Image Processing
IT D12

Page No.
Date

Image: An image is a 2-D function, $f(x,y)$ where x and y are the spatial ~~special~~ plane coordinates, and the amplitude of f at any pair of coordinates (x,y) is called the intensity of the image at the level.

Digital Image: It is composed of finite number of elements called pixel each of which has a particular location & value.

Image types:

(i) Monochrome / Bitmap Images:
1 bit ~~2 bits to represent~~ for pixel (0,1)
0 → black
1 → white

(ii) Gray Level Image:
8 bits ~~8 bits~~ (0 – 7) for a pixel
* 256 variations or gray levels

(iii) Color Image:
* 24 bits
* (0-7) R, (0-7) G, (0-7) B
* 3 planes

Gray level, $X = 0.30R + 0.59G +$ ~~0.4B~~ $0.11B$

(iv) Half toned image:
Grayscale image using only black & white colors

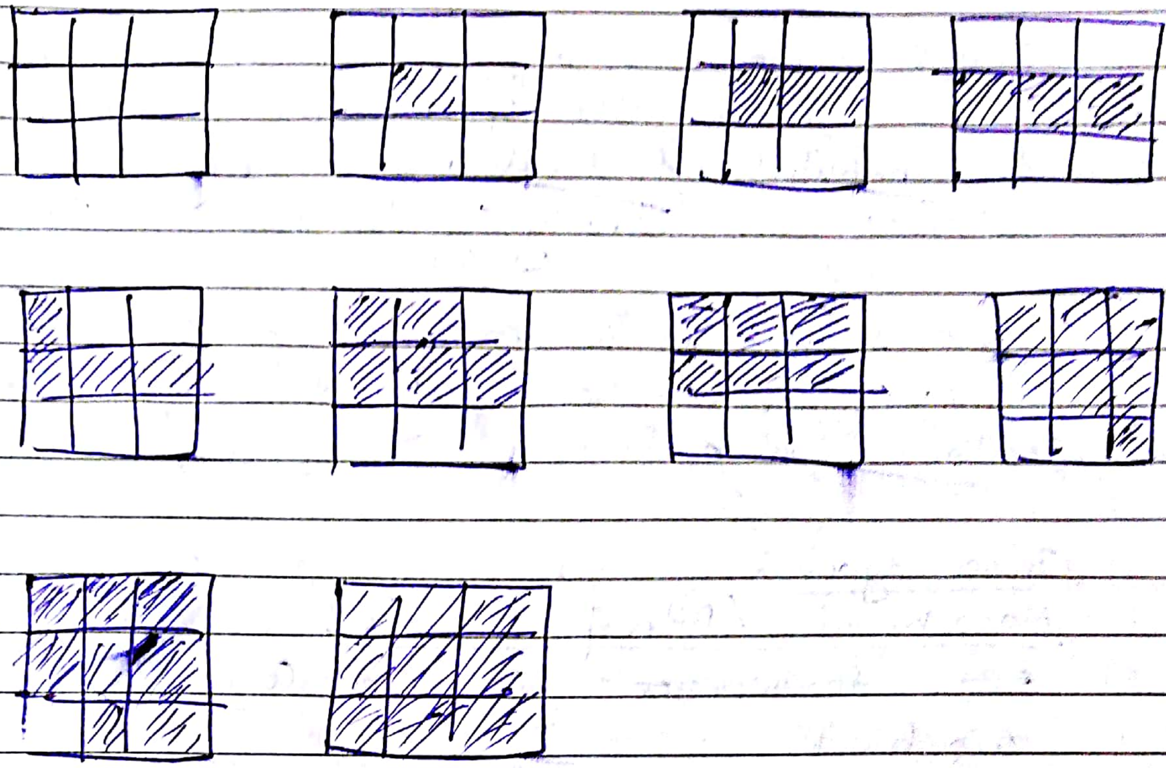


Image Processing:

It is a method to convert an image into a digital form and perform some operations on it in order to get an enhance image or to extract some useful information of it.

imp

Purpose of image processing:

(i) Visualisation: Observe the objects that are not visible. Eg sharpen blurred image, x-ray image

(ii) Image sharpening & restoration: To create a better image.

(iii) Image retrieval: Seek for the image of interest

(iv) Measurement of pattern: ~~Calculate length~~ Measure various ~~of~~ objects in a image.

(v) Image recognition: Distinguish the objects in an image.

Applications:

(i) Intelligent transport system:

(ii) Remote Sensing: Processing satellite images to predict floods, ~~etc~~ etc.

(iii) Moving object tracking: Measure motion parameters

(iv) Defence surveillance:

(v) Biomedical imaging techniques: X-ray, MRI, CT scan.

(vi) Automatic visual inspection system:

Low level Processing:

* Both input and output are images. ~~For~~
* Image processing operations such as noise reduction, contrast ~~enh~~ enhancement, image sharpening are performed.

Mid level Processing:

* Input are images but output are characteristics expected from those images such as ages, contours (boundary extraction) and identity of individual

Objects

* Processing images to render for further computer processing.

