

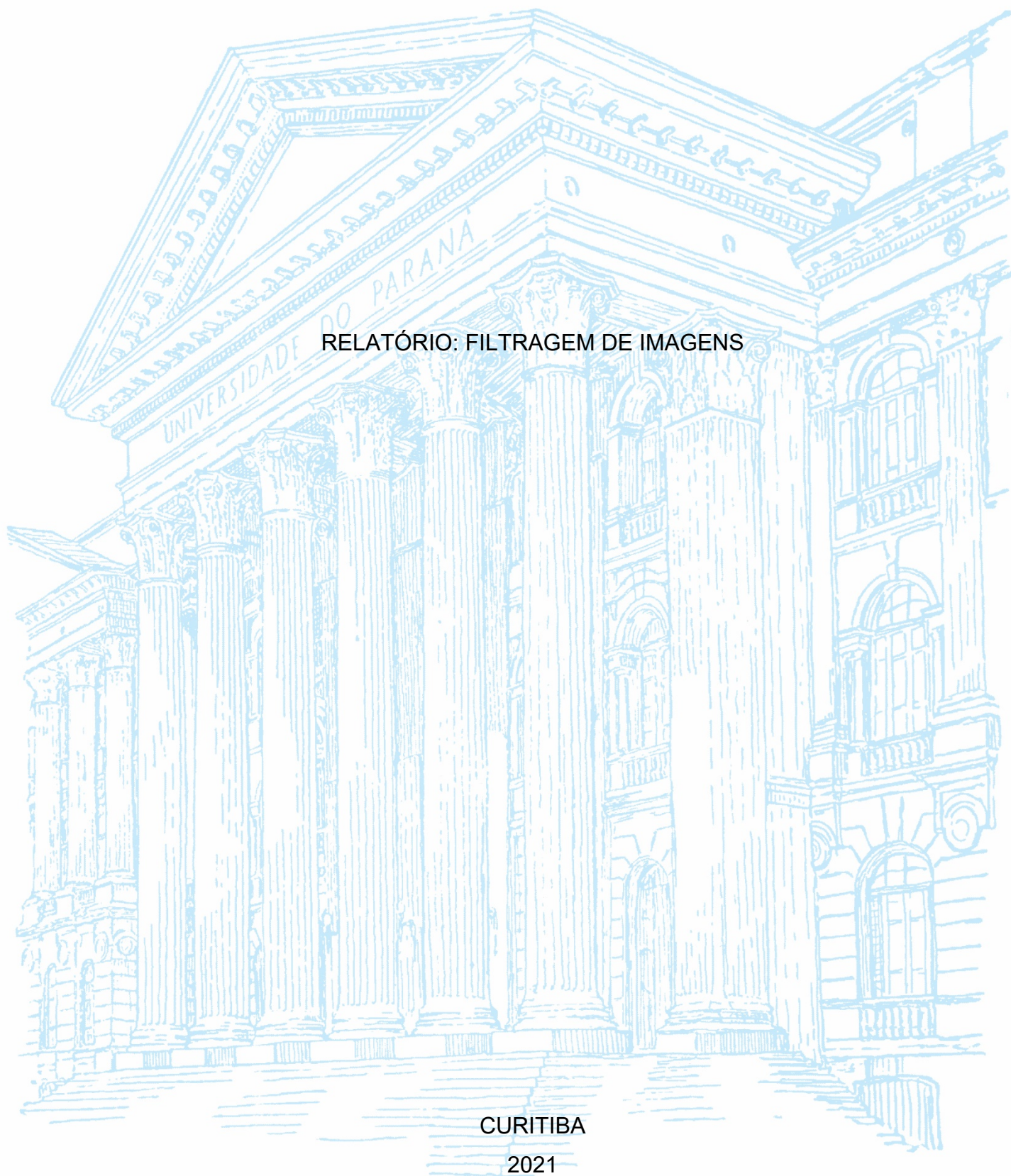
UNIVERSIDADE FEDERAL DO PARANÁ

MATEUS FELIPE DE CÁSSIO FERREIRA

RELATÓRIO: FILTRAGEM DE IMAGENS

CURITIBA

2021



MATEUS FELIPE DE CÁSSIO FERREIRA

RELATÓRIO: FILTRAGEM DE IMAGENS

Relatório apresentado como requisito parcial à conclusão da disciplina CI394 – Processamento de Imagens, no Curso de Bacharelado em Informática Biomédica, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Orientador: Prof. Dr. Luiz Eduardo Soares de Oliveira

