

UNIVERSIDADE DO ESTADO DE SANTA CATARINA

UDESC

NICOLLE BEATRICE ASQUINO

TRABALHO 2 – PIM

APLICAÇÃO DO OPERADOR GRADIENTE NA DETECÇÃO DE BORDAS

JOINVILLE

2025

1. INTRODUÇÃO

A detecção de bordas é uma etapa fundamental no processamento de imagens, permitindo identificar regiões de transição brusca de intensidade que frequentemente representam contornos de objetos. Neste trabalho, utilizamos operadores de gradiente para detectar bordas em imagens em tons de cinza. O gradiente é calculado a partir das derivadas direcionais horizontais (G_x) e verticais (G_y), e fornece informações tanto sobre a magnitude quanto sobre a direção da borda.

2. OBJETIVOS

- Aplicar filtros derivativos (Prewitt e Scharr) para calcular o gradiente de imagens.
- Suavizar imagens com filtro Gaussiano para reduzir ruídos.
- Calcular a magnitude e direção do gradiente.
- Aplicar a supressão dos não-máximos para afinar bordas.
- Testar diferentes valores do parâmetro K na supressão.
- Comparar os resultados das máscaras utilizadas.
- Analisar os resultados obtidos para diferentes imagens.

3. METODOLOGIA

3.1 Pré-processamento

As imagens fornecidas foram convertidas para tons de cinza, e um filtro Gaussiano com kernel 3x3 foi aplicado para redução de ruídos. Observe a imagem da Lua, para observação da suavização.

Imagem Original- Lua



Imagem 1: Lua original

Imagem Suavizada - Lua



Imagem 2: Lua suavizada

3.2 Máscaras de Gradiente

Foram utilizadas duas máscaras derivativas:

Máscara de Prewitt:

```
def aplicarPrewitt(imagem):  
    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]])  
  
    Gx = cv2.filter2D(imagem, -1, prewitt_x)  
    Gy = cv2.filter2D(imagem, -1, prewitt_y)
```

Imagem 3: Máscara Prewitt

Máscara de Scharr:

```
def aplicarScharr(imagem, nome='Imagem'):  
    scharr_x = np.array([[ -3,  0,  3],  
                        [-10,  0, 10],  
                        [ -3,  0,  3]])  
  
    scharr_y = np.array([[ -3, -10, -3],  
                        [  0,  0,  0],  
                        [  3, 10,  3]])  
  
    Gx = cv2.filter2D(imagem, -1, scharr_x)  
    Gy = cv2.filter2D(imagem, -1, scharr_y)
```

Imagem 4: Máscara Scharr

3.3 Cálculo da Magnitude e Direção

A magnitude foi calculada por: $M(i, j) = \sqrt{(Gx^2 + Gy^2)}$

```
magnitude = np.sqrt(Gx**2 + Gy**2).astype(np.uint8)
```

Imagem 5: Cálculo da magnitude

A direção foi calculada por: $D(i, j) = \arctan(Gy / (Gx + \epsilon))$

```
def calcularDirecao(Gx, Gy):  
    direcao = np.arctan2(Gy, Gx + 1e-8)  
    direcao = np.degrees(direcao)  
    direcao[direcao < 0] += 180  
    return direcao
```

Imagem 6: Cálculo da direção

4. SUPRESSÃO DE NÃO-MÁXIMOS E PARÂMETRO K

A supressão dos não-máximos visa manter apenas os pixels com magnitude local máxima na direção do gradiente. Foi adicionado o parâmetro K, que atua como um multiplicador de exigência:

- $K = 1.0$: padrão
- $K > 1.0$: torna a detecção mais seletiva

Foi testado os valores: $K = 1.0$, 1.2 e 1.5.

5. RESULTADOS

As imagens foram aplicadas nas imagens: moedas.png, Lua1_gray.jpg, chessboard_inv.png e img2.jpg.

