

ତାର୍କା 2:00am

11-11-2024  
Monday

## Image Compression

- Q) Consider the simple  $3 \times 3$ , 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-Welch(LZW) fixed length coding algo.
- Generate your new code book using 9 bits and illustrate your steps LZW encoding process.
  - What kind of redundancy of image data it reduces in your encoding process?
  - Any compression achieved by employing LZW?  
Proof it.

Ans:

i

For 8bit image codebook will be 9bit.

3	9	7
3	9	7
3	4	7

Dictionary location

Entry

1

1

2

2

3

3

4

4

5

5

6

6

7

7

8

8

9

9

10

10

11

11

12

12

13

13

14

14

15

15

[828-828-828-255] of compressed string will

$$\begin{array}{c} 3 \times 4 \times \\ \sqrt{3} \quad \sqrt{4} \quad \sqrt{7} \\ \sqrt{3} \quad \sqrt{4} \quad \sqrt{7} \\ \sqrt{3} \quad \sqrt{4} \quad \sqrt{7} \end{array}$$

Current Recognized Sequence	Pixel Being Processed	Encoded Output [location]	Dictionary Location (code word)	Dictionary Entry
3-4-7	3	256	256	3-4-1
3-4-7	4	257	257	4-7
3-4-7	7	258	258	7-3
3-4-7	-	-	-	-
3-4-7	256	259	259	3-4-7
3-4-7	-	-	-	-
3-4-7	258	260	260	7-3-4
3-4-7	-	-	-	-
3-4-7	257	257	257	7-3-4

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

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

In my LZW coding process it reduces spatial data redundancy.

It assigns fixed-length code words to variable length sequence of source symbols.

### Before compression,

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

Original image size =  $9 \times 8 \text{ bit} = 72 \text{ bit}$

### After Compression

Total Encoded output =  $6 \times 6 = 36$

Each encoded output  $\rightarrow 6 \text{ bit}$

Compressed image size =  $9 \times 6 = 54 \text{ bit}$

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

Yes, compression achieved.

② Why image compression needed?

\* Data storage at low storage

\* Data transmission at minimum cost

① How many and what types of data redundancy are -

3 types:

\* Coding redundancy  $\rightarrow$  most 2D intensity array contain more bits than are needed to represent intensity.

\* Spatial and temporal redundancy  $\rightarrow$  Pixels of 2D intensity arrays are correlated spatially and video sequence are temporally correlated.

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

- (iii) A  $512 \times 512$  8 bit image with 5.3 bits/pixel entropy is to be Huffman Coding. What is the maximum compression that can be expected?

$$\Rightarrow \text{Entropy} = 5.3 \text{ bits/pixel}$$

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

$$\text{Compression ratio} = \frac{512 \times 512 \times 8}{512 \times 512 \times 5.3}$$

This is the maximum compression ratio that can be expected.

- (3)  $3 \times 3$ , 8 bit image

- ① Illustrate your Huffman encoding process for image.

Ans: Symbol — Pixel value — Probability ( $3 \times 3 = 9$ )

$$a_1 \quad 11 \quad \frac{1}{9} = 0.33$$

$$a_2 \quad 14 \quad \frac{2}{9} = 0.22$$

total count
total number

$$a_3 \quad 16 \quad \frac{3}{9} = 0.33$$

$$a_4 \quad 18 \quad \frac{1}{9} = 0.11$$

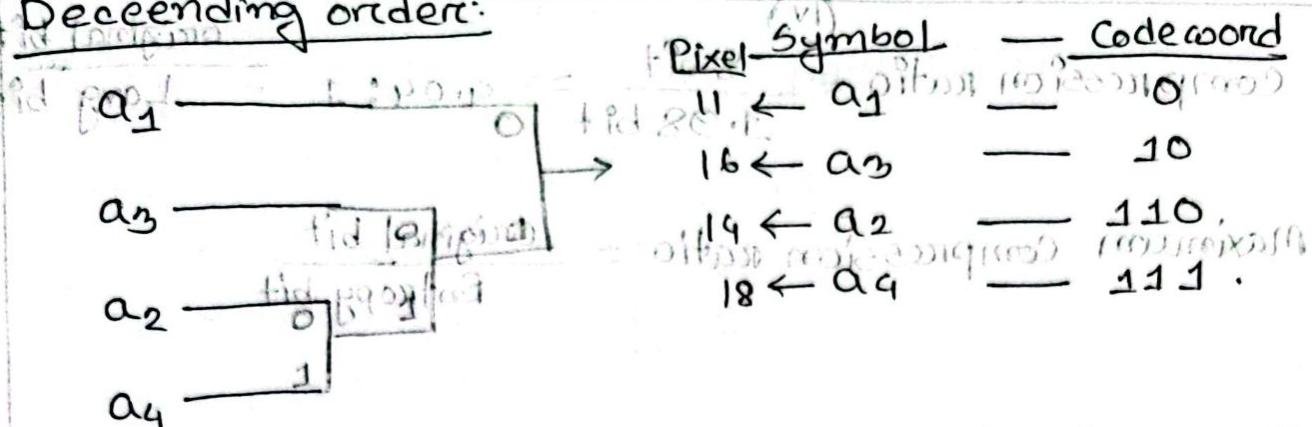
Sort the symbols in ascending order of probability.