CURITIBA

2021

SUMÁRIO

1 INTRODUÇÃO	3
2 EXECUÇÃO	4
3 RESULTADOS.....	5
3.1 MÉDIA.....	5
3.2 MEDIANA.....	5
3.3 EMPILHAMENTO DE IMAGENS	6
3.4 LAPLACIANO E SOBEL.....	7
4 CONCLUSÃO	9

1 INTRODUÇÃO

Este relatório busca apresentar os resultados obtidos referentes ao terceiro Laboratório da disciplina de CI394 – Processamento de Imagens, que consiste em testar e comparar diferentes tipos de filtros para a remoção de ruído "*salt and pepper*".

O objetivo deste relatório é o de analisar o desempenho de filtros na redução de ruído aleatório aplicado sobre uma imagem de modelo. Foram aplicados diferentes níveis de ruído [0.01, 0.02, 0.05, 0.07, 0.1] a uma imagem e, então, utilizado alguns filtros para reduzir o ruído provocado. Os métodos de filtragem implementados foram:

- Média (filter2D);
- Média (blur);
- Média (GaussianBlur);
- Média (boxFilter);
- Mediana;
- Empilhamento de Imagens;
- Método de Laplace;
- Método de Sobel.

Para os filtros das médias, mediana, Laplace e Sobel, foi testado diferentes tamanhos de máscara (*kernel*). Por outro lado, para o filtro do empilhamento de imagens, foi testado diferentes números de imagens ruidosas a serem empilhadas, com o mesmo nível de ruído cada uma delas.

Por fim, para classificar o melhor filtro para a redução do ruído provocado, foi utilizado a métrica PSNR (*Peak Signal-to-Noise Ratio*). Assim, quanto maior o valor do PSNR, mais similares são as duas imagens.

2 EXECUÇÃO

O *script* filtro.py implementa a geração de ruído aleatório (sp_noise), que foi integrada ao código pela especificação do trabalho, e os métodos de filtragem. Os parâmetros para a execução correta do código são:

PARÂMETRO 1: imagem original

PARÂMETRO 2: nível de ruído a ser aplicado

PARÂMETRO 3: filtro a ser aplicado

- 0 - Média (filter2D);
- 0.1 - Média (blur);
- 0.2 - Média (GaussianBlur);
- 0.3 - Média (boxFilter);
- 1 - Mediana;
- 2 - Empilhamento de Imagens;
- 3 - Método de Laplace;
- 4 - Método de Sobel.

PARÂMETRO 4: imagem de saída

Um exemplo de execução é:

```
python3 filtro.py original.png 0.1 2 saida.png
```

3 RESULTADOS

Nesta seção serão apresentados os resultados obtidos para diferentes filtros implementados no script filtro.py.

3.1 MÉDIA

A Figura 1 apresenta um conjunto de tabelas que reúne os resultados obtidos. Percebe-se que até o nível de ruído 0,05, o filtro do **GaussianBlur** foi o que obteve um melhor resultado comparativamente com os demais filtros. Acima desse nível de ruído, o filtro que mais obteve um valor maior de PSNR foi o **filter2D**. Ainda, nota-se que os maiores valores de PSNR estão concentrados em valores de máscaras menores. Fazendo uma média dos maiores valores de máscara encontrados em diferentes situações de ruído (3, 5, 7, 7, 9), e arredondando o valor para cima, chegamos a um valor médio de máscara de 7 (esse valor foi usado como padrão no script implementado).

