CMPEN/EE455: Digital Image Processing I

Computer Project # 4:

Image Enhancement: Histogram Modification

Georges Junior Naddaf, Zian Wang, Ritvik Muralidharan

Date: 11/1/2017

A. Objectives

This project's main purpose is to:

- Introduce CDF and histogram formation in Matlab
- Introduce gamma correction methods in Matlab
- Introduce contrast stretching methods in Matlab
- Introduce histogram/image equalization in Matlab

B. Methods

We have created 3 files in our directory. The first is Project4.m that produces the CDF and histogram of the original image. It also performs histogram/image equalization needed for question 3. The second file is Gamma.m. This file produces the gamma corrected images with gamma = 0.2 and gamma = 5. It also creates the CDF and histograms for these files. The third file is HistEq.m and it performs the histogram/image equalization of the original image and produces the CDF and histogram of that image.

To run the files, simply hit run. If questions 1 and 3 are being evaluated, run Project4.m. To evaluate question 2, run Gamma.m. To evaluate question 4, run HistEq.m. The files save modified images and show the figures needed.

Part 1:

For this question, we start by inputting the truck image in Project4.m. We go through the image and count the occurrence of each pixel intensity to be able to draw out the histogram. We then populate an array t with the cumulative sums. When the loop is finished, we plot the array, we use stem for aesthetic purposes and we label it. For the CDF, we sum the values in the t array using the sum() function. We then calculate the ratio of each element and store in n_sum. We then loop over the t array and calculate the CDF at each point by summing the ratios up until the iterator i.

Part 2:

In this question, we start out by setting the gamma value as 5 and 0.2 in Gamma.m. We then perform gamma correction using the power law. We set each pixel's intensity on a scale from 0 to 1. Then, we apply the gamma power and multiply the result by 255 back to the original scale. That way, we will get new pixel intensity values. This method produces two images corresponding to the two gammas. Then, we use the method in part 1 to produce the corresponding CDF and histogram for each image.

Part 3:

In this part, we start with the CDF already established for part 1 in Project4.m. We go through the CDF and set pixels with intensity less than 0.1 to 0 and more than 0.9 to 1. While iterating through the CDF, we use two variables L1 and L2 to keep track of the X-axis values at which a pixel has an intensity between 0.1 and 0.9. We are then asked to linearly stretch the range between 10% and 90%. We first stretch the CDF for illustrative purposes. To do that, we go through the CDF and scale the 0% and 100% values. The intensities that reside in the middle of the range are linearly stretched using a simple equation. We then proceed to linearly stretch the image. To do that, we set all intensities small than L1 to 0 and bigger than L2 to 255. In the case where the intensity falls in the range, we perform a simple equation similar to the one performed to the CDF. This results in the desired stretched image. We then produce the CDF and histogram for that image.

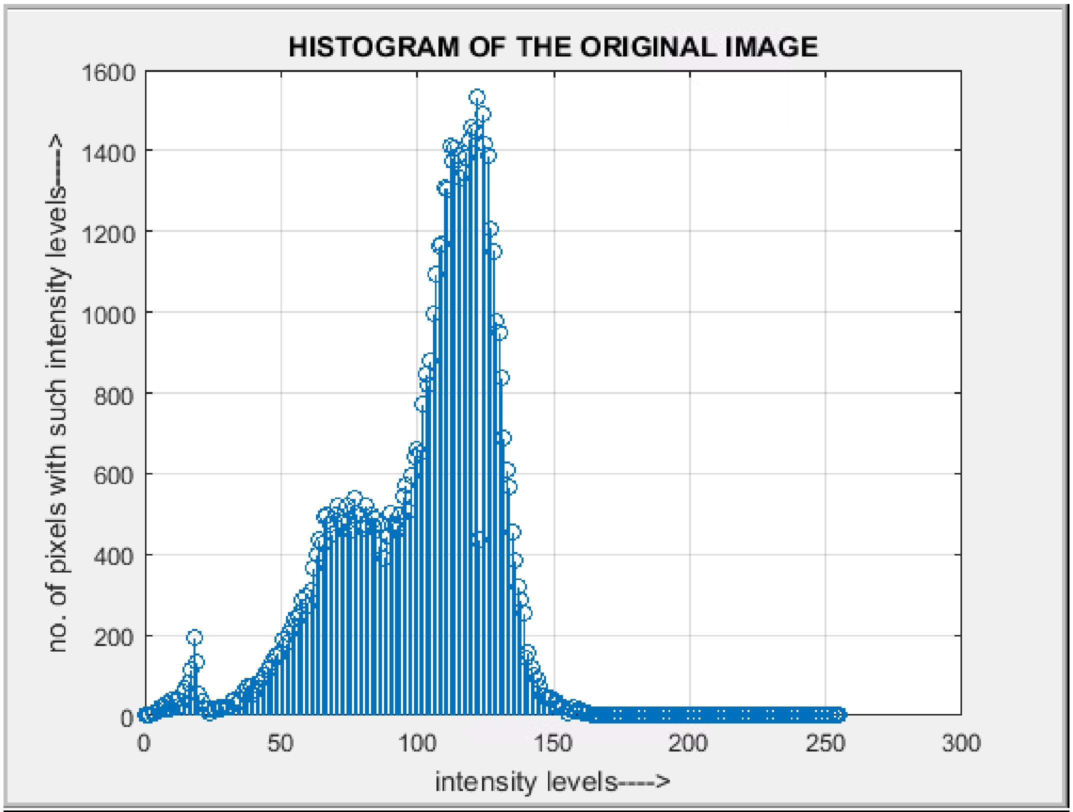
Part 4:

