Image Processing

Sampling & Quantization:-

→ In order to become suitable for digital processing, an image function must be digitized both, spatially & in amplitude. This digitization process involves two main process called

i, Sampling – Digitizing the co-ordinate value is called sampling.

ii, Quantization – Digitizing the amplitude value is called quantization.

Relationship b/w Pixels :-

1, Neighbourhood of Pixels –

Any image is represented as :

(c)

(y+1)	(x-1, y+1)	(x, y+1)	(x+1, y+1)
(y)	(x-1, y)	(x, y)	(x+1, y)
(y-1)	(x-1, y-1)	(x, y-1)	(x+1, y-1)
	(x-1)	(x)	(x+1)

⇒ $N_8(P)$ 8-Neighbours :- Union of 4-Neigh & Diagonal Neigh

⇒ $N_8(P) = N_4(P) + N_D(P)$

a, $N_4(P)$ 4-neighbours : the set of horizontal & vertical neighbours.

	(x, y+1)	
(x-1, y)	(x, y)	(x+1, y)
	(x, y-1)	

b, $N_D(P)$ Diagonal neighbours : The set of 4 Diagonal Neighbours

(x-1, y+1)		(x+1, y+1)
	(x, y)	
(x-1, y-1)		(x+1, y+1)

2, Connectivity / Adjacency :-

Two pixels that are neighbours & have the same gray level are adjacent.

a, 4-adjacency & 8-adjacency

Binary Image

0	1	0	1	→ 8-adjacency
0	0	$\boxed{1}$	0	→ 4-adjacency
0	0	1	0	
1	0	0	0	

V = {1}

Gray-Scale Image

54	10	20	8	→ 8
81	150	$\boxed{2}$	34	→ 4
201	20	3	35	
7	40	14	56	

V = {1, 2, ..., 10}
(Connection can be made with those elements in the set)

b, m-adjacency (mixed adj.)

→ In mixed adjacency the elements can connect with both 8 & 4-adj.

→ Then a single image has ~~multi~~ multiple connections, redundancy will occur / ~~du~~ duplicats.

→ So when a element can connect with 4 & 8 then the priority is more for 4-adjacency.

Example:-

0	1	1		0	1	
0	1	0		0	1	0
0	0	1		0	0	1

(without priority)

0	1	1	→ 4 → ignored 8-adjacency
0	1	0	→ 4
0	0	1	→ 8

2, Connectivity / Adjacency :-

Two pixels that are neighbours & have the same gray level are adjacent.

a, 4- adjacency & 8- adjacency

Binary Image

0	1	0	1
0	0	$\boxed{1}$	0
0	0	1	0
1	0	0	0

→ 8-adjacency

→ 4-adjacency

V = {1}

Gray-Scale Image

54	10	20	8
81	150	$\boxed{2}$	34
201	20	3	35
7	40	14	56

→ 8

→ 4

V = {1, 2, ...10}

(Connection can be made with those Elements in the set)

b, m-adjacency (mixed adj.)

→ In mixed adjacency the elements can connect with both 8 & 4-adjacency.

→ Then a single image has ~~multi~~ multiple connections, redundancy will occur / ~~duplicats~~ duplicats.

→ So when a element can connect with 4 & 8 then the priority is more for 4-adjacency.

Example:-

0	1	1		0	1	1
0	1	0		0	1	0
0	0	1		0	0	1

(without priority)

0	1	1
0	1	0
0	0	1

→ 4

→ ignored 8-adjacency

→ 4

→ 8

Distance measures b/w pixels :-

1, Euclidean Distance :- $D_e(P,q) = [(x-s)^2 + (y-t)^2]^{1/2}$

2, City block Distance :- $D_4(P,q) = |x-s| + |y-t|$

4-adjacency

3, Chess board Distance :-

$$D_8(P,q) = \max[|x-s|, |y-t|]$$

4, D_m Distance :- It is defined as the shortest m-path b/w the points. This distance b/w 2 pixels depends on the values of the pixels along the path as well as the values of their neighbours.

Note :- The D_4 & D_8 Distances b/w p & q are independent of any paths that might exist b/w the points because these distances involve only the coordinates of the points.

Q An image segment is shown below. Let V be the set of gray level values used to define connectivity in the image. Compute D_4, D_8 & D_m Distances b/w pixels 'p' & 'q' for : i, V = {2, 3}
ii, V = {2, 6}

(0,0) 2 (p)	3	2	6	1
6	2	3	6	2
5	3	2	3	5
2	4	3	5	2
4	5	2	3	6 (q) (4,4)

Coordinates of $p(x,y) = (0,0)$

$q(s,t) = (4,4)$

$\Rightarrow \boxed{D_4(P,q) = |x-s| + |y-t|}$

$= |0-4| + |0-4|$

$\boxed{D_4 = 8 \text{ units}}$

$D_8 = \max(|x-s|, |y-t|)$

$= \max(|0-4| + |0-4|)$,

$= \max(4, 4)$

$\boxed{D_8 = 4 \text{ units}}$

D_m, given 2 sets (i) & (ii) cases.

i, $V = \{2, 3\}$

(p) 2	3	2	6	1	
6	2	3	6	2	
5	3	2	3	5	
2	4	3	5	2	
4	5	2	3	6	(q)

⇒ There is no path b/w p & q for the given set, which not included 6 (q).

⇒ No path.

ii, $V = \{2, 6\}$

(p) 2	3	2	6	1	
6	2	3	6	2	
5	3	2	3	5	
2	4	3	5	2	
4	5	2	3	6	(q)

⇒ There is no path b/w p & q.

