

Lodha Genius
Programme

Project Exhibition

PLANT DISEASE DETECTION

Using a MobileNetV2 Neural Network

Rhea Manyala

WHAT IS THE ISSUE?

Farmers find it difficult to identify different crop diseases.



There are a large number of plant diseases that can be there

Farmers don't know what plant diseases are there and how to identify them

Even if farmers are aware it is difficult to identify the specific disease

THE SOLUTION

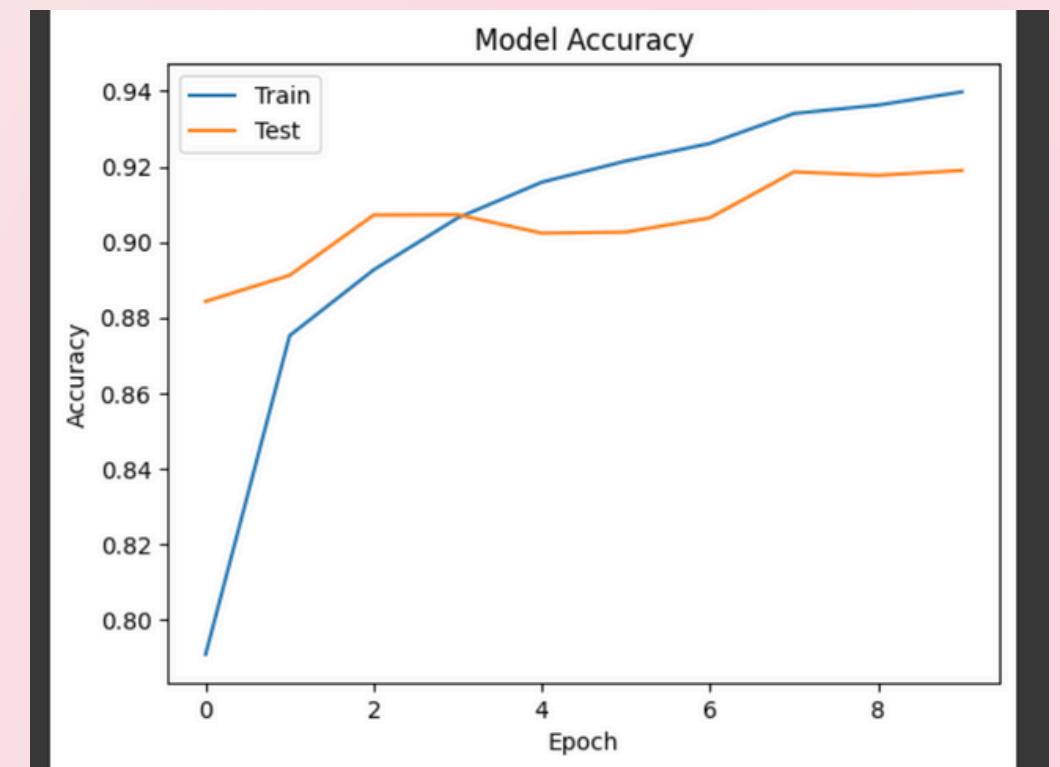
A Plant Disease Neural Network

- Use high-resolution leaf images to identify symptoms like spots, discoloration, or wilting.
- Employ CNN architectures like MobileNetV2 trained on labeled plant disease datasets to automatically learn visual patterns.
- Classify images into healthy or specific disease categories using softmax output layers.
- Implement a web app to upload images for prediction

