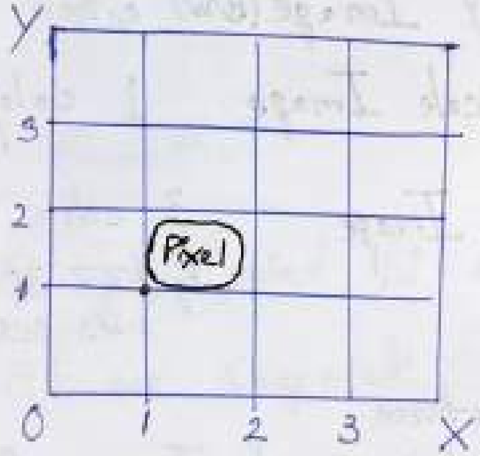
Pixel: Smallest individual element in an image

Image: Visual Representation of Something

Digital Image: Visual Representation of Something captured using some digital medium

$f(x,y)$



$f(1,1)$ indicates the position of the highlighted pixel

Image

Basic Steps of DIP:

Phase I / Pre-Processing

Image Acquisition ⇨ Enhancement ⇨ Filters ⇨ Segmentation

⇩

Representation ⇦ Decision ⇦ Classification / Detection of ROI (Region of Interest) ⇦ Feature Selection / Extraction

Phase II / Post-Processing

Pixel

R G B

Red Green Blue

0 - 255

256^3

Types of Image:

Each pixel has/is

1. Binary Image (B/W) either black or white
2. Grayscale Image 1 color value of Gray (0-255)
3. Color Image 3 color values for R.G.B
   Eg (177, 30, 250)
   0-255 0-255 0-255
   256^3 combinations

Difference Between

Image Analysis and Image Processing:

Image Analysis	Image Processing
Involves converting images into measurements.	Involves changing images (in ways that will help interpretation)

Digital Image: A representation of two-dimensional image as a finite set of digital values, called picture elements or pixels.

Each pixel has values (grey levels or colors)

0 - Black

1 - White

Common Image Format:

1. 1 sample per point [1 color channel]
   (B&W (0/1) or Grayscale (0-255))
2. 3 samples per point [3 color channels]
   (Red, Green, Blue)
3. 4 samples per point
   Red, Green, Blue, Alpha

   0-1

   transparent opaque

Each pixel is a sample of an original image

DPI & PPI :

DPI [Dots per inch] : A measure of printers' density of dot placement

PPI [Pixels per inch] : Number of pixels per square inch

Bits Per Pixel:

Number of distinct colors that can be represented by a pixel depends on the number of bits per pixel

n BPP = 2^n colors

1 bpp image has $2^1 = 2$ colors (monochrome)

8 bpp image has $2^8 = 256$ colors

* 16 bpp image has $2^{16} = 65,536$ colors (High color)
  ↳ 256^2

* 24 bpp image has $2^{24} = 16,777,216$ colors (True Color)
  ↳ 256^3
  (All RGB combinations)

Color Depth: Numbers of bits allocated to red, green & blue

15 bpp means 15 bits are used to represent each pixel.

Five bits for red, green and blue each

16 bpp

ⓐ 5 bits for red and blue

ⓑ 6 bits for green

Note: Red and Blue will have equal number of bits, with the remaining extra bits for Green. [Reason: Human eye is more sensitive to errors in green than in red/blue]

If transparency is considered, (Otherwise divide between red, green & blue only)

Extra bits are used for transparency or Alpha channel.

∴ 16 bpp ⇒ 5 for red, 5 for green, 5 for blue
1 for transparency

Digital Image Processing Focus:

① Improvement of pictorial information for human interpretation

② Processing of image data for storage, transmission and representation for autonomous machine perception

Input of DIP → Image
output of DIP → Image

Arithmatic Operations:

$$h(i,j) = f(i,j) + g(i,j)$$

① Addition $[h=f+g]$: Noise Reduction

② Subtraction $[h=f-g]$: Detection of changes/ ~~(bet^n f & g)~~
Motion Detection

③ Multiplication $[h=f\times g]$: Feature/Object Isolation

④ Division $[h=f/g]$: Illumination

◫ Key Stages in DIP:

Image Acquisition

Image Enhancement

Image Restoration

Morphological Processing

Segmentation

Object Recognition

Representation & Description

3 Levels of DIP:

① Low Level

- Input : Image
- Output : Image
- Example : Enhancement, De-noising

② Mid Level

- Input : Image
- Output : Object, Regions
- Example: Image Segmentation

③ High Level

- Input: Image
- Output: Class Labels
- Example : Recognition, Classification

Image Acquisition

03 main elements

(i) Illuminating Source

(ii) Scene

(iii) Sensors

Image Attributes

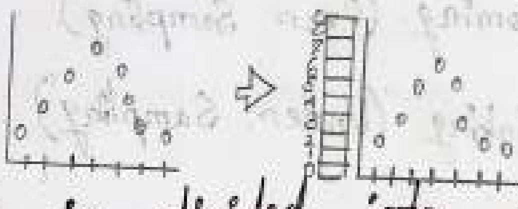
(1) A location $(x,y) \longrightarrow$ pixel

(2) A value $f(x,y)$ at pixel $(x,y) \longrightarrow$ gray level

* ## Image Digitization:

① Sampling Digitizing the coordinates values is called sampling

② Quantization Digitizing the amplitude values is called quantization



Intensity scale is divided into 08 discrete intervals [$\because 2^8 = 256$]

Image Size:

$\boxed{b = M \times N \times bpp}$

M= Height of Image
N= Width of Image
bpp= Bits per pixel (depends on no. colors/intensity values)

[Slide-77] Example:

Image Size= 1024x 512

Intensity Values = 72

What is the size of the image?

$6\ bpp = 2^6 = 64$ colors /intensity values (Tight Compression)

$7\ bpp = 2^7 = 128$ colors /intensity values (Loose Compression)

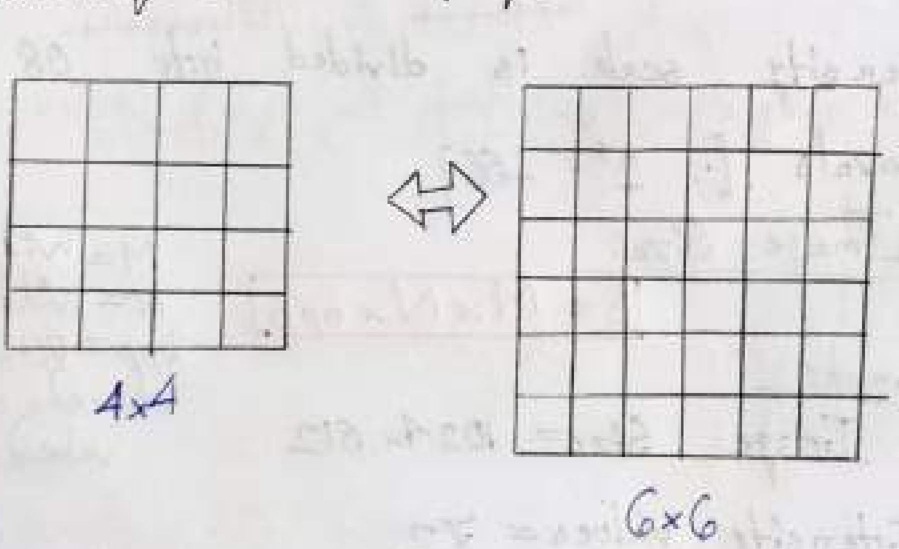
∴ Image Size = $1024 \times 512 \times 6$ or 7 bits

Image Interpolation:

The process of using known data to estimate values at unknown locations.