Q2: Consider the following image segment. Compute D_m, D_4 & D_8 distances b/w pixels 'p' & 'q' for $V = \{0, 1\}$ where V is the set of gray level values used to define connectivity. Repeat for $V = \{1, 2\}$

	3	1	2	1 (q)
	0	2	0	2
	1	2	1	1
(p)	1	0	1	2

A) co-ordinates of $q=(3,3)$ (s,t)

$p=(0,0)$ (x,y)

$D_4 = |x-s| + |y-t|$

$= |0-3| + |0-3|$

$\boxed{D_4 = 6 \text{ units}}$

$D_8 = \max(|x-s| + |y-t|)$

$= \max(|-3|, |-3|)$

$= \max(3, 3)$

$\boxed{D_8 = 3 \text{ units}}$

D_m, i, $V = \{0, 1\}$

$$\begin{array}{cccc} 3 & 1 & 2 & 1\ (q) \\ 0 & 2 & 0 & 2 \\ 1 & 2 & 1 & 1 \\ (p)\ 1 & 0 & 1 & 2 \end{array}$$

∴ path exist. $\boxed{D_m = 5 \text{ units}}$

ii, $V = \{1, 2\}$

$$\begin{array}{cccc} 3 & 1 & 2 & 1 \\ 0 & 2 & 0 & 2 \\ 1 & 2 & 1 & 1 \\ 1 & 0 & 1 & 2 \end{array}$$

∴ path exist $\boxed{D_m = 6 \text{ units}}$

Arithmetic Operations b/w Images

$s(x, y) = f(x,y) + g(x,y)$

$d(x,y) = f(x,y) - g(x,y)$

$p(x,y) = f(x,y) * g(x,y)$

$v(x,y) = f(x,y) \div g(x,y)$

Note:-

- If the result is a floating point numbers, round off its value.
- If the result is above the pixel range, Select the max range value.
- If the result is below the pixel range, Select the min range value.
- If the result is infinity, write it as zero.

Addition:-

$$\overset{A}{\begin{bmatrix} 0 & 100 & 10 \\ 4 & 0 & 10 \\ 8 & 0 & 5 \end{bmatrix}} + \overset{B}{\begin{bmatrix} 10 & 100 & 5 \\ 2 & 0 & 0 \\ 0 & 10 & 10 \end{bmatrix}} = \begin{bmatrix} 10 & 200 & 15 \\ 6 & 0 & 10 \\ 8 & 10 & 15 \end{bmatrix}$$

Range → 0-255

Uses:-

- Addition of noisy images for noise reduction
- Image averaging in the field of astronomy.

Subtraction:-

below the rage 0-255 then its '0'

$$\overset{A}{\begin{bmatrix} 0 & 100 & 10 \\ 4 & 0 & 10 \\ 8 & 0 & 5 \end{bmatrix}} - \overset{B}{\begin{bmatrix} 10 & 100 & 5 \\ 2 & 0 & 0 \\ 0 & 10 & 10 \end{bmatrix}} = \begin{bmatrix} 0 & 0 & 5 \\ 2 & 0 & 10 \\ 8 & 0 & 0 \end{bmatrix}$$

Uses:

- Enhancement of differences b/w images.
- mask mode radiography in medical imaging.

multiplication:

multiply pixel wise

above the range

$$\begin{bmatrix} 0 & 100 & 10 \\ 4 & 0 & 10 \\ 8 & 0 & 5 \end{bmatrix} * \begin{bmatrix} 10 & 100 & 5 \\ 2 & 0 & 0 \\ 0 & 10 & 10 \end{bmatrix} = \begin{bmatrix} 0 & 255 & 50 \\ 8 & 0 & 0 \\ 0 & 0 & 50 \end{bmatrix}$$

Uses:

- Shading correction
- masking (or) region of interest (RoI) operation

Division

$$\begin{bmatrix} 0 & 100 & 10 \\ 4 & 0 & 10 \\ 8 & 0 & 5 \end{bmatrix} / \begin{bmatrix} 10 & 100 & 5 \\ 2 & 0 & 0 \\ 0 & 10 & 10 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 2 \\ 2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Logical Operations on Images:-

⇒ AND, OR & NOT

i, AND:-

A	B	o/p
0	0	0
0	1	0
1	0	0
1	1	1

$$\underset{A}{\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}} \text{ and } \underset{B}{\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

ii, OR:-

A	B	o/p
0	0	0
0	1	1
1	0	1
1	1	1

$$A \text{ OR } B = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

iii, NOT:-

A	o/p
0	1
1	0

A NOT B =

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

Point Operations:- They are a method of Image processing in which Each pixel in the output image is only dependent upon the corresponding pixel in the input image & is independent of its location or neighbouring Pixels. Let 'r' be the gray value at a point (x,y) of the input image $f(x,y)$ & 's' be the gray value at a point (x,y) of the output image $g(x,y)$, then point operation can be defined as:

$S = T(r)$

where T is the point operation of a certain gray-level mapping relationship b/w the original image & the output image.

$$\begin{bmatrix} 4 & 3 & 5 & 2 \\ 3 & 6 & 4 & 6 \\ 2 & 2 & 6 & 5 \\ 7 & 6 & 4 & 1 \end{bmatrix}$$

1, Digital negative:-

$$s = (L-1) - r$$

First finding the highest 2 power pixel.

