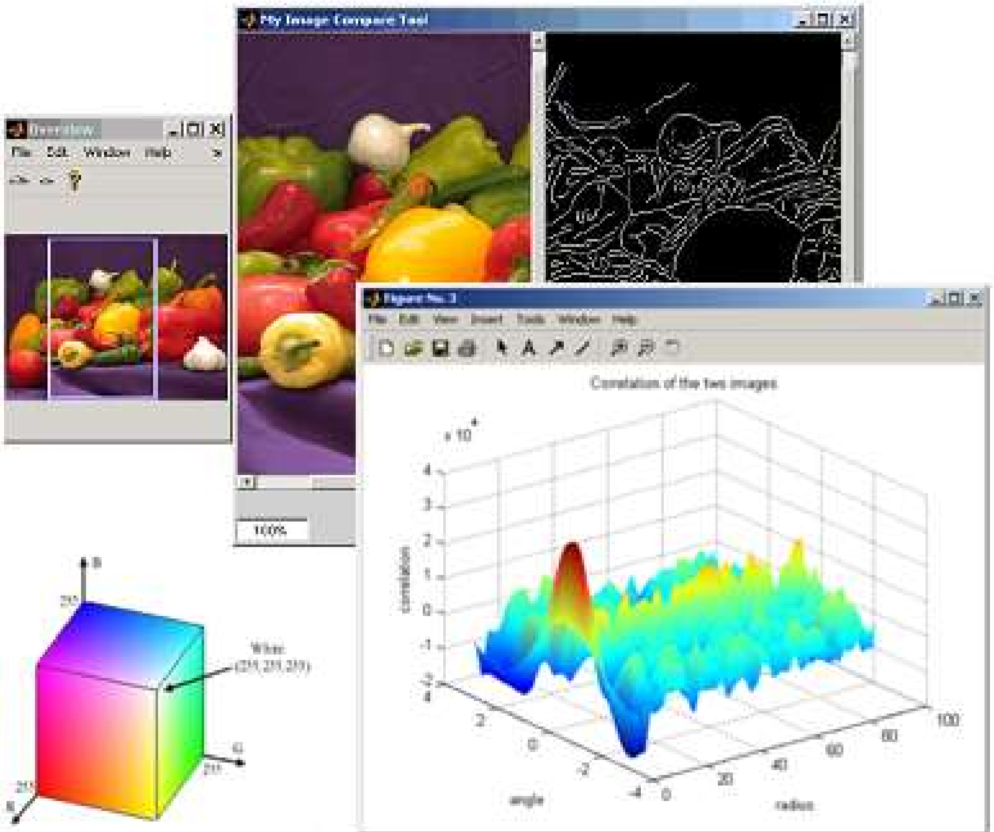
Digital Image Processing



Dr. Ajay Kumar Mahato
Assistant Professor
ECE Department
Gla University Mathura

DIGITAL IMAGE PROCESSING

LECTURE -17

Image Segmentation

Image Segmentation

- Segmentation is the ***process of partitioning a digital image*** into ***multiple regions*** and ***extracting meaningful regions*** known as region of interest (ROI) fir further image analysis. Segmentation is an important phase in the image analysis process.

20	20	10	10	10
20	20	10	10	10
20	20	10	10	10
15	15	10	10	10
15	15	10	10	10

Image Segmentation

20	20	10	10	10
20	20	10	10	10
20	20	10	10	10
15	15	10	10	10
15	15	10	10	10

20	20	10	10	10
20	20	10	10	10
20	20	10	10	10
15	15	10	10	10
15	15	10	10	10

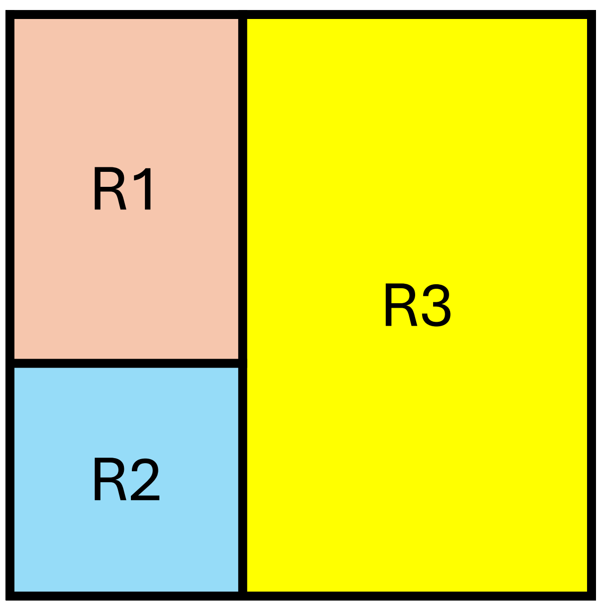


Image Segmentation

❑ The characteristics of segmentation process are the following:

1. If the subregions are combined, the original region can be obtained. Mathematically, it can be stated that $\cup R_i = R$ for $i = 1, 2, \ldots n$. For example, if there are three regions of figure R_1, R_2 and R_3 are combined, the whole region R is obtained.

2. The subregions R_i should be connected. In other words, the region cannot be open-ended during the tracing process.

3. The regions R_1, R_2... R_n do not share any common property. Mathematically, it can be stated as $R_i \cap R_j = \varphi$ for all i and j where $i \neq j$. Otherwise, there is no justification for the region to exist separately.

4. Each region satisfies a predicate or a set of predicates such as intensity or other image statistics that is , the predicate (P) can be colour, grey scale value, Texture, Or any other image statistics, mathematically, this is stated as $P(R_i) = \text{TRUE}$.

Classification of Image Segmentation Algorithm

Image Segmentation

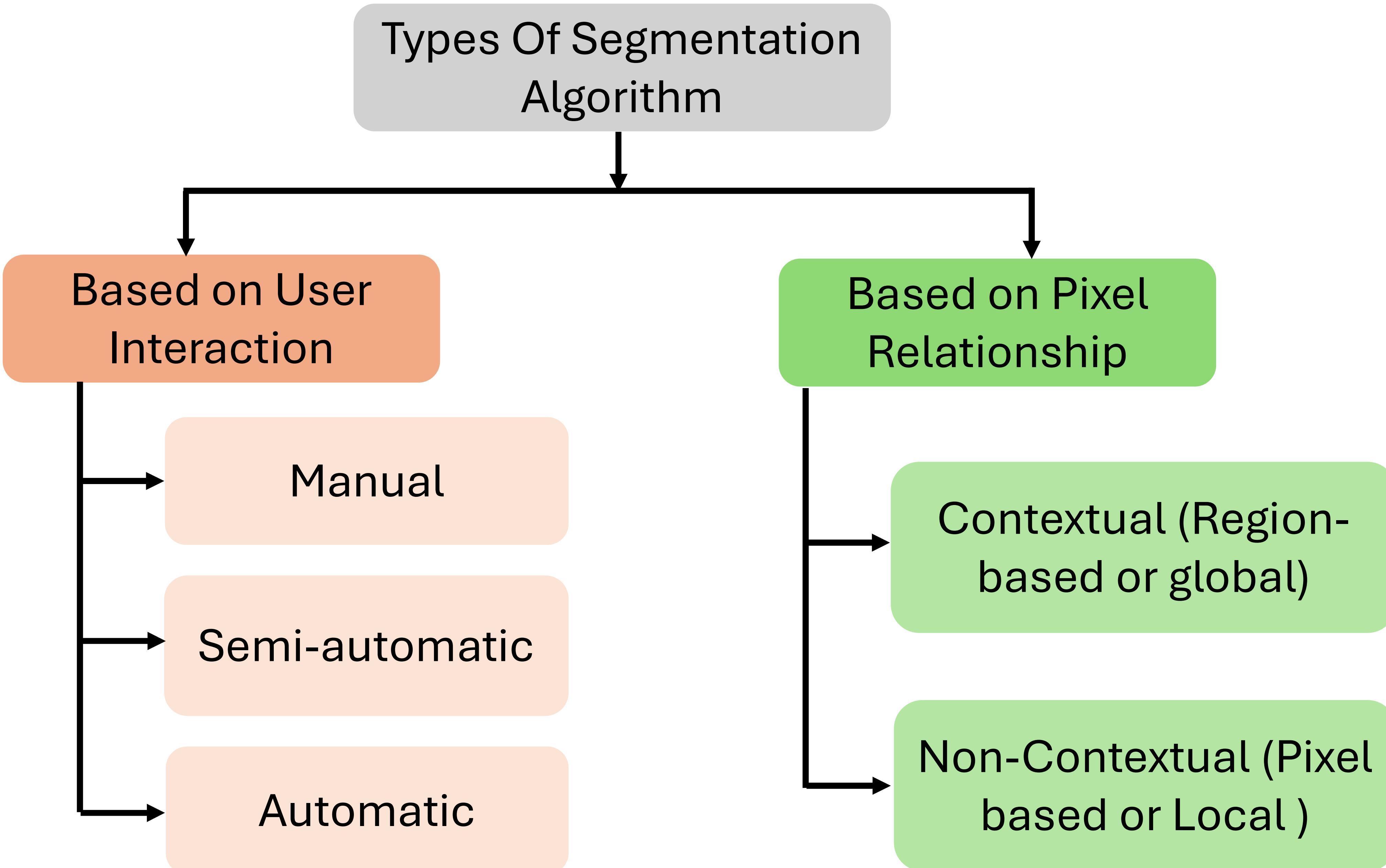


Image Segmentation

- Another way of Classification of Image Segmentation Techniques:

Image Segmentation

- **Discontinuity**
  1. Isolated Points
  2. Lines
  3. Edges
- **Similarity**
  1. Thresholding
  2. Region Growing
  3. Region Spitting and Merging

Image Segmentation

Types Of Segmentation Algorithm

- **Based on User Interaction**
  - Manual
  - Semi-automatic
  - Automatic
- **Based on Pixel Relationship**
  - Contextual (Region-based or global)
  - Non-Contextual (Pixel based or Local)

Image Segmentation

❑ Based on user interaction:

1. **Manual:**

   - In the manual method. the object of interest is observed by an expert who traces its ROI boundaries as well, with the help of software.
   - Hence, the ***decision related to segmentation are made by the by the human observers***.
   - A manual method of extraction is ***time consuming***, highly subjective, ***prone to human error*** and has poor intra-observer reproducibility.
   - However, manual method are still ***used commonly by experts to verify and validate the results of automatic segmentation algorithm.***

Image Segmentation

2. **Automatic:**

- Automatic segmentation algorithm are a preferred choice as they segment the structures of the objects ***without any human intervention.***
- They are preferred if the tasks need to be ***carried out for a large number of images.***

3. **Semiautomatic:**

- Semi-automatic algorithm are a ***combination of automatic and manual algorithm.*** In semi-automatic algorithm, human intervention is required in the initial stage.
- Normally, the ***human observer is supposed to provide the initial seed points*** indicating the ROI.
- Then the extraction process is ***carried out automatically as dictated by the logic of the segmentation algorithm***.

