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

def image_convolution(f, w, debug=False):
    N, M = f.shape
    n, m = w.shape

    a = int((n-1)/2)
    b = int((m-1)/2)
```

```
    # obtem filtro invertido
    w_flip = np.flip(np.flip(w, 0), 1)
    g = np.zeros(f.shape, dtype=np.uint8)
```

```
    # para cada pixel na imagem:
    for x in range(a, N-a):
        for y in range(b, M-b):
            # obtem submatriz a ser usada na convolucao
            sub_f = f[x-a : x+a+1, y-b:y+b+1]
            if debug:
                print(f"{x},{y} - subimage:\n{sub_f}")
            # calcula g em x,y
            g[x,y] = np.sum(np.multiply(sub_f, w_flip)).astype(np.uint8)
    return g
```

```
def media_vizinhanca(img, tamanho=3):
    w_med = np.ones((tamanho, tamanho)) / (tamanho * tamanho)
    return image_convolution(img, w_med)
```

```
def media_k_vizinhos(img, tamanho=3, k=5):
    pad = tamanho // 2
    img_padded = np.pad(img, pad, mode='reflect')
    result = np.zeros_like(img)
    for i in range(img.shape[0]):
        for j in range(img.shape[1]):
            vizinhos = img_padded[i:i+tamanho, j:j+tamanho].flatten()
            vizinhos_ordenados = np.sort(vizinhos)
            media_k = np.mean(vizinhos_ordenados[:k])
            result[i, j] = media_k
    return result.astype(np.uint8)
```

```
def mediana(img, tamanho=3):
    return cv2.medianBlur(img, tamanho)
```

```
def laplaciano(img):
    return cv2.Laplacian(img, cv2.CV_64F)
```

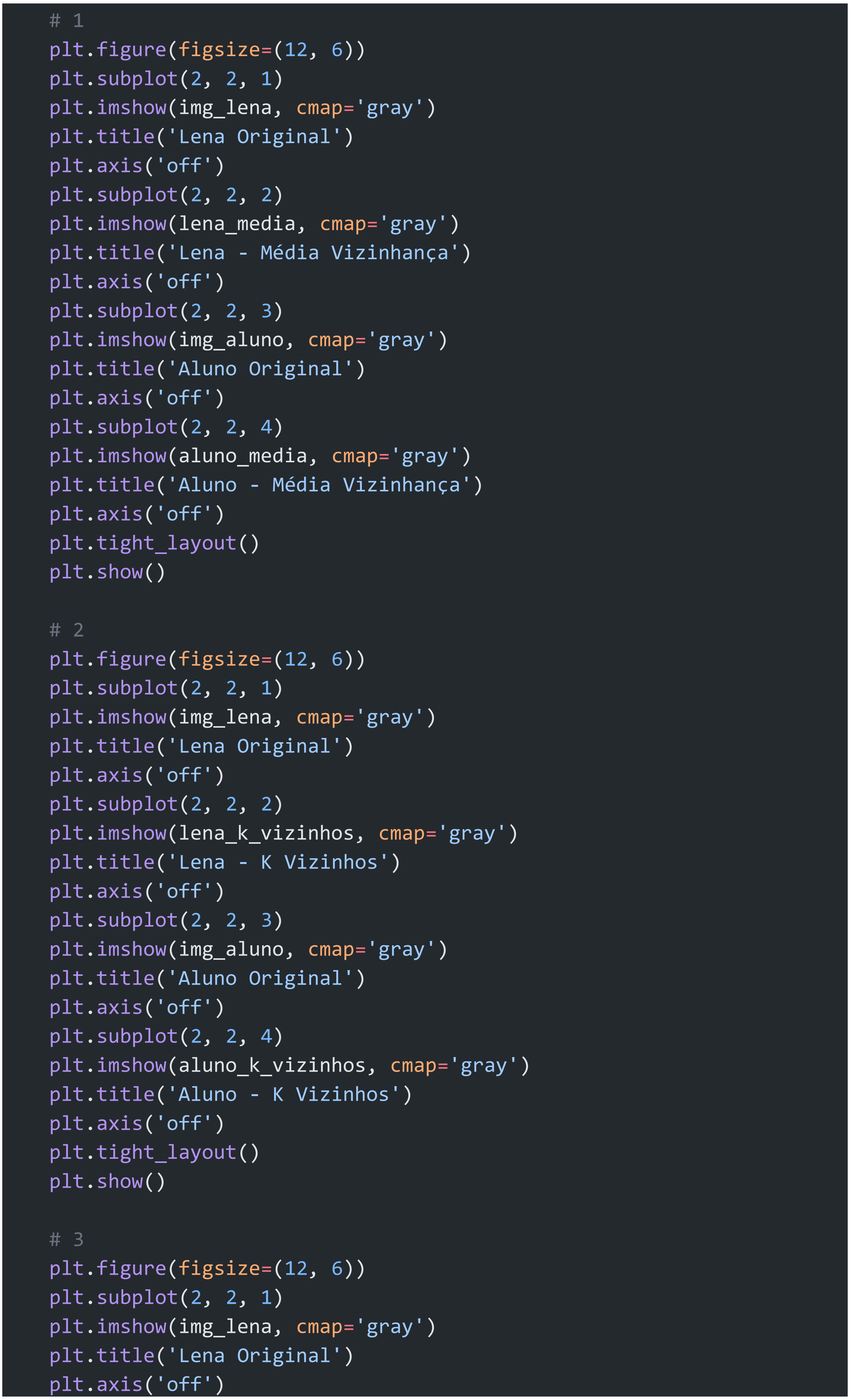
```
def roberts(img):
    kernelx = np.array([[1, 0], [0, -1]], dtype=np.float32)
    kernely = np.array([[0, 1], [-1, 0]], dtype=np.float32)
    gx = cv2.filter2D(img, -1, kernelx)
    gy = cv2.filter2D(img, -1, kernely)
    grad = np.sqrt(np.square(gx.astype(np.float32)) +
np.square(gy.astype(np.float32)))
    return np.clip(grad, 0, 255).astype(np.uint8)
```