Zooming (Over Sampling) 8×8 to 12×12

Shrinking (Under Sampling) 12×12 to 8×8



* Neighbors of a Pixel:

03 types

1. $N_4(p)$: 4 neighbors +
2. $N_D(p)$: Diagonal neighbors ×
3. $N_8(p)$: 8 neighbors ✳

Pixel Adjacency:

Adjacency depends on

- Neighborhood (N_4, N_D, N_8)
- Pixel Gray values

03 types

1. 4 Adjacency
2. 8 Adjacency
3. m Adjacency

Common Requirement ⇨ p & q must have same pixel gray level value

4-adjacency

Pixels p and q are 4-adjacent if

① p & q has same gray level value AND

② $N_4(p) = q$

8-adjacency

Pixels p and q are 8-adjacent if

① p & q has same gray level value AND

② $N_8(p) = q$

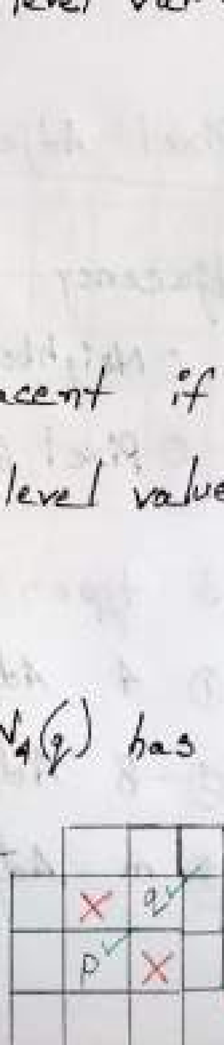
m-adjacency

Pixels p and q are m-adjacent if

① p & q has same gray level value AND

② $N_4(p) = q$ OR

③ $N_D(p) = q$ AND $N_4(p) \cap N_4(q)$ has no pixel with same gray level value



√→ Same gray level value

X→ Different gray level value

Image CANNOT be reconstructed from Histogram plot

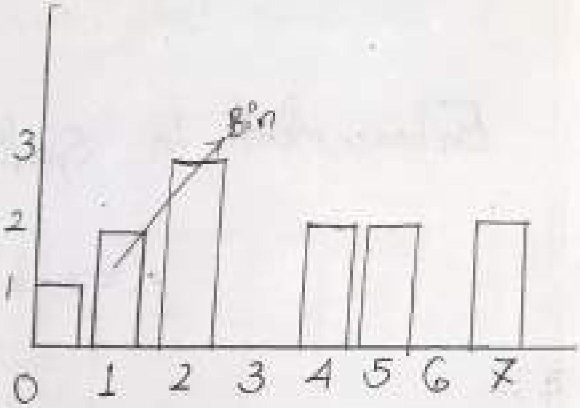
See 2nd Slide-35 Example

Histogram Plot ⇨ Plots the frequency of gray level values in an image

7	5	1
4	2	0
2	2	1

Image

⇨



h:

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

2nd Slide-43

- Dark Image ⟶ Bins clustered in low end (Around 0)
- Bright Image ⟶ Bins clustered in high end (Around 255)
- Low Contrast Image → Bins situated in a narrow range
- High Contrast Image → Bins scattered throughout entire range (0-255)

Normalized Image Histogram [Similar to Probability]

h:

0	1	2	3	4	5	6	7
$\frac{1}{9}$	$\frac{2}{9}$	$\frac{1}{3}$ (3/9)	0	$\frac{1}{9}$	$\frac{1}{9}$	0	$\frac{1}{9}$

Explanation: Divide original histogram values by total number of pixels.

Mask:

Image Enhancement:

Enhancement in Spatial Domain

$$g(x,y) = T[f(x,y)]$$

$f(x,y)$ = input image
$g(x,y)$ = output image

T = An operator defined over a neighbourhood of a point (x,y)

Spatial Filtering

Filter size must be much smaller than original image size

Types

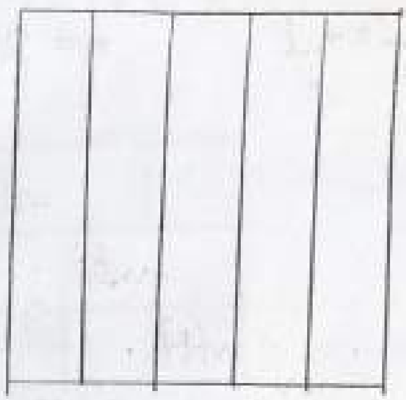
① Linear Spatial Filter

② Non-linear Spatial Filter

Bit Plane Slicing

Gray level value of each pixel stored as 1 or more bytes

6	7	6	6	7
0	0	0	1	2



Max gray level value = 7

$\therefore$ Bits required to represent = $(2^3 = 8) = 3$

$\therefore$ Bit planes = 3

Spatial Filtering

Mask / Filter / Template / Kernel / Window

Mask size $\Rightarrow m \times n$ [Always odd number]

$m = 2a+1$ $\quad n = 2b+1$ $\quad$ a & b are any integers

		w(1,1)
	w(0,0)	
w(-1,-1)		

Mask Coefficient

Spatial Correlation

f

0 0 0 1 0 0 0 0

w

1 2 3 2 8

$0\times1 + 0\times2 + 0\times3 + 0\times2 + 1\times8 = 8$

0 0 0 8 2 3 2 1

Zero Padding

Same on right side

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

Mirror Padding 0100

Mirror Padding 0001

1 2 3 2 8

1 2 3 2 8

1 2 3 2 8

1 2 3 2 8

Output Image Bit size = Input Image Bit size

Note: In answer script, denote which padding used
1st align w with f, then calculate how many padding bits required

Practice

f: 0 1 3 2 1 0 8 5

w: 1 2 3 2 8

$0\times1+0\times2+0\times3+0\times2+0\times8=0$

$0\times1+0\times2+0\times3+0\times2+1\times8=8$

0 8 26 25 23 15 74 60

0 0 0 0 0 1 3 2 1 0 8 5 0 0 0 0

1 2 3 2 8

1 2 3 2 8

1 2 3 2 8

1 2 3 2 8

1 2 3 2 8

1 2 3 2 8

1 2 3 2 8

1 2 3 2 8

Spatial Convolution

Reverse the mask (1 2 3 2 8 → 8 2 3 2 1)

Repeat steps of correlation

Example:

Convolute 00010000 by 12328
↓
8 2 3 2 1

0 0 0 1 2 3 2 8

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

8 2 3 2 1

8 2 3 2 1

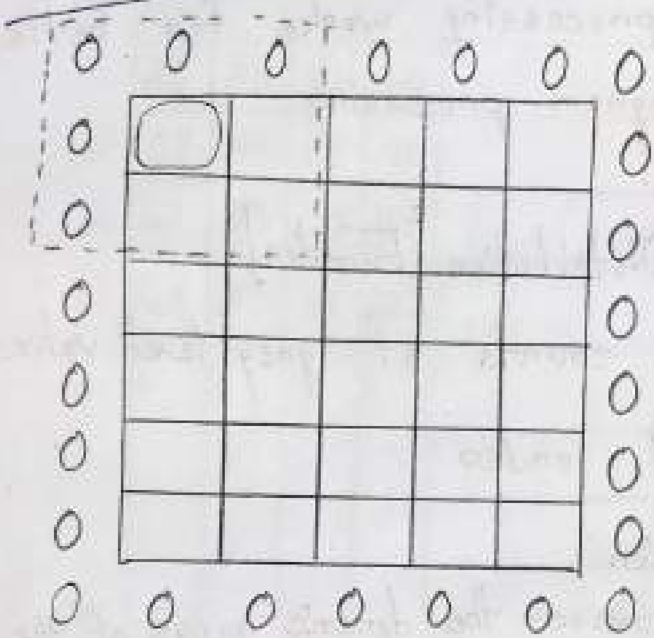
8 2 3 2 1

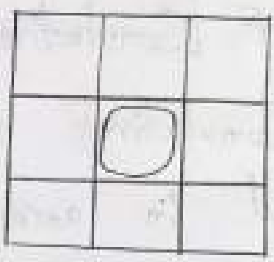
8 2 3 2 1

8 2 3 2 1

8 2 3 2 1

2D Correlation and Convolution :





