BIRZEIT UNIVERSITY
Electrical and Computer Engineering Department
Computer Vision - ENCS5343
Assignment #1

Sondos Shahin 1200166
Instructor: Dr. Aziz Qaroush
Section 2
Date: Nov 9, 2024

Introduction

The aim of this assignment is to compare the performance of smoothing filters with other advanced filters on noisy images. Three different grey level images were used in the experiment, and for each image different types of noise were added to test the filters' performance. Noises such as gaussian noise and salt & pepper noise were added to the images, each with three different intensities (Low, Medium, High). The targeted filters in the testing were box filter, gaussian filter and median filter for the simple filters, and adaptive mean filter, adaptive median filter and bilateral filter for the advanced filters. Moreover, each filter was tested on different kernel sizes to compare the effect of the size change. Measuring the filters' performance was based on many factors, such as the values of mean squared error (MSE) and peak signal to noise ratio (PSNR), edge preservation and the effect of kernel size.

Experiments

Three different images were used in the experiment, the first step was converting the images to grey level before starting the procedure. The next step was to add different types of noise to the images. Three levels of Gaussian noise were added to each image, their intensities were determined by changing the values of the variance. Another three levels of salt & pepper noise were added to the images, their intensities were determined by varying the probabilities of salt and pepper.

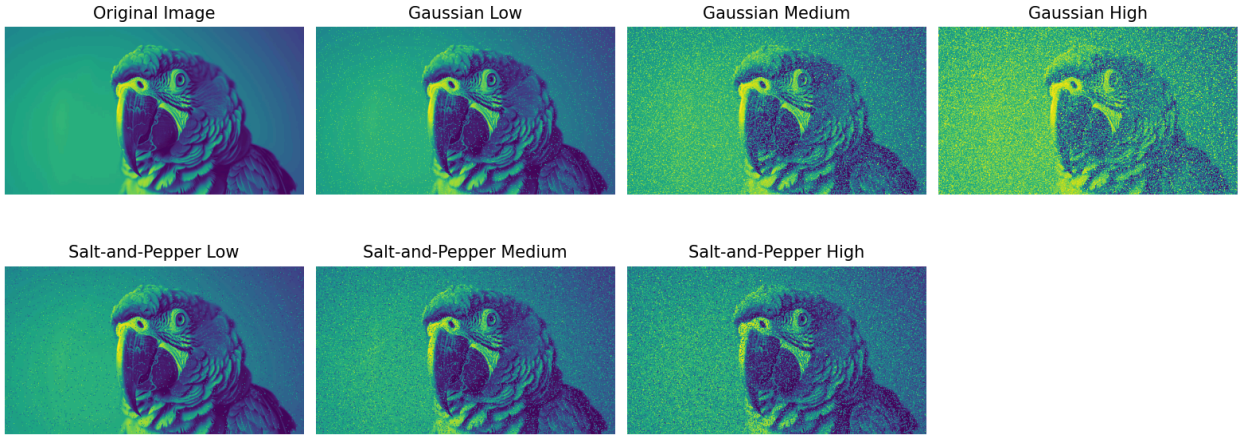


Figure 1: Image 1 with different noises

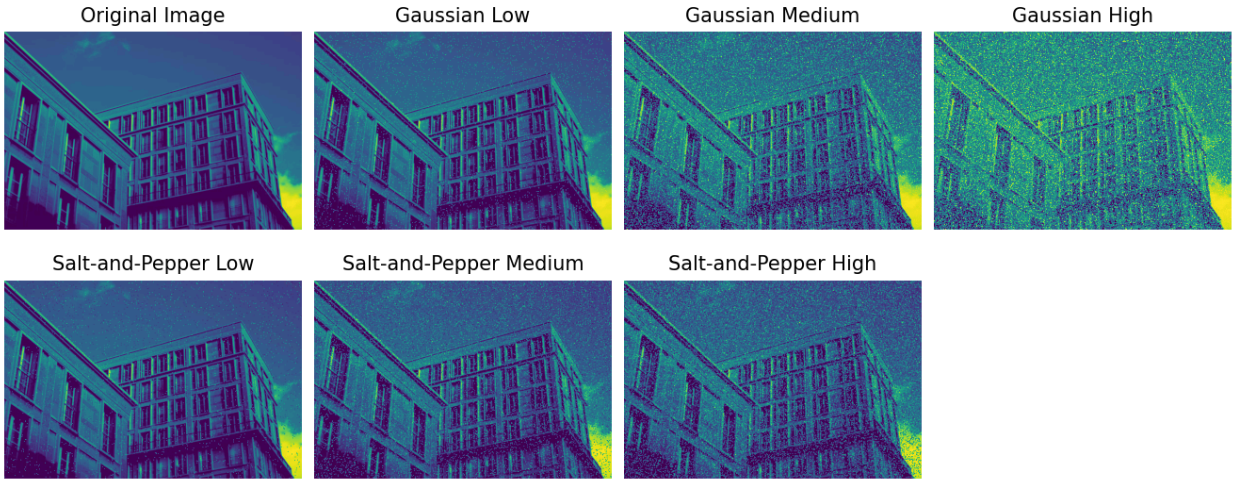


Figure 2: Image 2 with different noises

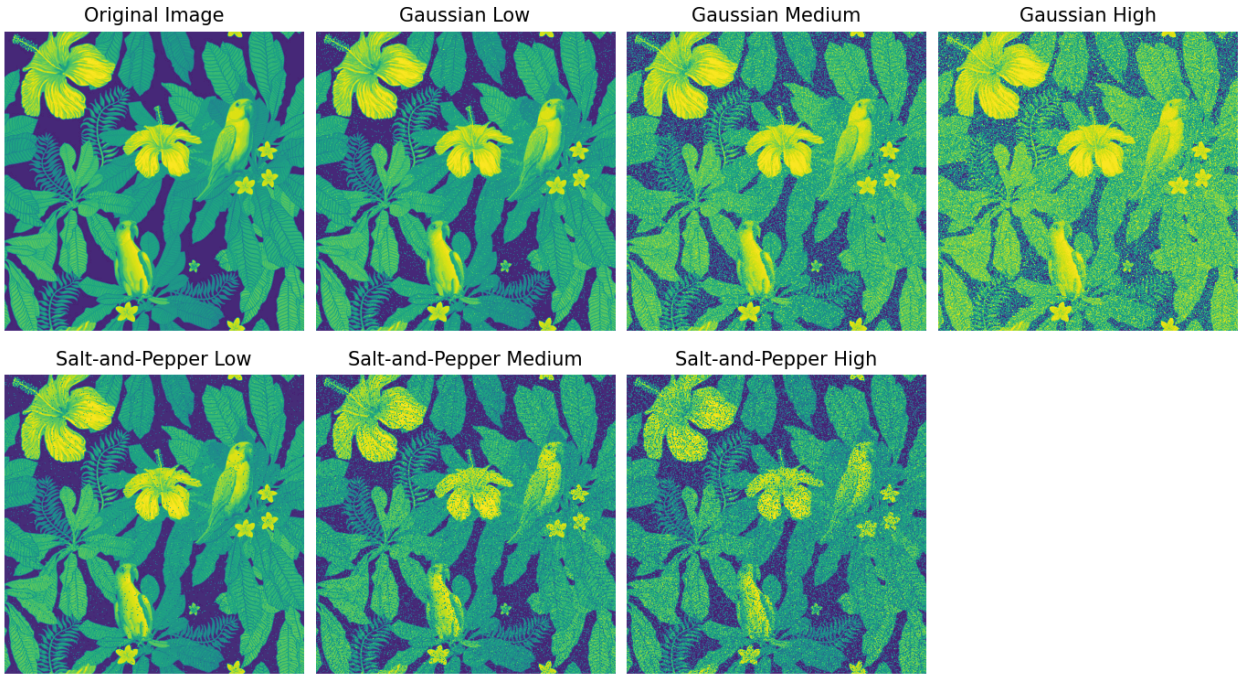


Figure 3: Image 3 with different noises

Multiple smoothing filters were applied to all variations of the images, such as box filter, gaussian filter, median filter, bilateral filter, adaptive mean and adaptive median filters. Each filter was applied three times, each with a different kernel size (3x3, 5x5 and 7x7). For each filter, the mean squared error (MSE) and peak signal to noise ratio (PSNR) were calculated. Moreover, the computation time for each filter was computed for different kernels. Finally, the performance of each filter was evaluated based on their ability to preserve the edges in the images, the used edge detector in the experiments was the Canny edge detector. Padding for the images before applying the filters was not explicitly done except for the adaptive filters, since only these two didn't have an API in openCv. All other filters have the padding implemented in their APIs. For the adaptive mean and median filters, the zero padding was used.

Results and Discussion

The MSE and PSNR for each filter:

Kernel size 3x3:

```
Box filter with kernel 3x3 : image  1
Gaussian Low
mse =  2.404566269224123
PSNR =  44.32043610249615
Gaussian Medium
mse =  17.45362450319682
PSNR =  35.71194732536382
Gaussian High
mse =  34.51823483670295
PSNR =  32.75031781828218
Salt-and-Pepper Low
mse =  2.1364351131847243
PSNR =  44.8339065370475
Salt-and-Pepper Medium
mse =  10.290444098842233
PSNR =  38.00646243101128
Salt-and-Pepper High
mse =  19.392642992915153
PSNR =  35.25443358441255
Gaussian filter with kernel 3x3: image  1
Gaussian Low
mse =  2.404566269224123
PSNR =  44.32043610249615
Gaussian Medium
mse =  17.45362450319682
PSNR =  35.71194732536382
Gaussian High
mse =  34.51823483670295
PSNR =  32.75031781828218
Salt-and-Pepper Low
mse =  2.1364351131847243
PSNR =  44.8339065370475
Salt-and-Pepper Medium
mse =  10.290444098842233
PSNR =  38.00646243101128
Salt-and-Pepper High
mse =  19.392642992915153
PSNR =  35.25443358441255
```

```
Median filter with kernel 3x3: image  1
Gaussian Low
mse =  2.404566269224123
PSNR =  44.32043610249615
Gaussian Medium
mse =  17.45362450319682
PSNR =  35.71194732536382
Gaussian High
mse =  34.51823483670295
PSNR =  32.75031781828218
Salt-and-Pepper Low
mse =  2.1364351131847243
PSNR =  44.8339065370475
Salt-and-Pepper Medium
mse =  10.290444098842233
PSNR =  38.00646243101128
Salt-and-Pepper High
mse =  19.392642992915153
PSNR =  35.25443358441255
Bilateral filter with kernel 3x3: image  1
Gaussian Low
mse =  2.404566269224123
PSNR =  44.32043610249615
Gaussian Medium
mse =  17.45362450319682
PSNR =  35.71194732536382
Gaussian High
mse =  34.51823483670295
PSNR =  32.75031781828218
Salt-and-Pepper Low
mse =  2.1364351131847243
PSNR =  44.8339065370475
Salt-and-Pepper Medium
mse =  10.290444098842233
PSNR =  38.00646243101128
Salt-and-Pepper High
mse =  19.392642992915153
PSNR =  35.25443358441255
```

```
Adaptive Mean filter with kernel 3x3: image  1
Gaussian Low
mse =  2.404566269224123
PSNR =  44.32043610249615
Gaussian Medium
mse =  17.45362450319682
PSNR =  35.71194732536382
Gaussian High
mse =  34.51823483670295
PSNR =  32.75031781828218
Salt-and-Pepper Low
mse =  2.1364351131847243
PSNR =  44.8339065370475
Salt-and-Pepper Medium
mse =  10.290444098842233
PSNR =  38.00646243101128
Salt-and-Pepper High
mse =  19.392642992915153
PSNR =  35.25443358441255
Adaptive Median filter with kernel 3x3: image:  1
Gaussian Low
mse =  2.404566269224123
PSNR =  44.32043610249615
Gaussian Medium
mse =  17.45362450319682
PSNR =  35.71194732536382
Gaussian High
mse =  34.51823483670295
PSNR =  32.75031781828218
Salt-and-Pepper Low
mse =  2.1364351131847243
PSNR =  44.8339065370475
Salt-and-Pepper Medium
mse =  10.290444098842233
PSNR =  38.00646243101128
Salt-and-Pepper High
mse =  19.392642992915153
PSNR =  35.25443358441255
```

