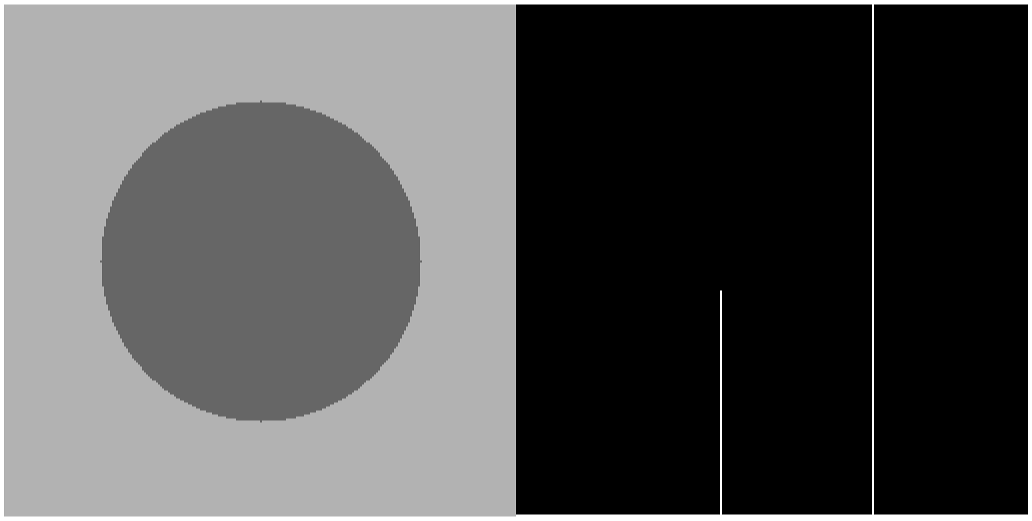
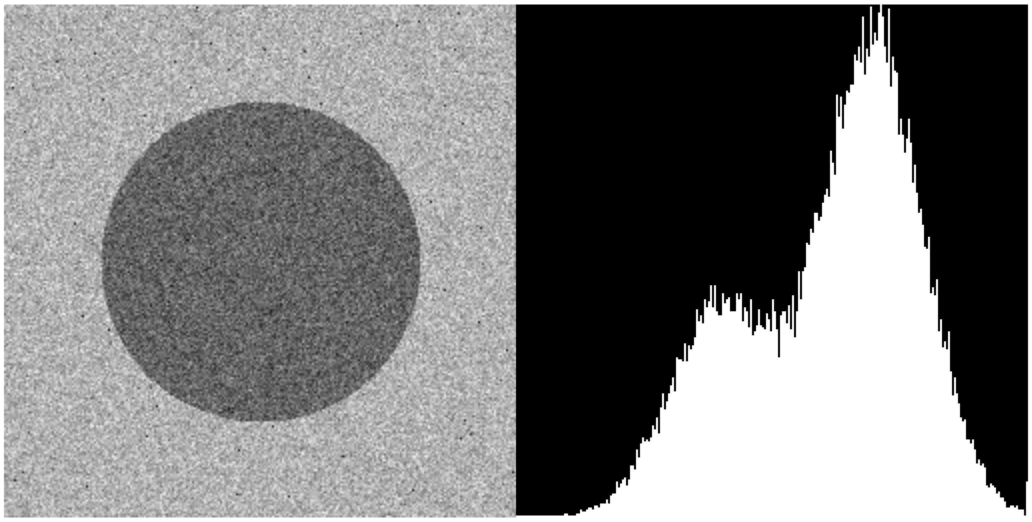
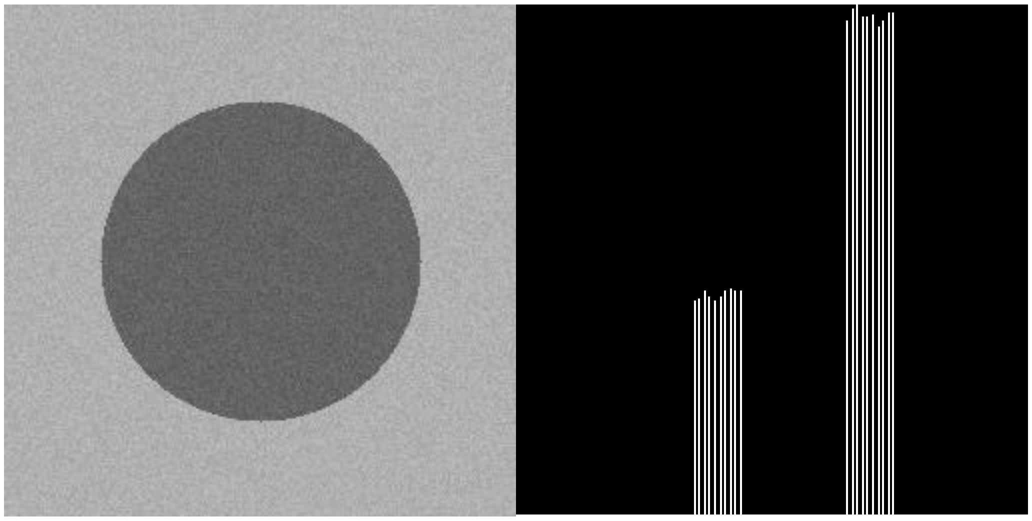
Nate Roberts
Homework 4