Histogram Processing:

Local histogram processing works far better than Global histogram processing

CDF [Cumulative Distribution Function]

Cumulatively add counts of grey level values listed in ascending order

Histogram Equalization

A method which increases the dynamic range of the grey-level in a low-contrast image to cover full range of grey-levels

General Histogram Equalization Formula

$$h(v) = round\left(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L-1)\right)$$

v = The grey level value for which is being calculated

cdf_{min} = cdf value of minimum grey level value

L = Number of grey level values (usually 256)

M = Width N = Height

Note: cdf(v) obtained from cdf Table

Example:

Image

52	55	61	66				
63	59	55	90				
62	59	68	113				
63	58	71	122				

Count Table

Value (v)	cdf	Value (v)	cdf
52	1	71	1
55	2	90	1
58	1	113	1
59	2	122	1
61	1		
62	1		
63	2		
66	1		
68	1		

cdf Table:

Value(v)	cdf
52	1
55	3
58	4
59	6
61	7
62	8
63	10

Value(v)	cdf
66	11
68	12
71	13
90	14
113	15
122	16

$$h(52) = \text{round}\left(\left(\frac{cdf(52) - cdf_{min}}{4 \times 4 - cdf_{min}}\right) \times (256-1)\right) = \text{round}\left(\frac{1-1}{16-1} \times 255\right)$$

$$= 0$$

$$h(55) = \text{round}\left(\frac{3-1}{16-1} \times 255\right) = 34$$

$$h(58) = \text{round}\left(\frac{4-1}{16-1} \times 255\right) = 51$$

$$h(59) = \text{round}\left(\frac{6-1}{16-1} \times 255\right) = 85$$

$$h(61) = \text{round}\left(\frac{7-1}{16-1} \times 255\right) = 102$$

$$h(62) = \text{round}\left(\frac{8-1}{16-1} \times 255\right) = 119$$

$$h(63) = \text{round}\left(\frac{10-1}{16-1} \times 255\right) = 153$$

$$h(66) = \text{round}\left(\frac{11-1}{16-1} \times 255\right) = 170$$

$$h(68) = \text{round}\left(\frac{12-1}{16-1} \times 255\right) = 187$$

$$h(71) = \text{round}\left(\frac{13-1}{16-1} \times 255\right) = 204$$

$$h(90) = \text{round}\left(\frac{14-1}{16-1} \times 255\right) = 221$$

$$h(113) = \text{round}\left(\frac{15-1}{16-1} \times 255\right) = 238$$

$$h(122) = \text{round}\left(\frac{16-1}{16-1} \times 255\right) = 255$$

Final Image

0	34	102	170
153	85	34	221
119	85	187	238
153	51	204	255

Q. 1st Order & 2nd Order X

Q. Apply Sharpening Spatial Filter

Sharpening Spatial Filter:

To highlight fine detail in an image

Sharpening Vs Blurring

Image Sharpening

○ 1st Order Derivative

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

○ 2nd Order Derivative

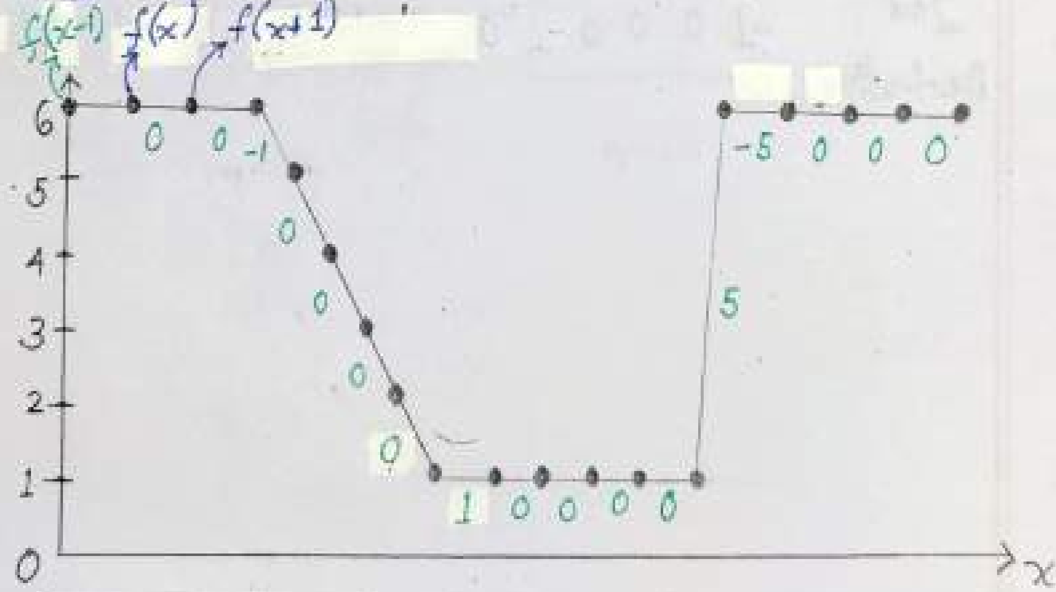
$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

Note: Start from 2nd point always ()

Value of 1st Order Derivative	Value of 2nd Order Derivative
① 0 at constant grey level	① 0 at constant grey level
	② Non-Zero at onset and end of
$f(x+1) - f(x)$	$f(x+1) + f(x-1) - 2f(x)$

Slide - 134

Q. Either Scan line or Graph given. Calculate 1st & 2nd Derivatives



Scan Line

1st Derivative

2nd Derivative

Slide-135

Image Strip	5 5 4 3 2 1 0 0 0 6 0 0 0 0 1 3 1 0 0 0 0 7 7 7 7
1st Derivative	-1 -1 -1 -1 -1 0 0 6 -6 0 0 0 1 2 -2 -1 0 0 0 7 0 0 0
2nd Derivative	-1 0 0 0 0 1 0 6 -12 6 0 0 1 1 -4 1 1 0 0 7 -7 0 0

Ramp Roof Roof step

Spatial Filtering

Mask / Filter / Template / Kernel / Window

Mask size ⇒ $m \times n$ [Always odd number]

$m = 2a + 1$ $n = 2b + 1$ a & b are any integer

		$w(1,1)$
	$w(0,0)$	
$w(-1,-1)$		

Mask Coefficient

Spatial Correlation

f

0 0 0 1 0 0 0 0

w

1 2 3 2 8

$0\times1 + 0\times2 + 0\times3 + 0\times2 + 1\times8$
$= 8$

$\boxed{0\ 0\ 0\ 8\ 2\ 3\ 2\ 1}$

Zero Padding

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

Same on right side

Mirror Padding

0100

1 2 3 2 8

1 2 3 2 8

1 2 3 2 8

1 2 3 2 8

Mirror Padding

0001

1st digit, w, with f, ...
bits required

Practice

f

0 1 3 2 1 0 8 5

w

1 2 3 2 8

$0\times1+0\times2+0\times3+0\times2+0\times8=0$

$0\times1+0\times2+0\times3+0\times2+1\times8=8$

0 8 26 25 23 15 24 60

0 0 0 0 0 1 3 2 1 0 8 5 0 0 0 0

1 2 3 2 8

1 2 3 2 8

1 2 3 2 8

1 2 3 2 8

1 2 3 2 8

1 2 3 2 8

1 2 3 2 8

1 2 3 2 8

Spatial Convolution

Reverse the mask (1 2 3 2 8 → 8 2 3 2 1)

