Computer Vision Assignment 4

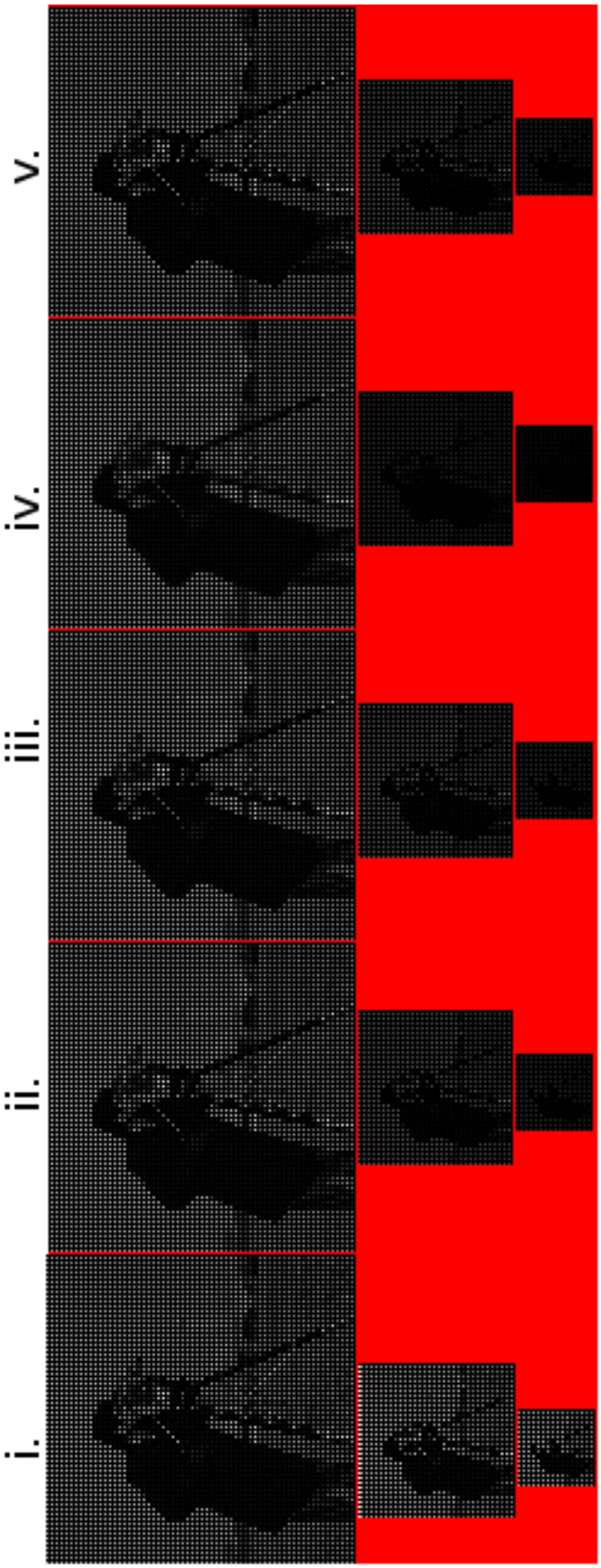
Ryan Cooper

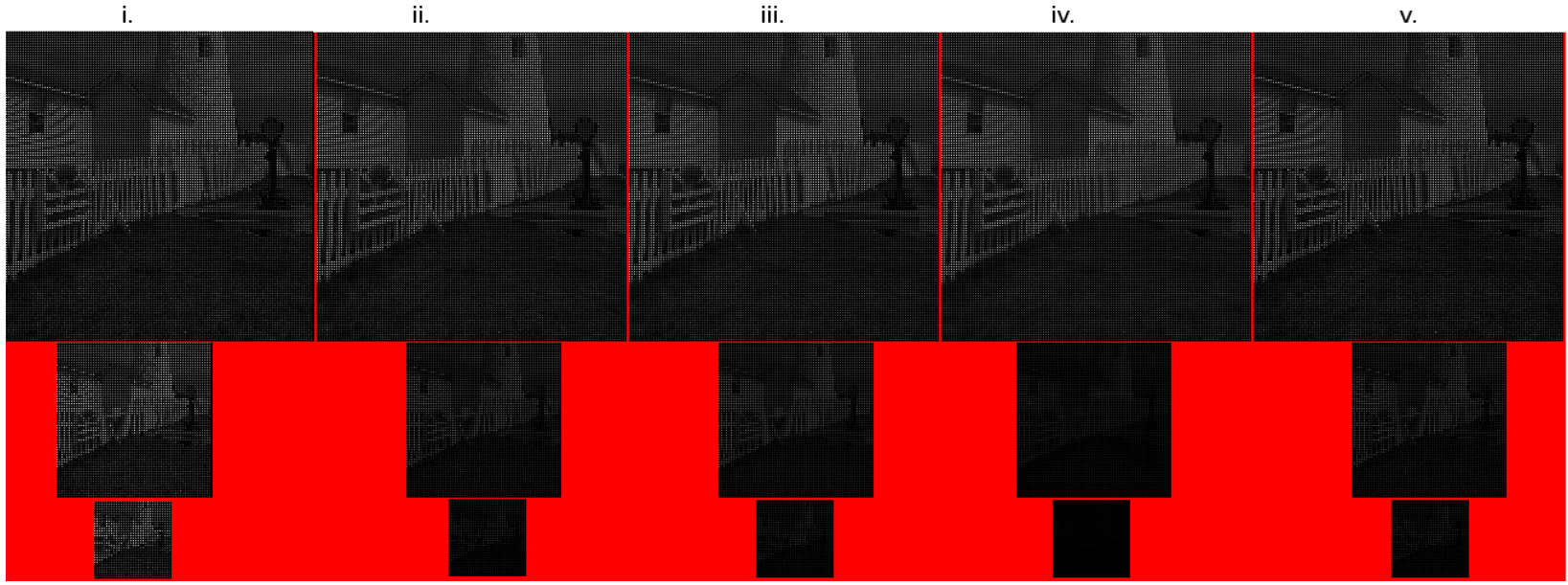
February 7, 2017

