SYDE 575

Lab 2 - Report

Prepared by:

Timothy Tsang 20556306

Jonathan Wen 20515413

Professor David A. Clausi

Wednesday October 3rd, 2018

1. Introduction

This lab focuses on investigating how noise generation affects images and ways to either enhance the noisy image or restore noisy images back to their original state, if possible. First, ways to add noise via MATLAB will be investigated, through adding noise with various distributions, such as Gaussian, Salt & Pepper, and Speckle. Next, various noise reduction filters were used to see how they effective they were on various types of noise. Small filters, such as a 3x3 filter was used, and large filters, such as 7x7 filters were used. Different distributions of filters were also used, such as Gaussian and the median filter. Next, enhancement features were investigated upon noisy images, such as sharpening through subtraction or multiplication. Results, figures, and tables of the investigation are noted below.

1. Noise Generation

First, the toy image was generated, which had two grayscale bands. Noise was added to that image, with a plot of the corresponding histogram. Here, an additive zero-mean Gaussian noise, with a variance of 0.01 was added to the image:



Figure #1: Toy Image with Gaussian Noise

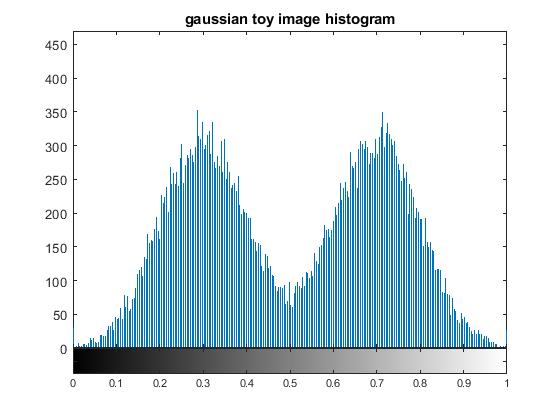


Figure #2: Toy Image Histogram with Gaussian Noise

Next, an additive Salt & Pepper noise, with a noise density of 0.05 was added to the image:

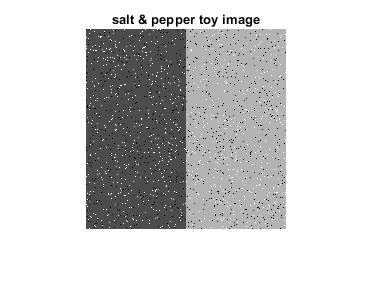


Figure #3: Toy Image with Salt & Pepper Noise

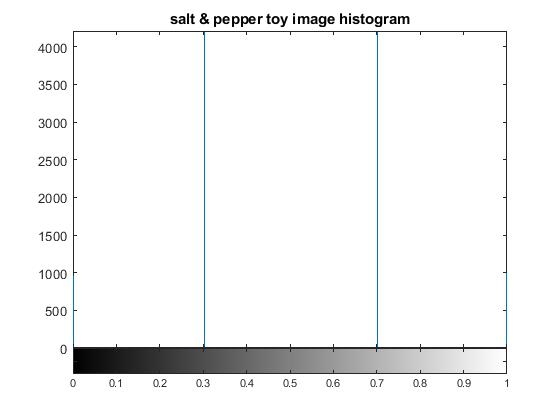


Figure #4: Toy Image Histogram with Salt & Pepper Noise

Lastly, a multiplicative Speckle noise, with a variance of 0.04 was added to the image:

