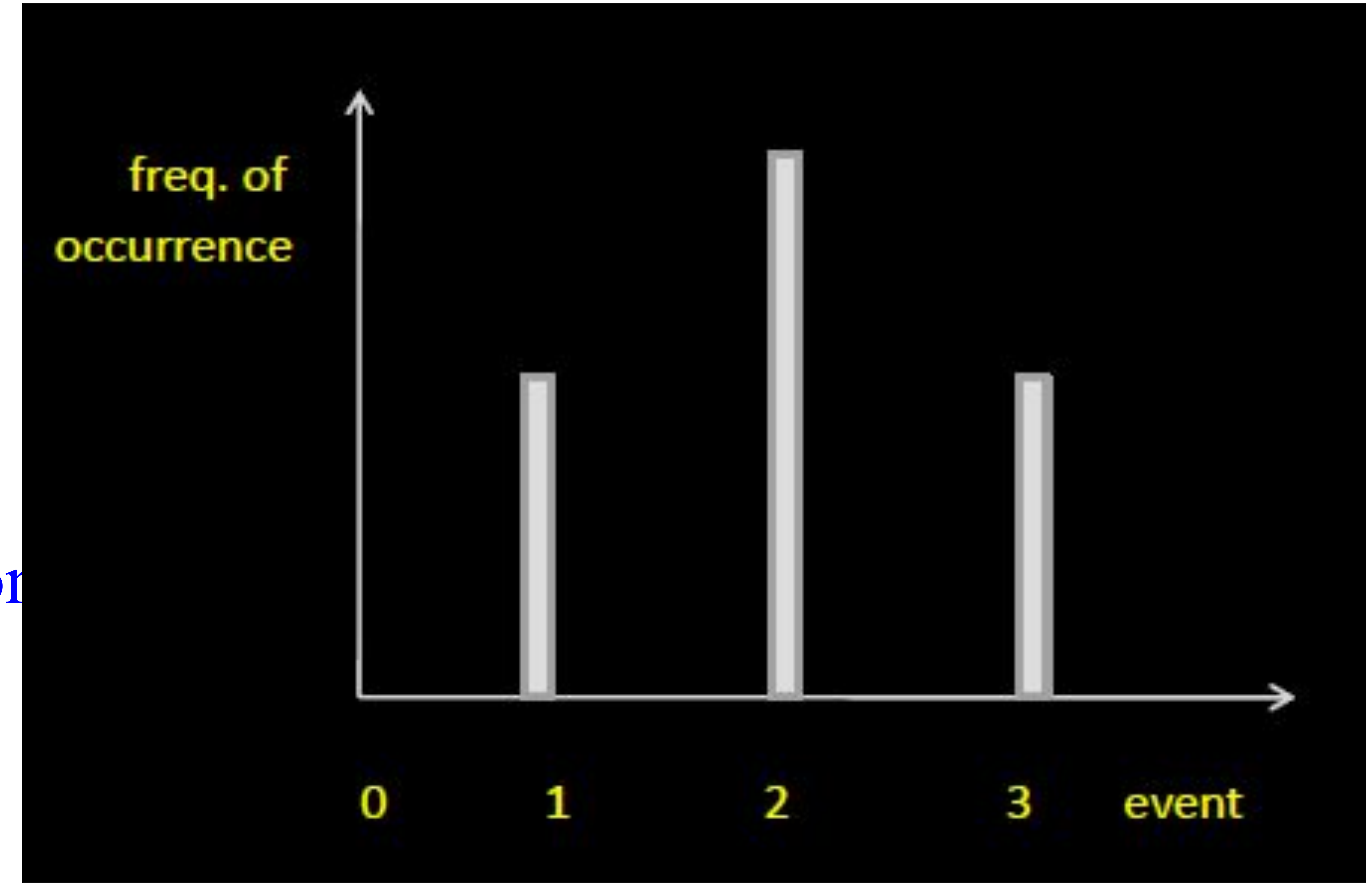
CST 304
Computer Graphics & Image Processing
Module - 5 Part-1

Gray level histogram

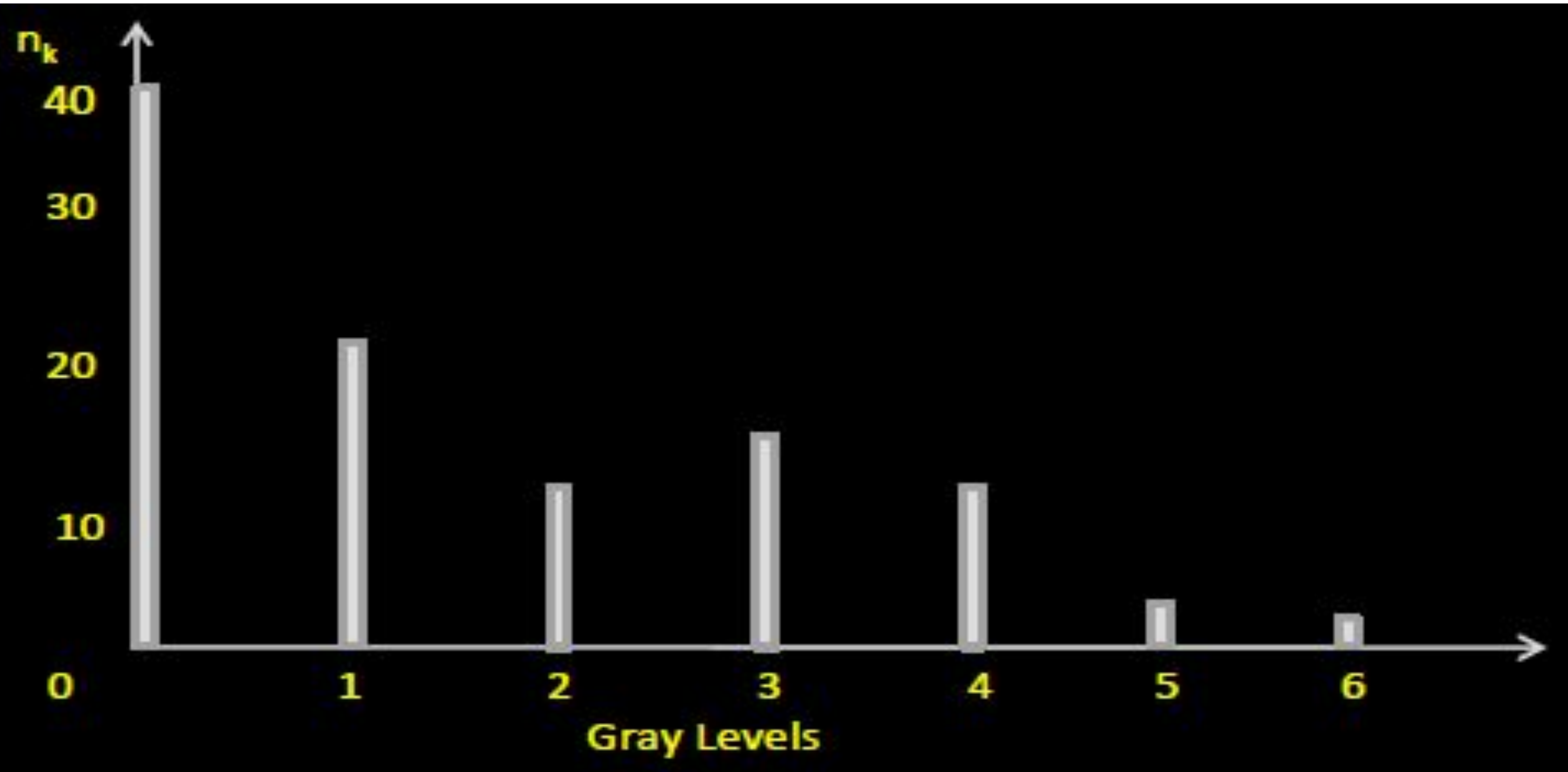
- In Statistics, Histogram is a graphical representation showing a visual impression of the distribution of data.
- An Image Histogram is a type of histogram that acts as a graphical representation of the lightness/color distribution in a digital image.
- It plots the number of pixels for each intensity value.
- Histogram of images provide a global description of their appearance.
- Histogram of an image represents relative frequency of occurrence of various gray levels.



Gray level histogram

- Histogram can be plotted in two ways:
- **First Method:**
  - X-axis has Gray levels & Y-axis has No. of pixels in each gray levels.
  - The histogram of a digital image with gray levels in the range [0, L-1] is a discrete function $\mathbf{h(r_k) = n_k}$, where r_k is the kth gray level and n_k is the number of pixels in the image having gray level r_k.

Gray Level	No. of Pixels (n_k)
0	40
1	20
2	10
3	15
4	10
5	3
6	2



Gray level histogram

- **Second Method:**
  - common practice to normalize a histogram
  - X-axis has gray levels & Y-axis has probability of occurrence of gray levels.

