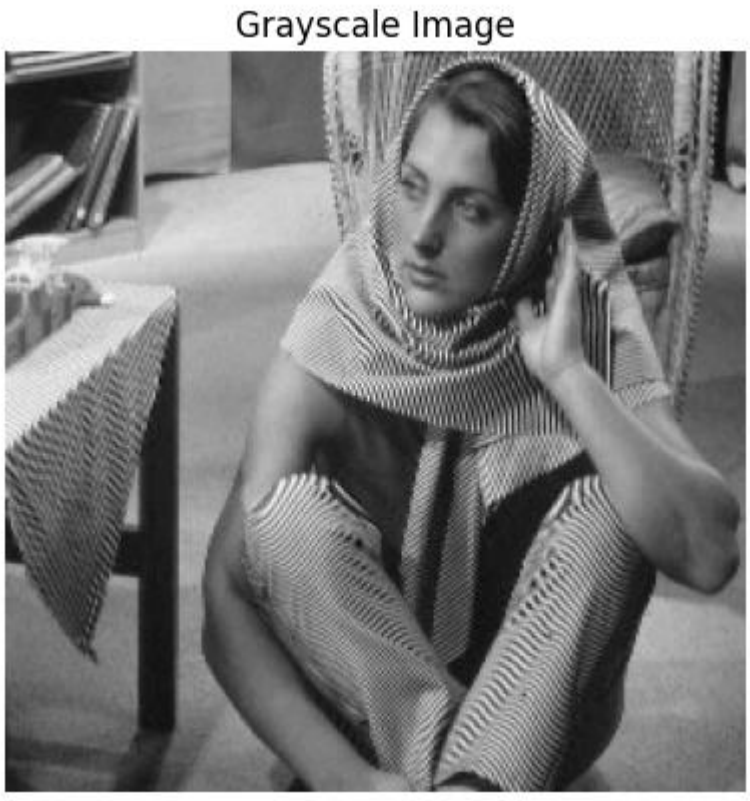
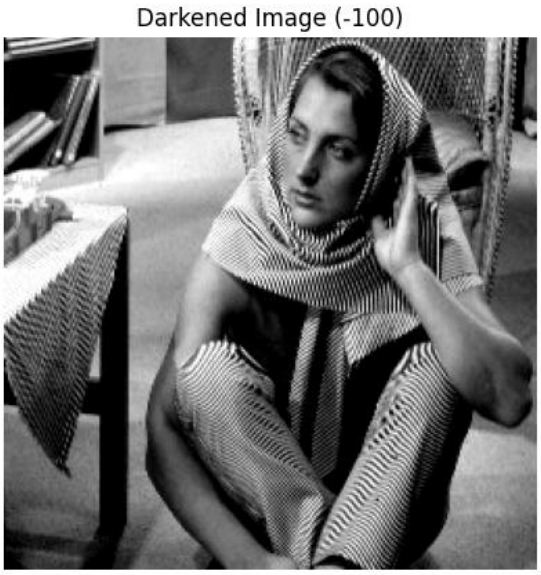
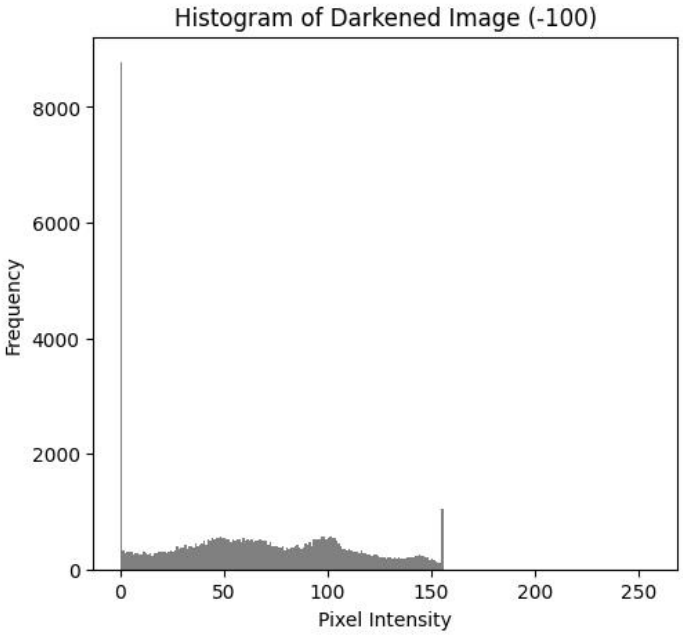
Lab 1

Task 1

Task 1 is about some general image programming (reading, changing to grayscale, resizing etc.) . Here are some of the images that were generated throughout the task 1.



