Laura Kissack
V00844693

ECE 435 Assignment 1

1.

a) Refer to AddNoise.m, OrgImg.ong, and GaussianNoise.png files included. This function reads a DICOM image and saves the first slice as a .png. It then generates a noisy image by adding a random Gaussian distribution to the original. This final image is saved. Both images are displayed.

b) Information will be lost no matter what when writing to a .png since the data is stored as integers. Converting from double will result in lost information.

c) Using the provided 'RIDER_Lung_CT.cdm' file, the following images were produced. Both images are fairly similar, however the second noisy image is slightly darker than the first and seems to have lower contrast within the noise.

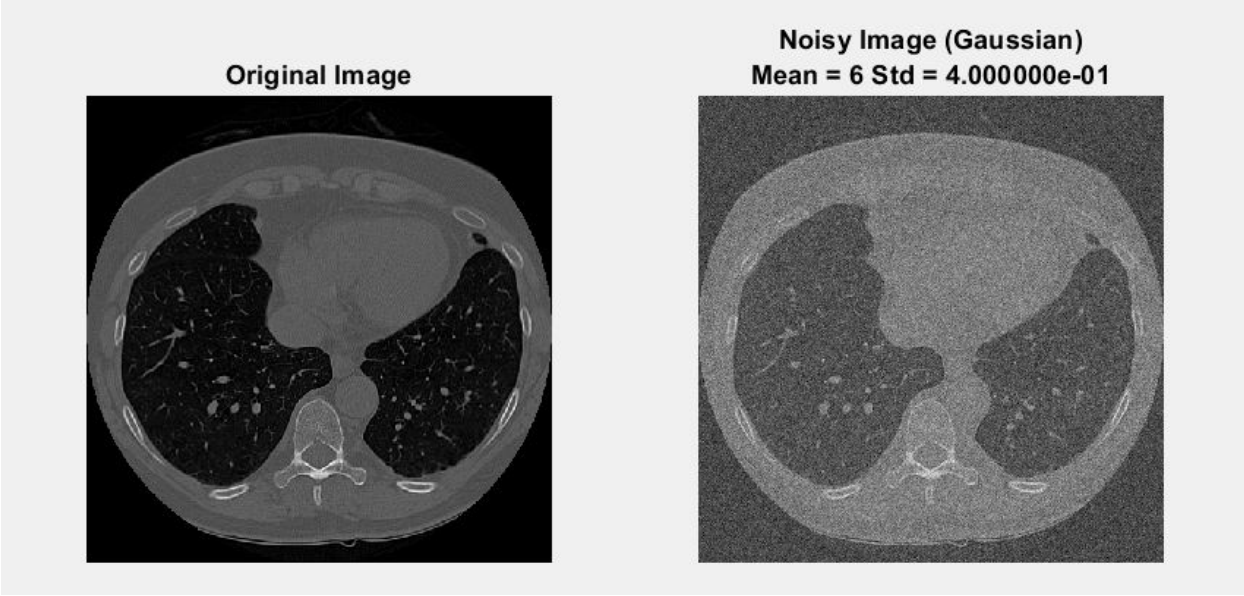

Figure 1: Original CT scan compared to scan with added Gaussian Noise (mean = 6, std = 0.4)