* Segmentation for object recognition and classification

High level Processing :

* Tracking or identifying objects in an image.
* Performing cognitive functions (involved in knowing learning and understanding typically associated with human vision)

Ques Given an 3x3 image : Plot its bit planes

1	2	0
4	3	2
7	5	2

Grayscale image
Max value = 7
3 bit required
∴ 3 bit planes

0	0	0
1	0	0
1	1	0

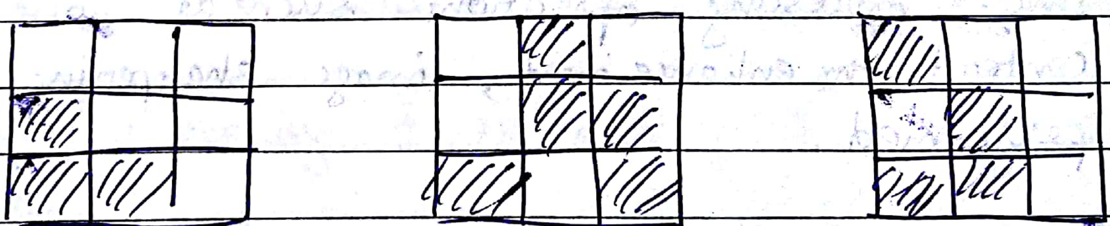
bit plane 2
MSB

0	1	0
0	1	1
1	0	1

bit plane 1
Mid SP

1	0	0
0	1	0
1	1	0

bit plane 0
LSB



Steganography :

(Embedding & extraction)

Page No.
Date

To hide a text in an image

150 → 10010110

50 → 00110010

51 ⇆ 00110011 very less difference for a gray scale image

50

Similar

To hide image

20 → 00010100

20	70	3
4	5	9
13	12	11

Image formats:

(i) JPEG (.jpg):

* Lossy but loss in quality is not very noticable
* Joint Photographic Expert Group
* Not good for sharp edges or text
* Quantization matrix, DCT

(ii) PNG (.png):

* Portable Network Graphics
* Lossless
* Compressed without loss in quality
* Great for sharp edges, texts

(iii) GIF

* Graphics Interchange Format
* Used for web application / graphics
* Limited to only 256 colours.

(iv) TIFF:

* Tagged Image File Format
* Lossless
* Huffman Coding

* Produce images of very high quality
* Large file size

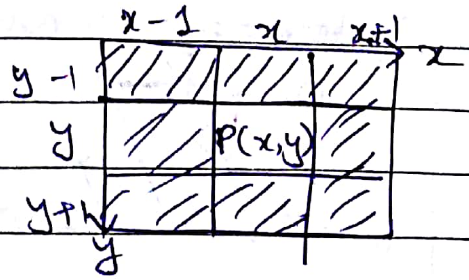
(v) Bitmap (.bmp):
* developed by microsoft
* no compression
* very high quality and very large size

(vi) RAW:
* unprocessed data

(vii) EPS:
* Encapsulated Post script
* Scalable to any size
* Can be opened by graphic design software
* Doesnot use color
* Recommended for print

Difference between gif, jpeg and png.

Neighborhood of pixel



$N_4(P)$: A pixel (x,y) has 2 horizontal & 2 vertical neighbour. This set of 4 pixels is called 4 neighbours of P & each of these neighbour is at a unit distance

$$N_4(P) = \{(x, y-1), (x, y+1), (y, x-1), (y, x+1)\}$$

for (x,y)

for (x,y) $N_D(P) = \{(x-1, y-1), (x+1, y+1), (x+1, y-1), (x-1, y+1)\}$
(Diagonal Pixel)

$$N_8(P) = \{\text{all neighbour of } (x,y)\}$$
$$= N_4(P) + N_D(P)$$

Connectivity of Pixels:

Let V be the set of gray levels used to define connectivity for 2 pts, P, Q ∈ V.

3 types of connectivity are defined:-

(i) 4-connectivity:

$P, q \in V$ & $q \in N_4(P)$

(ii) 8-connectivity:

$P, q \in V$ & $q \in N_8(P)$

(iii) M-connectivity (Mixed connectivity):

$P, q \in V$ are M-connected if

a) $q \in N_4(P)$

OR

b) $q \in N_D(P)$ and $N_4(P) \cap N_4(q) = \phi$ (empty)

Ques $V = \{1\}$

~~Determine~~ 4-connected Pixel.

$$\begin{array}{ccc} & p & q \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \qquad \begin{array}{ccc} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}$$

~~Determine~~ 8-connected pixels

$$\begin{array}{ccc} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \qquad \begin{array}{ccc} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}$$

~~Determine~~ M-connected Pixel

$$\begin{array}{ccc} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}$$

→ remove redundancy of path in $N_8(P)$

→ Duplicate path

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

$$\begin{array}{ccc} 0 & 1 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{array} \qquad \begin{array}{ccc} 0 & 1 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{array} \qquad \begin{array}{ccc} 0 & 1 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{array}$$

4-connected   8-connected   M-connected

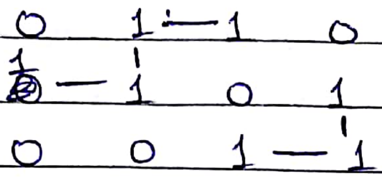
Ques $V = \{2, 3, 4\}$

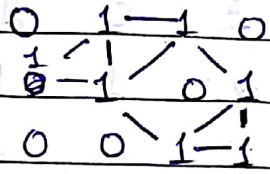
$$\begin{array}{cccc} & & q & \\ 4 & 1 & 3 & 2 \\ 2 & p\,3 & 4 & 5 \\ 5 & 2 & 1 & 4 \\ 2 & 2 & 4 & 3 \end{array}$$

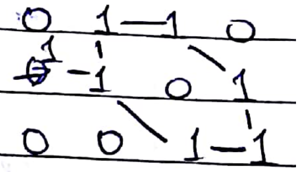
Ques Is p, q M-connected or not

Sol^n Not M-connected

Que $V = \{1\}$


4-connected


8-connected


M-connected

PATH & Length of Path:

A path from $P(x,y)$ to $Q(s,t)$ is defined as a sequence of distinct pixels.

$$(x_0, y_0),\ (x_1, y_1),\ (x_2, y_2) \ldots\ (x_n, y_n)$$

where

$$(x_0, y_0) = (x, y) \quad \& \quad (x_n, y_n) = (s, t)$$

(x_i, y_i) is adjacent to (x_{i-1}, y_{i-1}) for $1 \leq i \leq n$

$n \rightarrow$ length of path.