5.1. MAGNITUDE PREWITT E SCHARR

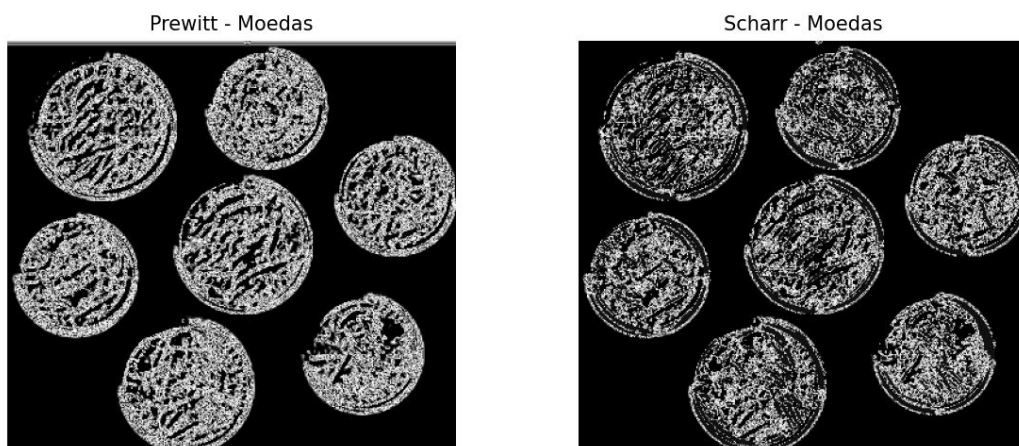


Imagem 7: Método Prewitt e Scharr para imagem Moedas

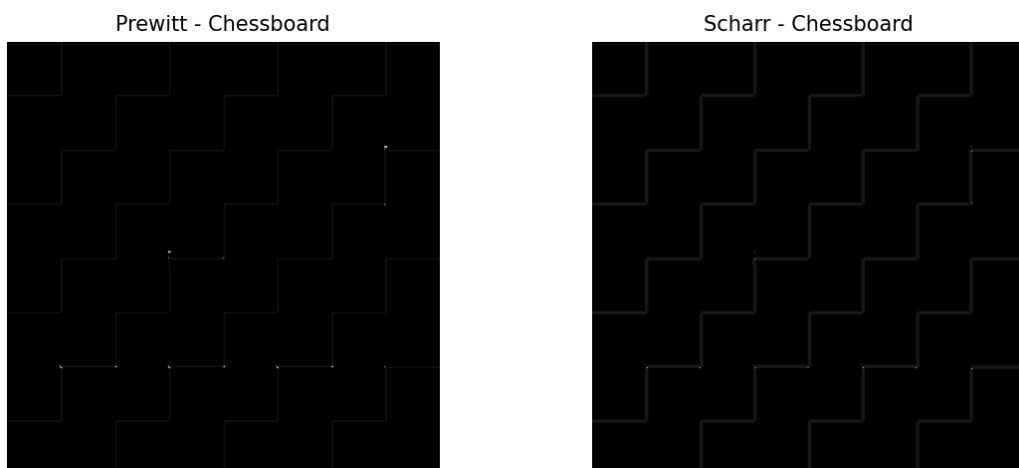


Imagem 8: Método Prewitt e Scharr para imagem Xadrez

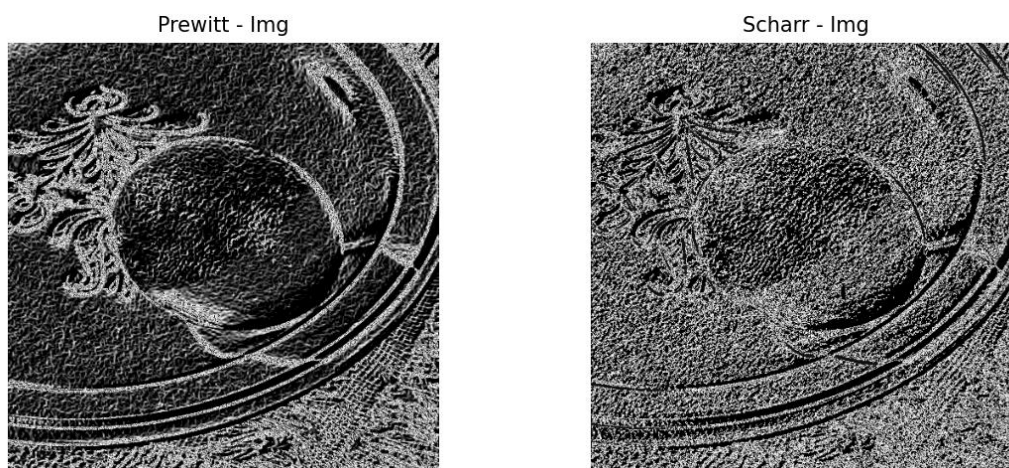