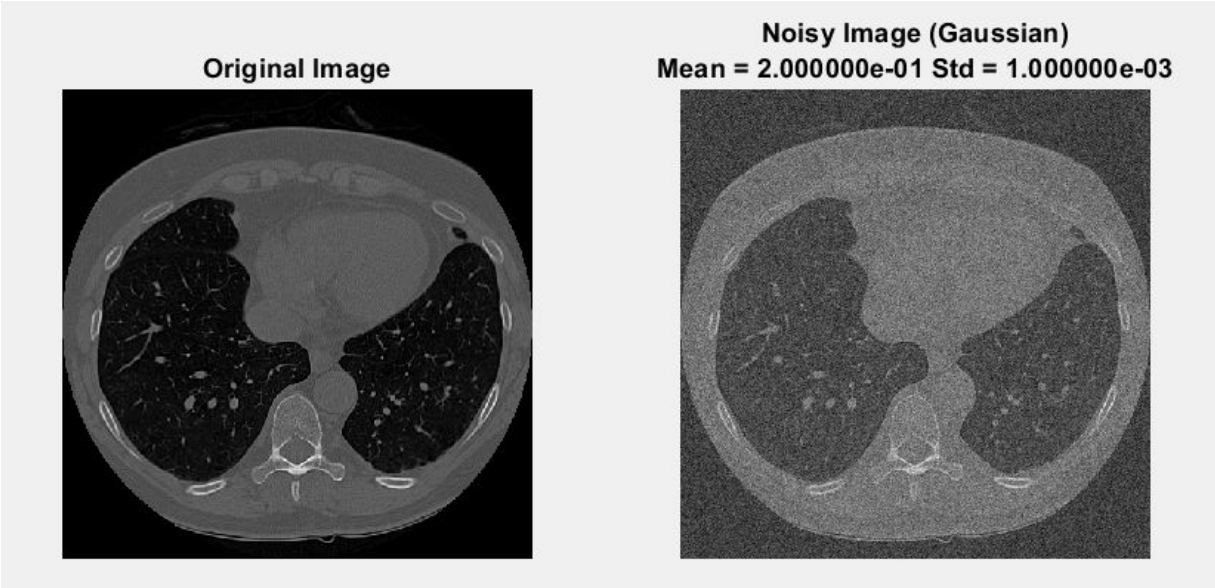

Figure 2: Original CT scan compared to scan with added Gaussian Noise (mean = 0.2, std = 0.001)

d) There are 51 slices in the RIDER_Lung_CT.dcm file. Each slice is has 512 rows and 512 columns. This was determined using the following MATLAB commands:

```
[rows, cols, samples, slices] = size(image)

rows =
   512

cols =
   512

samples =
       1

slices =
       51
```

2.

a) Refer to AddNLNoise.m, speckle.png, and 'salt & pepper.png'. This function adds the specified noise type (speckle or salt and pepper) to the first slice of a DICOM image using the pre-defined MATLAB imnoise() function. It also calculates the MSE between the original and noisy images. Finally both images are plotted together for comparison.

b) The function AddNLNoise was used to generate four noisy images (Figures 3 to 6). The .png files of Figure 4 and 6 are included as 'speckle.png' and 'salt & pepper.png' respectively. Higher noise characteristics corresponded to higher MSE values.

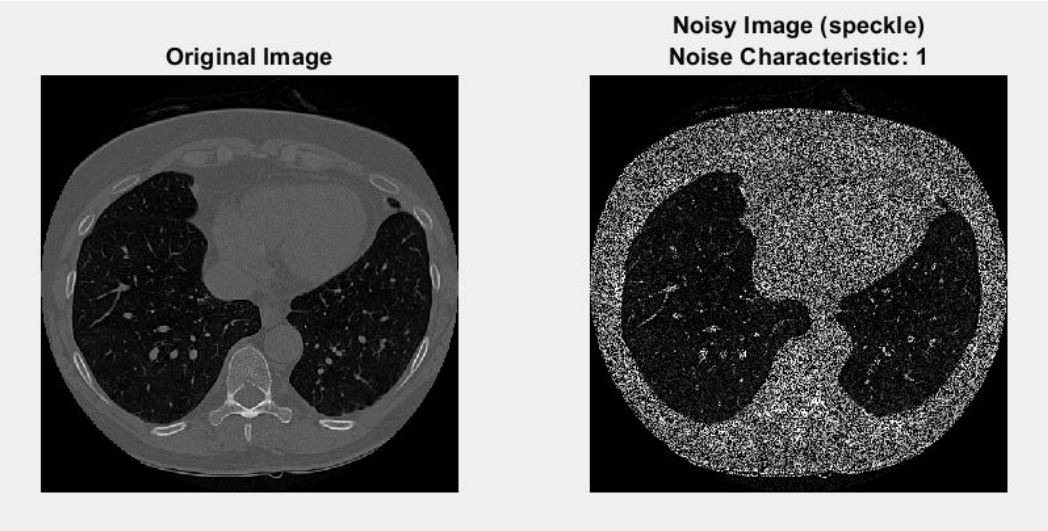

Figure 3: Original CT slice compared to CT slice with added speckle (variance = 1). MSE = 0.0449

