Lab 5: Histogram Processing.

Objective:

The objective of this lab is:

- To create and apply a piece-wise linear histogram transform.
- To create and apply contrast stretching on histograms.
- To create and apply histogram equalization.

Theory:

Piece-wise Linear Transform:

Piece-wise linear transformation is usually used to process a histogram in a specific way. For instance, consider a situation in which it is required from you to isolate picture elements whose gray levels fall in a pre specified range from A to B. This process can be done in various ways (such as writing a computer program in which "if" statement can isolate the pixels) but a generic approach is to create a Transformation function and then applying that transformation function using cv2.LUT or your own **mapping** routine.

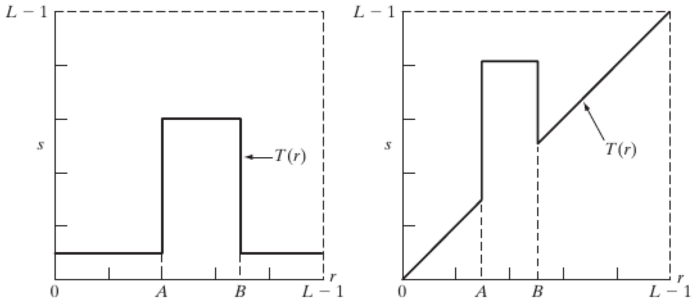


Figure 1: A transformation function highlights intensities within range [A, B] (left) and a transform which highlights [A,B] while retaining other pixels.

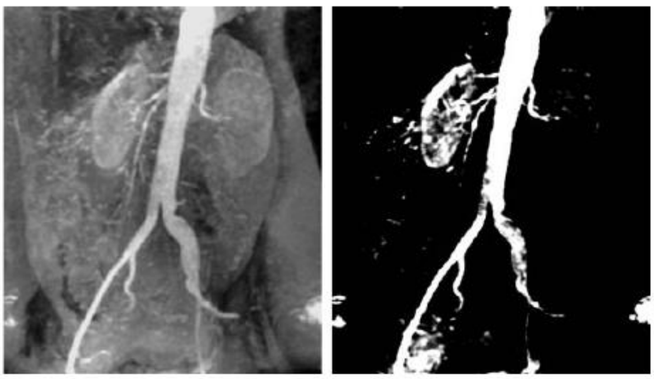


Figure 2: Aortic angiogram (left) and highlighted pixels within some range [A,B]

Contrast Stretching:

Contrast stretching is a process which involves extending a range of input histogram to an output histogram using following formula:

$$S=\left(\frac{S_{max}-S_{min}}{R_{max}-R_{min}}\right)(R-R_{min})+S_{min}$$

Where Smin and Smax are the goal ranges (Lower histogram in Figure 3) and Rmin and Rmax are the range you wish to stretch (Upper histogram in Figure 3). Since R is a range [Rmin, Rmax], a simple mathematical can be used to find out a one-to-one mapping for Ri (Rmin <= i <=Rmax). The process is robust enough to stretch one histogram of range [A,B] to [C,D] where both ranges can or cannot be of same gray-level values!

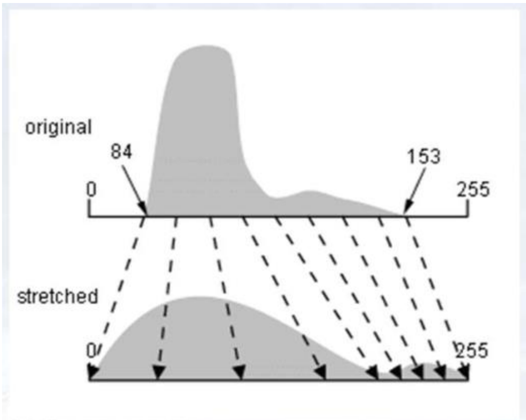


Figure 3: Contrast Streching from [84-153] range to [0-255]

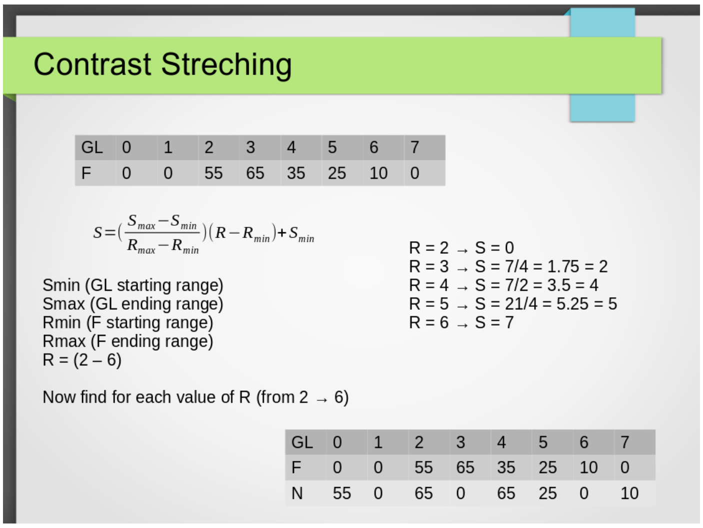


Figure 4: A self explanatory process (taught in class) to jog your memories.