$$P(\mu_k) = n_k / n$$

μ_k – gray level,

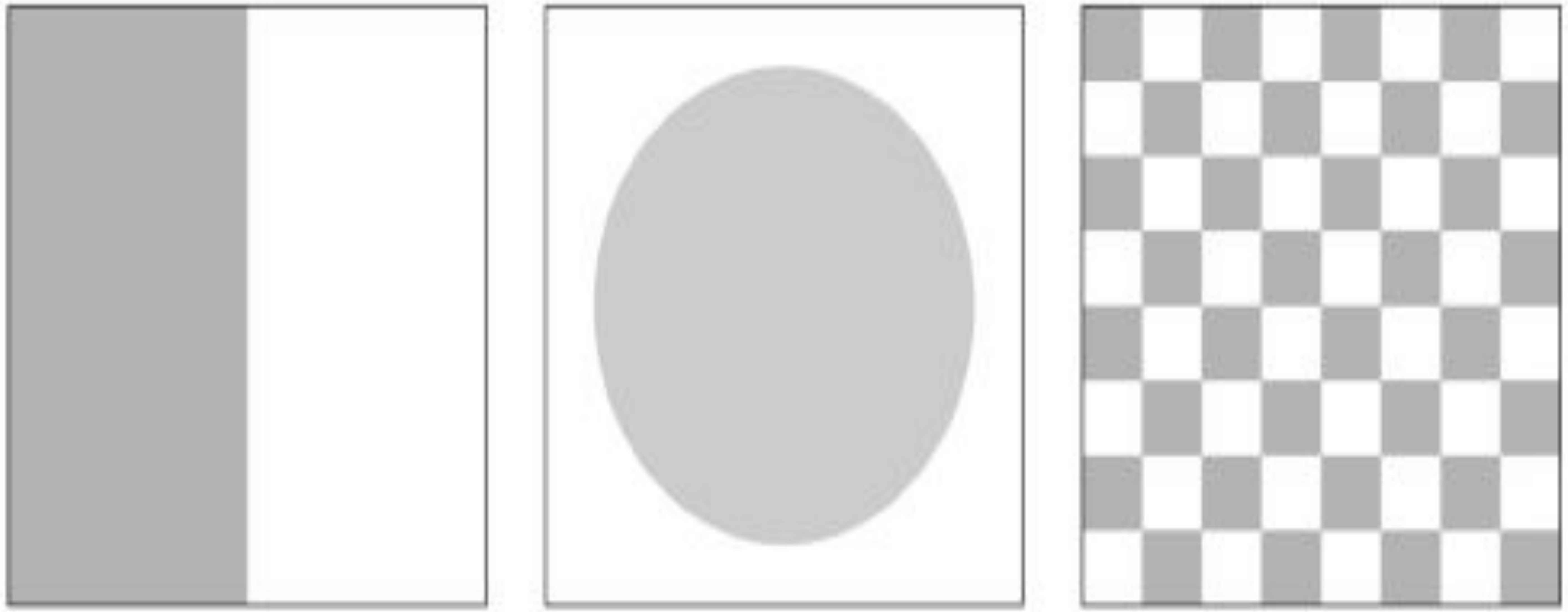
n_k –no. of pixels in kth gray level,

n –total number of pixels in an image

  - sum of all components of a normalized histogram is equal to 1.
  - Advantage of 2nd method: Maximum value plotted will always be 1.
  - White–1, Black–0.

Gray level histogram

- Different images can have same histogram
  - 3 images below have same histogram
  - Half of pixels are gray, half are white, so Same histogram = same statisics
  - Can we reconstruct image from histogram? No !



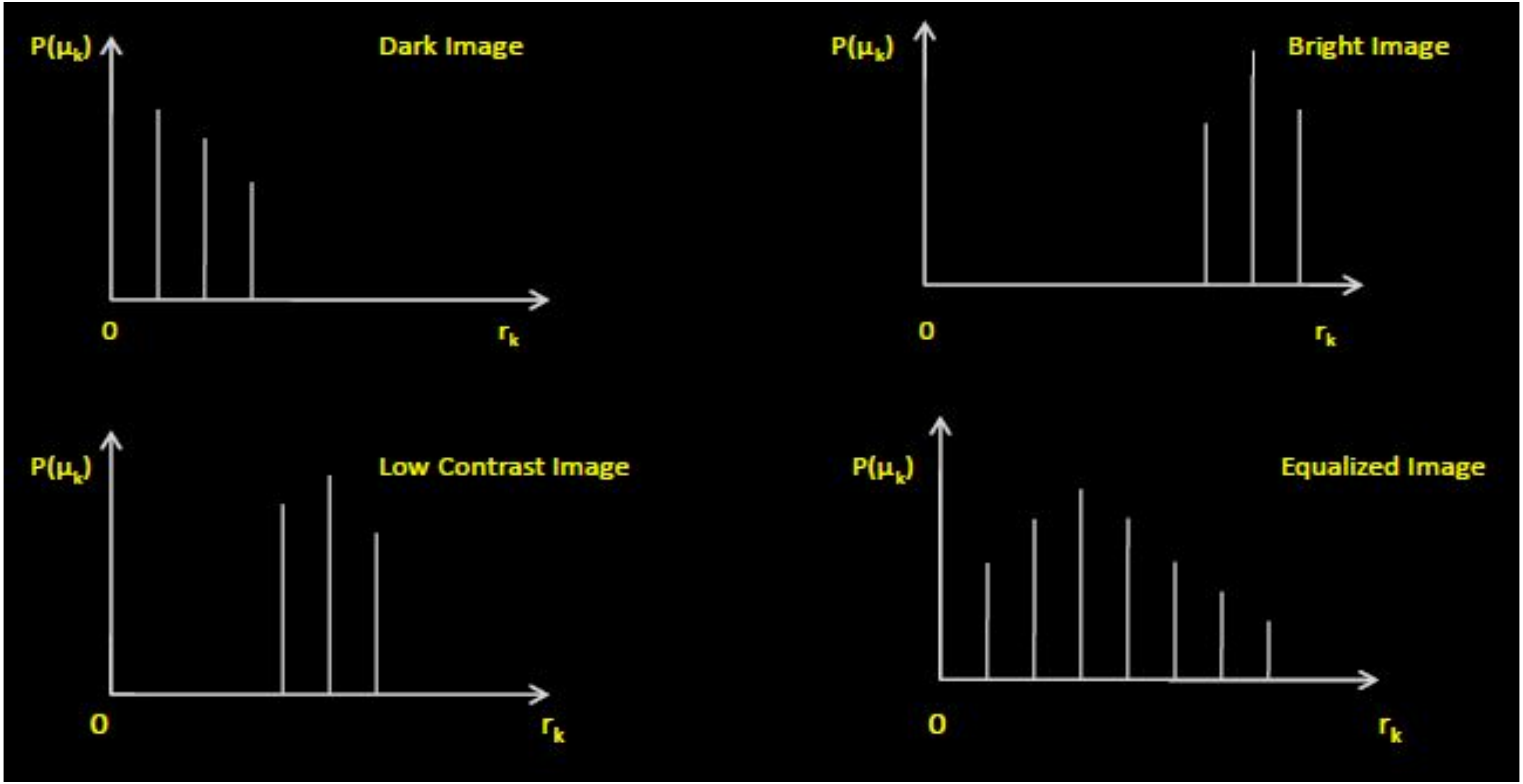
Gray level histogram

Why Histograms?

- Histograms are the basis for numerous spatial domain processing techniques
- Histogram manipulation can be used effectively for image enhancement
- Histograms can be used to provide useful image statistics
- Information derived from histograms are quite useful in other image processing applications, such as image compression and segmentation.

Gray level histogram

Types of Histograms



Gray level histogram

Types of Histograms

- In **dark Image** the components of the histogram are concentrated on the low (dark) side of gray scale
- The components of the histogram of the **bright image** are biased toward the high side of gray scale.
- A **low contrast image** histogram will be narrow and centered towards the middle of the gray scale.
- Components of the histogram in the **high-contrast image** cover a broad range of the gray scale and, further, that the distribution of pixels is not too far from uniform, with very few vertical lines being much higher than the others.
- It is reasonable to conclude that an image whose pixels tend to occupy the entire range of possible gray levels and, in addition, tend to be distributed uniformly, will have an appearance of high contrast and will exhibit a large variety of gray tones. The last graph is the best image; a high contrast image.
- Aim is to transform the first 3 histograms into the 4th type. That is try to increase the dynamic range of the image .This is called **Histogram Processing**

Gray level histogram

Histogram Processing

- There are two methods of enhancing contrast.

✔ Histogram Stretching

✔ Histogram Equalization

Histogram Processing - Histogram Stretching

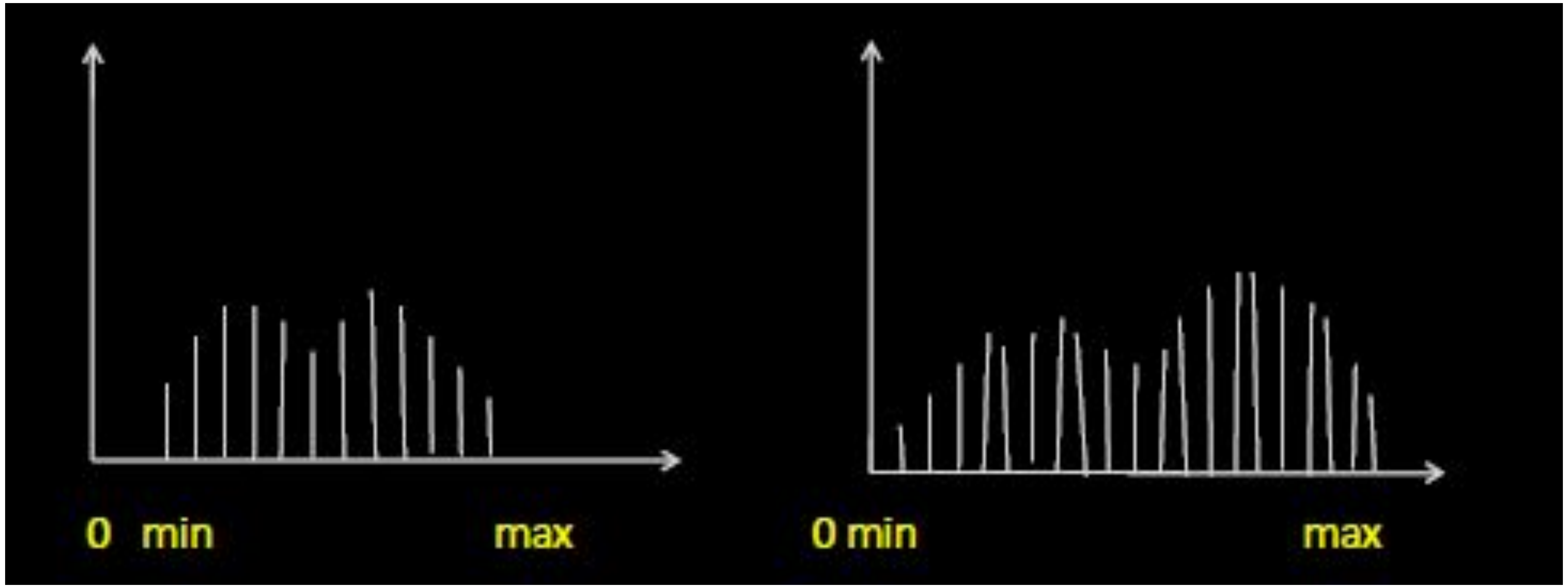
- Process of histogram stretching increases the dynamic range of the image and hence improves the contrast of the image.
- In this process, the basic shape of the histogram is not modified, but the entire range of image of histogram values is stretched.
- Before stretching process, it is necessary to specify the upper and lower limits for pixels of normalized range.
- Limit values depends on type of image. (eg: 8 bit image ,limit values ranges between 0 & 255)
- Simplest histogram function: $S = T(r) = ((S_{max} - S_{min}) / (r_{max} - r_{min}))\,(r - r_{min}) + S_{min}$

S_{max} – maximum limit value of image,

S_{min} – minimum limit value of image,

r_{max} – highest pixel value of image,

r_{min} – lowest pixel value of image



Histogram Processing - Histogram Stretching

Ex. 1) Perform Histogram Stretching so that the new image has a dynamic range of 0 to 7 [0, 7].

