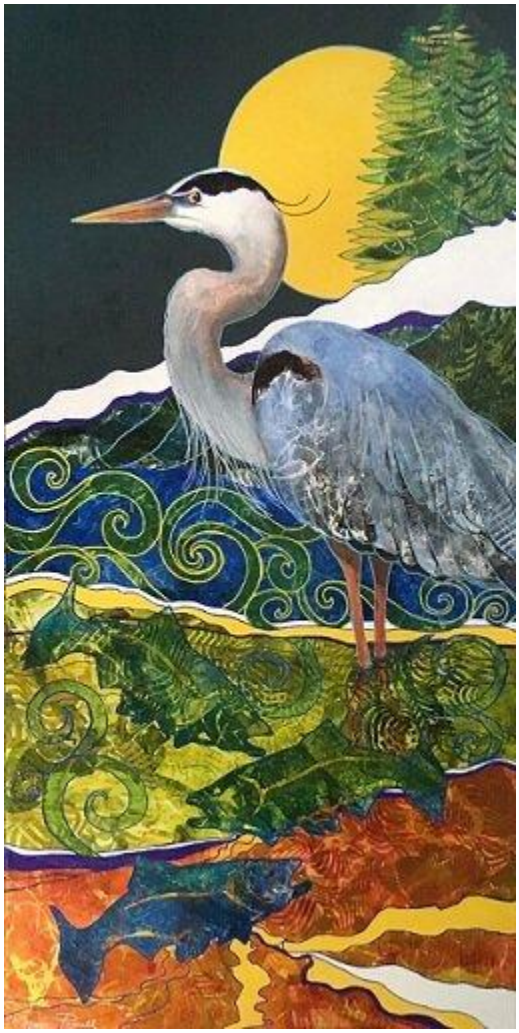
Jacob Kerstetter - PS0 Report

1a.



ps0-1-a-1.png

ps0-1-a-2.png

ps0-2-a-1.png

ps0-2-b-1.png

ps0-2-c-1.png

2d. I believe the red channel looks more like an expected monochromatic image. This is because side-by-side with the green channel image, you can make out more of the objects in the red channel image. For a computer vision algorithm, I would expect the red channel to perform better as there are more defined edges between objects due to a harsher transition of color.

ps0-3-a-1.png

ps0-3-b-1.png

4a. I computed these values using the numpy built-in methods. These methods flatten the 2D-array of the channel and compute the corresponding values.

```
#----- Problem 4 ----- #
# compute and print statistical values
maxGreen = np.max(green)
minGreen = np.min(green)
meanGreen = np.mean(green)
stddevGreen = np.std(green)
'''
print('Max of Green Pixels:', maxGreen)
print('Min of Green Pixels:', minGreen)
print('Mean of Green Pixels:', meanGreen)
print('Std Dev of Green Pixels:', stddevGreen)
'''
```

```
# import image 1
img1 = cv.imread('output/ps0-1-a-1.png')

# get channels from img1 and swap red and green
green = img1[:,:,1].copy()
```