FIGURA 1 – RESULTADOS OBTIDOS PARA DIFERENTES INTENSIDADES DE RUÍDO E EM DIFERENTES FILTROS

MÉDIA: noise_level = 0,01					MÉDIA: noise_level = 0,02					MÉDIA: noise_level = 0,05					MÉDIA: noise_level = 0,07					MÉDIA: noise_level = 0,1				
kernel size	filter2D PSNR	blur PSNR	GaussianBlur PSNR	boxFilter PSNR	kernel size	filter2D PSNR	blur PSNR	GaussianBlur PSNR	boxFilter PSNR	kernel size	filter2D PSNR	blur PSNR	GaussianBlur PSNR	boxFilter PSNR	kernel size	filter2D PSNR	blur PSNR	GaussianBlur PSNR	boxFilter PSNR	kernel size	filter2D PSNR	blur PSNR	GaussianBlur PSNR	boxFilter PSNR
3	25,42	25,64	26,48	25,63	3	24,45	24,49	24,88	24,49	3	22,14	22,04	21,87	22,03	3	20,92	20,81	20,52	20,80	3	19,47	19,32	18,90	19,34
5	23,65	24,06	25,76	24,05	5	23,64	23,67	24,89	23,68	5	22,80	22,49	22,78	22,48	5	22,21	21,68	21,64	21,67	5	21,10	20,51	20,15	20,53
7	22,70	23,36	24,85	23,35	7	22,86	23,14	24,39	23,14	7	22,72	22,34	22,99	22,33	7	22,40	21,72	22,09	21,71	7	21,76	20,71	20,79	20,74
9	22,19	22,88	24,37	22,88	9	22,31	22,73	24,03	22,73	9	22,44	22,10	22,92	22,10	9	22,21	21,57	22,14	21,56	9	22,03	20,68	20,95	20,70
11	21,83	22,52	23,97	22,52	11	21,93	22,40	23,71	22,40	11	22,00	21,87	22,78	21,86	11	22,04	21,39	22,09	21,39	11	21,96	20,57	21,00	20,60
13	21,50	22,24	23,62	22,24	13	21,69	22,14	23,41	22,13	13	21,73	21,66	22,62	21,66	13	21,74	21,22	22,00	21,22	13	21,68	20,46	20,98	20,48
15	21,25	22,01	23,36	22,01	15	21,41	21,92	23,18	21,92	15	21,43	21,48	22,47	21,48	15	21,43	21,07	21,90	21,07	15	21,25	20,34	20,94	20,36
17	21,03	21,81	23,13	21,81	17	21,21	21,73	22,98	21,73	17	21,26	21,32	22,33	21,32	17	21,15	20,93	21,80	20,93	17	20,95	20,23	20,88	20,25
19	20,83	21,64	22,92	21,64	19	21,00	21,56	22,78	21,56	19	21,10	21,18	22,19	21,18	19	20,84	20,81	21,68	20,81	19	20,99	20,13	20,81	20,15
21	20,58	21,49	22,77	21,49	21	20,74	21,41	22,64	21,41	21	20,91	21,05	22,08	21,04	21	20,72	20,69	21,60	20,69	21	20,86	20,04	20,75	20,06
23	20,39	21,35	22,59	21,35	23	20,59	21,28	22,47	21,28	23	20,58	20,93	21,95	20,92	23	20,58	20,58	21,49	20,58	23	20,76	19,95	20,67	19,97
25	20,25	21,23	22,46	21,23	25	20,44	21,16	22,35	21,16	25	20,48	20,82	21,85	20,81	25	20,49	20,48	21,41	20,48	25	20,64	19,87	20,61	19,88
27	20,12	21,11	22,34	21,11	27	20,30	21,04	22,23	21,05	27	20,32	20,72	21,75	20,71	27	20,37	20,39	21,32	20,39	27	20,46	19,79	20,55	19,81
29	20,01	21,01	22,23	21,00	29	20,18	20,94	22,13	20,94	29	20,13	20,63	21,66	20,62	29	20,29	20,30	21,25	20,30	29	20,32	19,71	20,49	19,73
31	19,88	20,91	22,12	20,91	31	20,06	20,85	22,02	20,85	31	19,98	20,54	21,57	20,53	31	20,20	20,22	21,17	20,22	31	20,17	19,64	20,43	19,67
33	19,78	20,82	22,04	20,82	33	19,89	20,76	21,94	20,76	33	19,79	20,46	21,51	20,45	33	20,09	20,15	21,11	20,15	33	19,98	19,58	20,38	19,60
35	19,65	20,74	21,93	20,73	35	19,77	20,68	21,84	20,68	35	19,62	20,38	21,42	20,37	35	20,01	20,08	21,03	20,07	35	19,80	19,52	20,32	19,54

FONTE: O autor (2021).