Image Segmentation

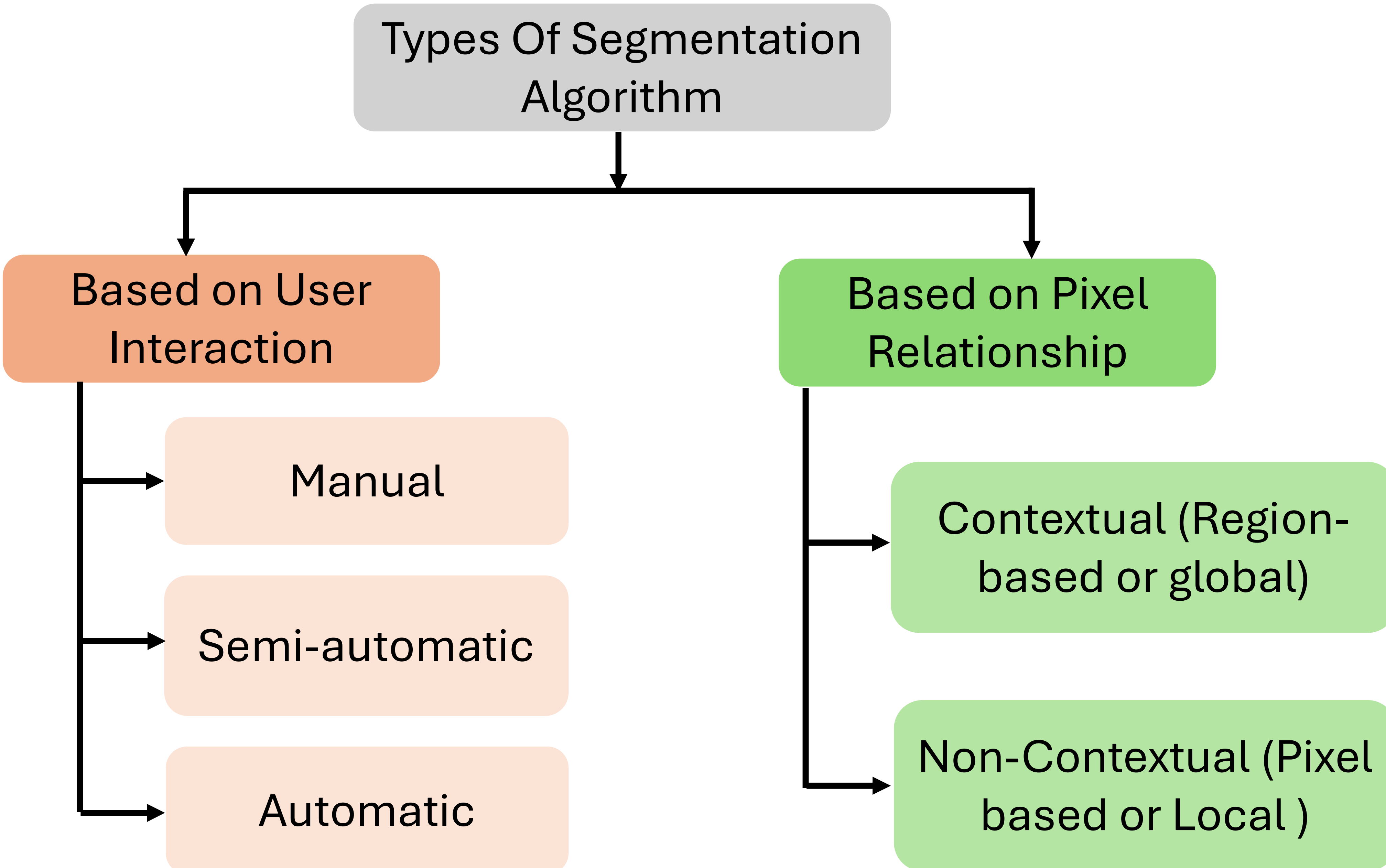


Image Segmentation

❑Based on Pixel relationship:

- Another way of classifying algorithms is to use the criterion of the pixel similarity relationships with neighbouring pixels.

- The similarity relationship can be based on colour, texture, brightness or any other image statistic.

- On this bases segmentation algorithms can be classified as follows:

  a) Contextual (region-based or global) algorithm:

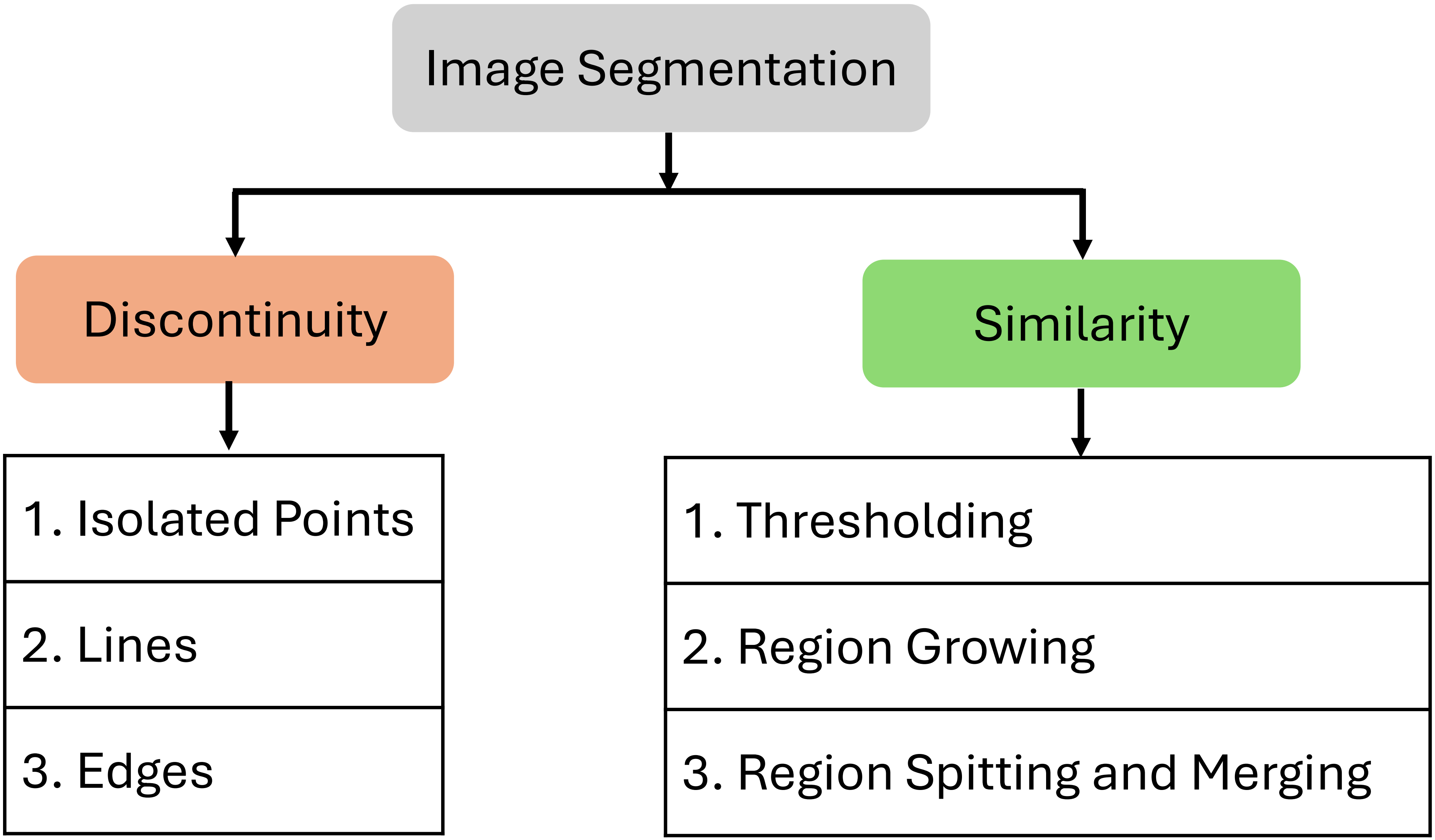
  b) Non-contextual (pixel-based or local) algorithm:

Image Segmentation

- **Contextual (region-based or global) algorithm:**
  - Contextual algorithms group pixels together based on common properties by exploiting the relationship that exist among the pixels.
  - These are also known as region-based or global algorithms. In region-based algorithms. ***The pixels are grouped based on some sort of similarity that exists between them***
- **Non-contextual (pixel-based or local) algorithm:**
  - Non-contextual algorithms are also known as pixel based or local algorithms. These algorithms ignore the relationship that exists between the pixels or features.
  - Instead, ***the data is to identify the discontinuities that are present in the image such as isolated lines and edges***. These are then simply grouped into a regions based on some global-level property. Example: Intensity based thresholding

