University of Waterloo

Faculty of Engineering

SYDE575 Lab 1: Fundamentals of Image Processing

Prepared By
Sarthak Tamboli | 20665466
Sheen Thusoo | 20728766

Date: 21 September, 2021
Prepared for: Professor David A. Clausi

1. Introduction

The purpose of this lab is to study fundamental image enhancement techniques such as digital zooming, spatial filtering, and point operations for image enhancement. It also looks at quantifiably measuring the quality of an image using Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR). These quantities help measure the quality of a reconstructed image. Further, digital zooming was performed in order to increase resolution of a downsampled image back to its original resolution using Nearest Neighbours, Bilinear, and Bicubic Interpolations. Lastly, point operations were applied, such as the power-law transformations, on the images to improve image contrast. For this lab, the 'lena.tiff', 'cameraman.tif', and 'tire.tif' image files were used. All the code written for this lab was done using MATLAB and is included in Appendix A.

2. Image Quality Measures

Image enhancement can be very subjective, thus it is difficult to evaluate whether one method produces a better image than another. However, it is important to quantify image enhancement techniques on images by using empirical measures for image quality. This allows comparison between difference techniques and helps to evaluate which algorithm produces better results. One of these empirical measures is Peak Signal-to-Noise Ratio or PSNR. This is the ratio of the maximum possible value of a signal and the value of the distorting noise.

In Section 2, a function to compute PSNR was developed. The function takes in two inputs, the reference image f and the test image g, and outputs a PSNR value, *PSNR_out*. The code for this method can be seen in Appendix A Figure A.1. This method uses a MAX_f value of 255 since the images use 8-bit data making 255 the largest possible value. It also calculates the dimensions (number of rows and columns) of the reference image which are used to compute the Mean Squared Error (MSE). In order to compute the MSE, the squared difference between the both of the images, f and g, is also computed. Originally these images were read as integers, but they were converted to the double data type to rescale the intensity of the image and perform the required functions on MATLAB. The rows and columns of the squared difference matrix are summed using the *sum* function in MATLAB. Once the MSE is calculated, it can be used to compute the PSNR. Here, 10 is multiplied by the log of the maximum intensity squared divided by the MSE. The equations for MSE and PSNR are included below as shown in Equation (1) and (2).

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1} ||f(i,j) - g(i,j)||^2 \tag{1}$$

$$PSNR = 10log_{10}\left(\frac{MAX_f^2}{MSE}\right) \tag{2}$$

3. Digital Zooming

Section 3 focused on studying the effects of different digital zooming techniques on image quality of images that were upsampled. Here, the Lena and Cameraman images were used. First, the Lena image was converted to grayscale using the *rgb2gray* function (the Cameraman image was already in grayscale). Then, the resolution of both images was reduced by a factor of 4 using Bilinear Interpolation. These reduced images are shown in Figure 1. The code for this section is separated across three figures and can be seen in Appendix A, Figures A.2.1, A.2.2, and A.2.3.

Figure 1: Lena and Cameraman Images Reduced using Bilinear Interpolation

Then, digital zooming was performed in order to upsample the images to their original resolution; the techniques used to upsample the images were Nearest Neighbour, Bilinear, and Bicubic Interpolations. The digital zooming results for each image and each interpolation method are shown in Figures 2, 3, 4, 5, 6, and 7.

Figure 2: Digital Zooming of Lena Image with Nearest Neighbour Interpolation

Figure 3: Digital Zooming of Cameraman Image with Nearest Neighbour Interpolation

Figure 4: Digital Zooming of Lena Image with Bilinear Interpolation

Figure 5: Digital Zooming of Cameraman Image with Bilinear Interpolation

Figure 6: Digital Zooming of Lena Image with Bicubic Interpolation

Figure 7: Digital Zooming of Cameraman Image with Bicubic Interpolation

The PSNR values between the original images and the upsampled images were also computed and can be seen in Table 1.

Table 1: PSNR Values Computed Between Original images and Upsampled Images.

Interpolation Method	**Image**	**PSNR Value**
Nearest Neighbour	Lena	26.6708
	Cameraman	21.5412
Bilinear	Lena	27.2977
	Cameraman	21.8190
Bicubic	Lena	28.0850
	Cameraman	22.2680

1. For the Nearest Neighbour method, the upsampled images (Figures 2, 3) had poor resolution especially in regions where there is a distinct boundary such as the Cameraman's coat. The reason for this poor upsampling is because this technique is vulnerable to the blocking effect. This effect is the cause of the coat boundary having a 'staircase effect'. This method only looks at the closest pixel's intensity value.

The Bilinear Interpolation method did better than the previous method mentioned (Nearest Neighbour) since the upsampled images (Figures 4, 5) had better resolutions. These resolutions are improved as the images have a much more smooth look. They do not contain any staircase patterns, but the images have a blurred appearance. This method looks at the pixel intensities of the four closest neighbours.

Finally, the Bicubic Interpolation method seems to perform the best out of the three as the resolution of the upsampled images (Figures 6, 7) are much higher. The images are both very smooth. This technique does not suffer from the staircase effect and also preserves fine details better than Bilinear Interpolation. This method approximates pixel intensity using the closest 16 neighbours. Thus, this method produces the best resolution for both the upsampled images.

