SYDE 575
Lab 1 - Report

Prepared by:
Timothy Tsang 20556306
Jonathan Wen 20515413

Professor David A. Clausi

Wednesday September 19th, 2018

1. Introduction

This lab provided examples of quantitatively measuring the quality of an image through calculating their respective Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR). Consequences of digital zooming on images were investigated through reducing the resolution of certain images and then attempting to reconstruct and restore the images back to their original state using various interpolation methods. Next, convolution was used in the context of image processing to determine the role of an impulse function when convolved with an image and how it produces such results. Lastly, various point operations were introduced to specific images in order to attempt to enhance the contrast of the images. Histograms were created and analyzed to determine the spread of point intensity throughout the image. All in all, the lab introduced many important concepts that are used in many applications of image processing.

2. Image Quality Measures

Section 2 outlines the creation of the function PSNR, which calculates the Mean Squared Error and the Peak Signal to Noise Ratio of two images as the input parameters respectively. First, it takes the size of the rows and columns of the test image, labelled as g, which has been modified somehow. These values will be used via comparison with the original image, labelled as f. Next, it calculates the MSE of the two images. It takes the squared difference of the two images, sums up the entire matrix, and then divides itself by the product of the rows and columns of the test column. One important thing to note is the result of the squared difference:

```
squared_difference = (double(f)-double(g)).^2;
```

It is important to note that the the images are first converted to double precision. Originally, the image is inputted as 8-bit unsigned integers; however, by converting to double precision, the intensity of the image would be rescaled and many operations that are done to the image would be much more easier since they are in double precision floating point. One clear example is when the the difference is found between the first and second image. If that operation was done using 8-bit unsigned integers, if the second image value was greater than the first, the result would be 0 since unsigned integers cannot be represented by negative numbers.

The MSE is found by summing all of the rows and columns of the squared differences, divided the by count of the rows and columns. Next, the PSNR is calculated. One important thing to note is that the R value (or max f value) in the PSNR equation can be represented as 255, since that is the largest value of the 8-bit unsigned integer data type.

3. Digital Zooming

In section 3, the cameraman and lena images were used. The lena image was first converted to grayscale using the rgb2gray function. The cameraman did not need to be converted since the original format was already grayscale. The two images were then resized using the imresize function. Using the *bilateral* method, the two images were reduced by a factor of four (scale was 0.25) using the bilateral interpolation method. Bilateral interpolation involves determining the pixel intensity through the use of 4 nearest neighbours, performing linear interpolation in the x and y direction. The reduced images are shown below:

Figure #1: Reduced Images using Bilateral Interpolation

Next, an attempt to restore the images back to their original form via digital zooming was done, using three methods: nearest neighbours, bilateral interpolation, and bicubic interpolation. The digital zoomed images for each method are shown below:

Figure #2: Nearest Neighbours Interpolation of Cameraman

Figure #3: Nearest Neighbours Interpolation of Lena

Figure #4: Bilateral Interpolation of Cameraman

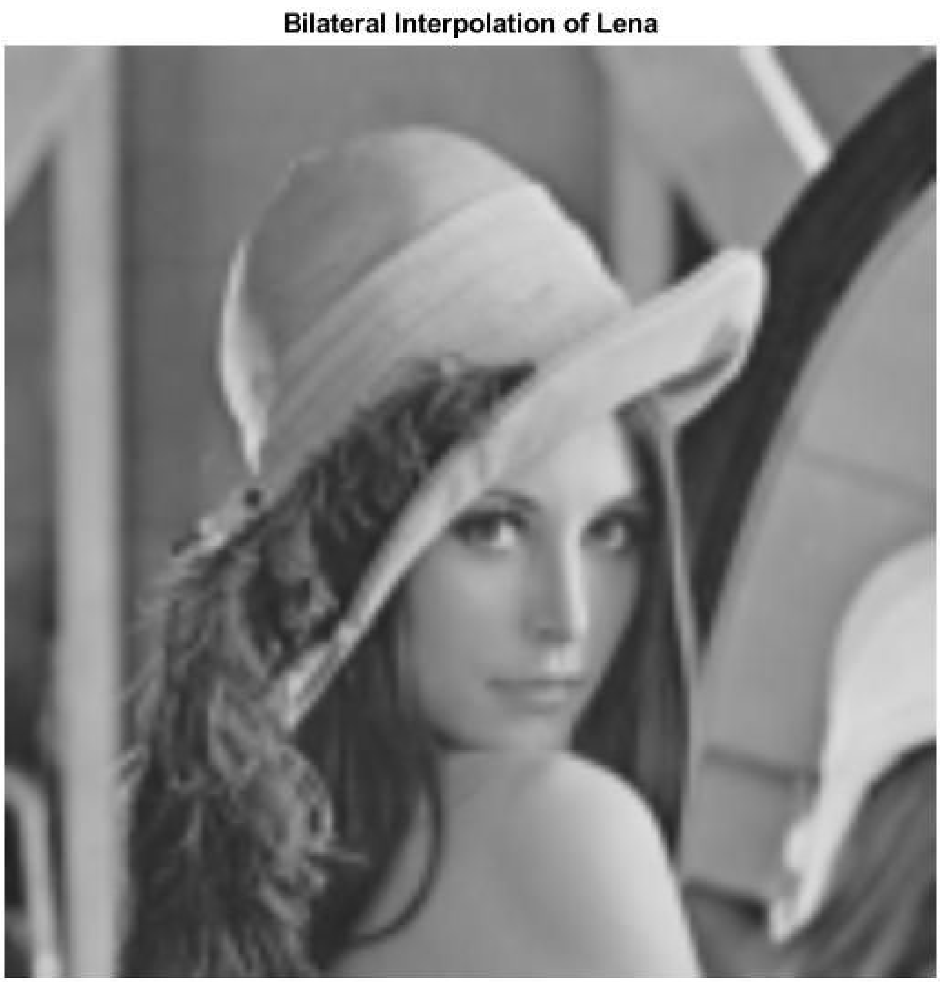

Figure #5: Bilateral Interpolation of Lena

Figure #6: Bicubic Interpolation of Cameraman

Figure #7: Bicubic Interpolation of Lena

In general, from the figures above, it is clear that the bicubic interpolation produces the best resolution upon digital zooming. In addition to the figures, the PSNR values were calculated for each up-sampled image, through comparing the modified image with their corresponding gray-scaled original image:

Table #1 - PSNR Values for Up-Sampled Images

Image:	PSNR Value:
NN_Cameraman	21.5412
NN_Lena	26.6709
Bilateral_Cameraman	21.8190
Bilateral_Lena	27.2977
Bicubic_Cameraman	22.2680
Bicubic_Lena	28.0850

1. For each up-sampled image, we can see an improvement in the image resolution as each method progresses. For the Nearest Neighbours method, the up-sampled image has a poor resolution if compared to the original images. This is due to the method not having enough information on the original pixel intensity, which would create a blocking effect. It is clearly the method that produced the images with the worst resolution due to only looking for the closest pixel, which was approximated in the mapped data.
   For the Bilateral Interpolation method, the pixel intensity value is approximated using 4 nearest neighbours, which greatly improves the resolution of the up-sampled images. In doing so, the image is a lot more smoother rather than having blocks in the up-sampled image.
   The last method, which is the Bicubic Interpolation method, provides the most optimal method where approximates the pixel intensity value by using 16 nearest neighbours. This method produces the best resolution for each up-sampled images.

