

Trabalho 1 - Convolução e aplicação de filtros em imagens

Nome: Rebecca Moreira Messias RA: 186416

14/04/2020

1. Introdução

Uma forma de se pensar numa imagem é como uma matriz de duas dimensões contendo valores de cor dos pixels no intervalo de 0 a 255. Dessa maneira é possível fazer manipulações de imagens com operações matemáticas em matriz.

Este relatório tem como objetivo apresentar o resultado de algumas manipulações realizadas em imagens coloridas e monocromáticas no domínio espacial web.

Na primeira parte, são manipuladas imagens coloridas na forma RGB com dois filtros diferentes. O primeiro transforma a imagem em tons amarelados. Já o segundo altera a imagem colorida para que ela tenha apenas uma banda de cor.

Na segunda parte, são manipuladas imagens monocromáticas e aplicado alguns filtros com convolução. A convolução de imagens basicamente é um processo em que tendo uma imagem e filtro na forma matricial conseguimos gerar uma nova imagem percorrendo a imagem original e manipulando os pixels e os pixels ao redor de acordo com o filtro.

2. Código

Acompanhando esse relatório tem um arquivo .zip chamado *MC920 - Rebecca Moreira Messias - 186416.zip*, onde se encontra o código no arquivo *trabalho1.py*. Como opção alternativa também tem um Jupyter Notebook *trabalho1.ipynb*.

3. Solução

3.1. Dependências

As bibliotecas utilizadas no código são: matplotlib, numpy, math, cv2, Pillow.

3.2. Limitações

O algoritmo possui o caminho das imagens fixo no código. Além disso, não suporta convolucionar imagens coloridas e só suporta o formato PNG.

3.3. O algoritmo de convolução

O algoritmo de convolução possui algumas funções:

- `converteImagemMatriz`: A função acima retorna uma matriz NumPy 2D dimensional contendo os valores de pixel.

- subMatrizes: Essa função retorna uma matriz gigante contendo submatrizes do kernel de tamanho que será usada novamente mais tarde.
- aplicacaoFiltro: A matriz gigante é passada como o argumento junto com o filtro de kernel na função para realizar a convolução.
- plotImagem: Função que faz plotagem e compara a imagem original com a imagem transformada após a convolução.

Os filtros foram definidos com a biblioteca numpy e a combinação de h1 e h2 foi feita aplicando primeiro h1 na imagem original e h2 nessa imagem resultante.

As imagens resultantes são salvas na raiz do diretório.

4. Resultados

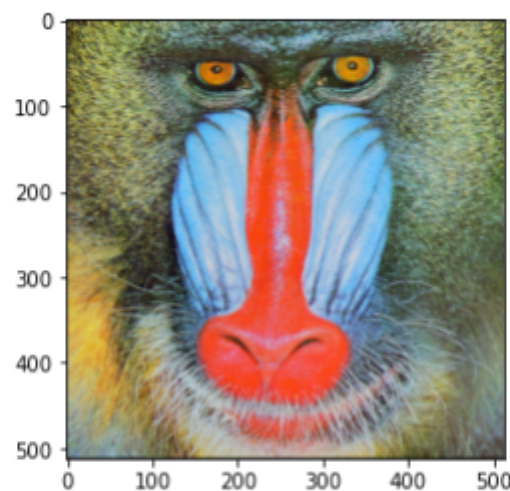


Imagem 1: Imagem colorida original utilizada nos itens na seção 1.1 do enunciado.