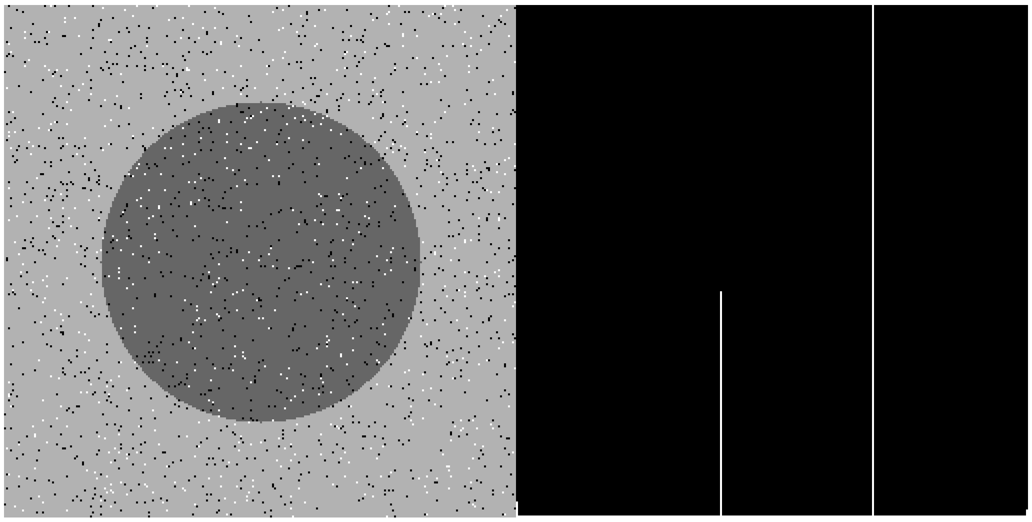
In order to demonstrate the generation of digital image noise and the use of filters to remove or reduce it, I first used Python to first generate a base two-tone image to work from. I also wrote a method to create a visualization of an image's histogram.


Basic two-tone image

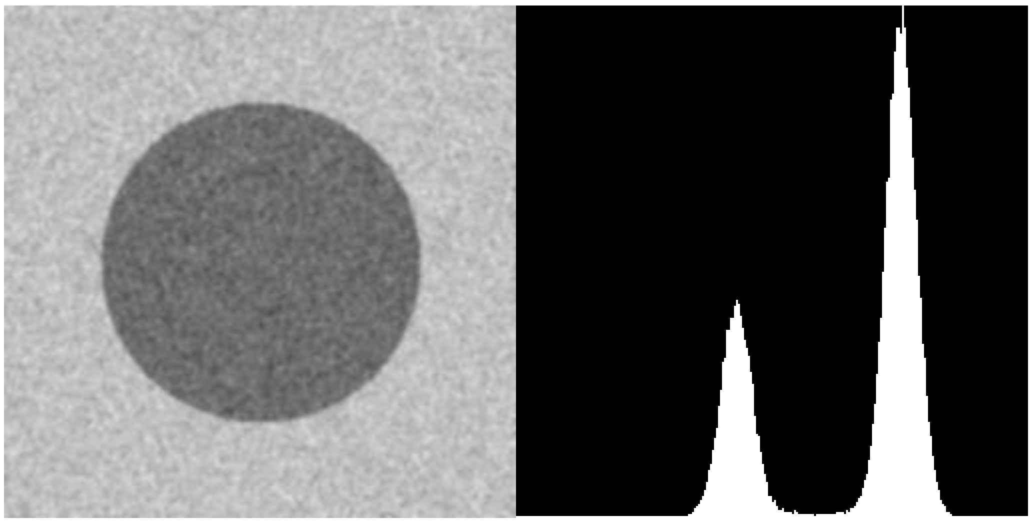
Next, I added several common types of noise to the base image. To generate variant values, I used NumPy's Random module. I modified the pixel values of the base image by a random value in the normal distribution to produce Gaussian noise, by a random value of uniform distribution to produce Uniform noise, and a random value of uniform distribution to select a given percentage of all pixels to be black or white for Salt & Pepper noise.

