

Trabalho 2 – Processamento Digital de Sinais

Semestre 1/2020

Professor: Ricardo de Queiroz

Aluno: Lindeberg Pessoa Leite

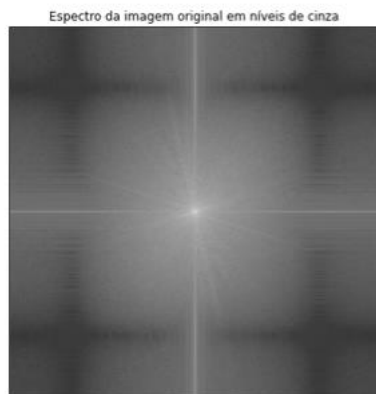
Para este trabalho, utiliza-se a foto pessoal do aluno (foto_lindeberg.jpeg) com resolução de 1024x1024, que segue junto com este documento. A foto é convertida para níveis de cinza, conforme a composição $Y = 0.30 R + 0.55 G + 0.15 B$. Ademais, para cada imagem (original ou resultado), sempre se mostra a imagem em si e seu espectro (centrado em $\log|z|$). Segue também juntamente a este documento o código fonte comentado (Trabalho2.ipynb), bem como uma versão gerada em html para rápida visualização (Trabalho2.html).

Passamos agora a responder os quesitos do trabalho.

1. Borrar a imagem por convolução (3 filtros diferentes)

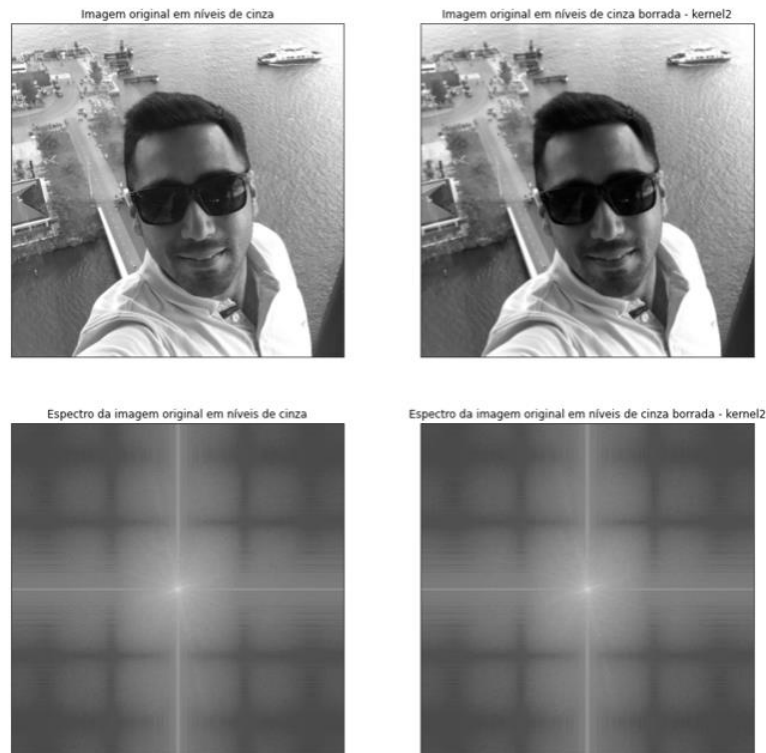
São utilizados os seguintes filtros: $\text{kernel1} = \text{np.ones}((3, 3), \text{np.float32}) / 9$; $\text{kernel2} = \text{np.ones}((5, 5), \text{np.float32}) / 25$; e $\text{kernel3} = \text{np.ones}((9, 9), \text{np.float32}) / 81$

