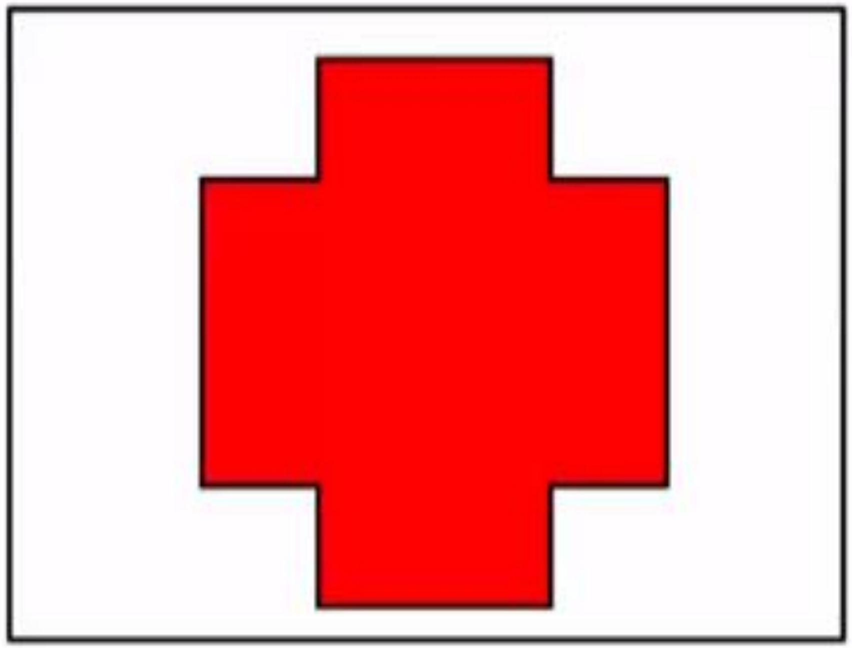
Image Segmentation

Introduction

Image Segmentation : To find out “what” is in the given image.



To extract the constituents that make up the image...

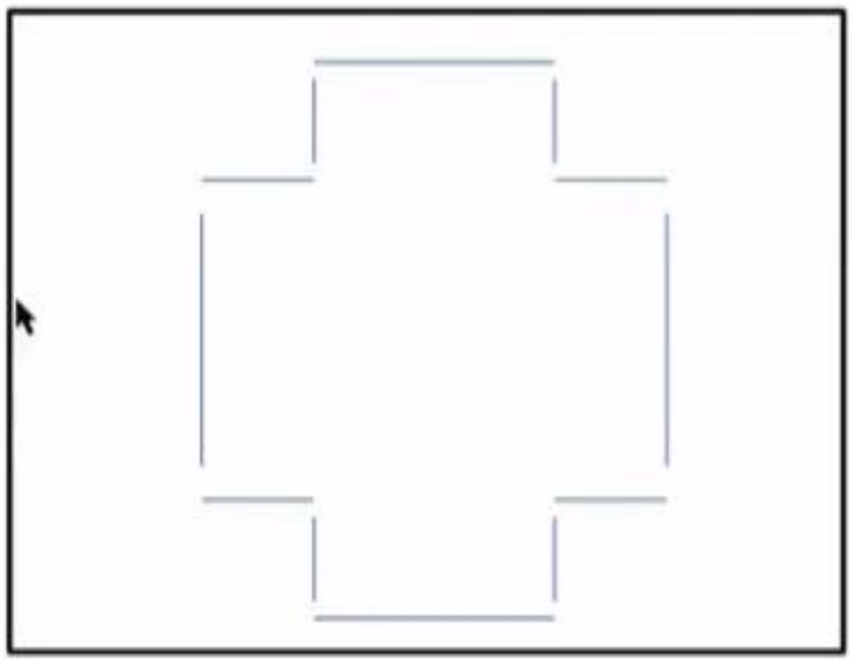
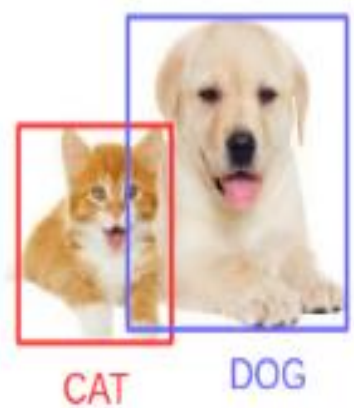