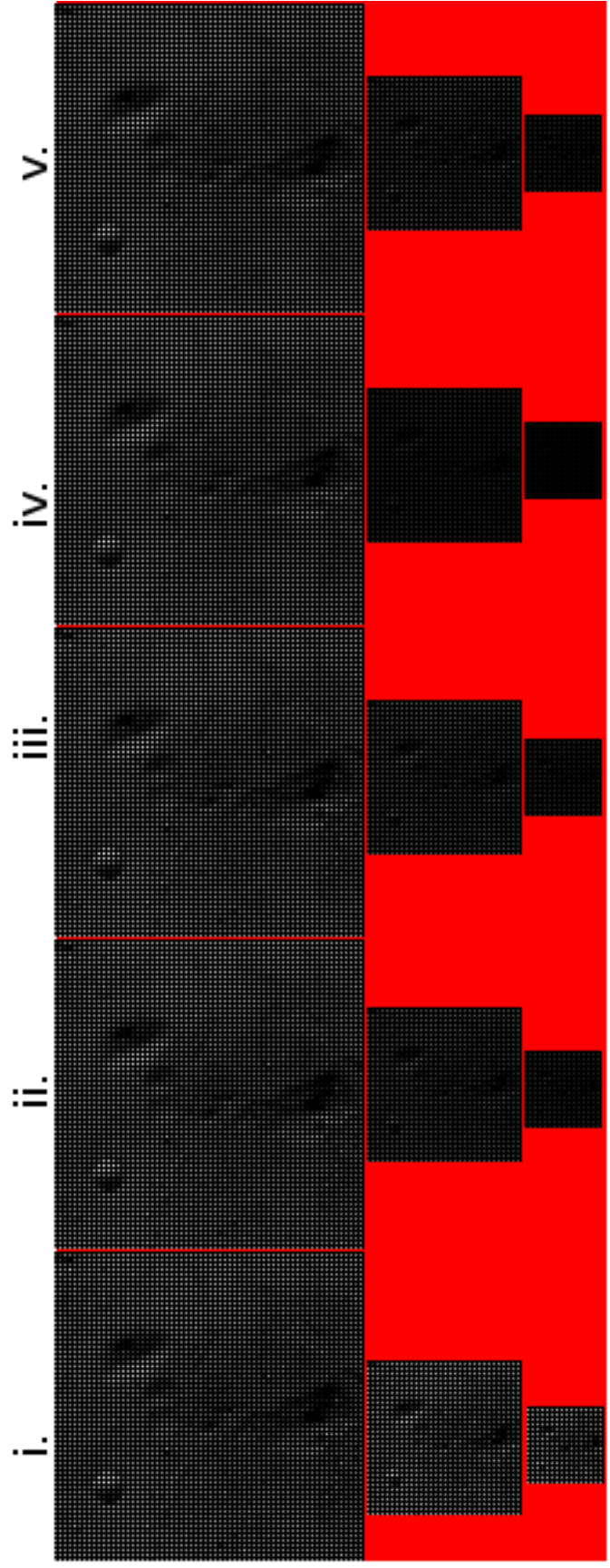
1. a. I am unsure if you wanted the code so here is a link to the a github page that has all of the code and images from this homework.