2. The PSNR values for each image and each zooming technique can be seen in Table 1. The PSNR value for the upsampled image reflects the quality of the reconstructed image; the higher the PSNR value, the better the quality of the reconstructed image. The PSNR value uses MSE, Equation (1), which calculates the difference between each pixel intensity in two images. A larger MSE value reflects that the two images, in this case the original and the reconstructed, are very different from one another. On the other hand, a smaller MSE value reflects two images that are closer and more similar in terms of pixel intensity. In the PSNR equation, Equation (2), the maximum possible intensity is divided by the MSE and a log function is applied to this fraction. Thus, a higher/lower PSNR value represents a higher/lower image resolution and a reconstructed image that is more similar/different than the original.

When looking at the PSNR for the images upsampled using Nearest Neighbour technique, Lena had a value of 26.6708 and Cameraman had a value of 21.5421. These are the lowest values of the three methods for these 2 images. This is in accordance with the visual quality of the images, as their resolutions are poor (the worst of the three methods) and produce the unwanted staircase effect.

The PSNR values of the images upsampled using Bilinear Interpolation are higher; Lena had a value of 27.2977 and Cameraman had a value of 21.8190. These values are slightly higher than the ones produced by the Nearest Neighbour technique. This is also in accordance with the visual quality of the images as they do not produce the unwanted staircase effect, but they do blur the appearance of the image.

The PSNR values of the images upsampled using Bicubic Interpolation were the highest with Lena having a value of 28.0850 and Cameraman having a value of 22.2680. These PSNR values are the highest of the three methods and are also in accordance with the visual quality of the upsampled images. These images do not produce the staircase effect like Nearest Neighbour and also do not blur the image like Bilinear Interpolation. These images preserve fine details.

3. For Bilinear Interpolation and Bicubic Interpolation, the boundaries or edges of the images (such as the hat in Lena or the coat in Cameraman) seem to work well. Since the original images also contained distinct boundaries, when they were downsampled, this contrast remained and when they were upsampled, the areas were easy to distinguish. However, the boundaries did not work well using the Nearest Neighbour technique, as they suffered from the 'staircase effect' due to the blocking effect. This effect occurs because this method approximates pixel intensity using only the nearest neighbour which can lead to the loss of the original pixel intensity. As well, the fine details of the image worked well using the Bicubic Interpolation method as it preserved these details (the building in the background of the Cameraman image). However, the Bilinear Interpolation method blurred these images so the fine details did not work well.

4. When comparing the zooming results between Lena and Cameraman, Lena outperforms, both visually and by PSNR value, in each zooming technique. Lena has PSNR values of 26.6708, 27.2977, and 28.0850 for Nearest Neighbour, Bilinear and Bicubic Interpolation, respectively. Comparatively, Cameraman has lower values of 21.5421, 21.8190, and 22.2680 for Nearest Neighbour, Bilinear and Bicubic Interpolation, respectively. Visually, for every digital zooming technique, Lena looks much better when restored to the original resolution. This is because the original Lena image had a much more even distribution of pixel intensities and a higher image quality in general. The original Cameraman image had a lower resolution to begin with; the 'staircase effect' is slightly noticeable even before any image downsampling/upsampling.

5. The PSNR value reflects the quality of a reconstructed image. A lower PSNR value reflects a higher quality with less error introduced to the reconstructed image whereas a higher PSNR value reflects a lower quality image with more error introduced. From Table 1, it can be observed

that the Bicubic Interpolation technique, according to the PSNR values, has the best image quality results for these two images. This is reflected in what was observed visually as the images upsampled using this technique had the best resolution. The other two techniques, Nearest Neighbour and Bilinear Interpolation, had much lower PSNR which reflected lower quality images. This was also reflected visually in the image resolution. The Nearest Neighbour upsampled images had the lowest PSNR values and the lowest resolution images visually. The Bilinear Interpolation upsampled images had slightly higher PSNR values and had slightly higher image resolutions visually.

4. Point Operations for Image Enhancement

In Section 4, point operations were studied to determine how they could improve image contrast. Here, the tire image was used as shown in Figure 8. First, a plot of the histogram of the original image was developed to analyze its pixel intensity distribution (Figure 9). Next, the image was negatively transformed (Figure 10) and its corresponding histogram (Figure 11) was plotted. Further, two power-law transformations were applied with $\gamma = 0.5$ and $\gamma = 1.3$.These transformed images (Figures 12, 14) as well as their respective histograms (Figure 13, 15) were plotted. Lastly, histogram equalization was performed on the original image and it was plotted along with its histogram (Figures 16, 17). The code for this section can be seen in Appendix A, Figures A.3.1 and A.3.2.