For the last part, we go through and create a histogram array followed by a CDF similar to the previous parts. Our implementation for histogram and CDF production in this part may differ from the other parts since we were testing some assumptions while writing the code. However, the code performs the same operations as the other parts and is logically equivalent. We then iterate over the CDF and fill an array d with rounded ratios. We finally create an equalized image by equalizing the histogram from the last loop and storing the result in a new image array. We then used the same methods used in other parts to produce the histogram and CDF.

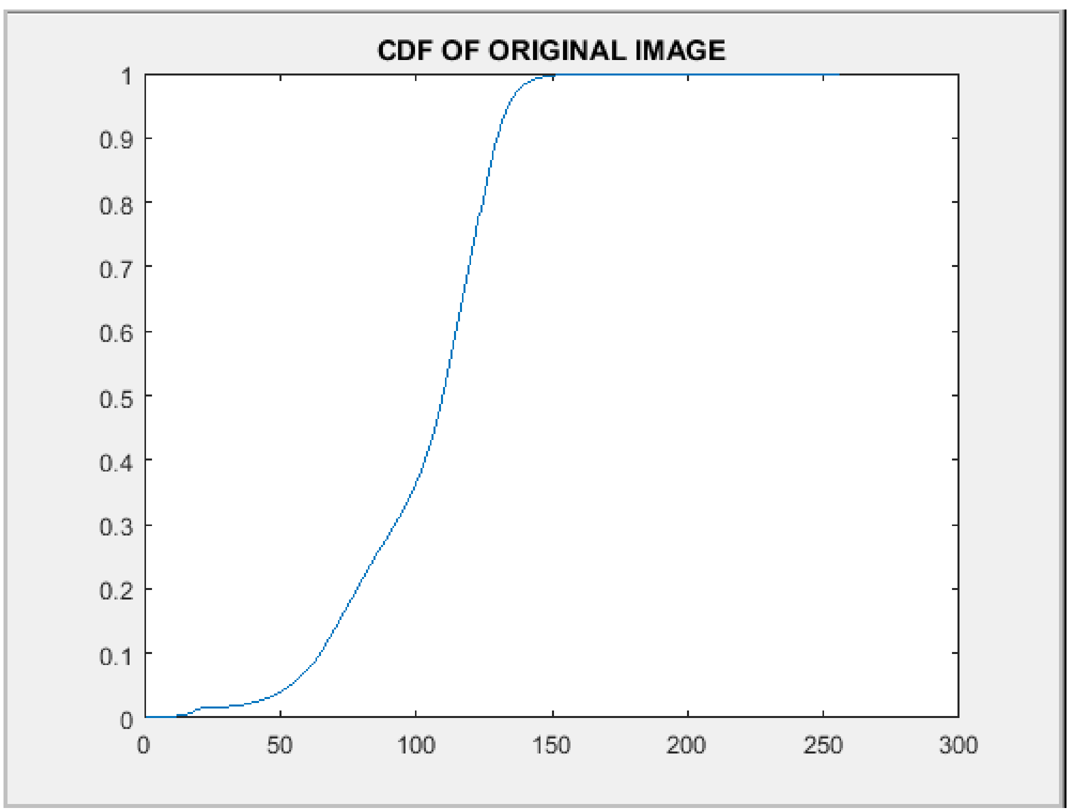
C. Results

Part 1:

Figure 1.1: Original Image



<u>Figure 1.2:</u> Histogram of original image



<u>Figure 1.3:</u> CDF of original image

Part 2:

