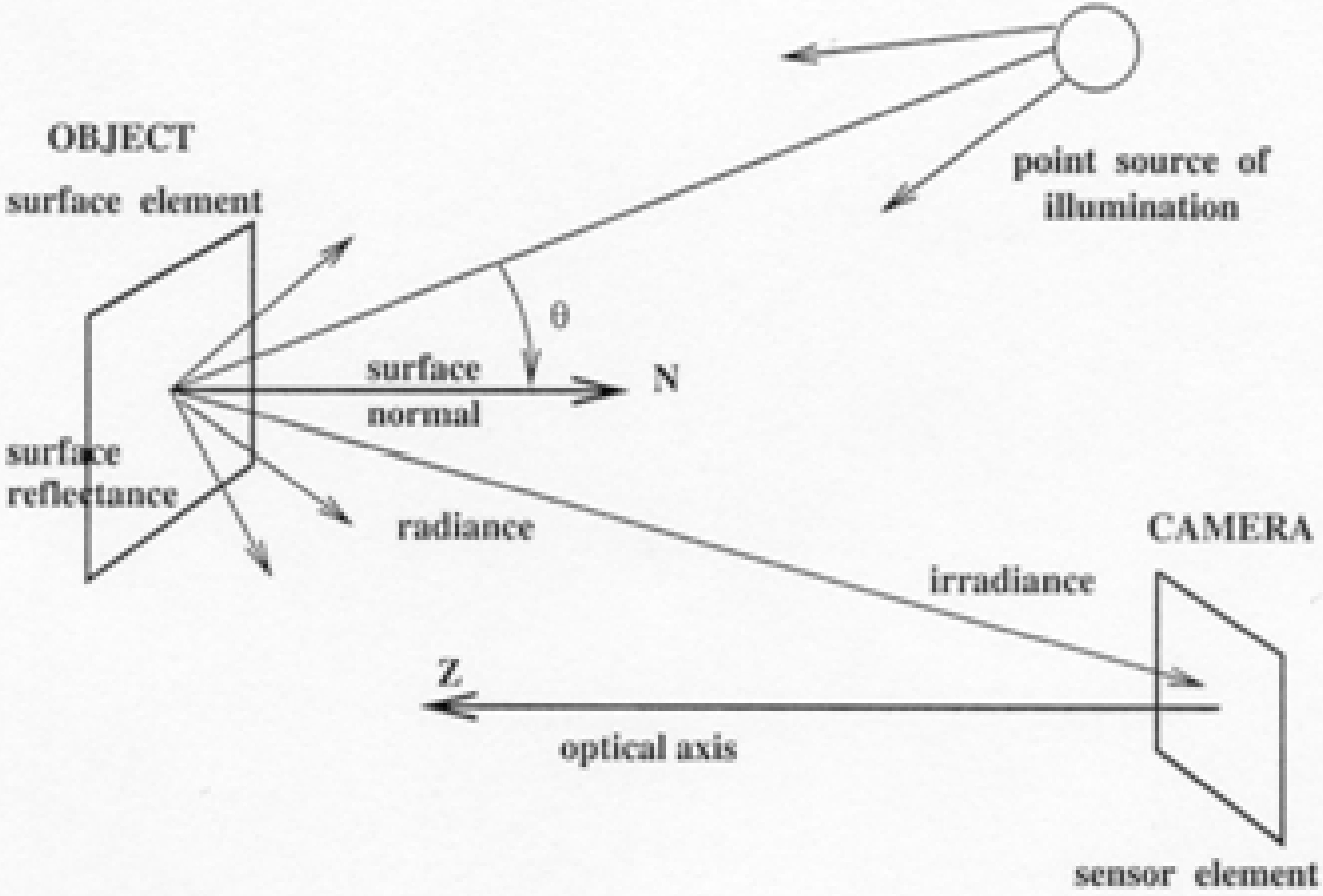
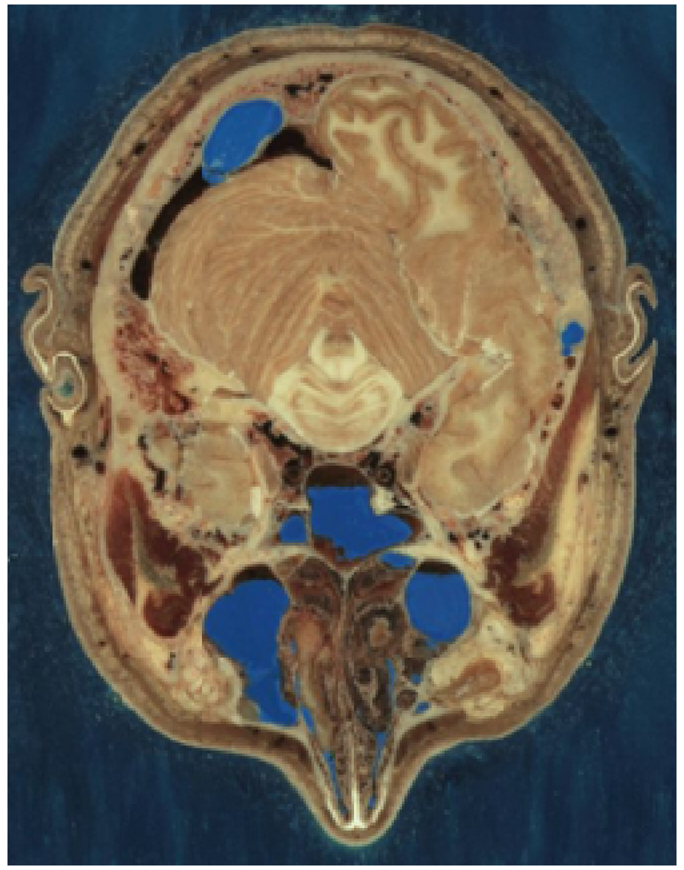
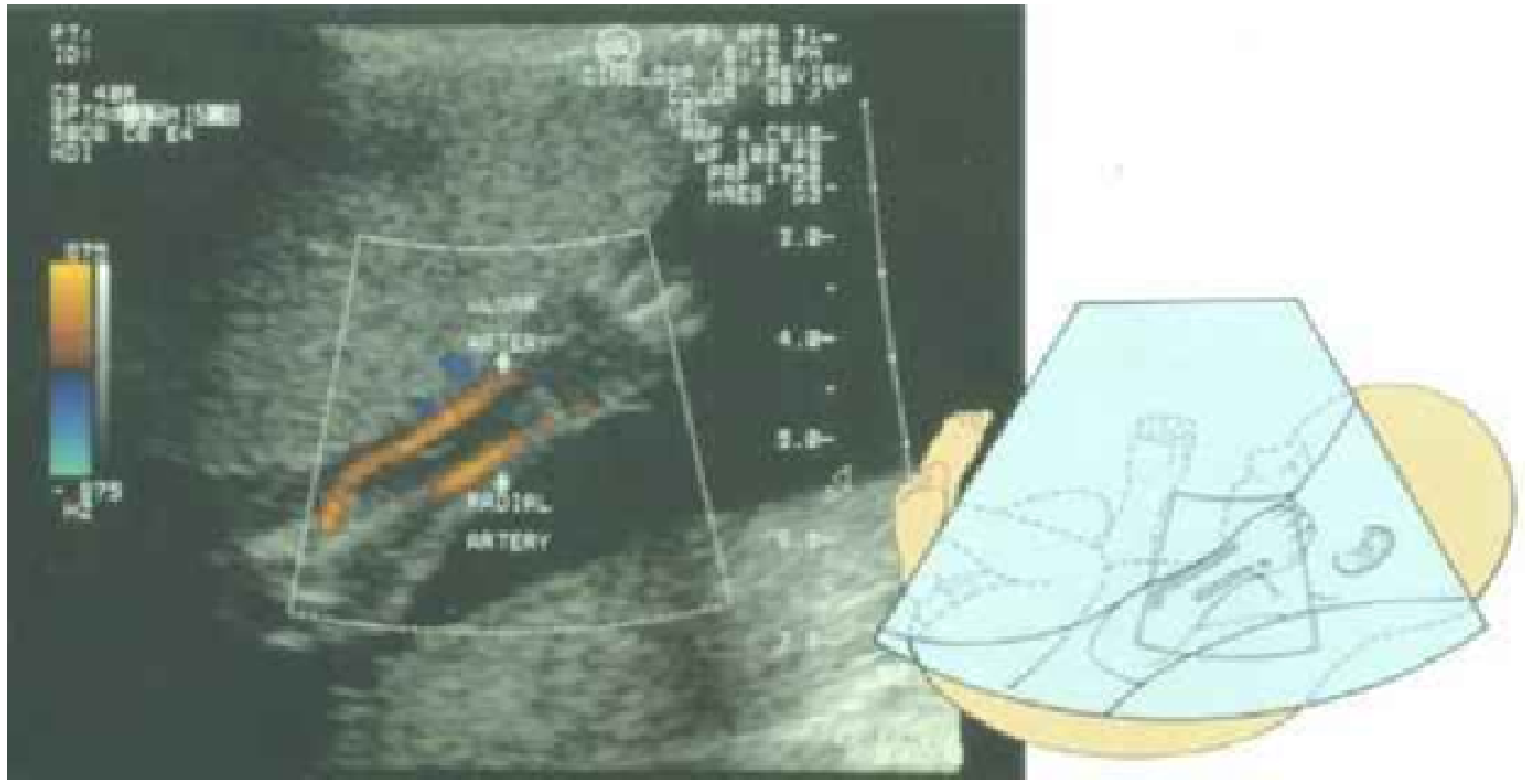
Image Acquisition

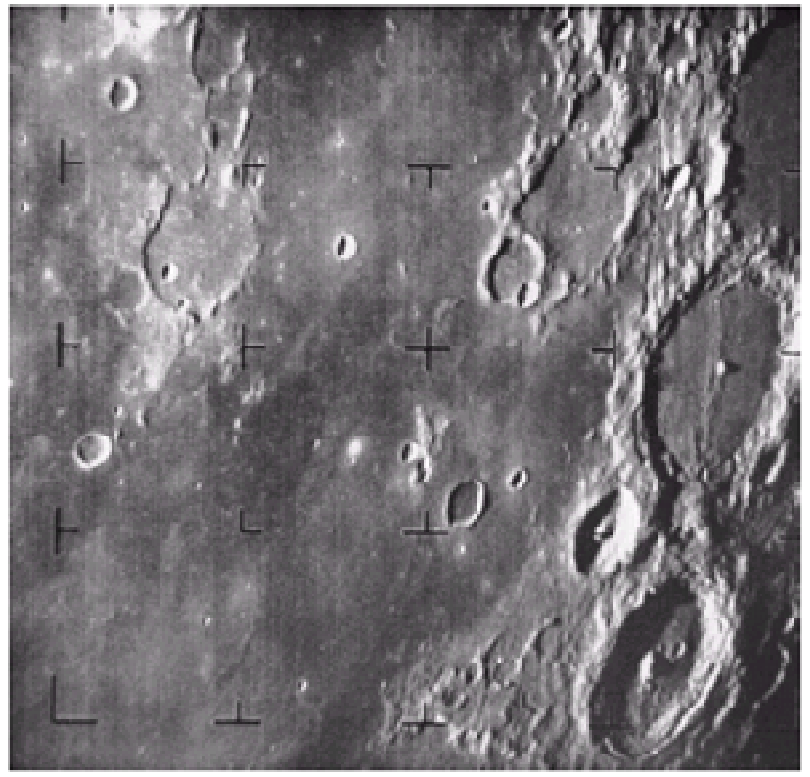






OBJECT
surface element
point source of illumination
θ
surface normal
N
surface reflectance
radiance
CAMERA
irradiance
Z
optical axis
sensor element




RADIAL
ARTERY



DIP Definition:

A Discipline in Which Both the Input and Output of a Process are Images.

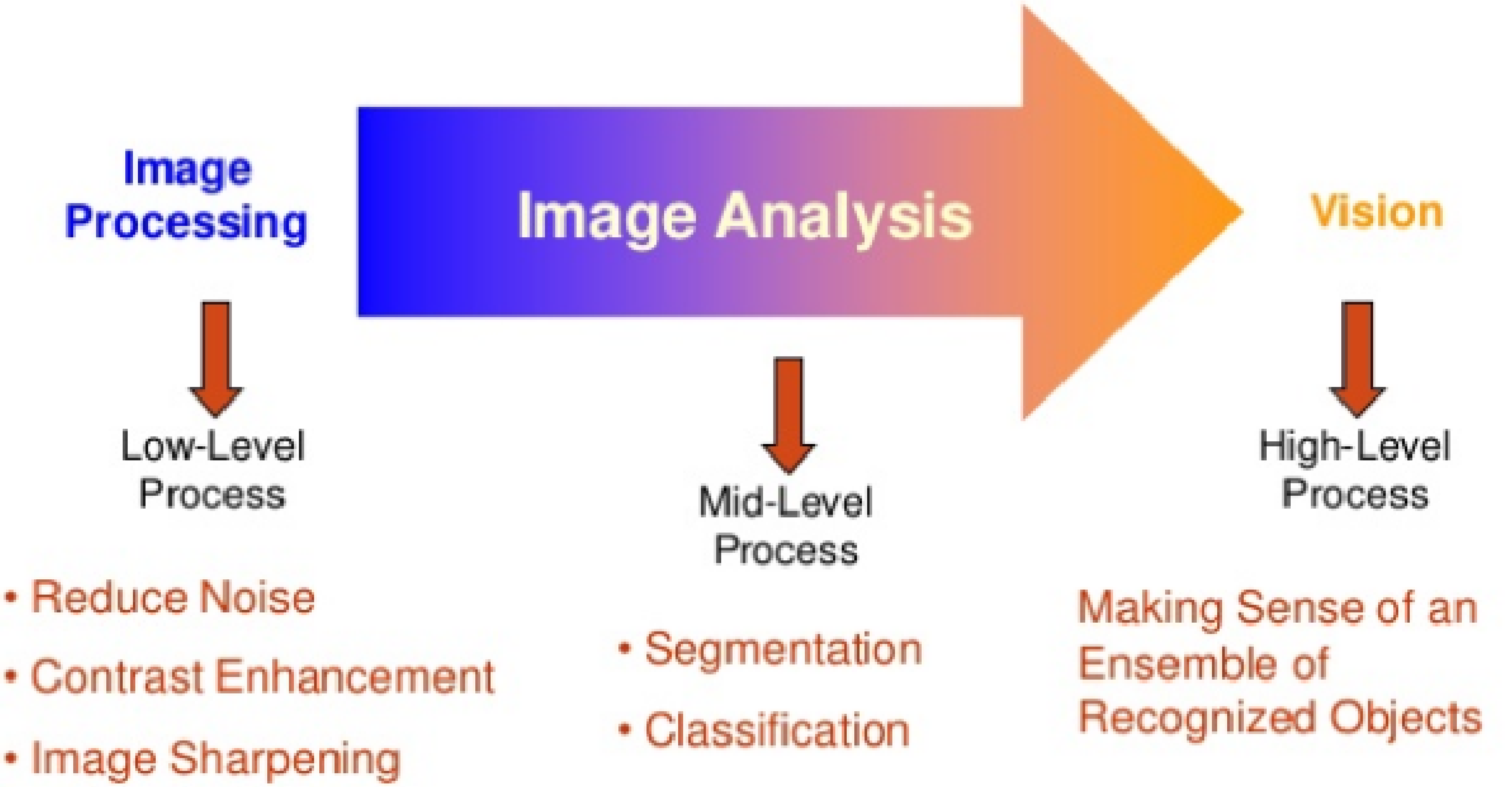
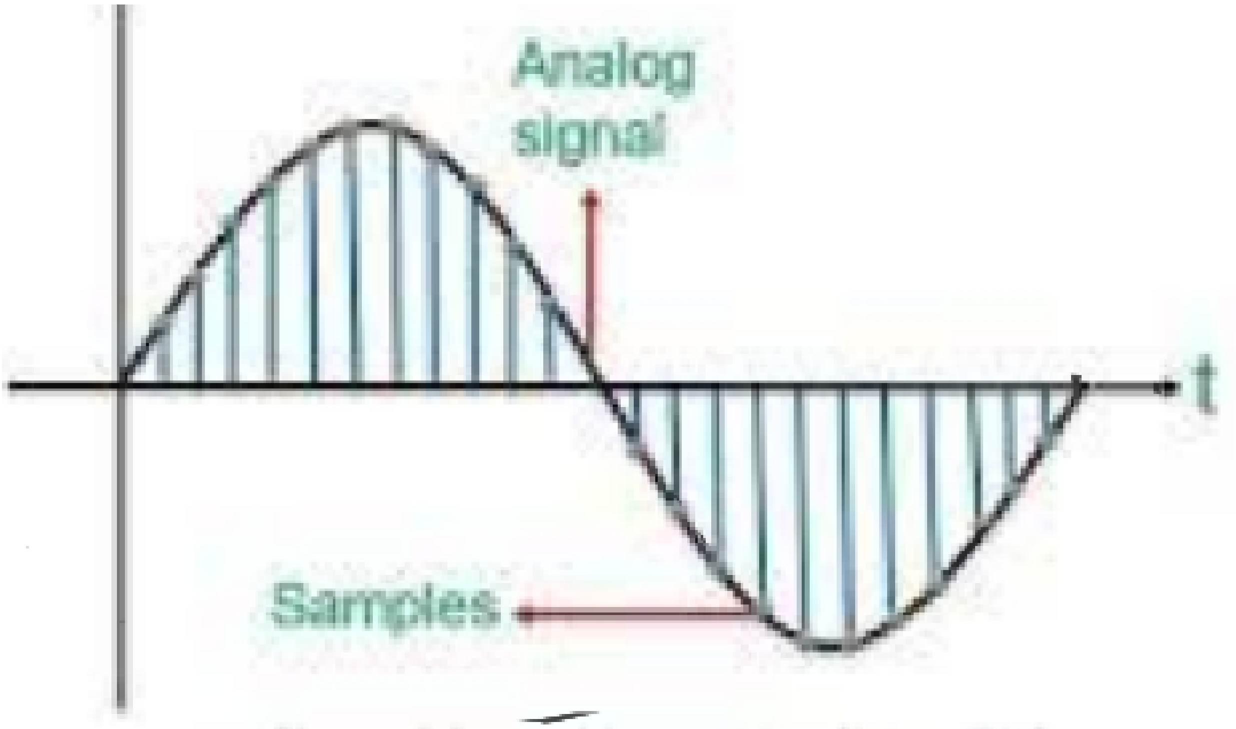


Image
Processing
Image Analysis
Vision
Low-Level
Process
Mid-Level
Process
High-Level
Process
• Reduce Noise
• Contrast Enhancement
• Image Sharpening
• Segmentation
• Classification
Making Sense of an
Ensemble of
Recognized Objects

Sampling and Quantization



Sampling of analog signal

Quantization of samples

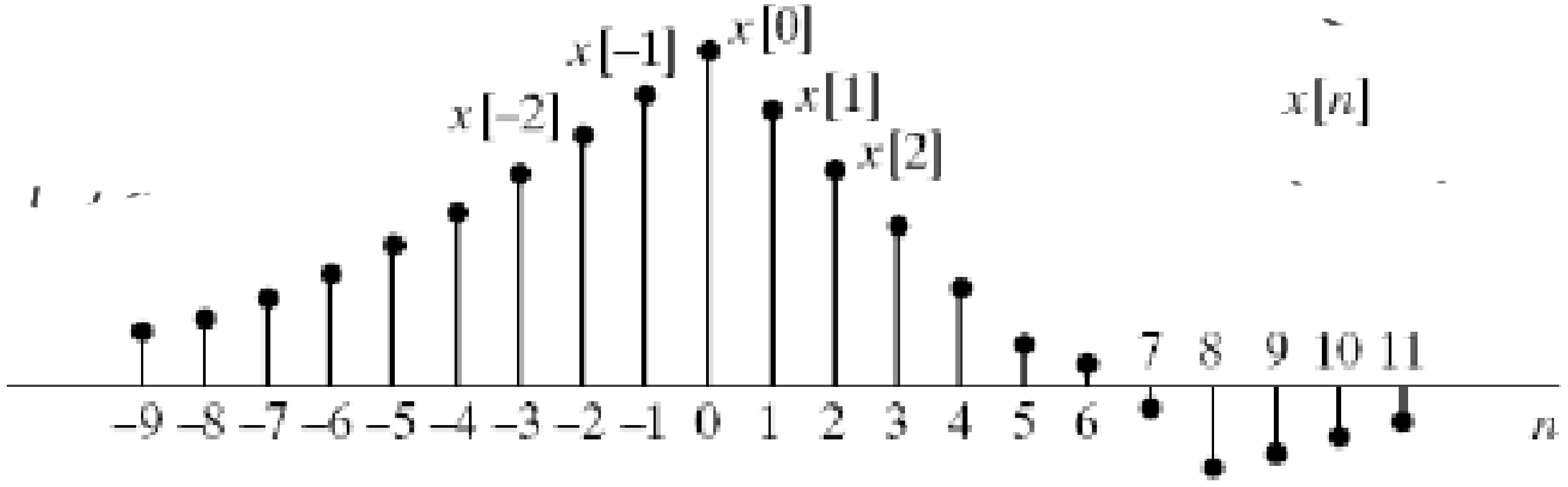
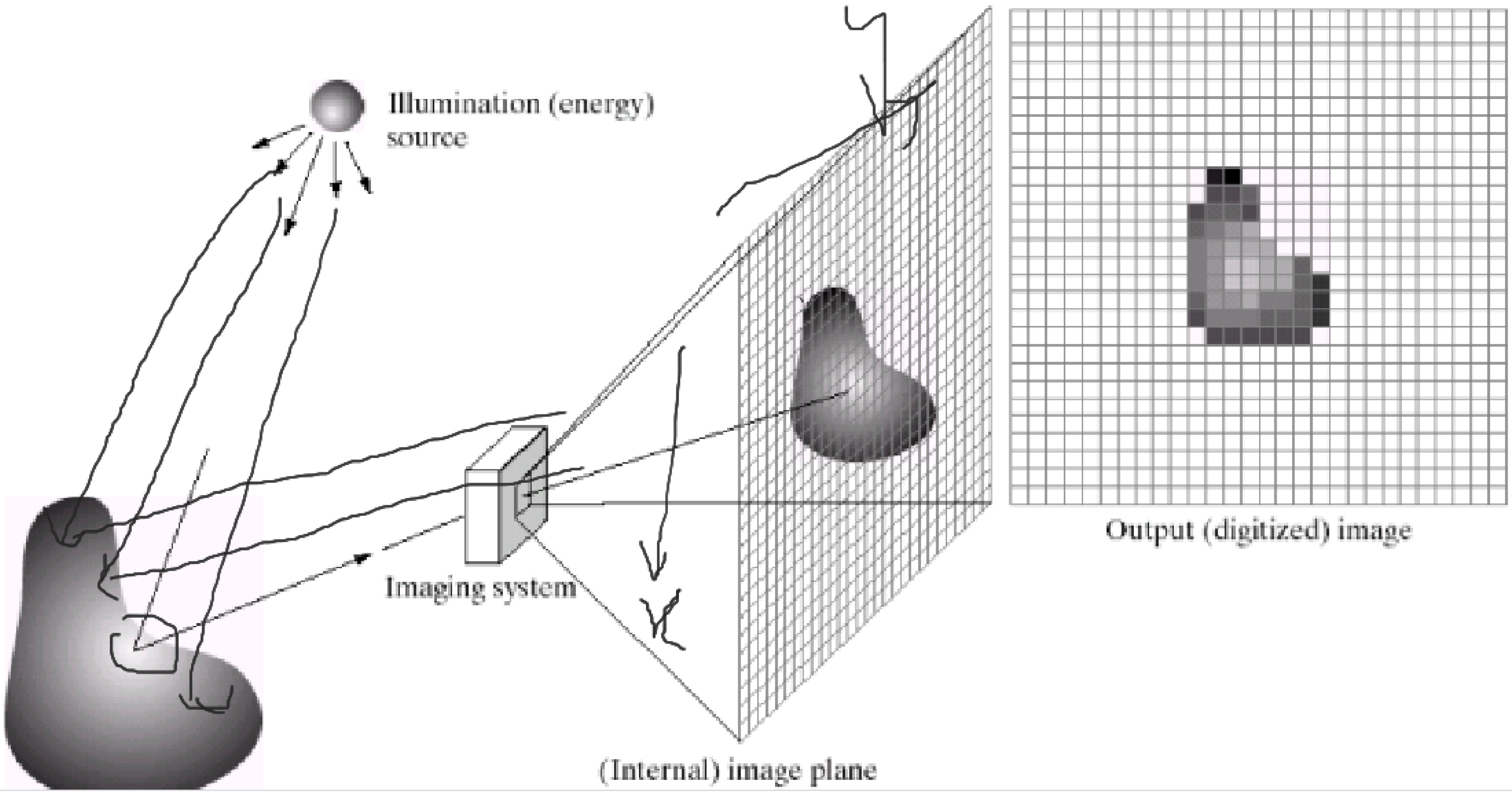
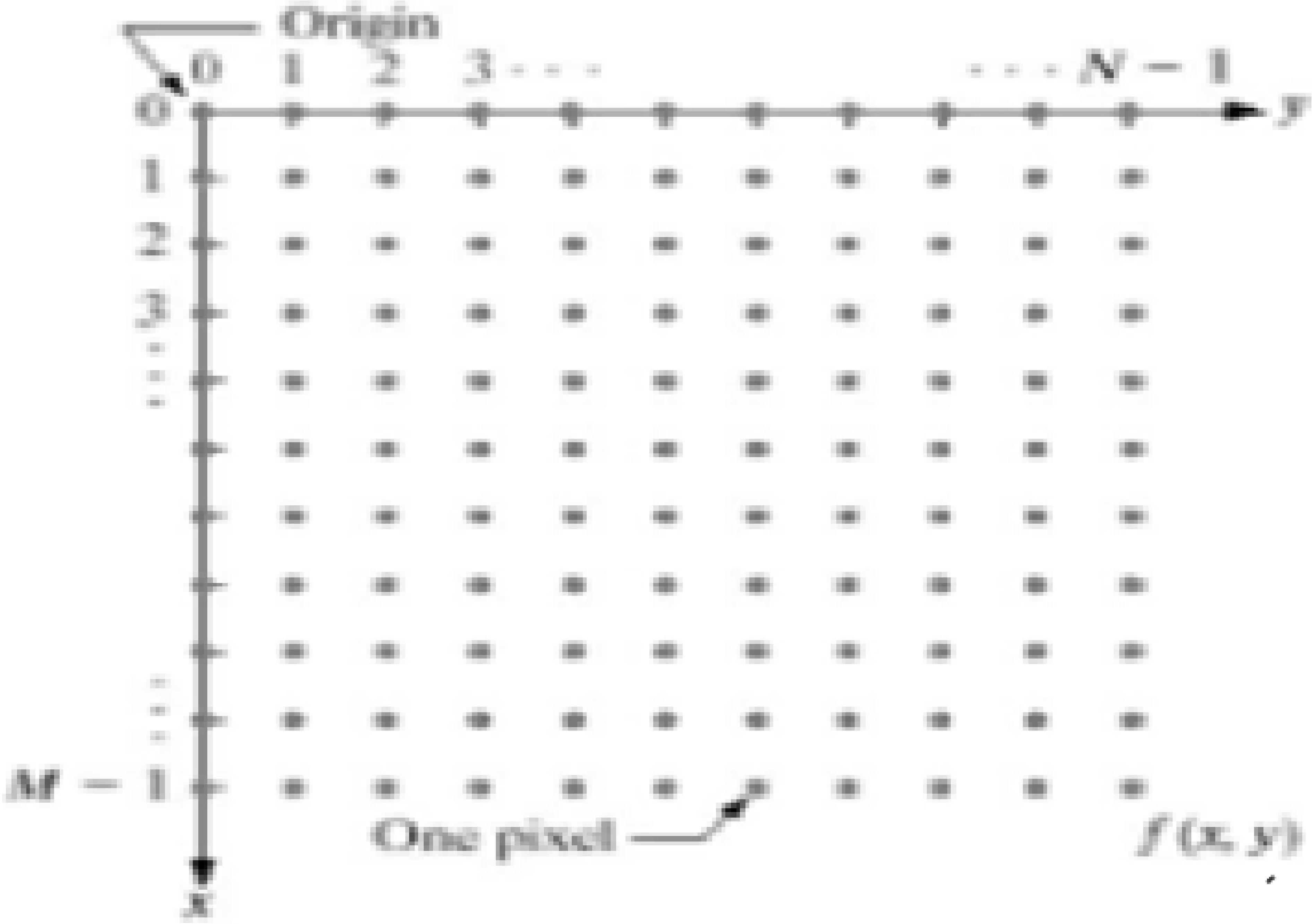


