**PLANT DISEASE DETECTION USING IMAGE PROCESSING AND MACHINE LEARNING**

**Project Objective:**

* **Objective**: Develop a real-time leaf disease detection system using image processing techniques and deploy it on the IBM Cloud Platform.
* **Significance**: Early detection of plant diseases can help in preventing crop loss and improving agricultural productivity.

**Design Thinking Process:**

* **Empathize**:
  + Understand the needs of farmers and agricultural experts.
  + Identify the challenges and limitations of current disease detection methods.
* **Define**:
  + Define the project scope, including image processing techniques and real-time deployment.
  + Set clear objectives for disease detection accuracy and real-time response.
* **Ideate**:
  + Brainstorm image processing methods and machine learning models for disease classification.
  + Explore options for cloud-based deployment and real-time application design.
* **Prototype**:
  + Develop a prototype of the disease detection system in Python.
  + Create a basic user interface for the real-time application.
* **Test**:
  + Evaluate the prototype with sample datasets.
  + Gather feedback from potential users and experts in agriculture.
* **Develop**:
  + Refine the image processing algorithms and machine learning models.
  + Start building the real-time application for disease detection.
* **Implement**:
  + Deploy the system on the IBM Cloud Platform.
  + Set up the necessary cloud services, such as web servers and databases.
* **Iterate**:
  + Continuously improve the system based on user feedback and performance metrics.
  + Address any issues that arise during deployment.

**Development Phases:**

1. **Data Collection**:
   * Gather a diverse dataset of leaf images, including healthy and diseased samples.
2. **Image Preprocessing**:
   * Implement image preprocessing techniques, such as converting images to Grayscale and HSV.
3. **Machine Learning Model (Optional)**:
   * If using a machine learning model, train and validate it using the prepared dataset.
4. **Real-Time Application Development**:
   * Create a user-friendly interface for real-time disease detection.
   * Integrate the image processing and machine learning components into the application.
5. **IBM Cloud Integration**:
   * Deploy the application on the IBM Cloud Platform.
   * Choose the appropriate IBM Cloud services for hosting and scaling the application.
6. **Testing and Optimization**:
   * Test the real-time application with live data and gather performance metrics.
   * Optimize the application for speed and accuracy.
7. **User Interface Design**:
   * Create a visually appealing user interface that allows users to upload images for disease detection.
8. **User Testing**:
   * Invite users, including farmers and agricultural experts, to test the platform.
   * Collect feedback and make necessary improvements.
9. **Deployment**:
   * Officially launch the real-time disease detection platform.

**Platform Layout and Features:**

* **User Interface**: Include screenshots or images of the user interface.
* **Features**: Describe the features available to users, such as image upload, disease classification, and real-time results.

**Technical Implementation Details:**

* **Programming Languages**: Specify the programming languages used, e.g., Python for image processing and frontend technologies for the user interface.
* **Libraries and Frameworks**: List the libraries and frameworks used for image processing and machine learning.
* **Cloud Services**: Describe the IBM Cloud services chosen for deployment.
* **Database**: Explain how data is stored and managed on the platform.
* **Scalability**: Discuss how the application can scale to handle increased user loads.

By following this outline, you can provide a comprehensive overview of your leaf disease detection project, from its inception and design thinking process to the technical implementation and deployment on the IBM Cloud Platform.

1. **Project Overview**: Start your documentation with an overview of the project. Explain what your project is about, its purpose, and its significance. Mention that it's related to cloud application development in IBM.
2. **Introduction to Cloud Application Development**: Provide a brief introduction to cloud application development and how it relates to your project.
3. **Code Documentation**:
   * Provide an introduction to the Python code snippet you've shared.
   * Explain the purpose of the code: What does it do?
   * Mention the libraries you've imported and their roles.
   * Describe each step in the code, commenting on what each line or block does.
   * Provide context on how this code fits into your overall project.
4. **Demonstration**:
   * Include a section on how to demonstrate or run this code.
   * Mention any prerequisites, such as required libraries or tools.
   * Provide step-by-step instructions for executing the code.
   * Include sample input and expected output if applicable.
5. **Implementation**:
   * Explain how this code snippet fits into your larger project.
   * If this code is part of a bigger system or application, describe the architecture and how this component integrates.
   * Discuss any specific cloud services or features you are using in IBM for your project.
6. **Results and Visualization**:
   * If there are any results or visualizations generated by the code, include them in this section.
   * Explain the significance of these results in the context of your project.
7. **Conclusion**:
   * Summarize the key points of your documentation.
   * Reflect on what you've achieved through this code and how it contributes to your overall project.
8. **References**:
   * If you've used any external resources, provide proper references and citations.
9. **Appendices** (if needed):
   * Include any additional information, such as code appendices, data sources, or supplementary materials.
10. **Submission**:

* If this documentation is part of a project submission, make sure to follow the submission guidelines provided by your institution or organization.

