MC920 Trabalho 1 - Filtragem de Imagens

182851 - Lucy Miyuki Miyagusiku Narita

17 de abril de 2019

1 Introdução

O objetivo deste trabalho é implementar alguns filtros nos domínios espacial e de frequências. Filtros permitem que sejam aplicados diversos efeitos em imagens, alterando os valores de intensidade dos pixels.

Juntamente desse relatório está sendo enviado um notebook *Jupyter* que contém o código e as imagens referentes ao projeto.

1.1 Filtragem no Domínio Espacial

O domínio espacial refere-se ao próprio plano da imagem, ou seja, ao conjunto de pixels que compõem uma imagem. [1]

O processo de filtragem neste domínio é normalmente feito através de matrizes denominadas *máscaras* que representam o peso do pixel em questão e de sua vizinhança. A máscara é então convolucionada com a imagem em que se queira aplicar o filtro.

1.2 Filtragem no Domínio de Frequência

Para trabalhar no domínio da frequência, aplicamos a transformada de Fourier na imagem. A expressão discretizada da transformada é dada por:

$$F[k] = \sum_{n=0}^{N-1} f[n] \exp\left(\frac{-i2\pi kn}{N}\right)$$

[2]

e sua inversa:

$$f[n] = \frac{1}{N}\sum_{k=0}^{N-1} F[k] \exp\left(\frac{i2\pi kn}{N}\right)$$

[2]

Para a aplicação do filtro, há uma relação de reciprocidade entre a convolução no domínio espacial e sua contrapartida no domínio da frequência:

$$\mathscr{F}(f(x,y)h(x,y)) \Longleftrightarrow \mathscr{F}(f(x,y)) * \mathscr{F}(h(x,y))$$

[2]

e

$$\mathscr{F}(f(x,y) * h(x,y)) \Longleftrightarrow \mathscr{F}(f(x,y))\mathscr{F}(h(x,y))$$

[2]

Portanto, executamos uma multiplicação termo a termo entre os elementos da matriz da transformada da imagem original com a matriz da transformada da máscara a ser aplicada.

2 Execução

O projeto foi desenvolvido em Python 2.7.15 e também foi testado com Python 3.6.7, utilizando os seguintes pacotes como dependências:

- Manipulação de dados
  - `numpy`
  - `opencv`
- Plotting
  - `matplotlib`
- Carregamento e download das imagens
  - `requests`
  - `PIL` (`Pillow Imaging Library`)

As imagens utilizadas se encontram na pasta `imgs/`. Por se tratar de um notebook, não há entrada de dados parametrizada.

3 Processo e Decisões Tomadas

3.1 O Processo de Convolução

A fórmula geral para o processo de convolução é dada por:

$$g(x,y) = \omega(x,y) * f(x,y) = \sum_{i=-a}^{a} \sum_{j=-b}^{b} \omega(i,j) f(x-i, y-j)$$

onde $g(x,y)$ é o resultado (a imagem filtrada), $f(x,y)$ é a entrada (a imagem original de dimensões (m,n)), ω é a matriz de convoluçao (máscara), $a = \lfloor \frac{m}{2} \rfloor$ e $b = \lfloor \frac{n}{2} \rfloor$

Para a aplicação das máscaras utilizamos a função `filter2D` do `openCV`, ao invés de criarmos nossa própria função de convolução, uma vez que estamos mais interessados na análise crítica do que na implementação da operação de convolução em si.

4 Resultados e Discussões

Os filtros foram aplicados à seguinte imagem:

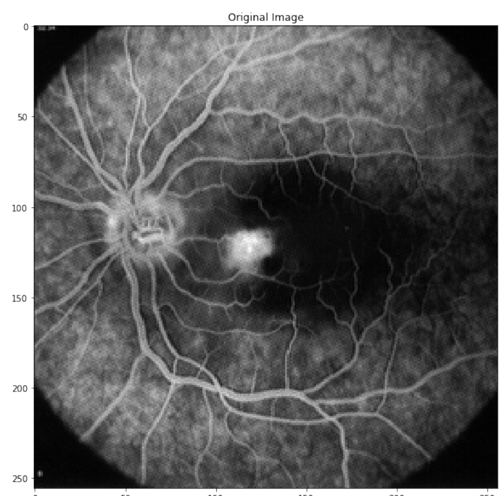

Figura 1: Imagem original

4.1 Filtragem no Domínio Espacial

4.1.1 Filtro h_1

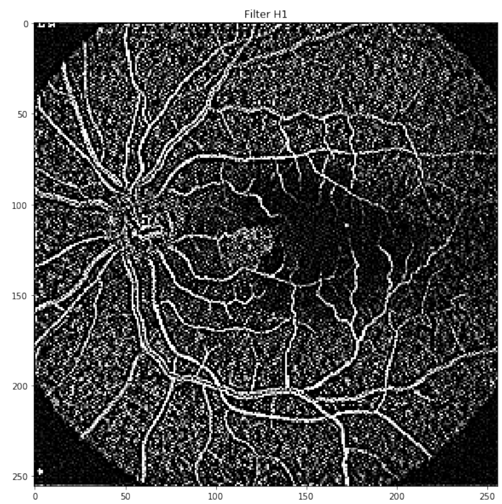

Figura 2: Aplicação do filtro h_1

O filtro h_1 é um filtro passa-alta. Por ser um filtro simples, com a soma dos pesos nula, ele realçará os pontos de alta frequência (**incluindo ruídos**), enquanto atenua as regiões com baixa variância.

4.1.2 Filtro h_2

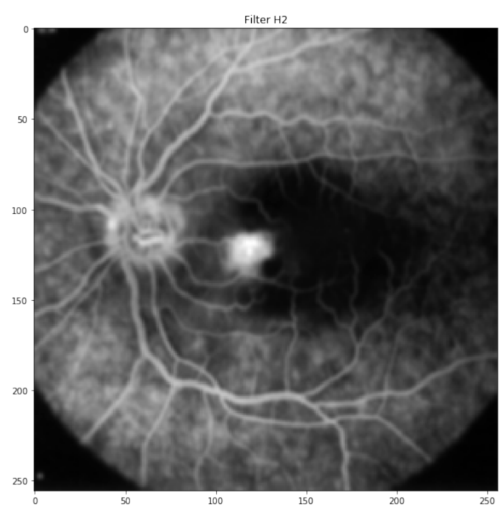

Figura 3: Aplicação do filtro h_2

O filtro h_2 é um filtro passa-baixa de suavização Gaussiano (5x5) com $\sigma = 1.0$ aproximado pela expansão binominal $(a+b)^n = \sum_{k=0}^{n} \frac{n!}{k!(n-k)!} a^{n-k} b^k$.

Média de tempo de aplicação do filtro em 5 amostras de mesmo tamanho:

```
CPU times: user 714 µs, sys: 500 µs, total: 1.21 ms
Wall time: 639 µs
```

O Filtro h_2 é separável e pode ser obtida a partir da convolução das seguintes máscaras unidimensionais:

$$h2' = \frac{1}{16}\begin{bmatrix}1\\4\\6\\4\\1\end{bmatrix}$$

e

$$h2'' = \frac{1}{16}\begin{bmatrix}1 & 4 & 6 & 4 & 1\end{bmatrix}$$

A separação da máscara é interessante pois permite uma aplicação mais rápida (uma vez que faz menos operações) em 2 passos.

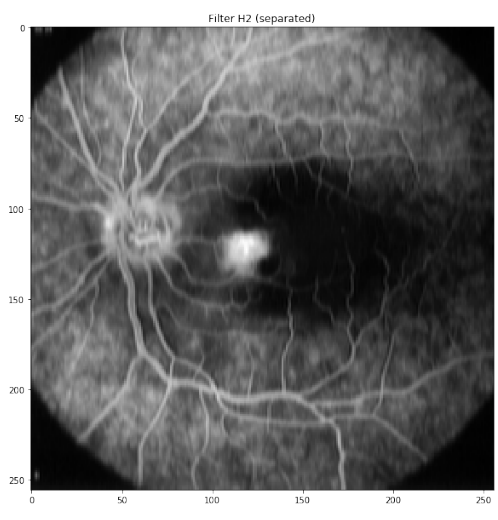

Figura 4: Aplicação dos filtros h_2' e h_2''

Média de tempo de aplicação das máscaras unidimensionais em 5 amostras de mesmo tamanho:

```
CPU times: user 133 µs, sys: 94 µs, total: 227 µs
Wall time: 235 µs
```