Image sampling and quantization



Representation of digital image

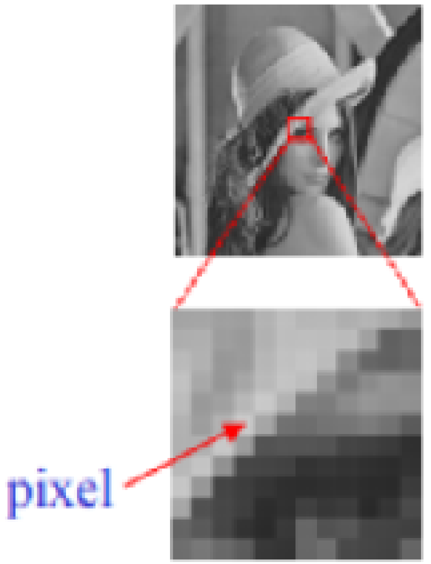


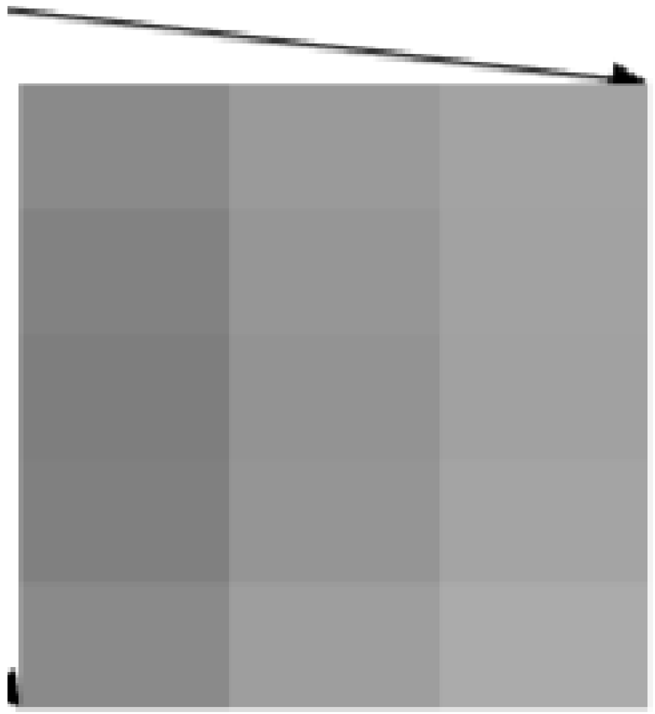
Representation of digital image

- To be suitable for computer processing an image, f(x,y) must be digitized both spatially and in amplitude
- Digitizing the spatial coordinates is called *image sampling*
- Amplitude digitization is called gray-level quantization
- f(x,y) is approximated by equally spaced samples in the form of an NxM array where each element is a discrete quantity

$$f(x,y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ . & & & \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

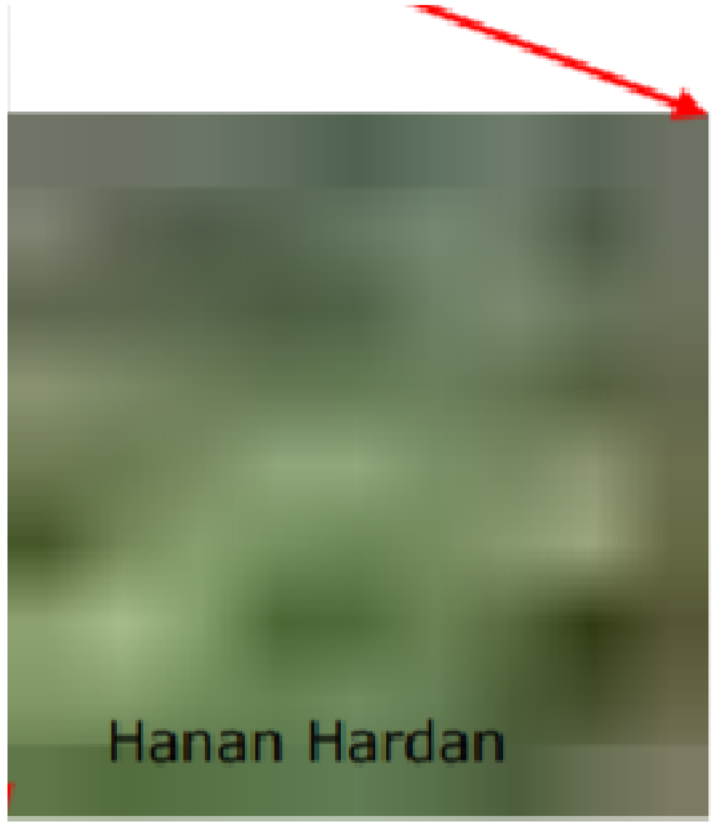
- ***Digital image***
  - *x*, *y*, *f(x, y)* are all finite and discrete
  - is composed of ***a finite number of elements***
  - These elements are referred to as
    - picture elements
    - image elements
    - pels
    - pixels - most widely used





Gray scale values

$$\begin{bmatrix} 10 & 10 & 16 & 28 \\ 9 & 6 & 26 & 37 \\ 15 & 25 & 13 & 22 \\ 32 & 15 & 87 & 39 \end{bmatrix}$$



RGB components

10 10 16 28
9
15
32

65 70 56 43
32
21
54

$$\begin{bmatrix} 99 & 70 & 56 & 78 \\ 60 & 90 & 96 & 67 \\ 85 & 85 & 43 & 92 \\ 32 & 65 & 87 & 99 \end{bmatrix}$$

Fundamental Steps of Digital Image Processing

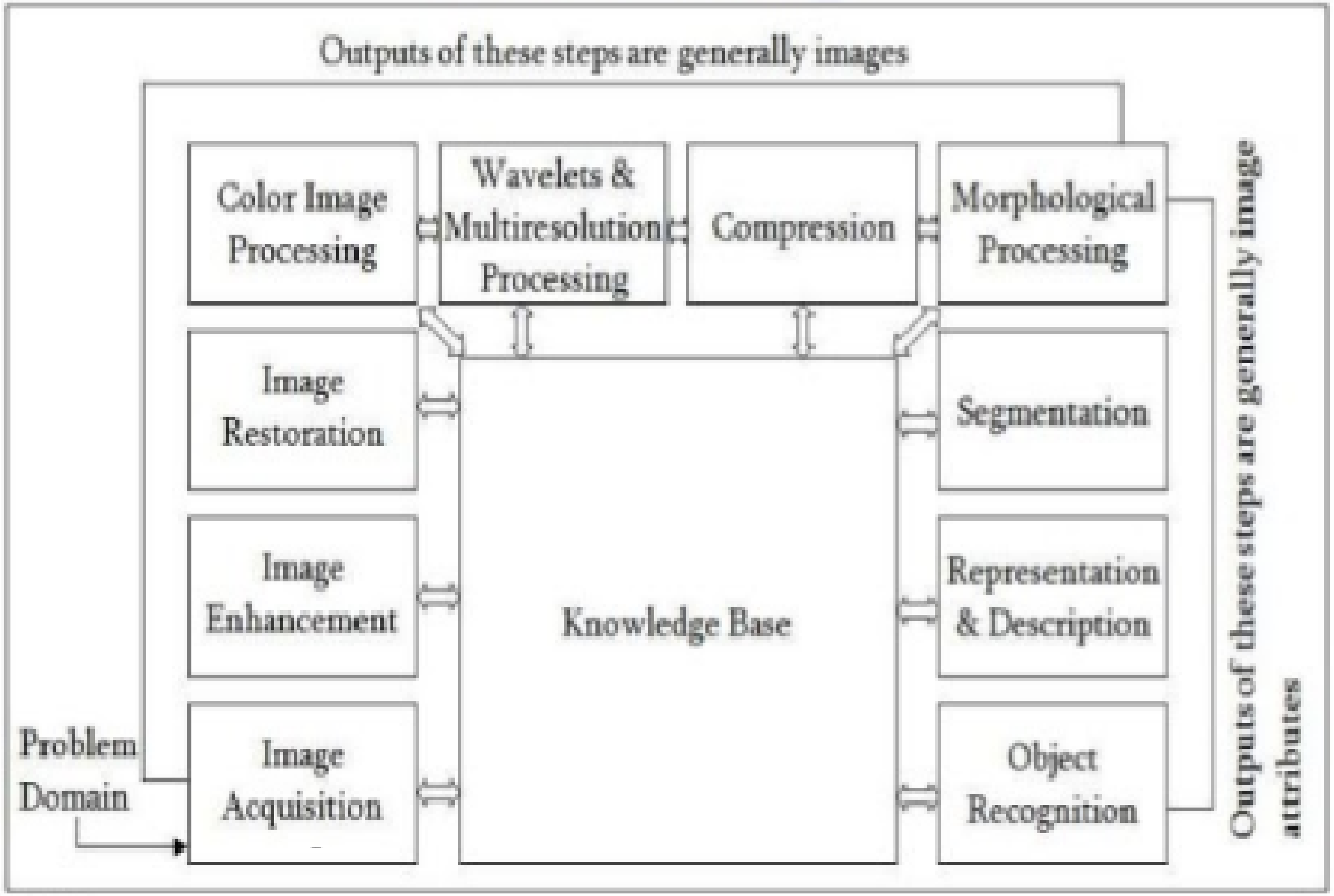


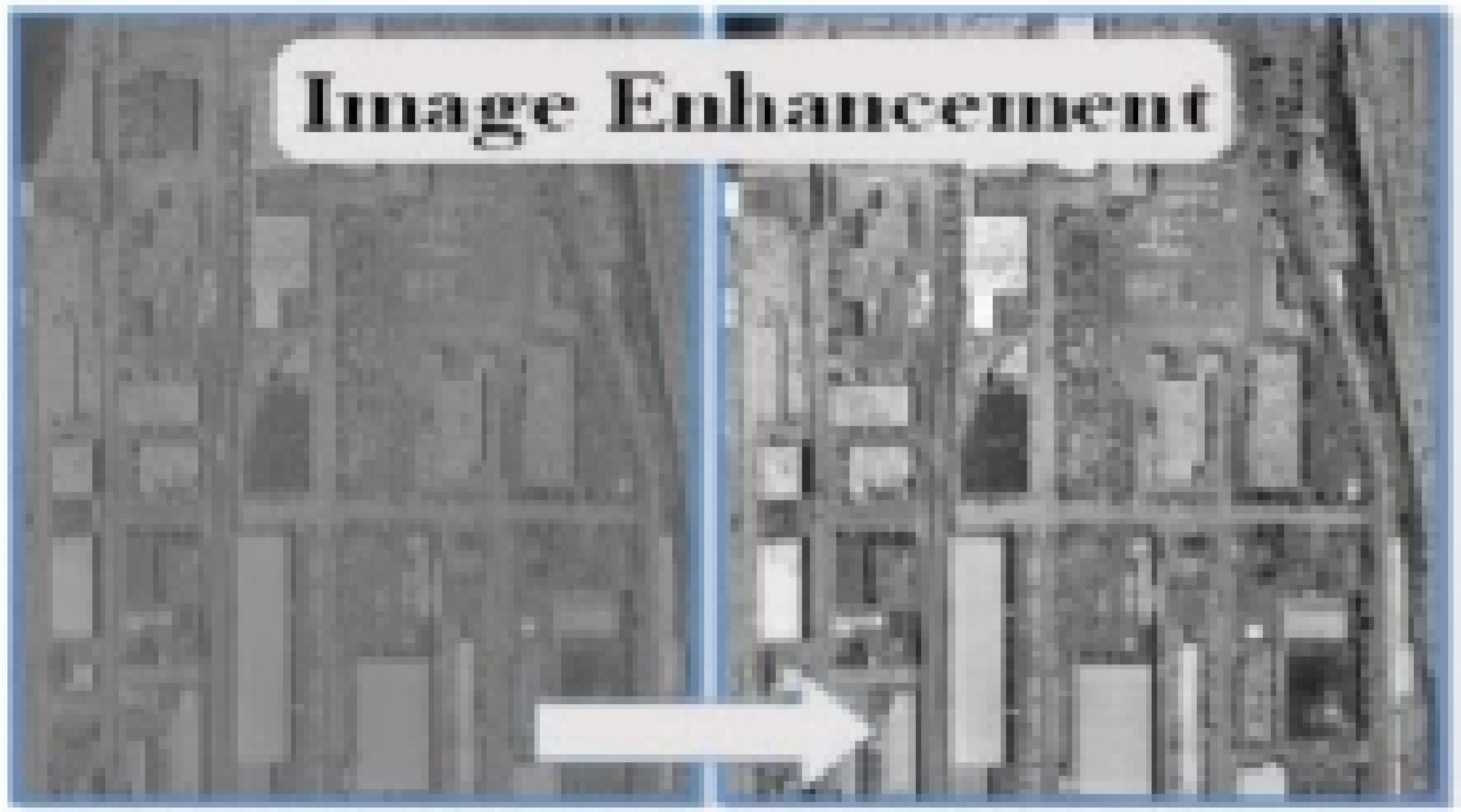
Image Acquisition

Image Enhancement

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

- The idea behind enhancement techniques is to bring out details that are hidden, or simple to highlight certain features of interest in an image.

Image Enhancement



What is Image Restoration?

- Image restoration attempts to restore images that have been degraded
  - ✓ Identify the degradation process and attempt to reverse it.
  - ✓ Almost Similar to image enhancement, but more objective.

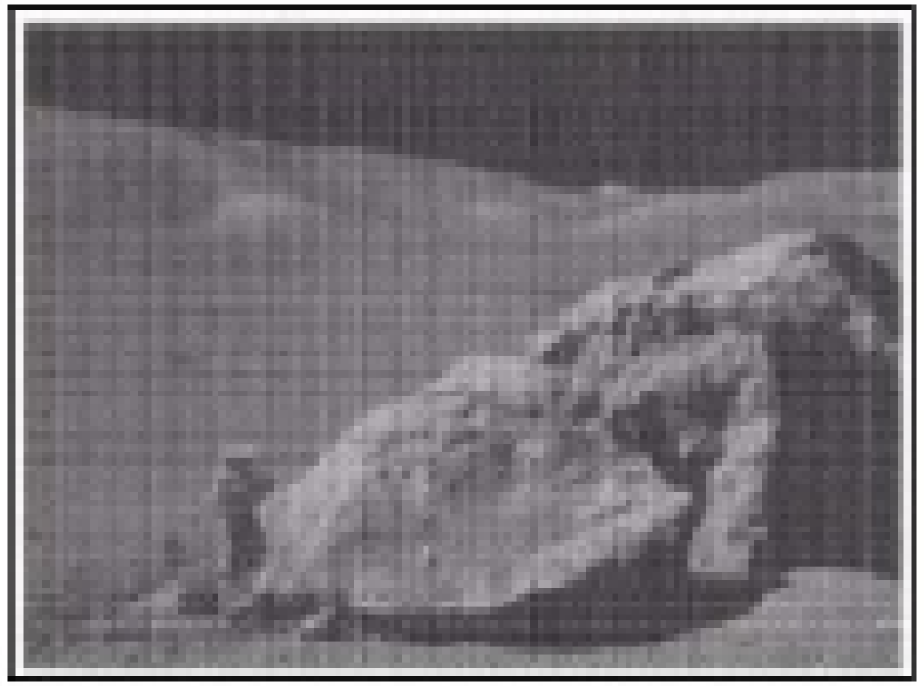


Fig: Degraded image

Fig: Restored image

Image Compression?

- How we stored the image: Reduce the size for storage .
- How analog image world is relate to digital processing world.
- Compression-Remove redundancies.
- Transmission with minimum bandwidth.
- Lossy Compression=redundancy +some information, but still acceptable.

Original Image
Size-116 KB

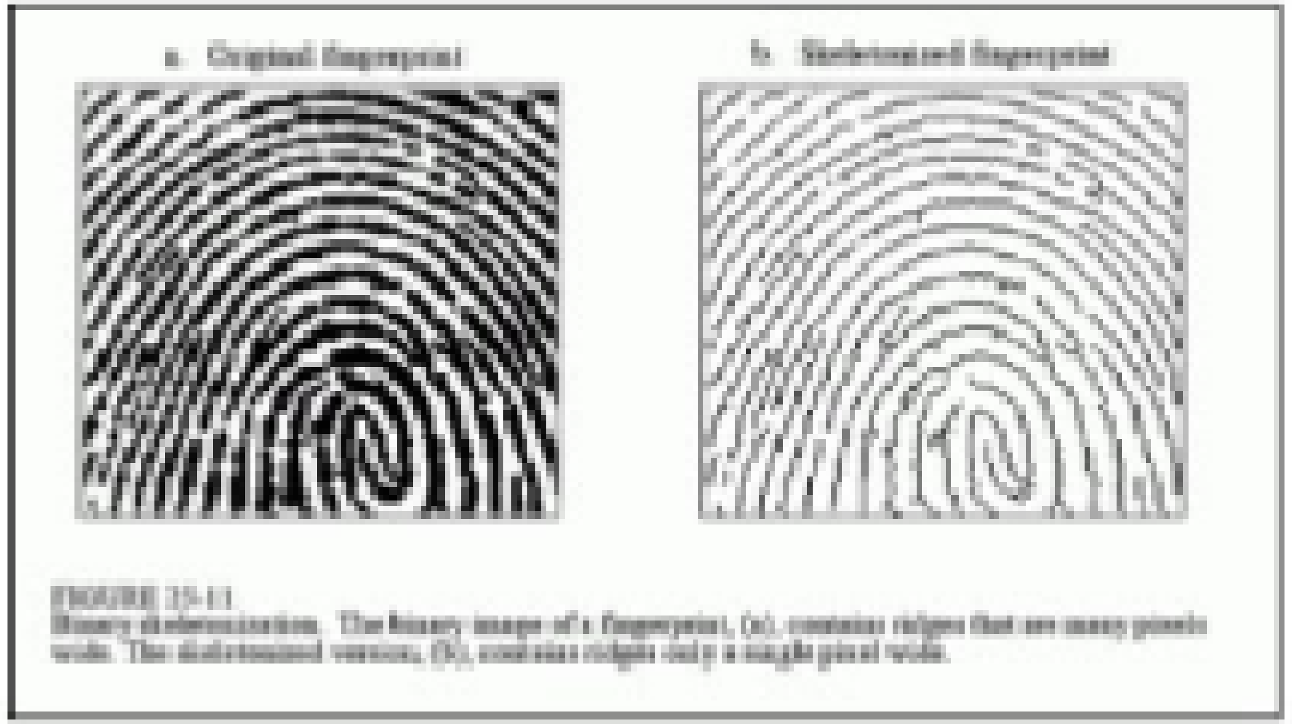
Compressed Image
Size-12.9 KB, 11 %

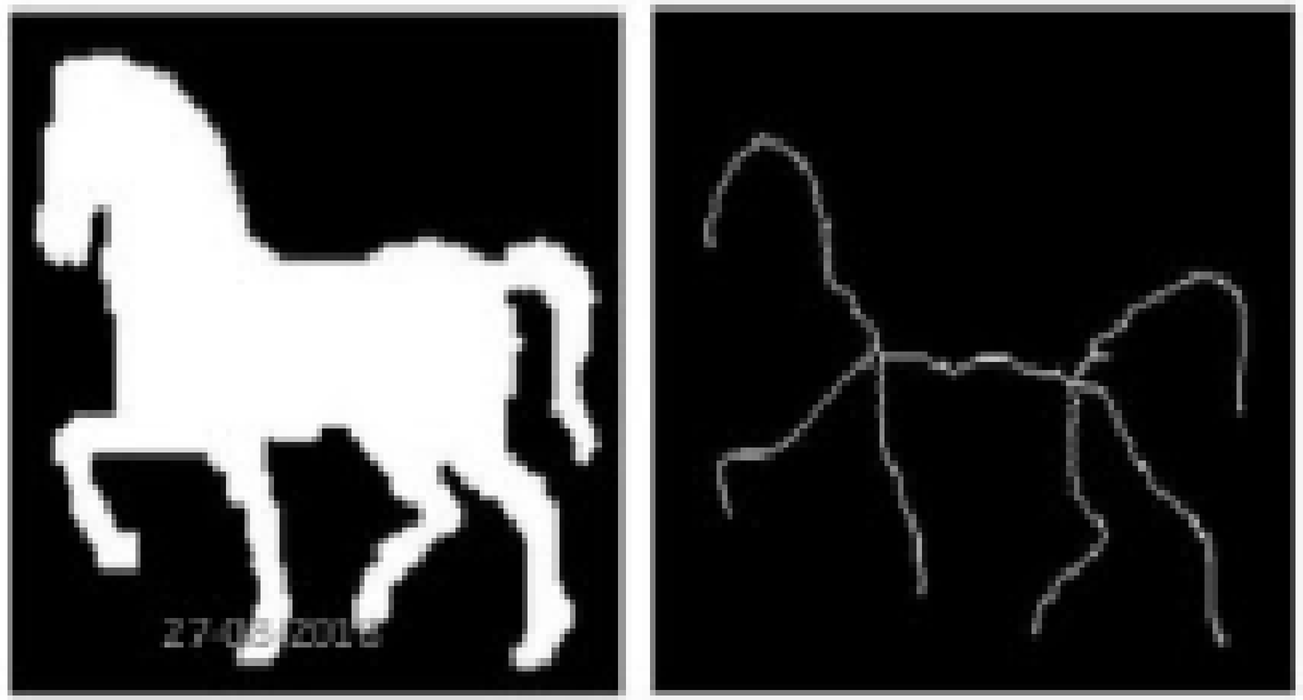
Compressed Image
Size-1.95 KB, 1.6 %

Morphological Image Processing

Extract image components that are useful in the representation and description of region shape, such as-