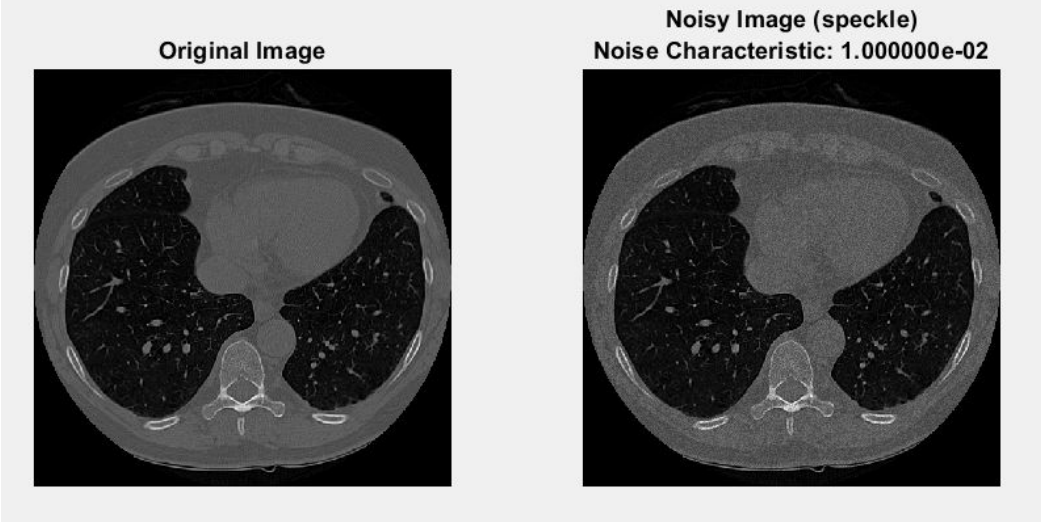

Figure 4: Original CT slice compared to CT slice with added speckle (variance = 0.01). MSE = 5.8415e-04

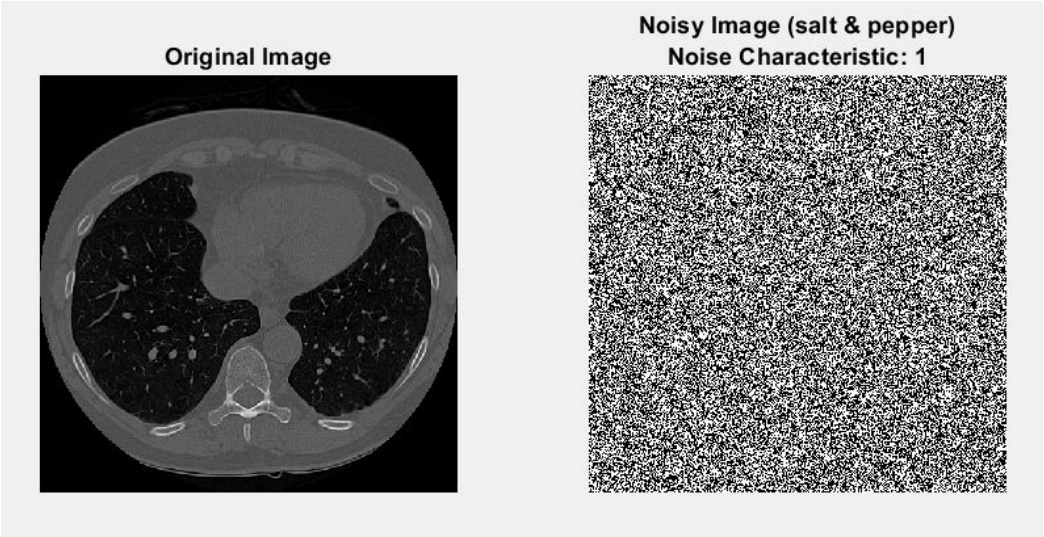

Figure 5: Original CT slice compared to CT slice with added salt and pepper (density = 1). MSE = 0.3813

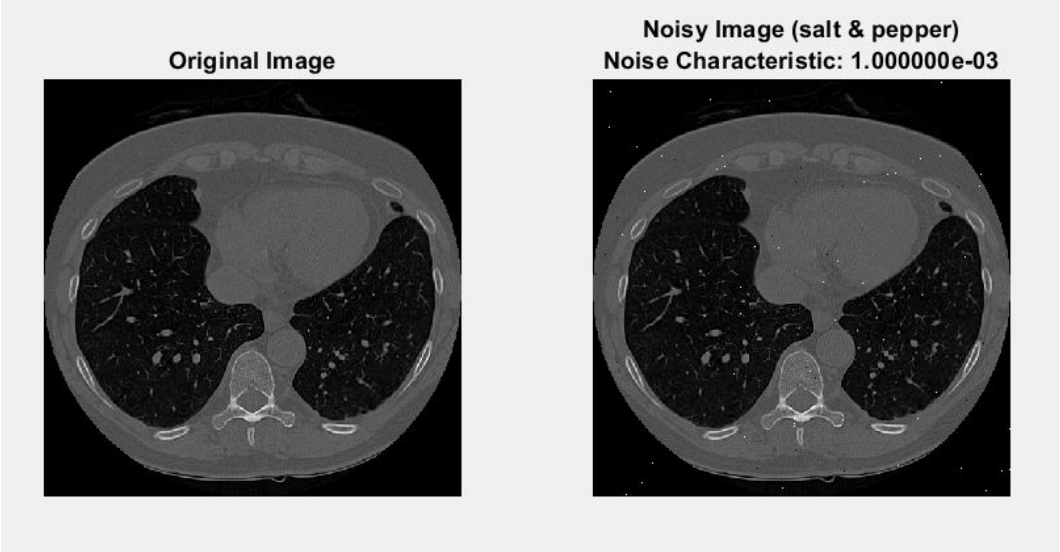

Figure 6: Original CT slice compared to CT slice with added salt and pepper (density = 0.001). MSE = 3.6942e-04

