

MC920 - Trabalho 1

Nome: Felipe Escórcio de Sousa - **RA:** 171043

Outubro 2020

1 Introdução

Este trabalho foi elaborado com o objetivo de implementar estudar o uso e aplicação de filtros em imagens. Uma das formas mais comuns na aplicação de filtros em imagens é por meio de aplicação de convoluções, usando-se matrizes simétricas.

2 Execução do código

O script foi executado usando-se Python 3.8.5, usando-se as bibliotecas NumPy e OpenCV2, executando-se o arquivo principal `lab01.py` pelo interpretador, como no exemplo:

```
python3 lab01.py -i input.png -o output.png -k 7
```

E os argumentos possíveis são

- input - caminho para a imagem png de entrada.
- output - caminho da imagem de saída pretendido.
- kernel - kernel pretendido entre os 11 disponíveis.
- combine - opção para usar a combinação entre os kernels 3 e 4.

como também a primeira letra de cada um dos parâmetros é aceita.

3 Entradas

Os kernels utilizados estão presentes no arquivo `kernels.py` e são importados diretamente pelo programa principal.

4 Execução

Os principais resultados da execução do script, para cada kernel aplicado estão demonstrados abaixo:

Kernel 1: Primeiramente, a imagem de teste tentada foi uma com um babuíno, porém como esta tem bem mais detalhes, ela foi novamente testada com uma imagem mais simples.

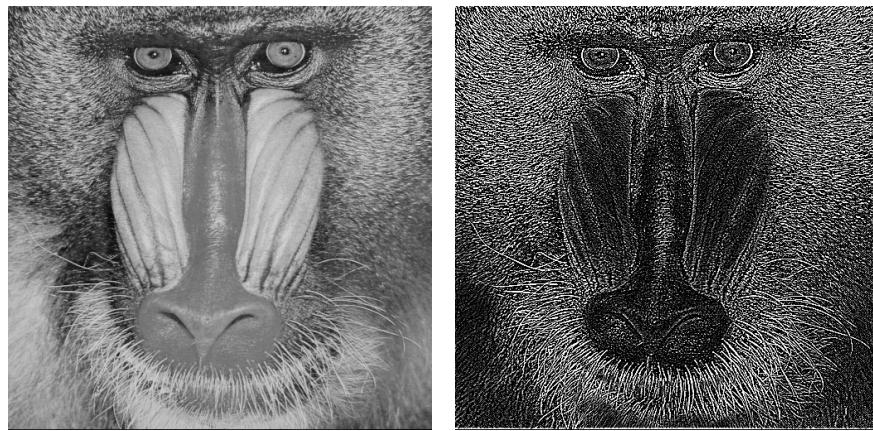


Figura 1: Kernel 1 aplicado a uma imagem detalhada.

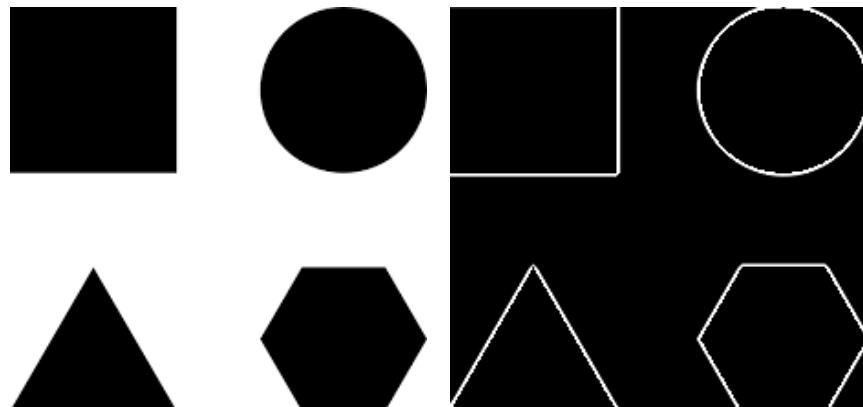


Figura 2: Kernel 1 aplicado a uma imagem com formas simples.

Com alguma pesquisa, encontramos que é um kernel que implementa uma laplaciana de uma gaussiana.

