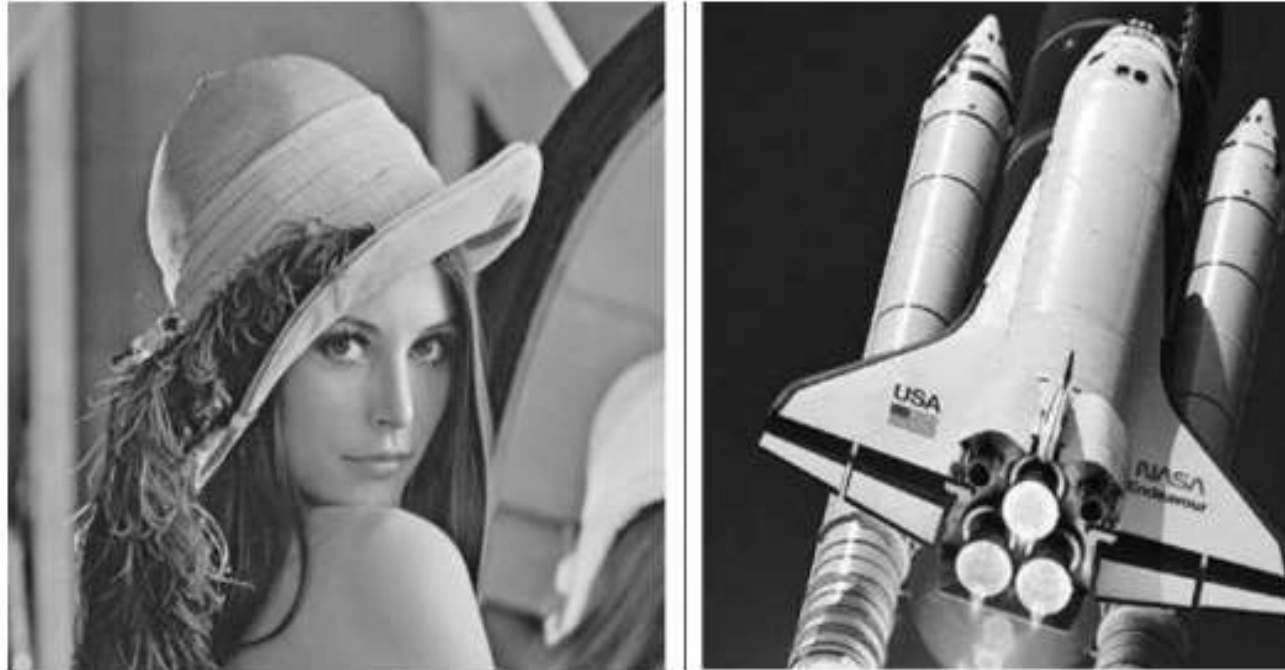




Each pixel is usually stored as a byte (value between 0 to 255). A dark pixel may have a value of 10; a bright one may be 240 (**dark=0; white=255**)

Example: find image size with 640*480 pixels

Total no. of bits = $640 * 480 * 8 \text{ bit} = 2457600 \text{ bit}$ Or $640 * 480 * 1 \text{ Byte} = 307200 \text{ Bytes}$

Convert to Byte: $2457600 / 8 = 307200 \text{ Byte}$

Convert to KByte: $307200 \text{ Byte} / 1024 = 300 \text{ KB}$

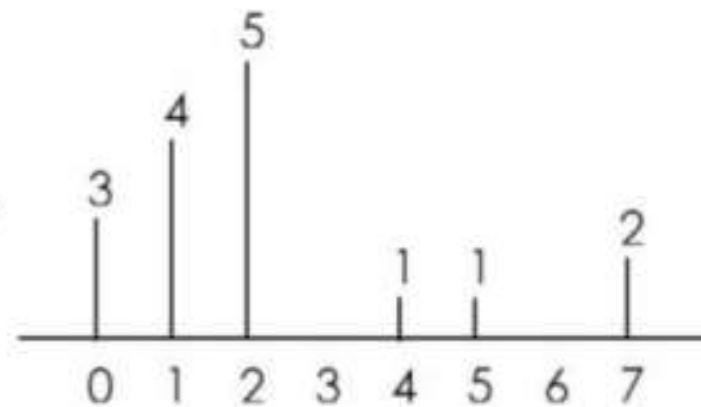
Examples: Color Image uses 32 bits with high = 200 and width = 200 pixels?

Size= $200 \times 200 \times 4$ Bytes = 160000B / 1024 = 156.25KB.

Example: Plot the Histogram of the following example with 4x4 matrix of a 3-bit image.