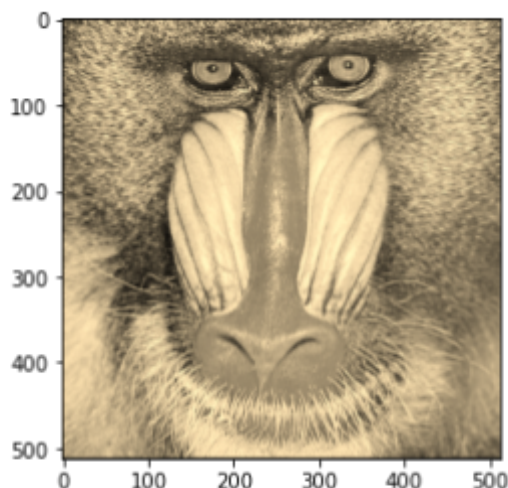


Imagem 2: Baboon.png com alterações para deixar a imagem com o filtro Sépia.

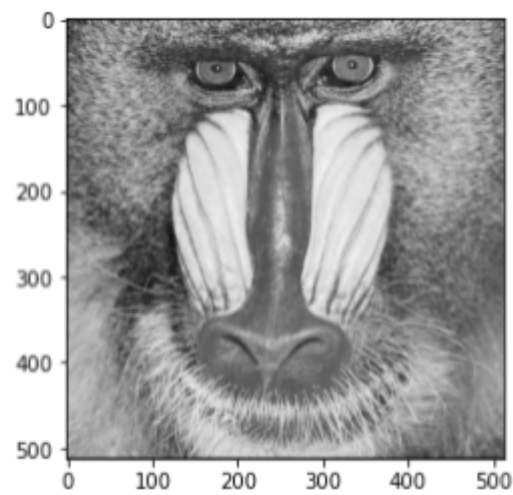


Imagem 3: Baboon.png com alteração para que a imagem fica com apenas uma banda de cor.

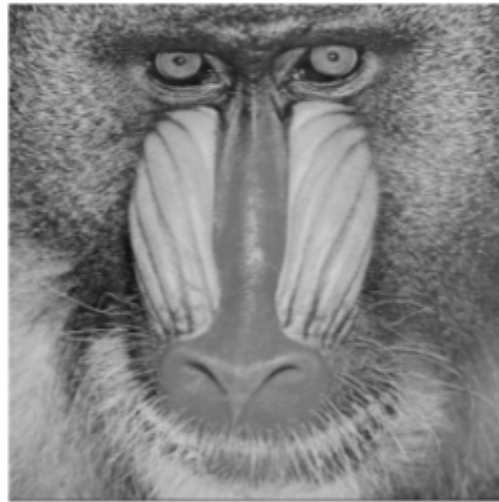


Imagem 4: Imagem monocromática utilizada para testar o algoritmo de convolução no item 1.2 do enunciado.