3.2 MEDIANA

A Figura 2 apresenta os resultados obtidos para o filtro da mediana. Como esperado, esse tipo de filtro é particularmente eficiente para eliminar ruídos do tipo “salt and pepper” e isso foi observado com os maiores valores de PSNR quando comparados com o filtro da média, para todos os níveis de ruído propostos. Além disso, nota-se que esse maior valor de PSNR está concentrado no tamanho de máscara igual a 3 para todos os níveis de ruído.

FIGURA 2 – RESULTADOS OBTIDOS PARA O FILTRO DA MEDIANA

noise level —— kernel size	MEDIANA				
	0,01	0,02	0,05	0,07	0,1
3	27,23	27,13	26,71	26,30	25,08
5	24,59	24,56	24,48	24,42	24,29
7	23,70	23,69	23,65	23,63	23,55
9	23,17	23,16	23,14	23,12	23,08
11	22,79	22,79	22,77	22,76	22,72
13	22,50	22,49	22,48	22,47	22,45
15	22,26	22,26	22,25	22,25	22,22
17	22,07	22,06	22,06	22,05	22,03
19	21,90	21,89	21,89	21,88	21,86
21	21,74	21,74	21,73	21,73	21,71
23	21,60	21,60	21,60	21,59	21,58
25	21,48	21,47	21,47	21,47	21,46
27	21,37	21,36	21,36	21,36	21,35
29	21,27	21,26	21,26	21,26	21,25
31	21,17	21,17	21,17	21,17	21,17
33	21,09	21,09	21,08	21,08	21,09
35	21,01	21,01	21,01	21,00	21,01

FONTE: O autor (2021).

3.3 EMPILHAMENTO DE IMAGENS

A Figura 3 apresenta os resultados obtidos para o filtro de empilhamento de imagens. Os resultados foram obtidos para diferentes níveis de ruído e utilizando diferentes conjuntos de imagens ruidosas para fazer o empilhamento das imagens. Nota-se que quanto maior o valor de n , maior é o valor de PSNR. Esse filtro foi o que mais obteve valores maiores dessa métrica comparativamente com os outros filtros propostos para um valor de $n = 200$.

Apesar do fato que de maiores valores de n conseguem reduzir ainda mais o ruído das imagens empilhadas, nesse intervalo apresentado [10, 200], existe um valor médio em que, a partir desse valor de n , o valor de PSNR não varia significativamente. O valor escolhido como padrão para o número de imagens a serem empilhadas é $n = 120$.

FIGURA 3 – RESULTADOS OBTIDOS PARA O FILTRO DE EMPILHAMENTO DE IMAGENS

EMPILHAMENTO					
noise level					
n	0,01	0,02	0,05	0,07	0,1
imagens					
10	31,751	28,580	24,173	22,430	20,513
20	34,575	31,236	26,431	24,507	22,323
30	36,070	32,647	27,561	25,496	23,171
40	37,063	33,553	28,252	26,099	23,652
50	37,820	34,214	28,719	26,478	23,964
60	38,501	34,766	29,085	26,768	24,186
70	38,929	35,148	29,356	26,987	24,358
80	39,339	35,469	29,570	27,154	24,484
90	39,677	35,759	29,746	27,304	24,600
100	39,918	35,988	29,883	27,405	24,680
110	40,328	36,210	30,004	27,494	24,747
120	40,466	36,376	30,109	27,582	24,812
130	40,631	36,531	30,190	27,658	24,864
140	40,817	36,665	30,264	27,712	24,910
150	40,956	36,783	30,337	27,771	24,947
160	41,111	36,885	30,394	27,815	24,980
170	41,237	36,993	30,453	27,849	25,007
180	41,335	37,087	30,508	27,892	25,041
190	41,473	37,157	30,547	27,924	25,056
200	41,612	37,238	30,592	27,957	25,082

FONTE: O autor (2021).

3.4 LAPLACIANO E SOBEL

A Figura 4 apresenta os resultados obtidos pelo Método de Laplace e o Método de Sobel. Sabe-se que esses dois filtros são ideias para o caso de realce em imagens, mais especificamente para o desafio de detecção de bordas. Em função da imagem de teste definida por padrão ser o de uma paisagem, com bordas não muito definidas, esse fato pode explicar o baixo desempenho desses dois métodos quando aplicados nesse contexto do experimento.

A Figura 5 mostra um exemplo de saída do Método de Laplace, utilizando uma imagem ruidosa com `noise_level = 0,07` e com um valor de `kernel_size = 3`.

