**IBM CLOUD DEVELOPMENT PROJECT**

**COLLEGE CODE:4224**

**UNIVERSITY COLLEGE OF ENGINEERING.TINDIVANAM**

**MELPAKKAM 604001**

**IMAGE RECOGNITION WITH IBM CLOUD**

**VISUAL RECOGNITION**

**PROJECT TITLE FOR IMAGE RECOGNITION**

**IS TO IMAGE EDGE DETECTION**

**SUBMITTED BY**

**K.KAVITHA**

**R.RAMYA**

**V.SRISHA**

## INTRODUCTION:

Edge detection is a fundamental technique in image processing and computer vision that identifies boundaries within images. By pinpointing sharp intensity variations or changes in color, edge detection algorithms can discern the outlines of objects and important features in a scene. Commonly used algorithms include the Sobel, Canny, and Prewitt operators. The ability to accurately detect edges is crucial for tasks such as object recognition, image segmentation, and feature extraction, making it an essential tool in a plethora of applications ranging from medical imaging to autonomous vehicles.

## LIBRARIES FOR IMAGE EDGE DETECTION:

## 1.OpenCV(open source computer vision library):

## Face Recognition:

You can use OpenCV to detect and recognize faces using various methods like Eigenfaces, Fisherfaces, or LBPH.

## Object Detection:

OpenCV offers pre-trained classifiers such as Haar Cascades and DNN modules that can detect objects like faces, eyes, and even full bodies.

## Augmented Reality:

By finding the relationship between 2D image points (from the camera) and 3D world points (object's physical location), you can overlay virtual 3D objects on a 2D image.

## Stitching Images:

If you have multiple images of a scene from different angles, OpenCV can stitch them together to produce a panoramic image.

## Motion Analysis and Object Tracking:

You can use OpenCV to track objects in a video or detect movement.

Gesture Recognition:

Recognize and interpret gestures made by hand motions.

## 3D Reconstruction:

With two or more cameras, you can capture different views of an object and create its 3D model.

## Video Editing:

OpenCV can be used to apply various effects, filters, and transformations to videos.

## Machine Learning:

OpenCV integrates with machine learning frameworks like TensorFlow and PyTorch, allowing you to use and train models for various tasks.

## Mobile and Web Apps:

OpenCV offers bindings for Python, Java, and JavaScript, enabling the integration of computer vision capabilities into mobile and web applications.

## 2.Scikit-image:

# Import necessary libraries

import numpy as np

import matplotlib.pyplot as plt

from skimage import io, color, filters

# Read the image

image = io.imread('https://image.shutterstock.com/image-photo/image-260nw-725888862.jpg') # Replace with your image URL or path

# Convert the image to grayscale

gray\_image = color.rgb2gray(image)

# Apply Gaussian filter

smooth\_image = filters.gaussian(gray\_image, sigma=1)

# Display the original and processed images

fig, ax = plt.subplots(1, 3, figsize=(15, 5))

ax[0].imshow(image)

ax[0].set\_title("Original Image")

ax[0].axis("off")

ax[1].imshow(gray\_image, cmap='gray')

ax[1].set\_title("Grayscale Image")

ax[1].axis("off")

ax[2].imshow(smooth\_image, cmap='gray')

ax[2].set\_title("Smoothed Image")

ax[2].axis("off")

## 3.Matplotlib:

x = [0, 1, 2, 3, 4, 5]

y = [0, 1, 4, 9, 16, 25]

# Create a figure and axis

fig, ax = plt.subplots()

# Plot the data

ax.plot(x, y, label='y = x^2')

# Set the labels and title

ax.set\_xlabel('x')

ax.set\_ylabel('y')

ax.set\_title('Simple Plot')

ax.legend()

## 4.PIL( Python imaging library)/Pillow:

## Purpose of PIL:

The main aim of PIL is to allow for quick image processing using Python. This means being able to do operations like reading image data, manipulating the data, and then writing the modified data back out again.

## Supported Formats:

PIL supports a variety of image formats, including popular ones like JPEG, PNG, BMP, TIFF, and others.

