

CMP_SC 7650: Digital Image Processing

Homework 3: Noise Filtering

By: Mikey Joyce

Due: 10/3/2023

Abstract:

The problem presented is to utilize various filtering mechanisms through OpenCV or SciPy and to implement the adaptive median filtering mechanism efficiently from scratch. This implementation was able to successfully implement mean squared error (MSE) calculation, SciPy filtering functions, and a custom adaptive median filtering function.

Introduction:

This assignment will have two separate experiments. The first experiment will utilize SciPy functions to apply a Gaussian and Median filter on two test images that have heavy noise and then calculate the MSE with respect to the image without noise. The second experiment will apply the efficient adaptive median filter on two different test images to recover the original image. The processing time, MSE, and a plot of the reconstructed pixels at each filter size will also be reported alongside the new images

Experiments and Results:

Experiment #1:

	MSE				
	Noisy	Gaussian sigma=2	Gaussian sigma=7	Median (7x7)	Median (19x19)
Test1Noise1	93.77	31.12	52.20	35.61	44.95
Test1Noise2	90.39	109.20	105.02	42.54	41.64

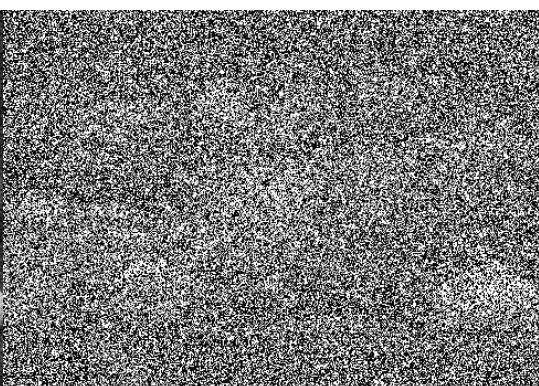
Original Image (Test1)