4.1.3 Filtros h_3 e h_4

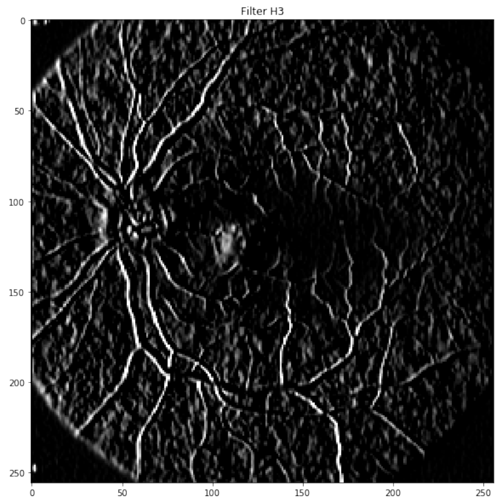

Figura 5: Aplicação do filtro h_3

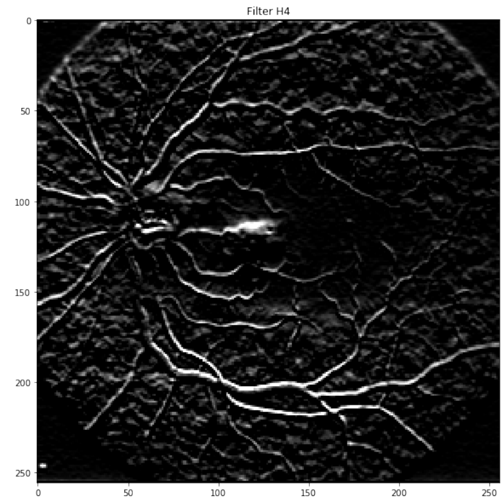

Figura 6: Aplicação do filtro h_4

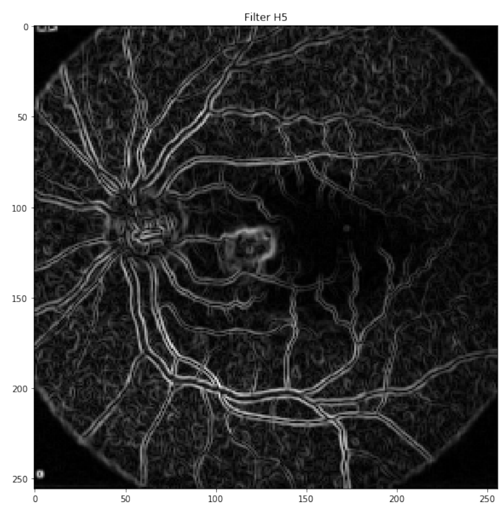

Figura 7: Aplicação dos filtros h_3 e h_4 combinados

Os filtro h_3 e h_4 são *Operadores de Sobel* ou *Filtros de Sobel* para detecção de bordas. O filtro h_3 detecta as bordas verticais enquanto o filtro h_4 as bordas horizontais. A combinação dos dois (h_5), portanto compõe as bordas **horizontais e verticais** da imagem.

4.2 Filtragem no Domínio de Frequência

Para as filtragem de frequência, deveríamos utilizar um filtro Gaussiano.

O filtro Gaussiano é dado pela seguinte função de transferência:

$$H(u,v) = \exp\left(-\frac{D^2(u,v)}{2D_0^2}\right)$$

em que D_0 é a frequência de corte e $D(u,v)$ é a distância do pixel (u,v) até a componente de frequência zero (componente de maior frequência).

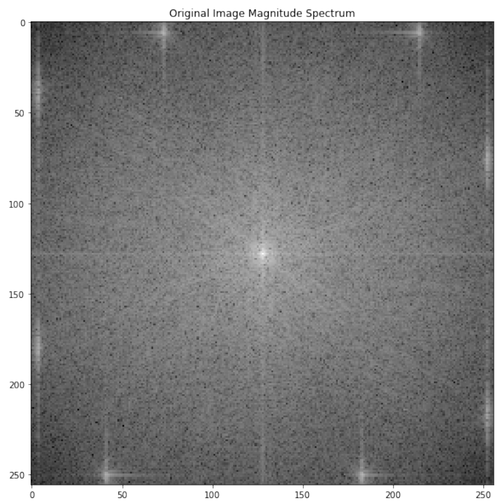

Figura 8: Imagem Original - Espectro de Frequência

A visualização do espectro de frequência da imagem original, sem nenhum filtro é vista na Figura 8. Como fizemos a translação da componente de frequência zero para o centro, as componentes de maior frequência, representados pelos pixels de maior intensidade, se encontrarão mais próximos ao centro.