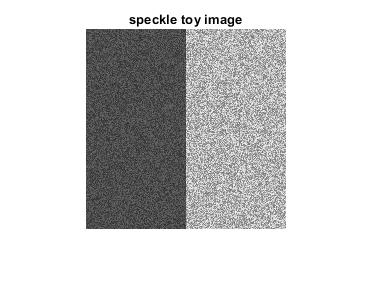


Figure #5: Toy Image with Speckle Noise

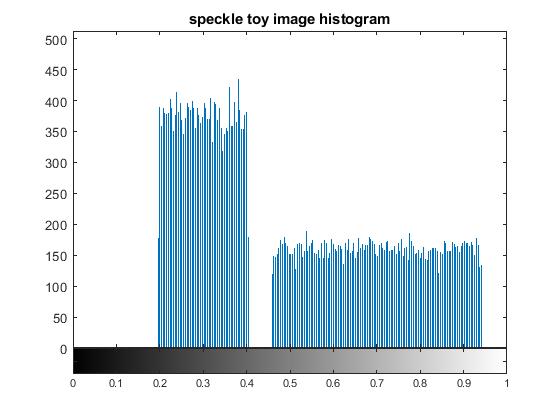


Figure #6: Toy Image Histogram with Speckle Noise

1. Gaussian creates a gaussian distribution of noise in the toy image. This means a smooth curve in the histogram. Here, we can see a distribution of grey tones throughout the image. On the other hand, salt and pepper sprinkles black and white pixels with intensities of 0 and 1 respectively. This means that there are 4 possible types of intensities: 0, 0.3, 0.7, and 1, which can be seen in the histogram. On the image, you can see only those intensities, nothing else. Speckle add multiplicative noise with the variance of 0.04 with uniform distribution. This is why we see 2 rectangles in the histogram, since the blacks and whites noise are distributed uniformly.
2. Gaussian noise makes the image look like white noise on a television. Salt and pepper adds white and black spots in both rectangles of the image. Speckle adds speckles of different intensities in the other rectangles. It seems like a mix of salt & pepper and gaussian noise.
3. The underlying distribution is a normal distribution with a set variance of 0.04. This is why the histogram looks like a box, and not a smooth curve (like in gaussian). You can see this in the image why looking at the noise. The noisy pixels are not really far from the ‘normal’ intensities, unlike in the gaussian noise.
4. hmmmm
5. Noise Reduction in the Spatial Domain

Next, various filters with different distributions were investigated, to see how well specific filters function on different types of noise. Below is the gray scaled Mandrill image, which was used to compare with images that are filtered to reduce the noise.

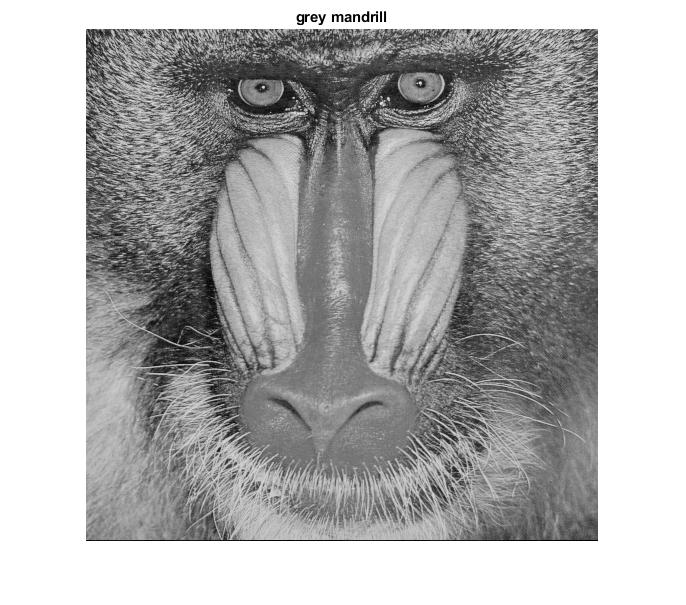


Figure #7: Grayscale Mandrill image

Next, the Mandrill image was contaminated with a zero-mean Gaussian noise, with a variance of 0.002. The image and corresponding histogram are displayed below.

