রাত 2:00am
11-11-2024
Monday

Image Compression

(1) Consider the simple 3x3, 8 bit image as (3,4,7) (3,4,7) (3,4,7) suppose you want to compress the image using a loss less lempel-ziv-wetch(LZW) fixed length coding algo.

(i) Generate your new code book using 9 bits and illustrate your steps LZW encoding process.

(ii) What kind of redundancy of image data it reduces in your encoding process?

(iii) Any compression achieved by employing LZW? Proof it.

Ans: (i)

For 8bit image codebook will be 9 bit.

3	4	7
3	4	7
3	4	7

Dictionary location	Entry
0	0
1	1
⋮	⋮
255	255
256	=
⋮	-
511	=

~~3×4×~~ 3 4 7
✓3 ✓4 ✓7
✓3 ✓4 7

Current Recognized Sequence	Pixel Being Processed	Encoded output [location]	Dictionary location (codeword)	Dictionary Entry
	3			
3	4	3	256	3-4 (3&4 combined)
4	7	4	257	4-7
7	3	7	258	7-3
3	4	—	—	—
3-4	7	256	259	3-4-7
7	3	—	—	—
7-3	4	258	260	7-3-4
4	7	—	—	—
4-7		257		

The output sequence [3-4-7-256-258-257]

Dictionary Location	Entry
0 : 255	0 : 255
256	3-4
257	4-7
258	7-3
259	3-4-7
260	7-3-4

(ii)

In my LZW coding process it reduces spatial data redundancy. It assigns fixed-length code words to variable length sequence of source symbol.

(iii)

Before Compression:

Total matrix value = $3 \times 3 = 9$

Original image size = 9×8 bit = 72 bit.

After Compression:

Total Encoded output = 6

Each encoded output → 9 bit

compressed image size = $9 \times 6 = 54$ bit

$$\text{Compression Ratio} = \frac{72}{54} = 1.33:1$$

Yes, compression achieved.

(2) (i) Why image compression needed?

- ✱ Data stored at less storage
- ✱ Data transmission at minimum cost

(ii) How many and what types of data redundancy are -

3 types:

✱ **Coding redundancy** → most 2D intensity array contain more bits than are needed to represent intensity.

✱ **Spatial and temporal redundancy** → Pixels of 2D intensity arrays are correlated spatially and vedio sequence are temporally correlated.

Irrelevant information: most 2D intensity arrays contain information that is ignored by the human visual system.

(iii) A 512×512 8 bit image with 5.3 bits/pixel entropy is to be Huffman coding. What is the maximum compression that can be expected?

⇒ Entropy = 5.3 bits/pixel

Means minimum 5.3 bits are required to represent a pixel without lossing any data/info.

Compression ratio = $\frac{512 \times 512 \times 8}{512 \times 512 \times 5.3} = 1.509:1$ expected.

This is the maximum compression ratio that can be

(3) 3×3, 8 bit image

18	16	16
11	11	14
11	14	16

① Illustrate your Huffman encoding process for image.

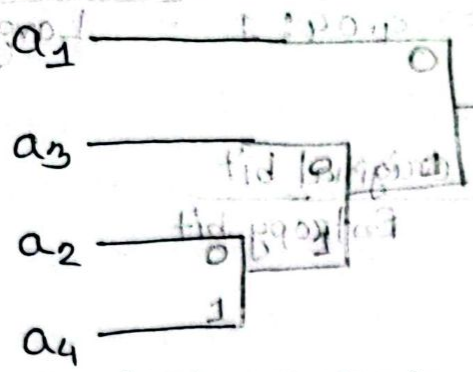
Ans:

Symbol	Pixel value	Probability (3×3=9)
a_1	11	3/9 = 0.33
a_2	14	2/9 = 0.22
a_3	16	3/9 = 0.33
a_4	18	1/9 = 0.11

[ascending order]

total count / total number

Deceending order:



Pixel	Symbol		Codeword
11	← a_1	—	0
16	← a_3	—	10
14	← a_2	—	110
18	← a_4	—	111

(11)

Compute Entropy of the above image.

Ans: Entropy $= \sum_{k=1}^{L-1} -P_r(r_k) \log_2\left(P_r(r_k)\right)$ L = # of intensity gray level

r_k = is input image intensity or gray level value k

$P_r(r_k)$ = normalized histogram of input image (Probability of r_k)

Entropy $= -(0.33)\log_2(0.33) - (0.22)\log_2(0.22) - (0.33)\log_2(0.33) - (0.11)\log_2(0.11)$

$= 1.758$ bits/pixel

~~minimum~~ each pixel need minimum 1.758 bit.

(111)

What kind of redundancy of image data it reduce in your encoding process? is it loss less?

Ans: It reduce the coding redundancy applying variable length code.

Average bit required per symbol,

$L_{avg} = \sum_{i=0}^{L} l_i P_i$ = Probability × number of bit

$= (0.33) \times \underset{11}{1} + (0.22) \times \underset{14}{3} + (0.33) \times \underset{16}{2} + (0.11) \times \underset{18}{3}$

$= 1.98$ bit/pixel

$H < L_{avg} \rightarrow$ so this compression is lossless.