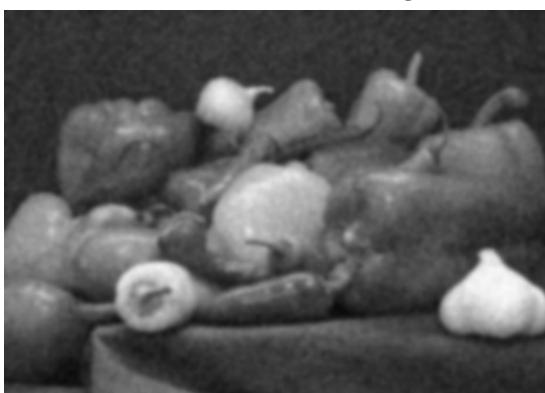
Test1Noise1



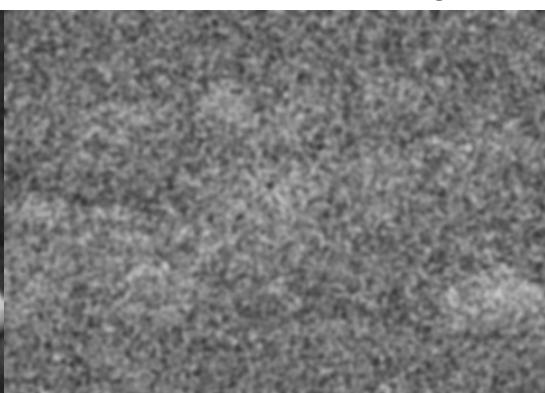
Test1Noise2



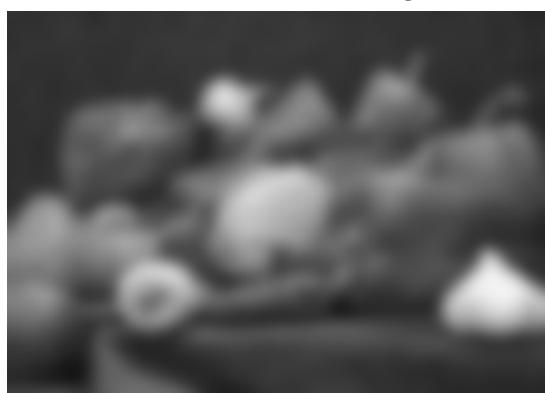
Test1Noise1→Gaussian: sigma=2



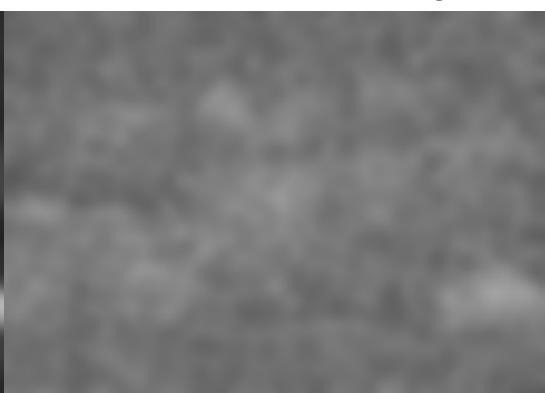
Test1Noise2→Gaussian: sigma=2



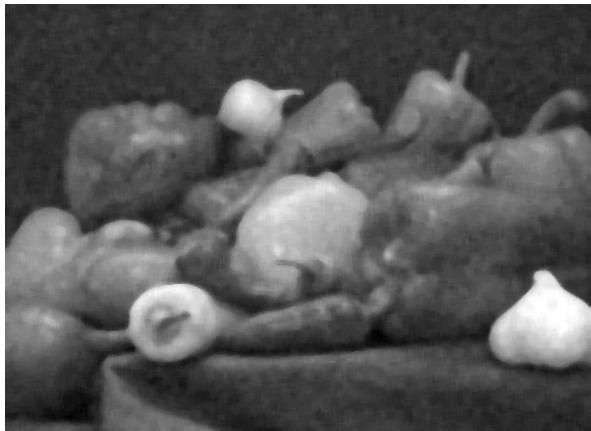
Test1Noise1→Gaussian: sigma=7



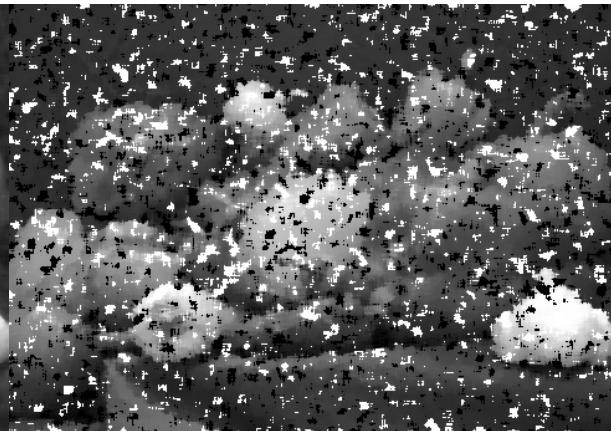
Test1Noise2→Gaussian: sigma=7



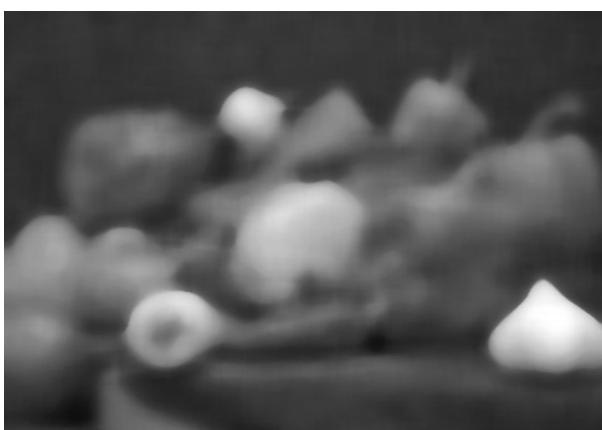
Test1Noise1→Median 7x7



Test1Noise2→Median 7x7



Test1Noise1→Median 19x19



