Whit Boland - jwb10028
Professor Gerig
CS6643 Computer Vision
22 February, 2024

Lab 1 Report

A1

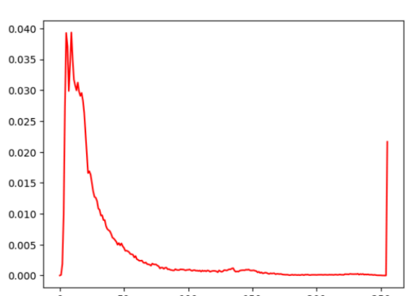
(a) Include a brief introduction and description of how histogram equalization works.

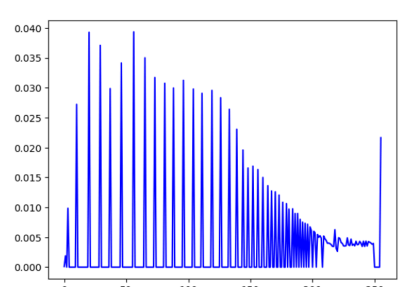
Histogram equalization is a process that includes the calculation of a source image's histogram values. Calculating the histogram of an image allows us to show a distribution of pixels based on their pixel intensity values. In the case of this report, 8-bit grayscale images are used as source images, and binned into an intensity value range of 256 [0-255]. Once the histogram of a source image is calculated the set of histogram values are used to calculate the cumulative distribution function. The cumulative distribution function, or CDF, is used to map the original source set of pixel values to a new target set of pixel intensity values that represent a cumulative probability of pixel intensities. Once the CDF is calculated, a normalization is performed to fit the original 8-bit range of [0-255], this helps in maximizing the contrast in the equalized image. After normalization each source pixel is mapped to its new normalized CDF pixel value, this results in the target equalized image.

(b) Show people.png before and after histogram equalization, and the corresponding histograms (PDFs).

people.png

people_processed.png

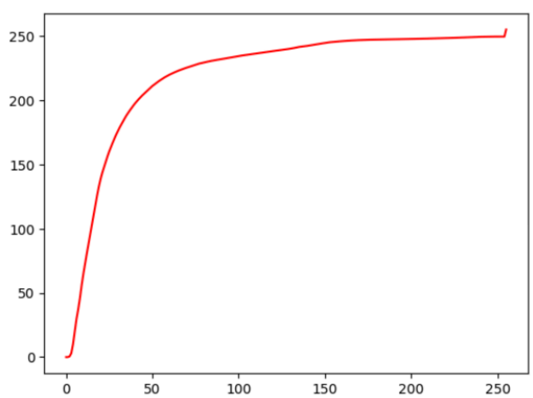




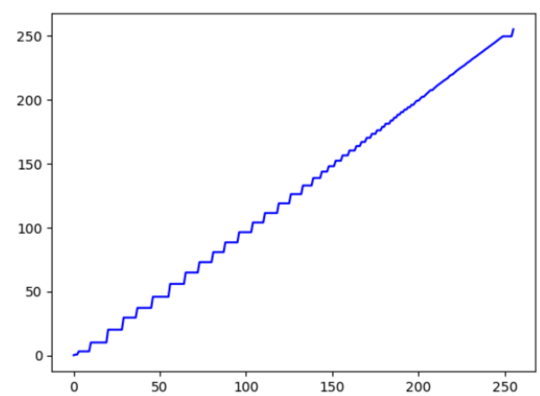
(c) Discuss how the image and histogram have changed, and connect it back to your description in 1.

Looking at the above source histogram, and the equalized histogram, it is fairly easy to notice that the target equalized histogram has been mapped to represent the cumulative probability value of the source set of pixels. This cumulative probability is normalized to fit the 8-bit range of [0-255], and then mapped to each corresponding pixel in order to return the target image. The target histogram above clearly depicts a normalized distribution from [0-255], which compared to the source image uses the maximum amount of the intensity range to map pixel values. Using the binning system pixels are effectively just redistributed across the whole of available intensity range in a manner that results in a target uniform distribution. As we can see from the above source image histogram the majority of pixel intensity values are weighted to the darker side of the x-axis. A huge difference with the side by side comparison of the target equalized image that is spread across the whole of the spectrum.