Adjacency:

Let V be the set of graylevels used to define adjacency.

a) 4-adjacency: 2 pixels P and Q with values from V are 4-adjacent if Q is in the set $N_4(P)$

b) 8-adjacency: 2 Pixels P and Q with values from V are 8-adjacent if Q is in the set $N_8(P)$

c) M-adjacency: $P, Q \in V$

(i) $Q \in N_4(P)$

OR

(ii) $Q \in N_D(P)$ and $N_4(P) \cap N_4(Q) = \Phi$ (empty)

Ques

$$\begin{matrix} 3 & 1 & 2 & 1\ (q) \\ 2 & 2 & 0 & 2 \\ 1 & 2 & 1 & 1 \\ 1 & 0 & 1 & 2 \end{matrix}$$

$V = \{0, 1\}$

(p)
Compute the length of shortest 4, 8 and M-Paths between P and Q, if particular path doesnot exist b/w these 2 pts, explain why.

$$\begin{array}{ccccc} & 3 & 1 & 2 & 1\ Q \\ & 2 & 2 & 0 & 2 \\ & 1 & 2 & 1 & 1 \\ P & 1 & 0 & 1 & 2 \end{array}$$

4-connected

4- Path doesnot exist
4- adjacency doesnot exist

$$\begin{array}{ccccc} & 3 & 1 & 2 & 1\ Q \\ & 2 & 2 & 0 & 2 \\ & 1 & 2 & 1 & 1 \\ P & 1 & 0 & 1 & 2 \end{array}$$

8- connected

length of 8-path = 4

$$\begin{array}{ccccc} & 3 & 1 & 2 & 1\ Q \\ & 2 & 2 & 0 & 2 \\ & 1 & 2 & 1 & 1 \\ P & 1 & 0 & 1 & 2 \end{array}$$

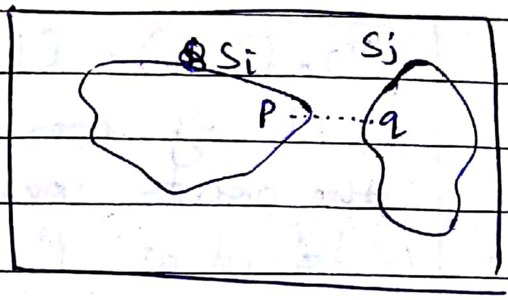
M- connected

length of M-path = 5

* Length of shortest path is 4 (8-Path)

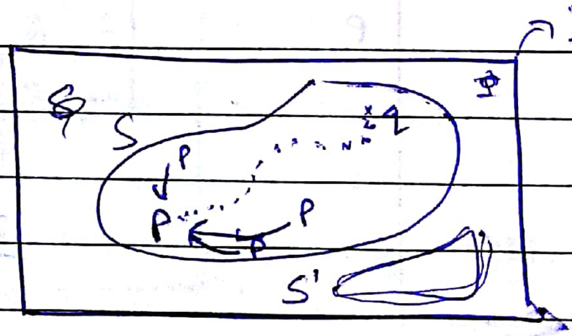
\# 2 Image Subsets, S_i & S_j are adjacent if there exist a $P \in S_i$ and there exist $Q \in S_j$ such that P and Q are adjacent.

Application: 2 objects ~~2 or image~~ can be considered 1 object because they are connected.



Connected Component:

If $S \subseteq I$ and $P, Q \in S$, then P is connected to q in S, if there is a path from $P \to q$, consisting entirely of pixel in S. For any $P \in S$, the set of pixels in S that are connected to P, is called connected component of S.
Any 2 pixels of a connected component are connected to each other. Distinct connected components are disjoint.



Distance Measure:

For pixels (P, q, z) with coordinates (x, y); (s, t); and (u, v) respectively, we call D to be a valid distance function or metric if it fulfils:

a) $D(P, q) \geq 0$ $[D(P, q) = 0$ if $P = q]$

b) $D(P, q) = D(q, P)$

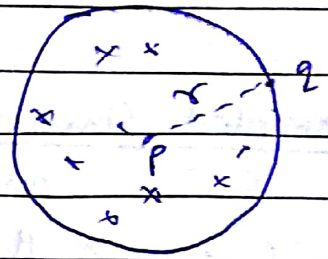
c) $D(P, z) \leq D(P, q) + D(q, z)$

Euclidean Distance b/w P & Q:

$P(x, y)$ & $Q(s, t)$

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

Set of ~~poo~~ points, $S = \{q \mid D(P,q) \le r\}$ are the points contained in a disk of radius, r centered at P



~~Mathattan~~ City - Block Distance (Manhattan):

D_4 distance or City- Block Distance,

$$D_4(P,Q) = |x-s| + |y-t|$$

Points having city block distance from P less than or equal to r ~~from~~ form a diamond centered at P

		2		
	2	1	2	
2	1	P	1	2
	2	1	2	
		2		

Chess- Board Distance:

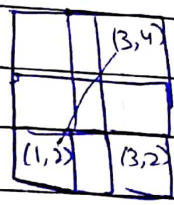
D_8 distance

$$D_8(P,Q) = \max(\cancel{(x-s)}|x-s|, |y-t|)$$

$S = \{q \mid D_8(P,Q) \le r\}$ forms a square centered at P

2	2	2	2	2
2	1	1	1	2
2	1	P	1	2
2	1	1	1	2
2	2	2	2	2

Eg $P(1,2)$, $Q(3,4)$



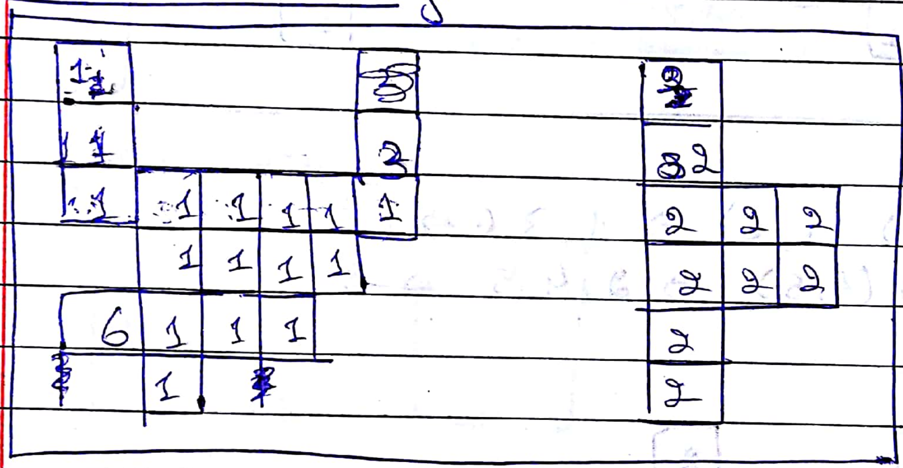
$$D_E(P,Q) = \sqrt{(1-3)^2+(2-4)^2}$$
$$= \sqrt{4+4} = 2\sqrt{2}$$

$$D_4(P,Q) = |1-3| + |2-4| = 4$$

$$D_8(P,Q) = \max(|1-3|, |2-4|)$$
$$= 2.$$

~~Component Labelling~~

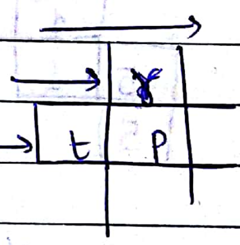
Component labelling:



level 1 level 2

Algorithm:

Scan the image from left to right



1st Pass:

$l(P)$ = Pixel value at position P

$L(P)$ = Label assigned to pixel ~~value~~ location P

if $l(P)=0$, move to the next scanning position

if $l(P)$ is in V & $l(t) = l(r) = 0$, assign new label to $l(P)$.

if $l(P)$ is in V & $L(r) \neq L(t)$, then assign one of the label to $l(P)$.

else $l(P) = l(r)$

Eg $P(1,2)$, $Q(3,4)$

$D_E(P,Q) = \sqrt{(1-3)^2 + (2-4)^2}$

$= \sqrt{4+4} = 2\sqrt{2}$

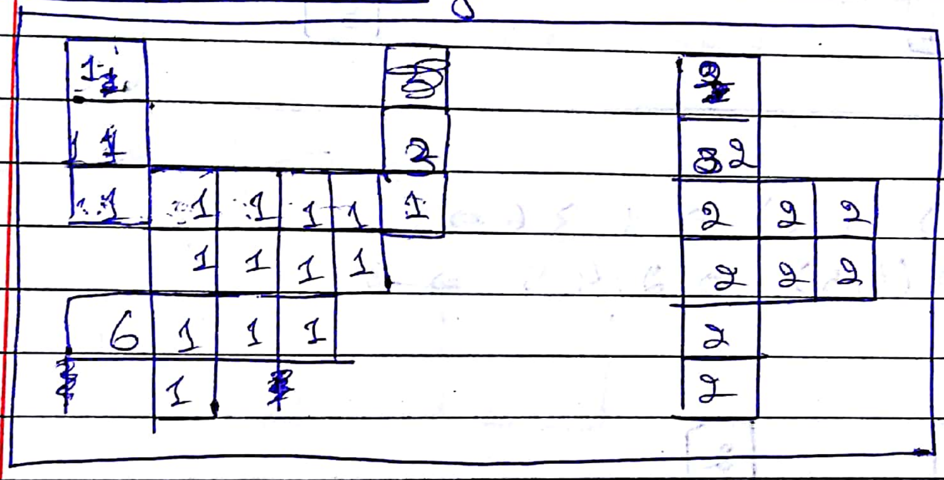
$D_4(P,Q) = |1-3| + |2-4| = 4$

$D_8(P,Q) = \max(|1-3|, |2-4|)$

$= 2.$

~~Component Labelling~~

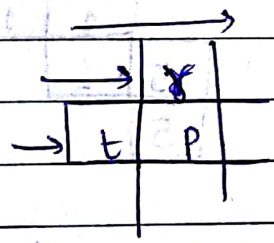
Component labelling:



level 1 level 2

Algorithm:

Scan the image from left to right



1st Pass:

$l(P)$ = Pixel value at position P

$L(P)$ = Label assigned to pixel ~~value~~ location P

if $l(P) = 0$, move to the next scanning position

if $l(P)$ is in V & $l(t) = l(r) = 0$, assign new label to $l(P)$.