Priority code to assign depends on the order of symbols.

Priority code to assign depends on the order of symbols.

Priority code to assign depends on the order of symbols.

Decreasing order:



→ Appr. 10x9 8xp (1)

(1)

Compute Entropy of the above image.

$$\text{Ans} \rightarrow \text{Entropy} = \sum_{k=1}^{L-1} -P_k(r_k) \log_2(P_k(r_k)) \quad L = \# \text{ of intensity gray level}$$

$$\text{Entropy} = -(0.33) \log_2(0.33) + -(0.22) \log_2(0.22) + -(0.33) \log_2(0.33) + -(0.11) \log_2(0.11)$$

Intensity or gray level value k

$= 1.758 \text{ bits/pixel}$

→ Appr. 10x9 histogram of input image (Probability of r<sub>k</sub>)

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,

$$L_{avg} = \sum_{i=0}^L L_i P_i = \text{Probability} \times \text{number of bit}$$

$$= (0.33) \times \frac{1}{16} + (0.22) \times \frac{3}{16} + (0.33) \times \frac{2}{16} + (0.11) \times \frac{3}{16}$$

$$= 1.98 \text{ bit/pixel}$$

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

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

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

④ 4x8 pixel image →

210	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

- ① Compute the entropy.
- ② Calculate the representative Huffman Codes for each pixel.
- ③ What is the compression ratio achieved by employing Huffman Coding instead of 8bit fixed length coding?

④ calculate the relative data redundancy of the given 8bit image.

⑤ Compute the effectiveness of the huffman coding.

$$\text{Entropy} = \sum_{k=1}^{L-1} -P_k(\pi_k) \log_2(P_k(\pi_k)) \quad \text{① Huffman symbol - pixel - Probability}$$

$$\text{total pixel} = 4 \times 8 = 32 \quad \begin{array}{l} a_1 = 21 \rightarrow 12/32 = 0.375 \\ a_2 = 95 \rightarrow 4/32 = 0.125 \\ a_3 = 169 \rightarrow 4/32 = 0.125 \\ a_4 = 243 \rightarrow 12/32 = 0.375 \end{array}$$

$$\text{Entropy} = -[(0.375) \log_2(0.375) + (0.125) \log_2(0.125) + (0.125) \log_2(0.125) + (0.375) \log_2(0.375)] = 1.811$$

$$= 1.811$$

fixed 8bit =

encoded dimensionless 1.811 < 8bit > H



(5)

5x5

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 have a source consider the pixel intensity are as random events then the intensity histogram is approximation of the probabilities of

whence,

$$H = \sum_{k=1}^{L-1} -P_{\pi}(r_k) \log_2 P_{\pi}(r_k) \quad (i) \quad L = \# \text{ of intensity or gray levels}$$

$r_k = \text{Input image intensity / gray level}$

$P_{\pi}(r_k) = \text{Normalized histogram}$

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.

prob.  $\leq 2^{-H}$   
threshold

⑥ ①

Compute Suppose  $(3 \times 3)$  of 3 bit 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 criterion.

② Compute the Golomb code for  $G_{4,2}(g)$

Given image

①

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

$f(x,y) + 2 = f'(x,y)$
$\begin{array}{ c c c } \hline 1 & 4 & 3 \\ \hline 2 & 3 & 4 \\ \hline 2 & 4 & 5 \\ \hline \end{array} \quad \begin{array}{ c c c } \hline 3 & 6 & 5 \\ \hline 4 & 5 & 6 \\ \hline 4 & 6 & 7 \\ \hline \end{array}$

$$\text{Row} = m = 3$$

$$\text{col} = N = 3$$

$$e_{\text{rms}} = \sqrt{\frac{1}{mN} \sum_{x=0}^{m-1} \sum_{y=0}^{N-1} [f(x,y) - f'(x,y)]^2}$$

$$= \sqrt{\frac{1}{3 \times 3} (2^2) \times 9}$$

$$= 2$$

যেখানে প্রতিটি 2 বাটু অঢ়ছে সো প্রতিটি difference 2 হবে and তার square power শিখবে গুরুতর।

$$DE = \text{Root mean square} \quad E = \left[ \frac{1}{n} \right] = (2)^{1/2} = \sqrt{2}$$

$$\text{Golomb Coding: } G_m(n) = G_{m,2}(g) \quad \text{where } g = \frac{n}{m} \quad \text{and } m = \text{divisor} = 4.$$

