16.Implement a Sobel algorithm using Open CV to filter the input image.

**AIM:**

To implement the **Sobel Edge Detection Algorithm** using OpenCV to filter an input image.

**PROCEDURE:**

1. Install OpenCV if not already installed using:

* pip install opencv-python

1. Import the required libraries (cv2 and numpy).
2. Read the input image using cv2.imread().
3. Convert the image to grayscale using cv2.cvtColor().
4. Apply the **Sobel filter** in the X and Y directions using cv2.Sobel().
5. Compute the absolute gradient magnitude using cv2.convertScaleAbs().
6. Combine the X and Y gradients using cv2.addWeighted().
7. Display the original, Sobel X, Sobel Y, and combined edge images using cv2.imshow().
8. Save the output images using cv2.imwrite(), if needed.
9. Wait for a key press and close all image windows using cv2.waitKey(0) and cv2.destroyAllWindows().

**PROGRAM:**

import cv2

import numpy as np

# Read the input image

image = cv2.imread("image.jpg") # Replace with your image file path

# Convert the image to grayscale

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

# Apply Sobel filter in X and Y directions

sobel\_x = cv2.Sobel(gray, cv2.CV\_64F, 1, 0, ksize=3) # Gradient in X direction

sobel\_y = cv2.Sobel(gray, cv2.CV\_64F, 0, 1, ksize=3) # Gradient in Y direction

# Convert gradients to absolute scale

sobel\_x = cv2.convertScaleAbs(sobel\_x)

sobel\_y = cv2.convertScaleAbs(sobel\_y)

# Combine the gradients

sobel\_combined = cv2.addWeighted(sobel\_x, 0.5, sobel\_y, 0.5, 0)

# Display images

cv2.imshow("Original Image", image)

cv2.imshow("Sobel X", sobel\_x)

cv2.imshow("Sobel Y", sobel\_y)

cv2.imshow("Sobel Combined", sobel\_combined)

# Save the results

cv2.imwrite("sobel\_x.jpg", sobel\_x)

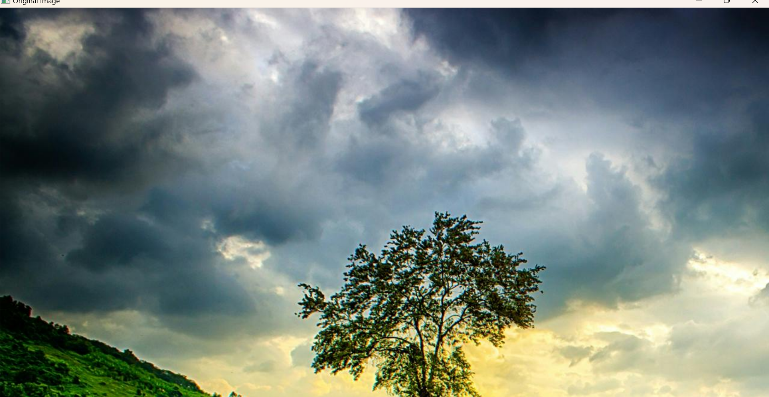
cv2.imwrite("sobel\_y.jpg", sobel\_y)

cv2.imwrite("sobel\_combined.jpg", sobel\_combined)

# Wait for a key press and close all windows

cv2.waitKey(0)

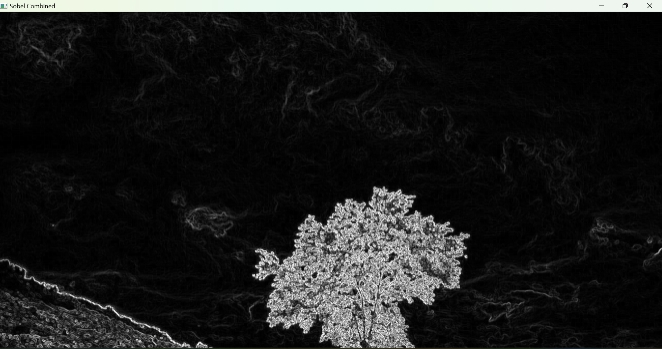
cv2.destroyAllWindows()

**INPUT:**

**OUTPUT:**



Sobel\_X Sobel\_Y



Sobel\_Combination

**RESULT :**

The program successfully applies the **Sobel Edge Detection Algorithm** to the given image, detects edges in both directions, displays the results, and saves them as "sobel\_x.jpg", "sobel\_y.jpg", and "sobel\_combined.jpg".