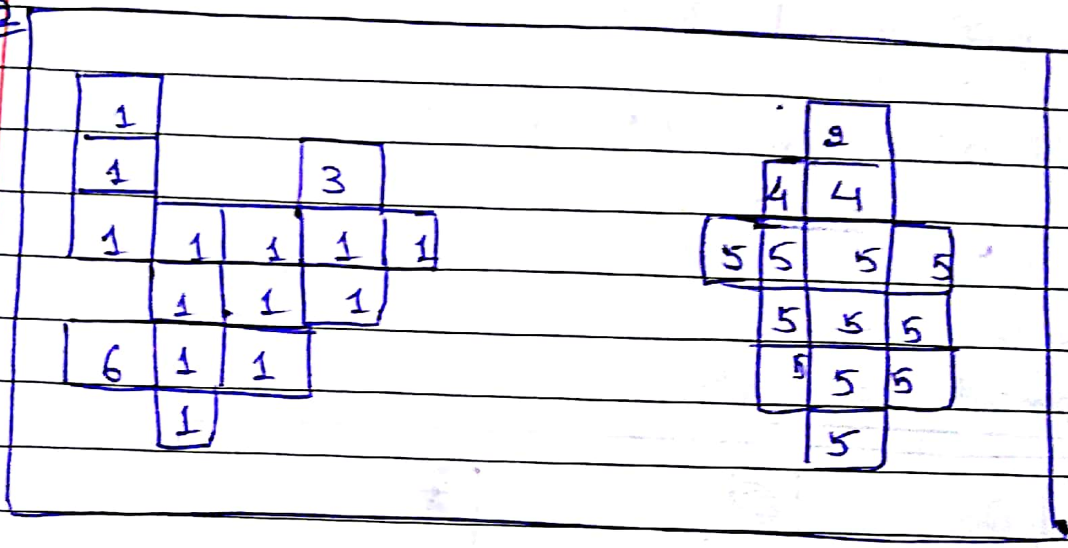
if $l(P)$ is in V & $L(r) \neq L(t)$, then assign one of the label to $l(P)$.

else $L(P) = l(r)$

2nd Pass: Process equivalent pairs to form equivalence classes.
Assign new label

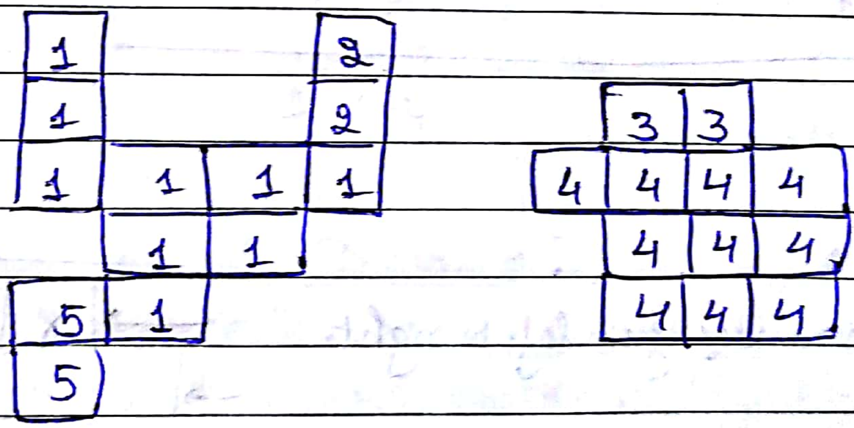
2 ↑ 4

Eg



(1, 3) (1, 6) ⇒ 1, 3, 6 ⇒ 1
(2, 4) (4, 5) ⇒ 2, 4, 5 ⇒ 2

Ques



(1, 2) (1, 5) ⇒ 1, 2, 5 ⇒ 1
(3, 4) ⇒ 3, 4 ⇒ 2

Image Algebra:

Addition:

Application: Image Morphing

Subtraction:

Application: ~~Moment~~ Movement Detection

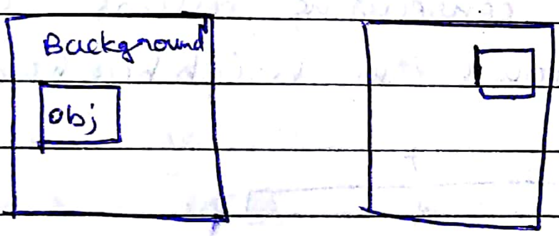
Multiplication:

Application: Brightness increases (Gives lighter image)
Gives a more saturated image

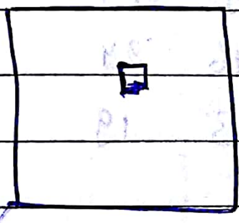
Divisions: (by a factor)

Application: Gives a ~~darker~~ darker image

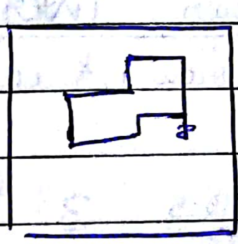
Logical Operations:



~~NOT ->~~ AND ->



OR ->



XOR ->

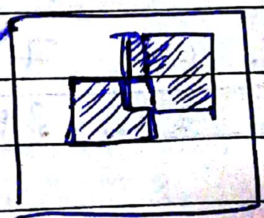


Image Zooming:

1. ~~Insert~~ columns b/w ~~& rows~~

24	14	54
68	24	26
18	19	84

Neighborhood Zooming Method

24	24	14	14	54	54	
68	68	24	24	26	26	→ stair case effect
18	18	19	19	84	84	

Linear Interpolation:

(Avg. of two consecutive columns or 2 consecutive rows in neighborhood zooming method)

24	19	14	34	54
68	46	24	25	26
18	18	19	52	84

~~Bilinear~~ Bilinear interpolation

Both row & column.