(iv)

$$\text{Compression ratio} = \frac{8 \text{ bit}}{1.98 \text{ bit}} = 4.04 : 1 \qquad \frac{\text{original bit}}{\text{Lavg bit}}$$

$$\text{Maximum compression ratio} = \frac{\text{original bit}}{\text{Entropy bit}}$$

(4) 4×8 pixel image →

21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243

(i) Compute the entropy.

(ii) Calculate the representative Huffman codes for each pixel.

(iii) What is the compression ratio achieved by employing Huffman coding instead of 8bit fixed length coding?

(iv) Calculate the relative data redundancy of the the given 8 bit image.

(v) Compute the effectiveness of the huffman coding.

(iii)

(1)

$$\text{Entropy} = \sum_{k=1}^{L-1} - Pr(r_k) \log_2 (Pr(r_k))$$

total pixel = 4×8 = 32

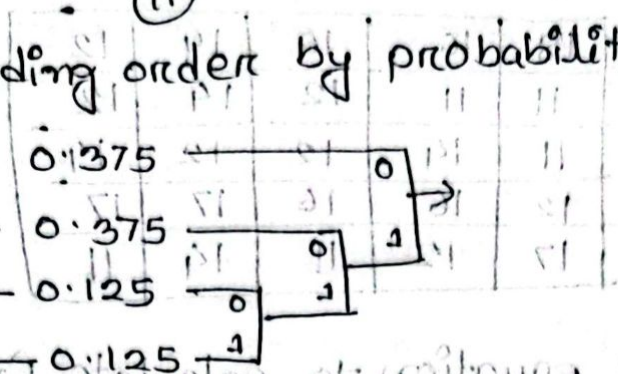
symbol	Pixel	Probability
a_1	21	$\rightarrow 12/32 = 0.375$
a_2	95	$\rightarrow 4/32 = 0.125$
a_3	169	$\rightarrow 4/32 = 0.125$
a_4	243	$\rightarrow 12/32 = 0.375$

$$\text{Entropy} = -[(0.375)\log_2(0.375) + (0.125)\log_2(0.125) + (0.125)\log_2(0.125) + (0.375)\log_2(0.375)]$$

$$= 1.811$$

(II)

Arrange deccending order by probability.



Symbol	Pixel	codeword
a_1	21	0 → 1bit
a_4	243	10 → 2bit
a_2	95	110 → 3bit
a_3	169	111 → 3bit

(III)

$$L_{avg} = \sum_{i=0}^{L} l_i P_i = (0.375) \times \underset{21}{1} + (0.375) \times \underset{243}{\overset{\text{bit}}{2}} + (0.125)\times 3 + (0.125)\times 3$$

$$= 1.875$$

Compression ratio, $C = \frac{8}{1.875} = 04.267 : 1$

(IV)

Relative Redundance, $R_D = 1 - \frac{1}{C} = 1 - \frac{1}{4.267} = 0.766$

= 76.55% data redundant.

(5) 5×5

18	16	16	19	12
11	11	12	14	12
11	14	12	12	14
12	16	16	17	17
17	12	11	14	11

① Derive the equation to calculate Entropy of an image from the information theory to understand the optimum required bit to represent image information.

Ans: Information Theory: If we consider the pixel intensity are as random events then the intensity histogram is approximation of the probabilities

$$H = \sum_{K=1}^{L-1} -P_r(r_k) \log_2 P_r(r_k)$$

where,

L = # of intensity or gray level

r_k = Input image intensity / gray level

$P_r(r_k)$ = Normalized histogram of input image.

Entropy is the measurement of the average information in an image. So it is not possible to encode input image with fewer than H bits/pixel.

(6) ①

Compute Suppose 3x3 of 3bit image is compressed by lossy compression technique, after de-compression, all the pixels value increased by 2. Calculate the Root-mean square Error as fidelity criteria.

② Compute the Golomb code for $G_4(9)$

Given image

$f(x,y)+2 = f'(x,y)$

1	4	3
2	3	4
2	4	5

3	6	5
4	5	6
4	6	7

① Each pixel of $f'(x,y)$ increased by 2.

Row = m = 3

col = N = 3.

$$e_{rms} = \sqrt{\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[\hat{f}(x,y)-f(x,y)\right]^2}$$

$$= \sqrt{\frac{1}{3\times3}(2^2)\times9}$$

$$= 2$$

[এখানে প্রত্যেক 2 করে বেড়েছে so প্রত্যেকটার difference 2 হবে and সবগুলোর square power নিলেও 4 থাকছে।]

② Golomb Coding: $G_m(n) = G_4(9)$ so $n=9$

m = divisor = 4.

Step 1: $\text{Floor}\left(\frac{n}{m}\right) = \text{Floor}\left(\frac{9}{4}\right) = 2.25 \approx 2 = q$

So unary code = 110.