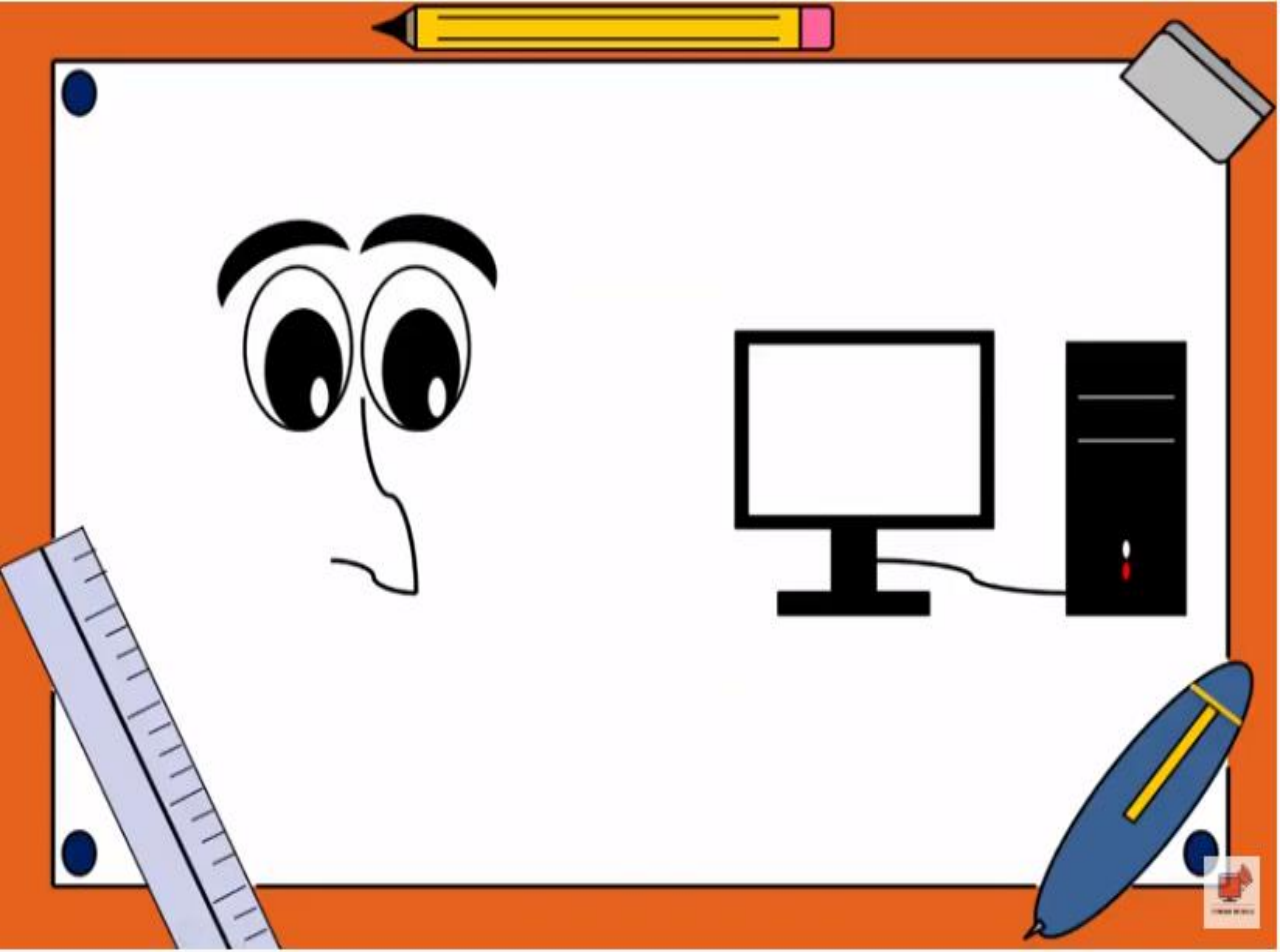
Image Segmentation

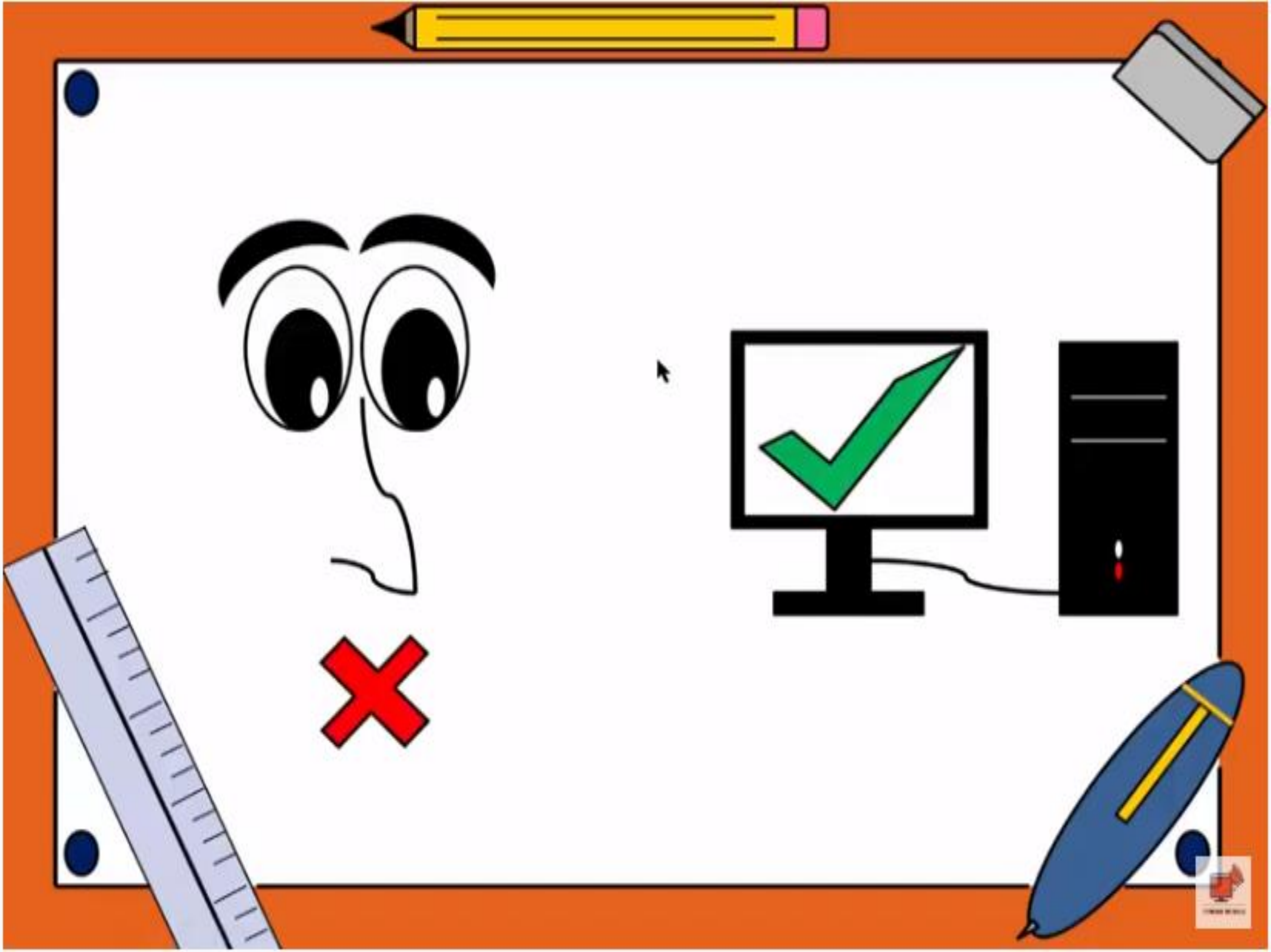
- Image Segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as image objects).

Image Localization



Object Detection

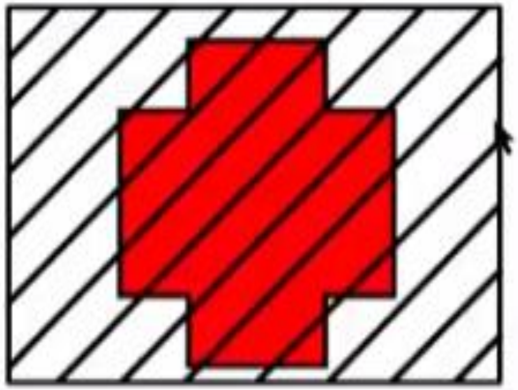


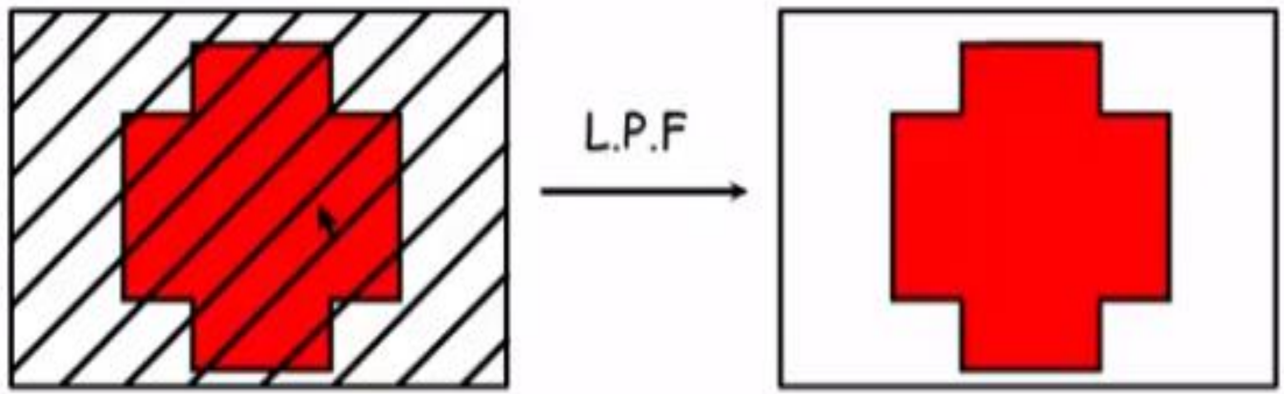


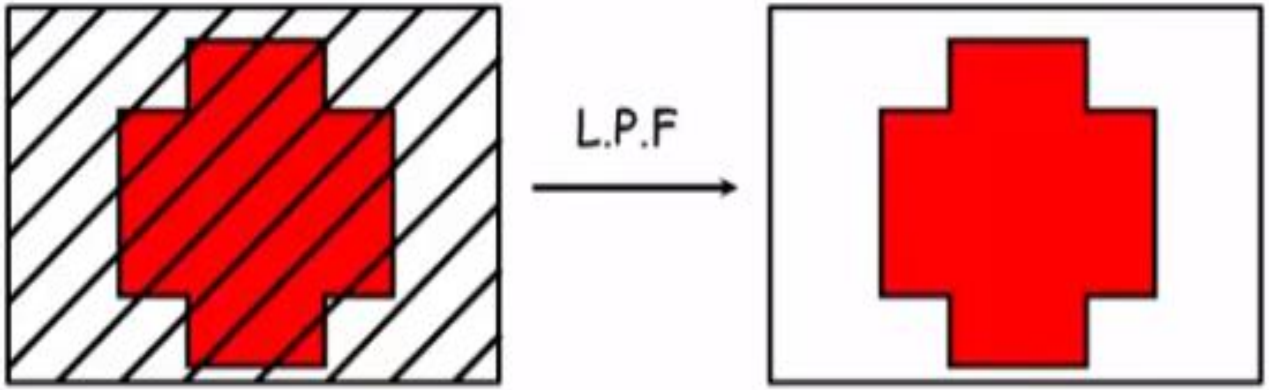
Special Segmentation is not Required.
Inherent Quality to segment : present

Segmentation => Automation

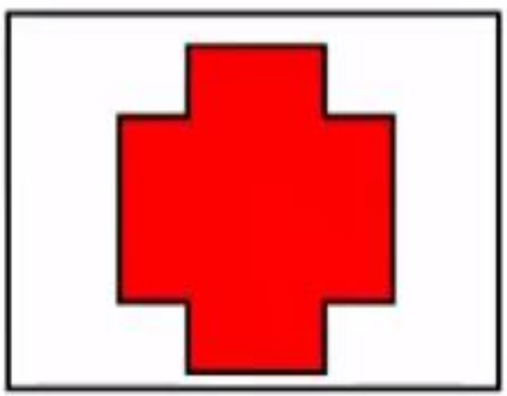
To enable the "Computer" to make decisions. {Computer Vision}

