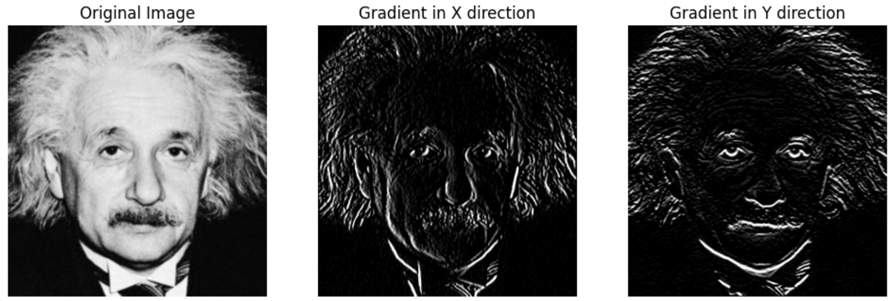
```
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt

from google.colab import drive
drive.mount("/content/drive")
image = cv.imread('/content/drive/MyDrive/images/einstein.png', cv.IMREAD_GRAYSCALE)
assert image is not None # Check if the image is loaded successfully

# Define the Sobel filter kernels
sobel_x = np.array([[-1, 0, 1],
                        [-2, 0, 2],
                        [-1, 0, 1]])

sobel_y = np.array([[-1, -2, -1],
                        [0, 0, 0],
                        [1, 2, 1]])

# Apply the Sobel operator using filter2D
gradient_x = cv2.filter2D(image, -1, sobel_x)
gradient_y = cv2.filter2D(image, -1, sobel_y)

# Compute the magnitude of gradients
gradient_magnitude = np.sqrt(np.square(gradient_x) + np.square(gradient_y))

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

plt.subplot(1, 3, 1)
plt.imshow(image, cmap='gray')
plt.title('Original Image')
plt.axis('off')

plt.subplot(1, 3, 2)
plt.imshow(gradient_x, cmap='gray')
plt.title('Gradient in X direction')
plt.axis('off')

plt.subplot(1, 3, 3)
plt.imshow(gradient_y, cmap='gray')
plt.title('Gradient in Y direction')
plt.axis('off')

plt.show()
```

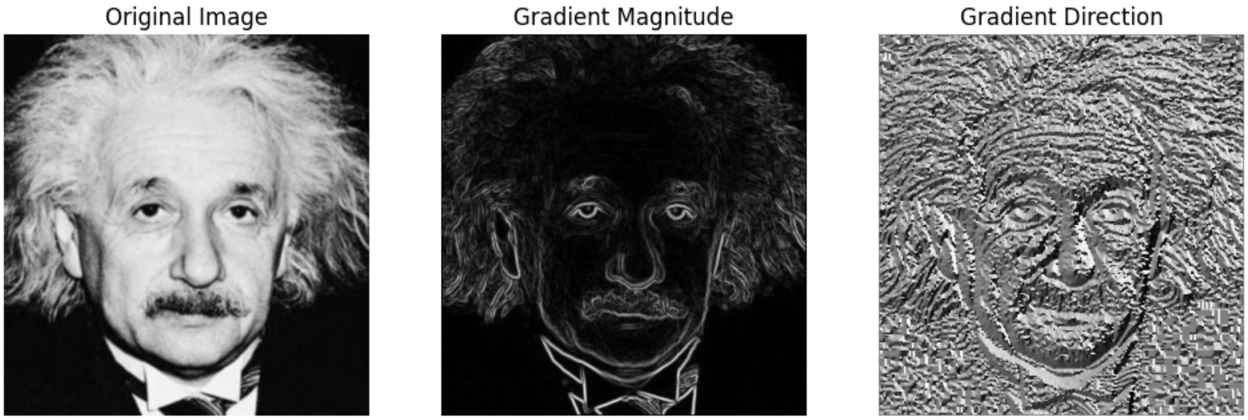
```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mour
```



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

from google.colab import drive
drive.mount("/content/drive")
image = cv.imread('/content/drive/MyDrive/images/einstein.png', cv.IMREAD_GRAYSCALE)
assert image is not None # Check if the image is loaded successfully

# Sobel kernels
```

```
sobel_x = np.array([[-1, 0, 1],
                        [-2, 0, 2],
                        [-1, 0, 1]])

sobel_y = np.array([[-1, -2, -1],
                        [0, 0, 0],
                        [1, 2, 1]])

