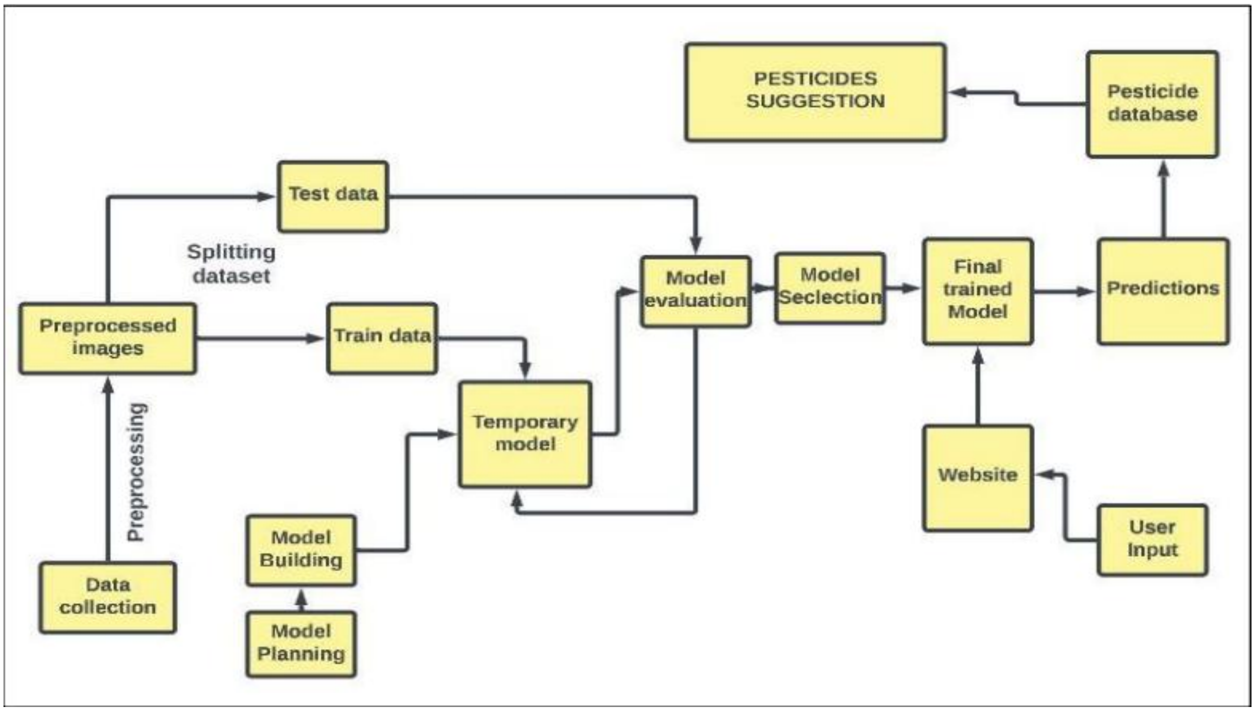
In this research paper we will be discussing about the plant disease detection and diagnosis in the field of agriculture using various machine learning algorithms, the various models used are random forests, support vector machine, fuzzy logic, K-means method has helped to improve the performance of these systems. In these cases, Random forests are considered to be flexible because they can handle a wide range of data types, including both continuous and categorical features. They also handle large datasets well and can be used for multi-class classification problems. Additionally, random forests are less prone to over fitting compared to other machine learning algorithms, which makes them a good choice for datasets with a limited number of images. In the proposed algorithm, the preprocessed images are passed through feature extraction to extract the desired features. The extracted features are then used to train the random forest classifier. During the classification stage, the classifier makes predictions on new images and assigns them to either the healthy or diseased class based on the learned patterns. Then the classifier then predicts whether the test image is of a healthy leaf or a diseased leaf.

Dataset Description:

The dataset contains images of plant leaves from three different plant categories: potato, tomato, and pepper. Each category has multiple classes, including healthy and diseased states. For potato, the classes are potato healthy, potato late blight, and potato early blight. For tomato, the classes include tomato target spot, tomato mosaic virus, tomato yellow leaf curl virus, tomato bacterial spot, tomato early blight, tomato healthy, tomato late blight, tomato leaf mold, tomato septoria leaf spot, and tomato spider mites two-spotted spider mite. For pepper, the classes are pepper bell bacterial spot and pepper bell healthy. The dataset aims to assist in the detection and classification of plant diseases to help farmers take necessary measures to protect their crops. In total, there are around 20,000 images in the dataset, with a varying number of images per class.

Dataset Link: https://www.kaggle.com/datasets/emmarex/plantdisease

Architecture Diagram:



Web Interface: - The interface will be built using ReactJS where the user can upload the leaf images which will be sent as a POST request to the FastAPI server. The FastAPI server retrieves the uploaded image and its metadata, such as filename and content type.

