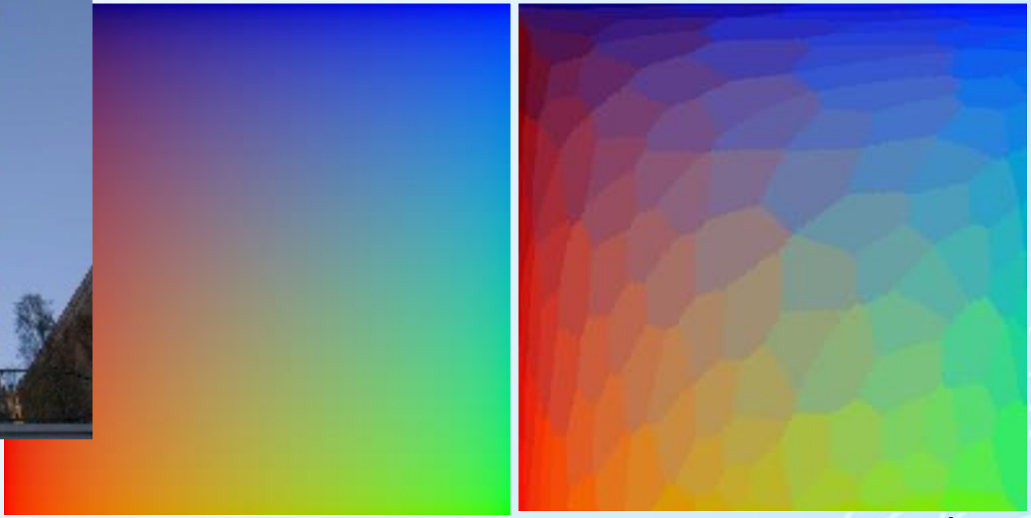
EE4704

Image Processing and Analysis

2 – DIGITAL IMAGING FUNDAMENTALS



ELEMENTS OF A VISION SYSTEM

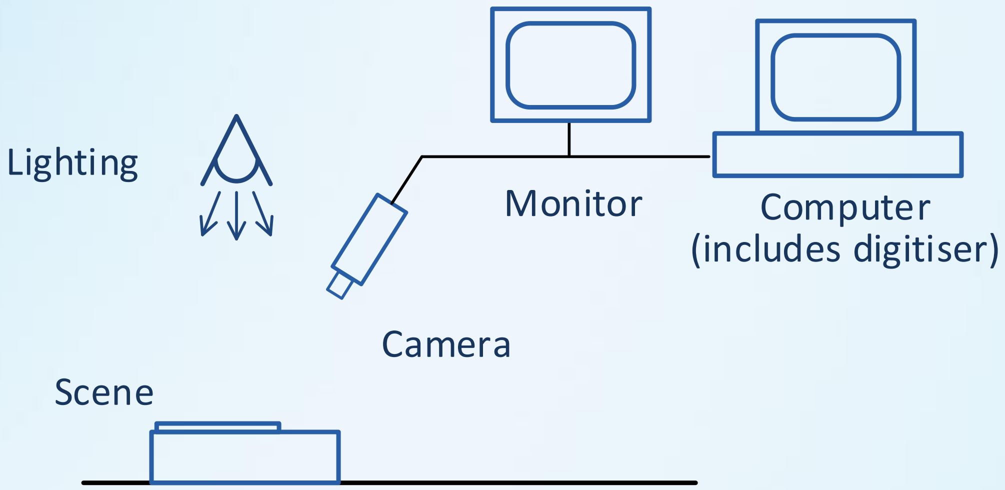
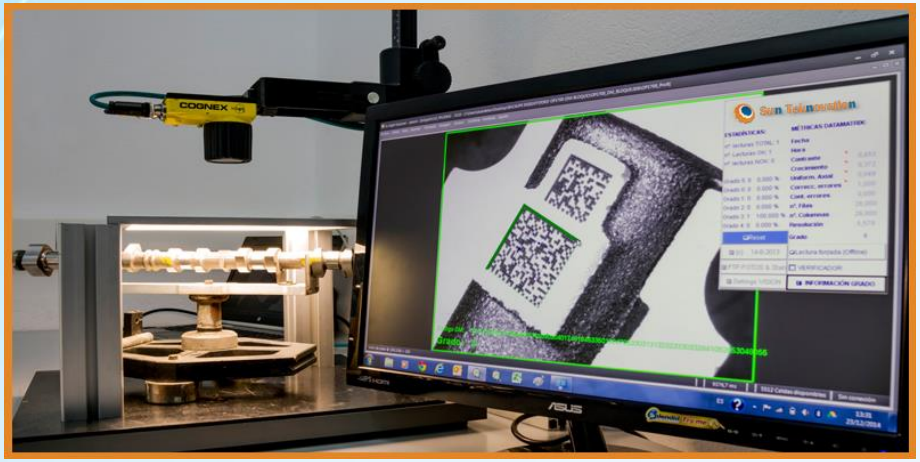


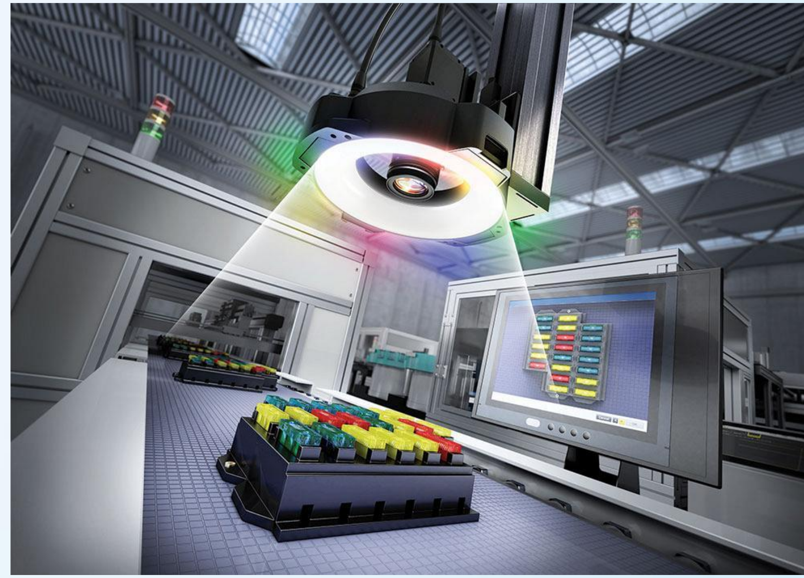
Image Acquisition
- sensor (e.g., video camera)
- digitiser (e.g., frame grabber)
- lighting