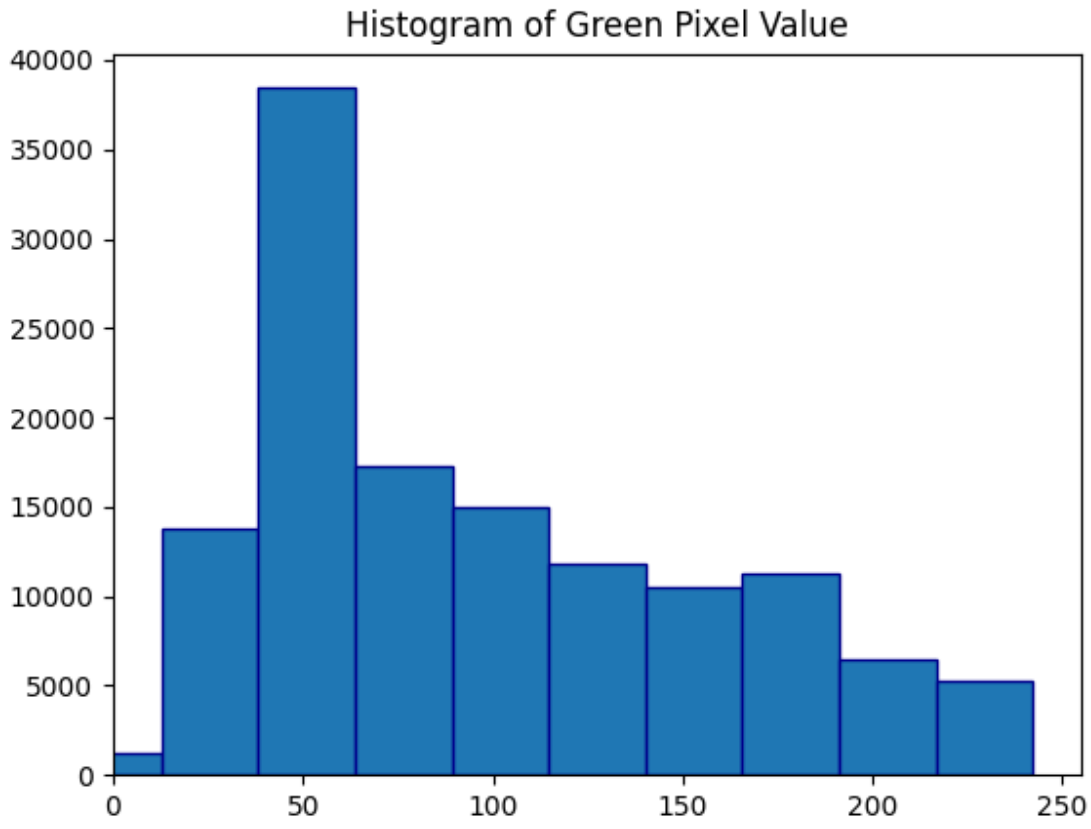
Max of Green Pixels: 255

Min of Green Pixels: 0

Mean of Green Pixels: 109.70905303955078

Std Dev of Green Pixels: 59.34478618969718

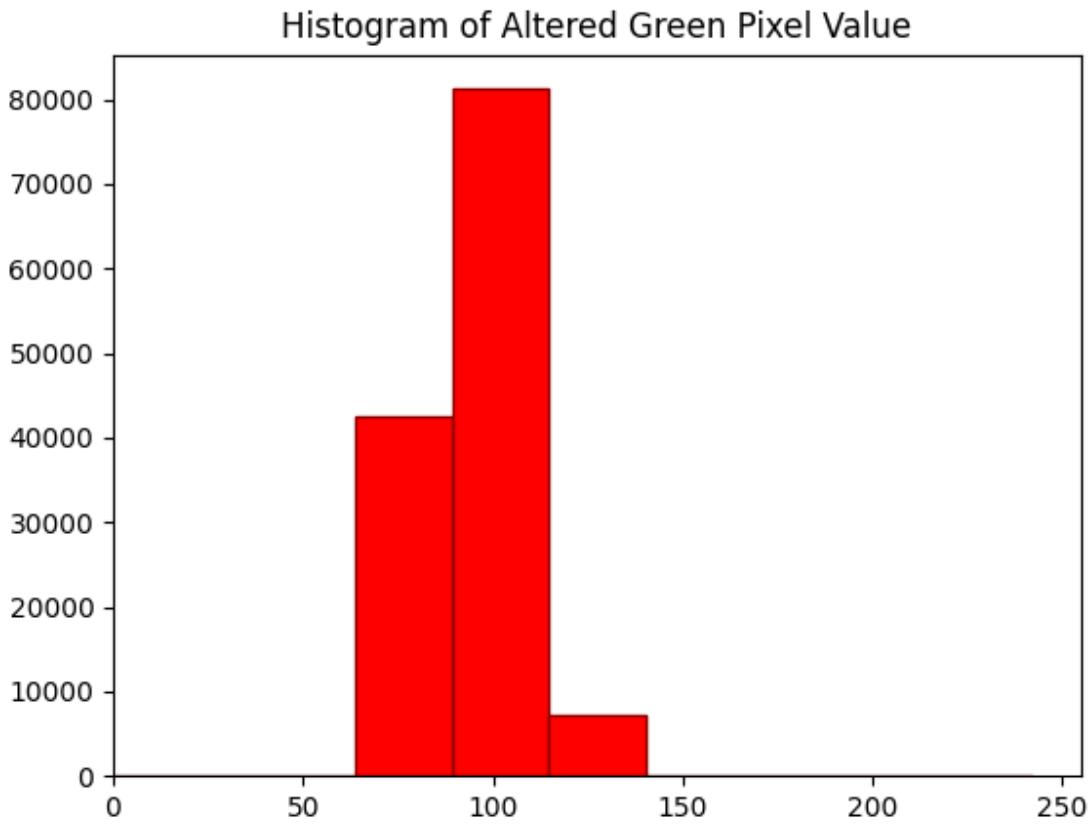
4b. The image histogram shows the frequency that a pixel contains a value in each bin range. Each bin is about 25 pixels in size, and because the majority of pixels fall in the lower intensity bins, this means the green channel in the image generally has lower values.



ps0-4-b-1.png

ps0-4-c-1.png

4d. The histograms in both plots are of a similar shape, but the resultant image histogram has been condensed in the range of values of its pixels.


