

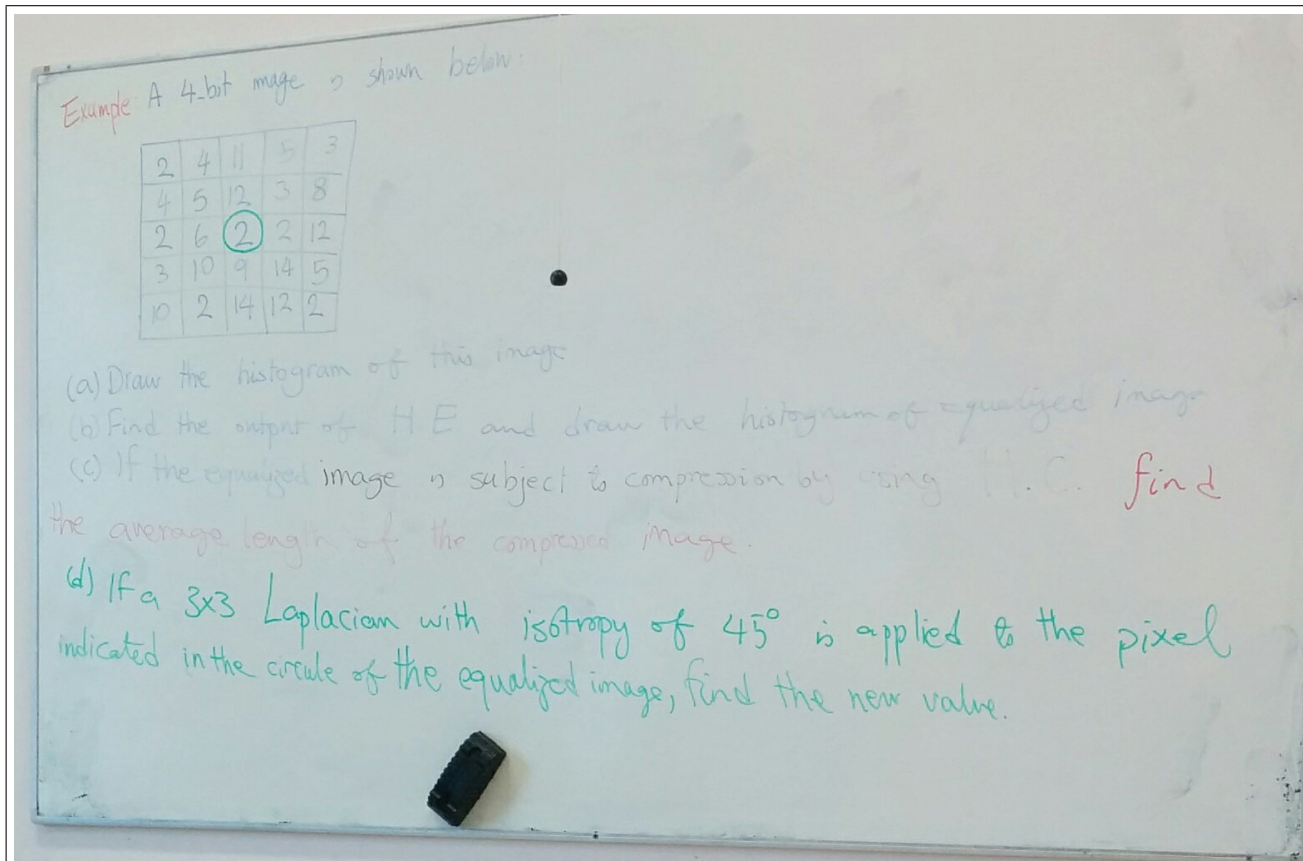
# Image Processing

## Home work 08

Aqeel Labash

**Lecturer:** Gholamreza Anbarjafari