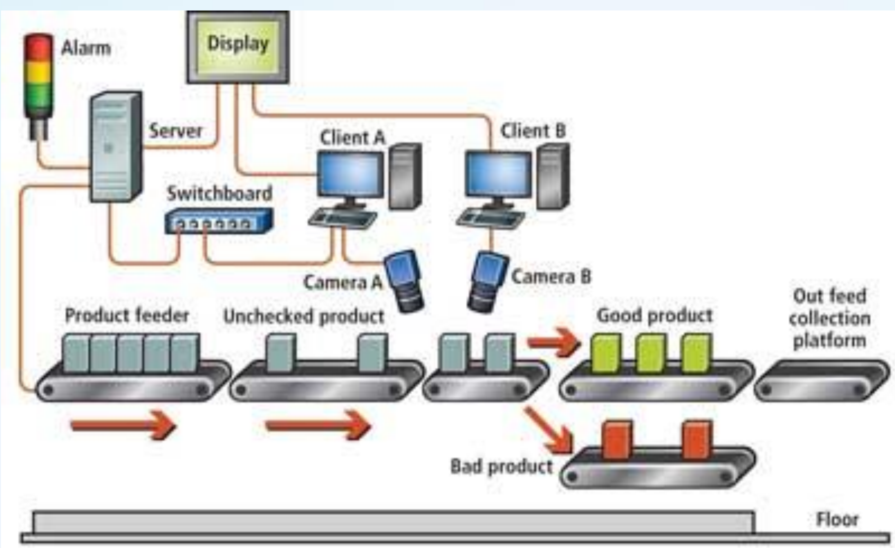
Processing
- computer
- dedicated hardware

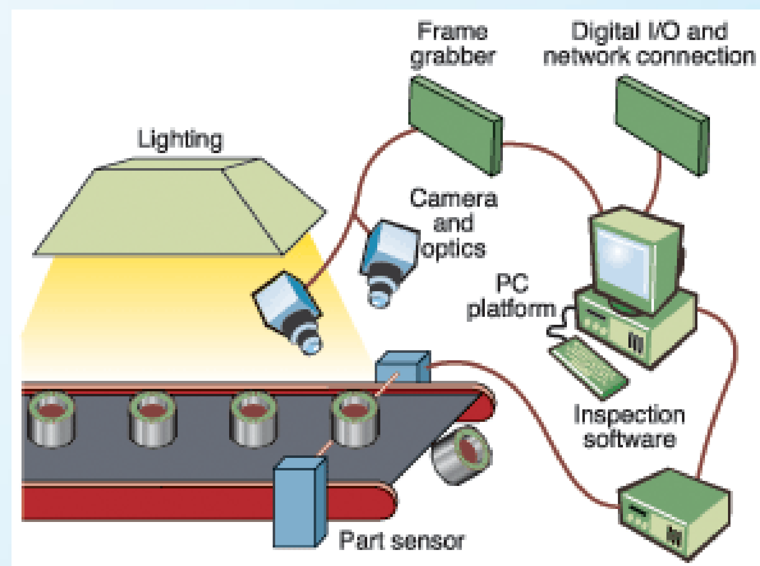
Storage
- hard disk
- optical disk
- digital video tape

Display
- video monitor


COGNEX




Alarm
Display
Server
Client A
Client B
Switchboard
Camera A
Camera B
Product feeder
Unchecked product
Good product
Out feed
collection
platform
Bad product
Floor


Frame
grabber
Digital I/O and
network connection
Lighting
Camera
and
optics
PC
platform
Inspection
software
Part sensor

ADVANCED VISION SYSTEMS

3D imaging
Mono, stereo, laser triangulation, pattern projection, structured light, time of flight

Non-visible imaging
Infrared, hyperspectral, multispectral

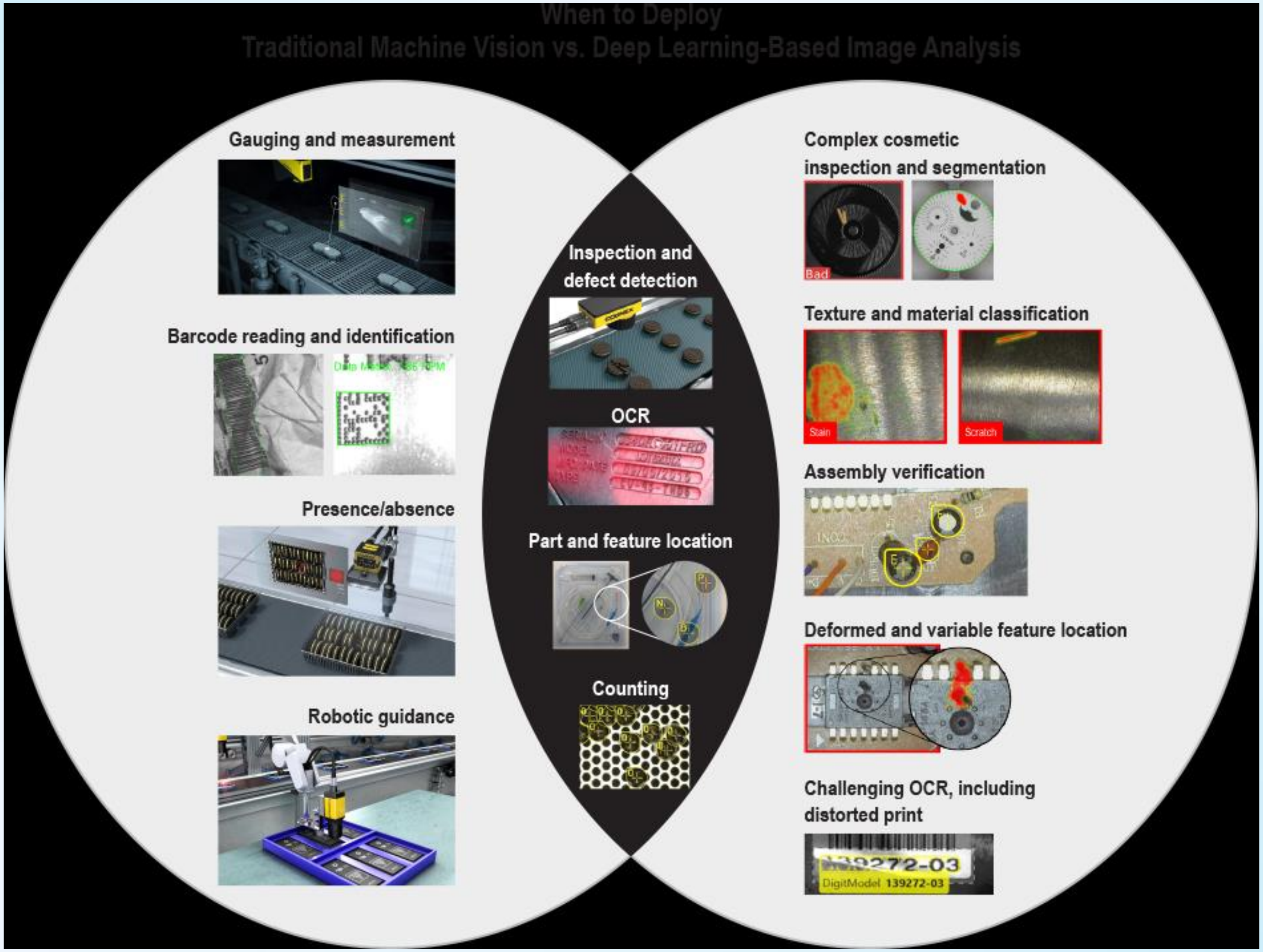
Embedded and mobile vision systems
Embedded computers, compact vision processors, autonomous vehicles, drones, robotics, mobile vision systems

