

MO443 - Processamento de Imagem Digital

Trabalho 02

Patrick de Carvalho Tavares Rezende Ferreira - 175480

24 de abril de 2020

1 Introdução

O objetivo deste trabalho é implementar alguns filtros de imagens no domínio espacial. A filtragem aplicada a uma imagem digital é uma operação local que altera os valores de intensidade dos pixels da imagem levando-se em conta tanto o valor do pixel em questão quanto valores de pixels vizinhos.

No processo de filtragem, utiliza-se uma operação de convolução de uma máscara pela imagem. Este processo equivale a percorrer toda a imagem alterando seus valores conforme os pesos da máscara e as intensidades da imagem.

2 Descrição do código elaborado

O código desenvolvido para este projeto possui uma função - denominada "convolui"(figura 1) - para a execução de todos os os filtros convolucionais pedidos no roteiro. O código contido no arquivo "main.py" foi desenvolvido para que o utilizador possa simplesmente importar todas as suas funções e utilizá-las independentemente. Para fins de demonstração, o código também possui um script dentro da função "main()" que é executado sempre que chamado como arquivo principal pelo interpretador python e cria um diretório "output" contendo as imagens solicitadas neste roteiro. Todas as funções com argumentos numéricos aceitam que estes sejam inteiros ou pontos flutuantes positivos, tratando de forma adequada e convertendo para o tipo "unit8" a matriz da imagem antes de salvar o arquivo de saída. O diretório das imagens de entrada e saída são passados também como argumentos da função.

Os parâmetros da função convolui são documentados na tabela 1.

Tabela 1: Referência da função convolui.

Parâmetro	Descrição
original file path	Caminho absoluto para a imagem que se deseja tratar
filtro	Matriz da máscara de filtro a ser aplicada
output file path	Caminho absoluto para o arquivo de imagem de saída (default: convolve-img.png)
retorno	Retorna a matriz da imagem calculada.

```
def convolui(original_file_path, filtro, output_file_path="convolve-img.png"):
    """
    Recebe uma imagem uint8 em escala de cinza, um filtro matricial e a retorna,
    salvando tambem no arquivo de saida especificado.
    :param original_file_path: A imagem original
    :param filtro: O filtro a se aplicar
    :param output_file_path: Path da imagem de saida
    :return: Matriz da imagem de saida
    """
```

Figura 1: Header da função utilizada para convoluir a imagem com o filtro.

```
$ python main.py
```

Figura 2: Comando para gerar as saídas requisitadas pelo roteiro.

2.1 Executando o script de demonstração

Dada uma imagem original, o roteiro solicita que sejam geradas imagens resultantes da aplicação de cada um dos filtros convolucionais listados no roteiro. Todas estas imagens são geradas quando se executa o script "main.py" como arquivo principal. Um exemplo de como executar o script neste modo sob a linha de comando é exibido na figura 2. A imagem usada como exemplo é a "baboon.png" (figura 3) e deve estar no mesmo diretório de chamada do script.

Todas as imagens solicitadas pelo roteiro serão então geradas em um diretório de "output" automaticamente criado no mesmo diretório onde o script é chamado. Os nomes de cada arquivo indicam a máscara de filtro aplicada em cada imagem, conforme o solicitado. As matrizes de cada filtro e imagem sendo operados são ecoadas na saída principal, mas não constituem parte essencial para a compreensão deste roteiro, apenas exibindo sua composição para fins de esclarecimento de eventuais dúvidas.

2.2 Utilizando as funções interativamente

A forma com que o script foi construído nos permite importar suas funções e utilizá-las independentemente. Para fazê-lo, basta realizar o *import* tradicional do python, como na figura 4.

2.3 Aplicação de filtros convolucionais

A convolução é uma importante ferramenta para o tratamento de imagens e a aplicação de filtros no formato de máscaras. O processo a ser realizado é semelhante àquele executado na operação de correlação, alterando-se apenas a ordem em que são percorridos os índices da máscara sendo aplicada, como mostra a equação 1.

$$\mathbf{w}(x, y) * \mathbf{f}(x, y) = \sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} w(i, j) f(x - i, y - j) \quad (1)$$

