Universidade Federal do Piauí
Campus Senador Helvídio Nunes de Barros - Picos PI
Disciplina: Tópicos em Visão Computacional
Professor: Romuere Rodrigues Veloso e Silva
Aluno: Mateus Pinto Garcia

Trabalho prático de aplicação de filtros espaciais com convolução.

Filtros utilizados

Remoção de ruídos: mean, media ponderada, gaussiano
Detecção de bordas: laplaciano, sobel

Resultados:

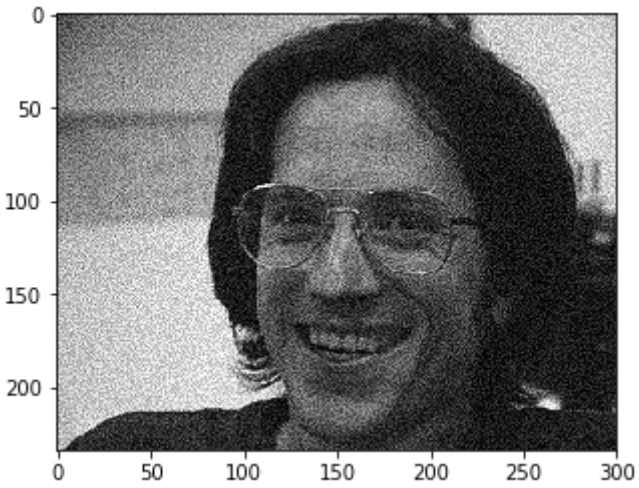

Figura 1: Imagem original

Resultado da convolução dos filtros de denoise:

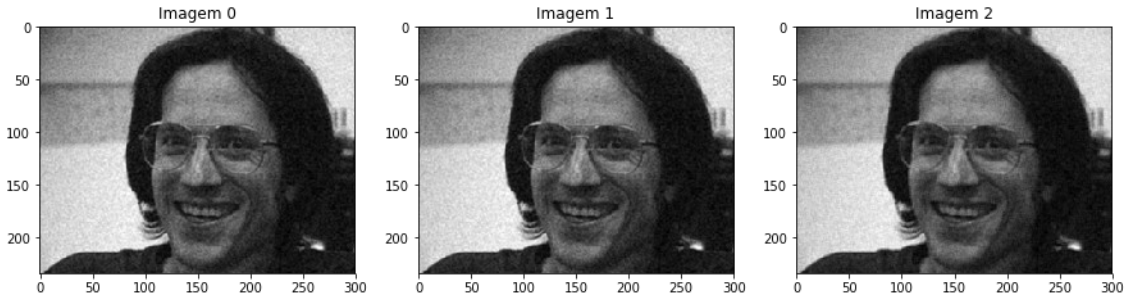

Figura 2: Filtros: Gaussiano, Média ponderada e Média, receptivamente.

Resultado da convolução dos filtros de detecção de borda

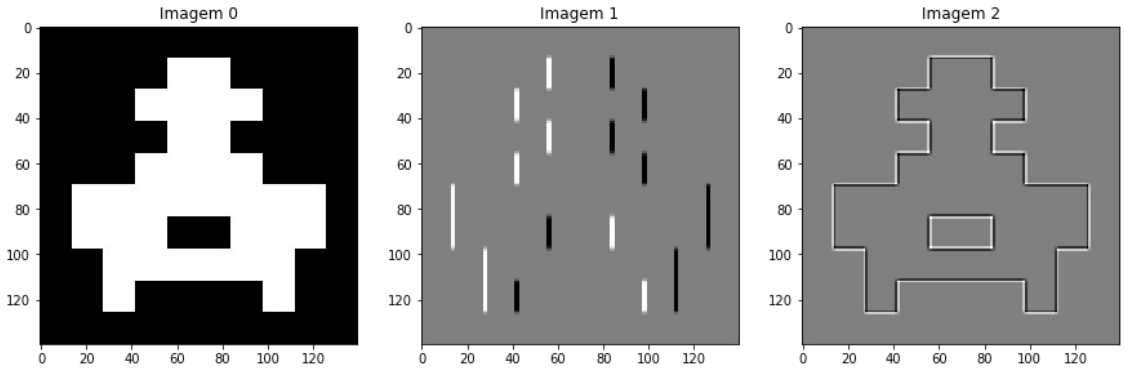

Figura 3: Imagem Original; Filtros: Sobel e Laplaciano

Para os filtros de denoise, média, média ponderada e gaussiano é nítido diferença entre a imagem original em comparação aos resultados da aplicação dos filtros, é possível ver que o nível de ruído diminuiu bastante para cada filtro aplicado.

