Digital Image Processing

Image Histograms

Image Histogram

- Histograms are collected *counts* of data organized into a set of predefined *bins.*
- *imhist() function calculates the histogram for the intensity image 'I' and displays a plot of the histogram. The number of bins in the histogram is determined by the image type.*

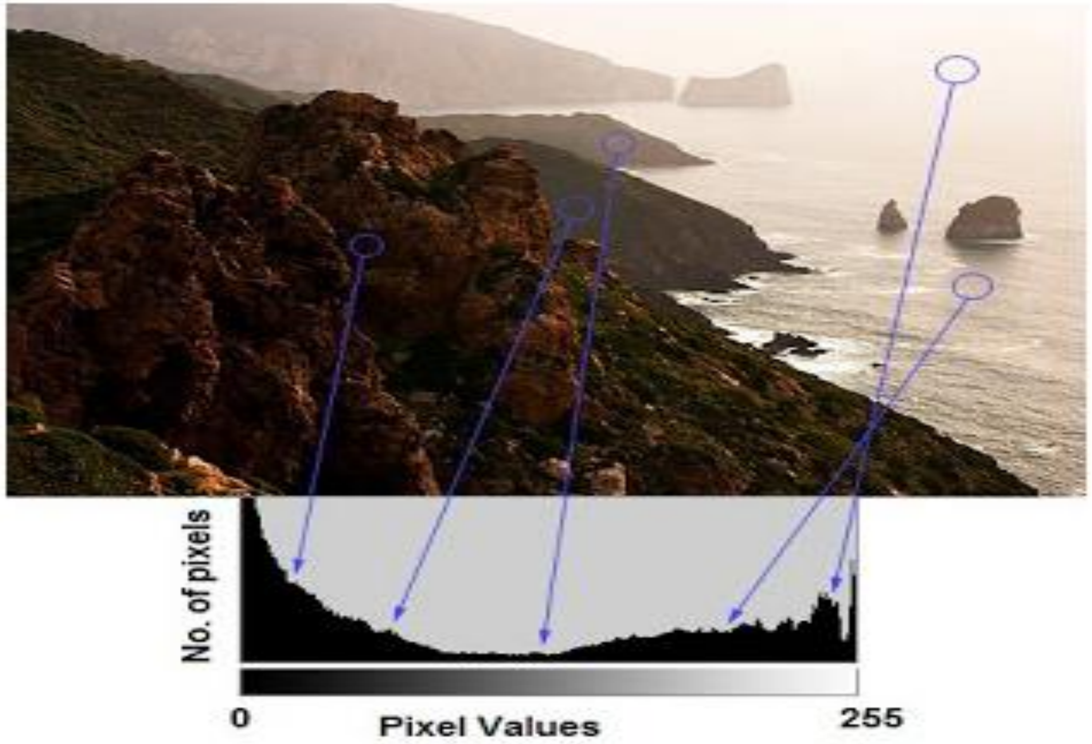
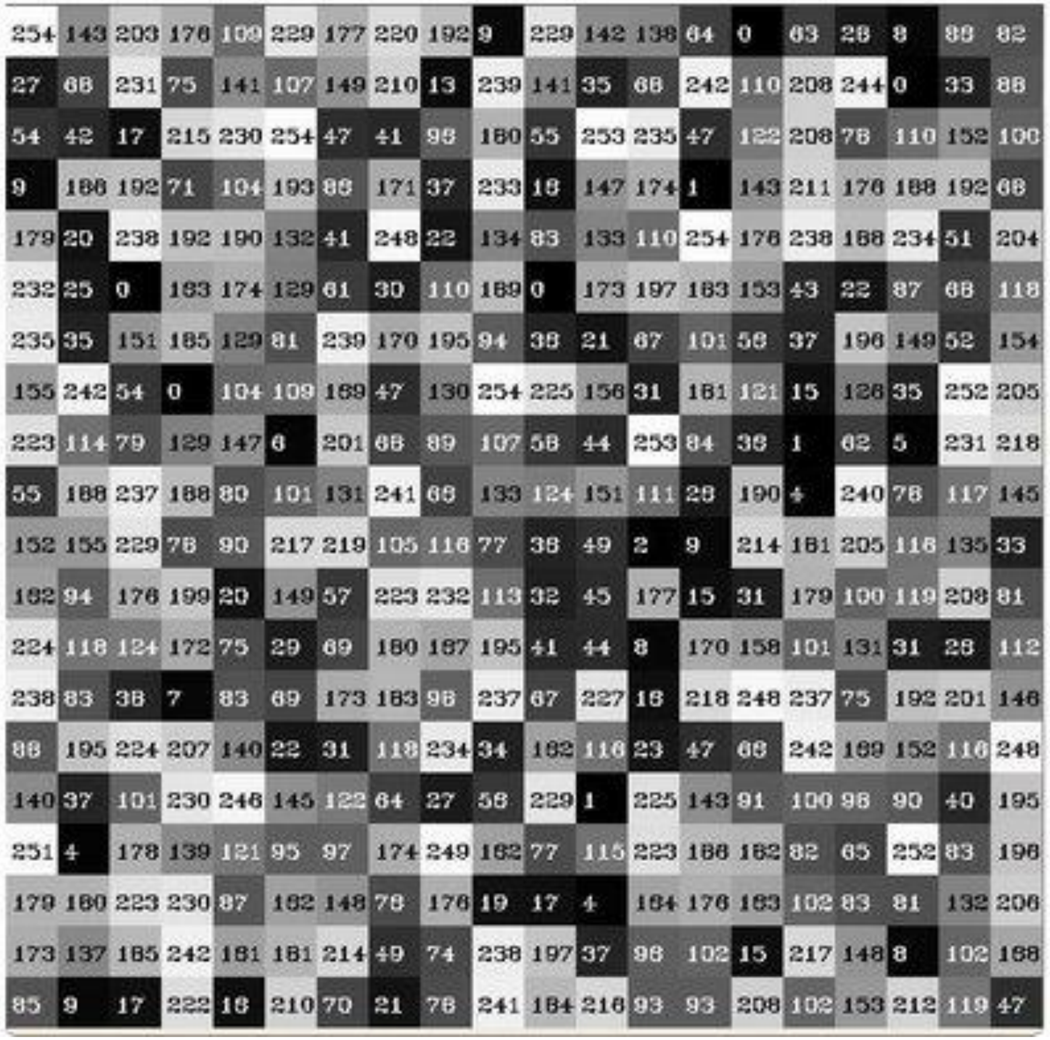


