SYDE 575 Lab #1: Fundamentals of Image Processing

Nedim Hodzic – 20627314

David Ramon Prados - 20695485

Section 1: Introduction

The contents of this report contain the results and findings found from lab #1: "The Fundamentals of Image Processing". This lab investigates different image processing evaluation techniques such as PSNR, up-sampling and down-sampling, and point operations. Some of the up/down sampling techniques examined include nearest-neighbor interpolation, bilinear interpolation and bicubic interpolation. The point operation methods include histograms and power-law transformations in order to gain insight regarding image enhancement. The findings of these different techniques can be found in the sections below.

Section 2: Image Quality Measures

MATLAB code included in appendix

Section 3: Digital Zooming

Figure 1: Original Images vs Down sampled Images

From figure 1 the top row of images contains the original greyscale images while the second row contain the images that have had their resolution reduced by a factor of 4.

Figure 2: Up sampled Images using Nearest Neighbor Interpolation

Figure 3: Up sampled Images using Bilinear Interpolation

Figure 4: Up sampled Images using Bicubic Interpolation

Up sampling Method	Lena Image PSNR Value	Cameraman Image PSNR Value

Nearest Neighbor Interpolation	33.97	32.92
Bilinear Interpolation	34.27	32.62
Bicubic Interpolation	34.71	32.89

Table 1: PSNR Values for each Up-Sampling Method

Question #1: From the figures and images above, it can be seen that the up sampled image fail to reach the quality/resolution of the original images. It can also be observed that the quality of the images vary based on the up-sampling methods used. The reduced image quality can be attested to the original down-sampling that was done on the images. The down sampling caused the pixel values to change within the images which makes it extremely hard to get them back to their original values especially using the fairly straightforward up-sampling algorithms.

Question #2: From examining figures 2,3 and 4, it can be seen that the quality of the up-sampled begin to vary in quality based on the up-sampling method used. For nearest neighbor interpolation, it can be seen in figure 2 that the images have the poorest quality. This makes sense because nearest neighbor interpolation is the simplest method of the ones used. When looking at the cameraman picture in figure 2, it can be seen that a distinct 'staircase' effect is formed in the image which is a common occurrence in nearest neighbor interpolations. In regards to PSNR, the lena image had its lowest PSNR value for nearest neighbour interpolation while the cameraman image had its second lowest score. Moving onto bilinear interpolation, in figure 3 it can be seen that the images are not as grainy but do appear blurred. The staircase effect is gone from the cameraman image but it does appear significantly blurrier than the original one. This is the case because bilinear interpolation interprets pixel values based on its 4 closest neighbors which can result in less precise pixel values. As a result, the lena image had its second highest PSNR score while the cameraman had its lowest PSNR value which can be attributed to its extremely blurry appearance. Looking at the final up-sampling method in bicubic interpolation, it can be seen from figure 4 that both images appear to look their best when compared to the original ones, although there is still a small amount of blurriness present. This makes sense because bicubic interpolation determines pixel values based on 16 of its closest neighbors which can result in much more precise pixel interpretations. As a result, both lena and cameraman images had their highest PSNR scores for bicubic interpolation.

Question #3: Based on the images in figures 2,3 and 4, it can be seen that all of the up-sampling methods do a good job in preserving the integrity of the image. They do a great job transitioning between darker and lighter areas in order to create distinct figures and shapes. It is easy identify what each image is regardless of the up-sampling method. However, all 3 methods do a poor job of up-scaling the images back to their original quality. As mentioned previously, this could be due to the down-sampling that originally occurred and changed the pixel values and dimensions of the images. The three up-sampling algorithms are simply not complex enough to effectively sample the images up back to their original quality. However, it can be seen that a method like

bicubic interpolation yielded the best results as it referenced a higher number of surrounding pixels in the upscaling process while a method like nearest neighbor interpolation yielded the worst results as it is the simplest algorithm out of the 3.

