Sophie-Gabrielle Hermanns (20882908), Nicolas Palmar (20883350)

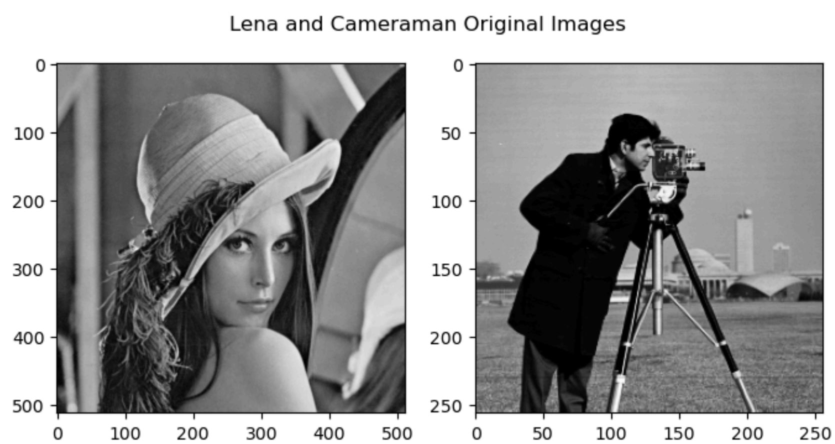
SYDE 575: Lab 1 Report

Introduction

The objective of this lab is to get practical experience with fundamental image processing concepts and techniques. This lab will explore image enhancement methods on stock images from the USC-SIPI image database, such as digital zooming and point operations. As well, this lab will contain image quality measures, using PSNR (Peak Signal to Noise Ratio) to quantitatively study the effects of image enhancement algorithms on image quality. These techniques will be applied to test images lena.tif, tire.tif, and cameraman.tif. Specifically, digital zooming and PSNR being implemented on lena.tif and cameraman.tif, whereas point operations will be applied to the tire.tif.

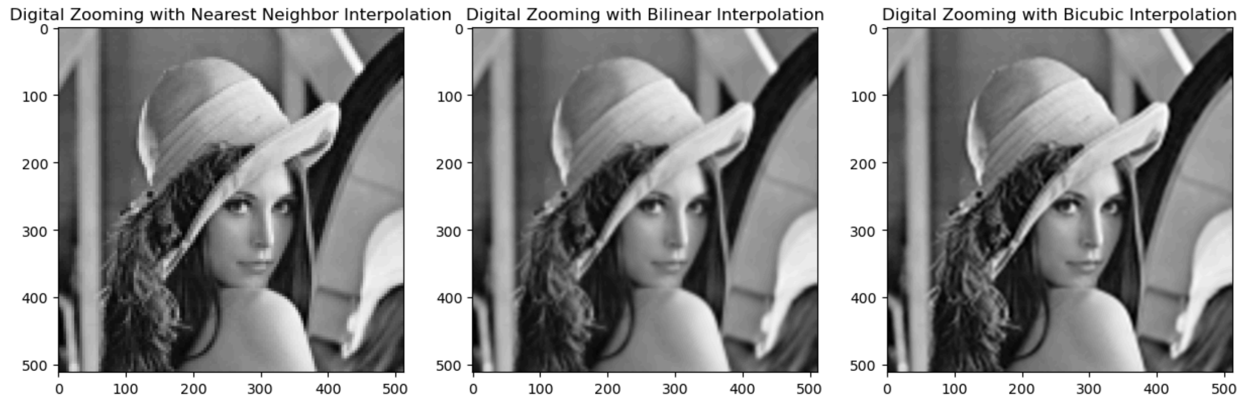
Digital Zooming

This first section of this lab was to perform digital zooming. The images of Lena and Cameraman were both reduced in resolution by a factor of 4 (both axes) using bilinear interpolation and then up-sampled using a variety of interpolation methods. We will step through the process, starting with the original images.