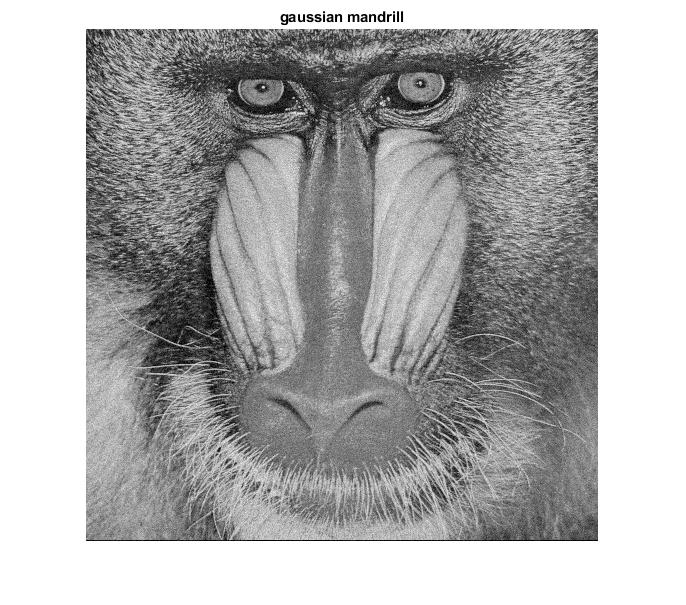


Figure #8: Mandrill with Gaussian Noise

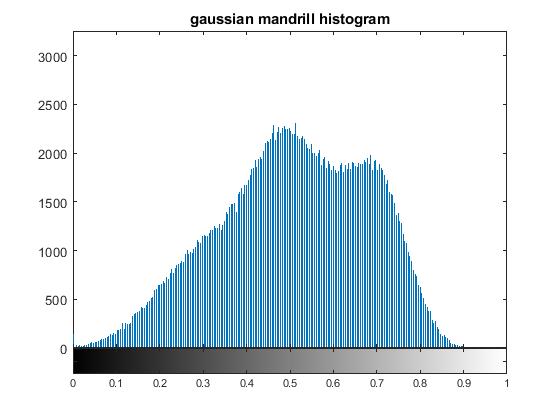


Figure #9: Mandrill Histogram with Gaussian Noise

In order to reduce the noise on the image, a 3x3 averaging kernel filter was created. The filter was then plotted and applied using the built-in MATLAB function imfilter. The PSNR was also compared with the original grayscale Mandrill image, calculated to be

23.057. Below displays the filter, the denoised image using the 3x3 averaging filter, and the corresponding histogram.

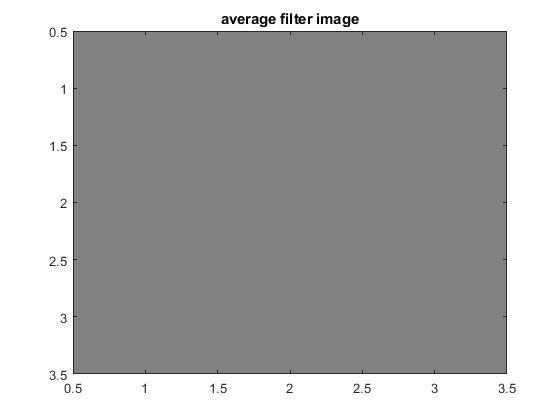


Figure #10: 3x3 Averaging Filter

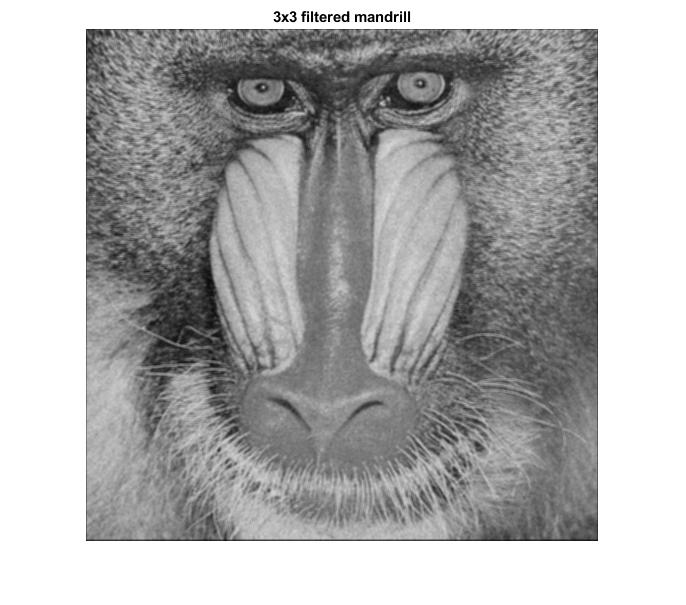


Figure #11: Denoised Mandrill Image using 3x3 Averaging Filter

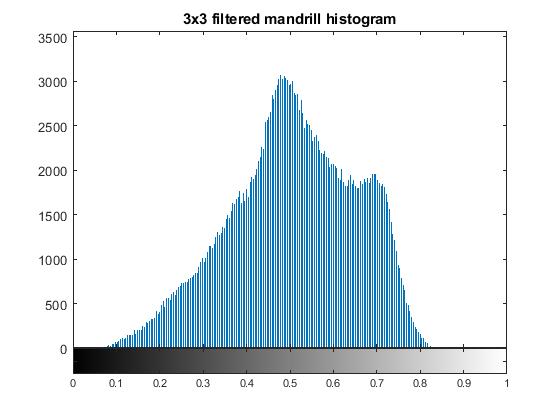


Figure #12: Denoised Mandrill Histogram using 3x3 Averaging Filter

1. Looking at the two images, the noise seemed to have decreased between the two images. This is because the 3x3 filter smoothed out the image, and the noisy pixels were averaged out. The PSNR decreased, which means that the image is further away from the original, noise-free image.
2. Not sure
3. One of the benefits of the average filter is that it visually decreased noisy images. This is very noticeable in the darker sections of the image. This is useful for photo editing applications, where images with high ISO can be smoothed out and denoised. One of the drawbacks of this method is that it makes the image slightly more blurry, and the edges less distinct.