Imagem borrada com kernel1



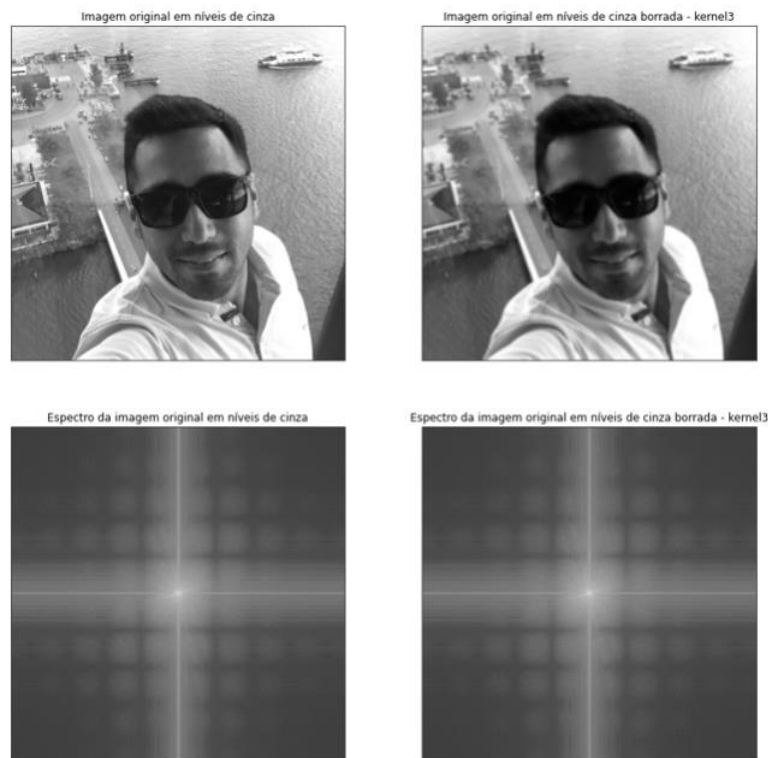
Note que para o kernel1, o borrado fica suave, sendo até difícil identificar a diferença da imagem original.

Imagem borrada com kernel2



Observe que para o kernel2 o borrado começa a ficar mais visível, já sendo possível identificar mais claramente quando comparado com a aplicação do kernel1.

Imagem borrada com kernel3



Já no kernel3, percebe-se claramente a imagem borrada. Resultado já esperado considerando o tamanho do kernel3.

2. Borrar a imagem no domínio de Fourier (filtros iguais ao da convolução).

São usados os mesmos filtros do item anterior

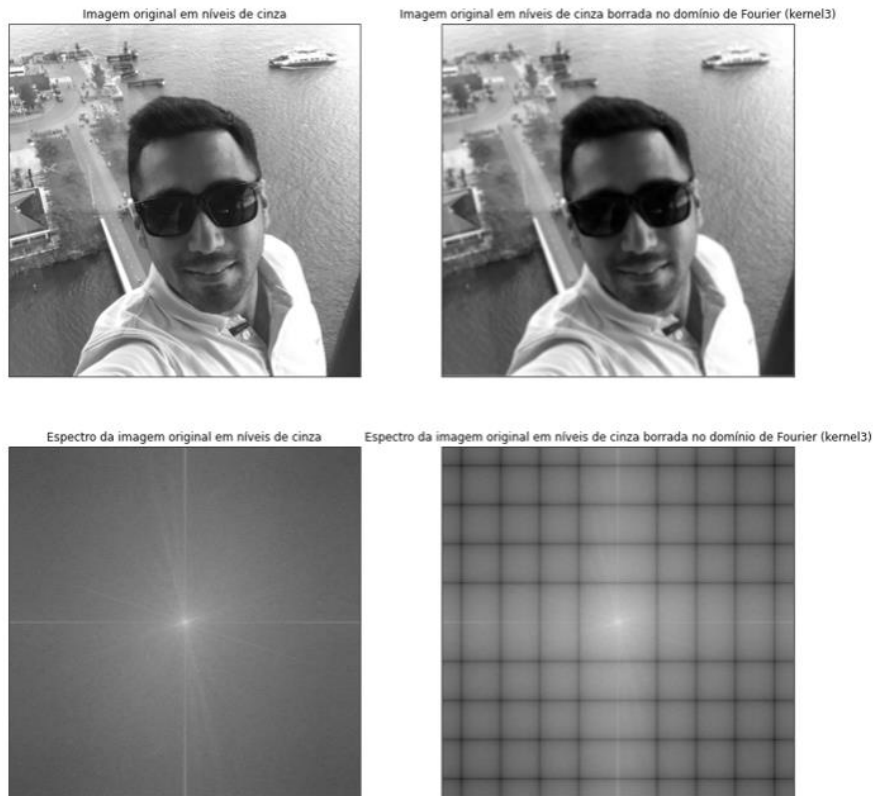
Imagem borrada no domínio de Fourier com kernel1