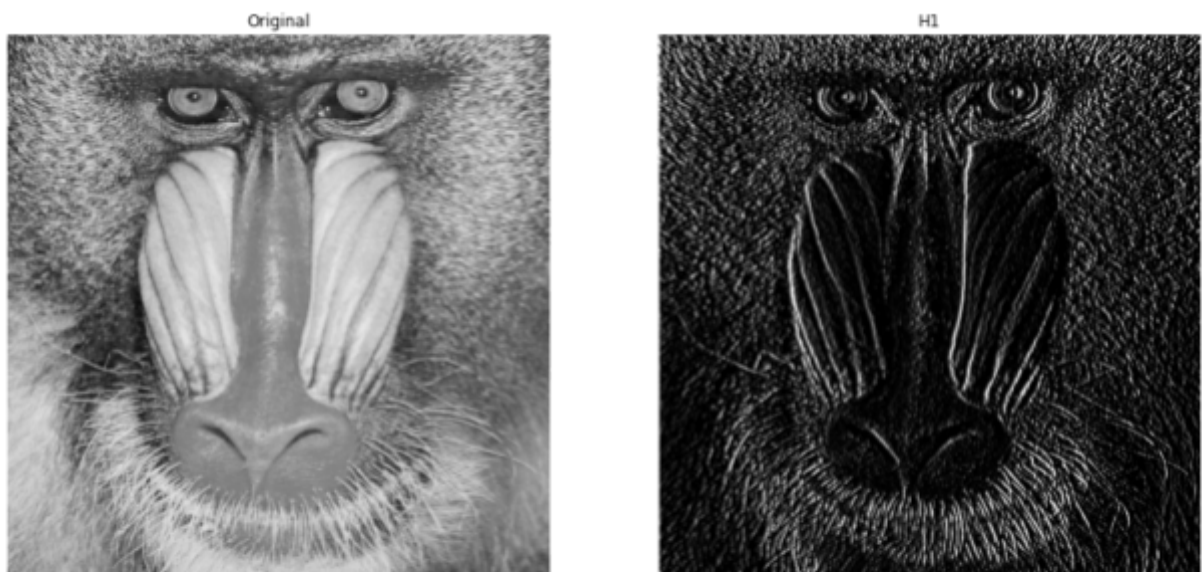


Imagem 5: Aplicação do filtro h1 na imagem monocromática.

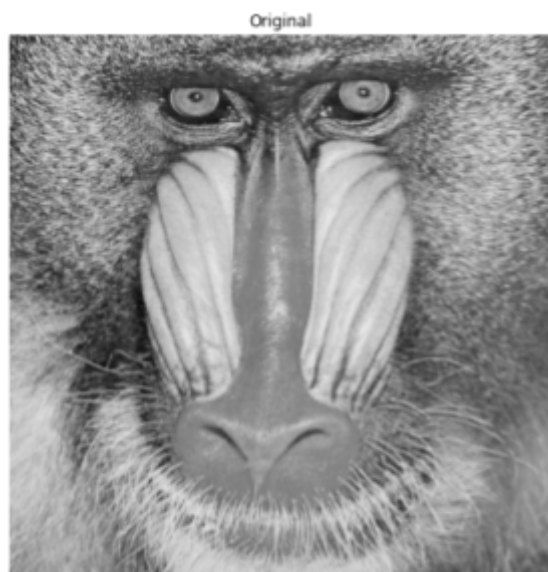


Imagem 6: Aplicação do filtro h2 na imagem monocromática.

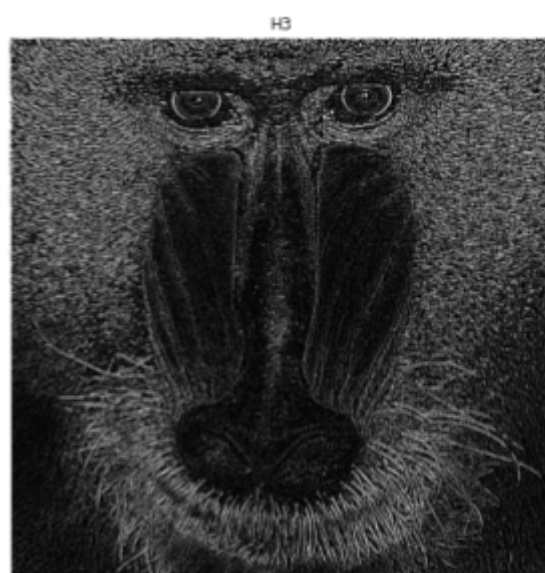
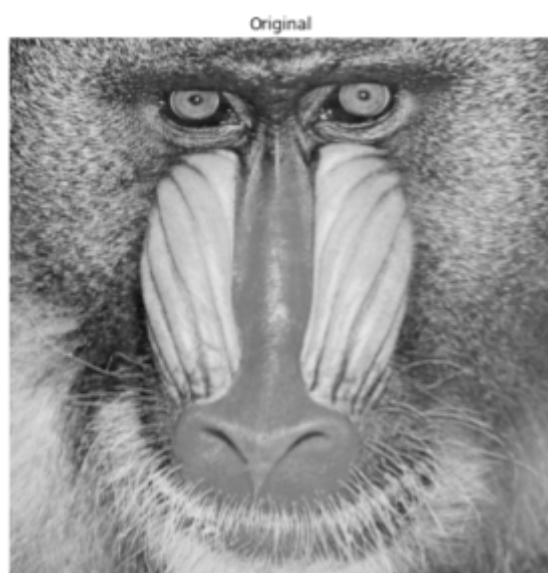


Imagem 7: Aplicação do filtro h3 na imagem monocromática.