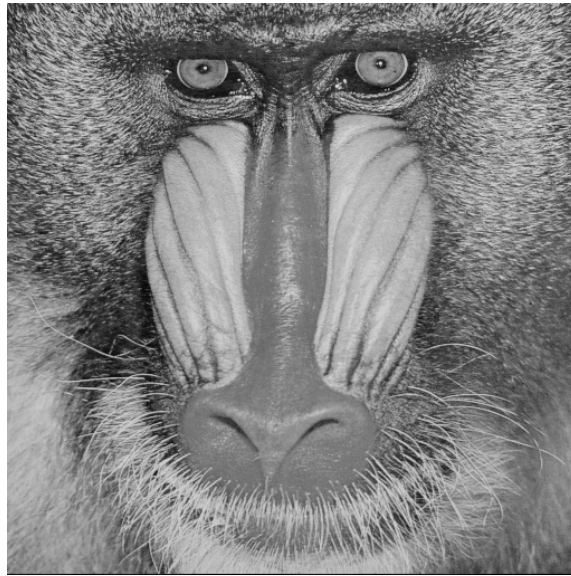


Figura 3: Imagem original usada como exemplo.

```
. python
Python 3.7.4 (default, Aug 13 2019, 20:35:49)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from main import *
>>> 
```

Figura 4: Importando funções desenvolvidas para este roteiro.

O processo de filtragem por convolução consiste em centralizar o filtro sobre o pixel sendo operado e substituir neste o valor da soma dos produtos entre cada elemento da máscara e o respectivo pixel da imagem alinhado, ressaltando a necessidade de fazer a inversão da máscara caso a operação seja de convolução real, e não correlação.

Para a implementação das operações, foi utilizada a função "ndimage.convolve" do pacote "scipy" em python. Esta função já realiza a operação de convolução real de maneira vetorizada, não sendo necessária a inversão do filtro previamente ou a execução de loops externos. Esta função também realiza o *padding* da imagem, que é o processo de adicionar pixels inexistentes na imagem original às bordas desta, permitindo aplicar o filtro nos pixels das bordas. Por padrão, como utilizado neste script, a função "convolve" repete os valores dos pixels das bordas nos pixels adicionados, o que parece mais coerente, dado o fato de que pixels próximos costumam ter valores relativamente próximos em fotos reais. Os filtros utilizados neste roteiro são expressos na figura 5.

2.3.1 Filtro h1

O resultado da aplicação da máscara h1 sobre a imagem original é mostrada na figura 6. A máscara h1 é também conhecida como "Unsharp masking"[2], e tem um efeito de aumentar a nitidez da imagem, ressaltando a definição da mesma, embora se distancie do conceito originalmente retratado. Os pixels multiplicados pelos valores negativos subtraem intensidade daquele no qual a máscara

$$h_1 = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

$$h_2 = \frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

$$h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$h_4 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$h_5 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$h_6 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$h_7 = \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}$$

$$h_8 = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

Figura 5: Filtros a serem aplicados na imagem.

está centralizada e sendo aplicada. Se este pixel for um contorno ou extremidade de um detalhe da imagem, ele já tende a ser um pouco mais escuro e ainda será subtraído pelo resultado da imagem, ressaltando este contorno.

2.3.2 Filtro h2

O resultado da aplicação da máscara h2 sobre a imagem original é mostrada na figura 7. A máscara h2 desempenha o papel de filtro gaussiano, o que é útil para suavizar a imagem e mitigar ruídos, embora possa eliminar certos detalhes relativamente pontuais, devido ao peso que coloca nos pixels da vizinhança serem relativamente próximos aos do pixels originais.

2.3.3 Filtro h3

O resultado da aplicação da máscara h3 sobre a imagem original é mostrada na figura 8. A máscara h3 é um filtro que resalta contornos verticais, o que se evidencia pela disposição de valores ao longo da matriz. O pixel resultante da aplicação da máscara não leva em consideração o próprio valor, mas sim o dos pixels nas colunas laterais, sendo que, quando as colunas de cada lado possuem valores qualitativamente opostos (uma delas for de pixels claros e, a outra, de escuros), o pixel sendo tratado terá seu valor mais intensamente alterado.

2.3.4 Filtro h4

O resultado da aplicação da máscara h4 sobre a imagem original é mostrada na figura 9. Ao contrário da máscara h3, a máscara h4 é um filtro que resalta contornos horizontais, o que fica evidente pela disposição de valores ao longo da matriz. O pixel resultante da aplicação da máscara não leva em consideração o próprio valor, mas sim o dos pixels nas linhas acima e abaixo dele, sendo que, quando as linhas de cada lado possuem valores qualitativamente opostos (uma delas for de pixels claros e, a outra, de escuros), o pixel sendo tratado terá seu valor mais intensamente alterado.