Figure 8: Tire Image

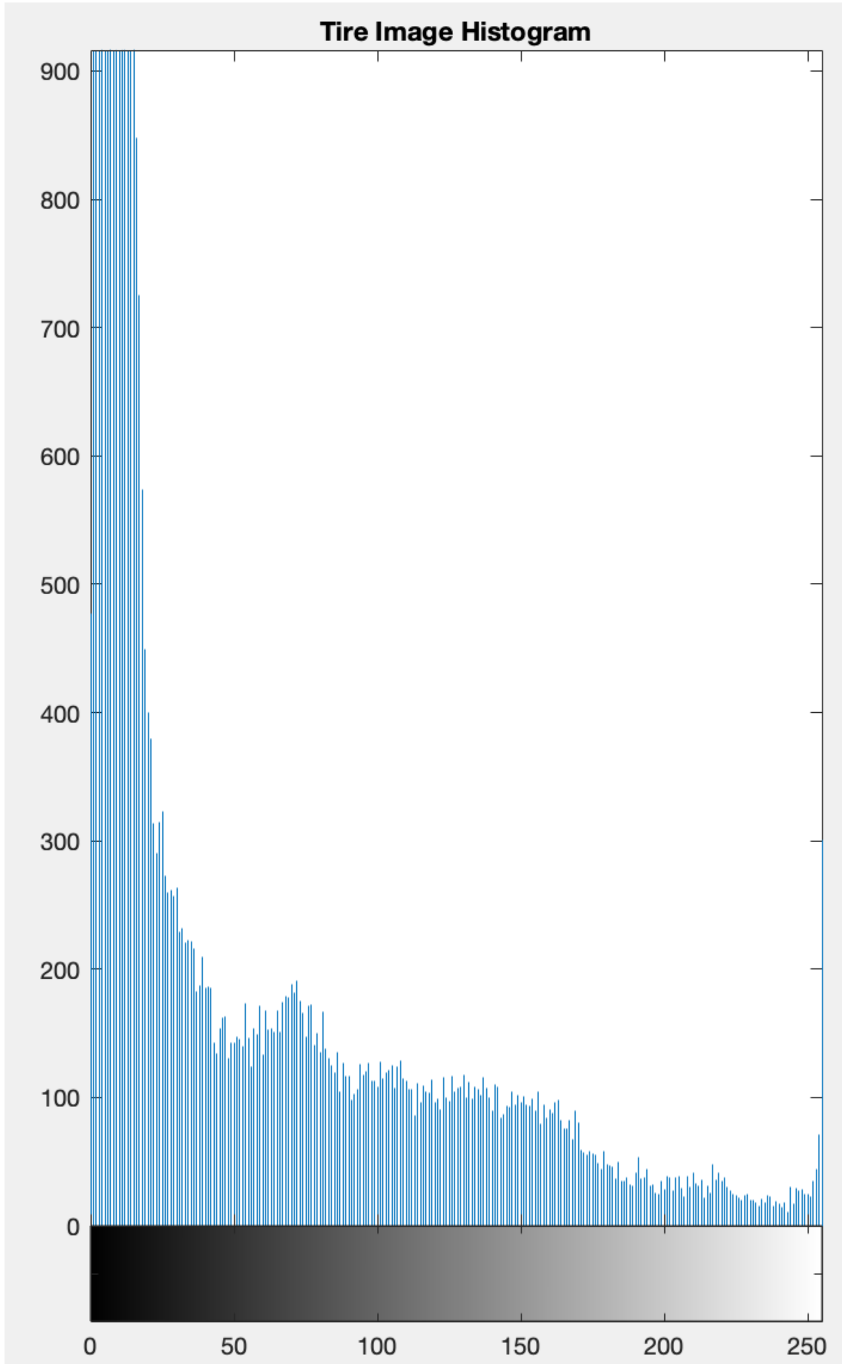

Figure 9: Histogram of Tire Image

6. The histogram shows the frequency of pixels at a specific intensity level ranging from [0, 255] for the tire image. It can be observed that the image is predominately dark as most of the pixels fall within the [0, 50] intensity levels. These pixels vary from blacks to dark greys. Considering the higher end of the intensity gradient, >200, there are significantly fewer light/white pixels as compared to the lower intensity darker pixels. The rest of the pixels fall within the grey intensity range.

It is useful in image segmentation and edge detection. For these algorithms, a histogram of an image, similar to Figure 9, can be used to determine the distribution of pixel intensities in an image. For example, from Figure 9, it is clear that there are many more pixels with lower intensities (darker) than there are higher (or brighter). A histogram can also help to show specific intensities where there is a peak or a low amount of pixels.

7. The x-axis of the histogram in Figure 9 is of the intensity values from [0, 255] (using 8-bit data). As explored in the previous question, the histogram is a graphical representation of the tonal distribution of the image, Figure 8. By inspection of the histogram, the tonal distribution or lightness of the image is seen quickly. With respect to the tire image in Figure 8, it is observed that the majority of pixels are dark with intensity values lower than 50 and there are relatively fewer light pixels with intensity values higher than 200. The histogram is also left-skewed which implies the image has a higher number of darker pixels.