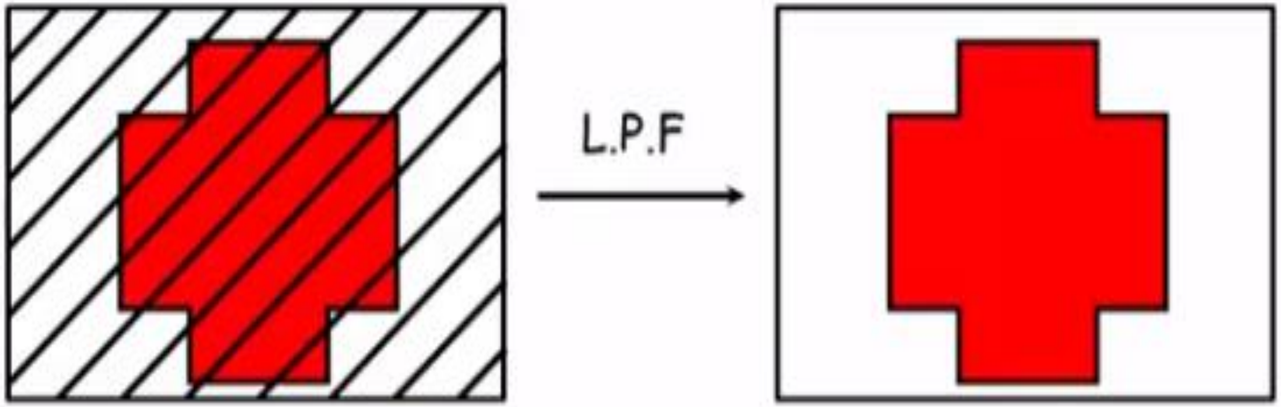



L.P.F

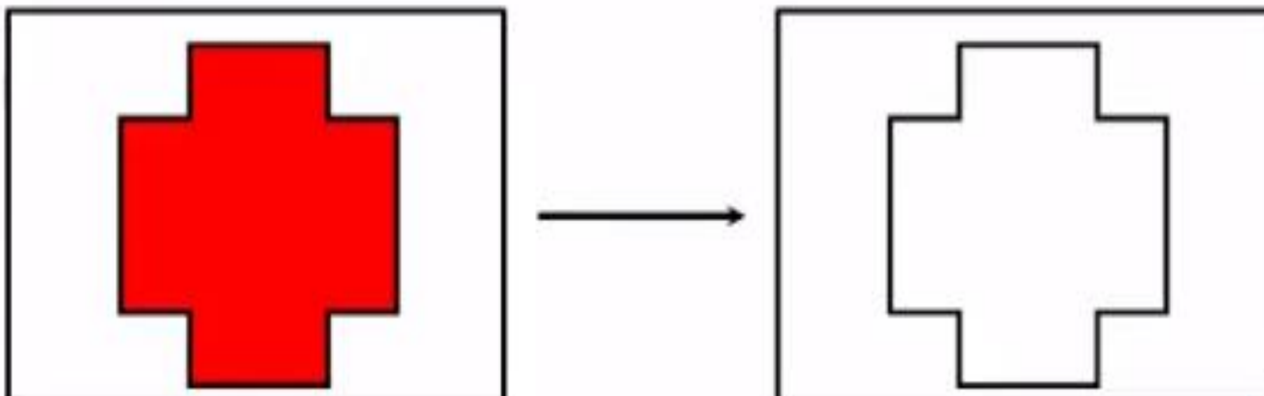


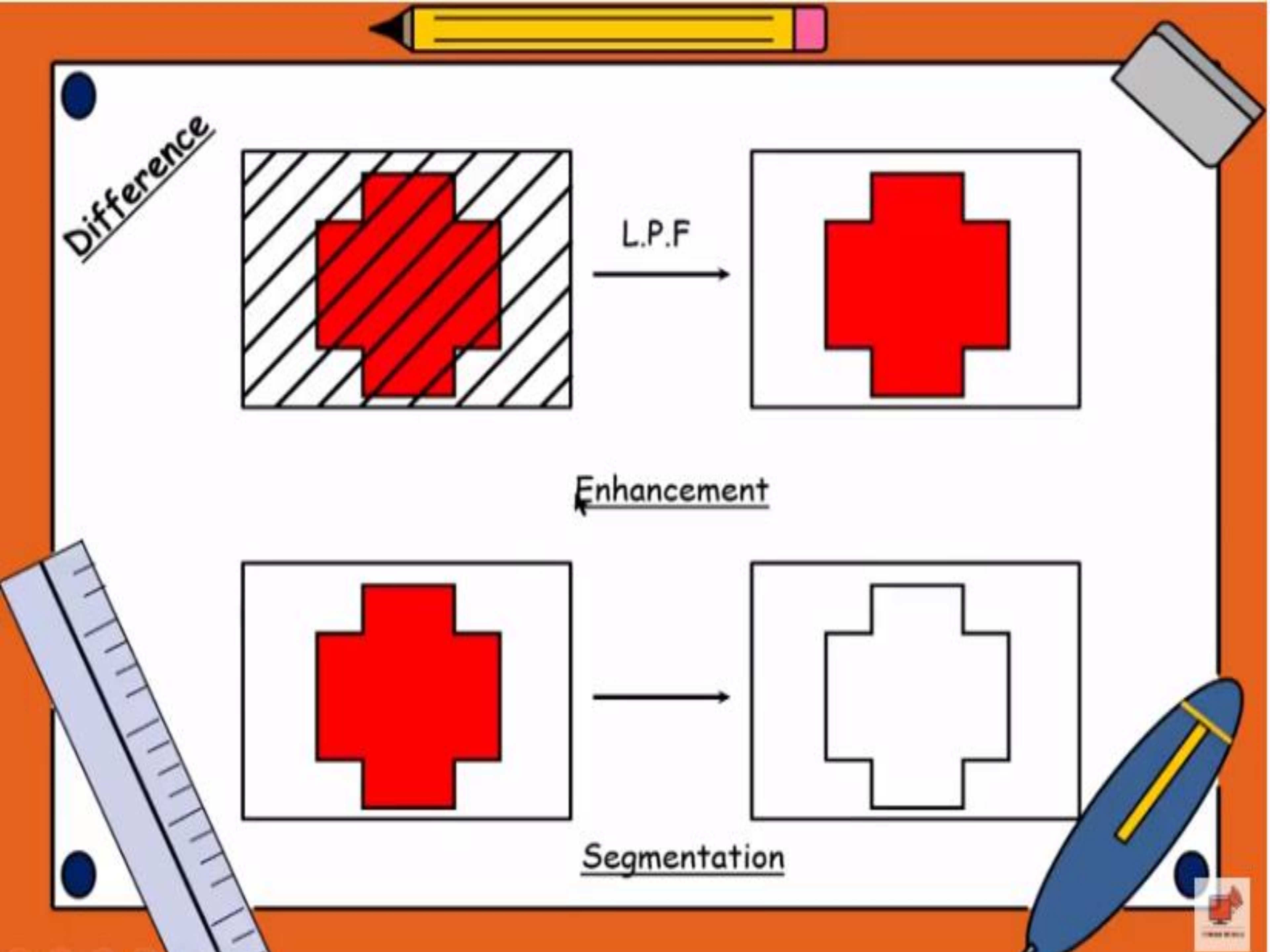
Enhancement





Enhancement




Difference
L.P.F
Enhancement
Segmentation

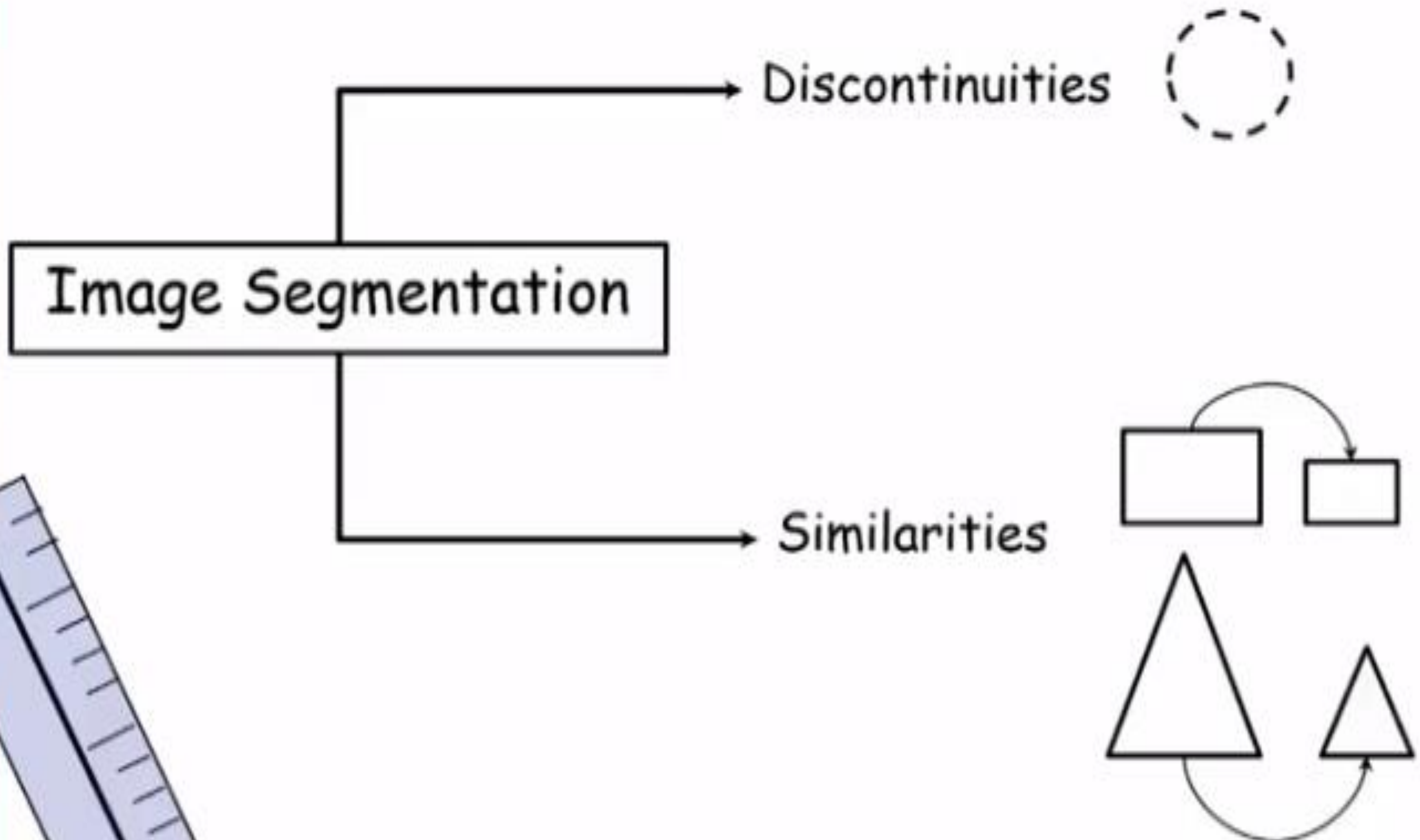


Image Segmentation
Discontinuities
Similarities

Image Segmentation Algorithms

- Image segmentation algorithms are generally based on one of the two basic properties such as