Artificial Intelligence, machine learning
Deep learning, neural networks

After training the Euresys EasySegment library with less than 100 sample images of good coffee beans without debris, the tool identifies objects it recognizes as non-coffee bean items.

Traditional Machine Vision

Deep Learning-Based Machine Vision



A SIMPLE IMAGE FORMATION MODEL

An image *is* a 2D light-intensity function, $f(x,y)$, where the value of f at spatial coordinates (x,y) gives the intensity (or brightness) of the image at that point.

The image function may be approximated by:

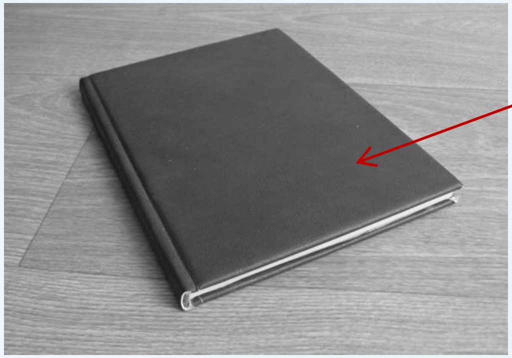
$$f(x, y) = i(x,y)\, r(x,y)$$

where

$i(x,y)$ = illumination component, $0 < i(x,y) < \infty$

$r(x,y)$ = reflectance component, $0 < r(x,y) < 1$

- $i(x, y)$ is determined by the light source(s)
- $r(x, y)$ is determined by the characteristics of the objects in a scene



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

Note: does not take into account orientation of surface with respect to the light source

Some Examples

Illumination:

- 100,000 lux on a clear day
- < 10,000 lux on a cloudy day
- 0.1 lux on a clear evening under full moon (on surface of earth)
- 1,000 lux in the office

The lux is a measure of illuminance (the luminous flux per unit area.)

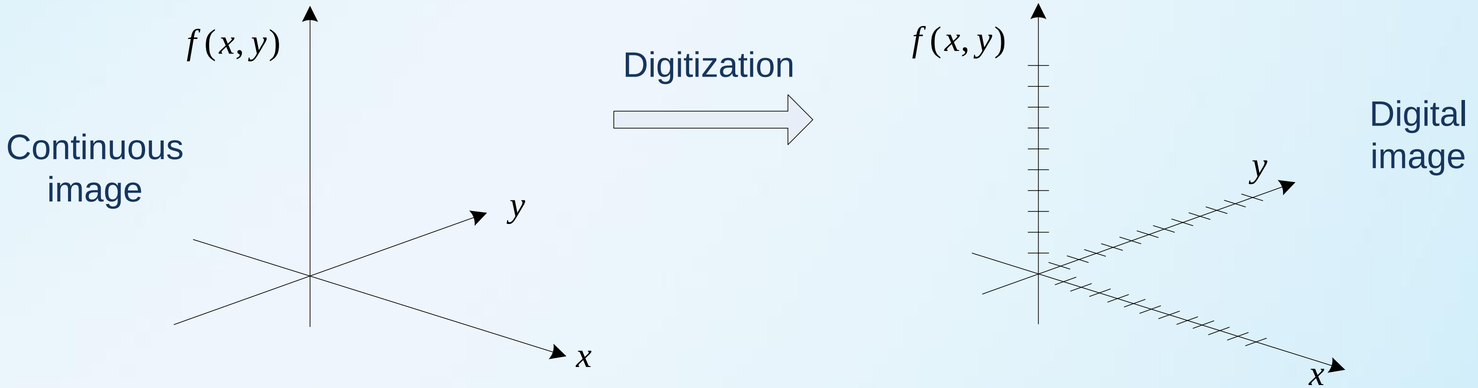
Reflectance:

- 0.01 for black velvet
- 0.65 for stainless steel
- 0.80 for flat-white wall paint
- 0.90 for silver-plated metal
- 0.93 for snow

SAMPLING AND QUANTIZATION

In order for a computer to process an image, the image must be digitized spatially and in amplitude.

- *Image sampling:* digitization of the spatial coordinates (x,y).
- *Gray-level quantization:* digitization of amplitude.



Note: $f(x,y)$ is used to denote both the continuous image function and the digital image

N columns

$$f(x,y)=\begin{pmatrix} f(0,0) & \cdots & f(0,N-1) \\ \vdots & \ddots & \vdots \\ f(M-1,0) & \cdots & f(m-1,n-1) \end{pmatrix}$$