Next, a 7x7 averaging filter kernel was created and used to remove noise from the Mandrill image. The PSNR was calculated to be 20.359. Below are plots of the denoised image using the 7x7 averaging filter and its corresponding histogram.

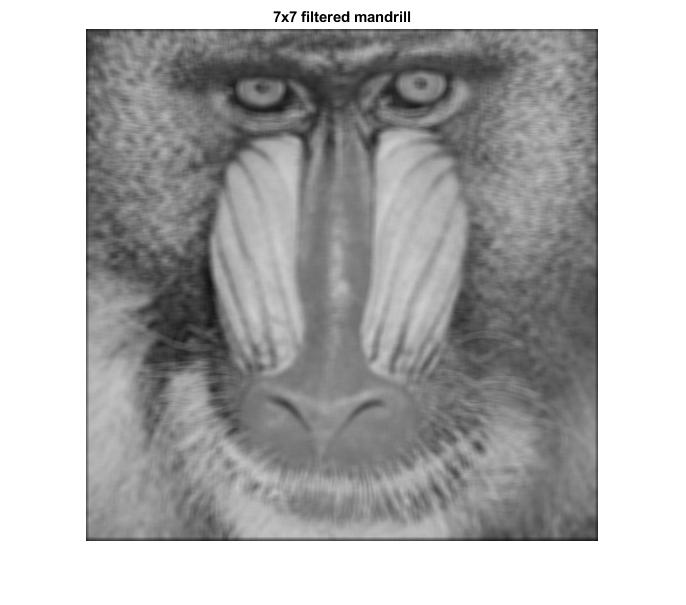


Figure #13: Denoised Mandrill Image using 7x7 Averaging filter

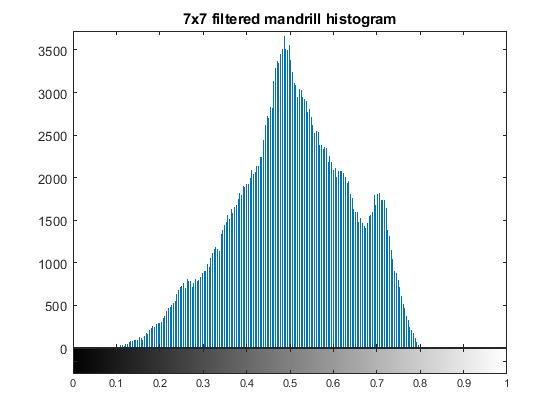


Figure #14: Denoised Mandrill Histogram using 7x7 Averaging filter

1. Visually, the image is much more blurry than the denoised image from the 3x3 filtering kernel. This is because a larger convolution kernel is used, which will result in a larger averaging of the image. Lines become even more blurred, and the overall noise of the image decreases. Because of this, the image is even further away than the 3x3 denoised image, and the PSNR decreases even more.
2. Dunnoe
3. Using a larger window size can be useful if you want to heavily remove noise, such as in a black background. However, if you want to keep some clarity and keep the image sharp, using a larger window would not be the best solution. This will heavily blur the edges in an image.

Next, a 7x7 Gaussian filter kernel with a standard deviation of 1 was created and used to remove noise from the Mandrill image. The PSNR was calculated to be 23.439. Below are plots of the 7x7 Gaussian filter, the denoised image using the 7x7 Gaussian filter and its corresponding histogram.