Repeat steps of correlation

Example:

Convolute 0001000 by 12328

↓

8 2 3 2 1

0 0 0 1 2 3 2 8

0000 0001000 0000

8 2 3 2 1

8 2 3 2 1

8 2 3 2 1

8 2 3 2 1

8 2 3 2 1

8 2 3 2 1

Q. 1st Order & 2nd Order X

Q. Apply Sharpening Spatial Filter

Sharpening Spatial Filter:

To highlight fine detail in an image

Differentiating **Sharpening Vs Blurring** Averaging

Sharpening	Blurring
○ Spatial Differentiating	○ Pixel Averaging
○ Used to highlight fine details and transitions in intensity	○ Used for noise reduction

Image Sharpening (Done for edge)

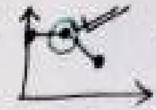

edge

○ 1st Order Derivative

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

○ 2nd Order Derivative

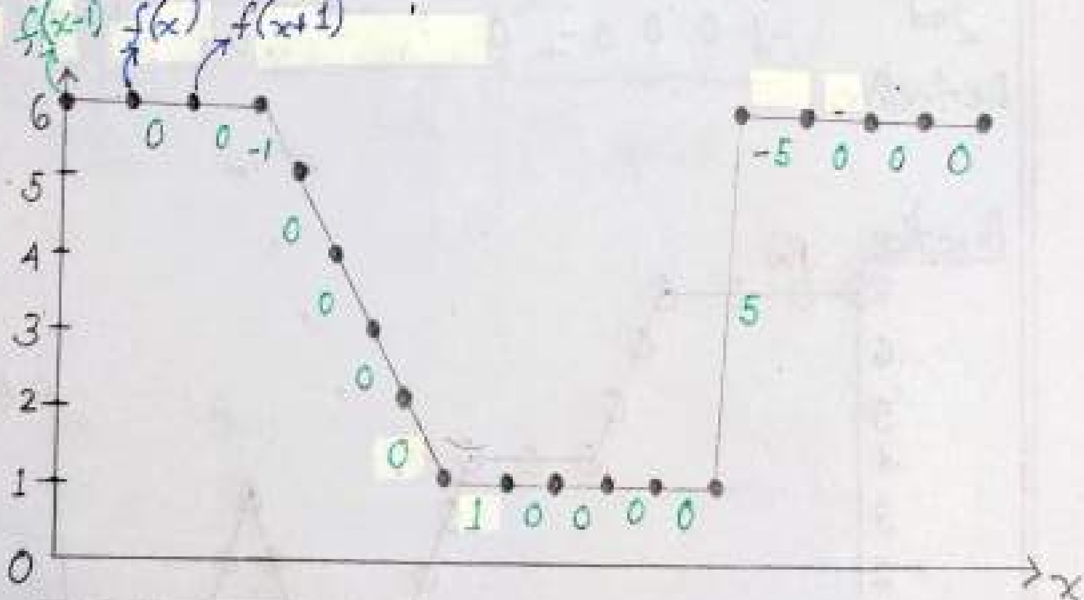
$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

Note: Start from 2nd point always ()

Value of 1st Order Derivative	Value of 2nd Order Derivative
① 0 at constant grey level	① 0 at constant grey level
② Non-Zero at onset of step and ramp	② Non-Zero at onset and end of step and ramp
③ Non-Zero along ramp	③ 0 along ramp
$f(x+1) - f(x)$	$f(x+1) + f(x-1) - 2f(x)$

Slide - 134

* Q. Either Scan line or Graph given. Calculate 1st & 2nd Derivatives



Scan Line

1st Derivative

2nd Derivative

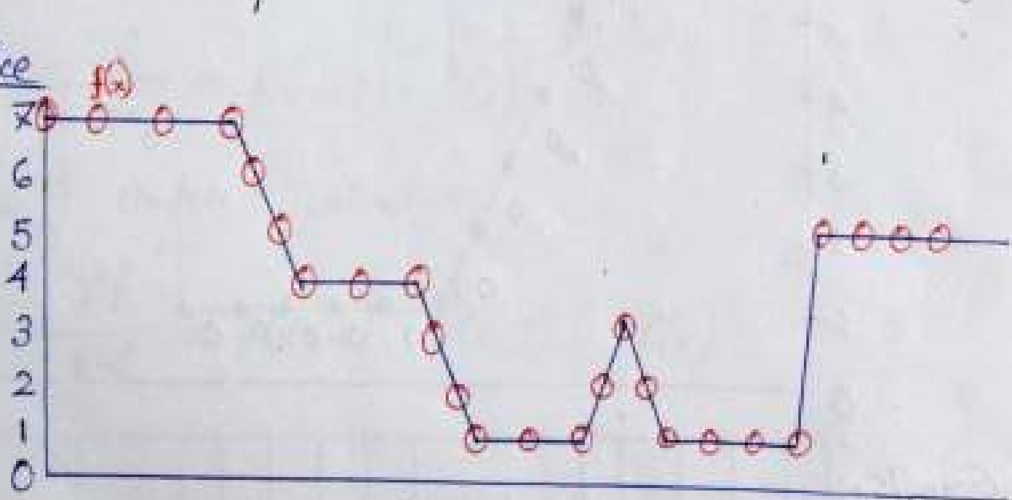
Slide-135

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

1st Derivative -1 -1 -1 -1 -1 0 0 6 -6 0 0 0 1 2 -2 -1 0 0 0 7 0 0 0

2nd Derivative $\underbrace{-1\ 0\ 0\ 0\ 0\ 1\ 0}_{\text{Ramp}}\ \underbrace{6\ -12\ 6}_{\text{Roof}}\ 0\ 0\ \underbrace{1\ 1\ -4\ 1\ 1}_{\text{Roof}}\ 0\ 0\ \underbrace{7\ -7}_{\text{step}}\ 0\ 0$

Practice



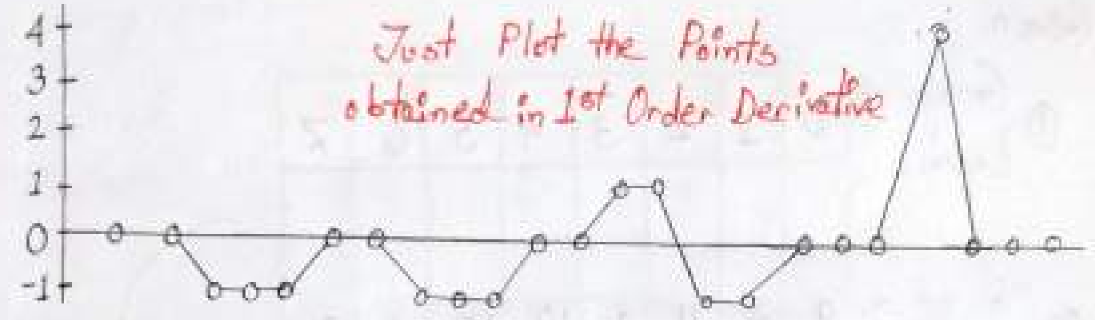
Scan Line 7 7 7 7 6 5 4 4 4 3 2 1 1 1 2 3 2 1 1 1 1 5 5 5 5

