

ISE-309
HW-2
Selin Dinç
150150229

In the first part of the code, I read all the images from sample_images1 and padded the image boundaries by zero. I calculated the pad width by using $\text{pad_width} = (\text{kernel}-1)/2$. After that, I put all of the padded original images in separate lists.

Then, I implemented the 1-dimensional Gauss function and created a plot (figure1) from the results.

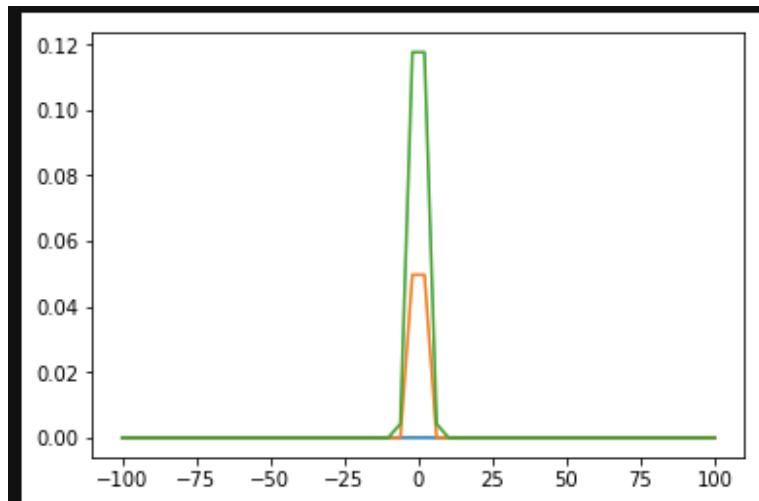
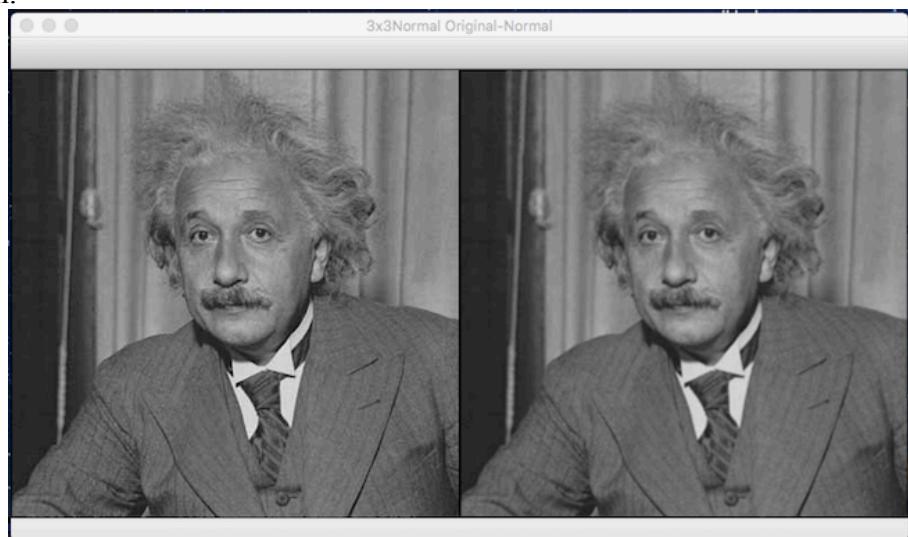