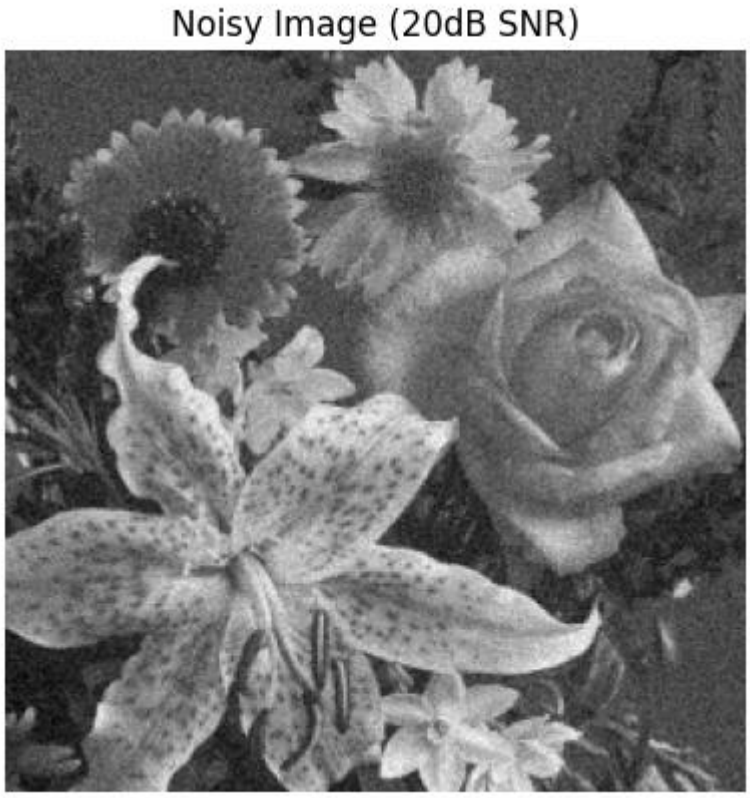




Task 2

Task 2 is about noise filtering and image restoration. The tasks included:

Corrupt an image with Gaussian noise of zero mean where the noisy image had a 20 dB Signal to Noise ratio. (This corresponds to a “somewhat” noisy image). This was then going to be removed using moving average and median filtering to remove the noise.



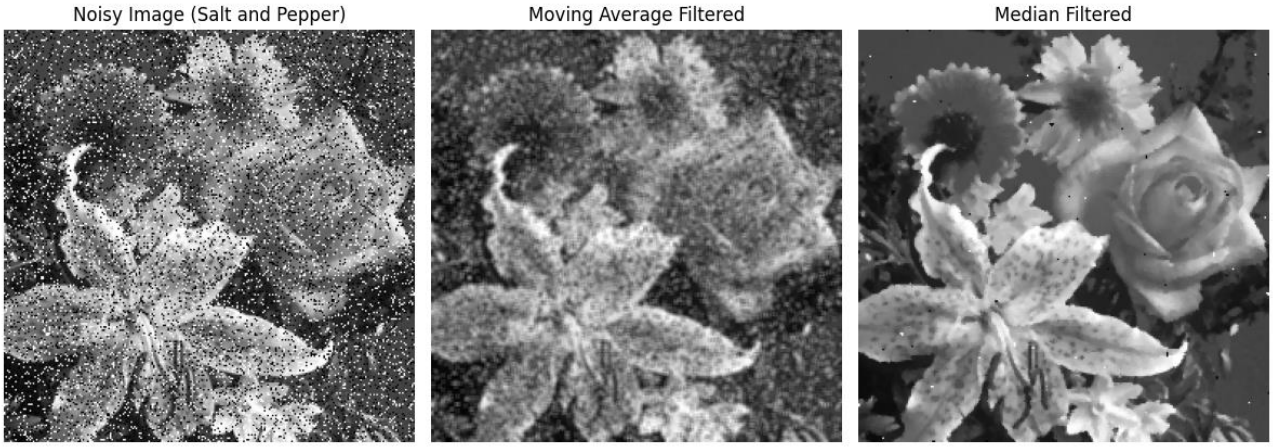
Moving Average Filtered Image

Median Filtered Image

As the pictures above show, is there a relvatively small difference amongst the three images. This is due to Gaussian noise has a “Gaussian distributed” (normal) noise. The image filtering

