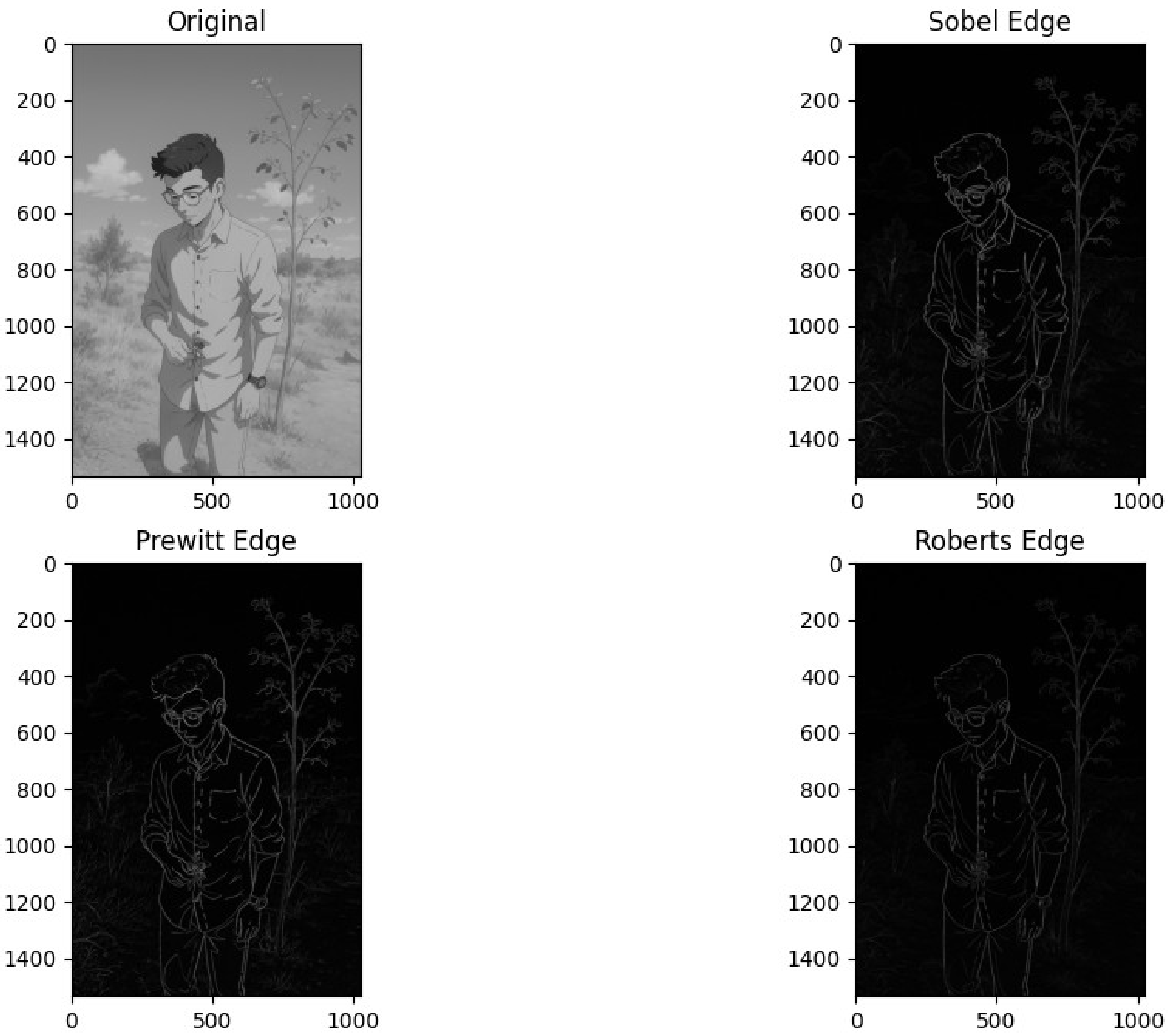
Gradient-Based

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

# Load grayscale image
img = cv2.imread("/content/Test 1.png", cv2.IMREAD_GRAYSCALE)

# Sobel operator
sobel_x = cv2.Sobel(img, cv2.CV_64F, 1, 0, ksize=3)
sobel_y = cv2.Sobel(img, cv2.CV_64F, 0, 1, ksize=3)
sobel_edge = cv2.magnitude(sobel_x, sobel_y)

# Prewitt operator (using kernels)
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]])

prewitt_x_img = cv2.filter2D(img, -1, prewitt_x)
prewitt_y_img = cv2.filter2D(img, -1, prewitt_y)
prewitt_edge = cv2.magnitude(np.float32(prewitt_x_img),
np.float32(prewitt_y_img))