figure1

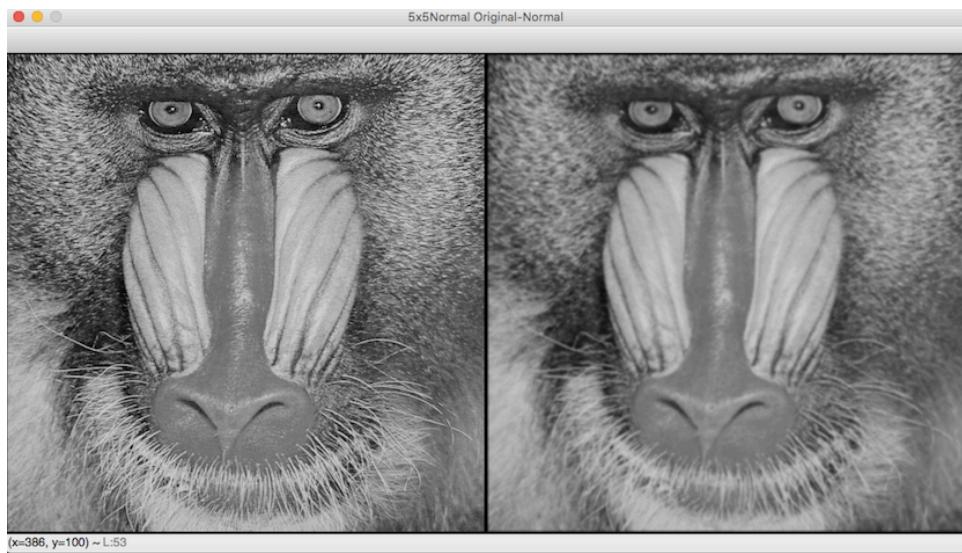
After that, I created the kernels which have been given in the question 2. Then, I did the normalization on those filters by dividing the sum of coefficient. I implemented those filters to the images.

I created new images which are going to show the filtered images. Then, padded them with zero. I created separate lists for each pad size.

In my code, normalization process is made by function called OperationFunc. In this function, I'm applying each filter to each image by iterating through the given images to normalize them.



(3x3 Normalized)

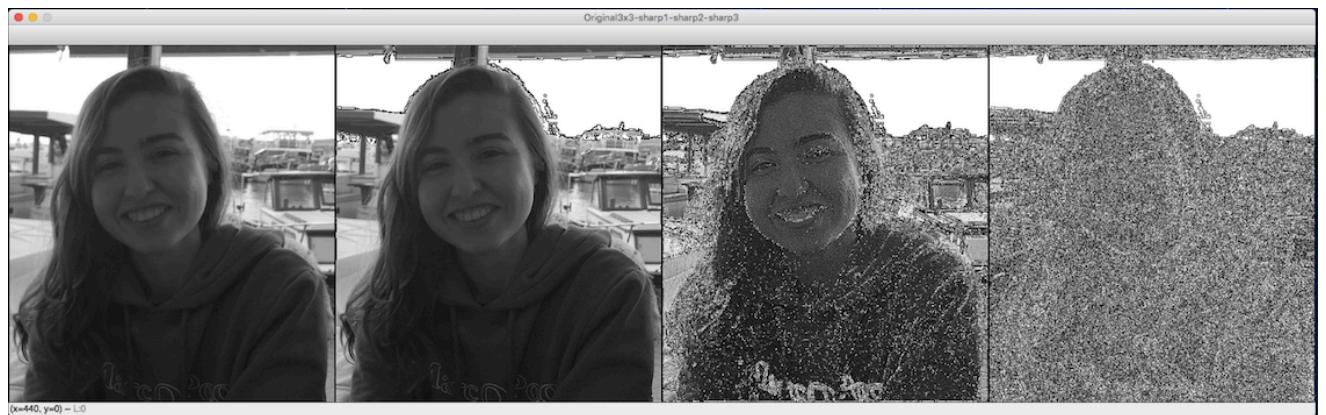


(5x5 Normalized)