Image Histogram

- *What are bins?*

The number of pixels for every pixel value (pixel value ranges from 0 to 255). We can find the number of pixels for every pixel value separately or the number of pixels in an interval of pixel values

Image Histogram

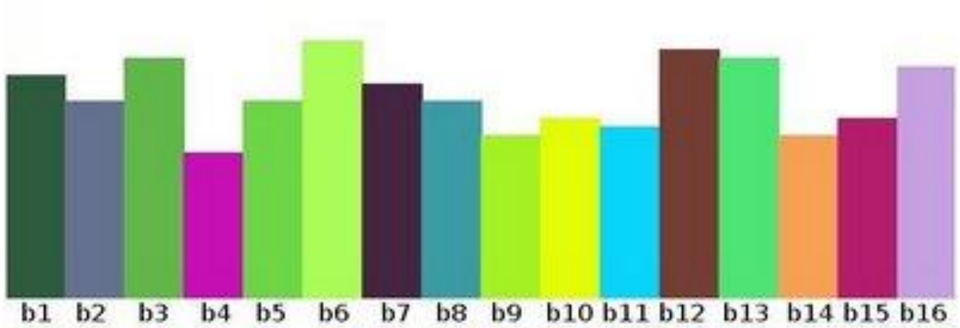


What happens if we want to *count* this data in an organized way?

The *range* of information value for this case is 256 values, we can segment the range in subparts (called **bins**) like:

$$[0,255] = [0,15] \cup [16,31] \cup .... \cup [240,255]$$
$$\text{range} = \text{bin}_1 \cup \text{bin}_2 \cup .... \cup \text{bin}_{n=15}$$