Step-2: $k = \lceil \log_2 m \rceil$; $c = 2^k - m$; $r = n \bmod m$

and compute truncated reminder r' such that

$$r' = \begin{cases} r \text{ truncated to } k-1 \text{ bit if } 0 \le r < c \\ r+c \text{ truncated to } k \text{ bit otherwise.} \end{cases}$$

Now,

$$k = \lceil \log_2(4) \rceil = 2 \; ; \; c = 2^2 - 4 = 0 \; ; \; r = 9 \bmod 4 = 1$$

$0 \le r < c$ is not accepted.

$r' = r + c = 1 + 0 = 1 = 01$

~~truncate 1 bit from k bit.~~

Step-3: Concaded result from step 1 and 2.

110 01 ✓

Another: $G_4(7)$ $n=7$ $m=4$

① $q = G_4(7) = \lfloor \frac{7}{4} \rfloor = 1$ Unary code = 10

② $k = \lceil \log_2 4 \rceil = 2$; $c = 2^k - m = 2^2 - 4 = 0$; $r = n \bmod m = 7 \bmod 4 = 3$

$r' = r + c = 3 = 11 \rightarrow$ (binary)

③ 10 11 → concaded (1) (2).

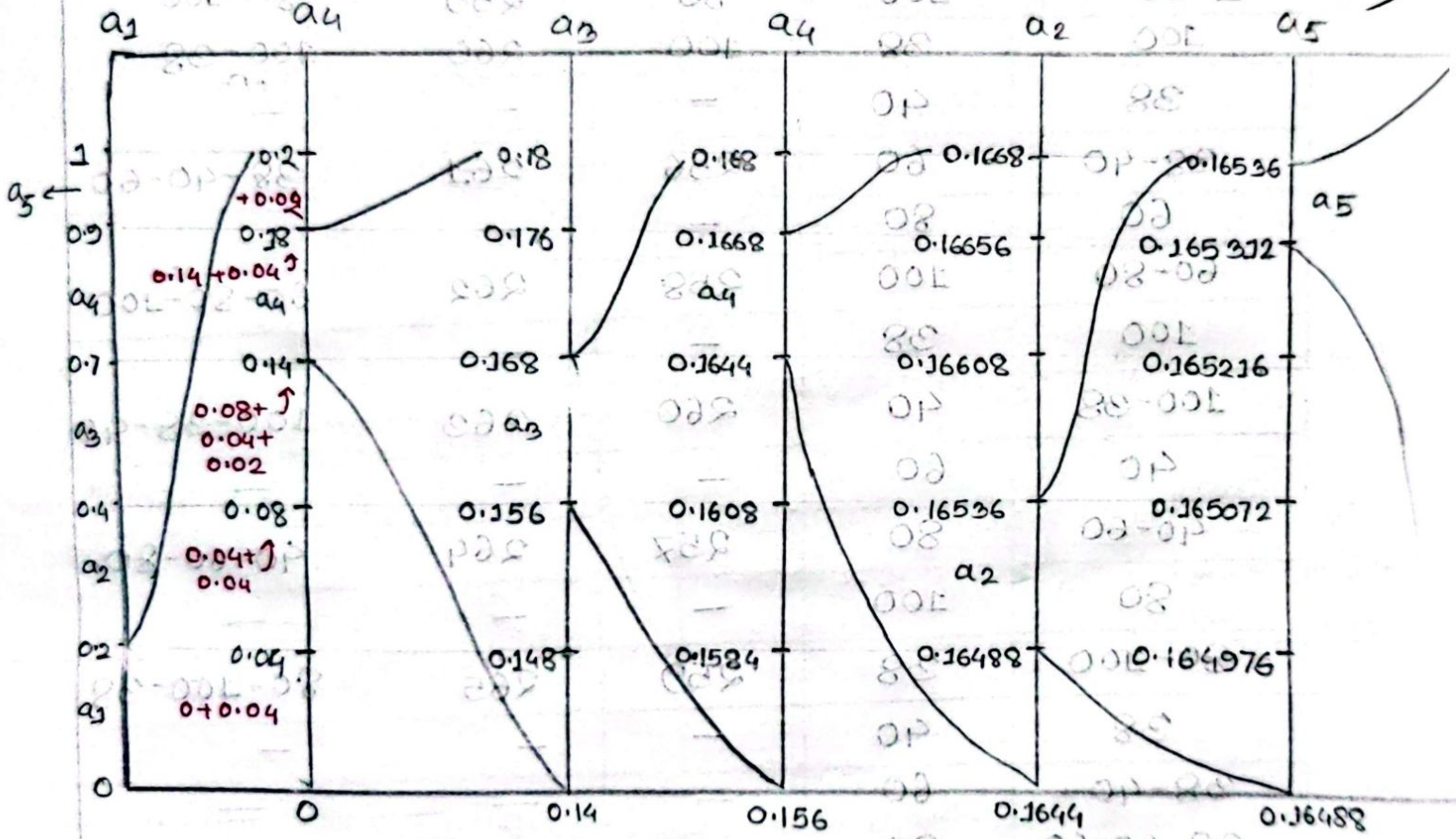
⑥ Consider The table-I below that represents the five symbol source with probabilities and initial subintervals What is the Arithmetic code for the sequence $a_1\ a_4\ a_3\ a_4\ a_2\ a_5$?
1 2 3 4 5 6