Disease Mapping: - Now the pre-processed image will be fed into the trained and selected deep learning model which analyzes the image to determine the type of disease based on the extracted features.

Showing results: - The model's predictions will be returned as a JSON object, which is sent back to the users front end interface through FastAPI server.

Pesticide Suggestion: - Once the trained deep learning model has identified the type of disease affecting the crop in the uploaded image, the system can access a database of known pesticides and their effectiveness against that disease. The system can then suggest appropriate pesticides that have been proven to be effective against that disease.

In addition to suggesting appropriate pesticides, the system can also provide information about the appropriate dosage, application frequency, and any precautions that need to be taken when using the recommended pesticide.

Deep learning model architecture

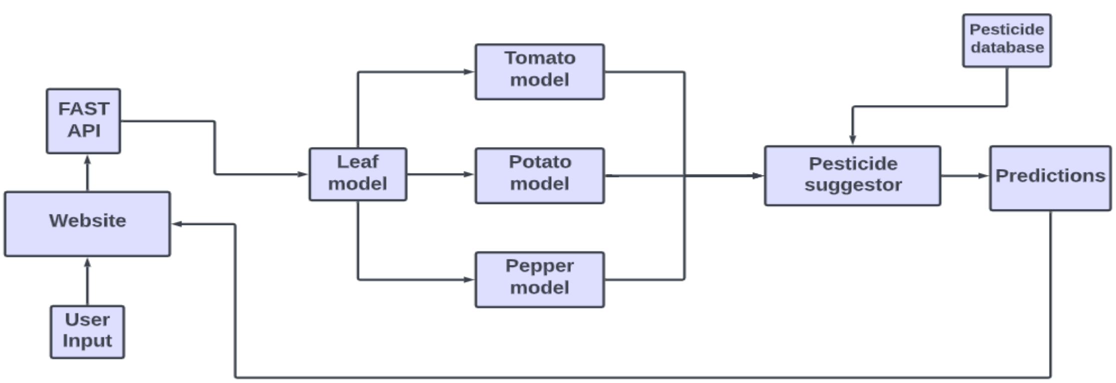
Dataset collection: Collecting datasets from Kaggle that have labeled images of diseased crops from various regions and weather conditions.

Preprocessing images: Before the images can be used to train a deep learning model, they need to be preprocessed to standardize their size and color and to remove any background noise or other artifacts. Preprocessing can include tasks such as resizing, cropping, normalization, and data augmentation.

Model planning and building: This step includes planning and building a customized model using several architectures like CNN, VGG, Inception, etc by adding a suitable number of convolutions, max-pooling layers, and activation functions.

Training model: Now these preprocessed images will be fed into the model, and optimized using regularization, and hyper parameter tuning.

Model evaluation and selection: The models built will be evaluated using several performance metrics such as precision, recall, and F1 score, based on which the final deployment model will be selected.



Proposed methodology:

The proposed methodology for our plant disease diagnosis and remediation system involves several key steps. First, we use image processing techniques to classify the uploaded image and identify the plant name. This is accomplished by training a deep learning model on a dataset of images of different plants, including potato, tomato, and pepper, each with several classes representing different disease states and healthy plants.

Next, we used the same approach to identify any diseases present in the uploaded image of the plant. This is done by training and testing separate models for each plant and disease class, using CNN and pre-trained VGG16 (using transfer learning). And ultimately selected CNN architecture based on several performance metrics. For example, if the uploaded image is of a tomato plant, the system will use the tomato disease model to determine if the plant has any of the common tomato diseases such as target spot or early blight.