2. The larger the Peak Signal to Noise Ratio (PSNR) value, the better the image resolution is. This is due to the fact that the PSNR value is calculated using the Mean Squared Error (MSE). Since MSE calculates how close the pixel intensity of two images are from each other, a larger MSE means there are more differences between the original and test image. Since PSNR is calculated through finding the logarithm of the quotient of the max value of the data type over the MSE, a larger PSNR would represent an image that has

higher resolution and an image that is closer to the original image.
In doing so, it would make sense that the Bicubic method would provide the highest PSNR since it outputs the image with the best resolution similar to the original. Next, the Bilateral method would provide the next highest PSNR, with the Nearest Neighbour method providing the smallest PSNR.

3. In both images, it seemed that the parts of the pictures where there were distinct edges worked well with the digital zooming methods. Areas where there is a stark contrast in the picture means that there will also be a contrast in the down-sampled picture. Therefore, when up-sampling the image, those areas will appear more crisp and clear, rather than blurry.

4. In terms of the variations of the two images, the Lena image has a larger PSNR compared to the Cameraman image. Clearly, the Lena image has a higher resolution compared to the Cameraman image, for any digital zooming method used. The original Lena image was less noisy and had more distribution of grey pixels, whereas the original Cameraman image was much noisier and lower quality to begin with.

5. The higher the PSNR value is, the better resolution and quality the reconstructed image has. Therefore, looking at Table #1, it is clear that the most optimal method that provides the best resolution of images is the Bicubic Interpolation method. Next is the Bilateral Interpolation and then the Nearest Neighbours method respectively. This is clearly reflected in the images, with the reconstructed, up-sampled images via the Bicubic Interpolation method having a much higher resolution compared to the other images.

4. Discrete Convolution for Image Processing

Next, the grayscale, normalized Lena image were subjected to convolution with 3 sets of impulse functions of different intervals. The images and three impulse functions are listed below:

- h1 = (1/6)*ones(1, 6);
- h2 = h1’;
- h3 = [-1 1]