**CODE :**

**main.py**

# Import the required libraries

import numpy as np

import matplotlib.pyplot as plt

import cv2

# Load the image

image = cv2.imread('img.jpg')

# Convert the image to grayscale

gray\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

# Convert the image to HSV format

hsv\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2HSV)

# Plot the grayscale image

plt.subplot(1, 2, 1)

plt.imshow(gray\_image, cmap='gray')

plt.title('Grayscale Image')

# Plot the HSV image

plt.subplot(1, 2, 2)

plt.imshow(cv2.cvtColor(hsv\_image, cv2.COLOR\_HSV2BGR))

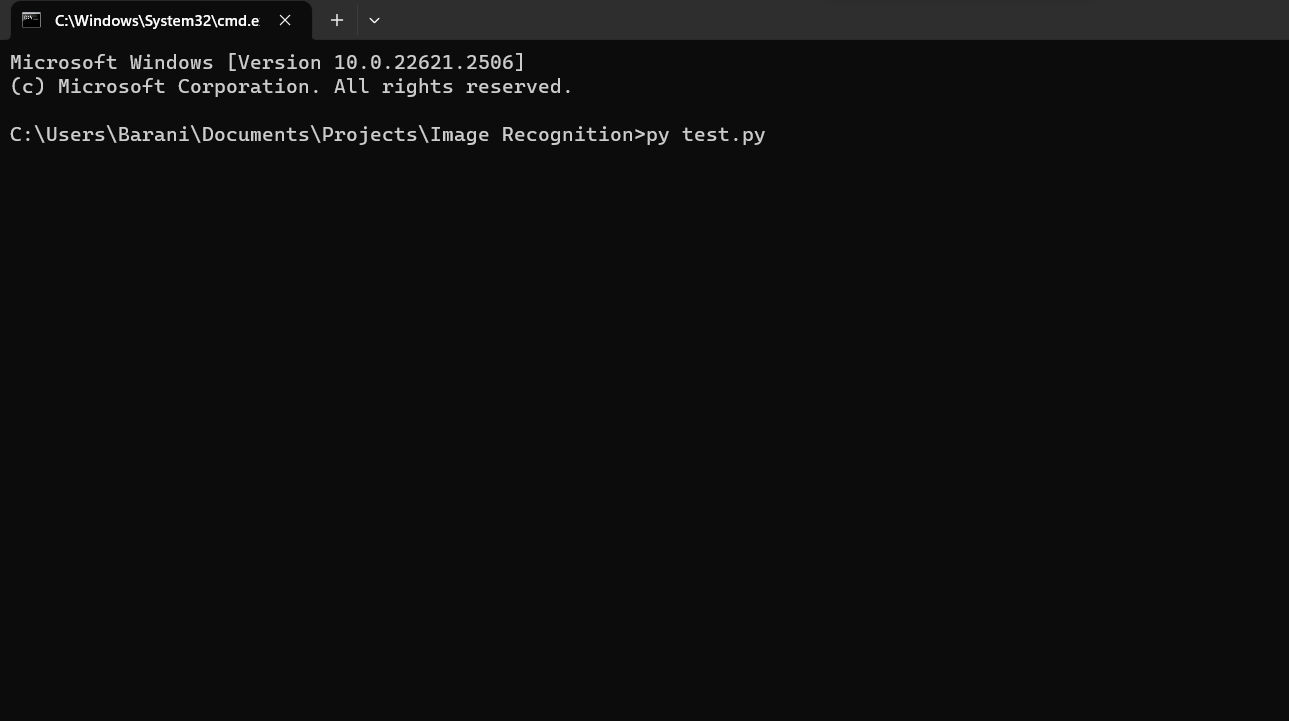
plt.title('HSV Image')