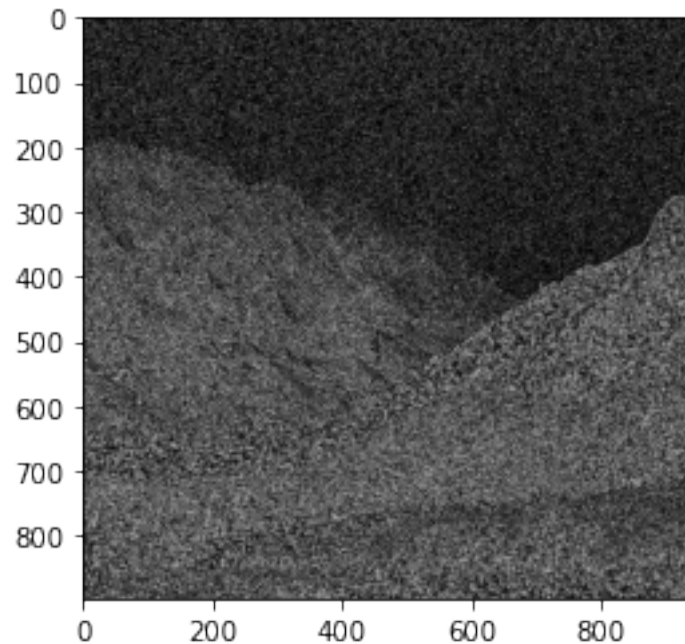
FIGURA 4 – RESULTADOS OBTIDOS PARA O FILTRO DE LAPLACE E SOBEL

LAPLACIANO					
noise level kernel size	0,01	0,02	0,05	0,07	0,1
3	4,37	4,38	4,41	4,41	4,42
5	3,93	4,07	4,25	4,29	4,35
7	4,12	4,10	4,13	4,22	4,36
9	4,11	4,01	4,13	4,26	4,42
11	4,08	3,99	4,17	4,31	4,45
13	4,06	3,99	4,21	4,34	4,47
15	4,02	4,00	4,25	4,36	4,47
17	4,29	4,20	4,34	4,36	4,36
19	4,39	4,18	4,13	4,13	4,12
21	4,12	4,10	4,10	4,10	4,10
23	4,10	4,10	4,10	4,10	4,10
25	4,10	4,10	4,10	4,10	4,10
27	4,10	4,10	4,10	4,10	4,10
29	4,10	4,10	4,10	4,10	4,10
31	4,10	4,10	4,10	4,10	4,10

SOBEL					
noise level kernel size	0,01	0,02	0,05	0,07	0,1
3	4,68	4,77	4,96	5,05	5,16
5	4,50	4,68	4,93	5,00	5,03
7	5,10	5,06	5,00	4,99	4,98
9	5,20	5,09	5,00	4,98	4,98
11	5,19	5,08	5,00	4,98	4,98
13	5,18	5,07	5,00	4,98	4,99
15	5,17	5,07	5,00	4,99	5,00
17	5,05	4,74	4,43	4,36	4,31
19	4,27	4,16	4,12	4,11	4,11
21	4,11	4,10	4,10	4,10	4,10
23	4,10	4,10	4,10	4,10	4,10
25	4,10	4,10	4,10	4,10	4,10
27	4,10	4,10	4,10	4,10	4,10
29	4,10	4,10	4,10	4,10	4,10
31	4,10	4,10	4,10	4,10	4,10

FONTE: O autor (2021).

FIGURA 5 – EXEMPLO DE IMAGEM OBTIDA UTILIZANDO O MÉTODO DE LAPLACE



FONTE: O autor (2021).

4 CONCLUSÃO

A partir dos resultados do experimento nota-se que existe uma certa tendência, no geral, de obter maiores valores de PSNR utilizando filtros menores (ou valores de kernels menores).

Dentre os filtros apresentados, aquele que se destacou melhor em reduzir o nível de ruído “*salt and pepper*” foi o empilhamento de imagens. Utilizando um valor de n imagens de 120, esse método conseguiu obter os melhores resultados para os níveis de ruído de 0,01 ; 0,02 ; 0,05 e 0,07. Já para o nível de ruído de 0,1 o método conseguiu atingir um valor de PSNR de 24,812 , enquanto que o método da mediana atingiu o valor de 25,080.