Image Segmentation

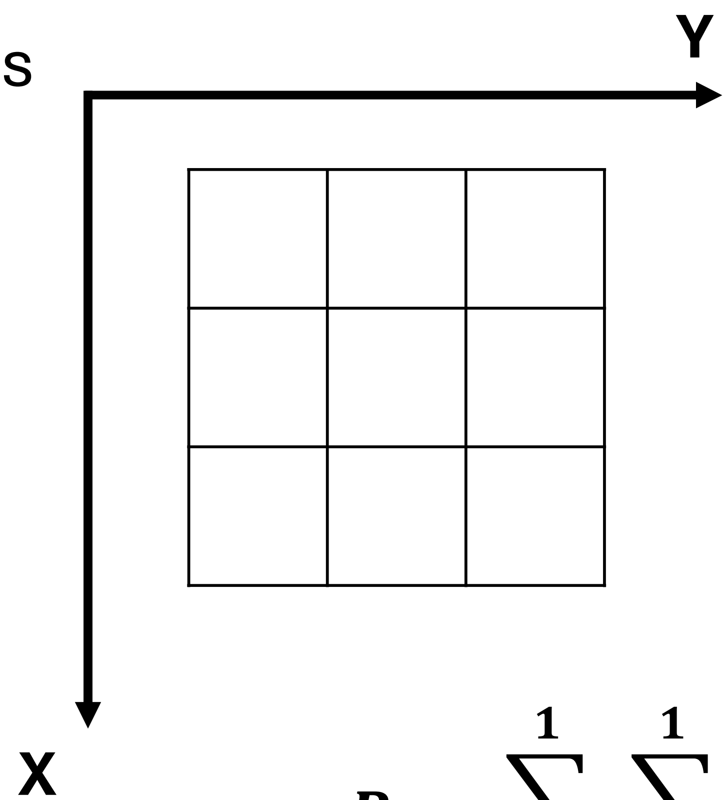
- **Another way of Classification of Image Segmentation Techniques:**



Detection of Discontinuity

Detection of Discontinuity

- The three basic types of grey level discontinuities in a digital image are the followings:
  - Isolated Points
  - Lines
  - Edges



W(-1,-1)	W(-1,0)	W(-1,1)
W(-1,0)	W(0,0)	W(0,1)
W(1,-1)	W(1,0)	W(1,1)

$$R = \sum_{i=-1}^{1} \sum_{j=-1}^{1} W(i,j) \,.\, f(x+i, y+j)$$

Point Detection

Isolated Point Detection

Example

Mask

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

Image 1

1	1	1
1	1	1
1	1	1

Image 2

1	1	1
1	0	1
1	1	1

Image 3

0	0	0
0	0	0
0	0	0

Isolated Point Detection

❑ Isolated Point Detection:

- An isolated point is a point whose grey level is significantly different from its background in a homogenous area.
- Mask Used for isolated point detection is as follows:

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

- If |R| > T, where T is non-negative threshold. Then an isolated point has been detected; and R is the sum of products of the coefficients with the gray levels contained in the region encompassed by the mark.

Isolated Point Detection

Example 1:

7	7	7	7	7	7	7
7	10	7	7	7	7	7
7	7	7	7	7	7	7
7	7	7	7	7	7	7
7	7	7	7	4	7	7
7	7	7	7	7	7	7
7	7	7	7	7	7	7

I

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

P

Isolated Point Detection

-3	-3	-3	0	0	0	0
-3	24	-3	0	0	0	0
-3	-3	-3	0	0	0	0
0	0	0	3	3	3	0
0	0	0	3	-24	3	0
0	0	0	3	3	3	0
0	0	0	0	0	0	0

$I^o P$

3	3	3	0	0	0	0
3	24	3	0	0	0	0
3	3	3	0	0	0	0
0	0	0	3	3	3	0
0	0	0	3	24	3	0
0	0	0	3	3	3	0
0	0	0	0	0	0	0

$|I^o P|$

Line Detection

Line Detection

- The next level of complexity is to detect lines.
- The masks below will extract lines that are one pixel thick and running in a particular direction
- Masks used for Line Detection are as follows:

-1	-1	-1
2	2	2
-1	-1	-1

Horizontal

-1	-1	2
-1	2	-1
2	-1	-1

+45°

-1	2	-1
-1	2	-1
-1	2	-1

Vertical

2	-1	-1
-1	2	-1
-1	-1	2

-45°

- Suppose at a certain line on the image $|R_i| > |R_j| \ \forall\ j \neq i$, then that line is more likely to be associated with the orientation of the mask.
- The final maximum response is defined by $max_{i=1}^{R}\{Ri\}$ and the line is associated with that mask.

Line Detection

Step 1:

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	3
1	1	1	1	1	1	3	1
1	1	1	1	1	3	1	1
3	3	3	3	3	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	1	1

I

-1	-1	-1
2	2	2
-1	-1	-1

H

Line Detection

0	0	0	0	0	0	-2	-4
0	0	0	0	0	-2	2	6
0	0	0	0	-2	2	0	0
-6	-6	-6	-6	0	0	2	-2
12	12	12	12	6	2	-2	0
-6	-6	-6	-6	-4	-2	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$I^o H$

0	0	0	0	0	0	2	4
0	0	0	0	0	2	2	6
0	0	0	0	2	2	0	0
6	6	6	6	0	0	2	2
12	12	12	12	6	2	2	0
6	6	6	6	4	2	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$|I^o H|$

Line Detection

Step 2:

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	3
1	1	1	1	1	1	3	1
1	1	1	1	1	3	1	1
3	3	3	3	3	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	1	1

$\boldsymbol{I}$

-1	-1	2
-1	2	-1
2	-1	-1

$\boldsymbol{D_{45^0}}$

Line Detection

0	0	0	0	0	0	-2	-4
0	0	0	0	0	-2	-4	6
0	0	0	0	-2	-4	12	0
0	0	0	0	0	12	-4	-2
0	0	0	0	6	-4	-2	0
0	0	0	0	-4	-2	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$I^{o}D_{45^{o}}$

0	0	0	0	0	0	2	4
0	0	0	0	0	2	4	6
0	0	0	0	2	4	12	0
0	0	0	0	0	12	4	2
0	0	0	0	6	4	2	0
0	0	0	0	4	2	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$\left|I^{o}D_{45^{0}}\right|$

Line Detection

Step 3:

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	3
1	1	1	1	1	1	3	1
1	1	1	1	1	3	1	1
3	3	3	3	3	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	1	1

I

-1	2	-1
-1	2	-1
-1	2	-1

V

Line Detection

0	0	0	0	0	0	-2	2
0	0	0	0	0	-2	2	0
0	0	0	0	-2	2	0	0
0	0	0	0	0	0	2	-2
0	0	0	0	0	2	-2	0
0	0	0	0	2	-2	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$$I^o V$$

0	0	0	0	0	0	2	2
0	0	0	0	0	2	2	0
0	0	0	0	2	2	0	0
0	0	0	0	0	0	2	2
0	0	0	0	0	2	2	0
0	0	0	0	2	2	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$$|I^o V|$$

Line Detection

Step 4:

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	3
1	1	1	1	1	1	3	1
1	1	1	1	1	3	1	1
3	3	3	3	3	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	1	1

I

2	-1	-1
-1	2	-1
-1	-1	2

D_{-45^0}

Line Detection

0	0	0	0	0	0	4	2
0	0	0	0	0	4	-4	0
0	0	0	0	4	-4	0	-6
0	0	0	0	-6	0	-4	4
0	0	0	0	0	-4	4	0
0	0	0	0	2	4	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$$I^{o}D_{-45^{0}}$$

0	0	0	0	0	0	4	2
0	0	0	0	0	4	4	0
0	0	0	0	4	4	0	6
0	0	0	0	6	0	4	4
0	0	0	0	0	4	4	0
0	0	0	0	2	4	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$$\left|I^{o}D_{-45^{0}}\right|$$

Line Detection

0	0	0	0	0	0	2	4
0	0	0	0	0	2	2	6
0	0	0	0	2	2	0	0
6	6	6	6	0	0	2	2
12	12	12	12	6	2	2	0
6	6	6	6	4	2	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

0	0	0	0	0	0	2	4
0	0	0	0	0	2	4	6
0	0	0	0	2	4	12	0
0	0	0	0	0	12	4	2
0	0	0	0	6	4	2	0
0	0	0	0	4	2	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

0	0	0	0	0	0	4	4
0	0	0	0	0	4	4	6
0	0	0	0	4	4	12	6
6	6	6	6	6	12	4	4
12	12	12	12	6	4	4	0
6	6	6	6	4	4	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

0	0	0	0	0	0	2	2
0	0	0	0	0	2	2	0
0	0	0	0	2	2	0	0
0	0	0	0	0	0	2	2
0	0	0	0	0	2	2	0
0	0	0	0	2	2	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

0	0	0	0	0	0	4	2
0	0	0	0	0	4	4	0
0	0	0	0	4	4	0	6
0	0	0	0	6	0	4	4
0	0	0	0	0	4	4	0
0	0	0	0	2	4	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$$L = max\{|\boldsymbol{I^o H}|, |\boldsymbol{I^o D_{45^0}}|, |\boldsymbol{I^o V}|, |\boldsymbol{I^o D_{-45^0}}|\}$$

Edge Detection

Edge Detection

❑ Edge:

- It is a boundary between ***two regions having distinct intensity levels*** or ***having distinct grey levels.***
- It play a very important role in many image processing applications. They provide an ***outline of an object.***
- **In the physical plane**, edges corresponds to the ***discontinuities in depth, surface orientation, change in material properties and light variations.***
- These variations are present in the image as gray scale discontinuities.
- An edge is a set of connected pixels that lies on the boundary between two regions that differ in grey value.
- **Most edges are unique in space,** that is, their position and orientation remain the same in space when viewed from different points.