- Boundaries extraction
- Skeletons
- Convex hull
- Morphological filtering
- Thinning
- Pruning...many More





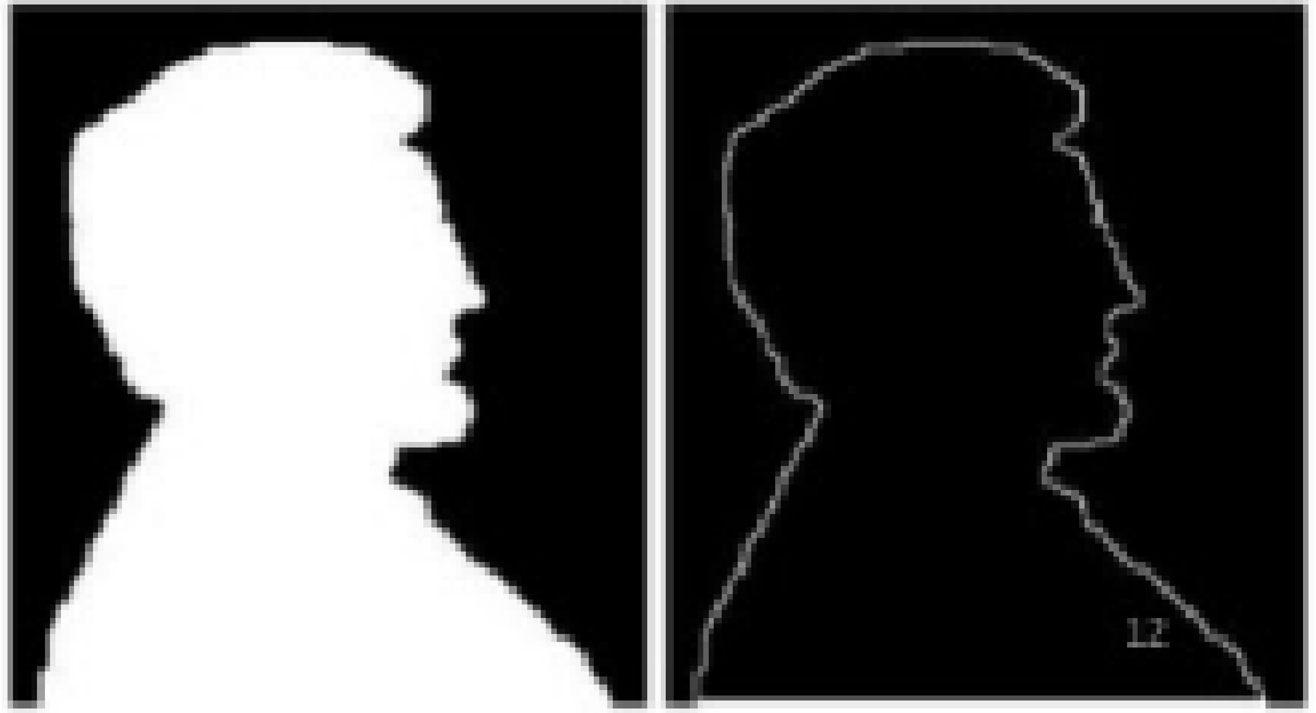
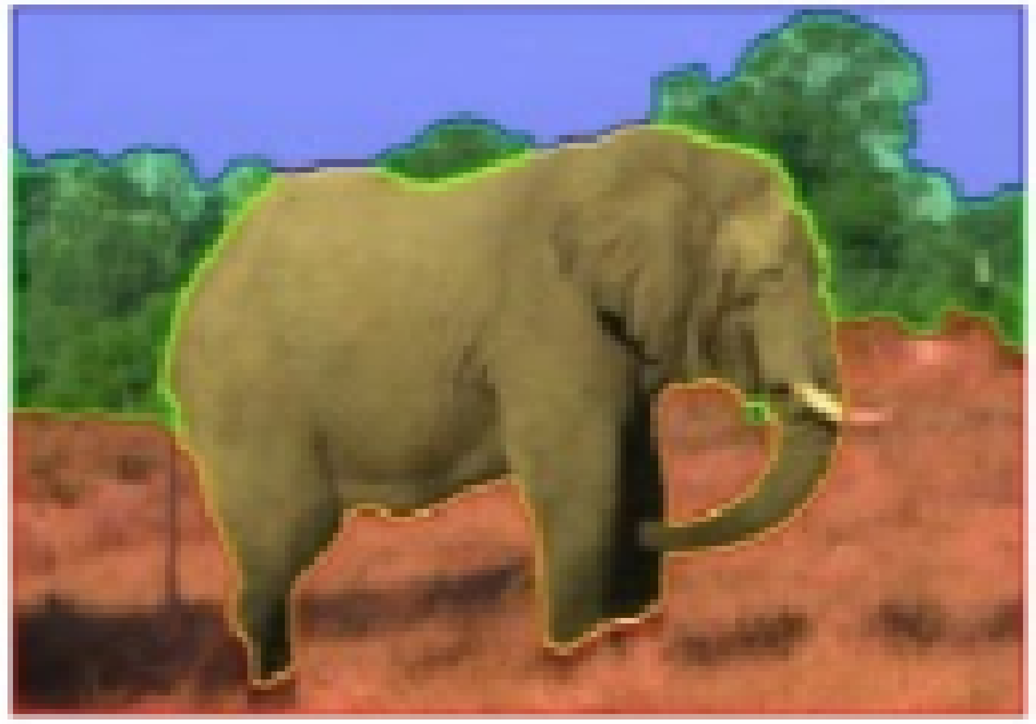


Image Segmentation

In computer vision, **Image Segmentation** is the process of partitioning a digital image into multiple segments. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze



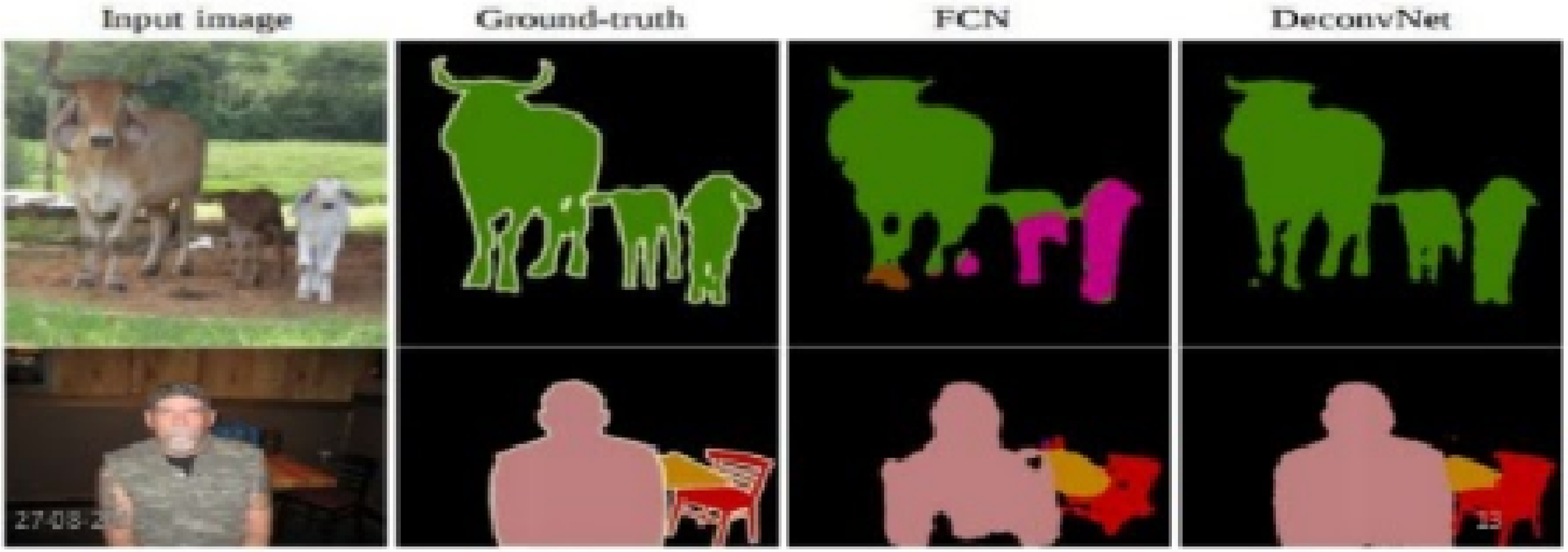


Image Representation & Description

- Image representation & description: After an image is segmented into regions; the resulting aggregate of segmented pixels is represented & described for further computer processing.

- Representation and Description
  - Representing regions in 2 ways:
    - Based on their external characteristics (its boundary):
    - Shape characteristics

- Based on their internal characteristics (its region):

– Regional properties: color, texture, and ...

- Both

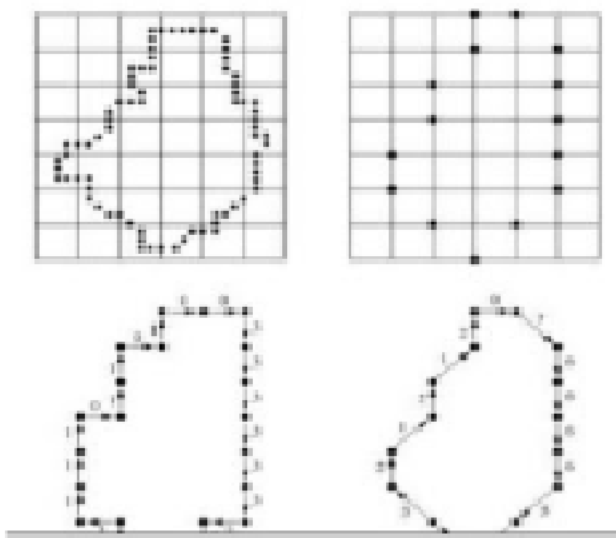
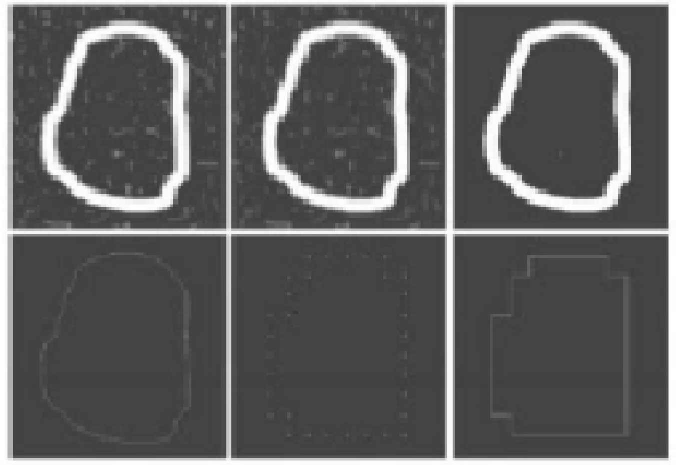
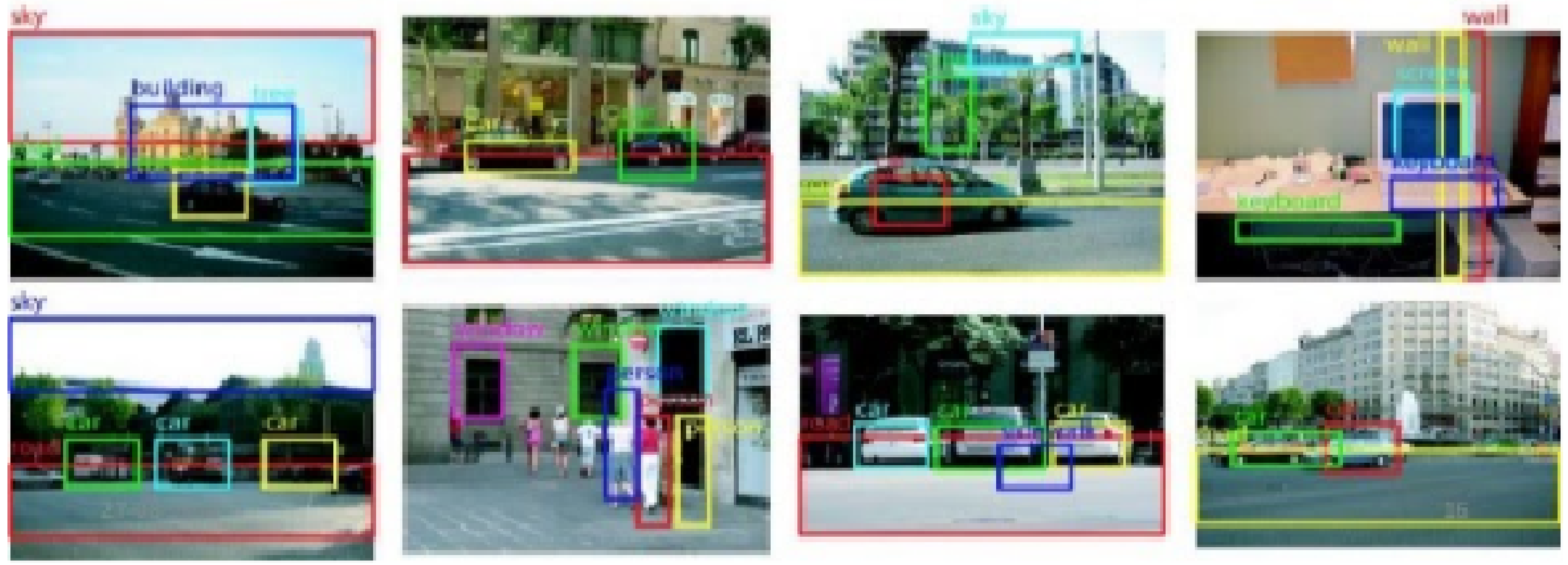


Image Representation & Description



Object Recognition

- Object Detection is the process of finding instances of objects in images. .This allows for multiple objects to be identified and located within the same image.
- Object recognition can be termed as identifying a specific object in a digital image or video. Object recognition have immense of applications in the field of monitoring and surveillance, medical analysis, robot localization and navigation etc.



Knowledge Base

Knowledge about a problem domain is coded into an image processing system in the form of a knowledge database.

Essential Components of an Image Processing System

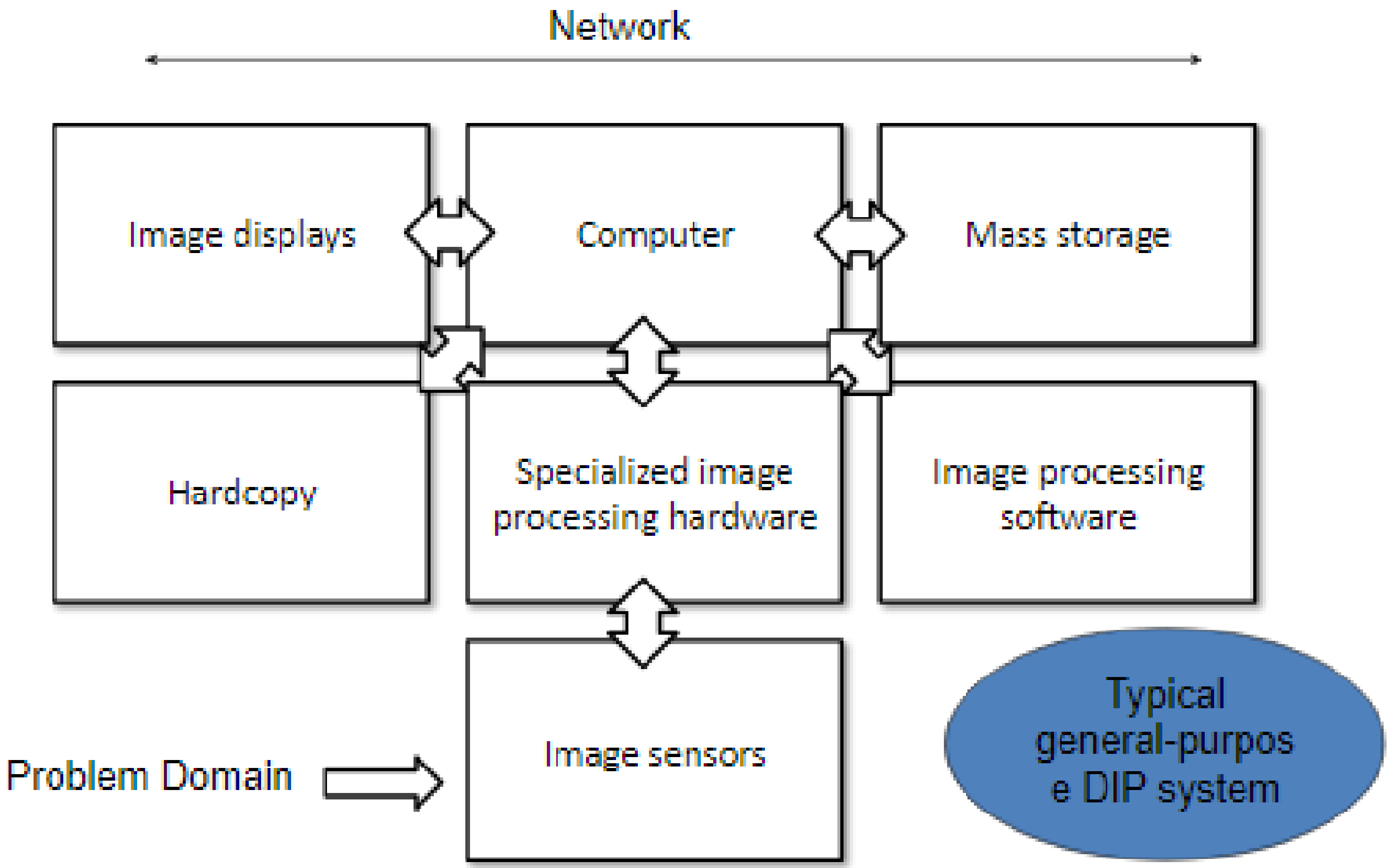


Image Sensors

Two elements are required to acquire digital images. The first is the physical device that is sensitive to the energy radiated by the object we wish to image (*Sensor*). The second, called a *digitizer*, is a device for converting the output of the physical sensing device into digital form.

Specialized Image Processing Hardware

- Usually consists of the digitizer, mentioned before, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images.

- This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs that the typical main computer cannot handle.

Computer

The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance.

Image Processing Software

Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules.

Mass Storage Capability

Mass storage capability is a must in a image processing applications. And image of sized 1024 * 1024 pixels requires one megabyte of storage space if the image is not compressed.

Digital storage for image processing applications falls into three principal categories:

1. Short-term storage for use during processing.
2. on line storage for relatively fast recall
3. Archival storage, characterized by infrequent access

Image Displays

The displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of the image and graphics display cards that are an integral part of a computer system.

Hardcopy devices

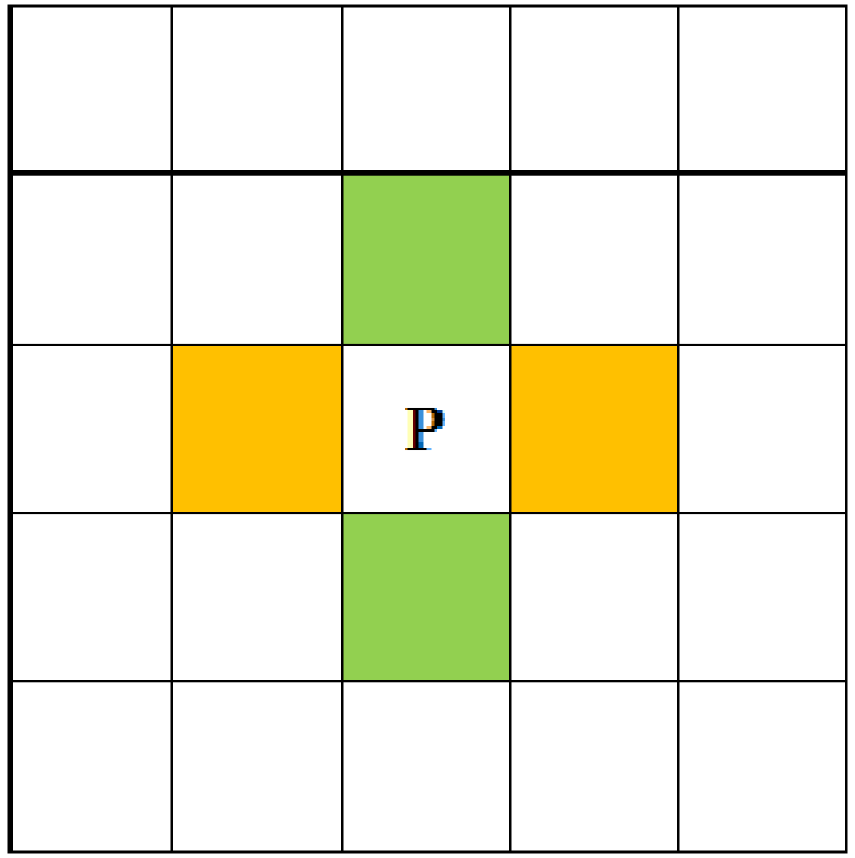
Used for recording images, include laser printers, film cameras, heat-sensitive devices, inkjet units and digital units, such as optical and CD-Rom disks.

