Brightness Adaptation

Brightness adaptation refers to the process by which our visual system adjusts its sensitivity to light based on the level of illumination in the environment. Our eyes have a remarkable ability to adapt to different levels of brightness and adjust our perception of brightness accordingly.

When we are in a low-light environment, our pupils dilate to allow more light into the eye, and our visual system becomes more sensitive to low levels of light. Conversely, in a bright environment, our pupils constrict to limit the amount of light entering the eye, and our visual system becomes less sensitive to light.

This adaptation is critical in image processing because it allows us to see images accurately in a range of lighting conditions. However, it can also create challenges in image processing since the same image can appear differently depending on the lighting conditions in which it is viewed. For example, an image that looks bright and clear in a well-lit room may appear dark and unclear in a dimly lit environment.

Brightness Discrimination

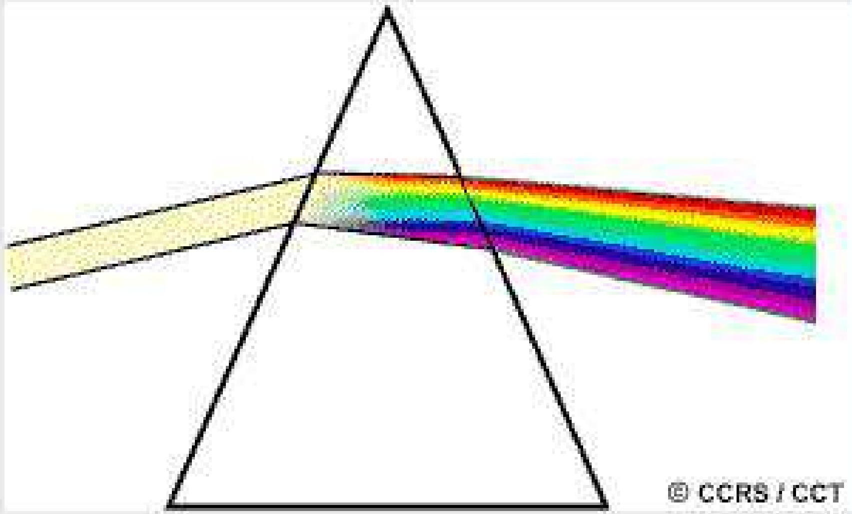
Brightness discrimination refers to the ability of our visual system to distinguish between different levels of brightness in an image. Our visual system can detect differences in brightness levels as small as one percent. This ability is crucial in image processing because it allows us to perceive and distinguish different features and objects in an image.

For example, if we look at a photograph of a person, our visual system can distinguish the brightness levels between the person's hair, skin, and clothing, allowing us to perceive each feature separately.

Brightness discrimination is also essential in image processing applications, such as image segmentation and edge detection, which require the detection of boundaries between different regions of an image. These boundaries are typically defined by changes in brightness levels, and the ability to discriminate these levels accurately is critical in these applications.

Conclusion

In conclusion, brightness adaptation and discrimination are two crucial concepts in image processing that underlie our ability to perceive and interpret images accurately. Brightness adaptation allows us to see images accurately in different lighting conditions, while brightness discrimination allows us to distinguish between different features and objects in an image. Understanding these concepts is critical in the development of image processing algorithms and applications that can accurately process and interpret images.



Blue, **green**, and **red** are the **primary colours** or wavelengths of the visible spectrum. They are defined as such because no single primary colour can be created from the other two, but all other colours can be formed by combining blue, green, and red in various proportions. Although we see sunlight as a uniform or homogeneous colour, it is actually composed of various wavelengths of radiation in primarily the ultraviolet, visible and infrared portions of the spectrum. The visible portion of this radiation can be shown in its component colours when sunlight is passed through a **prism**, which bends the light in differing amounts according to wavelength.

* Digital Image Processing

It is the use of a digital computer to process digital images through an algorithm. It covers low, mid and high-level processes.

→ low-level : inputs and outputs are images.

→ mid-level : outputs are attributes extracted from input images.

→ high-level : an ensemble of recognition of individual objects

* Pixel

It is the smallest controllable element of a picture represented on the screen.

A Simple Image Formation Model

→ We know that an object is represented in the form of a 2-D image.

→ For an image to be produced, there should be a light source illuminating the object.

→ When an image is generated from a physical process, its values are proportional to the energy radiated by a physical source (ex. electromagnetic waves).

→ Also, the intensity / amplitude of f at spacial coordinates is a positive scalar quantity whose physical meaning is determined by the source of the image.

→ Therefore, the function must be non-zero and finite, ie.

$$0 < f(x,y) < \infty$$

→ The function may be characterized by 2 components:

- the amount of source illumination incident on the scene being viewed
- the amount of illumination reflected by the objects in the scene.

→ These illumination and reflectance components can be ~~den~~ denoted by $i(x,y)$ and $r(x,y)$ respectively.

→ These 2 functions combine to form $f(x,y)$:

$$f(x,y) = i(x,y) * r(x,y)$$

→ Therefore, $\boxed{f(x,y) = i(x,y) * r(x,y)}$

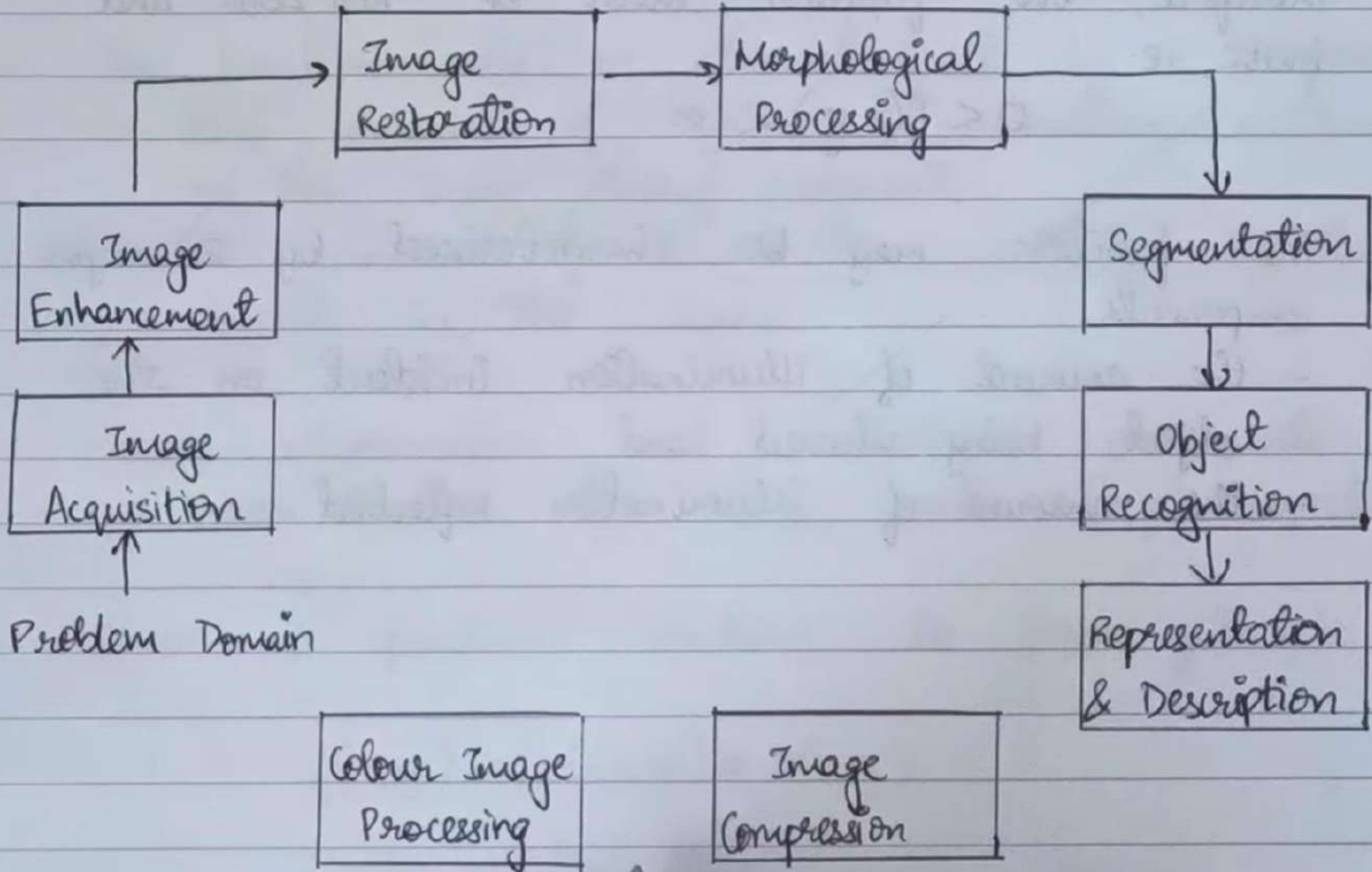
where $f(x,y)$: intensity at the point (x,y)
$i(x,y)$: illumination at the point (x,y)
$r(x,y)$: reflectance / transmissivity at the point (x,y)
$0 \leq i(x,y) < \infty$
$0 \leq r(x,y) \leq 1$

→ Reflectance is bounded by 0 (total absorption) and 1 (total reflectance).

→ The nature of $i(x,y)$ is determined by the illumination source.

→ The nature of $r(x,y)$ is determined by the characteristics of the imaged objects.

Key Stages in DIP



1) Image Acquisition

The image is captured by a sensor (ex. camera) and digitized if the output of the camera or sensor is not already in digital form, using analogue-to-digital convertor.

2) Image Enhancement

The process of manipulating an image so that the result is more suitable than the original for specific applications.

Enhancing an image brings out the hidden details of an image and highlights certain features which may be important.

Key stages in digital image processing

5) Image Segmentation

Segmentation procedures partition an image into its constituent parts or objects. The more accurate the segmentation, the more likely the recognition is to succeed.

6) Object Recognition

The process that assigns a label to an object based on the information provided by its description.

3:14 / 6:18 · Image Segmentation ›

7) Representation and Description

Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing (mainly recognition)

Description : also called feature selection, deals with extracting attributes that result in some information of interest.

8) Image Compression

It includes techniques for reducing the storage required to save an image or the bandwidth required to transmit it.

8) Image Compression

It includes techniques for reducing the storage required to save an image or the bandwidth required to transmit it.

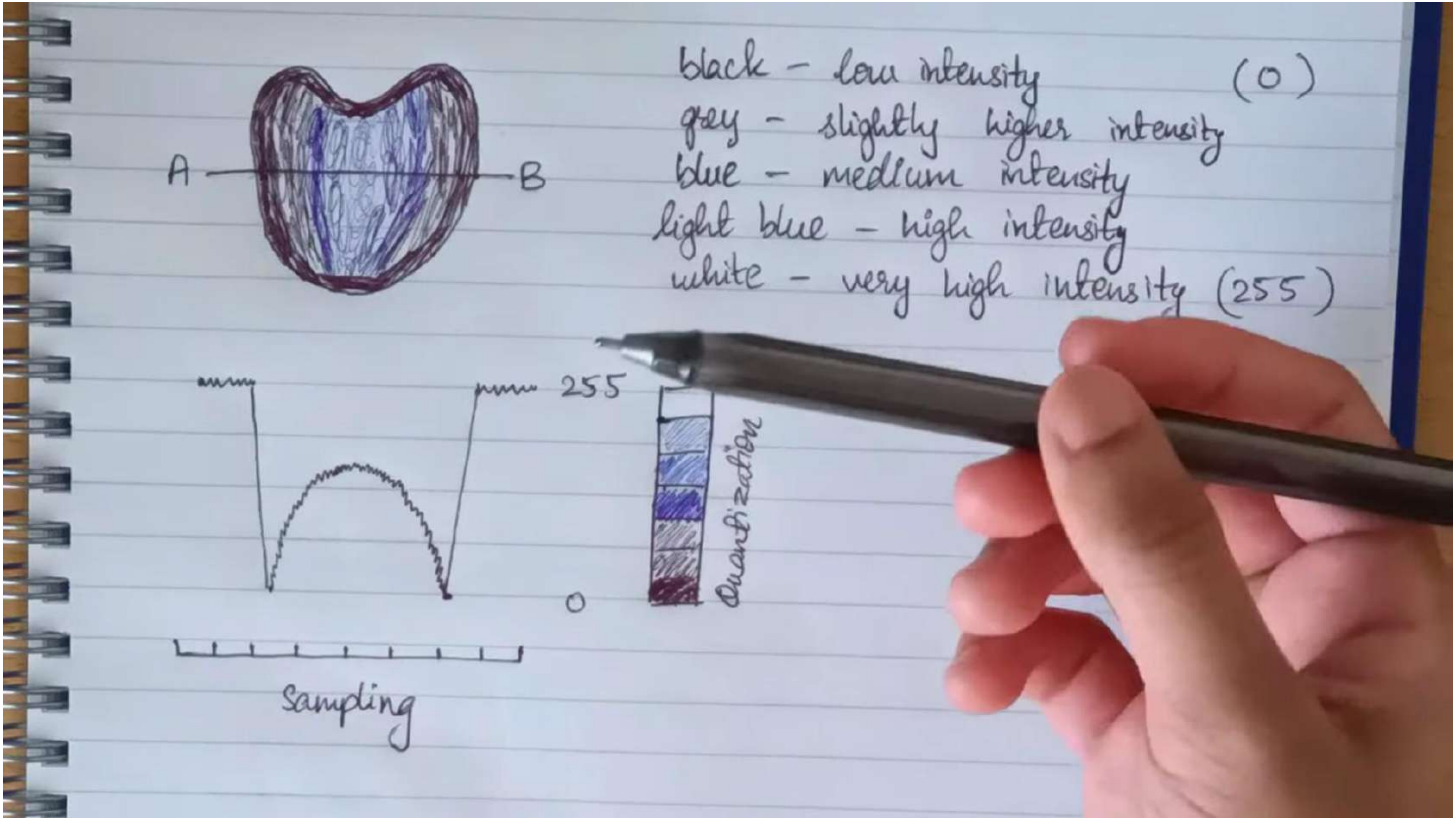
9) Colour image processing

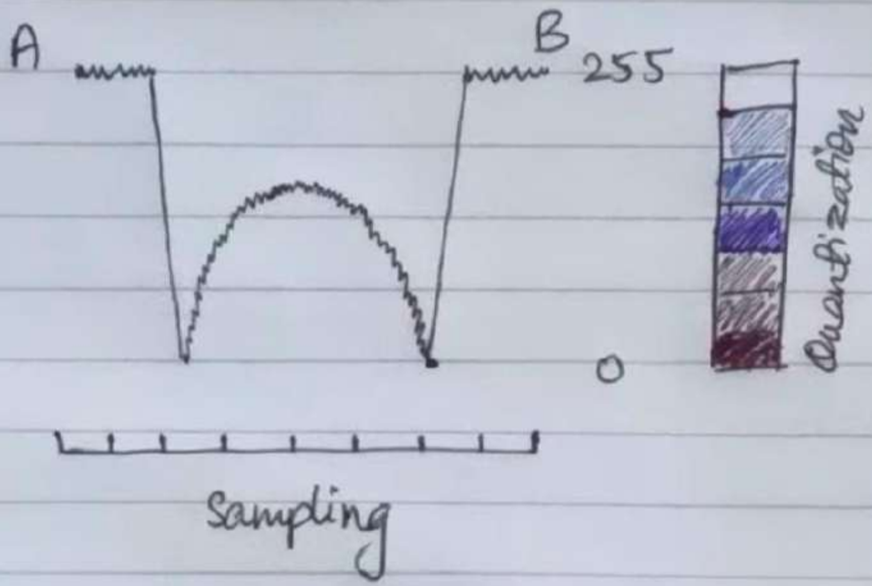
It is an area that has been gaining in importance because of the significant increase in the use of digital images over the internet. It includes the use of colour of the image to extract features of interest in the image.

Sampling and Quantization

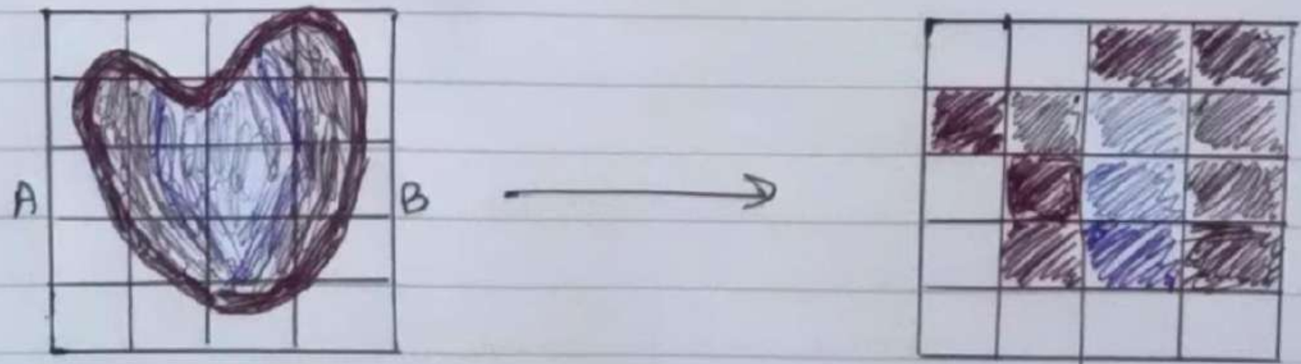
→ In order to become suitable for digital processing, an image function must be digitized both spatially and in amplitude. This digitization process involves two main processes called

1) Sampling - Digitizing the co-ordinate value is called sampling.
2) Quantization - Digitizing the amplitude value is called quantization.


black - low intensity (0)
grey - slightly higher intensity
blue - medium intensity
light blue - high intensity
white - very high intensity (255)
A
B
255
0
Quantization
sampling



analog → digital



Analog image

Digital image

Relationship between Pixels, Neighbourhood and Adjacency of Pixels

1) Neighbourhood of Pixels

Any image can be represented as the following:

y			
$(y+1)$	$(x-1, y+1)$	$(x, y+1)$	$(x+1, y+1)$
(y)	$(x-1, y)$	(x, y)	$(x+1, y)$
$(y-1)$	$(x-1, y-1)$	$(x, y-1)$	$(x+1, y-1)$
	$(x-1)$	(x)	$(x+1)$

x

a) $N_4(P)$ 4-neighbours : the set of horizontal and vertical neighbours

	$(x, y+1)$	
$(x-1, y)$	(x, y)	$(x+1, y)$
	$(x, y-1)$	

b) $N_D(P)$ diagonal neighbours : the set of 4 diagonal neighbours.

$(x-1, y+1)$		$(x+1, y+1)$
	(x, y)	
$(x-1, y-1)$		$(x+1, y-1)$

c) $N_8(p)$ 8-neighbours : union of 4-neighbours and diagonal neighbours

$(x-1, y+1)$	$(x, y+1)$	$(x+1, y+1)$
$(x-1, y)$	(x, y)	$(x+1, y)$
$(x-1, y-1)$	$(x, y-1)$	$(x+1, y-1)$

$$N_8(p) = N_4(p) + N_D(p)$$

2) Connectivity / Adjacency

Two pixels that are neighbours and have the same gray level are adjacent.

a) 4- adjacency

Binary image

0	1	0	1	8-adjacency
0	0	1	0	4-adjacency
0	0	1	0	
1	0	0	0	

not connected

$V = \{1\}$

b) 8-adjacency

Gray-scale image

54	10	100	8	8-adjacency
81	150	2	34	4-adjacency
201	200	3	45	
7	70	147	56	

not adjacent

$V = \{1, 2, 3, \text{---} 10\}$

c) m-adjacency (mixed adjacency)

$V = \{1\}$

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

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

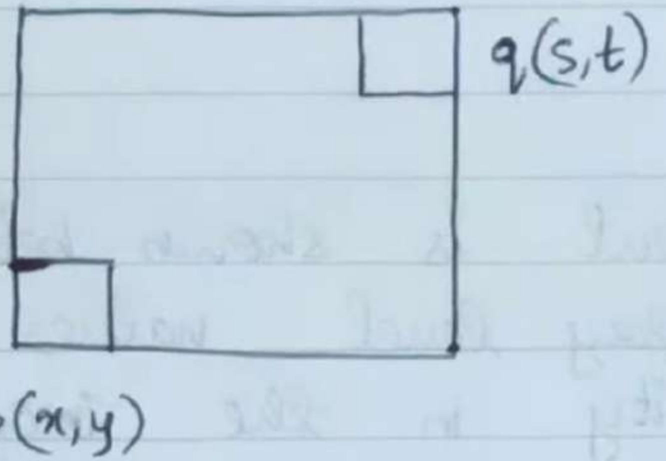
✓ 4-adjacency

× 8-adjacency

m-adjacen

Distance Measures between Pixels

Let p and q be pixels with coordinates (x, y) and (s, t) respectively.



1) Euclidean distance

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

2) City block distance

$$D_4(p,q) = |x-s| + |y-t|$$

Ex. The pixels with D_4 distance ≤ 2 from (x,y) form the following contours of constant distance:

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

3) Chessboard distance

$$D_8(p,q) = \max(|x-s|, |y-t|)$$

Ex. The pixels with D_8 distance ≤ 2 from (x,y) form the following contours of constant distance:

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

Distance Measures Between Pixels With Examples

4) Dm distance

It is defined as the shortest m-path between the points. This distance between 2 pixels depends on the values of the pixels along the path as well as the values of their neighbours.

p(x, y) ———————— q(s, t)

3:56 / 17:10

Distance Measures Between Pixels With Examples

Important note: The D_4 and D_8 distances between p and q are independent of any paths that might exist between the points because these distances involve only the coordinates of the points.

Question

1. An image segment is shown below. Let V be the set of gray level values used to define connectivity in the image. Compute D_4, D_8 and D_m distances between pixels 'p' and 'q' for:

(i) $V = \{2, 3\}$

4:31 / 17:10

Distance Measures Between Pixels With Examples

Question

1. An image segment is shown below. Let V be the set of gray level values used to define connectivity in the image. Compute D_4, D_8, and D_m distances between pixels 'p' and 'q' for:

(i) $V = \{2, 3\}$

(ii) $V = \{2, 6\}$

(0,0)	2(p)	3	2	6	1
	6	2	3	6	2
	5	3	2	3	5
	2	4	3	5	2
	4	5	2	3	6(q) (4,4)

Coordinates of $p(x, y) = (0,0)$.

Coordinates of $q(s, t) = (4,4)$.

$$D_4(p,q) = |x-s| + |y-t|$$
$$= |0-4| + |0-4|$$
$$= 4 + 4$$
$$= \underline{\underline{8 \text{ units}}}.$$

$$D_8(p,q) = \max(|x-s|, |y-t|)$$
$$= \max(|0-4|, |0-4|)$$
$$= \max(4, 4)$$
$$= \underline{\underline{4}}$$

7:28 / 17:10

Arithmetic Operations Between Images

These are array operations which are carried out between corresponding pixel pairs. The four arithmetic operations are denoted as:

$$s(x,y) = f(x,y) + g(x,y)$$

$$d(x,y) = f(x,y) - g(x,y)$$

$$p(x,y) = f(x,y) \times g(x,y)$$

$$v(x,y) = f(x,y) \div g(x,y)$$

Few Important Points:

→ If the result is a floating point number, round off its value.

→ If the result is above the pixel range, select the max range value.

→ If the result is below the pixel range, select the min range value.

→ If the result is infinity, write it as zero.

Arithmetic Operations and Logical Operations between Images in Digital image processing

Addition

0 - 255

$$\overset{A}{\begin{bmatrix} 0 & 100 & 10 \\ 4 & 0 & 10 \\ 8 & 0 & 5 \end{bmatrix}} + \overset{B}{\begin{bmatrix} 10 & 100 & 5 \\ 2 & 0 & 0 \\ 0 & 10 & 10 \end{bmatrix}} = \begin{bmatrix} 10 & 200 & 15 \\ 6 & 0 & 10 \\ 8 & 10 & 15 \end{bmatrix}$$

Uses

- Addition of noisy images for noise reduction.
- Image averaging in the field of astronomy.

2:34 / 10:49

Arithmetic Operations and Logical Operations between Images in Digital image processing

Subtraction

0 - 255

$$\overset{A}{\begin{bmatrix} 0 & 100 & 10 \\ 4 & 0 & 10 \\ 8 & 0 & 5 \end{bmatrix}} - \overset{B}{\begin{bmatrix} 10 & 100 & 5 \\ 2 & 0 & 0 \\ 0 & 10 & 10 \end{bmatrix}} = \begin{bmatrix} 0 & 0 & 5 \\ 2 & 0 & 10 \\ 8 & 0 & 0 \end{bmatrix}$$

Uses

- Enhancement of differences between images.
- Mask mode radiography in medical imaging.

4:13 / 10:49

Arithmetic Operations and Logical Operations between Images in Digital image processing

Multiplication

0-255

$$\begin{bmatrix} 0 & 100 & 10 \\ 4 & 0 & 10 \\ 8 & 0 & 5 \end{bmatrix} * \begin{bmatrix} 10 & 100 & 5 \\ 2 & 0 & 0 \\ 0 & 10 & 10 \end{bmatrix} = \begin{bmatrix} 0 & 255 & 50 \\ 8 & 0 & 0 \\ 0 & 0 & 50 \end{bmatrix}$$

Uses

- Shading correction
- Masking (or) Region of interest (ROI) operations.