1st Order Derivative 0 0 -1 -1 -1 0 0 -1 -1 -1 0 0 1 1 -1 -1 -1 0 0 0 4 0 0 0

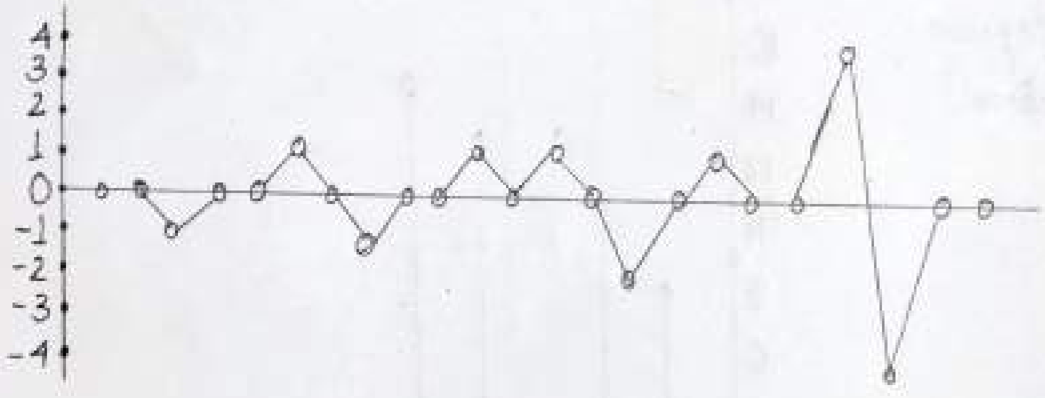
2nd Order Derivative 0 0 -1 0 0 1 0 -1 0 0 1 0 1 0 -2 0 1 0 0 4 -4 0 0

1st Order & 2nd Order Graph

1st Order Graph



2nd Order Graph



Laplacian Filter

Isotropic ⇒ Having a physical property which has the same value when measured in different directions

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

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

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

⇓

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

0	1 $x,y+1$	0
1 $x-1,y$	4	1 $x+1,y$
0	1 $x,y-1$	0

Laplacian Masks

0	1	0
1	-4	1
0	1	0

0	-1	0
-1	4	-1
0	-1	0

1	1	1
1	-8	1
1	1	1

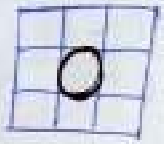
-1	-1	-1
-1	8	-1
-1	-1	-1

Scaling

Negative values cannot be present in image (realistically). So, replace with 0 (the lowest possible pixel gray level)

But theoretically (in Exam), show negative value

Simplification of Laplacian



$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) & \text{if center of coefficient} < 0 \\ f(x,y) - \nabla^2 f(x,y) & \text{if center of coefficient} > 0 \end{cases}$$

Previously,

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

$$\therefore g(x,y) = f(x,y) - \nabla^2 f(x,y)$$

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

0	-1	0
-1	5	-1
0	-1	0

-1	-1	-1
-1	9	-1
-1	-1	-1

Unsharp Masking ×

Frequency Domain ×

Example (Question: Calculate value for one pixel. Image & mask given. Padding may be required Replication or Zero padding)

Input Image

8	5	4
0	6	2
1	3	7

Mask

0	1	0
1	-4	1
0	1	0

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

$= 5 - 24 + 2 + 3$

$= -14$

Final Image

8	5	4
0	-14	2
1	3	7

Replication Padding Example

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

Enhanced Laplacian Filters [① Convert Mask ② Apply as like before]

1	1	1
1	-8	1
1	1	1

$\Rightarrow$

1	1	1
1	-9	1
1	1	1

Increase Magnitude of Center Point of Mask

0	1	0
1	4	1
0	1	0

$\Rightarrow$

0	1	0
1	5	1
0	1	0

Example

50	50	50	50	100	100	100	100
50	50	50	50	100	100	100	100
50	50	50	50	100	100	100	100
50	50	50	50	100	100	100	100
100	100	100	100	50	50	50	50
100	100	100	100	50	50	50	50
100	100	100	100	50	50	50	50
100	100	100	100	50	50	50	50

Input Image

1	1	1
1	-8	1
1	1	1

Mask

0	0	150	-150	0	0
0	0	150	-150	0	0
150	150	200	-200	-150	-150
-150	-150	-200	200	150	150
0	0	-150	150	0	0
0	0	-150	150	0	0

Output Image

Smoothing Spatial Filter: or Averaging filters or Lowpass filters

- Used for blurring and noise reduction
- Blurring used in pre-processing 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 and also by non-linear filtering

Types of Smoothing Spatial Filters

- Smoothing Spatial Filters
  - Linear Filters
    - Mean/Box Filter
    - Weighted Average Filter
    - Gaussian Filter
  - Non-Linear Filters
    - Median Filter
    - Max Filter
    - Min Filter

Linear Filters

① Box / Mean Filter

All coefficients are equal

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

② Weighted Average Filter

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

③ Gaussian Filter

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Non-linear (Order-Statistic) filters

① Median Filter

- Sort pixels in the neighbourhood
- Determine median (Eg: 1 3 6 10 15)
- Replace value of center pixel with median

② Min Filter

Replace value of center pixel with minimum pixel value in neighbourhood

③ Max Filter

Replace value of center pixel with maximum pixel value in neighbourhood

** Only names of filters will be given

Slide 24

Q1. Consider the image below and calculate the output of the pixel (2,2) if smoothing is done using 3x3 neighbourhood using the filters below:

ⓐ Box Filter

ⓑ Weighted Average filter

ⓒ Median filter

ⓓ Min filter

ⓔ Max filter

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

2,2

3x3

Input Image

Solution:

ⓐ Applying Box filter

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

$$\frac{1}{9} \times (1\times7+1\times9+1\times5+1\times4+1\times6+1\times8+1\times2+1\times0+1\times1)$$

$$= \frac{1}{9} \times 42 = 4.66 \approx 5$$

↑ round

Replace 6 with 5

ⓑ Weighted Average filter $\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

$= \frac{1}{16}\left[1\times7+2\times9+1\times5+2\times4+4\times6+2\times8 + 1\times2+2\times0+1\times1\right]$

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

$= 5.0625 \approx 5$ Replace ⑥ with 5 in image

ⓒ Median filter

> Sort values in neighborhood

⇨ 0 1 2 4 5 6 7 8 9

Replace ⑥ with 5

ⓓ Min filter

> Find min value

⇨ 0

Replace ⑥ with 0

ⓔ Max filter

> Find max value

⇨ 9

Replace ⑥ with 9

Filters Will be given in exam (hopefully)

First Order Derivative Filters

Mainly Used for Edge Detection

① Roberts Operator

-1	0
0	1

② Sobel Operator

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

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

③ Prewitt Operator

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

* Mark the pixel position of update & mask placement

Q2. Apply Roberts, Sobel and Prewitt's Operator on pixel (1,1) in the following image

50	50	100	100
50	50	100	100
50	50	100	100
50	50	100	100

Take any point of Robert's Operator as center

① Applying Robert's Operator [MUST Mark Center Pixels of Operators, specially for Roberts]

-1	0
0	1

$= 50 \times -1 + 100 \times 0 + 50 \times 0 + 1 \times 100$

$= 50$

② Applying Sobel's Operator

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

$= 50 \times (-1) + 50 \times (-2) + 100 \times (-1) + 50 \times 0 + 50 \times 0 + 100 \times 0 + 50 \times 1 + 50 \times 2 + 100 \times 1$

$= 0$

③ Applying Prewitt Operator

-1	-1	-1
0	0	0
1	1	1

$= 50\times(-1) + 50\times(-1) + 100\times(-1)$

$+ 50\times 0 + 50\times 0 + 100\times 0$

$+ 50\times 1 + 50\times 1 + 100\times 1$

$= 0$

Non-linear operation that correlates with shape or morphology of features

Morphological Processing

applied on Binary Image

Morphological processing pursues the goal of removing imperfections that arise when converting a greyscale image to a binary image

Defn: A collection of non-linear operations related to the shape or morphology of features in an image.