Gray Levels	0	1	2	3	4	5	6	7
No. of Pixels	0	0	50	60	50	20	10	0

Sol^n:- $r_{min} = 2$; $r_{max} = 6$; $s_{min} = 0$; $s_{max} = 7$;

slope = $((s_{max} - s_{min})/(r_{max} - r_{min})) = ((7 - 0)/(6 - 2)) = 7 / 4 = 1.75$.

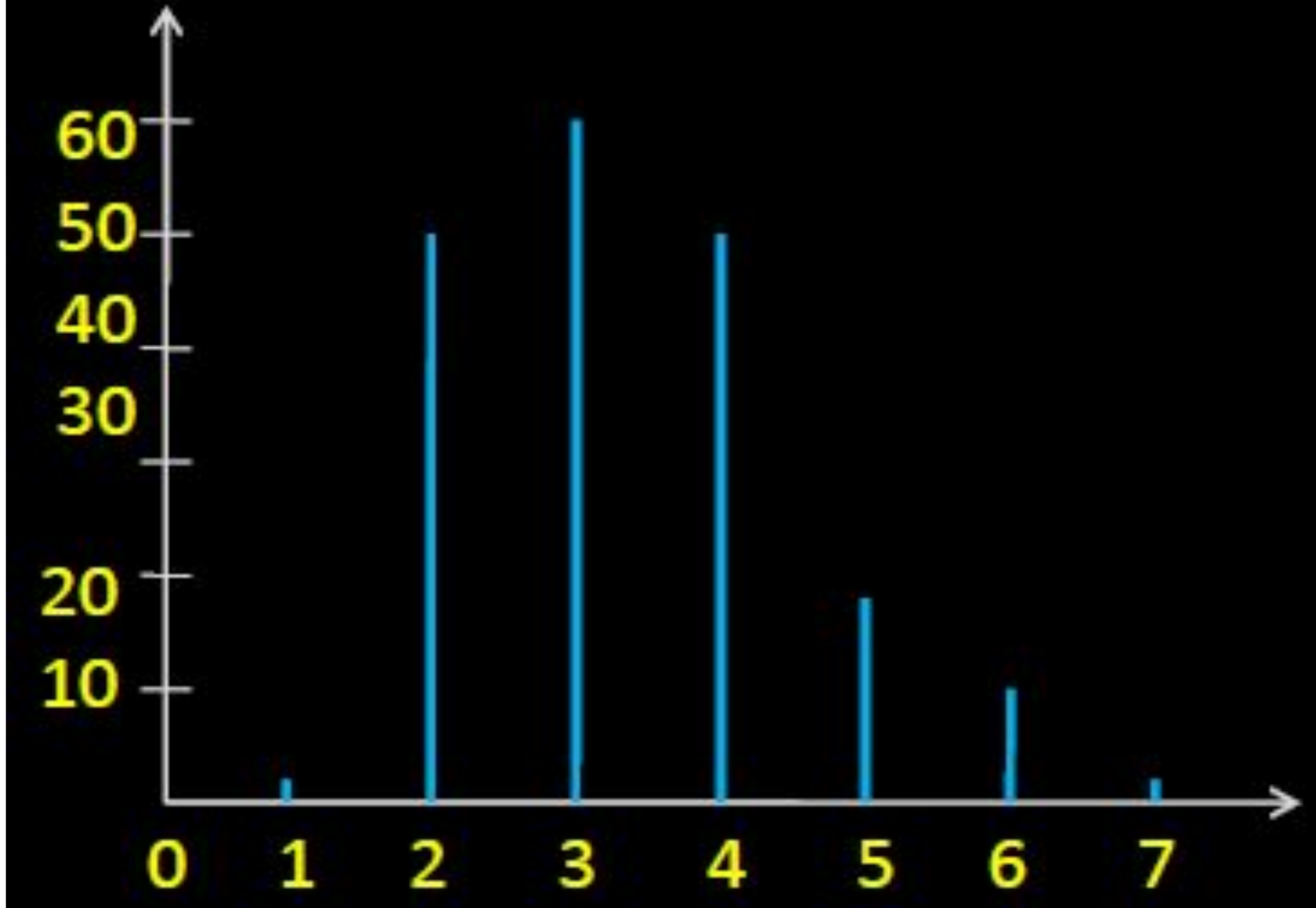
$S = (7 / 4)(r - 2) + 0;$

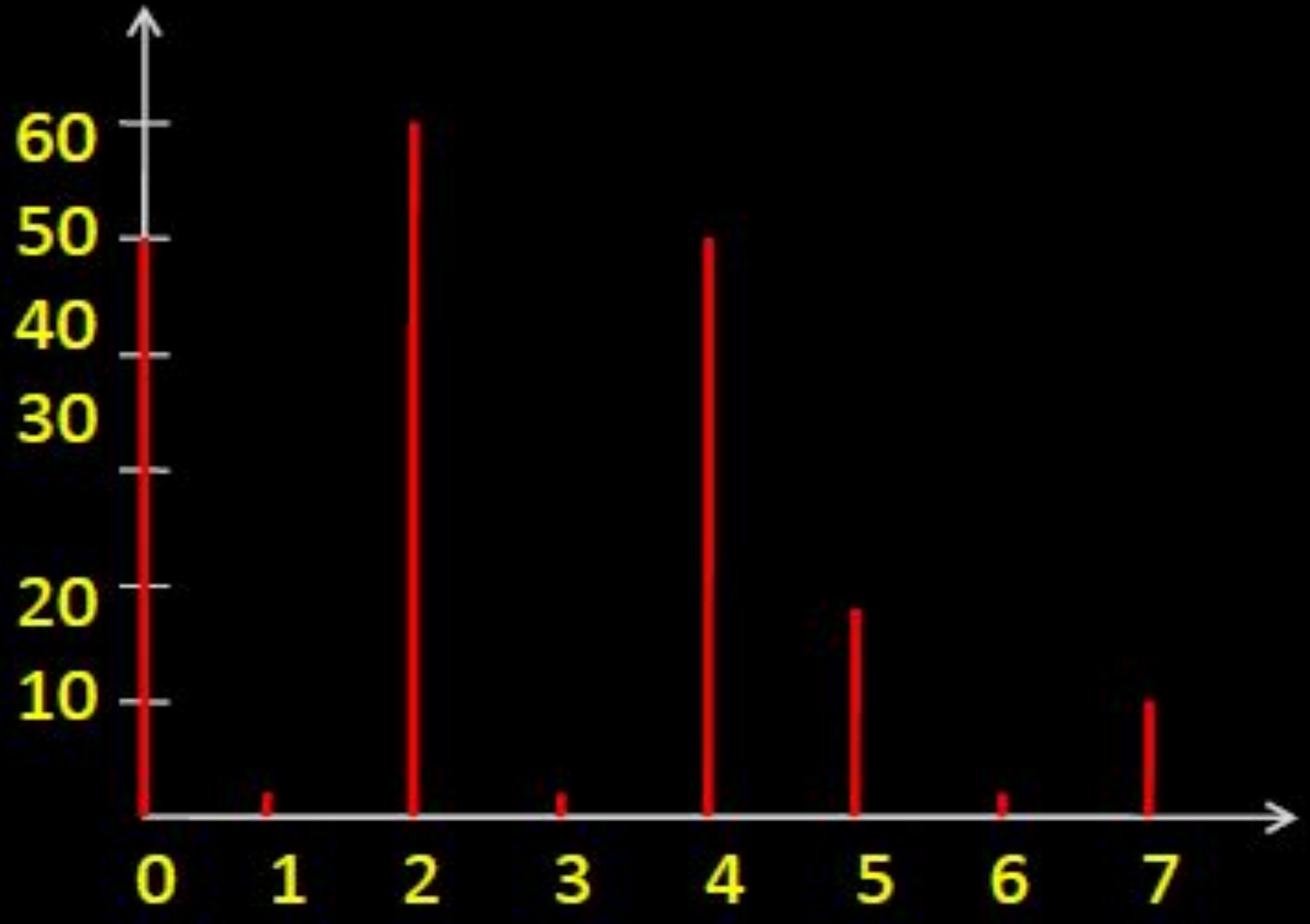
$S = (7 / 4)(r - 2)$

r	(7 / 4)(r - 2) = S
2	0 = 0
3	7/4 = 1.75 = 2
4	7/2 = 3.5 = 4
5	21/4 = 5.25 = 5
6	7 = 7

Histogram Processing - Histogram Stretching

Gray Levels	0	1	2	3	4	5	6	7
No. of Pixels	50	0	60	0	50	20	0	10





Histogram Processing - Histogram Equalization

- **Equalization** is a process that attempts to spread out the gray levels in an image so that they are evenly distributed across their range.
- It reassigns the brightness values of pixels based on image histogram.
- The histogram of resultant image is made as flat as possible.
- Provides more visually pleasing results across a wide range of images.
- Steps to perform Histogram Equalization:-
  - Find the running sum of histogram values.
  - Normalise the values from step1 by dividing by the total number of pixels
  - Multiply the values from step 2 by the maximum gray level value and round
  - Map the gray level values to the results from step 3 using a one to one correspondence.

Histogram Processing - Histogram Equalization

- Example:

Q. Perform histogram equalization of the image.

- Maximum value is found to be 5. We need a minimum of 3 bits to represent the number.
- There are 8 possible gray levels from 0 to 7
- Histogram of input image:-

$$\begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 3 & 4 & 5 & 4 & 3 \\ 3 & 5 & 5 & 5 & 3 \\ 3 & 4 & 5 & 4 & 3 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix}$$

Gray level	0	1	2	3	4	5	6	7
No of pixels	0	0	0	6	14	5	0	0

Histogram Processing - Histogram Equalization

Step1: Compute the running sum of histogram values.

- Running sum is otherwise known as Cumulative Frequency Distribution(CFD).