Histogram of Altered Green Pixel Value
80000
70000
60000
50000
40000
30000
20000
10000
0
0
50
100
150
200
250

ps0-4-d-1.png

ps0-4-e-1.png

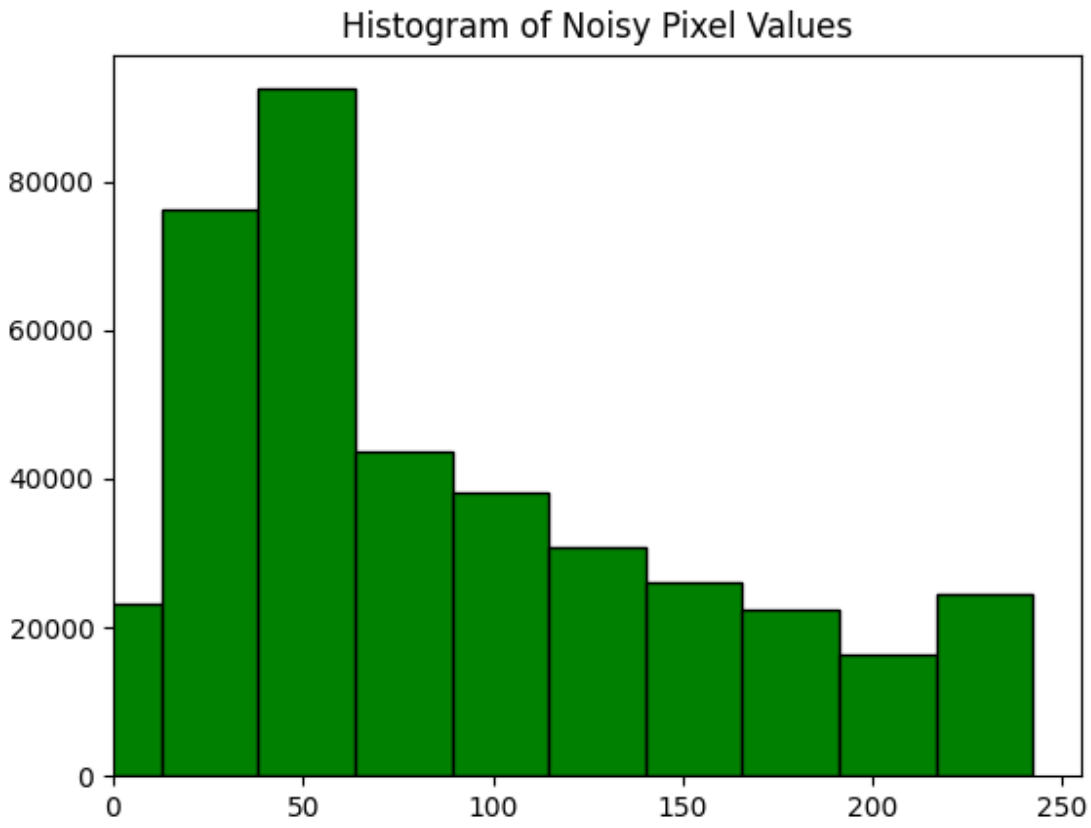
4f. Negative values are portrayed as black as their value is smaller than 0 (black, lowest intensity).

ps0-4-f-1.png

5a. The noise was noticeable at any value greater than 1 for sigma, but at a value of 10 for sigma, the noise became very prevalent.

ps0-5-a-1.png

5b. The relative levels of the bins are approximately the same, but the bins to the left and right of the most occurring bin both have higher frequency. The most frequent bin also has slightly decreased frequency. This is due to the Gaussian noise distorting the values of the most frequently occurring pixels to the nearby bin values.