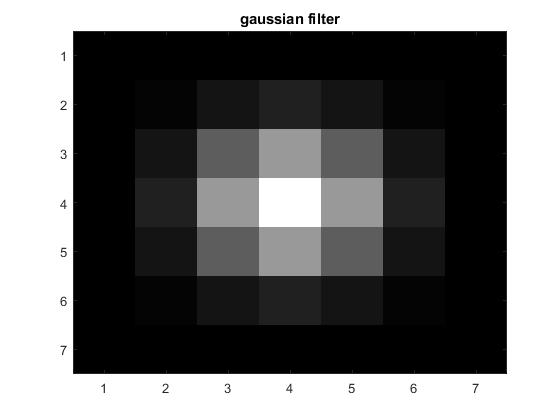


Figure #15: 7x7 Gaussian Filter

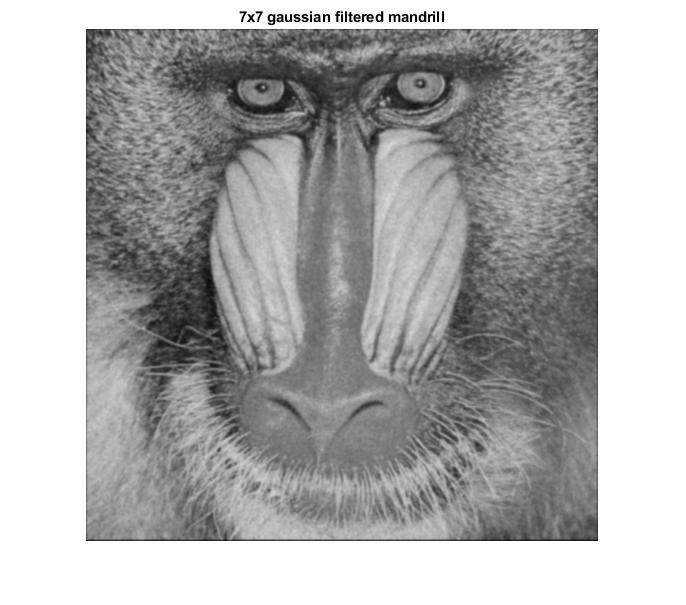


Figure #16: Denoised Mandrill Image using 7x7 Gaussian filter

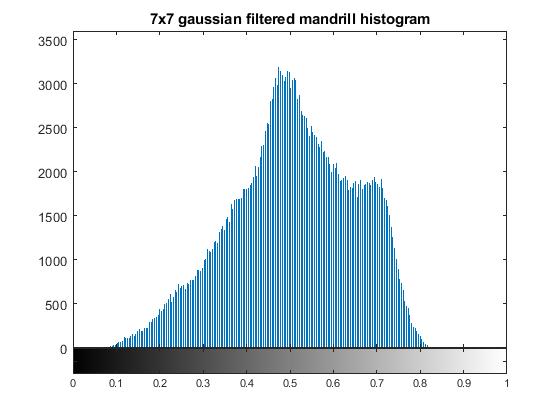


Figure #17: Denoised Mandrill Histogram using 7x7 Gaussian filter

1. There is significant difference between the gaussian filtered image compared to both the 3x3 and 7x7 average filtered image. This is because the filter is biased towards pixels that are closer to the center. Pixels on the edge are weighted less than pixels near the center. This improves the overall clarity of the image, while still removing the noise. Since the image is much more sharp than the 7x7 average filtered images, the PSNR is higher.
2. Dunnoe
3. If someone wants to filter an image but preserve the detail and clarity of the image, they should use the gaussian filtering method. This allows edges to not be as blurred as the other techniques. However, this method may not smooth out all the noisy pixels, unlike the other techniques.

Next, a new image was created using a Salt & Pepper noise distribution. This noise was applied to the Mandrill image. The 7x7 averaging filter kernel and the 7x7 Gaussian filter kernel was applied separately to this Salt & Pepper contaminated image, to see the results of the two various filters. Below illustrates the contaminated Salt & Pepper Mandrill image.

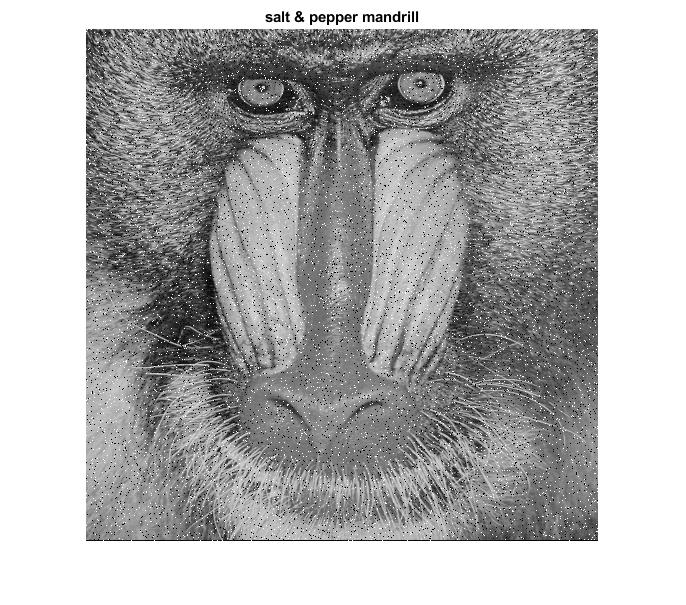


Figure #18: Salt & Pepper Mandrill image

Here, the 7x7 averaging filter kernel was used to remove noise from the Mandrill image. The PSNR was calculated to be 16.585. Below are plots of the denoised image using the 7x7 averaging filter and its corresponding histogram.