Para esta filtragem, geramos 4 filtros com frequências de corte 10 (Figura 9), 20 (Figura 10), 40 (Figura 11) e 80 (Figura 12).

Como podemos observar, quanto **maior** o valor da frequência de corte, **menor** é a taxa de suavização da imagem, em comparação com o filtro Gaussiano no domínio espacial que, quanto **maior** o σ, **maior** a taxa de suavização.

5 Conclusão

Neste exercício pudemos aplicar e observar o efeito de alguns filtros utilizados no pré processamento de imagens, tanto no domínio espacial quanto no domínio da frequência.

Os resultados obtidos foram conforme o esperado:

- O realce das bordas com muito ruído do filtro h_1;

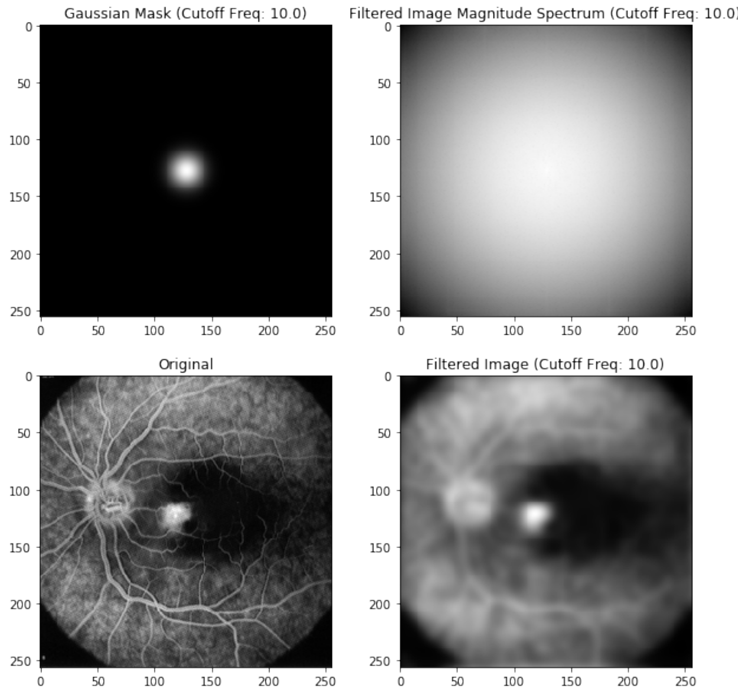

Figura 9: Filtro Gaussiano com Frequência de Corte 10

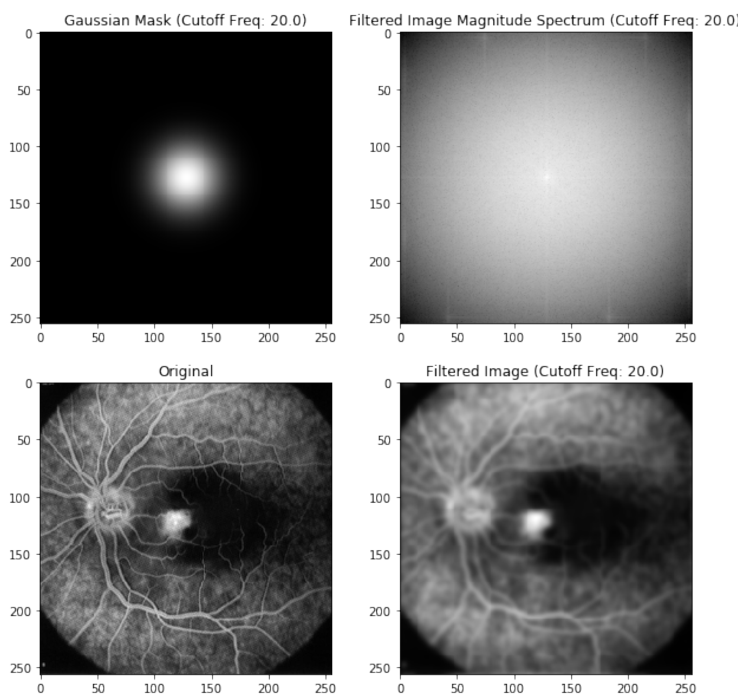

Figura 10: Filtro Gaussiano com Frequência de Corte 20

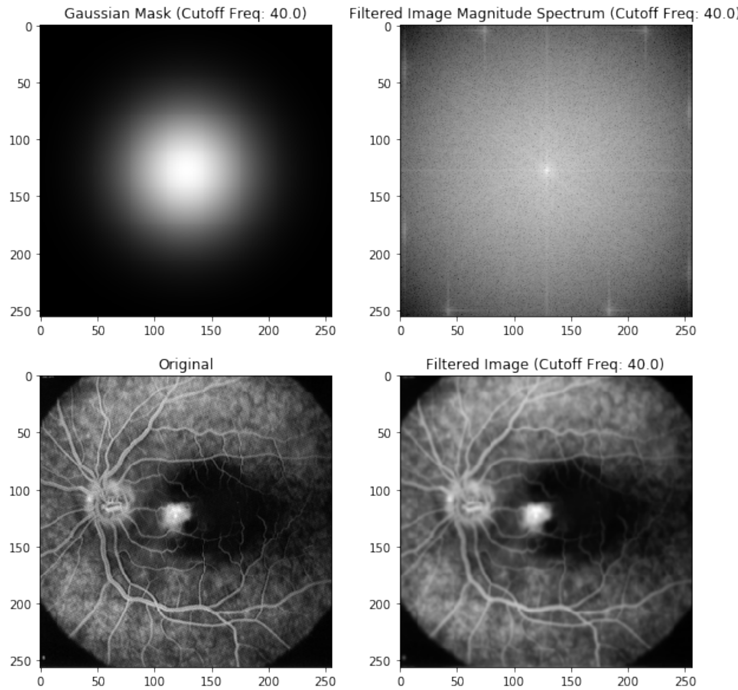

Figura 11: Filtro Gaussiano com Frequência de Corte 40

