Image Segmentation

Kiran TALELE

☎ 99870 30 881

Kiran TALELE

@ Bharatiya Vidya Bhavans' Sardar Patel Institute of Technology Andheri(w) Mumbai

- **Associate Professor,** Electronics Engineering Department (1997)
- **Dean,** Students, Alumni & External Relations (2022)

@ Sardar Patel Technology Business Incubator(SP-TBI), Funded by Department of Science & Technology(DST), Govt. of India

- **Head**, Academic Relations (2015)

@ IEEE Bombay Section

- **Treasurer** (2020)
- Executive Committee Member (2015)

Segmentation :

(i) Segmentation subdivides an image into its constituent parts or objects.

(ii) Segmentation is based on two basic properties of gray level values: **Discontinuity** and **Similarity**

[I] Segmentation using Discontinuity Property:

- In this category, image is partitioned into different parts or objects based on abrupt changes in gray level.
- This involves detection of isolated **POINTs** and **LINEs** and **EDGEs** in the image.

(1) **POINT** Detection

Consider a digital subimage F and point detector mask w

F =

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Point Detector

w =

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

A =

	R	

where,

$$R = z_1 w_1 + z_2 w_2 + z_3 w_3 + \ldots\ldots + z_9 w_9$$

For any point P at (x,y) position,

If |R| > Threshold
Then
POINT is said to be detected at (x,y) position

(2) **LINE** Detection

Consider a digital subimage F and point detector mask w

F =

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

→ **Line Detector** →

A =

	R	

Detection of a line can be done using the mask given below

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

Horizontal Line

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

Vertical Line

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

45^0 Direction Line

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

-45^0 Direction Line

F =

10	10	10
100	100	100
10	10	10

→ **Horizontal Line Detector W_1** →

A =

0	0	0
540	540	540
10	10	10

Horizontal Line Detected

F =

10	10	10
100	100	100
10	10	10

→ **Vertical Line Detector W_2** →

A =

0	0	0
0	0	0
0	0	0

NO Line Detected

F =

10	10	10
100	100	100
10	10	10

→ **45^0 Line Detector W_3** →

A =

0	0	0
0	0	0
0	0	0

NO Line Detected

F =

10	10	10
100	100	100
10	10	10

→ **-45^0 Line Detector W_4** →

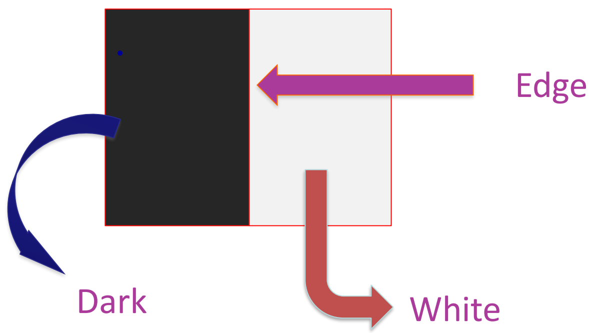
A =

0	0	0
0	0	0
0	0	0

NO Line Detected

(3) **EDGE** Detection

- Edges are abrupt discontinuities in the Gray Levels
- Edge is defined as rate of change of continuity



[A] Step Edge Detection

(i) Consider a Digital Image

$$F = \begin{bmatrix} 0 & 0 & 200 & 200 & 200 & 200 & 0 & 0 \\ 0 & 0 & 200 & 200 & 200 & 200 & 0 & 0 \\ 0 & 0 & 200 & 200 & 200 & 200 & 0 & 0 \end{bmatrix}$$

(ii) Pixel Values of Row -1 with repeated border values :

[0 0 0 200 200 200 200 0 0 0]

(iii) First Order Derivative i.e. Gradient :

[0 0 200 0 0 0 -200 0 0 0]

(iv) The Magnitude of First Order Derivative :

[0 0 **200** 0 0 0 **200** 0 0 0]

(v) The Edge detected image :

F =

0	0	200	0	0	0	200	0	0
0	0	200	0	0	0	200	0	0
0	0	200	0	0	0	200	0	0

Sharp edge Sharp edge

- **Magnitude of First order derivative produces desirable effect. i.e. Sharp Edges.**
  Therefore, first derivative mask such as Robert, Prewitt, Sobel and Fri-chen are suitable to detect Step Edges.

(iii) First Order Derivative i.e. Gradient :

[0 0 200 0 0 0 -200 0 0 0]

(iv) Second Order Derivative i.e. Laplacian :

[0 200 -200 0 0 -200 200 0 0 0]

(v) Magnitude of Second Order Derivative

[0 **200** **200** 0 0 **200** **200** 0 0 0]

(v) The Magnitude of Second Order Derivative :

[0 **200** **200** 0 0 **200** **200** 0 0]

(vi) The Edge detected image :

F =

0	0	200	200	0	0	200	200	0	0
0	0	200	200	0	0	200	200	0	0
0	0	200	200	0	0	200	200	0	0

Double edges **Double edges**

- **Magnitude of second order derivative produces double edges which is NOT desirable.**

  Therefore, Second order derivative mask such as Laplacian mask is NOT suitable to detect Step Edges.