In some cases, a piece-wise linear transformation function can also be created to achieve contrast stretching (as shown in Figure 5) in which a human expert selects two points (r1,s1) and (r2,s2) manually to enhance contrast of the image. The values of both points can also be automatically calculated using some predefined percentage of the available histogram! (e.g. 20% dark, 60% middle and 20% light gray-levels).

HINT: you can use CDF values to calculate the predefined percentages.

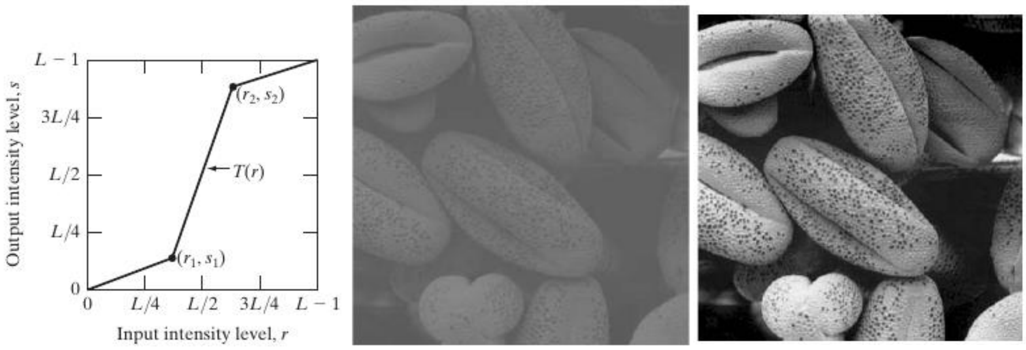
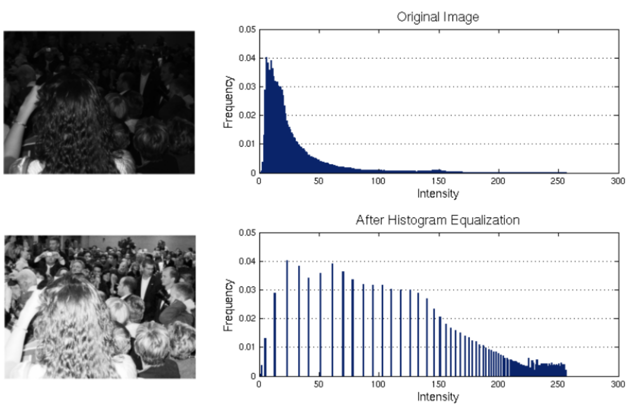


Figure 5: Contrast Stretching using piece-wise linear transformation function.

Histogram Equalization:

Histogram equalization is a method in image processing of contrast adjustment using the image's histogram. Histograms of an image before and after equalization. Some important points about the exercise.

- You should apply histogram equalization on any Grayscale image.
- You should not use the builtin histogram equalization method available in python or OpenCV
- You should use the formula mentioned below to implement the histogram equalization.



Submission Guidelines:-

Submit jupyter notebook containing code and output displayed.

Lab Tasks:

Task 1:

- Loadimage “kidney.tif”
- Make a piece-wise function capable of isolating areas of kidney and surrounding elements (as shown in Figure 2.
  - HINT: This is a mask version!
- Apply the mask and isolate pixels from the original image.
- Make a piece-wise function which keeps original image pixels except of range [A,B] (i.e. Figure 1.Left)
- Summarize your findings with images.

Task 2:

- Load a low contrast image “wiki.jpg”
- Create an algorithm which applies contrast stretching (pick any implementation you like i.e. either formula or points based)
  - **NOTE:** For now, you can select the stretching limits by your own.
- Apply same technique on “lowcon.tif”
- Summarize your findings on how to extend or automate the task!

Task 3:

- Write a program that equalizes the histogram of a given image. Consider the formula below

$$s_k = T(r_k) = \sum_{j=0}^{k} p_{in}(r_j) = \frac{(L-1)}{MN} \sum_{j=0}^{k} n_j$$

where

$$k = 0,1,2,\ldots,L-1$$

- Show the comparison of histograms before and after equalization obtained using:
  - Your Implementation of the algorithm
  - OpenCV’s implementation of Histogram Equalization
- Conclude your findings on following images and analyze the workings of Histogram
  - dark.tif
  - bright.tif
  - lowcon.tif
  - Wiki.jpg