Kernel 2: Nesta facilmente foi encontrado que é um kernel de Gaussian Blur, não tendo sido muito necessário o uso de uma outra imagem mais simples.



Figura 3: Kernel 2 aplicado a uma imagem.

Kernel 3: Aplicando-se o kernel 3 à ambas imagens, a fim, de melhor exemplificar, obtemos:

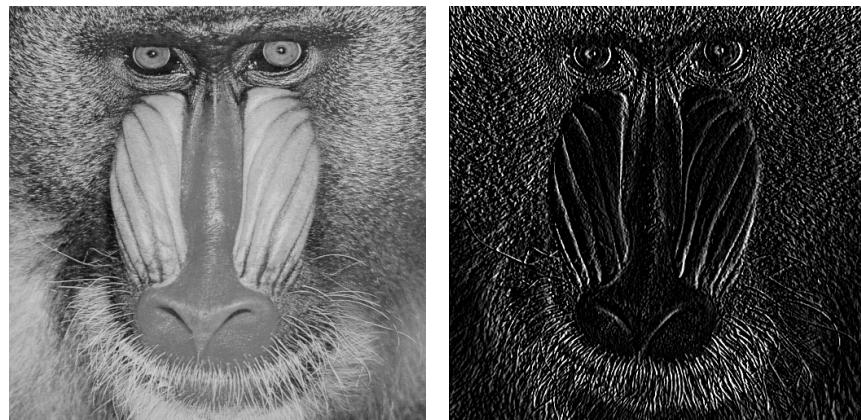


Figura 4: Kernel 3 aplicado a uma imagem com muitos detalhes.

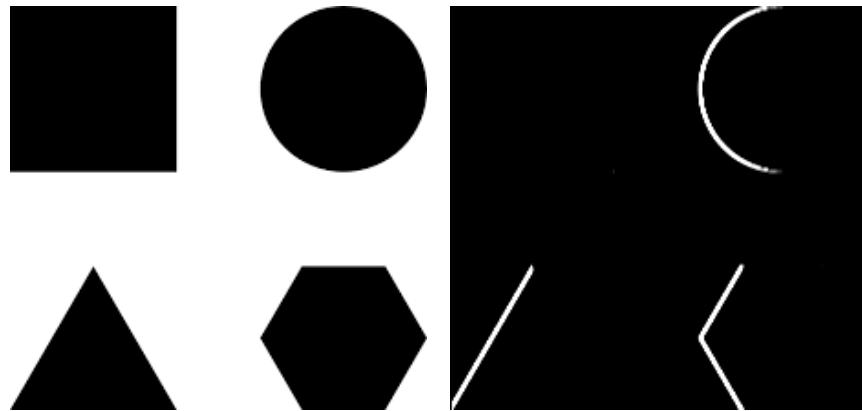


Figura 5: Kernel 3 aplicado a imagens menos detalhadas.

Tal que, com alguma pesquisa encontramos que temos um kernel de Sobel direito.

Kernel 4: Semelhantemente ao 3, mas como um Sobel inferior, temos:

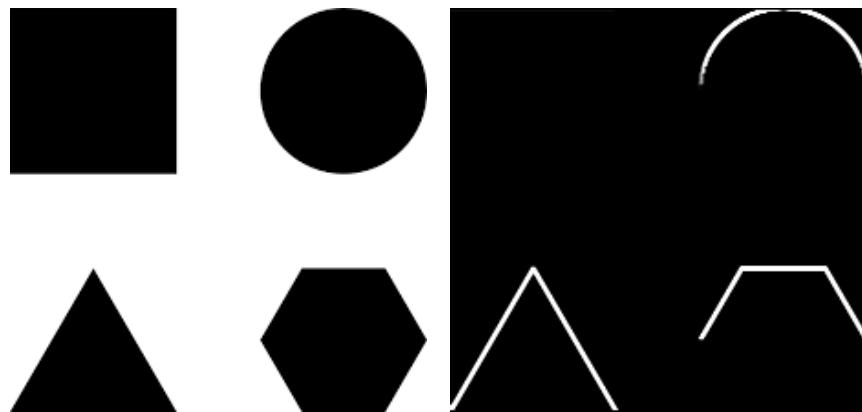


Figura 6: Kernel 4 aplicado a uma imagem simples.

