

Trabalho 2

MO443: Introdução ao Processamento de Imagem Digital

Henrique de Abreu Amitay - RA: 138501

23 de abril de 2020

1 Introdução

Este trabalho tem o objetivo de explorar o comportamento e aplicação de alguns filtros em imagens.

Serão exploradas nove filtros diferentes conforme imagem abaixo.

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

Figura 1: Filtros utilizados

Junto deste relatório se encontram os scripts para cada etapa do trabalho junto à algumas imagens de exemplo.

O trabalho foi implementado utilizando a linguagem de programação **Python** na versão **3.6.9**.

Além disso os seguintes pacotes foram utilizados para a implementação:

- OpenCv (cv2)
- Numpy

Todas as etapas deste exercício foram feitos a partir da seguinte imagem em escala de cinza:

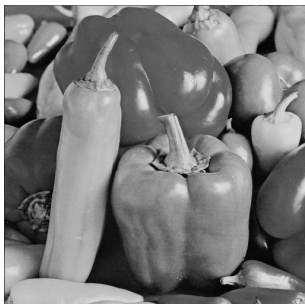


Figura 2: Imagem base utilizada em escala de cinza

2 Exercícios

2.1 Aplicação dos filtros

Todos os filtros puderam ser aplicados com o auxílio de métodos oferecidos pelos pacotes **Numpy** e **OpenCV**. O processo de aplicação se deu da seguinte maneira:

- Carregamento da imagem em escala de cinza
- Rotação em 180° do filtro (o que no caso destes filtros é desnecessário dado que as matrizes são simétricas, porém o código foi desenvolvido pensando em um caso geral). A operação de rotação é facilitada pela função **rot90** do **Numpy**.
- Por ultimo os filtros são aplicados com o auxílio da função **filter2D**, que permite diferentes implementações do cálculo de bordas, para este o método de borda constante.

O código abaixo ilustra um exemplo de aplicação de filtro

```
1 # Definicao do filtro
2 h1 = np.array([[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]], float)
3
4 # Rotacionamos o filtro 90 graus duas vezes
5 mirrored_filter = np.rot90(h1, 2)
6
7 filtered_image = cv2.filter2D(img, -1, mirrored_filter)
8
```

Todo este processo foi repetido para todos os filtros e está disponível no código **ex1.py**

2.2 Discussão dos resultados obtidos

O filtro **h1** é uma janela que engloba uma área quadrada de 9 pixels, porém apenas os 13 pixels centrais são considerados no cálculo da convolução, dado que os pixels mais "fronteiriços" possuem peso zero.

Podemos observar que este filtro realça as bordas da imagem porém com um ruído considerável. Isto provavelmente se dá pelo tamanho da janela e pelo peso consideravelmente maior do pixel central.

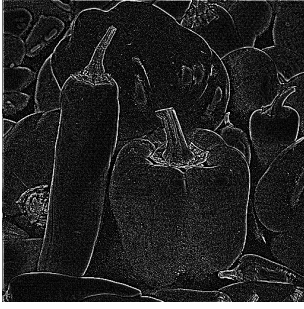


Figura 3: Imagem obtida após a aplicação do filtro $h1$



Figura 6: Imagem obtida após a aplicação do filtro $h4$

O filtro $h2$ também é uma janela que engloba uma área quadrada de 9 pixels, porém neste caso todos os pixels são multiplicados por valores menores que um no processo da convolução.

Isto cria um leve efeito de atenuação e desfocamento conforme imagem abaixo



Figura 4: Imagem obtida após a aplicação do filtro $h2$

Os filtros $h3$ e $h4$ mostram um comportamento curioso de detecção de borda. O filtro $h3$ mostra um maior realce nas bordas verticais, enquanto o filtro $h4$ nas bordas horizontais.

Isto se dá devido ao fato da coluna nula em $h3$ e da linha nula em $h4$.

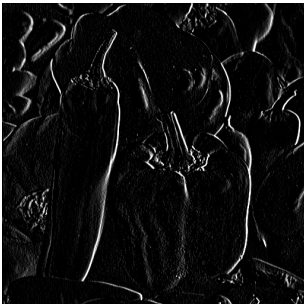


Figura 5: Imagem obtida após a aplicação do filtro $h3$

Podemos também combinar estes filtros com a operação $f(x) = \sqrt{h_3^2 + h_4^2}$. Esta operação mantém o comportamento de ambos os filtros, ou seja, uma detecção de bordas tanto horizontal quanto vertical conforme imagem abaixo. A combinação destes filtros é o chamado **Filtro de Sobel**.

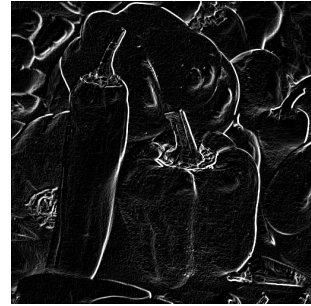


Figura 7: Imagem obtida após a aplicação do Filtro de Sobel: $\sqrt{h_3^2 + h_4^2}$

O filtro $h5$ demonstra um comportamento semelhante com $h1$, com uma detecção de borda bi-dimensional e ruído, porém ambos com menor intensidade. Isto se dá devido aos pesos menos intensos da matriz e sua menor dimensão.

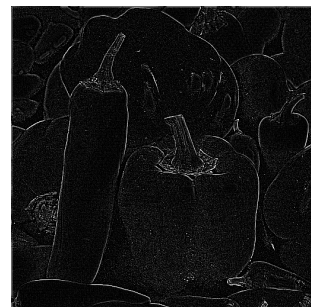


Figura 8: Imagem obtida após a aplicação do filtro $h5$

Da mesma maneira o filtro $h6$ demonstra um comportamento de atenuação semelhante à $h2$, porém com um desfoque menor, dado que possui pesos maiores.



Figura 9: Imagem obtida após a aplicação do filtro $h6$

Os filtros $h7$ e $h8$ apresentam comportamento semelhante, dado que são simétricos. Podemos observar as diferenças entre as operações de convolução e correlação de uma mesma imagem por um mesmo filtro. No caso ambos os filtros mostraram bastante ruído, porém de maneira homogênea.

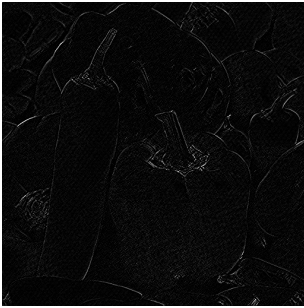


Figura 10: Imagem obtida após a aplicação do filtro $h7$

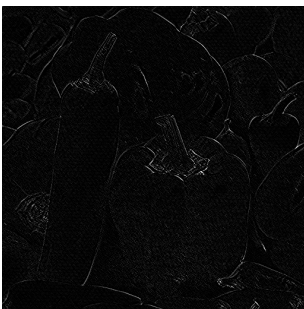


Figura 11: Imagem obtida após a aplicação do filtro $h8$