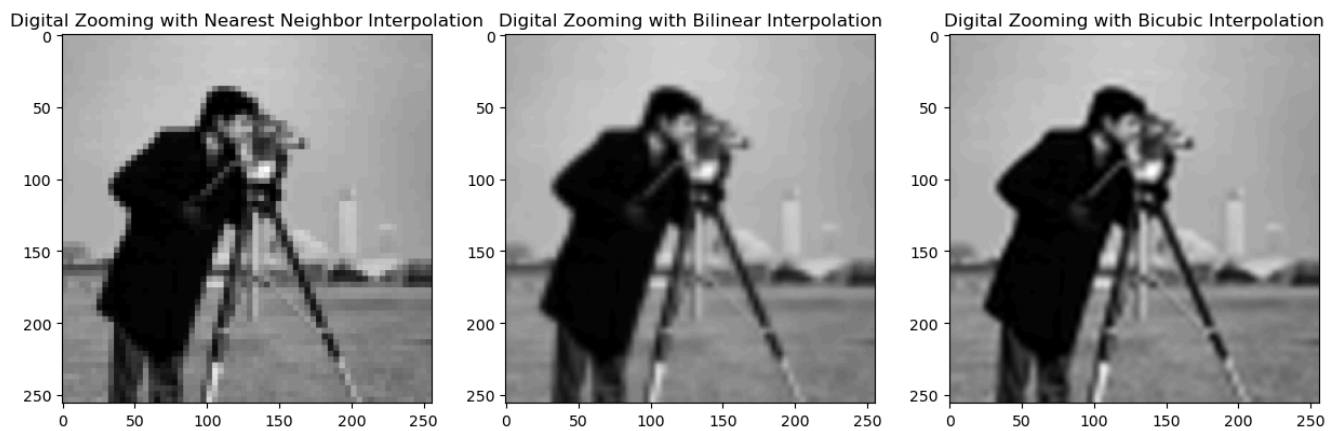
The first step was to reduce the resolution by a factor of 4 using bilinear interpolation, shown below.

Next, we will focus on Lena. The down-sampled image of Lena was up-sampled using nearest neighbor, bilinear, and bicubic interpolation. Furthermore, the PSNR values between Lena and the different up-sampled images were computed and are shown below.




```
PSNR between Lena image and up-sampled image using nearest neighbor interpolation: 26.41381416171506
PSNR between Lena image and up-sampled image using bilinear interpolation: 27.076245843461628
PSNR between Lena image and up-sampled image using bicubic interpolation: 28.14409121355567
```

Finally, we will look at Cameraman. The same process for Lena was completed with the Cameraman, showing the up-sampled images and PSNR values.




```
PSNR between Cameraman image and up-sampled image using nearest neighbor interpolation: 21.63163010175316
PSNR between Cameraman image and up-sampled image using bilinear interpolation: 21.917800814357268
PSNR between Cameraman image and up-sampled image using bicubic interpolation: 22.507738531349844
```

Digital Zooming: Questions

1. What can you observe about the up-sampled images produced by each of the methods?

All of the up-sampled images have lost information about the original image. The result produced is of lower quality, blurred images (compared to original). However, the upsampled bilinear and bicubic methods generally seem to produce a higher quality image compared to the down-sampled image. Furthermore, the upsampled images of Lena look more similar whereas those for the Cameraman vary.