Structuring Element

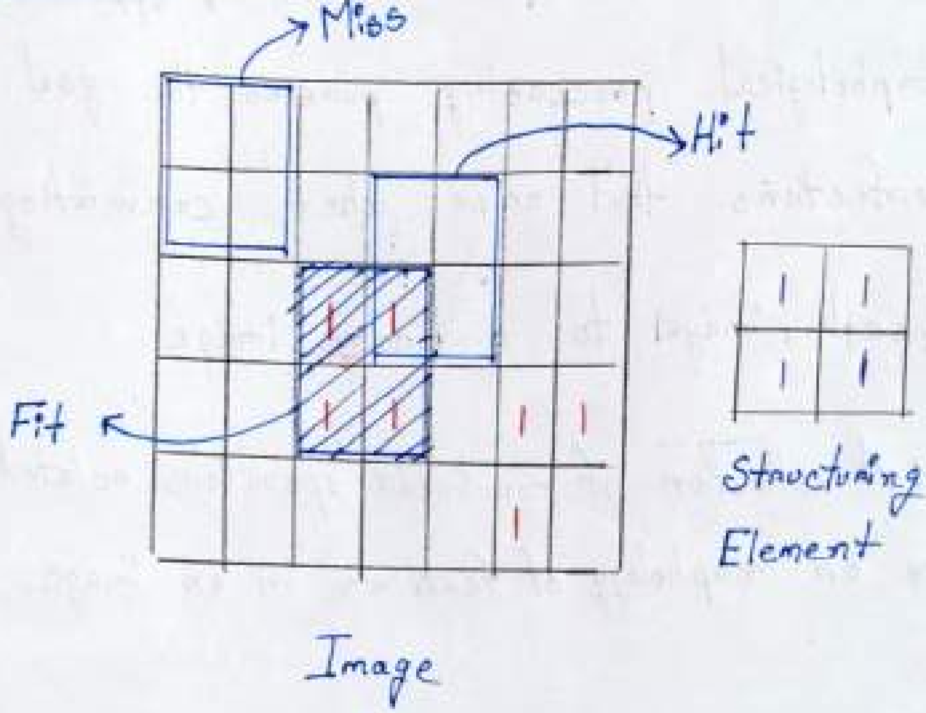
depends on number of 1s

Morphological techniques probe an image with a small shape/template (a small matrix of pixels each with a value of zero or one) called a Structuring Element

Matrix Dimensions = Size of the Structuring Element

Pattern of Ones/Zeros = Shape of the Structuring Element

Fit, Hit & Miss:



Calculating Origin of Structuring Element:

Consider (1,1) as first position of structuring element

Put values of ~~last~~ size coordinate in formula

[Mention in Exam which value you've used]

Mention starting point used (0,0) or (1,1)

Mention Structuring Element Size

$x = \text{Floor}[(\text{Number of Rows} + 1)/2]$

$y = \text{Floor}[(\text{Number of Column} + 1)/2]$

0,0		
	1,1	
		2,2

3x3

Formula [Center = (x,y)]

6x5 structuring element

$x = \text{Floor}[(6+1)/2]$

$= 3$

$y = \text{Floor}[(5+1)/2]$

$= 3$

Show Like This In Exam

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

6x5

Dilation & Erosion:

Dilation ⇨ A fundamental morphological operation that increases foreground depending on a structuring element

$f \oplus s$

Erosion ⇨ A fundamental morphological operation that shrinks/removes foreground depending on a structuring element

$f \ominus s$

(objects larger than S.E) (objects smaller than S.E)

Steps

① Calculate Center of Origin of Structuring Element (Draw SE on plain paper)

② Padding according to Structuring Element & its center

③ Observe Fits, Hits and Misses

Erosion

The erosion of a binary image f, by a structuring element s (denoted by $f \ominus s$) produces a new binary image $g = f \ominus s$ with ones in all locations (x,y) of a structuring element's origin at which that structuring element s fits the input image f.

i.e $g(x,y) = 1$ if s fits f

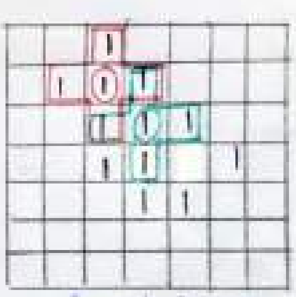
$\quad\quad\quad\;\; 0$ otherwise

[Considering background → 0 (black)
foreground → 1 (white)]

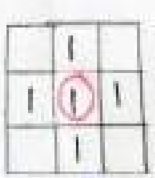
repeating for all pixel coordinates (x,y)

Shortcut:

1. In input image, find how many S.E present inside
2. For each S.E present in the input image, place 1 in center of origin
3. All other pixels will be 0.



input image



S.E

Draw Output Image & Structuring Element only
(Only Draw 1's)
No need to draw 0's

Erosion with 3x3 Structuring Element

Two techniques

① Only keep 1's (in center pixel) where the Structuring Element fits

Only for 3x3

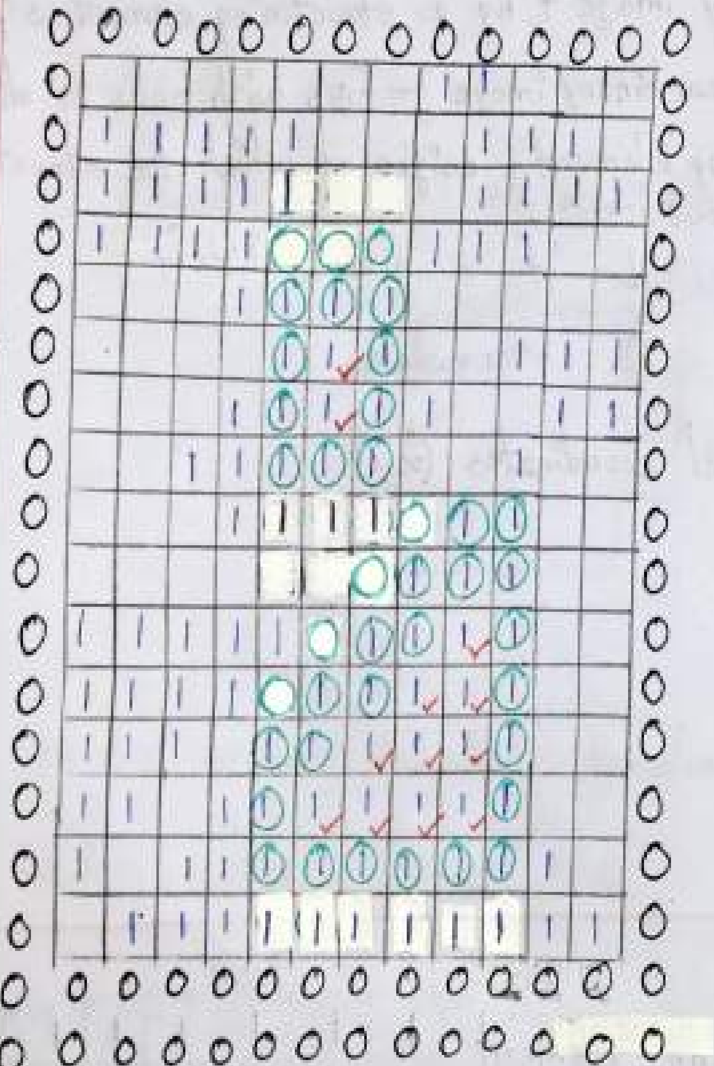
$$\begin{array}{ccc} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array}$$

S.E

② Remove all boundary 1's

$$\begin{array}{cccccc} 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{array} \Rightarrow \begin{array}{cc} & 1 \\ 1 & 1 \\ 1 & 1 \end{array}$$