Test1Noise2→Median 19x19

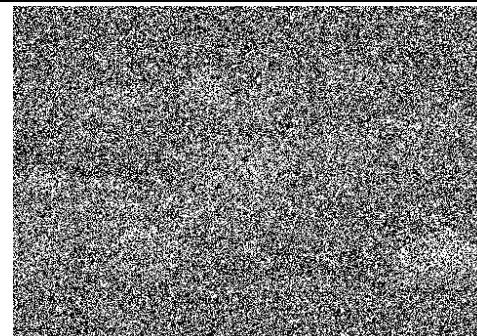


Experiment #2:

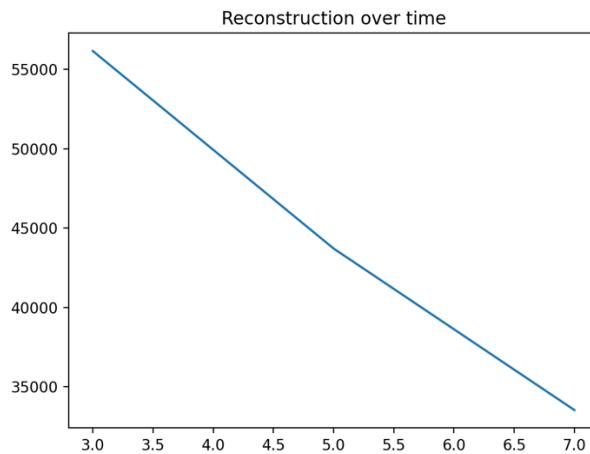
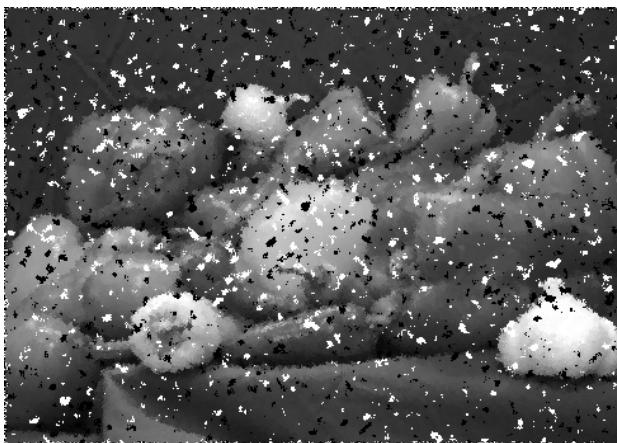
	MSE			Processing Time	
	Noisy	Adaptive Median Smax= 7x7	Adaptive Median Smax = 19x19	Adaptive Median Smax= 7x7	Adaptive Median Smax = 19x19
Test1Noise2	90.39	28.63	23.58	1.03 seconds	1.35 seconds
Test2Noise2	76.84	53.57	53.09	4.24 seconds	3.78 seconds



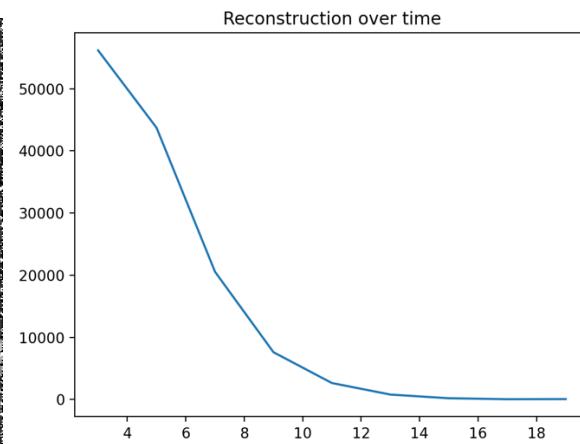
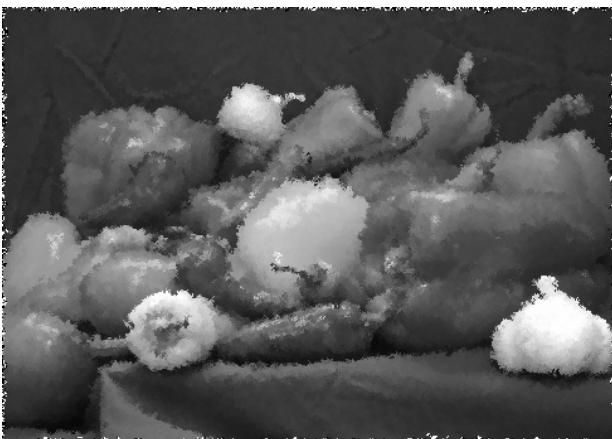
← Test1
Test1Noise2 →