1. **Based on discontinuities in intensity :** It partitions image based on abrupt changes in intensity such as edges.
2. **Based on similarity in intensity :** It partition an image into regions that are similar according to a set of predefined criteria

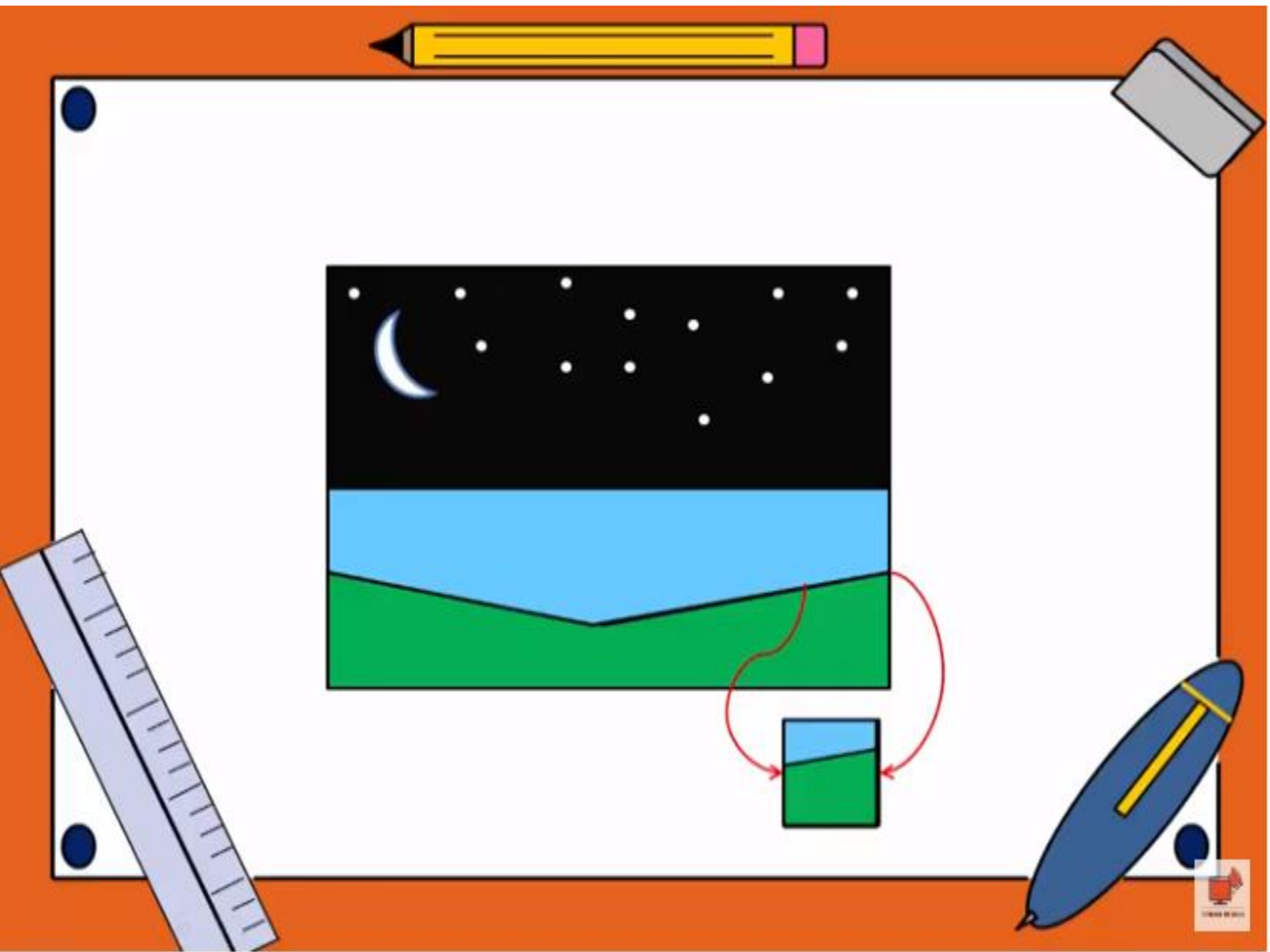
Image Segmentation

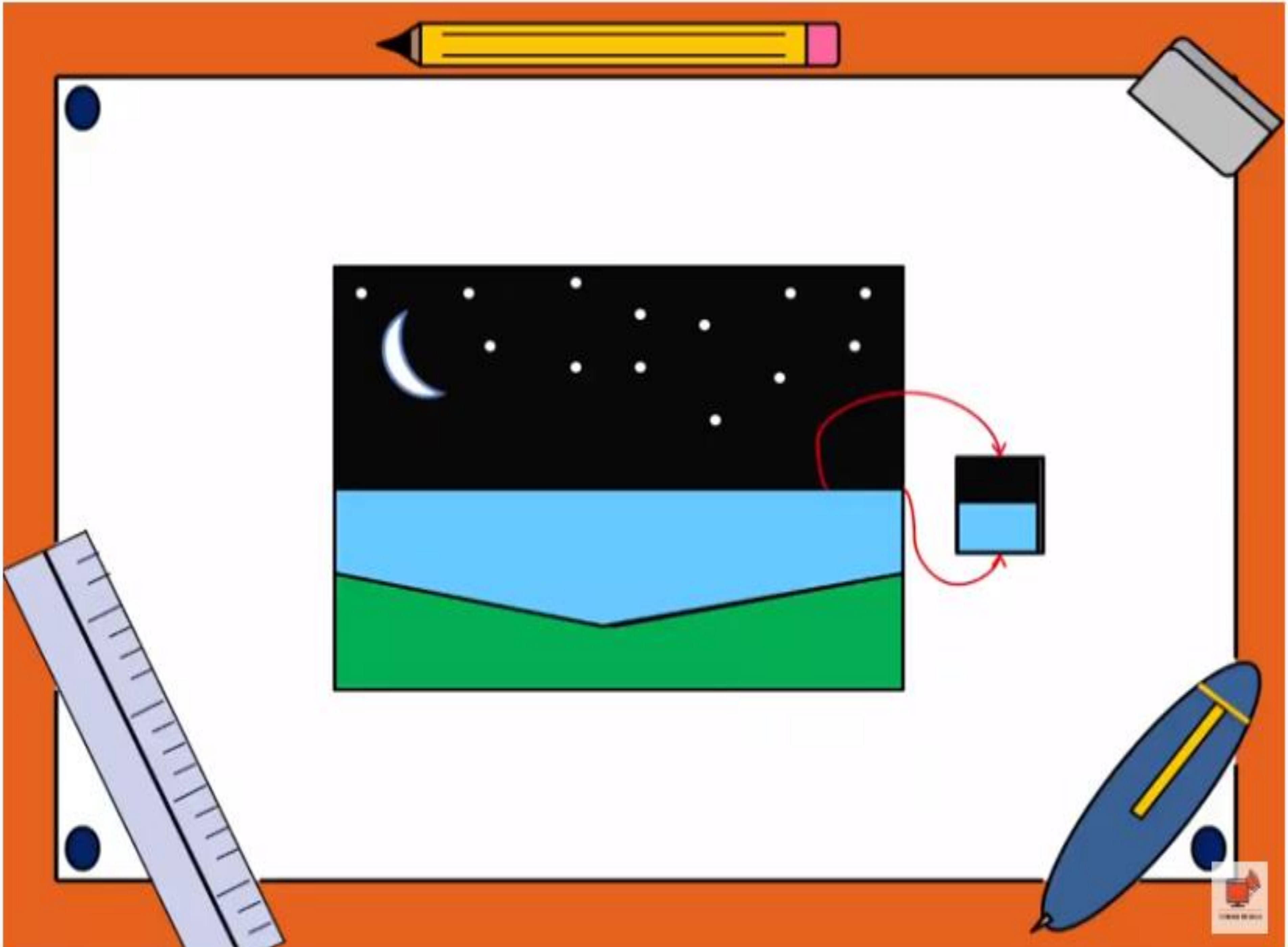