$\Rightarrow 2^0 = 1$

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

but the value was $6 \Rightarrow 2^{3}$

$\Rightarrow L = 8$

$n = 3$

Assuming the image to be 3-bit, we have $2^3 = 8$ levels

$n = 3$

$L = 2^n = 8$

$r = 0;\ s = (8-1) - 0$

$= 7$

$r = 1;\ s = 6$

$r = 2;\ = 5$

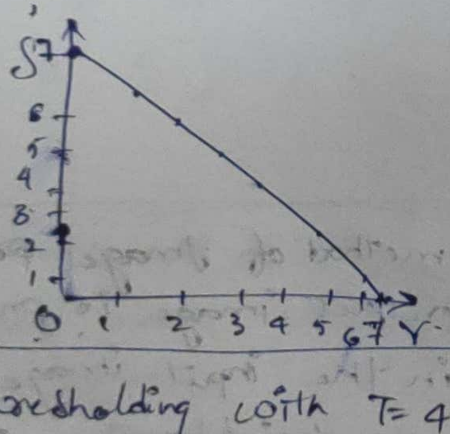
⋮

$r = 7;\ = 0$

The highest interval value.

$$\Rightarrow \begin{bmatrix} 3 & 4 & 2 & 5 \\ 4 & 1 & 3 & 1 \\ 5 & 5 & 1 & 2 \\ 0 & 1 & 3 & 6 \end{bmatrix}$$

Modifide but Digital negative



2, Thresholding with T = 4

$L = 8$

$L - 1 = 7$

$$s = \begin{cases} L-1 = 7; & r \geq 4 \\ 0; & r < 4 \end{cases}$$

$$\begin{bmatrix} 4 & 3 & 5 & 2 \\ 3 & 6 & 4 & 6 \\ 2 & 2 & 6 & 5 \\ 7 & 6 & 4 & 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} r = 0, 1, 2, 3, \longrightarrow s = 0 \\ r = 4, 5, 6, 7 \rightarrow s = 7 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 7 & 0 & 7 & 0 \\ 0 & 7 & 7 & 7 \\ 0 & 0 & 7 & 7 \\ 7 & 7 & 7 & 0 \end{bmatrix}.$$

output image

3. Clipping with $\boxed{r_1 = 2 \ \& \ r_2 = 5} \rightarrow$ range of r

$L = 8$

$L-1 = 7$

$$s = \begin{cases} L-1 ; & 2 \le r \le 5 \\ 0 ; & \text{otherwise} \end{cases}$$

$$\begin{bmatrix} r = 0, 1, 6, 7 \rightarrow s = 0 \\ r = 2, 3, 4, 5 \rightarrow s = 7 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 7 & 7 & 7 & 7 \\ 7 & 0 & 7 & 0 \\ 7 & 7 & 0 & 7 \\ 0 & 0 & 7 & 0 \end{bmatrix}$$

output image

4. Bit-plane slicing:

If $n = 3 \Rightarrow$ Convert all ~~decim~~ decimal pixels to binary

0 - 000
1 - 001
2 - 010
3 - 011
4 - 100
5 - 101
6 - 110
7 - 111

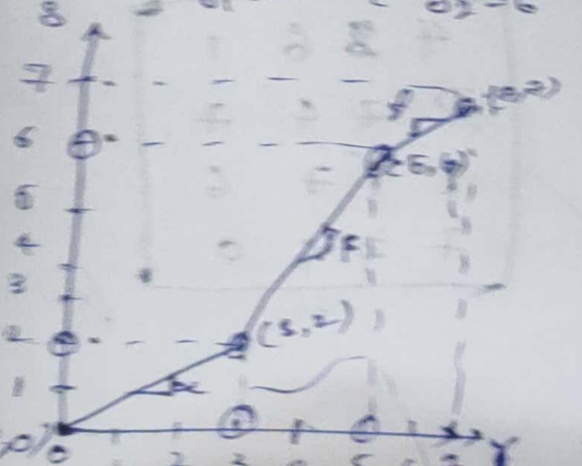
$$\Rightarrow \begin{bmatrix} 100 & 010 & 101 & 010 \\ 011 & 110 & 100 & 110 \\ 010 & 010 & 110 & 101 \\ 111 & 110 & 100 & 001 \end{bmatrix} //$$

5. Contrast Stretching

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

Given: $r_1 = 3 \ \& \ r_2 = 5$

$s_1 = 2 \ \& \ s_2 = 6$



$$\Rightarrow \text{Slope} \ \boxed{m = \frac{Y_2 - Y_1}{x_2 - x_1}}$$

$$\alpha = \frac{y_2 - y_1}{x_2 - x_1} = \frac{2-0}{3-0} = 0.66$$

$(x_1, y_1) = (0,0)$

$(x_2, y_2) = (3,2)$

$$\beta = \frac{\cancel{8} - 2}{5-3} = \frac{\cancel{4}^{2}}{\cancel{2}} = 2$$

$(x_1 y_1) = (3,2)$

$(x_2, y_2) = (5,6)$

$$\gamma = \frac{7-6}{7-5} = \frac{1}{2} = 0.5$$

$(5,6)$

$(7,7)$

6.39

$$S = \begin{cases} \alpha \times r & 0 \le r < 3 \\ \beta \times (r - r_1) + S_1 & 3 \le r < 5 \\ \gamma \times (r - r_2) + S_2 & 5 \le r \le 7 \end{cases}$$

$r_1 = 3$

$r_2 = 5$

$S_1 = 2$

$S_2 = 6$

$$\Rightarrow \begin{bmatrix} \alpha \times r = 0, 1, 2 \\ \beta(r - r_1) + S_1 = 3, 4 \\ \gamma(r - r_2) + S_2 = 5, 6, 7 \end{bmatrix}$$

r	
0	$\boxed{S = \alpha \times r} = 0.66 \times 0 = 0$
1	$= 0.66 \times 1 = 0.66 \approx 1$
2	$= 0.66 \times 2 = 1.32 \approx 1$
3	$\boxed{S = \beta(r - r_1) + S_1} = 2(3-3) + 2 = 2$
4	$= 2(4-3) + S_1 = 2(1) + 2 = 4$
5	$\boxed{S = \gamma(r - r_2) + S_2} = 0.5(5-5) + 6 = 6$
6	$= 0.5(6-5) + 6 = 6.5 \approx 7$
7	$= 0.5(7-5) + 6 = 7$

$\Rightarrow$ The final output image:

$$\begin{bmatrix} 4 & 2 & 6 & 1 \\ 2 & 7 & 4 & 7 \\ 1 & 1 & 7 & 6 \\ 7 & 7 & 4 & 0 \end{bmatrix}$$