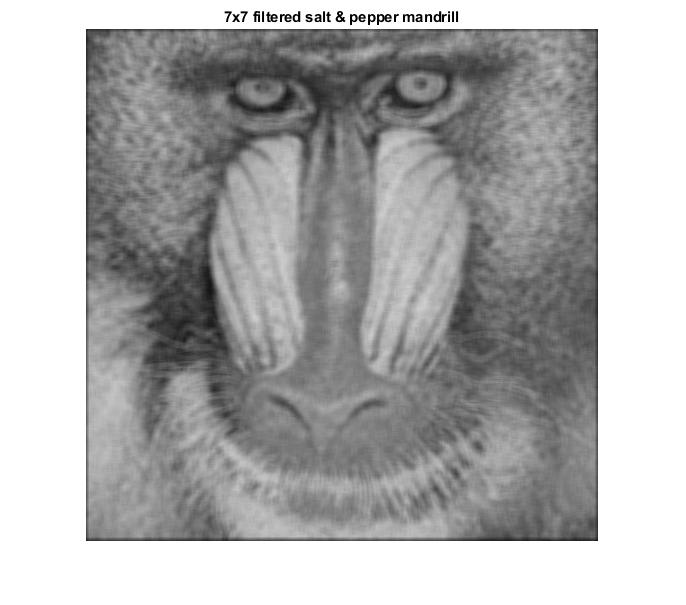


Figure #19: Denoised Mandrill image using 7x7 Averaging Filter

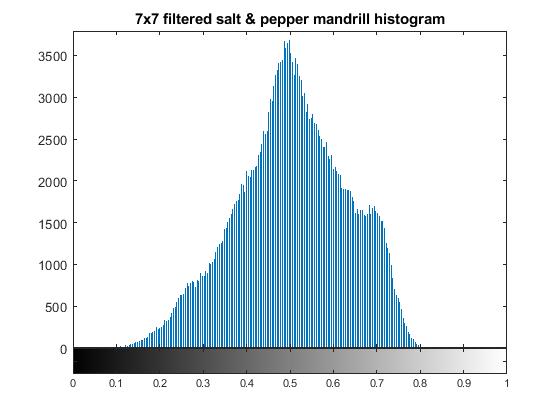


Figure #20: Denoised Mandrill Histogram using 7x7 Averaging Filter

Here, the 7x7 Gaussian filter kernel with a standard deviation of 1 was used to remove noise from the Mandrill image. The PSNR was calculated to be 18.357. Below are plots of the denoised image using the 7x7 Gaussian filter and its corresponding histogram.