Figure 2.1: gamma = 0.2 image

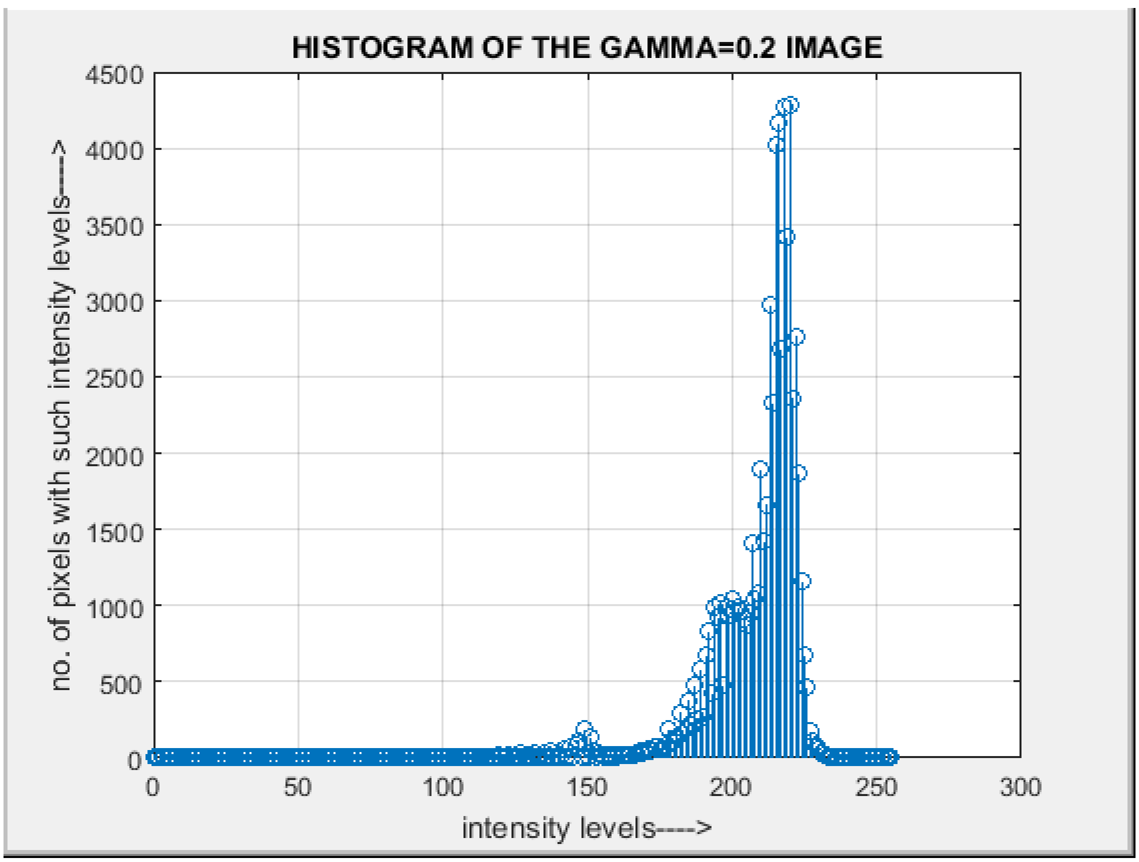


Figure 2.2: Histogram of gamma = 0.2 image

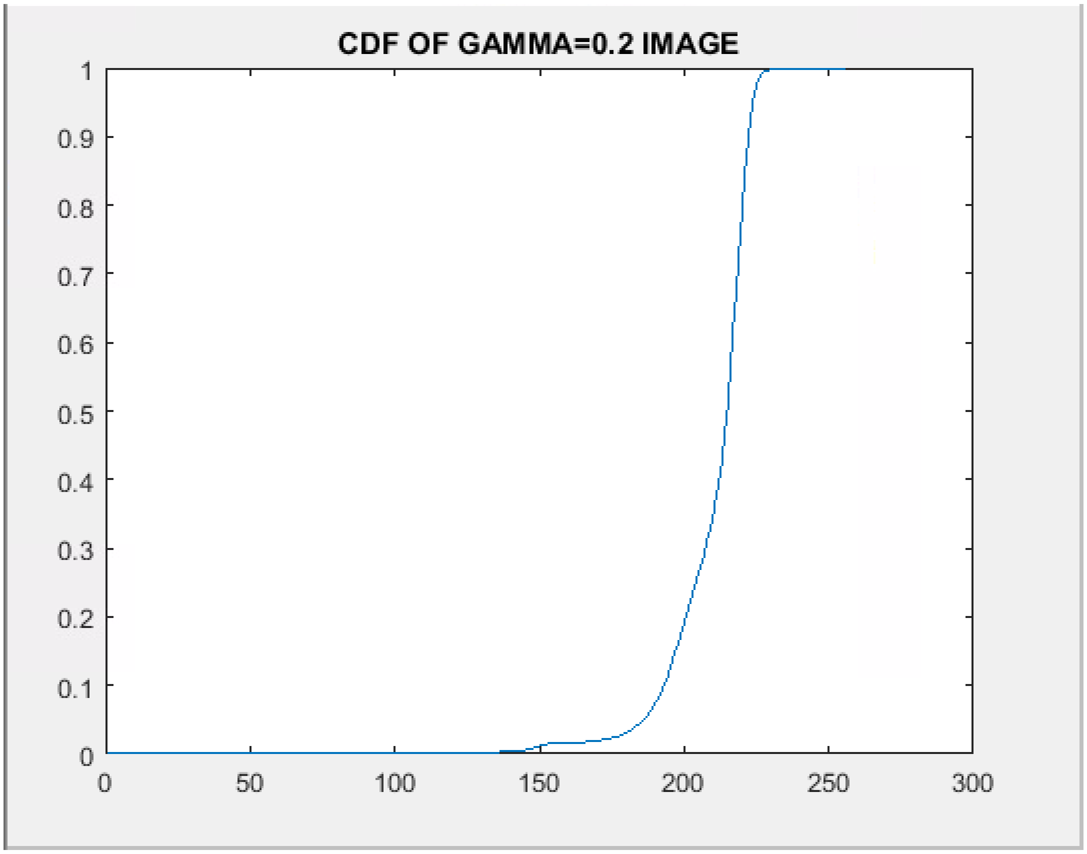


Figure 2.3: CDF of gamma = 0.2 image