Kernel 5: Analisando-se e com um pouco de pesquisa, descobriu-se que aqui temos um kernel que realiza detecção de bordas.

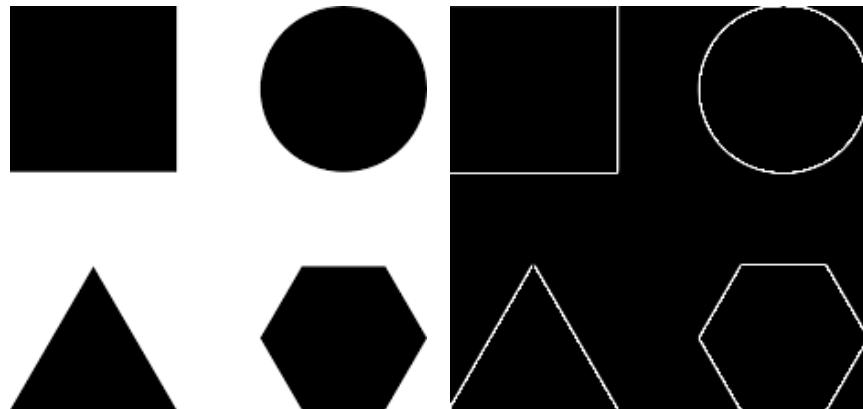


Figura 7: Kernel 5 já aplicado a uma imagem de poucos detalhes.

Kernel 6: Observamos, com uma imagem detalhada, que há um resultado borrado, parecido com anteriores.



Figura 8: Kernel 6 aplicado a uma imagem.

Esse efeito é de Simple box blur.

Kernel 7: Aqui temos um kernel de detecção de diagonais.



Figura 9: Kernel 7 aplicado a uma figura.

Kernel 8: Semelhante ao 7, mas para o outro lado.



Figura 10: Kernel 8 aplicado a uma imagem.

Kernel 9: Novamente temos um kernel que borra a imagem, ficando mais fácil de visualizar com uma imagem de formas geométricas.

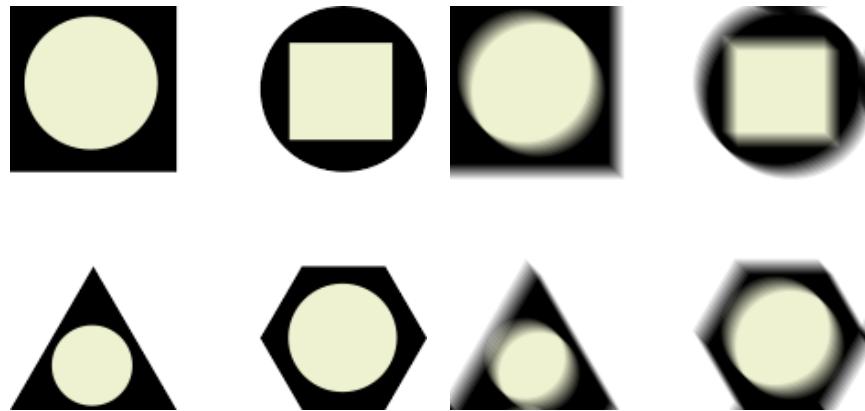


Figura 11: Kernel 9 aplicado a uma imagem com poucos detalhes.

Kernel 10: Este, especialmente fica mais fácil de visualizar com imagens com mais detalhes, pois esta faz realce de detalhes e cores.



Figura 12: Kernel 10 aplicado a uma imagem.

Kernel 11: Novamente, temos um kernel que detecta bordas nos sentidos inferiores e direitos da imagem.