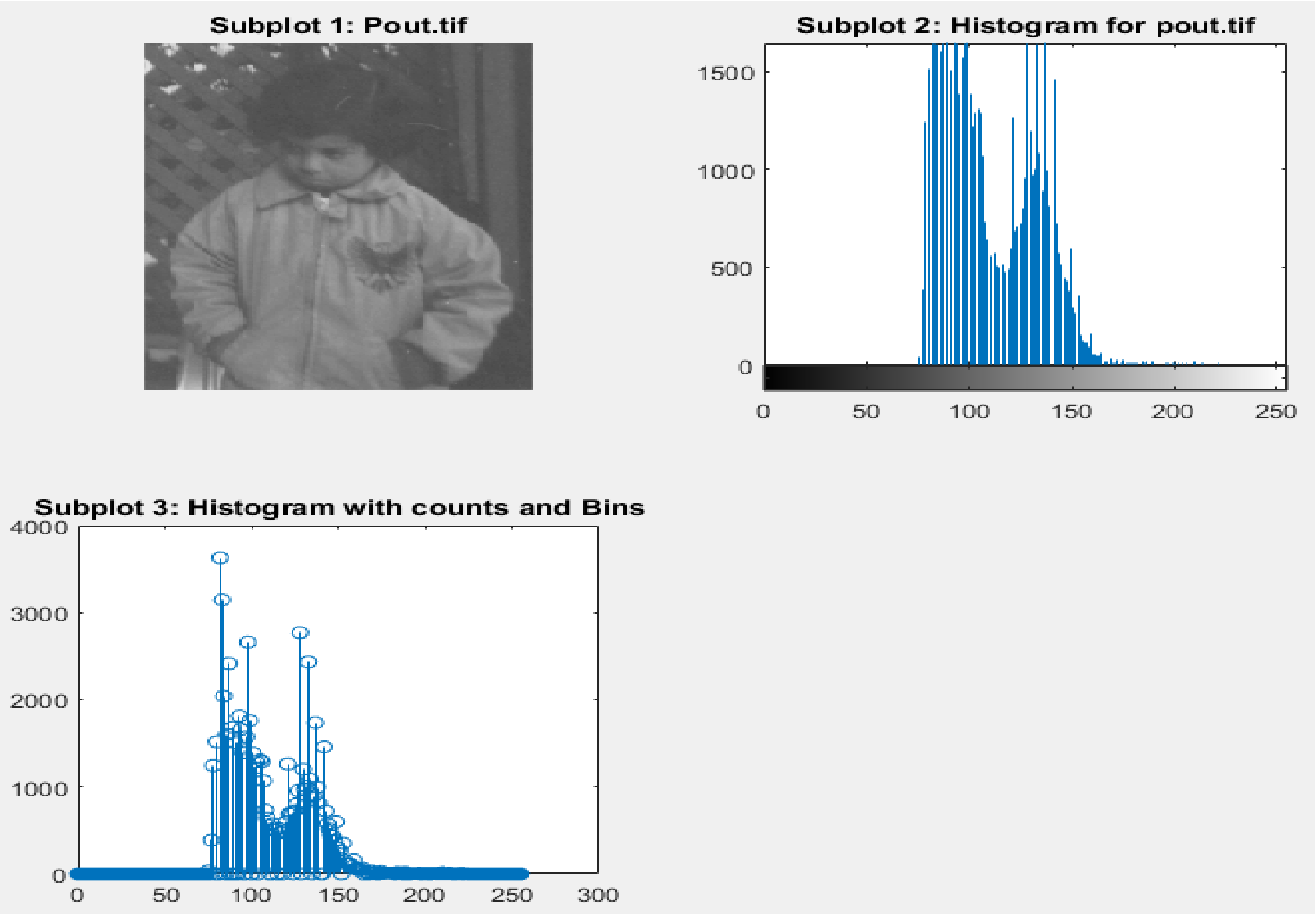
We can keep count of the number of pixels that fall in the range of each bin_i



Matlab Commands

- For plotting histogram
  - `imhist(img)`
- For finding counts and bin locations (bin refers to the pixel values as discussed in previous slide)
  - `[counts, binLocations] = imhist(img)`
- For plotting histogram with counts and bin locations
  - `stems(binLocations,counts)`

Results



Subplot 1: Pout.tif
Subplot 2: Histogram for pout.tif
1500
1000
500
0
0 50 100 150 200 250
Subplot 3: Histogram with counts and Bins
4000
3000
2000
1000
0
0 50 100 150 200 250 300

Image Histogram

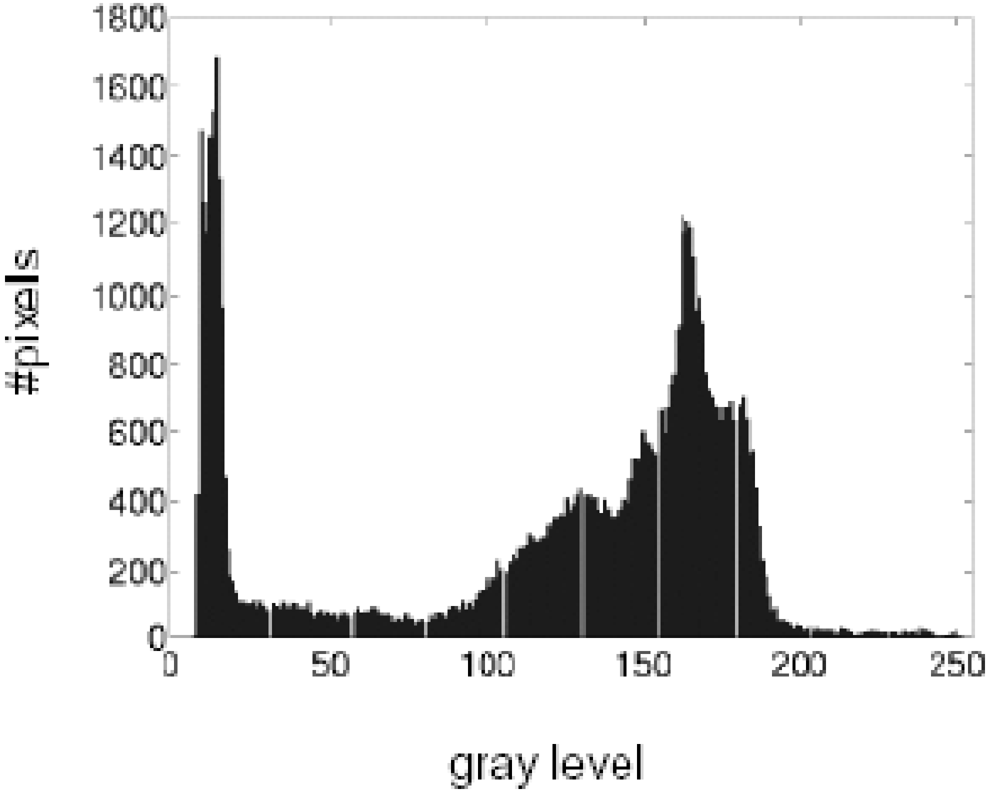
- The histogram of an image represents the frequency of pixels in an image
- The histogram of an image shows the distribution of gray levels in the image
- Massively useful in image processing especially in image segmentation
- Histogram of an image with gray level (0 to L-1):