Question #4: When comparing the PSNR values found in table 1, it can be seen that the lena image has the highest PSNR score for all 3 up-sampling methods when compared to the cameraman image. This is the case because the lena image is actually twice as large as the cameraman image in terms of pixels (512x512 vs 256x256). This gives the up-sampling method access to more pixel values which allows more accurate and precise interpretations to be made. This in turn results in higher image qualities and higher PSNR values.

Question #5: Looking back at table 1, it can be seen that the highest PSNR values occurred in the bicubic interpolations while the lowest values mostly occurred with the nearest neighbor interpolation. The higher the PSNR value, the more similar the up-scaled image is to the original image. Therefore, it makes sense that the bicubic interpolations yielded the highest PSNR scores as that specific up-scaling method is meant to be the most precise one. In terms of lowest PSNR, the lena image had its lowest score when using nearest neighbor interpolation while the cameraman image had its lowest score using bilinear interpolation. Although nearest neighbor interpolation is meant to be the least precise method out of the 3, bilinear interpolation produced a blurry cameraman image which contributed to the low PSNR score. It is also important to note that the PSNR score for bilinear interpolation was only slightly smaller than the score for nearest neighbor interpolation.

Section 4: Point Operations for Image Enhancement

We start by plotting the tire image and its histogram:

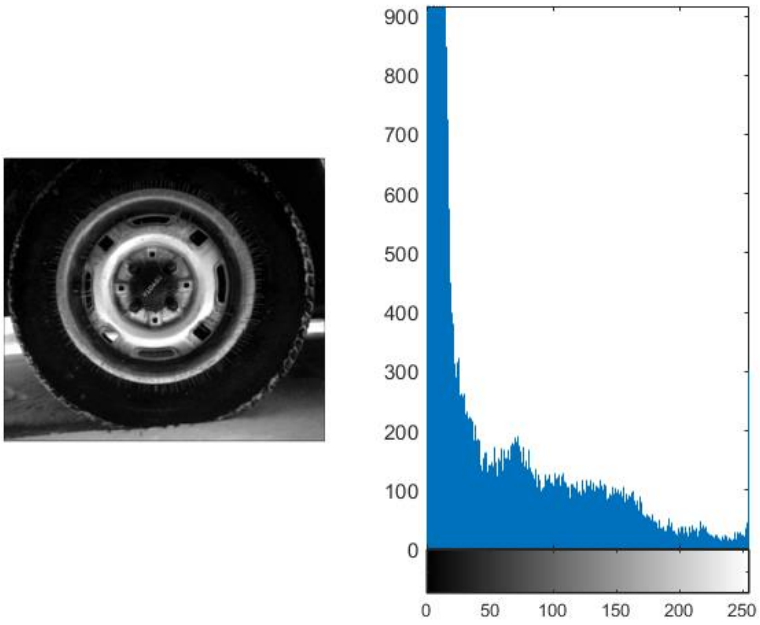

Figure 5: Tire Image and its Histogram

Question #6: The histogram of an image is a graphical representation of the pixel intensity values of a digital image. In the case of a black and white image such as the tire image shown in Figure 5, the histogram is simply the distribution of the grey levels in the image. Histograms are used for a wide variety of purposes since they summarize discrete or continuous data. They enable very useful point processing techniques, and modifications of original histograms are often used for image enhancement procedures.

Question #7: The histogram of the tire image shown in Figure 5 is heavily distributed towards the left. These means, that darker tones or grey levels are dominant in the image. The levels between 0 and 25 are the most frequent in the image, and they represent most of the pixel values. Consequently, the histogram shows how this is a very dark image, since most of the pixel values are situated in the left side of the histogram.