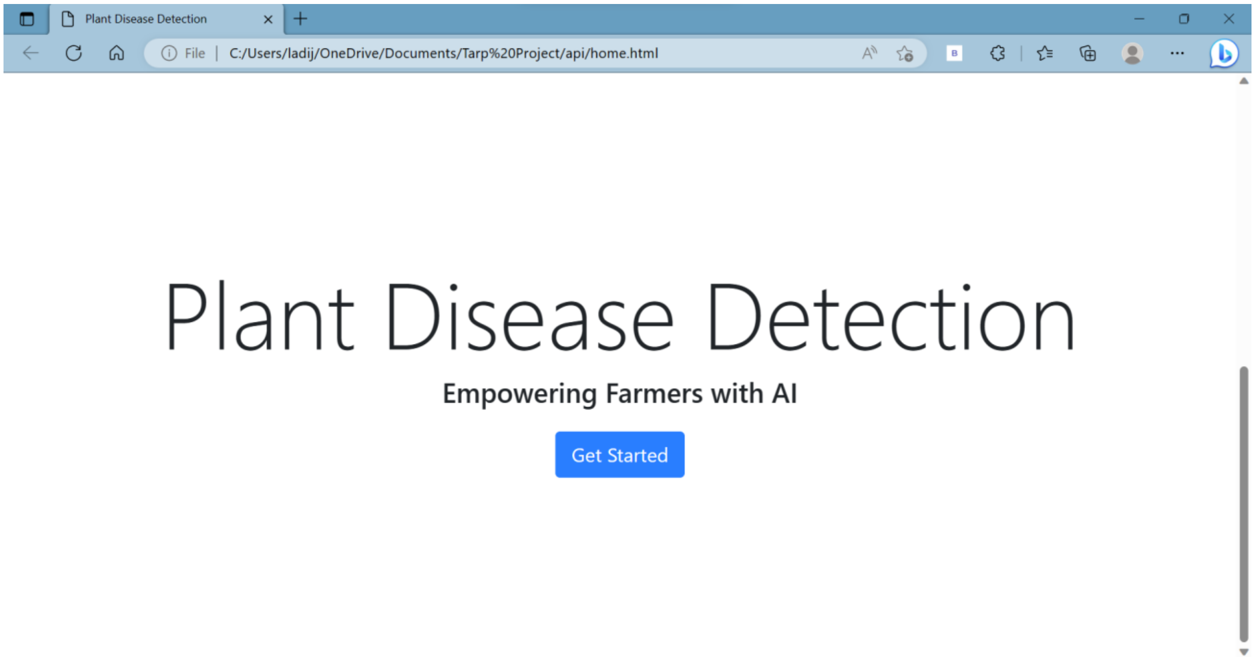
If the plant is found to be diseased, the system will recommend specific pesticides that are effective against the identified disease. These recommendations are from the dataset, which we created on own and are based on extensive research and manual note-taking on the most effective pesticides for each disease, and are presented to the user in an easy-to-understand format.

Finally, the system is implemented using the fast API framework to create a user-friendly web application. This allows farmers to easily upload images of their plants and receive quick and accurate diagnoses of any diseases present, as well as recommendations for appropriate remediation measures.

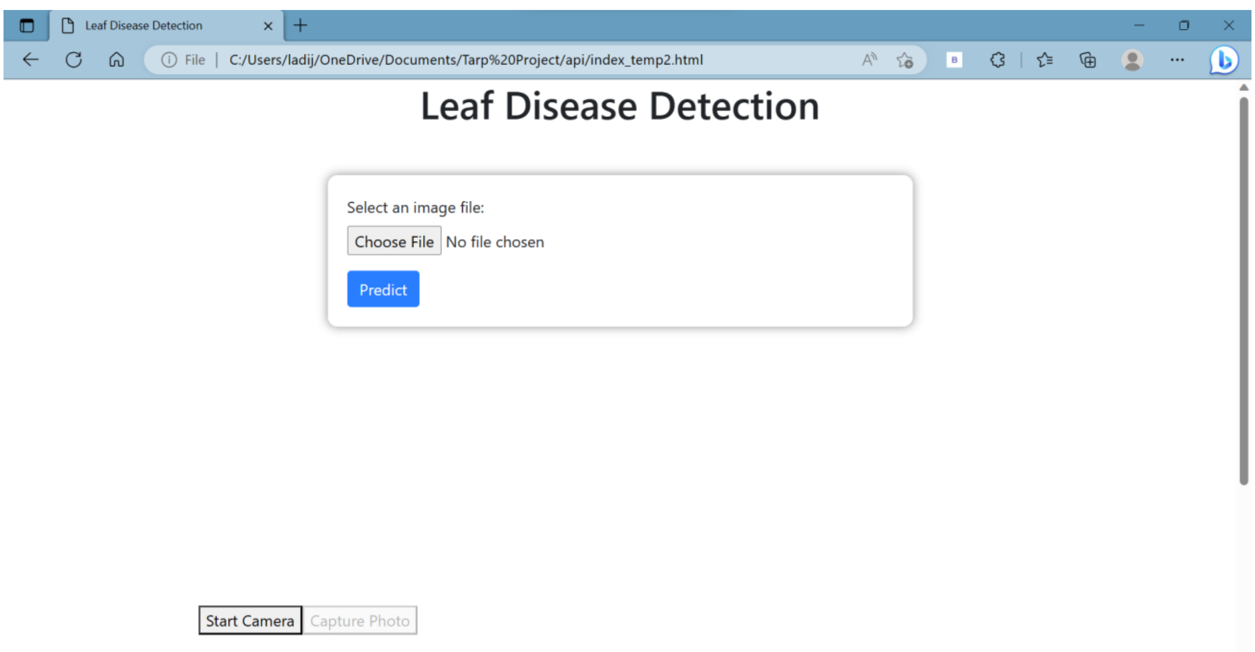
Overall, our proposed methodology leverages state-of-the-art deep learning techniques and extensive research on plant diseases and their treatments to create a powerful tool for farmers to diagnose and treat plant diseases quickly and effectively, leading to higher crop yields and a more sustainable agricultural sector