A discrete function $h(r_k) = n_k$, where r_k is the k^{th} gray level and n_k is the number of pixels in the image having gray level r_k .

Image Histogram

- How to obtain histogram without imhist() command?
  - For B bit image, initialize 2^B counters with 0
  - Loop over all pixels x,y
  - When encouraging gray level f(x,y)=I, increment counter # i

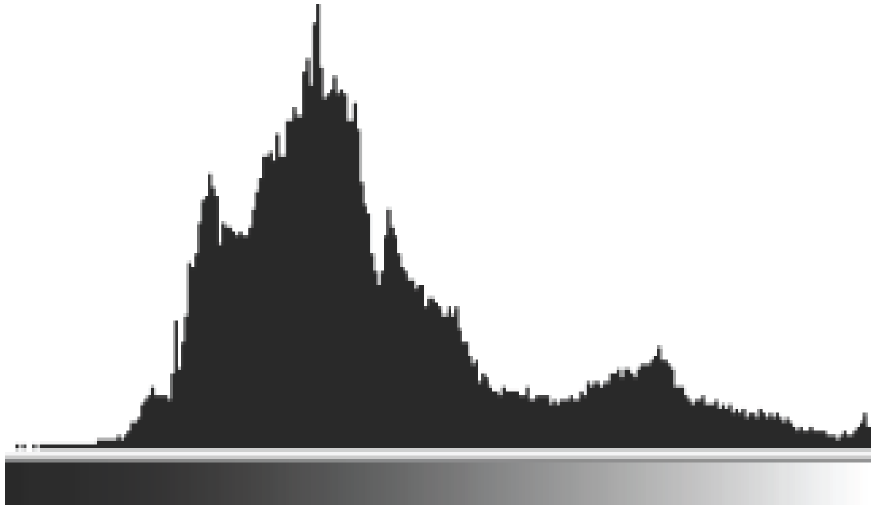
Histogram - Example



Cameraman
image

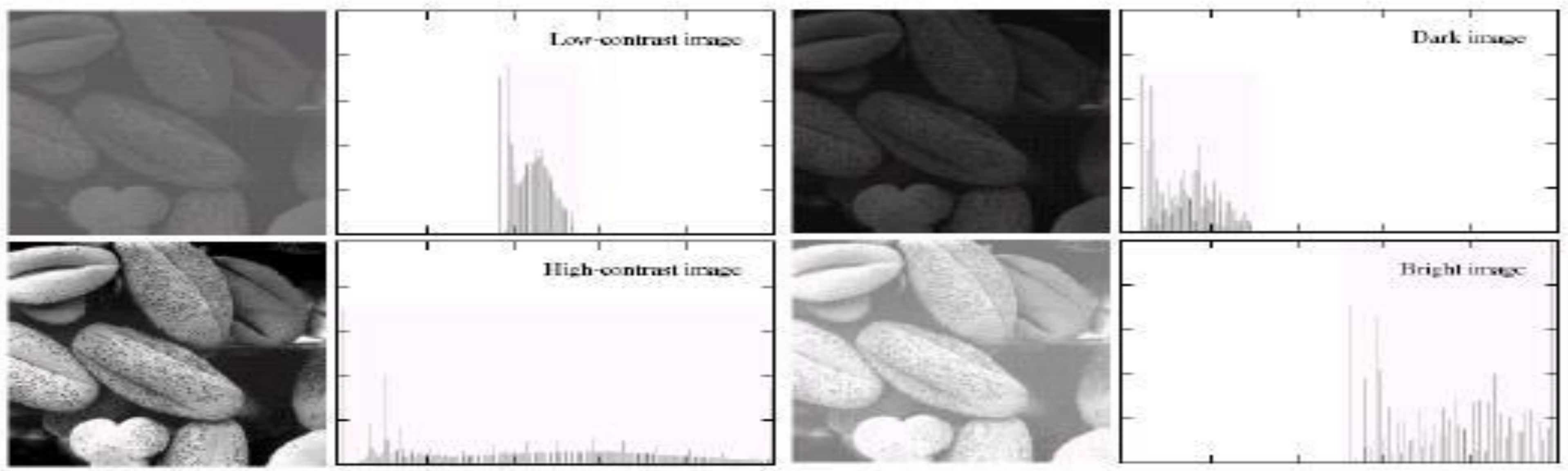
Histogram and Image Brightness/Contrast

- Changing the size of the image shrinks or stretches the histogram vertically.
- Adjusting brightness shifts the histogram horizontally.
- Adjusting contrast causes the histogram to stretch or shrink horizontally:
  - **Stretching = more contrast**
  - **Shrinking = less contrast**