Gray level	0	1	2	3	4	5	6	7
No of pixels	0	0	0	6	14	5	0	0
CFD	0	0	0	6	20	25	25	25

Gray level	0	1	2	3	4	5	6	7
No of pixels	0	0	0	6	14	5	0	0
CFD	0	0	0	6	20	25	25	25

Histogram Processing - Histogram Equalization

Step 2 : Divide the running sum obtained in step1 by the total number of pixels (which is 25 in this case).

Gray level	0	1	2	3	4	5	6	7
No of pixels	0	0	0	6	14	5	0	0
CFD	0	0	0	6	20	25	25	25
CFD/ total no of pixels	0/25	0/25	0/25	6/25	20/25	25/25	25/25	25/25

Histogram Processing - Histogram Equalization

Step3: Multiply the values from step 2 by the maximum gray level value (which is 7 in this case). The result is then rounded to closest integer.

Gray level	0	1	2	3	4	5	6	7
No of pixels	0	0	0	6	14	5	0	0
CFD	0	0	0	6	20	25	25	25
CFD/ total no of pixels	0/25	0/25	0/25	6/25	20/25	25/25	25/25	25/25
Multiply the values by max gray level	0/25*7	0/25*7	0/25*7	6/25*7	20/25*7	25/25*7	25/25*7	25/25*7
Rounded value	0	0	0	2	6	7	7	7

Histogram Processing - Histogram Equalization

Step 4: Map the gray level values to the results from step 3 using a one to one correspondence

Original gray level	Histogram equalized level
0	0
1	0
2	0
3	2
4	6
5	7
6	7
7	7

Original image

$$\begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 3 & 4 & 5 & 4 & 3 \\ 3 & 5 & 5 & 5 & 3 \\ 3 & 4 & 5 & 4 & 3 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix}$$

Histogram equalized image

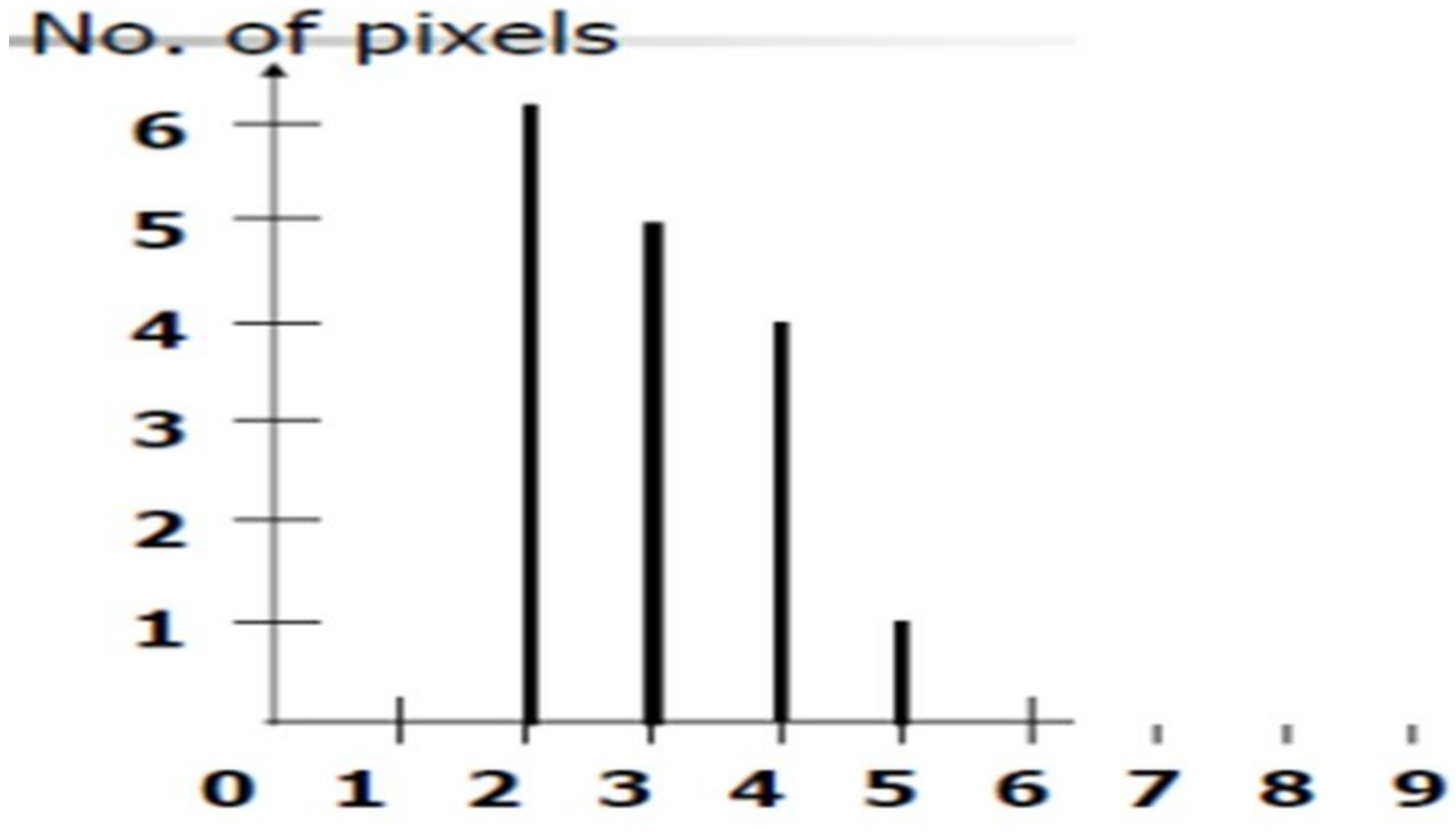
$$\begin{bmatrix} 6 & 6 & 6 & 6 & 6 \\ 2 & 6 & 7 & 6 & 2 \\ 2 & 7 & 7 & 7 & 2 \\ 2 & 6 & 7 & 6 & 2 \\ 6 & 6 & 6 & 6 & 6 \end{bmatrix}$$

Tutorial :- Histogram Equalization

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

4x4 image

Gray scale = [0,9]



histogram

Gray Level(j)	0	1	2	3	4	5	6	7	8	9
No. of pixels	0	0	6	5	4	1	0	0	0	0

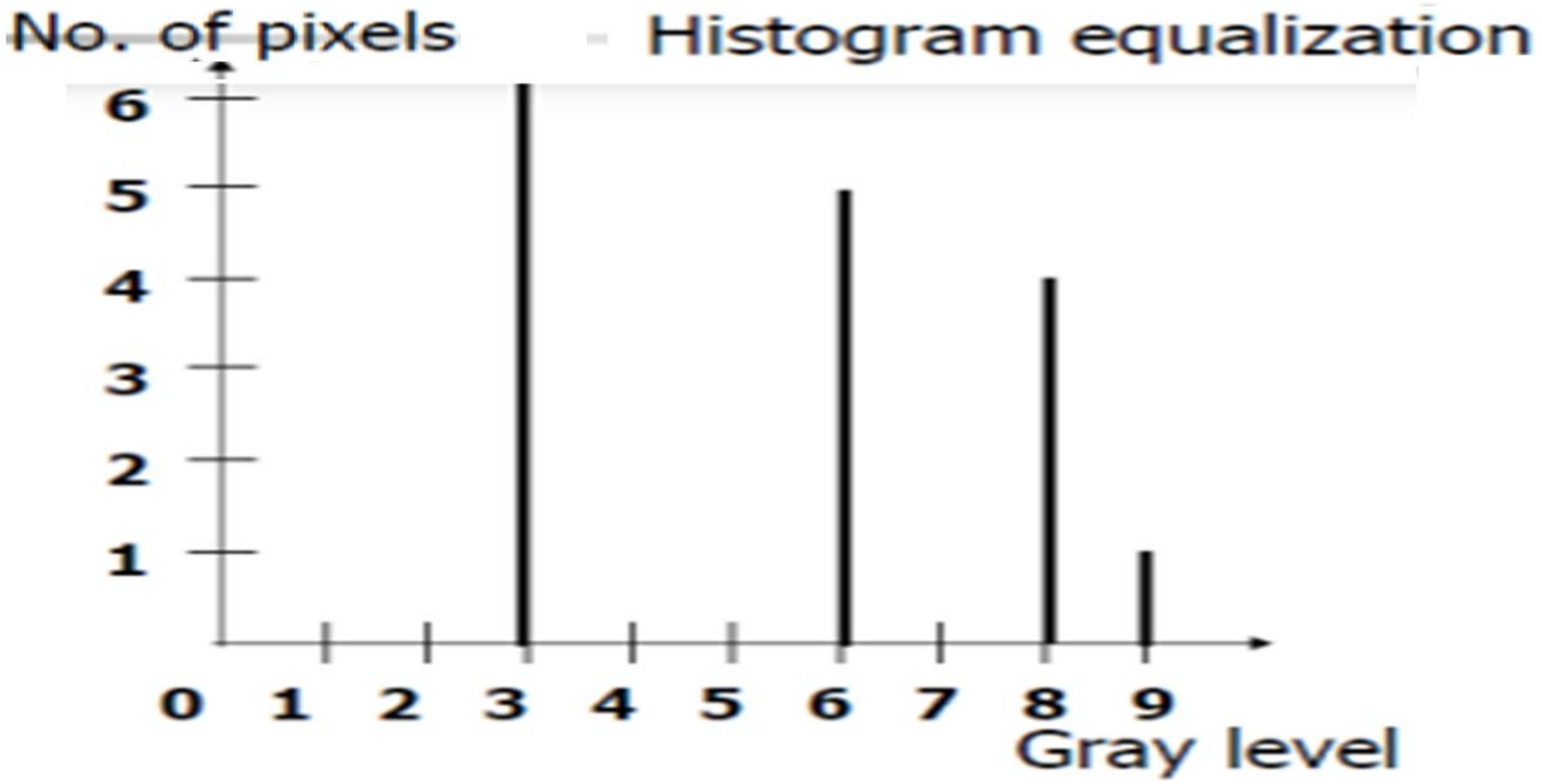
Tutorial :- Histogram Equalization

Gray Level(j)	0	1	2	3	4	5	6	7	8	9
No. of pixels	0	0	6	5	4	1	0	0	0	0
$\sum_{j=0}^{k} n_j$	0	0	6	11	15	16	16	16	16	16
$s = \sum_{j=0}^{k} \frac{n_j}{n}$	0	0	6 / 16	11 / 16	15 / 16	16 / 16	16 / 16	16 / 16	16 / 16	16 / 16
s x 9	0	0	3.3 ≈3	6.1 ≈6	8.4 ≈8	9	9	9	9	9

3	6	6	3
8	3	8	6
6	3	6	9
3	8	3	8

Output image

Gray scale = [0,9]



Smoothing Linear Filters

- Smoothing
  - elimination of high frequencies
  - usually performed using averaging (either weighted or non-weighted)
- Example: Smoothing using average 3x3 filters

$$H=\frac{1}{9}\begin{pmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{pmatrix}$$

non-weighted

$$H=\frac{1}{10}\begin{pmatrix}1 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 1\end{pmatrix}$$

weighted

$$H=\frac{1}{16}\begin{pmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{pmatrix}$$

weighted

Neighborhood processing in spatial domain: Here, to modify one pixel, values of the immediate neighboring pixels also considered. For this purpose, 3X3, 5X5, or 7X7 neighborhood mask can be considered.