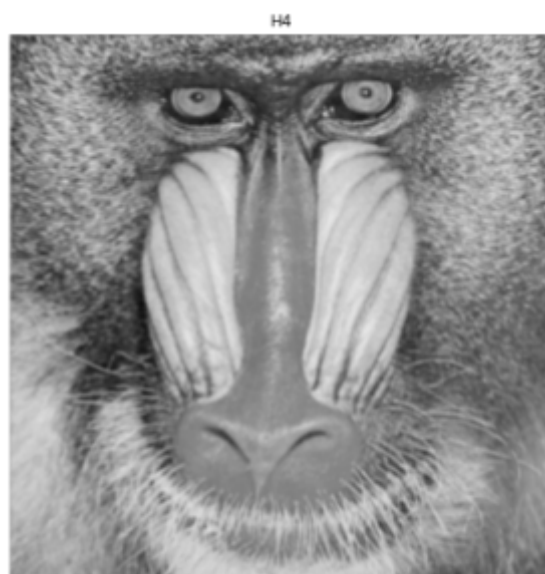
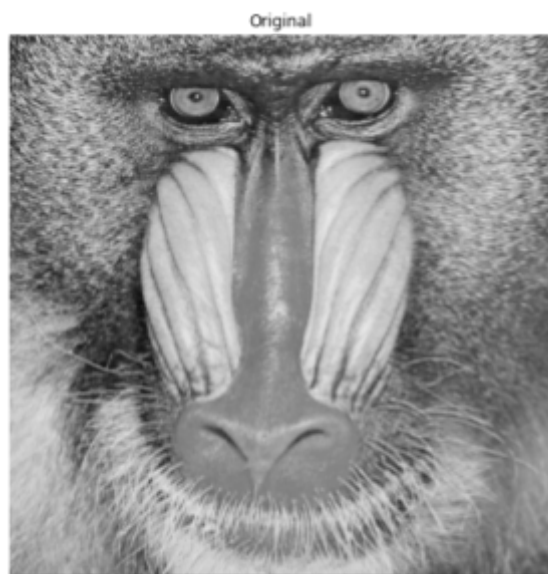


Imagem 8: Aplicação do filtro h4 na imagem monocromática.

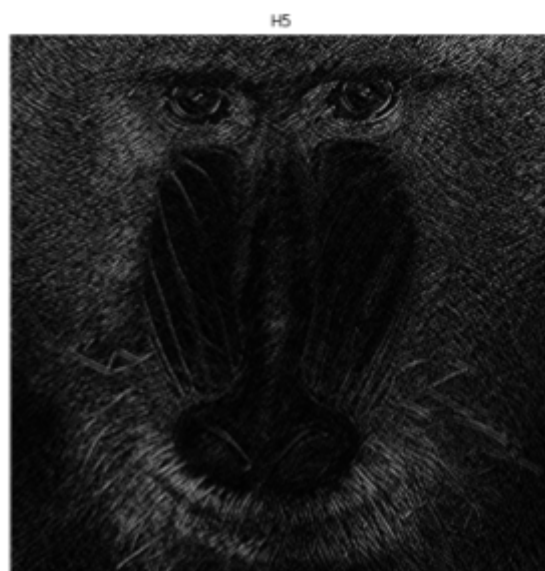
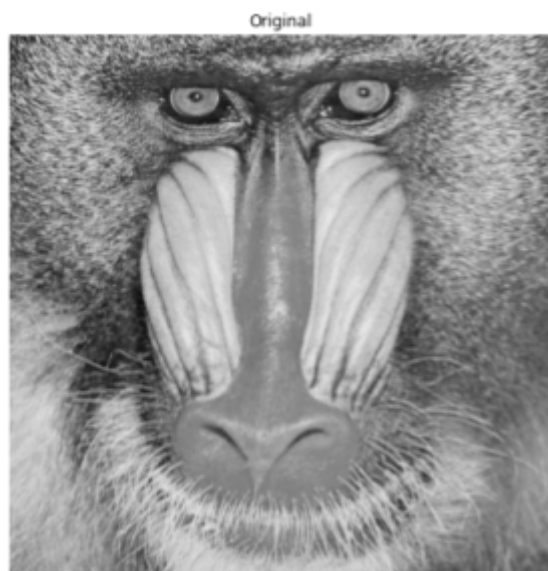