Opening Practice



16x12

	1	2	
1	1	1	1
2	1	1	1
	1	1	1
	1	1	1

4x3

Center of origin (x,y)

$y = \text{floor}[(4+1)/2] = 2$

$x = \text{floor}[(3+1)/2] = 2$

$(x,y) = (2,2)$

Erosion ✓

Dilation ○

Dilation

The dilation of a binary image f, by a structuring element s (denoted by $f \oplus s$) produces a new binary image $g = f \oplus s$ with ones in all locations (x,y) of a structuring element's origin at which the structuring element s hits the input image f.

ie $g(x,y) = \begin{cases} 1 & \text{if } s \text{ hits } f \\ 0 & \text{otherwise} \end{cases}$

repeating for all pixel coordinates (x,y)

Shortcut:

4x6 structuring element

$$\begin{array}{cccccc|c}
\multicolumn{6}{c}{4\times6} & \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & \\
1 & 1 & 1 & 1 & 1 & 1 & \\
1 & 1 & 1 & 1 & 1 & 1 & 2 \\
\multicolumn{2}{c}{2} & & & \multicolumn{2}{c}{3} &
\end{array}$$

In input image,

For every 1

① Fillup 2 rows above & 1 rows below

② Fillup 3 columns left & 2 columns right

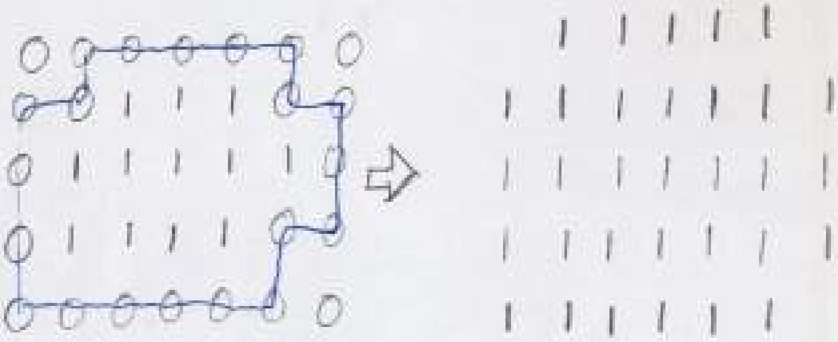
3x3 ⇨ Fill up / Increase one pixels around boundary

5x5 ⇨ Fill up / Increase two pixels around boundary

Dilation with 3x3 Structuring Element

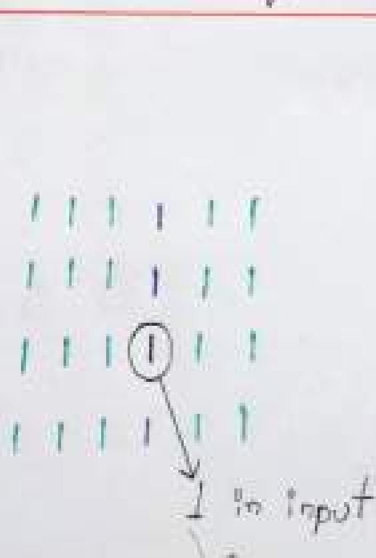
Technique

① Place 1's around boundary



Note: When drawing boundary, use only horizontal — and vertical lines | NOT diagonal lines

Note 02: Draw boundary twice if S.E is 5x5



• → 3 columns left, 2 columns right

• → 2 rows above, 1 row below

Opening & Closing

Opening ⇨ Erosion followed by Dilation $f \circ s = (f \ominus s) \oplus s$

Closing ⇨ Dilation followed by Erosion $f \bullet s = (f \oplus s) \ominus s$

Practice (Opening)

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \qquad \begin{array}{|c|} \hline 1 \\ \hline 1 \\ \hline 1 \\ \hline \end{array}$$

image

structuring element

$$\boxed{g = (f \ominus s_e) \oplus s_d}$$

Mention formula in exam *

* this part must be written in exam

1) Erosion

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

Eroded Image

2) Dilation of Eroded Image

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

Final Result after Morphological Opening

Practice (Closing)

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

image

$$\begin{matrix}1\\1\\1\end{matrix}$$

structuring element

$$\boxed{g = (f \oplus s) \ominus s}$$

① Dilation

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

Dilated Image

② Erosion of Dilated Image

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

Final Result after Morphological Closing

Two types of Morphological Filtering

① Top-hat filtering

Performed on grayscale/binary image and returns a filtered image

Computes the morphological opening of the image and then subtracts the result from the original image

$$\begin{matrix} 0&0&0&0&0&0\\ 0&0&1&1&0&0\\ 0&1&1&1&1&0\\ 0&0&1&1&0&0\\ 0&0&0&0&0&0 \end{matrix} \longrightarrow \begin{matrix} 0&0&0&0&0&0\\ 0&0&1&1&0&0\\ 0&0&1&1&0&0\\ 0&0&1&1&0&0\\ 0&0&0&0&0&0 \end{matrix} \longrightarrow \begin{matrix} 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&1&0&0&1&0\\ 0&0&0&0&0&0\\ 0&0&0&0&0&0 \end{matrix}$$

Original Image — Result Image after Opening — Result Image after Top-hat filtering

② Bottom-hat filtering

Performed on greyscale/binary image and returns a filtered image
Computes the morphological closing of the image and then subtracts the original image from the result

Segmentation [Last step of Pre-Processing]

Refers to the process of partitioning a digital image into multiple segments based on certain criteria or features

Each segment typically represents a set of pixels having similarities in:-

- intensity
- color
- texture
- shape

Note:

Only (usually) in Binary Image, shape of objects considered

Applications of Segmentation

① Object Recognition and Detection

② Image Compression

③ Image Editing and Enhancement

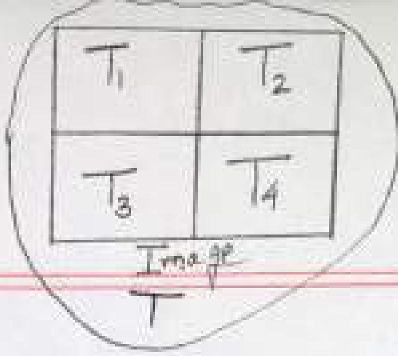
④ Medical Imaging

⑤ Remote Sensing

[4] Types of Segmentation:

① Thresholding

> Threshold value chosen



Local threshold ⇒ T_1, T_2

Global threshold ⇒ T

Image Binarization

One global threshold value T applied

$< T$ ⇨ One class (0 or Black)

$\geq T$ ⇨ Another class (1 or White)

Threshold Selection

Most frequently employed method for determining threshold value is based on histogram analysis of intensity levels.

Problems

1. Valley may be two broad, making it difficult to locate a significant minimum
2. Multiple valleys, multiple minima
3. No visible valley
4. Noise

Notes

Histogram Plotting cannot be done for color image

Color image must be split into its 3 channels and separate binary images must be created then merged.