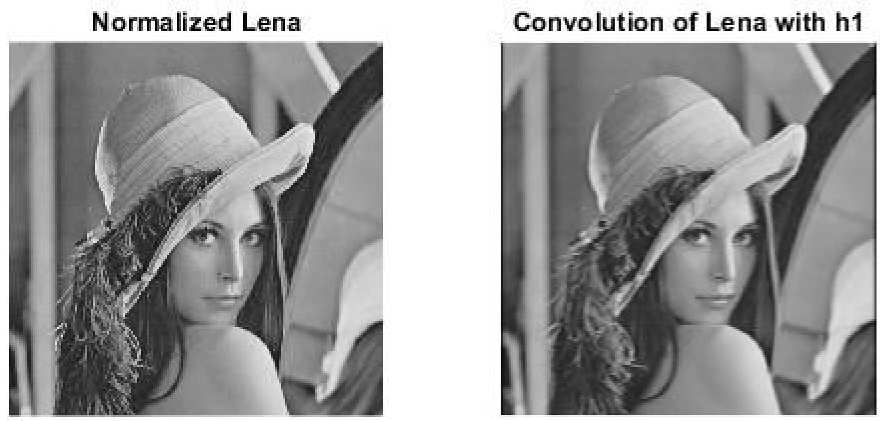

Figure #8 - Convolved with impulse function h1

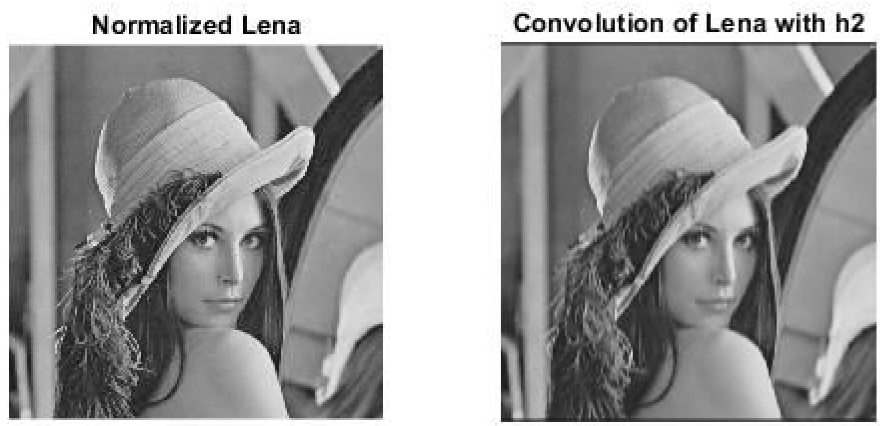

Figure #9 - Convolved with impulse function h2

Figure #10 - Convolved with impulse function h3

1. Convolving the Lena image with the h1 impulse function provides a smoother image. The h1 impulse function is a smoothing or averaging filter, which smoothes with a value of 6 along the horizontal axis. Therefore, any lines in the horizontal axis will be smoothed/blurred. .

2. Convolving the Lena image with the h2 impulse function that is the transposed version of the h1 impulse function creates an image that is smoothed along the vertical axis. This is because the h2 impulse function is a column array of ⅙. Lines along the vertical axis will be smoothed/blurred.

3. Convolving the Lena image with the h3 impulse function creates a black-white image, with most of the image having a 0 intensity and some edges of the image being white. The filter [-1 1] attempts to to perform edge detection. This is because an edge is defined by a change of pixel intensity. If there is no edge, then the intensity should stay the same between two consecutive pixels, and the convolution should cancel out to 0 (-1 cancels with 1). However, if there is an edge, the pixel intensities will vary, and the convolution result should not cancel out to 0.

4. Convolution is used in image processing, notably for enhancing and sharpening images, removing noise, and enhancing the resolution through removing blurring. It is also very useful for edge detections. As shown from the images above, convolving with an impulse function that acts as a smoothing or averaging filter greatly enhances the image to provide higher quality, removing a lot of noise and edges. On the other hand, convolution can also be useful for performing edge detections upon various images.

5. Point Operations for Image Enhancement

The last section involves subjecting the Tire image with various point operations, such as negatively transforming the image, transforming the image using power-laws of various exponents, and equalizing the original histogram of the image. A histogram was generated for each point operation and analyzed. First, the tire image and its corresponding histogram was plotted:

Figure #11 - Tire Image

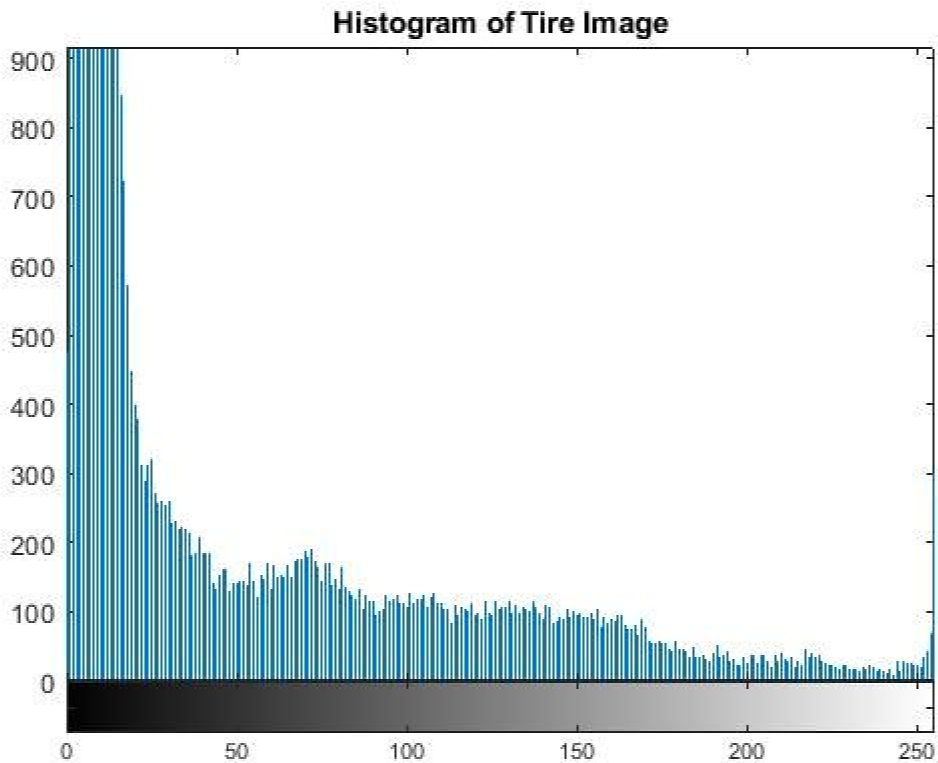

Figure #12 - Histogram of Tire Image

1. The histogram of the Tire image represents the pixel count (y-axis) within the image for a certain intensity (x-axis). From the histogram, the image clearly has a lot of pixels with an intensity less than 50, which means that the image contains a lot of dark grey or black colors. On the other hand, there are very little pixel count that has an intensity greater than 200, which means there are not as many light (close to white) pixels in the image. This is useful as someone can easily and clearly determine the tonal distribution of the entire image. Also, one can analyze the trends of the histogram, whether it has peaks or

valleys within the histogram. This is important and useful when performing processes of edge detection or segmenting an image.

2. As stated above, the intensity distribution is conveyed by displaying the pixel count for each intensity, ranging from 0-255. The histogram can quickly show a reader the primary intensity of the image. In this case, it is clear that there are a lot of pixels with an intensity of 0-50, while not having many pixels with an intensity greater than 200.

Negative transformation was applied to the tire image by subtracting 255 to the actual image. The transformed image and corresponding histogram are plotted below:

Figure #13 - Negative Transformed Image of Tire

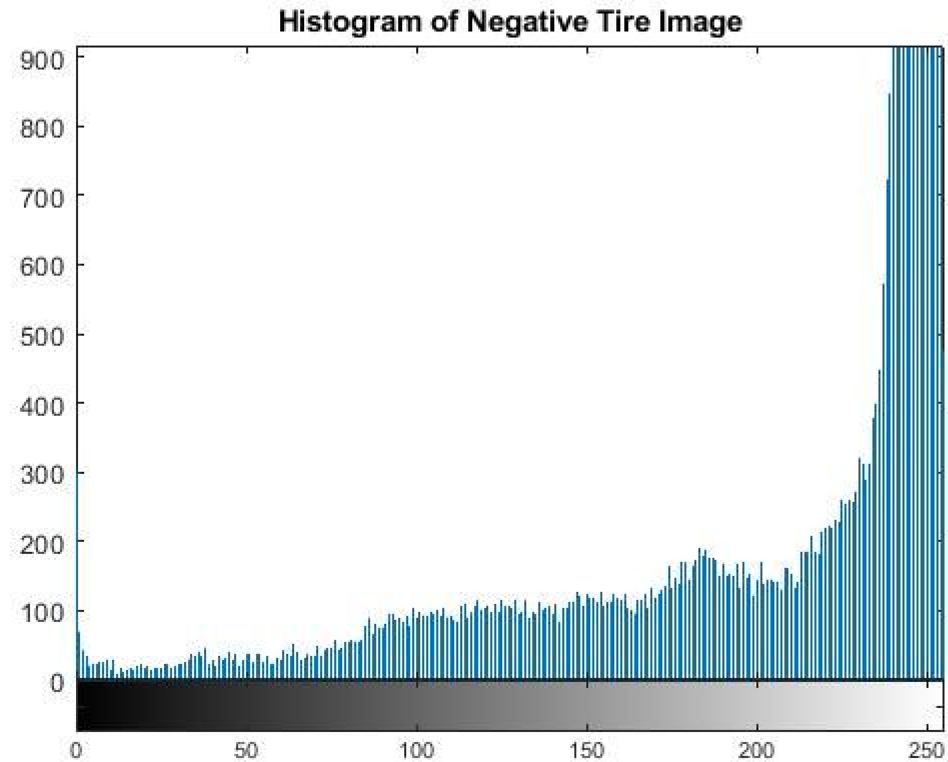

Figure #14 - Histogram of Negative Transformed Tire

1. The negative of the image is a reversal of the pixel intensity, with the lightest areas becoming the darkest areas, and vise versa. In this case, it is clear that the histogram has been reflected on the y-axis, such that most of the pixels have an intensity of 200-255, while having a lot less pixels that have an intensity from 0-50. This is the reversal of the original image and histogram shown above.

   Next, two power-law transformations was done on the image, with two exponent values of 0.5 and 1.3 respectively. Their images and histograms are plotted below:

Figure #15 - Power-law Transformed with Exponent Term of 0.5

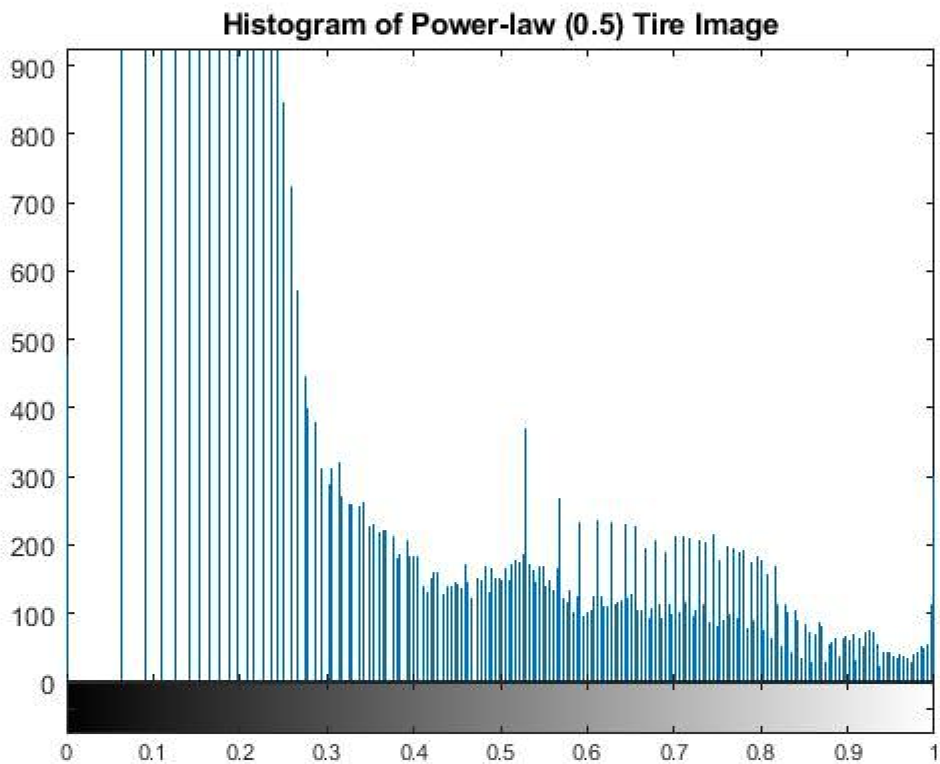

Figure #16 - Power-law Transformed Histogram with Exponent Term of 0.5

Figure #17 - Power-law Transformed with Exponent Term of 1.3

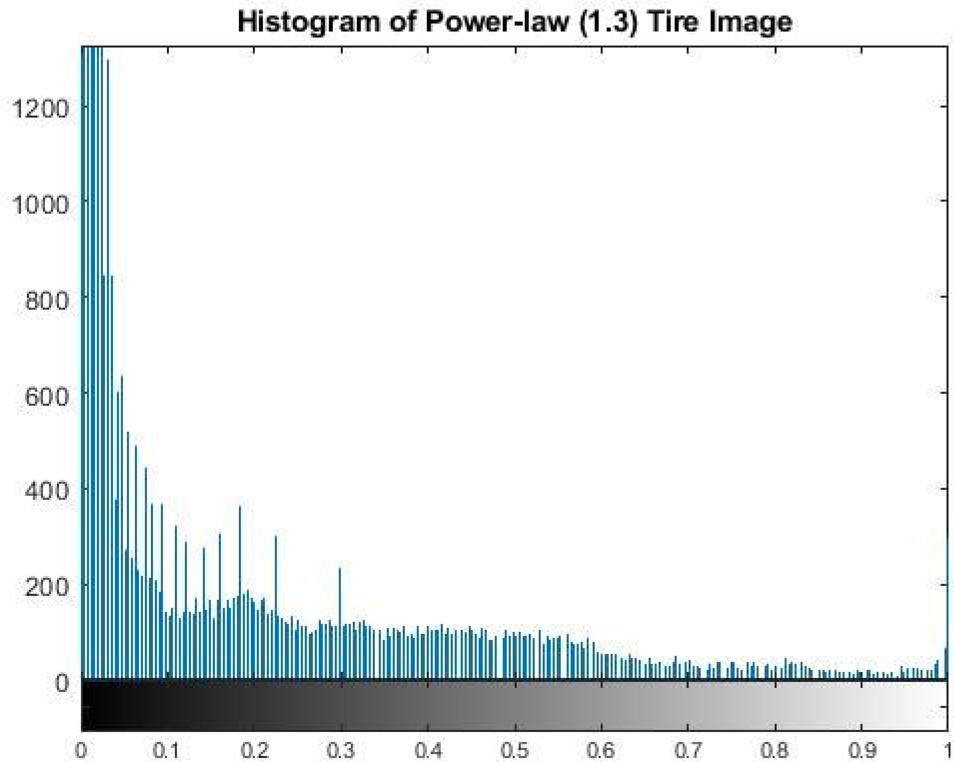

Figure #18 - Power-law Transformed Histogram with Exponent Term of 1.3

1. The transformed image with an exponent term of 0.5, the image appears to be whiter, with less number of pixels with a small intensity than the original image. On the other hand, the transformed image with an exponent term of 1.3 transforms the image to be more darker compared to the original image. Here, more pixels have a darker intensity. This is because for exponent terms that are less than 1, the image is enhanced to

distribute the amount of darker pixels while compressing the amount of brighter pixels, which would make the image be brighter. This is extremely useful for images that have darker regions; providing power-law transformations would allow readers to see the details of darker regions much better. On the other hand, for exponent terms greater than 1, the opposite happens in which the darker pixel regions would be compressed while the brighter pixel regions would be stretched and distributed to other pixel intensity values. This is extremely useful for images that have bright, overexposed, saturation spots, which would increase the contrast to allow readers to see details of brighter regions more better.

2. The histogram for the exponent term of 0.5 would still have a left-skewed histogram, in which there are high values of pixel counts in dark intensities. However, the distribution of darker pixels would be more stretched, meaning there would be more pixels within the dark-grey pixel intensity area.
   On the other hand, the histogram for the exponent term of 1.3 would have more darker pixels within the histogram and less amount of brighter pixels, since the bright regions of the image have been stretched whereas the darker regions of the image has been compressed.

3. The original image is quite dark and the details within the dark pixel regions are hard to see. Therefore, one would use an exponential term that is less than 1, since we want to see the details of the darker regions. Therefore, we would want to stretch the darker pixel regions and compress the brighter pixels, which would improve the image by providing more brighter pixels within the dark regions of the image. This would allow us to see the details of the dark regions of the image.