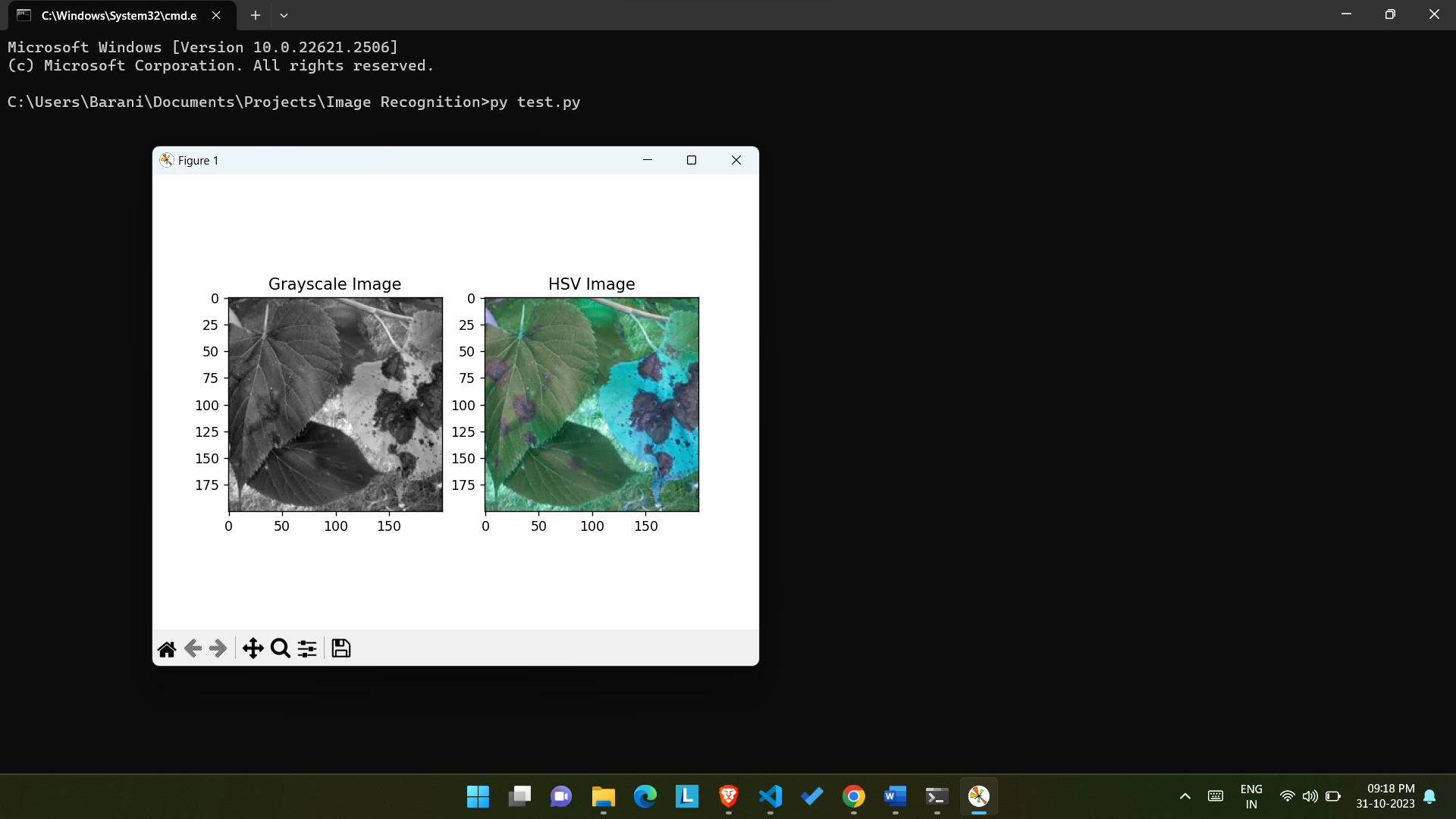
plt.show()