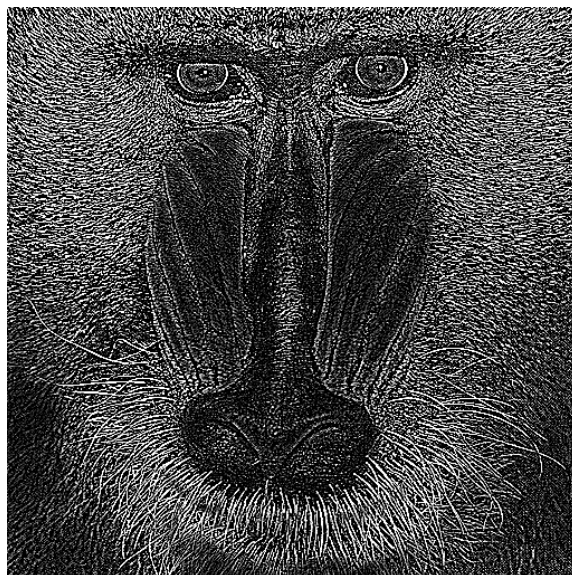


Figura 6: Resultado da aplicação da máscara de filtro h1.

2.3.5 Filtro h5

O resultado da aplicação da máscara h5 sobre a imagem original é mostrada na figura 10. Seu efeito de ressaltar contornos é evidente pela própria disposição de valores pelos elementos da máscara. Quando os pixels da vizinhança tiverem valores baixos, o pixels centralizado na máscara serão intensificados, ressaltando contornos em todas as direções na imagem.

2.3.6 Filtro h6

O resultado da aplicação da máscara h6 sobre a imagem original é mostrada na figura 11. A máscara h6 é também conhecida como "Box blur", por aplicar uma espécie de "borrão" igualmente por toda a imagem, fazendo como resultado de cada operação uma média entre os valores do pixel sendo operado e daqueles na vizinhança.

2.3.7 Filtro h7

O resultado da aplicação da máscara h7 sobre a imagem original é mostrada na figura 12. À semelhança das máscaras h3 e h4, esta máscara h7 também resalta contornos pelas contraposições entre elementos negativos e positivos na matriz, porém, os contornos enfatizadas são aqueles paralelas à diagonal principal da matriz da imagem.

2.3.8 Filtro h8

O resultado da aplicação da máscara h8 sobre a imagem original é mostrada na figura 13. À semelhança da máscara h7, esta máscara h8 também resalta contornos pelas contraposições entre

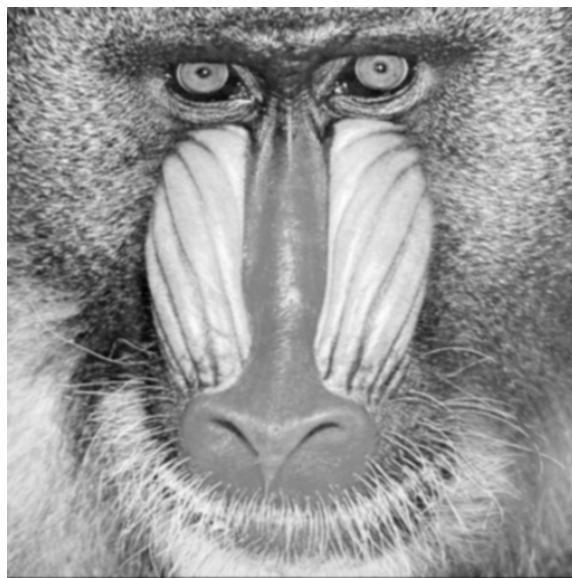


Figura 7: Resultado da aplicação da máscara de filtro h2.

elementos negativos e positivos na matriz, porém, os contornos enfatizados são aqueles perpendiculares à diagonal principal da matriz da imagem.

2.3.9 Combinação de h3 e h4

Após aplicadas as máscaras h3 e h4 separadamente na imagem original, teremos contornos verticais e horizontais ressaltados em cada imagem. A raiz da soma dos quadrados de forma "element wise" nestas duas imagens irá então, conforme se espera, ressaltar a existência de contornos em todas as direções, sendo conhecida como "filtro de Sobel"[1]. O resultado é expresso na imagem 14.

3 Conclusão

As imagens geradas deixam claro que os filtros por máscaras convolucionais são ferramentas poderosas e versáteis, ao passo que podem ser aplicados em diferentes imagens de maneira extremamente simples.

Para as imagens geradas neste roteiro, os pixels com valores que extrapolassem 255 foram "truncados" para 255, enquanto que pixels menores que zero tiveram seu valor alterado para zero. Desta forma, mais detalhes são evidenciados na imagem, embora isto implique no escurecimento de partes da imagem que não são o alvo do filtro, principalmente daqueles encarregados de ressaltar bordas e contornos.