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

Low Pass filtering: It is also known as the smoothing filter. It removes the high-frequency content from the image. It is also used to blur an image.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

High Pass Filtering: It eliminates low-frequency regions while retaining or enhancing the high-frequency components

A high pass filtered image may be computed as the difference between the original image and a low pass filtered version of that image as follows

High pass = Original – Low pass

Multiplying the original by an amplification factor yields a high boost or high frequency-emphasis

$$
\begin{aligned}
Highboost &= A(Original) - Lowpass \\
&= (A-1)(Original) + Original - Lowpass \\
&= (A-1)(Original) + Highpass
\end{aligned}
$$

A high pass filtering mask is as shown.

-1/9	-1/9	-1/9
-1/9	8/9	-1/9
-1/9	-1/9	-1/9

Median Filtering: It is also known as nonlinear filtering. It is used to eliminate salt and pepper noise. Here the pixel value is replaced by the median value of the neighboring pixel.

SPATIAL CONVOLUTION AND CORRELATION

Spatial filtering

- Filtering refers to accepting(passing) or rejecting certain frequency components. This effectively smoothens or sharpens the image.
  - E.g. Low pass filter, high pass filter, etc.
- If the operation performed on the image pixels is linear, then the filter is called a linear spatial filter, otherwise nonlinear.

Spatial Filters

Spatial filters can be classified by effect into:

Smoothing Spatial Filters: also called lowpass filters. They include:

Averaging linear filters

Order-statistics nonlinear filters.

Sharpening Spatial Filters: also called highpass filters.

eg: Laplacian linear filter.

Spatial Filters

Smoothing Spatial Filters

Used for blurring and for noise reduction.

Blurring is used in preprocessing steps to:

- remove small details from an image prior to (large) object extraction
- bridge small gaps in lines or curves.

Noise reduction can be accomplished by blurring with a linear filter and also by nonlinear filtering.

Averaging linear filters

The response of averaging filter is simply the average of the pixels contained in the neighborhood of the filter mask.

The output of averaging filters is a smoothed image with reduced "sharp" transitions in gray levels.

Noise and edges consist of sharp transitions in gray levels. Thus smoothing filters are used for noise reduction; however, they have the undesirable side effect that they blur edges.

Linear Spatial Filtering (Convolution)

The process consists of moving the filter mask from pixel to pixel in an image. At each pixel (x,y), the response is given by a sum of products of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask.

For the 3×3 mask shown in the previous figure, the result (or response), R, of linear filtering is:

$$R = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \cdots$$

$$+w(0,0)f(x,y) + \cdots + w(1,0)f(x+1,y) + w(1,1)f(x+1,y+1)$$

In general, linear filtering of an image *f* of size *M*× *N* with a filter mask of size *m*× *n* is given by the expression:

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

where a = (*m* - 1)/2 and b = (*n* - l)/2. To generate a complete filtered image this equation must be applied for x = 0,1, 2,..., *M*-1 and y = 0,1, 2,..., *N*-1.

Nonlinear Spatial Filtering

- This operation also consists of moving the filter mask from pixel to pixel in an image.
- The filtering operation is based conditionally on the values of the pixels in the neighborhood, and they do not explicitly use coefficients in the sum-of-products manner.
- For example, noise reduction can be achieved effectively with a nonlinear filter whose basic function is to compute the median gray-level value in the neighborhood in which the filter is located.
- Computation of the median is a nonlinear operation.

Linear Spatial Filtering

Use the following 3×3 mask to perform the convolution process on the shaded pixels in the 5×5 image below. Write the filtered image.

0	1/6	0
1/6	1/3	1/6
0	1/6	0

3×3 mask

30	40	50	70	90
40	50	80	60	100
35	255	70	0	120
30	45	80	100	130
40	50	90	125	140

5×5 image

Solution:

$$0 \times 30 + \frac{1}{6} \times 40 + 0 \times 50 + \frac{1}{6} \times 40 + \frac{1}{3} \times 50 + \frac{1}{6} \times 80 + 0 \times 35 + \frac{1}{6} \times 255 + 0 \times 70 = 85$$

$$0 \times 40 + \frac{1}{6} \times 50 + 0 \times 70 + \frac{1}{6} \times 50 + \frac{1}{3} \times 80 + \frac{1}{6} \times 60 + 0 \times 255 + \frac{1}{6} \times 70 + 0 \times 0 = 65$$

$$0 \times 50 + \frac{1}{6} \times 70 + 0 \times 90 + \frac{1}{6} \times 80 + \frac{1}{3} \times 60 + \frac{1}{6} \times 100 + 0 \times 70 + \frac{1}{6} \times 0 + 0 \times 120 =$$

$$0 \times 40 + \frac{1}{6} \times 50 + 0 \times 80 + \frac{1}{6} \times 35 + \frac{1}{3} \times 255 + \frac{1}{6} \times 70 + 0 \times 30 + \frac{1}{6} \times 45 + 0 \times 80 = 118$$

and so on ...

Filtered image =

30	40	50	70	90
40	**85**	**65**	**61**	100
35	**118**	**92**	**58**	120
30	**84**	**77**	**89**	130
40	50	90	125	140

Weighted average filter has different coefficients to give more importance (weight) to some pixels at the expense of others. The idea behind that is to reduce blurring in the smoothing process.

Averaging linear filtering of an image *f* of size *M*× *N* with a filter mask of size *m*× *n* is given by the expression given here.

To generate a complete filtered image this equation must be applied for

$x = 0,1, 2,..., M-1$ and $y = 0,1, 2,..., N-1$.

$$g(x,y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)}$$

Effects of Averaging linear filter

Figure 6.2 Effect of averaging filter. (a) Original image. (b)-(f) Results of smoothing with square averaging filter masks of sizes n = 3,5,9,15, and 35, respectively.

SPATIAL CONVOLUTION AND CORRELATION

Correlation & Convolution

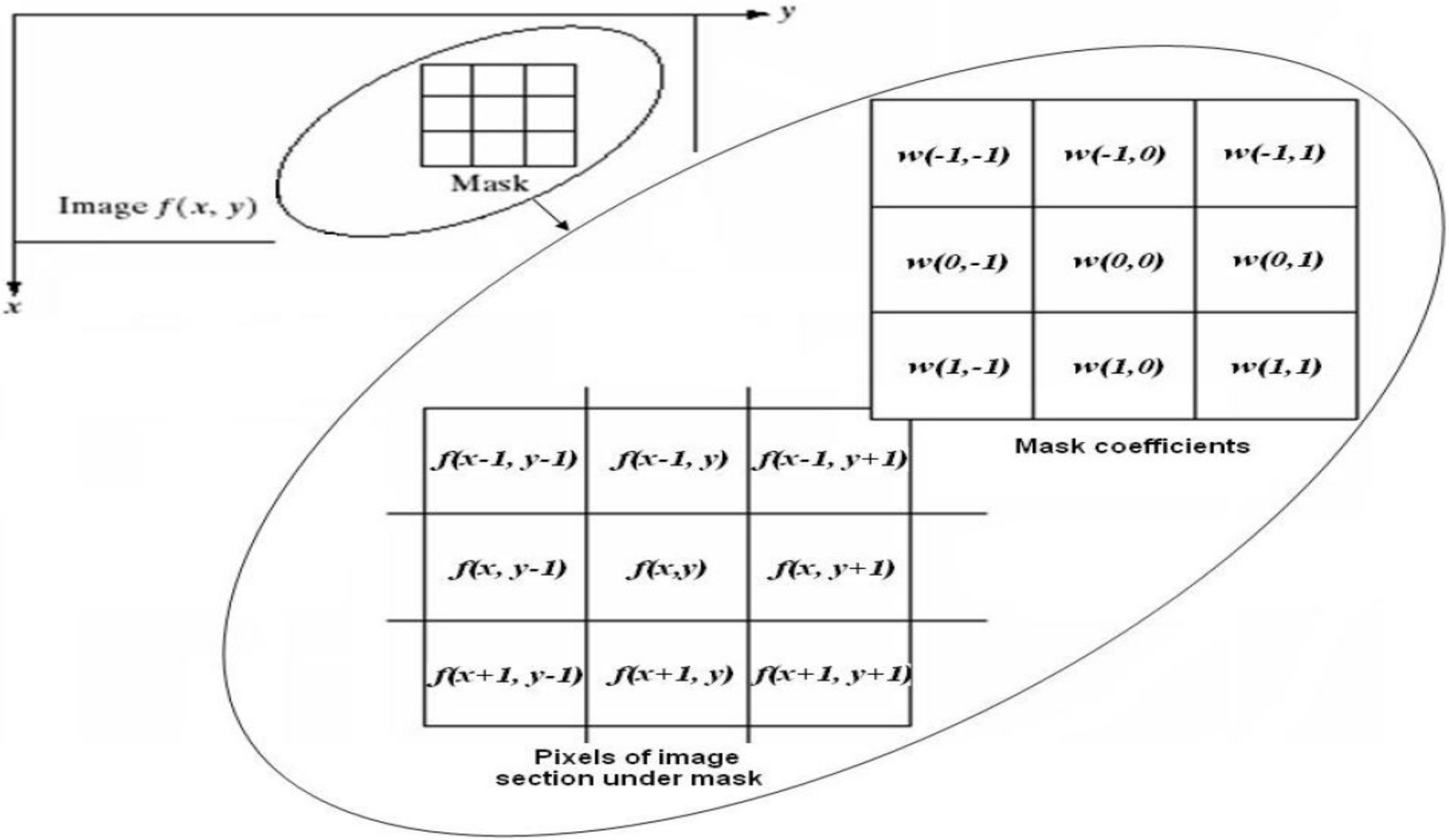
- Correlation & Convolution are two closely related concepts used in linear spatial filtering.
- Both are used to extract information from images.
- Both are basically linear and shift invariant operations.
- Term linear indicates that a pixel is replaced by the linear combination of its neighbours.
- Term shift variant means that same operations are performed at every point in the image.
- Correlation: It is a process of moving a filter mask over an image & computing the sum of products at each location.
- Convolution: mechanics are same, except that the filter is first rotated by 180°.