https://github.com/Pillager225/ComputerVisionAssignments/tree/master/Ass4

b. Images at each stage of the pyramid greater than 0 for each filter are shown on the next pages. The images were filtered with a 101 mirror padding.



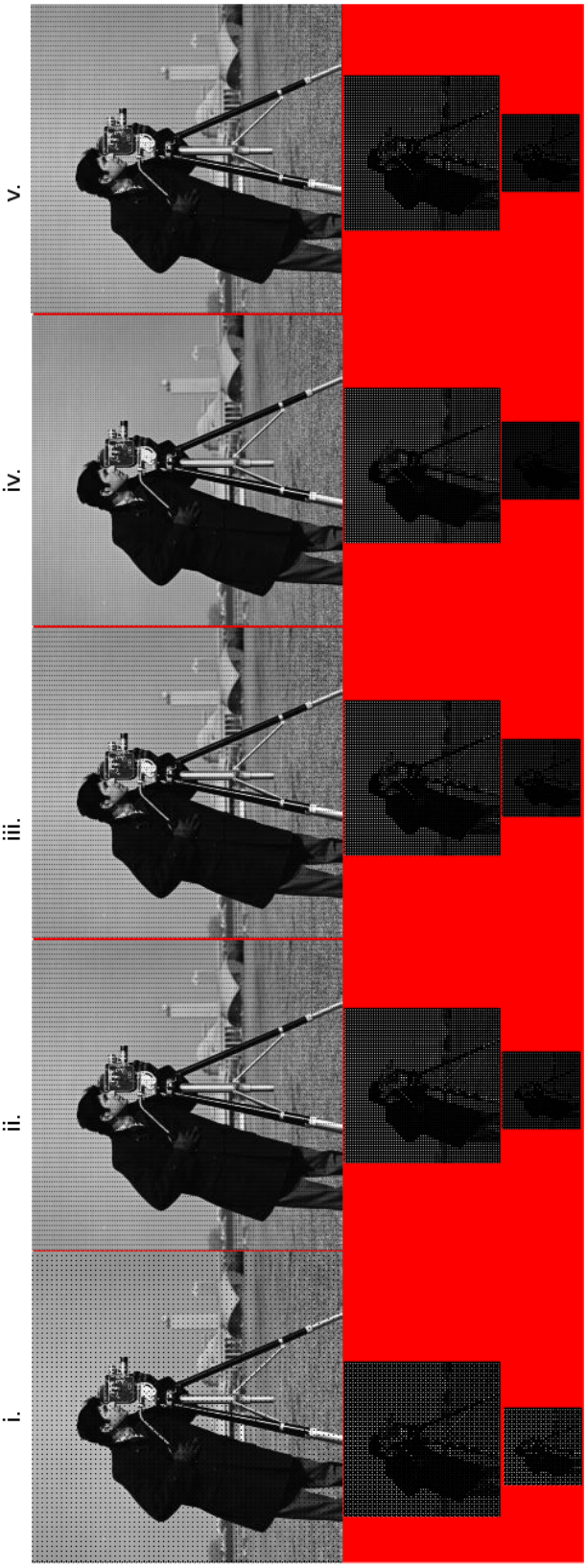

i.
ii.
iii.
iv.
v.