Imagem 9: Método Prewitt e Scharr para imagem Img2

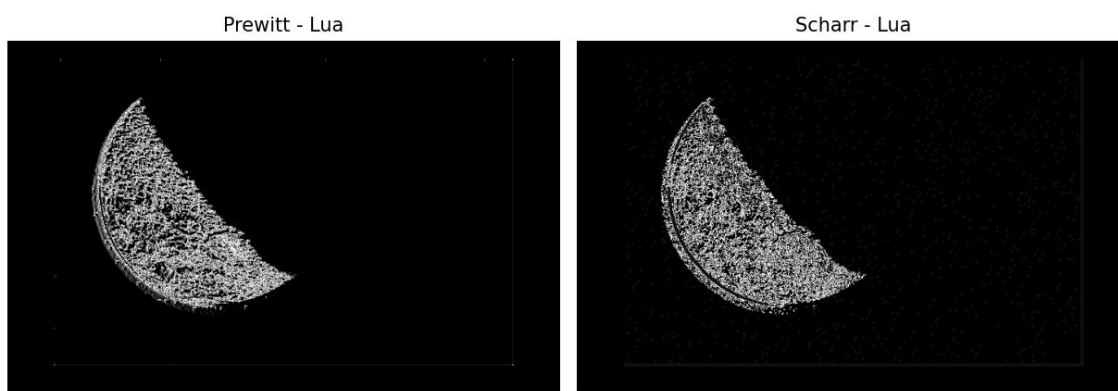


Imagem 10: Método Prewitt e Scharr para imagem Lua

5.3 SUPRESSÃO COM DIFERENTES VALORES DE K

5.3.1 Prewitt

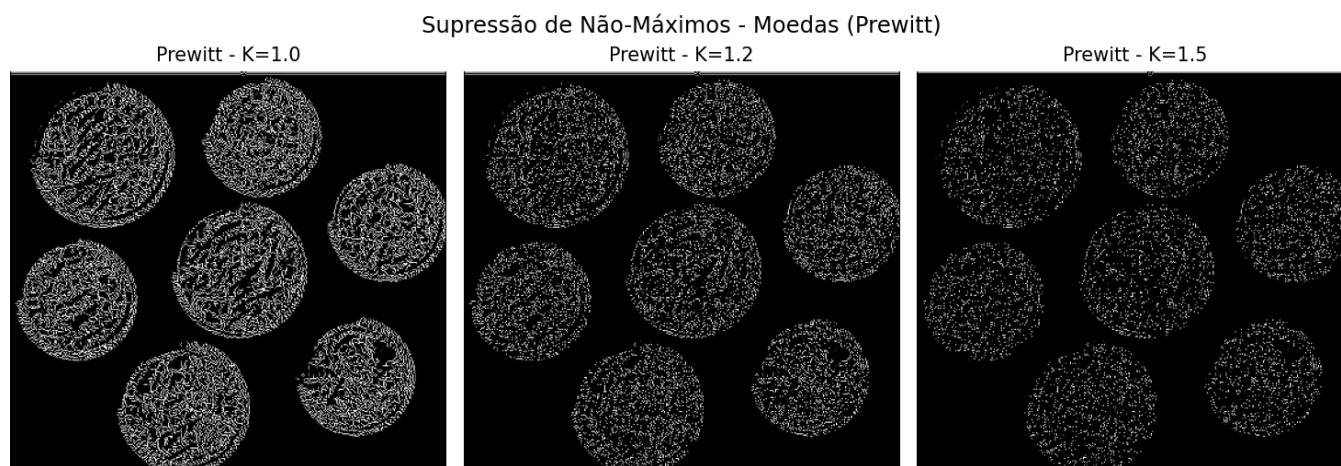
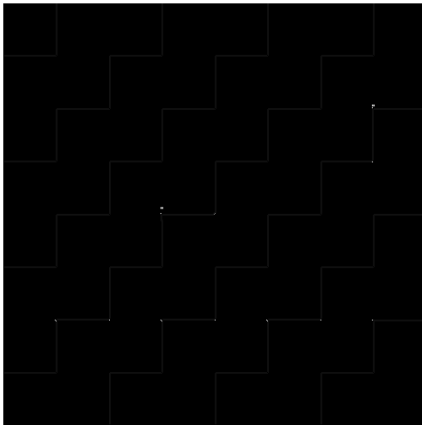