Histogram of Noisy Pixel Values
80000
60000
40000
20000
0
0
50
100
150
200
250

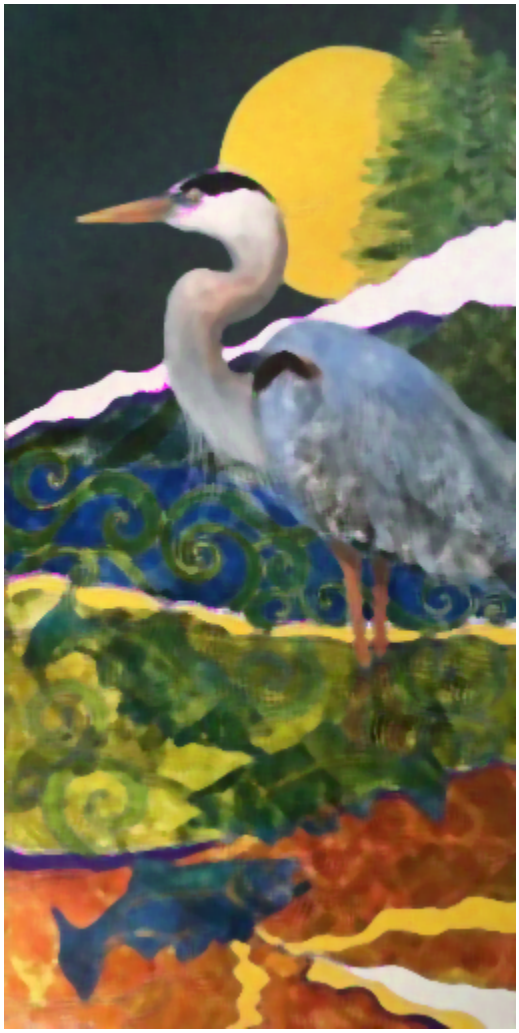
Ps0-5-b-1.png

ps0-5-c-1.png

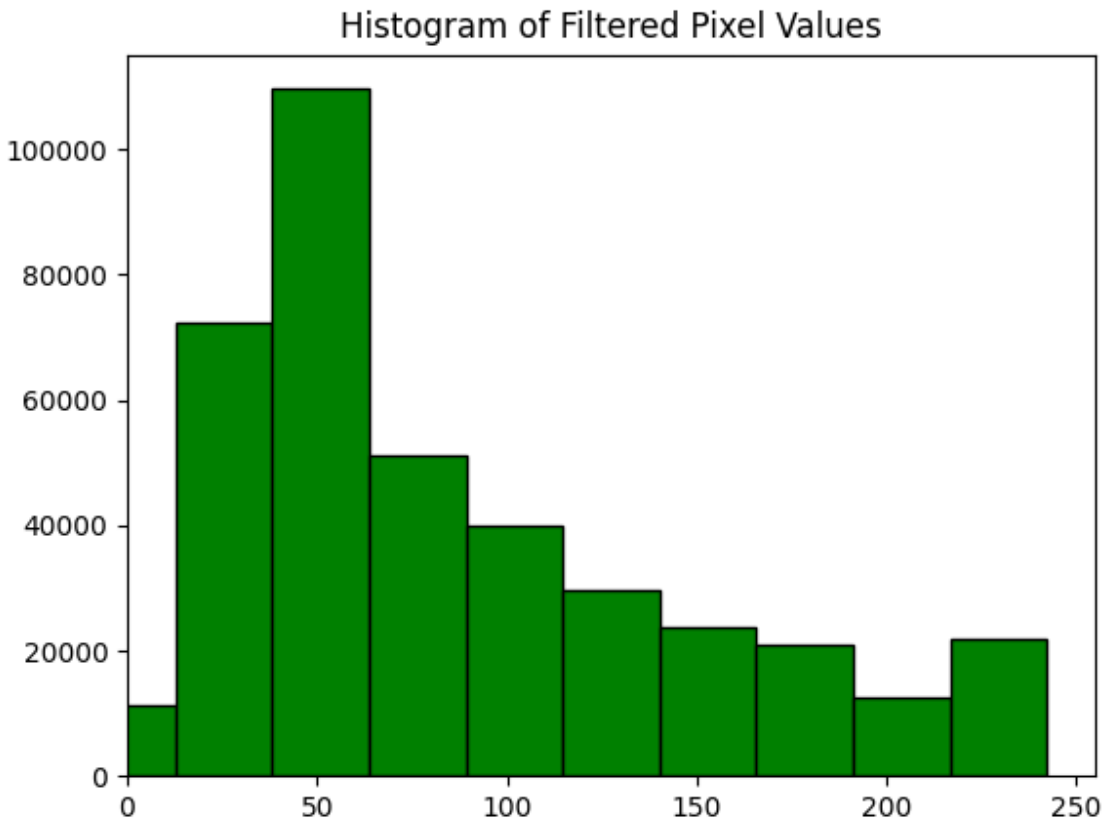
5d. The image with blue channel noise looks better in most areas. This is because the pixels that are white in the original image become more yellow, which is not a big distinction to the human eye, while in the image with green channel noise they become pink. In the originally green areas of the image, the image with blue channel noise looks worse while the image with green channel noise looks better in these areas.