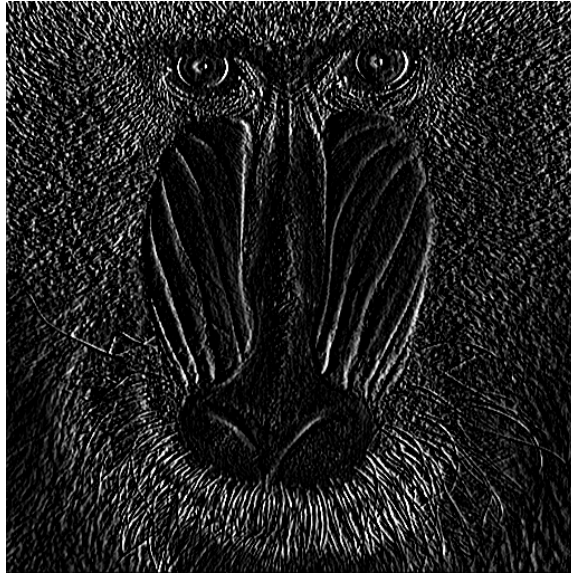


Figura 8: Resultado da aplicação da máscara de filtro h3.

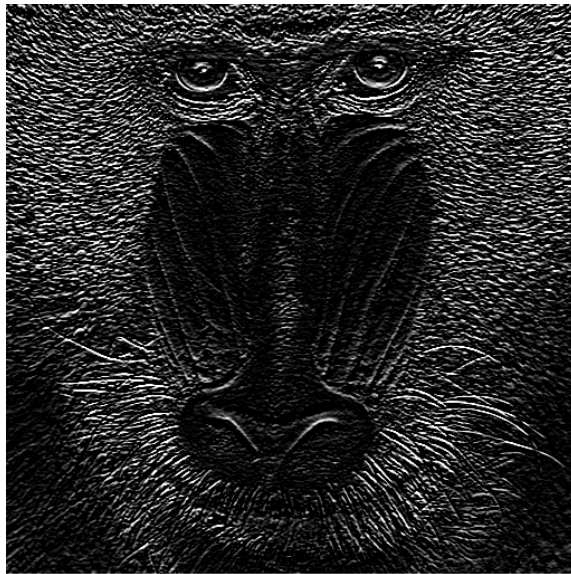


Figura 9: Resultado da aplicação da máscara de filtro h4.

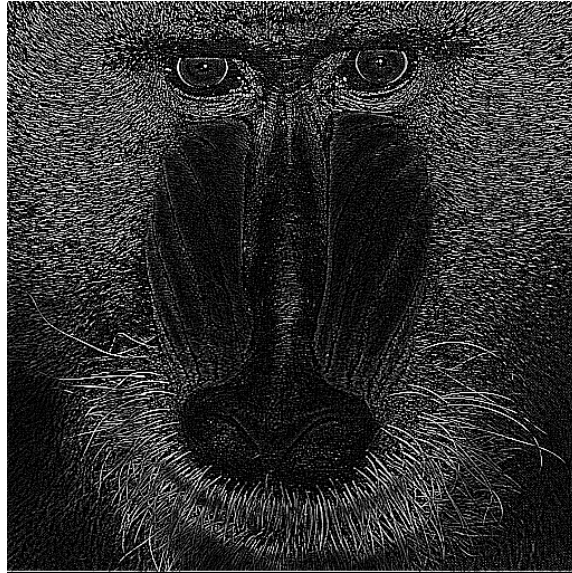


Figura 10: Resultado da aplicação da máscara de filtro h5.

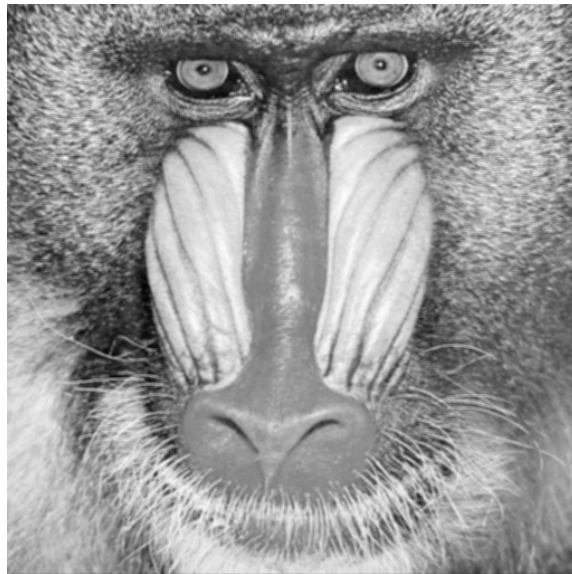


Figura 11: Resultado da aplicação da máscara de filtro h6.

