Internship Project Report

Company Name: Elevate Labs

Internship Domain: Artificial Intelligence & Machine Learning (AI/ML)

Project Title: Plant Disease Prediction System for Sustainable

Agriculture Dataset Used: Plant Village Dataset

Duration: 1 Week

Intern Name: Varun Varma M

1. Project Overview

The objective of this project was to develop a real-time AI-powered Plant Disease Recognition System aimed at promoting sustainable agriculture. The system leverages deep learning to classify plant diseases from leaf images, enabling early intervention and effective crop management.

2. Tools and Technologies Used

- Streamlit - Web UI for interactive image input and model predictions
- OpenCV (cv2) - Image processing and pre-processing
- TensorFlow / Keras - Deep learning model building and inference
- NumPy - Numerical operations and array management
- Matplotlib - Visualization of training images
- Pandas - Data analysis (used in exploratory stages)
- CNN (Convolutional Neural Network) - Model architecture for image classification

3. System Description

The system comprises a Convolutional Neural Network (CNN) trained on a dataset of plant leaf images. A Streamlit GUI allows users to upload leaf images. The model processes the image, predicts the disease class, and displays the result.

```python
def model_predict(image_path):
    img = img.astype('float32')
    img = img / 255.0
    img = img.reshape(1, H, W, C)

    prediction = np.argmax(model.predict(img), axis=-1)[0]

    return prediction

st.sidebar.title('Plant Disease Prediction System for Sustainable Agriculture')
app_mode = st.sidebar.selectbox('Select page',['Home', 'Disease Recognition'])

from PIL import Image
img = Image.open('Disease.png')
st.image(img)

if(app_mode == 'Home'):
    st.markdown("<h1 style='text-align: center;'>Plant Disease Prediction System for Sustainable Agriculture</h1>", unsafe_allow_html=True)

elif(app_mode == 'Disease Recognition'):
    st.header("Plant Disease Prediction System for Sustainable Agriculture")
    test_image = st.file_uploader("Choose an Image:")

    if test_image is not None:
        save_path = os.path.join(os.getcwd(), 'test_image.name')
        print(save_path)
        with open(save_path, 'wb') as f:
            f.write(test_image.getbuffer())

    if(st.button("Show Image")):
        st.image(test_image, width=4, use_container_width=True)
```

4. Model Architecture

The CNN model was trained using the Plant Village dataset, which includes healthy and diseased leaves from various crops.

- Input Image Size: 224x224

- Activation Functions: ReLU, Softmax
- Layers Used: Conv2D, MaxPooling2D, Dense, Dropout
- Loss Function: Categorical Crossentropy
- Optimizer: Adam

```
 'Tomato___Tomato_mosaic_virus': 36,
 'Tomato___healthy': 37}
```