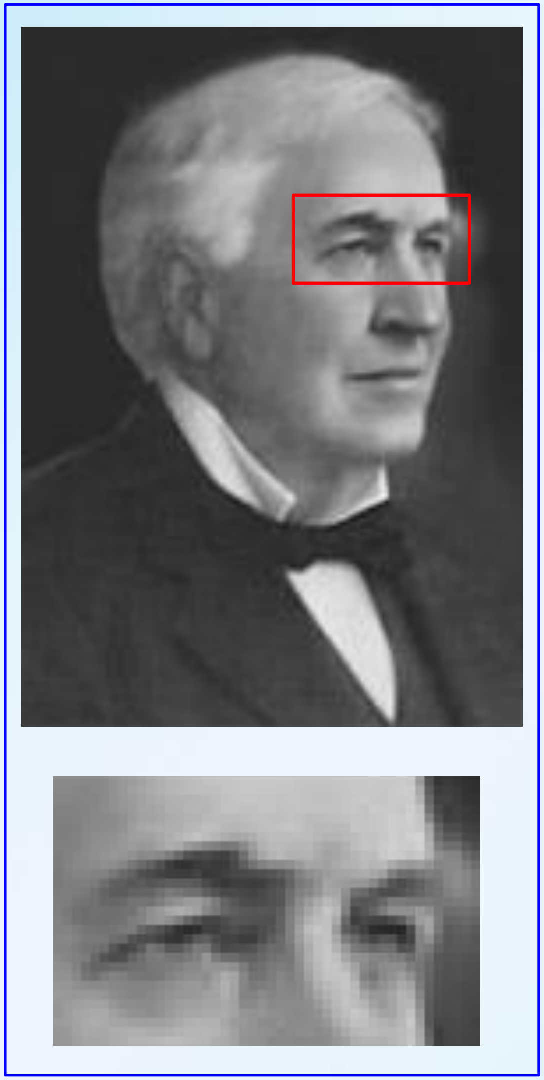
M rows

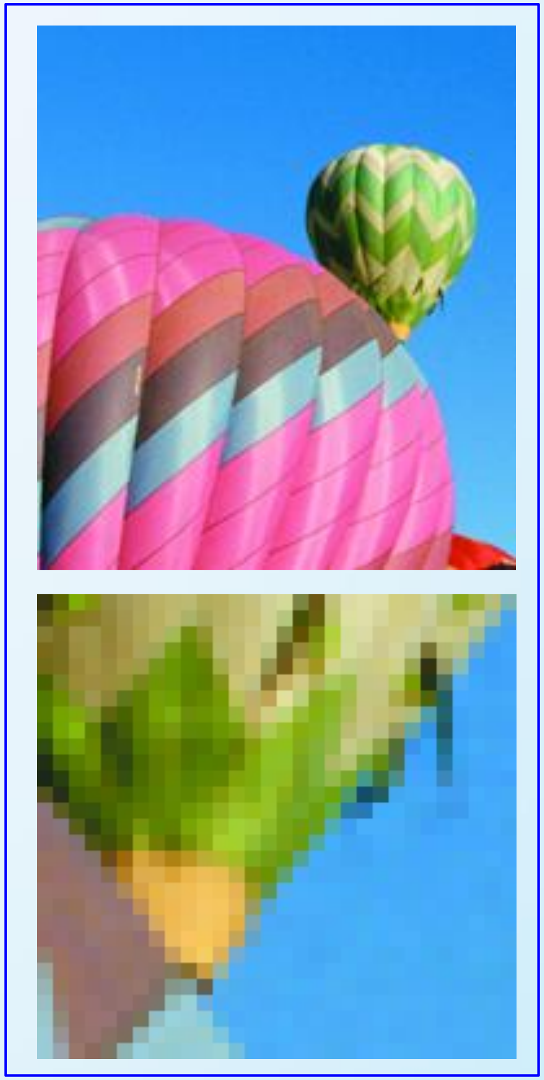
Digital image

- The digital image function is an $M \times N$ array.
- Each element of the array is referred to as an image element, picture element or pixel.
- The equation implies that the x axis points downwards and the y axis points to the right. Note that this is not the only convention.

- The *brightness* of a monochrome image f at coordinates (x,y) is called the gray level (r_k)
- r_k is an integer in the range $[0,L\text{-}1]$, e.g., [0,255]
- 0 is black and L-1 is white.

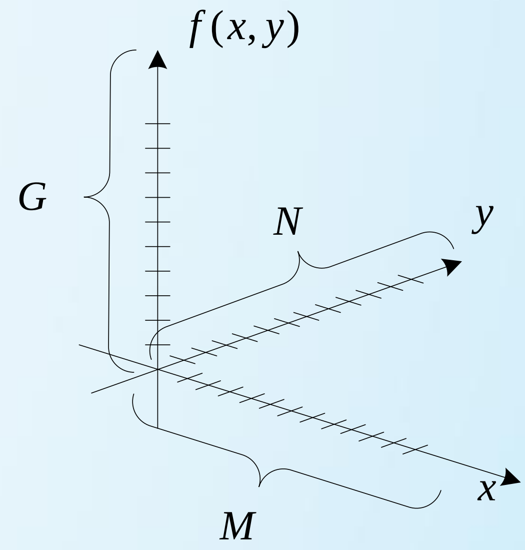
Digital images:





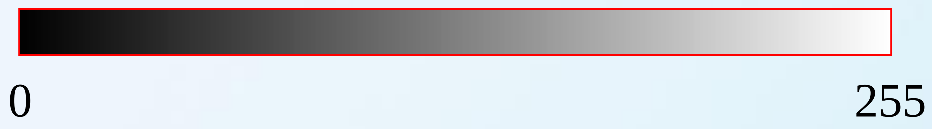
The resolution (the degree of discernible detail) of an image depends on

- the number of samples $M \times N$ (*spatial resolution*)
- the number of gray levels G *(gray-level resolution)*



The larger these parameters, the closer the digitized array approximates the original image, but storage and processing requirements increase with *M*, *N*, and *G*.

If a pixel is stored in a byte (G = 256), as is usually the case for monochrome images, the number of bytes required for storage is $M \times N$.



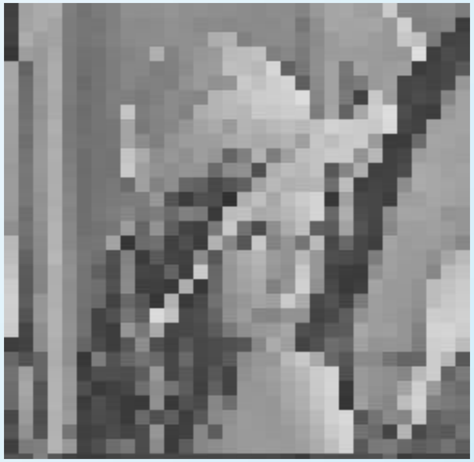
Insufficient spatial resolution → checkerboard effect (pixellation)