6, Intensity level slicing with $r_1 = 3$ & $r_2 = 5$

i, without background — clipping

ii, with background

i, Input image:

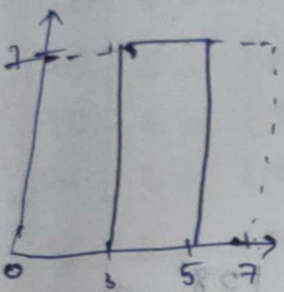
4	3	5	2
3	6	4	6
2	2	6	5
7	6	4	1

$$s = \begin{cases} L-1 = 7 \ ; & 3 \le r \le 5 \\ 0 \ ; & \text{otherwise} \end{cases}$$

$r = 0, 1, 2, 6, 7 \quad s \to 0$

$r = 3, 4, 5 \quad s \to 7$

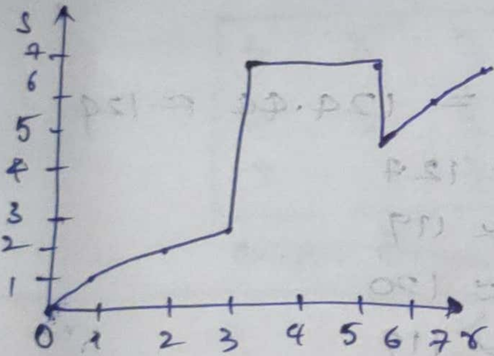
∴ The output Image:



7	7	7	0
7	0	7	0
0	0	0	7
0	0	7	0

ii, Same input Image

$$s = \begin{cases} L-1 = 7 \ ; & 3 \le r \le 5 \\ r \ ; & \text{Otherwise} \end{cases}$$



⇒

7	7	7	2
7	6	7	6
2	2	6	7
7	6	7	1

7, logarithmic Transformation:

$$\boxed{s = c.\log(1+r)}$$

where c is a constant & it is assumed that $r \geq 0$

110	120	90
91	94	98
90	91	99

$2^7 = 128$ 0–128

$L = 128$

$n = 7$

i, $c = 1$

ii, $c = \dfrac{L}{\log_{10}(1+L)}$

i, $\underline{C=1}$

r	s
110	$s = C\log(1+r) = 1\times\log(1+110) = \log(111) = 2.04 \approx 2$
120	$s = 1\times\log(1+120) = \log(121) = 2.08 \approx 2$
90	$s = 1\times\log(1+90) = \log(91) = 1.95 \approx 2$
91	$s = 1\times\log(1+91) = \log(92) = 1.96 \approx 2$
94	$s = 1\times\log(1+94) = \log(95) = 1.95 \approx 2$
98	$s = 1\times\log(1+98) = \log(99) = 1.99 \approx 2$
99	$s = 1\times\log(1+99) = \log(100) = 2 \approx 2$

Output Image:

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

ii, $C = \dfrac{L}{\log_{10}(1+L)} = \dfrac{128}{\log_{10}(1+128)} = \dfrac{128}{\log_{10}(129)} = \dfrac{128}{2.11} = 60.66$
≈ 61

$\Rightarrow$

r	
110	$s = C\times\log(1+r) = 61\times 2.04 = 124.44 \approx 124$
120	$s = 61\times 2.08 = 126.88 \approx 127$
90	$s = 61\times 1.95 = 118.95 \approx 119$
91	$s = 61\times 1.96 = 119.56 \approx 120$
94	$s = 61\times 1.95 = 118.95 \approx 119$
98	$s = 61\times 1.99 = 121.39 \approx 121$
99	$s = 61\times 2 = 122$

$$\begin{bmatrix} 124 & 127 & 119 \\ 120 & 119 & 121 \\ 119 & 120 & 122 \end{bmatrix}$$

Output Image.

Power-law transformation :

$$s = c r^{\gamma}$$

: c & γ are positive constants.

Given

$$\boxed{\begin{aligned} c &= 1 \\ \gamma &= 0.2 \end{aligned}}$$

$s = 1 \times r^{0.2}$
$= (r)^{0.2}$

· Same Input Image

r	s
110	$s = 1 \times (r)^{0.2} = 1 \times (110)^{0.2} = 2.56 \approx 3$
120	$s = (120)^{0.2} = 2.60 \approx 3$
90	$s = (90)^{0.2} = 2.45 \approx 2$
91	$s = (91)^{0.2} = 2.46 \approx 2$
94	$s = (94)^{0.2} = 2.48 \approx 2$
98	$s = (98)^{0.2} = 2.50 \approx 3$
99	$s = (99)^{0.2} = 2.50 \approx 3$

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

output Image.

Image Enhancement :-

The process of manipulating the image so that the result is more suitable than the original for specific applications

Techniques of Image Enhancement

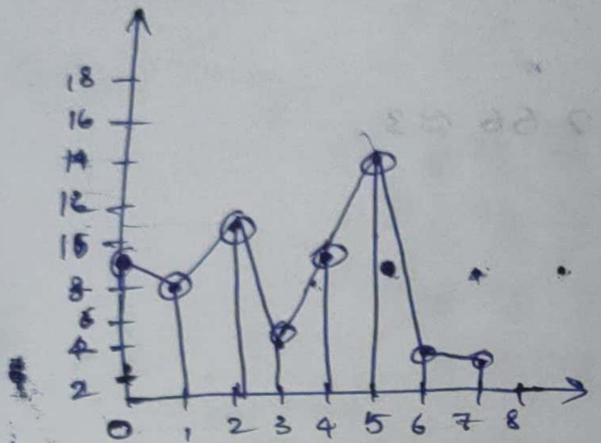
- **Spatial domain** – Involves direct manipulation of Pixels.
- **Frequency domain** – Involves modifying the fourier transform of an Image
- **Combination** – Involves a combination of spatical & frequency domains