2. How do the different methods compare to each other in terms of PSNR as well as visual quality? Why?

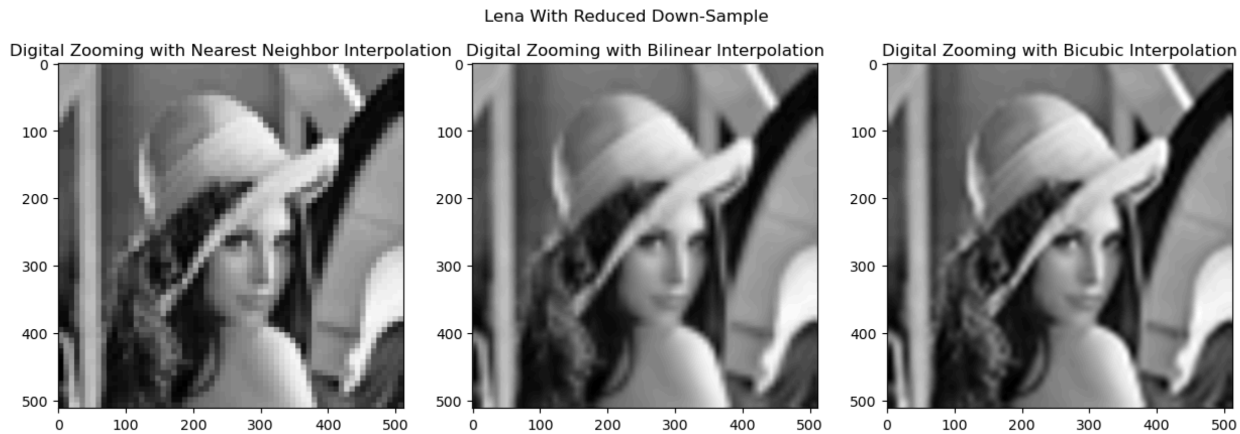
In terms of PSNR, the lowest order interpolation (nearest neighbor) was the worst, followed by bilinear, and finally followed by the cubic method (in both cases), although the results were all relatively similar. Additionally, with increasing PSNR values, the visual quality of the images generally increased. However, the visual quality only increased slightly, which is evident by the small increase in PSNR values; it is difficult to see the quality difference between each zooming method for Lena, but it is clear that the quality improves for the Cameraman. The increasing PSNR values match with the expected visual; we expect a higher peak signal to noise ratio which implies that there is a smaller MSE for the images. Furthermore, the higher order interpolation for digital zooming preserves fine details better as a result of using more information from the down-sampled image; the bicubic method uses the 16 nearest neighbors (expect best result), while bilinear uses 4 nearest neighbors (expect 2nd best result), and nearest neighbor only uses single closest neighbor (worst result).

3. What parts of the image seem to work well using these digital zooming methods? What parts of the image don't? Why?

Digital zooming seems to do well at preserving the general contents of the image; in particular parts of the image with low frequency signal. For example, Lena's general facial shape, body, her hat shape, and objects in the background are well preserved. This makes sense since the interpolation methods are able to keep high level spatial information during the zooming. However, high frequency signals such as the details in Lena's eye lashes, the feathers on Lena's hat, or other details on Lena's hat are not well preserved by digital zooming as the fine details get lost by interpolations and aggregations of information. A similar observation can be made by the Cameraman, where bilinear and bicubic methods are able to keep the low frequency spatial information very well, but the high frequency details of the man's face and camera are lost.

4. Compare the zooming results between Lena and Cameraman. Which image results in higher PSNR? Which image looks better when restored to the original resolution using digital zooming methods? Why?

In general, the Lena image had higher PSNR across all zooming methods. Furthermore, the Lena image visually looks better compared to the Cameraman image when restored to the original resolution; the Lena image is relatively clear, however, for the Cameraman we notice that the zoomed results are either blurry or pixelated. The main reason is that the Lena image started at a resolution that was 2x that of the camera man. Thus, the down-scaled images of Lena were able to preserve 2x as much information/signal (since they were both scaled by the same factor), which produced better results after using digital zooming. More signal implies better results. This statement was tested by reducing the image of Lena to the same resolution as the Cameraman (64x64; reduction by a factor of 8). The results show that Lena now becomes blurry or pixelated - as expected from the Cameraman.