Lastly, histogram equalization was done on the tire image. The transformed image and corresponding histogram are plotted below:

Equalized Image of Tire Image

Figure #19 - Equalized Image of Tire

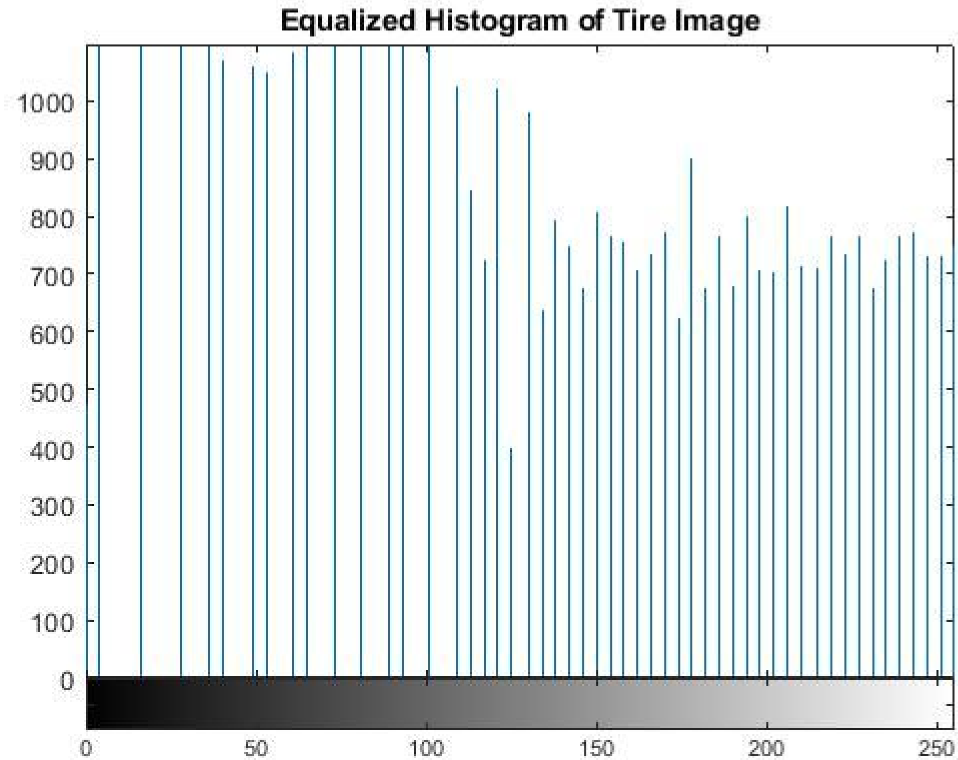

Figure #20 - Equalized Image Histogram of Tire

1. The equalized image contains a lot more white and brighter grey tones/colors compared to the original image. The image was stretched to increase the contrast of the original image, which is reflected in the new image.

2. The equalized histogram has distributed the pixel intensity to spread throughout the 255 levels. Originally, there was a skew of dark pixels in which the original histogram had a lot more pixels from values 0-50. Once equalized, the distribution of pixel intensity was spread across the entire range of values. For the intensities with high number of pixels, it was stretched while intensities of lower pixel numbers were compressed. This resulted in an image that had a much more flat intensity profile. In short, the contrast of the image was transformed due to modifying the distribution of intensities of the pixels within the original image.

6. Conclusion

Through this lab, we were able to discover ways to measure, validate, and improve the overall quality of an image using Matlab. Using PSNR and MSE to measure the amount of error between two images and the quality of the image via its resolution was done, which provided a quantifiable means to measure how close one image is to another image. Those calculations were used to determine the performance of various digital zooming methods, including Nearest Neighbours, Bilateral Interpolation, and Bicubic Interpolation. Next, an investigation on how convolution impacts the image quality was done. It was shown that through discrete convolution calculations, images can be filtered to either improve or decrease the quality of an image. Lastly, an attempt for image enhancements were done through the use of point operations, notably transforming to negative images, using power-law transformations of various exponent terms, and performing histogram equalization.