[B] Ramp Edge Detection

(i) Consider a Digital Image

$$F = \begin{bmatrix} 0 & 0 & 80 & 160 & 240 & 240 & 240 & 240 \\ 0 & 0 & 80 & 160 & 240 & 240 & 240 & 240 \\ 0 & 0 & 80 & 160 & 240 & 240 & 240 & 240 \end{bmatrix}$$

(ii) Pixel Values of Row -1 with repeated border Values :

[0 0 0 80 160 240 240 240 240 240]

(iii) First Order Derivative i.e. Gradient :

[0 0 80 80 80 0 0 0 0]

(iv) The Magnitude of First Order Derivative :

[0 0 **80** **80** **80** 0 0 0 0]

(v) The Edge detected image :

F =

0	0	80	80	80	0	0	0
0	0	80	80	80	0	0	0
0	0	80	80	80	0	0	0

Thick edge

- **Magnitude of First order derivative produces Thick Edge which is Not desirable**
- **Therefore, first derivative mask such as Robert, Prewitt, Sobel and Fri-chen are NOT suitable to detect Ramp Edges**

(iii) First Order Derivative i.e. Gradient :

[0 0 80 80 80 0 0 0 0]

(iv) Second Order Derivative i.e. Laplacian :

[**0** 80 0 0 -80 0 0 0 **0**]

(v) Magnitude of Second Order Derivative

[0 **80** **0** **0** **80** 0 0 0 0]

(v) The Magnitude of Second Order Derivative :

[0 **80** 0 0 **80** 0 0 0 0]

(vi) The Edge detected image :

F =

0	0	80	0	0	80	0	0	0	0
0	0	80	0	0	80	0	0	0	0
0	0	80	0	0	80	0	0	0	0

Sharp edge Sharp edge

- **Second order derivative produces desirable results..i.e. Sharp Edges.**

  Therefore, Second order derivative mask such as Lapacian mask is the most suitable to detect Ramp Edges.

Edge Linking and Boundary Detection

- Edge Detection Algorithm is followed by Linking procedures to assemble edge pixels into meaningful edges.
- Edge Linking Methods :

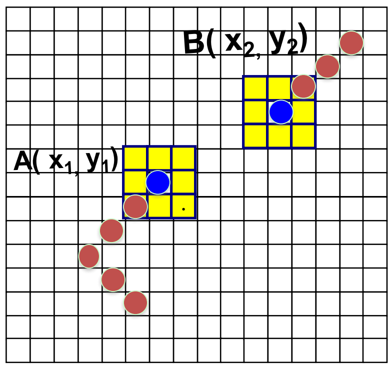
  (1) Edge Linking using Local Processing

  (2) Edge Linking using Global Processing Hough Transform Method

  (3) Edge Linking using Global Processing Graph Theoretic Method

• Edge Linking using Local Processing

(i) Consider two edge points A(x_1, y_1) and B(x_2, y_2) as shown in figure below



(ii) Select a small neighbourhood say 3x3.

(iii) Analyse the characteristics of edge pixels in a small neighbourhood say 3x3.

(iv) Link all the points that are similar according to a set of predefined criteria

(v) Two basic properties of pixels are :

a) Strength of Gradient
b) Direction of Gradient

Edge Linking using Local Processing . . . .

(v) An edge pixels with co-ordinates (x', y') is similar in magnitude to the pixel at (x,y) if the following condition is satisfied

$$\text{If } \left|\nabla f(x, y) - \nabla f(x', y')\right| \leq \text{Magnitude Threshold}$$

Where

$$\nabla f(x, y) = |Gx| + |Gy|$$

Edge Linking using Local Processing . . . .

(vi) An edge pixels with co-ordinates (x', y') is similar in direction to the pixel at (x,y) if the following condition is satisfied

$$\text{If } \left|\alpha(x, y) - \alpha(x', y')\right| \leq \textbf{Angle} \text{ Threshold}$$

Where

$$\alpha(x, y) = \tan^{-1}\left(\frac{Gy}{Gx}\right)$$

Edge Linking using Local Processing . . . .

That means,

If $\left| \nabla f(x, y) - \nabla f(x', y') \right| \leq$ Magnitude Threshold

AND

$\left| \alpha(x, y) - \alpha(x', y') \right| \leq$ **Angle** Threshold

Then

Two pixels at (x,y) and (x',y') can be linked.

- **Edge Linking using Global Processing Hough Transform Method**

(1) Hough Transform :

Consider two points A(x_1, y_1) and B(x_2, y_2) in xy plane

The equation of the line passing through AB is then given by,

$$y = m x + c$$

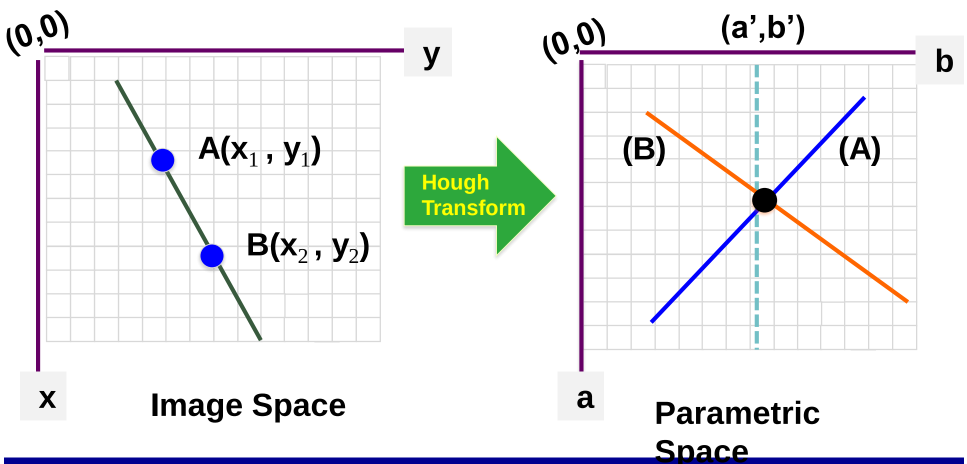
Let **y = a x + b** where **a** is slope value and **b** is y-intercept value

$$b = -a x + y$$

At A(x_1 , y_1), b = - a x_1 + y_1

At B(x_2 , y_2), b = - a x_2 + y_2

b = -a x + y



(i) **Hough Transformation is POINT to LINE and LINE to POINT Transformation from xy plane (Image Space) to ab plane (Parametric Space)**