Já com as bordas é notável a diferença os filtros laplaciano e sobel, é nítido a performance do filtro laplaciano para a detecção de borda, onde há a mudança abrupta de cor, como é o caso da imagem acima.

Para aplicação dos filtros foi utilizada a função **convolution(img, filtro, borda)**, que retorna a uma imagem com cada pixel calculado com base no tamanho e valores de uma mascara de entrada.

Função:

```
def convolution(img, filtro, borda = 'ignore'):

    img_out, x_origin, y_origin =  resize_matrix(img, filtro, borda)
    matrix_img_out = np.zeros(img.shape)

    for x in range(img.shape[0]):
        for y in range(img.shape[1]):
            try: #Try Except para ignorar tratamento de bordas
                matrix_img_out[x, y] = sum(
                    sum(filtro * img_out[x: x + x_origin, y: y + y_origin])
                )
            except: continue
    return matrix_img_out
```

Figura 4: Algoritmo - Convolução de filtros

Onde é feito o somatório da multiplicação de uma matriz filtro por uma janela de pixels de recorte, o resultado é posto no centro do recorte na imagem resultante.

Exemplo de Mascara:

"sobel": np.array(([-1, 0, 1], [-2, 0, 2], [-1, 0, 1]))

Filtro Laplaciano

O filtro laplaciano, tem a capacidade de realçar as bordas e detalhes em objetos em uma imagem. Como antes visto, este filtro é ótimo para o realce. Exemplo:

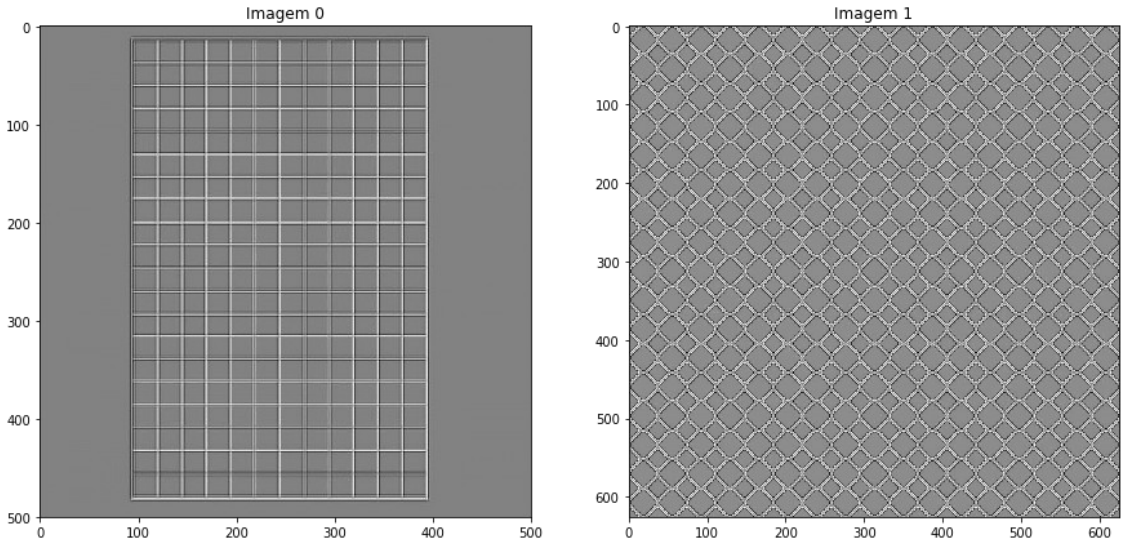

Figura 5: Aplicação do filtro laplaciano

Máscara de nitidez e uma filtragem high-boost

k = 0.4

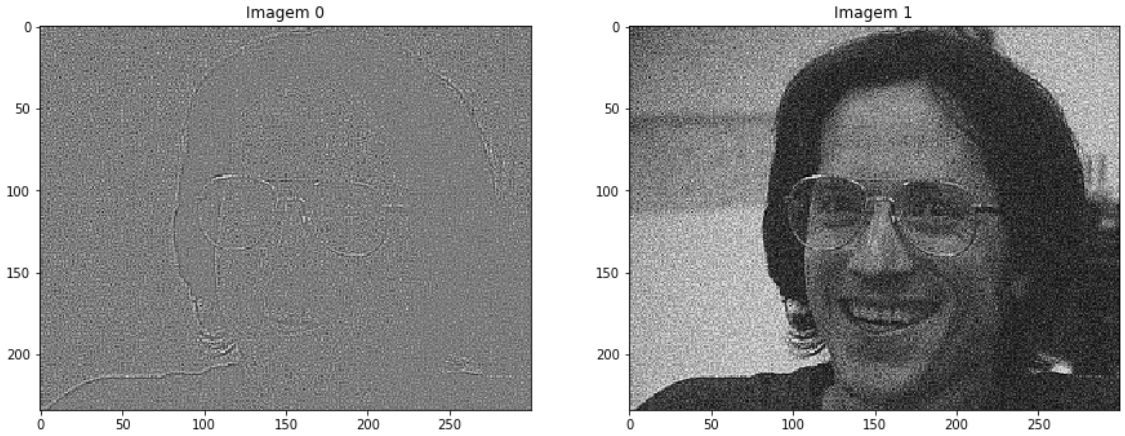

Figura 6: Mascara e high-boost

k = 1

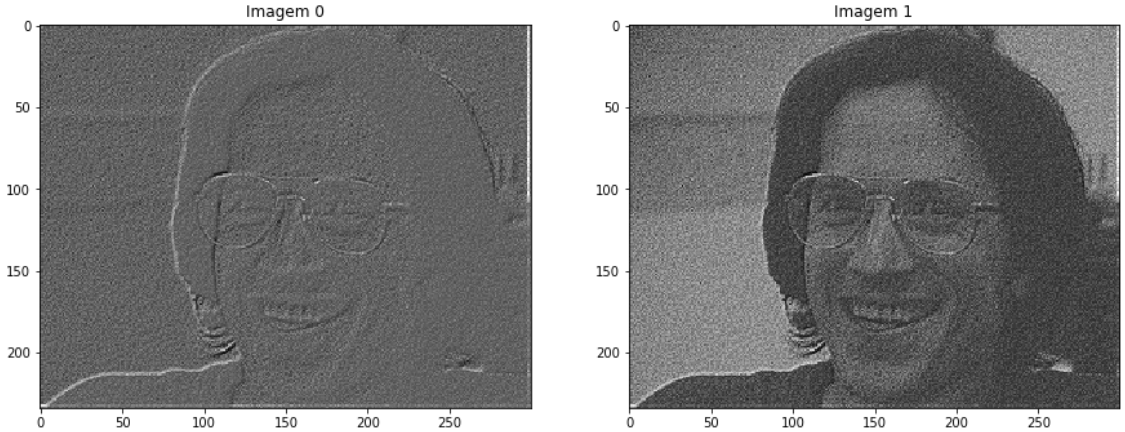

Figura 7: Mascara e high-boost

k = 5

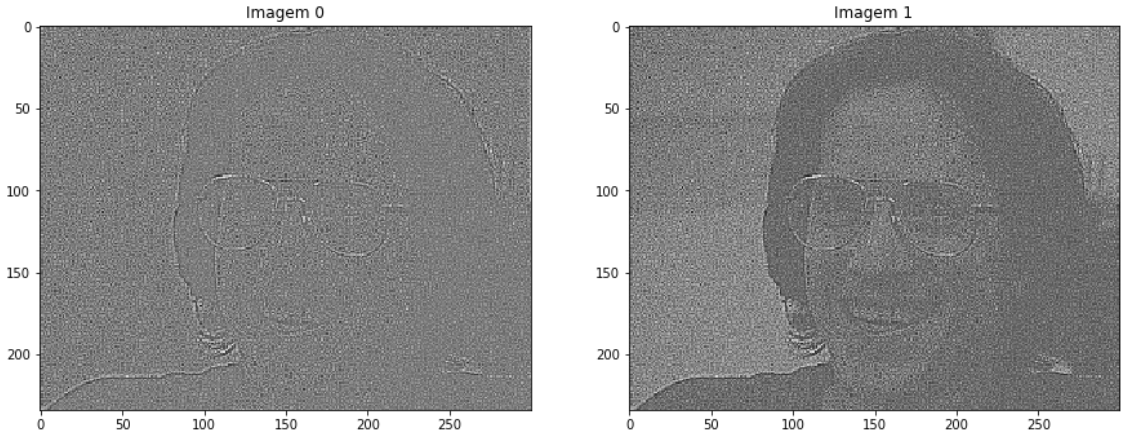

Figura 8: Mascara e high-boost

A mascara de nitidez é o resultado somente da diferença de uma imagem pela convolução dela por algum filtro, ou seja o processo: borrar a imagem original, subtrair a imagem borrada da imagem original obtendo a mascara. Somar a máscara à imagem original se obtem o high-boost.