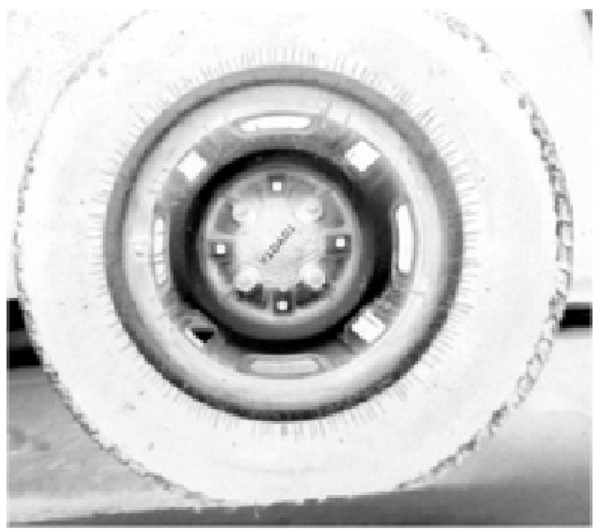

Figure 10: Negative Transformation of Tire Image

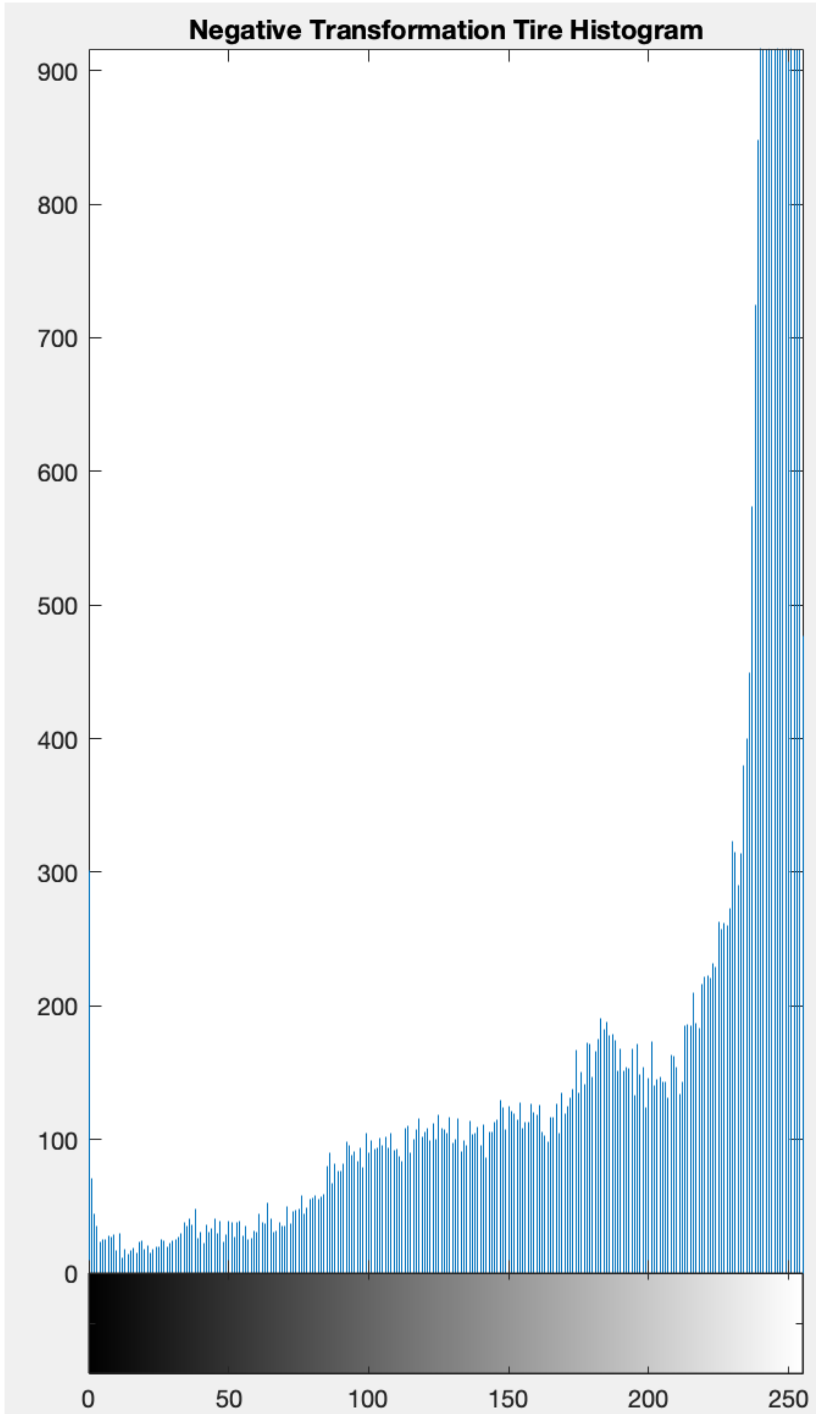

Figure 11: Histogram of Negative Transformation of Tire Image

8. The histogram in Figure 11 is the negative or reverse of the histogram in Figure 9. Each of the pixels in the original image are reversed thus there are now more bright pixels where there used to be darker pixels and vice versa. The current histogram shows the tonal distribution of the negative image, Figure 10. The image has a large number of light pixels, with intensity values

greater than 200, and the image has a small number of dark pixels, with intensity values less than 50. This reflects that the most frequent pixels in the negative tire image are light and close to white, with small amounts of dark pixels. The negative histogram shows the dark and light pixels being swapped/reversed. The negative histogram is the original histogram reflected upon the y-axis. Essentially, 255 is subtracted by each pixel's intensity value and that produces the negative histogram. This changed the skew of the histogram to a right-skew, which is exemplified in the increased brightness of the image.