(ii) **At each intersecting point in ab-plane, we get slope value and y-constant value of line that exist in xy-plane.**

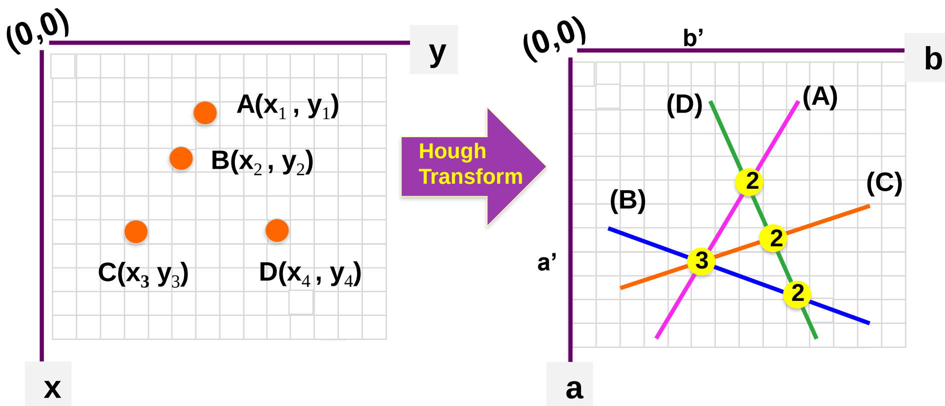
(iii) **The equation of line is given by,**

Y = a’ x + b’ ➔ THIS IS LINE AB

(2) Edge Linking using Hough Transform

(I) Map all the input points from xy plane to ab plane using Hough Transform

e.g. Consider 4 input points A, B, C and D as shown in figure.



Edge Linking using Hough Transform . . . .

(II) Count the number of intersecting lines at each point in ab plane.

(III) Select the point with maximum value of count.

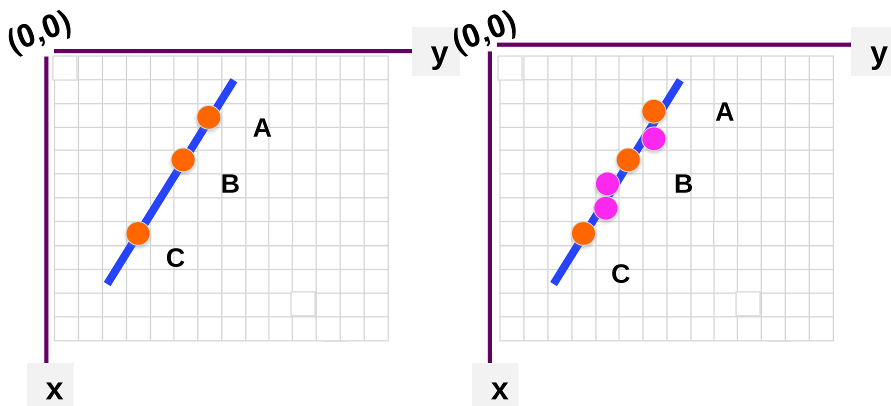
E.g. Max value of count is 3 at point (a',b')

(III) Define line with slope value = **a'** and Y-constant value = **b'** . The equation of line is **y = a' x + b'**

(IV) Determine co-linear points..

E.g. Point A, B and C are co-linear points.

(V) Link all co-linear points.



(3) **Limitation of Hough Transform :**

(i) Hough Transform method is not suitable for vertical edges

(ii) For vertical line, slope value is ∞

Paul V. C. Hough

The Hough transform was first proposed by **Paul Hough** in 1962 as a method for detecting lines in images.

It was made popular by Duda and Hart: Duda, R. O. and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures", 1972.

Ref.
The Hough transform was patented as U.S. Patent 3,069,654 in 1962 and assigned to the U.S. Atomic Energy Commission with the name "Method and Means for Recognizing Complex Patterns". This patent uses a slope-intercept parametrization for straight lines

Overview of Hough Transform :

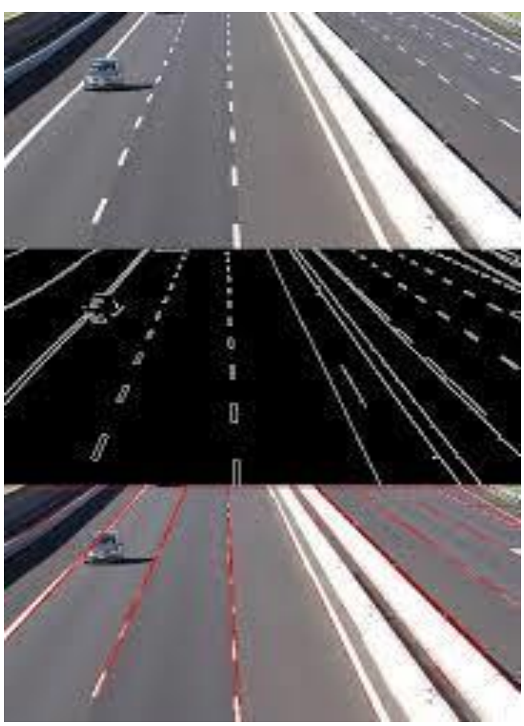
- The Hough transform is a mathematical technique used in computer vision and image analysis to detect simple geometric shapes like lines, circles, and ellipses.
- The basic idea behind the Hough transform is to represent lines or curves in an image as points in a parameter space.
- The Hough transform is a powerful tool for line and curve detection in images, particularly when traditional edge detection techniques fail.
- Its ability to handle broken and incomplete lines makes it a valuable addition to any image processing toolkit

Q1. What is Hough Transform?

- The Hough transform in image processing is a technique used to detect simple geometric shapes in images.
- It works by transforming the image space into a parameter space, where the geometric shapes can be detected through the identification of patterns in the parameter space.

Q2. Why is Hough Transform Needed?