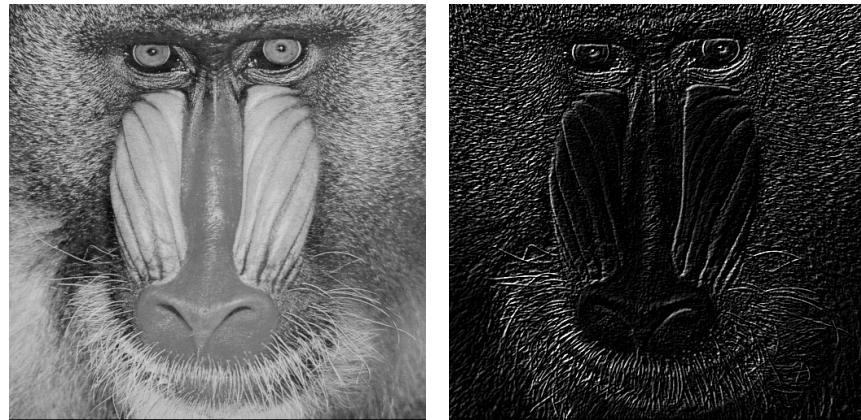


Figura 13: Kernel 11 aplicado a uma imagem detalhada.

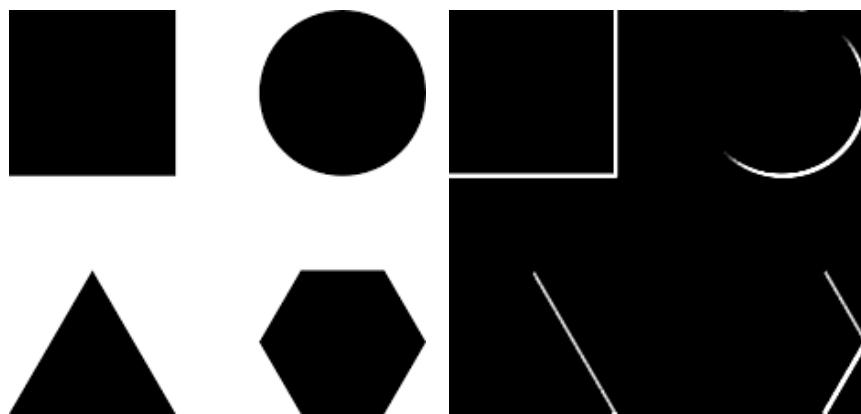


Figura 14: Kernel 11 aplicado a uma imagem de poucos detalhes.

Combinação entre o 3 e o 4: Este fica a combinação das duas imagens, traçando as linhas inferiores e direita.



Figura 15: Combinação entre o Kernel 3 e 4 aplicada a uma imagem detalhada.

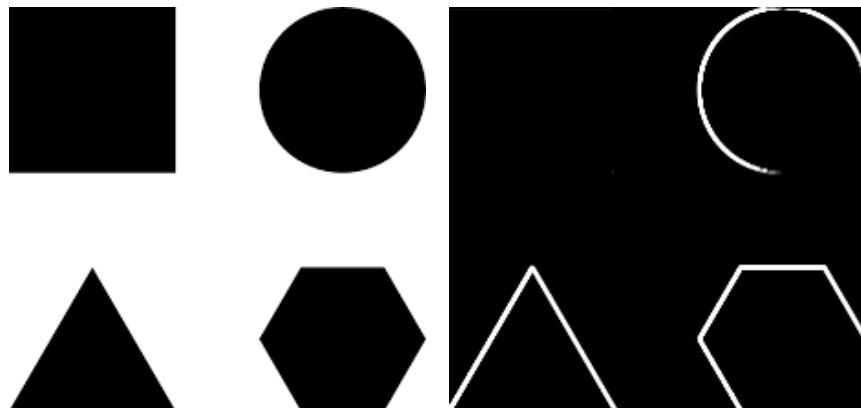


Figura 16: Combinação entre o Kernel 3 e 4 aplicada a uma imagem de poucos detalhes.

5 Referências:

<https://www.ic.unicamp.br/~afalcao/mo443/slides-aula6.pdf> acesso em 19 de outubro de 2020. <https://aishack.in/tutorials/image-convolution-examples/> acesso em 19 de outubro de 2020. [https://en.wikipedia.org/wiki/Kernel_\(image_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing)) acesso em 19 de outubro de 2020.

6 Dificuldades:

Algumas imagens apresentaram overflow, criando imagens muito escuras ou muito claras, necessitando uma normalização para que o valor das cores se adeque as representações numéricas das cores.