Arithmetic Operations and Logical Operations between Images in Digital image processing

$0.5 \rightarrow \underline{0}$

Division

$$\overset{A}{\begin{bmatrix} 0 & 100 & 10 \\ 4 & 0 & 10 \\ 8 & 0 & 5 \end{bmatrix}} \Big/ \overset{B}{\begin{bmatrix} 10 & 100 & 5 \\ 2 & 0 & 0 \\ 0 & 10 & 10 \end{bmatrix}} = \begin{bmatrix} 0 & 1 & 2 \\ 2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Uses

- Shading correction

6:47 / 10:49

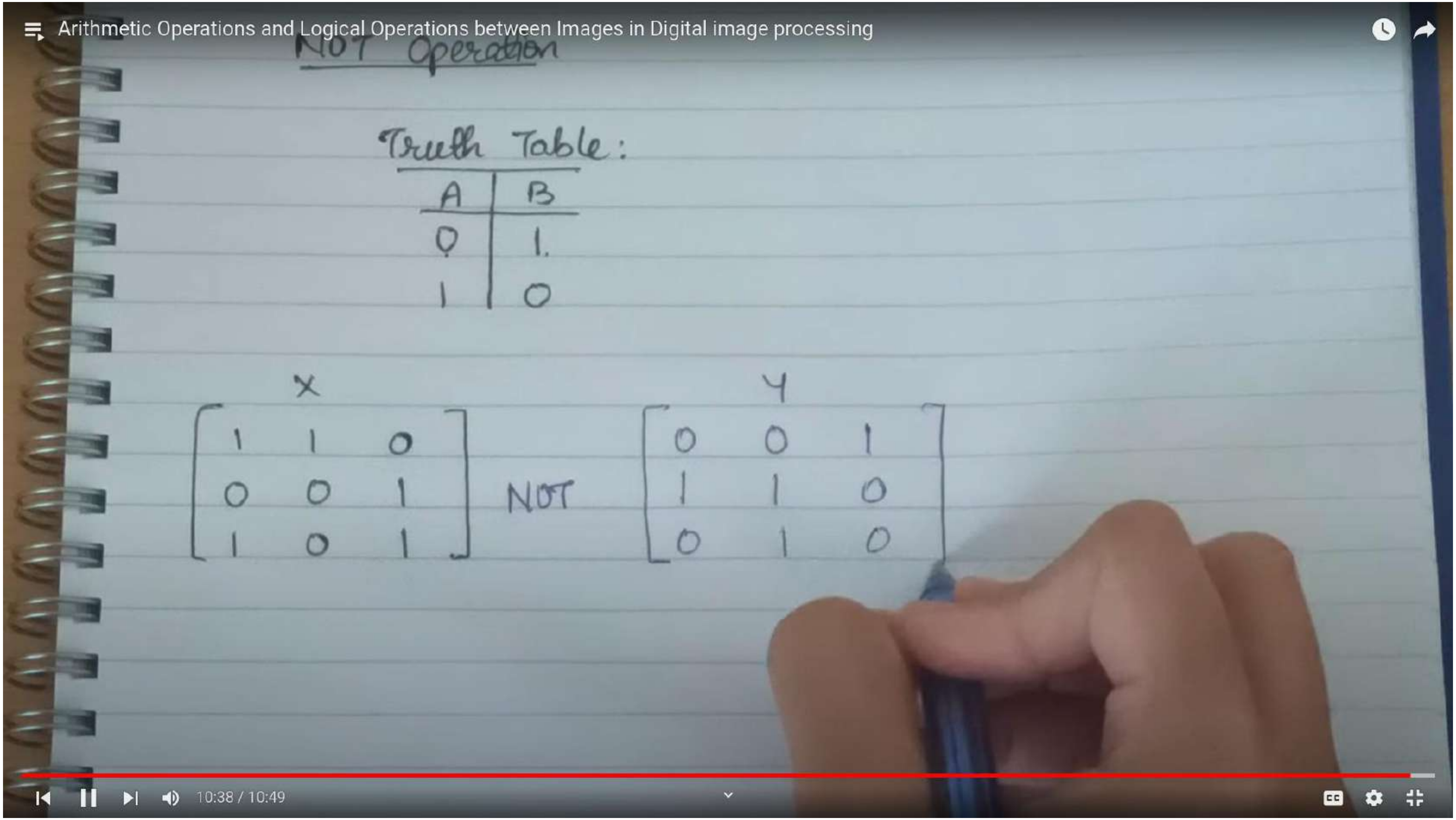
Logical Operations on Images

AND Operation

Truth Table :

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

$$\overset{X}{\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}} \text{ AND } \overset{Y}{\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}} = \overset{Z}{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}$$


Arithmetic Operations and Logical Operations between Images in Digital image processing
NOT Operation
Truth Table:
A | B
0 | 1.
1 | 0
X
1 1 0
0 0 1
1 0 1
NOT
Y
0 0 1
1 1 0
0 1 0
10:38 / 10:49

Point Operations

They are a method of image processing in which each pixel in the output image is only dependent upon the corresponding pixel in the input image and is independent of its location or neighbouring pixels.

Let 'r' be the gray value at a point (x, y) of the input image $f(x, y)$ and 's' be the gray value at a point (x, y) of the output image $g(x, y)$, then the point operation can be defined as :

pixels.

$$\begin{bmatrix} 2 & 1 \\ 2 & 3 \end{bmatrix} \longrightarrow \begin{bmatrix} \; & - \\ - & - \end{bmatrix}$$

Let 'r' be the gray value at a point (x, y) of the input image $f(x, y)$ and 's' be the gray value at a point (x, y) of the output image $g(x, y)$, then the point operation can be defined as :

$$s = T(r)$$

where T is the point operation of a certain gray-level mapping relationship between the original image and the output image.

1) Digital negative

$[0, 7]$ $[(0) - (L-1)]$

$0 \to 7$

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

$$S = (L-1) - r$$
$$S = (8-1) - r$$
$$= 7 - r.$$

Assuming the image to be 3-bit, we have $2^3 = 8$ levels.

$2^0 = 1$
$2^1 = 2$
$2^2 = 4$
$2^3 = \underline{8}$ $\quad 0 - 8$

$r = 0, \; S = 7 - 0 = 7.$
$r = 1, \; S = 7 - 1 = 6$
$r = 2, \; S = 7 - 2 = 5.$
$\vdots$
$r = 7, \; S = 7 - 7 = 0$

$n = 3$
$L = 2^n = 2^3 = \underline{\underline{8}}.$

$S = (L-1) - r$

$S = (8-1) - r$

$= 7 - r.$

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

Assuming the image to be 3-bit, we have $2^3 = 8$ levels.

$2^0 = 1$

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$ 0 - 8

$r = 0, \; S = 7 - 0 = 7.$

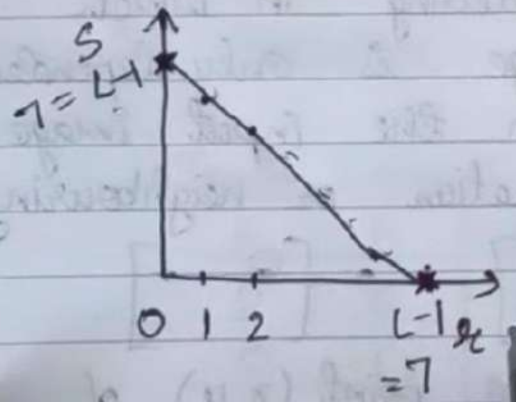
$r = 1, \; S = 7 - 1 = 6$

$r = 2, \; S = 7 - 2 = 5.$

$[0, L-1]$ $n = 3$

$L = 2^n = 2^3 = 8.$

$r = 7, \; S = 7 - 7 = 0.$



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

Digital Negative

2) Thresholding with $T = 4$

$L = 8$

$L - 1 = 7$

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

$$s = \begin{cases} L-1 = 7 \ ; & r \geq 4 \\ 0 \ ; & r < 4 \end{cases}$$

$$\left[\begin{array}{l} r = 0, 1, 2, 3 \rightarrow s = 0 \\ r = 4, 5, 6, 7 \rightarrow s = 7 \end{array}\right]$$

7	0	7	0
0	7	7	7
0	0	7	7
7	7	7	0

Output imag

Point operations in digital image processing with examples

2) Thresholding with T = 4

$L = 8$

$L - 1 = 7$

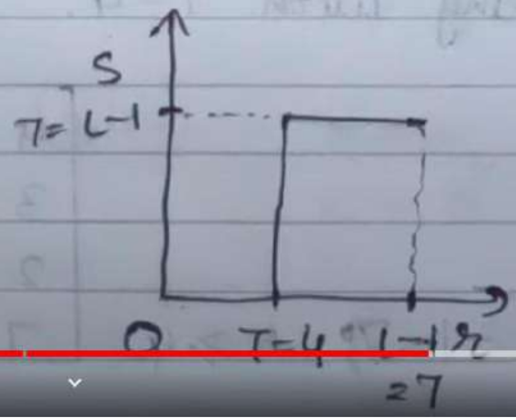
4	3	5	2
3	6	4	6
2	2	6	5
7	6	4	1

$$s = \begin{cases} L-1 = 7 & ; r \geq 4 \\ 0 & ; r < 4 \end{cases}$$

$$\left[\begin{array}{l} r = 0, 1, 2, 3 \rightarrow s = 0 \\ r = 4, 5, 6, 7 \rightarrow s = 7 \end{array}\right]$$

7	0	7	0
0	7	7	7
0	0	7	7
7	7	7	0

Output image



12:46 / 19:06 • Thresholding ›

3) Clipping with $r_1 = 2$ and $r_2 = 5$

$L = 8$

$L - 1 = 7$

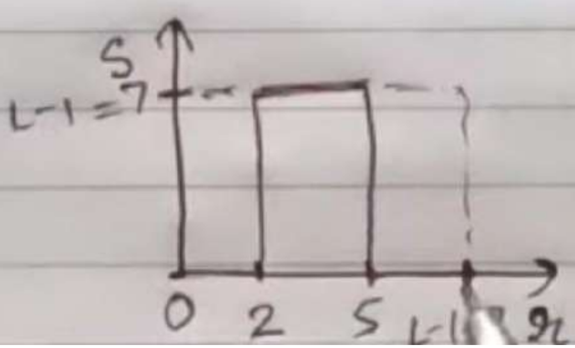
4	3	5	2
3	6	4	6
2	2	6	5
7	6	4	1

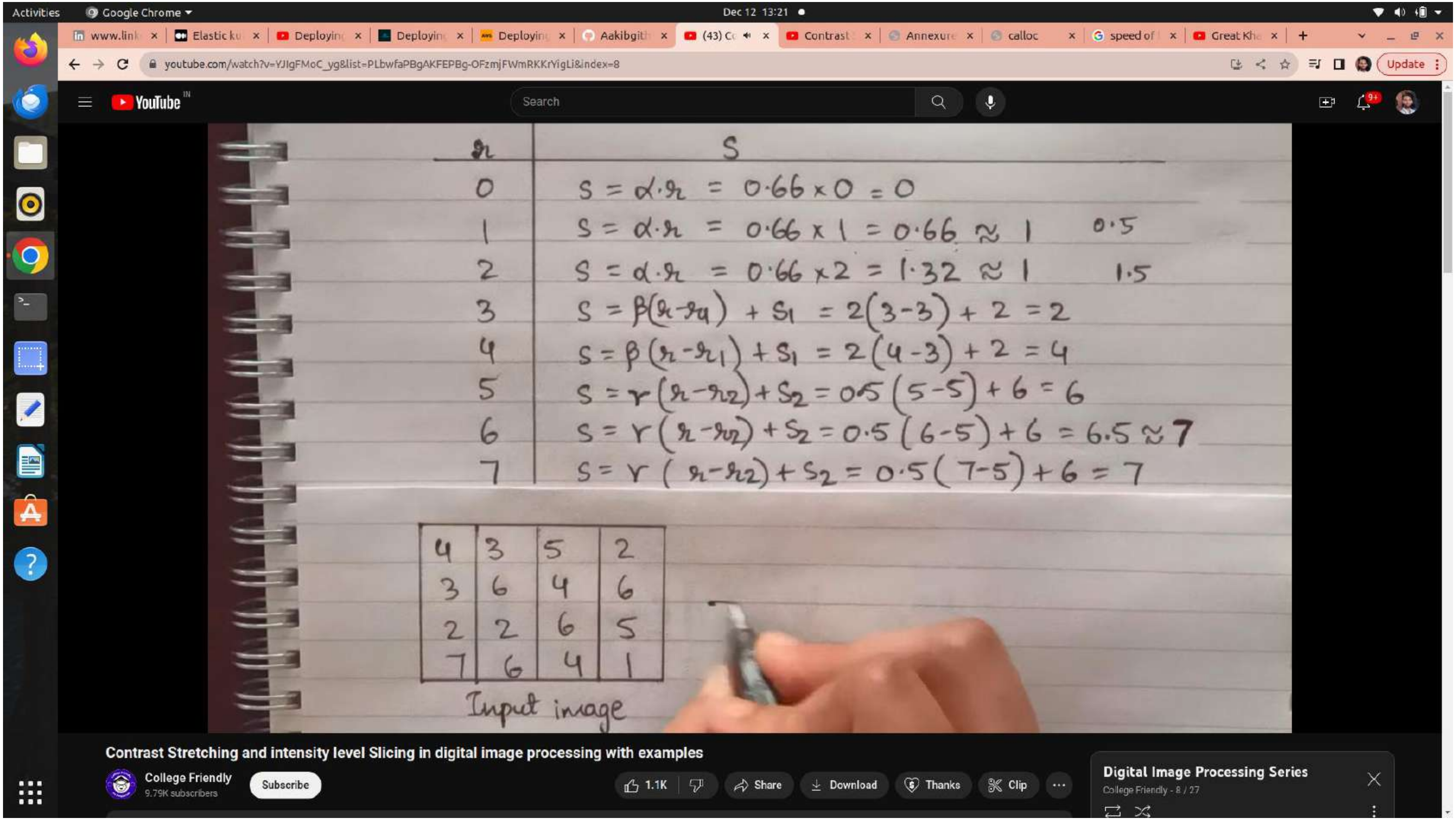
$$s = \begin{cases} L-1=7 \ ; & 2 \le r \le 5 \\ 0 \ ; & \text{otherwise} \end{cases}$$

$$\begin{bmatrix} r = 0,1,6,7 \rightarrow s = 0 \\ r = 2,3,4,5 \rightarrow s = 7 \end{bmatrix}$$

7	7	7	7
7	0	7	0
7	7	0	7
0	0	7	0

Output image





Activities
Google Chrome
Dec 12 13:21
youtube.com/watch?v=YJIgFMoC_yg&list=PLbwfaPBgAKFEPBg-OFzmjFWmRKKrYigLi&index=8
YouTube
Search
r
s
0
s = α·r = 0·66 × 0 = 0
1
s = α·r = 0·66 × 1 = 0·66 ≈ 1
0·5
2
s = α·r = 0·66 × 2 = 1·32 ≈ 1
1·5
3
s = β(r − r1) + s1 = 2(3 − 3) + 2 = 2
4
s = β(r − r1) + s1 = 2(4 − 3) + 2 = 4
5
s = γ(r − r2) + s2 = 0·5(5 − 5) + 6 = 6
6
s = γ(r − r2) + s2 = 0·5(6 − 5) + 6 = 6.5 ≈ 7
7
s = γ(r − r2) + s2 = 0·5(7 − 5) + 6 = 7
4 3 5 2
3 6 4 6
2 2 6 5
7 6 4 1
Input image
Contrast Stretching and intensity level Slicing in digital image processing with examples
College Friendly
9.79K subscribers
Subscribe
1.1K
Share
Download
Thanks
Clip
Digital Image Processing Series
College Friendly - 8 / 27

(ii)

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

$\downarrow$

7	7	7	2
7	6	7	6
2	2	6	7
7	6	7	1

Output image with back

$$s = \begin{cases} L-1 = 7 & ; \ 3 \le r \le 5 \\ r & ; \ \text{otherwise} \end{cases}$$

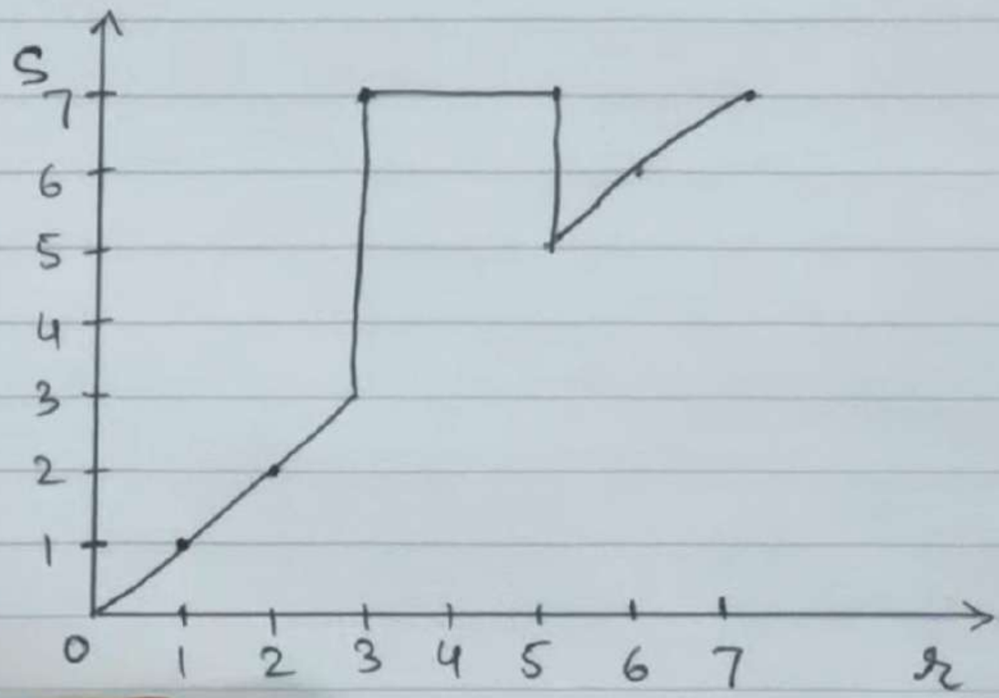


Image Enhancement

The process of manipulating the image so that the result is more suitable than the original for specific applications.

Techniques of Image Enhancement

Spatial domain
Involves direct manipulation of pixels.

Frequency domain
Involves modifying the Fourier transform of an image.

Combination
Involves a combination of spatial and frequency domains.

Image Enhancement in digital image processing with Histogram Equalization

Spatial Domain

1) Histogram Equalization

Q.1. Perform the histogram equalization for an 8×8 image shown below.

Gray levels	0	1	2	3	4	5	6	7
No. of pixels	9	8	11	4	10	15	4	3

1:33 / 12:52

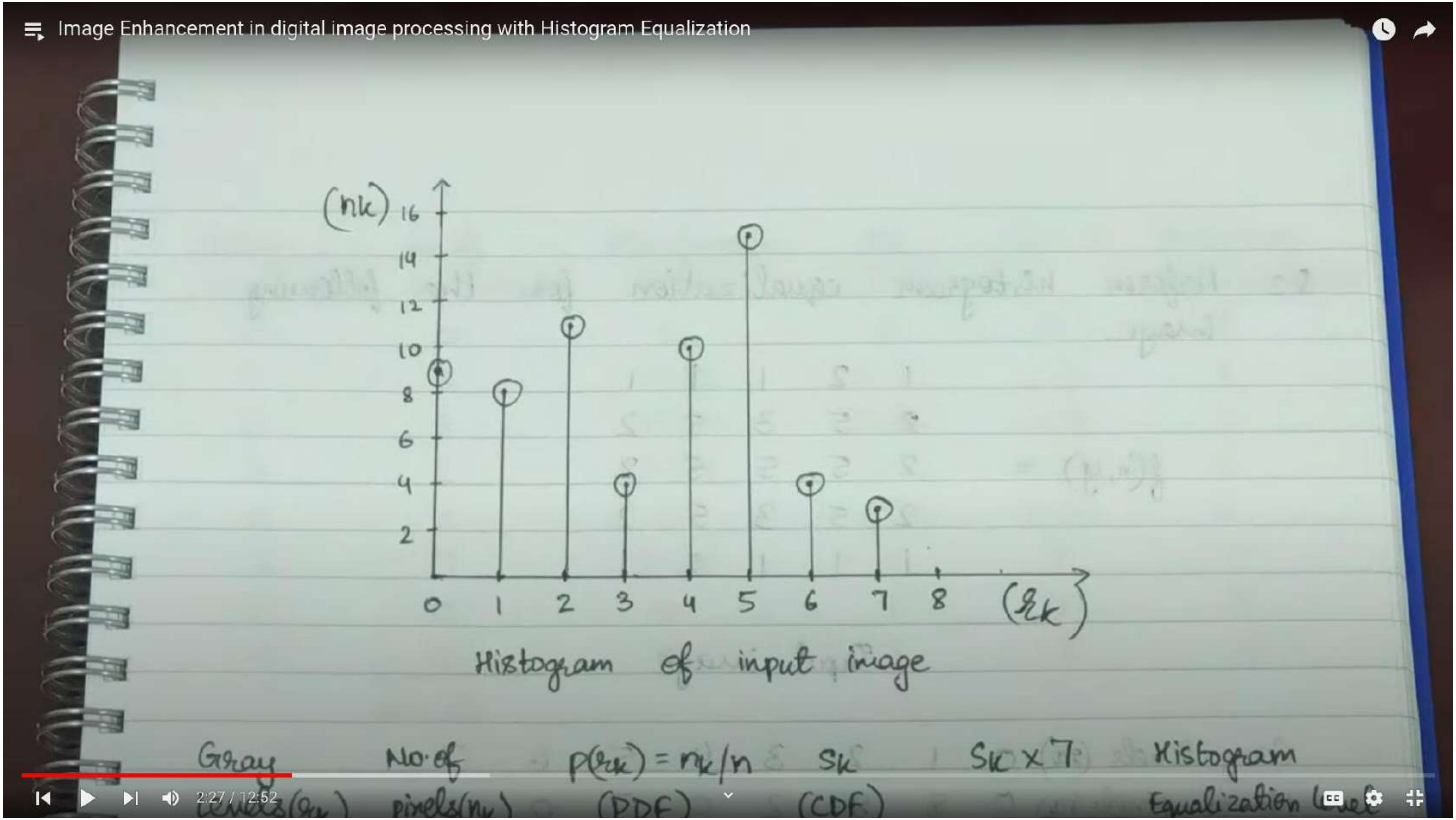
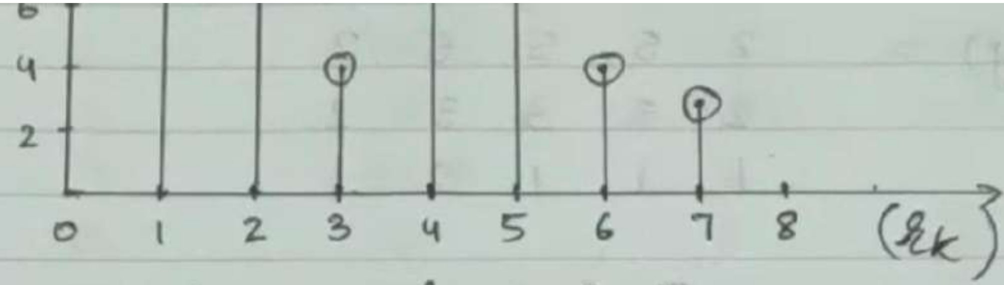


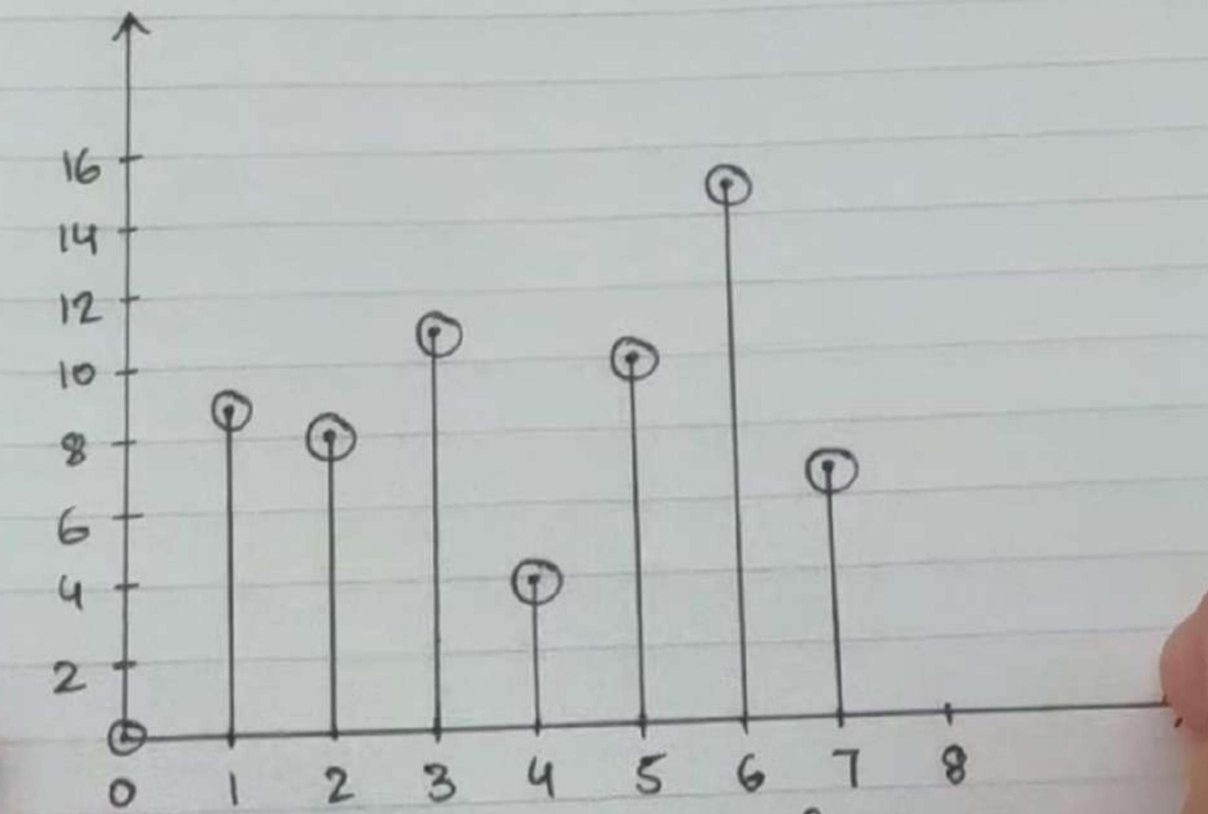
Image Enhancement in digital image processing with Histogram Equalization
(nk)
16
14
12
10
8
6
4
2
0
1
2
3
4
5
6
7
8
(rk)
Histogram of input image
Gray levels(rk)
No. of pixels(nk)
p(rk) = nk/n
(PDF)
Sk
(CDF)
Sk x 7
Histogram
Equalization level
2:27 / 12:52



Histogram of input image

Gray Levels (r_k)	No. of pixels (n_k)	$p(r_k) = n_k/n$ (PDF)	S_k (CDF)	$S_k \times 7$	Histogram Equalization level
0	9	0.141	0.141	0.987	1
1	8	0.125	0.266	1.862	2
2	11	0.172	0.438	3.066	3
3	4	0.0625	0.5005	3.5035	4
4	10	0.156	0.6565	4.5955	5
5	15	0.234	0.8905	6.2335	6
6	4	0.0625	0.953	6.671	7
7	3	0.047	1	7	7
	$n = 64$				

Gray levels	1	2	3	4	5	6	7
No. of pixels	9	8	11	4	10	15	7



Histogram of processed image

Image Enhancement in digital image processing with Histogram Equalization

$[0, L-1]$

Q.2. Perform histogram equalization for the following image.

$$f(x,y) = \begin{bmatrix} 1 & 2 & 1 & 1 & 1 \\ 2 & 5 & 3 & 5 & 2 \\ 2 & 5 & 5 & 5 & 2 \\ 2 & 5 & 3 & 5 & 2 \\ 1 & 1 & 1 & 2 & 1 \end{bmatrix}$$

Input image

Max. value = 5.