0	0	0	4
1	1	1	5
1	2	2	7
2	2	2	7

Answer 

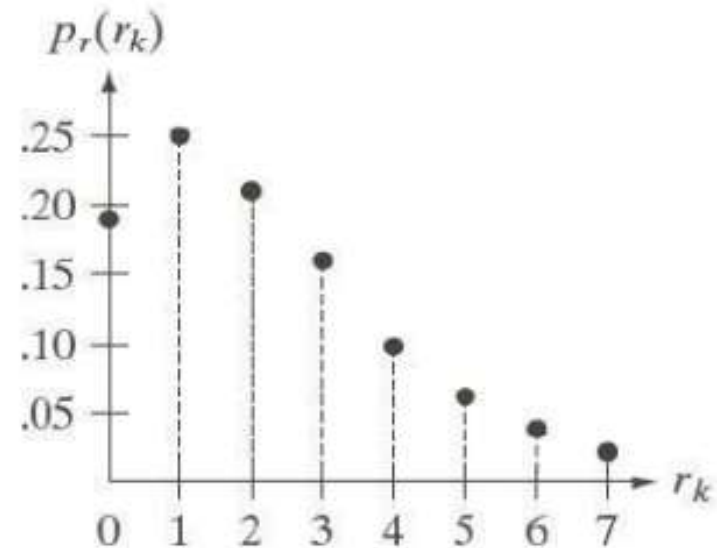


Example: Suppose that a 3-bit image ($L = 8$) of size 64×64 pixels has the gray level (intensity) distribution shown in the table below. **Perform normalized histogram.**

r_k	n_k
$r_0 = 0$	790
$r_1 = 1$	1023
$r_2 = 2$	850
$r_3 = 3$	656
$r_4 = 4$	329
$r_5 = 5$	245
$r_6 = 6$	122
$r_7 = 7$	81

Solution: $M \times N = 4096$. We compute the normalized histogram:

r_k	n_k	$p(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02



Normalized histogram

Example: Apply histogram stretching for the following sub image :

$$\begin{bmatrix} 7 & 12 & 8 \\ 20 & 9 & 6 \\ 10 & 15 & 1 \end{bmatrix}$$

Where: **Max = 255** ;

Min = 0

Solution:

$$St(r, c) = \left[\frac{I(r, c) - I(r, c)_{min}}{I(r, c)_{max} - I(r, c)_{min}} \right] (Max - Min) + Min$$

$$I(r, c)_{min} = 1 ;$$

$$I(r, c)_{max} = 20 ;$$

$$Max = 255 ;$$

$$Min = 0$$

$$I_{(0,0)} = [7-1 / 20-1] * [255 - 0] + 0 = 80.5$$

$$I_{(0,1)} = [12-1 / 20-1] * [255 - 0] + 0 = 147.6$$

$$I_{(0,2)} = [8-1 / 20-1] * [255 - 0] + 0 = 93.9$$

$$I_{(1,0)} = [20-1 / 20-1] * [255 - 0] + 0 = 255$$

$$I_{(1,1)} = [9-1 / 20-1] * [255 - 0] = 107.3$$

$$I_{(1,2)} = [6-1 / 20-1] * [255 - 0] + 0 = 67.1$$

$$I_{(2,0)} = [10-1 / 20-1] * [255 - 0] + 0 = 120.7$$

$$I_{(2,1)} = [15-1 / 20-1] * [255 - 0] + 0 = 187.8$$

$$I_{(2,2)} = [1-1 / 20-1] * [255 - 0] + 0 = 0$$

$$\begin{bmatrix} 80 & 147 & 93 \\ 255 & 107 & 67 \\ 120 & 187 & 0 \end{bmatrix}$$

Example: Apply histogram shrink for the following sub image :

70	120	80
[200	90	60]
100	150	10

Where: **Shrink max** =100 ; **shrink min** =20

Solution:

$$\text{Shrink}(I(r, c)) = \left[\frac{\text{Shrink}_{\text{MAX}} - \text{Shrink}_{\text{MIN}}}{I(r, c)_{\text{MAX}} - I(r, c)_{\text{MIN}}} \right] [I(r, c) - I(r, c)_{\text{MIN}}] + \text{Shrink}_{\text{MIN}}$$