```
def prewitt(img):
    kernelx = np.array([[1,1,1],[0,0,0],[-1,-1,-1]], dtype=np.float32)
    kernely = np.array([[-1,0,1],[-1,0,1],[-1,0,1]], dtype=np.float32)
    gx = cv2.filter2D(img, -1, kernelx)
    gy = cv2.filter2D(img, -1, kernely)
    return cv2.convertScaleAbs(gx + gy)
```

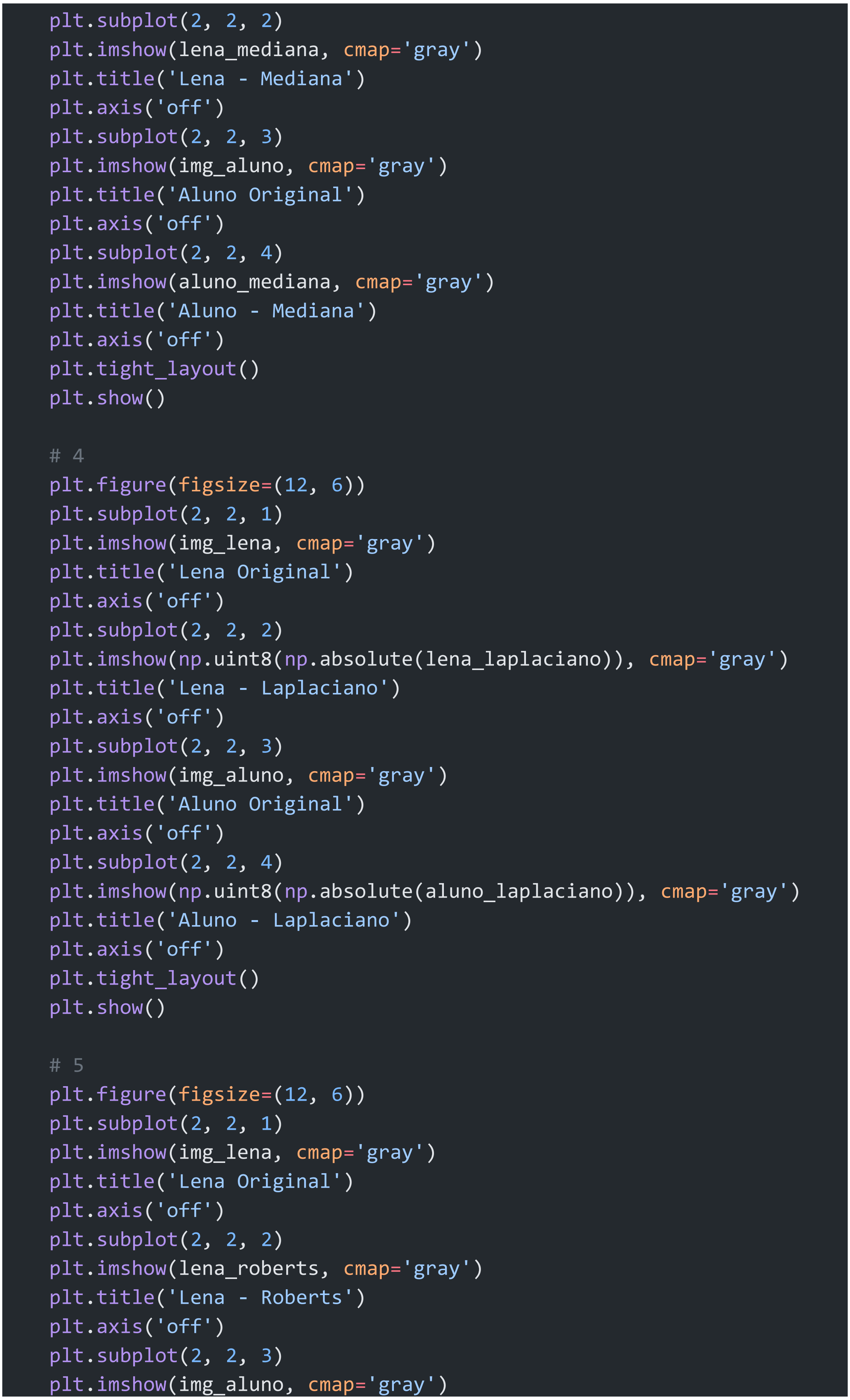
```
def sobel(img):
    gx = cv2.Sobel(img, cv2.CV_8U, 1, 0, ksize=5)
    gy = cv2.Sobel(img, cv2.CV_8U, 0, 1, ksize=5)
    return gx + gy
```