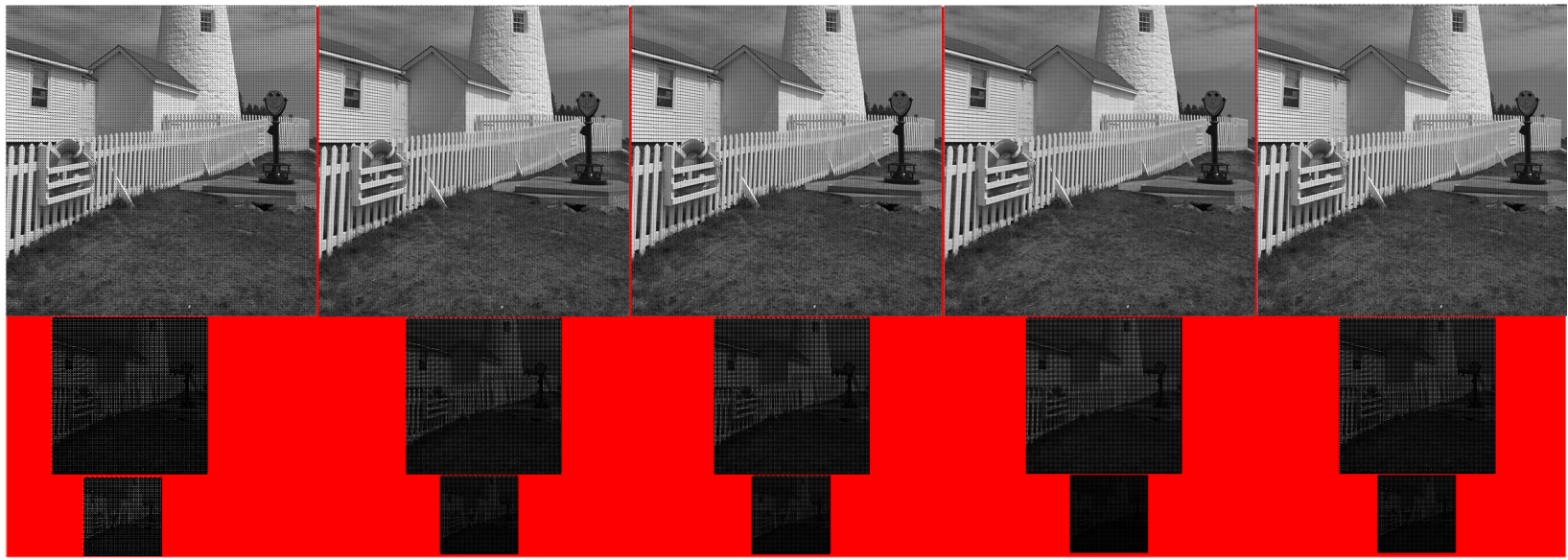
I did all of this work using opencv so I had to download the Matlab images from the Internet. The best version of the lighthouse was not the right size, but was instead 512x512.

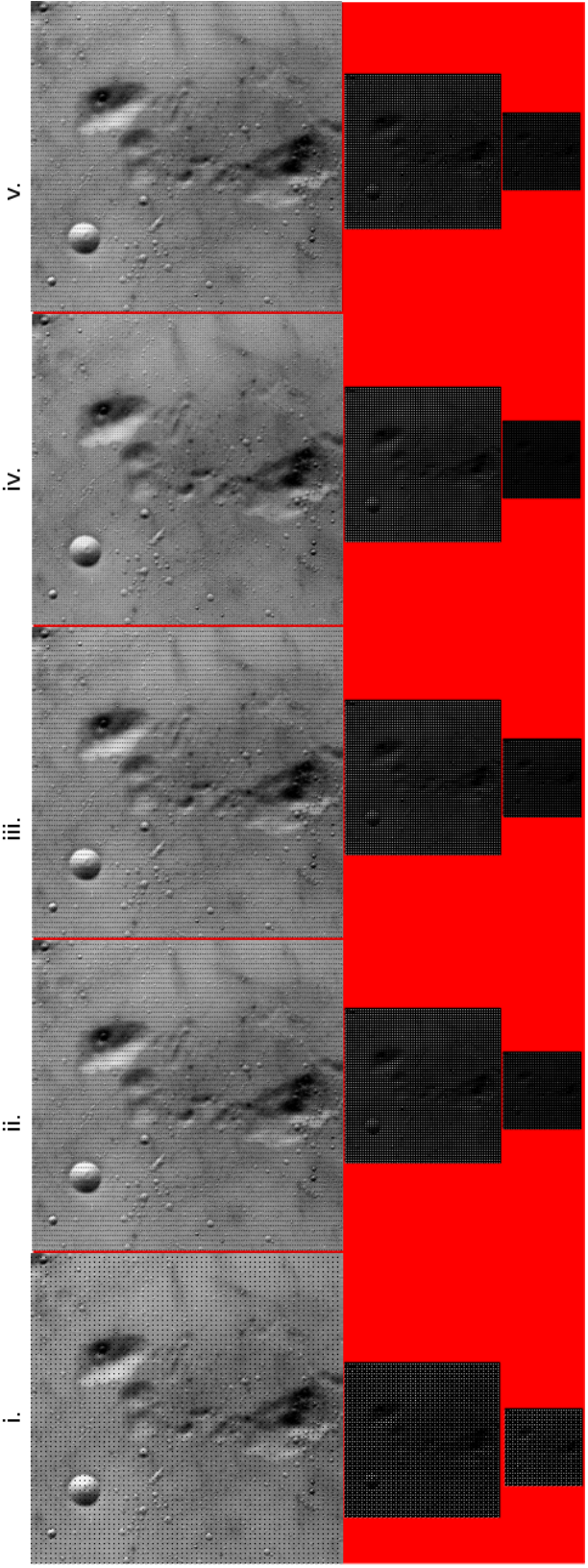
The last three filters all seem very similar until level 2. Then the Gaussian fitler is much darker. All of the 3rd levels except the one without a filter are dark from filtering so much.