Figure 2.4: gamma = 5.0 image

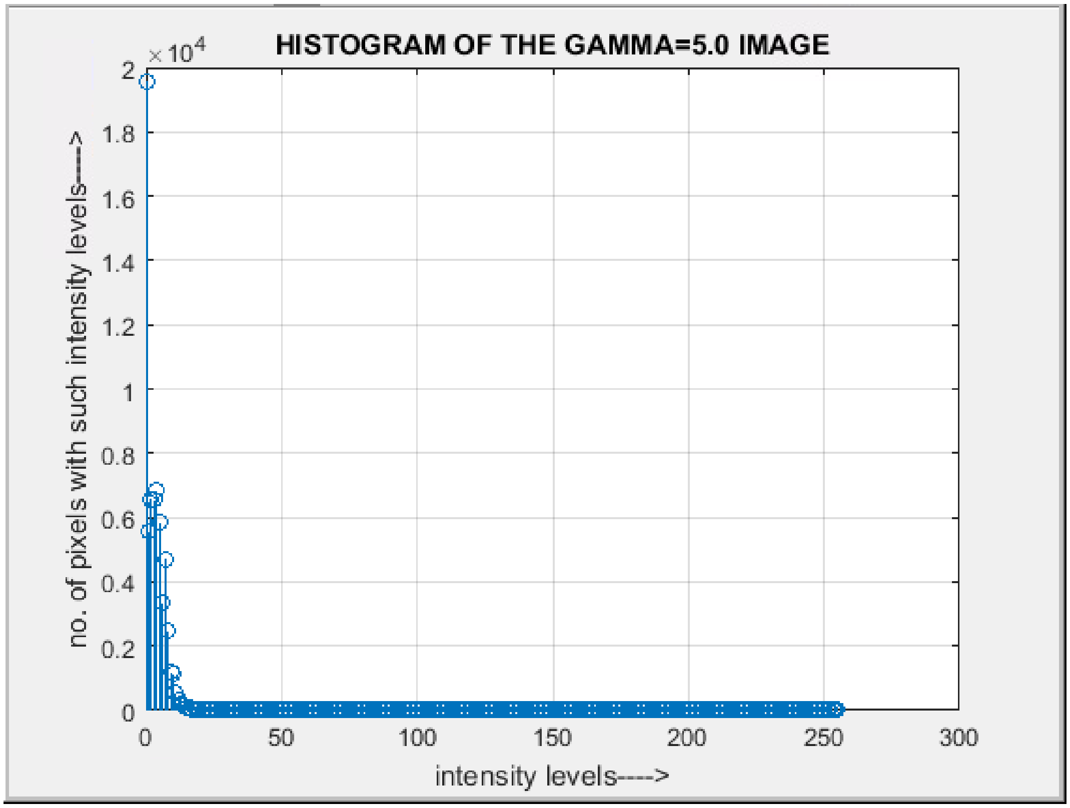


Figure 2.5: Histogram of gamma = 5.0 image

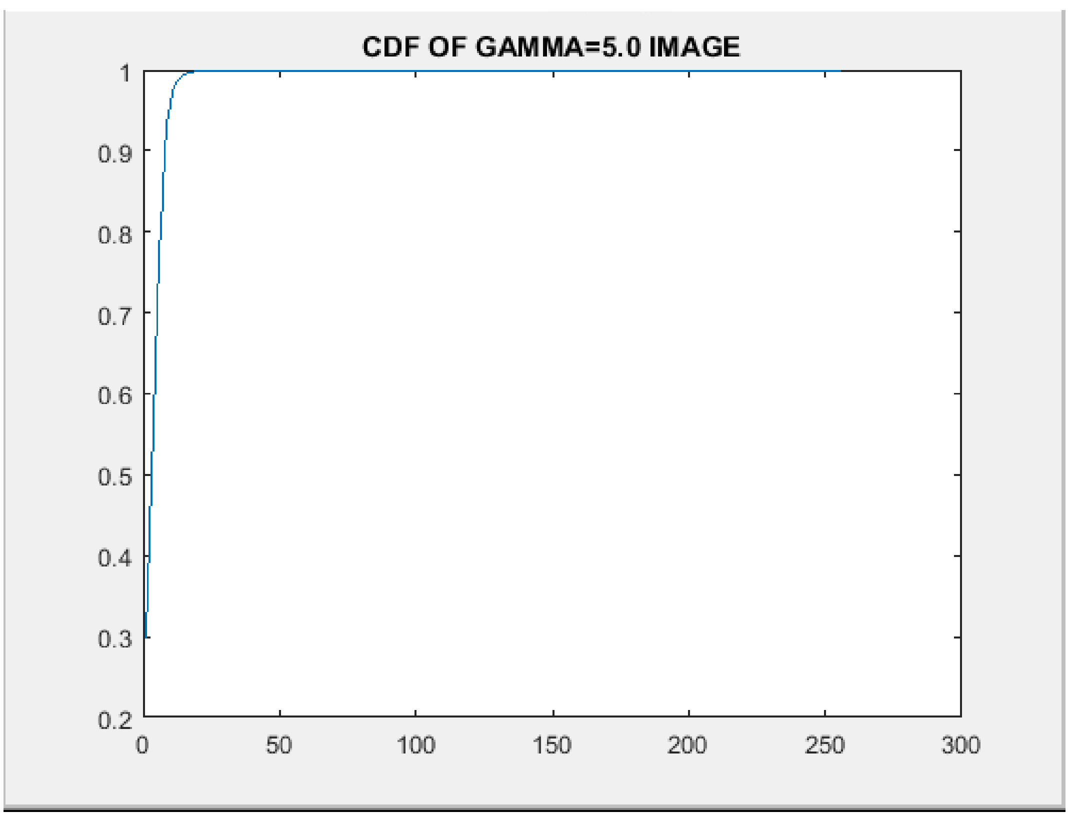


Figure 2.6: CDF of gamma = 5.0 image

Part 3:

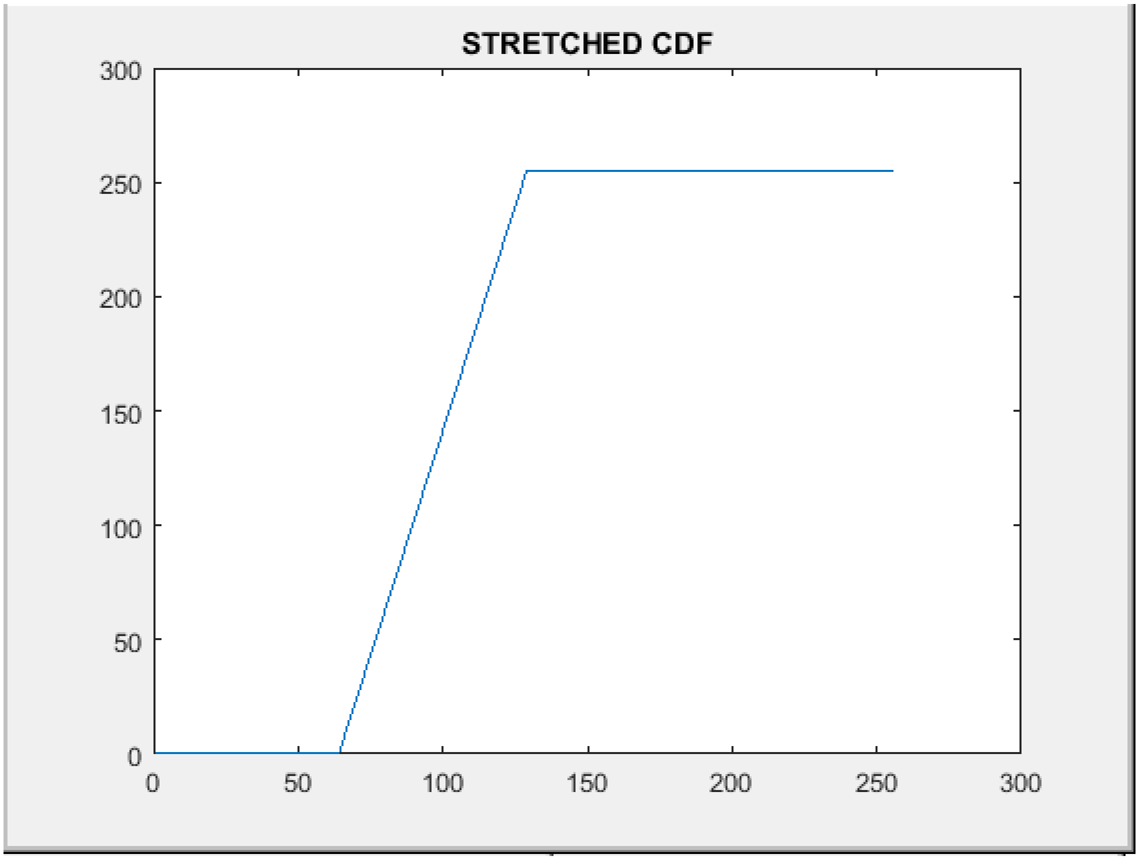
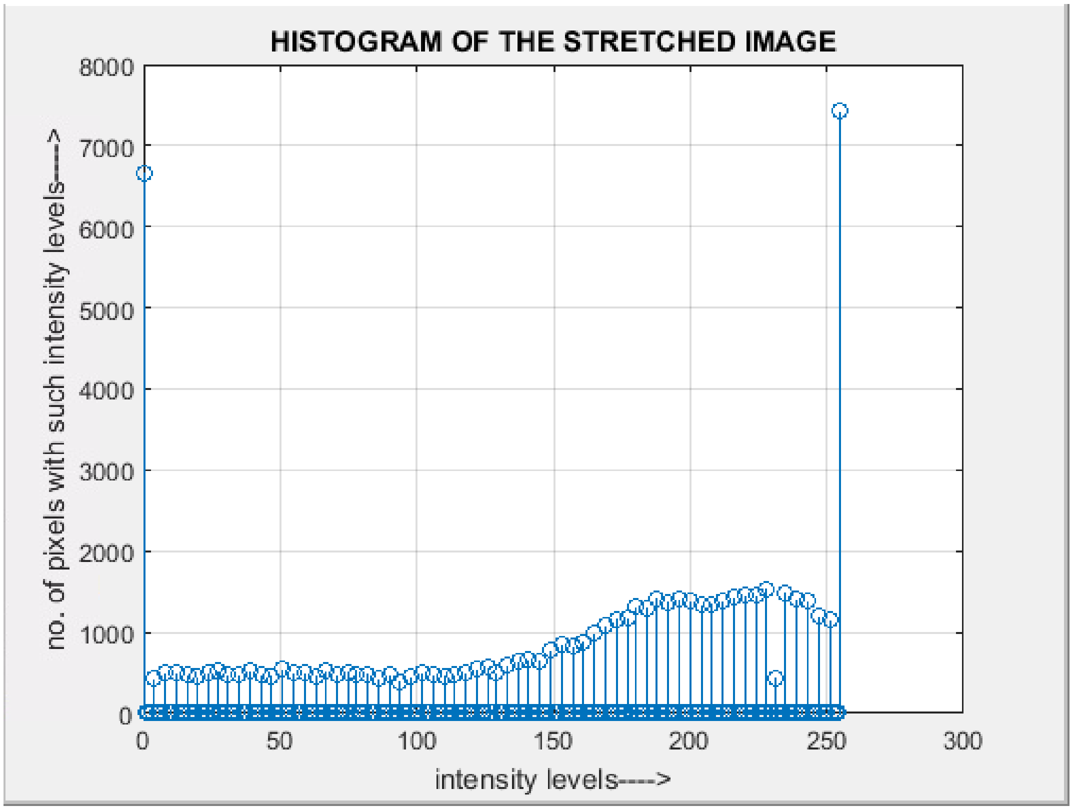