```
# carrega as imagens em cinza
img_lena = cv2.imread('lena.png', cv2.IMREAD_GRAYSCALE)
img_aluno = cv2.imread('img_aluno.jpg', cv2.IMREAD_GRAYSCALE)
```

```
if img_lena is not None and img_aluno is not None: # teste por que tava
dando ruim as imagens

    print("Aplicando filtros nas imagens...")

    lena_media = media_vizinhanca(img_lena, 5)
    aluno_media = media_vizinhanca(img_aluno, 5)

    lena_k_vizinhos = media_k_vizinhos(img_lena, 5, 9)
    aluno_k_vizinhos = media_k_vizinhos(img_aluno, 5, 9)

    lena_mediana = mediana(img_lena, 5)
    aluno_mediana = mediana(img_aluno, 5)

    lena_laplaciano = laplaciano(img_lena)
    aluno_laplaciano = laplaciano(img_aluno)

    lena_roberts = roberts(img_lena)
    aluno_roberts = roberts(img_aluno)

    lena_prewitt = prewitt(img_lena)
    aluno_prewitt = prewitt(img_aluno)

    lena_sobel = sobel(img_lena)
    aluno_sobel = sobel(img_aluno)
```



```
# 1
plt.figure(figsize=(12, 6))
plt.subplot(2, 2, 1)
plt.imshow(img_lena, cmap='gray')
plt.title('Lena Original')
plt.axis('off')
plt.subplot(2, 2, 2)
plt.imshow(lena_media, cmap='gray')
plt.title('Lena - Média Vizinhança')
plt.axis('off')
plt.subplot(2, 2, 3)
plt.imshow(img_aluno, cmap='gray')
plt.title('Aluno Original')
plt.axis('off')
plt.subplot(2, 2, 4)
plt.imshow(aluno_media, cmap='gray')
plt.title('Aluno - Média Vizinhança')
plt.axis('off')
plt.tight_layout()
plt.show()