Figure 12: Tire Image with Power Transformation ($\gamma = 0.5$)

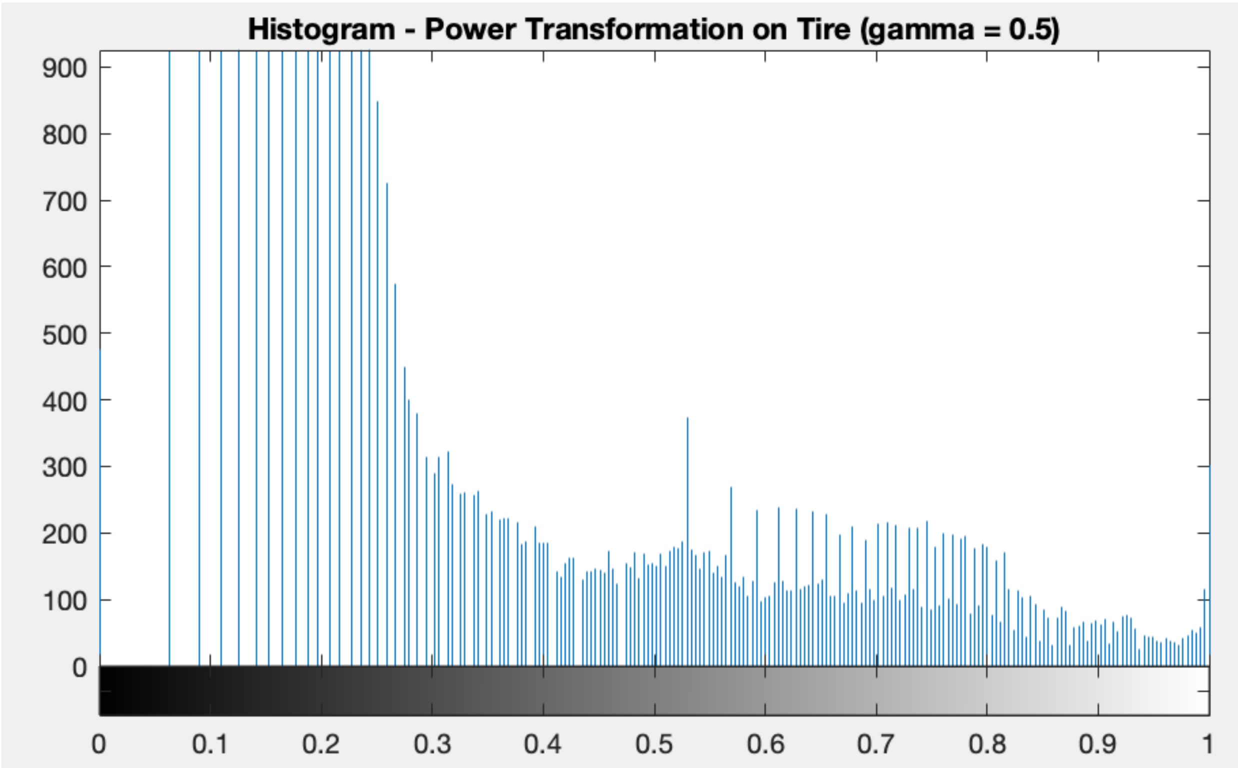

Figure 13: Tire Image Histogram with Power Transformation ($\gamma = 0.5$)

Figure 14: Tire Image with Power Transformation ($\gamma = 1.3$)