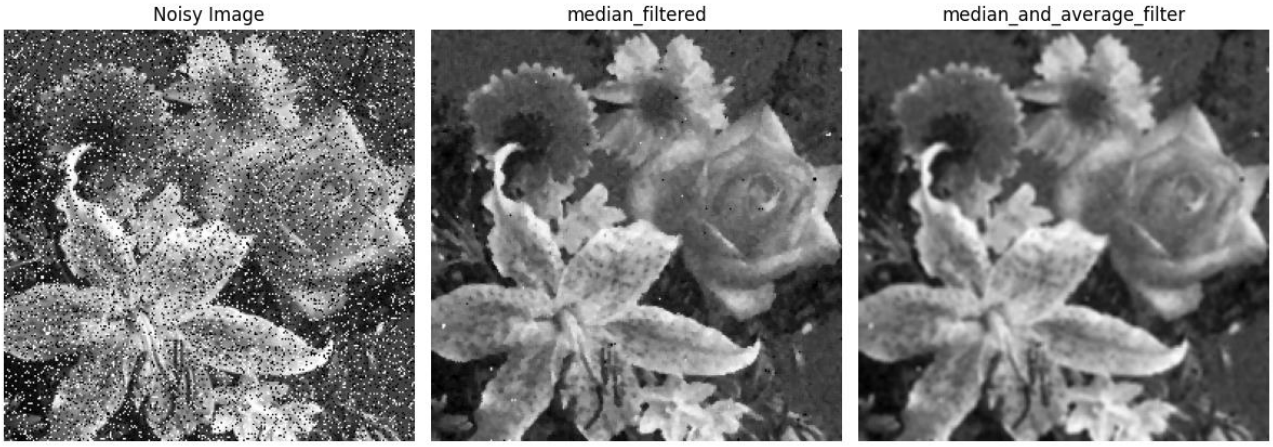
methods (median and mean) are all taking each kernel, and making the operation on all. This means that either of the filtering methods are not very efficient for these type of noise removal tasks.

Salt and pepper noise (randomly occurring black and white “dots” is another noise, and the second sub task was filtering with the same filtering methods mentioned above.



Here it becomes evident that median filtering is much more efficient against salt and pepper noise compared to Moving average. This is due to that the median filtering will take the median (and outliers wont matter) whereas the moving average will take the mean of all.

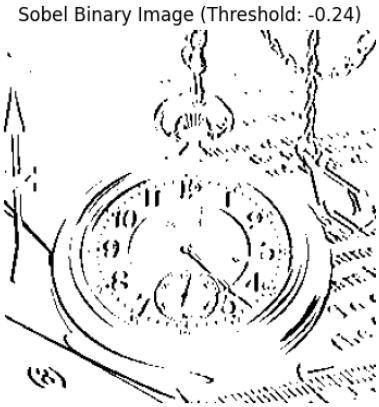
Lastly, the noises were applied sequentially (first gaussian noise then salt and pepper). After, the task was to pick filtering methods (either separately or sequentially in any order).

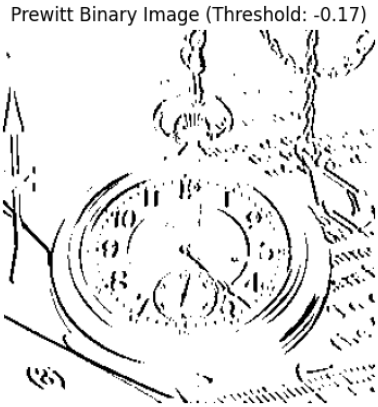


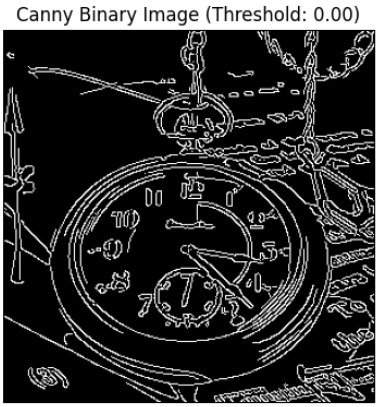
I chose to first use the median filter (to get rid of salt and pepper noise) and then use the median average filter. This resulted in the picture above, where the median and average filter is very similar to only using median average.