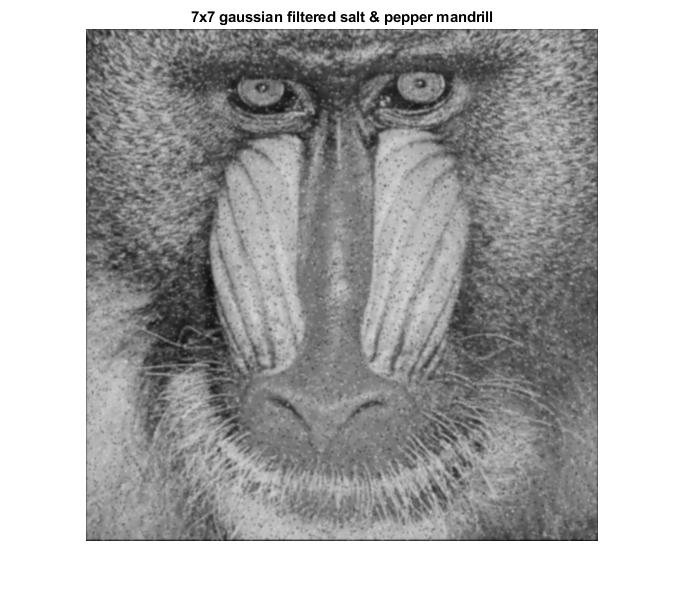


Figure #21: Denoised Mandrill Image using 7x7 Gaussian Filter

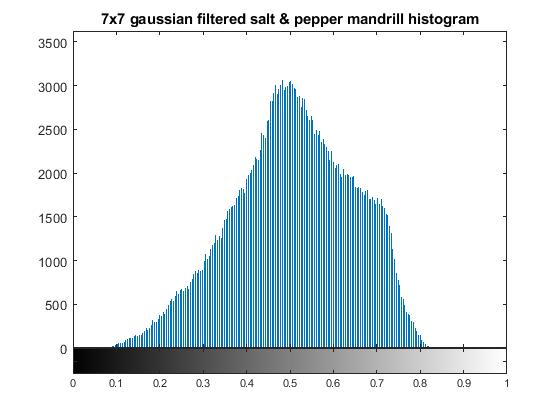


Figure #22: Denoised Mandrill Histogram using 7x7 Gaussian Filter

1. Comparing the averaging filter and the gaussian filter, the results are similar to what we saw previously. The averaging filter produced a more blurry image but with more of the noise reduced, while the gaussian filter produced a sharper image but did not reduce the noise as much. This is because with salt and pepper noise, the noise is contained to certain pixels, and not every pixel is affected by the noise (unlike gaussian noise). Therefore, when a gaussian filter is applied, the kernel will land on a salt/pepper pixel, and its intensity will not change much because the center is weighed much more heavily. Noise is therefore not reduced as well.   
     
   In terms of PSNR, both filtering techniques had much lower PSNR values. The gaussian filtered image had a high PSNR compared to the averaged image, which was expected.
2. Dunnoeeeee

Lastly, a median filter kernel was created to remove noise from the Salt & Pepper contaminated Mandrill image. The PSNR was calculated to be 17.617. Below are plots of the denoised image using the median filter and its corresponding histogram.

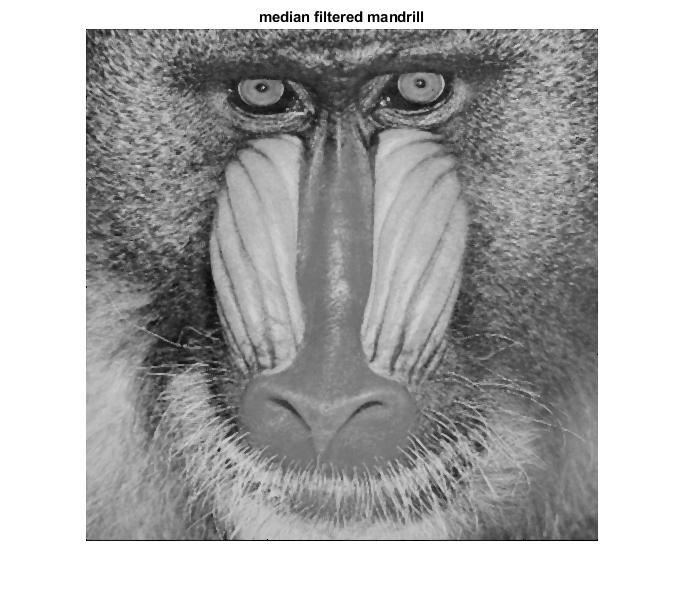


Figure #23: Denoised Mandrill Image using a Median Filter

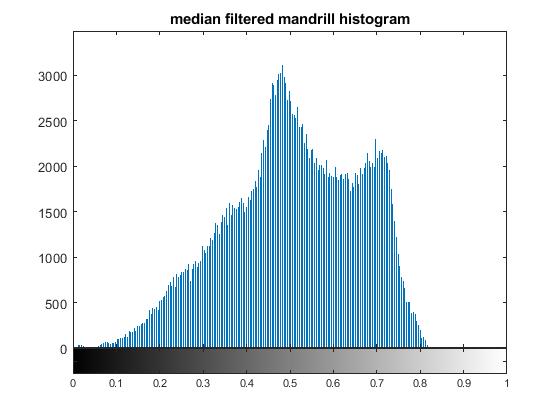


Figure #24: Denoised Mandrill Histogram using a Median Filter

1. The denoised image using the median filter performs better visually compared to the gaussian filter and the averaging filter. More of the salt and pepper pixels are being filtered out. This is because with salt and pepper, the pixel intensities will fall on the opposite spectrums. By using the median filter, the pixel intensities will never affect the final result, because they will not be accounted for. In terms of PSNR, it gives a higher value than the averaging filter method, but lower than the gaussian filter method.