Discontinuities

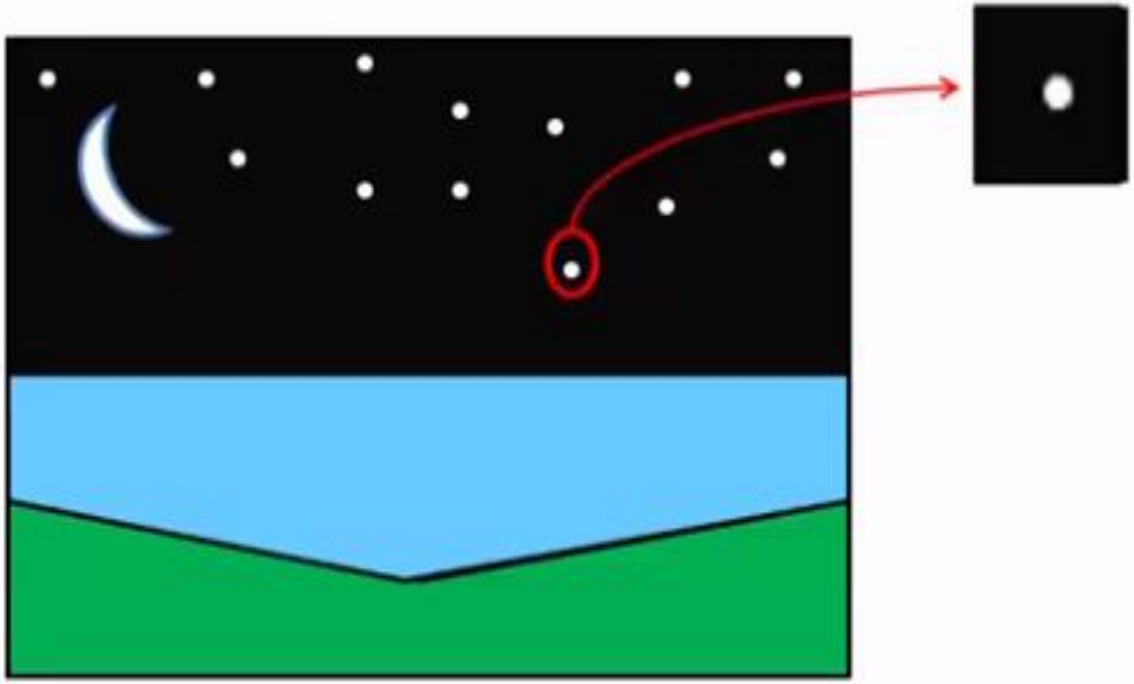
Point, Line
Edge

Discontinuities = Abrupt changes

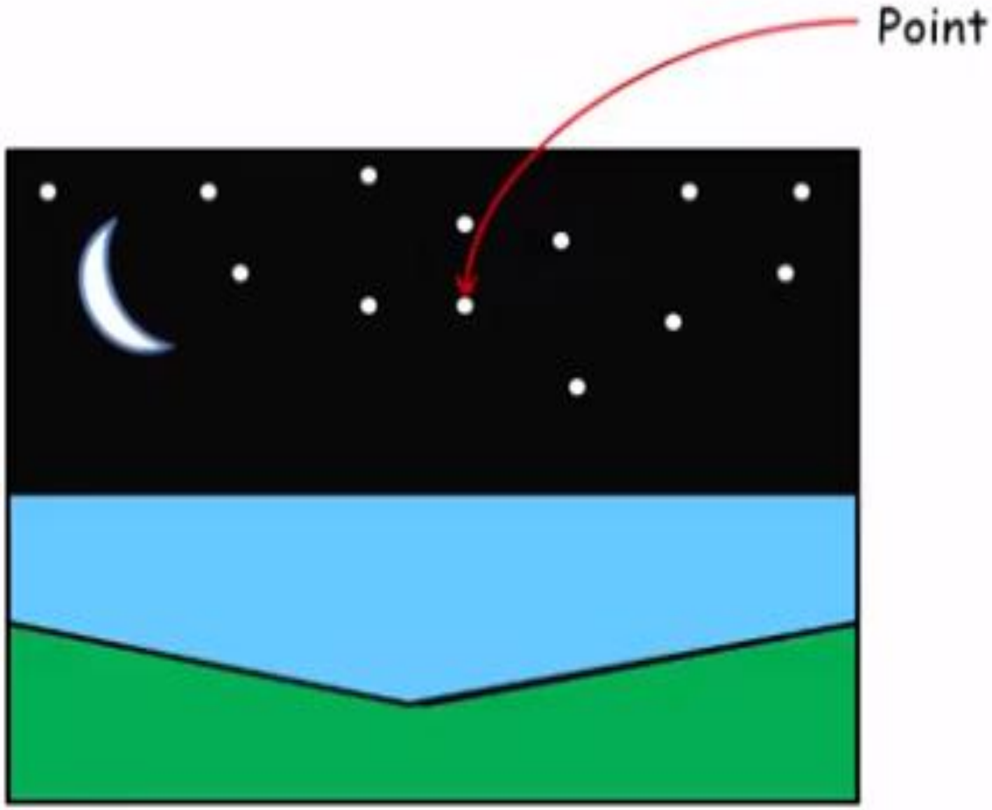
Frequency Changes (in the IP context)







Types of Discontinuities



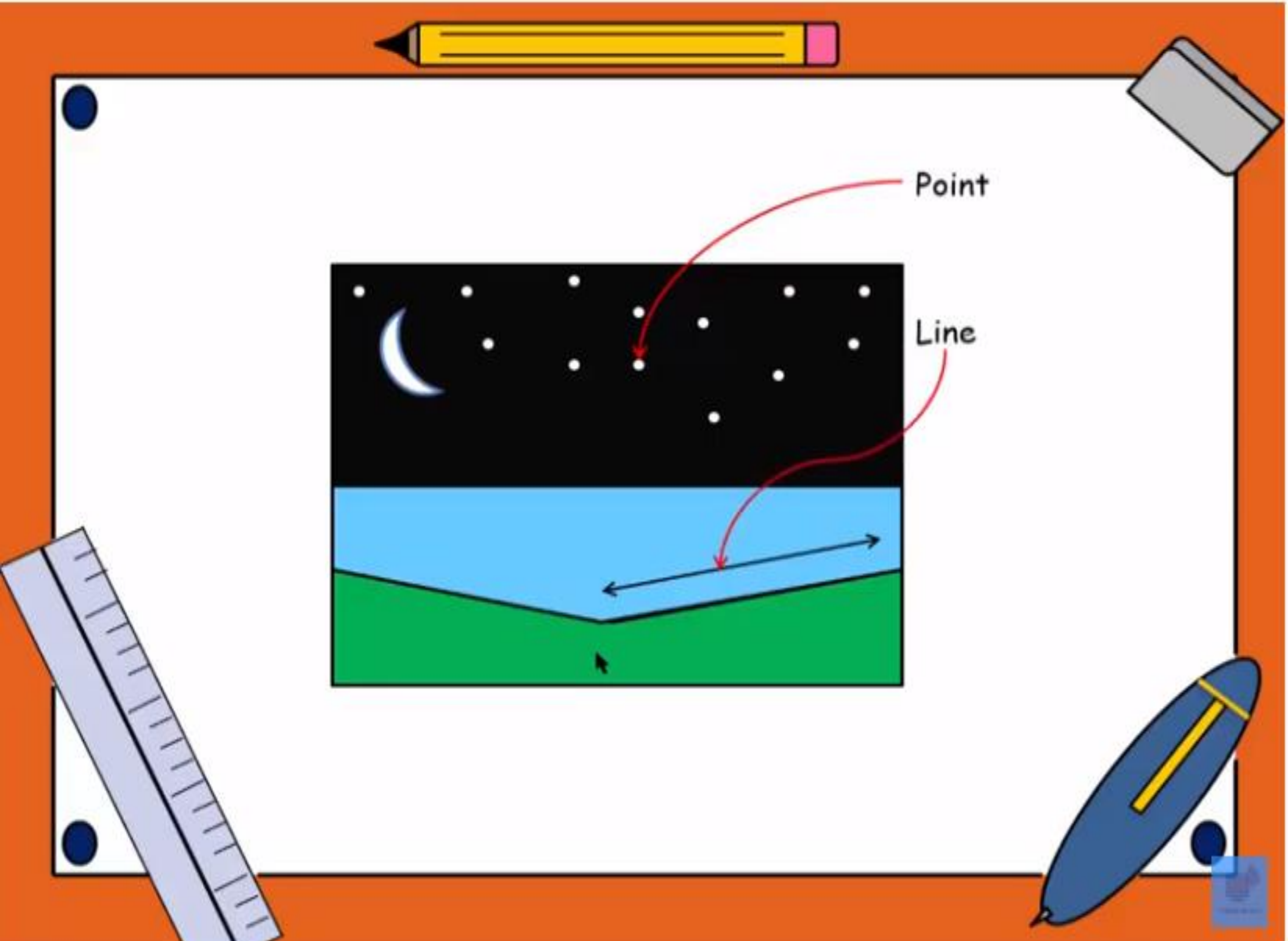
Every "discontinuity" is equivalent to a "change" in frequency.