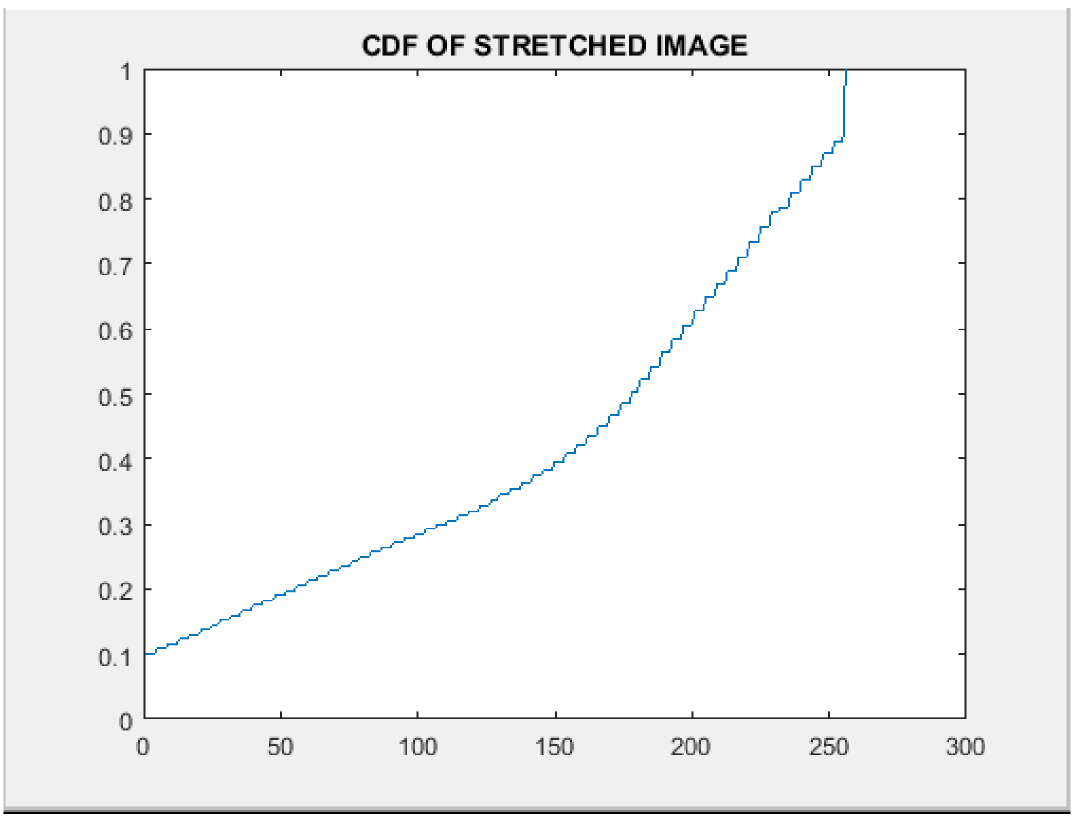