(d) Show the cumulative distribution function before and after histogram equalization in a side-by-side figure. Describe what you see.



pre-equalization cdf



post-equalization cdf

From the source image CDF it can be seen that the pixel intensity values had a very limited range of pixel intensities. Looking at the source image itself it can be seen that these pixel intensity values are likely skewed towards the darker side of the intensity range with some sparse highlights in the image. On the other hand, the post-equalization CDF clearly depicts a uniform distribution of pixel intensity values. This depiction is for the most part the expected result of histogram equalization, a uniform distribution of pixel intensity values normalized in a defined intensity range, in this case [0-255] for an 8-bit image. The best way to describe what is seen from the side by side comparison of cumulative distribution functions is that while the source image uses a very limited amount of the available intensity range, the target equalized image provides a uniform distribution across the whole of the available intensity range.

(e) Reapply the histogram equalization procedure on the corrected image. Show and discuss the results.

people_processed.png

people_processed_x2.png

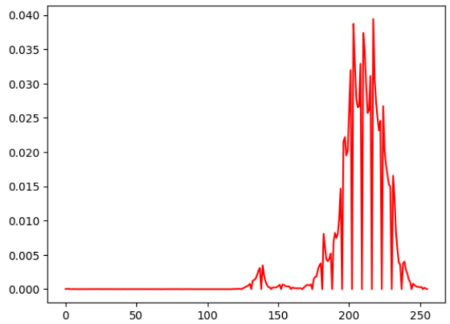
It can be noted from the above results that re-equalizing a source image that has already been equalized will result in a target image that represents the same equalization. When the equalization process has already been performed, the resulting histogram depicts a uniform distribution. When this uniform distribution is re-mapped into the CDF for the second time the resulting distribution will be the same, namely because the cumulative distribution function's job is to provide a uniform distribution of pixel intensity values. Since the input set is already a uniform distribution, at each step each pixel value will be mapped to its equalized value, and no work needs to be done.

(f) Apply histogram equalization to another low contrast image (greyscale). Show and discuss the results.

image.png

image_processed.png

image.png (pre-histogram)



image_processed.png (histogram)

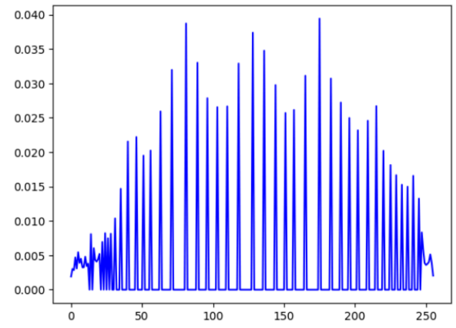
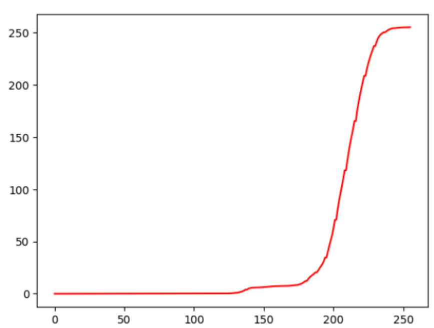
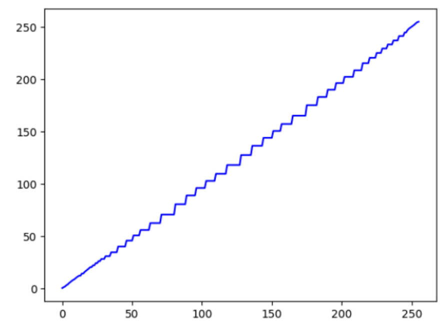


image.png (pre-cdf)



image_processed.png(cdf)

