

```
import cv2
import numpy as np
from skimage.metrics import structural_similarity as ssim

# Load an image in grayscale
image = cv2.imread('image.jpg', cv2.IMREAD_GRAYSCALE)

# Apply Sobel edge detection
sobel_x = cv2.Sobel(image, cv2.CV_64F, 1, 0, ksize=3)
sobel_y = cv2.Sobel(image, cv2.CV_64F, 0, 1, ksize=3)
sobel_edges = cv2.magnitude(sobel_x, sobel_y).astype('uint8')

# Apply Canny edge detection
canny_edges = cv2.Canny(image, 100, 200)

# Calculate Mean Squared Error (MSE)
mse_sobel = np.mean((image - sobel_edges)**2)
mse_canny = np.mean((image - canny_edges)**2)

# Calculate Structural Similarity Index (SSI)
ssi_sobel = ssim(image, sobel_edges)
ssi_canny = ssim(image, canny_edges)

# Display the metrics
print(f"MSE (Sobel): {mse_sobel:.2f}, MSE (Canny): {mse_canny:.2f}")
print(f"SSI (Sobel): {ssi_sobel:.2f}, SSI (Canny): {ssi_canny:.2f}")

# Display the original, Sobel edges, and Canny edges
cv2.imshow('Original Image', image)
cv2.imshow('Sobel Edges', sobel_edges)
cv2.imshow('Canny Edges', canny_edges)
cv2.waitKey(0)
cv2.destroyAllWindows()
```