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

$$\text{so unary code} = 110.$$

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

and compute truncated remainder  $r'$  such that  
 $r \leq r' < c$

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

Now,  $n = 2 + (r \cdot 2^k)$

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

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

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

~~truncate 1 bit from k bit~~

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

**110 01**

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

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

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

$$r' = r+c = 3 = 11 \rightarrow \text{Binary}$$

$$\textcircled{3} \quad 10\ 11 \rightarrow \text{concatenated } (1)(2).$$



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 Recognize Sequence (1)	Pixel being Processed (2)	Encoded Output (3)	Dictionary Location (4)	Dictionary Entry (5)
[38]	38			
(38, 40)	40	38	256	38-40
(38, 40, 60)	60	40	257	40-60
(38, 40, 60, 80)	80	60	258	60-80
(38, 40, 60, 80, 100)	100	80	259	80-100
(38, 40, 60, 80, 100)	38	100	260	100-38
(38, 40, 60, 80, 100)	40	-	-	-
(38-40, 60, 80, 100)	60	256	261	38-40-60
(38-40, 60, 80, 100)	80	-	-	-
(38-40, 60, 80, 100)	100	258	262	60-80-100
(38-40, 60, 80, 100)	38	-	-	-
(100-38, 40, 60, 80, 100)	40	260	263	100-38-40
(100-38, 40, 60, 80, 100)	60	-	-	-
(40-60, 80, 100, 38, 40)	80	257	264	40-60-80
(40-60, 80, 100, 38, 40)	100	-	-	-
(80-100, 38, 40, 60, 80)	38	259	265	80-100-38
(80-100, 38, 40, 60, 80)	40	-	-	-
(38-40-60, 80, 100, 38, 40)	60	-	-	-
(38-40-60, 80, 100, 38, 40)	80	261	266	38-40-60-80
(38-40-60, 80, 100, 38, 40)	100	-	-	-
(80-100, 38, 40, 60, 80)	38	-	-	-
(80-100, 38, 40, 60, 80)	40	265	267	80-100-38-40
(80-100, 38, 40, 60, 80)	60	-	-	-
(40-60, 80, 100, 38, 40)	80	-	-	-
(40-60, 80, 100, 38, 40)	100	264	268	40-60-80-100

Last → 100 - 100 - -

Location - Entry	
0	0
255	255
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
269	-
270	-
271	-

### Before compression

$$\text{Total output} = 25 \times 8 \times (30 + 30) = 7200$$

$$\text{Bit size} = 8$$

$$\text{Image size} = 25 \times 8$$

$$= 200$$

$$\text{Compression ratio} = \frac{200}{7200}$$

$$= \frac{1}{36}$$

Compression is achieved.

01	01	00000
13	01	01000
03	01	00100
05	01	00010
07	01	00001
09	01	00000
11	01	00000

Encoded output after decode

000000	000001	000010	000011	000100	000101	000110	000111	001000	001001	001010	001011	001100	001101	001110	001111	010000	010001	010010	010011	010100	010101	010110	010111	011000	011001	011010	011011	011100	011101	011110	011111
000000	000001	000010	000011	000100	000101	000110	000111	001000	001001	001010	001011	001100	001101	001110	001111	010000	010001	010010	010011	010100	010101	010110	010111	011000	011001	011010	011011	011100	011101	011110	011111
000000	000001	000010	000011	000100	000101	000110	000111	001000	001001	001010	001011	001100	001101	001110	001111	010000	010001	010010	010011	010100	010101	010110	010111	011000	011001	011010	011011	011100	011101	011110	011111
000000	000001	000010	000011	000100	000101	000110	000111	001000	001001	001010	001011	001100	001101	001110	001111	010000	010001	010010	010011	010100	010101	010110	010111	011000	011001	011010	011011	011100	011101	011110	011111
000000	000001	000010	000011	000100	000101	000110	000111	001000	001001	001010	001011	001100	001101	001110	001111	010000	010001	010010	010011	010100	010101	010110	010111	011000	011001	011010	011011	011100	011101	011110	011111

Total output = 38, 40, 60, 80, 100

256, 258, 260, 257, 259, 261,

265, 264, 100.