Below is a summary table of all of the calculated PSNR for the various filters used in the lab. Note that the top 4 rows used the Gaussian-distributed noise whereas the last 3 rows used the Salt & Pepper contaminated image.

Table #1 - PSNR Comparison

|  |  |
| --- | --- |
| Gaussian Noisy Image | 26.995385039359753 |
| 3x3 Denoised Image | 23.0571036198121 |
| 7x7 Denoised Image | 20.359602489641834 |
| 7x7 Gaussian Denoised Image | 23.439170128946520 |
| 7x7 Salt & Pepper Denoised Image | 16.585291337386000 |
| 7x7 Salt & Pepper Gaussian Denoised Image | 18.357674015710050 |
| Median Filter Denoised Image | 17.617066153500890 |

1. Sharpening in the Spatial Domain

Next, an investigation using sharpening in the spatial domain was done. With the 7x7 Gaussian filter that was used in Part 3, that filter was applied to the Cameraman image. Next, the Gaussian image was subtracted with the filtered image, plotting the difference. Below shows the 7x7 Gaussian filtered image and the difference image.



Figure #25: Cameraman Image using a 7x7 Gaussian Filter

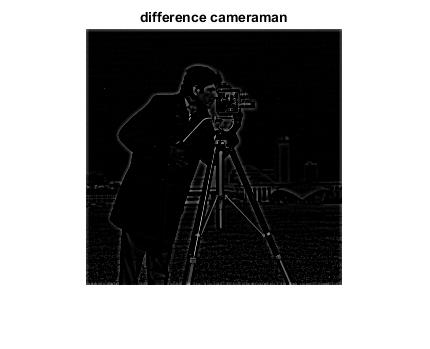


Figure #26: Difference between Filtered and Original Cameraman Image

1. The subtracted image is a black image with the edges of the original image being outlined. This works because of the properties of a gaussian smoothing filter. When an image is being smoothed, rough edges are being smoothed by sampling the neighboring pixels. This means that regions in the image where there is no contrast (change in pixel intensity) will not change much, but regions with high contrast (edges) will change. When subtracting the original image from the filtered image, the smooth regions will be 0, while the edges where there has been a lot of change will stay.

Next, instead of subtracting, the addition of the subtracted image and the original image was done. Below shows the sharpened addition image.

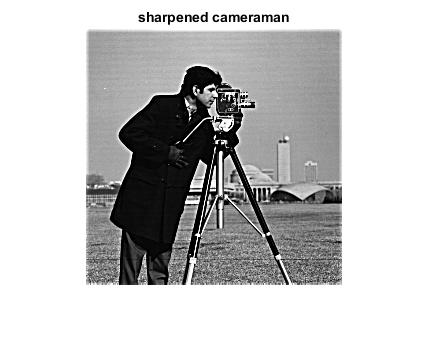


Figure #27: Addition of Subtracted Image and Original Cameraman Image

1. The resulting image is much more sharp than the original. When adding the subtracted image to the original, the edge values will be increased. Regions where the subtracted image is 0 (everywhere that is not an edge) will stay the same. Hence, the edges of the original image will be enhanced, making the image look much more sharp.

Lastly, the subtracted image was multiplied by 0.5 (or divided by 2) and then was added to the original image. Below shows the less sharpened addition image.

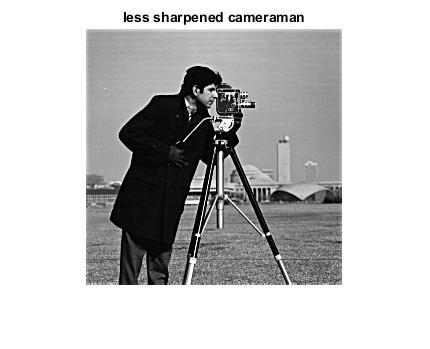


Figure #28: Addition of Half Subtracted Image and Original Cameraman Image

1. When multiplying the subtracted image by 0.5 and adding it to the original image, the result is a sharpened image, but much less sharp than the one above. This is because we are dividing each value by 2, reducing the intensities of the edges in the subtracted image. Therefore, the edge values will be increased by less than before, resulting in a less sharpened image.
2. Multiplying by a factor of less than 1 will reduced the sharpening, while multiplying by a factor of greater than 1 will increase the sharpening. Therefore one can control the amount of sharpening by changing the multiplying factor to greater or less.
3. Conclusion

Conc