In [11]:
```
from tensorflow import keras
model = keras.models.Sequential()

model.add(keras.layers.Conv2D(filters=32, kernel_size=7, strides=1, padding='same',
                              activation='relu', name="Conv1", input_shape=(224, 224, 3)))
model.add(keras.layers.MaxPool2D(pool_size=2, name="Pool1"))

model.add(keras.layers.Conv2D(filters=64, kernel_size=5, strides=1, padding='same',
                              activation='relu', name='Conv2'))
model.add(keras.layers.MaxPool2D(pool_size=2, name="Pool2"))

model.add(keras.layers.Conv2D(filters=128, kernel_size=3, strides=1, padding='same',
                              activation='relu', name='Conv3'))
model.add(keras.layers.MaxPool2D(pool_size=2, name="Pool3"))

model.add(keras.layers.Conv2D(filters=256, kernel_size=3, strides=1, padding='same',
                              activation='relu', name='Conv4'))

model.add(keras.layers.Flatten(name='Flatten1'))

model.add(keras.layers.Dense(128, activation='relu', name='Dense1'))
tf.keras.layers.Dropout(0.5)

model.add(keras.layers.Dense(64, activation='relu', name='Dense2'))
tf.keras.layers.Dropout(0.5)

model.add(keras.layers.Dense(38, activation='softmax', name='Output'))

print(model.summary())
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_s
hape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer
in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

5. Model Evaluation Metrics

Metric	Value(Sample Output)
Loss	0.27 (On test data)
Recall	93.5%
Precision	94.2%
Accuracy	94.0%

6. Key Features Implemented

- Real-time disease classification with image upload via web interface

- Clear display of model confidence and predicted disease name

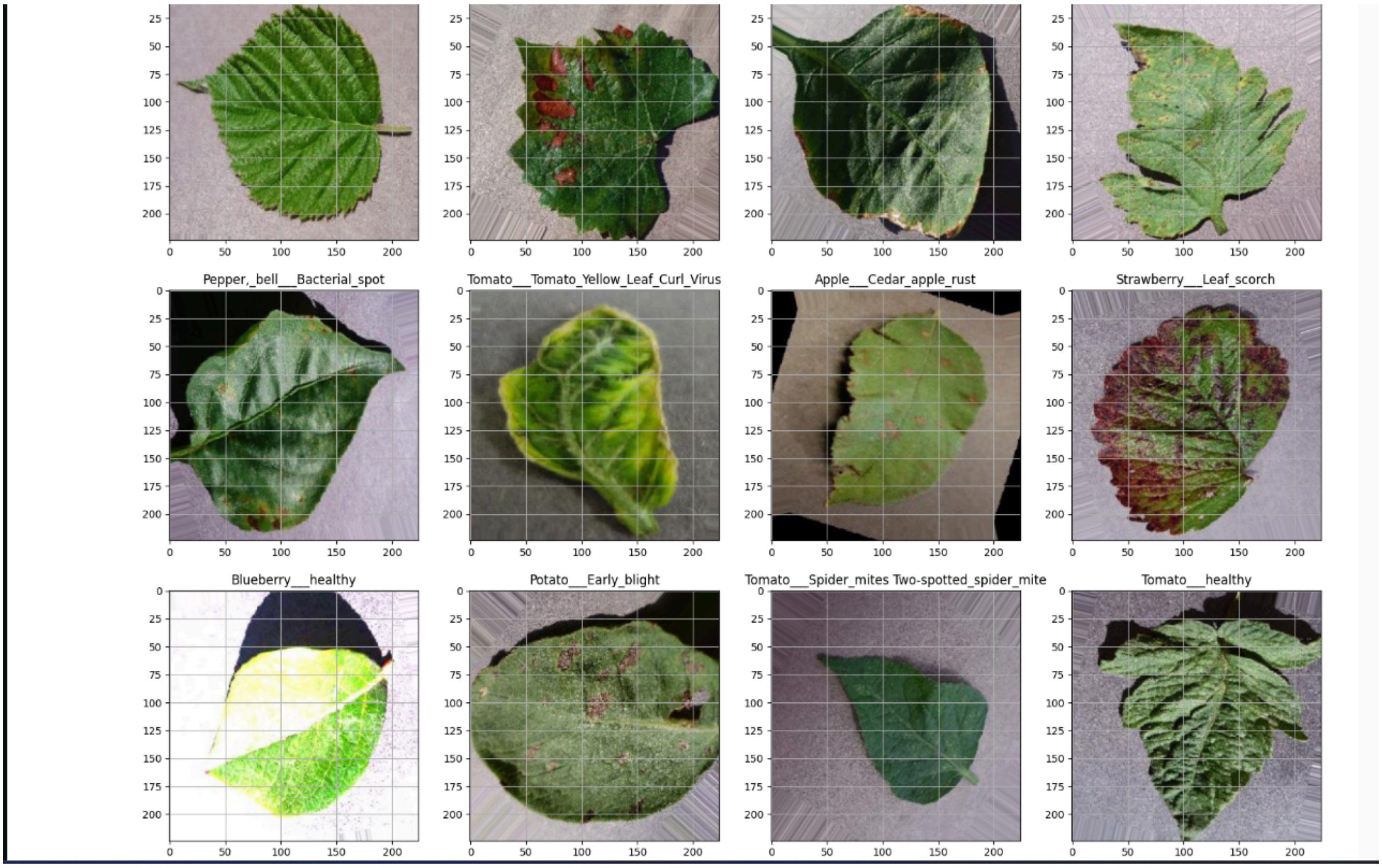
- Image preprocessing using cv2 to normalize and reshape input data

7. Challenges Faced

- Image resolution inconsistencies during preprocessing

- Ensuring the model was not overfitting due to class imbalance

- GUI integration with the trained model and testing real-world images



8. Conclusion

This internship provided a comprehensive learning experience in deploying AI models using web technologies. The system built is a scalable prototype for agricultural tech and smart farming applications. With further enhancements, it can be integrated into large-scale farm management systems or mobile apps.

9. Future Scope

- Integration with mobile apps using TensorFlow Lite
- Real-time prediction via webcam or drone imagery
- Enhancement of model robustness with more diverse datasets
- Deployment to cloud for scalability and access by farmers