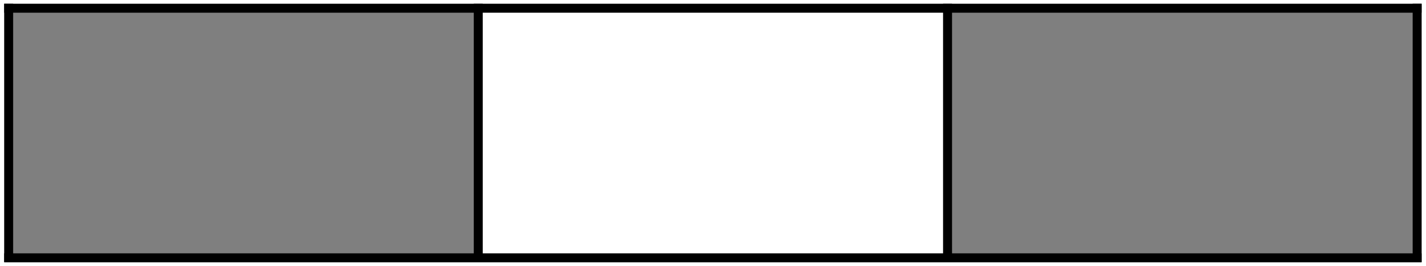
Edge Detection

Example 1

Example 2

Edge Detection

Example 3:



	0	1	2	3	4	5	6	7	8
$f(x)$	0	0	0	0	1	0	0	0	0
1st Derivative $= f(x+1) - f(x)$									
2nd Derivative $= f(x+1) + f(x-1) - 2f(x)$									

Edge Detection

Example 3:

	0	1	2	3	4	5	6	7	8
f(x)	1	1	1	1	0	1	1	1	1
1st Derivative = f(x+1) – f(x)									
2nd Derivative = f(x+1) +f(x-1) -2f(x)									

GLA UNIVERSITY MATHURA
Accredited with A+ Grade by NAAC
12-B Status from UGC

Edge Detection

❑ Observation From 1st Derivative and 2nd Derivative

First (1st) Derivative	▪ It is positive at the leading edge (Dark to Brighter) ▪ and Negative at the tailing edge (Brighter to Dark).
Second (2nd) Derivative	▪ Changes its sign from positive to negative, when pixel value changes from dark to bright. ▪ Changes its sign from negative to positive, when pixel value changes from bright to dark.

Example

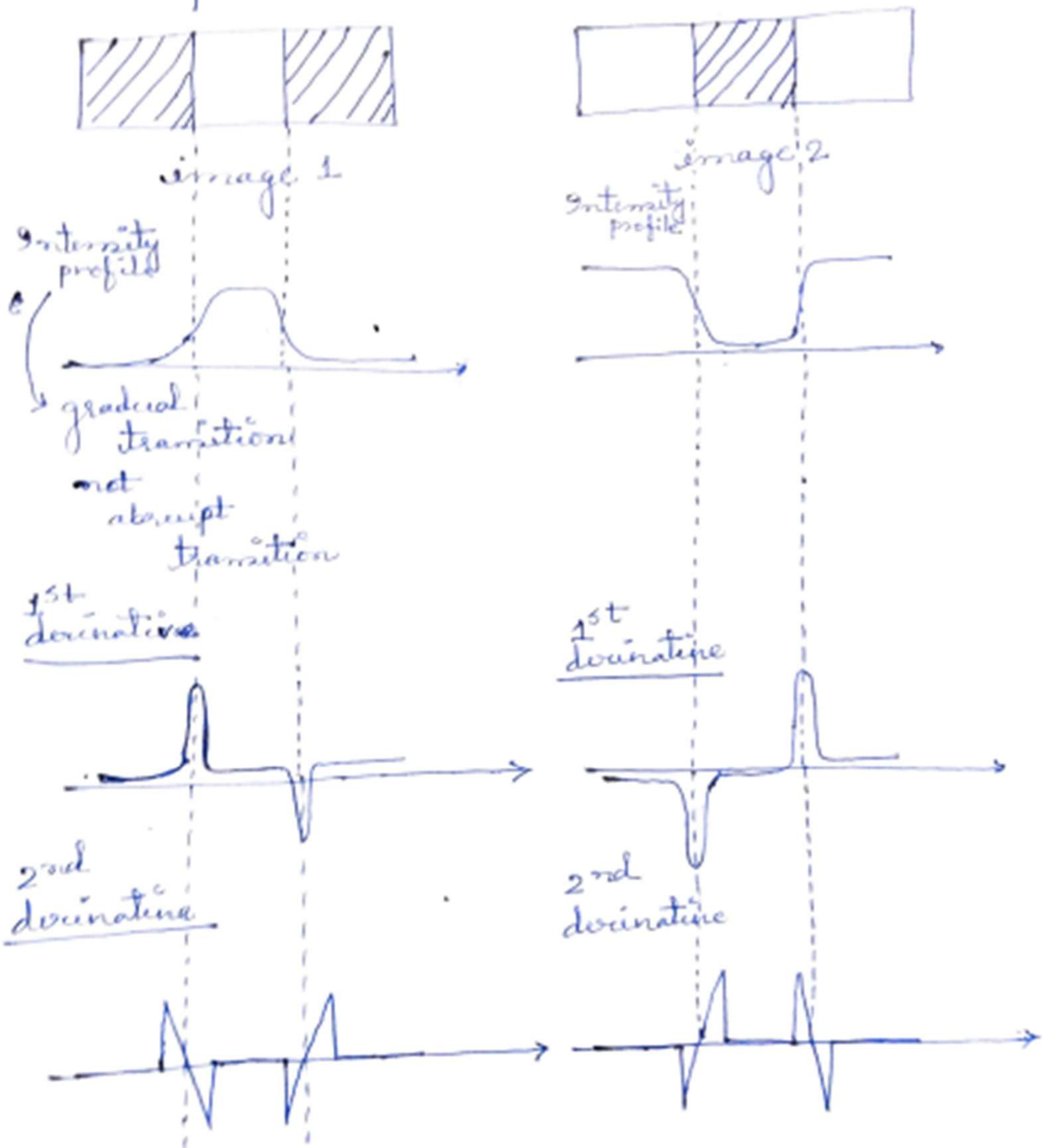


image 1
image 2
Intensity profile
Intensity profile
gradual transition
not abrupt transition
1st derivative
1st derivative
2nd derivative
2nd derivative

Edge Detection

- 2nd derivative is ***very very sensitive to the noise*** and that is the reason that the 2nd derivative operators are not usually used for edge detection operation.

- However, 2nd derivative can be used for ***some secondary information i.e.***

  - **The sign of the 2nd derivative** can be sued to determine whether the point is lying on the darker side of the edge or a point ins lying on the brighter side of the e edge.

  - **Moreover, there are some zero crossing points,** theses zero crossing points can be used to exactly identify the location of an edge whenever there is a gradual transition of the intensity from dark to bright or bright to dark.

1st Derivative

1st Derivative

- Here we use the **GRADIENT OPERATOR**

$$\overrightarrow{\nabla f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

g_x

-1	0
1	0

1. $\nabla f = magnitude\ of\ \left(\overrightarrow{\nabla f}\right)$

$$= \left[G_x^2 + G_y^2\right]^{1/2} \cong |G_x| + |G_y|$$

g_y

-1	1
0	0

2. **Direction of** $\overrightarrow{\nabla f} = \alpha\,(x, y) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$

1^{st} Derivative (Gradient)

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

1^{st} Derivative (Gradient)

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

out1 =

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

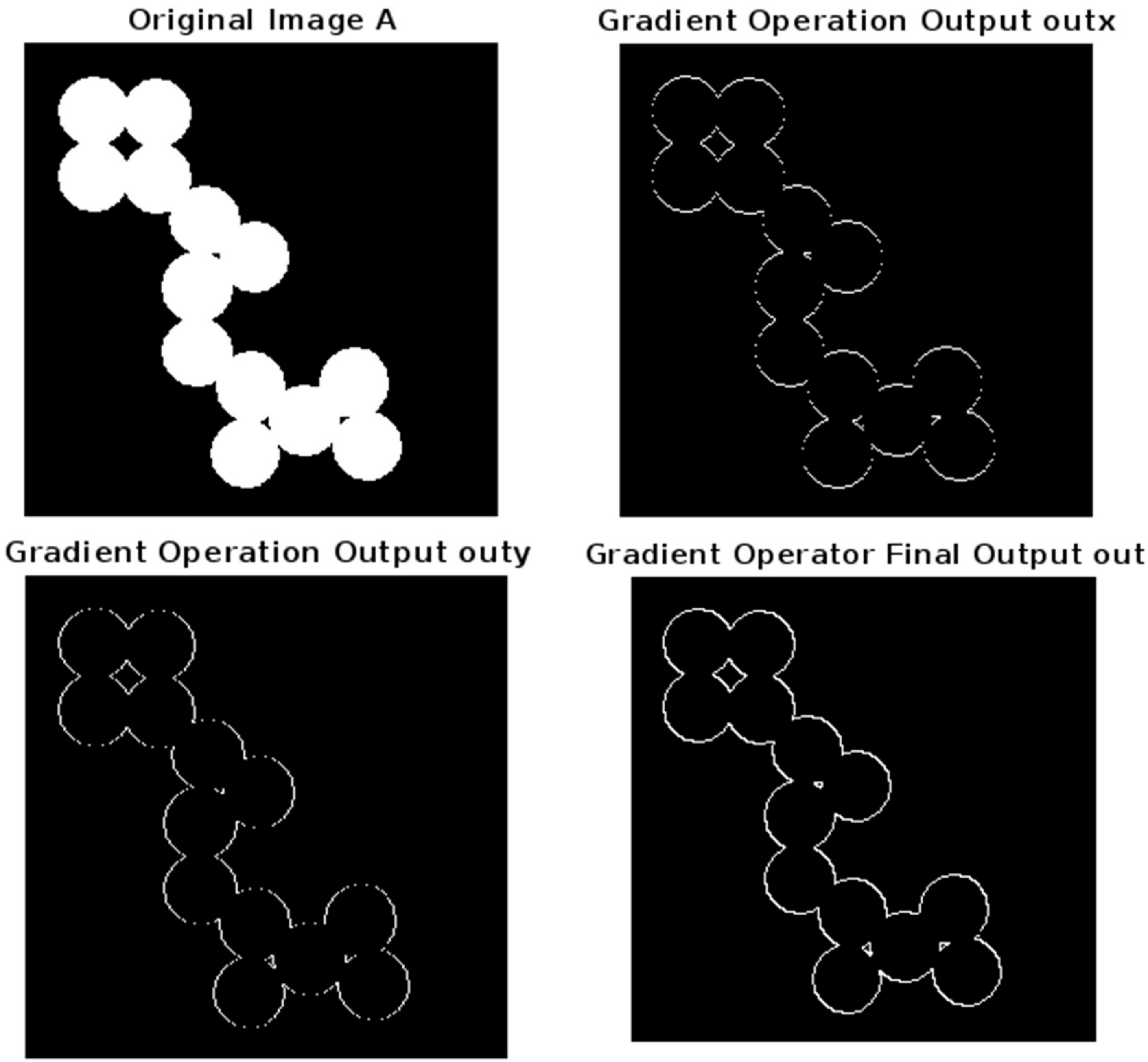
out2 =

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

out =

0	0	0	1	0	0	0
0	0	0	1	0	0	0
1	1	1	2	1	1	1
0	0	0	1	0	0	0
0	0	0	1	0	0	0
0	0	0	1	0	0	0
0	0	0	1	0	0	0

1st Derivative (Gradient)



1^{st} Derivative

- Mask used for Edge detection (First Order):
  - **Robert Operator:**

$$G_{45^\circ} = \begin{array}{|c|c|}\hline -1 & 0 \\ \hline 0 & 1 \\ \hline \end{array} \qquad G_{-45^\circ} = \begin{array}{|c|c|}\hline 0 & -1 \\ \hline 1 & 0 \\ \hline \end{array}$$

$$R = \left|I^0 G_{45^0}\right| + \left|I^0 G_{-45^0}\right|$$

Roberts Kernel are derivative with respect to the diagonal elements. Hence, they are called ***cross-gradient operators.*** They are based on the cross-diagonal differences.