3.

a) For both the convolution and correlation filtering the image was zero-padded. The following commands were used to generate the correlated and convoluted images respectively.The script can be found in the MATLAB function CorrVsConv.m included.

```
>> Fhor = [-1 0 1; -1 0 1; -1 0 1];
>> image = imread('ChestXray.png');
>> correlated = imfilter(image, Fhor,'corr');
```

>> convoluted = imfilter(image, Fhor,'conv');

The resulting images were then plotted next to each other.

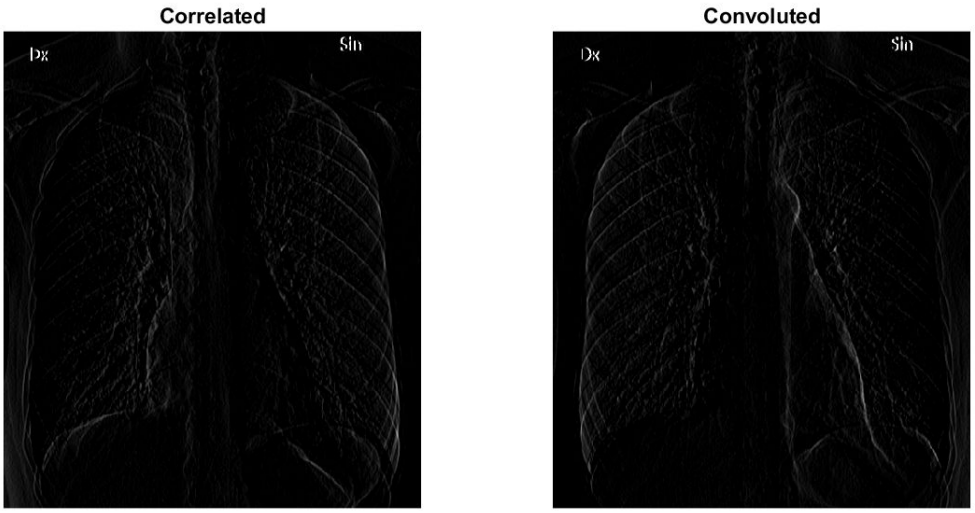

Figure 7: Correlated and Convoluted versions of the ChestXray.png

The images are very similar in appearance however different features have greater contrast. The convoluted image shows greater contrast of the left half of the chest and other structures on the right half of the rib cage. The correlated image shows the right edges more strongly than the left.

This is due to the kernel used in the convolution and correlation. The kernel detects the edges of the structures. Both images show detected edges, the depicted edges seem to depend on the orientation of the kernel (convolution = left, correlation = right)

b) Convolution is better for filtering because it is associative which allows multiple filters to be applied and generate the same result. Correlation is not associative, therefore the order that filters are applied impacts the final result.

4.

a) Refer to the GaussianFiltering.m file included. This function compares the original image and a noisy one through the MSE. The function then applies a Gaussian Filter to the original image by padding the image and performing convolution with a Gaussian Matrix. The result is compared with the original with the MSE.

   However, this since the original image is filtered (rather than the noisy one), the same final image will be generated independent of the noisy image selected. To see how

Gaussian Filtering impacted the different noisy images (part c), another function GaussianFilteringModified.m was created to apply the filter to the noisy image.

b) For this function, there was no difference between convolution and correlation. To convolute the image the filter is rotated 180 degrees. However, since the filter is a gaussian filter, the rotation does not change the result, meaning that correlation and convolution are the same.

c) Running the GaussianFiltering function on the three noisy images produced in part one, with n = 10, and std = 2, produced the same image. For all three images, the MSE increased after filtering.