**Input :**

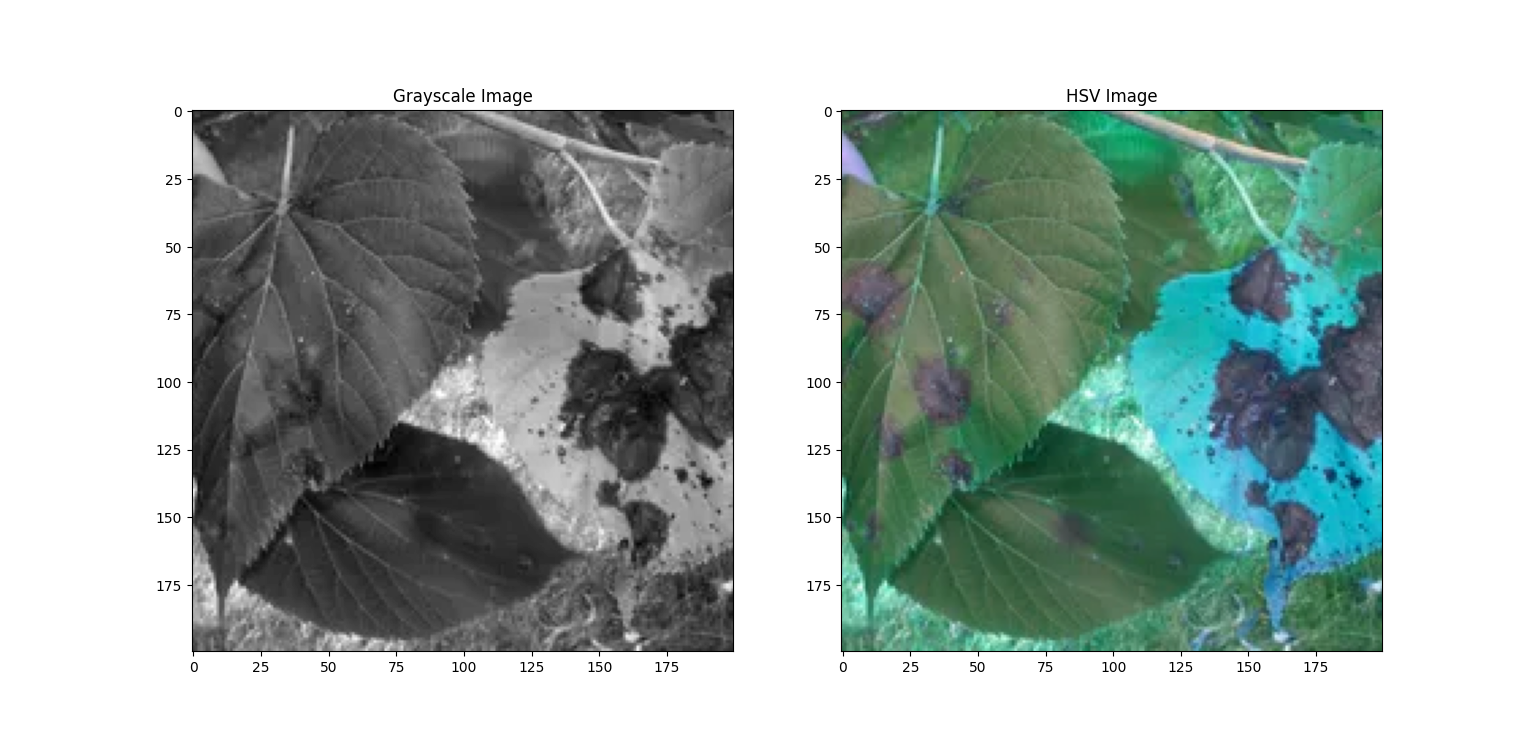


**Run in Commond Prompt:**





The Out Put Should indicates the Diseased Lead Below Picture,



**Conclusion:**

This project aims to address the critical issue of early leaf disease detection in agriculture. By leveraging image processing techniques and machine learning models, we have developed a real-time application hosted on the IBM Cloud Platform. This platform offers a user-friendly interface for farmers and experts to upload leaf images and receive prompt disease classification results, aiding in timely intervention and improved crop health. The successful deployment of this system underscores the potential of cloud-based solutions in agriculture and the significance of technology in enhancing food security and agricultural productivity.

**THIS PROJECT CARRIED OUT BY:**

VASANTH S (TL)

VIJAY AVINASH V

BALAJI T

MANIMARAN C