***Leaf Disease Using Fuzzy Inference System***

**1. Preprocessing:**

* The preprocessing step involves resizing the input image to a fixed size (in this case, 224x224 pixels) to ensure uniformity in the dataset.

**2. Feature Extraction:**

* After preprocessing, the image undergoes feature extraction. In this example, we extract two features: the mean and standard deviation of pixel intensities. These features are numerical representations of the image characteristics.

**3. Fuzzy Logic Classification:**

* Fuzzy logic allows for imprecise reasoning and decision-making by defining linguistic variables and fuzzy rules.
* The FuzzyLeafDiseaseClassifier class initializes fuzzy variables (lesion size, lesion density, and disease degree) along with their membership functions.
* Membership functions define how numerical inputs map to linguistic terms (e.g., small, medium, large).
* Fuzzy rules define the relationship between input variables and output classes (healthy or diseased).
* The degree of membership of the output variable (degree of disease) is calculated based on the input features (lesion size and density) using the defined fuzzy rules and membership functions.
* The degree of membership represents the confidence level of the classification result.

**4. Classification Result:**

* The degree of membership is compared to a threshold (0.5 in this case) to determine the final classification result.
* If the degree of membership is greater than 0.5, the leaf is classified as "Healthy"; otherwise, it's classified as "Diseased".

**Example:**

* We provide an example usage where an image of a leaf with a disease (potato late blight) is loaded and processed.
* The image is pre-processed and features (mean intensity and standard deviation) are extracted.
* The FuzzyLeafDiseaseClassifier class is used to classify the leaf based on the extracted features.
* The classification result ("Healthy" or "Diseased") is printed based on the degree of membership calculated by the fuzzy logic classifier.

This outlines the workflow of leaf disease classification using fuzzy logic, from preprocessing and feature extraction to fuzzy logic classification and obtaining the final classification result. Let me know if you need further clarification on any aspect!

ode

Code

import numpy as np

import skfuzzy as fuzz

from skfuzzy import control as ctrl

from PIL import Image

class Preprocessor:

def \_\_init\_\_(self, image\_size=(224, 224)):

self.image\_size = image\_size

def preprocess\_image(self, image\_path):

try:

image = Image.open(image\_path)

image = image.resize(self.image\_size)

return np.array(image)

except Exception as e:

print(f"Error processing image: {e}")

return None

class FeatureExtractor:

def \_\_init\_\_(self):

pass

def extract\_features(self, image):

# Example: Using mean and standard deviation of pixel intensities as features

mean\_intensity = np.mean(image)

std\_intensity = np.std(image)

return mean\_intensity, std\_intensity

class FuzzyLeafDiseaseClassifier:

def \_\_init\_\_(self):

# Define fuzzy variables and membership functions

self.lesion\_size = ctrl.Antecedent(np.arange(0, 256, 1), 'lesion\_size')

self.lesion\_density = ctrl.Antecedent(np.arange(0, 256, 1), 'lesion\_density')

self.disease = ctrl.Consequent(np.arange(0, 1.1, 0.1), 'degree\_of\_disease')

# Define membership functions for lesion size

self.lesion\_size['small'] = fuzz.trimf(self.lesion\_size.universe, [0, 100, 200])

self.lesion\_size['medium'] = fuzz.trimf(self.lesion\_size.universe, [0, 50, 100])

self.lesion\_size['large'] = fuzz.trimf(self.lesion\_size.universe, [50, 150, 255])

# Define membership functions for lesion density

self.lesion\_density['low'] = fuzz.trimf(self.lesion\_density.universe, [0, 50, 100])

self.lesion\_density['medium'] = fuzz.trimf(self.lesion\_density.universe, [50, 100, 150])

self.lesion\_density['high'] = fuzz.trimf(self.lesion\_density.universe, [100, 150, 255])

# Define membership functions for disease

self.disease['healthy'] = fuzz.trimf(self.disease.universe, [0, 0, 0.5])

self.disease['diseased'] = fuzz.trimf(self.disease.universe, [0.5, 1, 1])

# Define fuzzy rules

self.rule1 = ctrl.Rule(self.lesion\_size['small'] & self.lesion\_density['low'], self.disease['healthy'])

self.rule2 = ctrl.Rule(self.lesion\_size['large'] | self.lesion\_density['high'], self.disease['diseased'])

# Define fuzzy system

self.system = ctrl.ControlSystem([self.rule1, self.rule2])

self.simulator = ctrl.ControlSystemSimulation(self.system)

def classify(self, lesion\_size, lesion\_density):

try:

self.simulator.input['lesion\_size'] = lesion\_size

self.simulator.input['lesion\_density'] = lesion\_density

self.simulator.compute()

degree\_of\_membership = self.simulator.output['degree\_of\_disease']

print(f"Degree of Membership: {degree\_of\_membership}")

if degree\_of\_membership > 0.5:

return "Healthy"

else:

return "Diseased"

except Exception as e:

print(f"Error classifying leaf: {e}")

return None

if \_\_name\_\_ == "\_\_main\_\_":

# Load and preprocess a single image

image\_path = r"C:\Users\Satoshi\OneDrive\Desktop\Data\PERSONAL\_GROWTH\mini-projects\Images\Leaf\_image\_detection\potato class\potato\Potato\_\_\_Late\_blight\5d392db5-bf54-41f4-b76b-e3935dfe0154\_\_\_RS\_LB 3133.JPG"

preprocessor = Preprocessor()

feature\_extractor = FeatureExtractor()

classifier = FuzzyLeafDiseaseClassifier()

image = preprocessor.preprocess\_image(image\_path)

if image is not None:

features = feature\_extractor.extract\_features(image)

if features is not None:

result = classifier.classify(features[0], features[1])

if result is not None:

print(f"Prediction: {result}")e:

Output:

C:\Users\Satoshi\OneDrive\Desktop\Data\PERSONAL\_GROWTH\mini-projects\Images\Leaf\_image\_detection\potato class\potato\Potato\_\_\_Late\_blight\5d392db5-bf54-41f4-b76b-e3935dfe0154\_\_\_RS\_LB 3133.JPG

Degree of Membership: 0.5333069893535619

Prediction: Healthy