Test1Noise2 Adaptive Median Smax=7x7



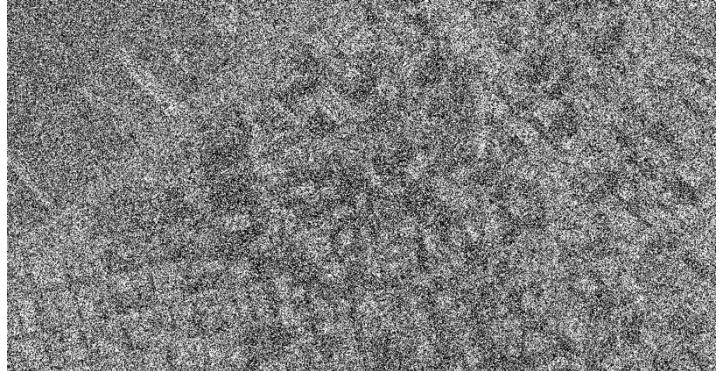
Test1Noise2 Adaptive Median Smax=19x19



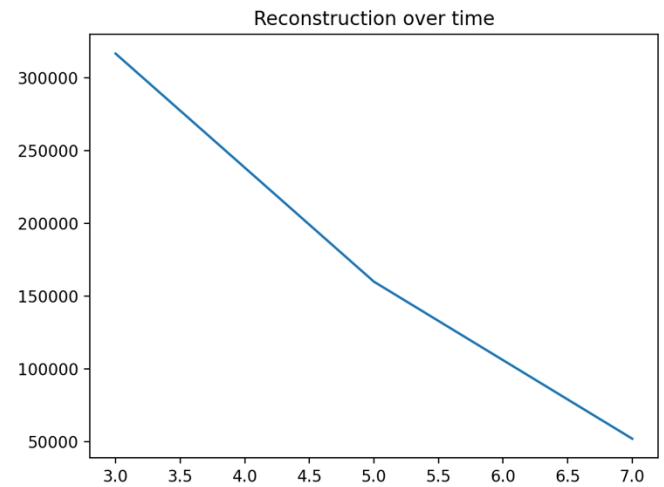
Test2



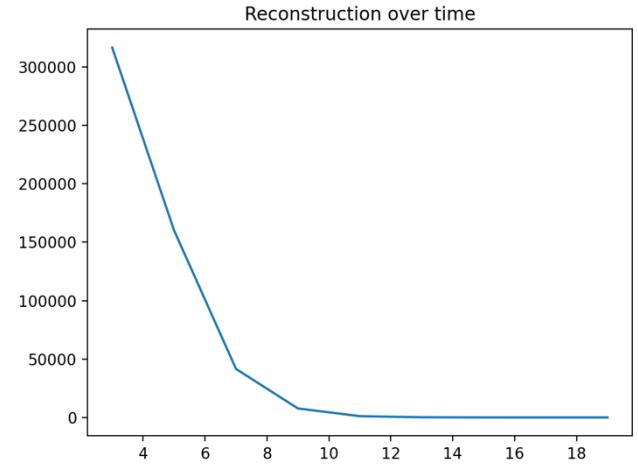
Test2Noise2



Test2Noise2 Adaptive Median Smax=7x7



Test2Noise2 Adaptive Median Smax=19x19



Conclusion:

In this assignment various filters such as the Gaussian filter, the median filter, and the adaptive median were explored. Through experimentation we were able to see how they affect the given image and what situations would be good for each filter. We also implemented a MSE function to calculate the MSE between two images, which was crucial for the reporting of metrics in each experiment.