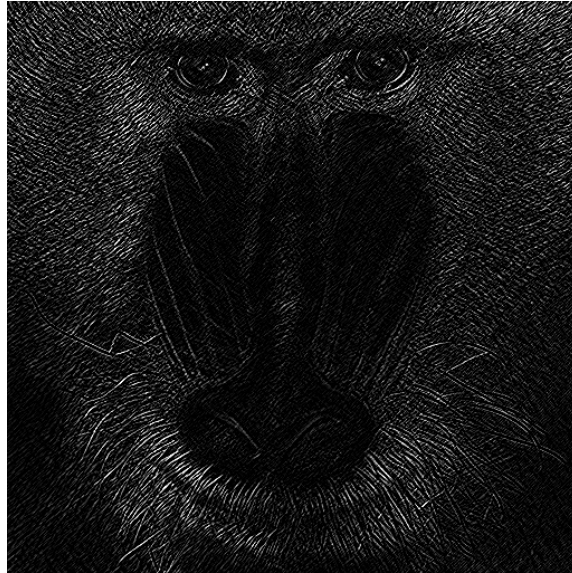


Figura 12: Resultado da aplicação da máscara de filtro h7.

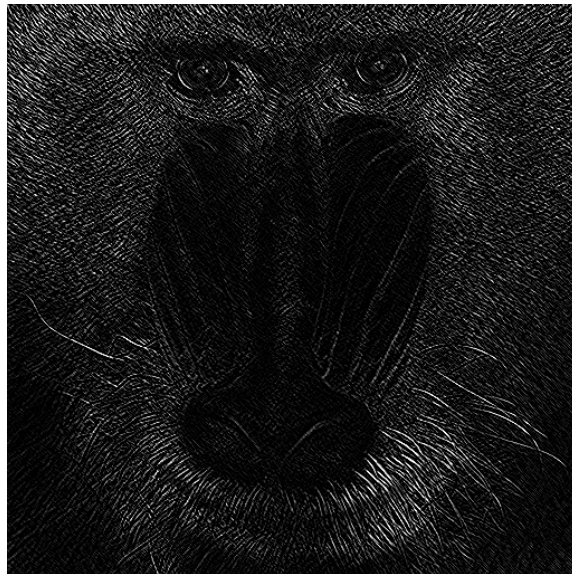


Figura 13: Resultado da aplicação da máscara de filtro h8.

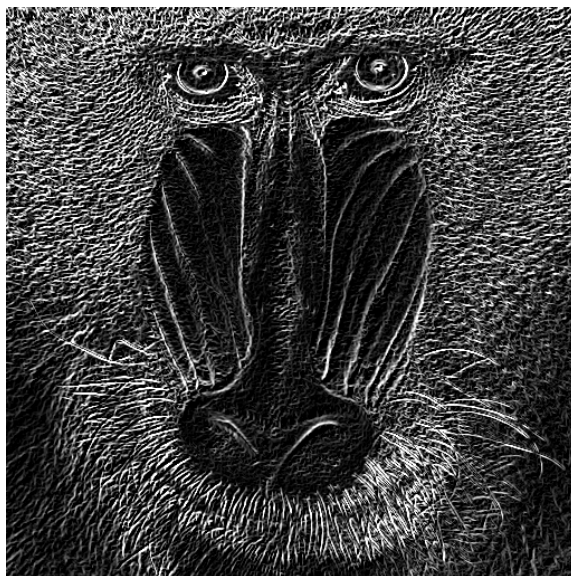


Figura 14: Resultado da aplicação das máscaras de filtro h3 e h4.

Referências

- [1] Wikipédia. *Filtro Sobel* — *Wikipédia, a enciclopédia livre*. [Online; accessed 28-março-2013]. 2013. URL: https://pt.wikipedia.org/w/index.php?title=Filtro_Sobel&oldid=35047926.
- [2] Wikipedia contributors. *Kernel (image processing)* — *Wikipedia, The Free Encyclopedia*. [Online; accessed 25-April-2020]. 2020. URL: [https://en.wikipedia.org/w/index.php?title=Kernel_\(image_processing\)&oldid=951378908](https://en.wikipedia.org/w/index.php?title=Kernel_(image_processing)&oldid=951378908).