Noise (a high frequency component)

To detect a high freq comp, we use....

... a H.P.F spatial mask

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


Point
Line

Lines too, can be treated as high freq...

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

But with a "slightly" different mask...

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

But with a "slightly" different mask...

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

But with a “slightly” different mask…

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

This is a mask for a horizontal line...

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

This is a mask for a horizontal line...

Similar masks for lines with diff inclination

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

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

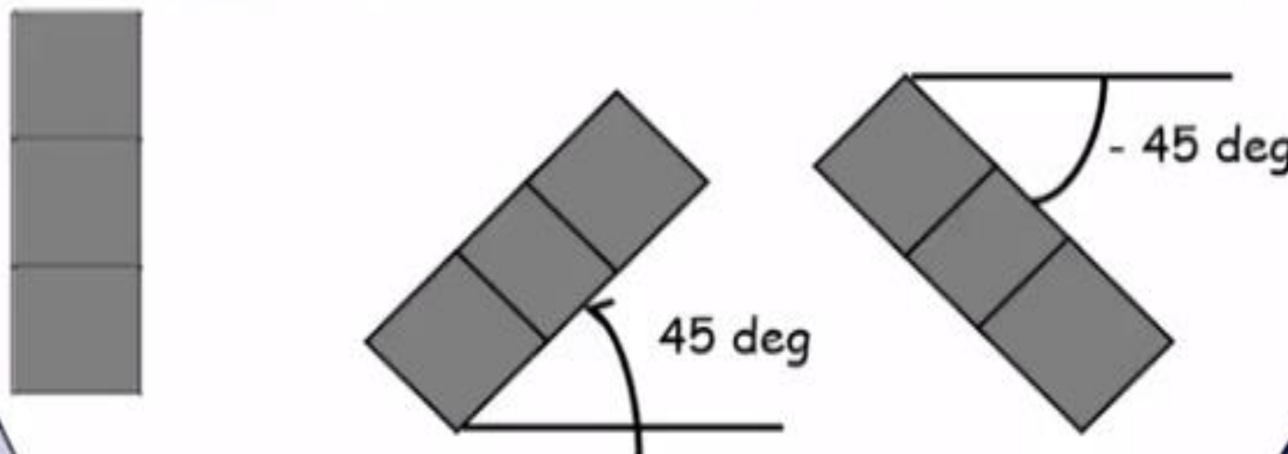
2	-1	-1
-1	2	-1
-1	-1	2

Similar masks for lines with diff inclination

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

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

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

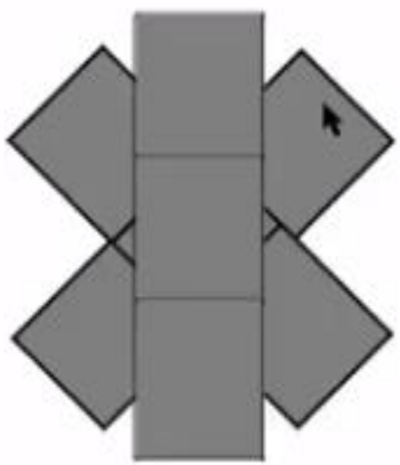


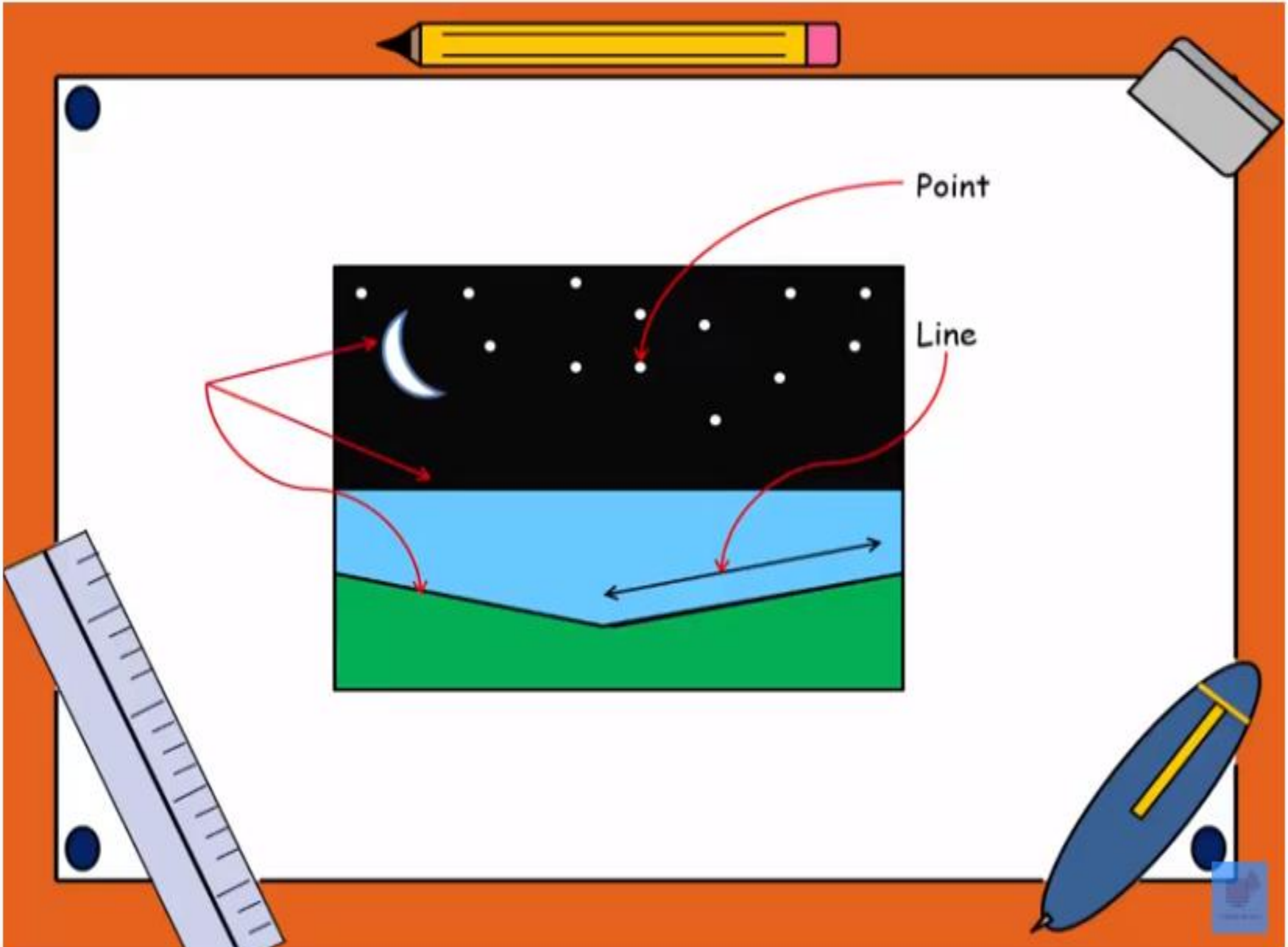
Similar masks for lines with diff inclination

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

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

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




Point
Line

Edge : A boundary between two different areas

Edge : A boundary between two different areas

Line Profile

An edge in a real world image

Line Profile

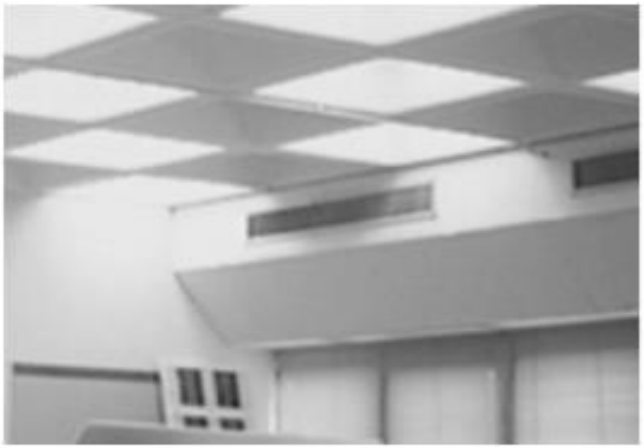
Point Detection

- A point has been detected at the location p(i,j) on which the mask is centered if |R |>T, where T is a nonnegative threshold, and R is obtained with the following mask.

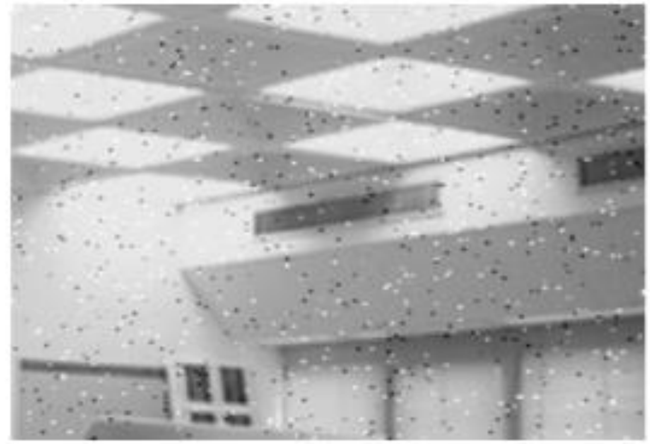
$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- The idea is that the gray level of an isolated point will be quite different from the gray level of its neighbors.