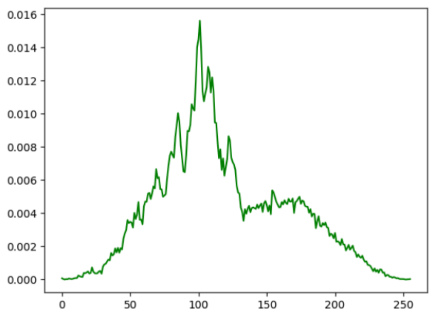


From the above source image it can be seen that for the most part pixel intensity value skews towards higher intensity values (closer to white). The histogram before equalization shows that the majority of pixels have been binned at higher pixel intensity values which confirms the previous assumption. In terms of the cumulative distribution curve it can be noted that equalization of a source image that is unevenly distributed to higher pixel intensity values results in generally the same uniform distribution of the source 'people.png' image which seemed to skew towards lower pixel intensity values. With that being said, it seems that summed up histogram equalization can be defined as a process that brings a source image's cumulative distribution curve into a target uniform state. This uniform distribution ensures that the entire set of pixel's intensity values are "flattened" into an evenly distributed histogram, resulting in our equalized image and linear cdf. Equalization is the result of this application of cdf values, c(I), to a range of histogram values h(I), after application the goal is a linear diagonal curve indicating the target uniform distribution. In terms of the above source image, histogram equalization provided an output with much higher visual contrast. The reflections on the lake after equalization are more defined, the topology of the landscape is much easier to make out. Overall, it seems that histogram equalization in this case would provide useful information regarding the source image and details that can be seen within the source image.

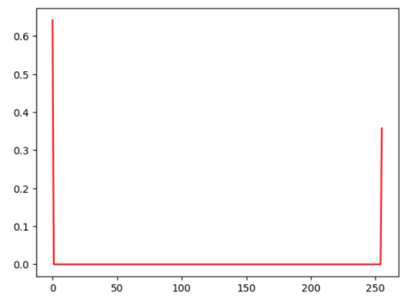
A2

(a) Show the histograms for each image.