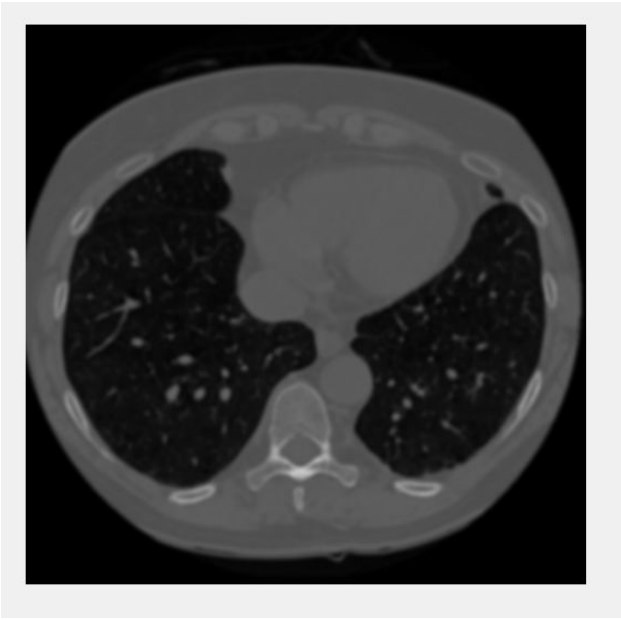

Figure 8: Resulting Gaussian Filtering of first slice (n = 10, std = 2)

When applied to the Gaussian Noise image (GaussianNoise.png). MSE pre = 0.0577, MSE post = 0.0064. When applied to the second speckle image (speckle.png), the MSE increased from 5.8617e-04 to 0.0064. Similar results also occurred with the salt and pepper noise image (salt & pepper.png. The MSE increased from 3.6942e-04 to 0.0064.

When the function GaussianFilteringModified was applied to the three noisy images different results were produced. When applied to the Gaussian Noise, the MSE decreased from 0.0577 to 0.0550. The resulting filtered image is shown in Figure 9. The Gaussian filter was 2 x 2 with a standard deviation of 0.5.

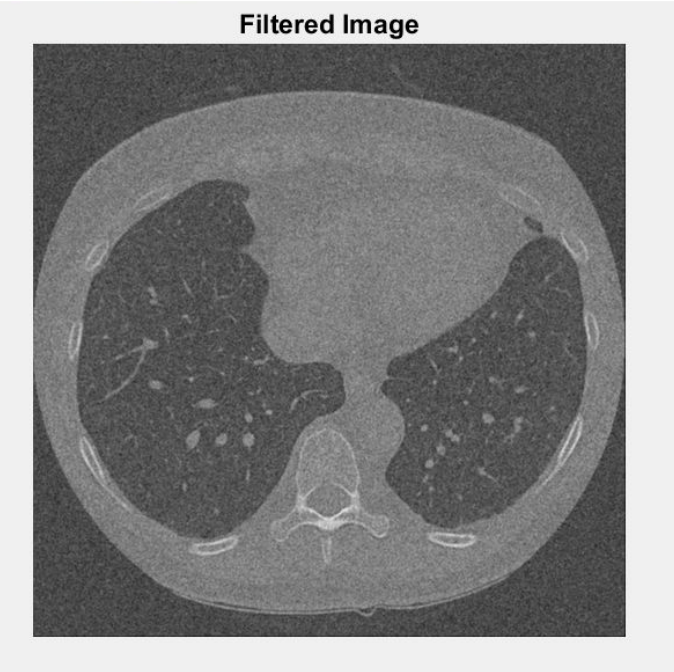

Figure 9: Gaussian Noise image after Gaussian Filtering

When applied to the ‘speckle.png’ image, the MSE increased from 5.8617e-04 to 9.1294e-04.

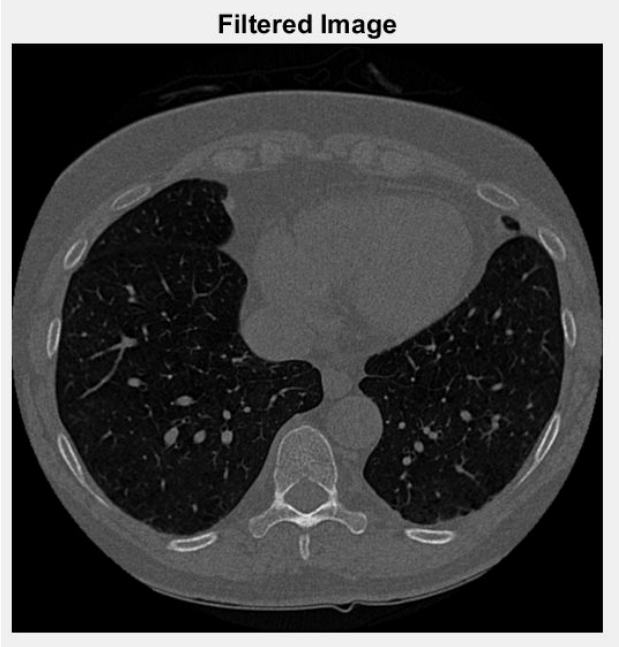

Figure 10: Speckle noise image after Gaussian Filtering

When applied to the ‘salt & pepper.png’ image, the MSE increased from 3.6942e-4 to 8.5967e-04. The filter size and standard deviation were also adjusted to 2 and 0.5 respectively.