① Spatial Domain

i. Histogram Equalization :-

Q. Perform the histogram equalization for an 8×8 image shown below.

Gray levels	0	1	2	3	4	5	6	7	(r_k)
No of pixels	9	8	11	4	10	15	4	3	(n_k)

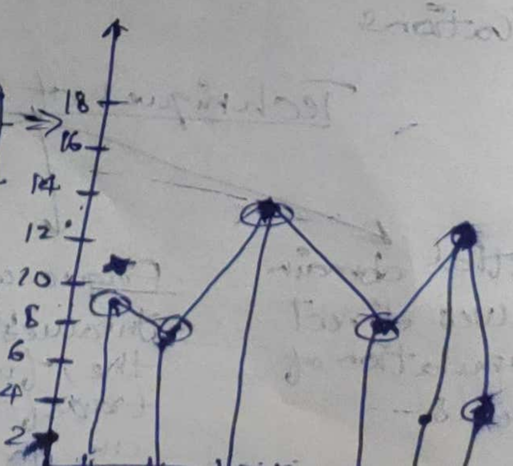


Gray levels (r_k)	No. of Pixels (n_k)	$P(r_k)=n_k/n$ (PDF)	S_k (CDF)	$S_k \times L-1$ $\Rightarrow S_k \times 7$	Histogram Equalization level
0	9	0.140	0.140	0.98	≈ 1
1	8	0.125	0.265	1.855	≈ 2
2	11	0.172	0.437	3.059	≈ 3
3	4	0.0625	0.4995	3.496	≈ 3
4	10	0.156	0.655	4.585	≈ 5
5	15	0.234	0.8895	6.226	≈ 6
6	4	0.0625	0.952	6.664	≈ 7
7	3	0.047	0.999	6.993	≈ 7

$n = 64$

New table:

r_k	1	2	3	5	6	7
n_k	9	8	15	10	15	7



Q. Perform histogram Equalization for the following image.

$$f(x,y) = \begin{bmatrix} 1 & 2 & 1 & 1 & 1 \\ 2 & 5 & 3 & 5 & 2 \\ 2 & 5 & 5 & 5 & 2 \\ 2 & 5 & 3 & 5 & 2 \\ 1 & 1 & 1 & 2 & 1 \end{bmatrix}$$

$\Rightarrow$ max value = 5

$2^0 = 1$

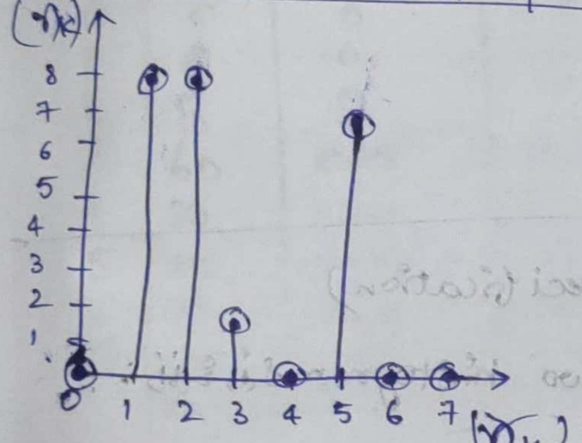
$2^1 = 2$

$2^2 = 4$

$2^3 = 8 \longrightarrow L = 8 \Rightarrow \boxed{L-1=7}$

$\boxed{n=3}$

Gray levels (r_k)	0	1	2	3	4	5	6	7
No. of Pixels (n_k)	0	8	8	2	0	7	0	0

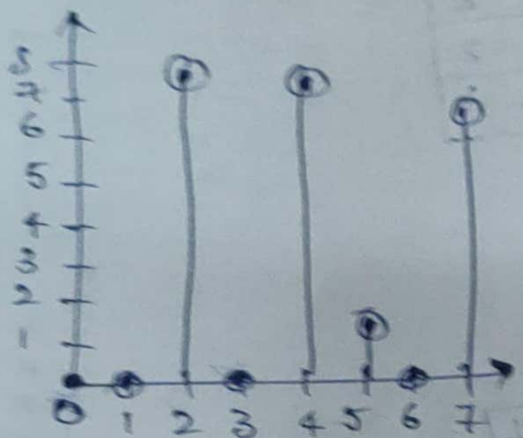


r_k	n_k	(PDF) $s_k = n_k/n$	(CDF) s_k	$s_k \times 7$	
0	0	0	0	0	≈ 0
1	8	0.32	0.32	2.24	≈ 2
2	8	0.32	0.64	4.48	≈ 4
3	2	0.08	0.72	5.04	≈ 5
4	0	0	0.72	5.04	≈ 5
5	7	0.28	1	7	≈ 7
6	0	0	1	7	≈ 7
7	0	0	1	7	≈ 7

$\boxed{n=25}$

New table :

r_k	0	2	4	5	7
n_k	0	8	8	2	7



$$\Rightarrow \begin{bmatrix} 2 & 4 & 2 & 2 & 2 \\ 4 & 7 & 5 & 7 & 4 \\ 4 & 7 & 7 & 7 & 4 \\ 4 & 7 & 5 & 7 & 4 \\ 2 & 2 & 2 & 4 & 2 \end{bmatrix}$$

ii, Histogram Matching (Specification)

Q. Given below are the two histograms i, & ii, . Modify i, by ii,

i,

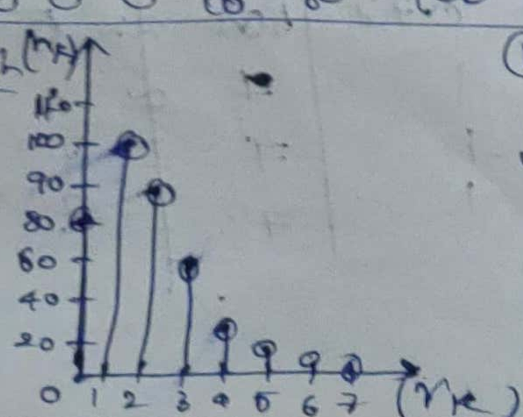
r_k	0	1	2	3	4	5	6	7
n_k	80	100	90	60	30	20	10	0