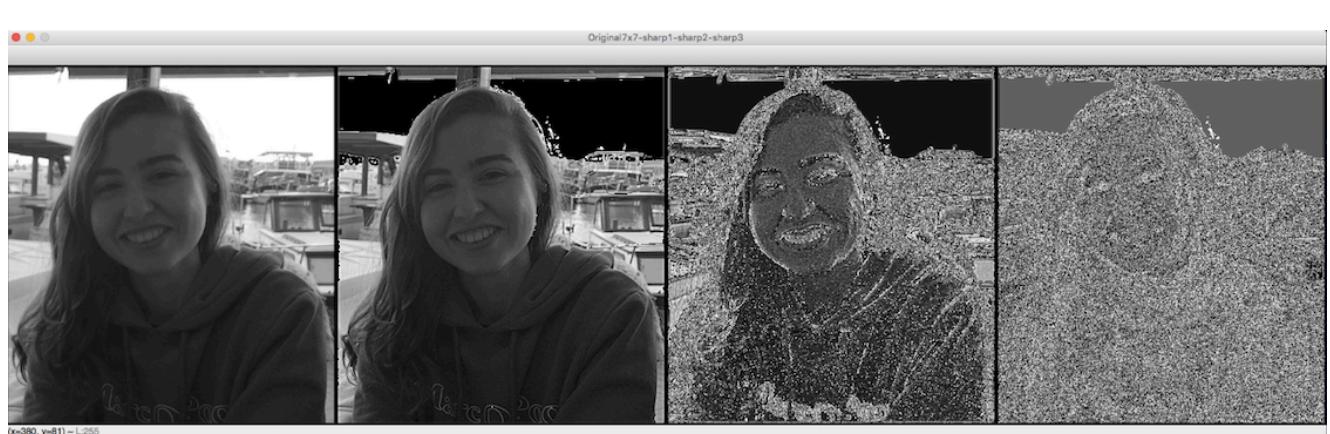
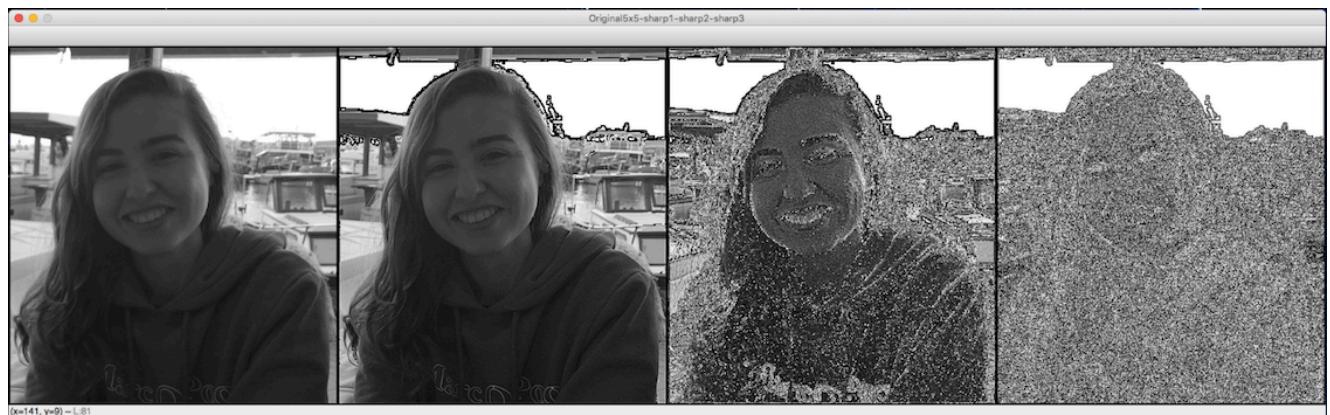


(7x7 Normalized)