256x256

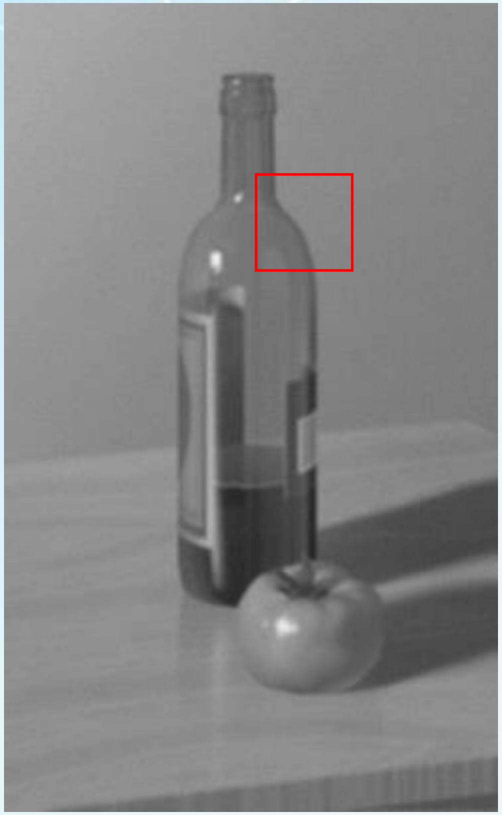
128x128

64x64

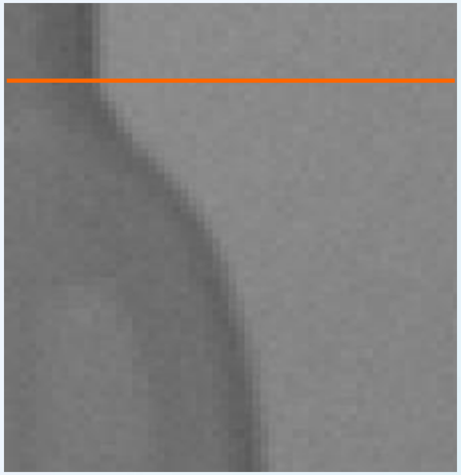
32x32



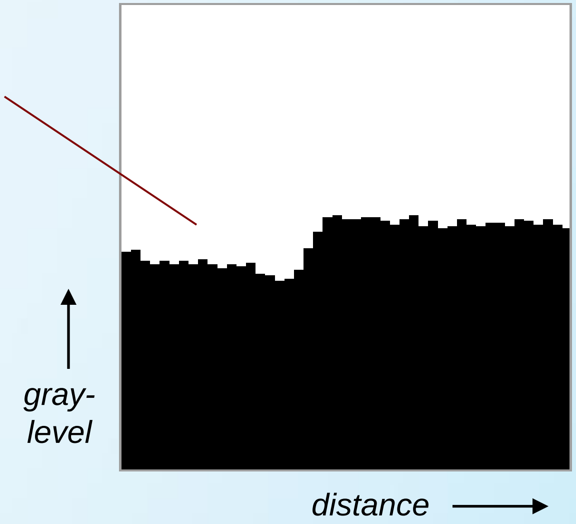
Sampling effects



Image



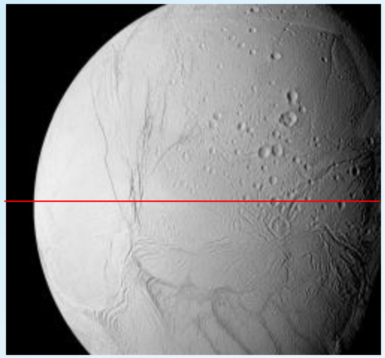
Magnified
portion



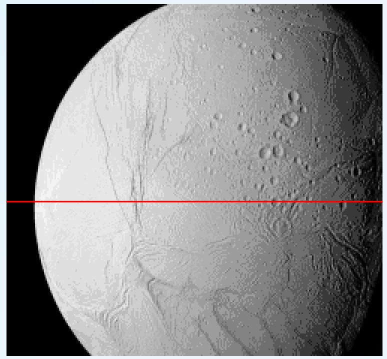
Gray-level profile
or
scan line

Insufficient gray-level resolution → false contouring

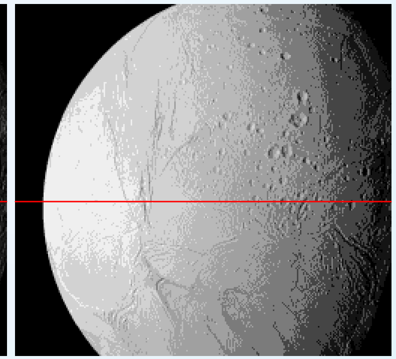
Good gray-level resolution requires $G \geq 256$



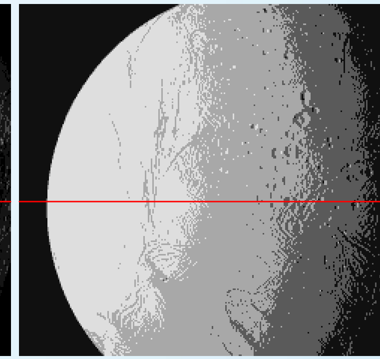
8 bits



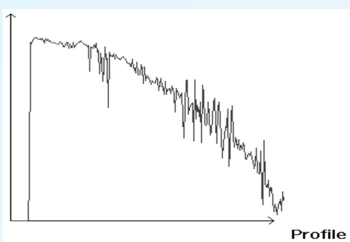
4 bits

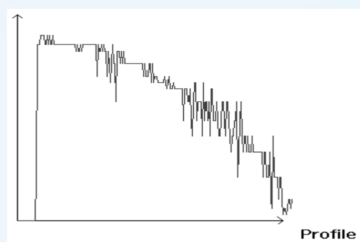


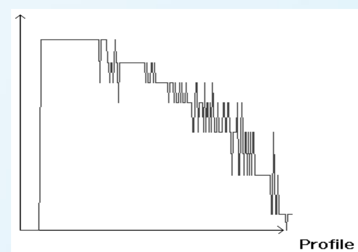
3 bits

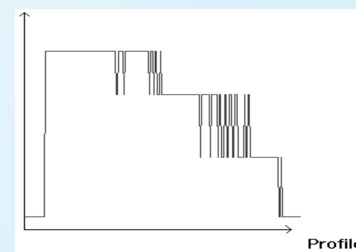


2 bits

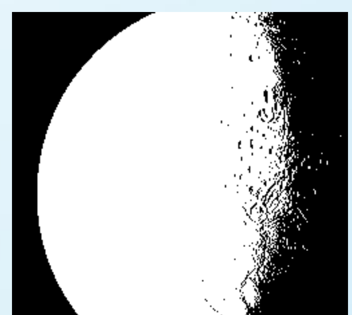








Binary images: two brightness levels (logical 0 and 1)



36435 colours

256 colours

BASIC RELATIONSHIPS BETWEEN PIXELS

Neighbours of a Pixel

	a	b	c	
	d	p	e	
	f	g	h	

4-neighbours of p: $N_4(p) = \{b, d, e, g\}$

Diagonal neighbours of p: $N_D(p) = \{a, c, f, h\}$

8-neighbours of p: $N_8(p) = N_4(p) \cup N_D(p) = \{a, b, c, d, e, f, g, h\}$

Connectivity

The concept of pixel connectivity is used in establishing boundaries of objects and connected components in an image.