Imagem 9: Aplicação do filtro h5 na imagem monocromática.

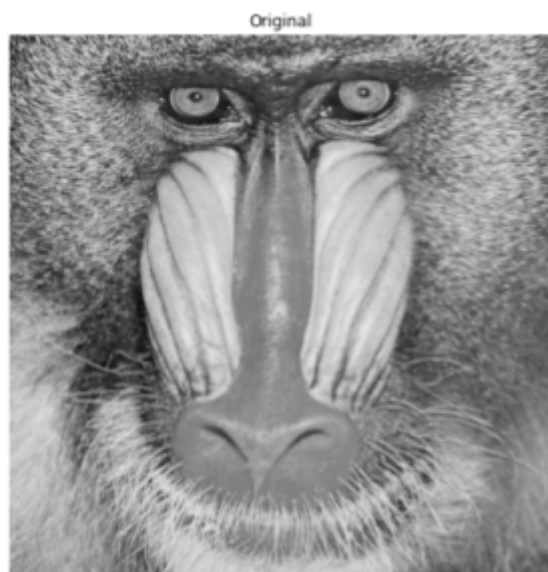


Imagem 10: Aplicação do filtro h6 na imagem monocromática.

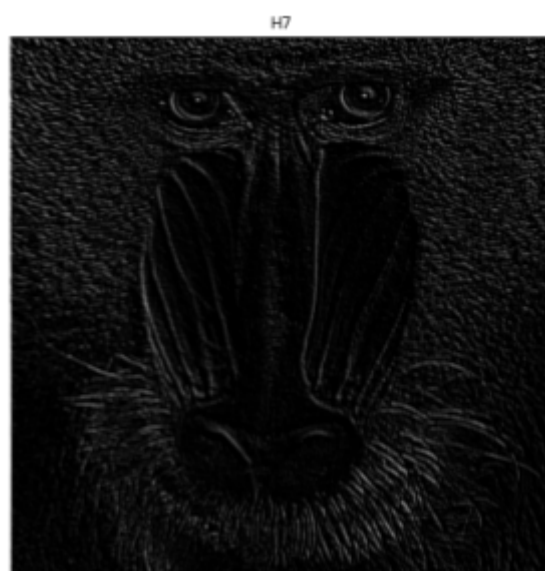
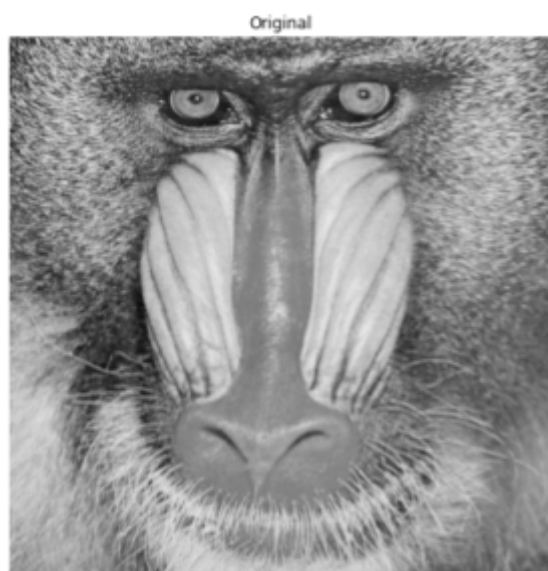


Imagem 11: Aplicação do filtro h7 na imagem monocromática.