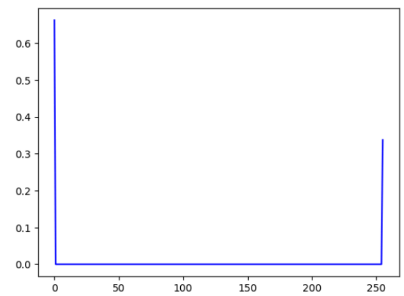
b2_b



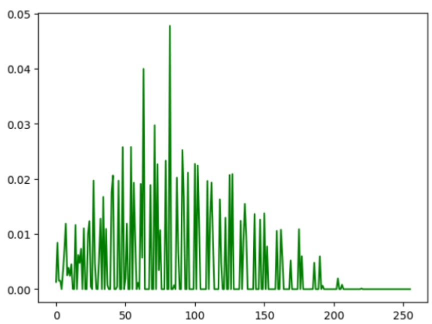
b2_a_manual



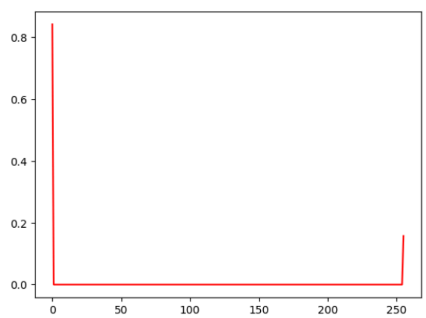
b2_a_otsu



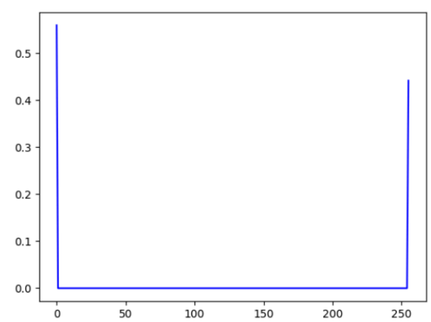
b2_b



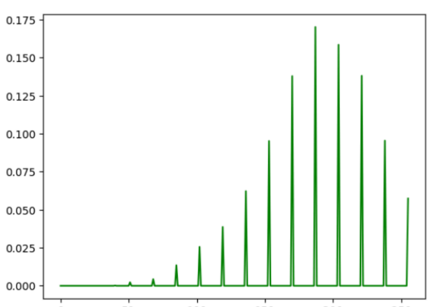
b2_b_manual



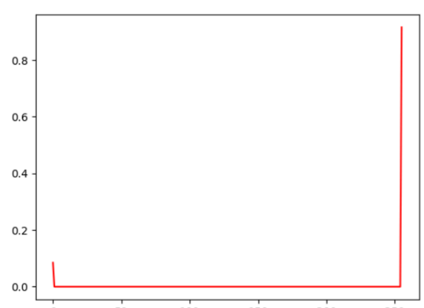
b2_b_otsu



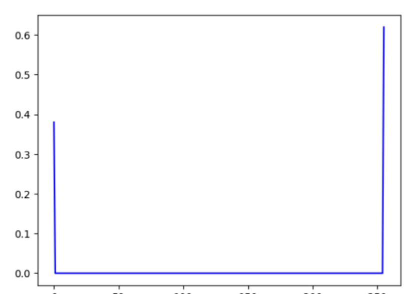
b2_c



b2_c_manual



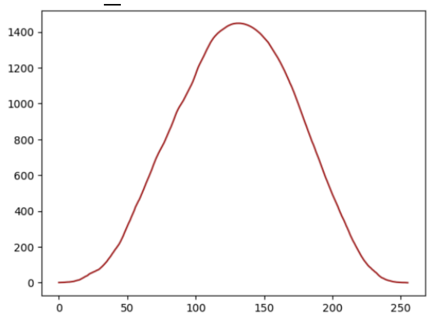
b2_c_otsu



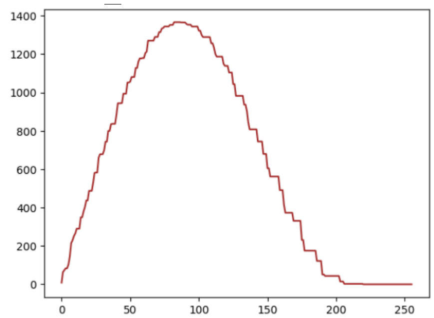
GLOBAL MANUAL THRESHOLD: 127

(b) Generate a plot of the inter-class variance as a function of the chosen threshold (i.e., x-axis with each possible threshold from 0-255, y-axis with the resulting variance)

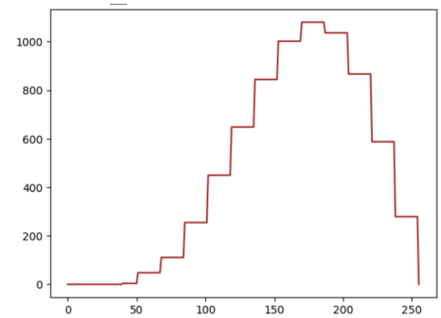
b2_a



b2_b



b2_c



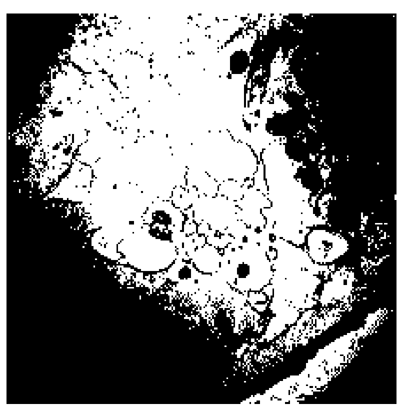
(c) State the inter-class variance of the image upon completion of the algorithm.