Histogram - Example

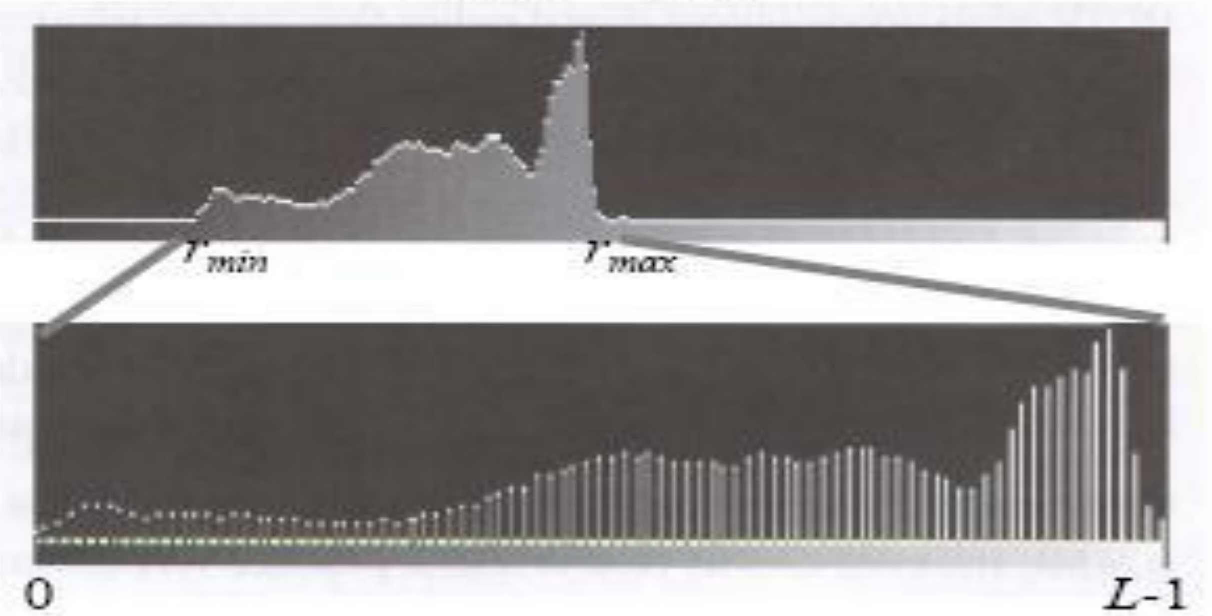
- A selection of images and their Histograms
- Note that the high contrast image has the most evenly spaced histogram
- Histograms of low contrast images are located in certain portions and not in the entire gray scale range



Contrast Stretching through Histogram

- If r_{max} and r_{min} are the maximum and minimum gray level of the input image and L is the total gray levels of output image The transformation function for contrast stretching will be

$$s = T(r) = (r - r_{\min})\left(\frac{L}{r_{\max} - r_{\min}}\right)$$



Histogram Equalization

- The idea is to find a transformation **s=T(r)** to be applied to each pixel of the input image **f(x,y)** such that a uniform distribution of gray levels in the entire range results for the output image **g(x,y)**
- The transformation function for histogram equalizations is:

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) \quad \text{for} \quad 0 \le k \le L-1$$

where $p_r(r_j) = \frac{n_j}{n}, \quad j = 0, \ldots\ldots, L-1 \quad and \quad n = \sum_{j=0}^{L-1} n_j$

n_j : number of pixels with gray level r_j

n : total number of pixels

Histogram Equalization

- Spreading out the frequencies in an image (or equalising the image) is a simple way to improve dark or washed out images
- The formula for histogram equalization is given where

✓ r_k: input intensity

✓ s_k: processed intensity

✓ k: the intensity (e.g 0.0 – 1.0)

✓ n_j: the frequency of intensity j

✓ n: the sum of all frequencies

$$s_k = T(r_k) = \sum_{j=1}^{k} p_r(r_j) = \sum_{j=1}^{k} \frac{n_j}{n}$$

Histogram Equalization

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

Value	Count	Value	Count	Value	Count	Value	Count	Value	Count
52	1	64	2	72	1	85	2	113	1
55	3	65	3	73	2	87	1	122	1
58	2	66	2	75	1	88	1	126	1
59	3	67	1	76	1	90	1	144	1
60	1	68	5	77	1	94	1	154	1
61	4	69	3	78	1	104	2		
62	1	70	4	79	2	106	1		
63	2	71	2	83	1	109	1		