MODEL TRAINING

Google Colab

```
▶ history = model.fit(  
    train_generator,  
    steps_per_epoch = train_generator.samples // batch_size,  
    epochs = 10,  
    validation_data = validation_generator,  
    validation_steps = validation_generator.samples // batch_size  
)  
  
→ Epoch 1/10  
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset`  
    self._warn_if_super_not_called()  
1358/1358 ━━━━━━ 142s 90ms/step - accuracy: 0.7020 - loss: 1.0598 - val_accuracy: 0.8843 - val_loss: 0.3565  
Epoch 2/10  
1358/1358 ━━━━━━ 124s 85ms/step - accuracy: 0.8717 - loss: 0.3919 - val_accuracy: 0.8912 - val_loss: 0.3182  
Epoch 3/10  
1358/1358 ━━━━━━ 113s 84ms/step - accuracy: 0.8933 - loss: 0.3161 - val_accuracy: 0.9072 - val_loss: 0.2783  
Epoch 4/10  
1358/1358 ━━━━━━ 142s 84ms/step - accuracy: 0.9083 - loss: 0.2691 - val_accuracy: 0.9073 - val_loss: 0.2735  
Epoch 5/10  
1358/1358 ━━━━━━ 113s 83ms/step - accuracy: 0.9194 - loss: 0.2393 - val_accuracy: 0.9024 - val_loss: 0.3036  
Epoch 6/10  
1358/1358 ━━━━━━ 144s 84ms/step - accuracy: 0.9248 - loss: 0.2183 - val_accuracy: 0.9027 - val_loss: 0.3011  
Epoch 7/10  
1358/1358 ━━━━━━ 140s 83ms/step - accuracy: 0.9288 - loss: 0.2048 - val_accuracy: 0.9064 - val_loss: 0.2992  
Epoch 8/10  
1358/1358 ━━━━━━ 142s 83ms/step - accuracy: 0.9337 - loss: 0.1832 - val_accuracy: 0.9186 - val_loss: 0.2602  
Epoch 9/10  
1358/1358 ━━━━━━ 141s 83ms/step - accuracy: 0.9381 - loss: 0.1743 - val_accuracy: 0.9177 - val_loss: 0.2684  
Epoch 10/10  
1358/1358 ━━━━━━ 113s 83ms/step - accuracy: 0.9410 - loss: 0.1676 - val_accuracy: 0.9190 - val_loss: 0.2726
```



WEB APP DEVELOPMENT

The image shows a dual-monitor setup for web application development. Both monitors display the Microsoft Visual Studio Code interface.

Left Monitor: This monitor displays the Python file `app.py`. The code implements a Flask application for plant disease prediction. It imports necessary libraries like Flask, tensorflow, and numpy. The `predict_disease` function loads an image from a path, processes it, and uses a pre-trained Keras model to predict the disease. The code also handles static files and uploads.

```
from flask import Flask, request, jsonify, render_template
import os
from werkzeug.utils import secure_filename
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing import image # type: ignore
import os

app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = 'uploads'
os.makedirs(app.config['UPLOAD_FOLDER'], exist_ok=True)

def predict_disease(image_path):
    class_indices = {0: 'Apple__Apple_scab', ...
    model = tf.keras.models.load_model("/Users/rheamanyala/vscode/plant website/maybe_better_plant_disease_prediction_model.h5")

    img = image.load_img(image_path, target_size=(224, 224))
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)
    img_array = img_array / 255.0

    prediction = np.argmax(model.predict(img_array))
    predicted_class = class_indices[prediction]
```