Source symbol	Probably	Initial sub Interval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.3	[0.4, 0.7)
a_4	0.2	[0.7, 0.9)
a_5	0.1	[0.9, 1.0)

(After table) calculation

$a_1 a_4 a_3 a_4 a_2 a_5$ → Arithmatic code

0.16536



$\frac{0.2}{5} = 0.04$

যাদের diff = 0.02

তারা বাড়বে

+0.04

আর diff = 0.03

0.04+0.02

diff = 0.01

+0.02

(0.18 − 0.14)

= 0.04

$= \frac{0.04}{5}$

$= 0.008/2$

$= 0.004$

d = 0.02;

+0.008

d = 0.03;

0.008+0.004

d = 0.01;

+0.004

(0.168 − 0.156)

$= 0.012/5$

$= 2.4 \times 10^{-3}$

$= 0.0024/2$

$= 0.0012$

(0.1668 − 0.1644)

= 0.0024

= 0.00048

= 0.00024

(0.16536 − 0.16488)

= 0.00048

$= 0.000096/2$

= 0.000048

⑦ use 9bit

38	40	60	80	100
38	40	60	80	100
38	40	60	80	100
38	40	60	80	100
38	40	60	80	100

Using LZW:

Current Recognized sequence ①	Pixel being Processed ②	Encoded output ③	Dictionary Location ⑤	Dictionary Entry ④
	38			
38	40	38	256	38-40
40	60	40	257	40-60
60	80	60	258	60-80
80	100	80	259	80-100
100	38	100	260	100-38
38	40	—	—	—
38-40	60	256	261	38-40-60
60	80	—	—	—
60-80	100	258	262	60-80-100
100	38	—	—	—
100-38	40	260	263	100-38-40
40	60	—	—	—
40-60	80	257	264	40-60-80
80	100	—	—	—
80-100	38	259	265	80-100-38
38	40	—	—	—
38-40	60	—	—	—
38-40-60	80	261	266	38-40-60-80
80	100	—	—	—
80-100	38	—	—	—
80-100-38	40	265	267	80-100-38-40
40	60	—	—	—
40-60	80	—	—	—
40-60-80	100	264	268	40-60-80-100

last → 100 — 100 — —

Location	Entry
0	0
⋮	⋮
255	255
~~38-40~~ 256	38-40
257	40-60
258	60-80
259	80-100
260	100-38
261	38-40-60
262	60-80-100
263	100-38-40
264	40-60-80
265	80-100-38
266	38-40-60-80
267	80-100-38-40
268	40-60-80-100
266 ⋮ 511	— —

Encoded output ~~কিভাবে~~ decode

5×5

38 40 60 80 100
38 40 60 80 100
38 40 60 80 100
38 40 60 80 100
38 40 60 80 100

Total output = 38, 40, 60, 80, 100
256, 258, 260, 257, 259, 261,
265, 264, 100.
= 14

Before Compression

Total output = 25
Bit size = 8
Image size = 25×8
= 200

After compression

Total output = ~~13~~ 14
Bit size = 9
output image size = ~~13×9~~ 14×9
= ~~117~~ 126

Compression ratio = $\frac{200}{\text{~~117~~ } 126}$ = ~~1.709~~ 1.59

Compression is achieved.

Chapter-3

Histogram

① 4×4 image with 5bit gray values

16	30	8	10
25	1	12	0
6	13	9	5
19	14	29	4

$r_k = 2$

0,1,4,5,6,8,9,10,12,13,14,16,19,25,29,30

$L = 2^5 = 32$

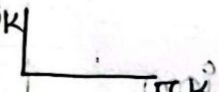
Ans: Histogram of an image is a graphical representation of the distribution of pixel intensities.

→ Compute Histogram → প্রতিটা pixel/Gray level কতবার আছে সেটা count করব। $h(r_k)$ ওখানে সবাই একবার আছে।