SPATIAL CONVOLUTION AND CORRELATION

Spatial convolution

- Method which replaces a pixel with the weighted average of the pixel and its neighbours of the convolution mask.
- Convolution is basically mathematical operation where each value in the output is expressed as the sum of values in the input multiplied by a set of weighting coefficients.
- An image can be either smoothened or sharpened by convolving the image with respect to low pass and high pass filter respectively.
- Application areas - image filtering, image segmentation, feature extraction etc.

The mechanism of linear spatial filtering



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

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

Linear Spatial Filtering

- For linear spatial filtering, the response is given by a sum of products of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask. For the 3 X 3 mask shown in the previous figure, the result (response), of linear filtering with the filter mask at a point (x,y) in the image is:

$$g(x,y) = w(-1,-1)\, f(x-1, y-1) + w(-1,0)\, f(x-1, y) + \dots + w(0,0)\, f(x,y) + \dots + w(1,0)\, f(x+1, y) + w(1,1)\, f(x+1, y+1)$$

which we see is the sum of products of the mask coefficients with the corresponding pixels directly under the mask.

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t)$$

Original image

	a	b	c		
	d	x	f		
	g	h	i		

$\otimes$

Filter mask

1	3	1
2	4	-2
-3	1	4

1a	3b	1c
2d	4x	-2f
-3g	1h	4i

a	b	c
d	y	f
g	h	i

Output image

$$y = 1a+3b+1c+2d+4x-2f-3g+1h+4i$$

Linear Spatial Filtering

- There are two closely related concepts that must be understood clearly when performing linear spatial filtering.
- One is correlation; the other is convolution.
- Correlation is the process of passing the filter mask w by the image and computing the sum of products at each location
- Mechanically, convolution is the same process, except that w is rotated by 180 degrees prior to passing it by f.

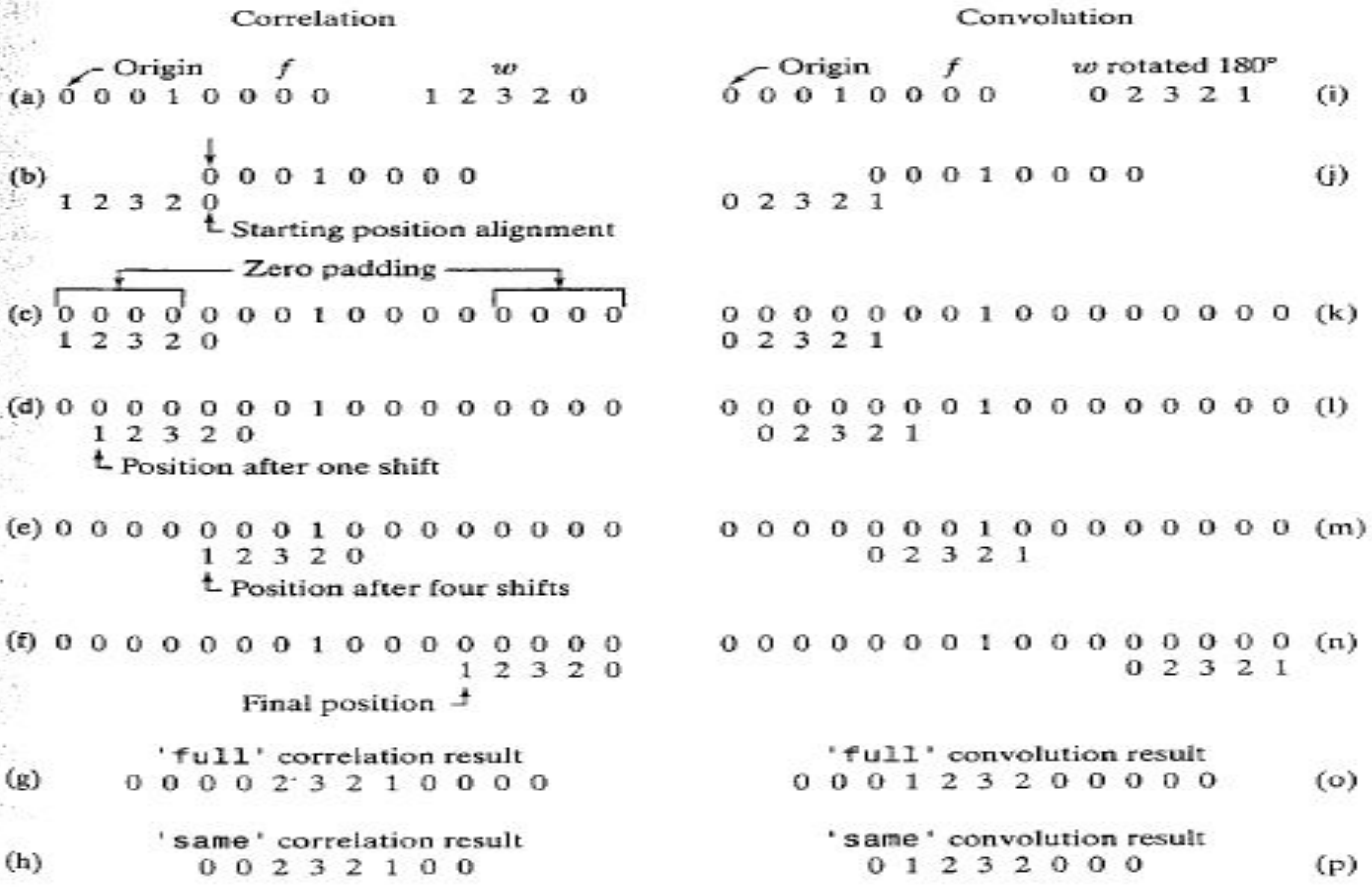


FIGURE 3.13 Illustration of one-dimensional correlation and convolution.

Origin of $f(x, y)$

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

$w(x, y)$

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

(a)

Padded f

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

(b)

Initial position for w

$$\begin{matrix} 1&2&3&0&0&0&0&0&0\\ 4&5&6&0&0&0&0&0&0\\ 7&8&9&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0 \end{matrix}$$

(c)

'full' correlation result

$$\begin{matrix} 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&9&8&7&0&0&0\\ 0&0&0&6&5&4&0&0&0\\ 0&0&0&3&2&1&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0 \end{matrix}$$

(d)

'same' correlation result

$$\begin{matrix} 0&0&0&0&0\\ 0&9&8&7&0\\ 0&6&5&4&0\\ 0&3&2&1&0\\ 0&0&0&0&0 \end{matrix}$$

(e)

Rotated w

$$\begin{matrix} 9&8&7&0&0&0&0&0&0\\ 6&5&4&0&0&0&0&0&0\\ 3&2&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0 \end{matrix}$$

(f)

'full' convolution result

$$\begin{matrix} 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&1&2&3&0&0&0\\ 0&0&0&4&5&6&0&0&0\\ 0&0&0&7&8&9&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0 \end{matrix}$$

(g)

'same' convolution result

$$\begin{matrix} 0&0&0&0&0\\ 0&1&2&3&0\\ 0&4&5&6&0\\ 0&7&8&9&0\\ 0&0&0&0&0 \end{matrix}$$

(h)

FIGURE 3.14 Illustration of two-dimensional correlation and convolution. The 0s are shown in gray to simplify viewing.

Exercise - Spatial convolution

F = { 0,0,2,0,0} and the mask or Kernel is { 7 5 1 }
Perform the Convolution Process ?

In Convolution Process we have to rotate the Kernel by 180°

Zero padding process for convolution

1	5	7						
0	0	0	0	2	0	0	0	0

$(1*0)+(5*0)+(7*0) = 0$

Convolution process

(a) Initial position

Template

1 5 7

0 0 | 0 0 2 0 0 | 0 0

0

Output is produced in the centre pixel.

(b) Position after one shift

Template is shifted by one bit.

1 5 7

0 0 | 0 0 2 0 0 | 0 0

0 0

Output produced is zero.

(c) Position after two shifts

Template is shifted again.

1 5 7

0 0 | 0 0 2 0 0 | 0 0

0 0 14

Output produced is 14.

(d) Position after three shifts

Template is shifted again.

1 5 7

0 0 | 0 0 2 0 0 | 0 0

0 0 14 10

Output produced is 10.

$(1*0)+(5*0)+(7*2) = 14$

(e) Position after four shifts

Template is shifted again.

1 5 7

0 0 0 0 2 0 0 0 0

0 0 14 10 2

Output produced is 2.

(f) Position after five shifts

Template is shifted again.

1 5 7

0 0 0 0 2 0 0 0 0

0 0 14 10 2 0

Output produced is 0.

(g) Final position

Template is shifted again.

1 5 7

0 0 0 0 2 0 0 0 0

0 0 14 10 2 0 0

Output produced is 0. Further shift crosses the range. Hence the process is stopped.

Correlation process

(a) Initial position

Template

7	5	1						
0	0	0	0	2	0	0	0	0
	0							

Output produced is 0.

(b) Position after one shift

Template is shifted by one bit.

	7	5	1					
0	0	0	0	2	0	0	0	0
	0	0						

Output produced is 0.

(c) Position after two shifts

Template is shifted again.

		7	5	1				
0	0	0	0	2	0	0	0	0
	0	0	2					

Output produced is 2.

(d) Position after three shifts

Template is shifted again.

			7	5	1			
0	0	0	0	2	0	0	0	0
	0	0	2	10				

Output produced is 10.

(e) Position after four shifts

Template is shifted again.

				7	5	1		
0	0	0	0	2	0	0	0	0
	0	0	2	10	14			

Output produced is 14.

(f) Position after five shifts

Template is shifted again.

					7	5	1	
0	0	0	0	2	0	0	0	0
	0	0	2	10	14	0		

Output produced is 0.

