MACHINE VISION LAB 5

Ex. Edge Detection Using Sobel, Robert and Prewitt.

NAME : OM SUBRATO DEY

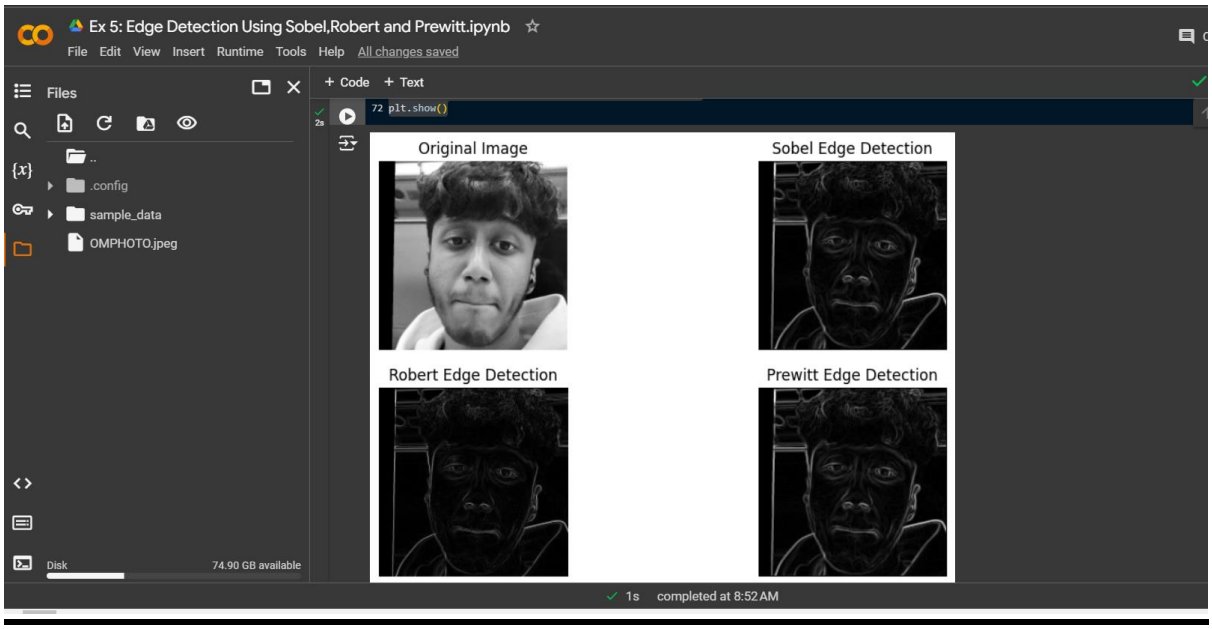
REGISTER NUMBER : 21BAI1876

CODE:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Load the image
img = cv2.imread('/content/OMPHOTO.jpeg', 0)

# Define the Sobel operators
sobel_x = np.array([[-1, 0, 1],[-2, 0, 2],[-1, 0, 1]])
sobel_y = np.array([[-1, -2, -1],[0, 0, 0],[1, 2, 1]])

# Define the Robert operators
robert_x = np.array([[1, 0],[0, -1]])
robert_y = np.array([[0, 1],[-1, 0]])

# Define the Prewitt operators
prewitt_x = np.array([[-1, 0, 1],[-1, 0, 1],[-1, 0, 1]])
prewitt_y = np.array([[-1, -1, -1],[0, 0, 0],[1, 1, 1]])

# Function to apply the operators
def apply_operator(img, operator):
    rows, cols = img.shape
    result = np.zeros((rows, cols))
    operator_rows, operator_cols = operator.shape
    for i in range(rows - operator_rows + 1):
        for j in range(cols - operator_cols + 1):
            sum = 0
            for k in range(operator_rows):
                for l in range(operator_cols):
```

```
                    sum += img[i + k, j + l] * operator[k, l]
            result[i + 1, j + 1] = sum
    return result

# Apply the Sobel operators
sobel_x_img = apply_operator(img, sobel_x)
sobel_y_img = apply_operator(img, sobel_y)
sobel_img = np.sqrt(sobel_x_img**2 + sobel_y_img**2)

# Apply the Robert operators
robert_x_img = apply_operator(img, robert_x)
robert_y_img = apply_operator(img, robert_y)
robert_img = np.sqrt(robert_x_img**2 + robert_y_img**2)

# Apply the Prewitt operators
prewitt_x_img = apply_operator(img, prewitt_x)
prewitt_y_img = apply_operator(img, prewitt_y)
prewitt_img = np.sqrt(prewitt_x_img**2 + prewitt_y_img**2)