= 14

= 14

### After compression

$$\text{Total output} = 14$$

$$\text{BTSIZE} = 14$$

$$\text{Output image size} = 13 \times 9$$

$$= 117 \times 126$$

## Chapter - 3

### Histogram

①

16	30	8	10	4x4 image with 5bit gray values
25	1	12	0	
6	13	9	5	$r_{ik} = 2$
19	4	29	4	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 →  $\text{প্রতিটি pixel/Gray level ক্ষেত্রে}$   
আছে ক্ষেত্রে count করুয়া।  $h(r_{ik})$  ক্ষেত্রে অবস্থার আছে।

$$\rightarrow \text{Normalize} = \frac{h(r_{ik})}{4 \times 4} = \text{PDF}$$

→ Equalize histogram বলতে CDF and  $s_k, n_k$

→ Transformation Curve

$$s_k = (L-1) \times \text{CDF}$$

Gray level ( $r_{ik}$ )	Histogram $h(r_{ik})$	$\text{PDF} = \frac{h(r_{ik})}{4 \times 4}$	$\text{CDF} = (\text{PDF}_1 + \text{CDF}_{l-1})$	$s_k = (L-1) \times \text{CDF}_i$	No. of pixel $n_k$
0	1	$1/16 = 0.0625$	0.0625	1.935	2
1	1	0.0625	0.125	3.875	4
4	1	0.0625	0.1875	5.8125	6
5	1	0.0625	0.25	7.75	8
6	1	0.0625	0.3125	9.6	10
8	1	0.0625	0.375	11.6	12
9	1	0.0625	0.4375	13.56	14
10	1	0.0625	0.5	15.5	16
12	1	0.0625	0.5625	17.4	17
13	1	0.0625	0.625	19.3	19
14	1	0.0625	0.6875	21.3	21
16	1	0.0625	0.75	23.2	23
19	1	0.0625	0.8125	25.18	25
25	1	0.0625	0.875	27.1	27
29	1	0.0625	0.9375	29.06	29
30	1	0.0625	1	31	31

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 $= \frac{h(r_k)}{n}$	$CDF = PDF^i + CDF^{i-1}$	$s_k = \frac{(L-1)}{m \times N} CDF$	Round off $s_k$	No of pixels · 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	4.56	5	850
3	656	0.1602	0.8109	5.67	6	$656 + 329 = 985$
4	329	0.08032	0.89072	6.24	6	
5	245	0.0598	0.95054	6.65	7	$245 +$
6	122	0.0298	0.98033	6.86	7	$122 + 81 = 448$
7	81	0.01978	1	7	7	

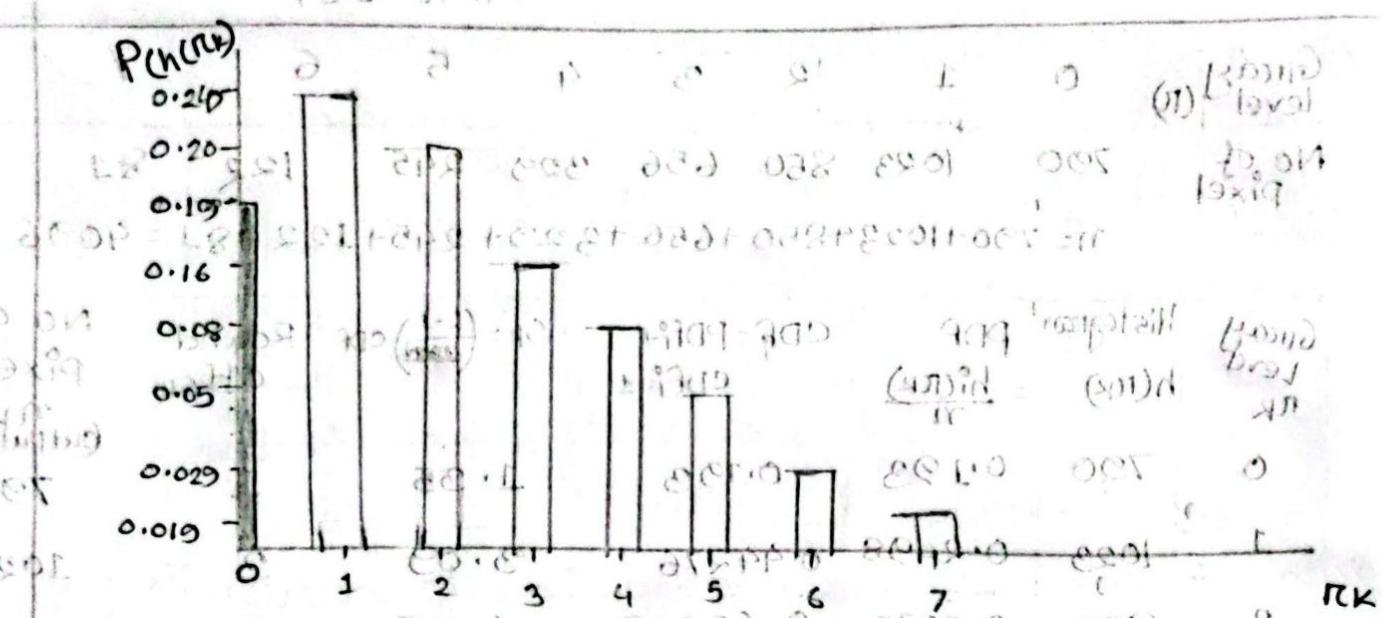
\* যদি  $CDF \rightarrow PDF$  value দিয়ে calculate করা হয়ে গেলে