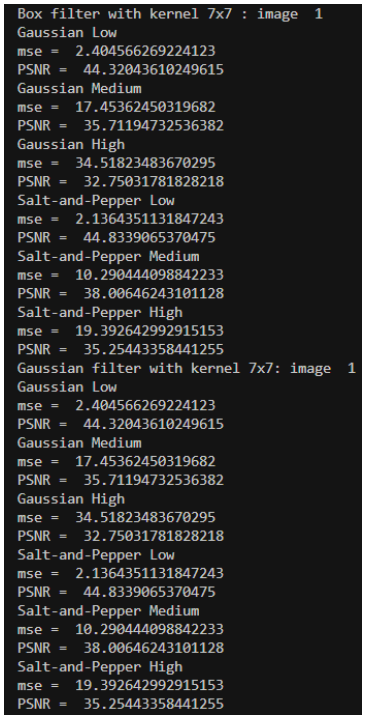
Kernel size 5x5:

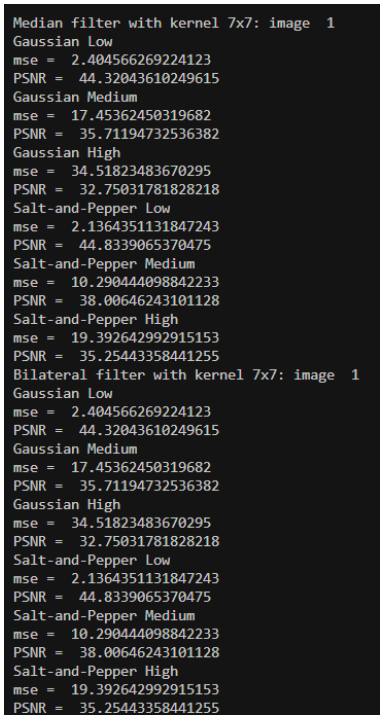
```
Box filter with kernel 5x5 : image  1
Gaussian Low
mse =  2.404566269224123
PSNR =  44.32043610249615
Gaussian Medium
mse =  17.45362450319682
PSNR =  35.71194732536382
Gaussian High
mse =  34.51823483670295
PSNR =  32.75031781828218
Salt-and-Pepper Low
mse =  2.1364351131847243
PSNR =  44.8339065370475
Salt-and-Pepper Medium
mse =  10.290444098842233
PSNR =  38.00646243101128
Salt-and-Pepper High
mse =  19.392642992915153
PSNR =  35.25443358441255
Gaussian filter with kernel 5x5: image  1
Gaussian Low
mse =  2.404566269224123
PSNR =  44.32043610249615
Gaussian Medium
mse =  17.45362450319682
PSNR =  35.71194732536382
Gaussian High
mse =  34.51823483670295
PSNR =  32.75031781828218
Salt-and-Pepper Low
mse =  2.1364351131847243
PSNR =  44.8339065370475
Salt-and-Pepper Medium
mse =  10.290444098842233
PSNR =  38.00646243101128
Salt-and-Pepper High
mse =  19.392642992915153
PSNR =  35.25443358441255
```

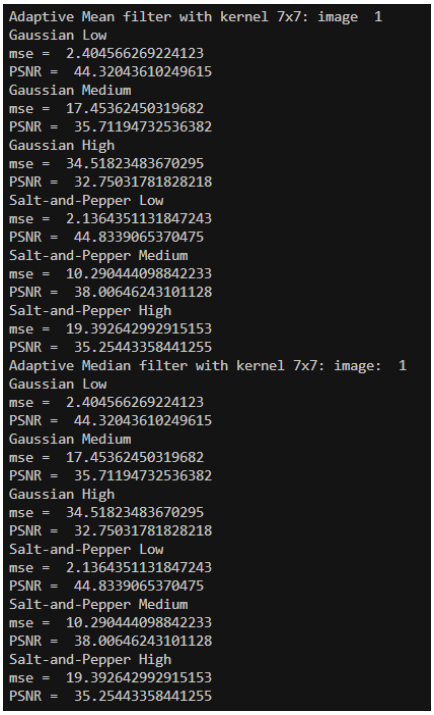
```
Median filter with kernel 5x5: image  1
Gaussian Low
mse =  2.404566269224123
PSNR =  44.32043610249615
Gaussian Medium
mse =  17.45362450319682
PSNR =  35.71194732536382
Gaussian High
mse =  34.51823483670295
PSNR =  32.75031781828218
Salt-and-Pepper Low
mse =  2.1364351131847243
PSNR =  44.8339065370475
Salt-and-Pepper Medium
mse =  10.290444098842233
PSNR =  38.00646243101128
Salt-and-Pepper High
mse =  19.392642992915153
PSNR =  35.25443358441255
Bilateral filter with kernel 5x5: image  1
Gaussian Low
mse =  2.404566269224123
PSNR =  44.32043610249615
Gaussian Medium
mse =  17.45362450319682
PSNR =  35.71194732536382
Gaussian High
mse =  34.51823483670295
PSNR =  32.75031781828218
Salt-and-Pepper Low
mse =  2.1364351131847243
PSNR =  44.8339065370475
Salt-and-Pepper Medium
mse =  10.290444098842233
PSNR =  38.00646243101128
Salt-and-Pepper High
mse =  19.392642992915153
PSNR =  35.25443358441255
```

```
Adaptive Mean filter with kernel 5x5: image  1
Gaussian Low
mse =  2.404566269224123
PSNR =  44.32043610249615
Gaussian Medium
mse =  17.45362450319682
PSNR =  35.71194732536382
Gaussian High
mse =  34.51823483670295
PSNR =  32.75031781828218
Salt-and-Pepper Low
mse =  2.1364351131847243
PSNR =  44.8339065370475
Salt-and-Pepper Medium
mse =  10.290444098842233
PSNR =  38.00646243101128
Salt-and-Pepper High
mse =  19.392642992915153
PSNR =  35.25443358441255
Adaptive Median filter with kernel 5x5: image:  1
Gaussian Low
mse =  2.404566269224123
PSNR =  44.32043610249615
Gaussian Medium
mse =  17.45362450319682
PSNR =  35.71194732536382
Gaussian High
mse =  34.51823483670295
PSNR =  32.75031781828218
Salt-and-Pepper Low
mse =  2.1364351131847243
PSNR =  44.8339065370475
Salt-and-Pepper Medium
mse =  10.290444098842233
PSNR =  38.00646243101128
Salt-and-Pepper High
mse =  19.392642992915153
PSNR =  35.25443358441255
```

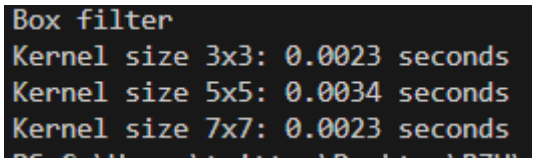
Kernel size 7x7:

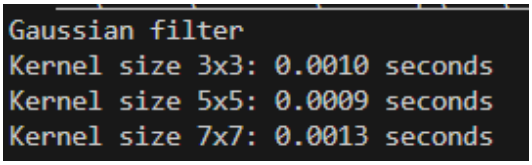


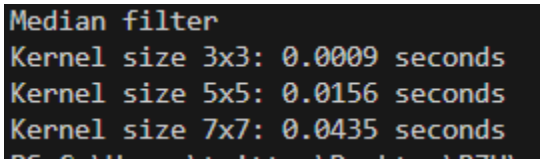


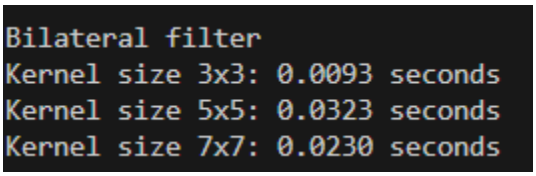


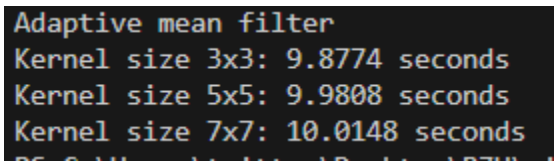
Computation time for each filter:

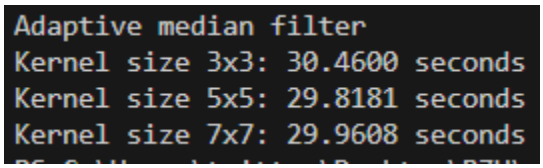








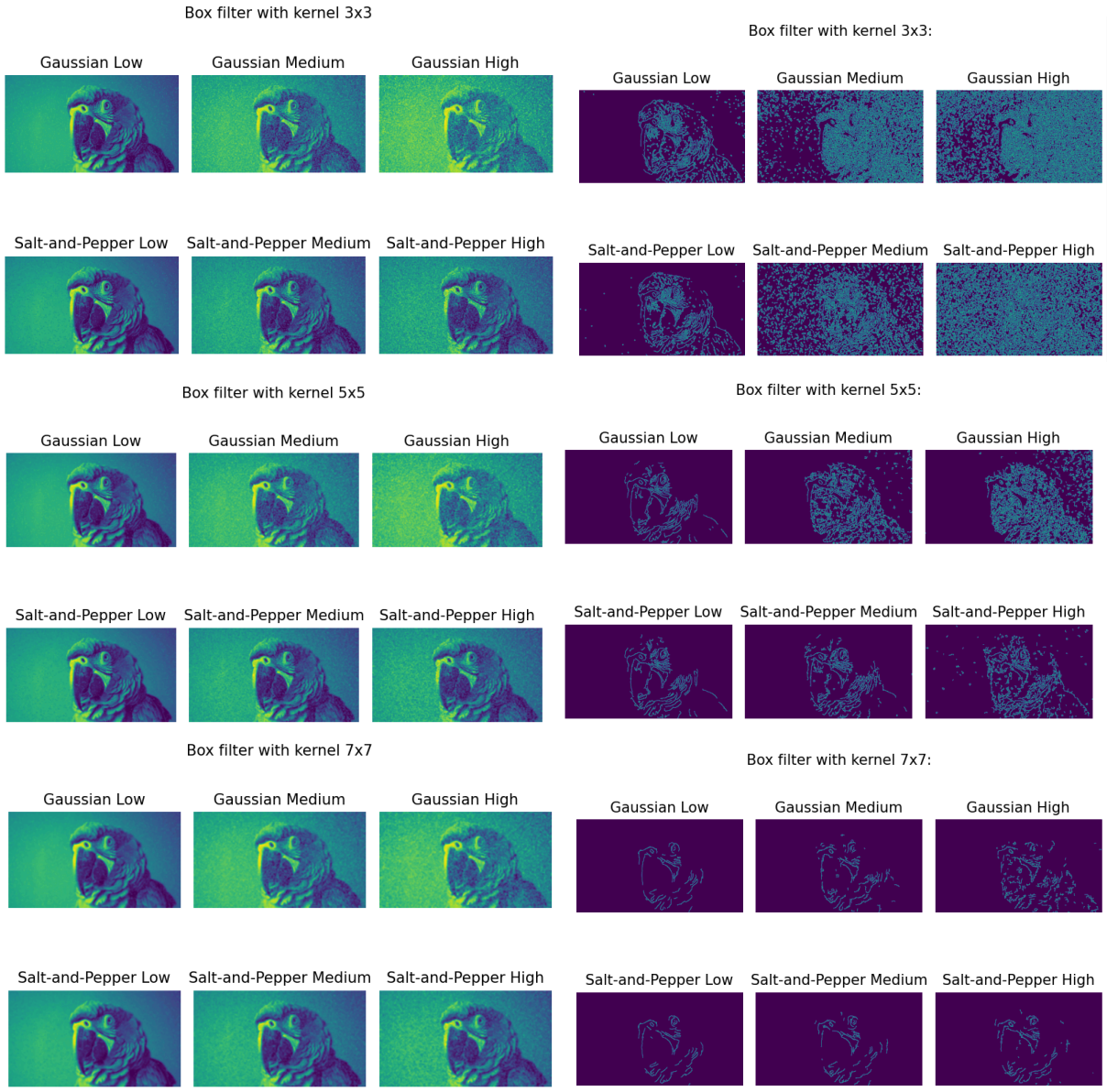




Applying Filters and Edge Detection:

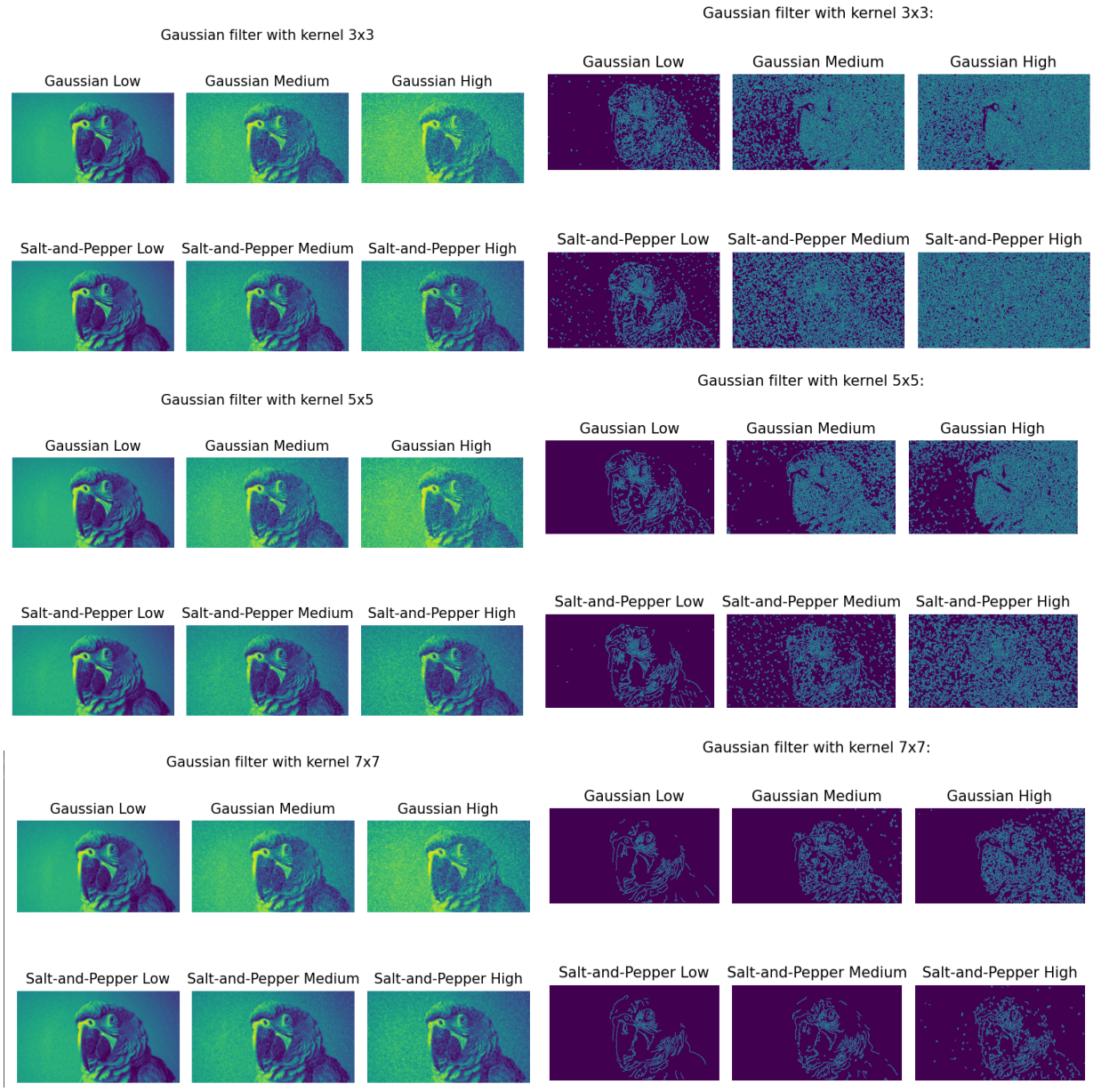
Image 1:

Box filter:



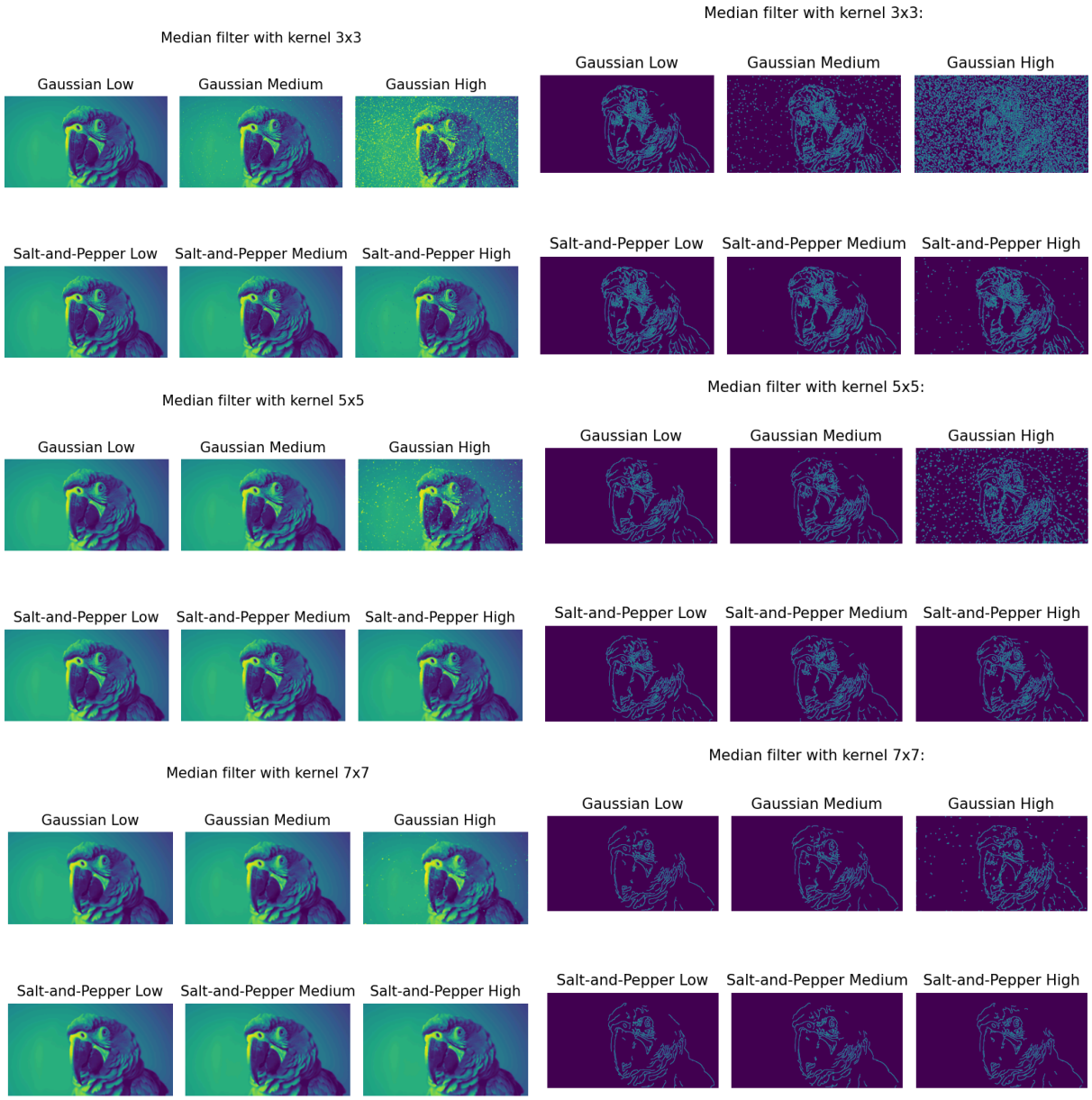
As noticed in the figures, the box filter did well when the noise intensity was low for both the gaussian and the salt & pepper noises for the 3x3 kernel, but was very bad when the noise intensity rose. The 5x5 kernel did a little better on high intensity noise, and the 7x7 kernel caused a loss in many details from the image.

Gaussian filter:



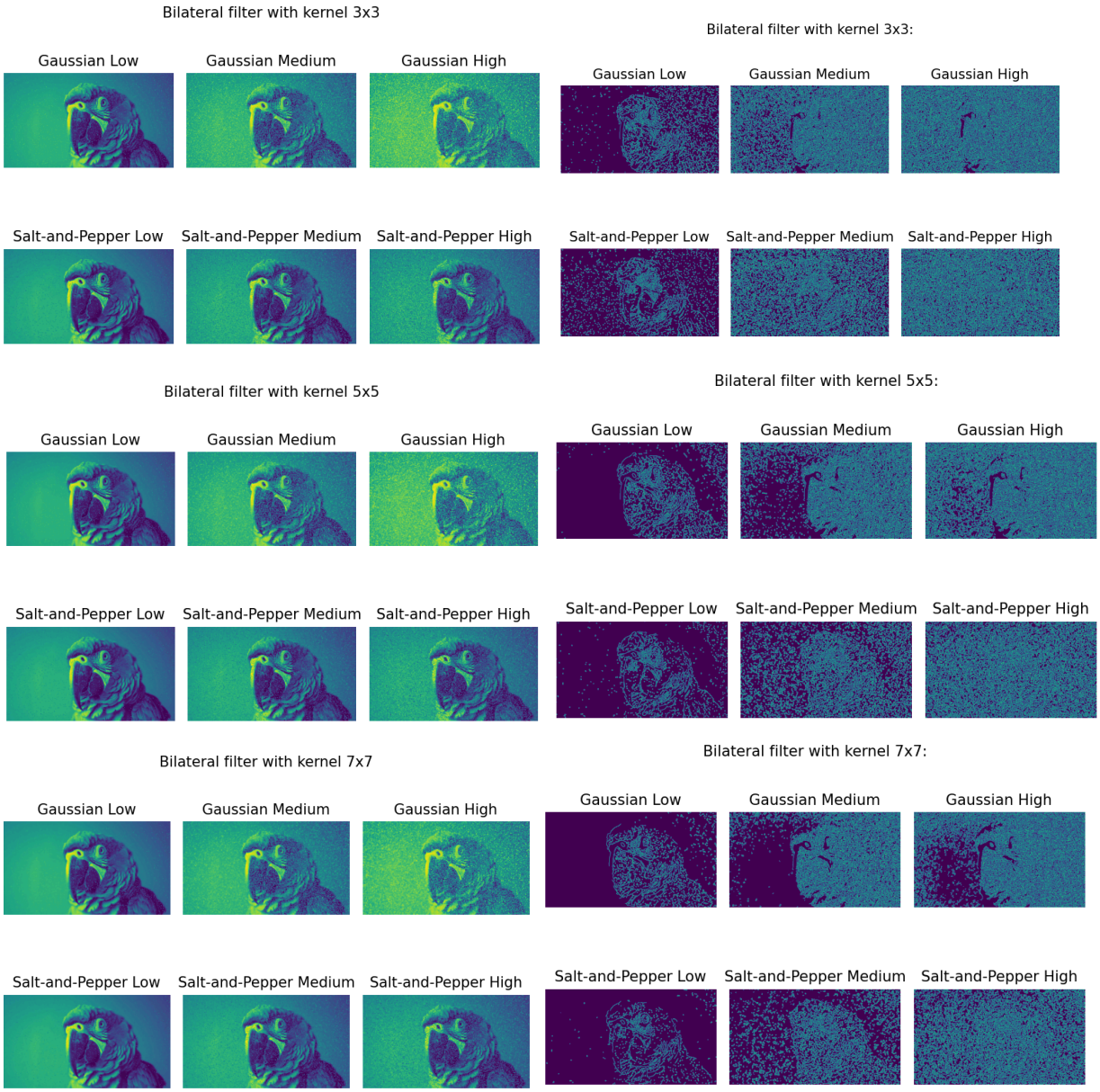
The gaussian filter with 3x3 kernel worked only for low intensity noises, kernel 5x5 and 7x7 did better on the higher intensities but still not good enough.