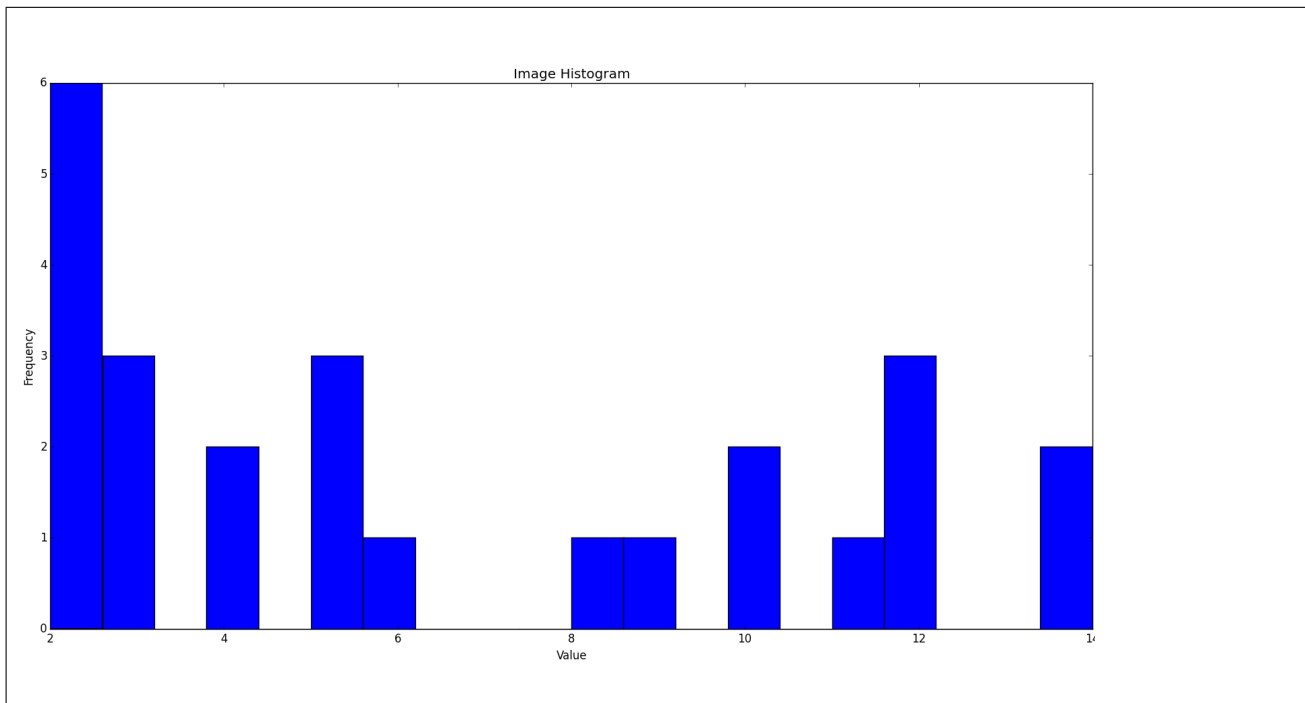
May 17, 2016



**A**

The histogram is just the frequency vs Value. The values as in the following table :