Histogram Equalization

52	55	61	66	70	61	64	73
63	59	55	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

Initial Image

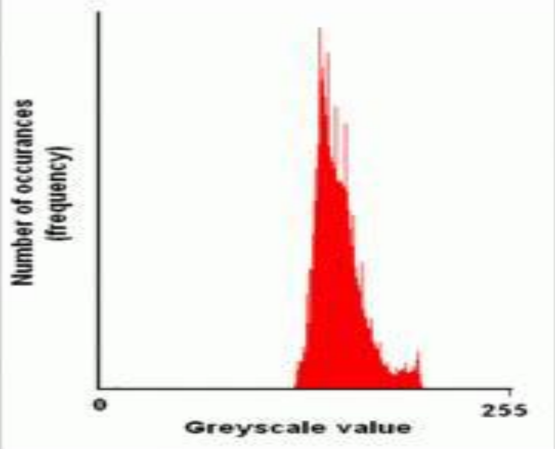
0	12	53	93	146	53	73	166
65	32	12	215	235	202	130	158
57	32	117	239	251	227	93	166
65	20	154	243	255	231	146	130
97	53	117	227	247	210	117	146
190	85	36	146	178	117	20	170
202	154	73	32	12	53	85	194
206	190	130	117	85	174	182	219

Image after Equalization

- **Notice that the minimum value (52) is now 0 and the maximum value (154) is now 255.**

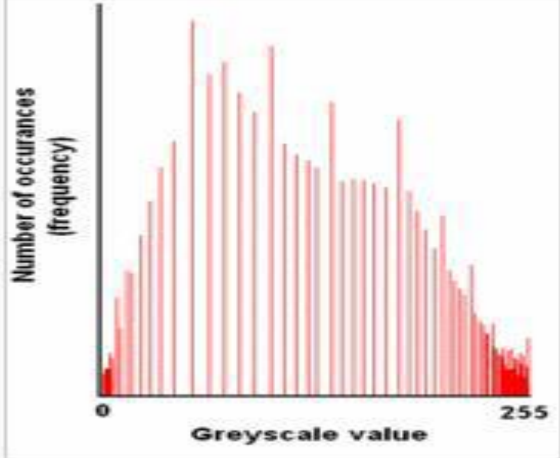
Histogram Equalization


An unequalized image



Corresponding histogram


The same image after histogram equalization



Corresponding histogram

Histogram Equalization

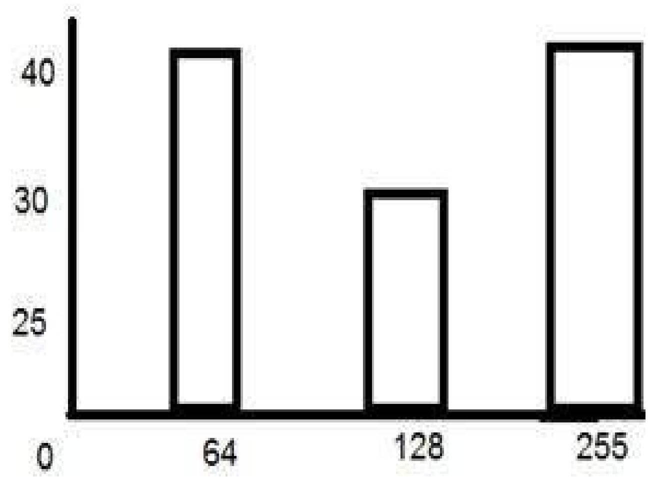
- While performing histogram equalization two most important concepts are

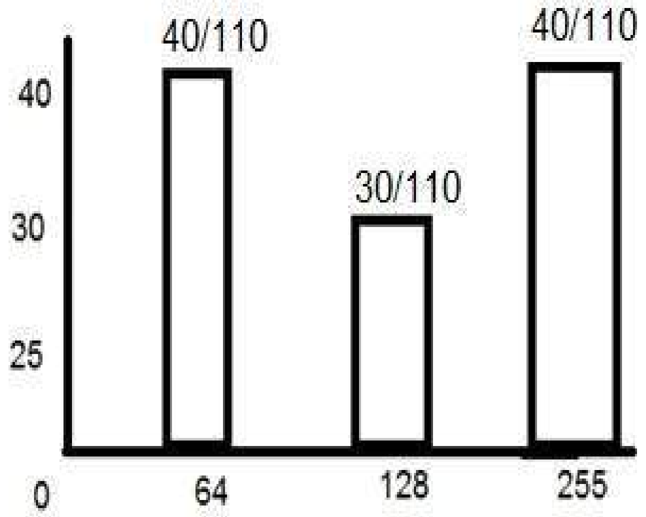
    - Probability Density Function (PDF)
    - Cumulative Distributive Function (CDF)

- How PDF and CDF are computed?