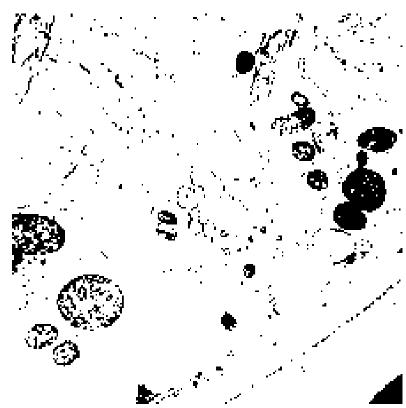
b2_a_otsu.png inter-class max-variance: 1448.013
b2_b_otsu.png inter-class max-variance: 1366.648
b2_c_otsu.png inter-class max-variance: 1079.709

(d) Note the intensity threshold chosen by the algorithm.

b2_a_otsu.png threshold value: 131
b2_a_otsu.png threshold value: 85
b2_a_otsu.png threshold value: 170

(e) Show the resulting binary image produced by the algorithm

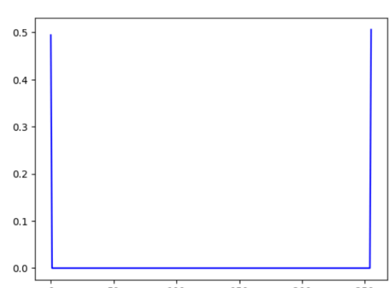


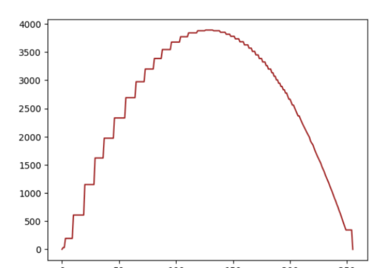


(f) Discuss the results.

Overall, the otsu method seems to provide an efficient way of selecting the optimal threshold value for each source image. Looking at image A, the otsu algorithm chose a threshold value of 131 to threshold the image. The above depictions of manual thresholding are depicted with a global threshold of 127 (near the midpoint). It seems that for image A the otsu algorithm chose a threshold value fairly close to the midpoint threshold, and does a fair job at showing the rough shape of the monkey's face from the input image. On the other hand, looking at image B the algorithm chose a threshold value of 85 which seems to skew towards the lower pixel intensity values due to the dark vignette-like aberrations in the photo. If the source image is taken into consideration, these aberrations likely cause B's histogram to be weighted towards darker intensity values. Therefore, it makes sense that otsu's algorithm would choose a threshold value lower than the midpoint threshold in order to ensure effective thresholding. The same principle applies for image C, except the histogram of the image depicts pixel intensity values that are weighted toward the high end of intensity scale. In turn, the otsu algorithm selects a threshold value above the midpoint to compensate for the distribution of the histogram. According to the article "A Threshold Selection Method from Gray-Level Histograms" by Nobuyuki Otsu, the best explanation of the algorithm is a method used to select a threshold automatically from a gray level histogram in which the optimal threshold selection is made using the measure of separability of the resultant classes in gray levels (Otsu, 1979, p. 66). Are there any predictions that can be made from this methodology? For one, if we perform histogram equalization on an image it stands to reason that the threshold chosen by the algorithm should be right at the midpoint. In terms of improvements that could be made to the algorithm, it seems to be a full proof method for dynamically determining the optimal threshold for an input image. The only improvement in theory that seems to be apparent would be to adapt the otsu algorithm to perform locally as opposed to globally. If the algorithm performs locally the algorithm should have more control over various regions or neighborhoods in the image. To answer the question asked above, for the sake of clarification on otsu's method an output thresholded image of the equalized 'people.png' photo will be posted below. When performing otsu thresholding on an equalized image the algorithm chose a threshold value of 126 (very close to the midpoint threshold).

