**PRACTICAL 2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name:** | Harsh Shah | **Semester:** | VII | **Division:** | 6 |
| **Roll No.:** | 21BCP359 | **Date:** | 30-07-24 | **Batch:** | G11 |
| **Aim:** | Extracting Region features and Boundary features from Images | | | | |

**Program**

import requests

from PIL import Image

import numpy as np

import cv2

from io import BytesIO

*# List of image URLs*

image\_urls = [

    "https://images.pexels.com/photos/56866/garden-rose-red-pink-56866.jpeg",

    "https://cdn.pixabay.com/photo/2015/10/09/00/55/lotus-978659\_640.jpg",

    "https://s28151.pcdn.co/wp-content/uploads/sites/2/2022/03/Coyote-animal-sentience-research.jpg",

    "https://i.natgeofe.com/k/9acd2bad-fb0e-43a8-935d-ec0aefc60c2f/monarch-butterfly-grass\_3x2.jpg",

    "https://image.shutterstock.com/image-photo/green-leaves-philodendron-plant-nature-260nw-2477697533.jpg"

]

*# Download images*

images = []

for url in image\_urls:

    response = requests.get(url)

    img = Image.open(BytesIO(response.content))

    images.append(img)

*# Resize images to 256x256 pixels*

resized\_images = [img.resize((256, 256)) for img in images]

*# Convert images to grayscale*

gray\_images = [cv2.cvtColor(np.array(img), cv2.COLOR\_RGB2GRAY) for img in resized\_images]

*# Extract boundary features using Canny edge detection*

boundary\_features = [cv2.Canny(img, 100, 200) for img in gray\_images]

*# Extract region features (using image moments)*

region\_features = [cv2.moments(img) for img in gray\_images]

*# Convert region features to a feature vector*

feature\_vectors = []

for moments in region\_features:

    if moments["m00"] != 0:

        cx = *int*(moments["m10"] / moments["m00"])

        cy = *int*(moments["m01"] / moments["m00"])

    else:

        cx, cy = 0, 0

    feature\_vectors.append([cx, cy])

*# Display results*

print("Boundary Features (Canny edges):")

for i, bf in enumerate(boundary\_features):

    print(*f*"Image {i+1}:")

    print(bf)

print("\nRegion Features (Centroid coordinates):")

for i, fv in enumerate(feature\_vectors):

    print(*f*"Image {i+1}: Centroid = {fv}")

print("\nFeature Vectore:")

print(feature\_vectors)

**Output:**