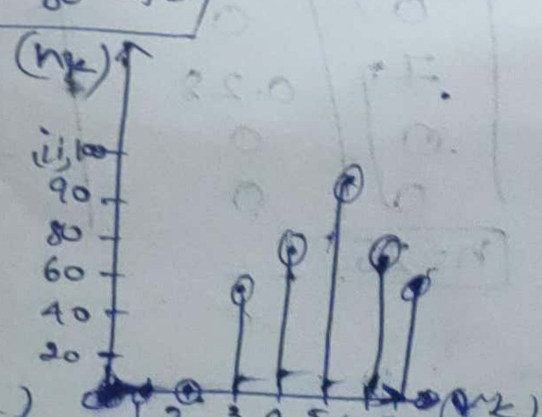
ii,

r_k	0	1	2	3	4	5	6	7
n_k	0	0	0	60	80	100	80	70

Sol: graphs

i,





Equalize the Histogram:,

r_k	n_k	$s_k = n_k/n$	s_k	$s_k \times 7$	New s_k	New n_k
0	80	0.20	0.20	1.4	1	80
1	100	0.25	0.45	3.15	3	100
2	90	0.23	0.68	4.76	5	90
3	60	0.15	0.83	5.81	6	90
4	30	0.07	0.9	6.3	6	
5	20	0.05	0.95	6.65	7	30
6	10	0.02	0.97	6.79	7	
7	0	0	0.97	6.79	7	

$\boxed{n = 390}$

Histogram Equalization level

ii,

r_k	n_k	$s_k = n_k/n$	s_k	$s_k \times 7$	New s_k
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	60	0.15	0.15	1.05	1
4	80	0.20	0.35	2.45	2
5	100	0.25	0.60	4.2	4
6	80	0.20	0.80	5.6	6
7	70	0.17	0.97	6.79	7

$\boxed{n = 390}$

iii, Mapping

→ Take the first & last columns of Histogram (ii)

→ Take the last 2 columns of Hg (i)

ii,

Gray level	Histogram Equalization level
0	0
1	0
2	0
3	1
4	2
5	4
6	6
7	7

i,

Hg Equalization level	New r_k
1	80
3	100
5	90
6	90
6	
7	30
7	
7	

⇒ 3 = 80

⇒ 4 = 80

⇒ 5 = 100

⇒ 6 = 90

⇒ 7 = 30

New table (merged)

Gray level	0	1	2	3	4	5	6	7
No. of Pixels	0	0	0	80	80	100	90	30

Fundamentals of Spatial filtering

→ The name filter is borrowed from frequency domain processing. It basically refers to accepting (Passing) or rejecting certain frequency components.

Ex: A filter that passes low frequencies is called lowpass filter.

→ we can accomplish a similar smoothing directly on the image itself using spatial filters. (also called masks, kernels, templates & windows)

(1) Convolution:

Q, let $I = \{0, 0, 1, 90\}$ be an image. Using the mask $R = \{3, 2, 8\}$, Perform the convolution.

Sol

a, Zero padding process for convolution

In Convolution process, we have to rotate the kernel by 180°.

```
8 2 3         8 2 3
0 0 0   0 1 0 0 0 0
```

b, Initial position

Template:

at center bit

```
8 2 3
0 0 0 0 1 0 0 0 0
  ↓
  0
```

$\Rightarrow (8 \times 0) + (2 \times 0) + (3 \times 0)$

$= 0$

output is 0 located at the center pixel

c, position after one shift

Template is shifted by one bit.

8 2 3
0 0 0 0 1 0 0 0 0
0 0

output is '0'.

d, position after two shifts

8 2 3
0 0 0 0 1 0 0 0 0
0 0 3

$\Rightarrow (8\times0)+(2\times0)+(3\times1)$
$= 3$

output is 3.

e, position after 3 shifts.

8 2 3
0 0 0 0 1 0 0 0 0
0 0 3 2 0 0 0 0

$\Rightarrow (8\times0)+(2\times1)+(3\times0)$
$= 2$

output is 2.

f, 4 shifts

8 2 3
0 0 0 0 1 0 0 0 0
0 0 3 2 8 0 0 0

$\Rightarrow (8\times1)+(2\times0)+(3\times0)$
$= 8$

$\Rightarrow$ output is 8.

g, 5 shifts

8 2 3
0 0 0 0 1 0 0 0 0
0 0 3 2 8 0 0 0

$\Rightarrow$ output is 0.

So in the final position the output produced is
{0, 0, 3, 2, 8, 0, 0, 0}

(2) Correlation:-

Q2, let .... (same as Q1)

↓

but we don't rotate the Kernel by 180°.

The o/t is like {0, 0, 8, 2, 3, 0}

8
2
3

For 2D Images:-

Q, let $I = \begin{pmatrix} 3 & 3 \\ 3 & 3 \end{pmatrix}$ be an Image &

$K = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ be a Kernal (mask). Perform

Convolution & correlation

(1) Convolution:-

i, Zero Pading:-

Rotate the Kernel by 180°

$K' \Rightarrow \begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \Rightarrow \begin{matrix} 3 - 4 \\ 1 - 2 \end{matrix} \Rightarrow \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}$

a, $\begin{matrix} 0\times4 & 0\times3 & 0 & 0 \\ 0\times2 & 3\times1 & 3 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$ → $(0\times4)+(0\times3)+(0\times2)+(3\times1)$

$\Rightarrow \begin{matrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$

$K' = \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}$

b, $\begin{matrix} 3 & 0\times4 & 0\times3 & 0 \\ 0 & 3\times2 & 3\times1 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{matrix} \Rightarrow \begin{matrix} 3 & 9 & 0 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$