→ Normalize = $\frac{h(r_k)}{4\times4}$ = PDF

→ Equalize histogram বললে CDF and s_k, n_k

→ Transformation Curve



$s_k = (L-1) \times CDF$

Gray level (r_k)	Histogram $h(r_k)$	PDF = $\frac{h(r_k)}{4\times4}$	CDF = (PDF$_i$ + CDF$_{i-1}$)	$s_k = (L-1)\times CDF_i$ = 31 × CDF$_i$	no. of pixel n_k (output image)
0	1	1/16 = 0.0625	0.0625	1.9375	2	1
1	1	0.0625	0.125	3.875	4	1
4	1	0.0625	0.1875	5.8125	6	1
5	1	0.0625	0.25	7.75	8	1
6	1	0.0625	0.3125	9.6	10	1
8	1	0.0625	0.375	11.6	12	1
9	1	0.0625	0.4375	13.56	14	1
10	1	0.0625	0.5	15.5	16	1
12	1	0.0625	0.5625	17.4	17	1
13	1	0.0625	0.625	19.3	19	1
14	1	0.0625	0.6875	21.3	21	1
16	1	0.0625	0.75	23.2	23	1
19	1	0.0625	0.8125	25.18	25	1
25	1	0.0625	0.875	27.1	27	1
29	1	0.0625	0.9375	29.06	29	1
30	1	0.0625	1	31	31	1

Here, Gray level, L = 8 (0-7) $M \times N = 64 \times 64 = 4096$

$\sqrt{4096} = 64$

Gray level (r_k)	0	1	2	3	4	5	6	7
No of pixel	790	1023	850	656	329	245	122	81

$n = 790 + 1023 + 850 + 656 + 329 + 245 + 122 + 81 = 4096$

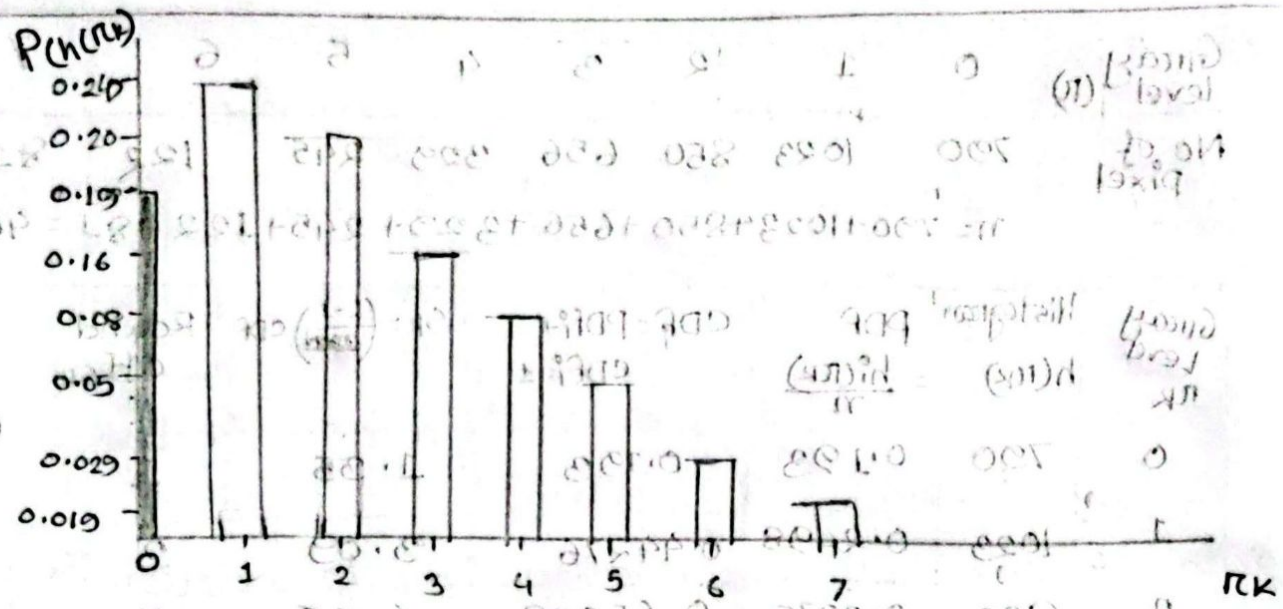
Gray Level r_k	Histogram $h(r_k)$	PDF $= \frac{h(r_k)}{n}$	CDF = PDF$_i$ + CDF$_{i-1}$	$S_k = \left(\frac{L-1}{M \times N}\right)$ CDF	Round off S_k	No of pixels. n_k (output image)
0	790	0.193	0.193	1.35	1	790
1	1023	0.2498	0.44276	3.09	3	1023
2	850	0.2075	0.65028	4.55	5	850
3	656	0.1602	0.8104	5.67	6	656+329 = 985
4	329	0.08032	0.89072	6.24	6	
5	245	0.0598	0.95054	6.65	7	245+122+81 = 448
6	122	0.0298	0.98033	6.86	7	
7	81	0.01978	1	7	7	