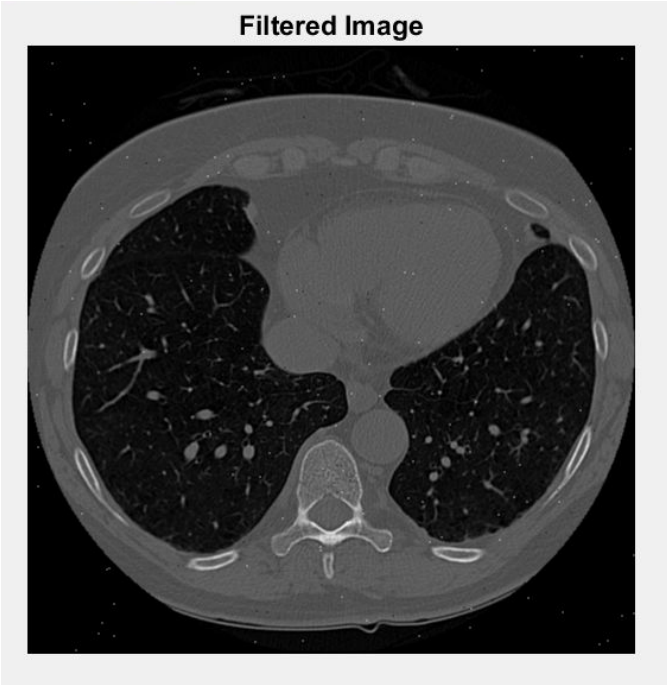

Figure 11: Salt and Pepper noise image after Gaussian Filtering

d) Refer to NLFiltering.m included. This function applies an n x n median filter to the provided image using the MATLAB function nlfilter.

e) When the NLFiltering is applied to the Gaussian Noise image, the image is still quite noisy but had less contrast within the noise. Increasing the size of n reduced the contrast of the noise.

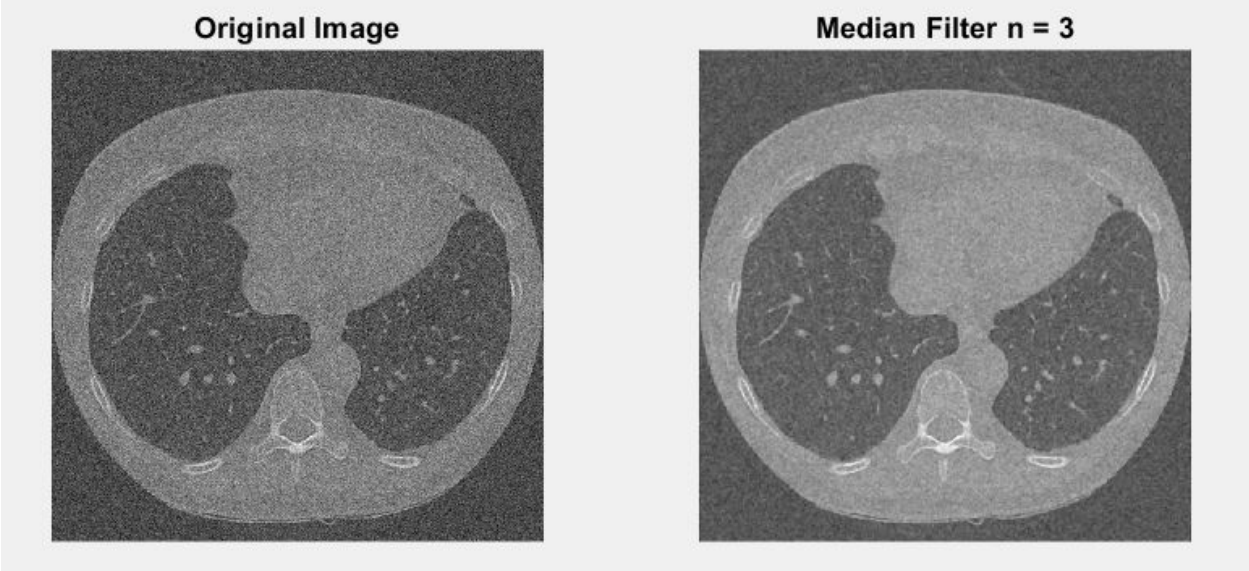

Figure 12: NLFiltering applied to the Gaussian Noise image

Applied to speckle.png, the result is not as grainy. However, upon closer inspection the edges of the different structures are blurrier.