$\Rightarrow (0\times4)+(0\times3)+(3\times2)+(3\times1)$

$= 9$

c, 3 9 0x4 0x3

0 3 3x2 0x1

0 3 3 0

0 0 0 0

$\Rightarrow$

3 9 6 0

0 3 3 0

0 3 3 0

0 0 0 0

d, 3 9 6 0

0x4 3x3 3 0

0x2 3x1 3 0

0 0 0 0

$\Rightarrow$

3 9 6 0

12 3 3 0

0 3 3 0

0 0 0 0

e, 3 9 6 0

12 3x4 3x3 0

0 3x2 3x1 0

0 0 0 0

$(3\times4)+(3\times3)+(3\times2)+(3\times1)$

$= 12+9+6+3$

$= 30$

$\Rightarrow$

3 9 6 0

12 30 3 0

0 3 3 0

0 0 0 0

f, 3 9 6 0

12 30 3x4 0x3

0 3 3x2 0x1

0 0 0 0

$\Rightarrow$

3 9 6 0

12 30 18 0

0 3 3 0

0 0 0 0

g, 3 9 6 0

12 30 18 0

0x4 3x3 3 0

0x2 0x1 0 0

$\Rightarrow$

3 9 6 0

12 30 18 0

9 3 3 0

0 0 0 0

h, 3 9 6 0

12 30 18 0

9 3x4 3x3 0

0 0x2 0x1 0

$\Rightarrow$

3 9 6 0

12 30 18 0

9 21 3 0

0 0 0 0

i,
$$\begin{bmatrix} 3 & 9 & 6 & 0 \\ 12 & 30 & 18 & 0 \\ 9 & 21 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 3 & 9 & 6 & 0 \\ 12 & 30 & 18 & 0 \\ 9 & 21 & 12 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

2, For correlation: Perform the same steps without rotating the kernel by 180°.

Smoothing Spatial filters

→ Smoothing filters are used for blurring & noise reduction.

→ Blurring is used in preprocessing tasks, such as removal of small details from an image prior to (large) object extraction.

→ Noise reduction can be accomplished by blurring with a linear filter & also by non-linear filtering.

Smoothing Spatial filters

- i, linear filters
  - Mean/Box filter
  - weighted ~~avegar~~ average filter
  - Gaussian filter.
- ii, Non-linear filters (order-statistic filter)
  - Median filter
  - max filter
  - min filter.

i, linear filters: They are also known as averaging filters (or) lowpass filters as they are simply the average of the pixels contained in the neighbourhood of the filter mask.

The process results in an image with reduced 'sharp' transitions in intensities which ultimately leads to noise reduction.

1, Box filter - all coefficients are equal

$$\underbrace{\frac{1}{9}}_{\text{Sum of all pixels}} \times \underbrace{\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}}_{3\times3} \rightarrow \text{mask}$$

2, weighted average - give more (less) weight to pixels near (away from) the output location.

$$\underbrace{\frac{1}{16}}_{\text{Sum of pixels}} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \rightarrow \text{mask}$$

3, Gaussian filter - the weights are samples of 2D Gaussian function.

$$\boxed{G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}}$$

(2D Gaussian function

$\sigma \rightarrow$ standard deviation

→ Used to blur edges & reduce contrast.

→ Similar to median filter but is faster.

---

ii, Non-linear (order-statistic filters):

Their response is based on ordering (ranking) the pixels contained in the image area encompassed by the filters, & then replacing the value of the center pixel with the value of the center pixel with the value determined by the ranking result.

1, Median: Find the median of all the pixel values

2, & 3, max & min also same.

Q, Consider the image below & calculate the output of the pixel (2,2) if smoothing is done using 3x3 neighborhood using all the filters below.

a, Box/Mean filter
b, weighted average filter
c, Median filter
d, Min "
e, Max "

$$\begin{matrix} 1 & 8 & 8 & 0 & 7 \\ 4 & \boxed{7} & 9 & 5 & 7 \\ 5 & 4 & 6 & 8 & 6 \\ 4 & 2 & 0 & 1 & 5 \\ 0 & 1 & 0 & 2 & 0 \end{matrix}$$

Center pixel

3x3

a, Box filter: $\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

$\Rightarrow \frac{1}{9} \times [7+9+5+4+6+8+2+0+1]$

$= \frac{1}{9} \times [42]$

$= 4.66$

≈ 5 ∴ The center pixel is replaced by '5'.

b, weighted avg filter: mask $= \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

$= \frac{1}{16} \times [(7\times1)+(9\times2)+(5\times1)+(4\times2)+(6\times4)+(8\times2)+(2\times1)+(0\times2) + (1\times1)]$

$= \frac{1}{16} [81]$

$= 5.0625$

≈ 5 ∴ Center pixel is replaced by '5'.

c, Median:

The image is written in ascending order.

$\Rightarrow$ 0, 1, 2, 4, 5, 6, 7, 8, 9

Median = 5 (replace)

d, Min:

$\Rightarrow$ minimum value is '0' (replace)

e, Max:

$\Rightarrow$ maximum value is '9' (replace)

Few Important Questions:

1Q, Why median filter is better than mean filter?

Ans:- Median filter is normally used to reduce noise in an image, similar to the mean filter. However, it often does a better to the mean filter in preserving useful detail in an image.

Median filter has 2 main advantages:-

1, The median is an more robust average than the main & so a single very unrepresentative pixel, in a neighborhood will not affect the median value significantly.

2, Since the median value must actualy be the value of one of the pixels in the neighborhood the median filter does not create new unrealistic pixel values when the filter straddles an edge. Therefore, it is much better at ~~preserving~~ preserving sharp edges than the mean filter.

Note:
- Mean filter is better at dealing with Gaussian noise than median filter.
- Median filter is better at dealing with salt & pepper noise than mean filter.