Results:

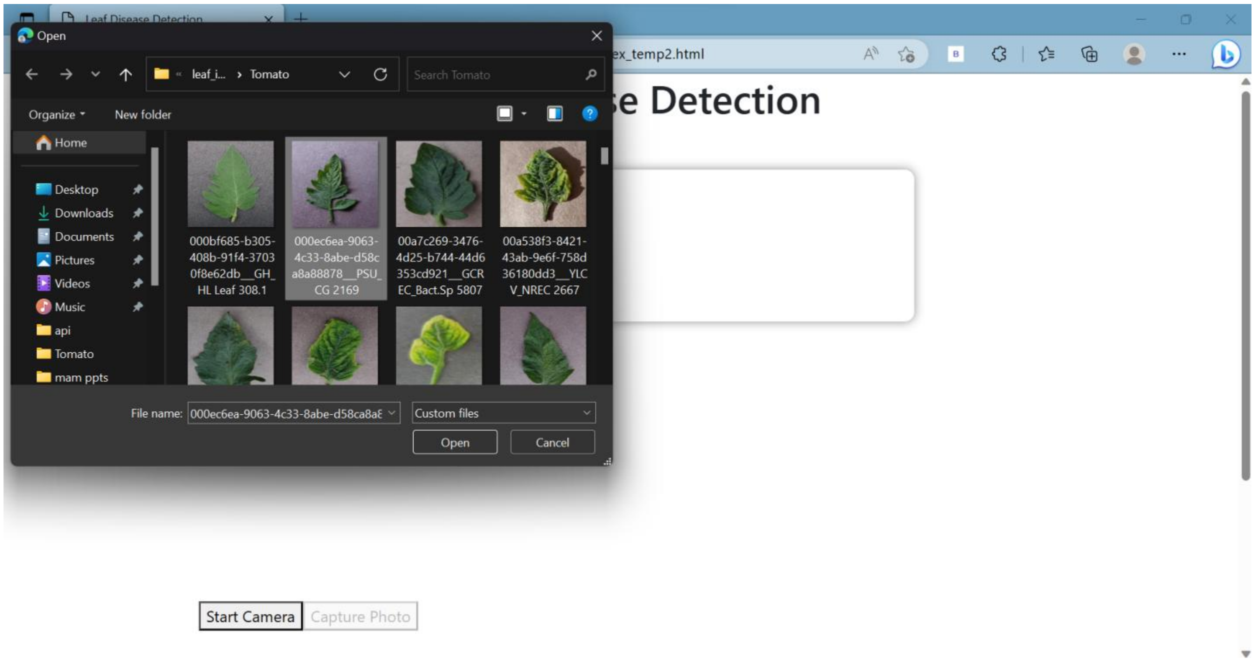
Home page of the website:



After clicking on the get started:

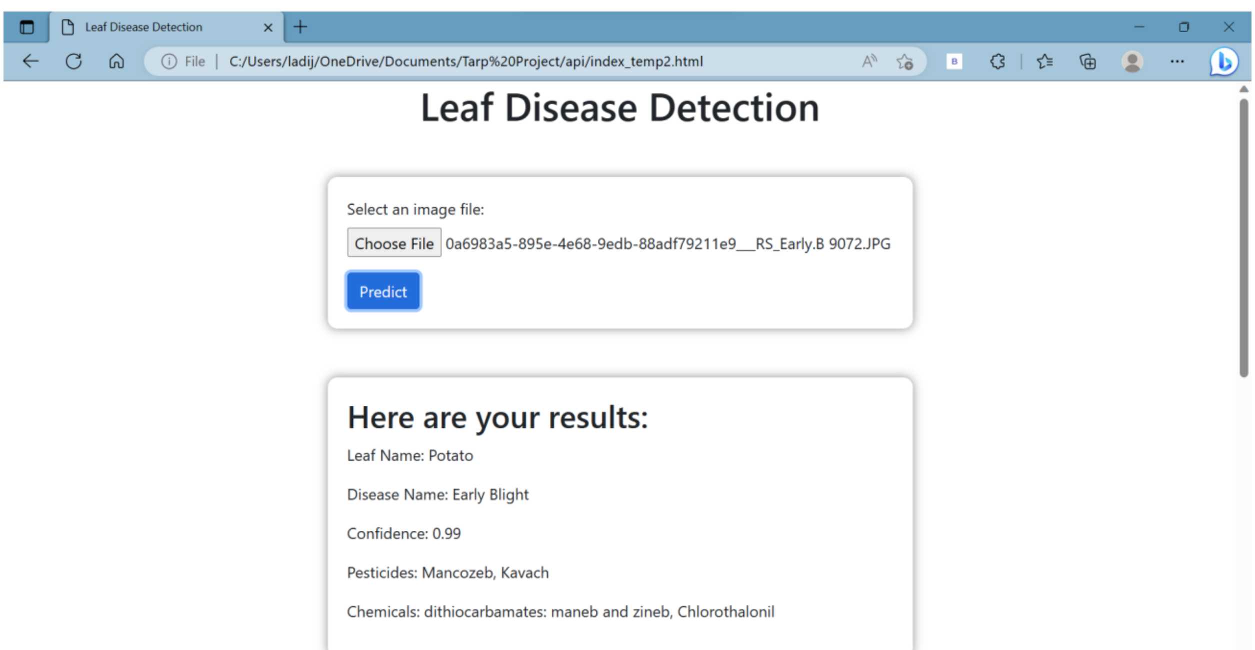


If the farmer clicks on choose file:

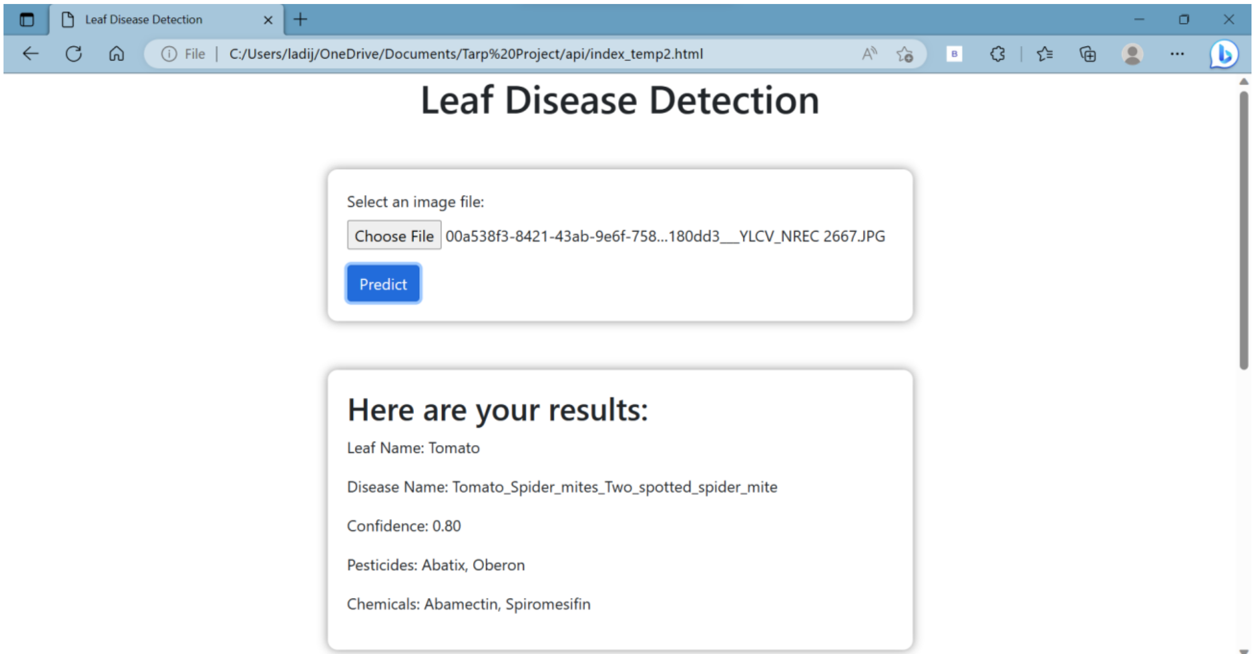


Here he can choose whatever leaf image, for which he wants to know the plant name, leaf disease and pesticides and chemicals.