Imagem borrada no domínio de Fourier com kernel2

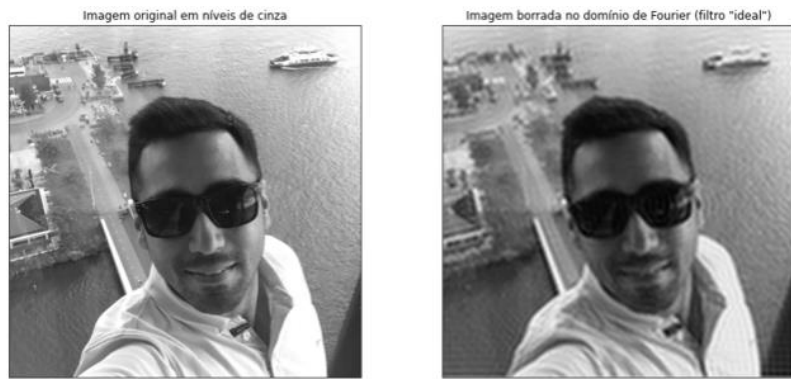


Imagem borrada no domínio de Fourier com kernel3

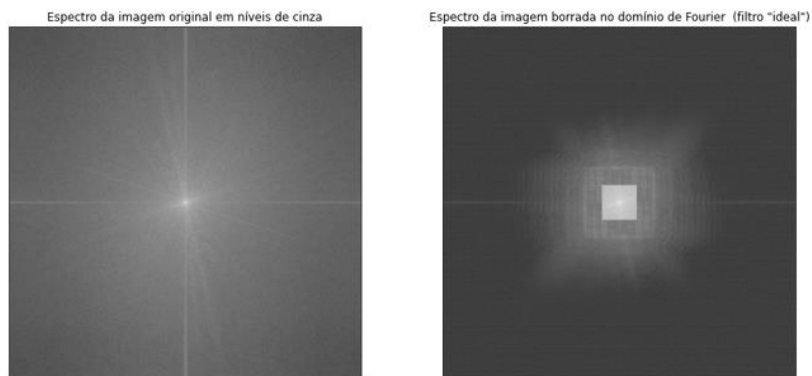


Verifique que o resultado, pelo menos visualmente, é bastante semelhante quando comparados com o método de convolução. Entretanto, note que os espectros são diferentes. Isso se deve ao fato do produto no domínio de Fourier corresponder a uma convolução circular no domínio do tempo e não a uma convolução simples, conforme aplicada no item anterior

3. Borrar a imagem no domínio de Fourier (filtro "ideal")

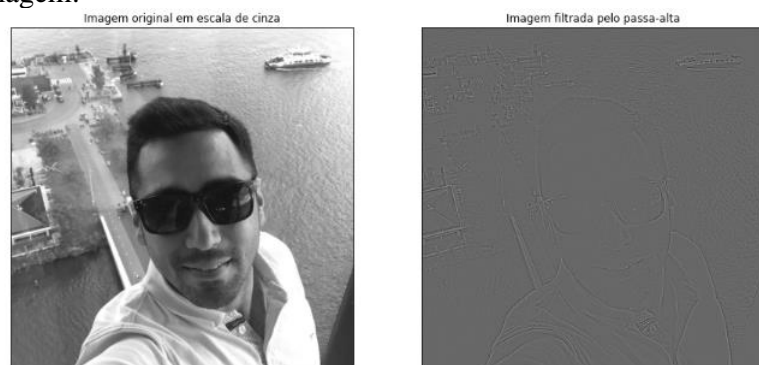


```
(Text(0.5, 1.0, 'Espectro da imagem borrada no domínio de Fourier (filtro "ideal")'),  
([]), ([]),  
([]), ([]))
```