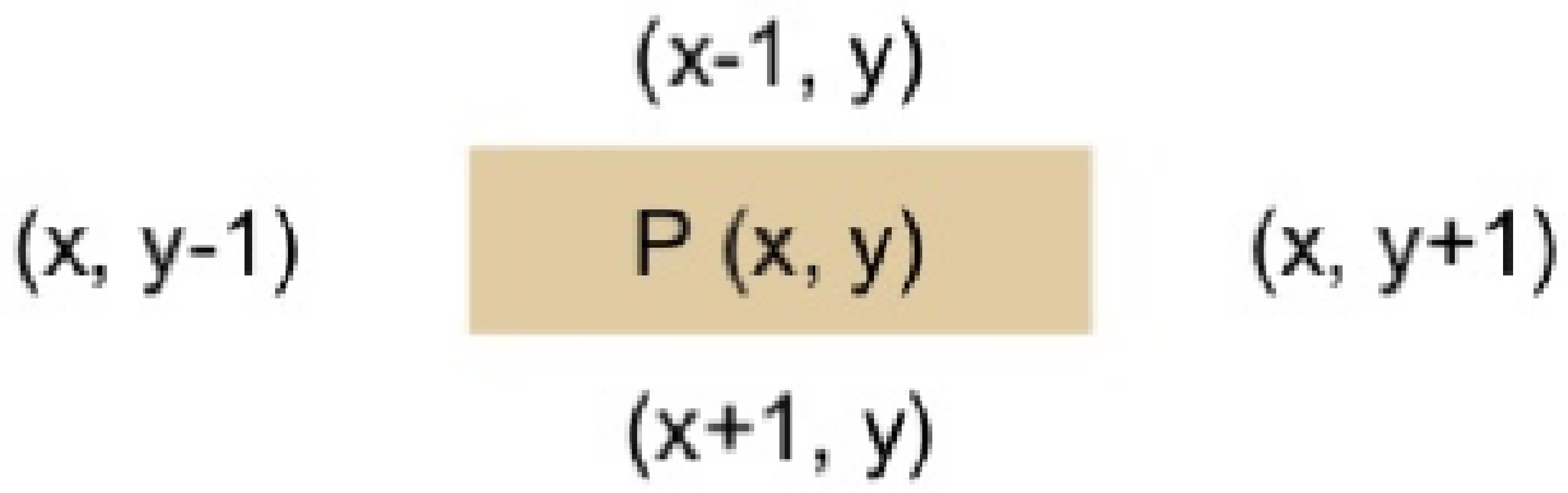
Basic relationships between the pixels

Neighbors of a pixel

4- Neighbours:

- A pixel p at location (x,y) has 2 horizontal and 2 vertical neighbour. In total a pixel p has four neighbours

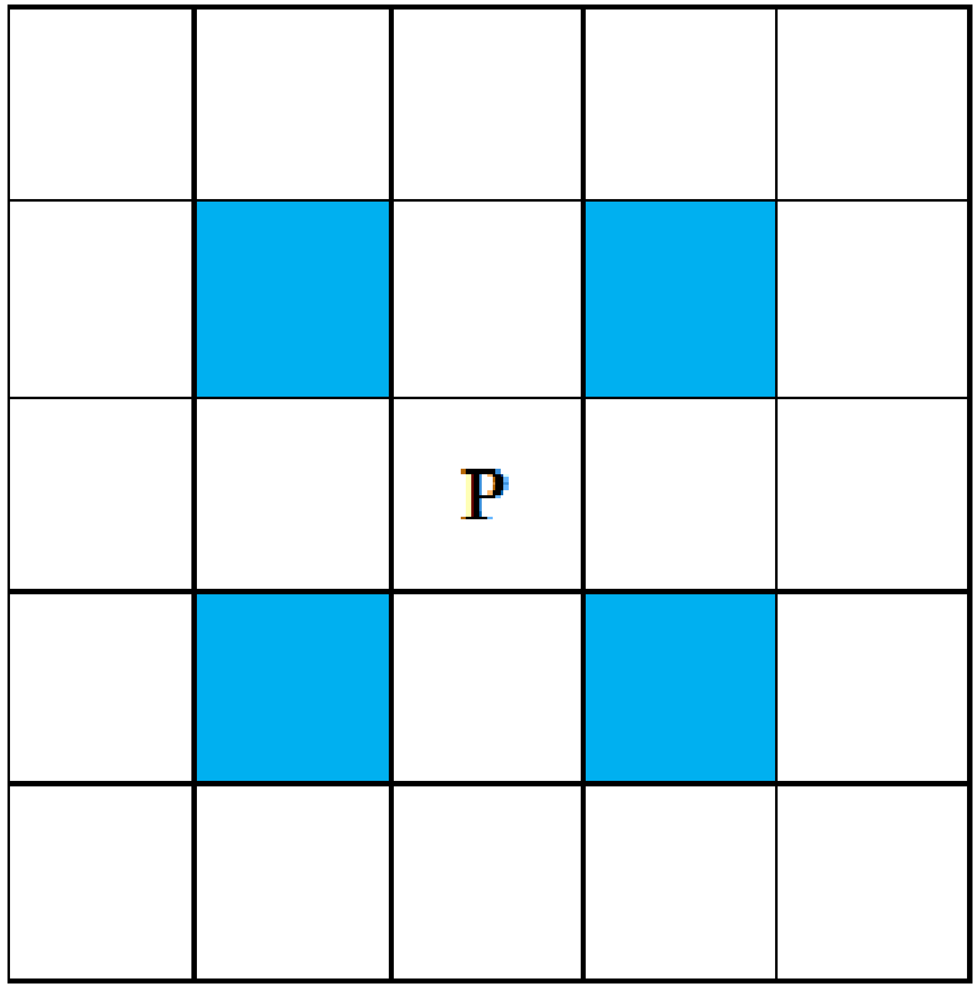




- This set of four pixel is called 4 neighbour of p = $N_4(p)$
- Each of this neighbour is at a unit distance from p

$$N_4(P) = \{(x, y-1), (x, y+1), (x-1, y), (x+1, y)\}$$

Diagonal Neighbors: A pixel P at location (x, y) has 4 diagonal neghbors



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

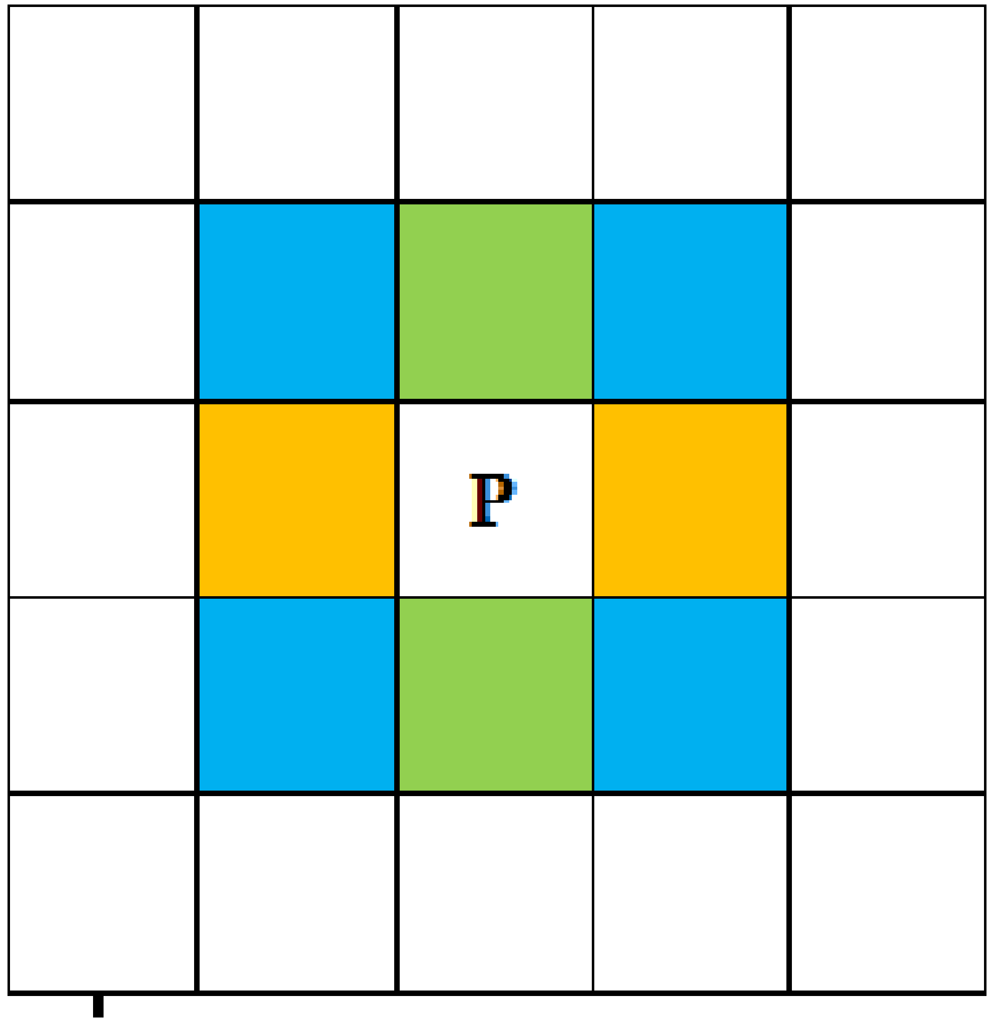
P (x,y)

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

$$N_D(P) = \{(x-1, y-1), (x-1, y+1), (x+1, y-1), (x+1, y+1)\}$$

8- neighbors:

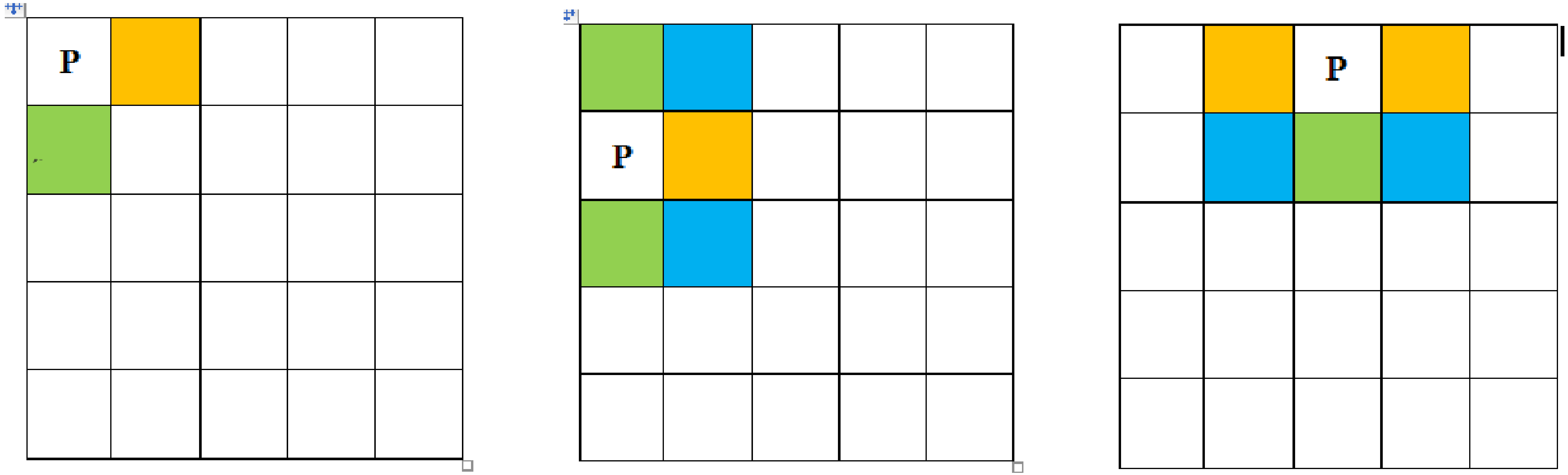
$N_D(P)$ together with the N4 (P), are called the 8-neighbors of p, denoted by $N_8(p)$.



(x-1, y-1)	(x-1, y)	(x-1, y+1)
(x, y+1)	*P (x,y)*	(x, y+1)
(x+1, y-1)	(x+1, y)	(x+1, y+1)

$N_8(P) = \{(x-1, y-1), (x-1, y), (x-1, y+1), (x, y-1), (x, y+1), (x+1, y-1), (x+1, y), (x+1, y+1)\}$

If pixel P is a boundary pixels, pixel P has partial neighbors



Connectivity

Pixel connectivity is a central concept of both edge- and region-based approaches to segmentation

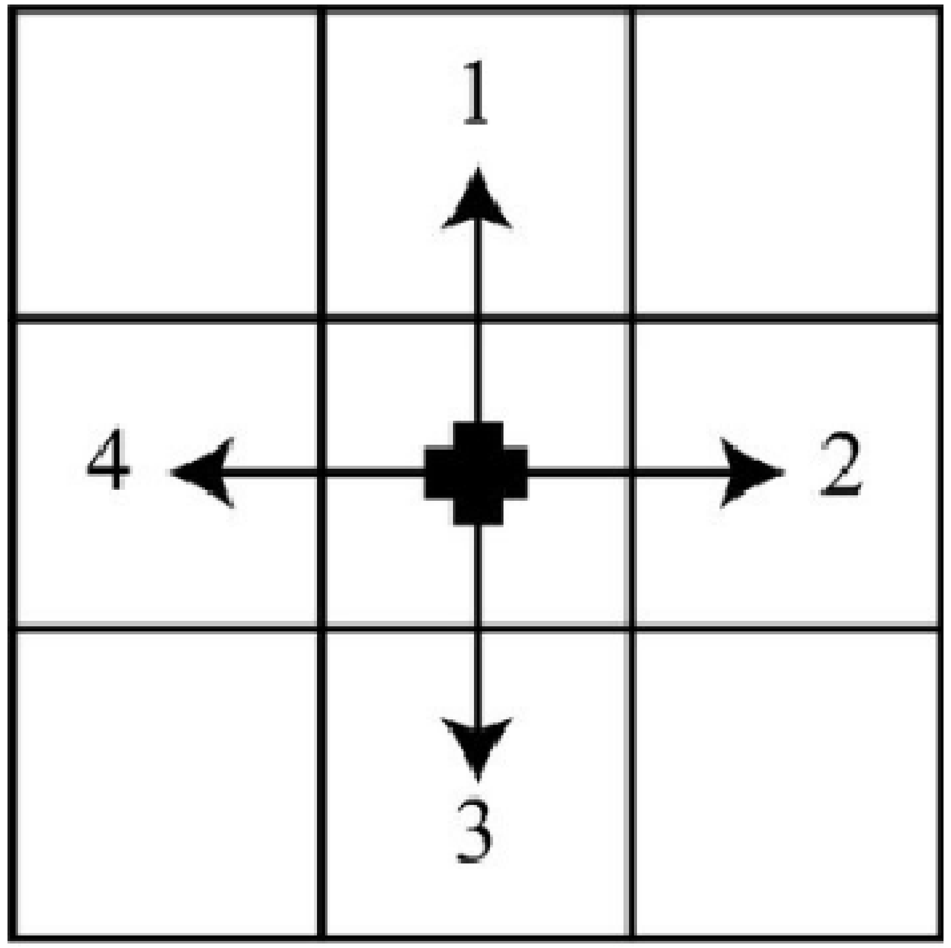
The notation of pixel connectivity describes a relation between two or more pixels. For two pixels to be connected they have to fulfill certain conditions on the pixel brightness and spatial adjacency.

First, in order for two pixels to be considered connected, their pixel values must both be from the same set of values *V*. For a grayscale image, *V* might be any range of graylevels, *e.g.* $V=\{22,23,...40\}$, for a binary image we simple have $V=\{1\}$.

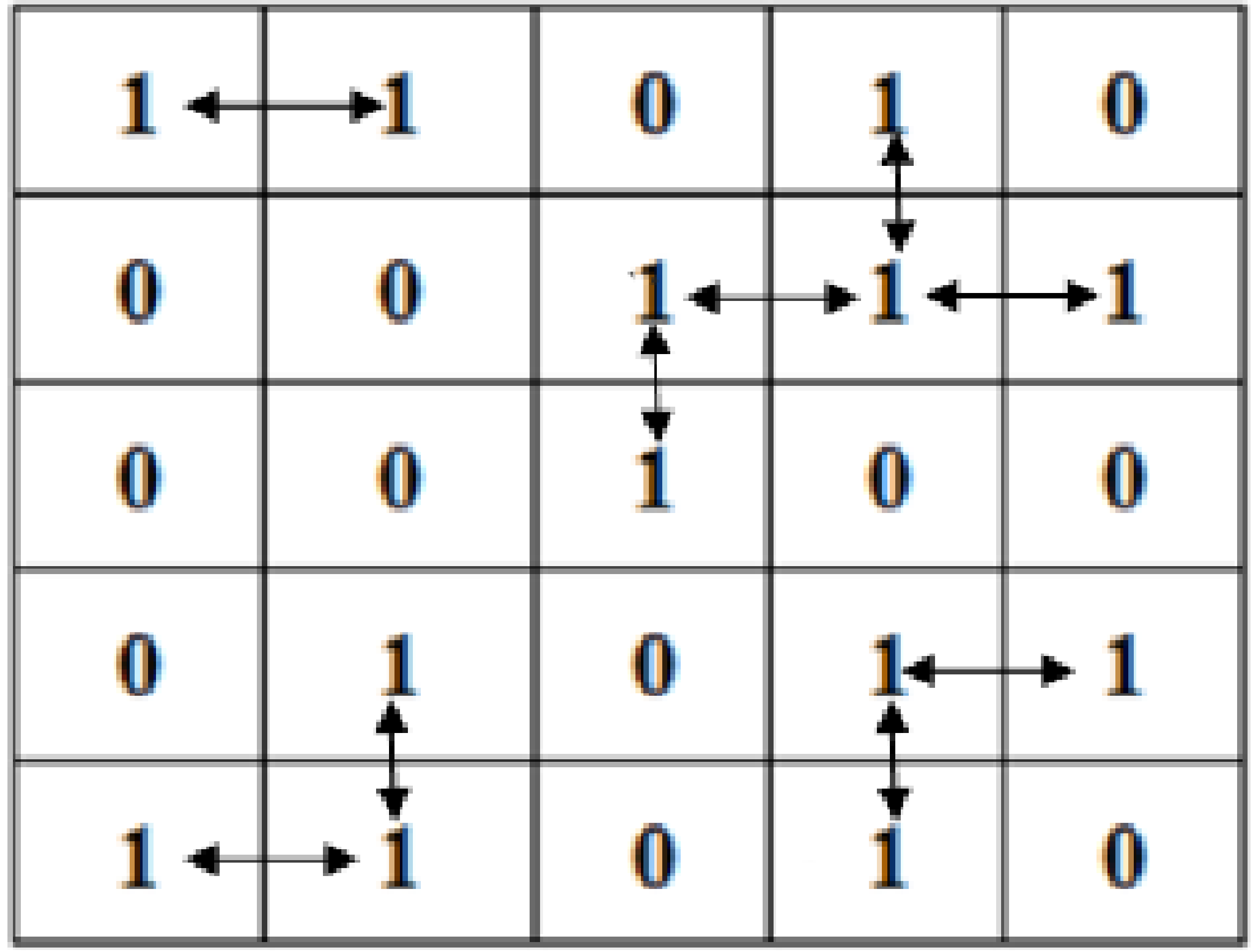
A set of pixels in an image which are all *connected* to each other is called a *connected component*.

4-connectivity:

Two pixels p and q with values from set V are 4- connected if q is in N_4 (p)



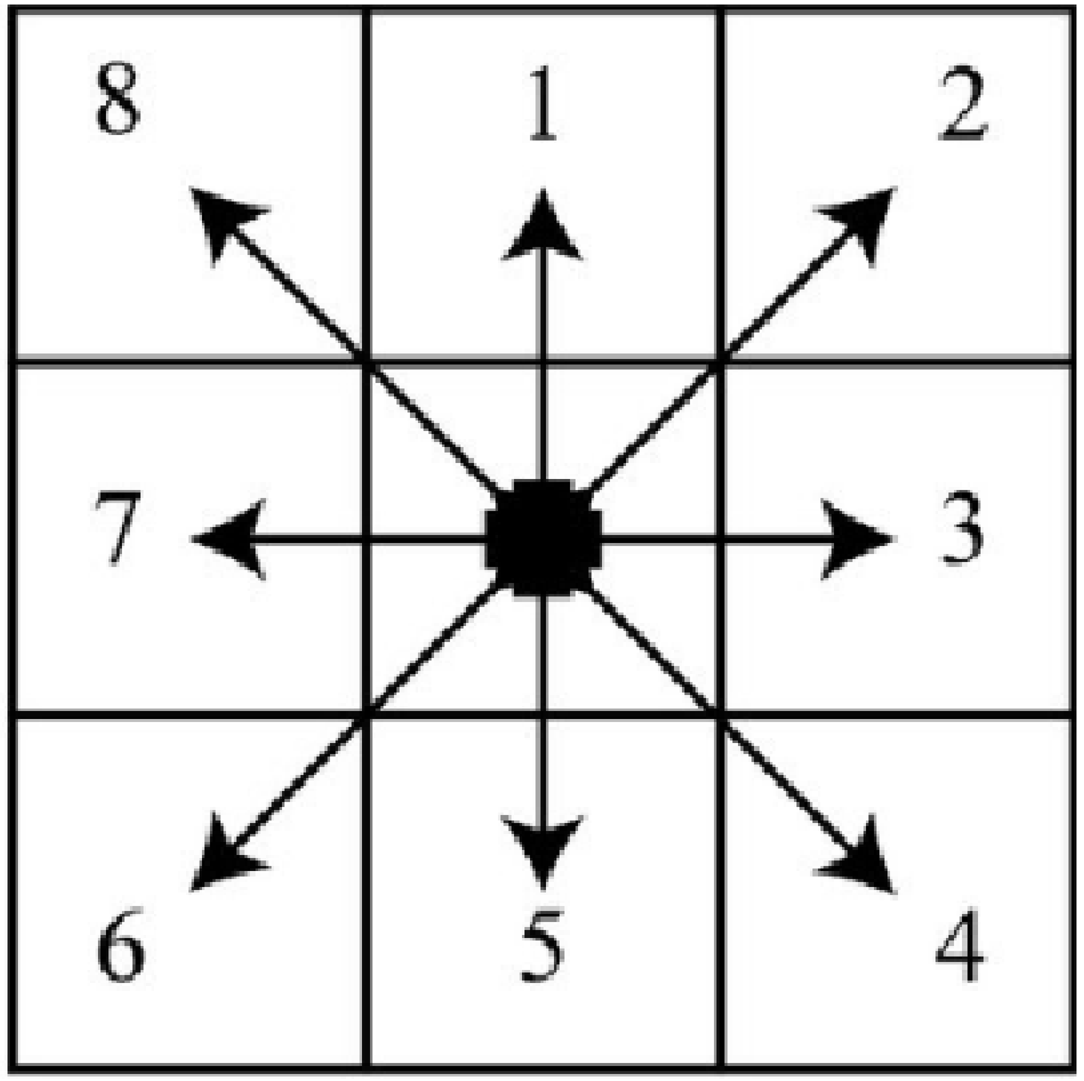
4-Connected



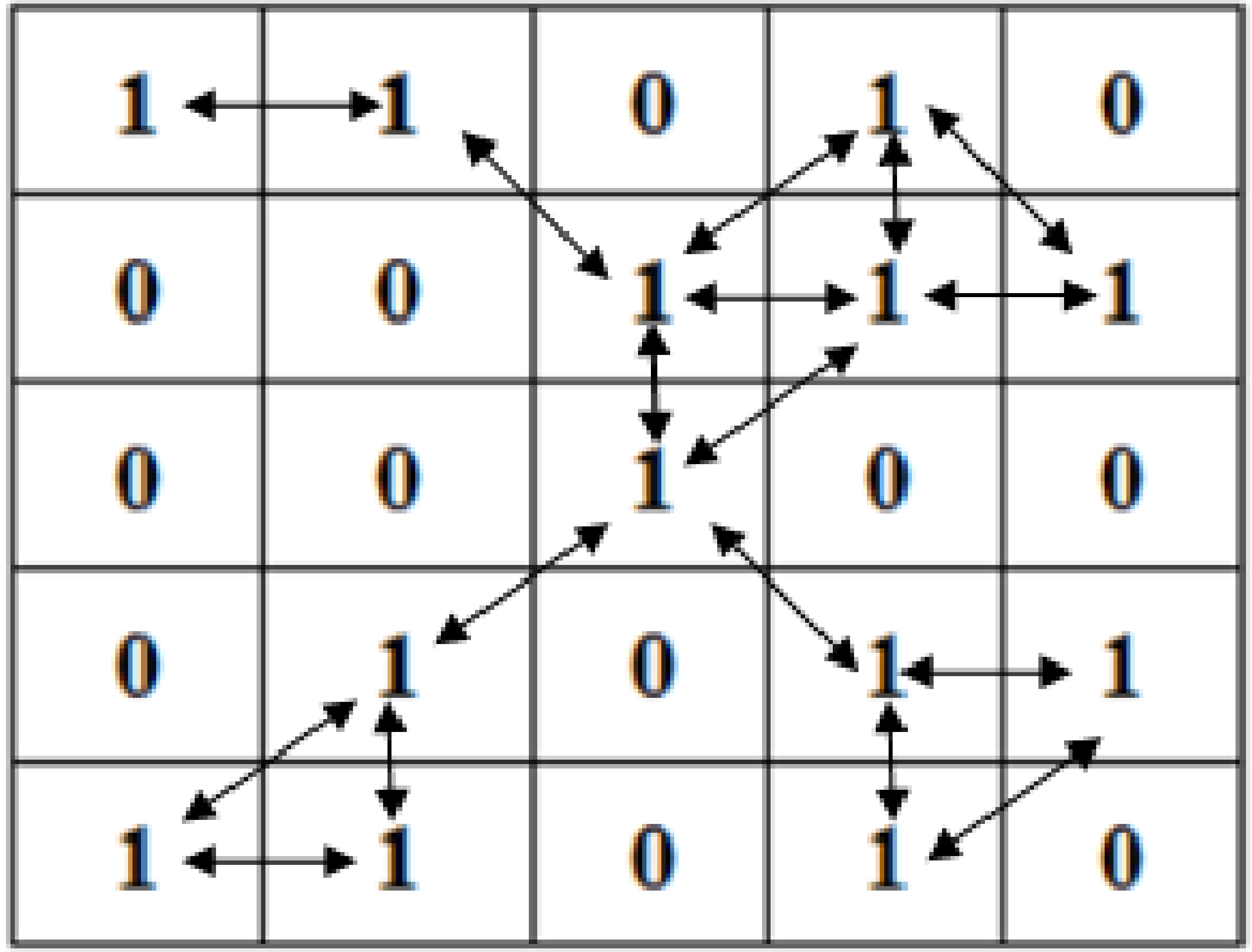
Example

8-connectivity :

Two pixels p and q with values from set V are 8- connected if q is in N_8 (p)