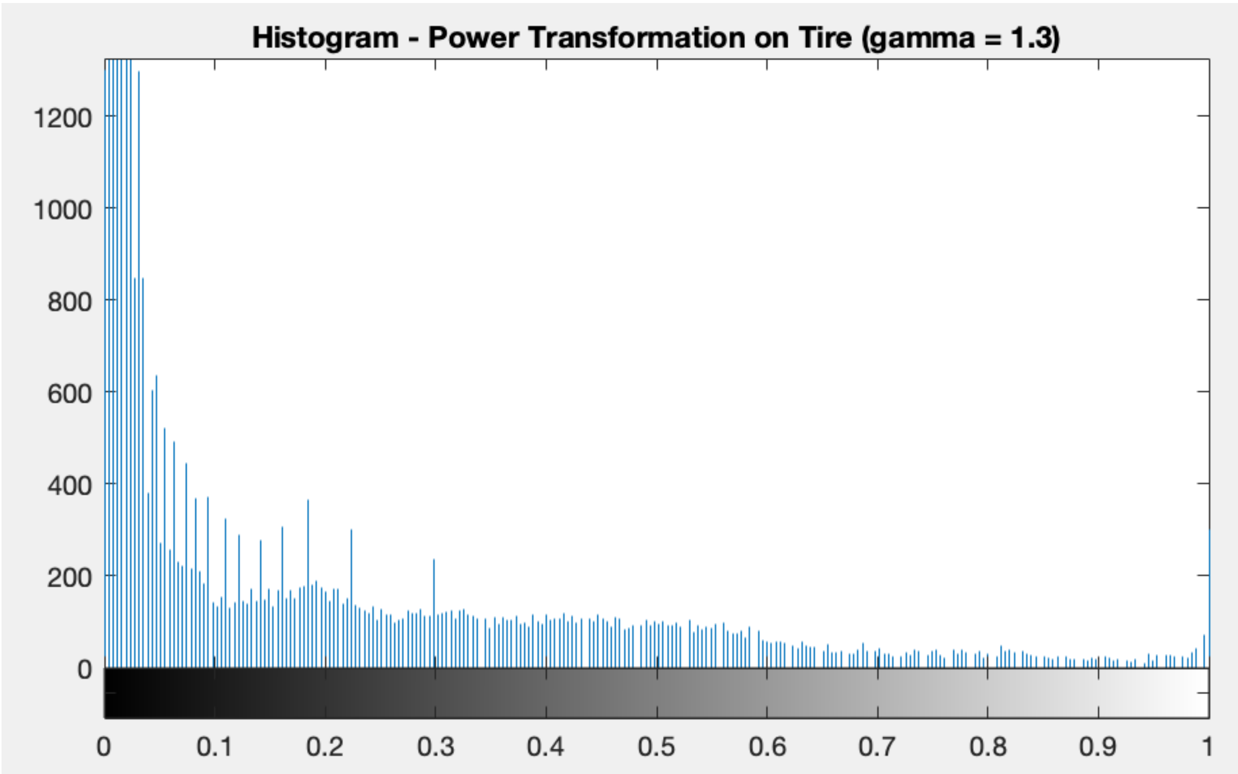

Figure 15: Tire Image Histogram with Power Transformation ($\gamma = 1.3$)

9. The transformed images are shown in Figure 12 and Figure 14. Figure 12 is of the power transformation with $\gamma = 0.5$and that appears as a brighter, whiter image when compared to the original. Figure 14 is of the power transformation with $\gamma = 1.3$and that transforms the original image to a darker, blacker version.

The effect of γon an image is dependent on its value. If it is greater than 1 then, the dark pixels and their related intensity values are compressed while the brighter pixels and their related intensity values are stretched then distributed over the remaining intensity values. This makes the image, as seen in Figure 14, darker and is useful for saturation spots, overexposed and overall bright images. Doing so increases the contrast and detailing in bright areas of the image. If $\gamma$is lower than 1 then, the bright pixels and their related intensity values are compressed while the darker pixels and their related intensity values are stretched then distributed over the remaining intensity values. This makes the image, as seen in Figure 12, brighter and is useful for increasing the detailing in dark areas of the image.

10. The histogram of the brighter, $\gamma = 0.5$ power transformation, Figure 13, still maintains a similar shape to that of the original histogram. This is because the darker, lower intensity valued pixels are stretched to the dark-grey lighter regions. The brighter pixels are compressed which results in a brighter image, reflected on the histogram. Thus, the histogram is skewed to the left,

however, the pixel intensities are more spread out or stretched as compared to the original. Thus, the image, Figure 12, has more pixels in the dark to light grey range (it is brighter).

The histogram of the darker, $\gamma = 1.3$ power transformation, Figure 15, still maintains a similar shape to that of the original histogram. However, there are more darker pixels and less lighter pixels. The histogram has compressed the dark pixel intensity ranges and stretched out the brighter ones towards the darker end. This results in a histogram with predominantly dark pixels and low intensity values. As such, the image, Figure 14, has darker pixel intensities and more pixels in this lower intensity range.

11. Since the original image is dark, the majority of pixels exist within the low intensity range. The detailing in the dark regions of the image is unclear and it is difficult to visibly differentiate the components in the image. To increase the quality of the image and increase the detailing, the image should be brightened. Brightening the image implies compressing the bright/lighter pixel regions and then stretching the darker pixel regions towards the lighter end of the intensity scale. This operation would make the dark spots brighter, increasing the detailing. To make this transformation, a power transformation can be applied to the original tire image with a γvalue less than 1. Thus, the $\gamma = 0.5$ power-law transformation should be used to enhance the image.