Next, we find the negative image transform. This can be done using Matlab built in function *imcomplement*. However, since we are using a gray scale image, it can be easily achieving by subtracting the image to 255:

$$image = 255 - image$$

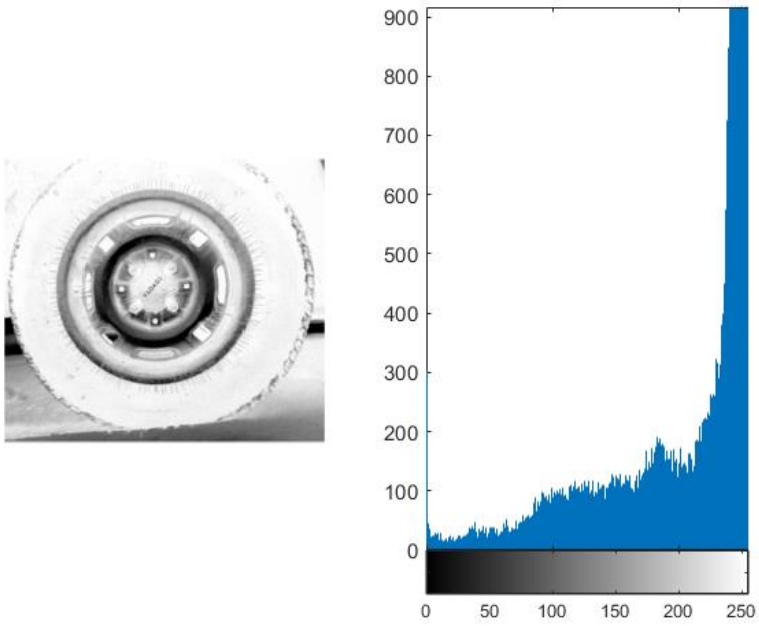

Figure 6: Negative Tire Image and its Histogram

Question #8: As expected, the histogram of the negative image is the inverse of the original. The histogram shown in Figure 6 is heavily distributed towards the right. These means that this is a very bright image with many saturated pixels towards the max intensity (255). Most of the data points are between 230 and 255, supporting the idea that the brightness of the image is large. This makes sense since the image was calculated by subtracting the pixel values to 255, which translates to flipping the grey levels. Since the histogram of a grey image is the distribution of the grey levels, the resultant histogram will be the opposite of the original one.

We continue by applying power-law transformations to the image. This can be done by iterating through the image values and applying the formula given, or we can use Matlab vector operations to avoid for loops. However, when using vector operations, the image must be converted a double first. After the transformation is done, we round the result (convert back to UINT8) since we are working in the discrete domain. Figures 7 and 8 show the result of the power transformations, and the code can be found in the appendix,