2×4 ↓ After zooming 4×8

							X
1	1.5	2	2.5	3	3.5	4	2
3	3.5	4	4.5	5	5.5	6	3
5	5.5	6	6.5	7	7.5	8	4
2.5	2.75	3	3.25	3.5	3.75	4	2

Scanned by CamScanner

Image Shrinking:

~~to~~ Retain visual quality

Remove ~~a~~ row and column

Image Resolution:

↳ Bits per pixel

Unit → ~~a~~ bpi (bits per inch)

For eg for gray scale image

~~2^8~~ $256 = 2^8$

8 bits per pixel.

Gray levels → 256

Ques Determine f no. of gray levels, with a gray level resolution of 2^k, for

(i) k = 5 , ~~no. of~~

(ii) k = 3

$$f = \begin{bmatrix} 123 & 162 & 200 & 147 & 93 \\ 137 & 157 & 165 & 232 & 189 \\ 151 & 155 & 152 & 141 & 130 \\ 205 & 101 & 100 & 193 & 115 \\ 250 & 50 & 75 & 88 & 100 \end{bmatrix}$$

Sol^n (i) No. of gray levels, $2^5 = 32$

Multiply ~~Divide~~ each pixel by $\frac{32}{256}$ or $\frac{1}{8}$

i.e Divide each pixel by 8. $[2^{8-5} = 2^3]$

Size of image = M×N×K
where m and n is no. of rows & columns & K is bits per pixel

$$f = \begin{bmatrix} 15 & 20 & 25 & 18 & 11 \\ 17 & 19 & 20 & 29 & 23 \\ 18 & 19 & 19 & 17 & 16 \\ 25 & 12 & 12 & 24 & 14 \\ 31 & 6 & 9 & 11 & 12 \end{bmatrix}$$

(ii) $k = 3$

No. of gray level, $L = 2^3 = 8$
Divide each pixel by 32 $[2^{8-3} = 2^5]$

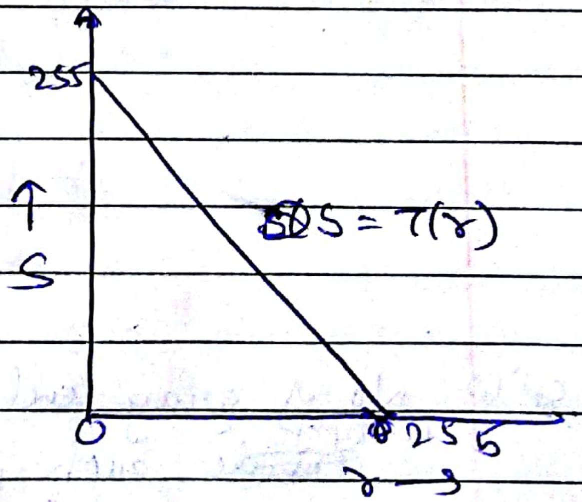
$$f = \begin{bmatrix} 3 & 5 & 6 & 4 & 2 \\ 4 & 4 & 5 & 7 & 5 \\ 4 & 4 & 4 & 4 & 4 \\ 6 & 3 & 3 & 6 & 3 \\ 7 & 1 & 2 & 2 & 3 \end{bmatrix}$$

Image Enhancement: Modify an image pixel values so that the o/p image suits for some specific application

1) **Digital Negative Image:**

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

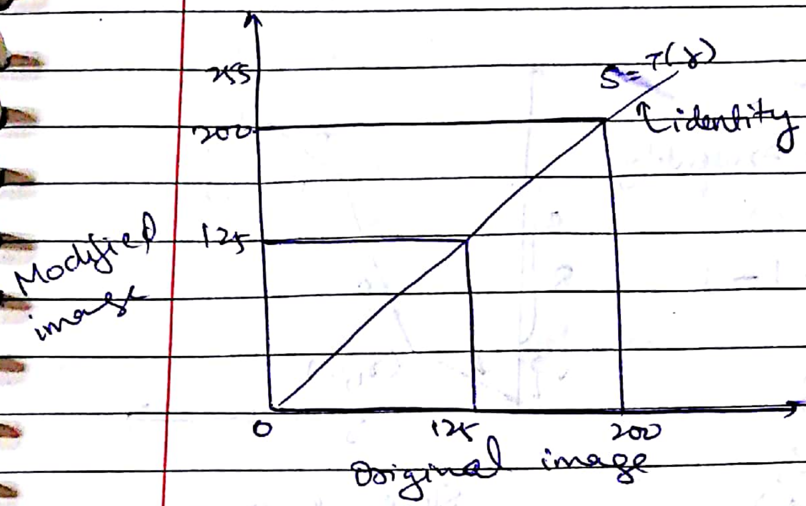
Applications: In medical x-rays



Point Processing:

Higher bits ~~are~~ carry visual significant information
Lower bits stores intensity of image

Page No.
Date



2) Intensity level Slicing :