Median filter:



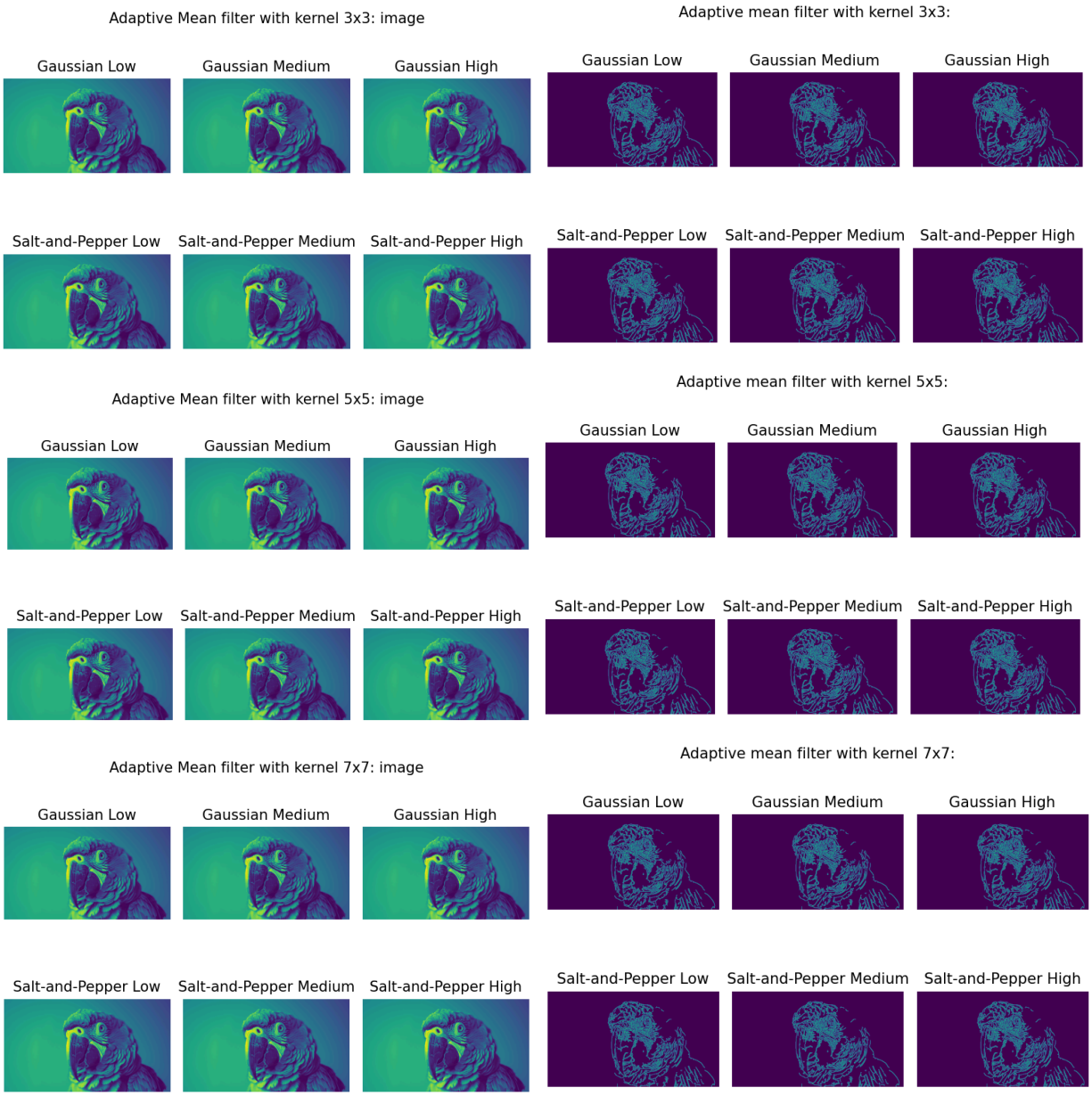
Kernel 3x3 failed completely when the noise was gaussian with high intensity, but did well for other noises. Kernel 5x5 did better on all types of noise, while kernel 7x7 worked well but caused some details to be lost.

Bilateral filter:



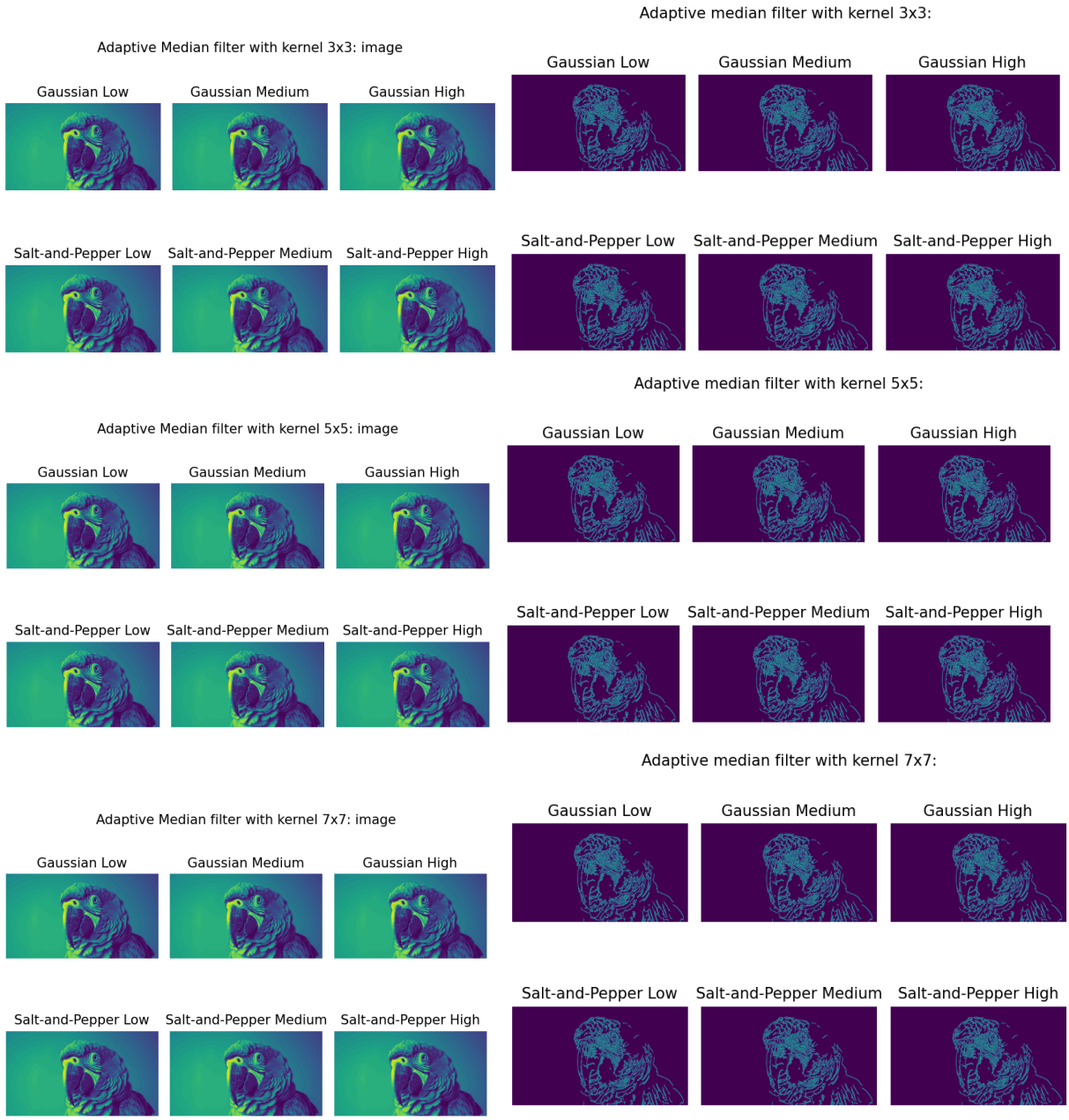
The bilateral filter with 3x3, 5x5 and 7x7 kernel sizes did very poorly in preserving the edges of the medium and high intensity noisy images, while working fine for low intensity noise.

Adaptive mean filter:



The adaptive mean filter has shown the best results in edge preservation so far, for all used kernels and with all intensities of the gaussian and salt & pepper noises. However, the cost was high computational time.

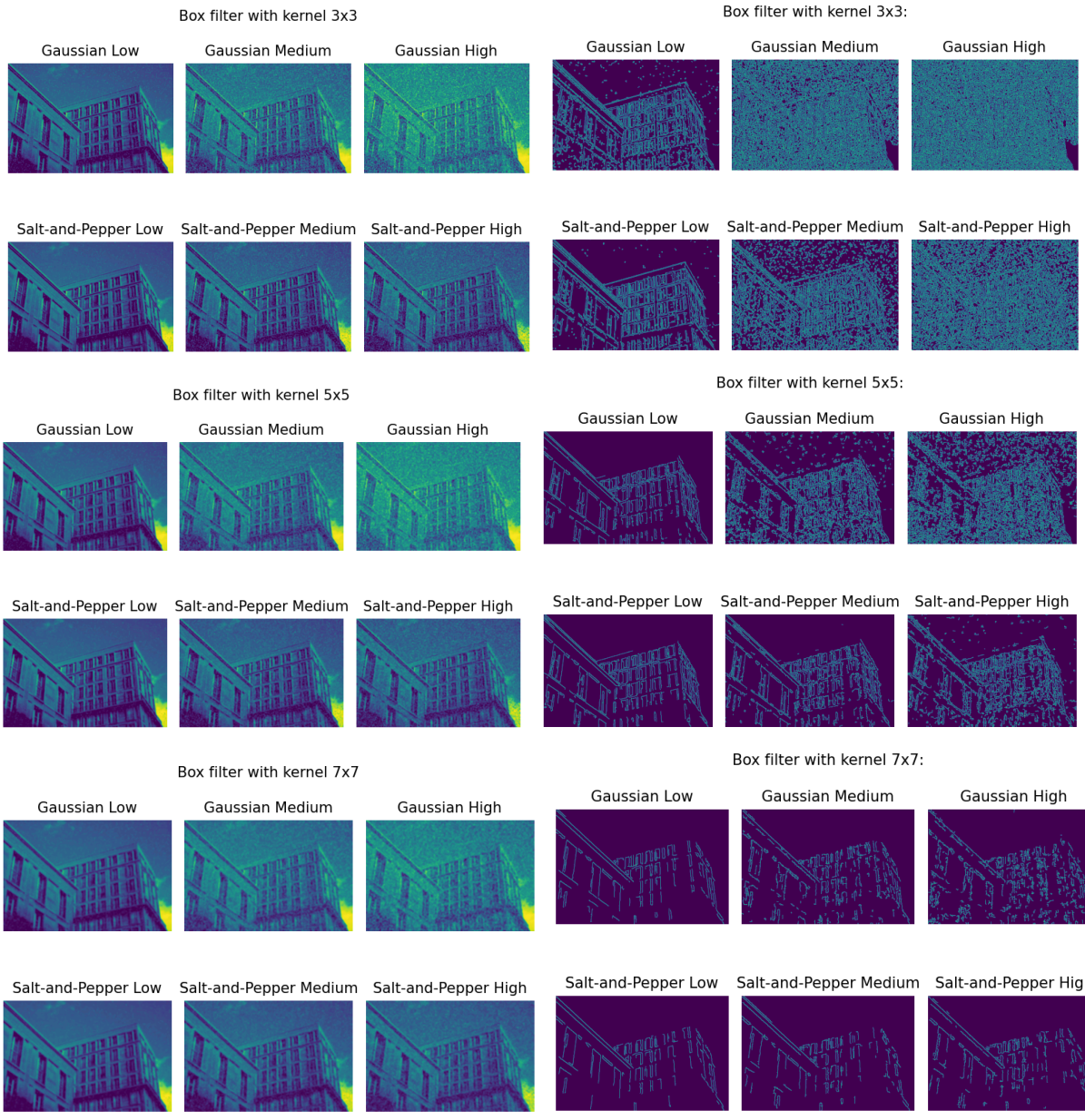
Adaptive median filter:



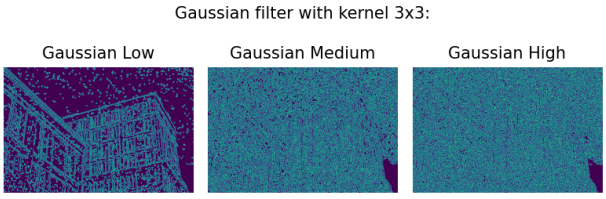
The adaptive median filter has shown similar results in edge preservation as the adaptive mean filter. However, the computational time in the adaptive median was even more than the adaptive mean filter.