## Performance:

Though written in Python, certain parts of PIL are implemented in C for performance. This makes many operations in PIL reasonably fast.

## Limitations & The Rise of Pillow:

Over time, the development of PIL slowed down, leading to the lack of support for some newer image formats and some lingering bugs.Pillow, a fork of PIL, came into the picture to address these issues. It's essentially an improved and maintained version of PIL, adding user-friendly features and fixing old bugs.

## Interoperability:

PIL works well with many other Python libraries, like NumPy (for numerical operations on image data) and Matplotlib (for displaying images).

## Usage Areas:

Web development (e.g., creating thumbnails, applying watermarks).Scientific research (e.g., analyzing visual data, modifying microscopy images).Automation (e.g., batch processing images, converting image formats).Graphics design (e.g., creating basic graphics, applying effects).

## 5.Tensorflow and PyTorch:

TensorFlow has a number of tools and features specifically designed for deploying models to production and monitoring their performance, while PyTorch has a more flexible and modular design that is well suited to rapid prototyping and experimentation.

## 6.Simple ITK:

## Purpose:

SimpleITK provides a simplified interface to ITK, making it easier for developers to create image analysis applications.

Languages:

While ITK is primarily a C++ library, SimpleITK offers bindings for several programming languages, including Python, R, Java, C#, Lua, Ruby, and TCL.

## Functionalities:

SimpleITK retains the powerful image processing capabilities of ITK, such as filtering, segmentation, and registration, but with an easier-to-use interface.

## Usage:

Import simpleITK as sitk

image = sitk.ReadImage("path\_to\_your\_image.nii")

smoothed\_image = sitk.SmoothingRecursiveGaussian(image, sigma=2.0)

sitk.WriteImage(smoothed\_image,

"path\_to\_save\_smoothed\_image.nii")

## Documentation & Community:

SimpleITK has a comprehensive set of documentation, tutorials, and an active community which makes it easier for newcomers to get started and for experienced developers to solve problems.

## 7.Mahotas:

A computer vision and image processing library for Python.

Provides fast functions for edge detection like sobel and laplacian.

import mahotas

image = mahotas.imread('image.jpg', as\_grey=True)

edges = mahotas.sobel(image)

## 8.Edge Detection using deep learning:

## Popular Deep Learning Models for Edge Detection:

**HED (Holistically-Nested Edge Detection):**

A pioneering work in this field, HED connects deep supervision to every layer of a CNN, ensuring even the shallower layers are useful for edge detection.

**CEDN (Contour Edge Detection Network):**

A combination of low-level features and high-level semantics for better edge and contour detection.

## Dataset:

BSDS500 is a widely used dataset for edge detection. It contains 500 natural images with ground truth edge maps annotated by humans. When training or fine-tuning models for edge detection, this dataset can serve as a benchmark.

## PROGRAM:

% importing the image

I = rgb2gray(imread("flowers.jpg"));

subplot(2, 4, 1),

imshow(I);

title("Gray Scale Image");

% Sobel Edge Detection

J = edge(I, 'Sobel');

subplot(2, 4, 2),

imshow(J);

title("Sobel");

% Prewitt Edge detection

K = edge(I, 'Prewitt');

subplot(2, 4, 3),

imshow(K);

title("Prewitt");

% Robert Edge Detection

L = edge(I, 'Roberts');

subplot(2, 4, 4),

imshow(L);

title("Robert");

% Log Edge Detection

M = edge(I, 'log');

subplot(2, 4, 5),

imshow(M);

title("Log");

% Zerocross Edge Detection

M = edge(I, 'zerocross');

subplot(2, 4, 6),

imshow(M);

title("Zerocross");

% Canny Edge Detection

N = edge(I, 'Canny');

subplot(2, 4, 7),

imshow(N);

title("Canny");

## OUTPUT:

## 

## CONCLUSION:

Image detection, a subset of computer vision, refers to the capability of software and machines to identify objects, features, or activities in images. It can be used for various applications, such as facial recognition, autonomous vehicles, and medical imaging.