Figure 16: Equalized Tire Image

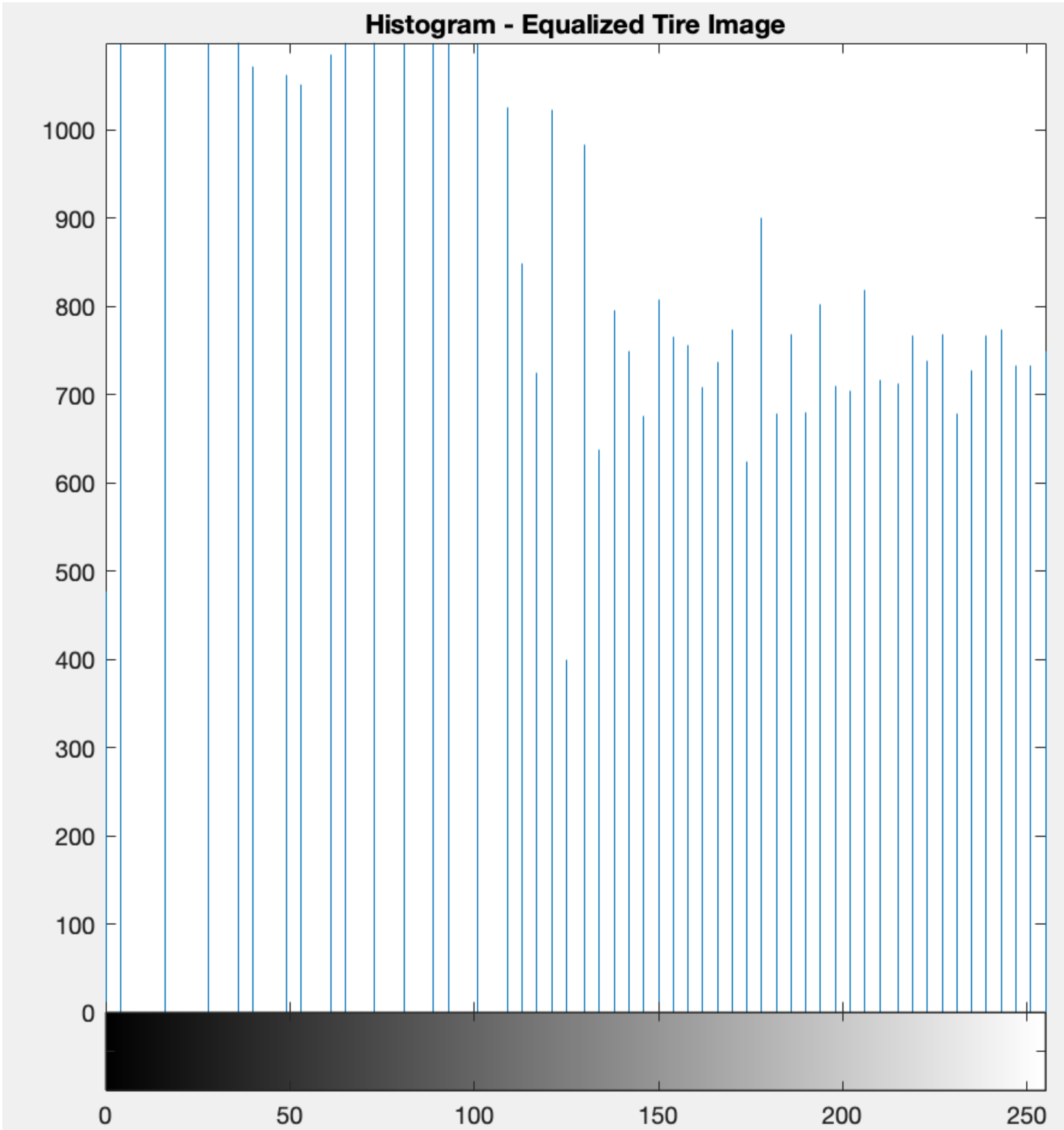

Figure 17: Equalized Tire Image Histogram

12. The equalized image, Figure 16, has greater detailing in the dark and light regions. The detailing in the tire and rim sections are more visible. The original tire image has been stretched to increase the contrast as there are more regions with higher pixel intensity (brighter). The different components of the image are clearer to the viewer. The image is more neutral in terms of lightness. By visual inspection, the image appears more balanced with equal amounts of dark and light regions when compared to the original.

13. The histogram in Figure 17 has an intensity distribution that is much more spread out than the original. The original image had an associated histogram that was skewed left towards lower

intensity pixels, however, this histogram is generally flat (no skewing). This is because histogram equalization was performed. Thus, the image has an almost equal amount of pixels in a variety of intensities from lower to higher. Equalization affected the image in that the intensities with a higher amount of pixels (low intensities) were stretched whereas the intensities with lower pixels (high intensities) were compressed. The equalized histogram is flat which is why the image has even amounts of dark and light pixels. The original pixel intensities were equally distributed to achieve this result.

5. Conclusion

In summary, this lab explored image enhancement techniques and ways to measure and evaluate the quality of images resulting from the use of these methods. This lab involved first establishing empirical measures for image quality to study the effects of image enhancement algorithms, specifically focusing on Nearest Neighbour, Bilinear Interpolation, and Bicubic Interpolations. These empirical measures included Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE) which measure the noise or error in reconstructed images. This helps to inform one about how similar or different the resampled image is to the original. It was observed that when looking at the PSNR value and the image visually, the Bicubic Interpolation method worked best for both Lena and Cameraman, producing the best resolution and highest PSNR values in the resampled images. Section 4 involved a study of how point operations are used to enhance image contrast using the tire image. Here, the image underwent negative transformation, power-law transformation (with $\gamma = 0.5$ and $\gamma = 1.3$), and histogram equalization. Since the original image had more dark or low intensity pixels, it was observed that the power transformation of $\gamma = 0.5$improved the image most due to its ability, since $\gamma < 1$, to stretch the lower (darker) intensity pixels and compress the higher (brighter) intensity pixels. This improved image contrast as there were more bright pixels in the darker areas, allowing one to see the details in these areas easily.

Appendix A

```matlab
%% Section 2 - Image Quality Measures
% Create a new function PSNR that, given a reference image f and a test image
% g as inputs, outputs the PSNR value PSNRout.

function PSNR_out = PSNR(f, g)
    % f - Reference Image
    % g - Test image

    MAX_f = 255; % Assume that MAXf= 255 since using 8-bit data

    % Get size of reference image f
    m = size(f, 1); % num of rows
    n = size(f, 2); % num of cols

    squared_diff = (double(f) - double(g)).^2; % convert to double so image is not read as integer

    % Calculate Mean Squared Error
    MSE = (1/(m*n)) * sum(squared_diff(:));

    % Calculate Peak Signal to Noise Ratio
    PSNR_out = 10 * log10(MAX_f^2 / MSE);

end
```