•The Hough transform is needed because traditional image processing techniques like edge detection and thresholding are not always effective at detecting simple geometric shapes in images. These techniques can be particularly ineffective when the shapes are distorted, incomplete, or partially obscured.
•The Hough transform can detect these shapes by transforming the image space into a parameter space where the shapes can be more easily identified.



Input image

Edge detected image

Edge detected image processed using Hough Transform

▪ Edge Linking using Graph Theoretic Method

(i) In this Method, edge is represented in the form of Graph

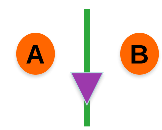
(ii) Edge is also called as Path and Low cost path corresponds to the most significant edge in the image

(iii) Consider Two pixels A and B

(iv) The Cost between A and B is given by,

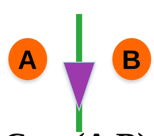
Cost (A,B) = H - | F(A)-F(B) |

Where H is the max gray value in the image
F(A) and F(B) are pixel values at point A and at point B respectively



Cost (A,B)

e.g.



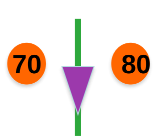
Cost (A,B)

Cost (A,B) = H - | F(A)-F(B) |

Assume H = 255

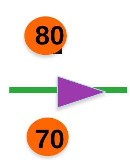
Cost (A,B) = 255 - | F(A)-F(B) |

Case-1 :



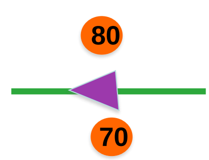
Cost = 255-| 70-80|
Cost = 245

Case-2 :



Cost = 255-| 70-80|
Cost = 245

Case-3 :



Cost = 255-| 80-70|
Cost = 245

Example :

Assume that the edge starts in the first Row and ends in the last row for the following Gray Image.

F =

5	6	1
6	7	0
7	1	3

Sketch all possible paths and determine the edge corresponding to minimum cost path

Solution :

The Cost between tow pixels A and B is given by,

Cost (A,B) = H - | F(A)-F(B) |

Where H = 7 is the max gray value in the image

F(A) and F(B) are pixel values at point A and at point B

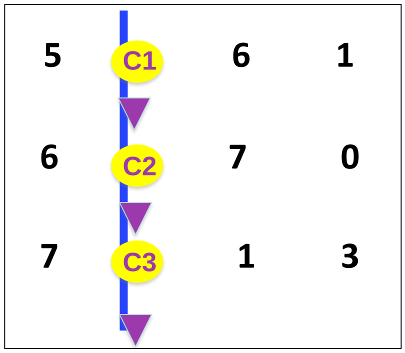
(1) For Edge-1 :

5	6	1
6	7	0
7	1	3

Cost (A,B) = H - | F(A)-F(B) |

Cost (A,B) = 7 - | F(A)-F(B) |

(1) For Edge-1 :



Cost (E1) = C1 + C2 + C3
= 6 + 6 + 1
= 13

(i) To find C1

C1 = 7 - | F(A)-F(B) |

C1 = 7 - | 5-6 |

C1 = 6

(ii) To find C2

C2 = 7 - | F(A)-F(B) |

C2 = 6

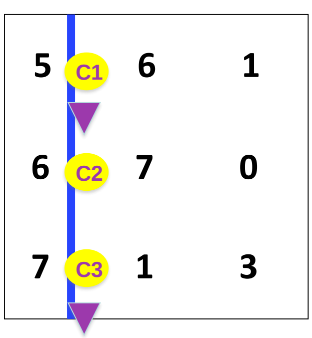
(ii) To find C3

C3 = 7 - | F(A)-F(B) |

C3 = 7 - | 7-1 |

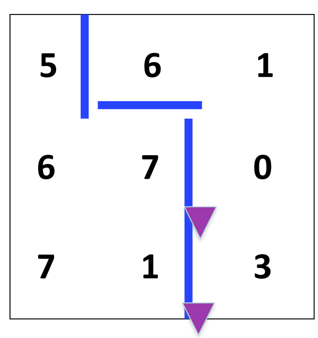
C3 = 1

(1) For Edge-1 :



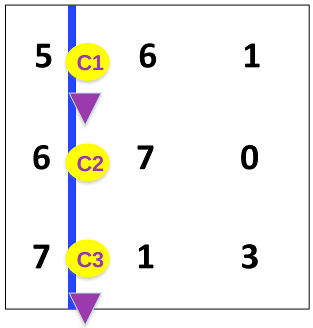
Cost (E1) = C1+ C2 +C3
Cost (E1) = 6 + 6 + 1
Cost (E1) == 13

(2) For Edge-2 :

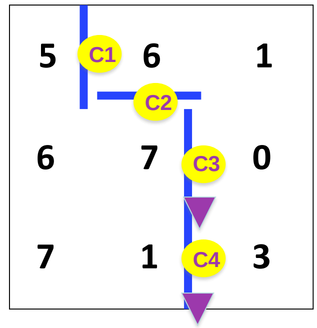


Cost (E1) = C1 + C2 + C3 + C4
Cost(E2) = 6 + 6 + 0 + 5
Cost (E2) == 13

(1) For Edge-1 :



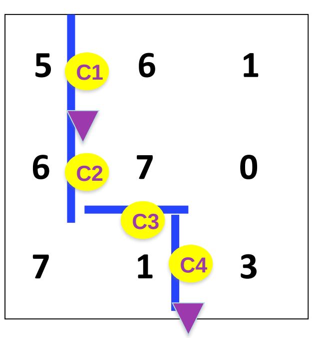
(2) For Edge-2 :