2.a. Images at each stage of the pyramid less than 3 for each filter are shown on the next pages. The images were filtered with a 101 mirror padding.

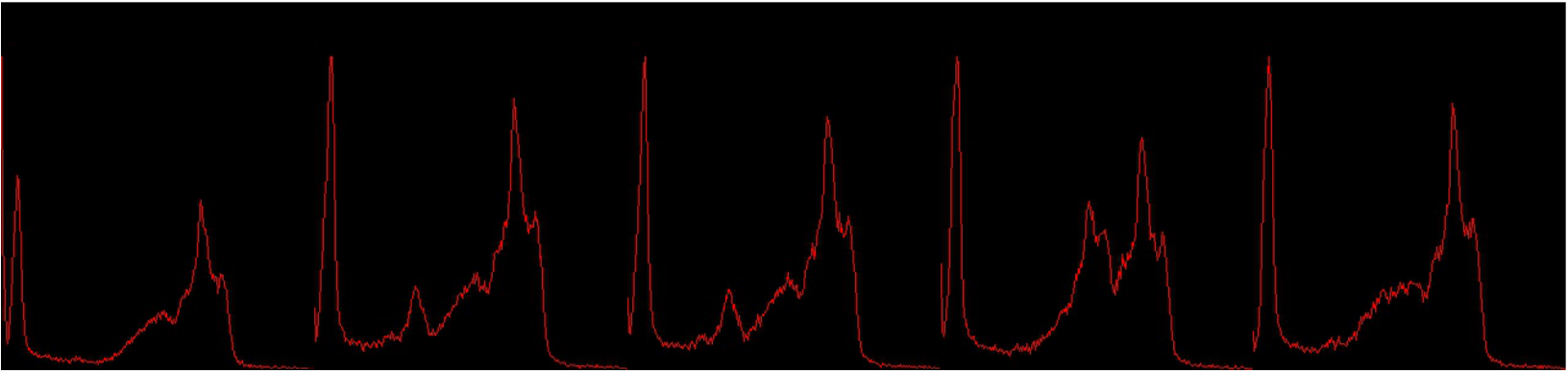


i. ii. iii. iv. v.