Figure A.1: MATLAB code for PSNR Function

```matlab
%% Section 3 - Digital Zooming

% Load lena and Cameraman Images and convert Lena to Grayscale using rgb2gray
% function (Cameraman is already a grayscale image)

lena_image = imread("lena.tiff");
cameraman_image = imread("cameraman.tif");

lena_grayscale = rgb2gray(lena_image);

% Reduce resolution of images by factor of 4 in horizontal & vertical
% directions using bilinear interpolation
lena_reduced = imresize(lena_grayscale, 0.25, 'bilinear');
cameraman_reduced = imresize(cameraman_image, 0.25, 'bilinear');

% Plot downsampled images
figure;
subplot(1,2,1), imshow(lena_reduced);
title('Image 1: Reduced Lena');
subplot(1,2,2), imshow(cameraman_reduced);
title('Image 2: Reduced Cameraman');

% Perform digital zooming to increase resolution of downsampled images back
% to resolution of original images (with a factor of 4):

% 1. Using Nearest Neighbour Interpolation
lena_nn = imresize(lena_reduced, 4, 'nearest');
cameraman_nn = imresize(cameraman_reduced, 4, 'nearest');

```

Figure A.2.1: MATLAB code for Section 3

```
% 2. Using Bilinear Interpolation
lena_bilinear = imresize(lena_reduced, 4, 'bilinear');
cameraman_bilinear = imresize(cameraman_reduced, 4, 'bilinear');

% 3. Using Bicubic Interpolation
lena_bicubic = imresize(lena_reduced, 4, 'bicubic');
cameraman_bicubic = imresize(cameraman_reduced, 4, 'bicubic');

% Plot the upsampled images
figure;
subplot(3,3,1), imshow(lena_nn);
title('Subplot 1: Nearest Neighbour - Lena');
subplot(3,3,2), imshow(lena_bilinear);
title('Subplot 2: Bilinear Interpolation - Lena');
subplot(3,3,3), imshow(lena_bicubic);
title('Subplot 3: Bicubic Interpolation- Lena');

subplot(3,3,4), imshow(cameraman_nn);
title('Subplot 4: Nearest Neighbour - Cameraman');
subplot(3,3,5), imshow(cameraman_bilinear);
title('Subplot 5: Bilinear Interpolation - Cameraman');
subplot(3,3,6), imshow(cameraman_bicubic);
title('Subplot 6: Bicubic Interpolation - Cameraman');

% Compute PSNR between original images and each of the upsampled images
% 1. Nearest Neighbours
lena_nn_psnr = PSNR(lena_grayscale, lena_nn);
cameraman_nn_psnr = PSNR(cameraman_image, cameraman_nn);

```

Figure A.2.2: MATLAB code for Section 3 (cont'd)

```
% 2. Bilinear Interpolation
lena_bilinear_psnr = PSNR(lena_grayscale, lena_bilinear);
cameraman_bilinear_psnr = PSNR(cameraman_image, cameraman_bilinear);

% 3. Bilinear Interpolation
lena_bicubic_psnr = PSNR(lena_grayscale, lena_bicubic);
cameraman_bicubic_psnr = PSNR(cameraman_image, cameraman_bicubic);
```

Figure A.2.3: MATLAB code for Section 3 (final part)

```
%% Section 4 - Point Operations for Image Enhancement

% Plot the tire image and the histogram using imhist
tire_image = imread("tire.tif");

figure;
subplot(1,2,1), imshow(tire_image);
title('Tire Image');
subplot(1,2,2), imhist(tire_image);
title('Tire Image Histogram')

% Apply image nagtive transform on the image. Plot image and histogram
neg_tire = 255 - tire_image; % Using 255 as max because 8-bit data

figure;
subplot(1,2,1), imshow(neg_tire);
title('Negative Tranformation of Tire Image');
subplot(1,2,2), imhist(neg_tire);
title('Negative Transformation Tire Histogram');

% Apply 2 power-law transformations
% Normalize tire image and convert to double so it isn't read as int
tire_image_norm = double(tire_image)/255;

% γ = 0.5
gamma_1 = 0.5;
tire_power_1 = tire_image_norm.^gamma_1;
% γ = 1.3
gamma_2 = 1.3;
tire_power_2 = tire_image_norm.^gamma_2;

```

Figure A.3.1: MATLAB code for Section 4

```
% Plot two transformed images and histograms
figure;
subplot(2,2,1), imshow(tire_power_1);
title('Power Transformation on Tire Image (gamma = 0.5)');
subplot(2,2,2), imhist(tire_power_1);
title('Histogram - Power Transformation on Tire (gamma = 0.5)');
subplot(2,2,3), imshow(tire_power_2);
title('Power Transformation on Tire Image (gamma = 1.3)');
subplot(2,2,4), imhist(tire_power_2);
title('Histogram - Power Transformation on Tire (gamma = 1.3)');

% Perform Histogram Equalization on tire image using histeq
tire_equalization = histeq(tire_image);

% Plot equalized image and its histogram
figure;
subplot(1,2,1), imshow(tire_equalization);
title('Equalized Tire Image');
subplot(1,2,2), imhist(tire_equalization);
title('Histogram - Equalized Tire Image');
```

Figure A.3.2: MATLAB code for Section 4 (cont'd)