Histogram Equalization

- Calculating PDF from histogram

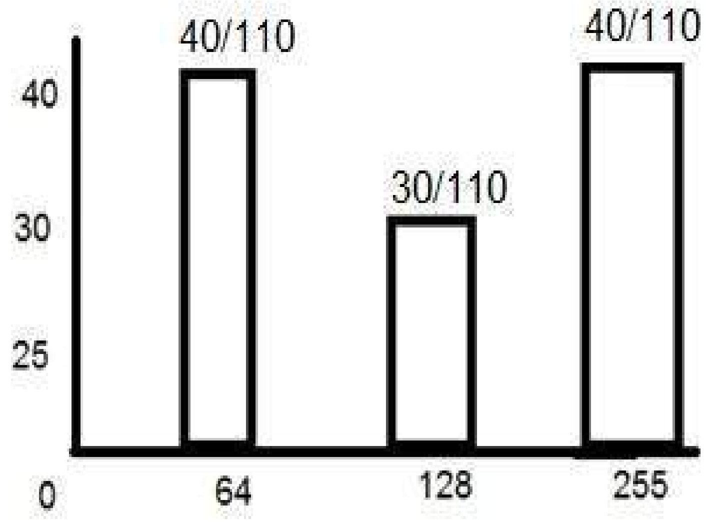


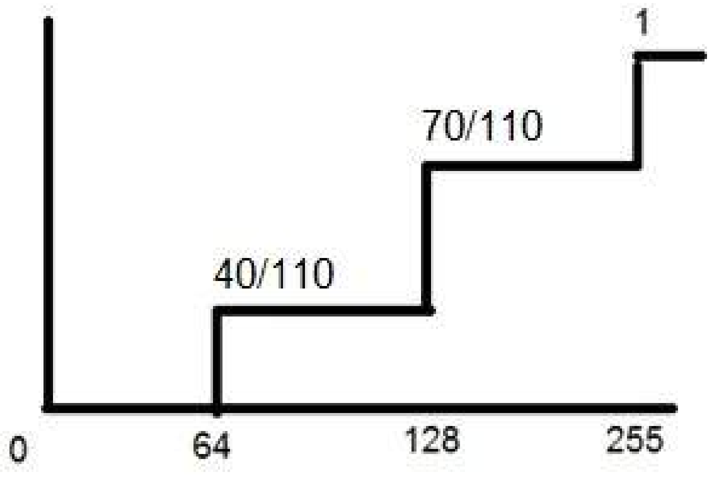


- Look at the count of each bar from vertical axis and then divide it by total count.

Histogram Equalization

- Calculating CDF from the histogram
- Keep the first value as it is, and then in the 2nd value , add the first one and so on.