$I(r, c)_{\min} = 10$; $I(r, c)_{\max} = 200$; **Shrink max** =100 ; **shrink min** =20

$$I_{(0,0)} = [100-20 / 200-10] * [70 - 10] + 20 = 45.2$$

$$I_{(0,1)} = [100-20 / 200-10] * [120 - 10] + 20 = 66.3$$

$$I_{(0,2)} = [100-20 / 200-10] * [80 - 10] + 20 = 49.4$$

$$I_{(1,0)} = [100-20 / 200-10] * [200 - 10] + 20 = 100$$

$$I_{(1,1)} = [100-20 / 200-10] * [90 - 10] + 20 = 53.68$$

$$I_{(1,2)} = [100-20 / 200-10] * [60 - 10] + 20 = 41.05$$

$$I_{(2,0)} = [100-20 / 200-10] * [100 - 10] + 20 = 57.89$$

$$I_{(2,1)} = [100-20 / 200-10] * [150 - 10] + 20 = 78.94$$

$$I_{(2,2)} = [100-20 / 200-10] * [10 - 10] + 20 = 20$$

45	66	49
[100	53	41]
57	78	20

Example: Apply histogram slide for the following sub image, where OFFSET= 10 :

7	12	8
[20	9	6]
10	15	1

Solution:

$$\text{Slide}(I(r,c)) = I(r,c) + \text{OFFSET}$$

17	22	18
[30	19	16]
20	25	11

• • •

✚ Example: Original sequence: 111122233333311112222
can be encoded as: (4,1),(3,2),(6,3),(4,1),(4,2)

Example: The source of information A generates the symbols {A0, A1, A2, A3 and A4} with the corresponding probabilities {0.4, 0.3, 0.15, 0.1 and 0.05}. Encoding the source symbols using binary encoder and Shannon-Fano encoder gives:

Source Symbol	P_i	Binary Code	Shannon-Fano
A0	0.4	000	0
A1	0.3	001	10
A2	0.15	010	110
A3	0.1	011	1110
A4	0.05	100	1111
L_{avg}	$H = 2.0087$	3	2.05

Example: Design Huffman codes for $A = \{a_1, a_2, \dots, a_5\}$, having the probabilities $\{0.2, 0.4, 0.2, 0.1, 0.1\}$.

Symbol	Step 1	Step 2	Step 3	Step 4	Codeword
a_2	0.4	0.4	0.4	0.6 0	1
a_1	0.2	0.2	0.4	0.4 1	01
a_3	0.2	0.2	0.2		000
a_4	0.1	0.2			0010
a_5	0.1				0011

Letter	Probability	Codeword
a_2	0.4	1
a_1	0.2	01
a_3	0.2	000
a_4	0.1	0010
a_5	0.1	0011

Lossless Compression Algorithms

- 1) Run-length Encoding (RLE)
- 2) Simple Repetition Suppression
- 3) Pattern Substitution
- 4) Shanon-Fano Method
- 5) Huffman Code

Example

If you know that original image is **256×256** pixel, single band (**gray scale**), **8-bit per pixel**. After compression the image file is **6,554 byte**.

Calculate the compression ratio

Solution

Size U = Size of image = $256 * 256 * 1 = 65,536$ bytes

$$\text{Compression ratio} = \frac{\text{Size U}}{\text{Size C}} = \frac{65536}{6554} = 9.999 = 10$$

This is called a “10 to 1” compression

Note: repeat the previous question for image with three bands.

Example :

Suppose you have the following sub of image. Perform the following operation

1. Stretch histogram.
2. Shrink (compress) histogram
3. Slide the histogram

7	12	8	7
11	9	6	4
10	5	1	5

Where: Image with 6 bit for each pixel

$\text{Shrink}_{\max} = 30$, $\text{Shrink}_{\min} = 10$

OFFSET values = 50

Solution :

1. Stretch histogram

$I(r,c)_{\max} = 12$, $I(r,c)_{\min} = 1$, $\text{Max} = 63$, $\text{Min} = 0$

$$\text{Stretch } (I(r, c)) = \left[\frac{I(r, c) - I(r, c)_{\min}}{I(r, c)_{\max} - I(r, c)_{\min}} \right] [\text{MAX-MIN}] + \text{MIN}.$$

$$I(r, c) = ((7-1)/(12-1)) * (63-0)+0 \\ = (6/11)*63 = \mathbf{34.36}$$

34.36			

12

2. Shrink histogram

$I(r,c)_{\max} = 12$, $I(r,c)_{\min} = 1$,
 $\text{Shrink}_{\max} = 30$, $\text{Shrink}_{\min} = 10$

7	12	8	7
11	9	6	4
10	5	1	5

$$\text{knirhS } (r(I, c)) = \left[\frac{\text{Shrink}_{\max} - \text{Shrink}_{\min}}{I(r, c)_{\max} - I(r, c)_{\min}} \right] [I(r, c) - I(r, c)_{\min}] \text{knirhS} + \text{min}$$

$$I(r, c) = ((30 - 10) / (12-1)) * (7 - 1) + 10 \\ 10 + 6*(20/11) = \mathbf{20.86}$$