Task 3

Task 3 was about edge detection. Here I picked Canny, Sobel and Prewitt edge detection. Canny is a rather complex method, which detectes the edges in several stages. First it uses gaussian filters to reduce the noise, it then uses gradient calculations and edge tracking. Sobel and prewitt is similar, whereas Prewitt is simpler than Sobel. They both have kernels that detect both horizontal and vertical edges.







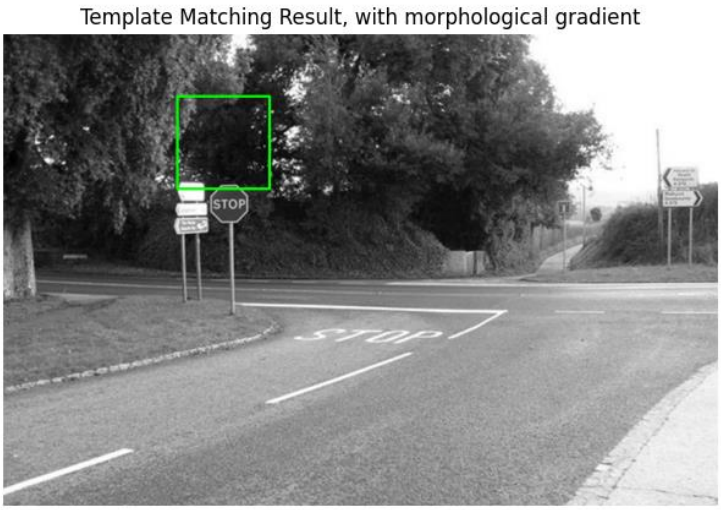
From the binary images above it is clear that Canny edge detection performs best of the three. Sobel and Prewitt performs similar, where many of the edges are not present.

Task 4

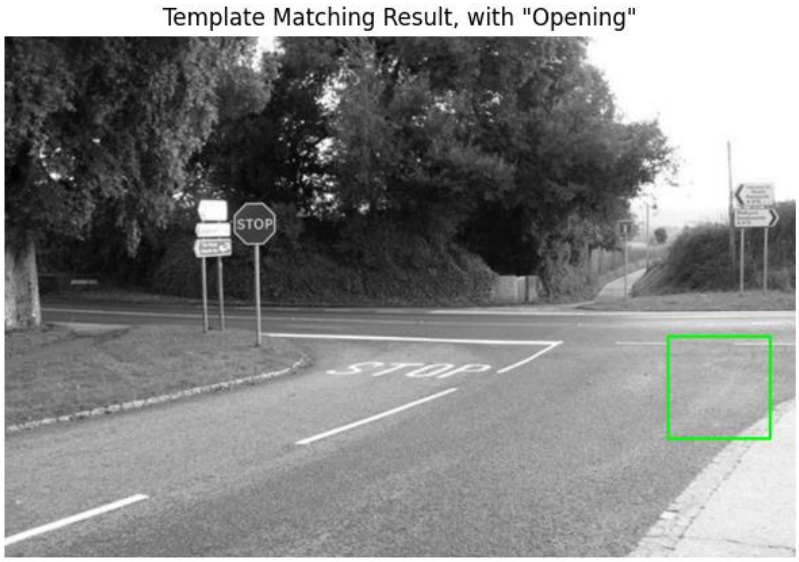
The fourth task was about template matching. Here is the image from where no morphological operation was applied.

The image clearly shows that the template is far from the actual image, and no edges of the template are detected.

When instead applying morphological gradient (which is Dilation (white areas expanded, black shrunken) subtracted by Erosion (shrinking of white regions and expands black regions)) the template is much closer to where it is in the image:



Here it can be seen that the template is detecting the edge of the actual image (although it is the top part of it). It is still much closer to the previous image.



The third image is from “Opening” (similar to Gradient, but the other way around). It can be seen that the template is very far away from where it actually is, and this is the worst results out of the three.