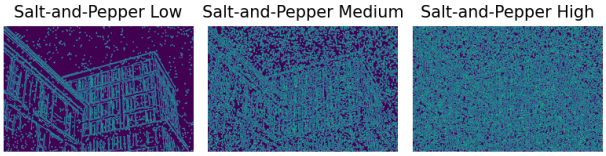
Image 2:

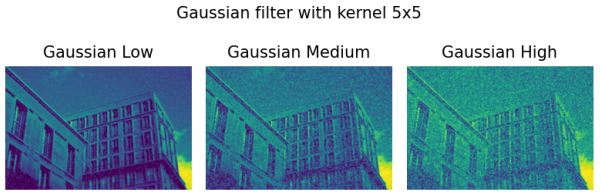
Box filter:

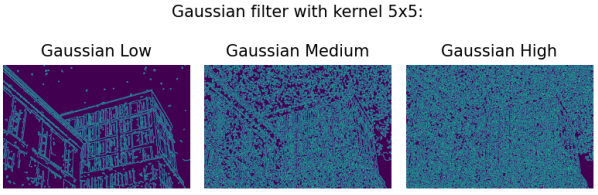


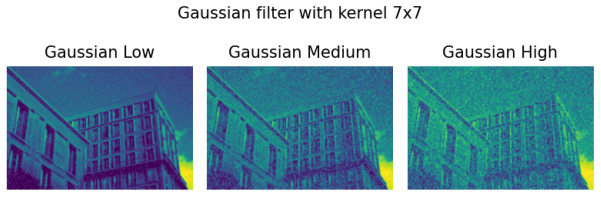
Gaussian filter:



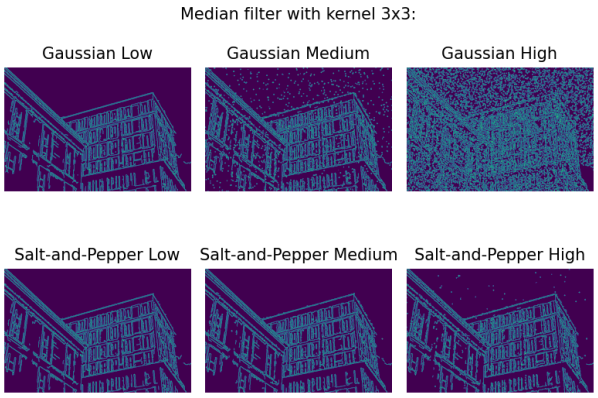


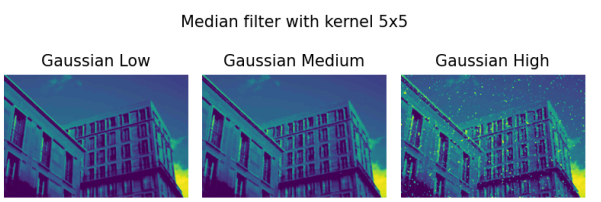


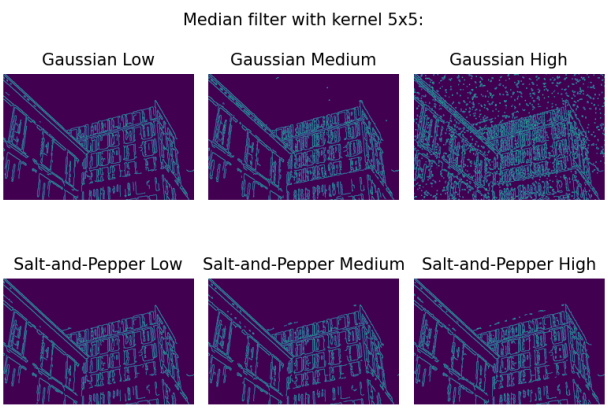


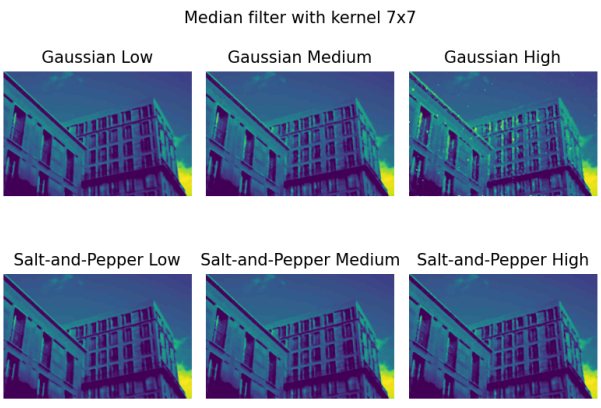


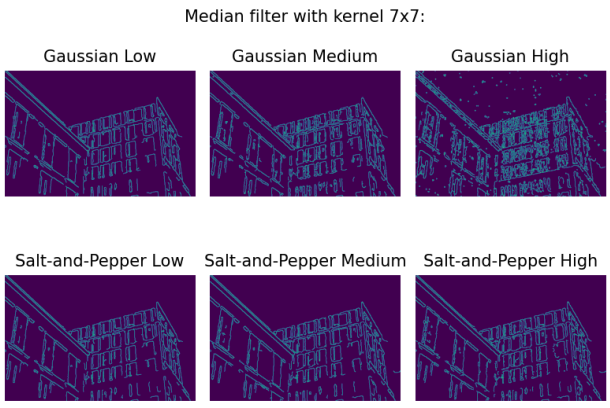
Median filter:



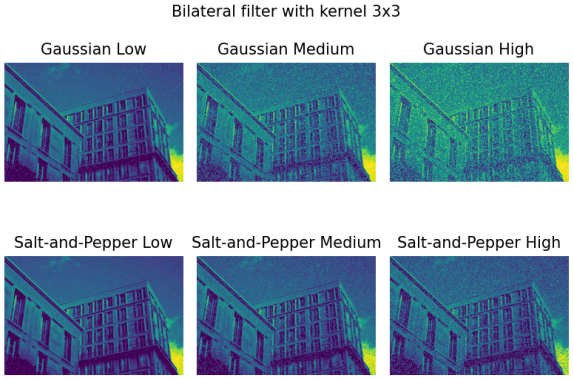


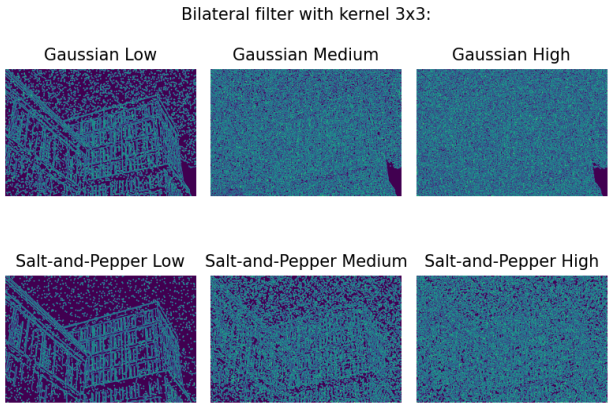


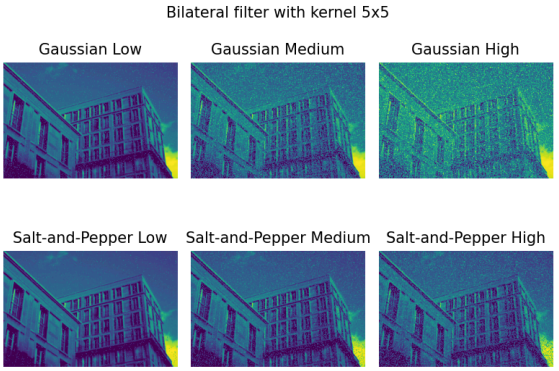


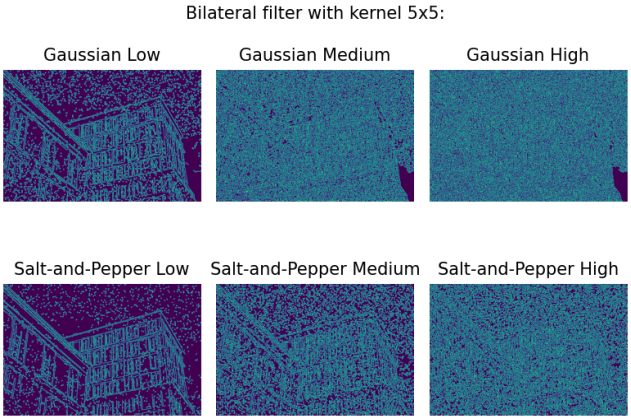


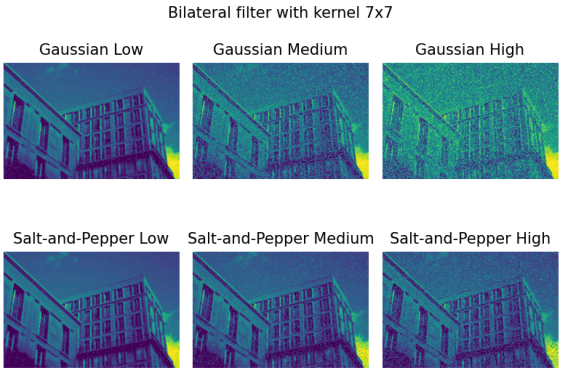
Bilateral filter:

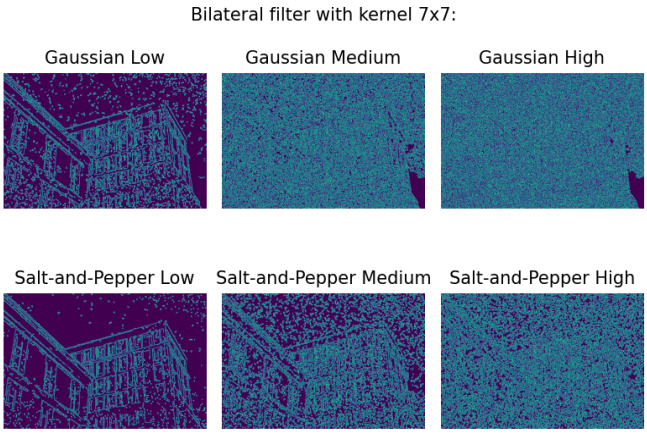




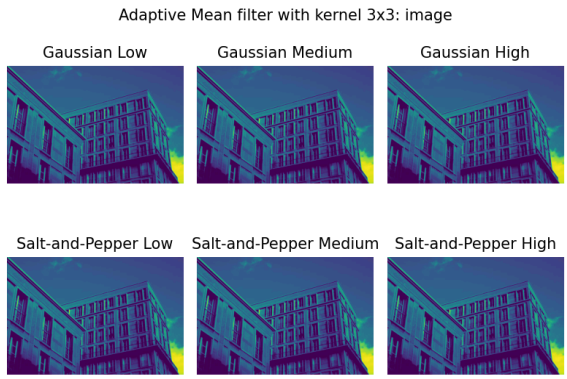


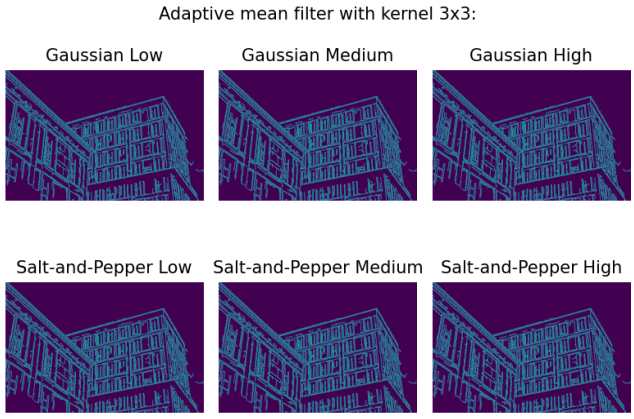


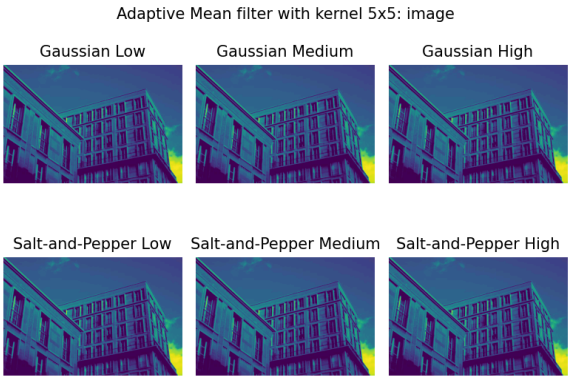


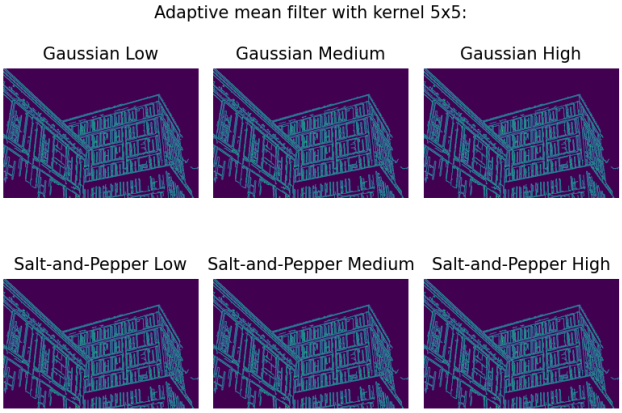


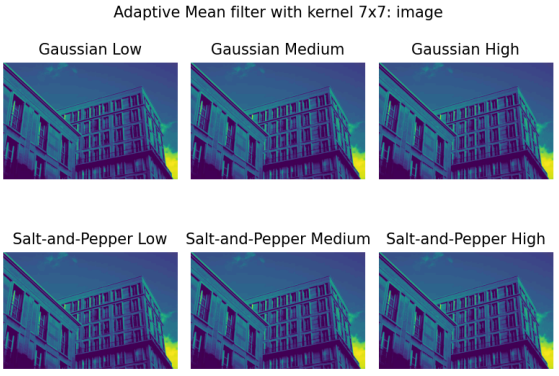
Adaptive mean filter:

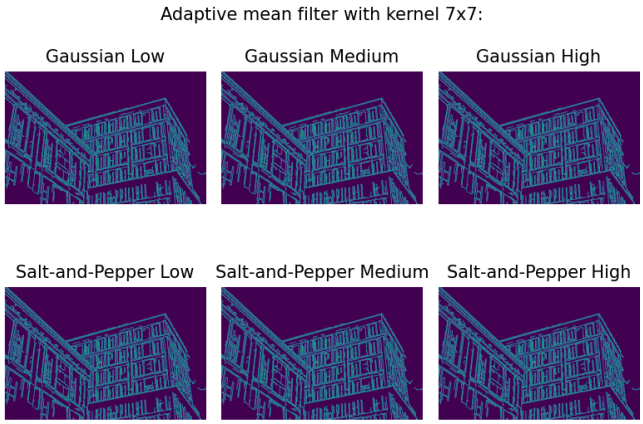






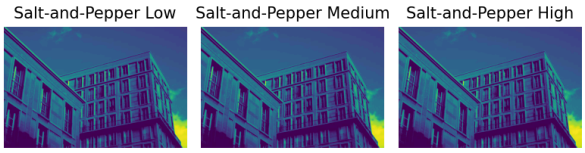






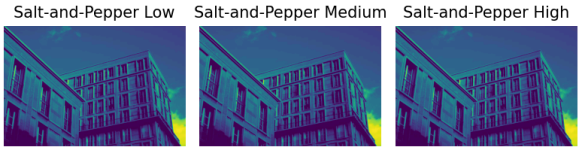
Adaptive median filter:

Adaptive Median filter with kernel 3x3: image



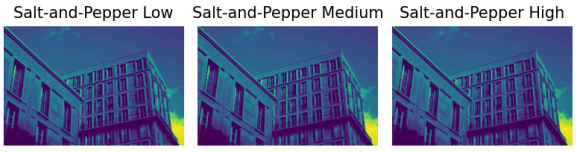
Adaptive median filter with kernel 3x3:

Adaptive Median filter with kernel 5x5: image



Adaptive median filter with kernel 5x5:

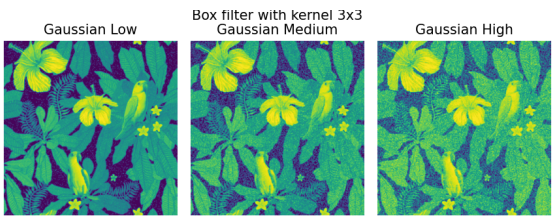
Adaptive Median filter with kernel 7x7: image

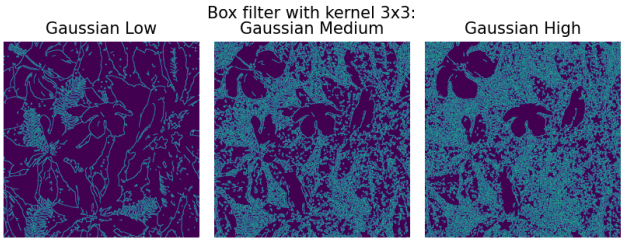


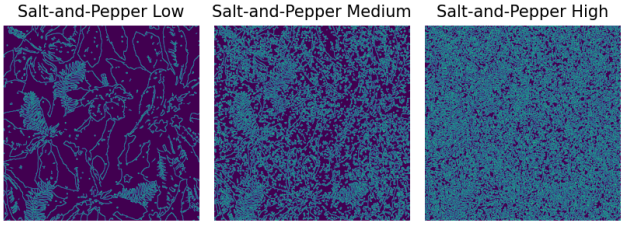
Adaptive median filter with kernel 7x7:

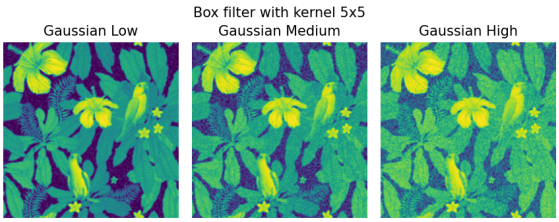
Image 3:

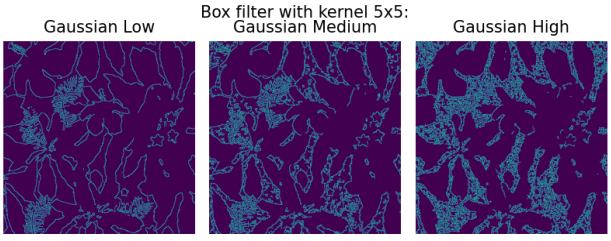
Box filter:

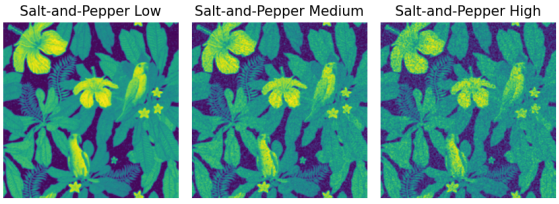


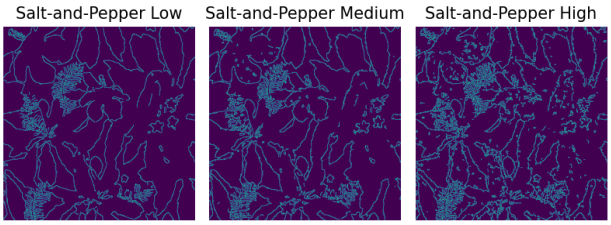


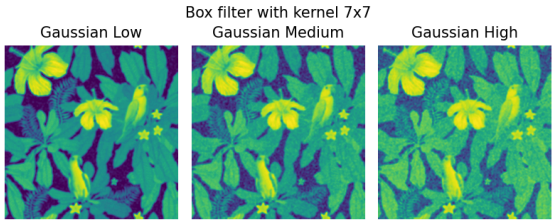


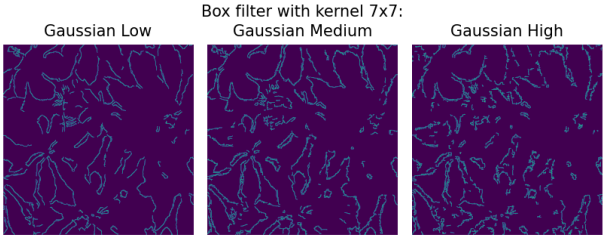


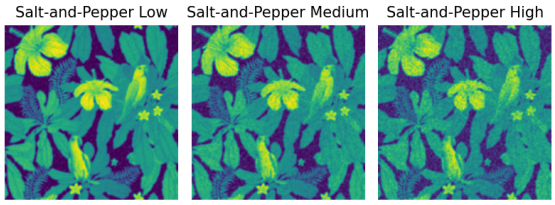


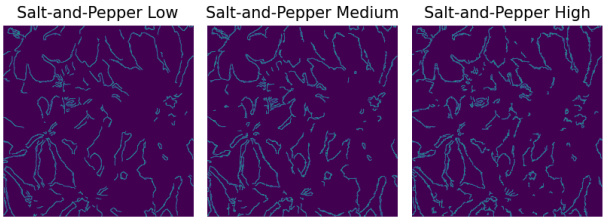




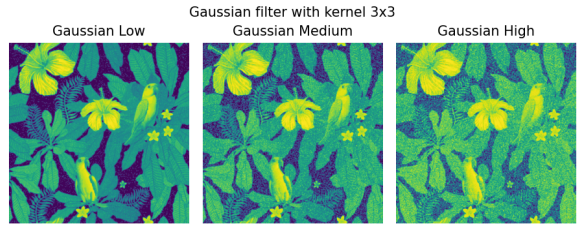


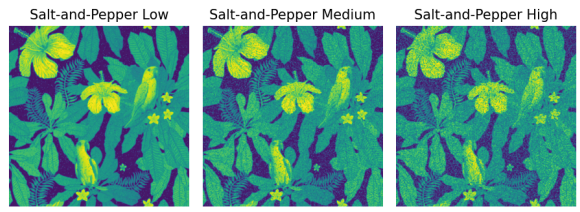


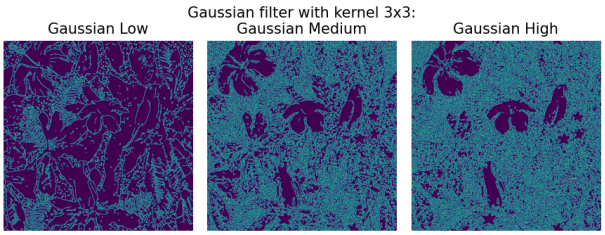


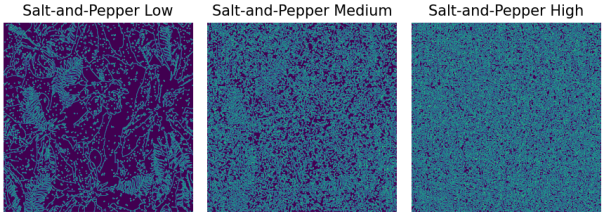


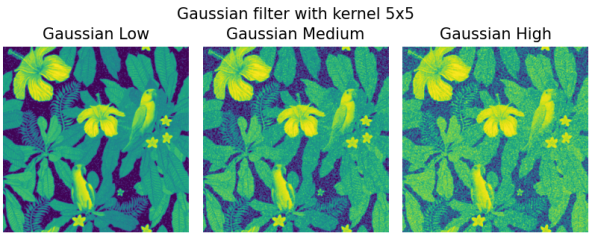
Gaussian filter:

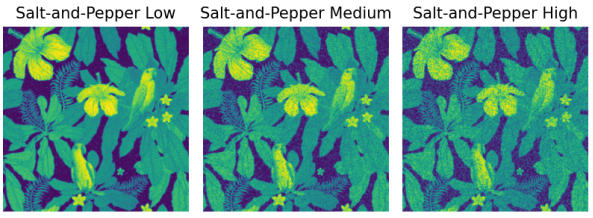


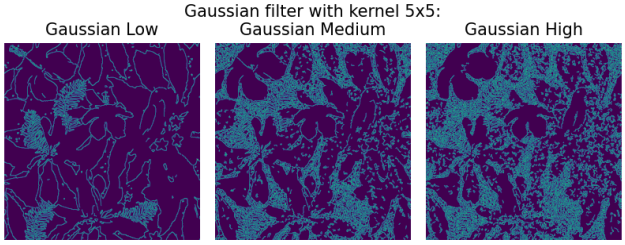


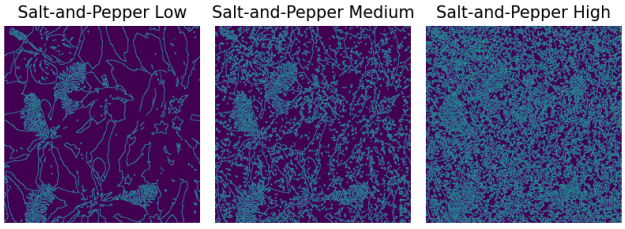


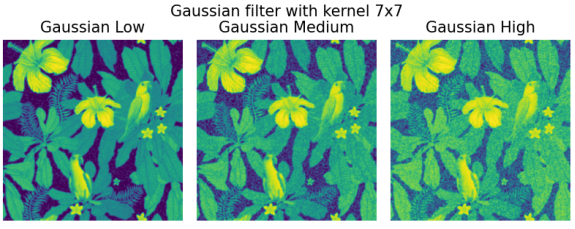


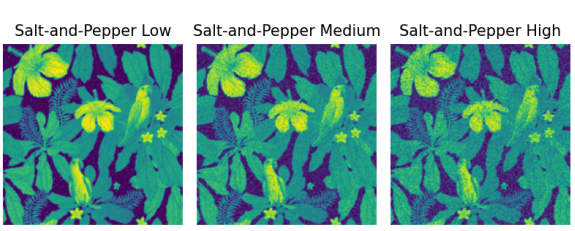


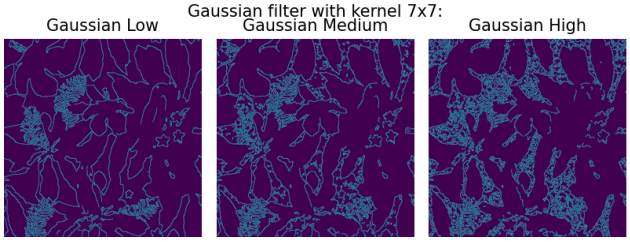


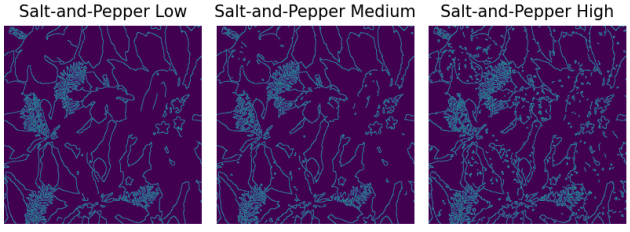




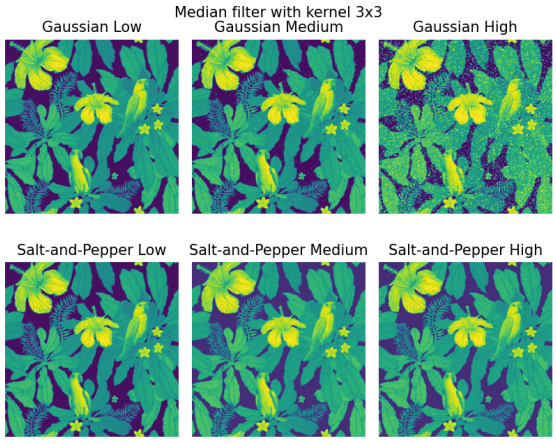


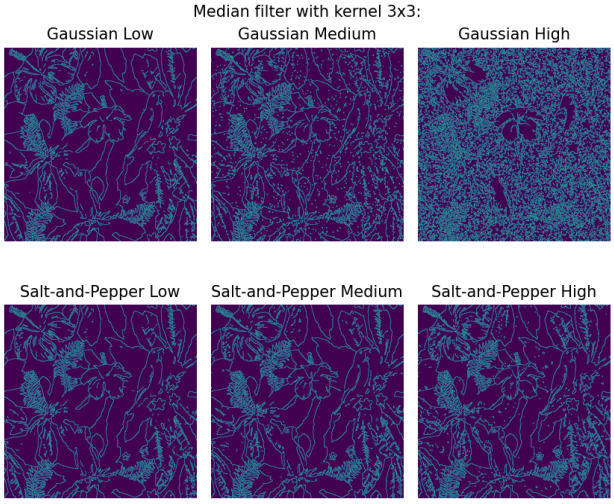


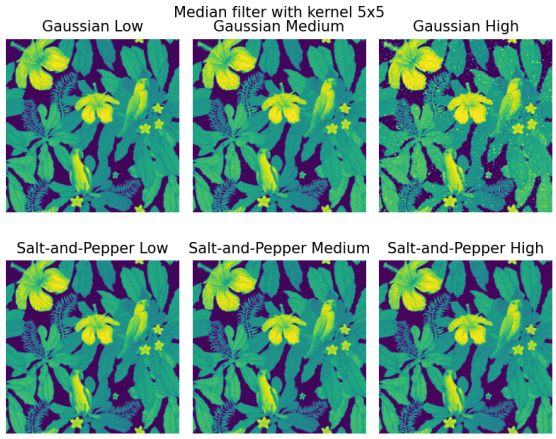


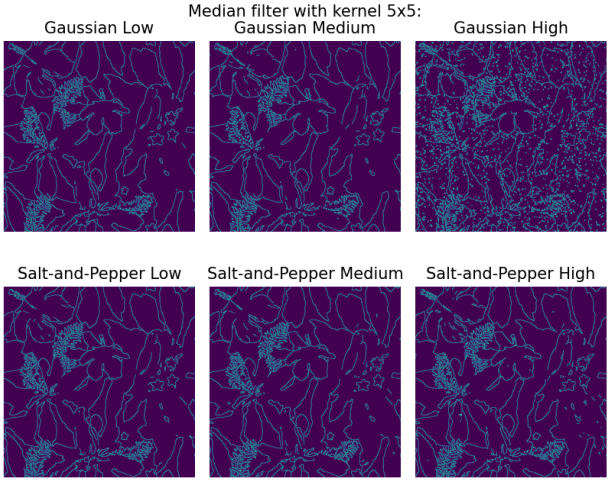


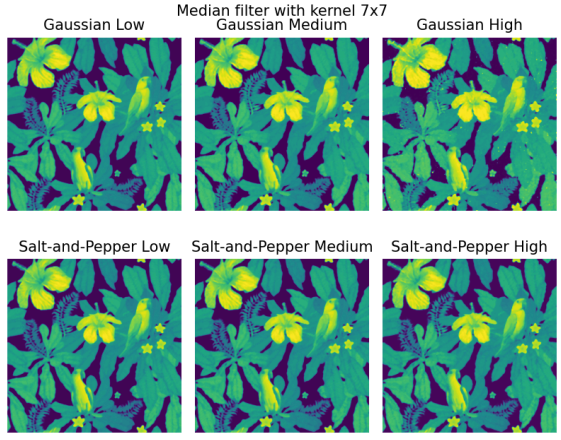
Median filter:

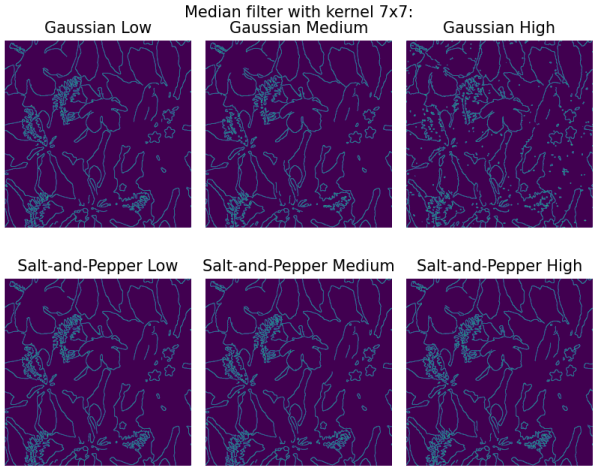




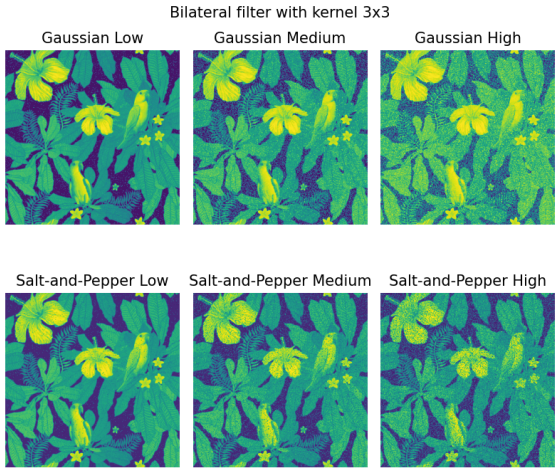


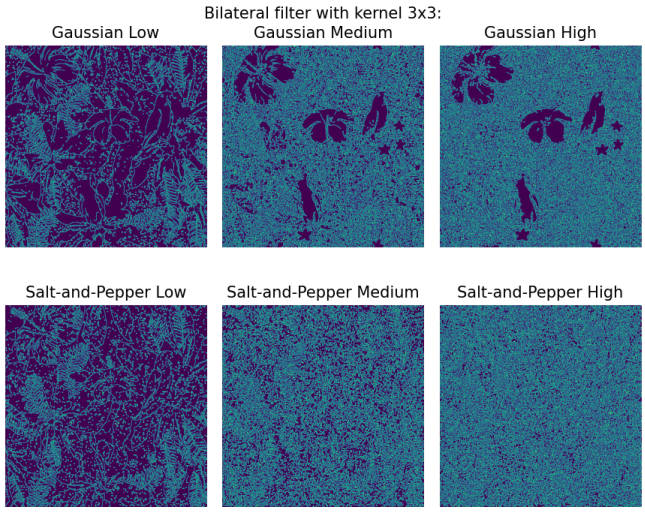


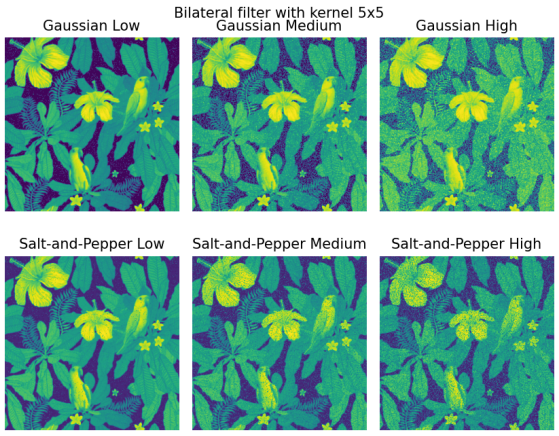


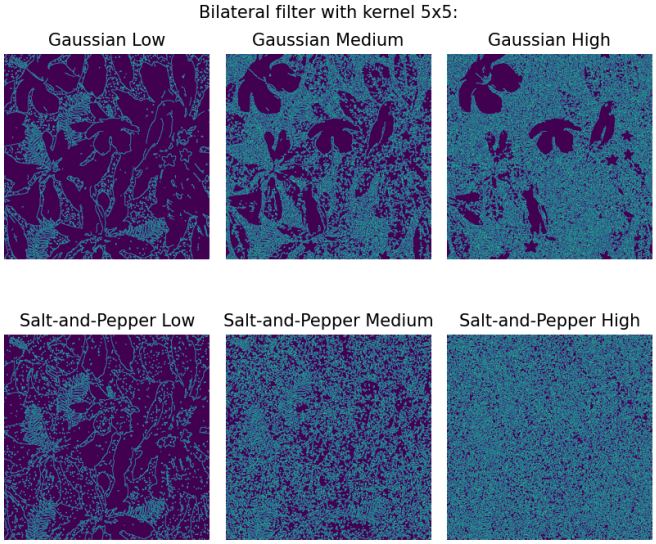


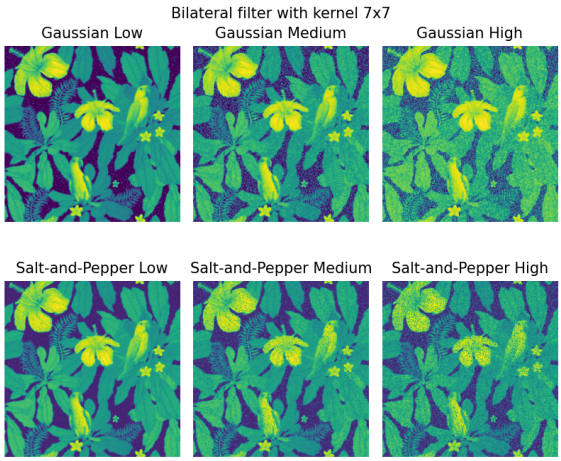
Bilateral filter:

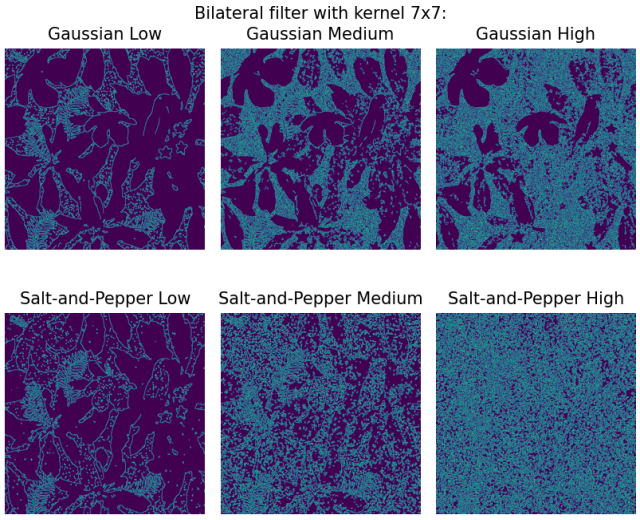




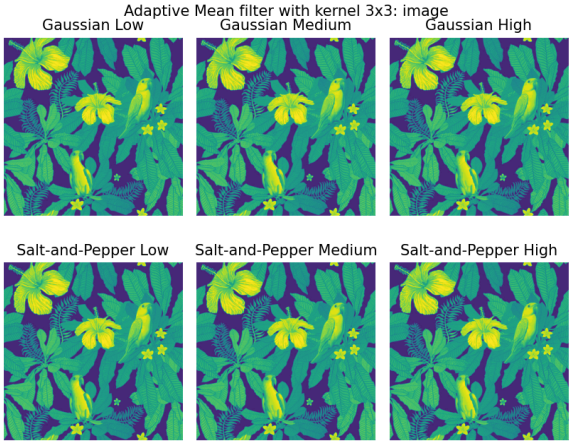


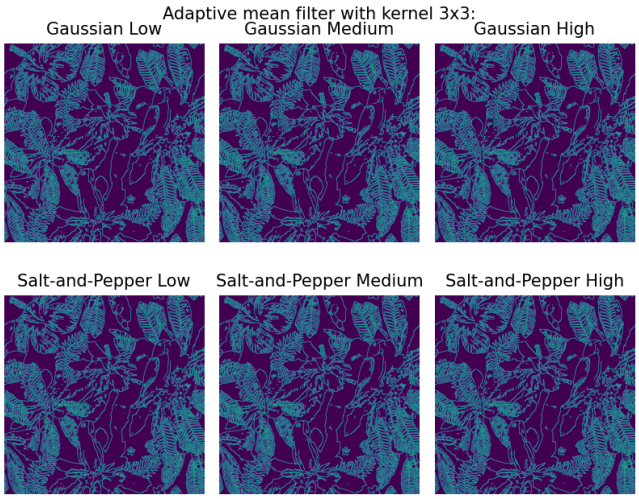


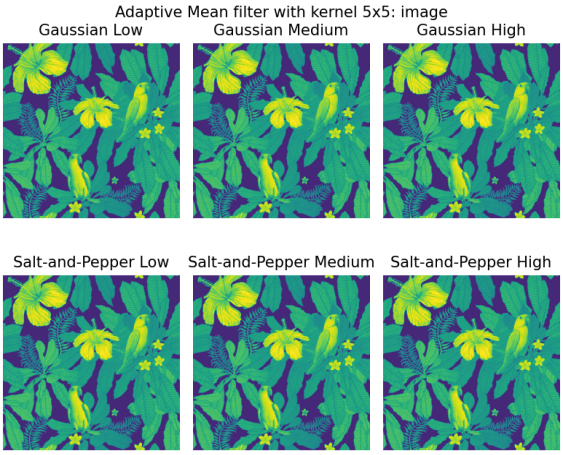


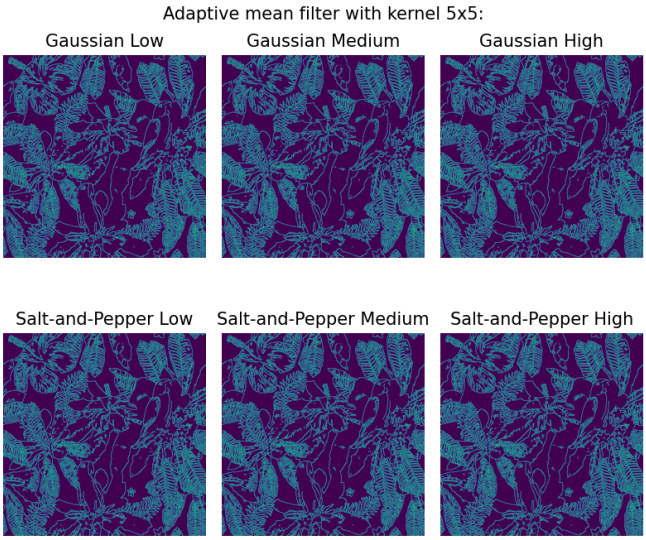


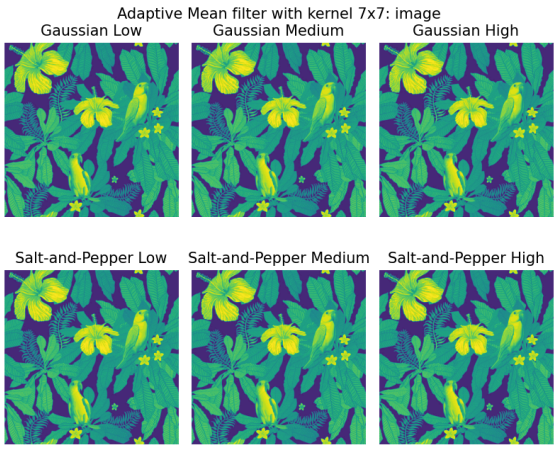
Adaptive mean filter:

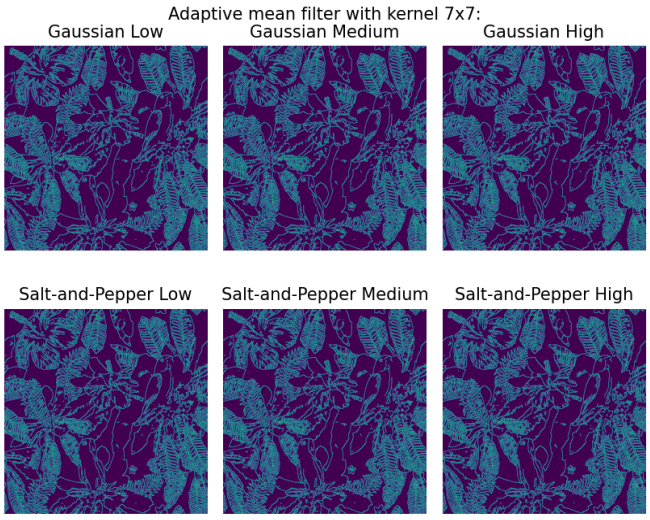




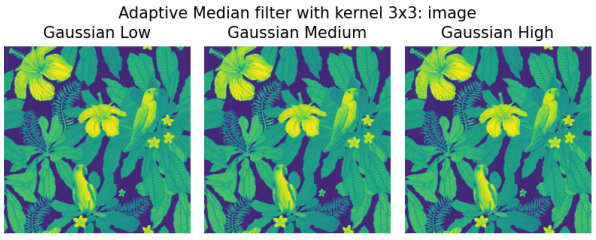


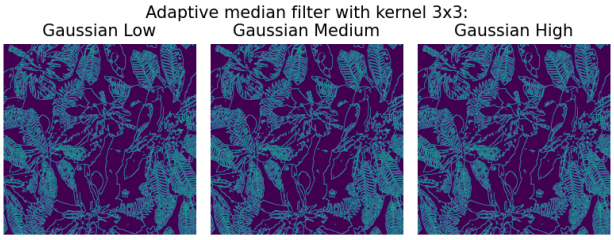


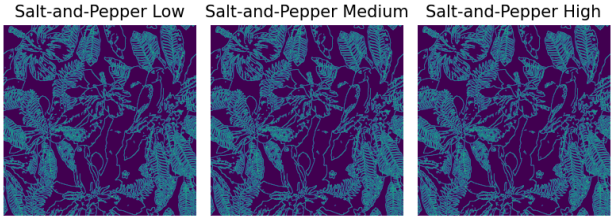


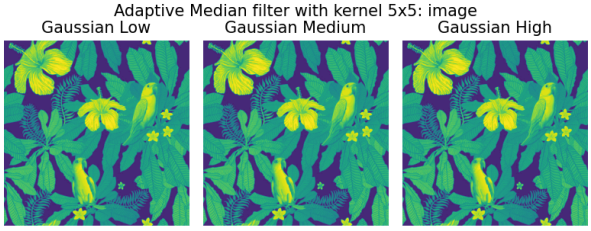


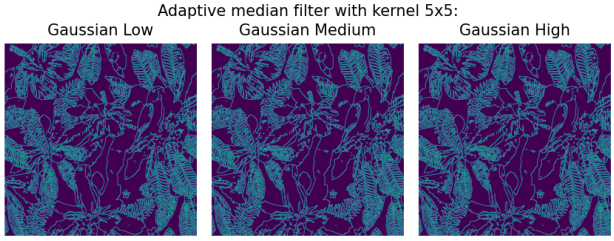
Adaptive median filter:

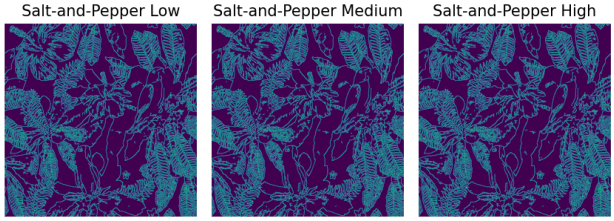


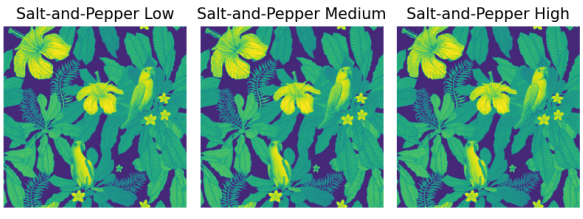


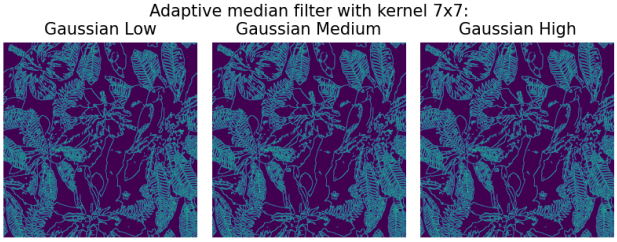












Conclusion

To sum up, the simple smoothing filters that were used in this assignment have shown their capability of fast computing, but with poor to medium noise reduction and edge preservation. However, the advanced filters, such as the adaptive mean and adaptive median, have shown their capability of performing very good noise reduction and still preserve edges as well, but at the cost of high computation requirements.