Experiment #1:

In this experiment 8 images were generated. There were two test images (Test1Noise1 and Test1Noise2) that were to be manipulated utilizing the Gaussian filter with sigma=2, a Gaussian filter with sigma=7, a 7x7 median filter, and lastly a 19x19 median filter. Noise1 is much less noise when compared to Noise2, because of this it was surprising to me that the MSE of the Noise2 image initially was lower than that of Noise1. Now we will discuss the results for each filter. The first filter was a Gaussian filter with sigma=2. For the Test1Noise1 image, it was able to clear up some of the noise without removing too many of the details, this is a satisfactory result despite there still being a bit of noise present. However, for the Test1Noise2 image, it just blurred the noise, making the resulting image worse than the original image when you factor in the MSE values calculated. The next filter was a Gaussian filter with sigma=7. For the Test1Noise1 image, it appears that it blurred the image too much and lost some of the details. We can confirm this result is worse than the image generated with the Gaussian sigma=2 due to the higher MSE. For the Test1Noise2 image, it is just more of the same blurring the noise rather than getting rid of it. It can be concluded that the Gaussian filter is not applicable to images with heavy salt and pepper noise. The next filter in question is the 7x7 median filter. For the Test1Noise1 it appears to be a very similar result to the Gaussian sigma=2 image, which is good as it diminished some of the noise. As for the Test1Noise2 image, some more of the structure seems to have been recovered, but there is still a large amount of noise in the photo, except this time the noise is large white and black blobs. The last filter for this experiment is the 19x19 median filter. For the Test1Noise1 image, the median 19x19 filter seems to do something similar to the Gaussian sigma=7, where it removes the noise, but blurs the image in a way that removes the details. The Test1Noise2 image however is impressively cleared up compared to the original noisy image. While the resulting image is not perfect, a lot of the structure was recovered where the original image you couldn't tell what the image was.

Experiment #2:

In this experiment 4 images were generated utilizing the adaptive median filter that I implemented. The goal of this was to implement the adaptive median filter in an efficient manner. There were two given test images that were to be utilized. The first test image within this experiment was Test1Noise2. Which was the image Test1 except

with a large amount of noise present. Utilizing the adaptive median filter with a maximum sliding window size of 7x7 a decent amount of the pixels were able to be recovered. It gives a similar result of the median 7x7 filter where there is still some white and black blobs that are blocking the image. When the maximum sliding window size is increased to 19x19, most of the image is recovered. You can see lots of the structure including some of the details such as the stems, which gave a better result than the median 19x19 filter. The next test image is Test2Noise2 which is the image Test2 except with lots of noise. Utilizing the adaptive median smax=7x7 filter, most of the image was able to be recovered, there is still some salt and pepper noise within the image, but it is mostly recovered. When smax is increased to 19x19, the image almost looks identical to the original image, the only difference is it is not a crisp and it is rough around the edges. The adaptive median filter was also implemented in an efficient manner as it always was around 4 seconds of computation or less. One note is that in Test2Noise2 the 19x19 filter performed faster than the 7x7 filter which was a bit unexpected. This is due to the sorting algorithm being $O(n \log n)$ on average, sometimes it is a bit faster and other times it is a bit slower.

All in all, everything that was asked for this assignment was accomplished. The MSE calculation and adaptive median filters were successfully implemented in an efficient manner. And the experimentation utilizing SciPy's Gaussian and median filters was also successful.

References:

- Libraries and tools: PyCharm, OpenCV, NumPy, Matplotlib, Preview
- Lec06_Filtering_FB_v2.pdf
- Lec07_Filtering_Adaptive_Integral_Histogram_v2.pdf