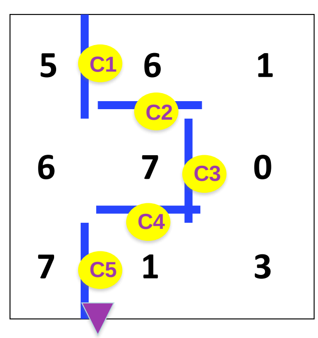
Cost (E1) = C1+ C2 +C3
Cost (E1) = 6 + 6 + 1
Cost (E1) == 13

Cost (E1) = C1 + C2 + C3 + C4
Cost(E2) = 6 + 6 + 0 + 5
Cost (E2) == 13

(3) For Edge-3 :



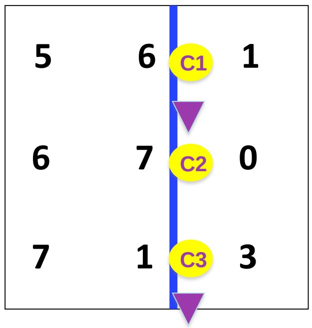
(4) For Edge-4 :



Cost (E3) = C1+C2+C3+C4
Cost (E3) = 6 + 6 + 1 + 5
Cost (E3) == 18

Cost (E4) = C1+C2+C3+C4+C5
Cost(E4) = 6 + 6 + 0 + 1 + 1
Cost (E4) == 14

(5) For Edge-5 :

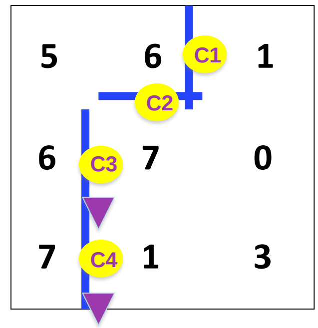


Cost (E5) = C1+ C2 +C3

Cost (E5) = 2 + 0 + 5

Cost (E5) == 13

(6) For Edge-6 :

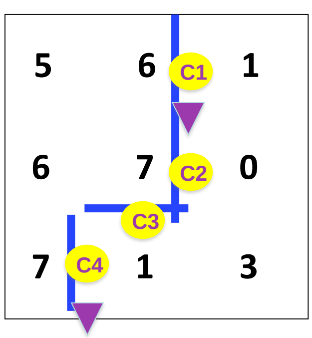


Cost (E6) = C1 + C2 + C3 + C4

Cost(E6) = 2 + 6 + 6 + 1

Cost (E6) == 15

(7) For Edge-7 :

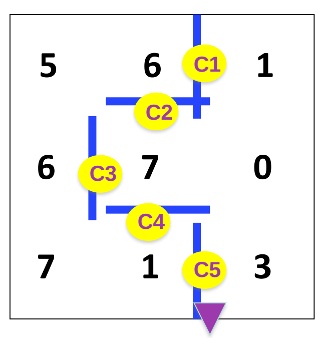


Cost (E7) = C1+C2+C3+C4

Cost (E7) = 2 + 0 + 1 + 1

Cost (E7) == 4

(8) For Edge-8 :



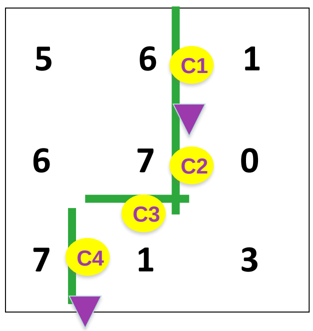
Cost (E8) = C1+C2+C3+C4+C5

Cost(E8) = 2 + 6 + 6 + 1 + 5

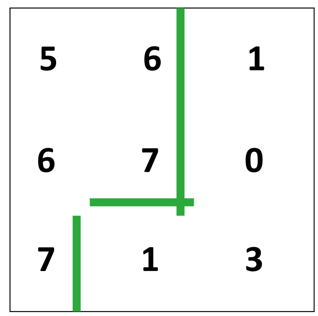
Cost (E8) == 20

Edge-7 has the minimum value of Cost.
Therefore, Edge-7 is the most significant edge in the image

For Edge-7, Let F(A) = 7 and F(B) = 0 and F(other) = 0



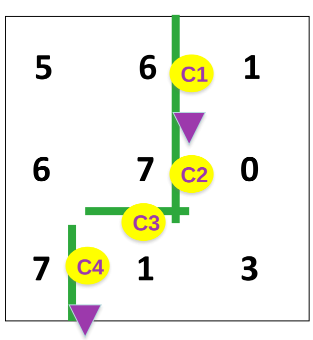
Input Image



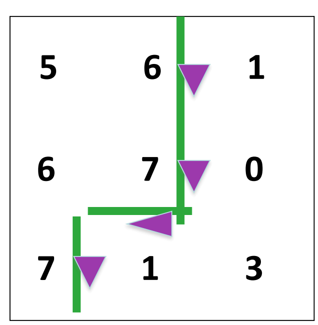
Output Image

Edge-7 has the minimum value of Cost.
Therefore, Edge-7 is the most significant edge in the image

For Edge-7, Let F(A) = 7 and F(B) = 0 and F(other) = 0



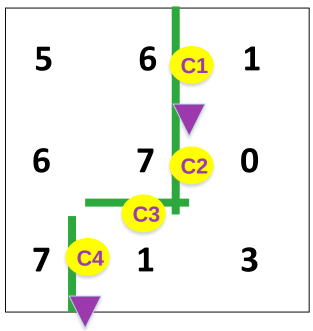
Input Image



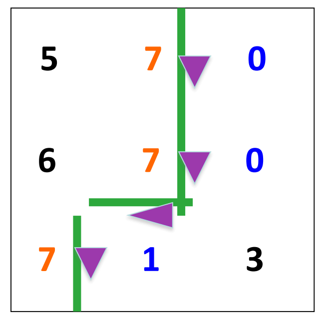
Output Image

Edge-7 has the minimum value of Cost.
Therefore, Edge-7 is the most significant edge in the image