Right Monitor: This monitor displays the CSS file `index.html` from a folder named `templates`. The CSS defines styles for various elements, including `.profile-pic` (a circular background with a border-radius of 50%), `.username` (font-weight 700, font-size 1.1rem, color #b85c74, font-family 'Urbanist'), and `.post-image` (width 100%, height auto, max-height 400px, object-fit contain, background #fffeef1, display block, margin-bottom 10px). The file also includes comments for `.post-caption`, `.post-actions`, and `.post-actions span:hover`.

```
<html lang="en">
<head>
<style>
.profile-pic {
    width: 50px;
    border-radius: 50%;
    background: #ffdce0;
    flex-shrink: 0;
    margin-right: 12px;
}

.username {
    font-weight: 700;
    font-size: 1.1rem;
    color: #b85c74;
    font-family: 'Urbanist', sans-serif;
}

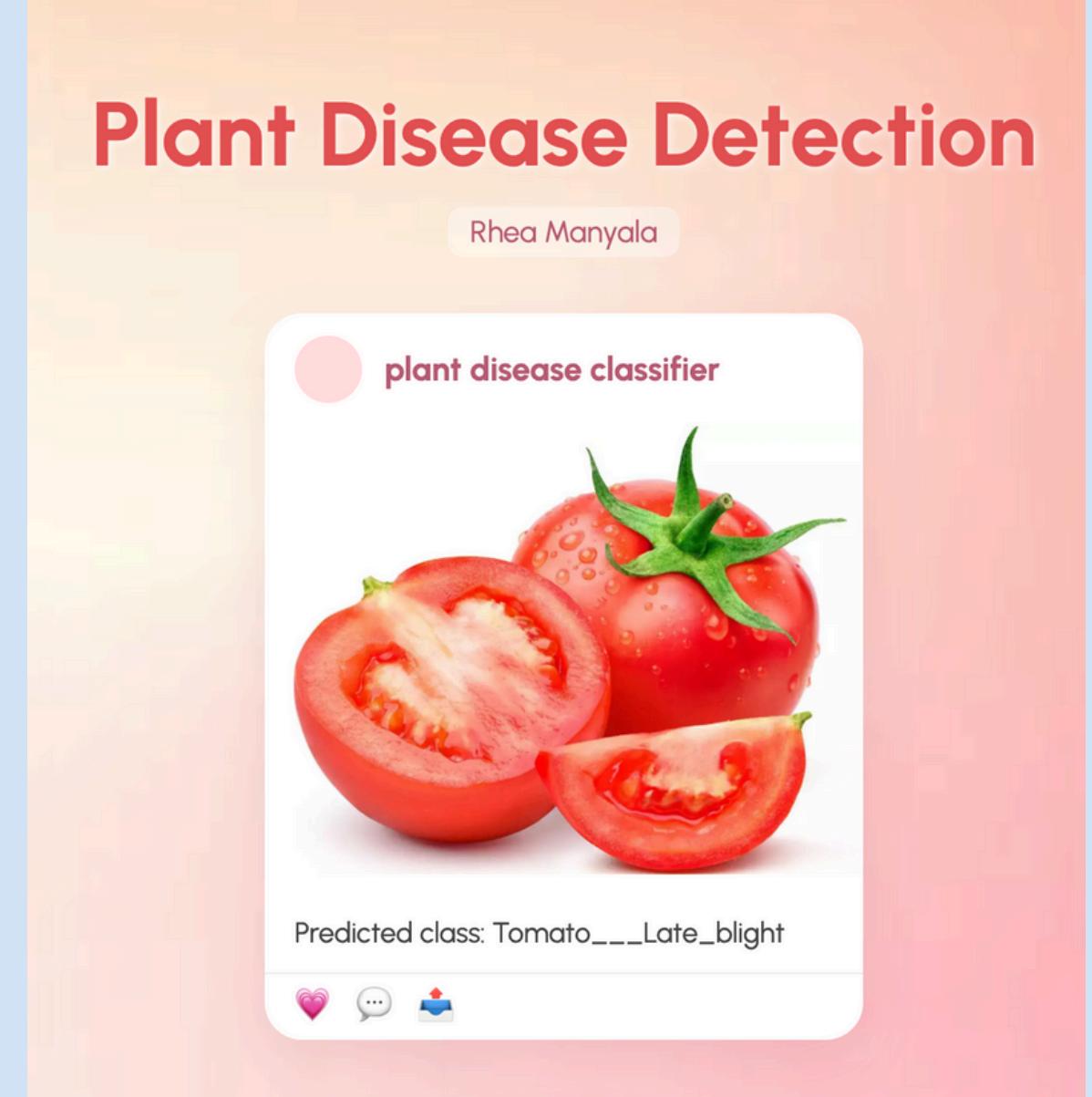
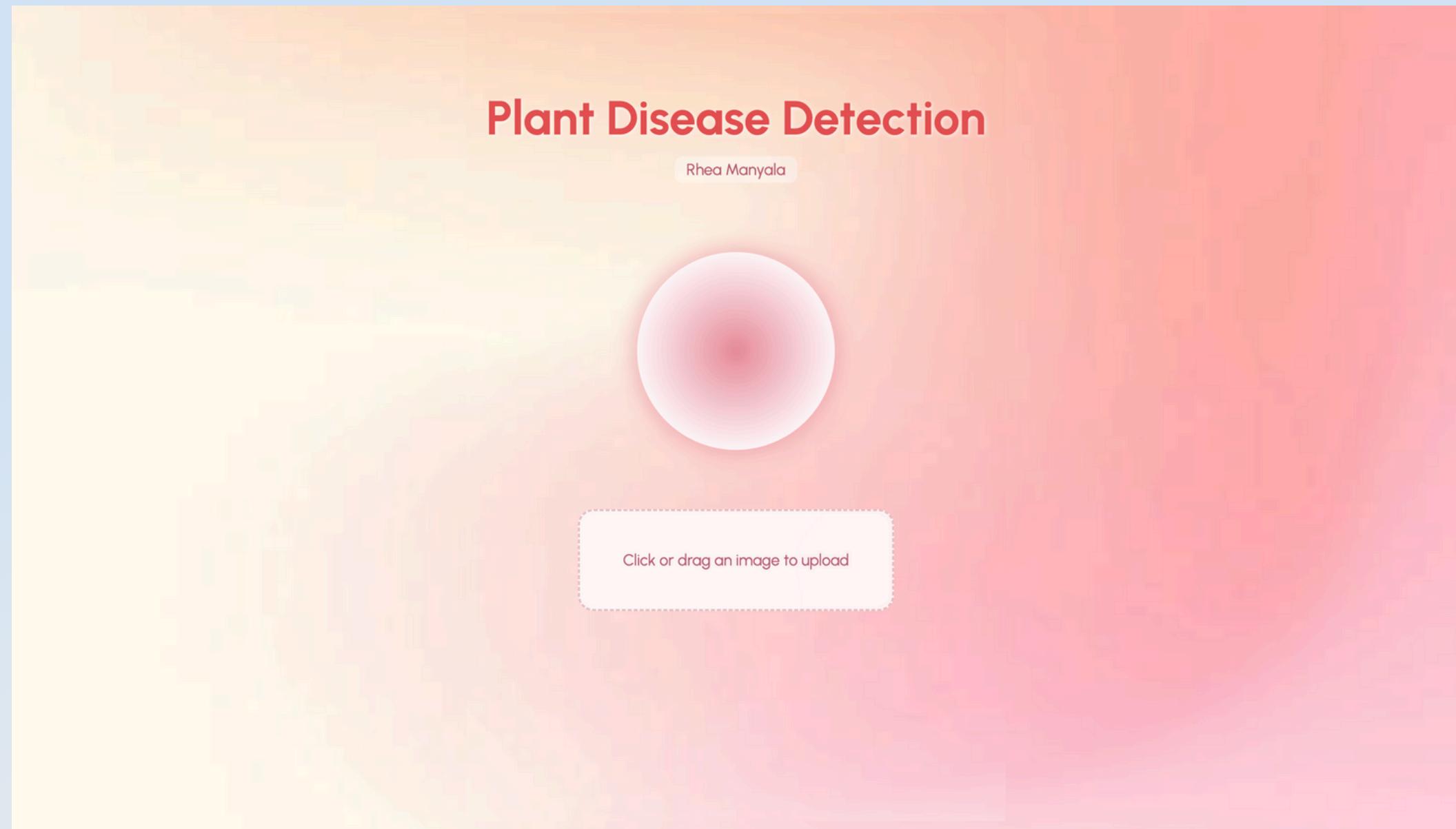
.post-image {
    width: 100%;
    height: auto;
    max-height: 400px;
    object-fit: contain;
    background: #fffeef1;
    display: block;
    margin-bottom: 10px;
}

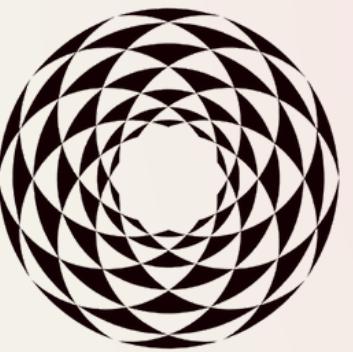
.post-caption {
    padding: 12px 16px;
    font-size: 0.95rem;
    color: #444;
    min-height: 1.4em;
}

.post-actions {
    padding: 4px 16px 6px;
    display: flex;
    gap: 14px;
    font-size: 1.2rem;
    color: #b85c74;
    border-top: 1px solid #eee;
}

.post-actions span:hover {
    color: #05277;
    cursor: pointer;
}
```

CLASSIFIER UI





THANK YOU

Rhea Manyala
LGP 2025 AI & Robotics