# Display the results
plt.figure(figsize=(10, 6))
plt.subplot(2, 2, 1), plt.imshow(img, cmap='gray')
plt.title('Original Image'), plt.axis('off')
plt.subplot(2, 2, 2), plt.imshow(sobel_img, cmap='gray')
plt.title('Sobel Edge Detection'), plt.axis('off')
plt.subplot(2, 2, 3), plt.imshow(robert_img, cmap='gray')
plt.title('Robert Edge Detection'), plt.axis('off')
plt.subplot(2, 2, 4), plt.imshow(prewitt_img, cmap='gray')
plt.title('Prewitt Edge Detection'), plt.axis('off')
plt.show()
```

OUTPUT:


Ex 5: Edge Detection Using Sobel,Robert and Prewitt.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Files
+ Code + Text
72 plt.show()
.config
sample_data
OMPHOTO.jpeg
Disk 74.90 GB available
Original Image
Sobel Edge Detection
Robert Edge Detection
Prewitt Edge Detection
1s completed at 8:52 AM

MACHINE VISION LAB 6

Ex. Image Segmentation (Thresholding and Region Growing)

NAME : OM SUBRATO DEY

REGISTER NUMBER : 21BAI1876

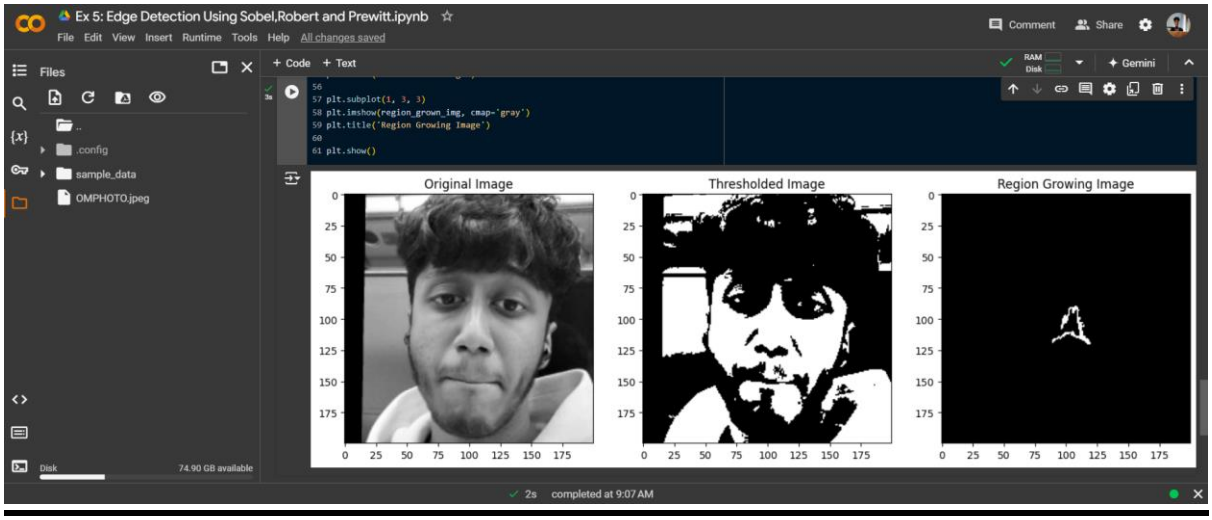
CODE:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Load the image in grayscale
image_path = '/content/OMPHOTO.jpeg'
img = cv2.imread(image_path, 0)  # 0 for grayscale

# Thresholding (binary segmentation)
_, thresh_img = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY)

# Region Growing function
def region_growing(img, seed, threshold=10):
    height, width = img.shape
    segmented_img = np.zeros_like(img)
    visited = np.zeros_like(img, dtype=bool)

    # Initialize the region with the seed point
    region_pixels = [seed]
    region_intensity = img[seed]
    segmented_img[seed] = 255
    visited[seed] = True

    # Define the 4-connectivity (up, down, left, right)
    directions = [(-1, 0), (1, 0), (0, -1), (0, 1)]

    while region_pixels:
        current_pixel = region_pixels.pop(0)
```

```
        for direction in directions:
            x = current_pixel[0] + direction[0]
            y = current_pixel[1] + direction[1]

            if 0 <= x < height and 0 <= y < width and not
visited[x, y]:
                if abs(int(img[x, y]) - int(region_intensity)) <
threshold:
                    segmented_img[x, y] = 255
                    visited[x, y] = True
                    region_pixels.append((x, y))

    return segmented_img

# Seed point for region growing (you can adjust this)
seed_point = (100, 100)  # Example: manually set a seed
region_grown_img = region_growing(img, seed_point, threshold=10)

# Plot original, thresholded, and region growing images
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
plt.imshow(img, cmap='gray')
plt.title('Original Image')

plt.subplot(1, 3, 2)
plt.imshow(thresh_img, cmap='gray')
plt.title('Thresholded Image')

plt.subplot(1, 3, 3)
plt.imshow(region_grown_img, cmap='gray')
```

```
plt.title('Region Growing Image')
plt.show()
```

OUTPUT:



COLAB NOTEBOOK LINK:

https://colab.research.google.com/drive/1QThx-6z5IP0Ux9SDSUZPih0YkHE_zjPl?usp=sharing