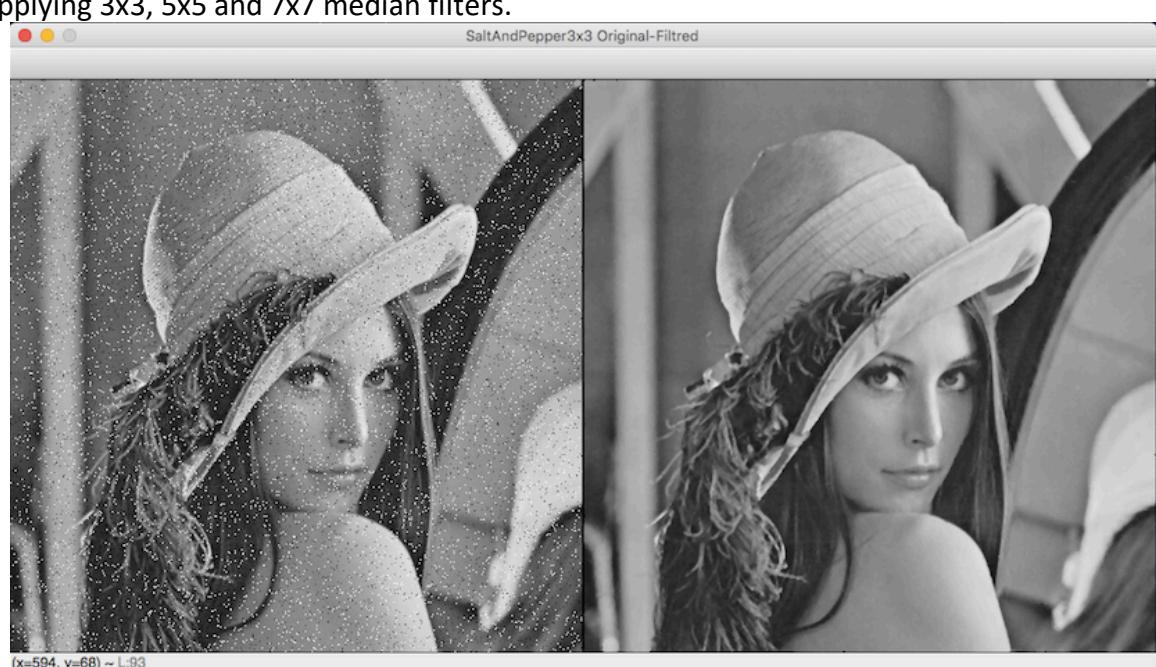
After that, I used the Gaussian filters from the 2nd question as sharpening filters. I implemented the each filter to my own face image by using different alfa values.

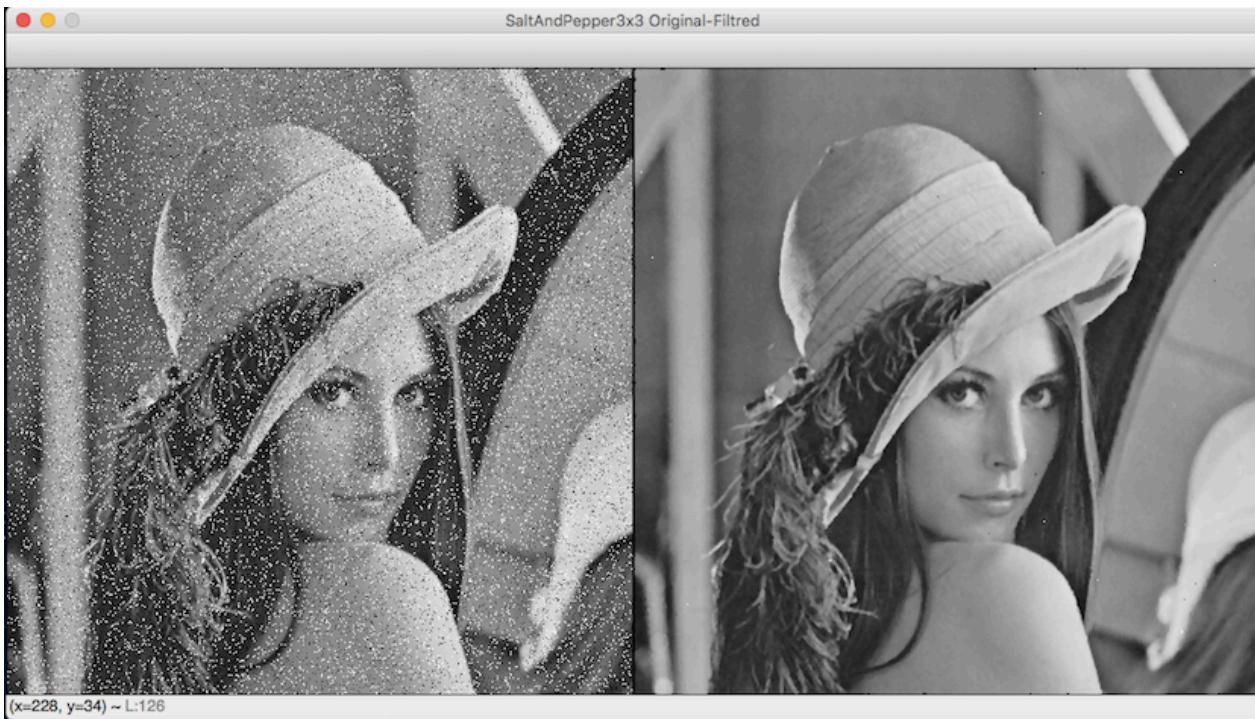


(original,3x3 with alfa values 1,20,100)