# Initialize gradient images
gradient_x = np.zeros_like(image, dtype=np.float32)
gradient_y = np.zeros_like(image, dtype=np.float32)

# Apply Sobel filter using nested loops
rows, cols = image.shape
for i in range(1, rows - 1):
  for j in range(1, cols - 1):
    gx = np.sum(sobel_x * image[i-1:i+2, j-1:j+2])
    gy = np.sum(sobel_y * image[i-1:i+2, j-1:j+2])
    gradient_x[i, j] = gx
    gradient_y[i, j] = gy

    # Compute magnitude and direction of gradients
gradient_magnitude = np.sqrt(gradient_x**2 + gradient_y**2)
gradient_direction = np.arctan2(gradient_y, gradient_x)

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

plt.subplot(1, 3, 1)
plt.imshow(image, cmap='gray')
plt.title('Original Image')
plt.axis('off')

plt.subplot(1, 3, 2)
plt.imshow(gradient_magnitude, cmap='gray')
plt.title('Gradient Magnitude')
plt.axis('off')

plt.subplot(1, 3, 3)
plt.imshow(gradient_direction, cmap='gray')
plt.title('Gradient Direction')
plt.axis('off')

plt.show()
```

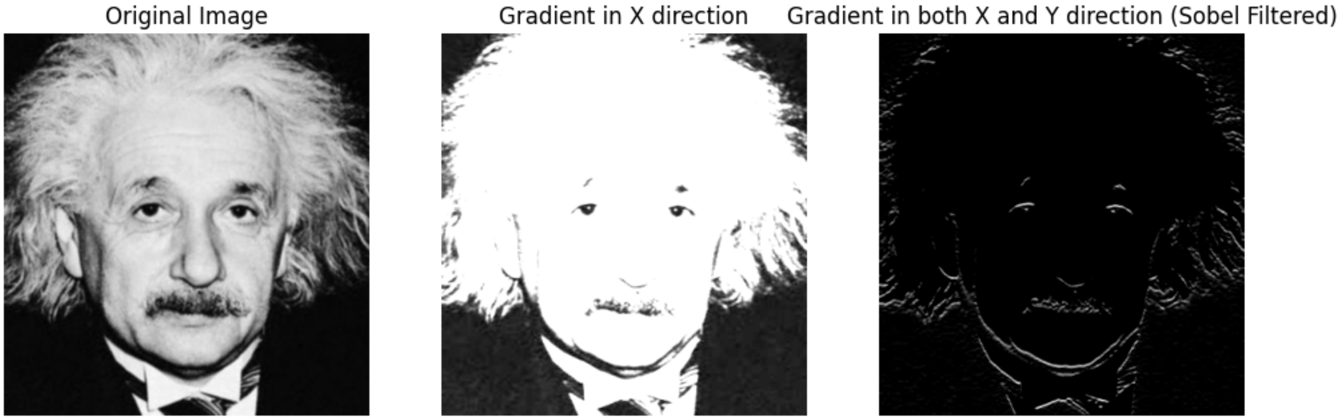
```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```



```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from google.colab import drive
drive.mount("/content/drive")
image = cv.imread('/content/drive/MyDrive/images/einstein.png', cv.IMREAD_GRAYSCALE)
assert image is not None # Check if the image is loaded successfully

# Check if the image is loaded successfully
if image is None:
    print("Error: Unable to load image.")
else:
```

```
# Define the separable Sobel kernels
sobel_horizontal = np.array([1, 2, 1])
sobel_vertical = np.array([1, 0, -1])

# Convolve image with the horizontal kernel
gradient_x = cv2.filter2D(image, -1, sobel_horizontal.reshape(1, 3))

# Convolve the result with the vertical kernel
gradient_xy = cv2.filter2D(gradient_x, -1, sobel_vertical.reshape(3, 1))

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

plt.subplot(1, 3, 1)
plt.imshow(image, cmap='gray')
plt.title('Original Image')
plt.axis('off')

plt.subplot(1, 3, 2)
plt.imshow(gradient_x, cmap='gray')
plt.title('Gradient in X direction')
plt.axis('off')

plt.subplot(1, 3, 3)
plt.imshow(gradient_xy, cmap='gray')
plt.title('Gradient in both X and Y direction (Sobel Filtered)')
plt.axis('off')

plt.show()
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```



Original Image

Gradient in X direction

Gradient in both X and Y direction (Sobel Filtered)