$2^0 = 1$

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$ $\quad L = 8$

$L - 1 = 7$

Gray levels (r_k)	0	1	2	3	4	5	6	7
No. of pixels (n_k)	0	8	8	2	0	7	0	0

8:18 / 12:52

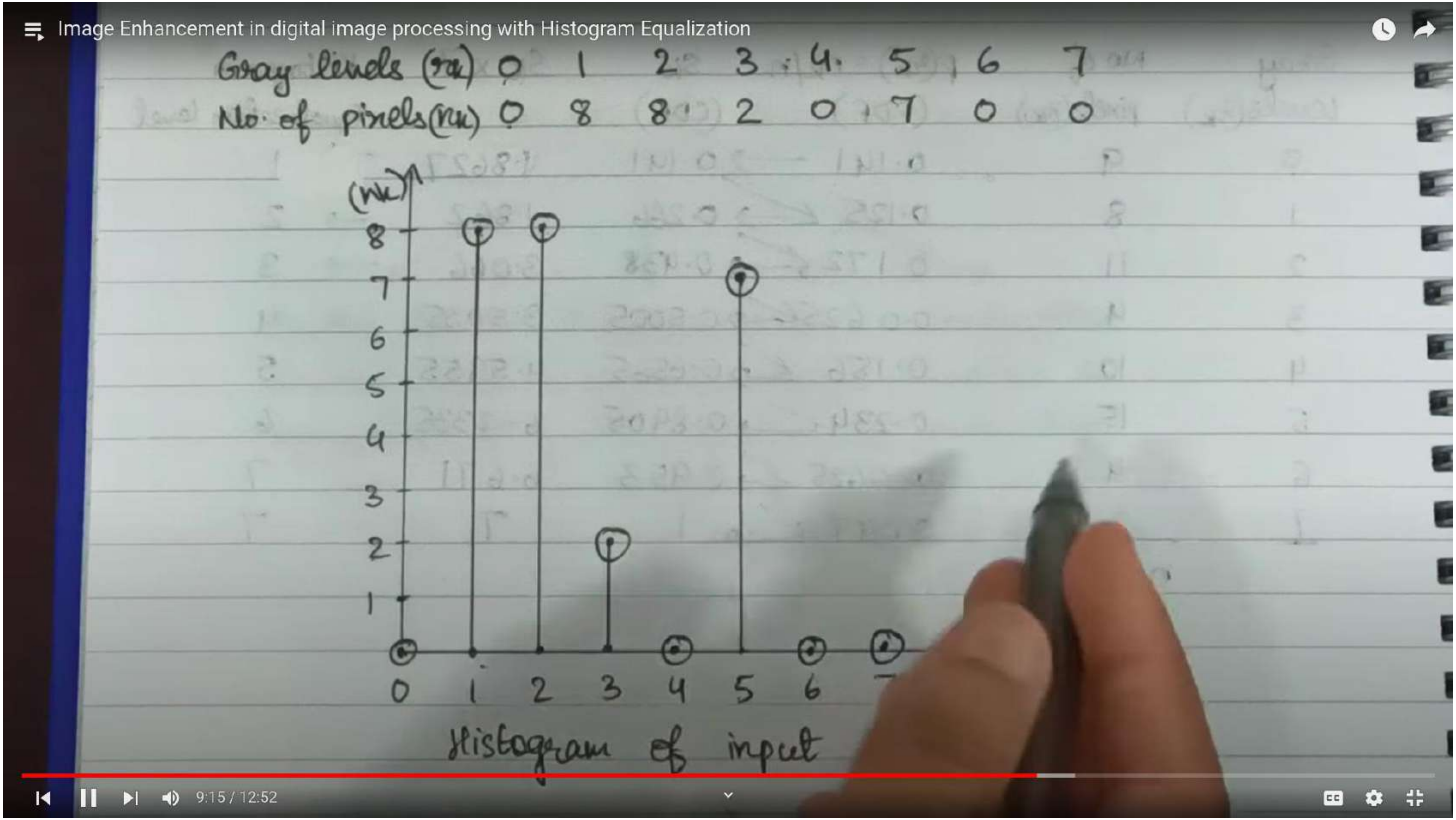
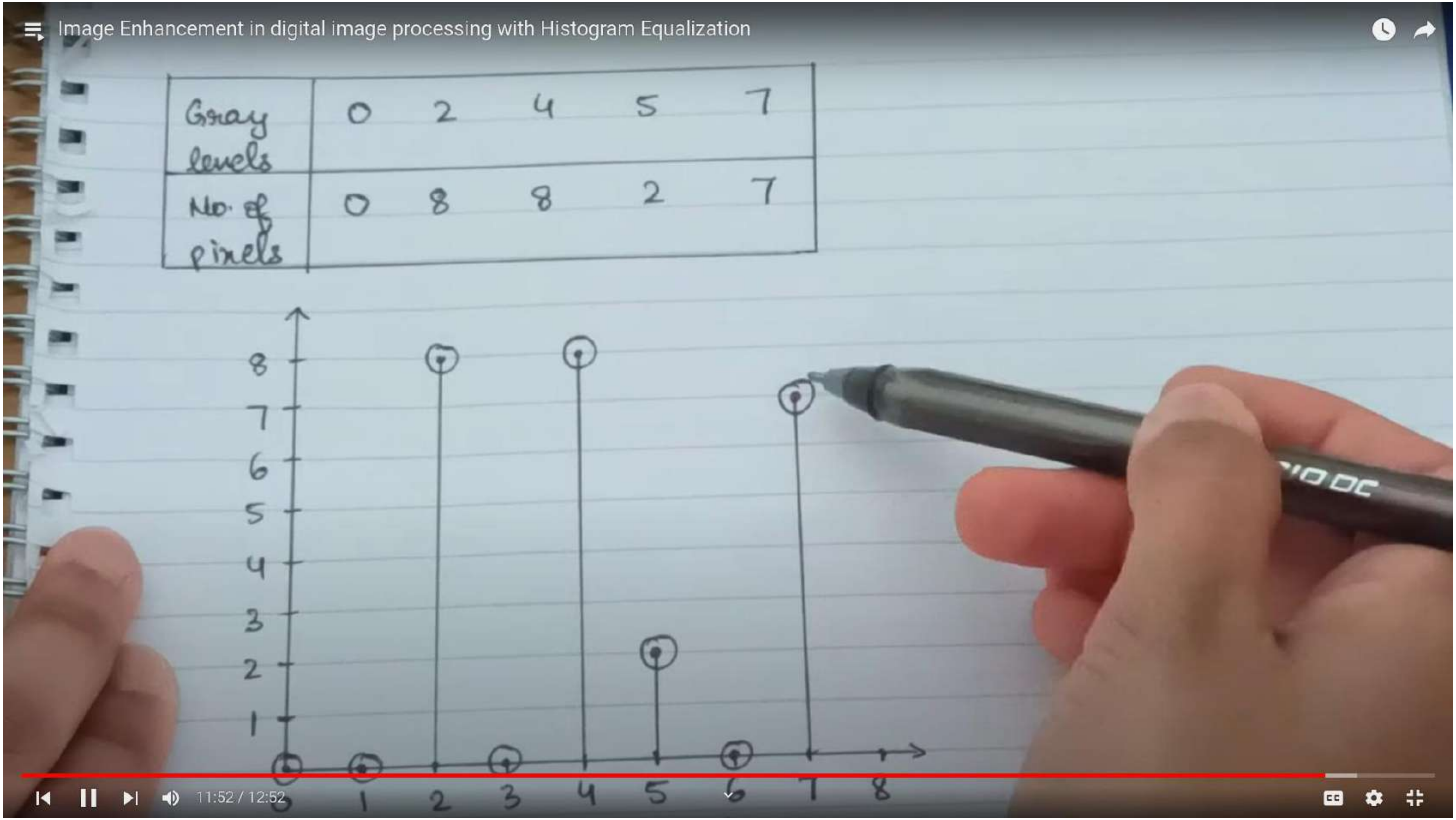


Image Enhancement in digital image processing with Histogram Equalization
Gray levels (rk) 0 1 2 3 4 5 6 7
No. of pixels (nk) 0 8 8 2 0 7 0 0
(nk)
8
7
6
5
4
3
2
1
0 1 2 3 4 5 6
Histogram of input
9:15 / 12:52

Image Enhancement in digital image processing with Histogram Equalization

Gray levels (r_k)	No of pixels (n_k)	$p(r_k) = n_k/n$ (PDF)	S_k (CDF)	$S_k \times 7$	Histogram Equalization level
0	0	0	0	0	0
1	8	0.32	0.32	2.24	2
2	8	0.32	0.64	4.48	4
3	2	0.08	0.72	5.04	5
4	0	0	0.72	5.04	5
5	7	0.28	1	7	7
6	0	0	1	7	7
7	0	0	1	7	7
	$n = 25$				

10:12 / 12:52



Gray levels	0	2	4	5	7
No. of pixels	0	8	8	2	7

Image Enhancement in digital image processing with Histogram Equalization

0	0	0	0	0	0
1	8	0.32	0.32	2.24	2
2	8	0.32	0.64	4.48	4
3	2	0.08	0.72	5.04	5
4	0	0	0.72	5.04	5
5	7	0.28	1	7	7
6	0	0	1	7	7
7	0	0	1	7	7

1	2	1	1	1
2	5	3	5	2
2	5	5	5	2
2	5	3	5	2
1	1	1	2	1

→

2	4	2	2	2
4	7	5	7	4
4	7	7	7	4
4	7	5	7	4
2	2	2	4	2

Input Image

12:39 / 12:52

2) Histogram Matching (Specification)

Q1. Given below are two histograms (i) and (ii). Modify the histogram (i) as given by histogram (ii).

(i)

Gray level (r_k)	0	1	2	3	4	5	6	7
No. of pixels (n_k)	80	100	90	60	30	20	10	0

(ii)

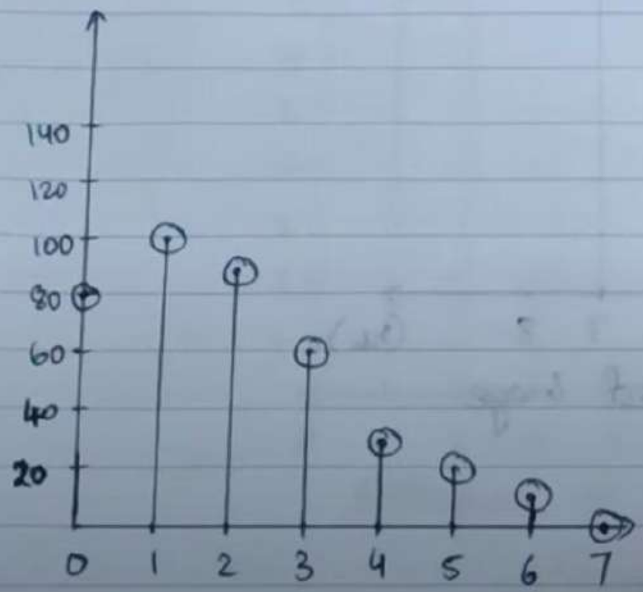
Gray level (r_k)	0	1	2	3	4	5	6	7
No. of pixels (n_k)	0	0	0	60	80	100	80	70

(i)

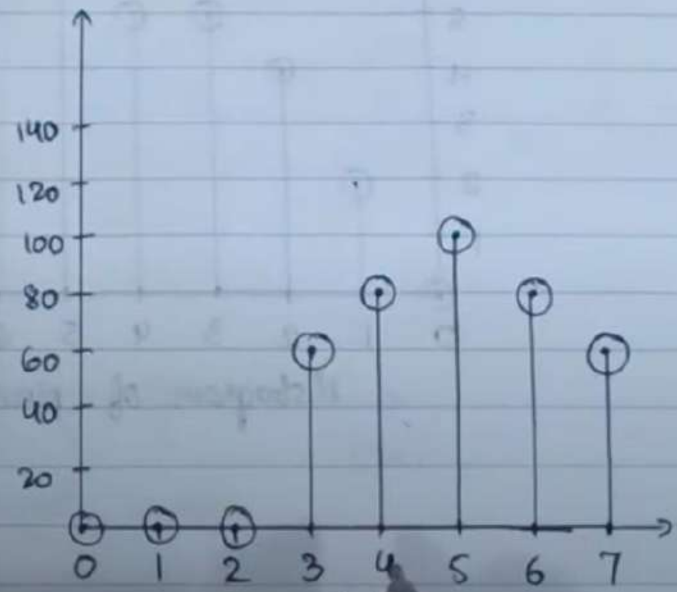
Gray level (r_k)	0	1	2	3	4	5	6	7
No. of pixels (n_k)	80	100	90	60	30	20	10	0

(ii)

Gray level (r_k)	0	1	2	3	4	5	6	7
No. of pixels (n_k)	0	0	0	60	80	100	80	70



(i)



(ii)

(1) Equalize histogram(i)

Gray level	n_k	$p(r_k) = n_k/n$ (PDF)	S_k (CDF)	$S_k \times 7$	Histogram equalization	New
0	80	0.20	0.20	1.4	1	80
1	100	0.25	0.45	3.15	3	100
2	90	0.23	0.68	4.76	5	90
3	60	0.15	0.83	5.81	6	60+30 = 90
4	30	0.07	0.9	6.3	6	
5	20	0.05	0.95	6.65	7	
6	10	0.02	0.97	6.79	7	20+10 = 30
7	0	0	0.97	6.79	7	
	$n = 390$					

Histogram matching in digital image processing

(1) Equalize histogram(i)

Gray level	n_k	$p(r_k) = n_k/n$ (PDF)	S_k (CDF)	$S_k \times 7$	Histogram equalization	New n_k
0	80	0.20	0.20	1.4	1	80
1	100	0.25	0.45	3.15	3	100
2	90	0.23	0.68	4.76	5	90
3	60	0.15	0.83	5.81	6	60+30 = 90
4	30	0.07	0.9	6:3	6	
5	20	0.05	0.95	6.65	7	20+10 = 30
6	10	0.02	0.97	6.79	7	
7	0	0	0.97	6.79	7	

$n = 390$

3:29 / 8:18

② Equalize histogram (ii)

Gray level	nk	p(rk) = nk/n (PDF)	Sk (CDF)	Sk × 7	Histogram equalization level
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	60	0.15	0.15	1.05	1
4	80	0.20	0.35	2.45	2
5	100	0.25	0.6	4.2	4
6	80	0.20	0.8	5.6	6
7	70	0.17	0.97	6.79	7
	n = 390				

Histogram matching in digital image processing

③ Mapping

→ Take the first and last columns of histogram (ii).

→ Take the last 2 columns of histogram (i).

(ii)

Gray level	Histogram equalization level
0	0
1	0
2	0
3	1
4	2
5	4
6	6
7	7

(i)

Histogram equalization level	New n_k
1	80
3	100
5	90
6	90
6	
7	
7	30
7	

5:13 / 8:18

Histogram matching in digital image processing

(ii)

Gray level	Histogram equalization level
0	0
1	0
2	0
3	1
4	2
5	4
6	6
7	7

(i)

Histogram equalization level	New n_k
1	80
3	100
5	90
6	90
6	
7	
7	30
7	

Gray level	0	1	2	3	4	5	6	7
No. of pixels				80	80	100	90	30

7:29 / 8:18

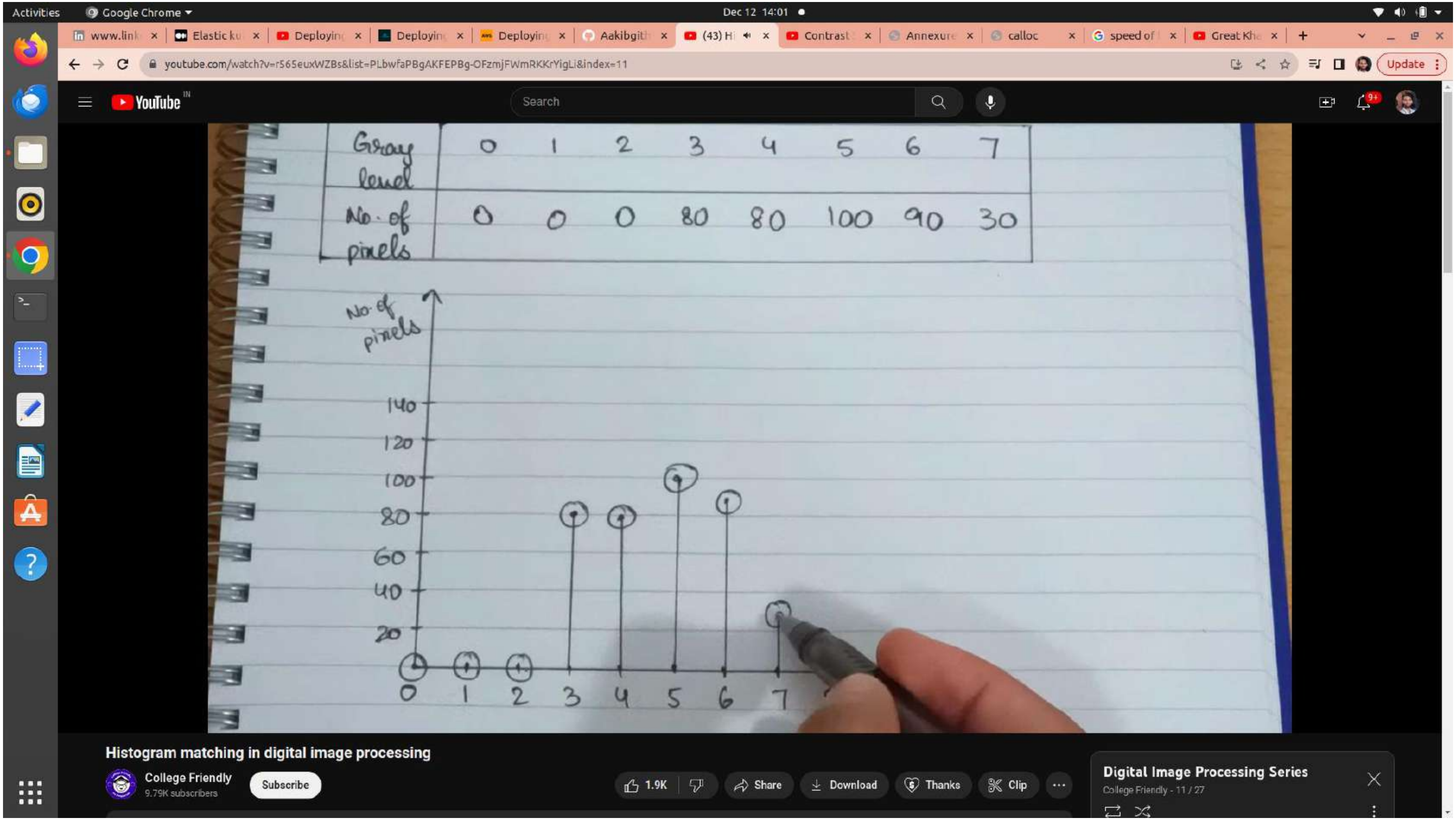
(ii)

Gray level	Histogram equalization level
0	0
1	0
2	0
3	1
4	2
5	4
6	6
7	7

(i)

Histogram equalization level	New nk
1	80
3	100
5	90
6	90
6	
7	
7	30
7	

Gray level	0	1	2	3	4	5	6	7
No. of pixels	0	[illegible]	[illegible]	[illegible]	80	100	90	30



Activities
Google Chrome
Dec 12 14:01
youtube.com/watch?v=r565euxWZBs&list=PLbwfaPBgAKFEPBg-OFzmjFWmRKKrYigLi&index=11
YouTube
Search
Update
Gray level
0 1 2 3 4 5 6 7
No. of pixels
0 0 0 80 80 100 90 30
No. of pixels
140
120
100
80
60
40
20
0 1 2 3 4 5 6 7
Histogram matching in digital image processing
College Friendly
9.79K subscribers
Subscribe
1.9K
Share
Download
Thanks
Clip
Digital Image Processing Series
College Friendly - 11 / 27

Fundamentals of Spatial Filtering

The name 'filter' is borrowed from frequency domain processing. It basically refers to accepting (passing) or rejecting certain frequency components.

Ex. A filter that passes low frequencies is called lowpass filter.

We can accomplish a similar smoothing directly on the image itself using spatial filters (also called masks, kernels, templates and windows).

1) Convolution

Q1. Let $I = \{0,0,1,0,0\}$ be an image. Using the mask $K = \{3,2,8\}$, perform the convolution.

$I = \{0,0,1,0,0\}$ $\quad\quad$ $K = \{3,2,8\}$

(i) zero padding process for convolution

In convolution process, we have to rotate the kernel by 180°.

$$\begin{array}{ccccccccc} 8 & 2 & 3 & & & & 8 & 2 & 3 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}$$

Fundamentals of Spatial Filtering in digital image processing

(ii) Initial position

Template

8 2 3

0 0 0 0 1 0 0 0 0

0

$(8\times0) + (2\times0) + (3\times0)$

$= 0$

Output is 0 located at the center pixel.

(iii) Position after one shift

Template is shifted by one bit.

8 2 3

0 0 0 0 1 0 0 0 0

0 0

Output is 0.

4:35 / 17:32 • Initial Position ›

(iv) Position after 2 shifts

Template is shifted again.

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

$(8\times 0) + (2\times 0) + (3\times 1)$
$= 3$

Output produced is 3.

(v) Position after 3 shifts

Template is shifted again.

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

$(8\times 0) + (2\times 1) + (3\times 0)$
$= 2$

Output produced is 2.

(vi) Position after 4 shifts

Template is shifted again.

8 2 3

0 0 0 0 1 0 0 0 0

0 0 3 2 8

$(8\times1) + (2\times0) + (3\times0)$
$= 8$

Output produced is 8

(vii) Position after 5 shifts

Template is shifted again

8 2 3

0 0 0 0 1 0 0 0 0

0 0 3 2 8 0

$(8\times0) + (2\times0) + (3\times0)$
$= 0$

Output produced is 0.

2) Correlation

Q2. Let $I = \{0, 0, 1, 0, 0\}$ be an image. Using the mask $K = \{3, 2, 8\}$, perform the correlation.

$I = \{0, 0, 1, 0, 0\}$ $K = \{3, 2, 8\}$

(i) Zero padding process for correlation

$$
\begin{array}{ccccccccc}
3 & 2 & 8 & & & & 3 & 2 & 8 \\
\underline{0} & \underline{0} & 0 & 0 & 1 & 0 & 0 & \underline{0} & \underline{0}
\end{array}
$$

(ii) Initial position

Template

$$
\begin{array}{ccccccccc}
3 & 2 & 8 & & & & & & \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 & 0 & & & & & & &
\end{array}
$$

Output produced is 0.

Output produced is 0.