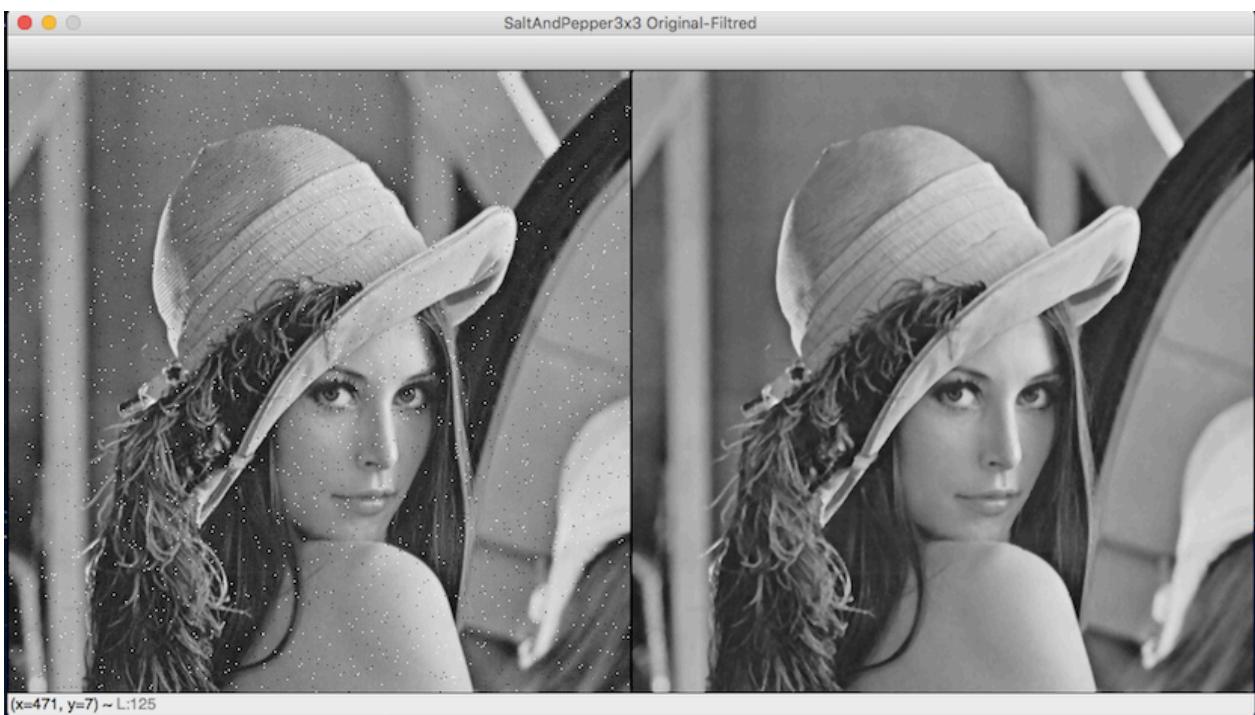


In the second part of my code, I read all the images from sample_file2 and I did the same process; padding images, creating new padded images. Then, I implemented a function called SaltPepper to remove the salt and pepper noise of the images from sample_file2 by applying 3x3, 5x5 and 7x7 median filters.