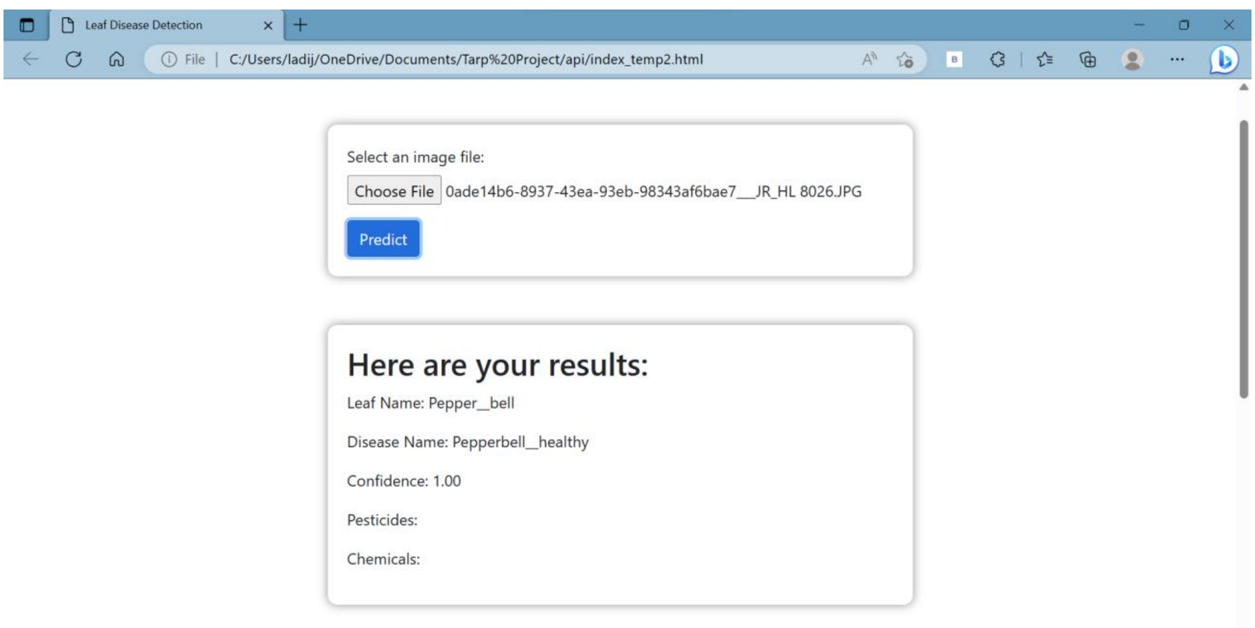
Our website is predicting the given image as Potato, and disease as Early Blight and recommending the pesticides and chemicals that can be used



Our website is predicting and recommending pesticides for tomato plant leaf:



Our website is predicting and recommending pesticides for Pepper plant leaf:



Conclusion

In conclusion, the Plant Disease Diagnosis and Remediation System is an effective tool for farmers to quickly and accurately diagnose plant diseases and identify the necessary pesticides for treatment. By using image processing techniques and machine learning algorithms, the system can classify the uploaded image and suggest the plant name and disease. The website interface provides an easy-to-use platform for farmers who may not have access to plant disease experts or may not have the resources to diagnose plant diseases.

Moreover, the system can lead to higher crop yields and a more sustainable agricultural sector by enabling farmers to take timely corrective measures. The accuracy of the model can be improved by collecting more data and including additional plant species and diseases. Future work can also involve automating the process of suggesting pesticides by integrating with a database of pesticide information. Overall, this project has the potential to make a significant impact in the agricultural sector by providing farmers with an efficient and effective tool to combat plant diseases and protect their crops.

Future Work:

There are several possibilities for enhancing the capabilities and applications of this website. One potential direction is to expand the dataset to include more plant species and more types of diseases. This would increase the scope of the website and enable it to provide even more comprehensive support for farmers.

Another possibility is to incorporate machine learning techniques to improve the accuracy of disease identification and pesticide suggestion. This could involve developing more sophisticated algorithms for image processing and feature extraction, as well as integrating more advanced models. With the increasing availability of large-scale datasets and advanced machine learning tools, this approach has the potential to significantly enhance the effectiveness of the website.

Furthermore, it could be beneficial to collaborate with agricultural experts and researchers to develop more targeted and effective solutions for combating plant diseases. This could involve conducting field experiments to test the effectiveness of different pesticides and developing customized treatment plans based on the specific conditions of each farm and crop. By leveraging the expertise of a broad range of stakeholders, this website could become an even more valuable resource for farmers seeking to optimize their crop yields and minimize the impact of plant diseases on their livelihoods.