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

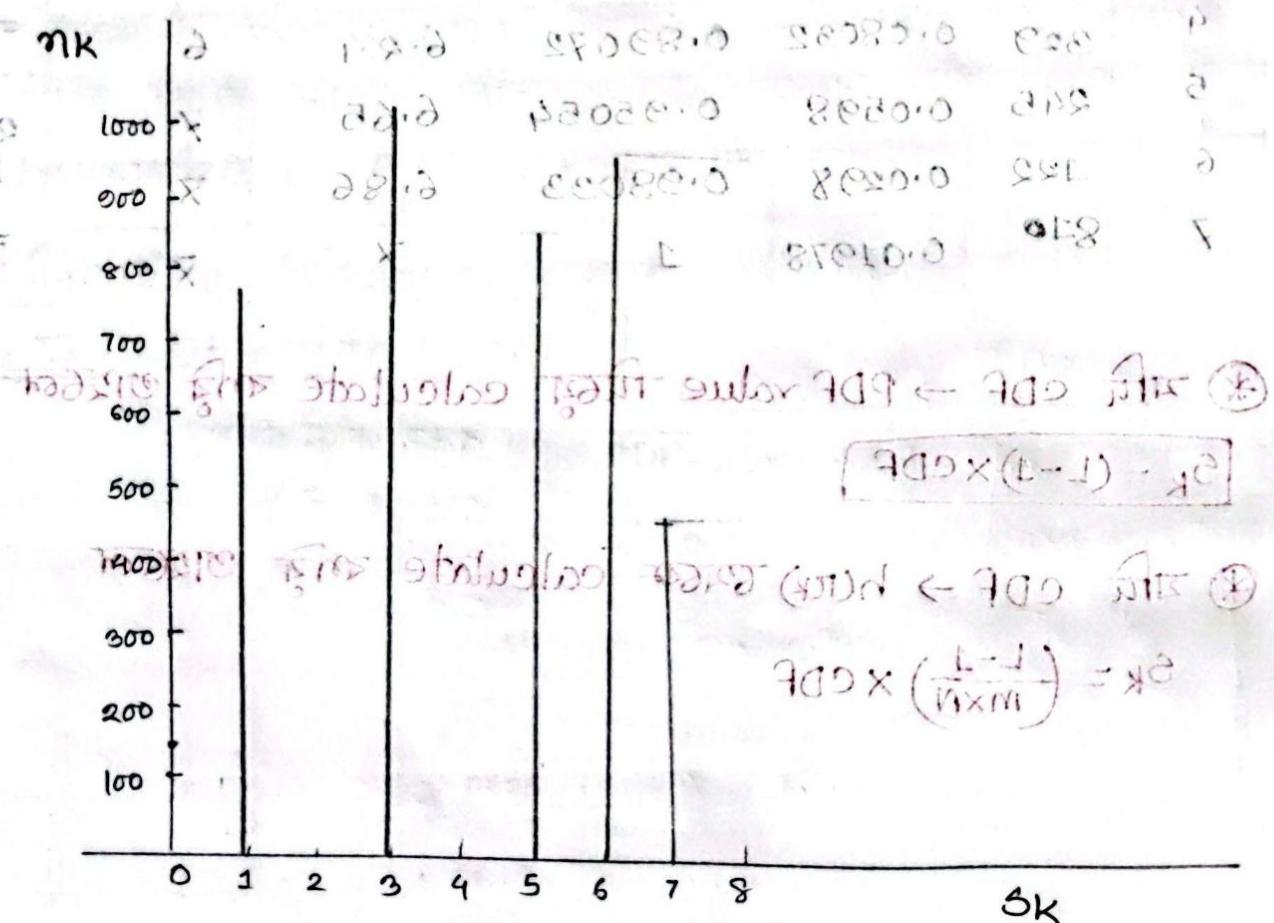
\* যদি  $CDF \rightarrow h(r_k)$  দিয়ে calculate করা হয়ে গেলে