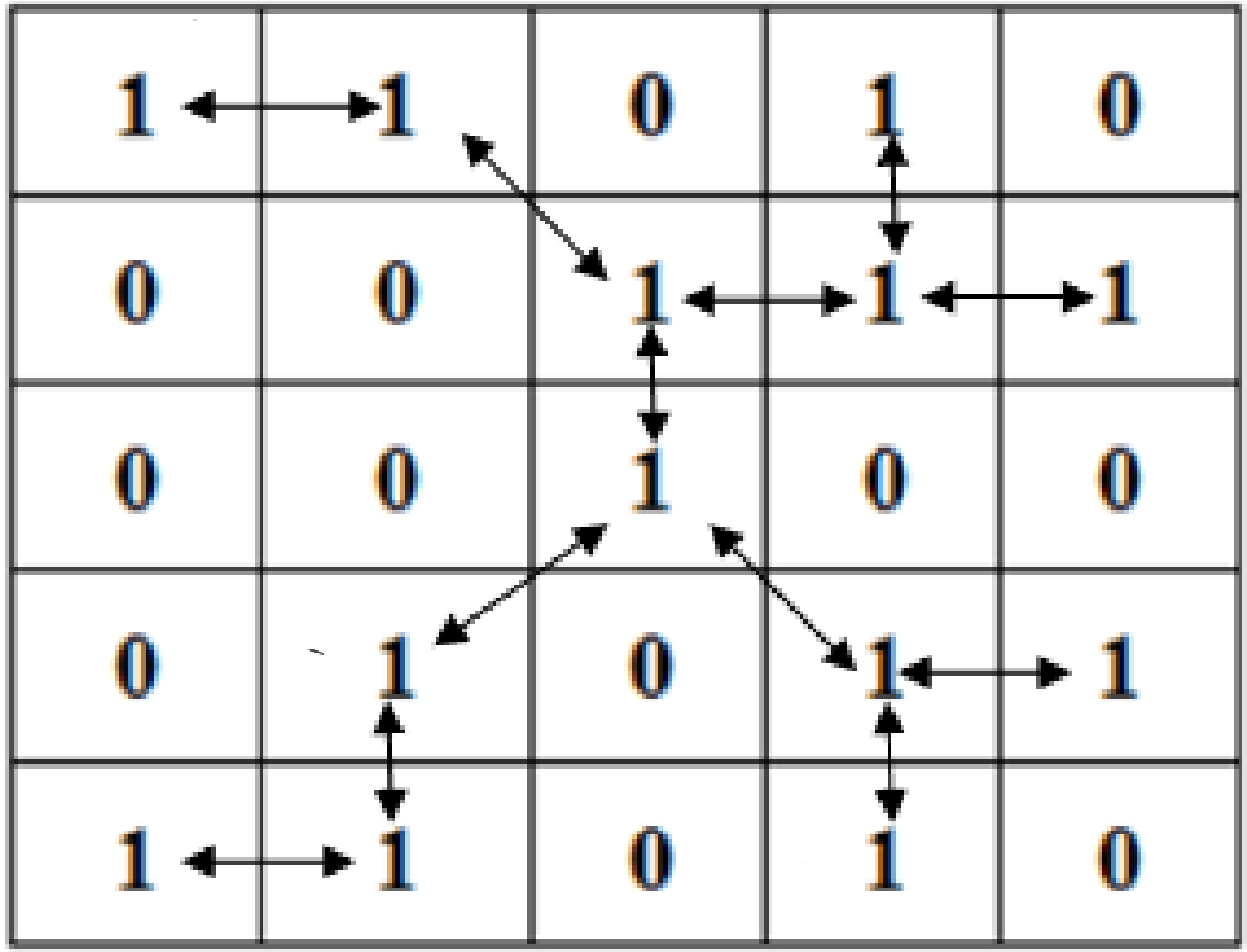
8-Connected



M-connectivity :

Mixed connectivity is modification of 8- connectivity and is introduced to eliminate the multiple paths that often arise when 8-connectivity is used

- Two pixels p and q with values from set V are M-connected
  - If q is in $N_4(p)$
  - If q is in $N_D(p)$, $N_4(p) \cap N_4(q)$ is empty i.e.
    $N_4(p) \cap N_4(q) = \emptyset$


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

Distance measures

For pixels p, q, and z, with coordinates (x,y), (s,t), and (v,w), respectively, D is a distance function or metric if

(a) $D(p,q) \geq 0$,

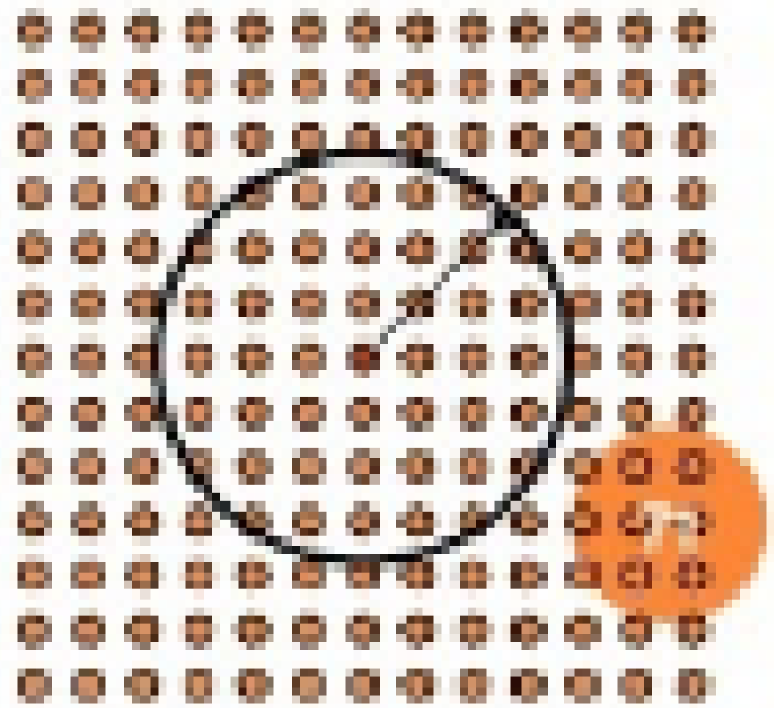
(b) $D(p,q) = D(q,p)$, (symmetry)

(c) $D(p,z) \leq D(p,q) + D(q,z)$ (triangular inequality)

Euclidean distance between p and q is

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

For this distance measure, the pixels having a distance less than or equal to some value r from (x,y) are the points contained in a disk of radius r centered at (x,y).

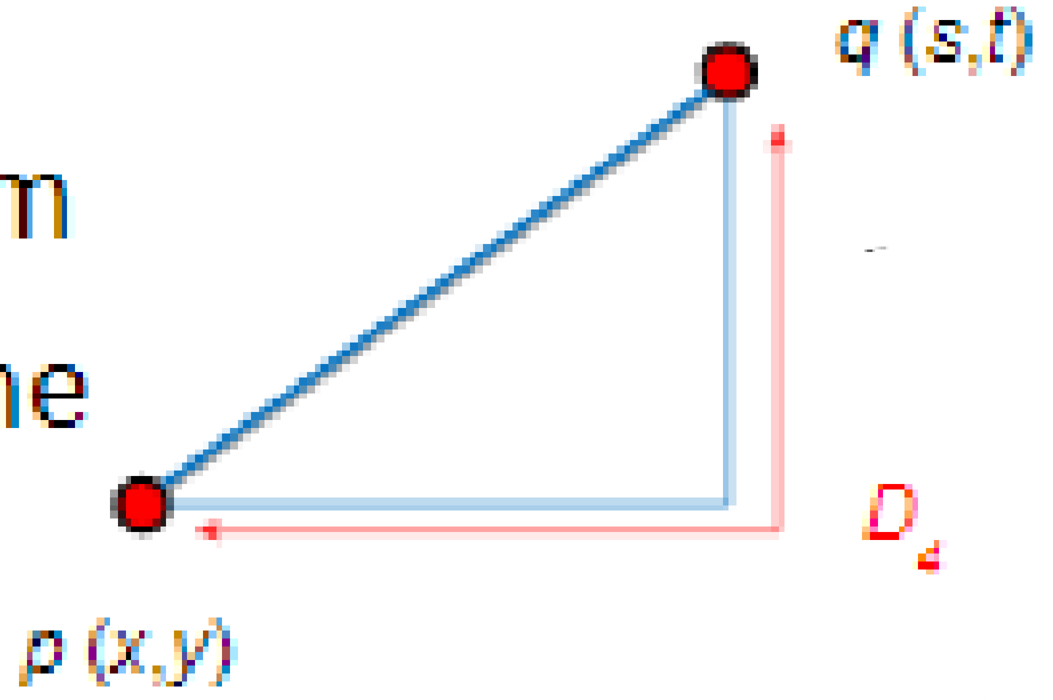


D_4 distance

- The D_4 *distance* (also called *city-block distance*) between *p* and *q* is defined as:

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

Pixels having a D_4 distance from (x,y), less than or equal to some value r form a Diamond centered at (x,y)

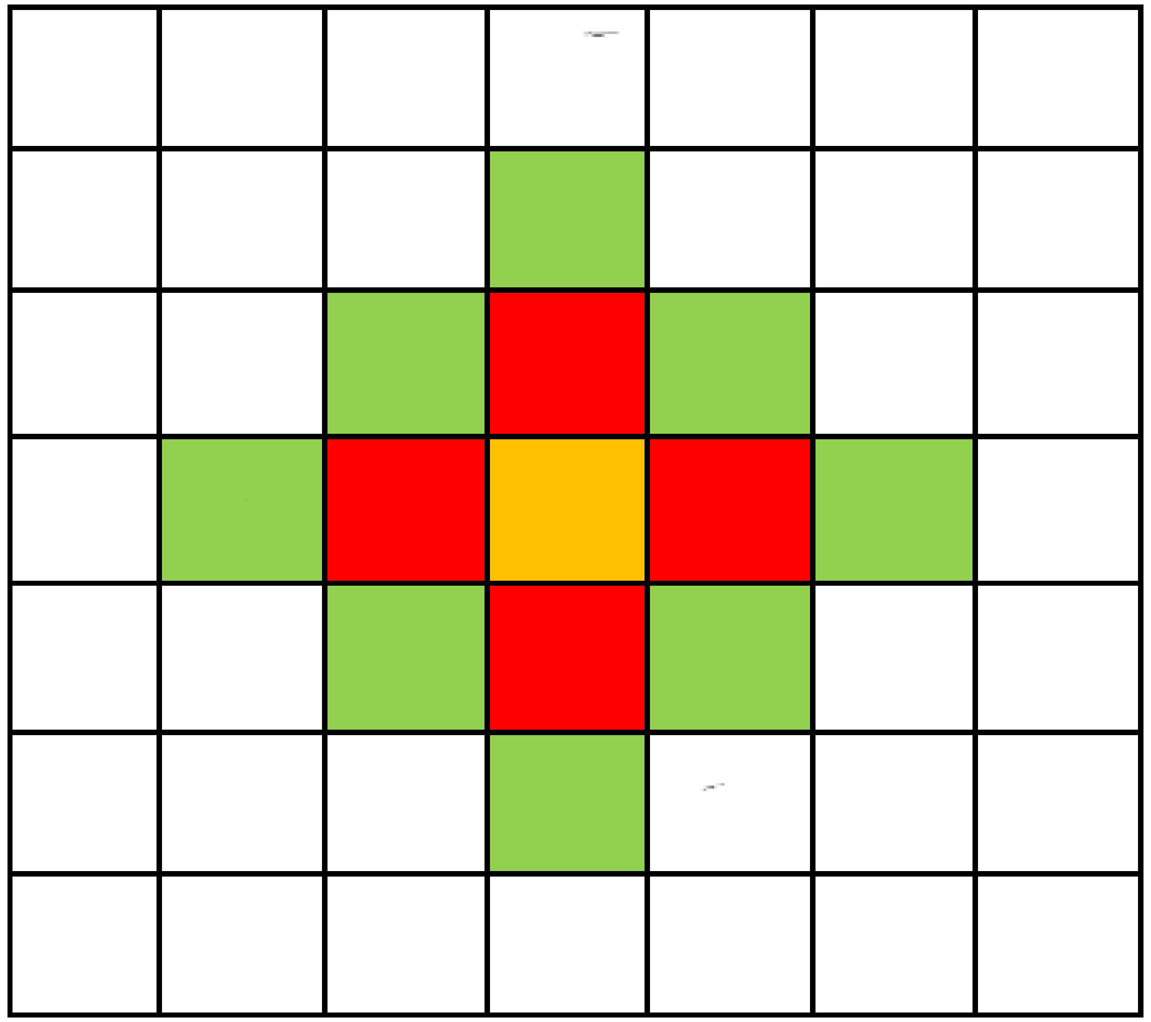


Example:

The pixels with distance $D_4 \leq 2$ from (x,y) form the following contours of constant distance.

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

The pixels with $D_4 = 1$ are the 4-neighbors of (x,y)

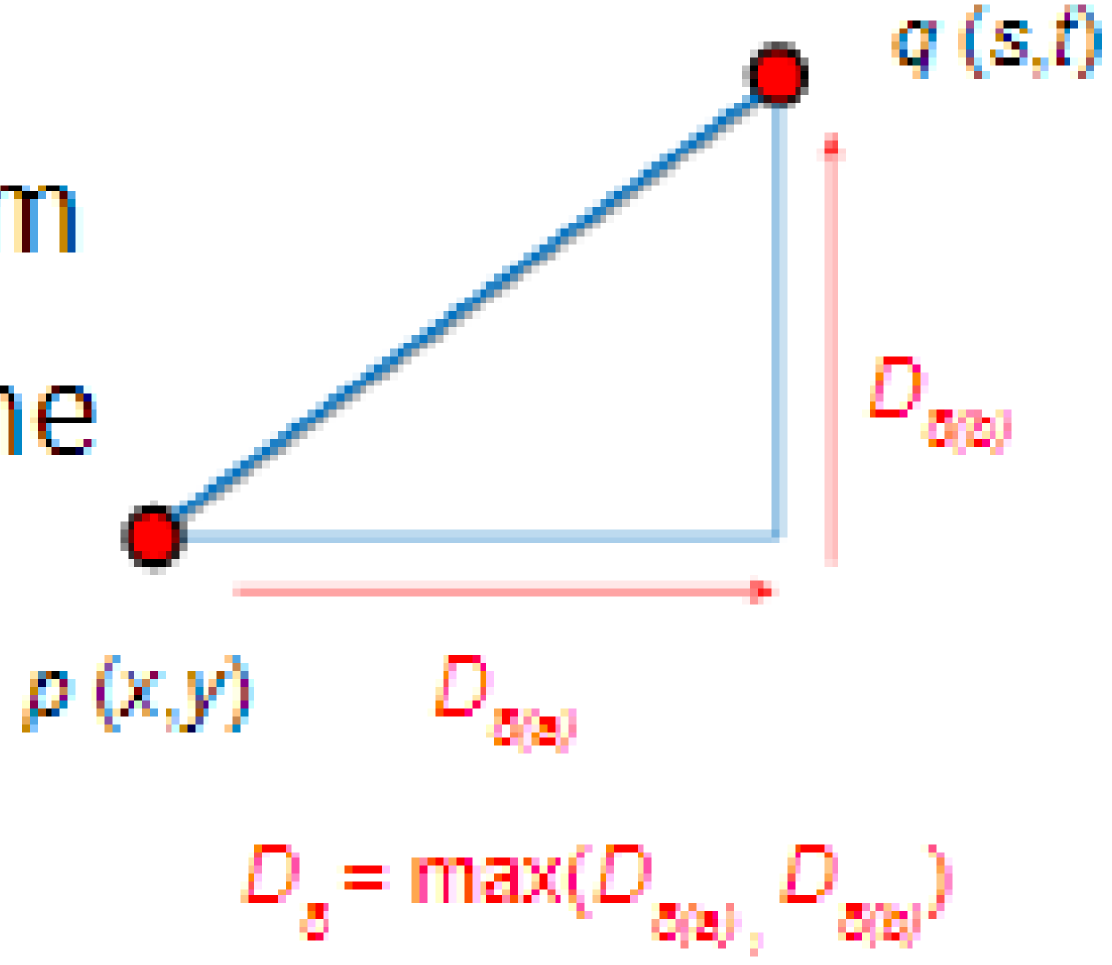


D_8 distance

- The D_8 *distance* (also called *chessboard distance*) between *p* and *q* is defined as:

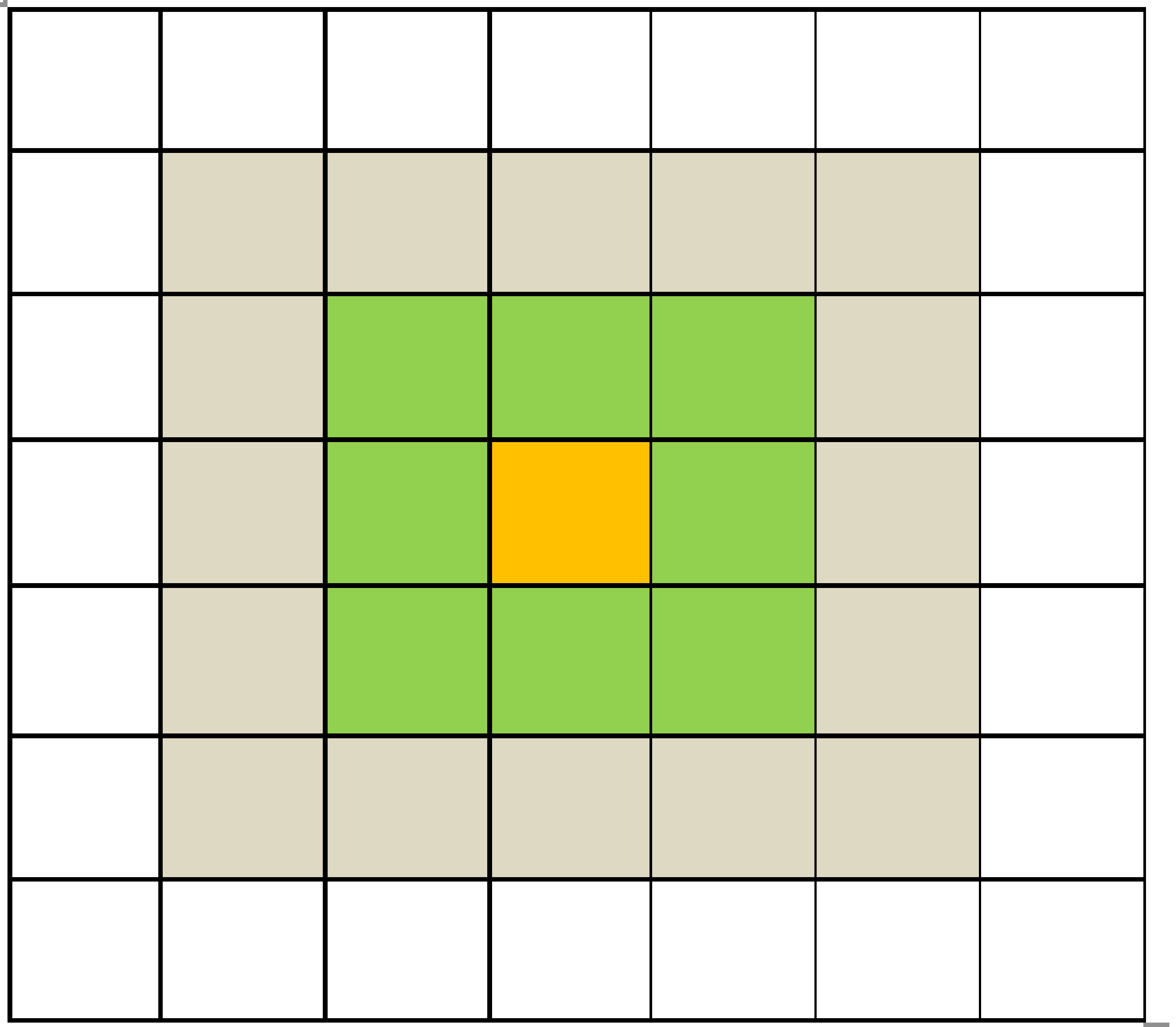
  $D_8(p,q) = \max(|x - s|, |y - t|)$

Pixels having a D_8 distance from (x,y), less than or equal to some value r form a square Centered at (x,y)



Example:

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

- **Dm distance:**

  is defined as the shortest m-path between the points.

  In this case, the distance between two pixels will depend on the values of the pixels along the path, as well as the values of their neighbors.

- Example:

  Consider the following arrangement of pixels and assume that p, p_2, and p_4 have value 1 and that p_1 and p_3 can have can have a value of 0 or 1

  Suppose that we consider the adjacency of pixels values 1 (i.e. $V = \{1\}$)

	p_3	p_4
p_1	p_2	
p		

- Cont. Example:

  Now, to compute the D_m between points p and p_4

  Here we have 4 cases:

  Case1: If p_1 =0 and p_3 = 0

  The length of the shortest m-path (the D_m *distance*) is 2 (p, p_2, p_4)

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

- Cont. Example:

  Case2: If p_1 =1 and p_3 = 0

  now, p_1 *and* p *will no longer be adjacent (see m-adjacency definition)*

  then, the length of the shortest path will be 3 (p, p_1, p_2, p_4)

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

- Cont. Example:

Case3: If p_1 =0 and p_3 = 1

The same applies here, and the shortest –m-path will be 3 (p, p_2, p_3, p_4)

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

	1	1
0	1	
1		

- Cont. Example:

Case4: If p_1 =1 and p_3 = 1

The length of the shortest m-path will be 4 (p, p_1, p_2, p_3, p_4)

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

	1	1
1	1	
1		

Arithmetic Operations

- Arithmetic operations between images are array operations. The four arithmetic operations are denoted as

$$f(x,y) = f1(x,y) + f2(x,y)$$

$$f(x,y) = f1(x,y) - f2(x,y)$$

$$f(x,y) = f1(x,y) \times f2(x,y)$$

$$f(x,y) = f1(x,y) \div f2(x,y)$$

Addition:

15	16	10	18	16
14	15	15	16	16
16	30	30	30	14
13	30	30	30	13
15	16	16	15	15

f1(x, y)

5	6	7	8	8
3	4	10	9	8
2	10	10	10	9
5	10	10	10	5
6	6	7	8	6

f2(x, y)

Example: Addition of Noisy Images for Noise Reduction

Noiseless image: f(x,y)

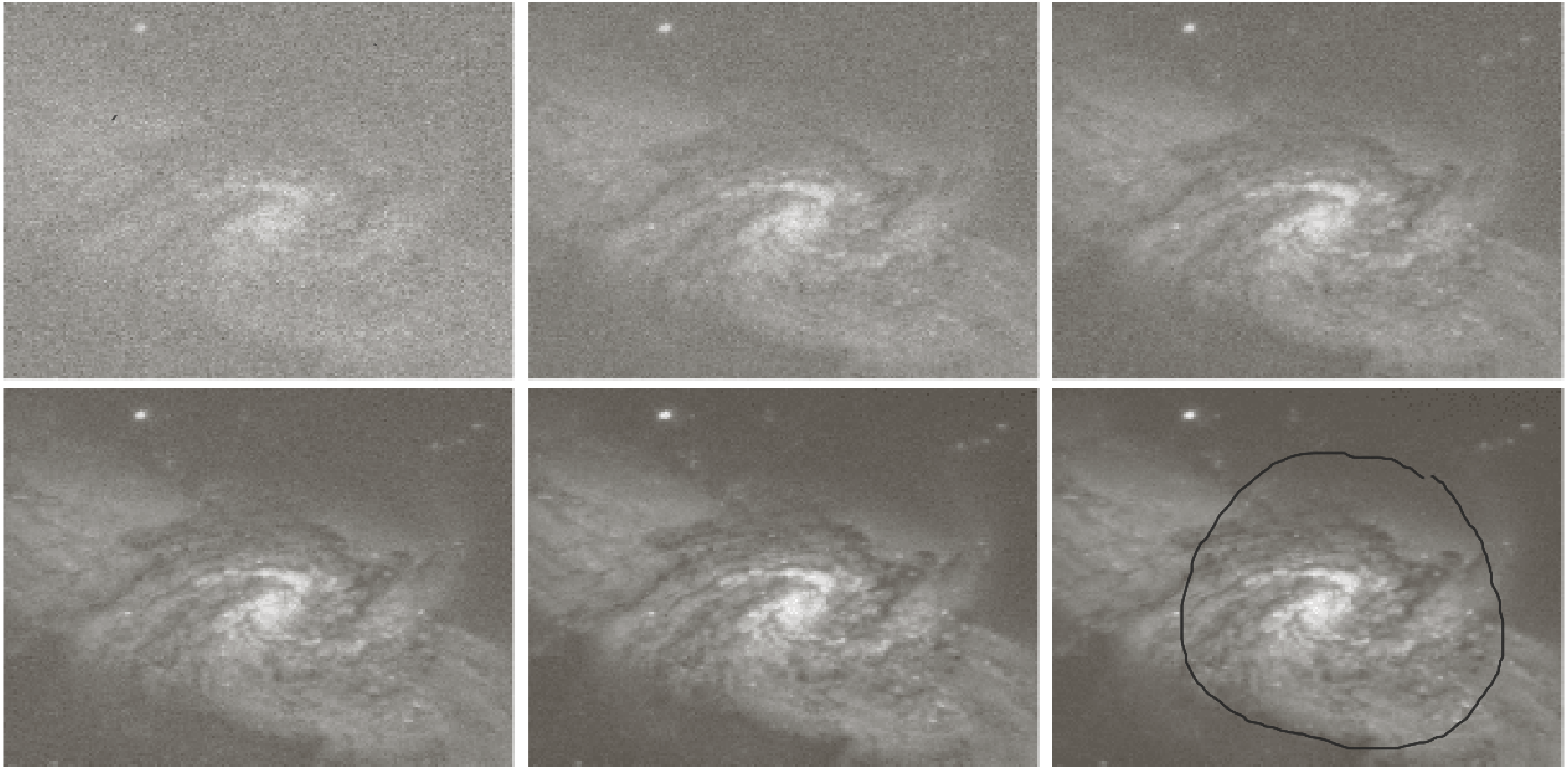
Noise: n(x,y) (at every pair of coordinates (x,y), the noise is uncorrelated and has zero average value)

Corrupted image: g(x,y)

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

Reducing the noise by adding a set of noisy images, $\{g_i(x,y)\}$

$$\overline{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y)$$



a	b	c
d	e	f

FIGURE 2.26 (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)–(f) Results of averaging 5, 10, 20, 50, and 100 noisy images, respectively. (Original image courtesy of NASA.)