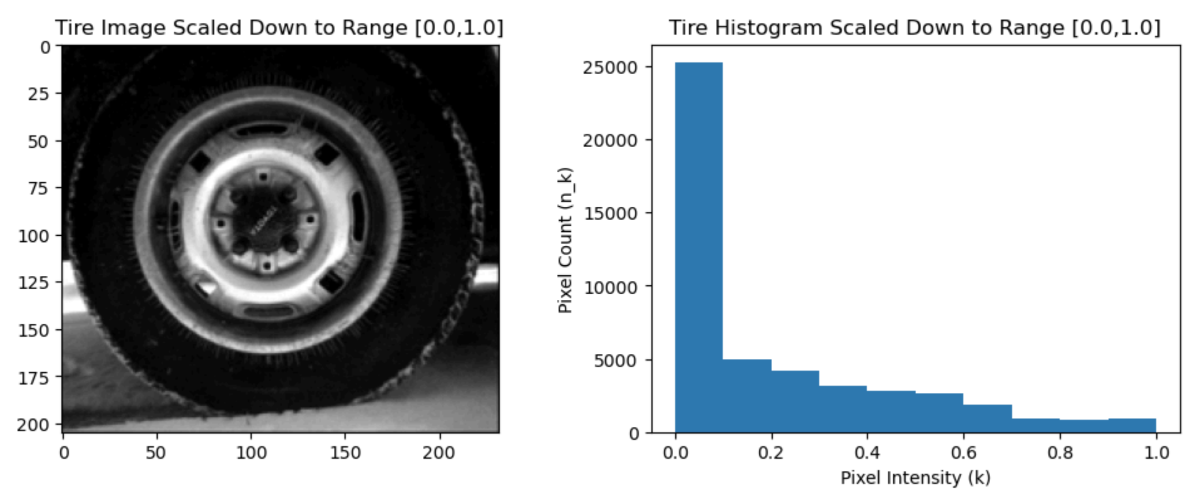


5. What does the PSNR tell you about each of the methods? Does it reflect what is observed visually?

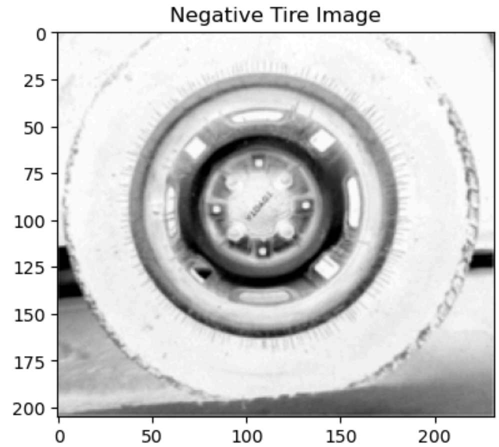
The PSNR tells us, in general, that the performance of each method is relatively similar; the values between different methods are much closer than the gap in PSNR between Lena and the Cameraman. Visually, there isn't a huge difference between the different methods (especially for Lena), which is shown by the close PSNR values. However, the PSNR does not provide a good metric in absolute terms for the quality of an image that is interpretable for humans (it is only useful in relative terms, compared to other PSNR values); the PSNR of an image cannot be used directly to tell much about the image.

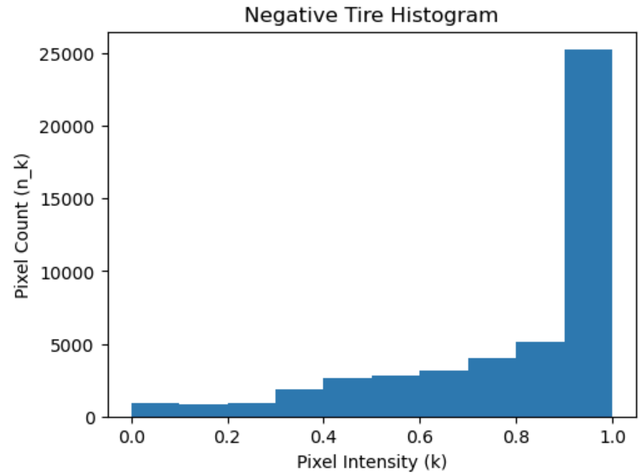
Point Operations For Image Enhancement

The second section of this lab focused on applying point operations for image enhancement. First, the image of the tire was scaled down to [0.0,1.0] by dividing the original image by 255.0. For the histogram, the tire array was first flattened using “tire.flatten()”.