1^{st} Derivative

- Mask used for Edge detection (First Order):
  - **Prewitt Operator:**

P_H =

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

For Horizontal Edges

P_V =

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

For Vertical Edges

$$P = \left|I^0 P_H\right| + \left|I^0 P_V\right|$$

1st Derivative

- Mask used for Edge detection (First Order):
  - **Sobel Operator:**

$S_H =$

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

For Horizontal Edges

$S_V =$

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

For Vertical Edges

$$S = |I^0 S_H| + |I^0 S_V|$$

1st Derivative: Comparison

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

Gradient

out =

0	0	0	1	0	0	0
0	0	0	1	0	0	0
1	1	1	2	1	1	1
0	0	0	1	0	0	0
0	0	0	1	0	0	0
0	0	0	1	0	0	0
0	0	0	1	0	0	0

Robert

out =

0	0	0	2	0	0	0
0	0	0	2	0	0	0
2	2	2	0	2	2	2
0	0	0	2	0	0	0
0	0	0	2	0	0	0
0	0	0	2	0	0	0
0	0	0	2	0	0	0

Prewitt

out =

0	0	0	3	3	0	0
0	0	0	3	3	0	0
3	3	3	2	2	3	3
3	3	3	2	2	3	3
0	0	0	3	3	0	0
0	0	0	3	3	0	0
0	0	0	3	3	0	0

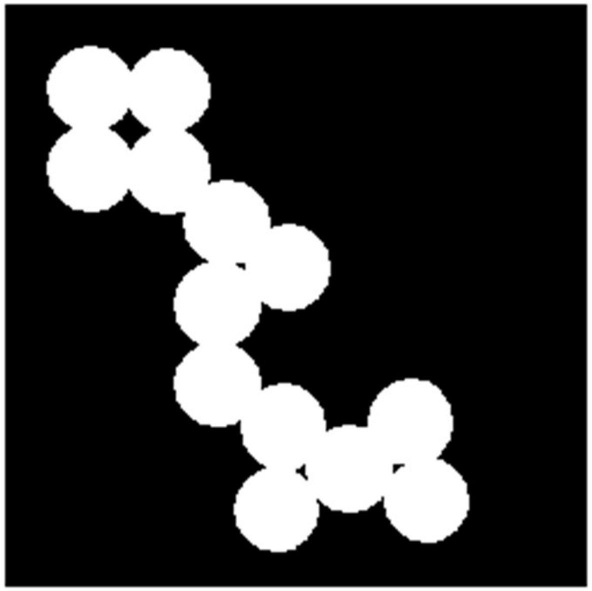
Sobel

out =

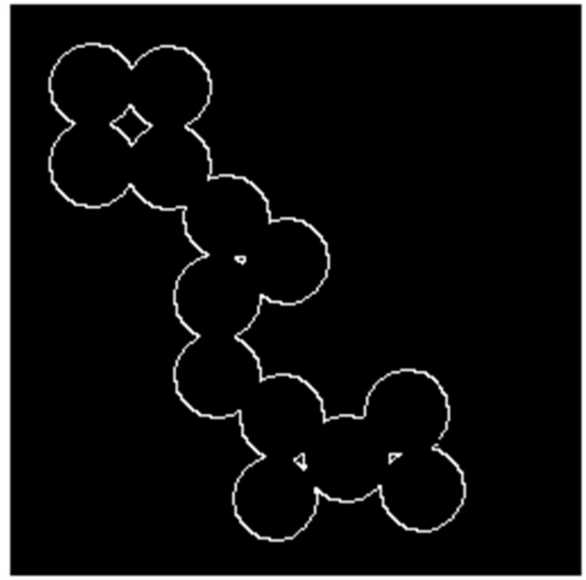
0	0	0	4	4	0	0
0	0	0	4	4	0	0
4	4	4	4	4	4	4
4	4	4	4	4	4	4
0	0	0	4	4	0	0
0	0	0	4	4	0	0
0	0	0	4	4	0	0

1st Derivative: Comparison

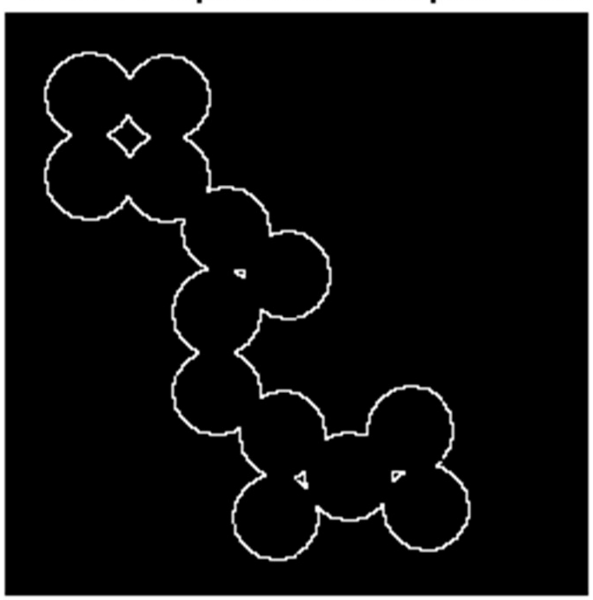
Original Image A



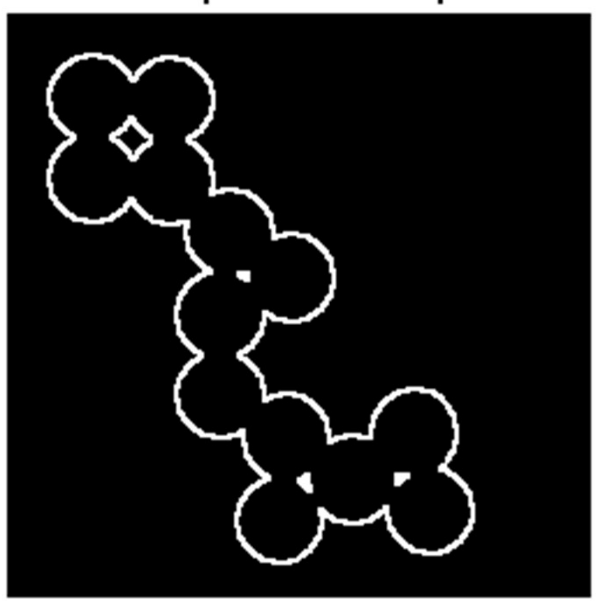
Gradient Operator Final Output out



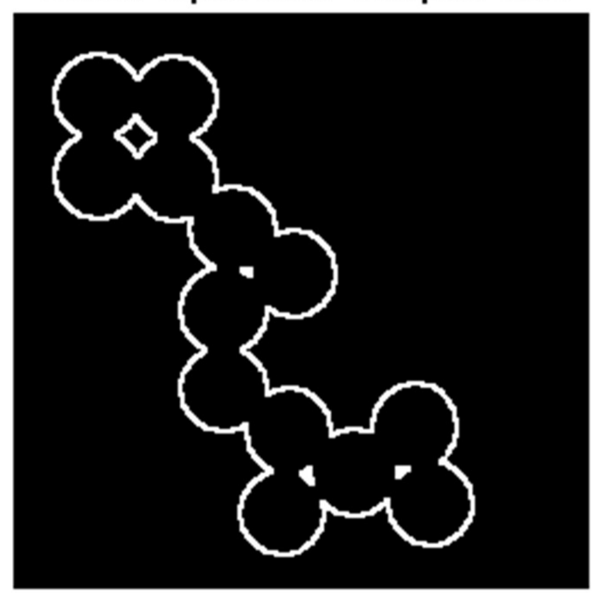
Robert Operator Output Out



Prewitt Operator Output Out

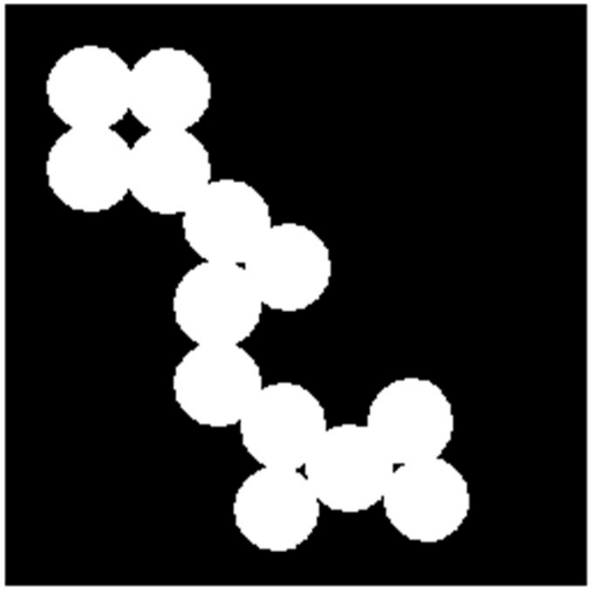


Sobel Operator Output Out

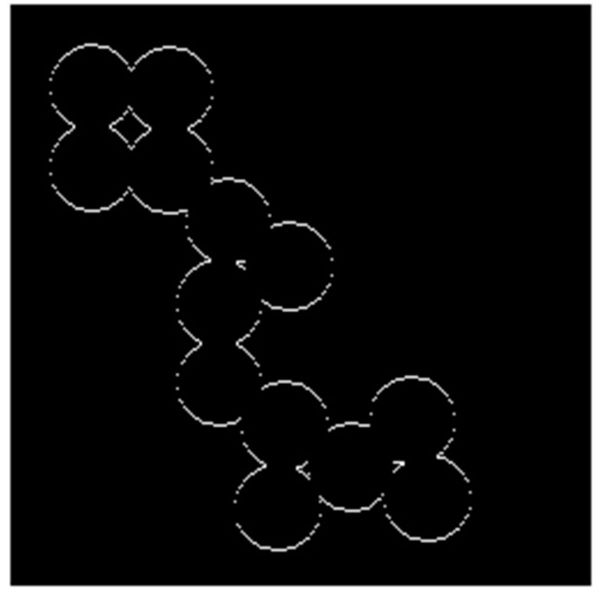


1st Derivative (Gradient)