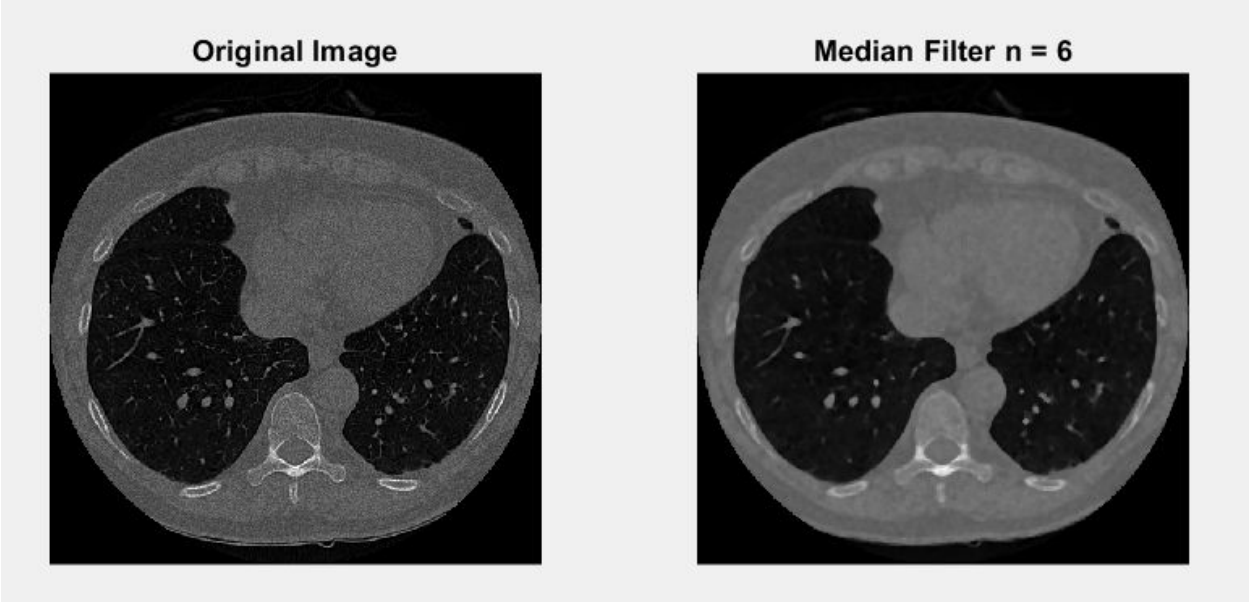

Figure 13: NLFiltering applied to the speckle noise image

When applied to the salt and pepper image, the filtering removes nearly all of the ‘salting’. The edges between the structures are blurrier.