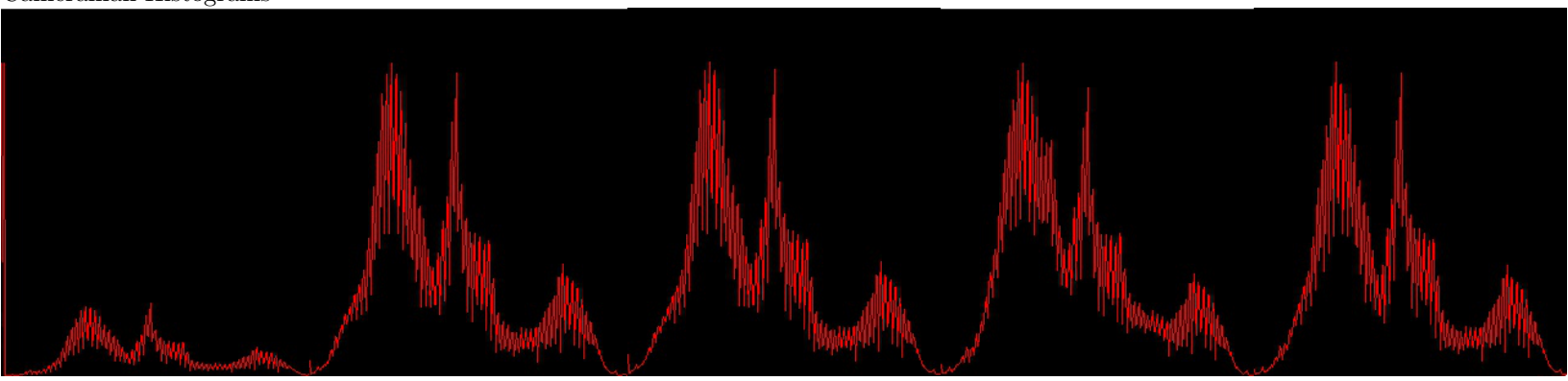




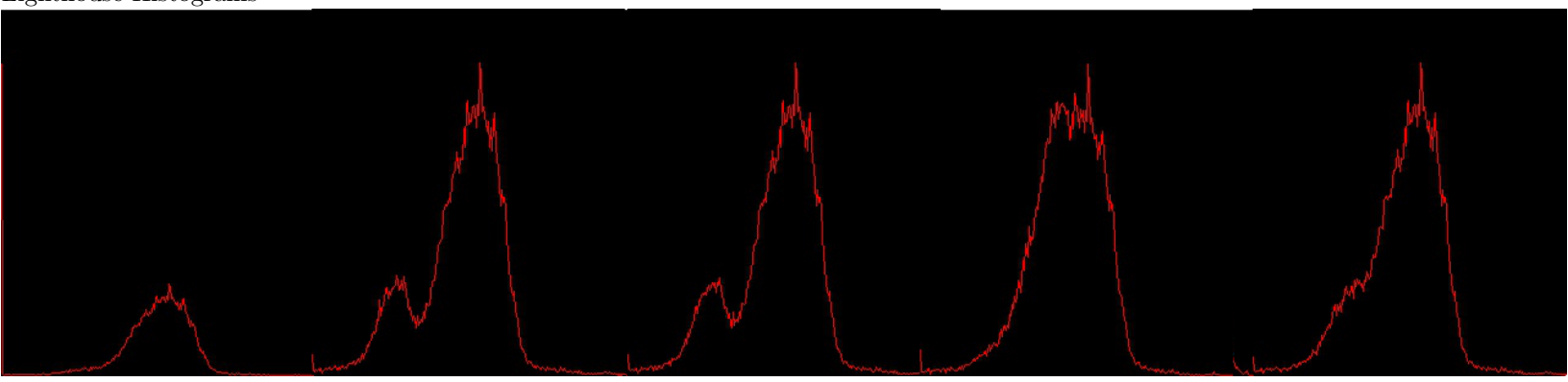
The Displayed images are pretty big, and had to be downsampled to be here, but the full resolution images can be found on the github page from earlier. b. The histograms are for the first layer of the Laplacian pyramid is shown. The other layer's histograms had barely anything on them, so I did not include them. They are on the github though.



Cameraman Histograms



Lighthouse Histograms



Moon Histograms

The mean squared values of the images, are show below.

cameraman	1	2	3
v.	16545.1276245	2867.67388916	726.753173828
iv.	16052.2112274	3877.77050781	259.756347656
iii.	16297.9644012	3501.07965088	867.102294922
ii.	16292.9105225	3514.70678711	883.551025391
i.	16822.1671295	3362.96630859	3370.28613281