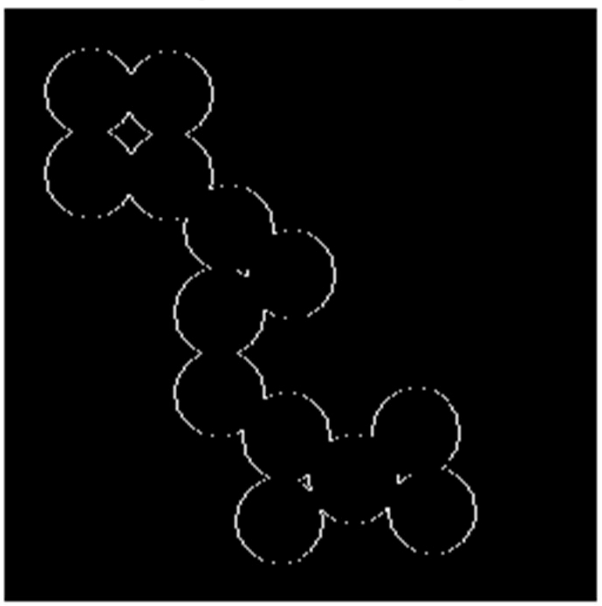
Original Image A



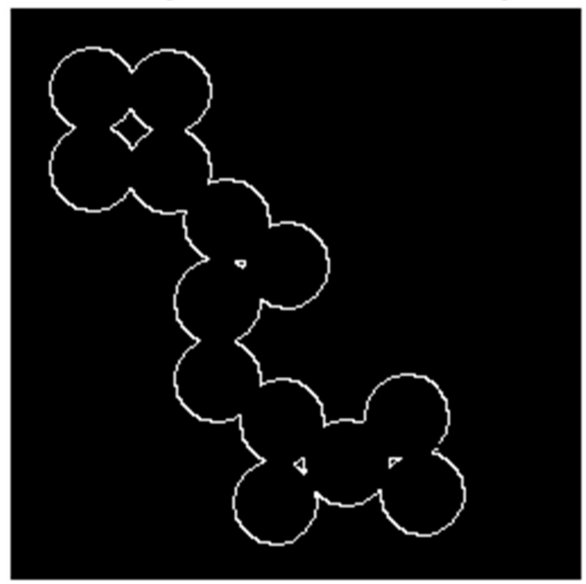
Gradient Operation Output outx



Gradient Operation Output outy

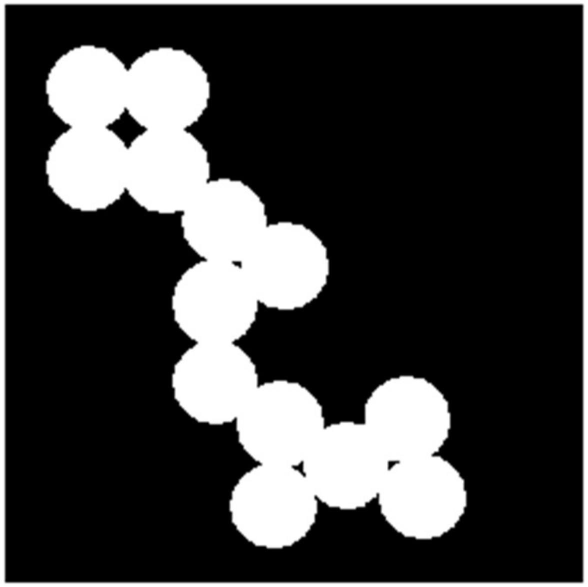


Gradient Operator Final Output out

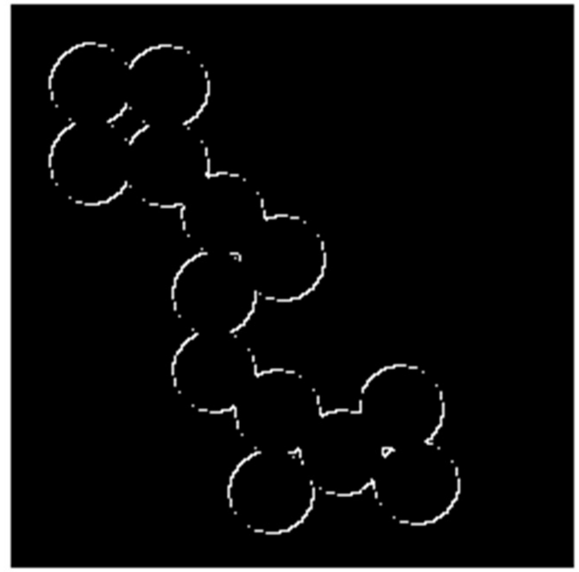


1st Derivative (Robert)