✱ যদি CDF → PDF value দিয়ে calculate করি তাহলে

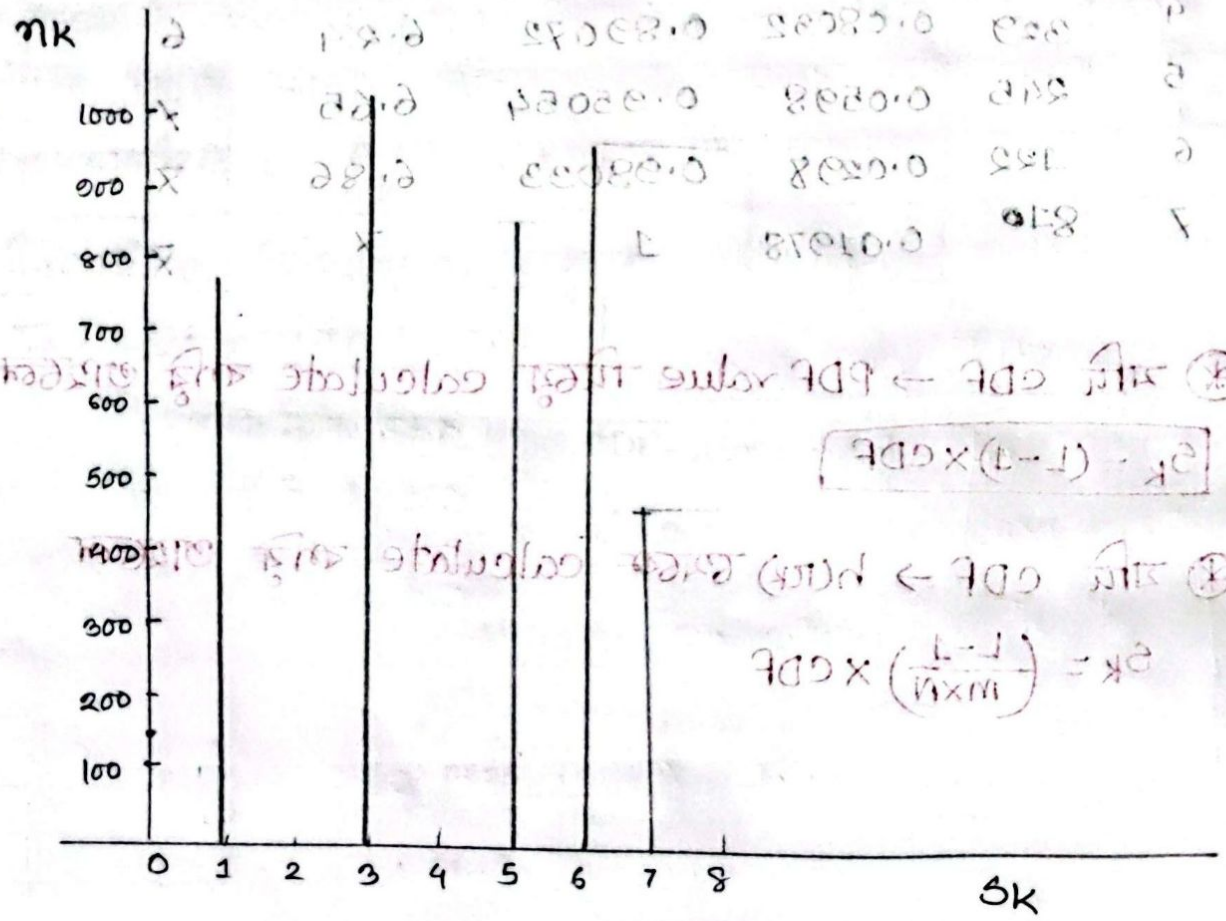
$$S_k = (L-1) \times CDF$$

✱ যদি CDF → $h(r_k)$ থেকে calculate করি তাহলে

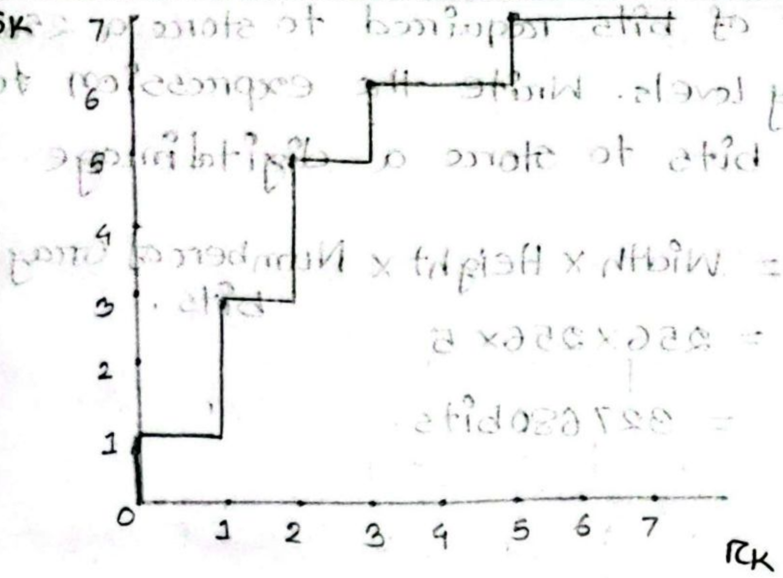
$$S_k = \left(\frac{L-1}{M \times N}\right) \times CDF$$



Graph-1: Normalized Graph.



Graph:2 Equalization.