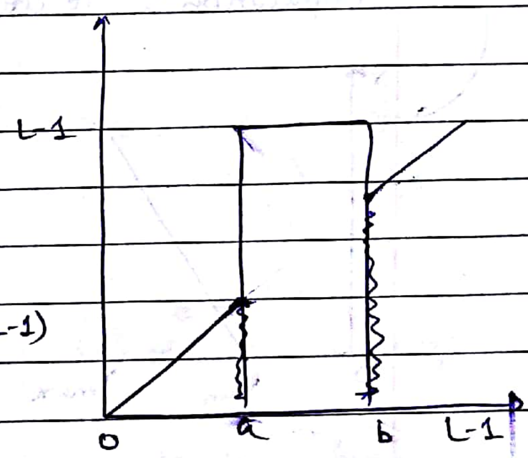
$$s = \begin{cases} L-1, & \text{if } a \le r \le b \\ 0, & \text{otherwise} \end{cases}$$

To retain color information

$$s = \begin{cases} L-1, & \text{if } a \le r \le b \\ r & \text{otherwise} \end{cases}$$

Application :- Satellite images.

River: ~~To~~ River intensity → known (range of gray scale value to L-1)



3) Bit plane Slicing :

Application :- Compression

Gray scale value calculation for a compressed 4-bit image

8	7	6	5
1	0	0	1
1	0	1	0
1	0	0	0

Eg

~~Gray scale~~ $= \cancel{8 \times 2^7 + 5 \times 2^4}$
$= \cancel{8 \times 128 + 5 \times 16}$
$= \cancel{1024 + 80} = \cancel{1104}$

Gray Scale (1001) $= \sum \text{bit value} \times 2^{n-1}$
$= 1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 1 \times 2^4$
$= 128 + 16 = 144$

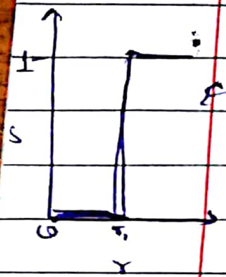
Gray Scale (1000) $= 128$

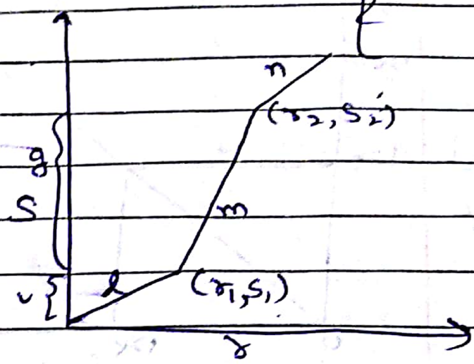
All other image enhancements techniques

$S = \{$

4) Contrast Stretching:

$r_1 = s_1$ & $r_2 = s_2 \Rightarrow$ identity line





$r_1 = r_2$, $s_1 = 0$ & $s_2 = L-1$

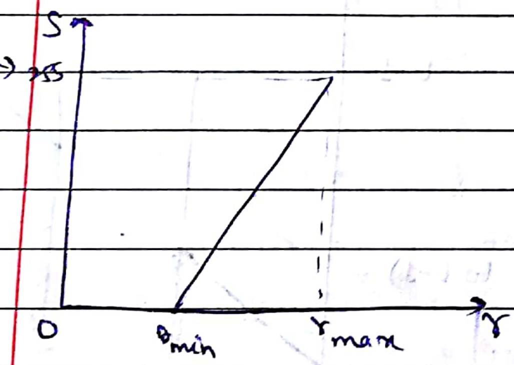
$(r_1, s_1) = (r_{min}, 0)$

&

$(r_2, s_2) = (r_{max}, L-1)$

Slopes = l, m, n

Application: To utilize dynamic range of camera



$$S = \begin{cases} l \cdot r, & 0 \le r \le a \\ m(r-a)+v, & a < r < b \\ n(r-b)+v+g, & b \le r \le L-1 \end{cases}$$

Filtering:

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

mask →

convolve

I/P image

	z_1	z_2	z_3
	z_4	z_5	z_6
	z_7	z_8	z_9

O/P →

		z_5'	

$$z_5' = \frac{w_1 z_1 + w_2 z_2 + w_3 z_3 + \ldots w_9 z_9}{n}$$

weighted sum

Linear filter (or Weighted sum)

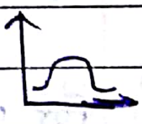
no need of mask # Non-linear filter (z_k, $\forall k = 1$ to 9) [min, max, median]

Noise:

Impulse noise: White intensity values exceeded in the image

Salt & Pepper noise: Black & white intensity values

Gaussian Noise:



Low Pass filters:

Low Pass regions are the smooth regions in an image. (no variation in pixel value)

~~(no variation)~~

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

High Pass region → where intensity (color) change in an image

Averaging filter → $\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

Eg.

$$\begin{matrix} 10 & 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 & 10 \\ 50 & 50 & 50 & 50 & 50 & 50 \\ 50 & 50 & 50 & 50 & 50 & 50 \\ 50 & 50 & 50 & 50 & 50 & 50 \end{matrix}$$

smooth region

High pass region

smooth region

After filtering

$$\begin{matrix} 10 & 10 & 10 & 10 \\ 23.3 & 23.3 & 23.3 & 23.3 \\ 36.6 & 36.6 & 36.6 & 36.6 \\ 50 & 50 & 50 & 50 \end{matrix}$$

smooth region

Blur

smooth region

* To maintain low frequency region, divide by 9.

\# To remove impulse (& salt & pepper) noise, use ~~imulse impulse~~ non-linear filters (median)

padding

\# To maintain size of image, duplicate boundary rows & columns or ~~pad~~ zero padding.