Subtraction:

15	16	10	18	16
14	15	15	16	16
16	30	30	30	14
13	30	30	30	13
15	16	16	15	15

5	6	7	8	8
3	4	10	9	8
2	10	10	10	9
5	10	10	10	5
6	6	7	8	6

An Example of Image Subtraction: Mask Mode Radiography

Mask h(x,y): an X-ray image of a region of a patient's body

Live images f(x,y): X-ray images captured at TV rates after injection of the contrast medium

Enhanced detail g(x,y)

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

The procedure gives a movie showing how the contrast medium propagates through the various arteries in the area being observed.

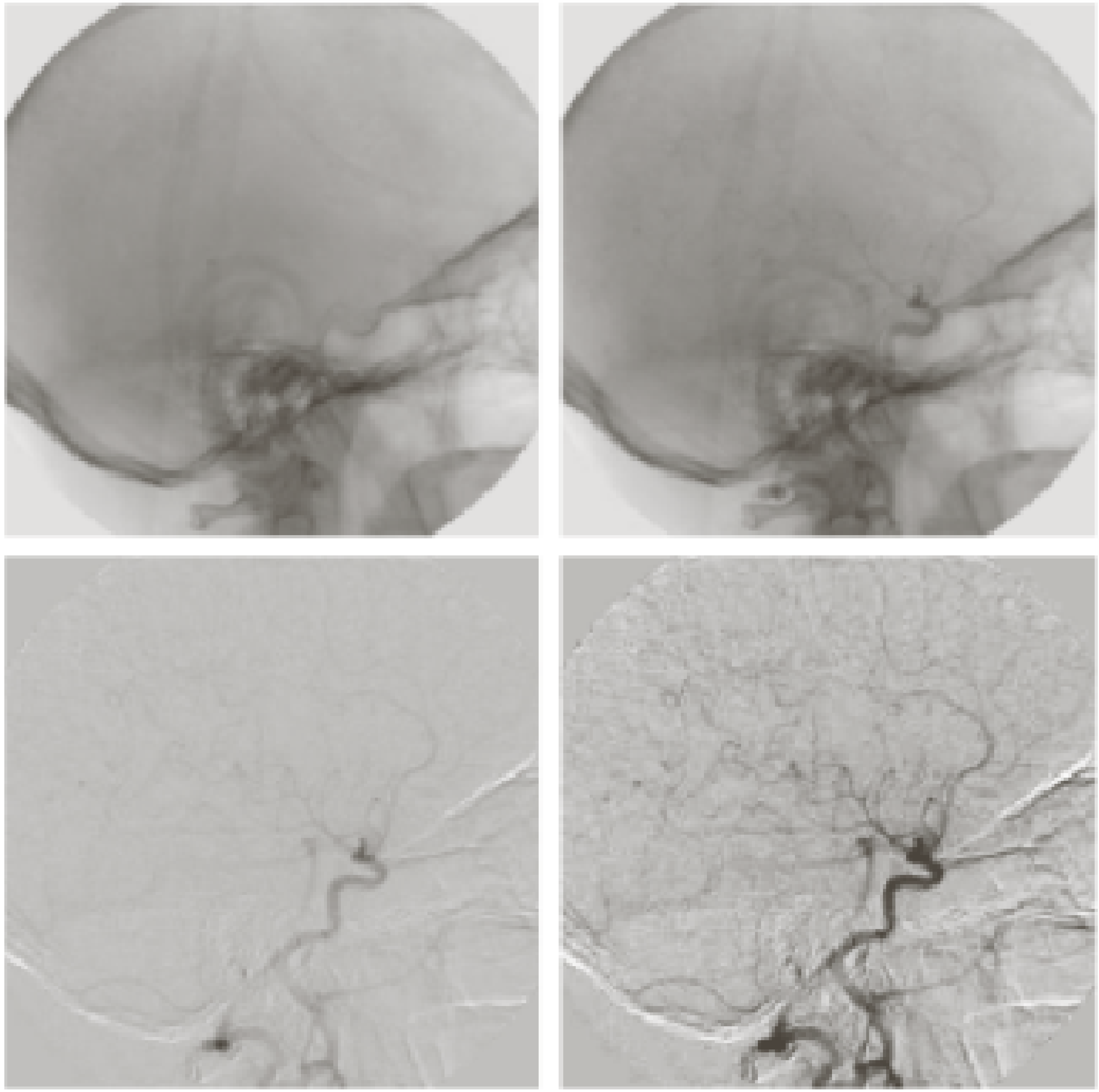


FIGURE 2.28 Digital subtraction angiography. (a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of The Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)

An Example of Image Multiplication

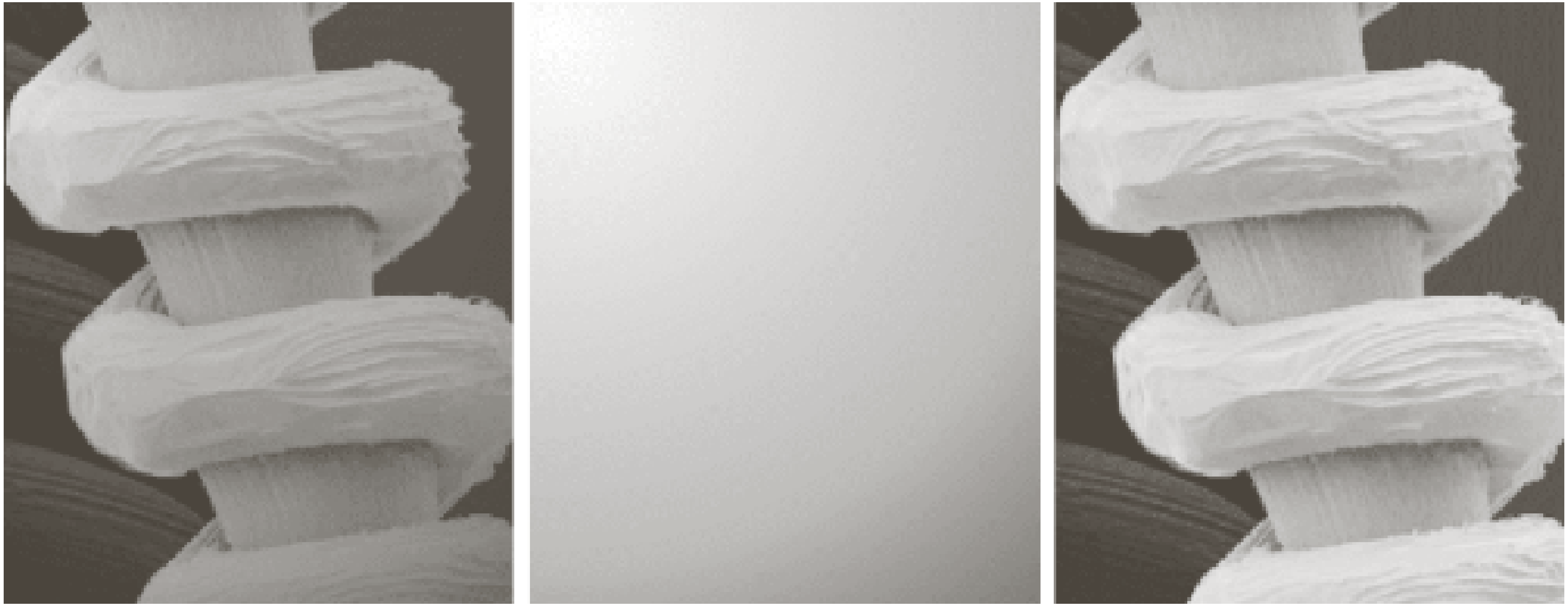


FIGURE 2.29 Shading correction. (a) Shaded SEM image of a tungsten filament and support, magnified approximately 130 times. (b) The shading pattern. (c) Product of (a) by the reciprocal of (b). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Logical Operations:

NOT, OR, AND, XOR

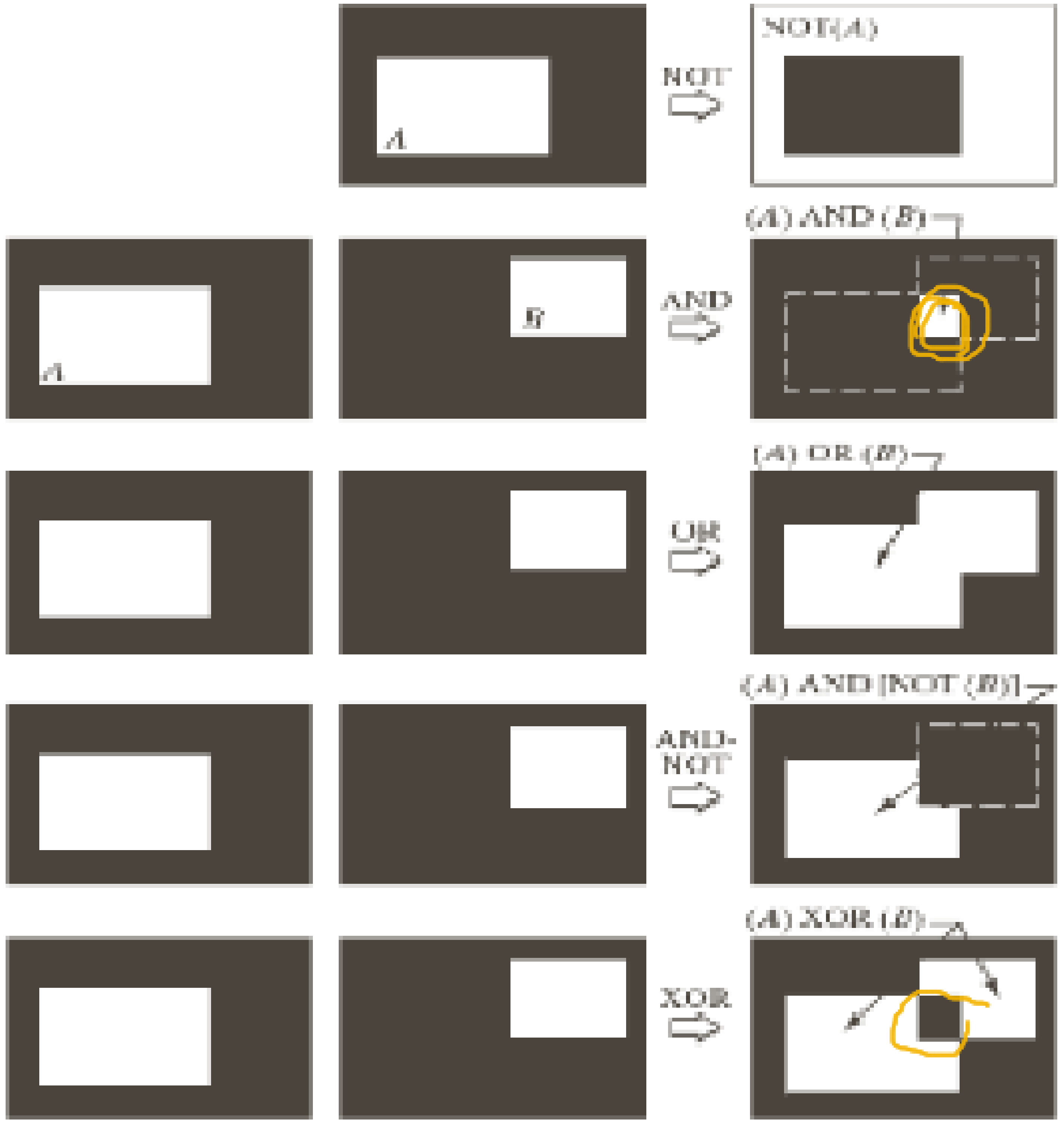
0	0	0	0	0	0
0	1	1	1	1	0
0	1	1	1	1	0
0	1	1	1	1	0
0	1	1	1	1	0
0	0	0	0	0	0

f1(x, y)

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

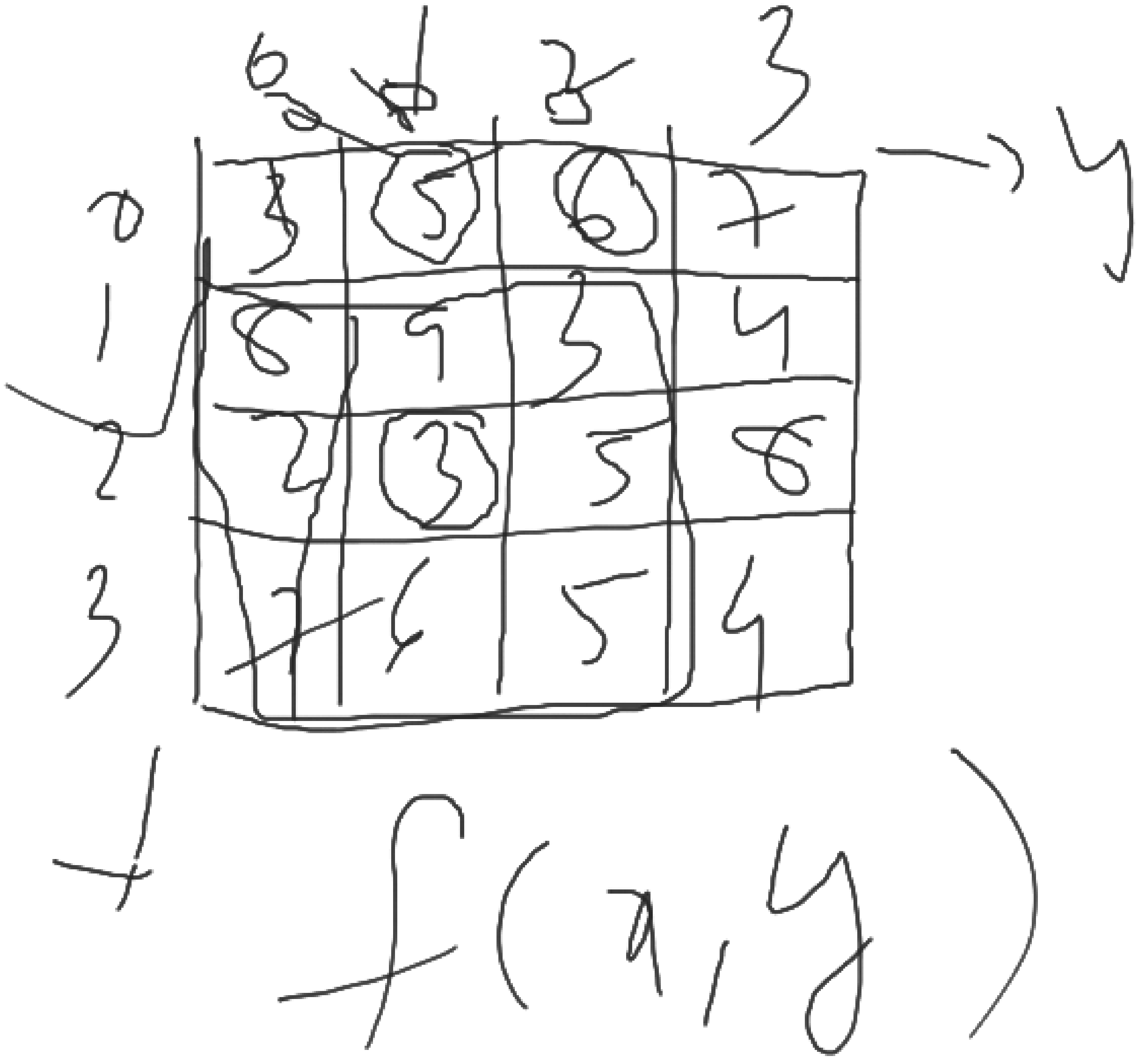
f2(x, y)

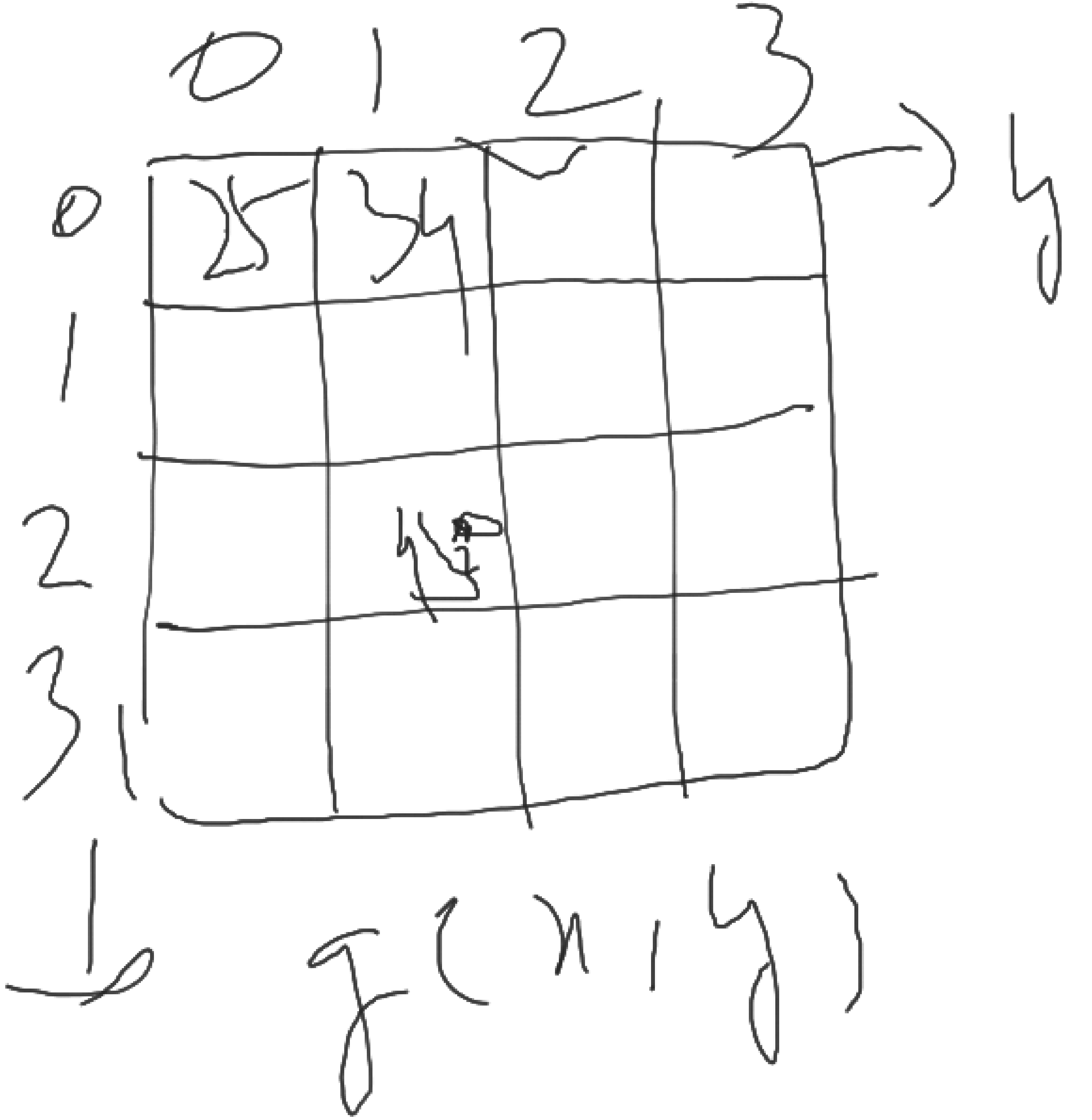
Logical Operations

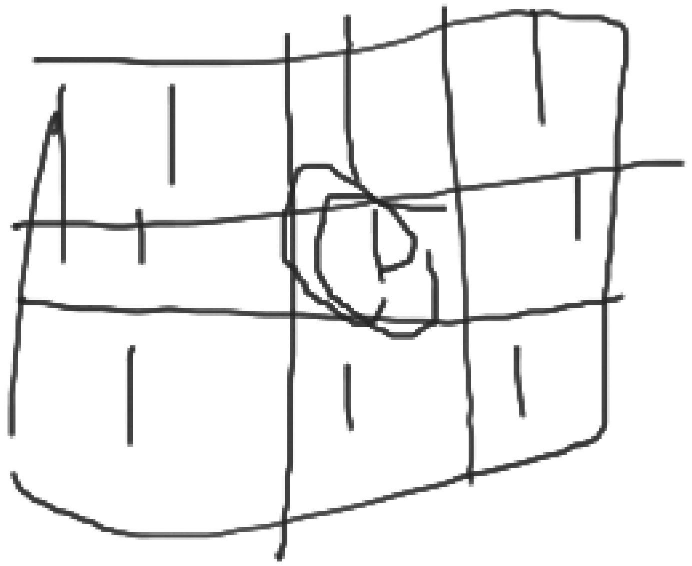


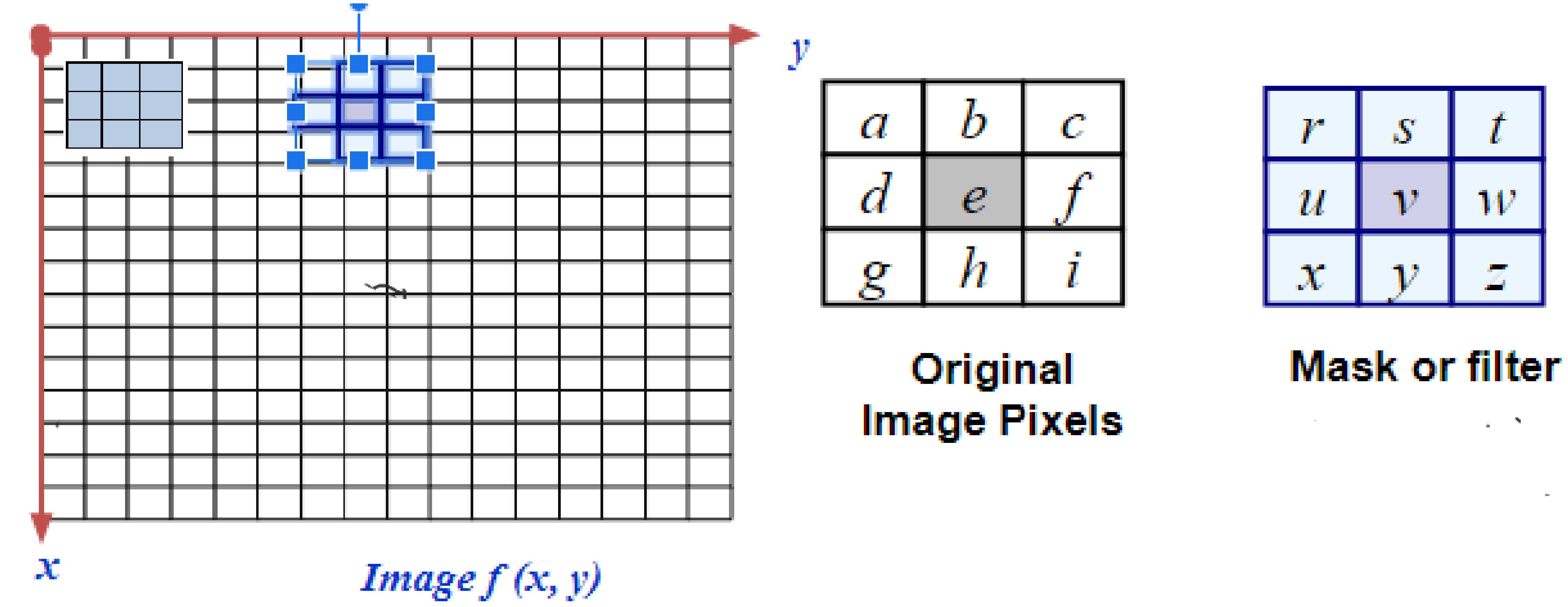
Neighbour hood oriented operations

- In addition to pixel by pixel operations on entire image, arithmetic and logical operations are used in neighbour hood oriented operations. These are called mask or window operations.


0 1 2 3
0 1 2 3
3 5 6 7
8 9 3 4
2 3 5 8
7 6 5 4
y
x
f(x, y)


0 1 2 3
0 1 2 3
25 34
y
x
g(x, y)





$$z_{processed} = v*e + r*a + s*b + t*c + u*d + w*f + x*g + y*h + z*i$$

Forward and inverse transform

Discrete Fourier transform (DFT) is a special class of transformation.
General **forward transformation** can be expressed as

$$T(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)g(x,y,u,v) \tag{1}$$

In case of DFT, $g(x,y,u,v) = \frac{1}{N}e^{-j\frac{2\pi}{N}(ux+vy)}$

Inverse transformation

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} T(u,v)h(x,y,u,v) \tag{2}$$

In case of I-DFT, $h(x,y,u,v) = \frac{1}{N}e^{j\frac{2\pi}{N}(ux+vy)}$

Walsh transform:

1-D Walsh transform:

When $N=2^n$, the kernel function is:

$$g(x,u)=\frac{1}{N}\prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)}$$

the discrete Walsh transform of a function *f(x)*, denote by *W(u)*, is:

$$W(u)=\frac{1}{N}\sum_{x=0}^{N-1}f(x)\prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)}$$

Where $b_k(z)$ is the kth bit in the binary representation of z.