(iii) Position after one shift
Template
3 2 8
0 0 0 0 1 0 0 0 0
0 0
Output produced is 0.

(iv) Position after 2 shifts
Template
3 2 8
0 0 0 0 1 0 0 0 0
0 0 8
Output produced is 8.

(v) Position after 3 shifts
Template

3 2 8

0 0 0 0 1 0 0 0 0

0 0 8 2

Output produced is 2.

(vi) Position after 4 shifts
Template

3 2 8

0 0 0 0 1 0 0 0 0

0 0 8 2 3

Output produced is 3.

(vii) Position after 5 shifts
Template

3 2 8

0 0 0 0 1 0 0 0 0

0 0 8 2 3 0

Output produced is 0.

(viii) Final position
Template

3 2 8

0 0 0 0 1 0 0 0 0

0 0 8 2 3 0 0

Output produced is 0. Further shifting exceeds the range.

So in the final position, the output produced is {0,0, 8,2,3,0,0}.

Q3. Let $I = \begin{pmatrix} 3 & 3 \\ 3 & 3 \end{pmatrix}$ be an image and

$K = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ be a kernel (mask). Perform

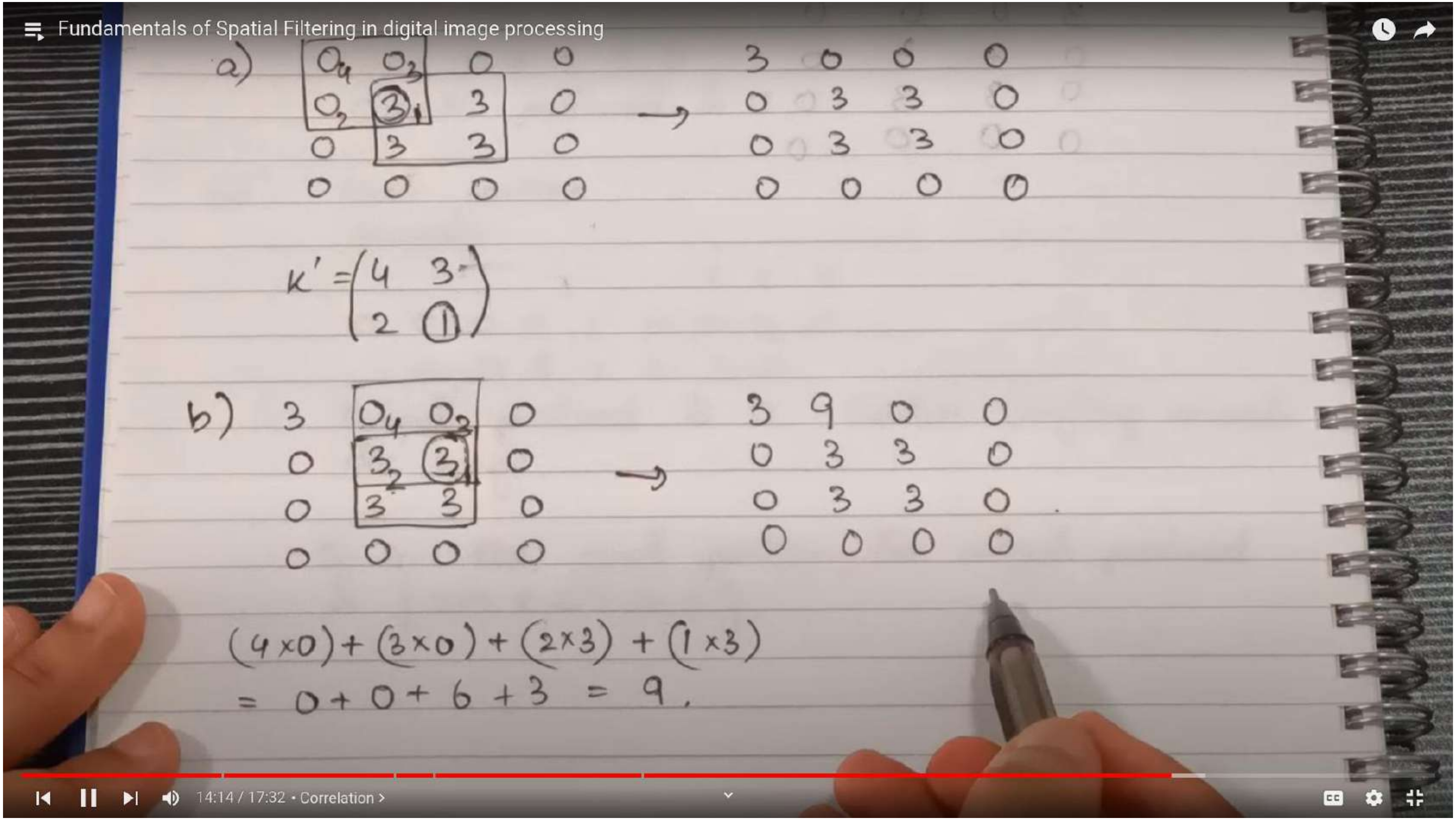
convolution and correlation.

i) Convolution

Rotate the kernel by 180°.

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \quad\downarrow\quad \begin{matrix} 3 & 4 \\ 1 & 2 \end{matrix} \quad\downarrow\quad \begin{matrix} 4 & 3 \\ 2 & 1 \end{matrix}$$

$K' = \begin{pmatrix} 4 & 3 \\ 2 & 1 \end{pmatrix}$.


Fundamentals of Spatial Filtering in digital image processing
a)
0 0 0 0
0 3 3 0
0 3 3 0
0 0 0 0
3 0 0 0
0 3 3 0
0 3 3 0
0 0 0 0
K' = (4 3 / 2 1)
b)
3 0 0 0
0 3 3 0
0 3 3 0
0 0 0 0
3 9 0 0
0 3 3 0
0 3 3 0
0 0 0 0
(4×0) + (3×0) + (2×3) + (1×3)
= 0 + 0 + 6 + 3 = 9.
14:14 / 17:32 • Correlation >

Fundamentals of Spatial Filtering in digital image processing

c)

$$\begin{matrix} 3 & 9 & 0_4 & 0_3 \\ 0 & 3 & 3_2 & 0_1 \\ 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{matrix} \longrightarrow \begin{matrix} 3 & 9 & 6 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$$

$$(4\times 0) + (3\times 0) + (2\times 3) + (1\times 0)$$

$$= 6.$$

15:29 / 17:32 • Correlation ›

d)

$$\begin{array}{cccc} 3 & 9 & 6 & 0 \\ 0_4 & 3_3 & 3 & 0 \\ 0_2 & 3_1 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{array} \longrightarrow \begin{array}{cccc} 3 & 9 & 6 & 0 \\ 12 & 3 & 3 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{array}$$

e)

$$\begin{array}{cccc} 3 & 9 & 6 & 0 \\ 12 & 3_4 & 3_3 & 0 \\ 0 & 3_2 & 3_1 & 0 \\ 0 & 0 & 0 & 0 \end{array} \longrightarrow \begin{array}{cccc} 3 & 9 & 6 & 0 \\ 12 & 30 & 3 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{array}$$

$$(4 \times 3) + (3 \times 3) + (2 \times 3) + (1 \times 3)$$
$$= 12 + 9 + 6 + 3$$
$$= 30.$$

f)

$$\begin{array}{cccc} 3 & 9 & 6 & 0 \\ 12 & 30_4 & 3_3 & 0 \\ 0 & 3_2 & 3_1 & 0 \\ 0 & 0 & 0 & 0 \end{array} \longrightarrow \begin{array}{cccc} 3 & 9 & 6 & 0 \\ 12 & 30 & 138 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{array}$$

$$120 + 9 + 6 + 3 = 138.$$

g) $$\begin{matrix} 3 & 9 & 6 & 0 \\ 12 & 30 & 138 & 0 \\ 0_4 & 3_3 & 3 & 0 \\ 0_2 & 0_1 & 0 & 0 \end{matrix} \longrightarrow \begin{matrix} 3 & 9 & 6 & 0 \\ 12 & 30 & 138 & 0 \\ 9 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$$

h) $$\begin{matrix} 3 & 9 & 6 & 0 \\ 12 & 30 & 138 & 0 \\ 9 & 3_4 & 3_3 & 0 \\ 0 & 0_2 & 0 & 0 \end{matrix} \longrightarrow \begin{matrix} 3 & 9 & 6 & 0 \\ 12 & 30 & 138 & 0 \\ 9 & 21 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$$

i) $$\begin{matrix} 3 & 9 & 6 & 0 \\ 12 & 30 & 138 & 0 \\ 9 & 21 & 3_4 & 0_3 \\ 0 & 0 & 0_2 & 0_1 \end{matrix} \longrightarrow \boxed{\begin{matrix} 3 & 9 & 6 & 0 \\ 12 & 30 & 138 & 0 \\ 9 & 21 & 12 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}}$$

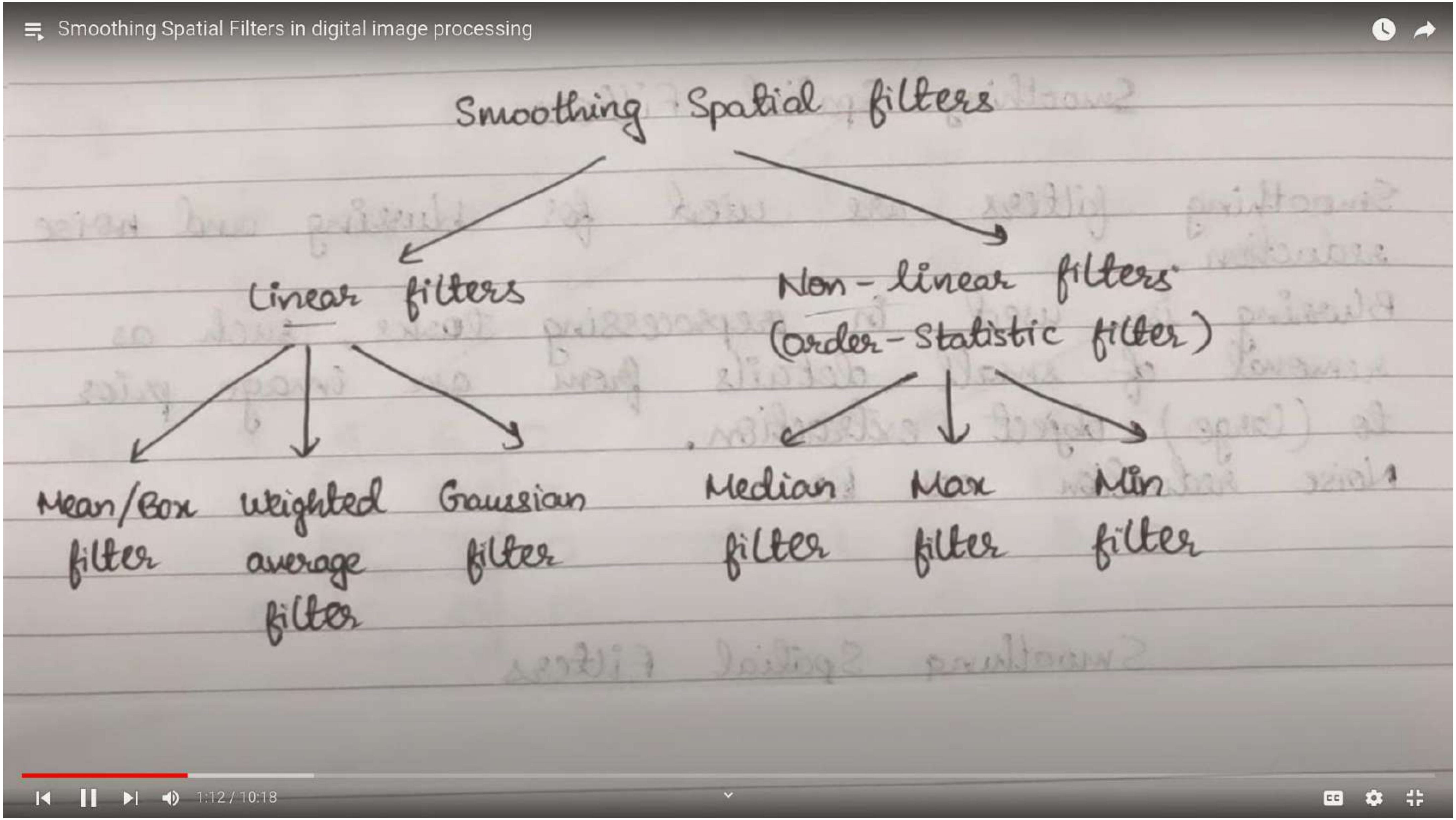
For correlation, perform the same steps without rotating the kernel by 180°.

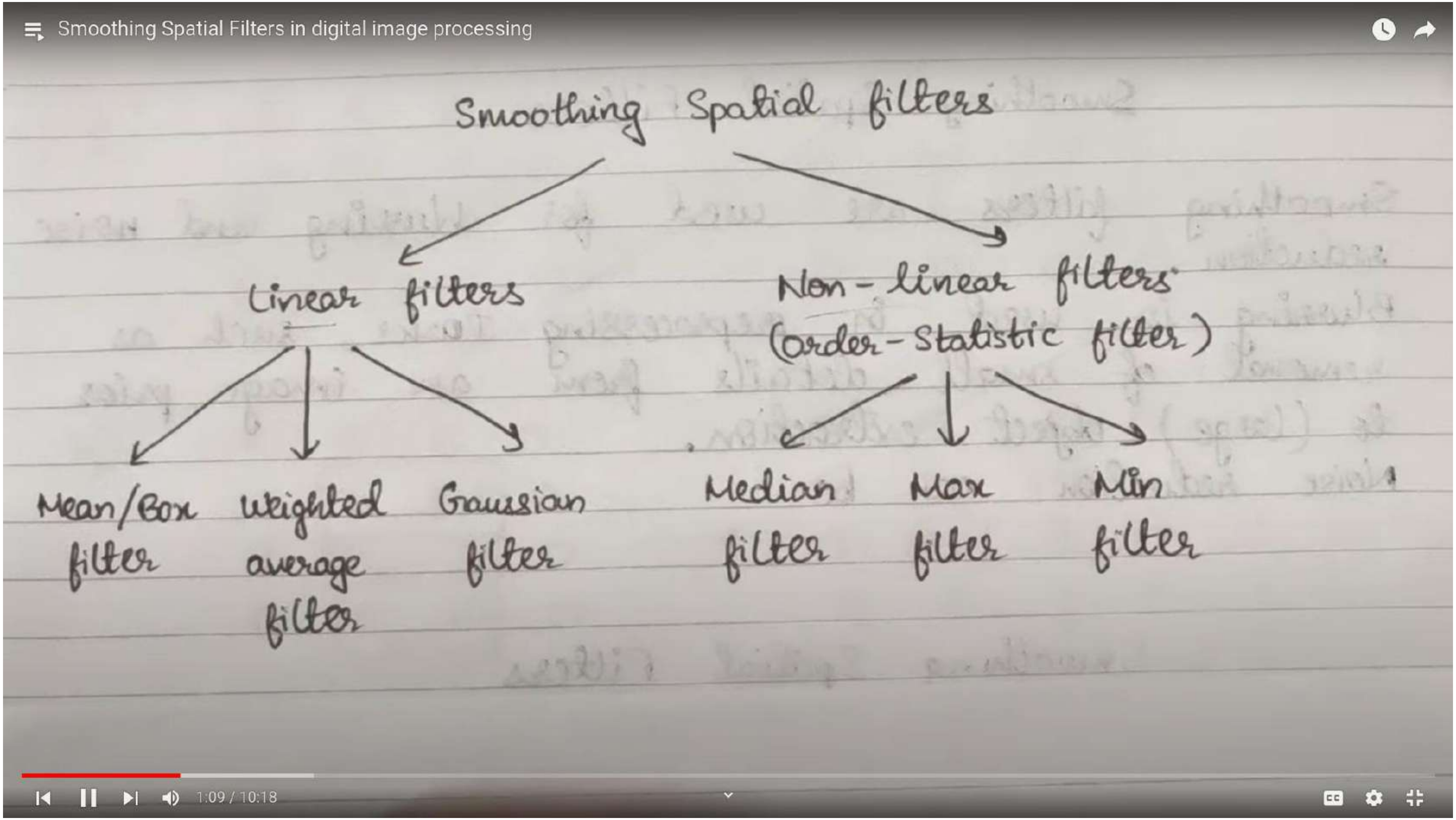
Smoothing Spatial Filters

→ Smoothing filters are used for blurring and noise reduction.

→ Blurring is used in preprocessing 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.


Smoothing Spatial Filters in digital image processing
Smoothing Spatial filters
Linear filters
Non-linear filters
(order-Statistic filter)
Mean/Box filter
weighted average filter
Gaussian filter
Median filter
Max filter
Min filter
1:12 / 10:18


Smoothing Spatial Filters in digital image processing
Smoothing Spatial filters
Linear filters
Non-linear filters
(Order-Statistic filter)
Mean/Box filter
weighted average filter
Gaussian filter
Median filter
Max filter
Min filter
1:09 / 10:18

* Smoothing linear filters

They are also known as averaging filters (or) lowpass filters as they are simply the average of the pixels contained in the neighbourhood of the filter mask.

The process results in an image with reduced 'sharp' transitions in intensities which ultimately leads to noise reduction.

1) Box filter - all coefficients are equal.

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

3x3

2) weighted average - give more (less) weight to pixels near (away from) the output location.

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ [illegible] & 4 & 2 \\ [illegible] & 2 & 1 \end{bmatrix} \longrightarrow \text{Mask}$$

3) Gaussian filter - the weights are samples of 2D Gaussian function:

$$\boxed{G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}}$$

(2D Gaussian function)

$\sigma \rightarrow$ standard deviation

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_{3\times 3} \rightarrow \text{Mask}$$

→ Used to blur edges and reduce contrast.

→ Similar to median filter but is faster.

* Non-linear (Order-Statistic) filters

Their response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result.

1) Median filter - Find the median of all the pixel values.
2) Min filter - Find the minimum of all the pixel values.
3) Max filter - Find the maximum of all the pixel values.

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

a) Box / Mean filter
b) Weighted average filter
c) Median filter
d) Min filter
e) 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

3x3

a) Box filter

$$= \frac{1}{9} \times \left[7+9+5+4+6+8+2+0+1 \right] \quad \frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

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

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

a) Box / Mean filter
b) Weighted average filter
c) Median filter
d) Min filter
e) Max filter

$$\begin{matrix} 1 & 8 & 8 & 0 & 7 \\ 4 & 7 & 9 & 5 & 7 \\ 5 & 4 & 6 & 8 & 6 \\ 4 & 2 & 0 & 1 & 5 \\ 0 & 1 & 0 & 2 & 0 \end{matrix}$$

3x3

Input image

b) Weighted average filter

$$= \frac{1}{16}\left[7\times1 + 9\times2 + 5\times1 + 4\times2 + 4\times6 + 8\times2 + 2\times1 + 0\times2 + 1\times1\right]$$

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

$$= \frac{1}{16}\left[81\right] = 5.0625 \approx \underline{\underline{5}}.$$

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

a) Box / Mean filter
b) Weighted average filter
c) Median filter
d) Min filter
e) Max filter

$$
\begin{array}{ccccc}
1 & 8 & 8 & 0 & 7 \\
4 & 7 & 9 & 5 & 7 \\
5 & 4 & 6 & 8 & 6 \\
4 & 2 & 0 & 1 & 5 \\
0 & 1 & 0 & 2 & 0
\end{array}
$$

3x3

c) Median filter

$\cancel{0}, \cancel{1}, \cancel{2}, \cancel{4}, 5, \cancel{6}, \cancel{7}, \cancel{8}, \cancel{9}$

Median = 5.

d) Min filter

= 0.

e) Max filter

= 9.

Few Important Questions

Q1. Why median filter is better than mean filter?

Ans. Median filter is normally used to reduce noise in an image, similar to the mean filter. However, it often does a better job than mean filter in preserving useful detail in an image.

Median filter has 2 main advantages:

1) The median is a more robust average than the mean and so a single very unrepresentative pixel in a neighborhood will not affect the median value significantly.

2) Since the median value must actually be the value of one of the pixels in the neighborhood, the median filter does not create new unrealistic pixel values when the filter straddles an edge. Therefore, it is much better at preserving sharp edges than the mean filter.

Note:

- Mean filter is better at dealing with Gaussian noise than median filter.
- Median filter is better at dealing with salt and pepper noise than mean filter.

Few Important Theory Questions

Q2. Write down a few approaches to deal with missing edge pixels.

Ans. A few approaches to dealing with missing edge pixels are:

1) Omit missing pixels
   - Only works with some filters.
   - Can add extra code and slow down processing.

2) Pad the image
   - Typically with either all white or all black pixels.

Few Important Theory Questions

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

Input Image

1	2	1
2	4	2
1	2	1

3:40 / 7:34

Q2. Write down a few approaches to deal with

0	0	0				
0	50	50	50	100	100	100
0	50	50	50	100	100	100
	50	50	50	100	100	100
	50	50	50	100	100	100
	50	50	50	100	100	100
	50	50	50	100	100	100

Input Image

1	2	1

0	0	0		100	100	100	100
0	50	50	50	100	100	100	100
0	50	50	50	100	100	100	
50	50	50	50	100	100	100	
50	50	50	50	100	100	100	
	50	50	50	100	100	100	
	50	50	50	100	100	100	

Input Image

1	2	1
2	4	2
1	2	1

Mask

* Wrapping of pixels

$$
\begin{array}{c|ccc|c}
9 & 7 & 8 & 9 & 7 \\
\hline
3 & 1 & 2 & 3 & 1 \\
6 & 4 & 5 & 6 & 4 \\
9 & 7 & 8 & 9 & 7 \\
\hline
3 & 1 & 2 & 3 & 1
\end{array}
\qquad\qquad
\begin{array}{c|ccc|}
 & 0 & 1 & 6 \\
\hline
4 & 9 & 8 & 4 \\
5 & 3 & 2 & 5 \\
6 & 0 & 1 & 6 \\
\hline
 & 9 & 8 & 4
\end{array}
$$

Sharpening Spatial Filters

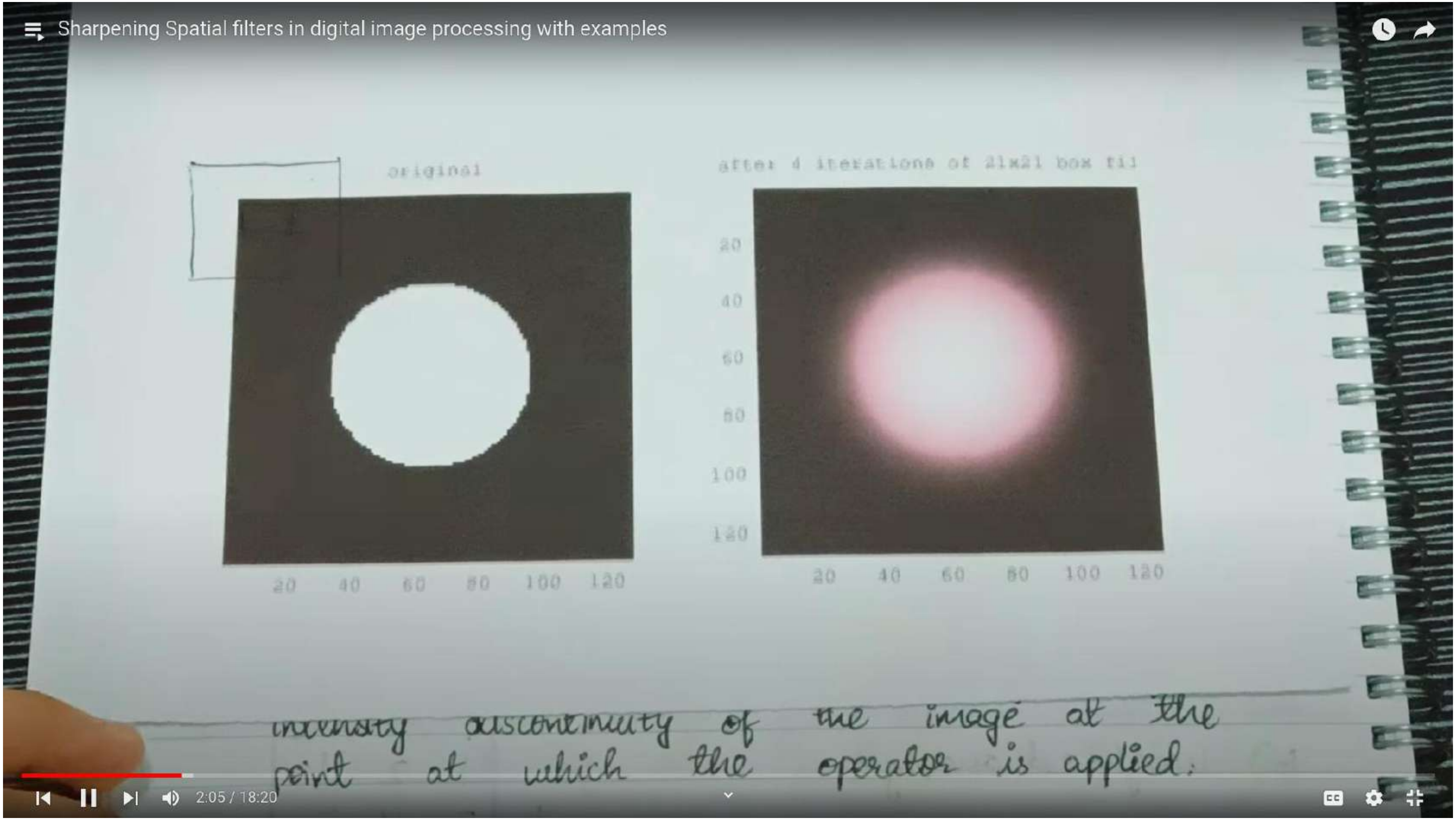
→ The principal objective of sharpening is to highlight transitions in intensity.