\# For a filter of size $(f * f)$ and image of $(n \times n)$, padding size

$$(n + 2p - f + 1) = n$$

$$\boxed{p = \frac{f-1}{2}}$$

\# Stride = 1

Replace negative value with 0 & positive value with 255

Page No.
Date

High Pass filter :

* Mask contains +ve & -ve weights
* Sum of the mask elements is 0.

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 4 & -1 \\ \hline 0 & -1 & 0 \\ \hline\end{array}$$

scaling function

Eg

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100

Soln with padding

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
-30	-270/9	-270/9	-270/9	-270/9	-270/9	-270/9	-30
30	+270/9	+270/9	+270/9	+270/9	+270/9	+270/9	30
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

replace -ve with 0 & +ve with 255

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
255	255	255	255	255	255	255	255
0	0	0	0	0	0	0	0

~~# Multiply~~

High Boost Filter :

* To retain background of image
  Multiply positive value
  with positive constant
  ~~A > C > 1~~
  C > 1.

$$\frac{1}{9} \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 8*c & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

Eg C = 1.1

8/9	8/9	8/9	8/9	8/9	8/9
8/9	8/9	8/9	8/9	8/9	8/9
-262/9					
250/9					
80/9					

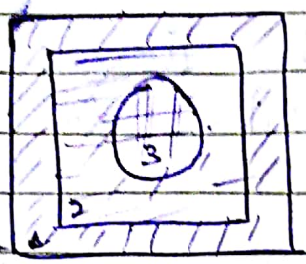
Increase constant value ~~so that~~ to get exact value of background pixel.

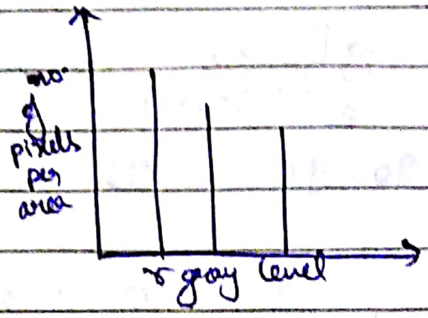
Eg ~~c = 2.125~~ C = 2.125

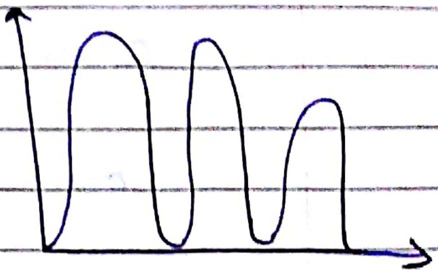
10	10	10	10	10	10
10	10	10	10	10	10
-20	-20	-20	-20	-20	-20
130	130	130	130	130	130
100	100	100	100	100	100
100	100	100	100	100	100

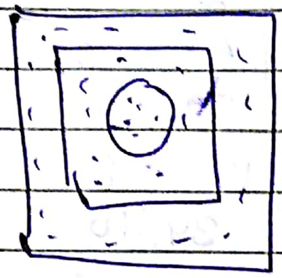
Scanned by CamScanner

Gaussian Noise:

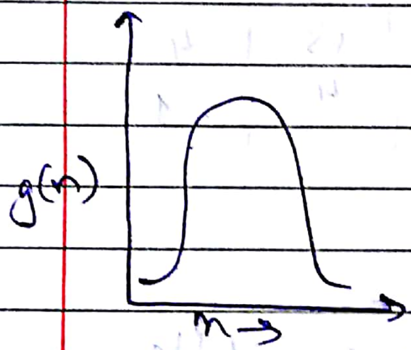








Gaussian Filter / Smoothing Filter



$$g(n) = e^{\frac{-n^2}{2\sigma^2}}$$

PDF of gaussian random variable

$$g[x,y] = c\, e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

Discrete gaussian distribution

c ← normalizing constant

For $\sigma^2 = 2$, $n = 7$

Mask Kernel → OR Gaussian Kernel

	-3	-2	-1	0	1	2	3
-3	0.011	0.038	0.082	0.105	0.082	0.038	0.011
-2	0.038	0.135	0.286	0.368	0.286	0.135	0.038
-1	0.082	0.286	0.606	0.779	0.606	0.286	0.082
0	0.105	0.368	0.779	1	0.779	0.368	0.105
1	0.082	0.286	0.606	0.779	0.606	0.286	0.082
2	0.038	0.135	0.286	0.368	0.286	0.135	0.038
3	0.011	0.038	0.082	0.105	0.082	0.038	0.011

$$\frac{g[3,3]}{c} = e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

$$c = \frac{g[3,3]}{e^{-(x^2+y^2)/(2\sigma^2)}} = \frac{1}{0.011}$$ [corner value should be 1]

$c = 90.91 = 91$

Gaussian Filter

scaling value

	-3	-2	-1	0	1	2	3
-3	1	4	8	10	8	4	1
-2	4	13	27	34	27	13	4
-1	8	13	56	71	56	13	8
0	10	34	71	91	71	34	10
1	8	13	56	71	56	13	8
2	4	13	27	34	27	13	4
3	1	4	8	10	8	4	1

* This filter smoothens image.
* Scaling ~~filter~~ function = sum of gaussian filter = 1086 (1115)
* To retain uniform intensity.
* Averaging filter → blur image
  Gaussian filter → sharp image

Ques For $n = 5$, $\sigma^2 = 1$

	-2	-1	0	1	2
-2	~~0.018~~ 1	5	8	5	1
-1	5	21	34	21	5
0	8	34	55	34	8
1	5	21	34	21	5
2	1	5	8	5	~~0.018~~ 1