Codes:

Deep learning Codes

Leaf Prediction:

https://drive.google.com/file/d/1p0dF35z86AcvkKJgZfU1JSHLKWKStZLJ/view?usp=share_link

Tomato Disease prediction:

https://drive.google.com/file/d/1E1ineCWPQOSmNhX5jBNkJX_02TSIHlNI/view?usp=share_link

Potato Disease prediction:

https://drive.google.com/file/d/1V3v6NfX0P-6s0rQWkVh-bRwrX6yfFYIq/view?usp=share_link

Pepper Disease prediction:

https://drive.google.com/file/d/1Jvy-_EOgMHxGNaNxAnyHWRZAgWkYoJBZ/view?usp=share_link

Fast API Code:

```
from fastapi import FastAPI, File, UploadFile

from fastapi.responses import JSONResponse
import uvicorn
import numpy as np
from io import BytesIO
from PIL import Image
import tensorflow as tf
from fastapi.middleware.cors import CORSMiddleware

from pest_display import pest_dict

app = FastAPI()
```

```
app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)


Leaf_MODEL =
tf.keras.models.load_model("C:/Users/ladij/OneDrive/Documents/Tarp
Project/Training2/models_leaf_3/1")
Potato_MODEL =
tf.keras.models.load_model("C:/Users/ladij/OneDrive/Documents/Tarp
Project/Training2/models_potato/1")
Tomato_MODEL =
tf.keras.models.load_model("C:/Users/ladij/OneDrive/Documents/Tarp
Project/Training2/models_tomato/1")
Pepper_MODEL =
tf.keras.models.load_model("C:/Users/ladij/OneDrive/Documents/Tarp
Project/Training2/models_pepper/1")


leaf_CN = ["Pepper__bell","Potato","Tomato"]
potato_CN = ["Early Blight", "Late Blight", "Healthy"]
tomato_CN =
["Tomato_Bacterial_spot","Tomato_Early_blight","Tomato_Late_blight","Tomato_L
eaf_Mold",

"Tomato_Septoria_leaf_spot","Tomato_Spider_mites_Two_spotted_spider_mite","To
mato__Target_Spot",

"Tomato_Tomato_YellowLeafCurl_Virus","Tomato_Tomato_mosaic_virus","Tomato_hea
lthy"]

pepper_CN = ["Pepper_bell_Bacterial_spot", "Pepperbell__healthy"]

@app.get("/ping")
async def ping():
    return "Hello, I am alive"

def read_file_as_image(data) -> np.ndarray:
    image = np.array(Image.open(BytesIO(data)))
    return image
```

```python
@app.post("/predict")
async def predict(
    file: UploadFile = File(...)
):

    image = read_file_as_image(await file.read())
    img_batch = np.expand_dims(image, 0)

    predictions = Leaf_MODEL.predict(img_batch)

    predicted_class = leaf_CN[np.argmax(predictions[0])]
    confidence = np.max(predictions[0])


    if predicted_class=='Potato':
        pred_potato = Potato_MODEL.predict(img_batch)
        pred_class_potato = potato_CN[np.argmax(pred_potato[0])]
        confidence_potato = np.max(pred_potato[0])


        x = pred_class_potato,
        y =  predicted_class,
        z = confidence_potato
        pests = pest_dict.get(pred_class_potato, {}).get("Pesticides", [])
        chems = pest_dict.get(pred_class_potato, {}).get("Chemicals", [])


    elif predicted_class=='Pepper__bell':
        pred_pepper = Pepper_MODEL.predict(img_batch)
        pred_class_pepper = pepper_CN[np.argmax(pred_pepper[0])]
        confidence_pepper = np.max(pred_pepper[0])

        x = pred_class_pepper,
        y = predicted_class,
        z = confidence_pepper
        pests = pest_dict.get(pred_class_pepper, {}).get("Pesticides", [])
        chems = pest_dict.get(pred_class_pepper, {}).get("Chemicals", [])


    elif predicted_class=='Tomato':
        pred_tomato = Tomato_MODEL.predict(img_batch)
        pred_class_tomato = tomato_CN[np.argmax(pred_tomato[0])]
        confidence_tomato = np.max(pred_tomato[0])
```

```
        x =  pred_class_tomato,
        y =  predicted_class,
        z = confidence_tomato
        pests = pest_dict.get(pred_class_tomato, {}).get("Pesticides", [])
        chems = pest_dict.get(pred_class_tomato ,{}).get("Chemicals", [])


    return JSONResponse({
        'leaf_name': y,
        'disease_name': x,
        'confidence': float(z),
        'pesticides': pests,
        'chemicals' : chems
    })


if _name_ == "_main_":
    uvicorn.run(app, host='localhost', port=8000)
```

Website HTML Code:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Leaf Disease Detection</title>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <style>
        body {
            background-color: #ffffff;
        }
        .container_2 {
            display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}
```