20.86			

3. Slide the histogram

OFFSET values = 50

$$\text{Slide } (I(r, c)) = [I(r, c)] + \text{OFFSET}$$

$$I(r, c) = 7 + 50 \\ = \mathbf{57}$$

57	62	58	

Example

Let's assume we have a simple grayscale image with the following pixel values and their corresponding frequencies:

Use the Shannon-Fano method To compress the image

Pixel Value	Frequency
0	50
1	100
2	12
3	25
4	6
5	3
6	1

solve this using the **Shannon-Fano algorithm**, we follow these steps:

Step 1: Sort the Pixel Values by Frequency (Descending)

Pixel Value Frequency

1 100

0 50

3 25

2 12

4 6

5 3

6 1

Step 2: Apply Shannon-Fano Algorithm

Goal: Assign binary codes by dividing the list into two parts where the total frequencies are as equal as possible, then assign 0 to one part and 1 to the other.

Step 2.1: First Split

Group A	Group B
---------	---------

1 (100) + 0 (50) = 150	3 (25) + ... = 47
------------------------	-------------------

Assign:

- Group A: 0
 - Group B: 1
-

Step 2.2: Split Group A [1, 0]

Group A1	Group A2
----------	----------

1 (100)	0 (50)
---------	--------

Assign:

- Pixel 1: 00
 - Pixel 0: 01
-

Step 2.3: Split Group B [3, 2, 4, 5, 6]

Group B1	Group B2
----------	----------

3 (25) + 2 (12) = 37	4 (6) + 5 (3) + 6(1) = 10
----------------------	---------------------------

Assign:

- B1: 10
 - B2: 11
-

Step 2.4: Split B1 [3, 2]

Group B1a Group B1b

3 (25) 2 (12)

Assign:

- Pixel 3: 100
- Pixel 2: 101

Step 2.5: Split B2 [4, 5, 6]

Group B2a Group B2b

4 (6) + 5 (3) 6 (1)

Assign:

- B2a: 110
- B2b: 111

Now split B2a:

- 4 (6): 1100
- 5 (3): 1101

Pixel 6: 111

Final Shannon-Fano Codes:

Pixel Value Frequency Code

1	100	00
0	50	01
3	25	100
2	12	101

Pixel Value Frequency Code

4	6	1100
5	3	1101
6	1	111

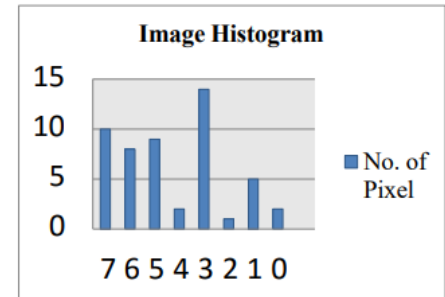
Let me know if you'd like to calculate the **average code length**, compression ratio, or visualize the Shannon-Fano tree!

Example- :

We have an image with 3 bit /pixel, so the possible range of values is 0 to . 7

We have an image with the following histogram

Gray level value	0	1	2	3	4	5	6	7
No. of Pixel histogram value	10	8	9	2	14	1	5	2



Answer:

Step :1

Great a running sum of histogram values. This means that the **first values** is 10, The **second** is **10+8=18**, next is **10+8+9=27**, and soon. Here we get

.10,18,29,43,44,49,51

Gray level value	0	1	2	3	4	5	6	7
No. of Pixel	10	8	9	2	14	1	5	2
Run sum	10	18	27	29	43	44	49	51

Step :2

Normalize by dividing by total number of pixels. The total number of pixels is **10+8+9+2+14+1+5+2=.51**

Gray level value	0	1	2	3	4	5	6	7
No. of Pixel	10	8	9	2	14	1	5	2
Run sum	10	18	27	29	43	44	49	51
Normalize	10/51	18/51	27/51	29/51	43/51	44/51	49/51	51/51

Step 3:

Multiply these values by the maximum gray level values in this case 7 , and then round the result to the closet integer. After this is done we obtain **1,2,4,4,6,6,7,7.**

Gray level value	0	1	2	3	4	5	6	7
No. of Pixel	10	8	9	2	14	1	5	2
Run sum	10	18	27	29	43	44	49	51
Normalize	10/51	18/51	27/51	29/51	43/51	44/51	49/51	51/51
Multiply by 7	1	2	4	4	6	6	7	7

Step 4:

Map the original values to the results from step 3 by a one-to-one correspondence.

Old	0	1	2	3	4	5	6	7
New	1	2	4	4	6	6	7	7