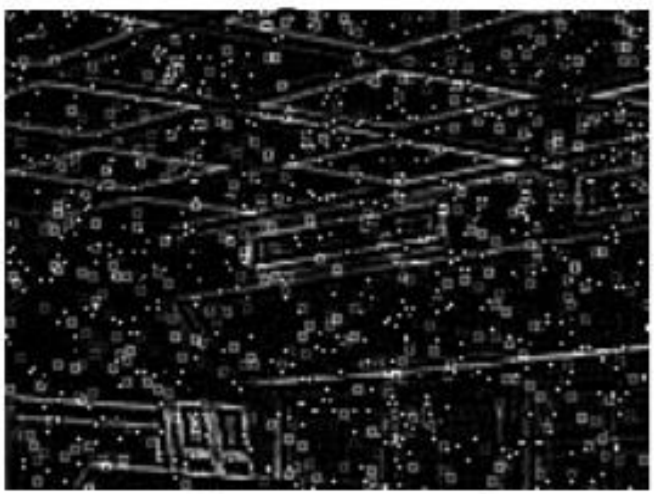
Example



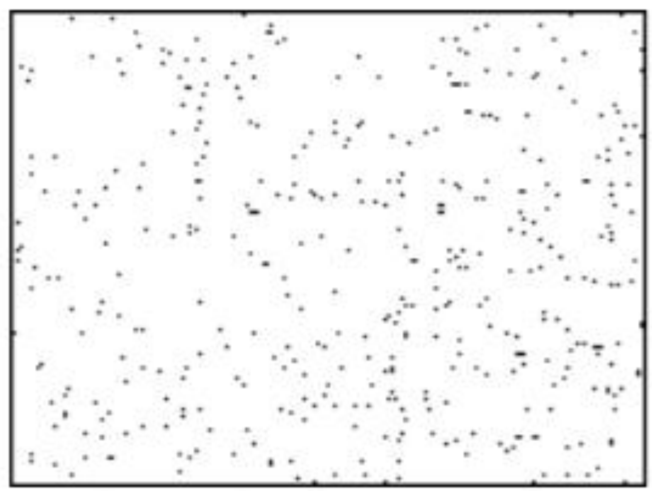
Original



Noise-added



Filtered o/p



Thresholded o/p

Line Detection

- It is based on pattern matching.
- The patterns are directly followed from the definition of a line.
- These pattern templates are designed with suitable coefficients and are applied at each point in an image.
- A set of such templates is shown in figure below:

–1	–1	–1
2	2	2
–1	–1	–1

Horizontal (R_1)

–1	–1	2
–1	2	–1
2	–1	–1

45° (R_2)

2	–1	–1
–1	2	–1
–1	–1	2

–45° (R_3)

–1	2	–1
–1	2	–1
–1	2	–1

Vertical (R_4)

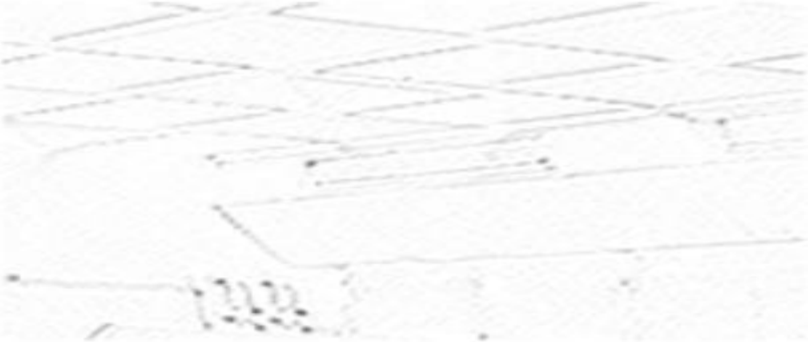
Line Detection

- Let R1, R2, R3 and R4 denote the responses of the masks in from left to right.
- If Ri denotes the response of kernel i, we can apply each of these kernels across an image, and for any particular point,if Ri > Rj for all j ≠ i that point is more likely to contain a line whose orientation (and width) corresponds to that of kernel i.
- One usually thresholds Ri to eliminate weak lines corresponding to edges.

Example

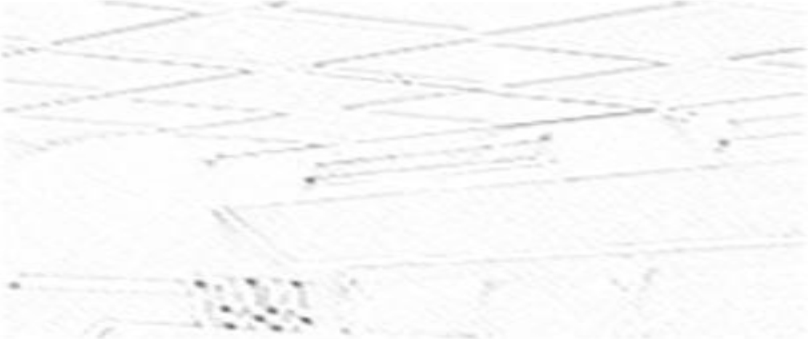
Original

Horizontal line



45° line

Vertical line



-45° line

Edge Detection

- Edge detection is an image processing technique for finding the boundaries of objects within images.
- It locates sharp changes in the intensity function. Edges are pixels where brightness changes abruptly.
- An edge is a property attached to an individual pixel and is calculated from the image function behavior in a neighborhood of the pixel.

Edge Detection

- Magnitude of the first derivative detects the presence of the edge.
- Sign of the second derivative determines whether the edge pixel lies on the dark sign or light side.
- There are many edge detection techniques.
- The discontinuity based edge detection techniques discussed are
    - Roberts Edge Detection
    - Sobel Edge Detection
    - Prewitt Edge Detection
    - Laplacian Edge Detection.

Roberts Edge Detection

- The Roberts edge detection is introduced by Lawrence Roberts (1965).
- It performs a simple, quick to compute, 2-D spatial gradient measurement on an image.
- This method emphasizes regions of high spatial frequency which often correspond to edges.

Roberts Edge Detection

- Robert’s method takes the forward difference between the neighboring pixels.
- Robert’s kernels are derivatives with respect to diagonal elements. Hence they are called cross-gradient operators.
- Thus they are based on cross diagonal differences.
- Robert’s mask for the cross difference is given by

-1	0
0	+1

G_x

0	-1
+1	0

G_y

Example

Roberts Edge Detection

Sobel Edge Detection

- The Sobel edge detection method was introduced by Sobel in 1970.
- The Sobel operator is based on central differences.
- This can be viewed as an approximation of the first Guassian derivative.
- In the Sobel operator, higher weights are assigned to the pixels close to the candidate pixels.

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

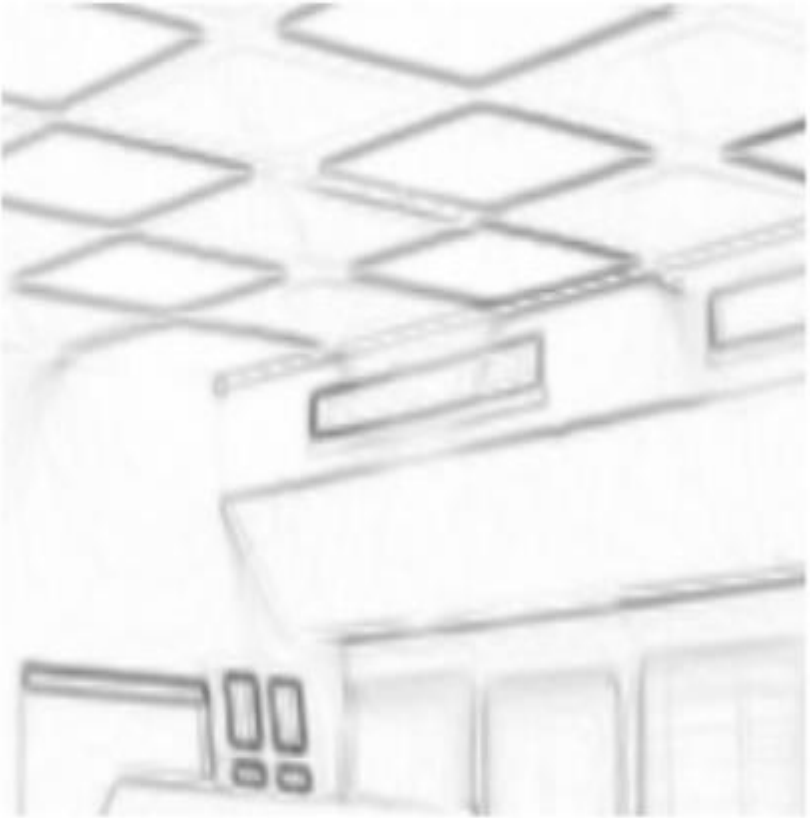
G_x

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

G_y

Example

Original



Processed image

Prewitt Edge Detection

- The Prewitt edge detection was proposed by Prewitt in 1970.
- The Prewitt method takes the central difference of the neighboring pixels.
- The central difference can be obtained using the mask [-1 0 1].
- This method is very sensitive to noise.
- Hence to avoid noise, Prewitt method uses some averaging.
- The prewitt approximation for the 3x3 mask is given as follows:

-1	-1	-1
0	0	0
+1	+1	+1

G_x

-1	0	+1
-1	0	+1
-1	0	+1

G_y

Image Segmentation using Second Derivative

- It is also possible to use second order derivatives to detect edges.
- In the case of a second derivative, the edge pixel is present at the location where the second derivative is zero.
- Thus zero-crossing can be observed as a sign change in pixel differences.
- Laplacian algorithm is one such zero crossing algorithm.