2Q, Write down a few approaches to deal with missing edge pixels.

Ans:- A few approaches to dealing with missing edge pixels are:-

1, **Omit missing pixels:-**
- only works with some filters
- can add extra code & slow down processing.

2, **Pad the image:-**
- Typically with either all white or all black pixels.

3, Replicate border pixels.
4, Truncate the Image.
5, Allow pixels wrap around the Image.
- Can cause some strange image artifacts.

→ Zero padding → pixel Replicate → wrapping of Pixels

Zero padding

0	0	0	0	0
0	1	2	3	0
0	4	5	6	0
0	7	8	9	0
0	0	0	0	0

pixel Replicate

1	1	2	3	3
1	1	2	3	3
4	4	5	6	6
7	7	8	9	9
7	7	8	9	9

wrapping of Pixels

9	7	8	9	7
3	1	2	3	1
6	4	5	6	4
9	7	8	9	7
3	1	2	3	1

Sharpening Spatial Filters :-

→ The principal objective of sharpening is to highlight transitions in intensity.

→ Applications of Image sharpening include electronic printing, medical imaging, industrial inspection & autonomous guidance in military systems.

Blurring → pixel averaging

Sharpening → spatial differentiation.

→ The strength of the response of a derivation operator is proportional to the degree of intensity discontinuity to the ~~degree~~ image at the point at which the operator is applied.

→ Therefore image differentiation Enhances Edges and other discontinuities (such as noise) & deemphasizes areas which have slowly varying intensities.

Foundation of Sharpening filters:

1. First-order derivative of a one-dimensional function $f(x)$:

$$\boxed{\frac{\partial f}{\partial x} = f(x+1) - f(x)}$$

2. Second order derivative of a one-dimensional function $f(x)$:

$$\boxed{\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)}$$

Laplacian filter

→ It highlights gray-level discontinuities in an Image

→ It deemphasized regions with slowly varying gray levels.

→ formula: $\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$

Where, $\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x,y)$ — (1)

$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x,y)$ — (2)

$$\Rightarrow \boxed{\nabla^2 f = f(x+1,y) + f(x,y+1) + f(x-1,y) + f(x,y-1) - 4f(x,y)}$$

laplacian mask:

$x-1, y+1$	$x, y+1$	$x+1, y+1$
$x-1, y$	x, y	$x+1, y$
$x-1, y-1$	$x, y-1$	$x+1, y-1$

$$\boxed{\begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix}} \qquad \begin{matrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{matrix}$$

$$\begin{matrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{matrix} \qquad \begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix}$$

Q. Apply laplacian filter on the given Image on the center pixel.

$$\begin{bmatrix} 8 & 5 & 4 \\ 0 & 6 & 2 \\ 1 & 3 & 7 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Input Image mask

$\Rightarrow (8\times0)+(5\times1)+(4\times0)+(0\times1)+(6\times-4)+(2\times1)+(1\times0)+(3\times1)+(0\times7)$

$\Rightarrow 0+5+0+0+(-24)+2+3+0$

$= -14$

$$\Rightarrow \begin{bmatrix} 8 & 5 & 4 \\ 0 & -14 & 2 \\ 1 & 3 & 7 \end{bmatrix}$$

Q2. Apply Enhanced laplacian filter on the given on the center pixel.

$$\begin{bmatrix} 8 & 5 & 4 \\ 0 & 6 & 2 \\ 1 & 3 & 7 \end{bmatrix}$$

Notes: Enhanced laplacian is Nothing but increment 1 in Center pixel. If its negative increment -1 to it.

$\Rightarrow$ Enhanced mask is

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

$\Rightarrow (8\times0)+(5\times1)+(4\times0)+(0\times1)+(6\times(-5))+(2\times1)+(1\times0)+(3\times1)+(7\times0) = -20$

$$\begin{bmatrix} 8 & 5 & 4 \\ 0 & -20 & 2 \\ 1 & 3 & 7 \end{bmatrix}$$

Unsharp Masking & Highboost filtering

- primarily used in the printing & publishing industry to sharpen images.
- The process involves subtracting an unsharp (smoothed) version of an image from the original image.

This process called unsharp masking consists of the following steps:

1. Blur the original image.
2. Subtract the blurred image from the original (The resulting difference is called the mask)
3. Add the mask to the original.

Unsharp masking

$$f_s(x, y) = f(x, y) - \bar{f}(x, y)$$

Sharpend image = original image − blurred image.

Subtracting a blurred version of an image from the original produces a sharpened image.

Highboost filtering

$$f_{hb}(x, y) = A f(x, y) - \bar{f}(x, y)$$

$$= A(f(x, y)) - [f(x, y) - f_s(x, y)]$$

$$\boxed{f_{hb} = (A-1) f(x, y) - f_s(x-y).}$$

This is the generalised form of Unsharp masking. Where $A \geq 1$. A specifies the amount of sharpening of the image.

If we use Laplacian ...

Sharpend image $f_s(x,y)$ with addition of the original, image. $f_s(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) \\ f(x,y) + \nabla^2 f(x,y) \end{cases}$

$$f_{hb}(x,y) = \begin{cases} Af(x,y) - \nabla^2 f(x,y) \\ Af(x,y) + \nabla^2 f(x,y) \end{cases}$$

High boost mask

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & A+4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & A+8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

where $A \geq 1$

Q, Apply highboost filter on the image given below on the center pixel. Use the mask with $A = 1.7$

$\Rightarrow$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \times \begin{bmatrix} -1 & -1 & -1 \\ -1 & A+8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Input Image

$\Rightarrow (1\times-1) + (2\times-1) + (3\times-1) + (4\times-1) + (6\times-1) + (7\times-1) + (8\times-1) + (9\times-1) + 5\times(1.7+8)$

$= -1(1+2+3+4+6+7+8+9) + 48.5$

$= -40 + 48.5$

$= 8.5$