To establish whether two pixels are connected, we must determine if they satisfy two criteria:

- *spatial* - they are *adjacent* in some sense (e.g., if they are 4-neighbours or 8-neighbours)
- *gray level* - their gray levels satisfy a specified *criterion of similarity* (e.g., if they lie within a range).

0	0	203	197	175
0	200	80	78	81
0	202	82	202	82
0	204	78	80	79
0	0	198	76	202

Pixel p (shaded)

0	0	203	197	175
0	200	80	78	81
0	202	82	202	82
0	204	78	80	79
0	0	198	76	202

8-neighbours of p

0	0	203	197	175
0	200	80	78	81
0	202	82	202	82
0	204	78	80	79
0	0	198	76	202

Connected pixels

Let V be the set of gray-level values used to define the *similarity criterion*. For example,

$$V = \{1\} \quad \text{(binary image) or } V = \{51, 52, \ldots, 60\} \quad \text{(gray-level image)}$$

Two types of connectivity are commonly used:

- 4-*connectivity*. Two pixels p and q with values from V are 4-connected if q is in the set $N_4(p)$, i.e., p and q are 4-neighbours
- 8-*connectivity*. Two pixels p and q with values from V are 8-connected if q is in the set $N_8(p)$, i.e., i.e., p and q are 8-neighbours

$$\begin{array}{ccc} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{array}$$

4-connectivity

$$\begin{array}{ccc} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{array}$$

8-connectivity

Examples

0	2	1	1	2	0	7
1	0	8	8	9	1	7
0	0	9	0	1	0	1
1	8	7	1	1	2	1
9	0	1	1	1	0	1

0	2	1	1	2	0	7
1	0	8	8	9	1	7
0	0	9	0	1	0	1
1	8	7	1	1	2	1
9	0	1	1	2	0	1

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

0	2	1	1	2	0	7
1	0	8	8	9	1	7
0	0	9	0	1	0	1
1	8	7	1	1	2	1
9	0	1	1	2	0	1

4-connectivity

0	2	1	1	2	0	7
1	0	8	8	9	1	7
0	0	9	0	1	0	1
1	8	7	1	1	2	1
9	0	1	1	2	0	1

8-connectivity

0	0	203	197	175
0	200	80	78	81
0	202	82	202	82
0	204	78	80	79
0	0	198	76	202

$\Rightarrow$

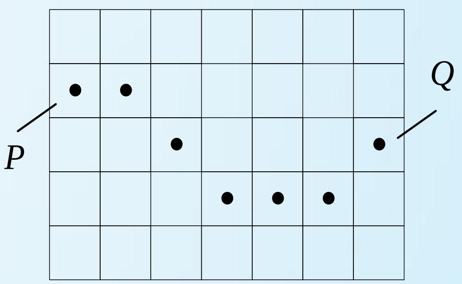
0	0	203	197	175
0	200	80	78	81
0	202	82	202	82
0	204	78	80	79
0	0	198	76	202

$$V = \{[195, 205]\}$$

8-connectivity

A *path* from pixel P with coordinates (x_1, y_1) to pixel Q with coordinates (x_n, y_n) is a sequence of distinct pixels with coordinates

$$(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots\ldots, (x_n, y_n)$$



where

(x_i, y_i) is connected to (x_{i-1}, y_{i-1}), $1 \leq i \leq n$

We can define 4- or 8-paths, depending on the type of connectivity specified.

P

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

Q

4-path

P

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

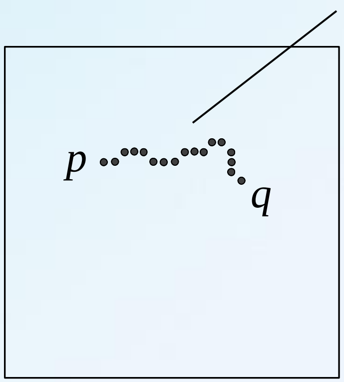
Q

8-path

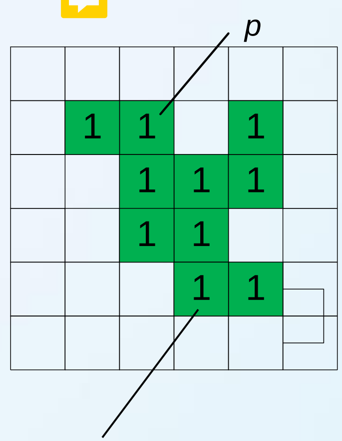
Connected components:

- Pixels *p* and *q* are connected if there is a path from *p* to *q*.
- The set of pixels that are connected to *p* is called a *connected component*. Hence, any two pixels of a connected component are connected to each other, and distinct connected components are disjoint.

path

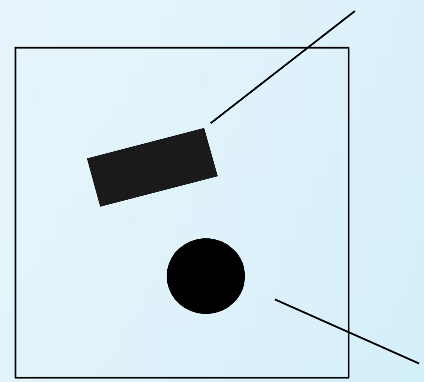


p is connected to *q*



connected component
(all 'green' pixels)

connected
component A



connected
component B

Distance Measures

Euclidean distance:

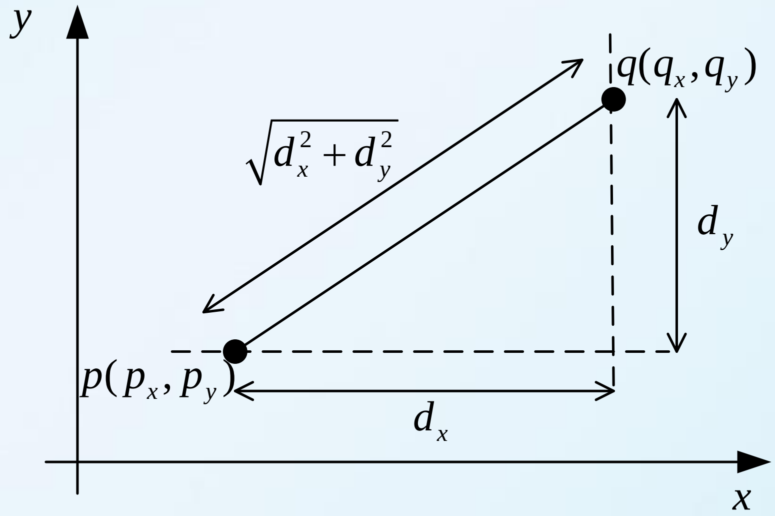
$$D_e(p,q) \triangleq (d_x^2 + d_y^2)^{1/2} = [(q_x - p_x)^2 + (q_y - p_y)^2]^{1/2}$$