Value	Count
2	6
3	3
4	2
5	3
6	1
8	1
9	1
10	2
11	1
12	3
14	2



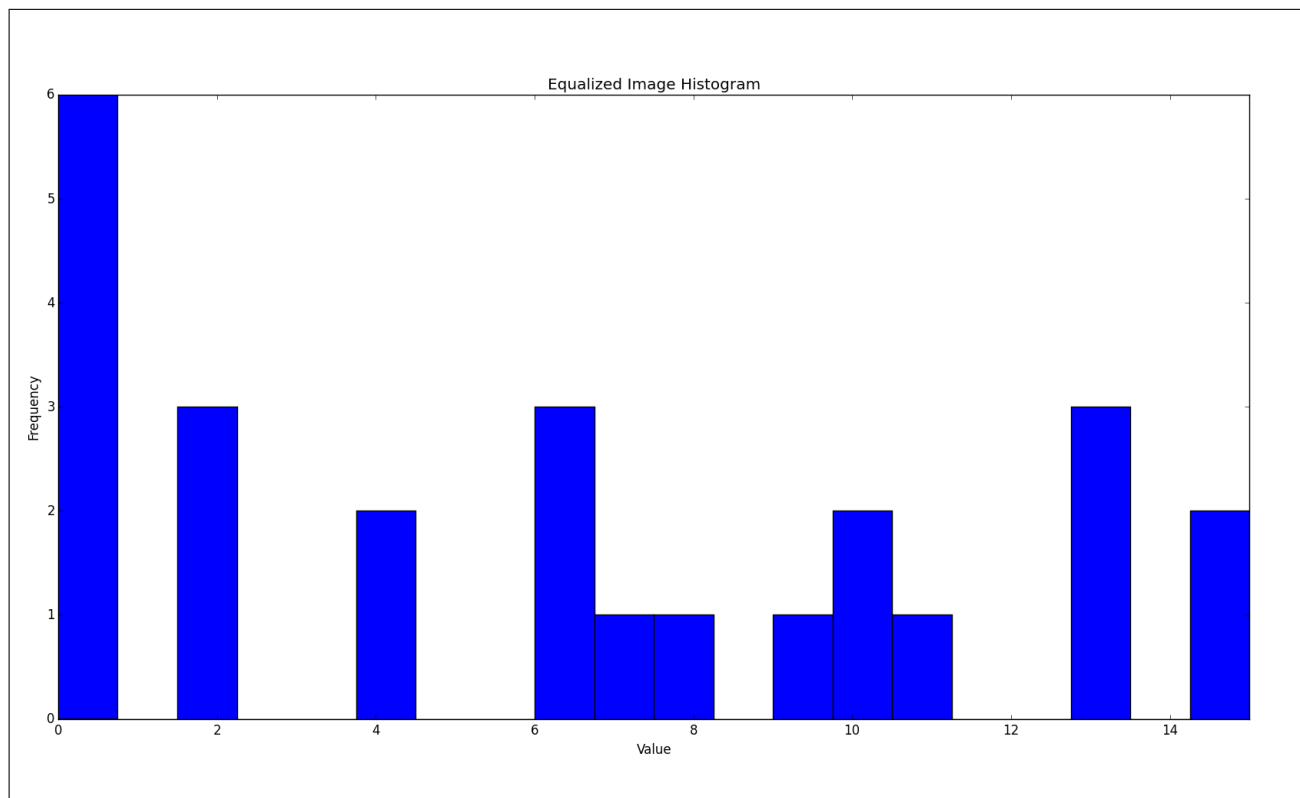
*Fig. 1:* Histogram of the image

## B

First here is the values :

value	Count	Probablitiy	CDF	HE
2	6	0.24	0.24	0
3	3	0.12	0.36	2
4	2	0.08	0.44	4
5	3	0.12	0.56	6
6	1	0.04	0.60	7
8	1	0.04	0.64	8
9	1	0.04	0.68	9
10	2	0.08	0.76	10
11	1	0.04	0.80	11
12	3	0.12	0.92	13
14	2	0.08	1.00	15

And here is the historam of the equalized image :



*Fig. 2:* Histogram of the equalized image

## C

For Huffman coding I found a way with numbers (same as probabilities but just with counts). I made the draw on the board here is the image :