→ Applications of image sharpening include electronic printing, medical imaging, industrial inspection and autonomous guidance in military systems.

Blurring → Pixel averaging

Sharpening → Spatial diffentiation

→ The strength of the response of a derivative operator is proportional to the degree of intensity discontinuity of the image at the point at which the operator is applied.


Sharpening Spatial filters in digital image processing with examples
original
after 4 iterations of 21x21 box fil
20 40 60 80 100 120
20 40 60 80 100 120
intensity discontinuity of the image at the
point at which the operator is applied.
2:05 / 18:20

and autonomous guidance in military systems.

Blurring → Pixel averaging

Sharpening → Spatial diffentiation

→ The strength of the response of a derivative operator is proportional to the degree of intensity discontinuity of the image at the point at which the operator is applied.

→ Therefore, image differentiation enhances edges and other discontinuities (such as noise) and deemphasizes areas which have slowly varying intensities.

Sharpening Spatial filters in digital image processing with examples

Foundation of sharpening filters

1) First-order derivative of a one-dimensional function $f(x)$:

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

2) Second-order derivative of a one-dimensional function $f(x)$:

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

5:12 / 18:20

Laplacian filter

→ It highlights gray-level discontinuities in an image.

→ It deemphasizes regions with slowly varying gray levels.

$f(x)$

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

$f(x,y)$

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

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

Laplacian filter

→ It highlights gray-level discontinuities in an image.

→ It deemphasizes regions with slowly varying gray levels.

$f(x)$

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

$f(x, y)$

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

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

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

* Laplacian mask

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

$f(x-1, y-1)$	$f(x, y-1)$	$f(x+1, y-1)$
$f(x-1, y)$	$f(x, y)$	$f(x+1, y)$
$f(x-1, y+1)$	$f(x, y+1)$	$f(x+1, y+1)$

Input image

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

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

0	1	0
1	-4	1
0	1	0

Sharpening Spatial filters in digital image processing with examples

Q1. Apply Laplacian filter on the given image on the center pixel.

$$\begin{bmatrix} 8 & 5 & 4 \\ 0 & 6 & 2 \\ 1 & 3 & 7 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Input image — Mask

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

$$= 0 + 5 + 0 + 0 - 24 + 2 + 0 + 3 + 0$$

$$= 10 - 24 = -14.$$

10:20 / 18:20

Enhanced Laplacian Filter

$$\begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix} \xrightarrow{\text{Enhanced}} \begin{matrix} 0 & 1 & 0 \\ 1 & -5 & 1 \\ 0 & 1 & 0 \end{matrix}$$

$$\begin{matrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{matrix} \xrightarrow{\text{Enhanced}} \begin{matrix} 1 & 1 & 1 \\ 1 & -9 & 1 \\ 1 & 1 & 1 \end{matrix}$$

Q2. Apply enhanced laplacian filter on the given image on the center pixel.

$$\begin{bmatrix} 8 & 5 & 4 \\ 0 & 6 & 2 \\ 1 & 3 & 7 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & -9 & 1 \\ 1 & 1 & 1 \end{bmatrix} = 8 + 5 + 4 + 0 - 54 + 2 + 1 + 3 + 7$$

$$= 30 - 54 = -24.$$

Sharpening Spatial filters in digital image processing with examples

Q3. Apply Laplacian filter on the given image.

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	[illegible]	100	100	50	50	50	50

1	1	1
1	-8	1
1	1	1

Mask

$50 \times 8 - 50 \times 8$

$= 0$

$50 \times 5 + 100 \times 3 - 50 \times 8$

$= 450 + 300 - 400$

$= 150.$

$50 \times 3 + 100 \times 5 - 400$

$= 150 + 500 - 400$

$= -150.$

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

18:11 / 18:20

Unsharp Masking and Highboost Filtering

→ Primarily used in the printing and publishing industry to sharpen images.

→ The process involves subtracting an unsharp (smoothed) version of an image from the original image. This process called unsharp masking consists of the following steps:

1) Blur the original image.
2) Subtract the blurred image from the original (the resulting difference is called the mask)
3) Add the mask to the original.

Unsharp masking

$$f_s(x,y) = f(x,y) - \bar{f}(x,y)$$

sharpened image = original image − blurred image

Subtracting a blurred version of an image from the original produces a sharpened image.

Highboost filtering

$$f_{hb}(x,y) = A f(x,y) - \bar{f}(x,y)$$
$$= A f(x,y) - [f(x,y) - f_s(x,y)]$$

$$f_{hb}(x,y) = (A-1) f(x,y) - f_s(x,y).$$

Highboost filtering

$$f_{hb}(x,y) = A f(x,y) - \bar{f}(x,y)$$
$$= A f(x,y) - [f(x,y) - f_s(x,y)]$$

$$\boxed{f_{hb}(x,y) = (A-1) f(x,y) - f_s(x,y)}$$

This is the generalized form of unsharp masking, where $A \geq 1$. A specifies the amount of sharpening of the image.

If we use Laplacian filter to create sharpened image $f_s(x,y)$ with addition of the ...age

$$f_s(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) \\ f(x,y) + \nabla^2 f(x,y) \end{cases}$$

$$f_{hb}(x,y) = \begin{cases} A f(x,y) - \nabla^2 f(x,y) \\ A f(x,y) + \nabla^2 f(x,y) \end{cases}$$

Unsharp Masking and High boost Filtering

Highboost Masks

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & A+4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 \\ -1 & A+8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

where $A \geqslant 1$

Q1. Apply highboost filter on the image given below on the center pixel. Use the mask with $A = 1.7$.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} * \begin{bmatrix} -1 & -1 & -1 \\ -1 & A+8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Input image

$$= -1(1+2+3+4+6+7+8+9) + 5(1.7+8)$$

$$= -1(40) + 5(9.7)$$

$$= -40 + 48.5 = \underline{\underline{8.5}}$$

8:04 / 8:13

First Order Derivative Filters

1) Roberts operators (Cross-gradient)

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

$$\begin{bmatrix} z1 & z2 & z3 \\ z4 & z5 & z6 \\ z7 & z8 & z9 \end{bmatrix}$$

Image

2) Sobel operators

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

3) Prewitt operators

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

1) Roberts operators (Cross - gradient)

-1	0
0	1

0	-1
1	0

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Image

2) Sobel operators

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

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

3) Prewitt operators

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

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

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

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

Input image

1) Roberts operator

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

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

$= -50 + 100 = \underline{\underline{50}}.$

2) Sobel operator

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$= 50(-1) + 50(-2) + 100(-1) + 50(1) + 50(2) + 100(1)$

$= \cancel{-50} - \cancel{100} - \cancel{100} + \cancel{50} + \cancel{100} + \cancel{100} = 0$

3) Prewitt operator

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

$*$

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

Input image

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

$= \underline{\underline{0}}.$

Image Transforms

→ Image transforms are mathematical tools that help us to convert images from spatial domain to frequency domain.

→ Advantages for transforming images :
- It may isolate critical components of image pattern so that they are directly accessible for analysis.
- It may place image data in a more compact form so that it can be stored and transmitted efficiently.
- It is useful for fast computation of 2D convolution and correlation.
- It is reversible, i.e, we can revert to the initial domain.

Discrete Fourier Transform

* One - Dimensional DFT

$$F(k) = \sum_{x=0}^{N-1} f(x)\, e^{-j2\pi kx/N} \quad \text{where} \quad k = 0, 1, \ldots N-1.$$

* Inverse DFT

$$f(x) = \frac{1}{N} \sum_{k=0}^{N-1} F(k)\, e^{j2\pi kx/N} \quad \text{where} \quad x = 0, 1, 2, \ldots N-1$$

Important Formulas

$e^{j\pi} = -1$ $\quad$ $e^{-j\pi} = -1$ $\quad$ $e^{j\pi/2} = j$ $\quad$ $e^{-j\pi/2} = -j$

$e^{-j2\pi} = 1$ $\quad$ $e^{-j3\pi} = -1$ $\quad$ $e^{-j3\pi/2} = j$ $\quad$ $e^{-j9\pi/2} = -j$

$$e^{-j\theta} = \cos\theta - j\sin\theta$$

$$e^{j\theta} = \cos\theta + j\sin\theta$$

Image Transforms and DFT (Discrete Fourier Transform) With Examples

Q1. Compute DFT of the sequence $f(x) = \{\underset{0}{1}, \underset{1}{0}, \underset{2}{0}, \underset{3}{1}\}$.

$$F[k] = \sum_{x=0}^{N-1} f(x)\, e^{-j2\pi kx/N} \quad \text{where } k = 0, 1, \ldots N-1$$

Here $N = 4$.

$$F[k] = \sum_{x=0}^{3} f(x)\, e^{-j2\pi kx/4}$$

$$= f(0)e^{0} + f(1)e^{-j2\pi k/4} + f(2)e^{-j\pi k} + f(3)e^{-j3\pi k/2}$$

$$= 1 + 0 + 0 + e^{-j3\pi k/2}$$

$$= 1 + e^{-j3\pi k/2}$$

6:03 / 11:16

Image Transforms and DFT (Discrete Fourier Transform) With Examples

$$= f(0)e^{0} + f(1)e^{-j2\pi k/4} + f(2)e^{-j\pi k} + f(3)e^{-j3\pi k/2}$$

$$= 1 + 0 + 0 + e^{-j3\pi k/2}$$

$$= 1 + e^{-j3\pi k/2}.$$

When $k=0$

$$F[0] = 1 + e^{0} = 1+1 = 2.$$

When $k=1$

$$F[1] = 1 + e^{-j3\pi/2} = 1+j.$$

When $k=2$

$$F[2] = 1 + e^{-j3\pi} = 1-1 = 0.$$

When $k=3$

$$F[3] = 1 + e^{-j9\pi/2} = 1-j.$$

8:28 / 11:16

Image Transforms and DFT (Discrete Fourier Transform) With Examples

<u>when $k=2$</u>

$$F[2] = 1 + e^{-j3\pi} = 1 - 1 = 0.$$

<u>when $k=3$</u>

$$F[3] = 1 + e^{-j9\pi/2} = 1 - j.$$

$$F[k] = \{2,\ 1+j,\ 0,\ 1-j\}$$

8:47 / 11:16

* Kernel of a 4-point DFT :

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \longrightarrow \text{kernel}$$

* 1D DFT : $F[k] = \text{Kernel} \times f(x)$

* 2D DFT : $F[k, l] = \text{Kernel} \times f(x, y) \times \text{Kernel}^T$

Q. Calculate 4-point DFT for the sequence $x(n) = \{0, 1, 2, 3\}$ using matrix method.

Ans. The 4-point DFT in one dimensional = Kernel × Input sequence.

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix}$$

$$= \begin{bmatrix} 0 + 1 + 2 + 3 \\ 0 - j - 2 + 3j \\ 0 - 1 + 2 - 3 \\ 0 + j - 2 - 3j \end{bmatrix} = \begin{bmatrix} 6 \\ -2 + 2j \\ -2 \\ -2 - 2j \end{bmatrix}$$

Q3. Compute the 2D DFT of the grayscale image is given by

$$f(m,n) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Ans. $F(k,l) = \text{Kernal} \times f(m,n) \times \text{Kernal}^T$

Kernal for the 4-point DFT is $= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}.$

$$F[k,l] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \times$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}.$$

$$F[k,l] = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

$$= \begin{bmatrix} 16 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Discrete Cosine Transform

* One Dimensional DCT

$$X[k] = \alpha(k) \sum_{n=0}^{N-1} x(n) \cos\left\{\frac{(2n+1)\pi k}{2N}\right\}$$

where $0 \le k \le N-1$,

$$\alpha(k) = \begin{cases} \sqrt{\frac{1}{N}}, & \text{if } k=0 \\ \sqrt{\frac{2}{N}}, & \text{if } k \neq 0 \end{cases}$$

* Kernel of a 4-point DCT

$$\begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.6532 & 0.2706 & -0.2706 & -0.6532 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.2706 & -0.6532 & 0.6533 & -0.2706 \end{bmatrix}$$

* 1D DCT : $F[k] = \text{Kernel} \times f(x)$

* 2D DCT : $F[k,l] = \text{Kernel} \times f(x,y) \times \text{Kernel}^T$

Q Find the DCT of $f(x) = (1, 2, 4, 7)$.

Ans. $F = \text{Basis function} \times f(x)$

$$F = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.6532 & 0.2706 & -0.2706 & -0.6532 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.2706 & -0.6533 & 0.6533 & -0.2706 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \\ 4 \\ 7 \end{bmatrix}$$

$$= \begin{bmatrix} 7 \\ -4.459 \\ 1 \\ -0.370 \end{bmatrix}.$$

Q Find 2[illegible] DCT of $f(x,y) = \begin{bmatrix} 1 & 2 & 2 & 1 \\ 2 & 1 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 2 & 1 & 2 & 1 \end{bmatrix}.$

Discrete Cosine Transform and Haar transform with Examples

Ans: $F = \text{Kernel} \times f(x,y) \times \text{Kernel}^T$

$$F = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.6532 & 0.2706 & -0.2706 & -0.6532 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.2706 & -0.6533 & 0.6533 & -0.2706 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 2 & 1 \\ 2 & 1 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 2 & 1 & 2 & 1 \end{bmatrix} \times$$

$$\begin{bmatrix} 0.5 & 0.6532 & 0.5 & 0.2706 \\ 0.5 & 0.2706 & -0.5 & -0.6533 \\ 0.5 & -0.2706 & -0.5 & 0.6533 \\ 0.5 & -0.6532 & 0.5 & -0.2706 \end{bmatrix}$$

$$= \begin{bmatrix} 6 & 0.3025 & -1 & 0.9235 \\ 0 & -0.1463 & -0.3825 & -0.3532 \\ 0 & 0 & 0 & 0 \\ 0 & -0.3532 & -0.923 & -0.8525 \end{bmatrix}.$$

3:57 / 7:29 · Questions ›

Haar Transform

* Kernel for a 2×2 matrix

$$H_2 = \frac{1}{\sqrt{2}} \times \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H_4 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

Q Find the Haar transform of the signal

$$f(m,n) = \begin{bmatrix} 4 & -1 \\ 2 & 3 \end{bmatrix}_{2\times 2}$$

Ans. The 2D Haar transform of the signal $f(m,n)$ is given by $F(k,l)$ where,

$$F(k,l) = H_2 \times f(m,n) \times H_2^T$$

$$H_2 = \frac{1}{\sqrt{2}} \times \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

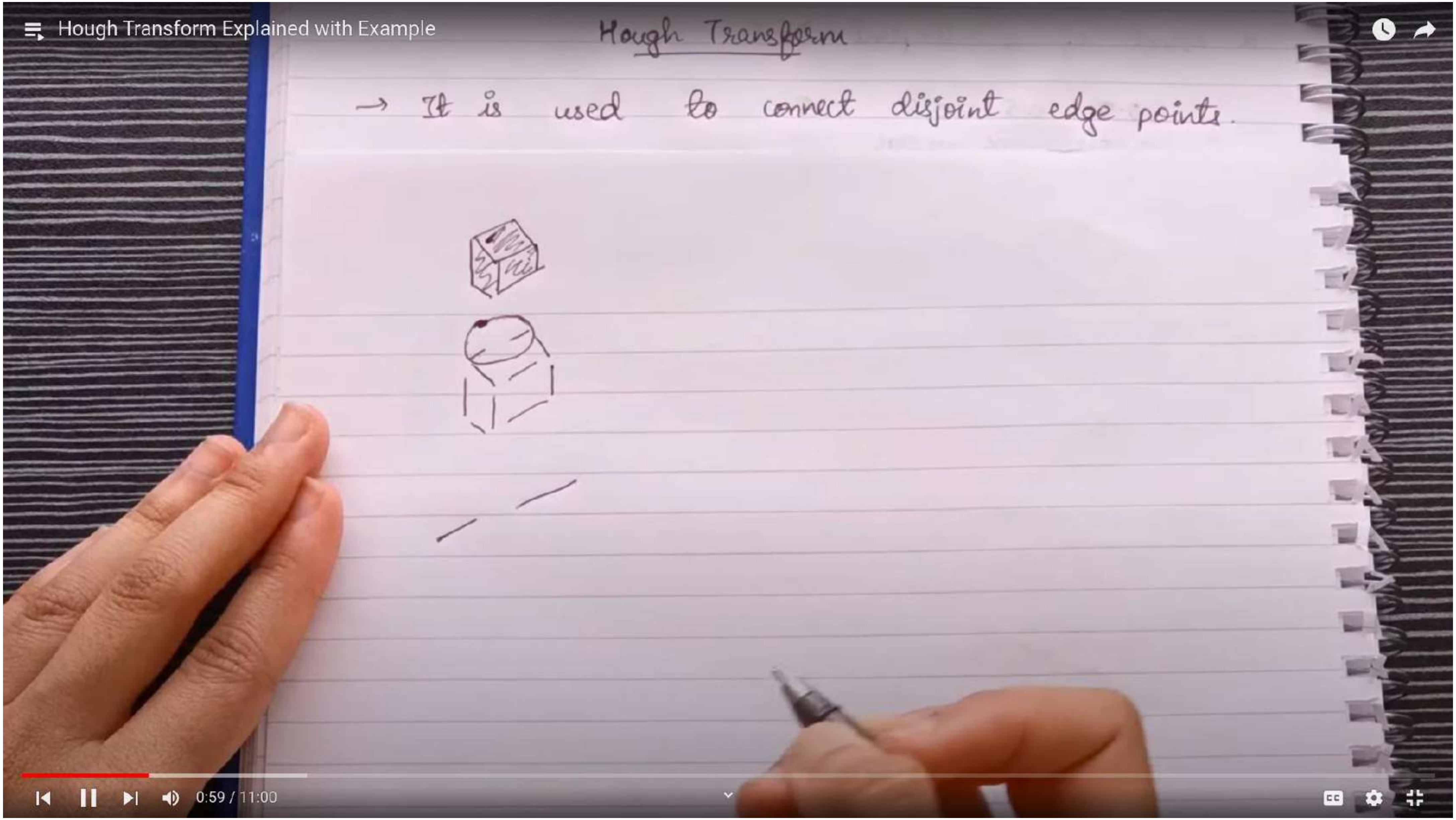
$$F(k,l) = \frac{1}{\sqrt{2}} \times \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \times \begin{bmatrix} 4 & -1 \\ 2 & 3 \end{bmatrix} \times \frac{1}{\sqrt{2}} \times \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

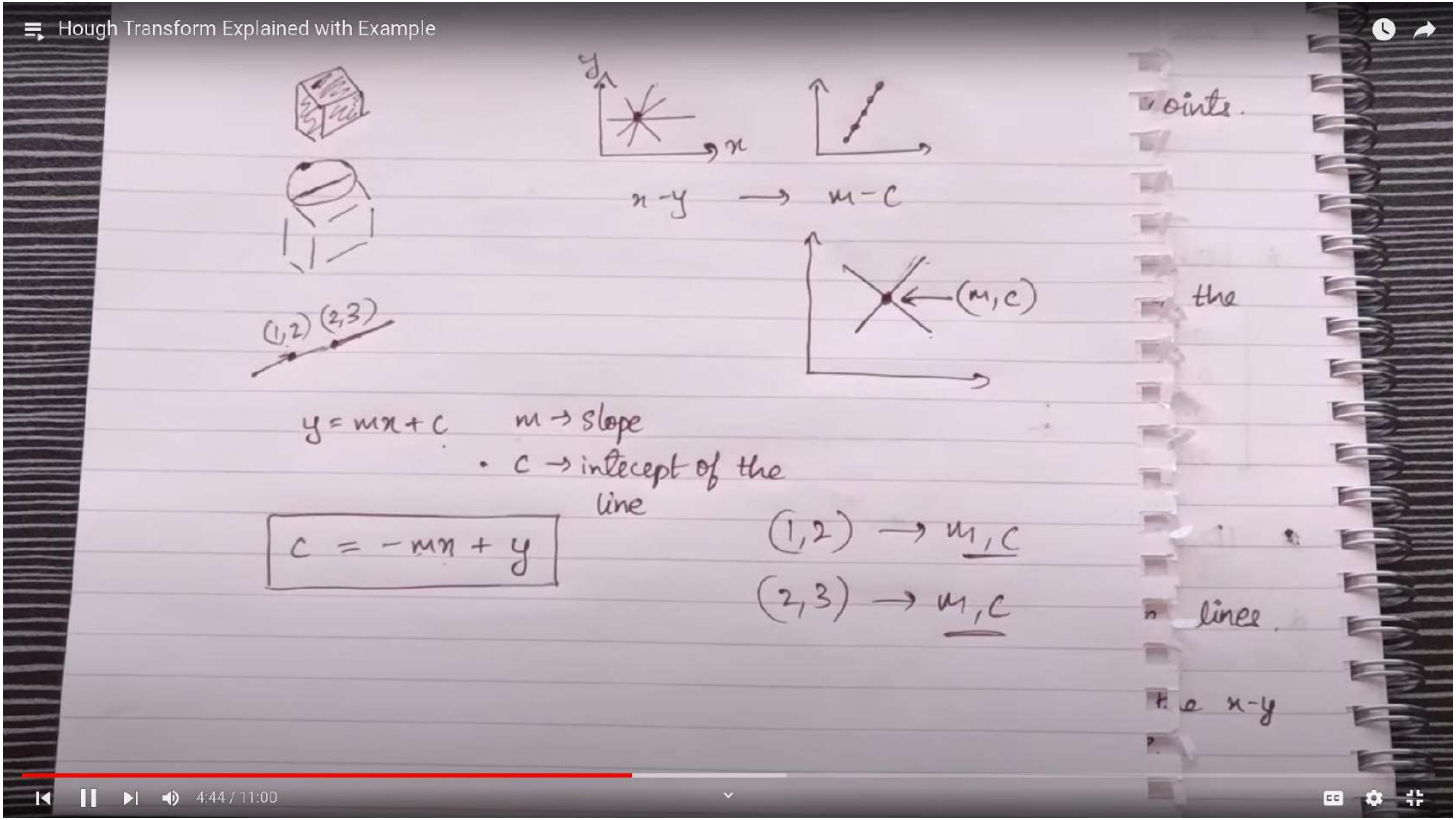
$$F(k,l) = \begin{bmatrix} 4 & 2 \\ -1 & 3 \end{bmatrix}.$$

Image Transforms

* 1D Formula : $F[k] = \text{kernel} \times f(x)$

* 2D Formula : $F[k, l] = \text{kernel} \times f(x, y) \times \text{kernel}^T$


Hough Transform Explained with Example
Hough Transform
→ It is used to connect disjoint edge points.
0:59 / 11:00


Hough Transform Explained with Example
y
x
x-y → m-c
(m,c)
(1,2) (2,3)
y = mx + c
m → slope
c → intecept of the line
c = -mx + y
(1,2) → m, c
(2,3) → m, c
oints.
the
lines.
x-y
4:44 / 11:00

Hough Transform

→ It is used to connect disjoint edge points.

* Equation of a line :

$y = mx + c$ where $m \rightarrow$ slope
$c \rightarrow$ intercept of the line

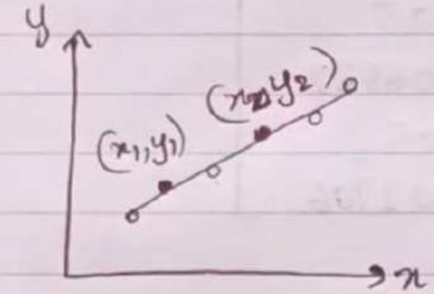
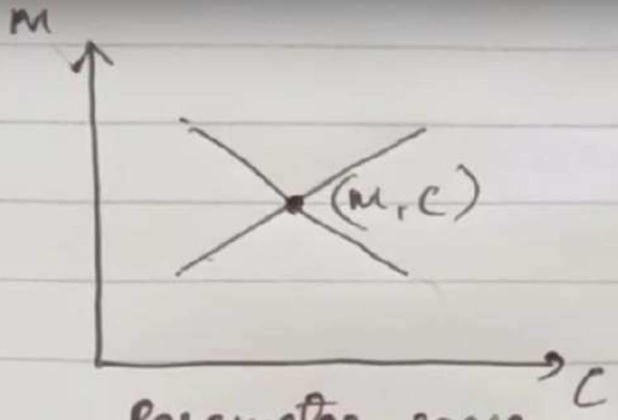


Image Space

A single point can be part of infinite lines.

Therefore, we transform that point in the x-y plane into a line in the c-m plane.

Hough Transform Explained with Example



Parameter space
(or)
Hough space

$(x_1, y_1) \longrightarrow (m_1, c_1)$

$(x_2, y_2) \longrightarrow (m_2, c_2)$

If A and B are two points connected by a line in the spatial domain, they will be intersecting lines in the Hough space.

6:25 / 11:00

Hough Transform Explained with Example

If A and B are two points connected by a line in the spatial domain, they will be intersecting lines in the Hough space.

Q1. Using Hough transform, show that the following points are collinear. Also find the equation of the line. (1,2), (2,3) and (3,4).