D_4 distance (city-block distance):

$$D_4(p,q) \triangleq |d_x| + |d_y| = |q_x - p_x| + |q_y - p_y|$$

D_8 distance (chessboard distance):

$$D_8(p,q) \triangleq \max\left(|d_x|,|d_y|\right) = \max\left(|q_x - p_x|,|q_y - p_y|\right)$$



Arithmetic/Logic Operations

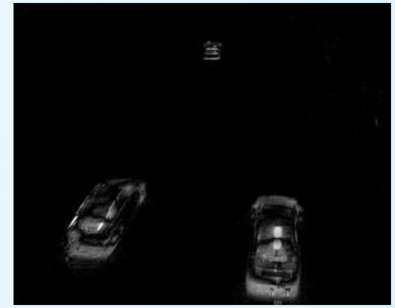
Arithmetic operations between two pixels *p* (value z_p) and *q* (value z_q) are

$$\text{Addition: } z_p + z_q$$
$$\text{Subtraction: } z_p - z_q$$
$$\text{Multiplication: } z_p \times z_q$$
$$\text{Division: } z_p \div z_q$$

Often, one of the pixels is a constant operand, as in the multiplication of an image by a constant.

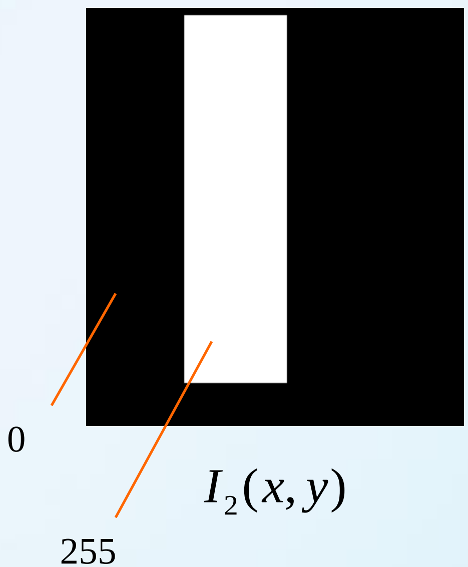
$I_1(x,y)$

$I_2(x,y)$



$I_3(x,y) = \left|I_1(x,y) - I_2(x,y)\right|$

$I_1(x,y)$



$I_2(x,y)$

$I_3(x,y) = I_1(x,y) \times I_2(x,y)$

rescaled to [0,255]

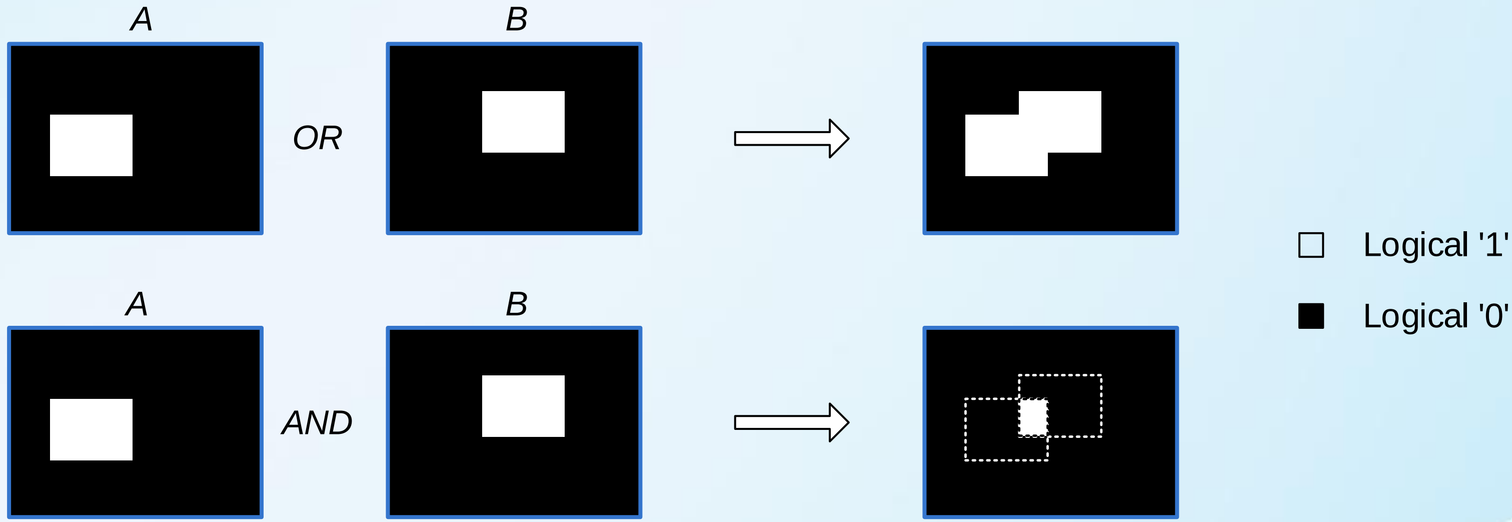
[Logic operations](#) are applicable only to binary images. With logical values z_p and z_q:

$$\text{AND} : z_p \bullet z_q$$

$$\text{OR} : z_p + z_q$$

$$\text{COMPLEMENT} : \overline{z}_p$$

Logic operations are basic tools in morphological image processing where they are used for tasks such as feature detection and shape analysis.

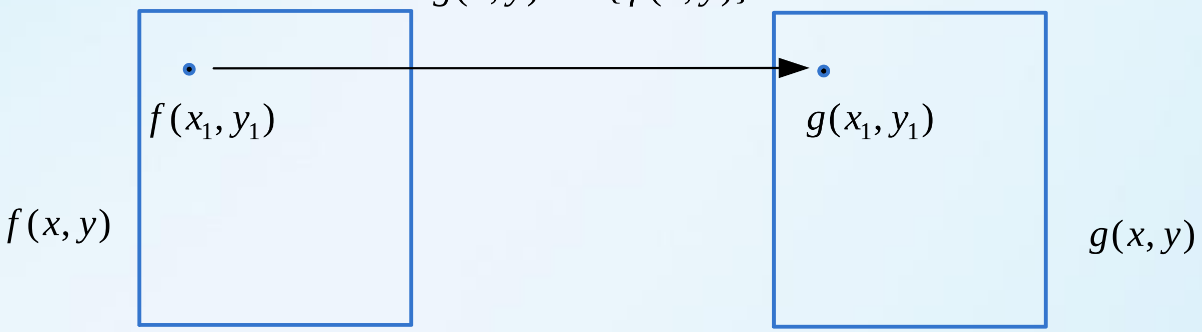


Arithmetic and logical operations are used in basically two ways: point operations and neighbourhood operations.