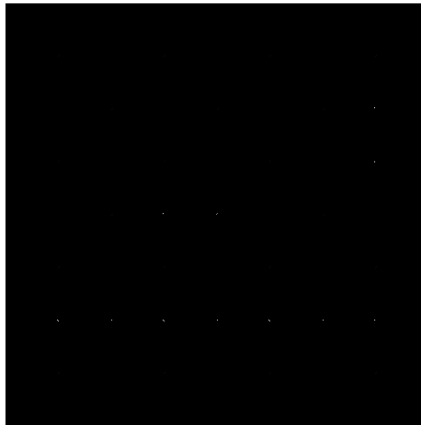
Imagem 11: Supressão para o método Prewitt na imagem Moedas

Supressão de Não-Máximos - Chessboard (Prewitt)

Prewitt - K=1.0



Prewitt - K=1.2



Prewitt - K=1.5

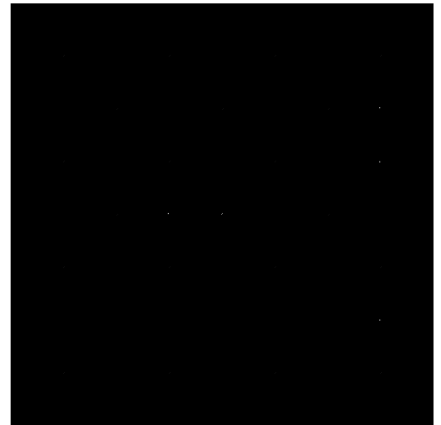
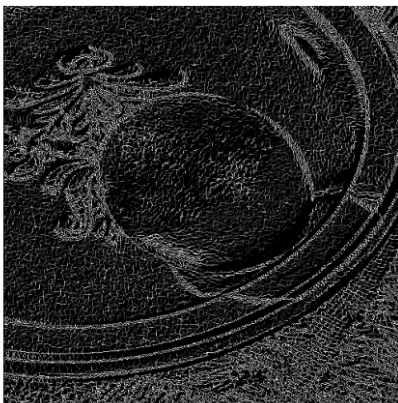


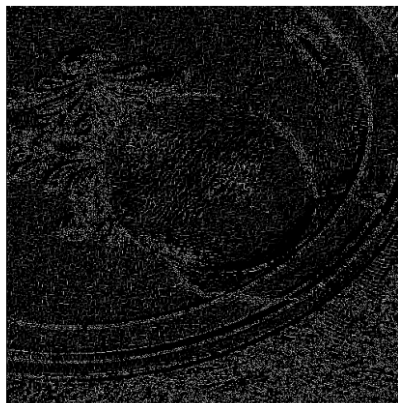
Imagem 12: Supressão para o método Prewitt na imagem Xadrez