Ans. Equation of the line:

$$y = mx + c$$

In order to perform Hough transform, we need to convert the line from (x,y) plane to (m,c) plane.

$$\therefore \quad \boxed{c = -mx + y}$$

6:53 / 11:00

(i) For $(x,y) = (1,2)$, $c = -m + 2$

if $c = 0$, $0 = -m + 2$

$\boxed{m = 2}$

if $m = 0$, $\boxed{c = 2}$

Thus, $(m,c) = (2,2)$.

(ii) For $(x,y) = (2,3)$, $c = -2m + 3$.

if $c = 0$, $0 = -2m + 3$

$2m = 3$

$m = 3/2 = 1.5$. $\boxed{m = 1.5}$

if $m = 0$, $\boxed{c = 3.}$

Thus, $(m,c) = (1.5, 3)$.

$m = 3/2 = 1.5$. $\boxed{m = 1.5}$

if $m = 0$, $\boxed{c = 3}$.

Thus, $(m, c) = (1.5, 3)$.

(iii) For $(x, y) = (3, 4)$, $c = -3m + 4$.

if $c = 0$, $0 = -3m + 4$

$3m = 4$

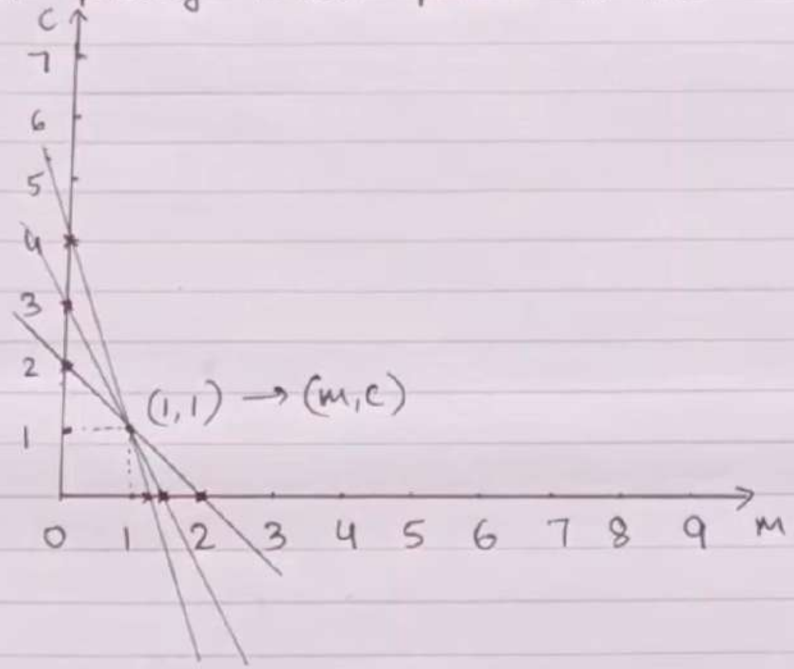
$m = 4/3 = 1.33$. $\boxed{m = 1.33}$

if $m = 0$, $\boxed{c = 4}$.

Thus $(m, c) = (1.33, 4)$.

$(m, c) = (2,2), (1.5,3), (1.33,4).$

On plotting these points in the m-c plane:



Original equation:

$$y = mx + c$$

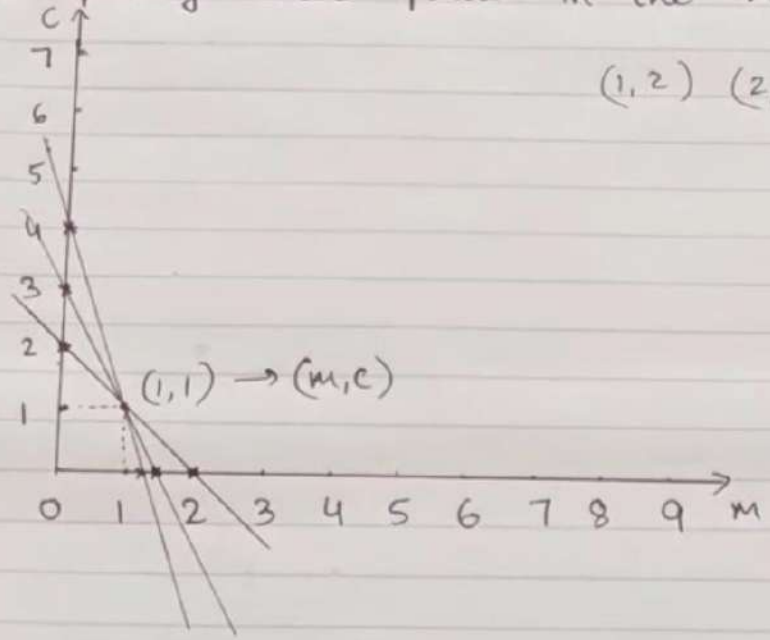
Substituting (1,1):

$$\boxed{y = x + 1}$$

$(m, c) = (2,2), (1.5, 3), (1.33, 4).$

On plotting these points in the m-c plane:

$(1,2)$ $(2,3)$ $(3,4)$



Original equation :

$$y = mx + c$$

Substituting $(1,1)$:

$$\boxed{y = x + 1} \rightarrow \text{Final equation}$$

Frequency Domain Filters

Spatial Domain

$$f(x,y) \longrightarrow \boxed{h(x,y)} \longrightarrow g(x,y)$$

$$g(x,y) = f(x,y) * h(x,y)$$

$h(x,y)$: impulse response

Frequency Domain

$$F(u,v) \longrightarrow \boxed{H(u,v)} \longrightarrow G(u,v)$$

$$G(u,v) = F(u,v) * H(u,v)$$

$$\left[g(x,y) = F^{-1}\left(F(u,v)\,H(u,v)\right) \right]$$

$H(u,v)$: transfer function

Basic steps for filtering in the frequency domain

Input image $f(x,y)$ ⟶ Pre-processing ⟶ Fourier transform ⟶ Filter function $H(u,v)$

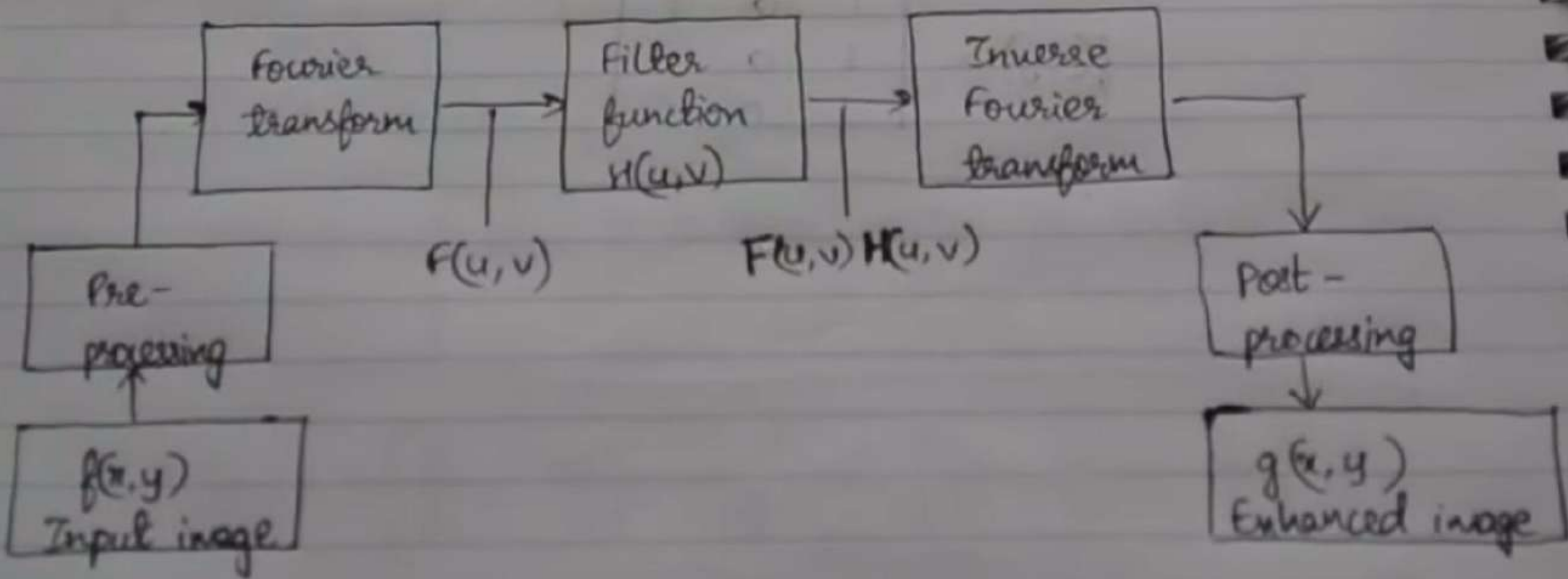
Fourier transform: $F(u,v)$

Basic steps for filtering in the frequency domain

Input image $f(x,y)$ → Pre-processing → Fourier transform → Filter function $H(u,v)$

$F(u,v)$

Basic steps for filtering in the frequency domain



1) Multiply the input image by $(-1)^{x+y}$ to center the transform.
2) Compute $F(u,v)$, the DFT of the image from (1).
3) Multiply $F(u,v)$ by a filter function $H(u,v)$.
4) Compute the inverse DFT of the result in (3).
5) Obtain the real part of the result in (4).
6) Multiply the result in (5) by $(-1)^{x+y}$.

Smoothening frequency filters

→ These are also called low-pass filters.
→ These are used to smoothen the image as they allow only low frequency components to pass through.
→ These are used to remove noise.

Types of low-pass filters

1) Ideal low-pass filter (ILPF)
2) Butterworth low-pass filter (BLPF)
3) Gaussian low-pass filter (GLPF)

1) Ideal low pass filter (ILPF)

$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$

[where D_0 is the cutoff frequency].

Smoothening frequency filters

→ These are also called low-pass filters.

→ These are used to smoothen the image as they allow only low frequency components to pass through.

→ These are used to remove noise.

Types of low-pass filters

1) Ideal low-pass filter (ILPF)
2) Butterworth low-pass filter (BLPF)
3) Gaussian low-pass filter (GLPF)

(0,0)	(0,1)	(0,2)
(1,0)	(1,1)	(1,2)
(2,0)	(2,1)	(2,2)

→

(-1,-1)	(-1,0)	(-1,1)
(0,-1)	(0,0)	(0,1)
(1,-[illegible]	[illegible]	[illegible]

Frequency domain filtering in image processing - Low pass and High pass filters

2) Butterworth low pass filter (BLPF)

$$H(u,v) = \frac{1}{1 + [D(u,v)/D_0]^{2n}}$$

[where D_0 is the cutoff frequency and n is the order of the Butterworth filter.]

3) Gaussian low pass filter (GLPF)

$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$

[where D_0 is the cutoff frequency]

10:55 / 24:15

Sharpening frequency filters

→ These are also called high-pass filters.
→ These are used to sharpen the image as they only allow high frequency components to pass through.
→ These are used to remove background of an image.

Types of High-pass filters

1) Ideal High-pass filter (IHPF)
2) Butterworth high-pass filter (BHPF)
3) Gaussian high pass filter (GHPF)

1) Ideal high-pass filter (IHPF)

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

[where D_0 is the cutoff frequency]

2) Butterworth high-pass filter (BHPF)

$$H(u,v) = \frac{1}{1 + [D_0 / D(u,v)]^{2n}}$$

[where D_0 is the cutoff frequency and n is the order of the Butterworth filter.]

3) Gaussian high-pass filter (GHPF)

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

[where D_0 is the cutoff frequency]

$(-1)^{x+y}$

$M = 4$
$N = 4$

Note: $D(u,v) = \sqrt{\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2} = \sqrt{(u-2)^2 + (v-2)^2}$

↳ Euclidean distance

center $= (2,2)$.

Q1. Convert the given spatial domain image using Fourier transform and perform Ideal low pass filter to smoothen the image. Choose D_0 as 0.5. Show the step by step procedure for doing the same.

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

Input image

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

Input image

1) Multiply the input image by $(-1)^{x+y}$ to center the transform.

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

$$= \begin{bmatrix} 1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 \\ 1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 \end{bmatrix}.$$

$(-1)^{0+0} \quad (-1)^{0+1} \quad (-1)^{0+2}$

Frequency domain filtering in image processing - Low pass and High pass filters

2) Compute the DFT of the image from (1).

$$F(u,v) = \text{kernel} \times f(x,y) \times \text{kernel}^T$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 \\ 1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 8 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

17:28 / 24:15

Frequency domain filtering in image processing - Low pass and High pass filters

3) Multiply $F(u,v)$ with the filter function $H(u,v)$.

For Ideal low pass filter,

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

4×4 $\quad (-1)^{x+y}$

$M=4, \; N=4$

$$D(u,v) = \sqrt{\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2}$$

$$= \sqrt{\left(u - \frac{4}{2}\right)^2 + \left(v - \frac{4}{2}\right)^2}$$

$$= \sqrt{(u-2)^2 + (v-2)^2}.$$

(0,0)	(0,1)	(0,2)	(0,3)
(1,0)	(1,1)	(1,2)	(1,3)
(2,0)	(2,1)	(2,2)	(2,3)
(3,0)	(3,1)	(3,2)	(3,3)

(Here we are calculating the distance between (u,v) from the center $(2,2)$ of the mask.)

20:14 / 24:15

Distance from each value of (u,v) from $(2,2)$:

$(0,0) \rightarrow \sqrt{(0-2)^2+(0-2)^2} = \sqrt{8} = 4.24$

$(0,1) \rightarrow \sqrt{(0-2)^2+(1-2)^2} = \sqrt{5} = 2.23$

$(0,2) \rightarrow \sqrt{(0-2)^2+(2-2)^2} = 2$

$(0,3) \rightarrow \sqrt{(0-2)^2+(3-2)^2} = \sqrt{5} = 2.23$

$(1,0) \rightarrow \sqrt{(1-2)^2+(0-2)^2} = \sqrt{5} = 2.23$

$(1,1) \rightarrow \sqrt{(1-2)^2+(1-2)^2} = \sqrt{2} = 1.41$

$(1,2) \rightarrow \sqrt{(1-2)^2+(2-2)^2} = 1$

$(1,3) \rightarrow \sqrt{(1-2)^2+(3-2)^2} = \sqrt{2} = 1.41$

$(2,0) \rightarrow \sqrt{(2-2)^2+(0-2)^2} = 2$

$(2,1) \rightarrow \sqrt{(2-2)^2+(1-2)^2} = 1$

$(2,2) \rightarrow \sqrt{(2-2)^2+(2-2)^2} = 0$

$(2,3) \rightarrow \sqrt{(2-2)^2+(3-2)^2} = 1$

$(3,0) \rightarrow \sqrt{(3-2)^2+(0-2)^2} = \sqrt{5} = 2.23$

$(3,1) \rightarrow \sqrt{(3-2)^2+(1-2)^2} = \sqrt{2} = 1.41$

$(3,2) \rightarrow \sqrt{(3-2)^2+(2-2)^2} = 1$

$(3,3) \rightarrow \sqrt{(3-2)^2+(3-2)^2} = \sqrt{2} = 1.41$

$(2,1) \rightarrow \sqrt{(2-2)^2 + (1-2)^2}$

$(2,2) \rightarrow \sqrt{(2-2)^2 + (2-2)^2} = 0$

$(2,3) \rightarrow \sqrt{(2-2)^2 + (3-2)^2} = 1$

$(3,0) \rightarrow \sqrt{(3-2)^2 + (0-2)^2} = \sqrt{5} = 2.23$

$(3,1) \rightarrow \sqrt{(3-2)^2 + (1-2)^2} = \sqrt{2} = 1.41$

$(3,2) \rightarrow \sqrt{(3-2)^2 + (2-2)^2} = 1$

$(3,3) \rightarrow \sqrt{(3-2)^2 + (3-2)^2} = \sqrt{2} = 1.41.$

$$\therefore D(u,v) = \begin{bmatrix} 4.24 & 2.23 & 2 & 2.23 \\ 2.23 & 1.41 & 1 & 1.41 \\ 2 & 1 & 0 & 1 \\ 2.23 & 1.41 & 1 & 1.41 \end{bmatrix}.$$

$D_0 = 0.5$ (Given)

$$\therefore H(u,v) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$G(u,v) = F(u,v) * H(u,v)$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 8 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

4) Compute the inverse DFT of the result in (3).

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \end{bmatrix} \rightarrow \text{IDFT Kernel}$$

4) Compute the inverse DFT of the result in (3).

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} \rightarrow \text{IDFT Kernel}$$

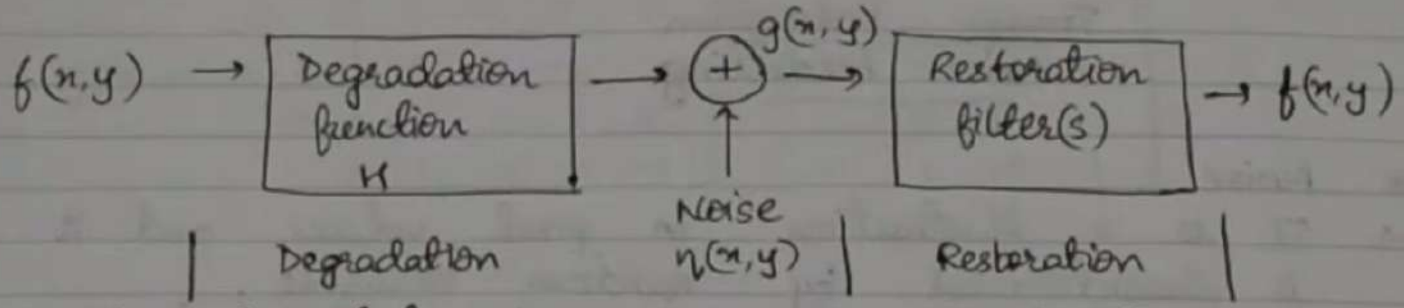
$$G^{-1}(u,v) = \frac{1}{4}\text{Kernel} \times \text{Image} \times \frac{1}{4}\text{Kernel}^T$$

$$= \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

$$= \frac{1}{16}\begin{bmatrix} 8 & 8 & 8 & 8 \\ -8 & -8 & -8 & -8 \\ 8 & 8 & 8 & 8 \\ -8 & -8 & -8 & -8 \end{bmatrix}.$$

Image Restoration

Image Degradation / Restoration Model



→ The degraded image in the spatial domain is given by:

$$g(x,y) = h(x,y) * f(x,y) + \eta(x,y)$$

where $h(x,y)$ is the spatial representation of the degradation function.

→ The degraded image in the frequency domain is given by:

$$G(u,v) = H(u,v)\,F(u,v) + N(u,v)$$

Noise Models

* Sources of noise

→ Image acquisition, digitization, transmission
→ Image sensors : noise occurs due to environmental conditions and by the quality of the sensing elements.

* White noise

→ In signal processing, white noise is a random signal having equal intensity at different frequencies, giving it a constant power spectral density.
→ The Fourier spectrum of noise is constant.

Types of Noise

1) Gaussian noise

→ Random noise that enters a system can be modelled as a Gaussian or normal distribution.

→ This noise affects both, dark and light areas of image.

→ The PDF of a Gaussian random variable, z, is given by

$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(z-\mu)^2/2\sigma^2}$$

where $z \rightarrow$ gray level

$\mu \rightarrow$ mean of average values of z

$\sigma \rightarrow$ standard deviation

$\sigma^2 \rightarrow$ variance

Image Restoration in digital image processing

2) Impulse noise

→ It is also known as shot noise, salt and pepper noise, and binary noise.

→ It occurs mostly because of sensor and memory problem because of which pixels are assigned incorrect maximum values

→ The PDF of (bipolar) impulse noise is given by

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$

If either P_a or P_b is zero, impulse noise is called unipolar.

5 seconds

5:26 / 13:11 • Image Noise ›

Q1. Why is impulse noise known as salt and pepper noise?

Ans. If neither of the two probabilities P_a or P_b are zero, and especially if they are approximately equal, impulse noise values resemble salt-and-pepper granules randomly distributed over the image. For this reason, bipolar impulse noise is also called salt-and-pepper noise.

3) Poisson noise

→ This type of noise manifests as a random structure or texture in images.

3) Poisson noise

→ This type of noise manifests as a random structure or texture in images.

→ It is very common in x-ray images.

→ The PDF of Poisson noise is given by:

$$p(z) = \frac{(nP)^z}{z!} e^{-nP}$$

where $n \rightarrow$ total no. of pixels

$z \rightarrow$ gray level

$P \rightarrow$ ratio of noise pixels to the total no. of pixels

where $n \rightarrow$ total no. of pixels
$z \rightarrow$ gray level
$P \rightarrow$ ratio of noise pixels to the total no. of pixels

4) Exponential noise
→ This type of noise occurs mostly due to the illumination problems
→ It is observed in laser imaging

→ The PDF of exponential noise is given by:

$$p(z) = \begin{cases} a e^{-az} & \text{for } z \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

where $a > 0$.

$\boxed{\begin{array}{l} \text{Mean} \rightarrow 1/a \\ \text{Variance} \rightarrow 1/a^2 \end{array}}$

5) Gamma noise (Erlang)

→ This type of noise also occurs mostly due to the illumination problems.

→ The PDF is given by:

$$p(z) = \begin{cases} \dfrac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$

where $a > 0$ and b is a positive integer.

Mean $\longrightarrow b/a$

Variance $\rightarrow b/a^2$

$a \rightarrow$ min. gray value
$b \rightarrow$ max. gray value

Image Restoration in digital image processing

6) Rayleigh noise

→ This type of noise is mostly present in range images.

→ Range images are mostly used in remote sensing applications.

→ The PDF is given by:

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-\frac{(z-a)^2}{b}} & \text{for } z \geq a \\ 0 & \text{otherwise} \end{cases}$$

Mean → $a + \sqrt{\frac{\pi b}{4}}$

Variance → $\frac{b(4-\pi)}{4}$.

8:59 / 13:11 • Impulse Noise ›

7) Uniform noise

→ It is also a very popular noise which occurs in images where different values of noise are equally probable.

→ It occurs because of Quantization noise.

→ The PDF is given by:

$$p(z) = \begin{cases} \frac{1}{b-a} & \text{for } a \le f(x,y) \le b \\ 0 & \text{otherwise} \end{cases}$$

Mean → $\frac{a+b}{2}$

Variance → $\frac{(b-a)^2}{12}$

Image Restoration in digital image processing

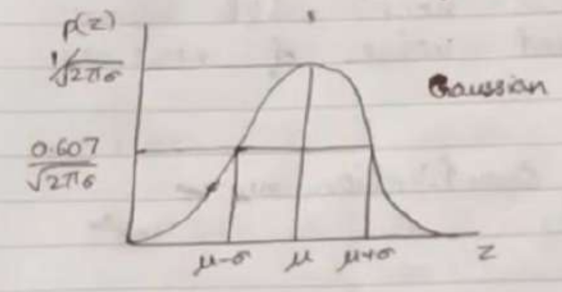
8) Periodic Noise

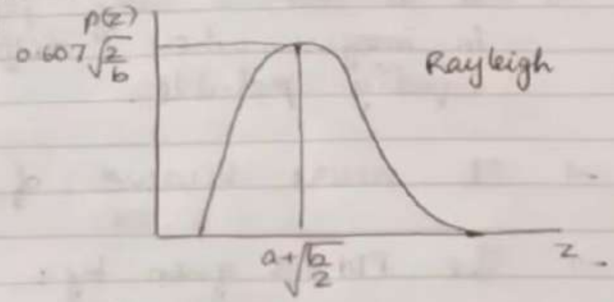
→ Arises typically from electrical or electromechanical interference.

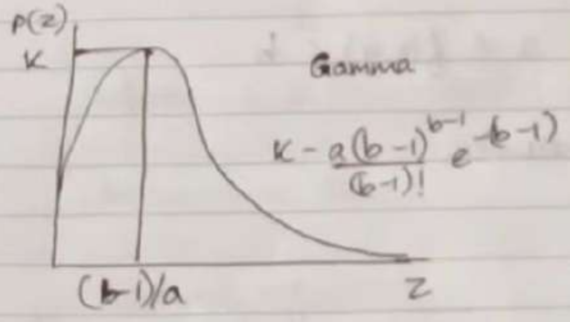
→ Reduced significantly via frequency domain filtering.

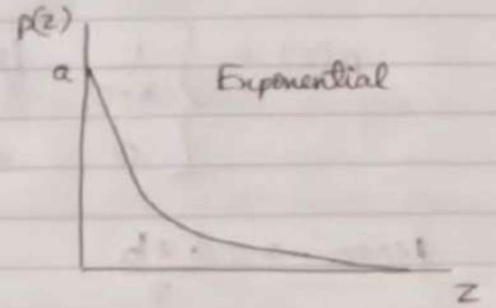
10:35 / 13:11 • Impulse Noise ›

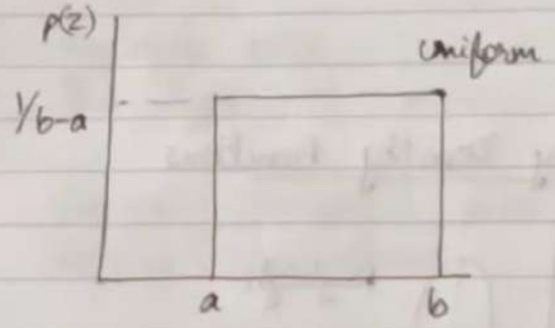
Probability Density Functions

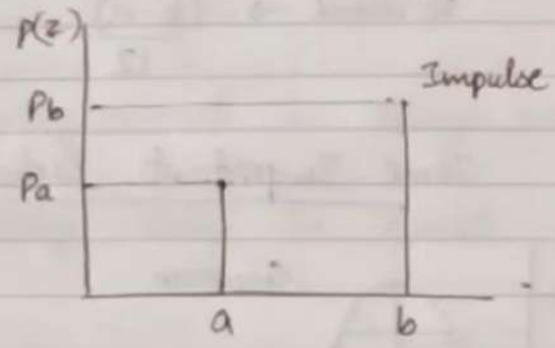












Noise Modelling

1) Gaussian
   - Electronic circuit noise, sensor noise due to poor illumination and/or high temperature.