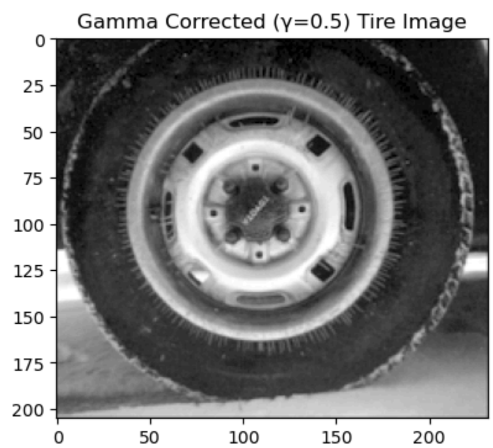


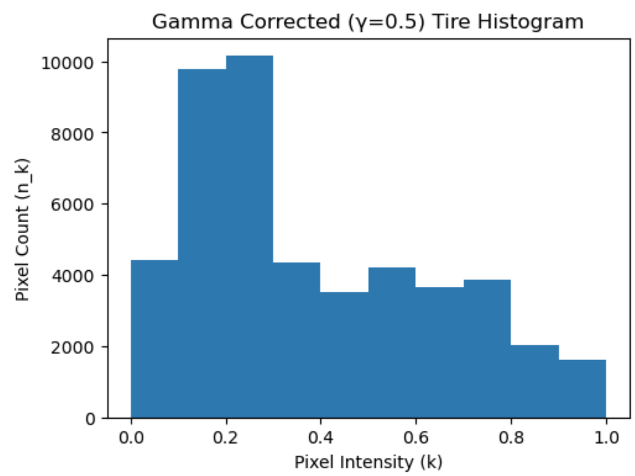
Afterwards, the image negative transform was executed on the tire image, producing the following image and histogram.

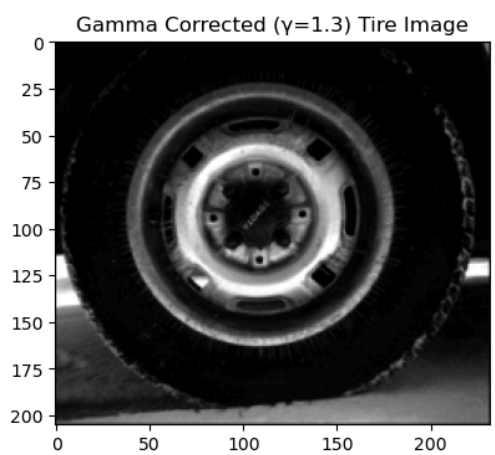


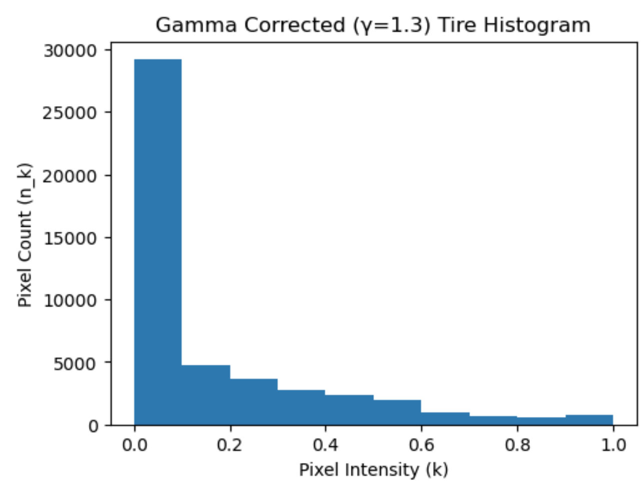


Next, two power-law transformations (gamma correction) were applied on the tire image, one with an exponent term γ = 0.5 and the other with γ = 1.3. The power-law transformation being $I_{out} = I_{in}^{\gamma}$, which resulted in the following outputs.

