

Problem No: 03

Problem Name: Perform edge detection operation (Canny, Sobel, Morphological edge detection)

Description: Edge detection is a key image processing technique used to identify the boundaries or outlines of objects within an image. It works by detecting sharp changes in pixel intensity, which often corresponds to object edges. The main goal is to simplify the image while preserving important structural details. Common methods include Canny, Sobel, and Morphological edge detection, each with its unique approach and accuracy. These methods are widely used in object recognition, medical imaging, and computer vision applications.

Canny Edge Detection

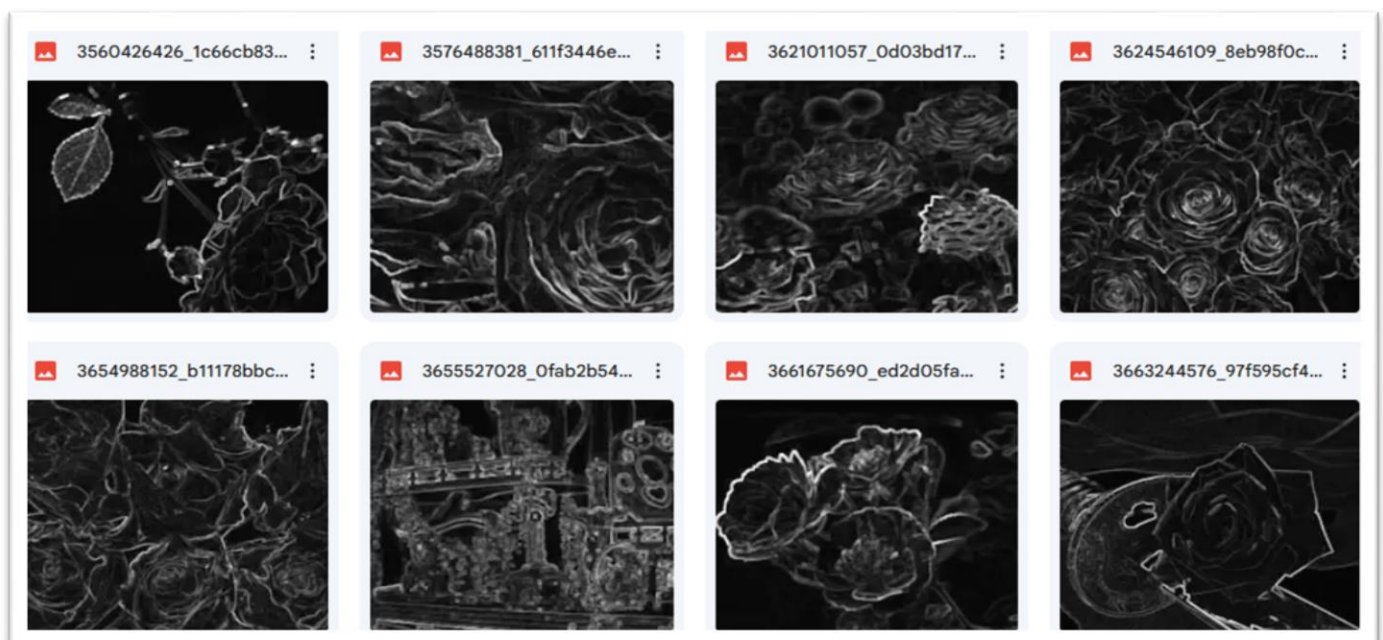
Canny edge detection is a multi-stage algorithm known for its accuracy and noise resistance. It includes steps such as noise reduction (using Gaussian blur), gradient calculation, non-maximum suppression, and hysteresis thresholding. This method produces thin and precise edges, making it ideal for detecting fine details in complex images.

Sobel Edge Detection

Sobel edge detection uses gradient operators to measure the rate of change in pixel intensity in both horizontal and vertical directions. It applies two 3×3 convolution masks (G_x and G_y) to highlight edges along the x and y axes. The final edge magnitude is computed by combining both gradients. It is simple, efficient, and suitable for detecting prominent edges in images.

Morphological Edge Detection

Morphological edge detection is based on mathematical morphology and uses operations like **dilation** and **erosion**. The edge is extracted by subtracting the eroded image from the dilated version of the same image. It works effectively on binary or grayscale images and is useful in shape-based analysis. This method is often used in segmentation and object boundary extraction. Images of grayscale folder:



Source code and Output:

Canny edge detection:

```
import os
import cv2
from glob import glob
output_dir_edge = '/content/drive/MyDrive/edge_flower'
os.makedirs(output_dir_edge, exist_ok=True)
# Process each image in the gray_folder
for img_path in gray_folder:
    img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
    edges = cv2.Canny(img, 100, 200) # Example thresholds
    filename = os.path.basename(img_path)
    output_path_edge = os.path.join(output_dir_edge, filename)
    cv2.imwrite(output_path_edge, edges)
print(f"Edge-detected images saved to {output_dir_edge}")
```