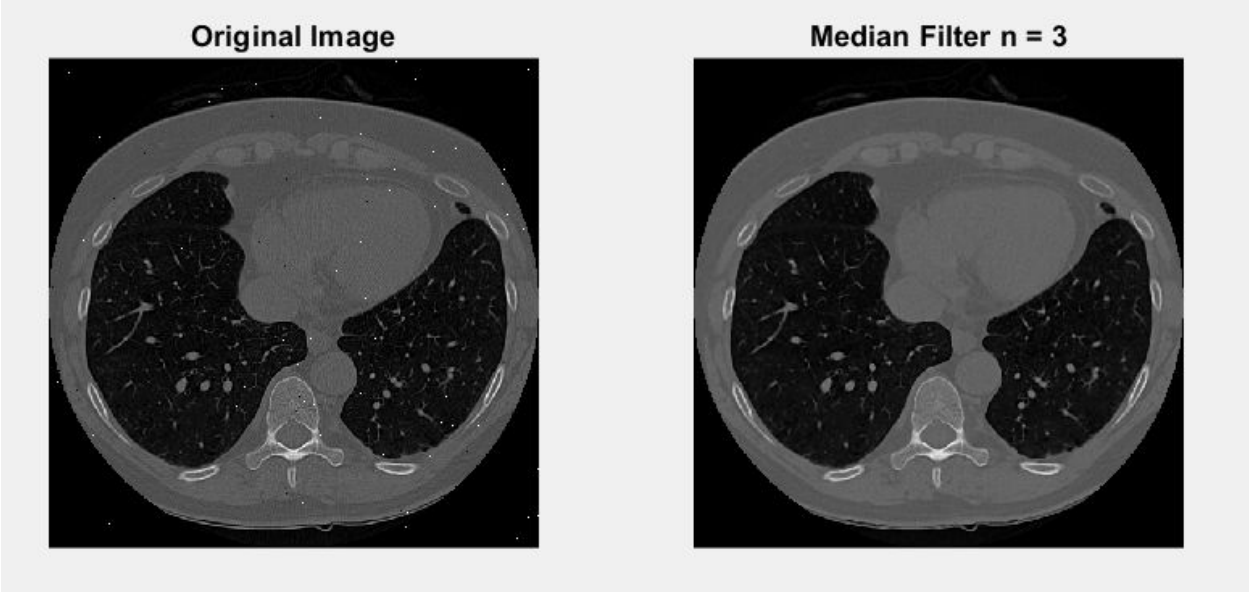

Figure 14: NLFiltering applied to salt and pepper noise.

f) All three MATLAB functions nlfilt, colfilt and blockproc filter an image based on a provided function. nlfilt applies the function to each sliding block of provided size. colfilt

applies the same function to chunks of specific size and attempts to use less memory by placing the region in a column of a temporary matrix. blockproc filters each section of the image separately and concatenates them at the end.

g) Refer to included function SpeckleDen.m. For large speckle variance levels, above 0.2, the temporal averaging method reduces the effect of speckle noise. This can be seen below. The MSE decreased from 0.0144 to 0.0034.

Figure 15: The first slice of the CT compared to temporal averaging on 30 slices with speckle variance of 0.5

For low values of speckle variance, the temporal averaging dramatically increases the MSE. The MSE increased from 3.0245e-04 to 0.0029.

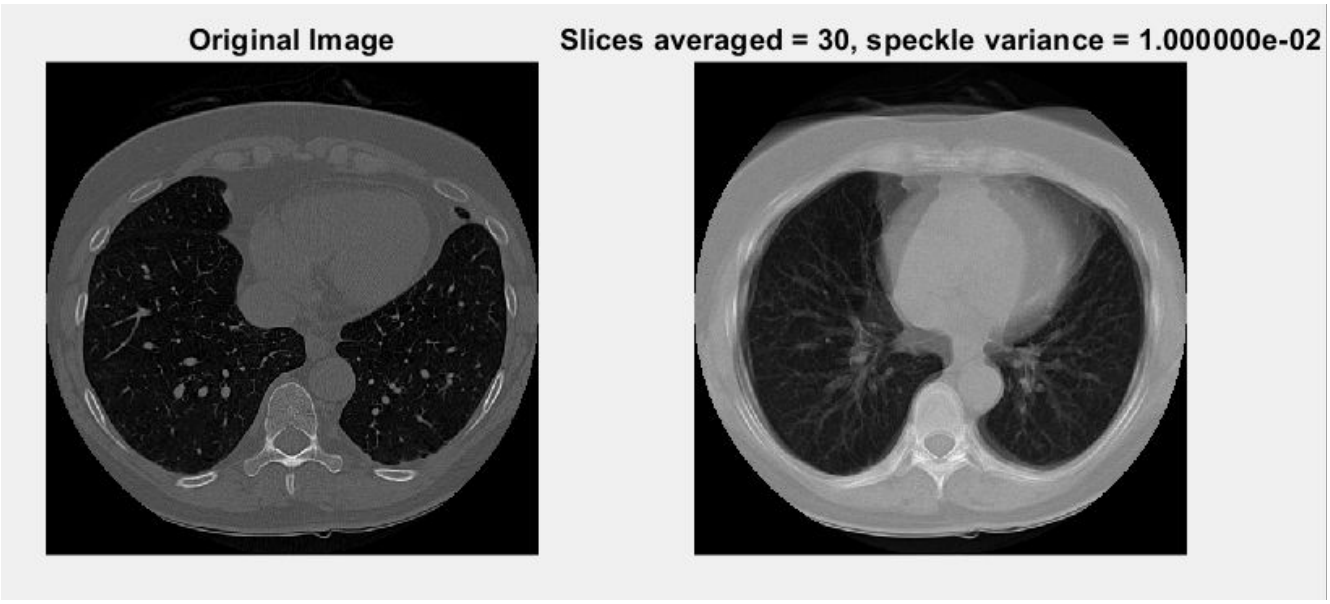

Figure 16: The first slice of the CT compared to temporal averaging on 30 slices with speckle variance of 0.01

h) The main drawback of this temporal averaging function is that multiple slices must be used. The end result is the average of the slices, which appears less definitive than a single slice. This could be rectified by using multiple samples of the same slice instead of multiple slices.