import cv2

import numpy as np

from matplotlib import pyplot as plt

# Load sample image (you can replace 'sample\_image.jpg' with your image path)

image\_path = r'C:\Users\admin\Desktop\abdul kalam.jpg'

img = cv2.imread(image\_path, cv2.IMREAD\_COLOR)

# Convert image to grayscale

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

# Apply Gaussian blur to reduce noise

blurred\_gray = cv2.GaussianBlur(gray, (3, 3), 0)

# Apply Sobel edge detection

edges\_x\_sobel = cv2.Sobel(blurred\_gray, cv2.CV\_64F, 1, 0, ksize=3)

edges\_y\_sobel = cv2.Sobel(blurred\_gray, cv2.CV\_64F, 0, 1, ksize=3)

edges\_sobel = cv2.magnitude(edges\_x\_sobel, edges\_y\_sobel)

# Apply Canny edge detection

edges\_canny = cv2.Canny(blurred\_gray, 100, 200)

# Apply Laplacian edge detection on blurred image

edges\_laplacian = cv2.Laplacian(blurred\_gray, cv2.CV\_64F, ksize=3)

# Display original and edge-detected images

plt.figure(figsize=(18, 12))

plt.subplot(231), plt.imshow(cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)), plt.title('Original Image')

plt.subplot(232), plt.imshow(edges\_sobel, cmap='gray'), plt.title('Sobel Edge Detection')

plt.subplot(233), plt.imshow(edges\_canny, cmap='gray'), plt.title('Canny Edge Detection')

plt.subplot(234), plt.imshow(edges\_laplacian, cmap='gray'), plt.title('Laplacian Edge Detection')

plt.show()