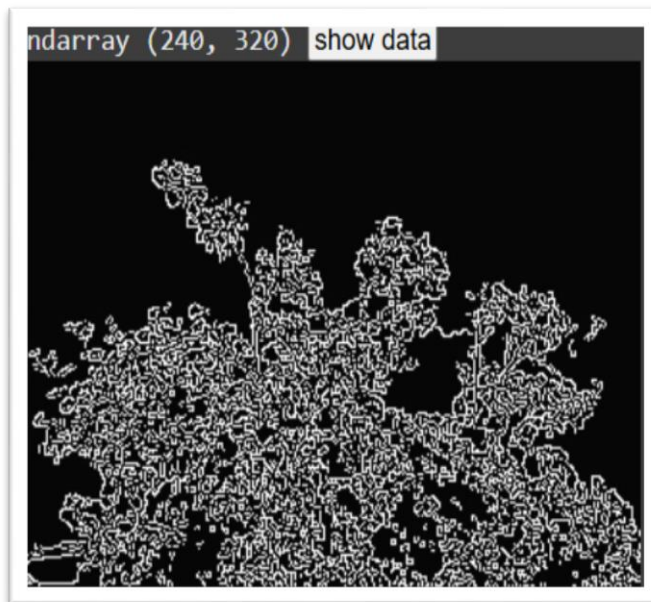


Figure name: Presentation of canny and sobel edge detection.

Sobel edge detection:

```
import matplotlib.pyplot as plt
import cv2
from glob import glob

# Assuming output_dir_sobel is defined from the previous step where Sobel images
# were saved
sobel_folder = glob('/content/drive/MyDrive/sobel_flower/*.jpg')
if len(sobel_folder) > 0:
    # Read and display the first Sobel edge-detected image
    img = cv2.imread(sobel_folder[0], cv2.IMREAD_GRAYSCALE)
    plt.imshow(img, cmap='gray')
    plt.title('Example Sobel Edge-Detected Image')
    plt.axis('off')
    plt.show()
else: print("No Sobel edge-detected images found in the specified directory.")
```

Morphological edge detection:

```
import os
import cv2
from glob import glob
import numpy as np
# Define the destination directory in Google Drive for morphological edge-detected
images
output_dir_morph = '/content/drive/MyDrive/morph_edge_flower'
# Create the directory if it doesn't exist
os.makedirs(output_dir_morph, exist_ok=True)
# Process each image in the gray_folder (assuming gray_folder is defined from a
previous step)
for img_path in gray_folder:
    # Read the grayscale image
    img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
    # Define a kernel for morphological operations (e.g., a 3x3 kernel)
    kernel = np.ones((3,3), np.uint8)
    # Apply morphological gradient (dilation - erosion) to find edges
    morph_edges = cv2.morphologyEx(img, cv2.MORPH_GRADIENT, kernel)
    # Get the filename and create the output path in Google Drive
    filename = os.path.basename(img_path)
    output_path_morph = os.path.join(output_dir_morph, filename)
    # Save the morphological edge-detected image to Google Drive
    cv2.imwrite(output_path_morph, morph_edges)
print(f"Morphological edge-detected images saved to {output_dir_morph}")
# Define morph_folder after saving the images
morph_folder = glob('/content/drive/MyDrive/morph_edge_flower/*.jpg')

if len(morph_folder) > 0:
    # Read the first image from the morph_folder
    img = cv2.imread(morph_folder[0], cv2.IMREAD_GRAYSCALE)
    # Display the image
    plt.imshow(img, cmap='gray')
    plt.title('First Morphological Edge-Detected Image')
    plt.axis('off')
    plt.show()
else: print("The morph_edge_flower folder is empty.")
```

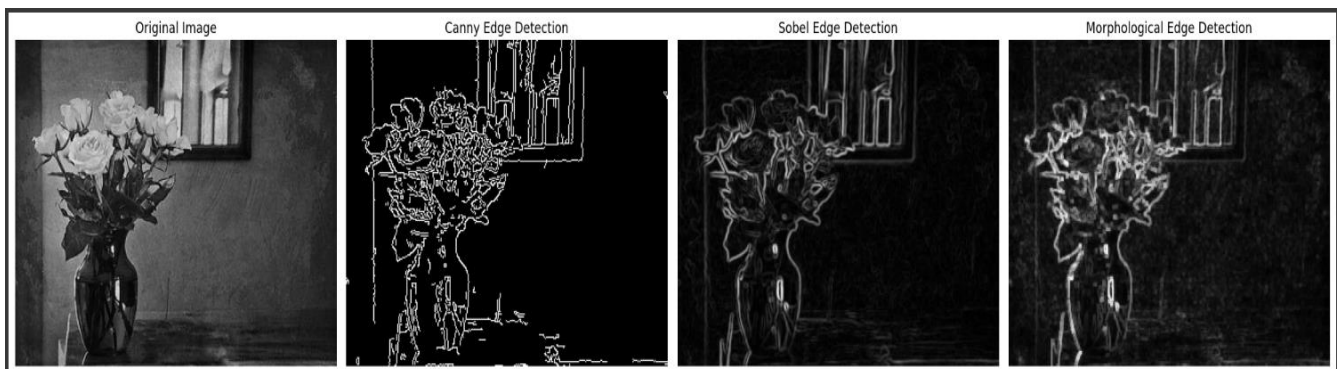


Figure: Comparison of canny, sobel and morphological edge detection.