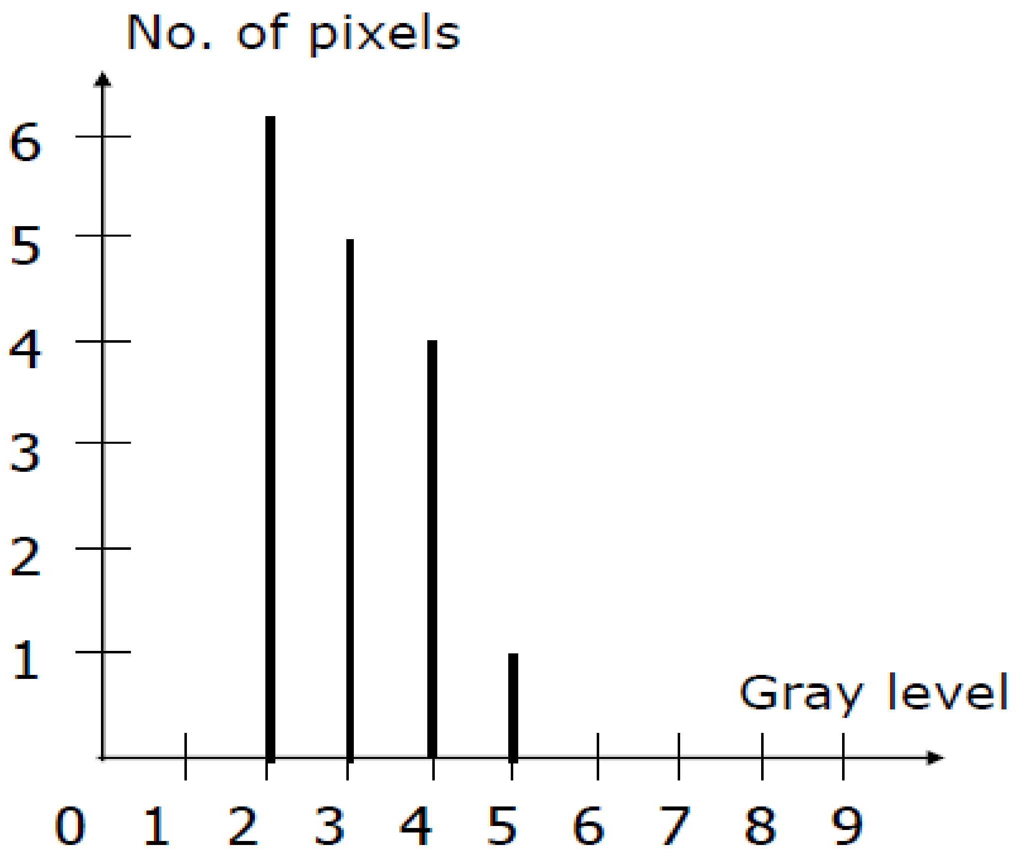


Histogram Equalization

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

4x4 image

Gray scale = [0,9]



histogram

Histogram Equalization

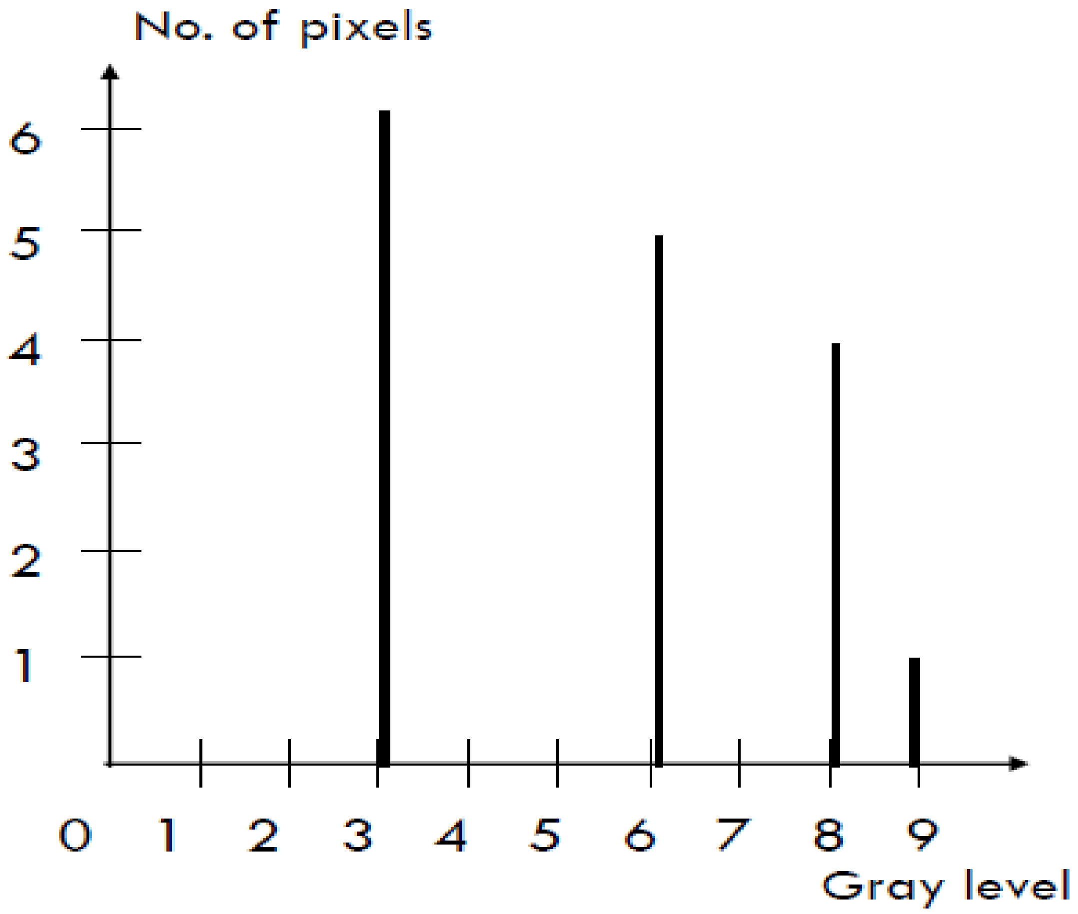
GL	0	1	2	3	4	5	6	7	8	9
n_p	0	0	6	5	4	1	0	0	0	0
nj/n	0	0	0.375	0.3125	0.25	0.0625	0	0	0	0
$s=\sum n_j/n$	0	0	0.375	0.6875	0.9375	1	1	1	1	1
sx9	0	0	3.375 ~=3	6.1875 ~= 6	8.4375 ~= 8	9	9	9	9	9
EQ	0	0	3	6	8	9	9	9	9	9

Histogram Equalization

3	6	6	3
8	3	8	6
6	3	6	9
3	8	3	8

Output image

Gray scale = [0,9]



Equalized histogram

Example: Histogram Equalization