# Roberts operator
roberts_x = np.array([[1, 0],
                      [0, -1]])
roberts_y = np.array([[0, 1],
                      [-1, 0]])
roberts_x_img = cv2.filter2D(img, -1, roberts_x)
roberts_y_img = cv2.filter2D(img, -1, roberts_y)
roberts_edge = cv2.magnitude(np.float32(roberts_x_img),
np.float32(roberts_y_img))

# Display results
plt.figure(figsize=(12,8))

plt.subplot(2,2,1), plt.imshow(img, cmap='gray'),
plt.title("Original")
plt.subplot(2,2,2), plt.imshow(sobel_edge, cmap='gray'),
plt.title("Sobel Edge")
plt.subplot(2,2,3), plt.imshow(prewitt_edge, cmap='gray'),
plt.title("Prewitt Edge")
plt.subplot(2,2,4), plt.imshow(roberts_edge, cmap='gray'),
plt.title("Roberts Edge")

plt.show()
```



Original

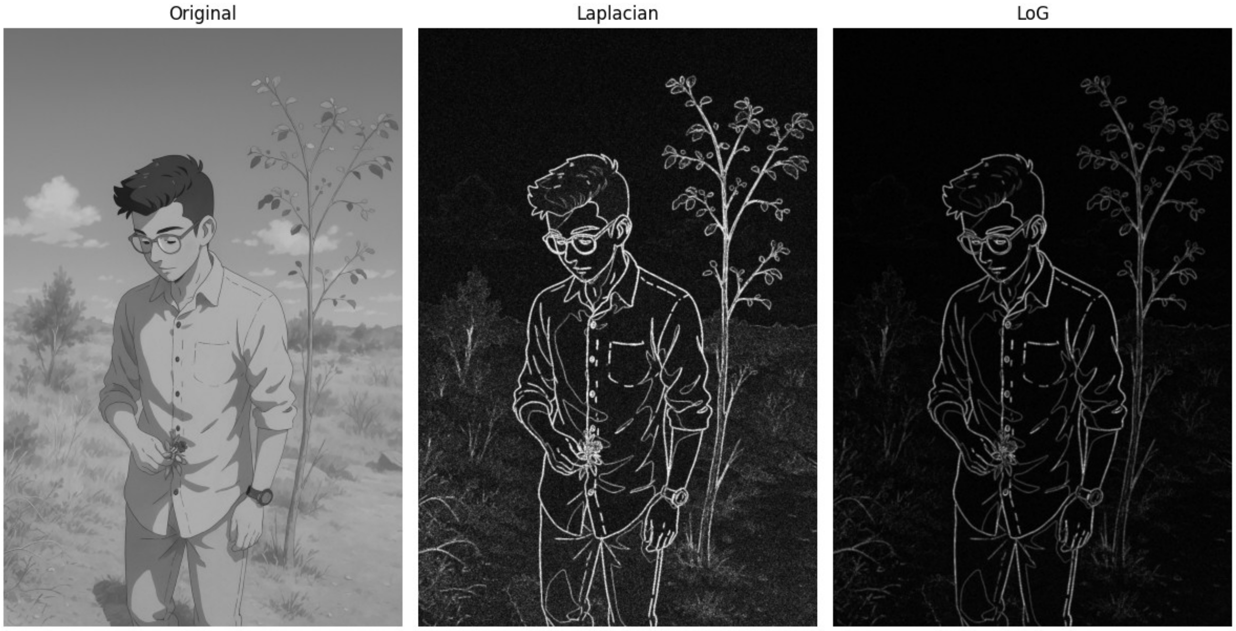
Sobel Edge

Prewitt Edge

Roberts Edge

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

# Load grayscale image
img = cv2.imread("/content/Test 1.png", cv2.IMREAD_GRAYSCALE)

# Laplacian operator (second-order derivative)
laplacian = cv2.Laplacian(img, cv2.CV_64F, ksize=3)
laplacian = cv2.convertScaleAbs(laplacian)

# Laplacian of Gaussian (LoG)
# Step 1: Apply Gaussian blur to reduce noise
gaussian_blur = cv2.GaussianBlur(img, (3,3), 0)
# Step 2: Apply Laplacian
log = cv2.Laplacian(gaussian_blur, cv2.CV_64F, ksize=3)
log = cv2.convertScaleAbs(log)
```

```python

# Display results
plt.figure(figsize=(12,6))

plt.subplot(1,3,1), plt.imshow(img, cmap='gray'),
plt.title("Original"), plt.axis("off")
plt.subplot(1,3,2), plt.imshow(laplacian, cmap='gray'),
plt.title("Laplacian"), plt.axis("off")
plt.subplot(1,3,3), plt.imshow(log, cmap='gray'), plt.title("LoG"),
plt.axis("off")

plt.tight_layout()
plt.show()
```


Original
Laplacian
LoG