For Edge-7, Let F(A) = 7 and F(B) = 0 and F(other) = 0



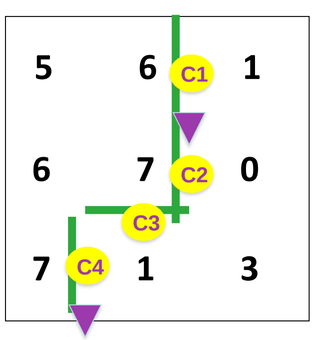
Input Image



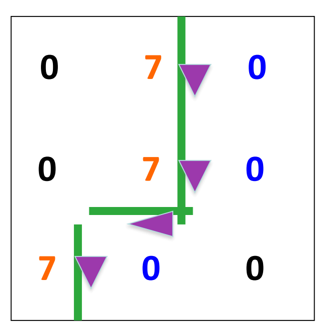
Output Image

Edge-7 has the minimum value of Cost.
Therefore, Edge-7 is the most significant edge in the image

For Edge-7, Let F(A) = 7 and F(B) = 0 and F(other) = 0



Input Image



Output Image

Edge-7 has the minimum value of Cost.
Therefore, Edge-7 is the most significant edge in the image

For Edge-7, Let F(A) = 7 and F(B) = 0 and F(other) = 0

5	6	1
6	7	0
7	1	3

Input Image

0	7	0
0	7	0
7	0	0

Output Image

[II] Segmentation using Similarity property:

- In this category, Image is partitioned into regions based on Similarity in gray level.
- This involves detection of region.
- Let R represents the entire image and R1, R2,--------R_N are segmented regions such that,

(i) $\bigcup_{i=1} Ri = R$

(ii) Ri is connected Region

(iii) Ri $\bigcap$ Rj = Null for all i and j

(iv) Predicate (Ri) = TRUE for all i

(A) Region Growing by Pixel Aggregation :-

(i) Region growing is a procedure that groups pixels into larger regions.

(ii) Pixel aggregation procedure starts with a seed point AND Neighbouring pixels that have similar values are appended

Example-1 :

F =

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

Predicate :
If F(x,y) = { 0, 1, 2, 3 }
Then F(x,y) = Red
Else F(x,y) = Yellow

F =

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

F =

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

Ex-2: Segment the following given image using Region Growing by Pixel Aggregation. Assume Threshold=2.

Predicate :
Let Seed Point be S(x',y')
If $| F(x,y) - S(x',y') | \leq T$
Then F(x,y) = Intensity of Segmented Region

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

(I) Let Seed Point $S_1(4,1) = 5$ and T = 2

Predicate :
If $| F(x,y) - S_1(4,1) | \leq T$
i.e. $| F(x,y) - 5 | \leq 2$

i.e. **F(x,y) = { 3, 4, 5, 6, 7 }**

Then F(x,y) = A

Segmented Region : R_1 = { A }

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

(II) Let Seed Point $S_2(0,0) = 3$ and $T = 2$

Predicate :

If $| F(x,y) – S_2(0,0) | \leq T$

i.e. $| F(x,y) – 3 | \leq 2$

i.e. F(x,y) = { 1, 2, 3, 4, 5 }

Then F(x,y) = B

Segmented Region : R_2 = { B }

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

(III) Let Seed Point $S_3(6,1) = 2$ and $T = 2$

Predicate :

If $| F(x,y) – S_3(6,1) | \leq T$

i.e. $| F(x,y) – 2 | \leq 2$

i.e. F(x,y) = { 0, 1, 2, 3, 4 }

Then F(x,y) = C

Segmented Region : R_3 = { C }

	0	1	2	3	4	5	6	7
0	B	B	B	B	B	B	A	A
1	B	B	B	B	A	A	A	A
2	B	B	B	A	A	A	A	A
3	A	A	A	A	A	A	C	C
4	A	A	A	A	A	C	C	C
5	A	A	A	A	C	C	C	C
6	C	C	C	C	C	C	C	C
7	C	C	C	C	C	C	C	C

No of Segmented Regions : 3

$R = R_1 \cup R_2 \cup R_3$

(B) Region Splitting & Merging :-

In this method,

- First an image is subdivided into a set of regions and
- Then subdivided adjacent regions are merged to form a new region.

Ex-1 :

10	10	80	80
10	10	80	80
50	50	80	80
50	50	80	80

Predicate : All the pixels in the image must have same intensity value.

By Splitting :

10	10	80	80
10	10	80	80
50	50	80	80
50	50	80	80

By Merging :

10	10	80	80
10	10	80	80
50	50	80	80
50	50	80	80

Ex-2 : Segment the following given image such that the difference between the Maximum intensity value and the minimum intensity value in the segmented region is less than 8 using Region Splitting & Merging Technique

10	9	30	4
7	6	33	37
50	52	54	53
55	52	56	58

Predicate for Splitting:

For Any Region R ;

IF | **Max Intensity Value – Min Intensity Value** | ≥ **8**

Then Split the region into 4 parts

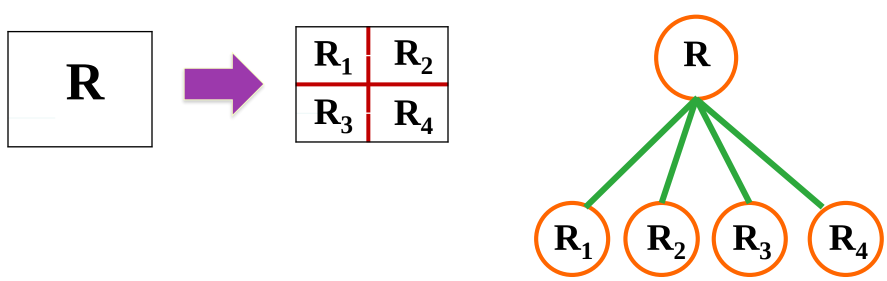
Step-1 : Region Splitting

Predicate for Splitting:

For Any Region R ;

IF | **Max Intensity Value – Min Intensity Value** | ≥ **8**

Then Split the region into 4 parts



(1) For Input Image R

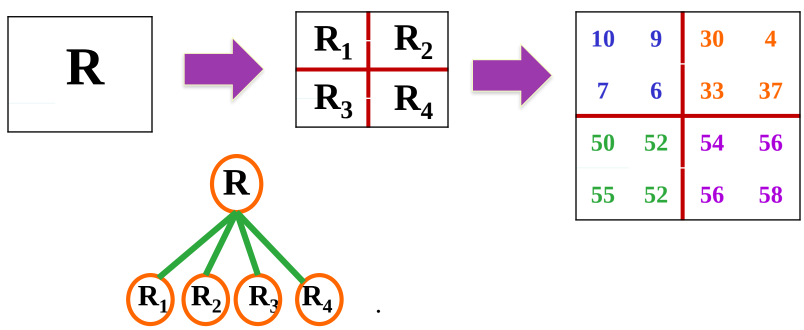
10	9	30	4
7	6	33	37
50	52	54	53
55	52	56	58

Imax = 58 Imin = 4

Imax – Imin = 54 $\geq$ **8**

Predicate (R) = **TRUE**

So, Split the region R into 4 parts



(2) For Region R1

Imax = 10 Imin = 6

Imax – Imax = 4

Predicate (R) = FALSE

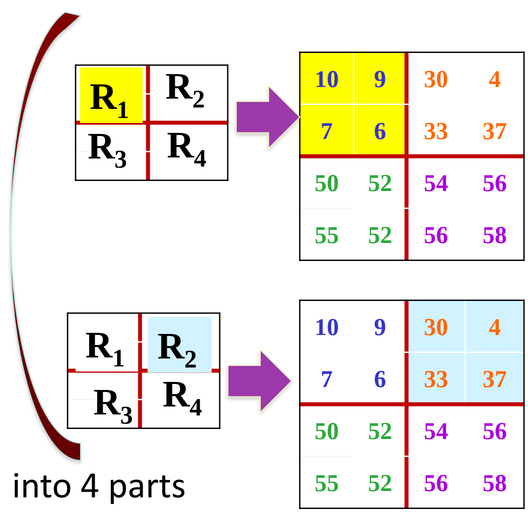
(3) For Region R2

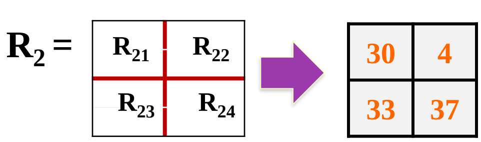
Imax = 37 Imin = 4

Imax – Imax = 33

Predicate (R) = **TRUE**

So, Split the region $\mathbf{R_2}$ into 4 parts





(4) For Region R3

Imax = 55 Imin = 50

Imax – Imax = 5

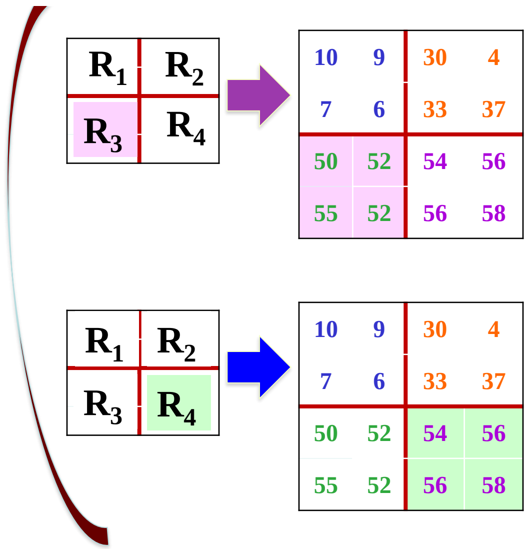
Predicate (R) = **FALSE**

(5) For Region R4

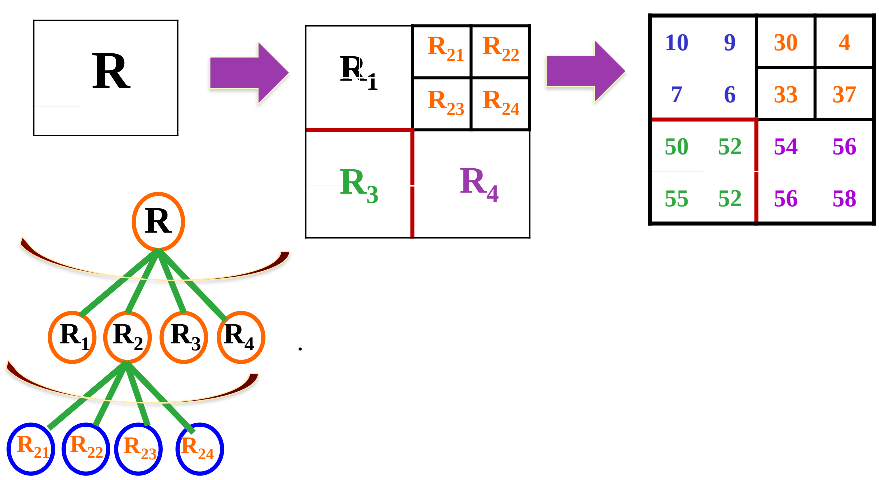
Imax = 58 Imin = 54

Imax – Imax = 4

Predicate (R) = **FALSE**



Result of Splitting



Step-1 : Region Merging

Predicate for Merging

For adjacent Regions R_A and R_B

IF | **Max Intensity Value – Min Intensity Value** | < **8**

Then Merge the regions :