(3x3 median)



(3x3 median)



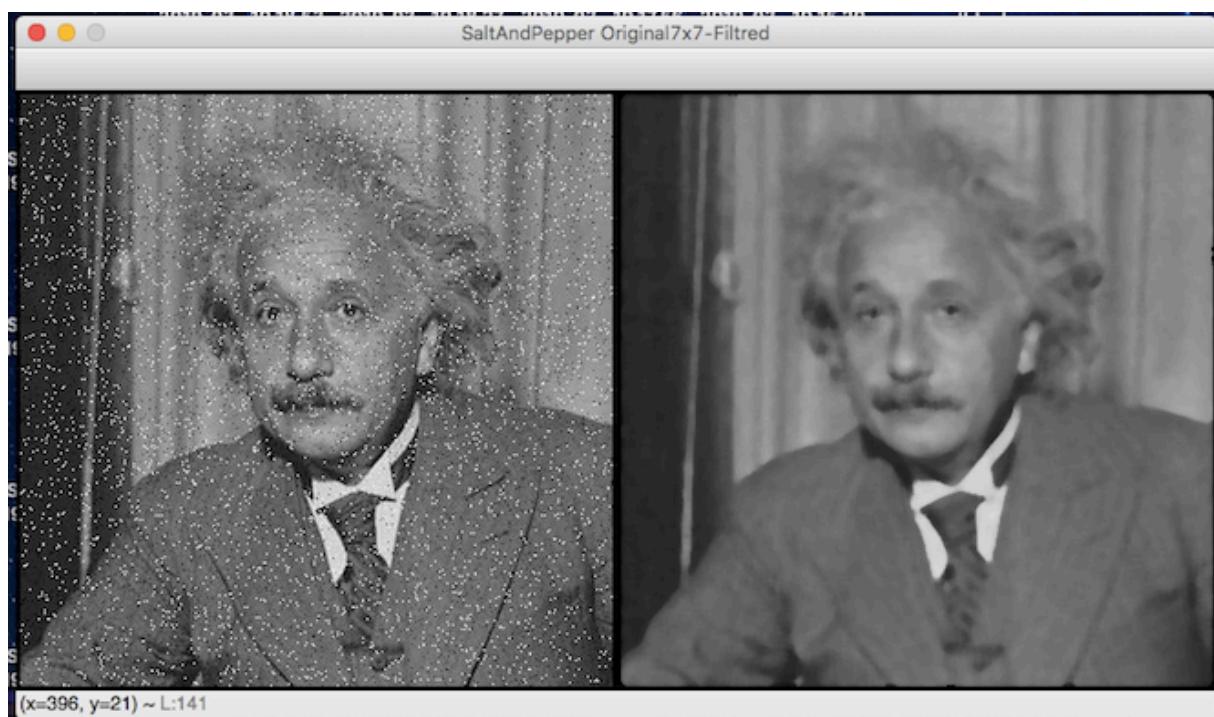
(5x5 median filtered)



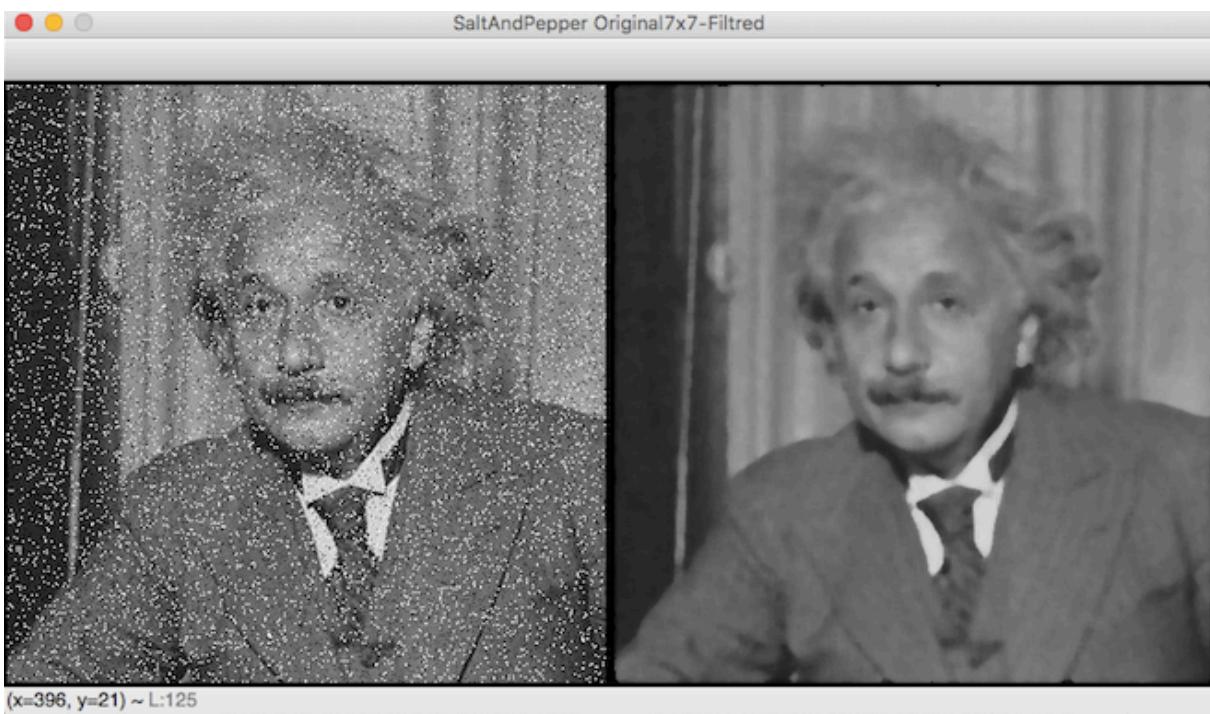
(5x5 median)