Graph-3: Transition graph

(III) Show thathe mapping of the new gray level values into number of pixels.

Ans: So,

New Gray leve	→	# of pixel
0	→	0
1	→	790
2	→	0
3	→	1023
4	→	0
5	→	850
6	→	656+329
7	→	245+122+81
		4096.

Gray level	No of pixel	Sk
0	790	1
1	1023	3
2	850	5
3	656	6
4	329	6
5	245	7
6	122	7
7	81	7

এটা nk এর কলামটাই।

③ Find the number of bits required to store a 256x256 image with 32 gray levels. Write the expression to find the number of bits to store a digital image.

Ans: Number of bit = Width x Height x Number of Gray level bits.

$2^5 = 32$

$= 256 \times 256 \times 5$

$= 327680$ bits.

Decipher – 3(f)

(i) Suppose you have a 3x3 image with values as [2,5,8,5,1,3,4,7,2] How can you smooth this image using an average filter of size 3x3 and zero padding? Show your filtered output image with detailed calculations.

Ans: Main image with zero padding

0				
0	0	0	0	0
0	2	5	8	0
0	5	1	3	0
0	4	7	2	0
0	0	0	0	0

3x3 Avg Filter

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\text{Output} = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix}$$

$T(1,1) = \frac{1}{9}(0+0+0+0+2+5+0+5+1) = 1.44$

$T(1,2) = \frac{1}{9}(0+0+0+2+5+8+5+1+3) = 2.67$

$T(1,3) = \frac{1}{9}(0+0+0+5+8+0+1+3+0) = 1.89$

$T(2,1) = \frac{1}{9}(0+2+5+0+5+1+0+4+7) = 2.67$

$T(2,2) = \frac{1}{9}(2+5+8+5+1+3+4+7+2) = 4.11$

$T(2,3) = \frac{1}{9}(5+8+0+1+3+0+7+2+0) = 2.89$

$T(3,1) = \frac{1}{9}(0+5+1+0+4+7+0+0+0) = 1.89$

$T(3,2) = \frac{1}{9}(5+1+3+4+7+2+0+0+0) = 2.44$

$T(3,3) = \frac{1}{9}(1+3+0+7+2+0+0+0+0) = 1.44$

Output Image = $\begin{bmatrix} 1.44 & 2.67 & 1.89 \\ 2.67 & 4.11 & 2.89 \\ 1.89 & 2.44 & 1.44 \end{bmatrix}$

(ii) obtain the output image by applying a 3x3 median filter on a 4x4 image with values [0,2,1,0,5,2,3,8,0,1,3,4,3,0,7,2]. Ignore the border pixels during calculation and put zero in the boarder of the output image.

Ans:

0	2	1	0
5	2	3	8
0	1	3	4
3	0	7	2

① median → 0 0 1 1 ②2 3 3 5, n=odd → $\frac{n+1}{2} = \frac{9+1}{2} = 5$

$T_{22} \rightarrow 2$

② median → 0 1 1 2 ②3 3 4 8 $T_{23} \rightarrow 2$

③ median → 0 0 1 2 ③3 3 5 7 $T_{32} \rightarrow 3$

④ median → 0 1 2 2 ③3 4 7 8 $T_{33} \rightarrow 3$

3x3 median output

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & T_{22} & T_{23} & 0 \\ 0 & T_{32} & T_{33} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

(iii) Derive a positive laplacian filter mask to enhance an image in a single filtering operation.

⇒ Positive laplacian = $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

When we apply Laplacian filter the output image is just about the edges of original picture. So to get enhanced picture by using Laplacian filter we have to add (for negative) or substract (for positive Lap) with/from the original image.

But these two step can be done 1 step.

Enhanced imge, $g(x,y) = f(x,y) + \nabla^2 f$

If center neg, (original − 2nd derivative)

$= f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)]$

$= f(x,y) - f(x+1,y) - f(x-1,y) - f(x,y+1) - f(x,y-1) + 4f(x,y)$

$= 5f(x,y) - f(x+1,y) - f(x-1,y) - f(x,y+1) - f(x,y-1)$

$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ = modified filter

So now we can enhance an image with modified filter in a single step.

⊛ original picture * $\begin{vmatrix}0&0&0\\0&1&0\\0&0&0\end{vmatrix}$ → no change.

filter
identical filter

⊛ original picture * $\begin{vmatrix}0&0&0\\0&0&1\\0&0&0\end{vmatrix}$ → Total picture will shift left by 1 pixel

⊛ original * $\left(\begin{vmatrix}0&0&0\\0&2&0\\0&0&0\end{vmatrix} - \frac{1}{9}\begin{vmatrix}1&1&1\\1&1&1\\1&1&1\end{vmatrix}\right)$ → Sharpening filter, Emphasize difference with local average.

