**--Lab-8--To perform various pattern classification and analysis techniques on images—**

import cv2

import numpy as np

from sklearn.svm import SVC

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

# --- 1. Feature Extraction (using SIFT) ---

def extract\_sift\_features(image\_path):

image = cv2.imread(image\_path, cv2.IMREAD\_GRAYSCALE)

if image is None:

print(f"Error: Could not load image {image\_path}")

return None, None

sift = cv2.SIFT\_create()

keypoints, descriptors = sift.detectAndCompute(image, None)

return descriptors

# --- 2. Prepare Dataset (Example with dummy data) ---

# In a real application, you would load images from different classes

# and extract features for each.

# For demonstration, let's create some dummy feature vectors and labels.

# Imagine these are descriptors from different images.

features\_class\_a = [np.random.rand(10, 128) for \_ in range(20)] # 20 images, 10 keypoints each, 128-dim descriptor

features\_class\_b = [np.random.rand(15, 128) for \_ in range(20)] # 20 images, 15 keypoints each, 128-dim descriptor

# Combine all descriptors into a single list and create corresponding labels

all\_descriptors = []

labels = []

for desc in features\_class\_a:

if desc is not None:

all\_descriptors.extend(desc)

labels.extend([0] \* len(desc)) # Class 0

for desc in features\_class\_b:

if desc is not None:

all\_descriptors.extend(desc)

labels.extend([1] \* len(desc)) # Class 1

# Convert to NumPy arrays

X = np.array(all\_descriptors)

y = np.array(labels)

# --- 3. Train a Classifier (Support Vector Machine) ---

# Split data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Initialize and train an SVM classifier

svm\_model = SVC(kernel='linear')

svm\_model.fit(X\_train, y\_train)

# --- 4. Evaluate the Classifier ---

y\_pred = svm\_model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"Classifier Accuracy: {accuracy:.2f}")