# 2
plt.figure(figsize=(12, 6))
plt.subplot(2, 2, 1)
plt.imshow(img_lena, cmap='gray')
plt.title('Lena Original')
plt.axis('off')
plt.subplot(2, 2, 2)
plt.imshow(lena_k_vizinhos, cmap='gray')
plt.title('Lena - K Vizinhos')
plt.axis('off')
plt.subplot(2, 2, 3)
plt.imshow(img_aluno, cmap='gray')
plt.title('Aluno Original')
plt.axis('off')
plt.subplot(2, 2, 4)
plt.imshow(aluno_k_vizinhos, cmap='gray')
plt.title('Aluno - K Vizinhos')
plt.axis('off')
plt.tight_layout()
plt.show()

# 3
plt.figure(figsize=(12, 6))
plt.subplot(2, 2, 1)
plt.imshow(img_lena, cmap='gray')
plt.title('Lena Original')
plt.axis('off')
```



```python
plt.subplot(2, 2, 2)
plt.imshow(lena_mediana, cmap='gray')
plt.title('Lena - Mediana')
plt.axis('off')
plt.subplot(2, 2, 3)
plt.imshow(img_aluno, cmap='gray')
plt.title('Aluno Original')
plt.axis('off')
plt.subplot(2, 2, 4)
plt.imshow(aluno_mediana, cmap='gray')
plt.title('Aluno - Mediana')
plt.axis('off')
plt.tight_layout()
plt.show()

# 4
plt.figure(figsize=(12, 6))
plt.subplot(2, 2, 1)
plt.imshow(img_lena, cmap='gray')
plt.title('Lena Original')
plt.axis('off')
plt.subplot(2, 2, 2)
plt.imshow(np.uint8(np.absolute(lena_laplaciano)), cmap='gray')
plt.title('Lena - Laplaciano')
plt.axis('off')
plt.subplot(2, 2, 3)
plt.imshow(img_aluno, cmap='gray')
plt.title('Aluno Original')
plt.axis('off')
plt.subplot(2, 2, 4)
plt.imshow(np.uint8(np.absolute(aluno_laplaciano)), cmap='gray')
plt.title('Aluno - Laplaciano')
plt.axis('off')
plt.tight_layout()
plt.show()

# 5
plt.figure(figsize=(12, 6))
plt.subplot(2, 2, 1)
plt.imshow(img_lena, cmap='gray')
plt.title('Lena Original')
plt.axis('off')
plt.subplot(2, 2, 2)
plt.imshow(lena_roberts, cmap='gray')
plt.title('Lena - Roberts')
plt.axis('off')
plt.subplot(2, 2, 3)
plt.imshow(img_aluno, cmap='gray')
plt.title('Aluno Original')
plt.axis('off')
plt.subplot(2, 2, 4)
plt.imshow(aluno_roberts, cmap='gray')
plt.title('Aluno - Roberts')
plt.axis('off')
plt.tight_layout()
plt.show()

# 6
plt.figure(figsize=(12, 6))
plt.subplot(2, 2, 1)
plt.imshow(img_lena, cmap='gray')
plt.title('Lena Original')
plt.axis('off')
plt.subplot(2, 2, 2)
plt.imshow(lena_prewitt, cmap='gray')
plt.title('Lena - Prewitt')
plt.axis('off')
plt.subplot(2, 2, 3)
plt.imshow(img_aluno, cmap='gray')
plt.title('Aluno Original')
plt.axis('off')
plt.subplot(2, 2, 4)
plt.imshow(aluno_prewitt, cmap='gray')
plt.title('Aluno - Prewitt')
plt.axis('off')
plt.tight_layout()
plt.show()

# 7
plt.figure(figsize=(12, 6))
plt.subplot(2, 2, 1)
plt.imshow(img_lena, cmap='gray')
plt.title('Lena Original')
plt.axis('off')
plt.subplot(2, 2, 2)
plt.imshow(lena_sobel, cmap='gray')
plt.title('Lena - Sobel')
plt.axis('off')
plt.subplot(2, 2, 3)
plt.imshow(img_aluno, cmap='gray')
plt.title('Aluno Original')
plt.axis('off')
plt.subplot(2, 2, 4)
plt.imshow(aluno_sobel, cmap='gray')
plt.title('Aluno - Sobel')
plt.axis('off')
```

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