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

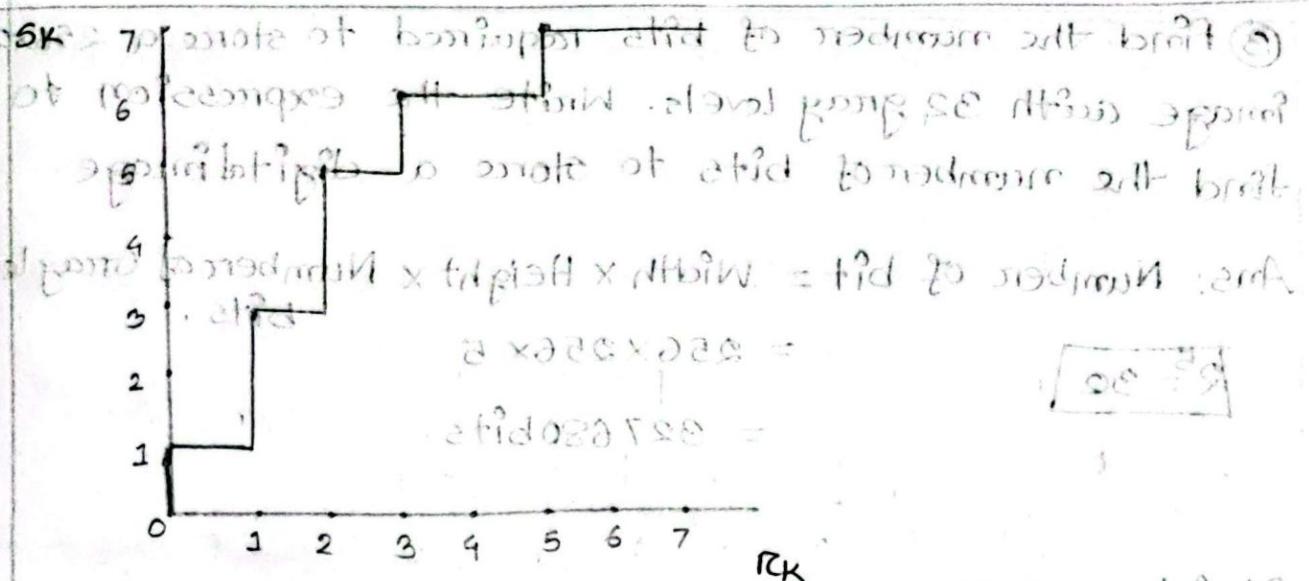
there (final level) =  $E = \frac{1}{2}mv^2$  (Eq. 8)



## Graph - 1: Normalized Graph



## Graphs 2 Equalization .



(b) 8 - mark question

### Graph-3: Transition graph

- (iii) Show that the mapping of the new gray level values into number of pixels.

Ans 50, Gray level  $\rightarrow$  # of pixels

Gray level	# of pixels
0	790
1	790
2	0
3	1023
4	0
5	850
6	656 + 329
7	245 + 122 + 81

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

$$\text{Ans. } 4096 \times (0.0625 + 0.0625 + 0.0625 + 0.0625 + 0.0625 + 0.0625 + 0.0625) = 0.625$$

$$\text{Ans. } 8 = (0.0625 + 0.0625 + 0.0625 + 0.0625 + 0.0625 + 0.0625 + 0.0625) \frac{1}{16} = 0.625$$

$$\text{Ans. } = (0.0625 + 0.0625 + 0.0625 + 0.0625 + 0.0625 + 0.0625 + 0.0625) \frac{1}{16} = 0.625$$

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

Ans: Number of bit = Width  $\times$  Height  $\times$  Number of Gray level bits.

$$R^5 = 32$$

$$= 256 \times 256 \times 5$$

$$= 327680 \text{ bits.}$$

### Decipher - 3(f)

(i) Suppose you have a  $3 \times 3$  image with values as

~~dimensions~~  $[2, 5, 8, 5, 1, 3, 4, 7, 2]$  How can you smooth this image using an average filter of size  $3 \times 3$  and zero padding? Show your filtered output image with detailed calculations.

Ans: Main image with zero padding

	0.00	0.00	0.00
0.00	0.00	0.00	0.00
0.25	0.25	0.80	0.00
0.50	0.50	1.30	0.00
0.75	0.75	2.00	0.00
1.00	1.00	2.00	0.00
1.25	1.25	2.00	0.00
1.50	1.50	2.00	0.00
1.75	1.75	2.00	0.00
2.00	2.00	2.00	0.00

$3 \times 3$  Avg Filter

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

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

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

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

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

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

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

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

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

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

$$T(3,3) = \frac{1}{9} (5+1+3+0+7+2+0+0+0) = 1.44$$

total of 9 pixels having 3x3 window to apply soft boundary

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

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

Ans:

$$\begin{bmatrix} 0 & 2 & 1 & 0 \end{bmatrix}$$

$3 \times 3$  median output

$$(B_{11}-x)^2 - (B_{11}-5)^2 - (B_{11}-2)^2 - (B_{11}-3)^2 -$$

$$(B_{12}-x)^2 - (B_{12}-5)^2 - (B_{12}-2)^2 - (B_{12}-3)^2 -$$

$$\textcircled{1} \quad \begin{bmatrix} 0 & 1 & 3 & 4 \\ 3 & 0 & 7 & 2 \end{bmatrix}$$

$$\text{Median} \rightarrow 0 \textcircled{1} 1 \textcircled{2} 2 3 3 5 \quad n=7 \text{ odd} \rightarrow \frac{n+1}{2} = \frac{7+1}{2} = 5$$