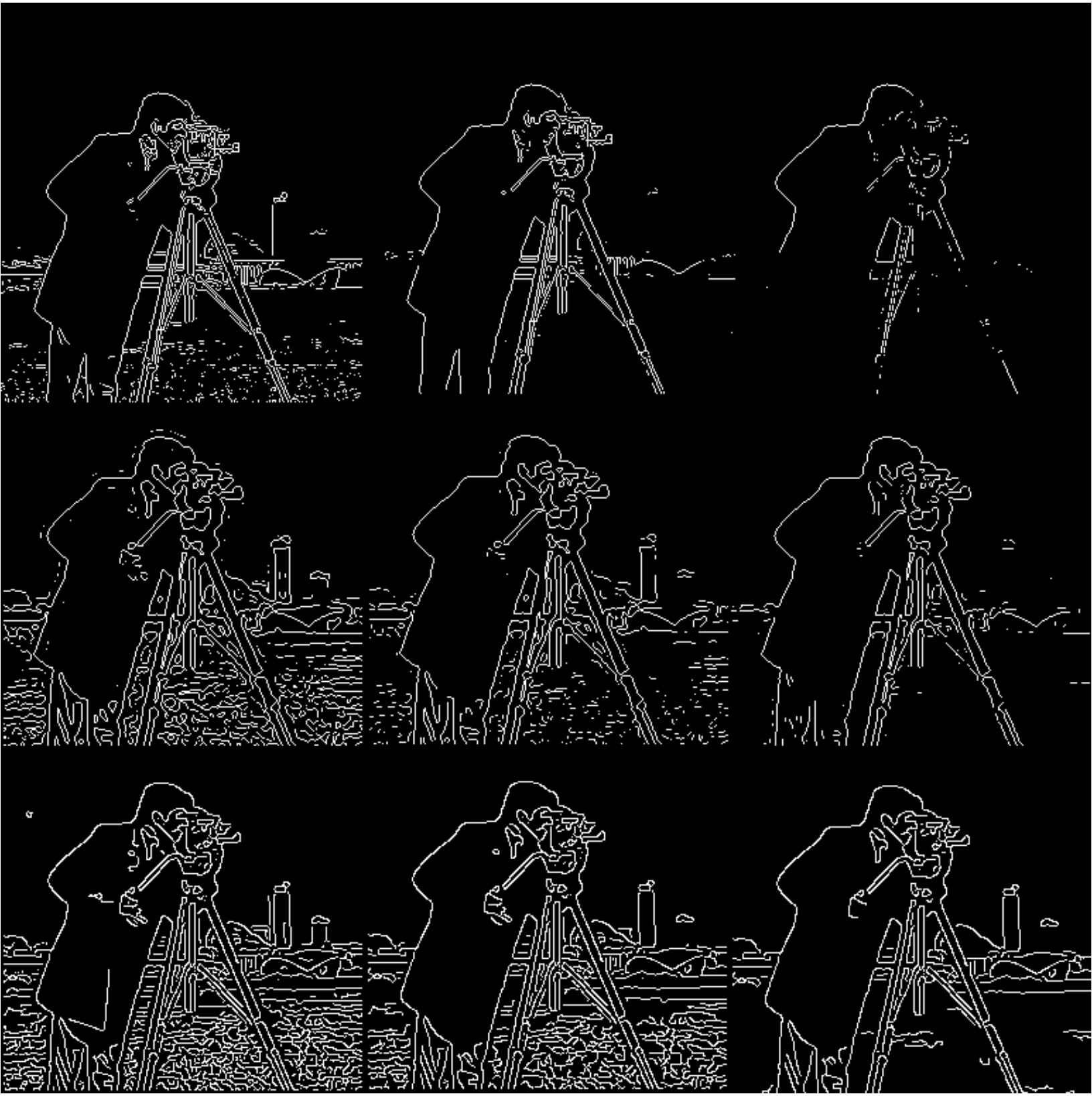
lighthouse	1	2	3
v.	14920.9052811	2602.87124634	651.192138672
iv.	14477.8607712	3496.52304077	230.006103516
iii.	14685.6674919	3182.27113342	788.322570801
ii.	14681.7614441	3188.94007874	797.257629395
i.	15172.8110313	3073.84780884	3068.74890137

moon	1	2	3
v.	15748.896225	2707.36590576	678.701171875
iv.	15236.0677185	3728.00512695	247.38671875
iii.	15495.5098267	3337.72790527	835.924804688
ii.	15494.3839417	3338.8682251	837.200927734
i.	16022.3848267	3206.07556152	3205.41430664

I am really not sure what to compare with the histograms, and the mean square numbers. They all look roughly the same. The tent filtered images have horizontal and vertical brighter and darker repeating lines, but that is probably because they aren't circular filters.