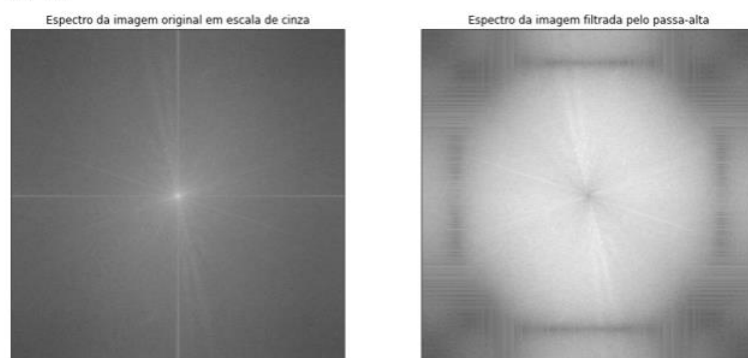


Para o filtro ideal, é criada uma máscara, no centro há uns e nas extremidades zeros. Essa máscara é multiplicada pela dft da imagem em níveis de cinza

4. Passa-alta da imagem.



```
(Text(0.5, 1.0, 'Espectro da imagem filtrada pelo passa-alta'),  
([]), ([]),  
([]), ([]))
```

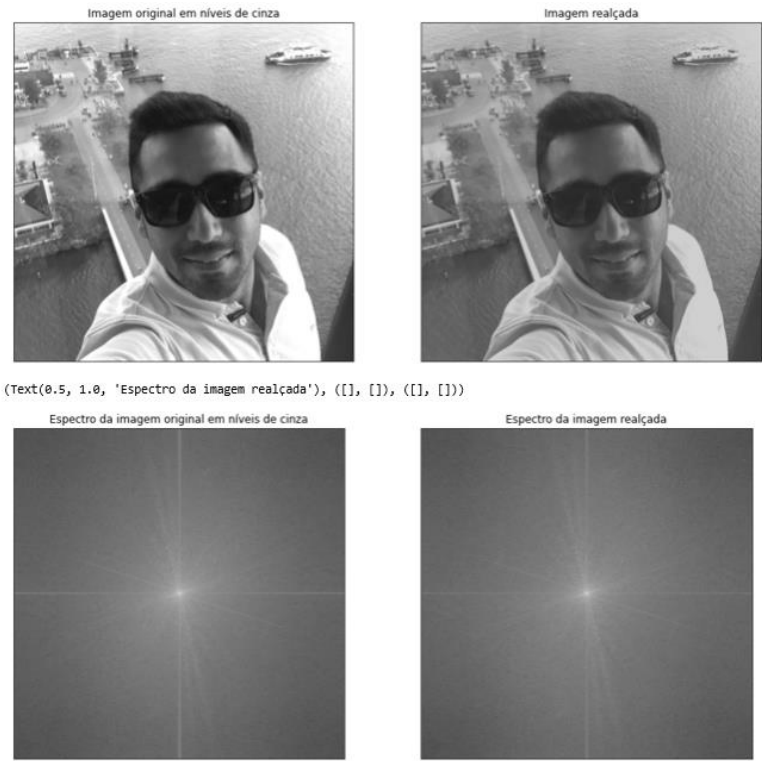


Neste item, vale destacar que a aplicação do filtro passa-alta realça as regiões de borda da imagem, conforme pode ser visualizado na imagem no canto superior direito.

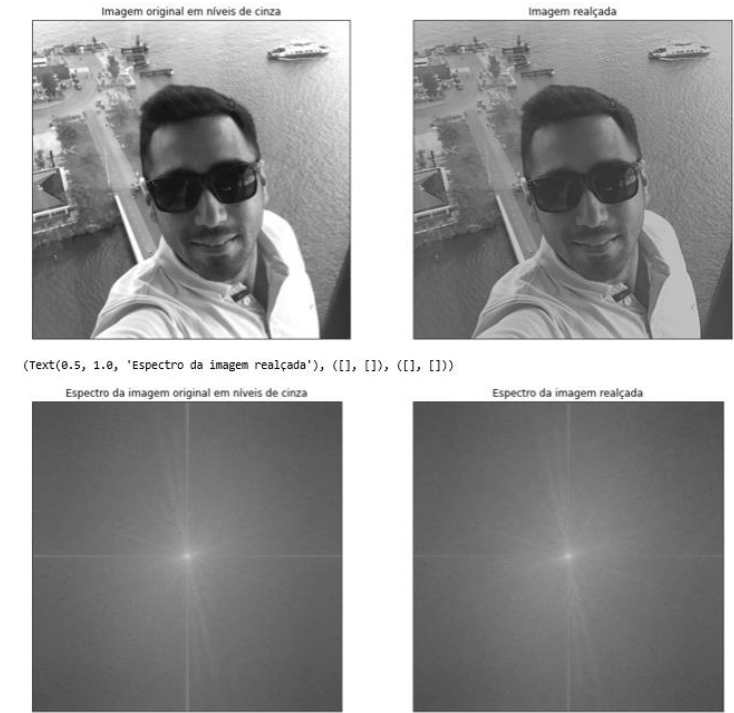
5. Realce de imagem usando o resíduo do passa-baixa (2 intensidades)

