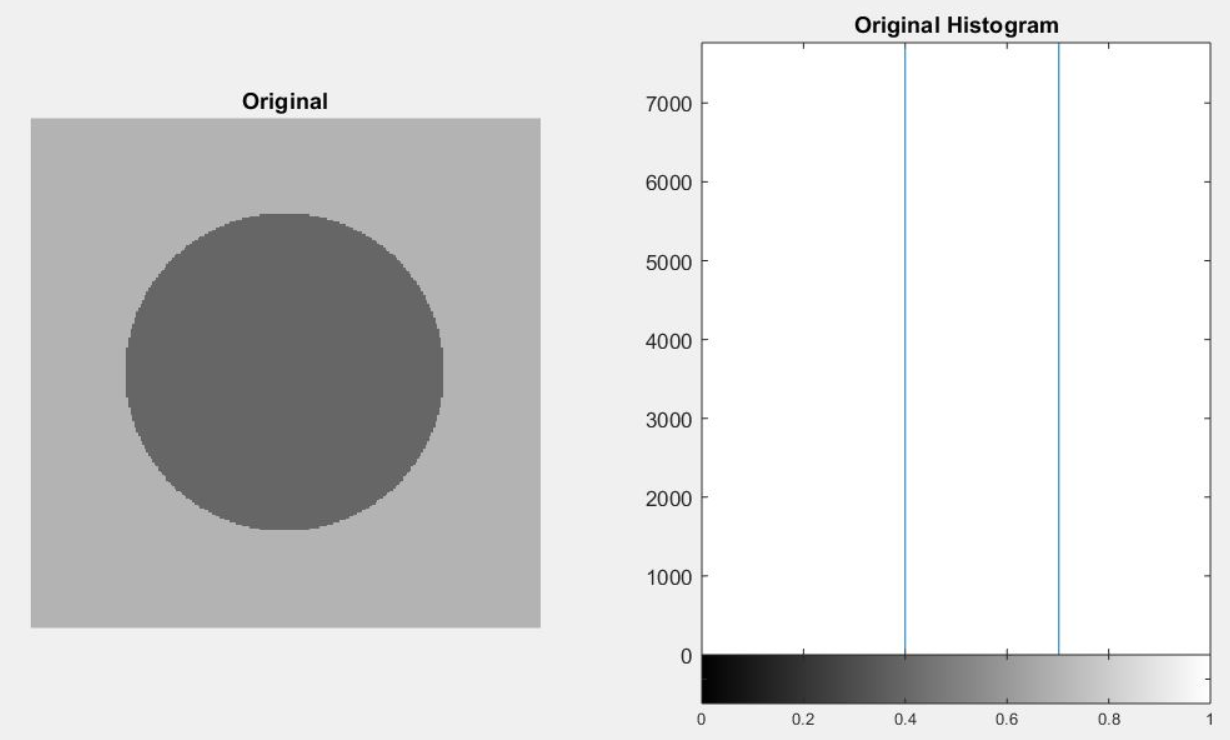
Kevin Valenzuela

October 20, 2018

Digital Image Processing

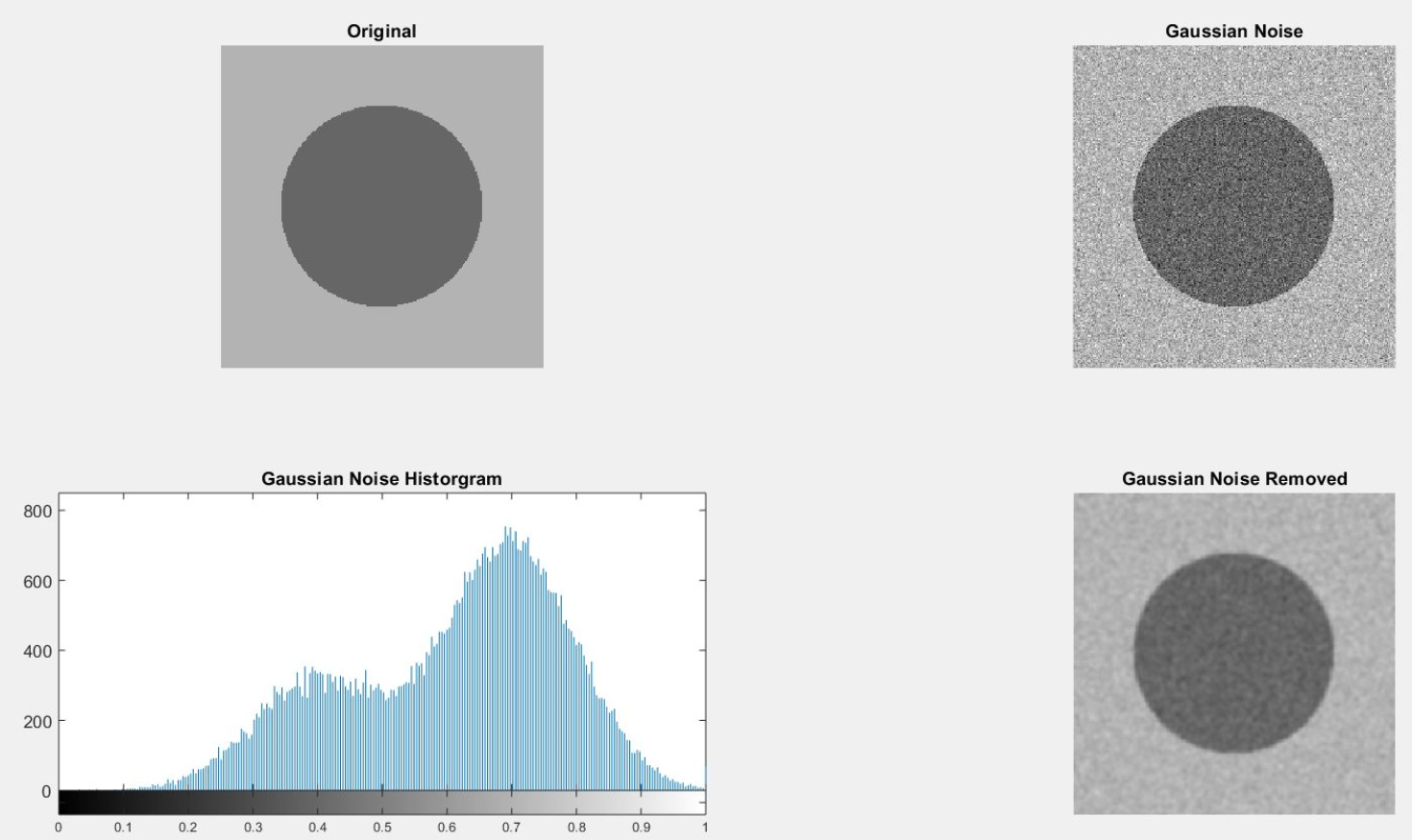
Image Restoration

For this assignment, I generated an image of a circle with the circle having an intensity of 0.4 and the area outside of that having 0.7. The image and its histogram can be seen below.