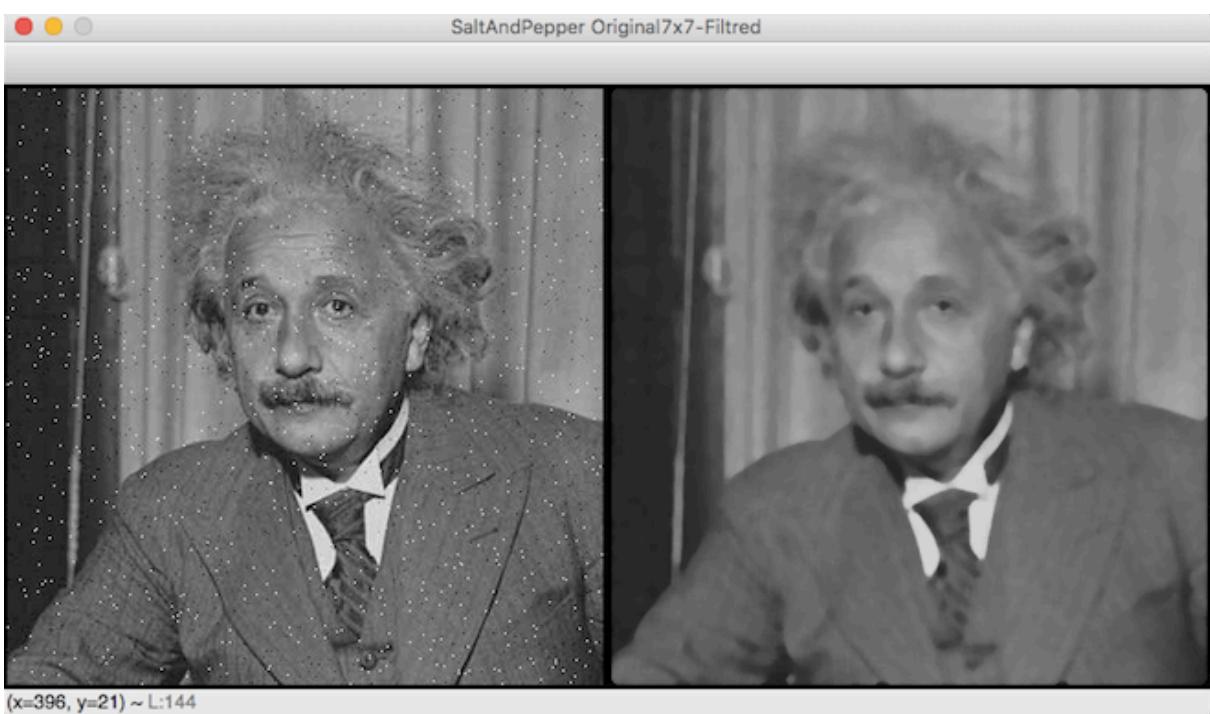
(5x5 median)



(7x7 median filtered)



(7x7 median)



(7x7 median)