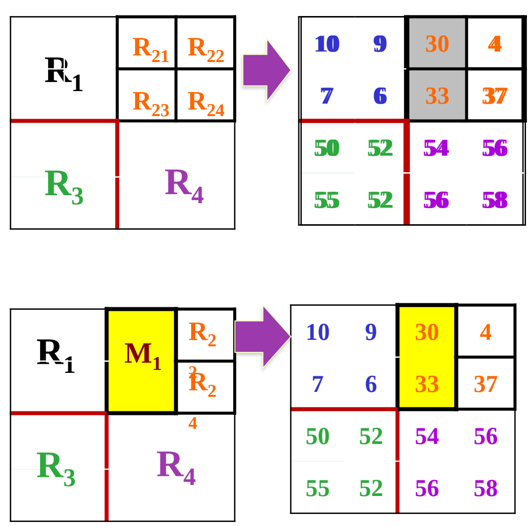
Let $M_{AB} = R_A \cup R_B$

(I) For adjacent Regions
R_{21} R_{22} R_{23} and R_{24}

1. For $\{ R_{21} \cup R_{23} \}$

Imax = 33 Imin = 30
Imax – Imax = 3 < 8
Predicate = **TRUE**
Merge the regions

Let $M_1 = \{ R_{21} \cup R_{23} \}$



2. For $\{ M_1 \cup R_{24} \}$

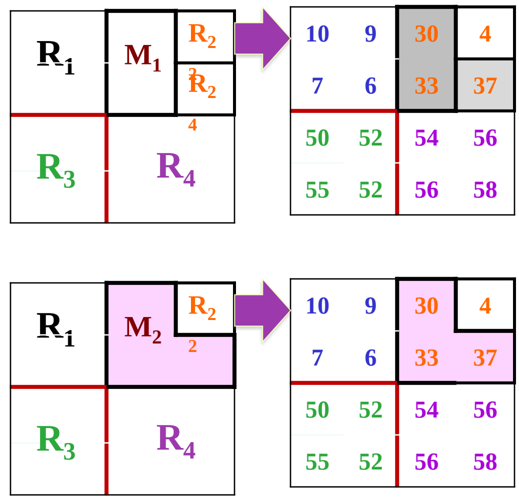
Imax = 37 Imin = 30

Imax – Imax = 7 < 8

Predicate = **TRUE**

Merge the regions

Let $M_2 = \{ M_1 \cup R_{24} \}$



3. For $\{ R_3 \cup R_4 \}$

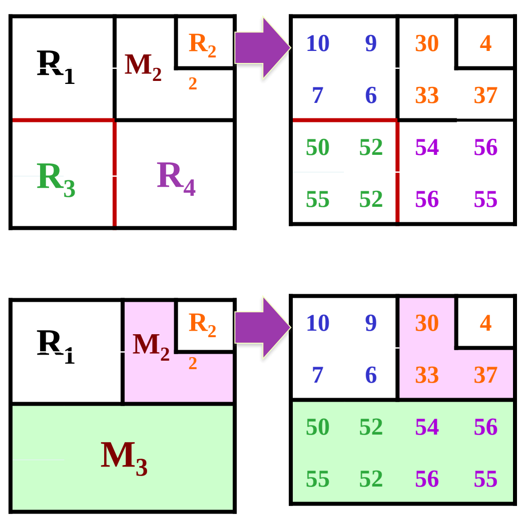
Imax = 56 Imin = 50

Imax – Imax = 6 < 8

Predicate = **TRUE**

Merge the regions

Let $M_3 = \{R_3 \cup R_4\}$



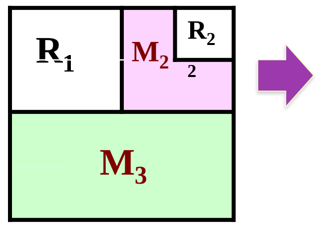
Result of Segmentation :

1) **No of Segmented Regions = 4**
   Such that R = R_1 U R_{22} U M_2 U M_3

2) **Let R_1 = { A }, R_{22} = { B }, M_2 = { C } and M_3 = { D }**

10	9	30	4
7	6	33	37
50	52	54	53
55	52	56	55

Input IMAGE



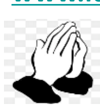
A	A	C	B
A	A	C	C
D	D	D	D
D	D	D	D

Segmented IMAGE

Dr. Kiran TALELE

- Academic : PhD
- Professional :
  - Dean-Students, Alumni & External Relations @ Bharatiya Vidya Bhavans' Sardar Patel Institute of Technology (SP-IT), Mumbai
  - Head-Academic Relation @ Sardar Patel Technology Business Incubator (SP-TBI), Mumbai
  - Treasurer-IEEE Bombay Section

091-9987030881
kiran.talele@spit.ac.in / ktvtalele@gmail.com
https://www.linkedin.com/in/k-t-v-talele/
www.facebook.com/Kiran-Talele-1711929555720263



Stay Connected

- **Dr. Kiran TALELE** is an Associate Professor in Electronics & Telecommunication Engineering Department of Bharatiya Vidya Bhavans' Sardar Patel Institute of Technology, Mumbai with 33+ years experience in Academics.

- He is a Dean of Students, Alumni and External Relations at Sardar Patel Institute of Technology, Andheri Mumbai.
  He is also a Head of Sardar Patel Technology Business Incubator, Mumbai.

- His area of research is Digital Signal & Image Processing, Computer Vision, Machine Learning and Multimedia System Design.

- **He has published 85+ research papers at various national & international refereed conferences and journals. He has filed published 12+ patents at Indian Patent Office. One patent is granted in 2021.**

- He is a Treasurer of IEEE Bombay Section and Mentor for Startup Incubation & Intellectual Asset Creation.

- He received incentives for excellent performance in academics and research from Management of S.P.I.T. in 2008-09. He is a recipient of P.R. Bapat IEEE Bombay Section Outstanding Volunteer Award 2019.