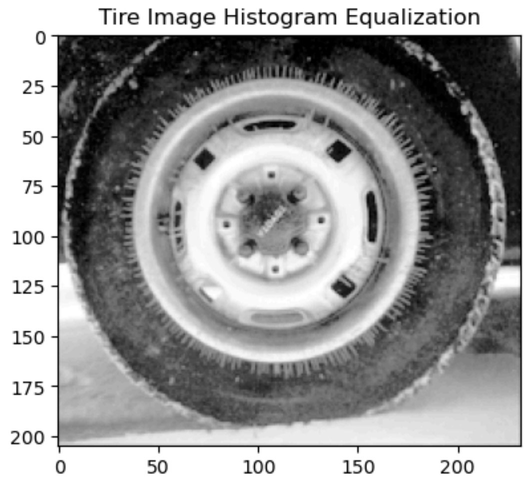


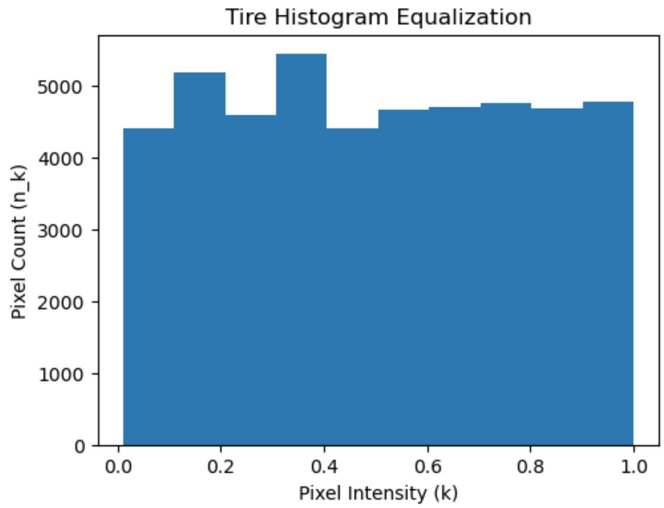






Lastly, histogram equalization was performed on the tire image using the skimage.exposure.equalize_hist function.





Point Operations For Image Enhancement: Questions

Tire image scaled down to [0.0,1.0]

6. Explain what the histogram of an image represents. Why is it useful?

A histogram of a single channel image (gray-scale) image represents the distribution of gray levels in an image. Specifically, the histogram represents the frequency of pixels, n(k), at each intensity level, k and can be used to infer the probability of occurrence for a certain pixel intensity. It's useful as it helps understand the distribution of pixel intensities numerically and enables point processing for image enhancement.

7. Describe how the histogram looks like in the context of intensity distribution. What does the histogram say about the image?

The intensity distribution is right-skewed. Based on the mean inferred from the histogram, as well as the spread of the data in the histogram, we can tell that the image is generally dark, which is also noticeable while observing the image.

Tire image negative transform

8. Describe how the histogram looks like in the context of intensity distribution. How does it differ from the histogram of the original image? Why?

The histogram is the opposite of the original image, meaning it is now left-skewed. This indicates that there is a larger amount of bright pixels in the image. This differs from the original as the negative is produced by subtracting each pixel from the maximum intensity value, 1 - n(k), where n(k) is the pixel count of each intensity value (k). Essentially, this is flipping each intensity value. Hence, the darkest pixels will become the brightest and vice-versa.

Tire image power-law transformations

9. Describe the appearance of the transformed images. Why do they appear this way?

For γ = 0.5, the image becomes brighter as the power-law transform raises the input intensities to a power less than 1, amplifying lower intensity values. As well, γ = 0.5 is essentially a square root transformation, indicating that the low pixel intensities are stretched (slope > 1), while the high pixel intensities are compressed (slope < 1). Since this image is mainly a dark image, stretching the dark region provides strong enhancement qualities. Thus, the final image is much lighter in general and has good contrast.

For γ = 1.3, it is the opposite scenario, as all pixel values have been made darker by the transformation, causing a loss of brightness, especially in the mid-tones and lighter areas. In this case, this transformation is closer to a quadratic transformation (we can infer the same properties as a quadratic transformation), so the low pixel intensities are compressed (slope < 1) while the high pixel intensities are stretched (slope > 1). Compressing the dark region ends up making the image look even darker.

10. Describe how each of the histograms look like in the context of intensity distribution. Why do they look like this? What does each histogram say about each transformed image?

For γ = 0.5, the histogram is more evenly distributed because the gamma correction makes the image brighter overall. Thus, this transformation has a mean that is much higher than the original image, as gamma values less than 1 transform the lower intensity values into higher intensities, creating a more uniform distribution of brightness across the image. Furthermore, the variance in the histogram is also much higher as a result of the non-linear gamma curve which spreads intensity values more (and can be seen in the histogram by a larger spread). It also indicates that the image has more contrast, which is evident.