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

$$T_{22} \rightarrow 2 (L-B-x)^2 - (L+B+x)^2 -$$

$$\textcircled{2} \text{ median} \rightarrow 0 1 1 2 \textcircled{3} 3 4 8 \quad T_{23} \rightarrow 2$$

$$\textcircled{3} \text{ median} \rightarrow 0 0 1 2 \textcircled{3} 3 3 5 7 \quad T_{32} \rightarrow 1 3$$

$$\textcircled{4} \text{ median} \rightarrow 0 1 2 2 \textcircled{3} 3 4 7 8 \quad T_{33} \rightarrow 3$$

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

border pixels of the original image are zeroed out

gate off the  $n^2$  matrix

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

$$\Rightarrow \text{Positive Laplacian} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

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 subtract (for positive lap) with from the original image.

But these two step can be done in one step.

$$\begin{aligned} \text{Enhanced image, } g(x,y) &= f(x,y) + \nabla^2 f \\ &= f(x,y) - [f(x+1,y) + f(x-1,y) \\ &\quad + f(x,y+1) + f(x,y-1) - 4f(x,y)] \\ &= f(x,y) - f(x+1,y) - f(x-1,y) \\ &\quad - f(x,y+1) - f(x,y-1) + 4f(x,y) \\ &= 5f(x,y) - f(x+1,y) - f(x-1,y) \\ &\quad - f(x,y+1) - f(x,y-1) \end{aligned}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

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

= modified filter

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

\* original picture  $\xrightarrow{\text{filter}} \text{no change}$

original picture  $\xrightarrow{\text{filter}} \text{Total picture will shift left by 1 pixel}$

~~original  $\xrightarrow{\left( \begin{array}{ccc} 0 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 0 \end{array} - \frac{1}{9} \left| \begin{array}{ccc} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array} \right| \right)}$  Sharpening filter~~

Emphasize difference with local average.

low pass filter

Smoothing filters

1. Standard average / mean  $\rightarrow$

$$\frac{1}{9} \left| \begin{array}{cccc} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{array} \right|$$

$$\left| \begin{array}{ccc} 2 & 3 & 4 \\ 5 & 6 & 7 \\ 8 & 9 & 10 \end{array} \right|$$

2. Weighted average  $\rightarrow$

$$\frac{1}{16} \left| \begin{array}{cccc} 1 & 4 & 2 & 1 \\ 2 & 4 & 2 & 1 \\ 1 & 2 & 4 & 2 \\ 1 & 2 & 4 & 2 \end{array} \right|$$

3. median filter

↳ non linear

Sharpening spatial filter

1. Laplacian  $\rightarrow$  use of 2nd derivative for image enhancement

2. Sobel  $\rightarrow$  use of 1st derivative for image enhancement.

(Gradient operators)  $|G_x| + |G_y| = \sqrt{G_x^2 + G_y^2}$  (sharpening threshold)

$$\left| \begin{array}{ccc} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{array} \right|$$

$$\left| \begin{array}{ccc} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{array} \right|$$

pos-Lap

neg-Lap

$$\frac{\partial f}{\partial y} = \left| \begin{array}{ccc} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{array} \right|$$

$$\frac{\partial f}{\partial x} = \left| \begin{array}{ccc} -1 & 0 & 1 \\ -2 & 0 & -2 \\ 1 & 0 & 1 \end{array} \right|$$

$$= \left| \begin{array}{ccc} -1 & 0 & 1 \\ -2 & 0 & -2 \\ 1 & 0 & 1 \end{array} \right| = 0$$

$$G_y = 0 \quad G_x (90^\circ \text{ of } G_y)$$

3. Prewitt  $\rightarrow$

$$\left| \begin{array}{ccc} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{array} \right|$$

Gy

$$\left| \begin{array}{ccc} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{array} \right|$$

Gx

high pass

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

Ans: Fixing Edges & Edges

→ Remove Blurring from image

→ Highlight edges.

→ Useful for emphasizing transitions in image intensity.

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

Using Sobel: Gx:

1	6	9
5	2	7
2	6	0

-1	0	1
-2	0	2
-1	0	1

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 9 \\ -4 & 10 & 14 \\ -2 & 0 & 0 \end{bmatrix}$$

$$G_x = -1 + 0 + 4 - (-2) + 0 + 14 - 2 + 0 + 0 = 5$$

Gy:

$$\begin{bmatrix} 1 & 6 & 9 \\ 5 & 2 & 7 \\ 2 & 6 & 0 \end{bmatrix} \times \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} -1 & -12 & -9 \\ 0 & 0 & 0 \\ 2 & 12 & 0 \end{bmatrix}$$

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

Gradient magnitude,  $G = \sqrt{|G_x|^2 + |G_y|^2} = \sqrt{5^2 + (-3)^2} = \sqrt{34} = 5.86$

$$\theta = \tan^{-1} \frac{G_y}{G_x} = \tan^{-1} \frac{-3}{5} = -30.96^\circ$$

$$G = 5.86$$

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

← It's a 3x3

3x3

B

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  $2 \times 2$  image with 8 bit gray values.