Figure 3.1: Stretched CDF

Figure 3.2: Stretched image



<u>Figure 3.3:</u> Histogram of stretched image



<u>Figure 3.4:</u> CDF of stretched image

Part 4:

Figure 4.1: Equalized image

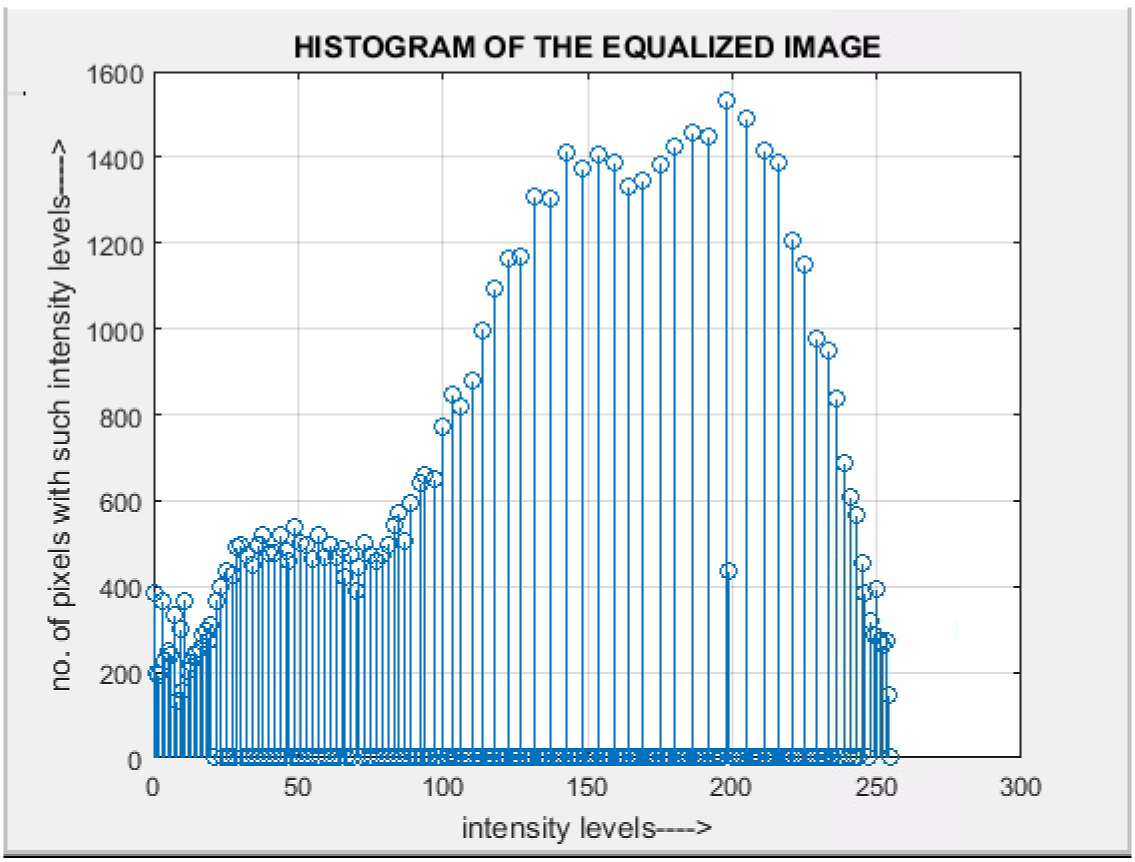


Figure 4.2: Histogram of equalized image

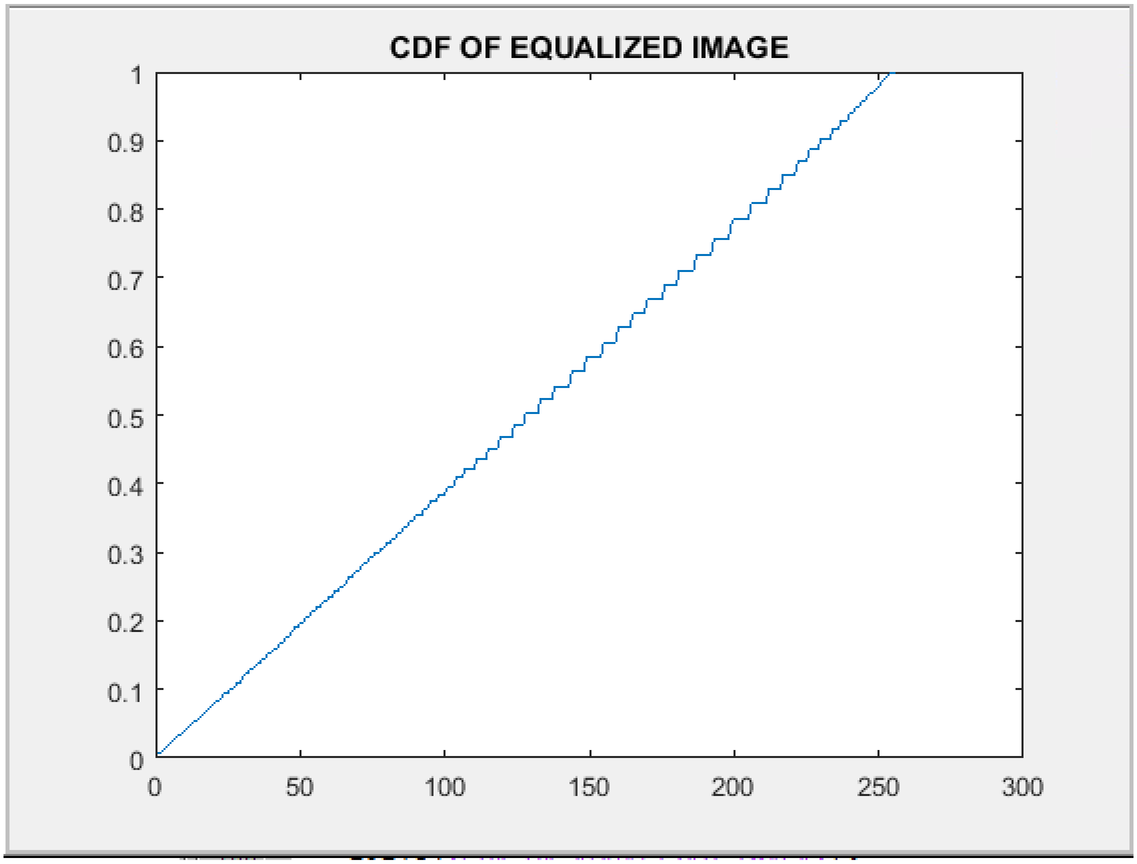


Figure 4.3: CDF of equalized image

Part 2:

Note that when gamma = 5.0, the image may appear black. However, in the monitor at the CSE lab, the image was visible but very dark and that is expected since gamma = 5.0 is a big value.

Part 3:

S = T(r) expression.

L1 = 64 and L2 = 129 by using disp() function in code.

T(r) = {

0 if 0 <= r < 64,

3.923 * r – 251.077 if 64 <= r <= 129,

255 if 129 < r <= 255

}

Observations/Part 5:

Part 1:

In part 1, we observe that the histogram of the original image is not balanced and the CDF does not uniformly span the intensity range. Both the histogram and CDF lean to the left. This indicates that there is an under-exposure, meaning the image is darker than it should be.

Part 2:

After performing gamma correction, we notice various observations. First and foremost, gamma correction changes the pixel intensity distribution. When a gamma of 0.2 is chosen, the image becomes overexposed and we obtain a very “bright” image. When using a gamma of 5.0, the image becomes extremely under-exposed. And so, both these gamma corrections do not create a better image. The changes are visible in the histograms of the images, as the peaks shift left or right depending on the case. The same can be said about the CDF as it shifts from left to right. A better image may be produced by choosing a gamma value in between 0.2 and 5 such as 0.9, which may produce a properly exposed image.

Part 3:

In this part, we perform linear contrast stretching, and this technique has some interesting effects. We can see that this stretching technique balances out the histogram and CDF of the image (with jumps at 0 and 255) and produces a much clearer image compared to the original. However, this technique, being slightly primitive, also produces an inevitable, unwanted effect. We notice that the relatively black colors in the image become pitch black and the relatively white colors in the image become pure white (by design). Thus, this technique provides a better image but with a drawback of less precise pixel intensities.

Part 4:

In this last part, we perform histogram equalization. We notice that the histogram and CDF are very balanced, indicating a properly exposed image (no jumps). Also, the image is as clear as when using linear contrast stretching but without the negative effect of having pure black/white colors. And so, we can imply that histogram equalization is the best image enhancement technique compared to gamma correction and contrast stretching.

D. Conclusions

The project addresses peculiar questions concerning image enhancement by histogram modification. The key points to take out are:

- The histogram and CDF are valuable clues that indicate the quality of image exposure.
- Using gamma correction is tricky as finding the right gamma value may require multiple trial and error attempts.
- If the gamma is too low, the image becomes over-exposed and if it is too high, the image becomes under-exposed.
- Linear stretching is a primitive method that can lead to proper exposure in an image.
- Linear stretching may harm pixel intensity accuracy.
- Histogram equalization is a robust and powerful method for creating a properly exposed image.

In conclusion, these image enhancement techniques are extremely useful in real life. For example, applying histogram equalization on an under-exposed image of an unidentified individual can help in producing a clearer image, which is very beneficial for security purposes. These techniques can also be applied on shots taken at night or in the bright of day allowing for clearer, more balanced images. Such applications can be used in modern electronic items such as mobile phones and cameras.