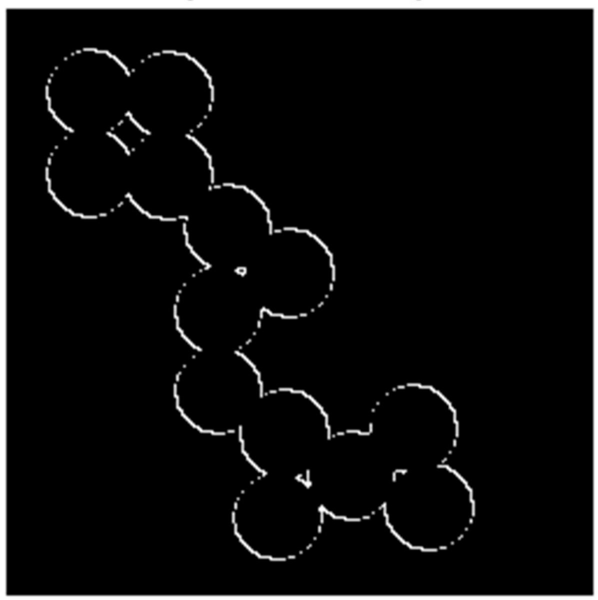
Original Image A



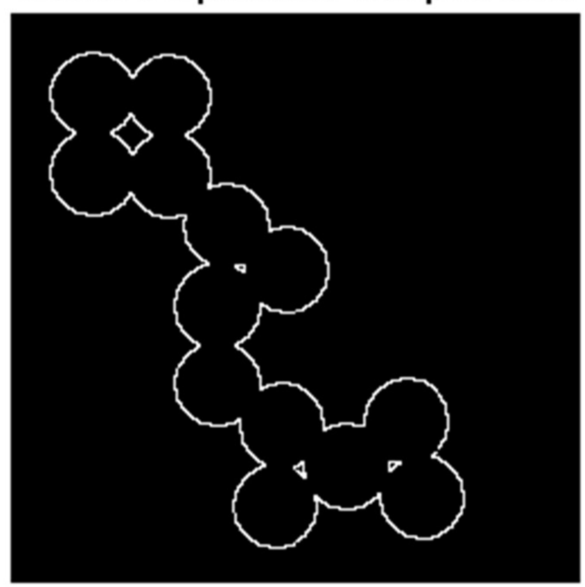
Robert Operator output Out1



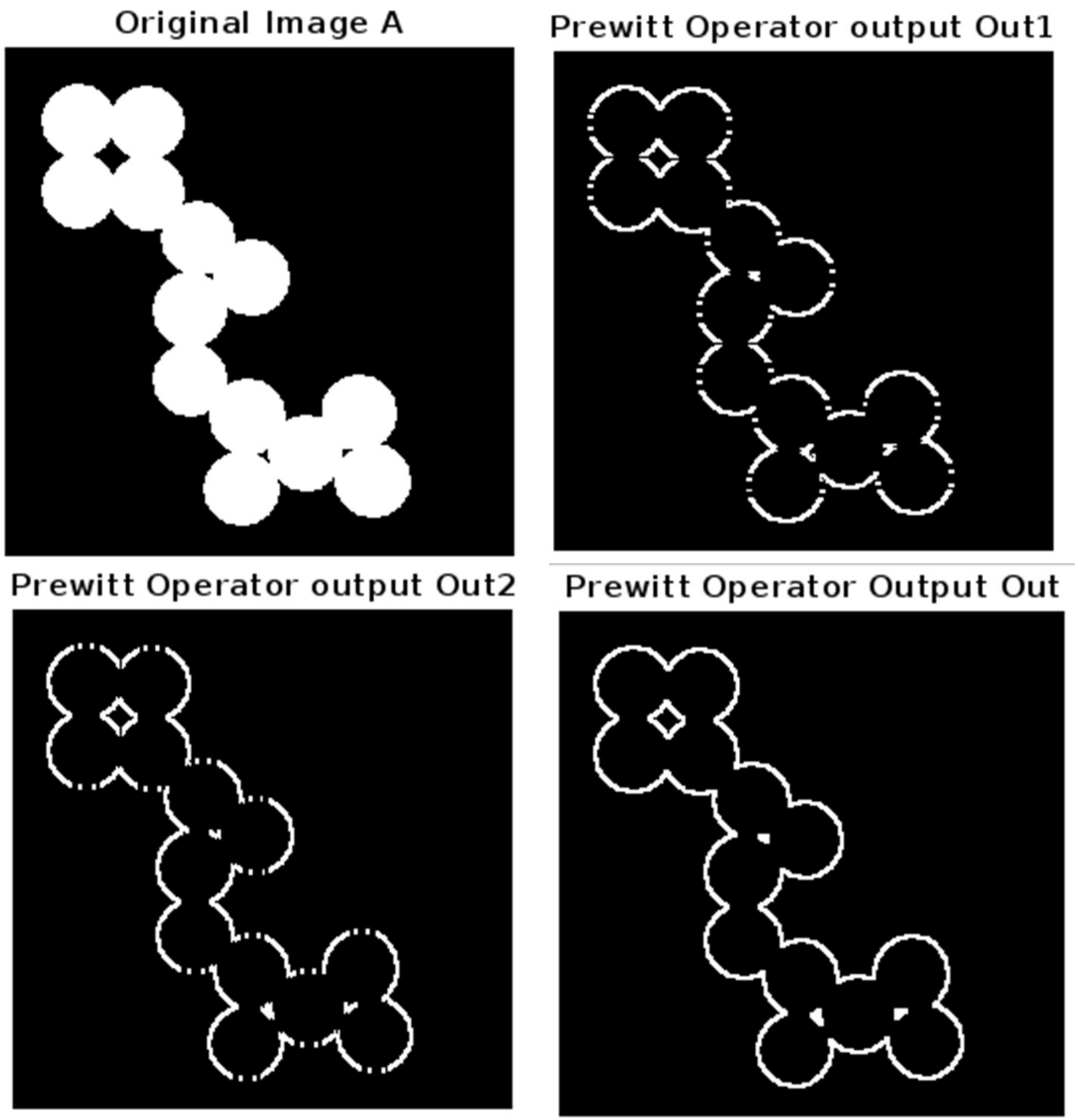
Robert Operator output Out2



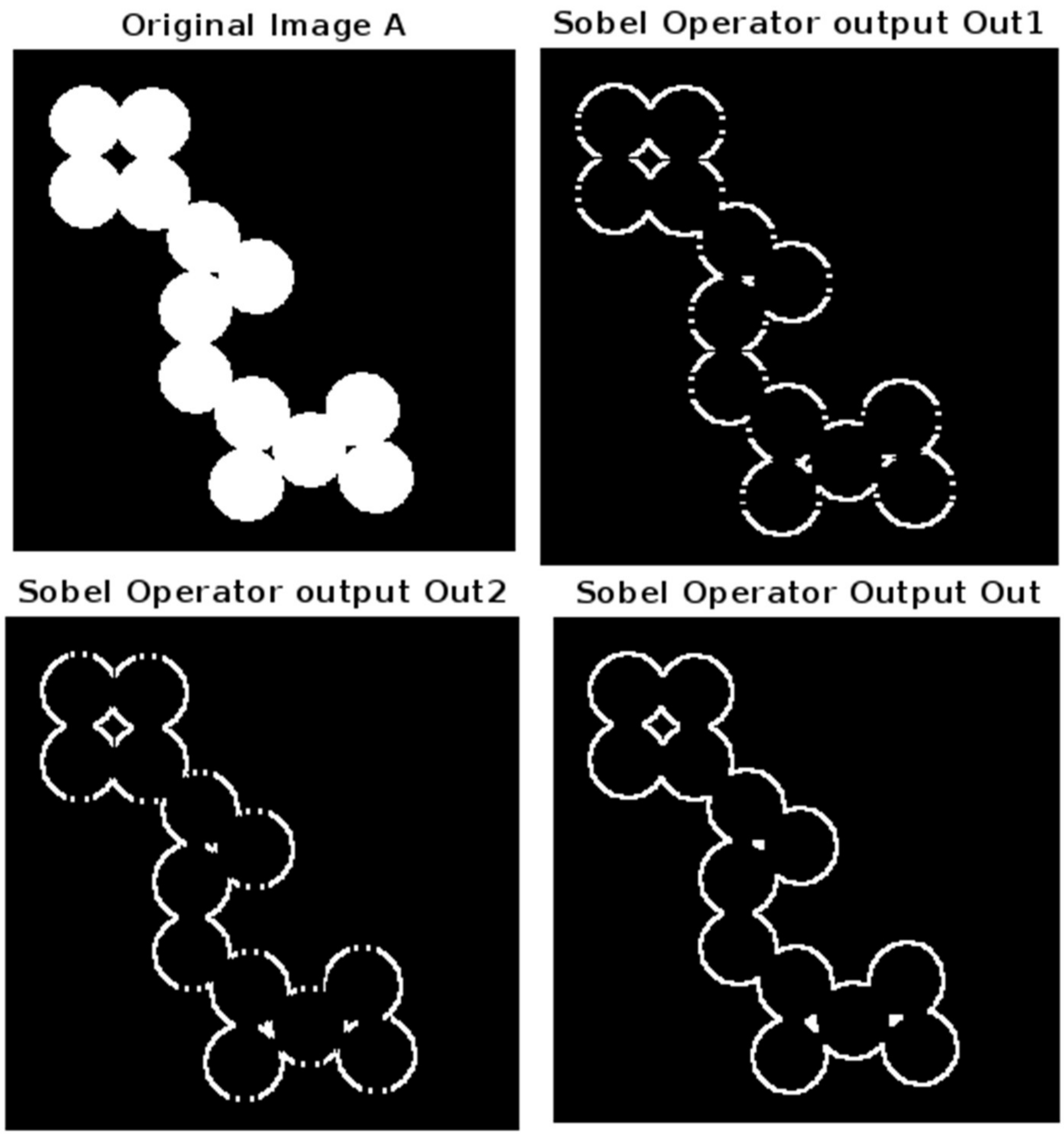
Robert Operator Output Out



1st Derivative (Prewitt)



1^{st} Derivative (Sobel)



1st Derivative

- The Prewitt Operators are simpler to implement than the Sobel mask, but the slight computational difference between them typically is not an issue.

- The fact that the Sobel mask have better noise-suppression (smoothing) characteristics makes them preferable.

- Noise suppression is an important issue when dealing with derivatives.

1st Derivative

Prewitt Operator: The additional mask can be used to detect the edges in the diagonal direction are as follows:

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

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

Sobel Operator: The additional mask can be used to detect the edges in the diagonal direction are as follows:

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

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

2^{nd} Derivative

GLA UNIVERSITY MATHURA
Accredited with A+ Grade by NAAC
12-B Status from UGC

2nd Derivative

- used in edge detection operation in image segmentation.

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

Where,

$\frac{\partial^2 f}{\partial x^2}$ = 2nd order derivative in x-direction

$\frac{\partial^2 f}{\partial y^2}$ = 2nd order derivative in y-direction.

2nd Derivative

❑ Masks used in 2nd Order Derivative:

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

Consider only the horizontal direction and vertical direction for the computation of the second derivative.

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

Consider the horizontal direction, vertical direction as well as the diagonal direction for the computation of the second derivative.

2nd Derivative (Laplacian)

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

out1 =

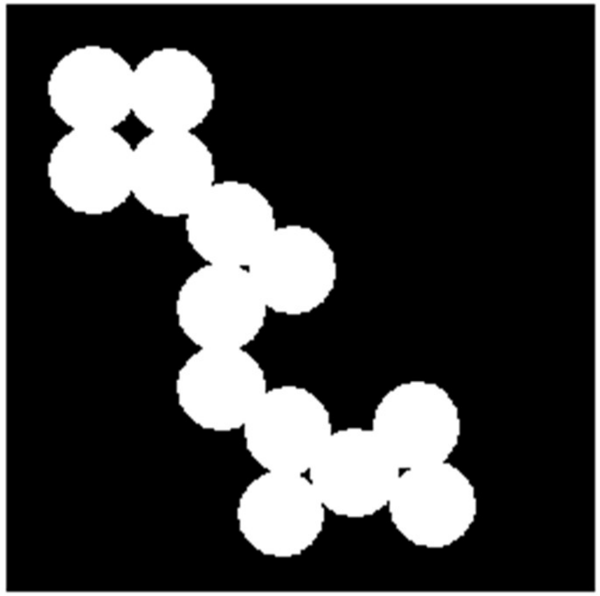
0	0	0	-1	1	0	0
0	0	0	-1	1	0	0
-1	-1	-1	-2	2	1	1
1	1	1	2	-2	-1	-1
0	0	0	1	-1	0	0
0	0	0	1	-1	0	0
0	0	0	1	-1	0	0

out2 =

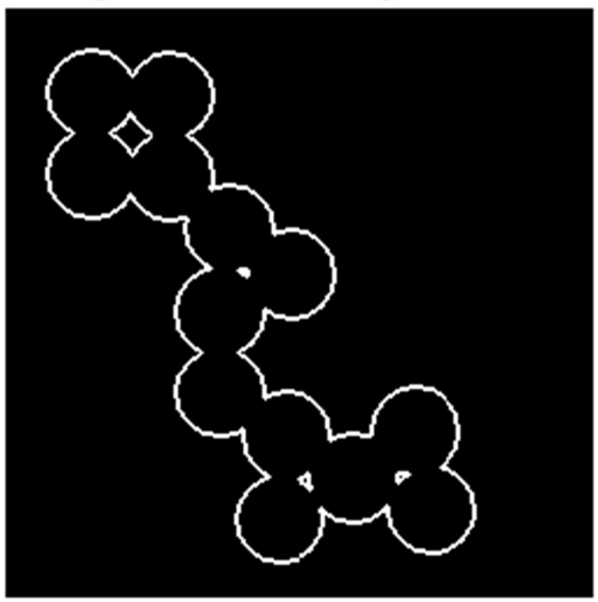
0	0	0	-3	3	0	0
0	0	0	-3	3	0	0
-3	-3	-3	-4	4	3	3
3	3	3	4	-4	-3	-3
0	0	0	3	-3	0	0
0	0	0	3	-3	0	0
0	0	0	3	-3	0	0

2nd Derivative (Laplacian)