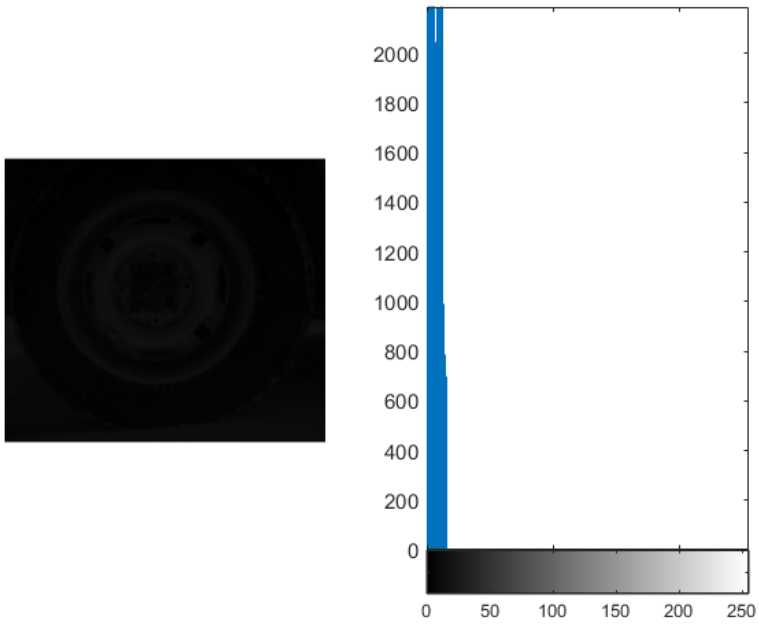

Figure 7: 0.5 Power Transformation

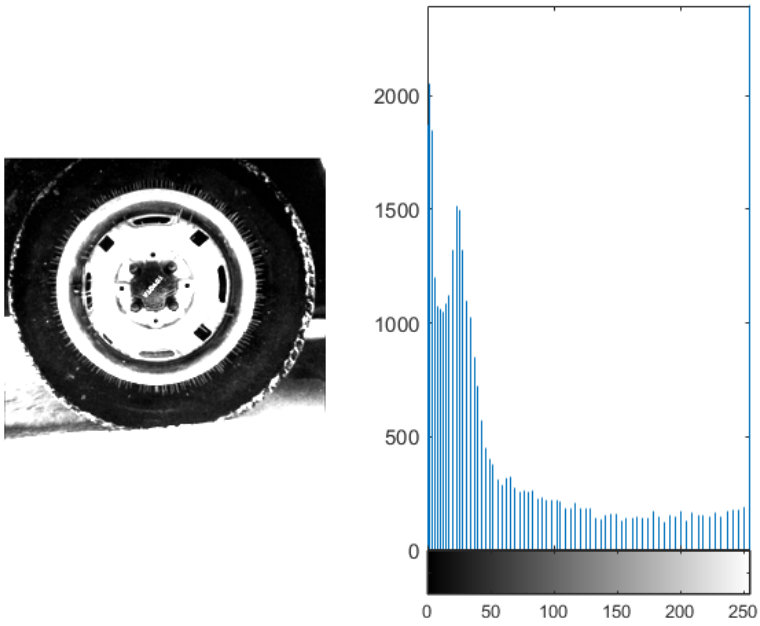

Figure 8: 1.3 Power Transformation

Question #9: For the 0.5 power transformation the result is shown in Figure 7. The result is even darker than the original image. This is because we used a square root transformation, which compresses the bright areas and stretches the darker ones. Since the original image was mostly

dark, these predominant grey levels got stretched to become even more dominant. On the other hand, the few bright tones got stretched and became almost imperceptible. This is a very low contrast image.

The result for the 1.3 power transformation is shown in Figure 8. In this case, the result is an enhanced image where the features can be differentiated easier. By using a quadratic like transformation, the dark areas get compressed while the brighter areas get stretched. The slope acts as the decision maker for at which grey level we switch from compressing to stretching. In result, the few bright pixels from the original image got stretched becoming more visible, while the number of dark pixels is reduced. In consequence, this transformation increased the image contrast making the features easier to discern.

Question #10: In terms of the histogram changes, there are some aspects common to both transformations, while others are specific to each transformation. Starting with the changes common to both histograms, we see that the density of bins has been reduced with respect to the original histogram. This is caused by the rounding operation in the transformation calculation. Alternatively, the scale on Y-axis increased from 900 to 2000. The number of pixels or data points is unaltered, but since we have less bins, the number of pixels per bin must increase.

Next, we look at each histogram individually. For the 0.5 power transformation, the resultant histogram is even more left sided than the original one. This explains why the transformed image is darker than the original one. Since we applied a square root transformation to an already dark image, the number per pixel for the dark levels (0-25) dramatically increased, while the few bins for the bright tones got stretched and almost disappeared. In result, this histogram has the shape of very low contrast image which matches the observations made about the image.

For the 1.3 power transformation, the resultant histogram is more balanced than the original one. The number of saturated tones (255 level) dramatically increased (stretched), while the darker levels got stretched. In the original histogram most data points for the darker tones were between 0 and 25, while on the 1.3 transformed one are more evenly distributed between 0 and 40. Additionally, the number of pixels for the darkest tone (0) also increased. It can also be seen that the bright tones between 200 and 255 also increased, but since the starting number was very low, the effect is less dramatic. These changes align with the observations made for the transformed image. The 1.3 transformation compressed dark tones and stretched brighter grey levels, resultant in a higher contrast image than the original.