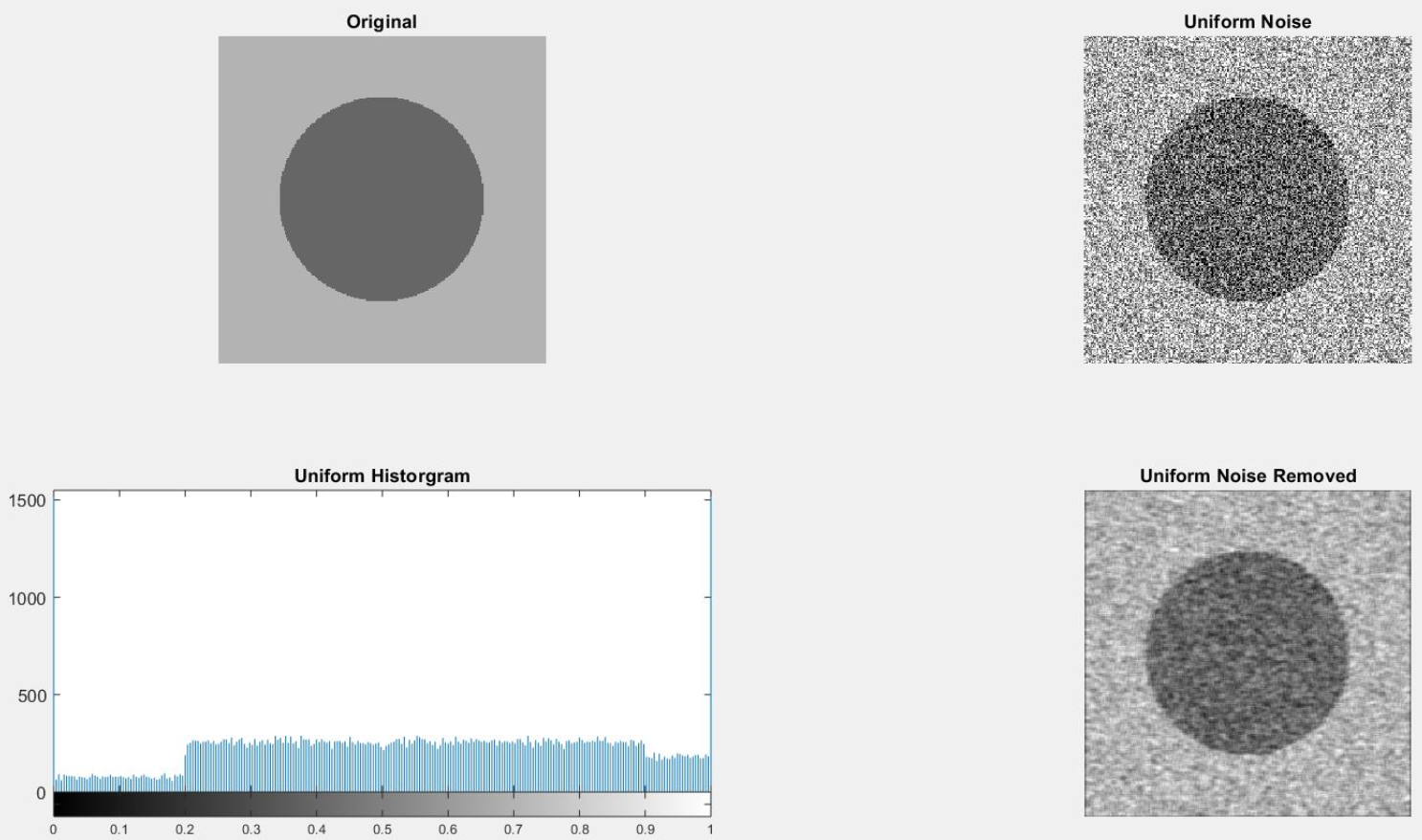
I added several types of noise to this original image. The noises that I added are Gaussian, Uniform, and Salt & Pepper noise. On the next sections I will show each of these noises added to the original image along with the noise and the histogram after the noise was added.