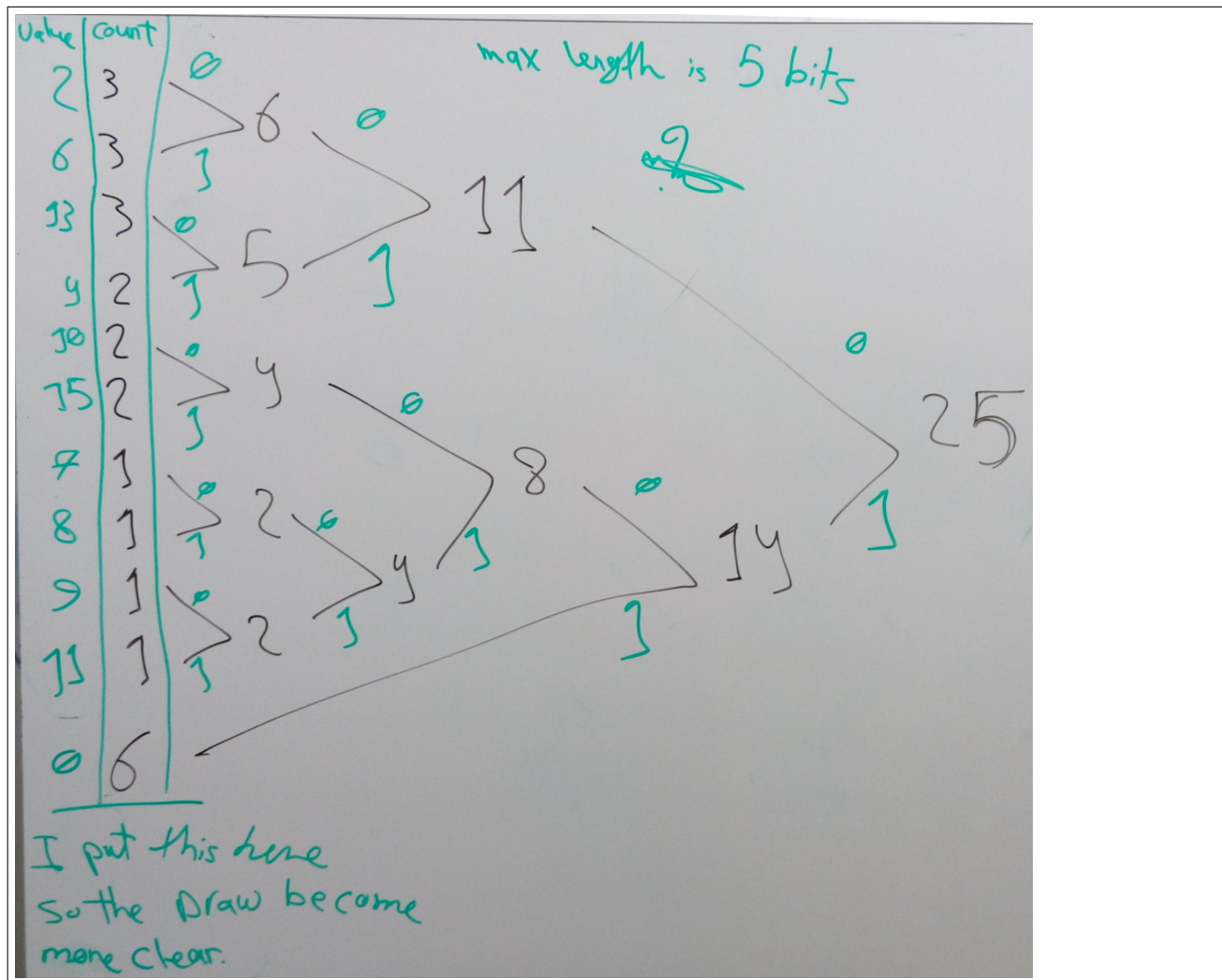


Fig. 3: The process of building Huffman Code

Here is the values:

value	Count	Pr	Code.Length
0	6	0.24	2
2	3	0.12	3
6	3	0.12	3
13	3	0.12	5
4	2	0.08	3
10	2	0.08	5
15	2	0.08	5
7	1	0.04	3
8	1	0.04	4
9	1	0.04	4
11	1	0.04	5

From the previous table we can see that the average length is  $3.4800000000000004 \approx 3.48$

**Note:** For more details please check my solution file (HTML in python with figures + tables at HW8.ipynb on git hub )

D)

The value of laplacian image for that specific pixel is 62 and the new value is -62.

Filter mask for digital laplacian :

$$L = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

The equalized image at the same point in circle :

IM =

6	13	2
7	0	0
10	9	15

$$Sum(L * IM) = 62$$

depending on the rule

$$g(x,y) = f(x,y) - \nabla^2 f(x,y) \text{ since the center coefficient of the laplacian mask is negative}$$

$$= -62$$

**Note:**All .py , .tex , .pdf , .png , etc.. exist on github

**E.O.F**