Supressão de Não-Máximos - Img (Prewitt)

Prewitt - K=1.0



Prewitt - K=1.2



Prewitt - K=1.5

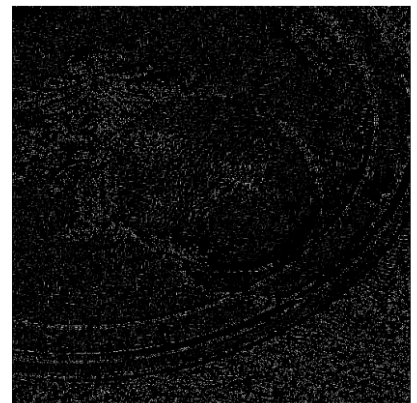
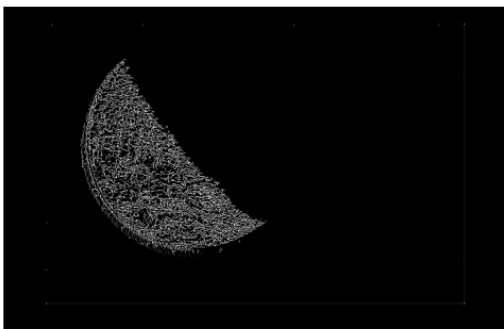


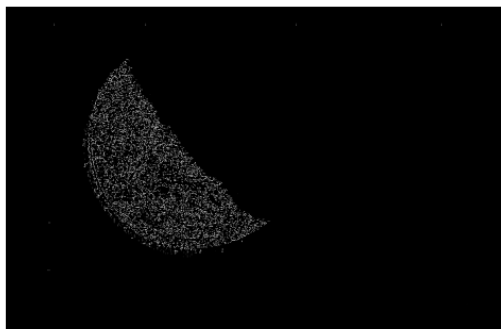
Imagem 13: Supressão para o método Prewitt na imagem Img2

Supressão de Não-Máximos - Lua (Prewitt)

Prewitt - K=1.0



Prewitt - K=1.2



Prewitt - K=1.5

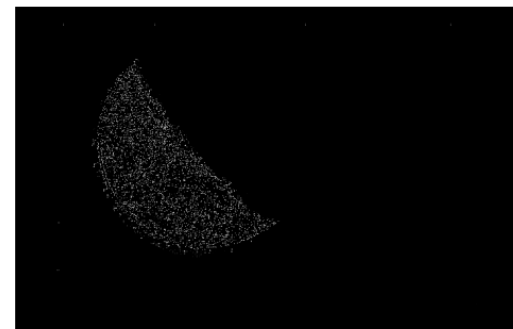


Imagem 14: Supressão para o método Prewitt na imagem Lua

5.3.1 Scharr

Supressão de Não-Máximos - Moedas (Scharr)