**Gaussian Noise**



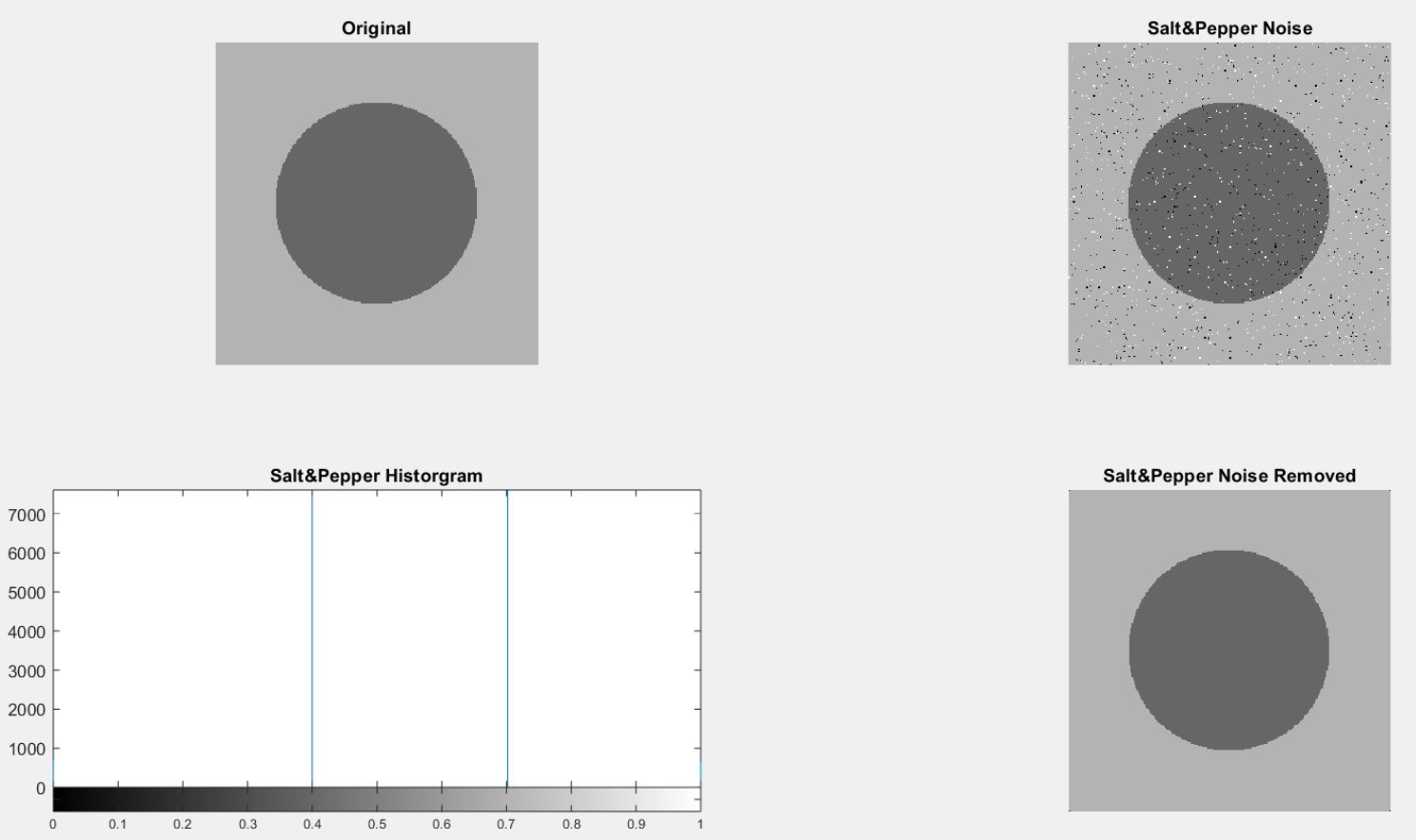
Gaussian noise can be reduced using a spatial filter. When smoothing an image, we reduce not only the noise, but also the fine-scaled image details because they also correspond to blocked high frequencies. The most effective basic spatial filtering techniques for noise removal include mean filtering, median filtering and Gaussian smoothing. I used Gaussian smoothing for this image.

**Uniform Noise**



The uniform noise is caused by quantizing the pixels of image to a number of distinct levels is known as quantization noise. It has approximately uniform distribution. I made it by creating a matrix of the same size as the input image and filling it with random values, after that I added it to the original image. In the uniform noise the level of the gray values of the noise are uniformly distributed across a specified range. Uniform noise can be used to generate any different type of noise distribution. I was able to fix some of this noise by using an **averaging** filter.

**Salt & Pepper**



Another common form of noise is data drop-out noise (commonly referred to as intensity spikes, speckle or salt and pepper noise). The noise is caused by errors in the data transmission. The corrupted pixels are either set to the maximum value (which looks like salt in the image) or have single bits flipped over. In some cases, single pixels are set alternatively to zero or to the maximum value, giving the image a `salt and pepper' like appearance. Unaffected pixels always remain unchanged. The best way I was able to restore the image was by using a **median** filter.