Laplacian Algorithm

- The algorithm is as follows:

1. Generate the mask
2. Apply the mask
3. Detect the zero-crossing. Zero crossing is the situation in which pixels in a neighborhood differ in sign.

Laplacian Mask

- A very popular second order operator is the Laplacian operator.
- The Laplacian has the same properties in all directions and is therefore invariant to rotation in the image.
- For a 3x3 region, the mask for horizontal and vertical edges is given as follows.

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Laplacian Mask

- To recognize the diagonal edges, the mask is rotated by 45 degree.

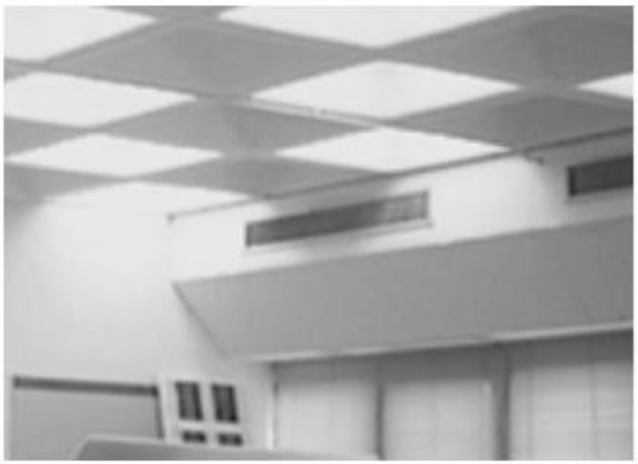
−1	0	−1
0	4	0
−1	0	−1

Laplacian Mask

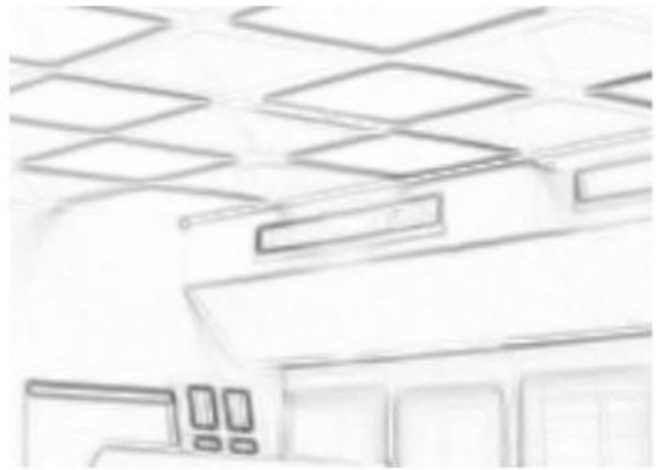
- The addition of these two results in a mask as given below:

−1	−1	−1
−1	8	−1
−1	−1	−1

Example:



Original



Processed image

Laplacian Algorithm

- It is seldom used in practice for edge detection for the following reasons:
  1. As a 2nd-order derivative, it is unacceptably sensitive to noise.
  2. It produces double edges and is unable to detect edge direction.
- The Laplacian usually plays the secondary role of detector for establishing whether a pixel is on the dark or light side of an edge.

Problems

Ex. 1. Apply horizontal and vertical line detection mask on the following image F. Use appropriate threshold value. Assume virtual rows and columns by repeating border pixel values.

F =

6	5	10
100	100	100
4	20	10

Image after replication

6	6	5	10	10
6	6	5	10	10
100	100	100	100	100
4	4	20	10	10
4	4	20	10	10

Line detection Mask

Horizontal

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

Vertical

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

Response of Horizontal Line Mask

6	6	5	10	10
6	-283	-279	-275	10
100	555	545	535	100
4	-272	-266	-260	10
4	4	20	10	10

Response of Vertical Line Mask

6	6	5	10	10
6	2	-12	10	10
100	-15	20	-5	100
4	-32	52	-20	10
4	4	20	10	10

R_H: Response of Horizontal mask. R_V: Response of Vertical mask

$|R_H| > |R_V|$: Point is more likely associated with a line in direction of mask.

$|R_H|$

6	6	5	10	10
6	283	279	275	10
100	555	545	535	100
4	272	266	260	10
4	4	20	10	10

$|R_V|$

6	6	5	10	10
6	2	12	10	10
100	15	20	5	100
4	32	52	20	10
4	4	20	10	10

Horizontal line is present

Th>=500 can be set

Ex. 2 For a given image F

F =

6	5	7
2	8	4
6	3	7

6	6	5	7	7
6	6	5	7	7
2	2	8	4	4
6	6	3	7	7
6	6	3	7	7

W1 =

0	0	0
-1	2	-1
0	0	0

W2 =

0	-1	0
0	2	0
0	-1	0

W3 =

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

Apply the following filter mask W1, W2, W3 on the input image F and obtain the output image.

F =

6	5	7
2	8	4
6	3	7

6	6	5	7	7
6	6	5	7	7
2	2	8	4	4
6	6	3	7	7
6	6	3	7	7

W1 =

0	0	0
-1	2	-1
0	0	0

6	6	5	7	7
6	1	-3	2	7
2	-6	10	-4	4
6	3	-7	4	7
6	6	3	7	7

F =

6	5	7
2	8	4
6	3	7

6	6	5	7	7
6	6	5	7	7
2	2	8	4	4
6	6	3	7	7
6	6	3	7	7

W2 =

0	-1	0
0	2	0
0	-1	0

6	6	5	7	7
6	4	-3	3	7
2	-8	8	-6	4
6	4	-5	3	7
6	6	3	7	7

F =

6	5	7
2	8	4
6	3	7

6	6	5	7	7
6	6	5	7	7
2	2	8	4	4
6	6	3	7	7
6	6	3	7	7

W3 =

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

6	6	5	7	7
6	5	-6	5	7
2	-14	18	-10	4
6	7	-12	7	7
6	6	3	7	7

Ex. 3 Derive 8 Directional Laplacian filter Mask (3×3)

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

Laplacian filter Mask (3×3)

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

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

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

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

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

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

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

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

Ex. 5 Write different line detection masks in an image. Detect 45^0 inclined line in the following image z

Z =

10	10	100
10	100	10
100	10	10

Different line detection masks in an image

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

Horizontal

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

$+45^0$

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

Vertical

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

-45^0

Use pixel replication at border to transform 3×3 image to 5×5.

z =

10	10	100
10	100	10
100	10	10

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

10	10	10	100	100
10	-90	0	90	100
10	0	0	0	10
100	90	0	-90	10
100	100	10	10	10

Horizontal

10	10	10	100	100
10	-90	0	360	100
10	0	540	0	10
100	360	0	-90	10
100	100	10	10	10

$+45^0$

10	10	10	100	100
10	-90	0	90	100
10	0	0	0	10
100	90	0	-90	10
100	100	10	10	10

Vertical

10	10	10	100	100
10	180	-270	-180	100
10	-270	0	-270	10
100	-180	-270	180	10
100	100	10	10	10

-45^0

Problem

- What will be the effect of applying the Laplacian filter for point detection in the image given below?

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

Solution

- After pixel replication

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

Applying Mask

−1	−1	−1
−1	8	−1
−1	−1	−1

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

0	0	0
-150	-150	-150
150	150	150
0	0	0

Problem

- Apply proper mask and detect 45^{o} inclined line in the following image. Assume T=100

10	10	100
10	100	10
100	10	10

Solution

- After pixel replication

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

Applying Mask

−1	−1	2
−1	2	−1
2	−1	−1

45°

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

Problem

- Apply Prewitt on the following image.

1	2	4
3	2	1
1	1	3

Solution

- After pixel replication

1	1	2	4	4
1	1	2	4	4
3	3	2	1	1
1	1	1	3	3
1	1	1	3	3

Applying Mask

-1	-1	-1
0	0	0
+1	+1	+1

1	1	2	4	4
1	1	2	4	4
3	3	2	1	1
1	1	1	3	3
1	1	1	3	3

G_x

Applying Mask

-1	0	+1
-1	0	+1
-1	0	+1

1	1	2	4	4
1	1	2	4	4
3	3	2	1	1
1	1	1	3	3
1	1	1	3	3

G_y

Thank you!!!