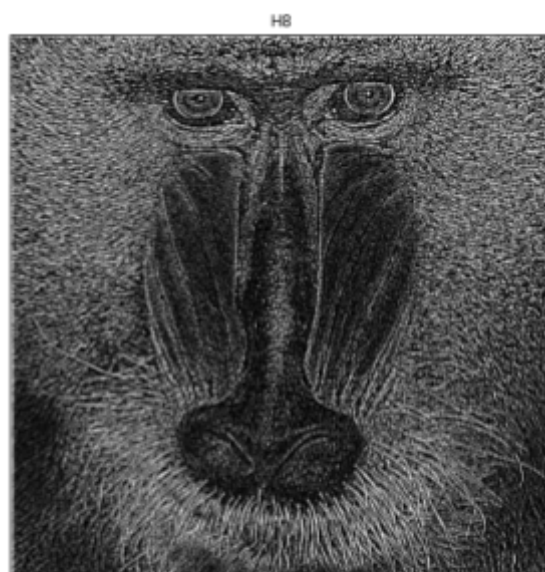
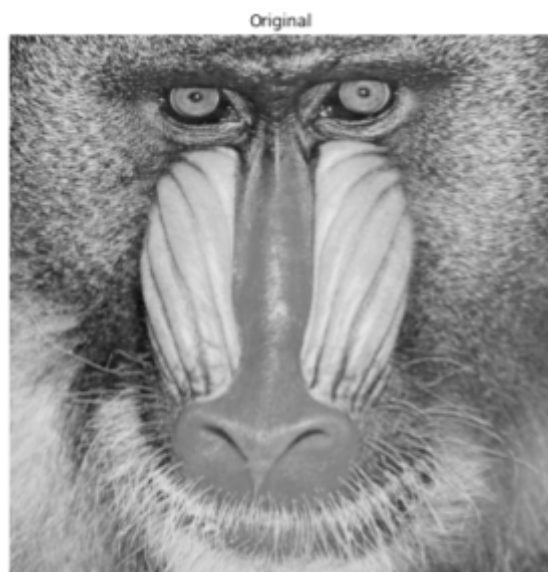


Imagem 12: Aplicação do filtro h8 na imagem monocromática.

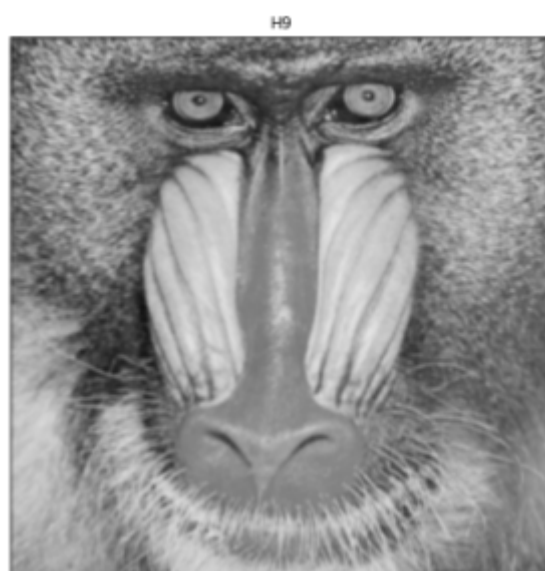
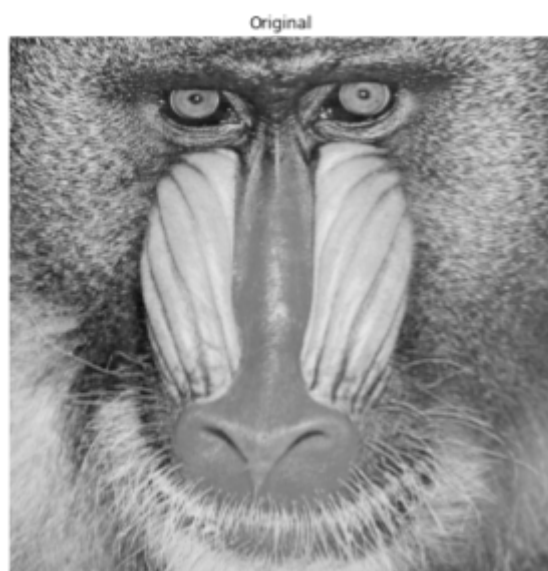