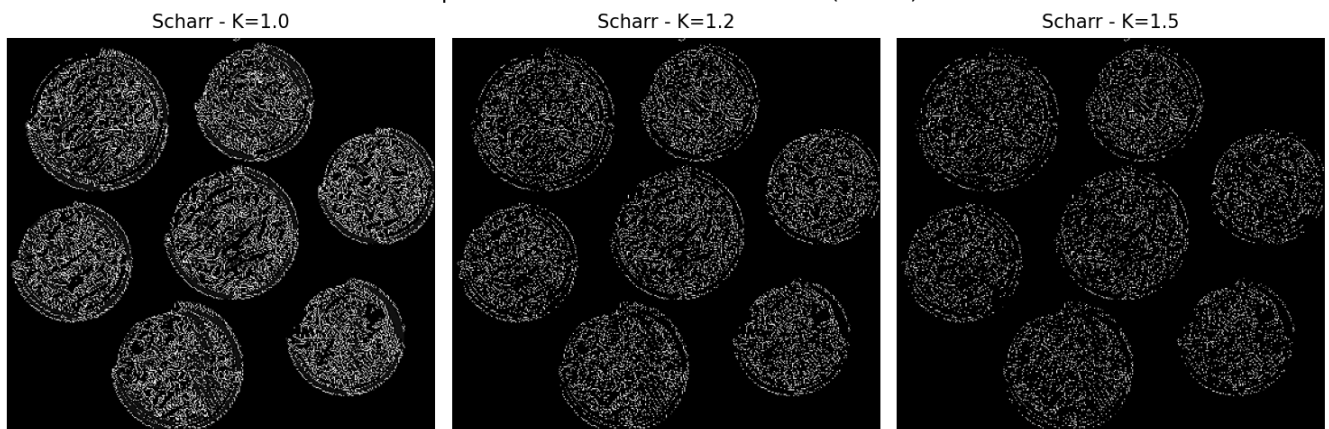


Imagem 15: Supressão para o método Scharr na imagem Moedas

Supressão de Não-Máximos - Chessboard (Scharr)

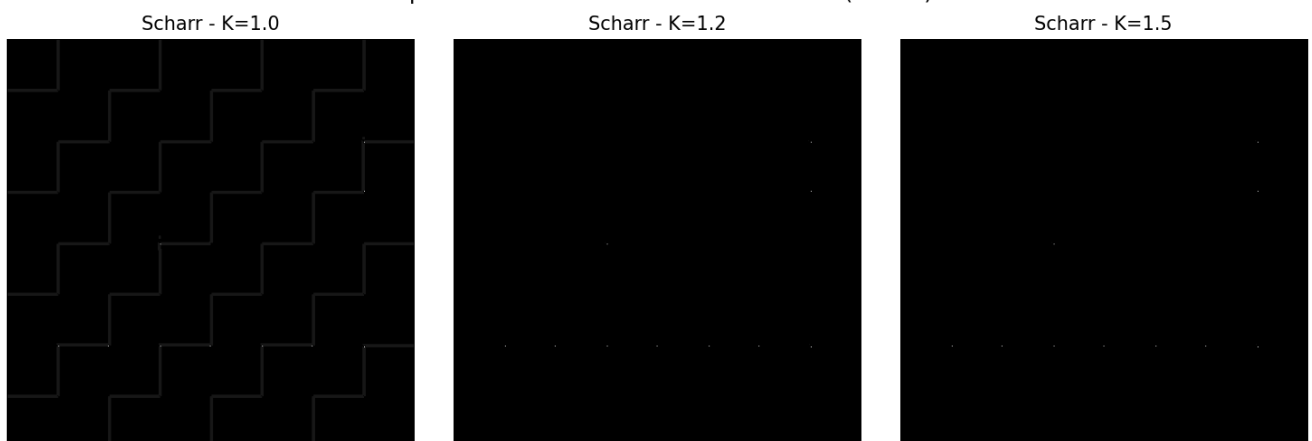


Imagem 16: Supressão para o método Scharr na imagem Xadrez

Supressão de Não-Máximos - Img (Scharr)