people_equalized_otsu



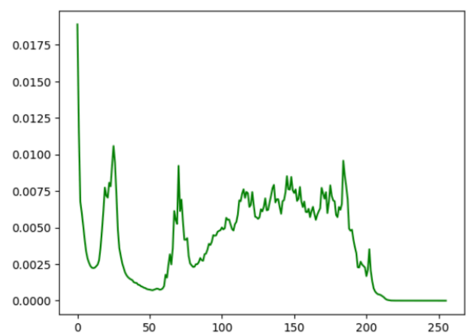


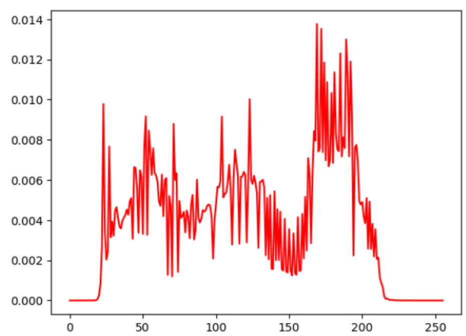
A3

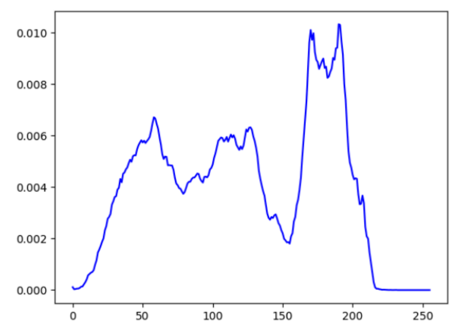
(a) Here are some creative examples of histogram matching from an input target image to a source input image:

Source Original A | Target A | Matched A



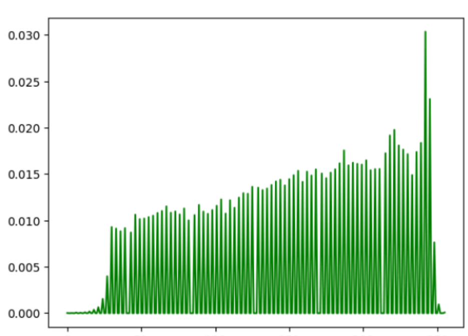


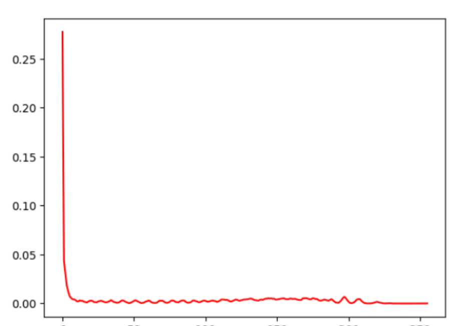


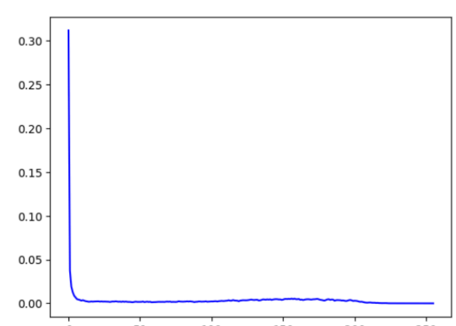


Source Original B | Target B | Matched B









(b) Brief Explanation of Histogram Matching:

The histogram_matching function is implemented using somewhat of a divide and conquer strategy. The function takes an input source image, and an input target image and performs several calculations on each to determine our result matched image. In order to successfully perform histogram matching it is necessary to calculate cumulative distribution for both the source and the target image. Therefore, the first step in histgroam matching is to calculate the probability density of both the source and the target image. Then these probability densities are used to caculate the respective cumulative distribution functions for both the source and the target image. After the cumulative distribution is calculated, the mapping is performed. In order to map the original source pixel values to the target matched pixel values a helper function called find_closest is implemented. This helper function helps to determine the closest matched cdf value for each pixel in the source cdf using the corresponding cdf value from the target image. Once this mapping is performed an image is written using the matched cdf mapping, this is our result matched image.

References

Zeng, S., & Otsu, N. (n.d.). Transactions on systems, man, and Cybernetics. Xing zheng yuan guo ke hui ke zi zhong xin.