```
        #formContainer {
            margin-top: 50px;
            max-width: 600px;
            margin-left: auto;
            margin-right: auto;
            padding: 20px;
            background-color: #ffffff;
            border-radius: 10px;
            box-shadow: 0px 0px 10px #888888;
        }
        #resultContainer {
            margin-top: 50px;
            max-width: 600px;
            margin-left: auto;
            margin-right: auto;
            padding: 20px;
            background-color: #ffffff;
            border-radius: 10px;
            box-shadow: 0px 0px 10px #888888;
        }
        #resultContainer h2 {
            margin-top: 0px;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1 class="text-center mb-5">Leaf Disease Detection</h1>
        <div id="formContainer">
            <form enctype="multipart/form-data">
                <div class="form-group">
                    <label for="fileInput">Select an image file:</label>
                    <input type="file" class="form-control-file"
id="fileInput" accept=".jpg, .jpeg, .png">
                </div>
                <button type="button" class="btn btn-primary"
onclick="predict()">Predict</button>
            </form>
        </div>

        <div id="resultContainer" style="display:none">
            <h2>Here are your results:</h2>
            <div id="result">
```

```
                <!-- The result will be displayed here -->
            </div>
        </div>
    </div>


    <div class="container_2">
        <button id="startButton">Start Camera</button>
    <button id="captureButton" disabled>Capture Photo</button>
    <br>
    <video id="video" width="640" height="480" style="display:block"></video>
    <br>
    <canvas id="canvas" width="640" height="480"
style="display:none"></canvas>


    </div>

    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@0.2.7/dist/tf.min.js"></sc
ript>
    <script>
        function predict() {
            const fileInput = document.getElementById('fileInput');
            const file = fileInput.files[0];
            if (!file) {
                alert("Please select an image file");
                return;
            }

            const formData = new FormData();
            formData.append("file", file);

            fetch('http://localhost:8000/predict', {
                method: 'POST',
                body: formData
            }).then(response => response.json())
            .then(result => {
                const resultContainer =
document.getElementById('resultContainer');
                const resultDiv = document.getElementById('result');
                resultDiv.innerHTML = `
```

```
                <p>Leaf Name: ${result.leaf_name}</p>
                <p>Disease Name: ${result.disease_name}</p>
                <p>Confidence: ${result.confidence.toFixed(2)}</p>
                <p>Pesticides: ${result.pesticides.join(", ")}</p>
                <p>Chemicals: ${result.chemicals.join(", ")}</p>
            `;
            resultContainer.style.display = 'block';
        })
        .catch(error => {
            alert('Prediction failed. Please try again.');
            console.error(error);
        });
    }

    const startButton = document.getElementById('startButton');
    const captureButton = document.getElementById('captureButton');
    const video = document.getElementById('video');
    const canvas = document.getElementById('canvas');
    const context = canvas.getContext('2d');
    let stream;

    // Start the camera when the user clicks the button
    startButton.addEventListener('click', function() {
        navigator.mediaDevices.getUserMedia({ video: true })
            .then(function(mediaStream) {
                stream = mediaStream;
                video.srcObject = stream;
                video.play();
            })
            .catch(function(error) {
                console.log("Unable to access the camera: " + error);
            });

        // Enable the capture button once the camera has started
        video.addEventListener('canplay', function() {
            startButton.disabled = true;
            captureButton.disabled = false;
        });
    });

    // Capture a photo and display it on the website
    captureButton.addEventListener('click', function() {
        context.drawImage(video, 0, 0, canvas.width, canvas.height);
```

```
            stream.getTracks().forEach(track => track.stop());
            video.style.display = "none";
            captureButton.style.display = "none";
            canvas.style.display = "block";

            // Save the captured photo
            let link = document.createElement('a');
            link.download = 'photo.png';
            link.href =
canvas.toDataURL("image/png").replace(/^data:image\/[^;]/,
'data:application/octet-stream');
            link.click();
        });

    </script>
</body>
</html>
```

References:

https://rspsciencehub.com/article_9834_e9c3d09ae4406e80dc2981886d728cd3.pdf

https://journalppw.com/index.php/jpsp/article/download/5753/3778/6597#:~:text=%5B6%5D%20studied%20the%20transfer%20learning,even%20under%20complex%20background%20conditions.

https://www.researchgate.net/profile/Sumit-Kumar-4/publication/360564285_Comparison_of_Artificial_Intelligence_Algorithms_in_Plant_Disease_Prediction/links/6281df143a23744a7283f587/Comparison-of-Artificial-Intelligence-Algorithms-in-Plant-Disease-Prediction.pdf

https://www.researchgate.net/profile/Refed-Adnan-Jaleel/publication/361903013_Improving_prediction_of_plant_disease_using_k-efficient_clustering_and_classification_algorithms/links/62cbe342d7bd92231faa3808/Improving-prediction-of-plant-disease-using-k-efficient-clustering-and-classification-algorithms.pdf

https://www.frontiersin.org/articles/10.3389/fpls.2016.01419/full

https://www.sciencedirect.com/science/article/pii/S2666285X22000218

https://arxiv.org/pdf/2106.10698