(g) Final position

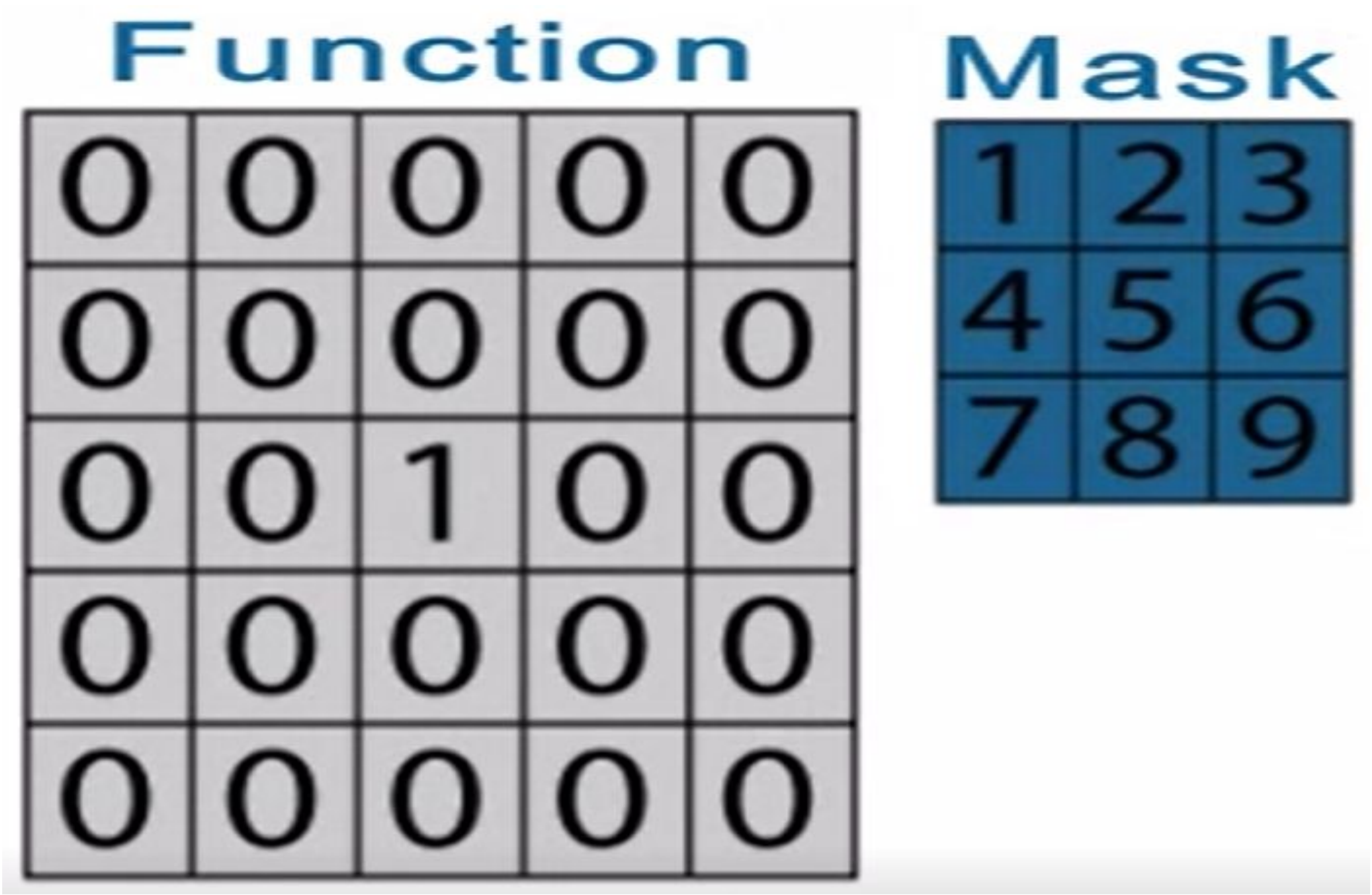
Template is shifted again.

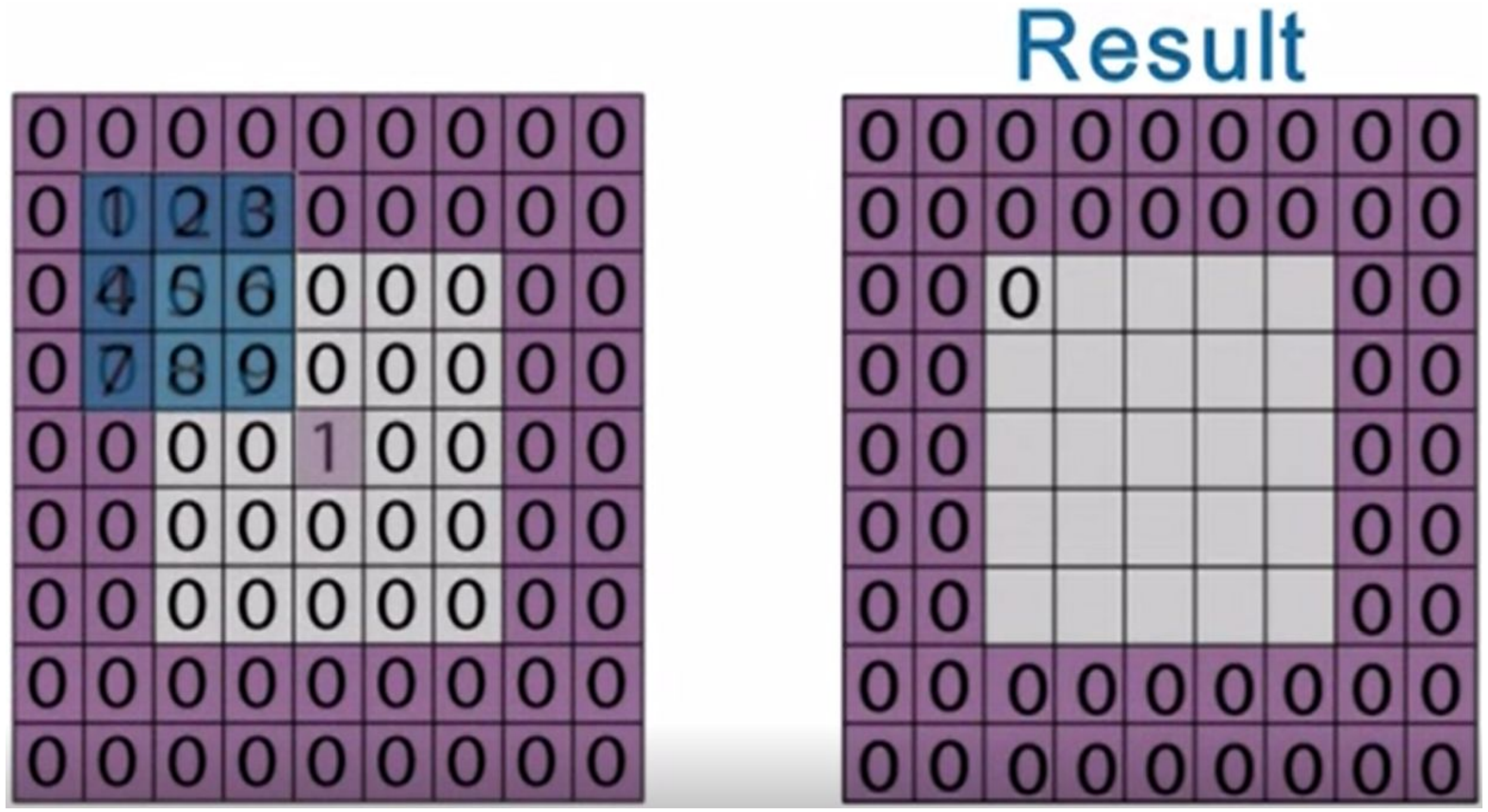
						7	5	1
0	0	0	0	2	0	0	0	0
	0	0	2	10	14	0	0	

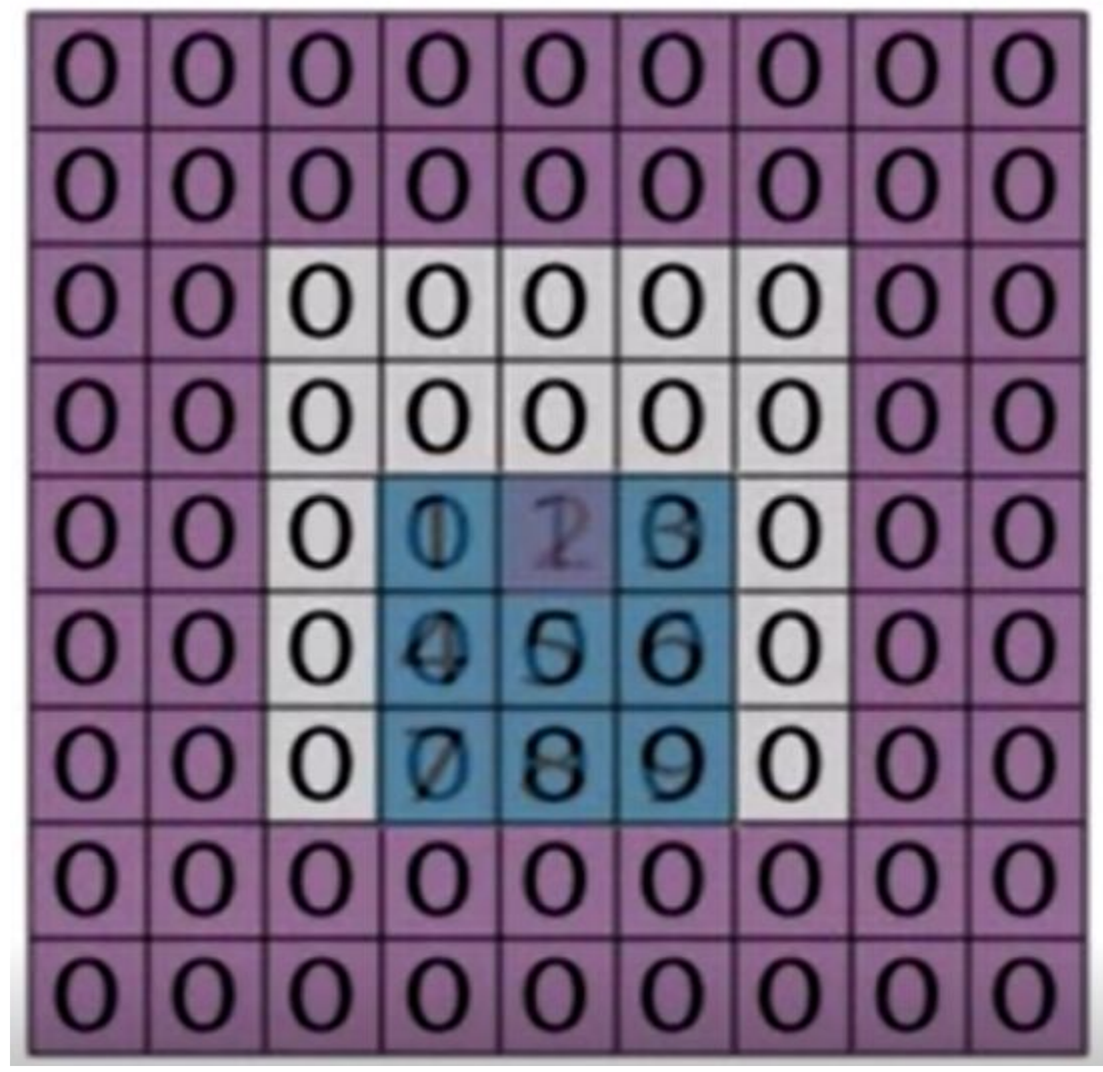
Output produced is 0.