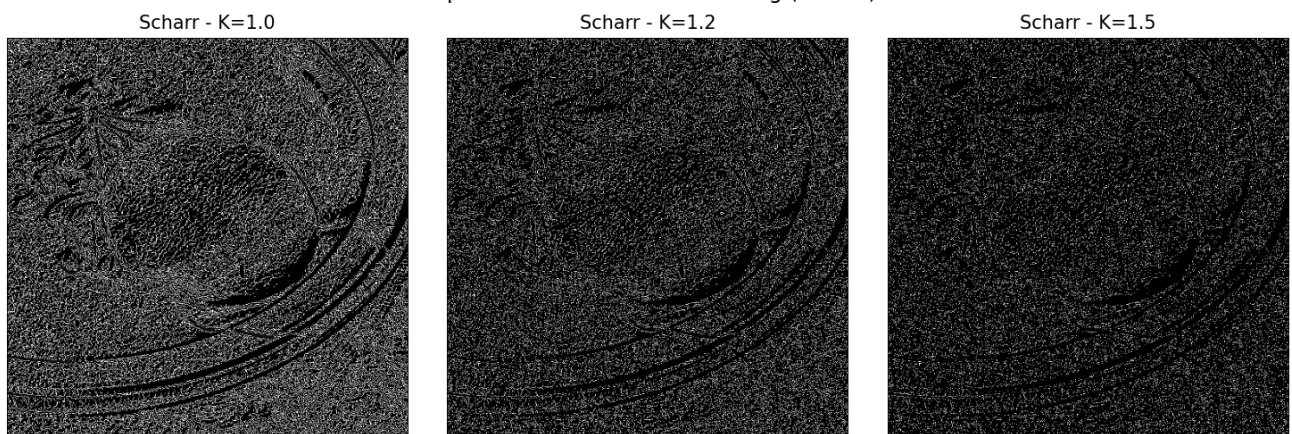


Imagem 17: Supressão para o método Scharr na imagem Img2

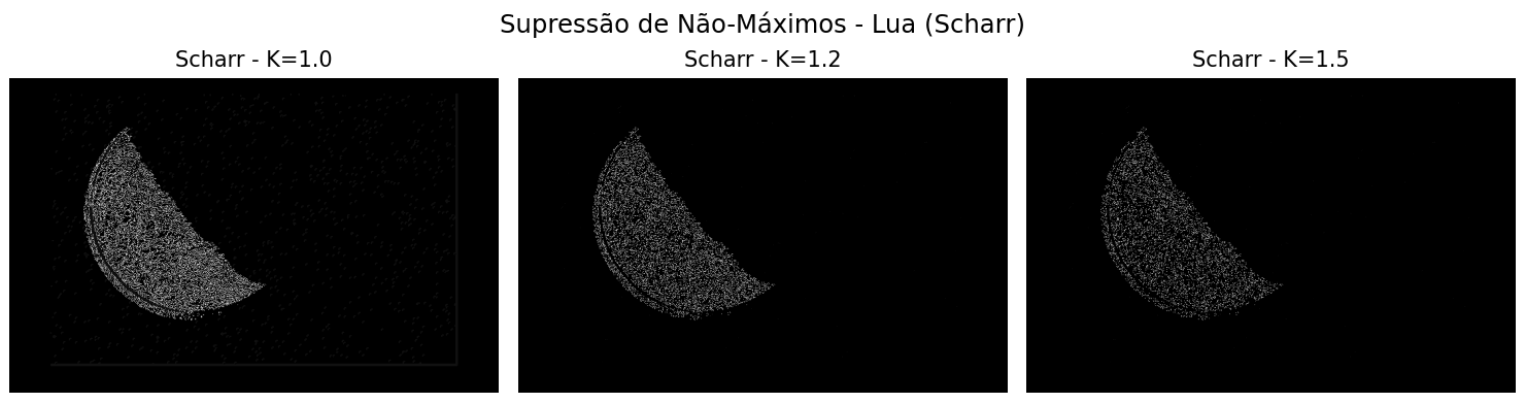


Imagem 18: Supressão para o método Scharr na imagem Lua

6. ANÁLISE COMPARATIVA

- Prewitt apresentou resultados satisfatórios, mas mais suscetível ao ruído.
- Scharr destacou melhor as bordas suaves e foi mais robusto.
- A variação de K influenciou diretamente na espessura e presença das bordas:
 - $K = 1.0 \rightarrow$ mais detalhes, mas com ruído
 - $K = 1.5 \rightarrow$ bordas mais finas e seletivas

7. CONCLUSÃO

O trabalho demonstrou com sucesso a aplicação do operador gradiente na detecção de bordas. A comparação entre Prewitt e Scharr mostrou que a escolha da máscara influencia a qualidade dos resultados. O parâmetro K se mostrou útil para controlar a seletividade na detecção de bordas.

Os resultados demonstram que a escolha da máscara pode depender do tipo de imagem, Scharr é mais indicada em contextos com detalhes suaves, enquanto Prewitt pode ser suficiente em imagens com alto contraste.