Imagem 13: Aplicação do filtro h9 na imagem monocromática.



Imagem 14: Aplicação conjunta do filtro h1 e h2 na imagem monocromática.

5. Conclusões

No item a da seção 1.1 foi pedido para que fossem feitas as operações da Imagem 15 em uma imagem colorida qualquer.

$$R' = 0.393R + 0.769G + 0.189B$$

$$G' = 0.349R + 0.686G + 0.168B$$

$$B' = 0.272R + 0.534G + 0.131B$$

Imagem 15: Operações para efeito sépia

De acordo com o resultado que vimos na Imagem 2, podemos dizer que essas operações fazem com que a imagem tenha efeito sépia.

Já no item b da seção 1.1 o objetivo foi deixar a imagem apenas em uma banda de cor e de acordo com os resultados obtidos na Imagem 3, percebemos que o filtro aplicado deixou a imagem em tons de cinza.

Na seção 1.2 foi proposto para que fossem aplicados os filtros da Imagem 16 em imagens monocromáticas. E além desses, também fazemos uma combinação do filtro h1 e h2 de acordo com a fórmula $\sqrt{(h1)^2 + (h2)^2}$.

$$\begin{aligned}
 h_1 &= \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} & h_2 &= \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} & h_3 &= \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \\
 h_4 &= \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} & h_5 &= \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} & h_6 &= \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} & h_7 &= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \\
 h_8 &= \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix} & h_9 &= \frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}
 \end{aligned}$$

Imagem 16: Filtros que devem ser aplicados nas imagens monocromáticas.

De acordo com os resultados obtidos podemos dizer que os filtros tem os seguintes efeitos:

- h_1 : Esse filtro é um detector de bordas. Por ele ter um efeito mais suave, a imagem resultante é menos afetada pelo ruído. Conhecido como Borda de Sobel Vertical.
- h_2 : Esse filtro é um detector de bordas. Por ele ter um efeito mais suave, a imagem resultante é menos afetada pelo ruído. Conhecido como Borda de Sobel Horizontal.
- h_3 : Esse é um filtro para detecção de bordas. Ele tem como objetivo identificar pontos para os quais o brilho da imagem muda. Conhecido como Detecção de Bordas de Canny.
- h_4 : Filtro em que cada pixel da imagem resultante tem um valor igual ao valor médio de seus pixels vizinhos. O filtro é conhecido como Box Blur e o resultado é uma imagem com um efeito blur suave.
- h_5 : Este é um filtro de detecção de linhas. No caso desse filtro, ele detecta linhas em 45° (diagonal secundária).
- h_6 : Também é um filtro de detecção de linhas, mas nesse caso a detecção ocorre em linha de 135° (diagonal principal).
- h_7 : É um filtro de detecção de arestas. Conhecido como Roberts Cross. Foi um dos primeiros detectores de borda.
- h_8 : Este é um filtro para detectar bordas conhecido como Laplacian of Gaussian. o filtro Laplaciano por si só é muito sensível ao ruído. Por conta disso, esse filtro propõe suavizar a imagem antes.

- h9: Tem como objetivo reduzir o ruído da imagem e reduzir os detalhes. O efeito é como se a imagem fosse vista por uma tela translúcida. É conhecido como Gaussian Blur.

Com isso foi possível visualizar alguns filtros que são muito utilizados nos para processamento de imagem digital e visão computacional e seus efeitos nas imagens.