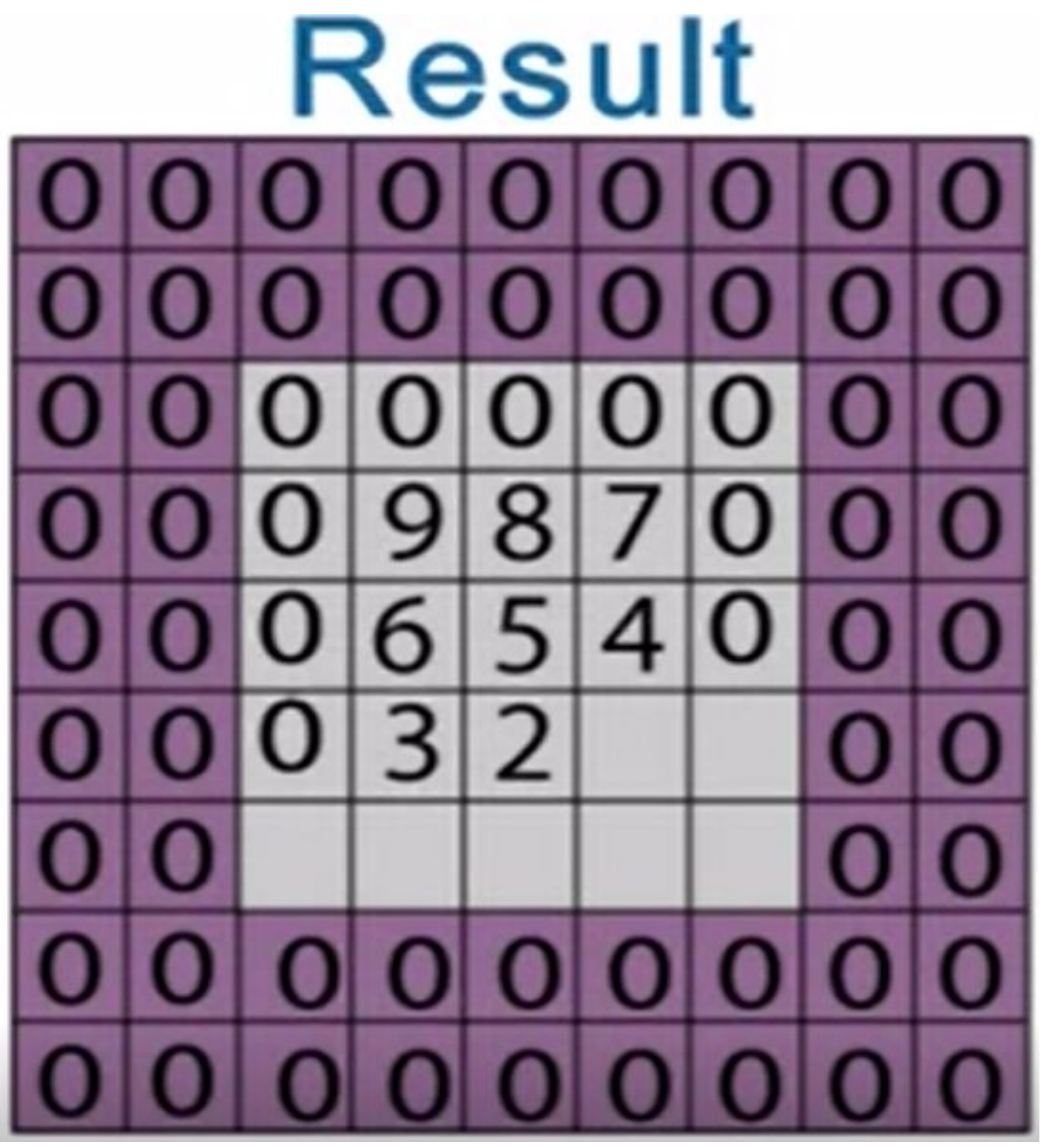
So in the final position, the output produced is [0 0 2 10 14 0 0].

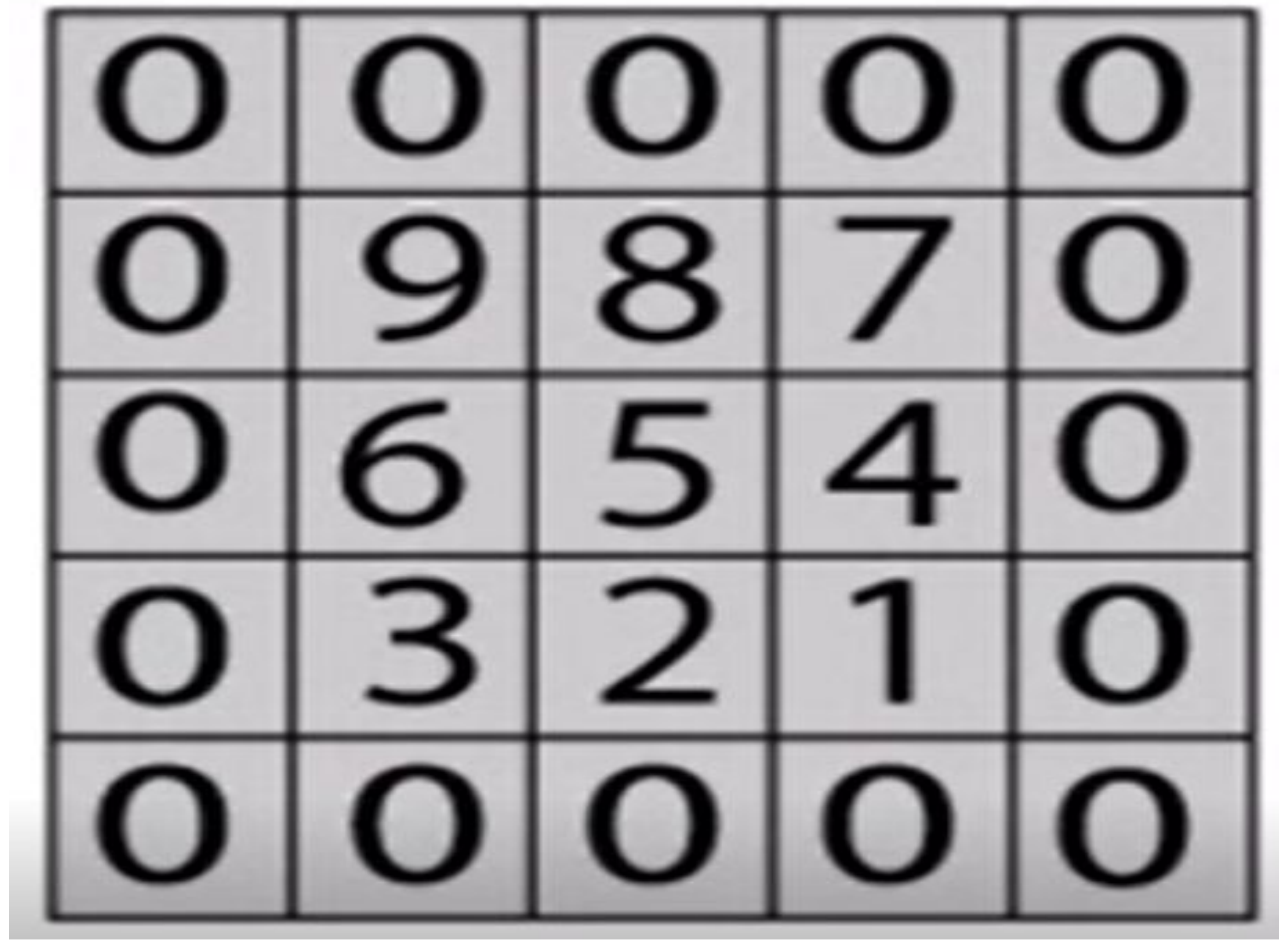
Spatial Correlation











Spatial Convolution

Function

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

Mask

9	8	7
6	5	4
3	2	1

Cropped Convolution Result

0	0	0	0	0
0	1	2	3	0
0	4	5	6	0
0	7	8	9	0
0	0	0	0	0

M x N pixels and G = number of gray levels ($G = 2^k$)

It is assumed that discrete levels are equally spaced between 0 to G-1 in the gray scale.

Therefore the number of bits required to store a digitized image of size **M x N** is

b = M x N x k .

Q. Find the number of bits required to store a 128 x 128 image with 256 gray levels.

128 x 128 image with 256 gray levels (ie 8 bits/pixel)

128 x 128 x 8 = 131072 bits

= 131072 /8 = 16384 bytes ~ 17000 bytes

Image resolution

A resolution can be defined as the total number of pixels in an image. Clarity of an image does not depends on number of pixels, but on the spatial resolution of the image.

Gray level resolution

Gray level resolution refers to the predictable or deterministic change in the shades or levels of gray in an image. It is equal to the number of bits per pixel.

Sampling is done on x axis and quantization is done in Y axis.
So that means digitizing the gray level resolution of an image is done in quantization.