For γ = 1.3, the histogram generally did not change in terms of distribution. The distribution looks similar thus no contrast change was observed. However, the values for lower intensity pixels did increase, showing a slight reduction in the mean value, which was observed by a slightly darker looking image.

11. Compared with the original image, which of the transforms should you use to enhance the image? Why?

The transform with γ = 0.5 should be used to enhance the image as the larger variance means higher contrast, and as it brightens the overall image, it can reveal hidden features and improve visibility. Thus, the higher contrast is desirable for image enhancement and brings out more details in the image which were not observed in the original image. On the other hand, the γ = 1.3 transform darkens the image, which could obscure those important details, making it less optimal for enhancement. Furthermore, the original image is right skewed, which points to using a technique which stretches the lower intensities while compressing the higher intensities (which is precisely what γ = 0.5 does).

Tire image histogram equalization

12. Describe the appearance of the equalized image.

The equalized image is much lighter (higher pixel intensity values) compared to prior tire images. Many details about the tire have been brought out, but so has noise in the image.

Additionally, the overall contrast in the image has increased as the tire and the background are more distinct. Overall, the originally low contrast image has been enhanced. The range of pixel intensities is equalized which produces a detailed image by showing more intensities.

13. Describe how the histogram looks like in the context of intensity distribution. Why does it look like this? What does each histogram say about each equalized image?

The histogram of intensity is close to following a uniform distribution, which makes sense as it is what histogram equalization is trying to achieve. Compared to the original image, the uniform distribution has a mean that is much higher (the image is lighter in general) and has a lot of contrast due to the very high variance of the transformed distribution. This distribution was achieved by stretching the dark pixel range (where most pixel intensities in the image exist), and compressing the light pixel values. The cumulative distribution function was used to make this transformation possible, and bring out all pixel intensities in the image to maximize contrast.

Conclusion

This lab examined 3 basic aspects of image processing: image quality measures, digital zooming, and point processing for image enhancement. For image quality measures, PSNR was implemented and used to compare the results of digital zooming. It was determined that out of the digital zooming techniques, nearest neighbor interpolation performed the worst, followed by bilinear, and then bicubic interpolation - however all the methods yielded similar PSNR values. Visually, the techniques looked very similar for zooming on Lena, and only the nearest neighbor method looked pixelated for Cameraman. Finally, different point processing techniques were used along with histograms to determine how transformations would shift the intensity distributions. The tire image was used which showed a generally dark image from the histogram, and then the negative transform was applied to turn it into a light image. Next, two power-law transformations with opposing effects were applied to the originally dark tire image; the transformation with gamma = 0.5 had a slope > 1 for dark pixel values, thus it stretched that region in the output image, making the resultant mean pixel intensity lighter and with higher variance. When gamma = 1.3 was applied, the opposite effect was observed as the slope < 1 occurred for low intensity values, and further compressed the dark image intensities. Finally, histogram equalization was used to obtain the “best” image creating a uniform intensity distribution. The histogram equalized image provided the most detail and contrast as a result of the high variance in the distribution, but also brought out some of the noise in the image.

