ES 114: Probability, Statistics, and Data Visualization
Lab Assignment 5 [5 points]

Shanmuganathan Raman

February 22, 2024

1 General Instructions

Please read the following instructions carefully towards your assignment submission.

1. Implement the assignment in python from scratch without using any advanced libraries. You can use numpy, scipy, pandas, matplotlib, opencv or PIL (only for reading, writing, displaying, or transforming images) and other related libraries for preliminary tasks on images.
2. You need to show the result on grayscale images only.
3. Submit only a single Colab link in the classroom with both code and report (your observations and/or comments in the Markdown).
4. Diligently cite all the sources that you may have referred for completing this assignment. Copying and/or altering the exiting code repositories is strictly **NOT allowed**. Discussion among peers is allowed without exchanging codes. Any sort of malpractice will be have serious repercussions.
5. You need to show the results on the images provided in the 'Dataset.zip' folder.

2 Histogram Matching [5 Marks]

In the case of images, histograms are the statistical way to represent the pixel intensity distribution. Two images can have similar intensity histograms, but they can differ spatially, i.e., how the intensities are distributed. For this assignment, you need to perform Histogram Matching between two grayscale images, A and B. Histogram matching transforms the source image (say A) histogram distribution to the target image (say B) histogram distribution. In order to match the histogram of images A and B, you need to first equalize the histogram of both images. Then, you need to map each pixel of A to B using the equalized histograms and later modify each pixel of A based on B. Histogram equalization is a method to transform the skewed distribution of image intensities into an approximately uniform distribution over the $[0, L]$, where L is the maximum possible intensity of an image, which in our case is 255 (8-bit, Grayscale).

Note: We have provided an image dataset along with a pre-defined pair in Table 1 over which you have to perform histogram matching. The sample result is shown in Fig. 1. **DO NOT** use any libraries for histogram matching. You can refer to the opencv/PIL/scipy documentation for understanding.

3 steps :
1. Create a histogram
2. Perform Histogram Equalization
3. Perform Histogram Matching

Figure 1: Sample result showing left image histogram matching the center image histogram. The result is shown on the right.

Source Image	Target Image	Matched Histogram
		A
		B
		C
		A

Table 1: Pair of images over which you have to show histogram matching. Replace the A, B, C, and A with the matched histogram image after applying your algorithm.

Histogram Equalization

A ↓

0	2	1	3	4
1	3	4	3	3
0	1	3	1	4
3	1	4	2	0
0	4	2	4	4

Image

Intensity	Number of Pixels
0	4
1	5
2	3
3	6
4	7

Histogram Table

(A)

	# pixels	P_x	CDF	CDF x 4	#
0	4	4/25	4/25 = 0.16	0.64 → 1	9
1	5	5/25	9/25 = 0.36	1.44 → 1	
2	3	3/25	12/25 = 0.48	1.92 → 2	3
3	6	6/25	18/25 = 0.72	2.88 → 3	6
4	7	7/25	25/25 = 1	4 → 4	7
	25				

0 → 1
1 → 9
2 → 3
3 → 6
4 → 7

2	1	2	1	0
3	3	2	4	4
1	3	2	4	4
0	0	3	2	1
1	3	1	4	0

Target: Image B

Pixel Value	Histogram
0	4
1	6
2	5
3	5
4	5

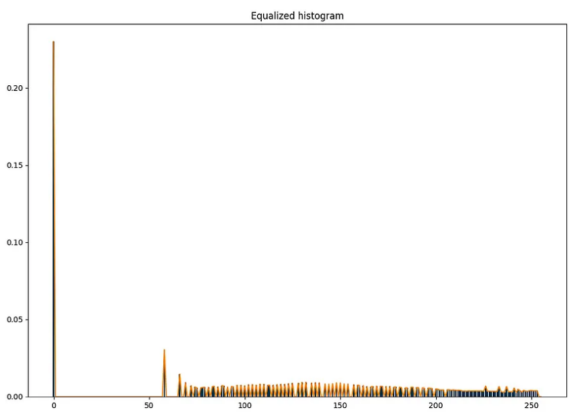
Equalized Histogra

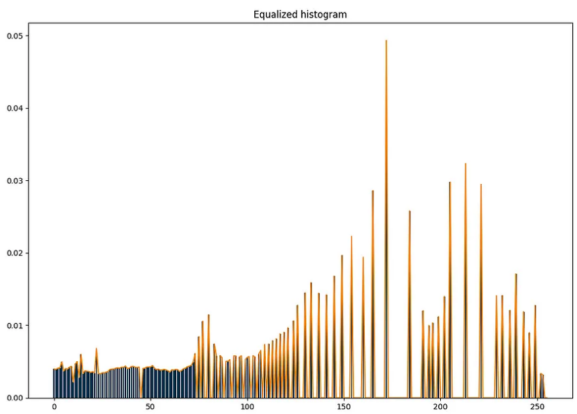
	# pixels	P_x	CDF	CDF x 4	#
0	4	4/25	4/25 = 0.16	0.64 → 1	4
1	6	6/25	10/25 = 0.4	1.6 → 2	11
2	5	5/25	15/25 = 0.6	2.4 → 2	
3	5	5/25	20/25 = 0.8	3.2 → 3	5
4	5	5/25	25/25 = 1	4 → 4	5

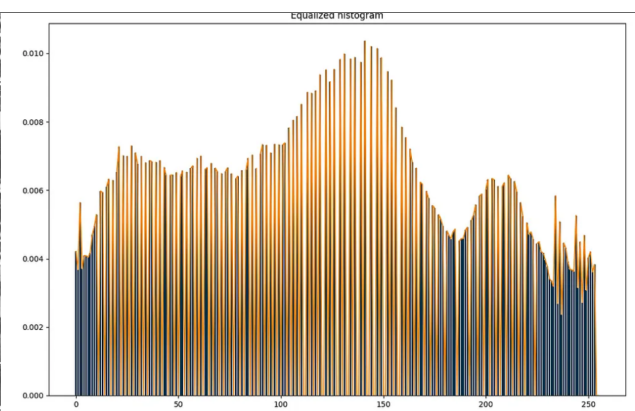
Equalized Histogram

Image A		
	0	0
	1	9
	2	3
	3	6
	4	7

Image B		
	0	0
	1	4
	2	11
	3	5
	4	5







0	2	1	3	4
1	3	4	3	3
0	1	3	1	4
3	1	4	2	0
0	4	2	4	4

input: Image A

Pixel Value	Histogram
0	4
1	5
2	3
3	6
4	7

2	1	2	1	0
3	3	2	4	4
1	3	2	4	4
0	0	3	2	1
1	3	1	4	0

Target: Image B

Pixel Value	Histogram
0	4
1	6
2	5
3	5
4	5

Image A

Int.	CDF	
0	4/25	4
1	9/25	9
2	12/25	12
3	18/25	18
4	25/25	25

Image B

CDF		Int
4/25	4	0
10/25	10	1
15/25	15	2
20/25	20	3
25/25	25	4

Final Mapping

A	A Matched with B
0	0
1	1
2	1
3	3
4	4

(nearest value mapping)