2) Rayleigh
   - Range imaging

3) Exponential and gamma
   - Laser imaging

4) Impulse
   - Quick transients, such as faulty switching

5) Uniform
   - Least descriptive
   - Basis for numerous random number generators

Image Restoration in presence of Noise Only

Spatial filtering, which is used for image smoothening and sharpening, can also be used to remove noise.

$\boxed{\text{Spatial filters for Image Restoration}}$

- Mean Filters
- Order Statistic Filters

Image Restoration in presence of Noise Only

Mean Filters

1) Arithmetic mean filter

→ This filter removes local variations within the image.

→ It is similar to low pass filter.

→ It is useful in removing Gaussian noise and uniform noise.

$$f(x,y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s,t)$$

where $g(s,t)$ → mask applied over blurred image.

1:27 / 13:13

Image Restoration in presence of Noise Only

2) Geometric mean filter

→ This filter eliminates Gaussian noise.

→ It is ineffective for pepper type of noise.

→ This filter achieves smoothing comparable to the arithmetic mean filter, but it tends to lose less image details in the process.

$$\hat{f}(x,y) = \left[\prod_{(s,t) \in S_{xy}} g(s,t) \right]^{1/mn}$$

2:13 / 13:13

2) Geometric mean filter

→ This filter eliminates Gaussian noise.

→ It is ineffective for pepper type of noise.

→ This filter achieves smoothing comparable to the arithmetic mean filter, but it tends to lose less image details in the process.

$$\hat{f}(x,y) = \left[\prod_{(s,t) \in S_{xy}} g(s,t) \right]^{1/mn}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}_{3\times3}$$

$$= (1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8 \times 9)^{1/9}$$

Image Restoration in presence of Noise Only

3) Harmonic mean filter

→ It works well for salt noise, but fails for pepper noise.

→ Also works well for Gaussian noise.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}_{3\times 3}$$

$$f(x,y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

$$= \frac{9}{\left(\frac{1}{1}+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}\cdots\frac{1}{9}\right)}$$

4) Contra-harmonic mean filter

→ It is well suited for reducing the effects of

4:10 / 13:13

Image Restoration in presence of Noise Only

3) Harmonic mean filter

→ It works well for salt noise, but fails for pepper noise.

→ Also works well for Gaussian noise.

$$\hat{f}(x,y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & & \\ 7 & 8 & \end{bmatrix}$$

$$= \frac{9}{\left(\frac{1}{1} + \frac{1}{2} + \ldots\right.}$$

4) Contra-harmonic mean filter

→ It is well suited for reducing the e

4:08 / 13:13

Image Restoration in presence of Noise Only

... order of the filter.

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}.$$

when $Q = 0$, it becomes arithmetic mean filter.
$Q = 1$, it becomes harmonic mean filter.

* Usage :

→ Arithmetic and geometric mean filters : suited for Gaussian or uniform noise.
→ Contraharmonic filters : suited for impulse noise.

5:36 / 13:13

Image Restoration in presence of Noise Only

Order Statistic Filters

1) Median filter.

- → It is an example of non-linear filter.
- → It works by sorting the list and finding the median. The center pixel is then replaced by the median value.
- → It is effective in the presence of both bipolar and unipolar impulse noise.

$$f(x,y) = \underset{(s,t) \in S_{xy}}{\text{median}} \{g(s,t)\}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

6:05 / 13:13

2) Maximum filter

→ It selects the largest value in the sorted list.
→ It is used for removing pepper noise

$$f(x,y) = \max_{(s,t)\in S_{xy}} \{g(s,t)\}$$

3) Minimum filter

→ It selects the minimum value in the sorted list.
→ It is used for removing salt noise.

$$f(x,y) = \min_{(s,t)\in S_{xy}} \{g(s,t)\}$$

Image Restoration in presence of Noise Only

4) Midpoint filter

→ This filter selects the midpoint, which is the average of the minimum and maximum values.

→ It is very effective in removing Gaussian noise and Uniform noise.

$$f(x,y) = \frac{1}{2}\left[\max_{(s,t)\in S_{xy}}\{g(s,t)\} + \min_{(s,t)\in S_{xy}}\{g(s,t)\}\right]$$

5) Alpha-trimmed me

7:33 / 13:13

5) Alpha-trimmed mean filter.

→ It is based on the concept of computation of the average of the pixels that falls within the window.

→ It works by deleting the $d/2$ lowest and the $d/2$ highest gray-level values.

→ It is useful in situations involving multiple types of noise, such as a combination of salt-and-pepper and Gaussian noise.

$$f(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} g_r(s,t)$$

[If $d=0$ → arithmetic mean filter]

[If $d = mn-1$ → median filter]

6) Trimmed mean filter

d/2 highest gray-level values.

→ It is useful in situations involving multiple types of noise, such as a combination of salt-and-pepper and Gaussian noise.

$$f(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} g_r(s,t)$$

[If $d = 0 \rightarrow$ arithmetic mean filter]

[If $d = mn-1 \rightarrow$ median filter]

6) Trimmed mean filter

→ It works by removing the max. and min. values and calculating the average of the remaining values.

$$\cancel{1}, \underbrace{2, 3, 4, 5, 6, 7, 8}_{7}, \cancel{9}$$

Q1. Assume that the image given below is affected by Gaussian and impulse noise. Identify the type of filter and apply on the image which removes both noises.

4	0	4	9
4	1	8	8
1	1	6	6
1	0	5	6
1	1	5	6

Input image

order $\rightarrow$ 5×4

$m = 5$

$n = 4$

$mn = 5 \times 4 = 20$.

Assuming $d = mn - 2$
$= 20 - 2$
$= 18.$

$\alpha = \frac{d}{2} = \frac{18}{2} = 9.$

$\underbrace{0,0,1,1,1,1,1,1,4}_{9},4,4,\underbrace{5,5,6,6,6,6,8,8,9}_{9}$

$$f(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} g_r(s,t)$$ $\rightarrow$ Alpha trimmed mean filter

$$= \frac{1}{20-18}\left[4+4\right]$$

$$= \frac{1}{2}(8)$$

$$= \underline{\underline{4}}.$$

$d = 0 \rightarrow$ arithmetic mean filter

$d = mn - 1 \rightarrow$ median filter

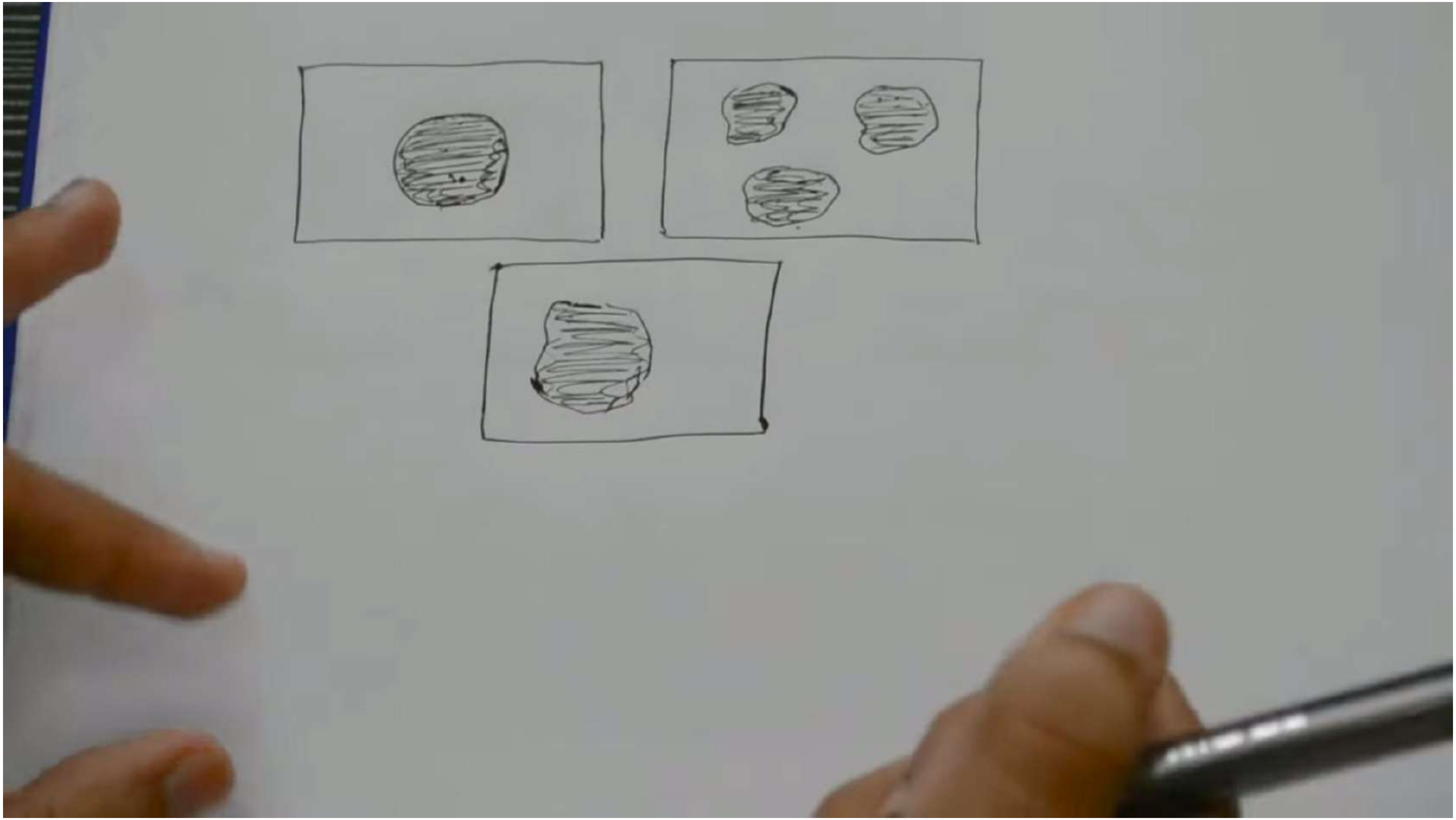
$= \underline{\underline{4}}.$

Image Segmentation

It is a stage of transition from image processing methods whose inputs and outputs are images, to methods in which the inputs are images but the outputs are attributes extracted from those images.

Segmentation refers to the process of partitioning an image into multiple regions.

Image segmentation is typically used to locate objects and boundaries in images.



Methods of finding Discontinuity and Similarity

* Discontinuity
  - Isolated point
  - Line detection
  - Edge detection

* Similarity
  - Thresholding
  - Region growing
  - Region split
  - Region merge

* Discontinuity

1) Detection of isolated points

→ It is done using sharpening filters. We use second order derivative using Laplacian.

→ A point has been detected at a location (x, y) on which the kernel is centered if the absolute value of the response of the filter at that point exceeds a specified threshold.

→ Such points are labeled 1 and all others are labeled 0 in the output image, thus producing a binary image.

→ A point has been detected at a location (x, y) on which the kernel is centered if the absolute value of the response of the filter at that point exceeds a specified threshold.

→ Such points are labeled 1 and all others are labeled 0 in the output image, thus producing a binary image.

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

$$g(x,y) = \begin{cases} 1 & \text{if } |Z(x,y)| > T \\ 0 & \text{otherwise} \end{cases}$$

Ke

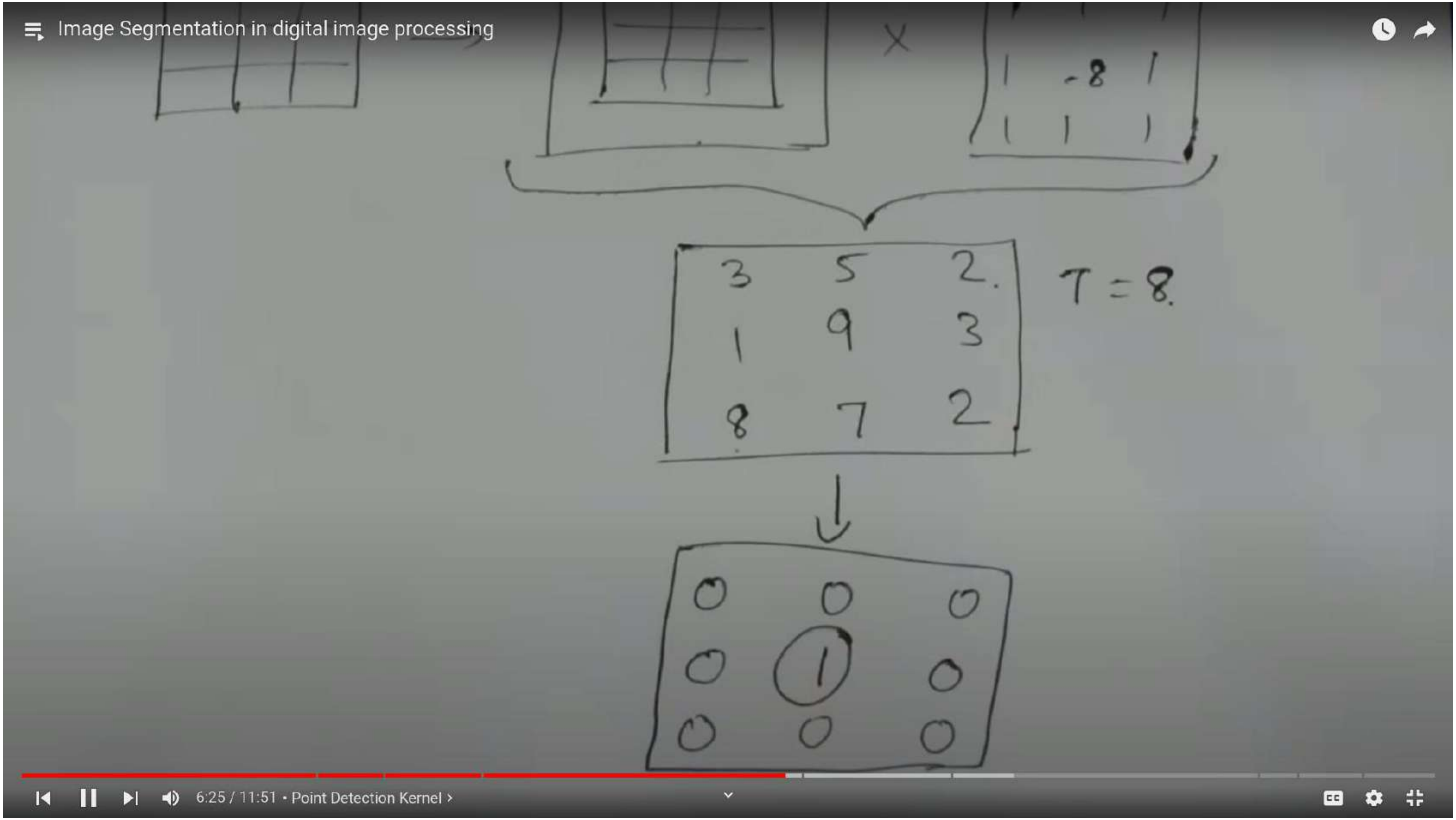


Image Segmentation in digital image processing
X
-8
3 5 2
1 9 3
8 7 2
T = 8
0 0 0
0 1 0
0 0 0
6:25 / 11:51 • Point Detection Kernel ›

2) Line detection

→ We use second order derivatives which result in a stronger filter response and produce thinner lines than first derivatives.

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}$$

Horizontal

$$\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

$+45°$

$$\begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$$

Vertical

$$\begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}$$

$-45°$

3) Edge detection

→ It is an approach used frequently for segmenting images based on abrupt (local) changes in intensity.

→ First order derivatives such as Roberts-cross, Prewitt and Sobel operators are preferred for thicker lines.

→ Second order derivatives (Laplacian) are used for detecting thinner lines

→ Kirsch compass kernels are used for finding the maximum edge strength in a few predetermined directions.

Edge Detection Kernels

1) Roberts cross-gradient operators

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

2) Prewitt operators

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

3) sobel operators (have better noise suppression (smoothing) characteristics)

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

4) Prewitt masks for diagonal edges

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

$+45^\circ$ $\qquad$ -45°

Image Segmentation in digital image processing

4) Prewitt masks for diagonal edges

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

$+45^\circ$ $\qquad$ -45°

5) Sobel masks for diagonal edges

$$\begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

$+45^\circ$ $\qquad$ -45°

6) Kirsch compass kernels

$$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$$

N

$$\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$$

N.W

$$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

W

$$\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

S

$$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}$$

SE

$$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}$$

E

$$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

NE

7) Second order kernels (The Laplacian)

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Thresholding

1) Thresholding

→ It is carried out with the assumption that the range of intensity levels covered by objects of interest is different from the background.

→ Steps :

a) A threshold T is selected.

b) Any point (x,y) in the image at which $f(x,y) > T$ is called an object point.

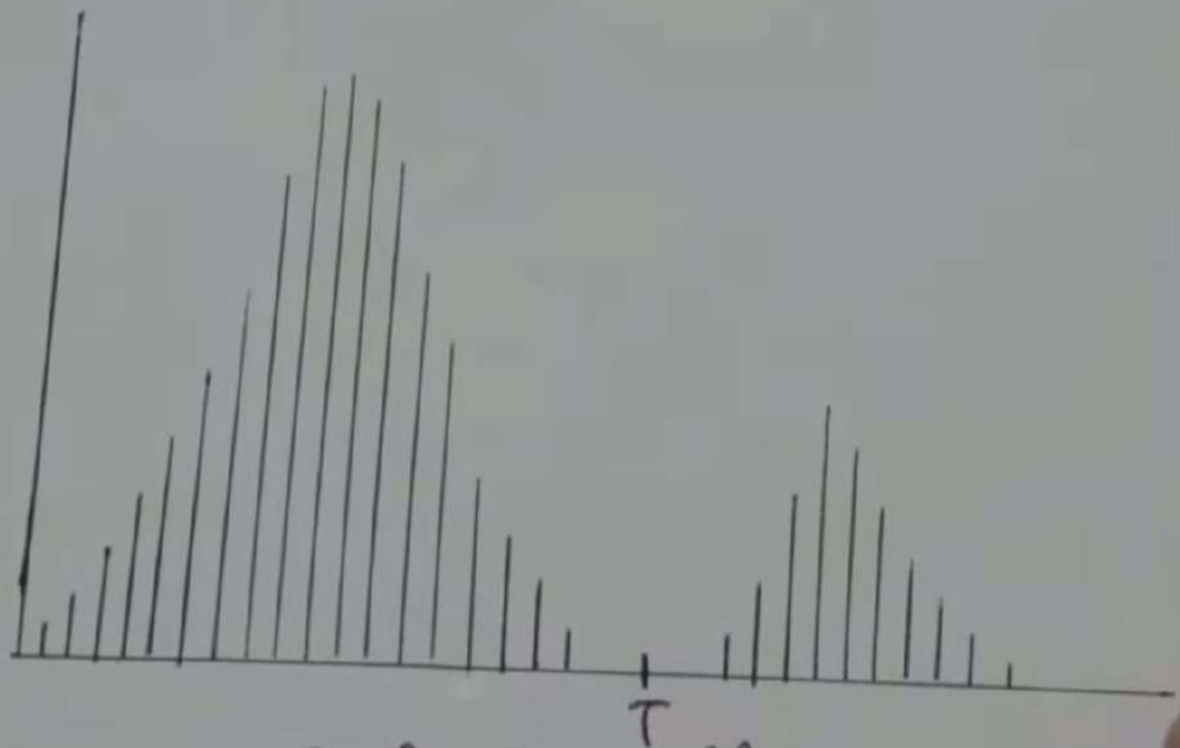
c) The segmented image, denoted by $g(x,y)$ is given by

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) \le T \end{cases}$$

Thresholding

1) Thresholding

→ It is carried out with the assumption that the



Single Threshold

Types of Thresholding

1) Global thresholding
- T is a constant.

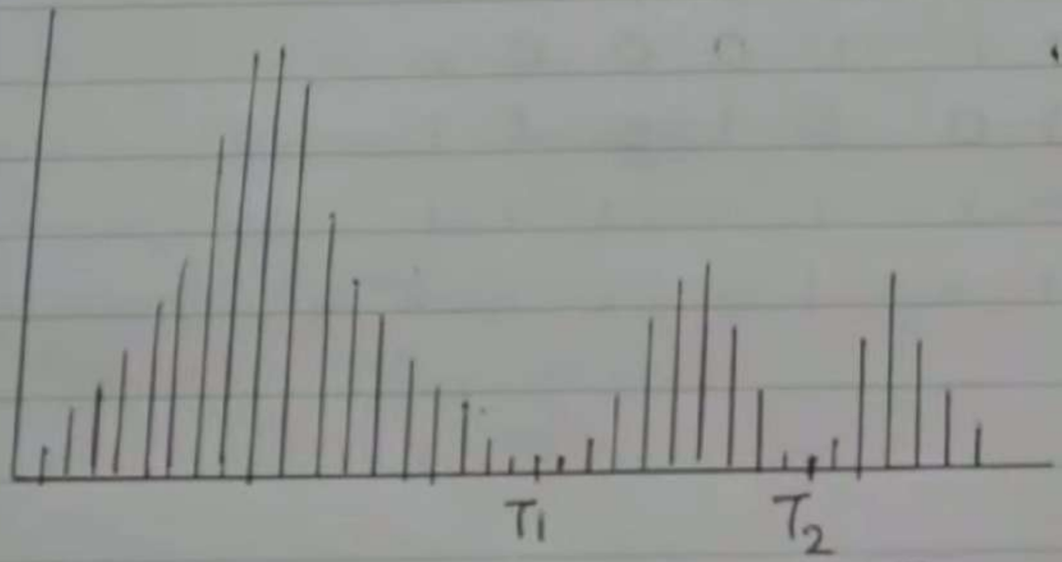
2) Variable thresholding
- T changes over an image.

3) Local or regional thresholding
- In variable thresholding, if the value of T at any point (x,y) in an image depends on the properties of a neighborhood of (x,y).

4) Dynamic or adaptive thresholding

- In variable thresholding, if the value of T depends on the spatial coordinates (x, y).

A Histogram with three dominant modes (two types of light objects on a dark background)



$$g(x,y) = \begin{cases} a & \text{if } f(x,y) > T_2 \\ b & \text{if } T_1 < f(x,y) \leq T_2 \\ c & \text{if } f(x,y) \leq T_1 \end{cases}$$

Procedure for Global Thresholding to Obtain T

1) Select an initial estimate for T. (This value should be greater than the minimum and less than the maximum intensity level in the image. It is better to choose the average intensity of an image.)

2) Segment the image using T. This will produce 2 groups of pixels : G1 consisting of all pixels with gray level values > T and G2 consisting of pixels with values <= T.

3) Compute the average gray level values μ1 and μ2 for the pixels in regions G1 and G2.

3) Compute the average gray level values $\mu 1$ and $\mu 2$ for the pixels in regions G1 and G2.