Realce de imagem com o kernel1 e kernel3

Kernel1



Kernel3



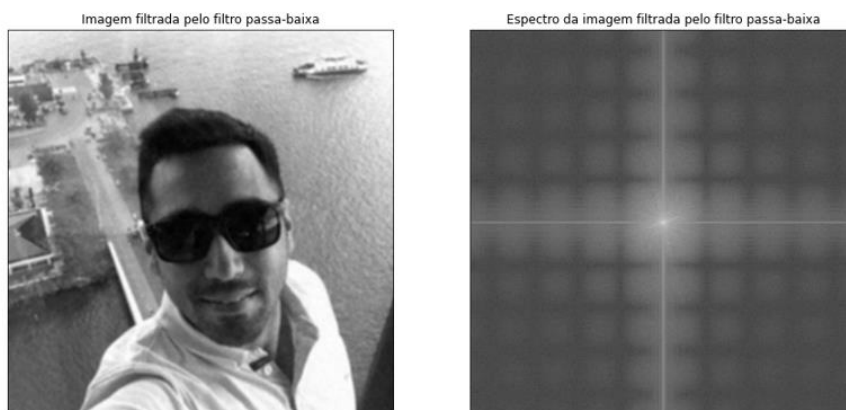
Neste item, para realce de imagem, são usados dois filtros passa-baixa (kernel1 e kernel3), cuja resultados podem ser visualizados nas imagens acima, que apresentam uma leve diferença de realce.

6. Filtragem de ruídos gaussianos usando passa baixas.

Inicialmente, vamos inserir um ruído gaussiano:



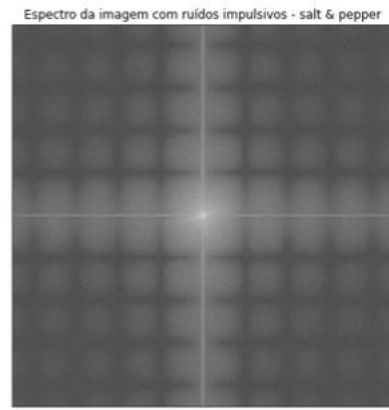
Agora, vamos filtrar usando passa-baixa (kernel3)



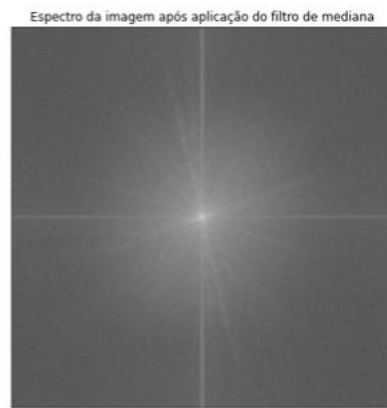
É possível visualizar que houve remoções de ruídos gaussianos após passar o filtro passa-baixa. Entretanto, o grande problema desse tipo de filtro é que a imagem resultante fica desfocada e acaba perdendo informações de detalhe.

7. Filtragem de ruídos impulsivos usando filtro de mediana

Inicialmente, vamos inserir ruídos impulsivos:



Agora, vamos filtrar usando filtro de mediana



Perceba que a filtragem utilizando filtro de mediana produz um resultado muito bom, sem desfocar a imagem.