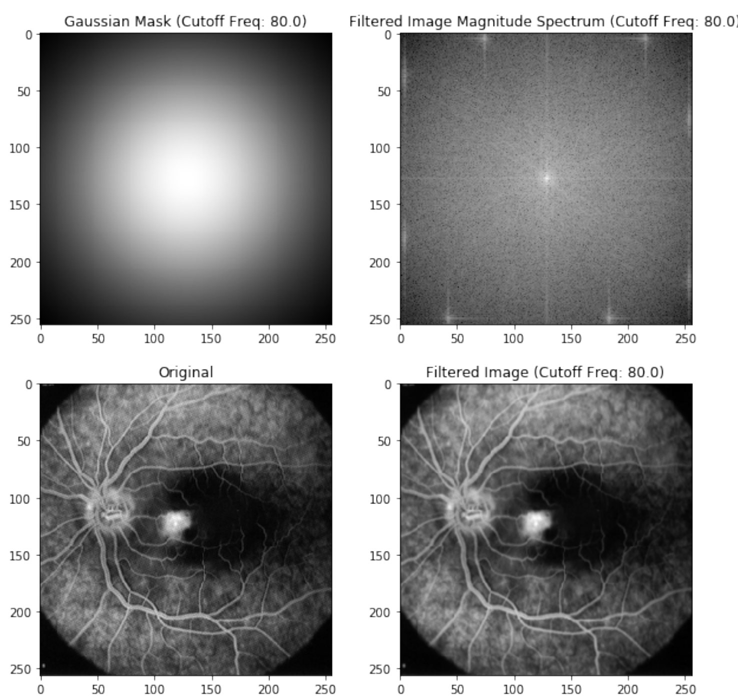

Figura 12: Filtro Gaussiano com Frequência de Corte 80

- A suavização da imagem com o filtro h_2;
- As bordas verticais, horizontais e o conjunto de bordas da imagem com os filtros h_3, h_4 e a combinação dos dois, respectivamente;
- A relação inversamente proporcional da frequência de corte com a taxa de suavização no filtro Gaussiano no domínio das frequências.

Referências

[1] Pedrini, H., *MC920 Introdução ao Processamento Digital de Imagens - Realce*, (Universidade Estadual de Campinas, UNICAMP. 2019). [Online] available at `http://www.ic.unicamp.br/ helio/disciplinas/MC920/aula_realce.pdf`

[2] Pedrini, H., *MC920 Introdução ao Processamento Digital de Imagens - Domínio de Frequência*, (Universidade Estadual de Campinas, UNICAMP. 2019). [Online] available at `http://www.ic.unicamp.br/ helio/disciplinas/MC920/aula_dominio_frequencia.pdf`