Question #11: The 1.3 transformation should be used to enhance the image. This is because the original image was very dark, with very few bright pixels and most of its data points between the 0 and 25 levels. In other words, the contrast on the original image was low with a left side histogram. With a quadratic like transformation, the darker tones get compressed and the bright levels get stretched. This operation balances the histogram and results in a higher contrast image were the features are clearer. Therefore, the 1.3 transformation is the correct point operation for image enhancement.

Last, we apply a histogram equalization operation to the original image.

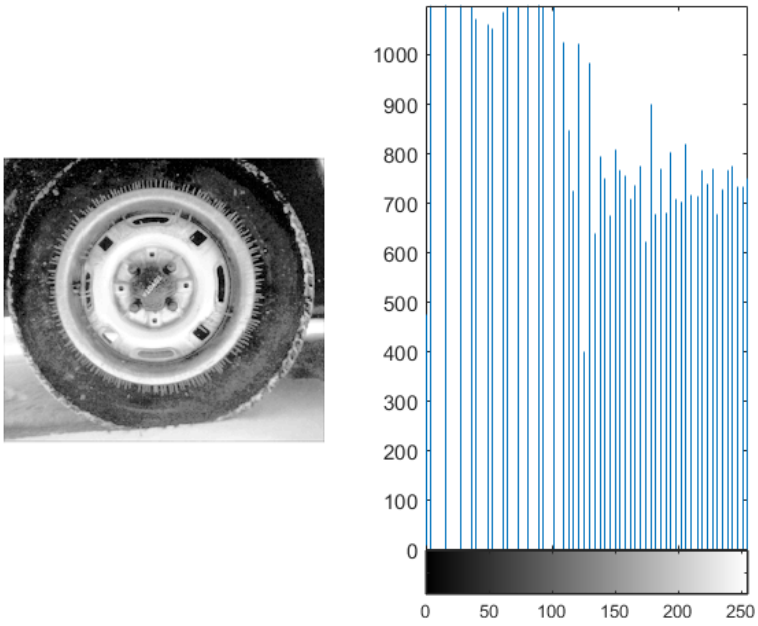

Figure 9: Histogram Equalization Result

Question #12: Figure 9 shows the results of the histogram equalization operation. The resulting image is an enhanced picture respect to the original one. The features are more visible, and it is possible to observe new ones that were hidden in the original image. The tire threads are visible, and the wheel does not blend with the background anymore. In addition, the range of grey levels are more evenly distributed, with less dark or very bright areas observed. While the feature boundaries are not as obvious as in the 1.3 transformation, this image offers much finer details.

Question #13: The equalized histogram is more evenly distributed for all grey levels. The probability for each intensity level is closer than in any of the previous histograms. The goal of a histogram equalization is to make each tonality equally probable. However, for a discrete case such as the one presented in this report this is not possible due to the rounding operations. In any case, this is still a valuable operation that outputs a better result than the original image. While the contrast is not as high as in the 1.3 transformation, an equalized histogram offers a higher level of detail with less saturated blocks. If we were in the continuous case, we would achieve a perfectly flat histogram. However, for the discrete case this is only possible in the trivial case.

Section 5: Conclusion

In conclusion, this lab taught us about various different imaging processing techniques that can be used to analyze different kinds of images. From sections 2 and 3, it can be seen that the usage of nearest neighbor, bilinear and bicubic interpolation can be used to up-sample images with reduced quality in order to produce an image. From the results found in that section, it could safely be said that bicubic interpolation does the best job of enhancing images while nearest neighbor interpolation does the poorest job. This is further supported by the fact that bicubic interpolated images had the highest PSNR scores while nearest neighbor interpolated images had the lowest score. In section 4, the applications of histograms for image analysis were investigated. From the results in that section it can be seen that applying an integer less than 1 for power-law transformations results in an extremely dark image while applying an integer greater than one produces a lighter and higher quality image. It can also be seen that applying an equalized histogram to an image also produces a more evenly distributed histogram which in turn produces a higher quality image than the original one being examined.