Eg: $n=3$, $z=6$ (110 in binary), we have that

$b_0(z)=0$, $b_1(z)=1$, and $b_2(z)=1$

Inverse transformation kernel

$$h(x, u) = \prod_{i=0}^{n-1} (-1)^{b_i(x) b_{n-1-i}(u)}$$

Inverse transform

$$f(x) = \sum_{u=0}^{N-1} W(u) \prod_{i=0}^{n-1} (-1)^{b_i(x) b_{n-1-i}(u)}$$

2-D Walsh transform:

In case of 2D signal (forward transformation kernel)

$$g(x, y, u, v) = \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{b_i(x) b_{n-1-i}(u) + b_i(y) b_{n-1-i}(v)}$$

(Inverse transformation kernel)

$$h(x, y, u, v) = \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{b_i(x) b_{n-1-i}(u) + b_i(y) b_{n-1-i}(v)}$$

Forward transform

$$W(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v)}$$

Inverse transform

$$f(x,y) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} W(u,v)\prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v)}$$

Walsh transform kernal matrix for an image f(x, y) of size 4X4

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	-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	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	-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	-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	-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

Properties of Walsh transform

- The transform is separable and symmetric.
- Transform is orthogonal transform.
- The coefficients near origin have maximum energy and it reduces as we go further away from the origin.
- It has energy compaction property but not strong as in DCT.
- Kernel values are real values +1 or -1.

Hadamard transform

When $N=2^n$, the kernel function is:

$$g(x,u)=\frac{1}{N}(-1)^{\sum_{i=0}^{n-1} b_i(x)b_i(u)}$$

Where the summation in the exponent is performed in modulo 2

1-D Hadamard transform of a function $f(x)$, denote by $H(u)$, is:

$$H(u)=\frac{1}{N}\sum_{x=0}^{N-1} f(x)(-1)^{\sum_{i=0}^{n-1} b_i(x)b_i(u)}$$

Inverse kernel and transform:

$$h(x,u) = (-1)^{\sum_{i=0}^{n-1} b_i(x) b_i(u)}$$

$$f(x) = \sum_{u=0}^{N-1} H(u)(-1)^{\sum_{i=0}^{n-1} b_i(x) b_i(u)}$$

For 2D signal

$$g(x, y, u, v) = \frac{1}{N}(-1)^{\sum_{i=0}^{n-1} b_i(x)b_i(u)+b_i(y)b_i(v)},$$

and

$$h(x, y, u, v) = \frac{1}{N}(-1)^{\sum_{i=0}^{n-1} b_i(x)b_i(u)+b_i(y)b_i(v)}$$

Forward and inverse kernel are identical.

Properties of Hadamard transform

- The transform is separable and symmetric.
- Transform is orthogonal transform.
- The coefficients near origin have maximum energy and it reduces as we go further away from the origin.
- It has energy compaction property but not strong as in DCT.
- Kernel values are real values +1 or -1.

$$k = 0, 1, 2, 3$$

$$h_0(z) = \frac{1}{\sqrt{N}} = \frac{1}{\sqrt{4}} = \frac{1}{2}$$

$$\text{for } z = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$$

$$k = 1, \; p = 0, \; q = 1$$

$$h_1(z) = \frac{1}{\sqrt{4}} \begin{cases} 2^{0/2} & \frac{1-1}{2^0} \le z < \frac{1/2}{2^0} \\ -2^{0/2}, & \frac{1-1/2}{2^0} \le z < \frac{1}{2^0} \\ 0 & \end{cases}$$

$$h_1(z) = \frac{1}{2} \begin{cases} 1, & 0 \le z < 1/2 \\ -1, & 1/2 \le z < 1 \\ 0, & \text{otherwise} \end{cases}$$

$k = 2, \; p = 1, \; q = 1$

$$h_2(z) = \frac{1}{\sqrt{4}} \begin{cases} 2^{1/2}, & \frac{1-1}{2^1} \le z < \frac{1-1/2}{2^1} \\ -2^{1/2}, & \frac{1-1/2}{2} \le z < \frac{1}{2} \\ 0, & \text{otherwise} \end{cases}$$

$$h_2(z) = \frac{1}{2}\begin{cases} \sqrt{2}, & 0 \le z < 1/4 \\ -\sqrt{2}, & \frac{1}{4} \le z < 1/2 \\ 0, & \text{otherwise} \end{cases}$$

$K = 3, \ p = 1, \ q = 2$

$$h_3(z) = \frac{1}{\sqrt{4}}\begin{cases} 2^{1/2}, & \frac{2-1}{2^1} \le z < \frac{2-1/2}{2^1} \\ -2^{1/2}, & \frac{2-1/2}{2^1} \le z < \frac{1}{1} \\ 0, & \text{otherwise} \end{cases}$$

$$h_3(t) = \frac{1}{2}\begin{cases} \sqrt{2}, & \frac{1}{2} \le t < \frac{3}{4} \\ -\sqrt{2}, & \frac{3}{4} \le t < 1 \\ 0, & \text{otherwise} \end{cases}$$

$$H_4 = \begin{array}{c|cccc} k \backslash t & 0 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} \\ \hline 0 & 1/2 & 1/2 & 1/2 & 1/2 \\ 1 & 1/2 & 1/2 & -1/2 & -1/2 \\ 2 & 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 \\ 3 & 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} \end{array}$$

$$\boxed{16 \times 16}$$

$$W(u,v) = 4 \times 4$$

$$\begin{bmatrix} W(0,0) & W(0,1) & \cdots & W(0,3) \\ W(1,0) & \cdots & \cdots & W(1,3) \\ \vdots & & & \\ W(3,0) & \cdots & \cdots & W(3,3) \end{bmatrix}$$

$f(x, y) \mid 4 \times 4 \quad 16 \times 16$

$H(u, v) \tau\ 4 \times 4$

$u, v = 0, 1 \ldots 3$

$x, y \mid$

$$H(u, v) = \sum_{x=0}^{3} \sum_{y=0}^{3} f(x, y)\, g(x, y, u, v)$$

$H(0, 0)$

Histogram processing

- The histogram of digital image with gray levels in the range [0, L-1] is a discrete function
- $h(r_k) = n_k$

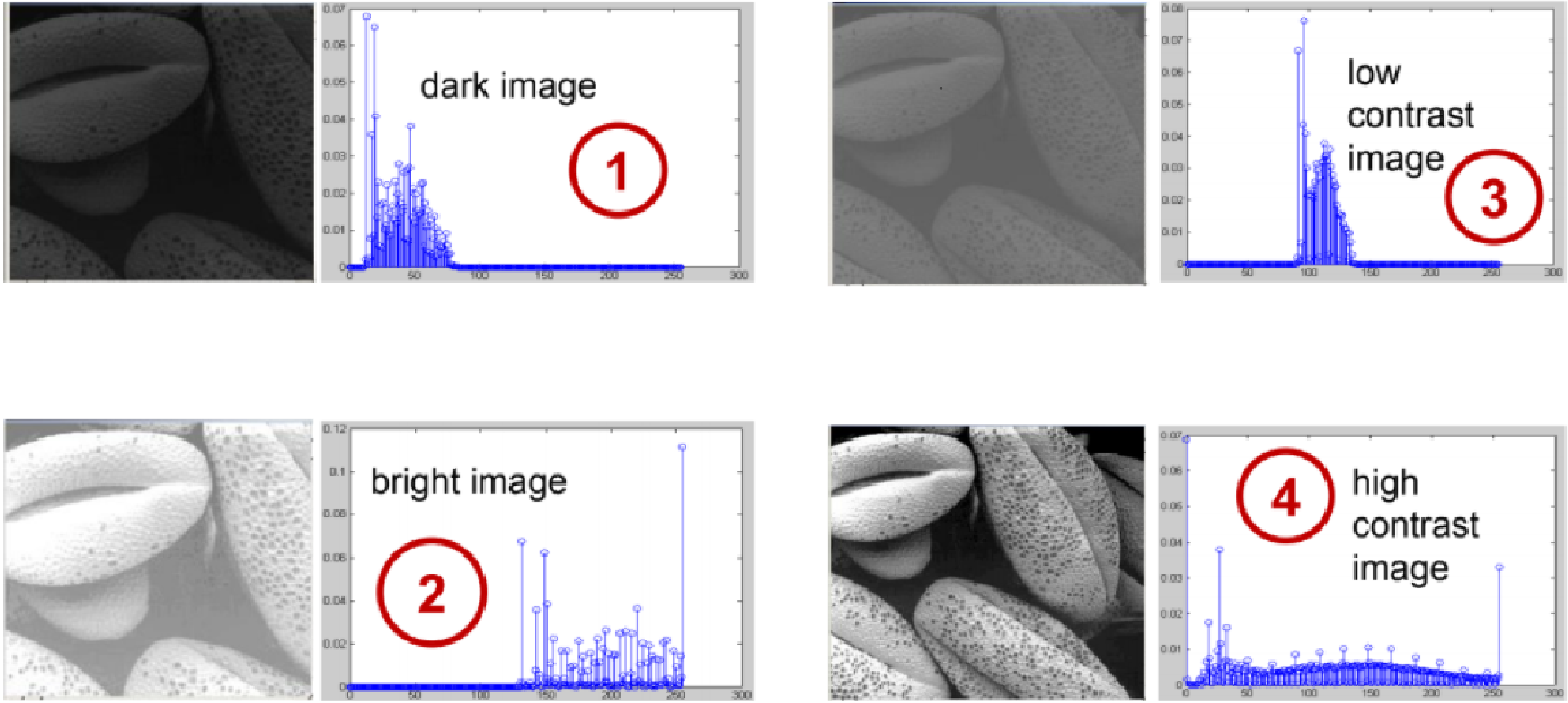
  r_k: kth gray level

  n_k: number of pixels in image having gray levels r_k
- Normalized histogram

  $p(r_k) = n_k/n$

  n: total number of pixels in image

  n = MN (M: row dimension, N: column dimension)


dark image
1
low contrast image
3
bright image
2
4
high contrast image

Histogram Techniques classified into two types

1. Histogram equalization
2. Histogram specification or matching

Histogram equalization: if r is continuous variable

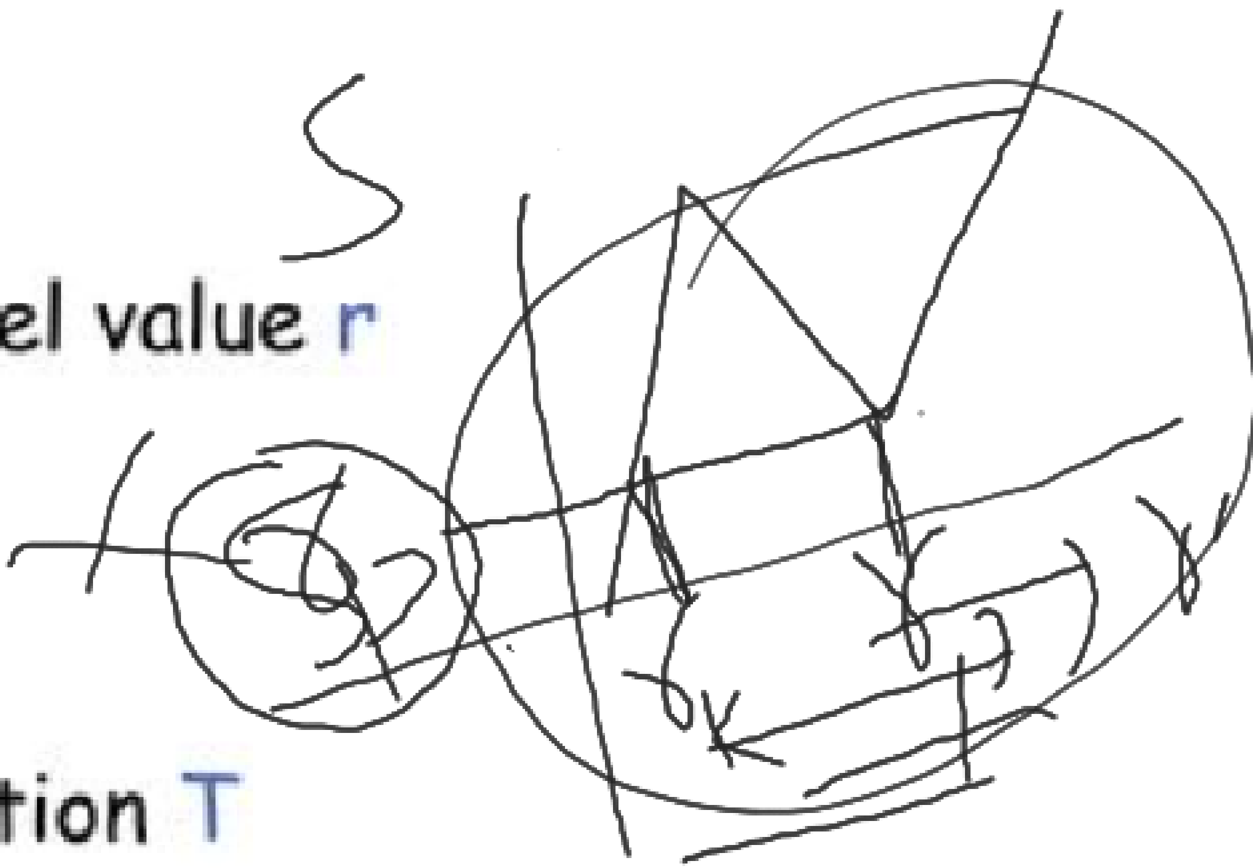
- r: intensities of the image to be enhanced

  r is in the range [0, L-1]

  r = 0: black, r = L-1: white

  s: processed gray levels for every pixel value r

- $s = T(r), \quad 0 \le r \le L-1$



- Requirements of transformation function T

(a) T(r) is a (strictly) monotonically increasing in the interval

$0 \le r \le L-1$

(b) $0 \le T(r) \le L-1$ for $0 \le r \le L-1$

- Inverse transformation

  $r = T^{-1}(s), \quad 0 \le s \le L-1$

Intensity levels: random variable in interval [0, L-1]

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

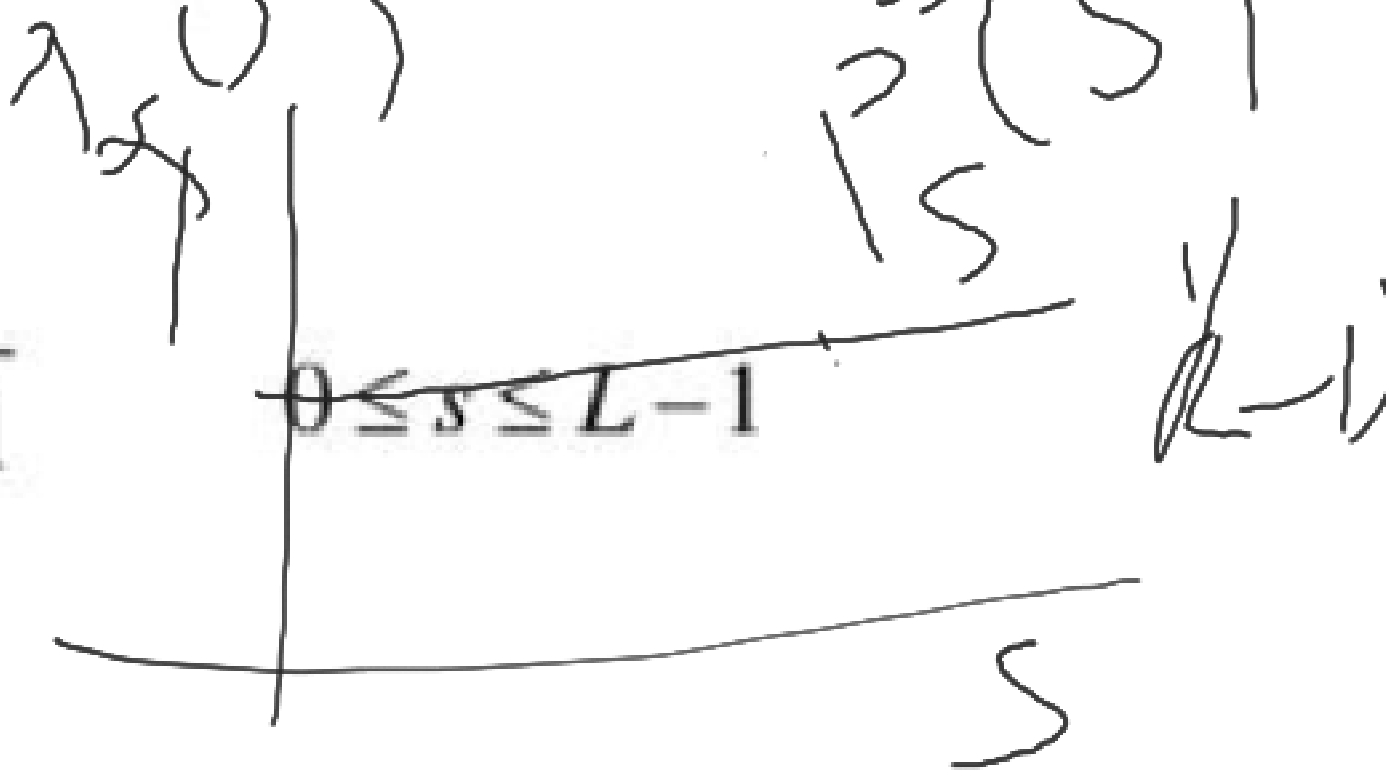
probability density function (PDF)

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$

cumulative distribution function (CDF)

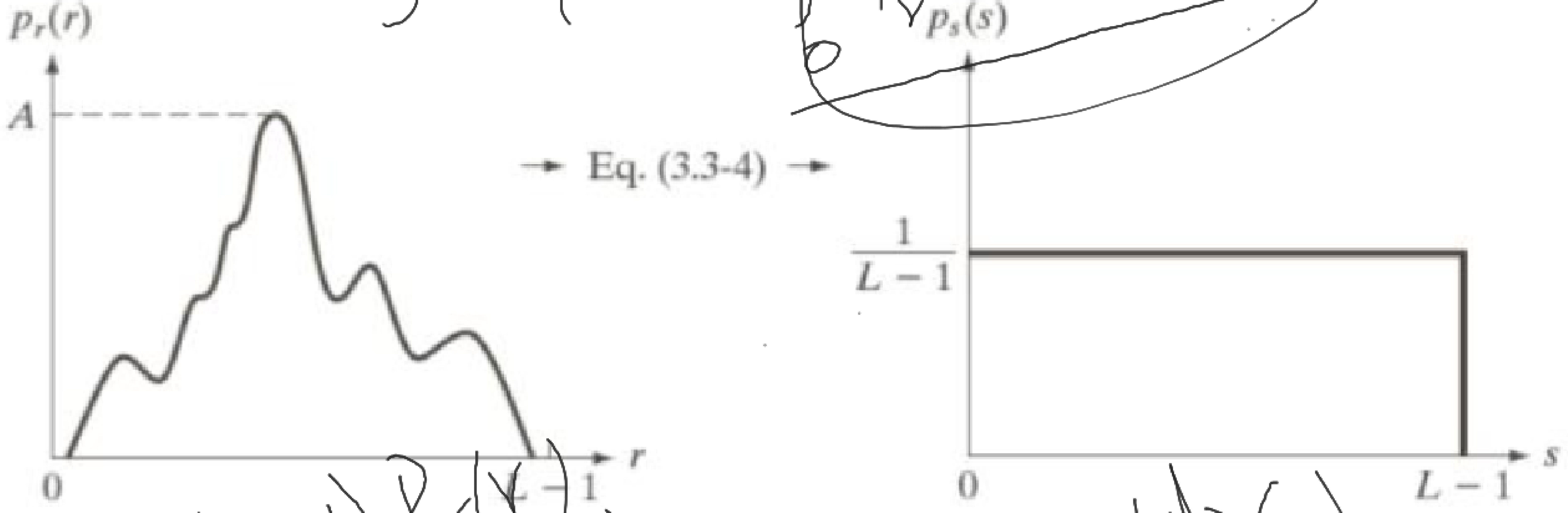
$$\frac{ds}{dr} = \frac{dT(r)}{dr} = (L-1)\frac{d}{dr}\left[\int_0^r p_r(w)\,dw\right] = (L-1)\,p_r(r)$$

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = p_r(r)\left|\frac{1}{(L-1)\,p_r(r)}\right| = \frac{1}{L-1} \qquad 0 \le s \le L-1$$



Uniform probability density function

$$s = (L-1)\int_0^r p_r(w)\,dw$$



$$\frac{ds}{dr} = (L-1)\,p_r(r)$$

$$\frac{dr}{ds} = \frac{1}{(L-1)\,p_r(r)}$$

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

$$p_s(s) = p_r(r) \times \frac{1}{(L-1)\,p_r(r)}$$

Discrete Case:

$$p_r(r_k) = \frac{n_k}{MN} \qquad k = 0, 1, 2, \ldots, L-1$$

MN: total number of pixels in image

n_k: number of pixels having gray level r_k

L: total number of possible gray levels

$r_0, r_1 \ldots r_{L-1}$

$r \text{ — } s$

$s_K = T(r_K)$

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{L-1}{MN}\sum_{j=0}^{k} n_j \qquad k = 0, 1, 2, .., L-1$$

- histogram equalization (histogram linearization):

  Processed image is obtained by mapping each pixel r_k (input image) into corresponding level s_k (output image)

$k = 0, 1, 2 \ldots L-1$

$r_0, r_1 \ldots r_{L-1}$

$$S_k = T(r_k) = (L-1) \sum_{j=0}^{k} P_r(r_j)$$

$$S_0 = T(r_0) = (L-1) \sum_{j=0}^{0} P_r(r_j)$$

$$= (L-1)\, P_r(r_0)$$

$$S_1 = T(r_1) = (L-1) \sum_{j=0}^{1} P_r(r_j) = (L-1)\left[P_r(r_0) + P_r(r_1)\right]$$

→ Perform histogram equalization of the image

$$f(x,y) \rightarrow \begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 3 & 4 & 5 & 4 & 3 \\ 3 & 5 & 5 & 5 & 3 \\ 3 & 4 & 5 & 4 & 3 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix}$$

$n = 25$

Highest gray level in image = 5

no. of bits used to represent each gray level = 3

$P_r(r_k) = n_k / n$, $n = 25$

r_k	r_0 = 0	r_1 = 1	r_2 = 2	r_3 = 3	r_4 = 4	r_5 = 5	r_6 = 6	r_7 = 7
n_k	0	0	0	6	14	5	0	0
$P_r(r_k)$	$\frac{0}{25}$	$\frac{0}{25}$	$\frac{0}{25}$	$\frac{6}{25}$	$\frac{14}{25}$	$\frac{5}{25}$	$\frac{0}{25}$	$\frac{0}{25}$
CDF	$\frac{0}{25}$	$\frac{0}{25}$	$\frac{0}{25}$	$\frac{6}{25}$	$\frac{20}{25}$	$\frac{25}{25}$	$\frac{25}{25}$	$\frac{25}{25}$
CDF × 7	0	0	0	1.68	5.6	7	7	7
S_k	0	0	0	2	6	7	7	7

$f(x,y)$ r_k

$f(x,y) \rightarrow$ (r_k)