Appendix: Code (Python file is also being attached in dropbox)

```
from skimage.color import rgb2gray
from skimage.io import imread
import skimage.transform as tf
import numpy as np
import matplotlib.pyplot as plt
import skimage.exposure

method_order_to_name = {
    0: "Nearest Neighbor",
    1: "Bilinear",
    3: "Bicubic"
}

def mse(f, g):
    """
    Mean squared error between 2 images
    """
    m, n = f.shape
    return np.sum(np.square(f - g)) * (1 / (m*n))

def PSNR(f, g):
    """
    Peak signal to noise ratio between 2 images
    """
    f = f.astype(np.float64)
    g = g.astype(np.float64)

    return 10 * np.log10(255.0**2 / mse(f, g))

def reduce_resolution(image, title, factor):
    """
      reduce  each  image  resolution  by  a  factor  of  4  (horizontally  and
vertically)
    """
    new_shape = image.shape[0] // factor, image.shape[1] // factor
    print(f"Old shape: {image.shape}, New shape: {new_shape}")
    image = tf.resize(image, new_shape, order=1)
    plt.imshow(image)
```

```
    plt.title(title)
    plt.show()
    print(image.shape)
    return image

def digital_zoom(image, factor, order, ax, title):
    """
    Perform digital zoom where order NN = 0, bilinear = 1, bicubic = 3
    """
    new_shape = image.shape[0] * factor, image.shape[1] * factor
    print(f"Old shape: {image.shape}, New shape: {new_shape}")
    image = tf.resize(image, new_shape, order=order)
    ax.imshow(image)
    ax.title.set_text(title)
    return image

def zoom_image_different_methods(image, factor):
    images = []
    fig, axs = plt.subplots(1, 3, figsize=(15, 6))
    for i in range(3):
        order = i
        if i == 2:
            order = 3
        zoomed_image = digital_zoom(image, factor, order, axs[i], f"Digital Zooming with {method_order_to_name[order]} Interpolation")
        images.append(zoomed_image)
    return images

def tire_plotting(tire_image, image_title, histogram_title):
    """
    Plotting image and histogram of tire
    """
    fig, axs = plt.subplots(1, 2, figsize=(12, 4))
    tire_flattened = tire_image.flatten()
    axs[0].imshow(tire_image)
    axs[1].hist(tire_flattened)
    axs[0].set_title(image_title)
    axs[1].set_title(histogram_title)
    axs[1].set_xlabel("Pixel Intensity (k)")
    axs[1].set_ylabel("Pixel Count (n_k)")
```

```
    plt.show()

lena= rgb2gray(imread('lena.tiff')) *255
cameraman = imread('cameraman.tif').astype(np.float64)
tire = imread('tire.tif').astype(np.float64) / 255.0

# plot original images
fig, axs = plt.subplots(1, 2, figsize=(8, 4))
for i, image in enumerate([lena, cameraman]):
    axs[i].imshow(image, cmap='gray')

fig.suptitle('Lena and Cameraman Original Images')
plt.show()
plt.gray()

# reduce resolution and plot
lena_reduced = reduce_resolution(lena, "Lena Reduced Resolution", 4)
cameraman_reduced = reduce_resolution(cameraman, "Cameraman Reduced
Resolution", 4)

# zoom images and plot
lena_images = zoom_image_different_methods(lena_reduced, 4)
plt.show()
cameraman_images = zoom_image_different_methods(cameraman_reduced, 4)
plt.show()

# print the PSNR values for Lena and the cameraman
for i, l_image in enumerate(lena_images):
    order = i
    if i == 2:
        order = 3
    upsample_method = method_order_to_name[order]
        print(f"PSNR between Lena image and up-sampled image using
{upsample_method.lower()} interpolation: {PSNR(lena, l_image)}")


for i, c_image in enumerate(cameraman_images):
    order = i
    if i == 2:
        order = 3
```

```
    upsample_method = method_order_to_name[order]
       print(f"PSNR  between  Cameraman  image  and  up-sampled  image  using
{upsample_method.lower()} interpolation: {PSNR(cameraman, c_image)}")

# plot original tire image and histogram
tire_plotting(tire,  "Tire  Image  Scaled  Down  to  Range  [0.0,1.0]",  "Tire
Histogram Scaled Down to Range [0.0,1.0]")

# calculate negative
tire_negative = 1.0 - tire
tire_plotting(tire_negative,   "Negative   Tire   Image",   "Negative   Tire
Histogram")

# calculate power-law transformation for gamma 0.5
I_out1 = tire ** 0.5
tire_plotting(I_out1,  "Gamma  Corrected  (γ=0.5)  Tire  Image",  "Gamma
Corrected (γ=0.5) Tire Histogram")

# calculate power-law transformation for gamma 1.3
I_out2 = tire ** 1.3
tire_plotting(I_out2,  "Gamma  Corrected  (γ=1.3)  Tire  Image",  "Gamma
Corrected (γ=1.3) Tire Histogram")

# apply histogram equalization
hist_equalizer = skimage.exposure.equalize_hist(tire)
tire_plotting(hist_equalizer,  "Tire  Image  Histogram  Equalization",  "Tire
Histogram Equalization")
```