Appendix:

Matlab code starts in the next page.

```
%Lab 1

%Image Quality Measures

%example useage of the PSNR function
lena_normal = imread('lena.tiff');
lena_noise = imnoise(lena_normal,'gaussian',0.05);

%figure(1)
%imshow(lena_normal);

%figure(2)
%imshow(lena_noise);

PNSR_out = PSNR(lena_normal,lena_noise);

%Digital Zooming

%loading the original images in
lena = imread('lena.tiff');
cameraman = imread('cameraman.tif');

%converting to grey (cameraman has no rgb values)
lena_gray = rgb2gray(lena);

%resizing using bilinear interpolation
lena_resize = imresize(lena_gray,0.25,'bilinear');
cameraman_resize = imresize(cameraman,0.25,'bilinear');

%comparing grayscale images to reduced resolution ones
figure;
subplot(2,2,1),imshow(lena_gray);
subplot(2,2,2),imshow(cameraman);
subplot(2,2,3),imshow(lena_resize);
subplot(2,2,4),imshow(cameraman_resize);

%Increasing resolution using nearest neighbour interpolation
lena_nn = imresize(lena_resize,4,'nearest');
cameraman_nn = imresize(cameraman_resize,4,'nearest');

figure;
subplot(1,2,1), imshow(lena_nn);
subplot(1,2,2), imshow(cameraman_nn);

lena_nn_PSNR = PSNR(lena_gray,lena_nn);
cameraman_nn_PSNR = PSNR(cameraman,cameraman_nn);

%Increasing resolution using nearest bilinear interpolation
lena_bilinear = imresize(lena_resize,4,'bilinear');
```

```
cameraman_bilinear = imresize(cameraman_resize,4,'bilinear');

figure;
subplot(1,2,1), imshow(lena_bilinear);
subplot(1,2,2), imshow(cameraman_bilinear);

lena_bilinear_PSNR = PSNR(lena_gray,lena_bilinear);
cameraman_bilinear_PSNR = PSNR(cameraman,cameraman_bilinear);

%Increasing resolution using nearest bicubic interpolation
lena_bicubic = imresize(lena_resize,4,'bicubic');
cameraman_bicubic = imresize(cameraman_resize,4,'bicubic');

figure;
subplot(1,2,1), imshow(lena_bicubic);
subplot(1,2,2), imshow(cameraman_bicubic);

lena_bicubic_PSNR = PSNR(lena_gray,lena_bicubic);
cameraman_bicubic_PSNR = PSNR(cameraman,cameraman_bicubic);

% Point Operations for Image Enhancement
target = 'tire.tif';

% Initial image
tire = imread(target);
histo = imhist(tire);
figure;
subplot(1,2,1), imshow(tire);
subplot(1,2,2), imhist(tire);
% Question 6 -> report
% Question 7 -> report

%Negative of the image
tire_neg = 255 - tire;
figure;
subplot(1,2,1), imshow(tire_neg);
subplot(1,2,2), imhist(tire_neg);

%Verfiiy it worked
tire_neg2 = imcomplement(tire);
figure;
subplot(1,2,1), imshow(tire_neg2);
subplot(1,2,2), imhist(tire_neg2);
%Question 8 report

%Power-law transformations
tire_05 = double(tire).^(0.5);
tire_05 = uint8(tire_05);
figure;
```

```
subplot(1,2,1), imshow(tire_05);
subplot(1,2,2), imhist(tire_05);

tire_13 = double(tire).^(1.3);
tire_13 = uint8(tire_13);
figure;
subplot(1,2,1), imshow(tire_13);
subplot(1,2,2), imhist(tire_13);

% Question 9 -> report
% Question 10 -> report
% Question 11 -> report

% Histogram equalization
tire_eq = histeq(tire);
figure;
subplot(1,2,1), imshow(tire_eq);
subplot(1,2,2), imhist(tire_eq);
% Question 12 -> report
% Question 13 -> report
```