low pass filter

Smoothing filter

1. Standard average / mean → $\frac{1}{9}\begin{vmatrix}1&1&1\\1&1&1\\1&1&1\end{vmatrix}$
2. Weighted average → $\frac{1}{16}\begin{vmatrix}1&2&1\\2&4&2\\1&2&1\end{vmatrix}$
3. Median filter
   - ↳ non linear

high pass

Sharpening Spatial Filter

1. Laplacian → use of 2nd derivative for image Enhancement
2. SOBEL → use of 1st derivative for image Enhancement. (Gradient operator)

$\begin{vmatrix}0&1&0\\1&-4&1\\0&1&0\end{vmatrix}$ pos-Lap $\quad \begin{vmatrix}0&-1&0\\-1&4&-1\\0&-1&0\end{vmatrix}$ neg-Lap $\quad \frac{\partial f}{\partial y} = \begin{vmatrix}-1&-2&-1\\0&0&0\\1&2&1\end{vmatrix}$ Gy $\quad \frac{\partial f}{\partial x} = \begin{vmatrix}-1&0&1\\-2&0&2\\-1&0&1\end{vmatrix}$ Gx (90° of Gy).

3. Prewitt → $\begin{vmatrix}-1&-1&-1\\0&0&0\\1&1&1\end{vmatrix}$ Gy $\quad \begin{vmatrix}-1&0&1\\-1&0&1\\-1&0&1\end{vmatrix}$ Gx

④ What is the objective of sharping image? 3bit image of size 5by5 in the square. calculate the gradient magnitude and angle using sobel mask at the highlighted center pixel.

0	2	6	7	3	
1	1	6	4	2	
4	5	(2)	7	4	
1	2	6	0	3	
2	1	5	7	5	

Ans:

→ Remove Blurring from image

→ Highlight edges.

→ Useful for emphasizing transitions in image intensity.

Using Sobel: Gx:

$$\begin{vmatrix} 1 & 6 & 4 \\ 5 & 2 & 7 \\ 2 & 6 & 0 \end{vmatrix} \times \begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix} = \begin{vmatrix} -1 & 0 & 4 \\ -10 & 0 & 14 \\ -2 & 0 & 0 \end{vmatrix}$$

$$Gx = -1+0+4-10+0+14-2+0+0 = 5$$

Gy:

$$\begin{vmatrix} 1 & 6 & 4 \\ 5 & 2 & 7 \\ 2 & 6 & 0 \end{vmatrix} \times \begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{vmatrix} = \begin{vmatrix} -1 & -12 & -4 \\ 0 & 0 & 0 \\ 2 & 12 & 0 \end{vmatrix}$$

$$Gy = +1+12-4+0+0+0+2+12+0 = -3$$

Gradient magnitude, $G = |Gx| + |Gy| = |5| + |-3|$

$$\boxed{G = 8}$$

$$\theta = \tan^{-1}\frac{Gy}{Gx} = \tan^{-1}\frac{-3}{5}$$

$$\boxed{\theta = -30.96}$$

⑤ Define the Bit Plane slicing method. What are the 3 main goals of bit plane slicing? Consider

$I = (150, 60), (60, 210)$ as a 2×2 image with 8 bit gray values.

(i) Give 8 bit planes of I

(ii) Determine the reconstructed image using bit planes 8 & 7

Ans: Bit plane slicing: is a method of representing an image with one or more bits of the byte used for each pixel.

One can only use MSB to represent the pixel which reduce the original gray level to a binary image.

3 main goals:

1. Converting Gray level image to binary image.
2. Representing image with fewer bits and that's why the image smaller size.
3. Enhancing image by focusing main characteristics.

(i)

$$\begin{matrix} 150 & 60 \\ 60 & 210 \end{matrix} \rightarrow \begin{vmatrix} 10010110 & 00111100 \\ 00111100 & 11010010 \end{vmatrix}$$

(8 7 6 5 4 3 2 1)

$$2^0 \times \begin{vmatrix} 0 & 0 \\ 0 & 0 \end{vmatrix}; \underset{\text{2nd}}{2^1 \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}}; \underset{\text{3rd}}{2^2 \begin{vmatrix} 1 & 1 \\ 1 & 0 \end{vmatrix}}; \underset{\text{4th}}{2^3 \begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix}}; \underset{\text{5th}}{2^4 \begin{vmatrix} 1 & 1 \\ 1 & 1 \end{vmatrix}}; \underset{\text{6th}}{2^5 \begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix}}$$

(1st)

$$\underset{\text{7th}}{2^6 \begin{vmatrix} 0 & 0 \\ 0 & 1 \end{vmatrix}}; \underset{\text{8th}}{2^7 \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}}$$

(11)

Reconstructed image using bit plane 8 and 7

$$2^7 \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} + 2^6 \begin{vmatrix} 0 & 0 \\ 0 & 1 \end{vmatrix} = \begin{vmatrix} 128 & 0 \\ 0 & 128 \end{vmatrix} + \begin{vmatrix} 0 & 0 \\ 0 & 64 \end{vmatrix}$$

$$= \begin{vmatrix} 128 & 0 \\ 0 & 192 \end{vmatrix}$$