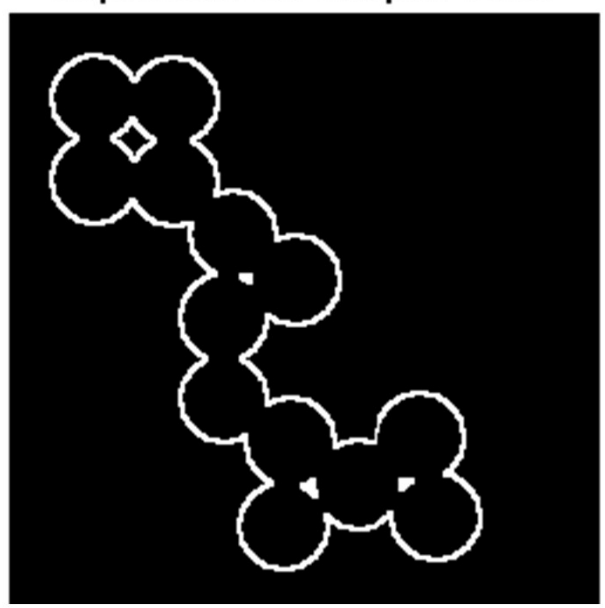
Original Image A



Laplacian L4 output out4



Laplacian L8 output out8



2nd Derivative

- Normally, Laplacian operator (2nd order derivative) is not used for the edge detection because of the following reasons:
  - Very very sensitive to the noise
  - It leads to double edge at every transition.
- However, it gives secondary information such as
  - Whether the particular point/pixels lies on the darker side or the brighter side
  - It accurately determine the location of edge (the location of zero-crossing point).
- To reduce the noise sensitivity of the Laplacian operator, input image is ***first processed with Gaussian operator*** then ***the smooth image is processed with Laplacian operator*** and these two operation is called as **"Laplacian of Gaussian (LOG) operator".**

LOG Operation

- Gaussian operator:

$$h(x,y) = \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

If $x^2 + y^2 = r^2$

$$\nabla^2 h = \left(\frac{r^2-\sigma^2}{\sigma^4}\right) exp\left(-\frac{r^2}{2\sigma^2}\right)$$

- LOG Mask:

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

LOG Operation

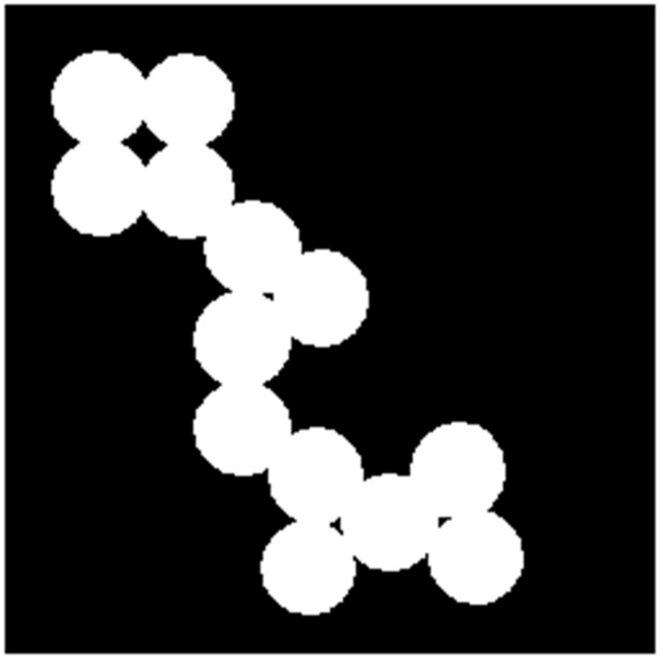
1	1	1	1	2	2	2
1	1	1	1	2	2	2
1	1	1	1	2	2	2
2	2	2	2	1	1	1
2	2	2	2	1	1	1
2	2	2	2	1	1	1
2	2	2	2	1	1	1

out =

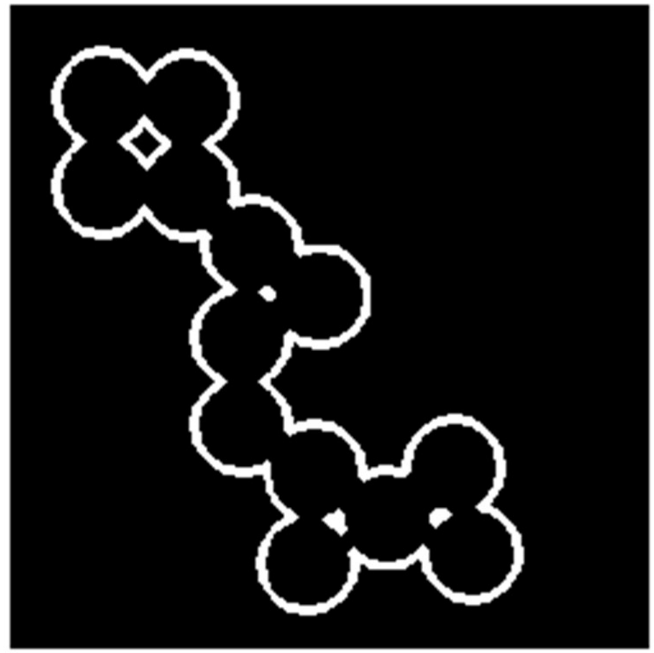
0	0	-1	-5	5	1	0
-1	-1	-2	-6	6	2	1
-5	-5	-6	-8	8	6	5
5	5	6	8	-8	-6	-5
1	1	2	6	-6	-2	-1
0	0	1	5	-5	-1	0
0	0	1	5	-5	-1	0

LOG Operation

Original Image A

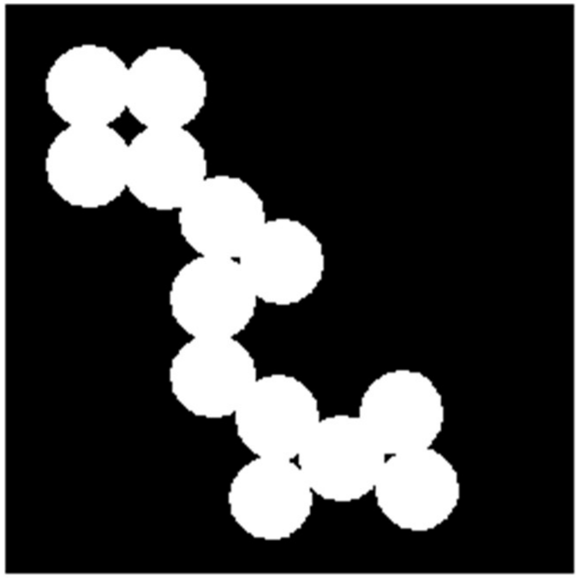


LOG Transformed Image out

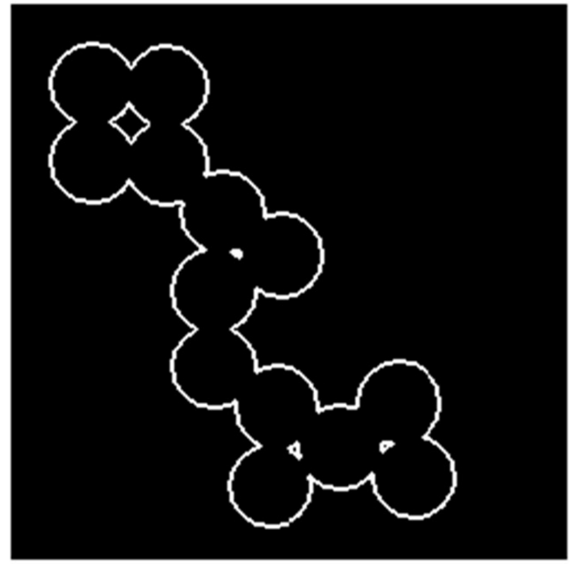


Comparison

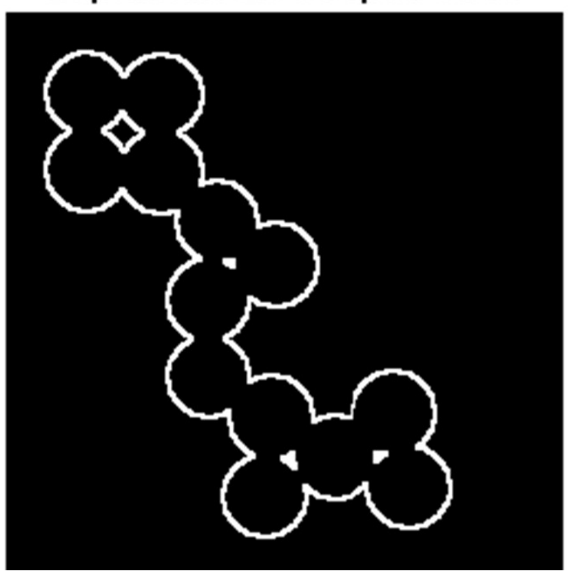
Original Image A



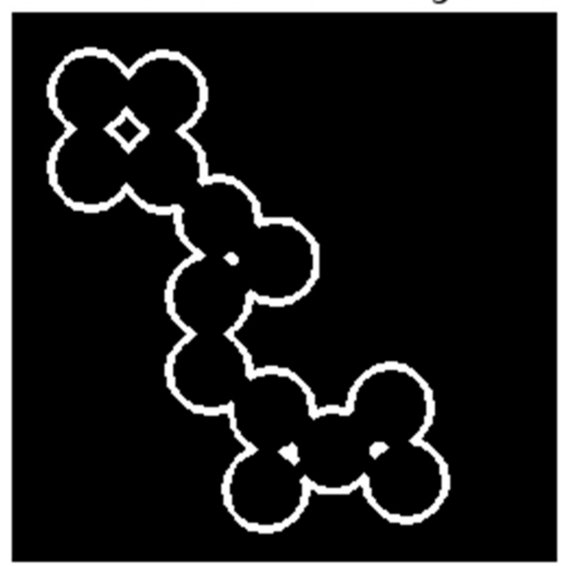
Laplacian L4 output out4



Laplacian L8 output out8



LOG Transformed Image out



Edge Linking

Edge Linking

- Ideally discontinuity detection technique should identify pixels lying on the boundary between regions.

- In practice, there may be breaks in boundary and spurious intensity discontinuities

  - Due to non-uniform illumination

  - Presence of noise

- Edge linking procedure assemble edge points into meaningful boundaries. There are two approaches for Edge linking

  - Local Processing

  - Global Processing

Thank You