5e. The output image has most of the colors of the noisy pixels corrected, but is smoothed/blurred. This is because the filtering finds a median of the pixels nearby to attempt to fix the noise, in turn smoothing edges between contrasting colors. The histogram of the filtered image has a higher peak at the 50 bin range and decreased peaks next to it, which aligns with the

idea that the noise was affecting this bin heavily. This histogram also matches that of the original image closely.



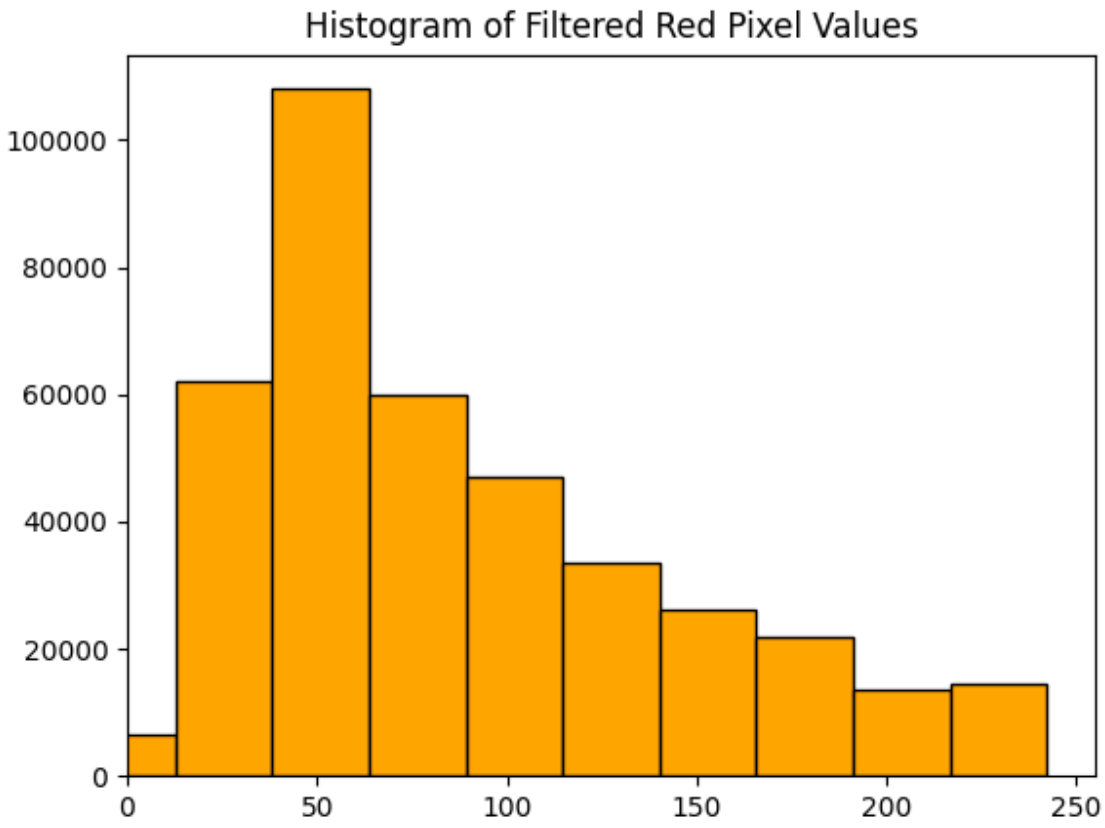
ps0-5-e-1.png



ps0-5-e-2.png

5f. The Gaussian filter does not take away much of the noise entirely. This filter only smooths the noise so it is less intense and less noticeable compared to the original noisy image. The histogram of the Gaussian filtered image compared to the noisy image has one peak bin rather than two and has the values more centered around that peak with a steady rate of decline as the bin value increases. Overall, the median filter works better to erase the noise from the image and restore original pixel values.

ps0-5-f-1.png


Histogram of Filtered Red Pixel Values
100000
80000
60000
40000
20000
0
0
50
100
150
200
250

ps0-5-f-2.png