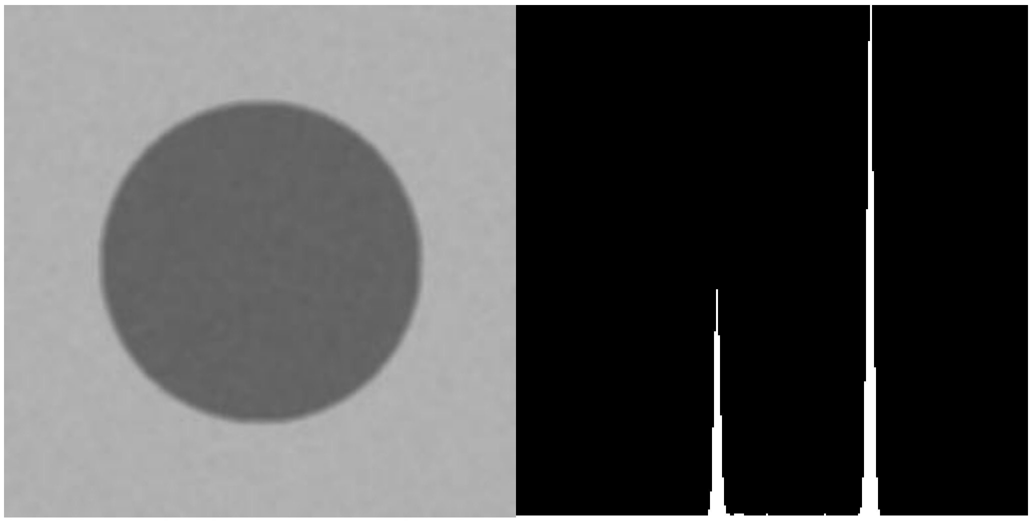

Gaussian noise added; mean 0, variance 0.01

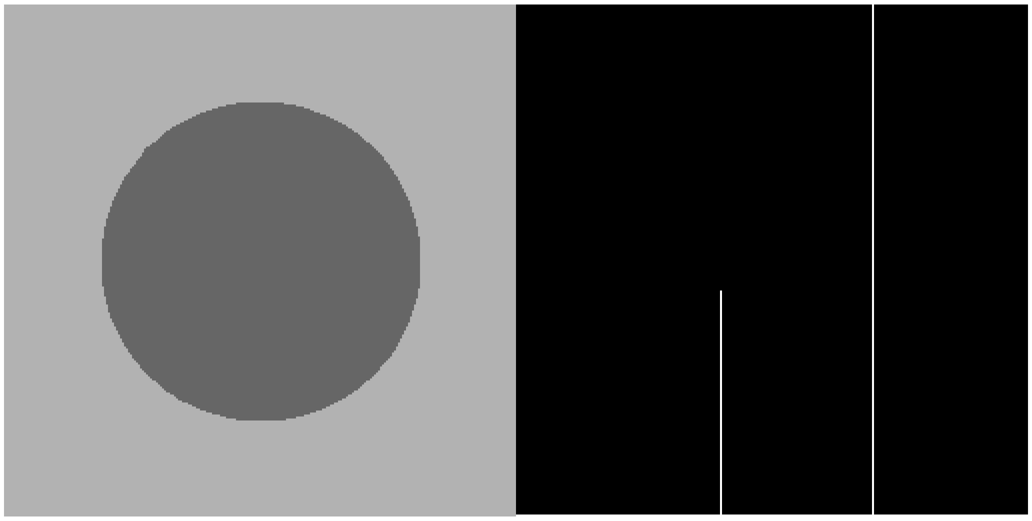

Uniform noise added, range -0.05 to 0.05


Salt & Pepper noise added, d = 0.02

I then used different filtering methods to attempt to return the noisy images to their original values. Each spatial filter used a 5x5 pixel neighborhood to do comparisons. On the image with Gaussian noise, I used a Gaussian blur; for the image with Uniform noise, I chose a geometric mean filter; and for the Salt & Pepper noise, I used a median filter.


Gaussian noise after Gaussian blur


Uniform noise after Geometric Mean filter


Salt & Pepper noise after Median filter

The Gaussian blur filter was not as effective at removing the Gaussian noise as I had expected, but as shown by the histograms, the Gaussian noise effected the greatest change on the base image, and it does show noticeable improvement in tone uniformity. The geometric mean filter applied to Uniform noise was more effective, returning quite smooth tones, with only a minor softening of edges. The most successful filter was the median applied to S&P noise – it returned the image very nearly to its original state. Of course, this degree of precision owes to the simplicity of the base image, but the filter was nevertheless remarkably effective.

Project written in Python;
Packages used: cv2 (OpenCV), NumPy, math