(i) Give 8 bit planes of I

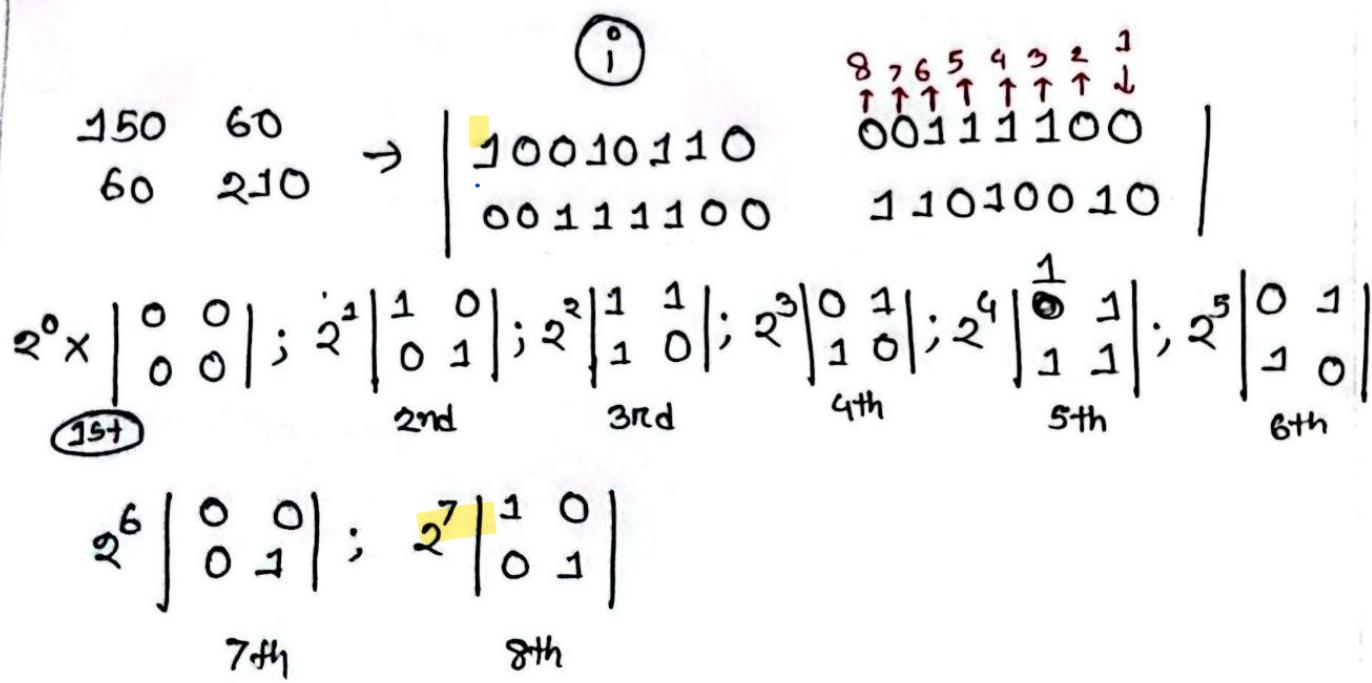
(ii) Determine the reconstructed image using bit planes 8 & 7

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.



Reconstructed image using bit plane 8 and 7

$$2^7 \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} + 2^6 \begin{vmatrix} 0 & 0 \\ 0 & 1 \end{vmatrix} = \begin{vmatrix} 128 & 0 \\ 0 & 128 \end{vmatrix} + \begin{vmatrix} 0 & 64 \\ 0 & 64 \end{vmatrix} = \begin{vmatrix} 128 & 0 \\ 0 & 192 \end{vmatrix}$$

Neighbors less than or equal to 128 are set to white  
Neighbors greater than 128 are set to black

• Generating binary image from grayscale image  
• Reconstruction of binary image from grayscale image

• Edge detection using Sobel operator

$$\begin{matrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{matrix} \quad \textcircled{1} \quad \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} \quad \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} \quad \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$$

$$\begin{matrix} 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 1 \\ 2 & 0 & -2 & 0 & 2 & 0 & -2 & 0 & 2 \\ 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 1 \end{matrix} \quad \textcircled{2} \quad \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

$$\begin{matrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{matrix} \quad ; \quad \begin{matrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$$