Over Segmentation

- Required regions removed
  or feature

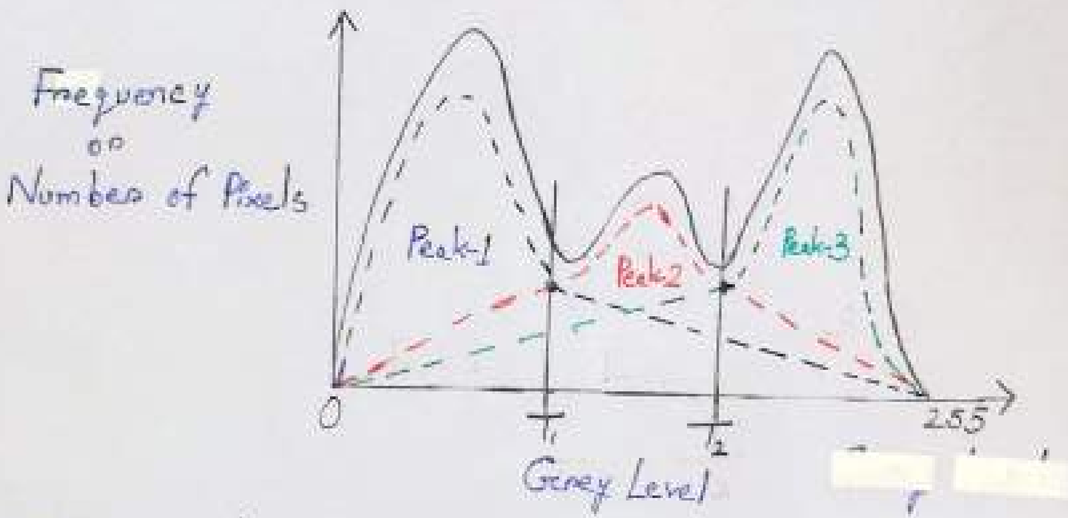
- Threshold value too low

Under Segmentation

- Unwanted regions still present
  or features

- Threshold value too high.

Optimal Threshold Value Selection



- For each peak, draw a curve as shown above from minimum (0) to maximum (255) grey level value
- Each intersection between the curves (of peaks situated next to one another, for example: Peaks 1 and 2 or Peaks 2 and 3) is a potential optimal threshold
- If multiple optimal thresholds found, each must be applied to observe whether over-segmentation or under-segmentation occurs.

Region Growing & Region Splitting

Region-based segmentation is based on the connectivity of similar pixels in a region

Two main approaches to region-based segmentation

① Region-growing

② Region-splitting

Procedure for Region-Growing

Q1:

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

Apply region-growing on the following image with initial point (2,2) and threshold value as 2. Use 4 connectivity

$T=2$

$|7-6| = 1 \leq 2$ ✓

$|7-4| = 3 \not\leq 2$ ×

$|7-5| = 2 \leq 2$ ✓

$|7-3| = 4 \not\leq 2$ ×

$|\text{Seed value} - \text{Pixel value}| \leq \text{Threshold}$

⇩

Same Region

Otherwise, Separate Region

Q1 If specific point given, region growed step-by-step

Q2 Q3 If intensity value given as seed point, region growed step-by step also

0	1	2	0
2	5	6	0
1	4 ×	7	3 ×
0	2	5	1

$|6-2| = 4 \nleq 2$ $|5-2| = 3 \nleq 2$

$|6-5| = 1 \leq 2$ $|5-1| = 4 \nleq 2$

$|6-0| = 6 \nleq 2$

Note: Place × where condition is NOT satisfied

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

$|5-1| = 4 \nleq 2$

$|5-2| = 3 \nleq 2$

$|5-4| = 1 \leq 2$ [This satisfies for 5 but not for 7] no need to even calculate

Note: If even one × DO NOT MERGE

Final Result (Segmented Image)

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

If connectivity not mentioned, 8 way connectivity

* Max difference between pixels in a region must be ≤ Threshold

Q2.

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

Apply region-growing on following image with seed point as 6 (intensity value) and threshold value as 3.

Final Result (Segmented Image)

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

① Put 1's where initial seed points (6) are

② Shortcut: 6-3 = 3
Put 1's where intensity value 3 or higher (≥3) and is 8-neighbor with 6's

③ The 1's put in step-2 are the new seed points. Check with neighbors, if threshold condition satisfied, put 1's in neighbor

** ④ Note: Max difference between pixels in a region must be ≤ Threshold

Q3.

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

Seed Point = 6

Threshold = 3

How many segments identified?

Segmented Image

a b c

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

d

5 segments

Region Splitting

Q4

Apply region-splitting on the following image. Assume the threshold value be $\leqslant 4$

i.e. $|Max - Min| \leqslant 4$ [Don't Split]

$|Max - Min| \nleq 4$ [Split]

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

Max Intensity Level = 7

Min Intensity Level = 0

$7 - 0 \nleq 4$ [∴ Split]

Image divided into a, b, c, d regions. Then, repeat process for each region

ⓐ $7-4 \leqslant 4$

ⓑ $7-2 \nleq 4$ Split

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

ⓒ $3-0 \leqslant 4$

ⓓ $7-0 \nleq 4$ Split

2 b1 b2

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

b4 d1 d2 d3 d4

$7-5 \leq 4$ ✓

$7-4 \leq 4$ ✓

$3-2 \leq 4$ ✓

$6-5 \leq 4$ ✓

$3-2 \leq 4$ ✓

$7-4 \leq 4$ ✓

$3-0 \leq 4$ ✓

$5-4 \leq 4$ ✓

No More Splitting Required

10 Regions

Merging

After region-splitting, check adjacent regions. (8 connectivity)

If in the union of two regions,

Max − Min $\leqslant$ Threshold [Merge]

Max − Min $\nleqslant$ Threshold [Don't Merge]

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

Q. Apply splitting and merging, on the image with threshold value equal to 3

Step-1: Splitting

Condition

Max value − Min value $\leqslant 3$

$7 - 0 \nleqslant 3$

a							b

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

c d

a-region

$7-4 \leqslant 3$

b-region

$7-2 \not\leqslant 3$

c-region

$3-0 \leqslant 3$

d-region

$7-0 \not\leqslant 3$

$\therefore$ Regions b and d split into 4 sub-regions

a b1 b2

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

b3 b4 d1 d2 c d3 d4

b_1-region

$7-5 \leqslant 3$

b_2-region

$7-4 < 3$

b_3-region

$3-2 \leqslant 3$

b_4-region

$6-4 \leqslant 3$

d_1-region

$3-2 \leqslant 3$

d_2-region

$7-5 \leqslant 3$

d_3-region

$3-0 \leqslant 3$

d_4-region

$5-4 \leqslant 3$

Step-2: Merging

a b1 b2

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

b3 b4 d1 d2

c d3 d4

① If in two adjacent regions (8-neighbors),

Max value - Min value $\leq$ Threshold

then, Merge

② However, the condition (Max value - Min value $\leq$ Threshold) must be satisfied by other regions in the merger as well

Threshold = 3

① Regions a and b_1

$7-4 \leq 3$

∴ Regions merged

② Regions a and b_3

$7-2 \not\leq 3$

③ Regions b_1 and b_2

$7-4 \leq 3$

∴ Regions merged

④ Regions b_2 and b_4

$7-4 \leq 3$

∴ Regions merged

⑤ Regions b_4 and d_2

$7-4 \leq 3$

∴ Regions merged

⑥ Regions d_2 and d_1

$7-2 \not\leq 3$

⑦ Regions d_2 and d_4

$7-4 \leq 3$

∴ Regions merged

⑧ Regions d_4 and d_3

$4-0 \not\leq 3$

∴ Regions $a, b_1, b_2, b_4, d_2, d_4$, merged into one region

Note: Regions d_4 and d_1 satisfy the threshold condition. However, d_2 has already been selected and merged, and d_2 and d_1 do not satisfy threshold

** Note: If not specified, start merging from region at (0,0)

① Regions b_3 and d_1

$3-2 \leqslant 3$

$\therefore$ Regions merged

② Regions b_3-d_1 and d_3

$3-0 \leqslant 3$

$\therefore$ Regions merged

③ Regions $b_3-d_1-d_3$ and d_4

$3-0 \leqslant 3$

Regions merged

$\therefore$ Regions b_3, d_1, d_3, c are merged into one region

a b b_3

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

b_4 d_3 d_4 c d_2

Note: Different answers possible depending on which region considered next at each step