4) Compute a new threshold value :

$$T = \frac{1}{2}(\mu 1 + \mu 2)$$

5) Repeat step 2 through 4 until the difference in T in successive iterations is smaller than a predefined parameter T_o.

$$\begin{bmatrix} 5 & 3 & 9 \\ 2 & 1 & 7 \\ 8 & 4 & 2 \end{bmatrix}$$

Let $T = \dfrac{5+3+9+2+1+7+8+4+2}{9}$

$T = \dfrac{41}{9} = 4.55 \approx 5.$

Segmenting the image using T, we would get

$G1 = \{9, 7, 8\}$

$G2 = \{5, 3, 2, 1, 4, 2\}$

$\mu_1 = \dfrac{9+7+8}{3}$

$= \dfrac{24}{3} = 8.$

$\mu_2 = \dfrac{5+3+2+1+4+2}{6}$

$= \dfrac{17}{6} = 2.83 \approx 3.$

$T = \dfrac{1}{2}(8+3)$

Thresholding in digital image processing

8	4	2

$$T = \frac{41}{9} = 4.55 \approx 5.$$

Segmenting the image using T, we would get

$G1 = \{9, 7, 8\}$ $\qquad$ $G2 = \{5, 3, 2, 1, 4, 2\}$

$$\mu_1 = \frac{9+7+8}{3} = \frac{24}{3} = 8.$$

$$\mu_2 = \frac{5+3+2+1+4+2}{6} = \frac{17}{6} = 2.83 \approx 3.$$

$$T = \frac{1}{2}(8+3)$$

$$= \frac{1}{2}(11) = 5.5 \approx 5.$$

8:36 / 12:10

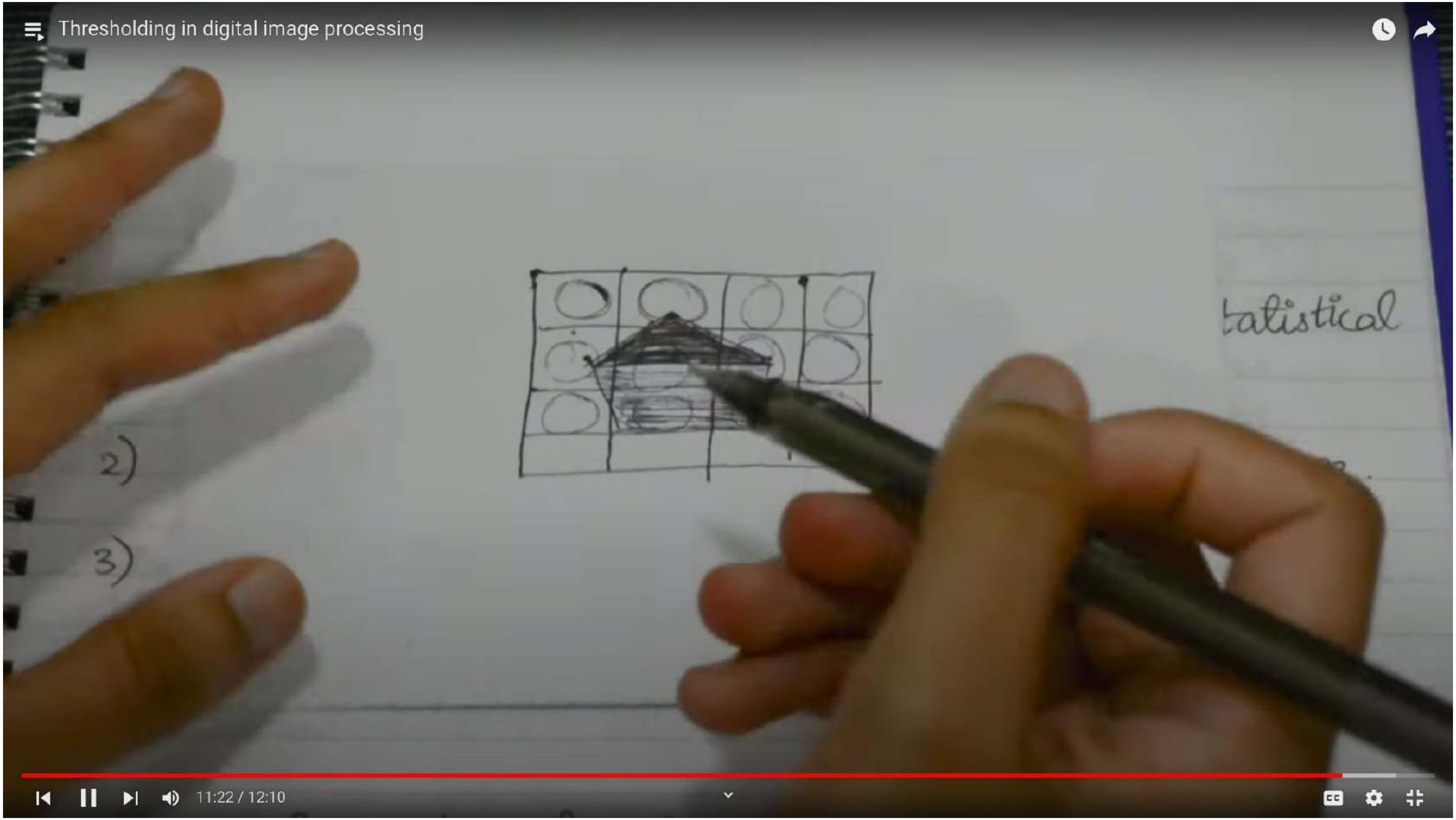
Procedure for Adaptive Thresholding

1) Convolve the image with a suitable statistical operator, ie. the mean or median.

2) Subtract the original from the convolved image.

3) Threshold the difference image with C.

4) Invert the threshold image.


Thresholding in digital image processing
tatistical
2)
3)
11:22 / 12:10

Region-based Segmentation

→ Edges and thresholds sometimes do not give good results for segmentation.

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

→ There are two main approaches to region-based segmentation : region-growing and region-splitting.

Procedure for Region - Growing

1) Find all connected components in $S(x,y)$ and reduce each connected component to one pixel. Label all such pixels found as 1. All other pixels in S are labeled 0.

2) Form an image f_Q such that, at each point (x,y), $f_Q(x,y) = 1$ if the input image satisfies a given predicate, Q, at those coordinates, and $f_Q(x,y) = 0$ otherwise.

2) Form an image f_Q such that, at each point (x, y), $f_Q(x, y) = 1$ if the input image satisfies a given predicate, Q, at those coordinates, and $f_Q(x, y) = 0$ otherwise.

3) Let g be an image formed by appending to each seed point in S all the 1-valued points in f_Q that are 8-connected to that seed point.

4) Label each connected component in g with a different region label (Ex. integers or letters). This is the segmented image obtained by region growing.

Procedure for Region splitting and Merging

1) If a region R is inhomogeneous ($P(R) = \text{False}$), then R is split into four sub-regions.

2) If 2 adjacent regions R_i, R_j are homogeneous ($P(R_i \cup R_j) = \text{True}$), then they are merged.

3) The algorithm stops when no further splitting or merging is possible.

Note: Condition for region growing:
abs(seed value - pixel value) $\leq$ Threshold

Q1. Apply region growing on the following image with initial point at (2,2) and threshold value as 2. Use 4 connectivity.

$T = 2$

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

Ans. The segmented region is shown in the following figure.

Condition → absolute difference <= 2.

4 way connectivity

Q1. Apply region growing on the following image with initial point at (2,2) and threshold value as 2. Use 4 connectivity.

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

$T = 2$

a

$7-6=1$ ✓

$7-4=3$ ✗

$7-5=2$ ✓

$7-3=4$ ✗

Ans. The segmented region is shown in the following figure.

Condition → absolute difference $<= 2$.

4 way connectivity

Ans. The segmented region is shown in the following figure.

Condition → absolute difference <= 2.

4 way connectivity

Here, we will compare each element which is 4 way connected to the seed element.

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

Segmented image obtained from region growing:

	0	1	2	3
0	0	0	0	0
1	0	1	1	0
2	0	0	1	0
3	0	0	1	0

Q2. Apply region growing on the following image with seed point as 6 and threshold value as 3.

Seed point = 6
T = 3

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

Ans. Condition → absolute difference <= 3.
8 way connectivity.

Ans. Condition $\rightarrow$ absolute difference $<= 3$.

8 way connectivity.

Seed point $= 6$.

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

Region based Segmentation

$$
\begin{matrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
\longrightarrow \text{Segmented Image}
$$

14:13 / 23:48

Region based Segmentation

Q3. Apply splitting and merging on the following image with threshold value equal to 3.

$T = 3$

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	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

Ans. Condition : absolute difference $<= 3$.

Max value $= 7$

Min value $= 0$

$|7-0| = \underline{7}$ which is greater than 3.

Therefore we will split the region into 4 sub-regions.

15:45 / 23:48

Splitting

	5	6	6	6	7	7	6	6	
(a)	6	7	6	7	5	5	4	7	(b)
	6	6	4	4	3	2	5	6	
	5	4	5	4	2	3	4	6	
	0	3	2	3	3	2	5	7	
(c)	0	0	0	0	2	2	5	6	(d)
	1	1	0	1	0	3	4	4	
	1	0	1	0	2	3	5	4	

In region (a):

Max = 7 Min = 4

$|7-4| = 3$ which is equal to threshold.

Therefore, no need to split.

Splitting

	5	6	6	6	7	7	6	6	
a	6	7	6	7	5	5	4	7	b
	6	6	4	4	3	2	5	6	
	5	4	5	4	2	3	4	6	
	0	3	2	3	3	2	5	7	
c	0	0	0	0	2	2	5	6	d
	1	1	0	1	0	3	4	4	
	1	0	1	0	2	3	5	4	

$|7-2| = 5$

$|3-0| = 3$

$|7-0| = 7$

In region a :

Max = 7   Min = 4

$|7-4| = 3$ which is equal to threshold.

Therefore, no need to split.

In region (b):

Max = 7 Min = 2

$|7-2| = 5$ which is greater than 3.

Therefore we will split the region (b) into 4 sub-regions.

In region (c):

Max = 3 Min = 0

$|3-0| = 3$.

So no need to split.

In region (d):

Max = 7 Min = 0

$|7-0| = 7$.

So we will split into 4 sub-regions.

						b1		b2	
	5	6	6	6	7	7	6	6	
a	6	7	6	7	5	5	4	7	b
						b3		b4	
	6	6	4	4	3	2	5	6	
	5	4	5	4	2	3	4	6	
						d1		d2	
	0	3	2	3	3	2	5	7	
c	0	0	0	0	2	2	5	6	d
						d3		d4	
	1	1	0	1	0	3	4	4	
	1	0	1	0	2	3	5	4	

Furthermore, we will check all the sub-regions. Since all of them are $<= 3$, no further splitting is required.

Region based Segmentation

					b1		b2		
	5	6	6	6	7	7	6	6	
a	6	7	6	7	5	5	4	7	b
					b3		b4		
	6	6	4	4	3	2	5	6	
	5	4	5	4	2	3	4	6	
					d1		d2		
	0	3	2	3	3	2	5	7	
c	0	0	0	0	2	2	5	6	d
					d3		d4		
	1	1	0	1	0	3	4	4	
	1	0	1	0	2	3	5	4	

Furthermore, we will check all the sub-regions. Since all of them are $<= 3$, no further splitting is required.

Merging

Check adjacent regions, if they are falling within the threshold, then merge.

Consider regions a and b1

Max = 7 Min = 4

$|7-4| = 3$ ✓ Merge.

20:33 / 23:48

Consider regions a b1 and b2

Max = 7 Min = 4

|7-4| = 3. ✓ Merge.

Consider regions a b1 b2 and b4

Max = 7 Min = 4

|7-4| = 3 ✓ Merge

Consider regions a b1 b2 b4 and d2

Max = 7 Min = 4

|7-4| = 3 ✓ Merge.

Similarly, merge a b1 b2 b4 d2 with d4.

Region based Segmentation

Similarly, merge (c), (d1), (b3) and (d3).

Final segmented image :

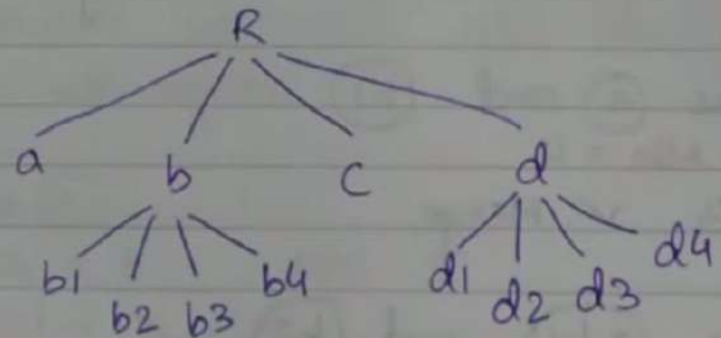
5	6	6	6	7	7	6	6
6	7	6	7	5	5	4	7
6	6	4	4	5	2	5	6
5	4	5	4	2	3	4	6
0	3	2	3	3	2	4	7
0	0	0	0	2	3	5	6
1	1	0	1	0	3	4	4
1	0	1	0	2	3	5	4

a b c d b1 b2 b3 b4 d1 d2 d3 d4

22:39 / 23:48

Final segmented image :

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



Quadtree Structure
for Splitting

Morphological Operators

Morphological image processing (or morphology) describes a range of image processing techniques that deal with the shape (or morphology) of features in an image.

Morphological operations are typically applied to remove imperfections introduced during segmentation, and so typically operate on bi-level images.

These operations are performed using the concepts of reflection and translation.

* Structuring Element (SE)

→ It is a small set to probe the image under study.

→ For each structuring element, we should define the origin.

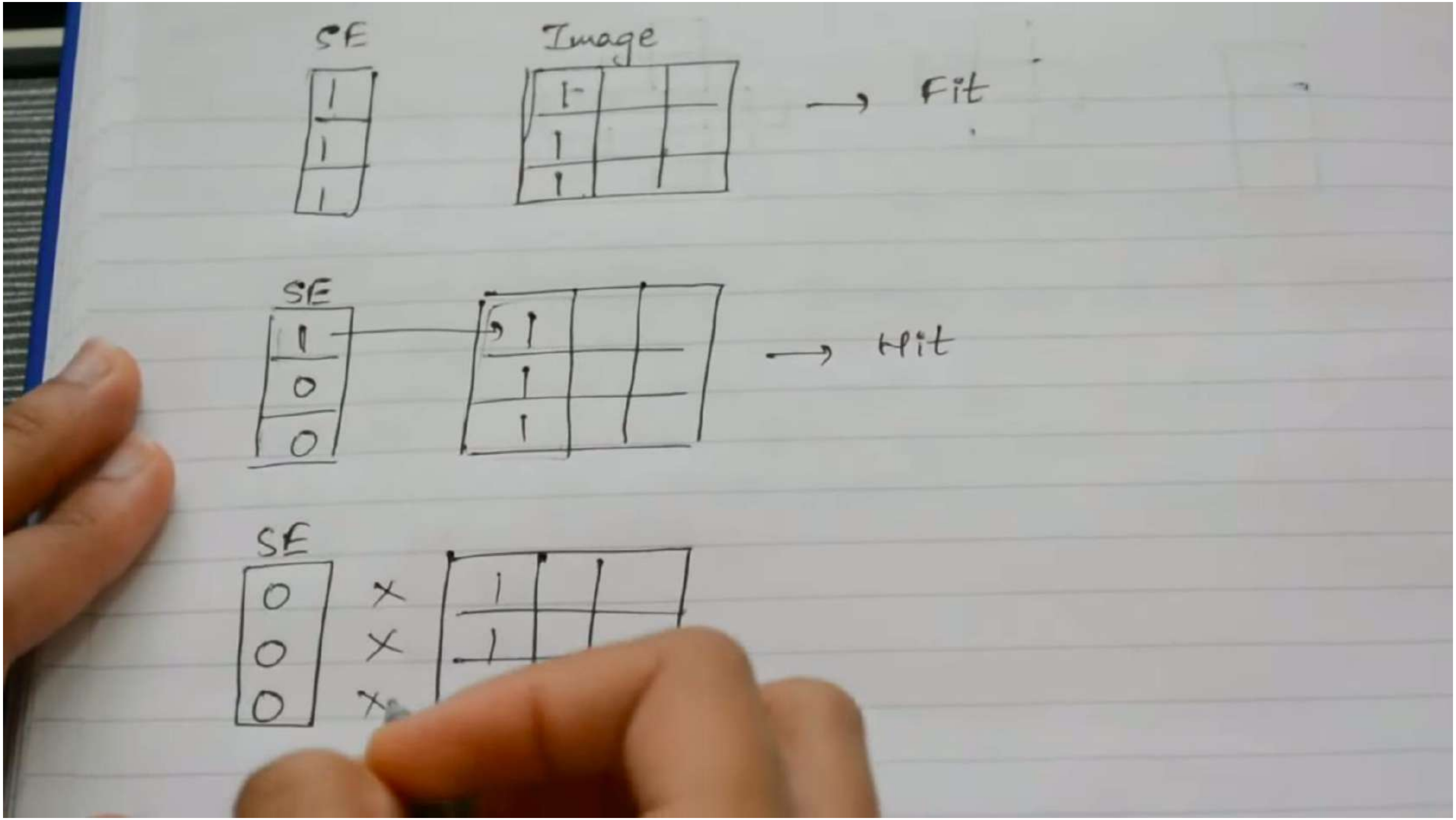
→ The shape and size must be adapted to geometric properties for the objects.

→ We check for three conditions while applying the SE:

↳ Fit

↳ Slit

↳ Miss


SE
Image
1
1
1
1
1
1
Fit
SE
1
0
0
1
1
1
Hit
SE
0
0
0
x
x
x
1
1

→ SEs can have varying sizes.

→ Usually the element values are 0, 1 and none(!).

→ Empty spots in the structuring elements are don't care's.

→ Examples of SE:

Box →

1	1	1
1	1	1
1	1	1

Disk →

	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
1	1	1	1	1	1	1
	1	1	1	1	1	
		1	1	1		

1	1	
1	0	
1		0

1	1	1
1	10	1
1	1	1

Morphological Operations

→ Basic operations are dual to each other.

1) Dilation and Erosion
→ Dilation enlarges foreground and shrinks background.
→ Erosion shrinks foreground and enlarges background.

* Dilation
→ It is the set of all points in the image where the structuring element "touches or hits" the foreground.
→ Consider each pixel in the input image. If the structuring element matches completely with the pixel values or even if atleast one match is found, write a "1" at the origin of the structuring element.

* Application of dilation

→ Repair breaks (bridging the gaps)

→ Repair intrusions

→ Enlarges the object

* Dilation is denoted by $A \oplus B$.

* Erosion

→ It is the set of all points in the image where the structuring element "fits into".

→ Consider each foreground pixel in the input image. If the structuring element matches completely with the pixel values, write a "1" at the origin of the structuring element.

* Application of erosion

→ Split apart joined objects

→ Strip away extrusions

→ Shrink the objects

* Erosion is denoted by $A \ominus B$.

Q1. Compute $A \oplus B$ for the following input images.

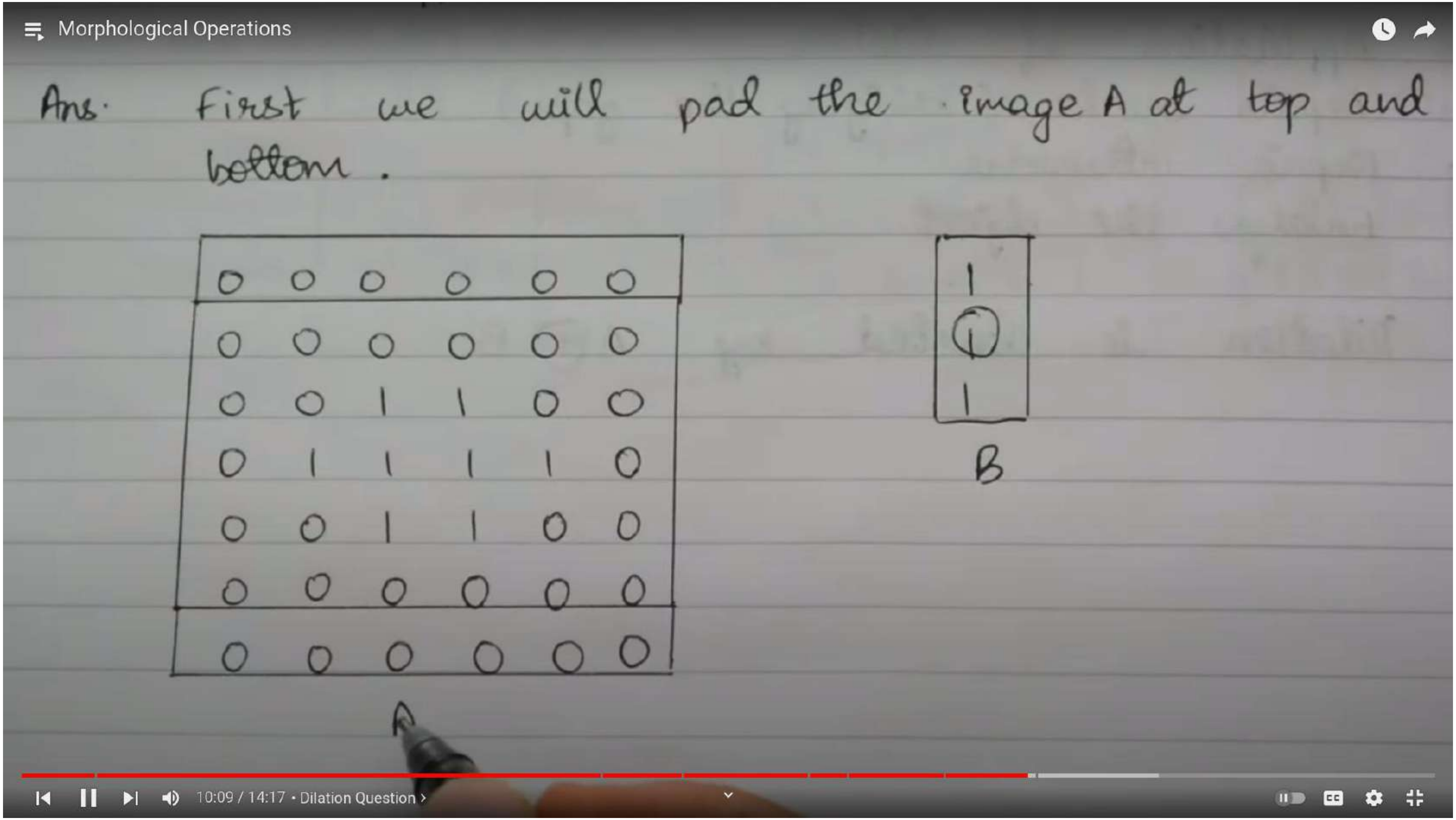
0	0	0	0	[illegible]	0
0	0	1	1	[illegible]	0
0	1	1	1	[illegible]	0
0	0	1	1	[illegible]	[illegible]
0	0	0	0	[illegible]	[illegible]

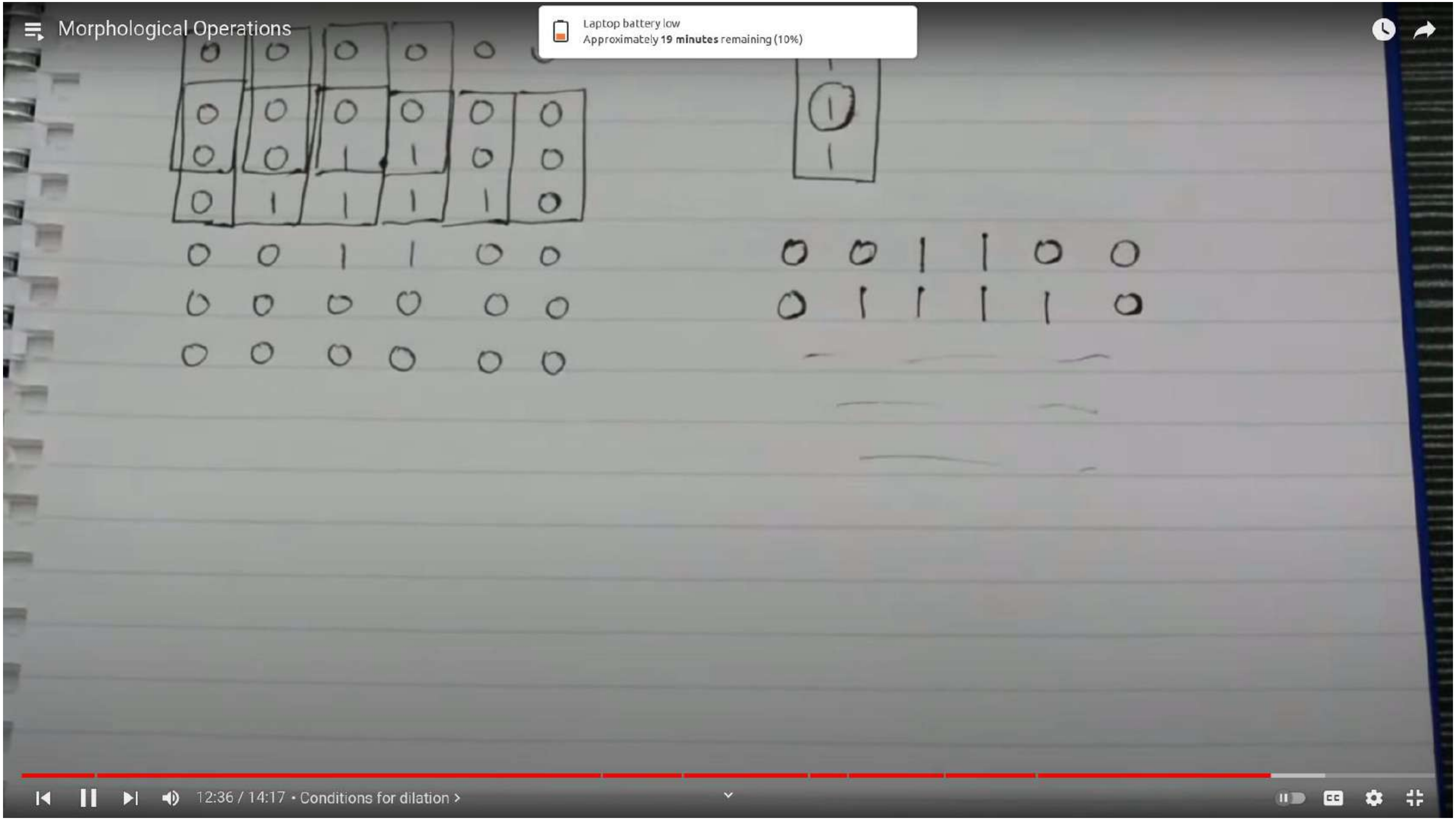
A

1
1
1

B

Ans. First w[illegible]e image A at top and bottom.


Morphological Operations
Ans. First we will pad the image A at top and bottom.
0 0 0 0 0 0
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
0 0 0 0 0 0
A
1
1
1
B
10:09 / 14:17 • Dilation Question ›


Morphological Operations
Laptop battery low
Approximately 19 minutes remaining (10%)
12:36 / 14:17 • Conditions for dilation >

Then we will place B over A and check for following conditions for dilation.

Full match = 1.
Some match or atleast one match = 1
No match = 0.

Output image

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