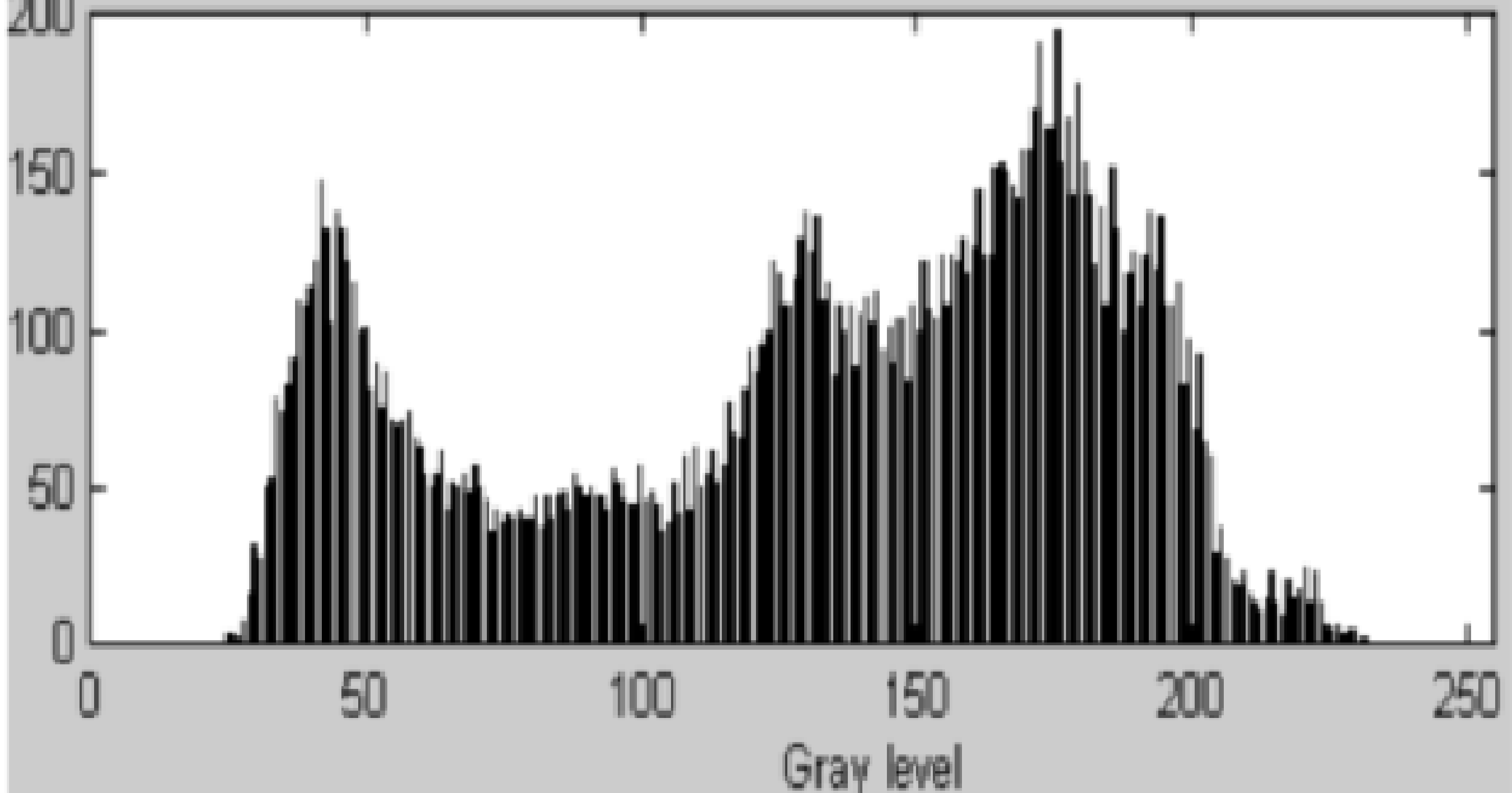
4	4	4	4	4
3	4	5	4	3
3	5	5	5	3
3	4	5	4	3
4	4	4	4	4

$g(x,y)$ s_k

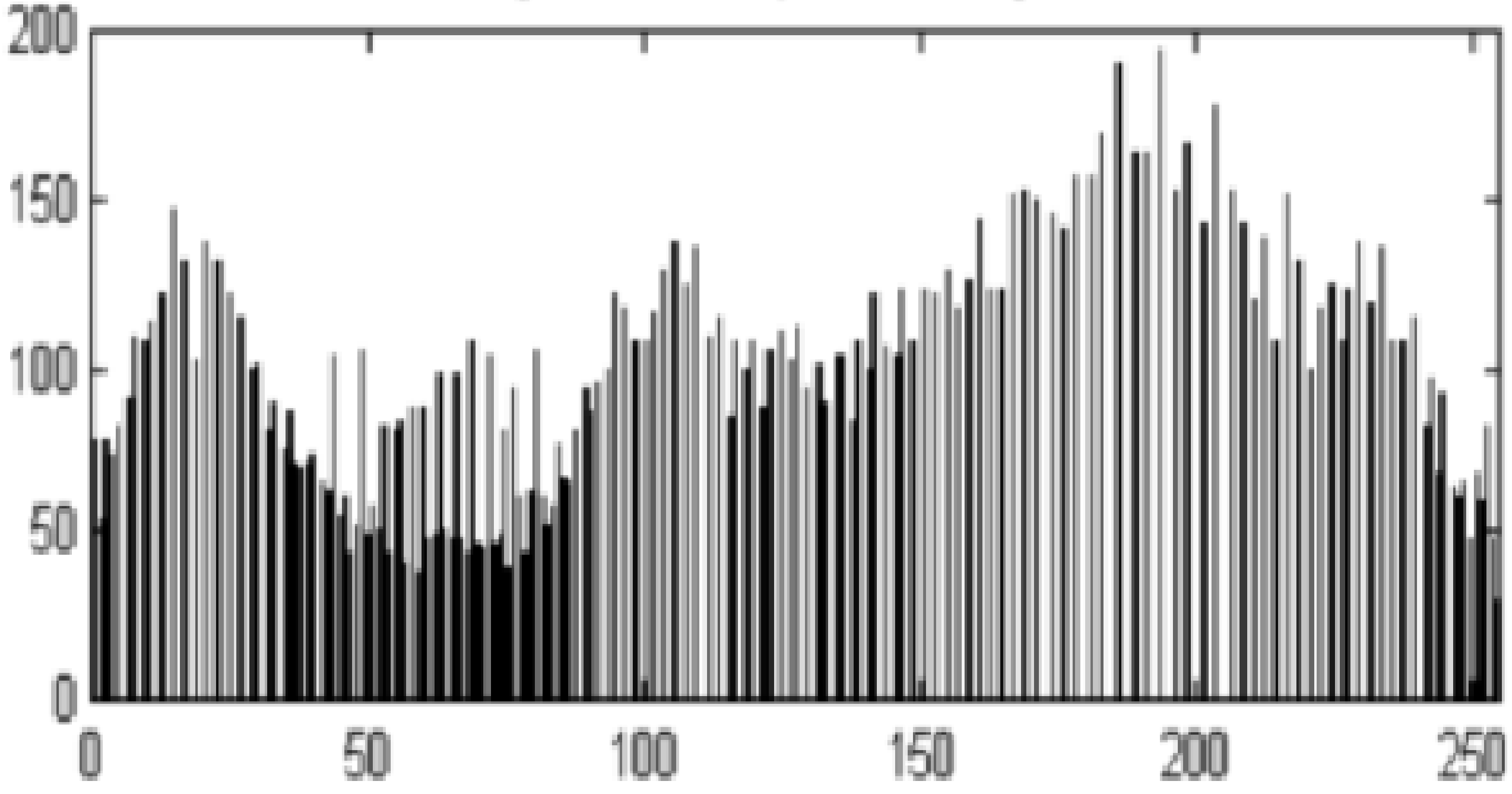
6	6	6	6	6
2	6	7	6	2
2	7	7	7	2
2	6	7	6	2
6	6	6	6	6

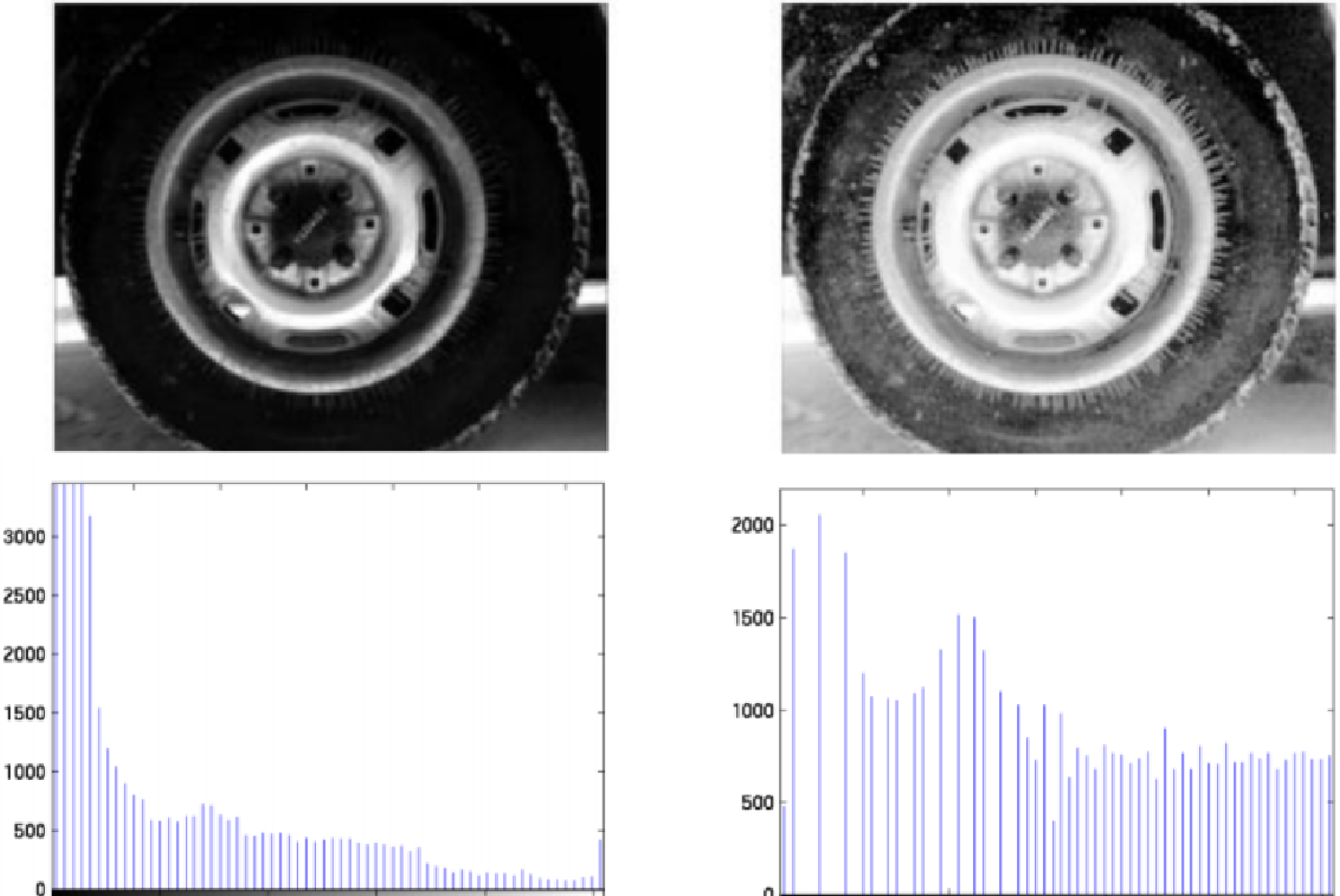
r_k	s_k
0	0
1	0
2	0
3	2
4	6
5	7
6	7
7	7

Histogram of the subimage

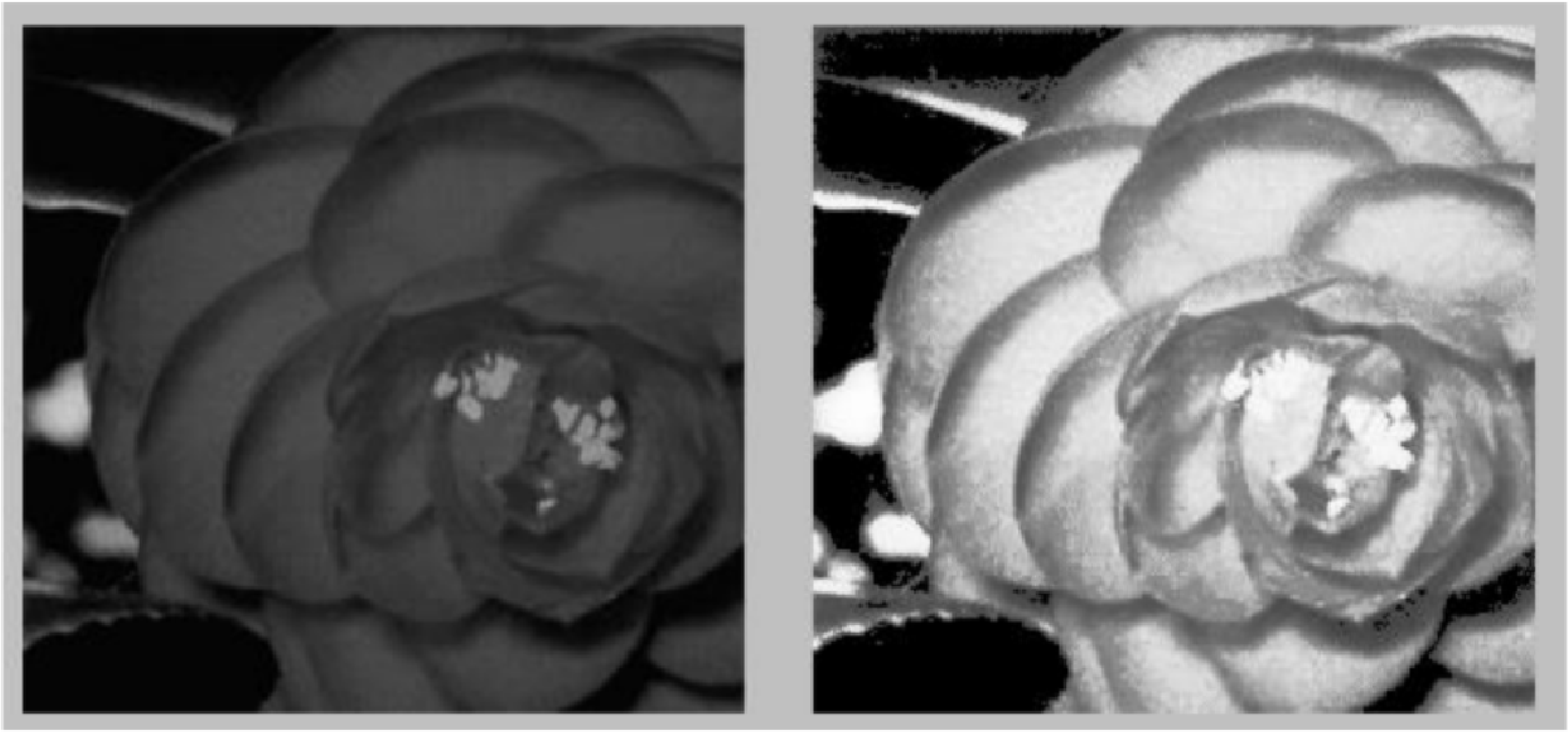


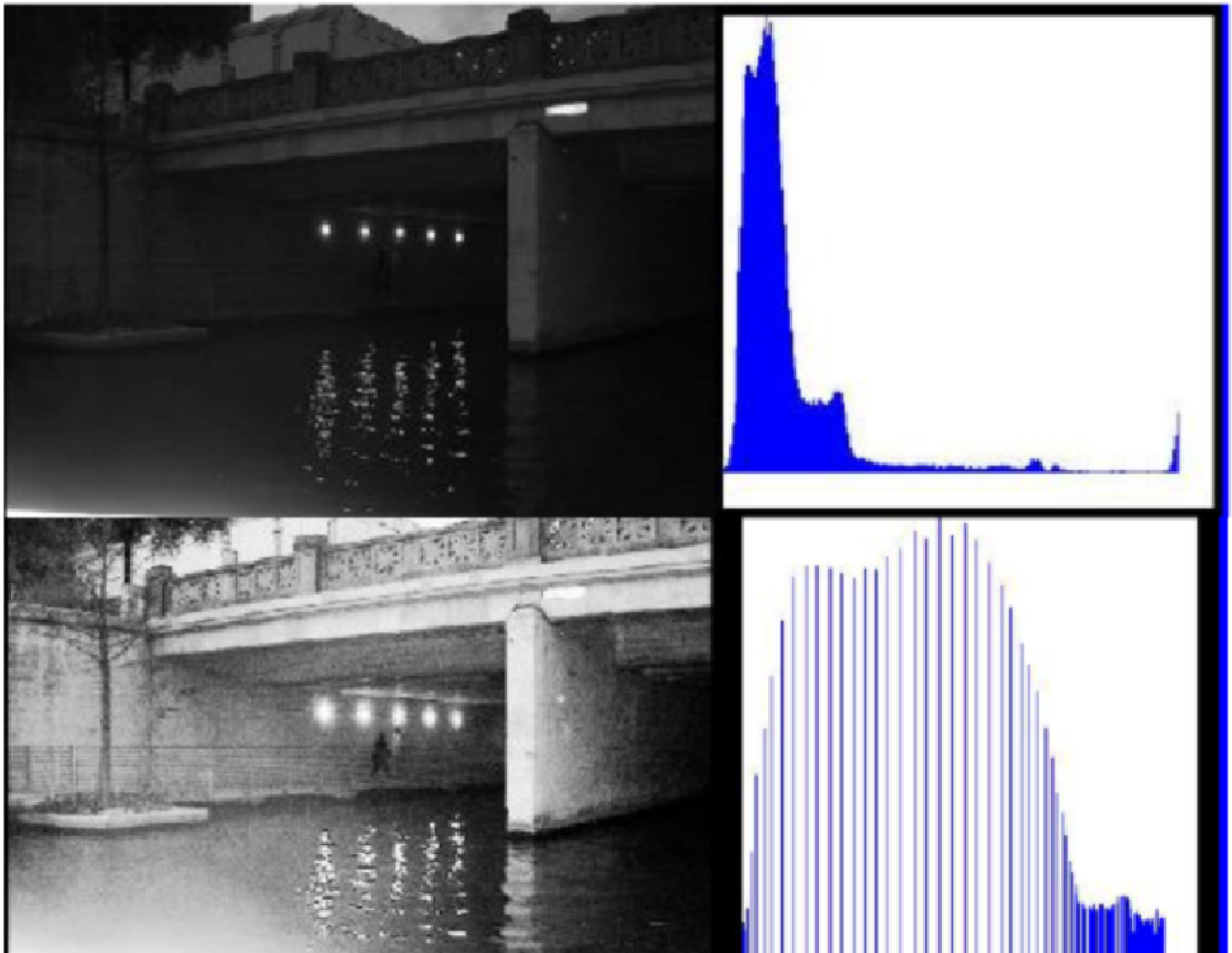
Histogram of the equalized image

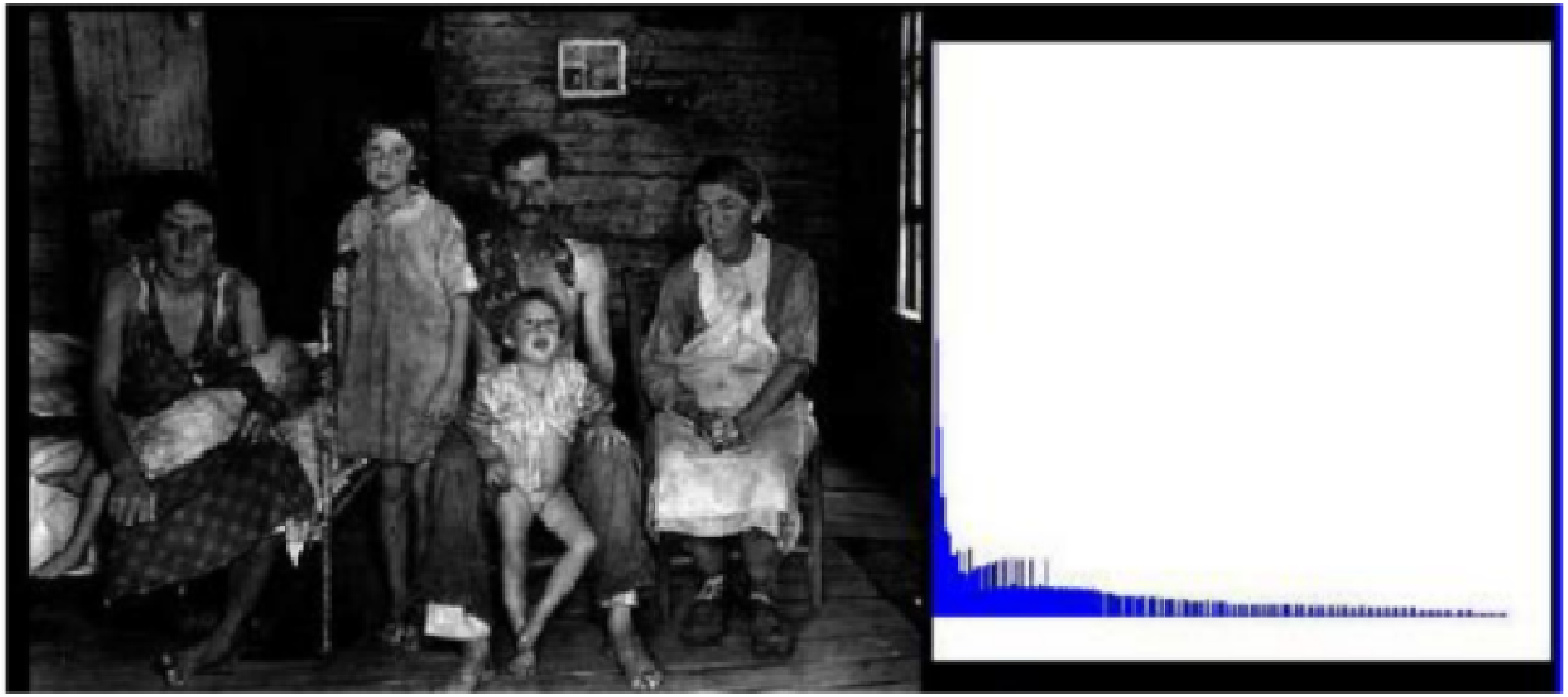




3000
2500
2000
1500
1000
500
0
2000
1500
1000
500
0

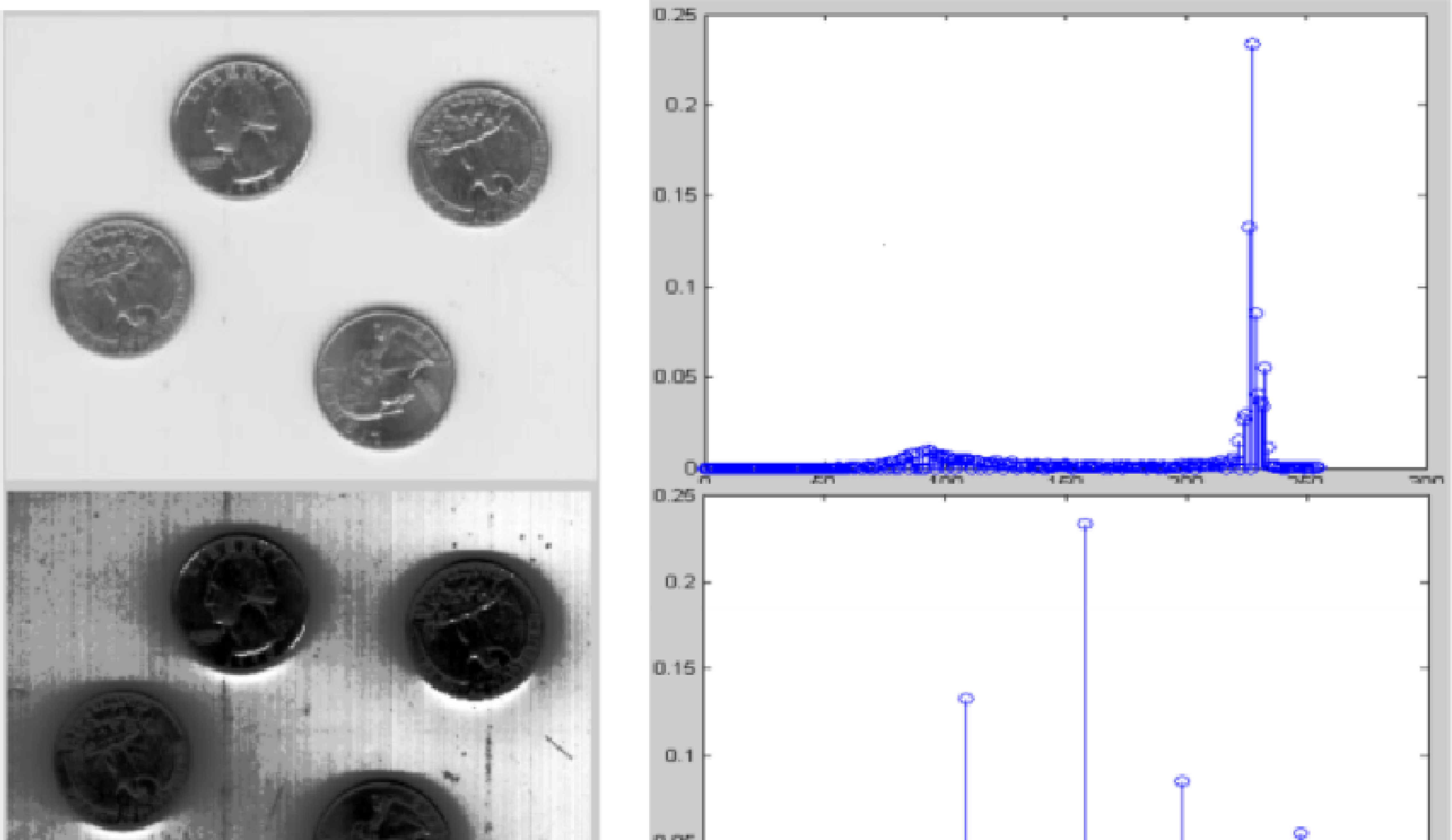






Histogram Equalization is not always desirable

Histogram equalization may not always produce desirable results, particularly if the given histogram is very narrow. It can produce false edges and false regions. It can also increase image "graininess" and "patchiness."



Example 2: apply histogram equalization to the 64X64 image given below

$n = 64 \times 64$

r_k	n_k
$r_0 = 0$	790
$r_1 = 1$	1023
$r_2 = 2$	850
$r_3 = 3$	656
$r_4 = 4$	329
$r_5 = 5$	245
$r_6 = 6$	122
$r_7 = 7$	81