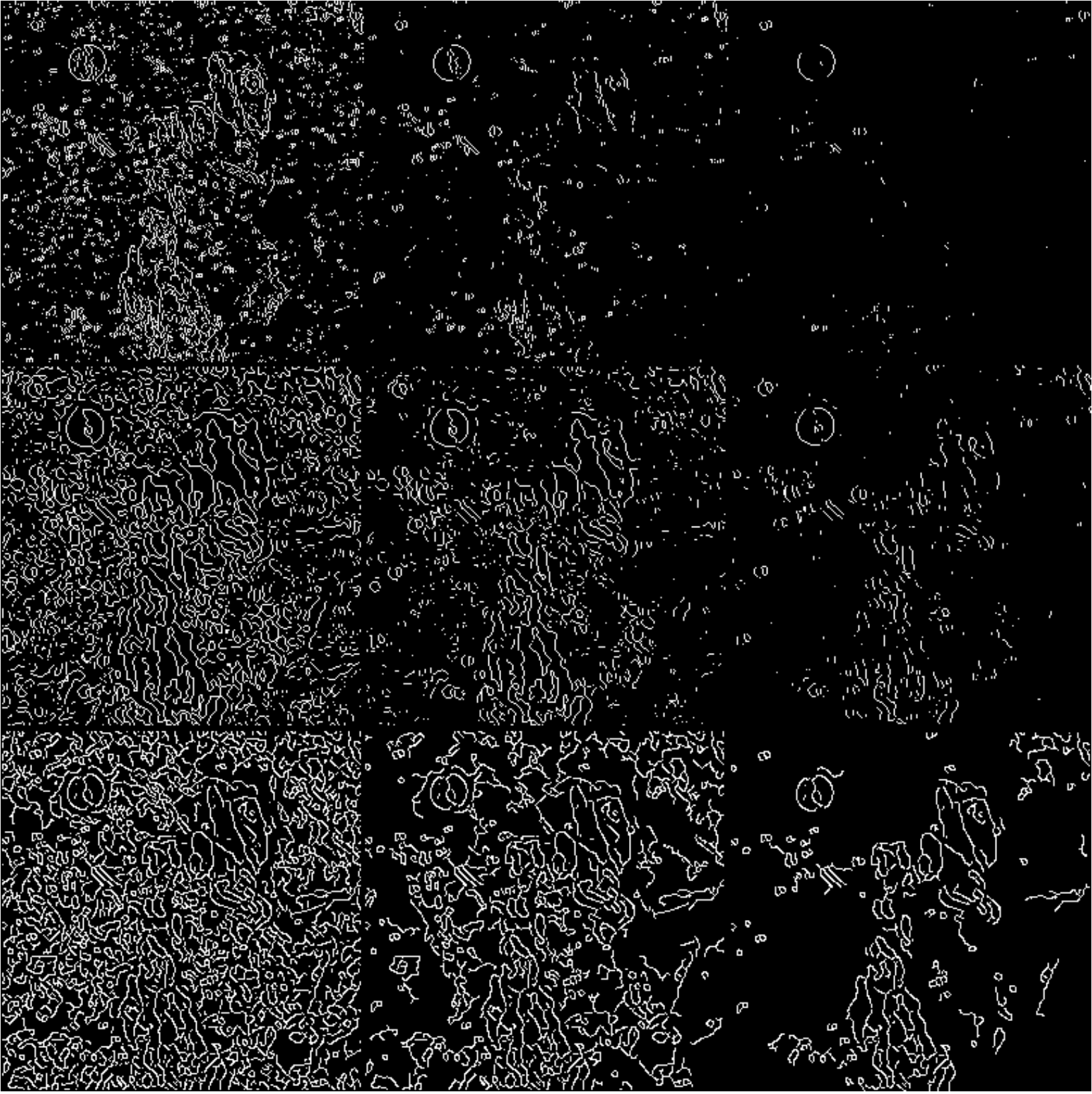
c. I would give the Gaussian filter rank 1 because it has fewer artifacts in the images, and looked the smoothest in downsampled and upsampled transitions. The FIR filter is the next best for the same reasons, then filter iii, filter ii, and finally no filter.

3. The images on in the column are one half the threshold, the middle column has the default threshold, and the images in the right column are double the threshold. Each image has the Sobel filter as the first row, the second row is the Log filter, and the last row is the the Canny filter. These images were actually done with Matlab because I needed the Matlab thresholds, so the lighthouse image is the correct resolution. The order of the images is cameraman, lighthouse, moon.



Cameraman Edges

Lighthouse Edges



Moon Edges

Decreasing the threshold by 2 increases the amount of detail the algorithms find, while increasing the threshold by 2 decreases the amount of detail the algorithms find. To be, I think that increasing the threshold by 2 removes the most amount of noise, while retaining enough information to still recognize the image. These attributes would be the best to use for object detection because it has the least amount of information and still has enough to depict the image. Sobel is the worst edge detector, but lowered threshold Log adds in a lot of extra noise because of the oscillations of the LoG function at values farther away from the mean. Canny can pick up on so much detail that the image begins to look like an art piece.