Point operations:

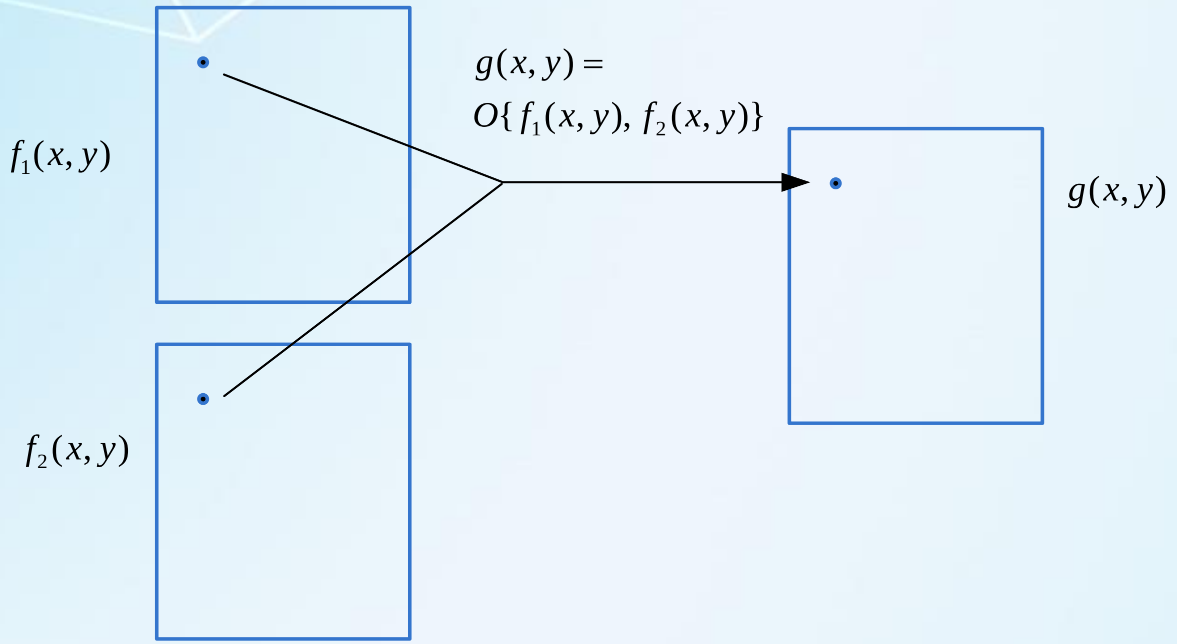
Point operations modify the gray values at individual pixels depending only on the gray value. The input may be one or more images. Examples are gray-level transformation or addition of two images.

$$g(x, y) = O\{f(x, y)\}$$



Example:

$$g(x, y) = 0.5 f(x, y) + 20$$



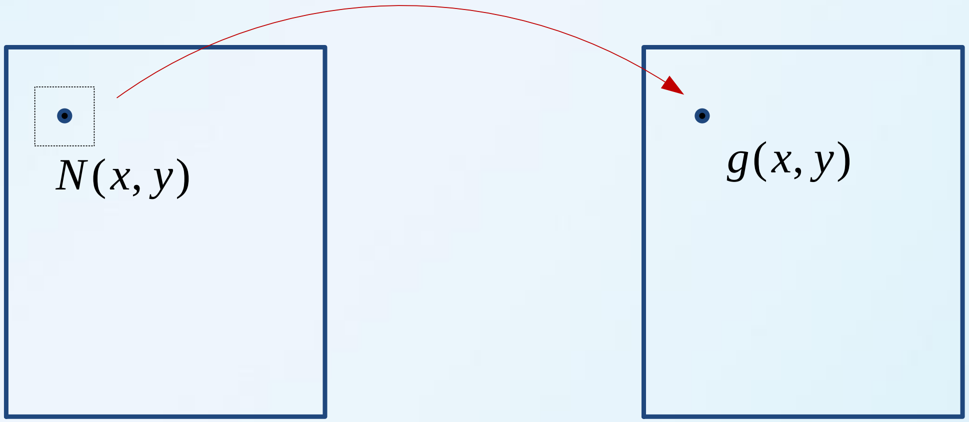
Example:

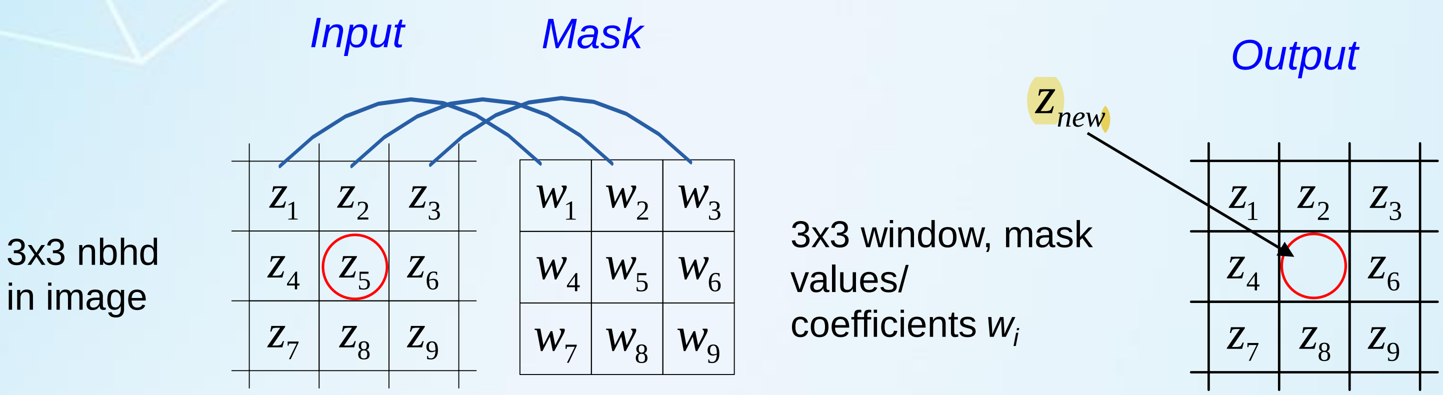
$$g(x, y) = f_1(x, y) + f_2(x, y) - 20$$

Neighbourhood operations:

Neighbourhood operations combine the values of the pixels in a neighbourhood to yield the output value. This requires the definition of appropriate masks (template, window, or filter). Mask sizes vary, e.g., 3×3, 3×1, 5×5.

$$g(x, y) = O\{N(x, y)\}$$





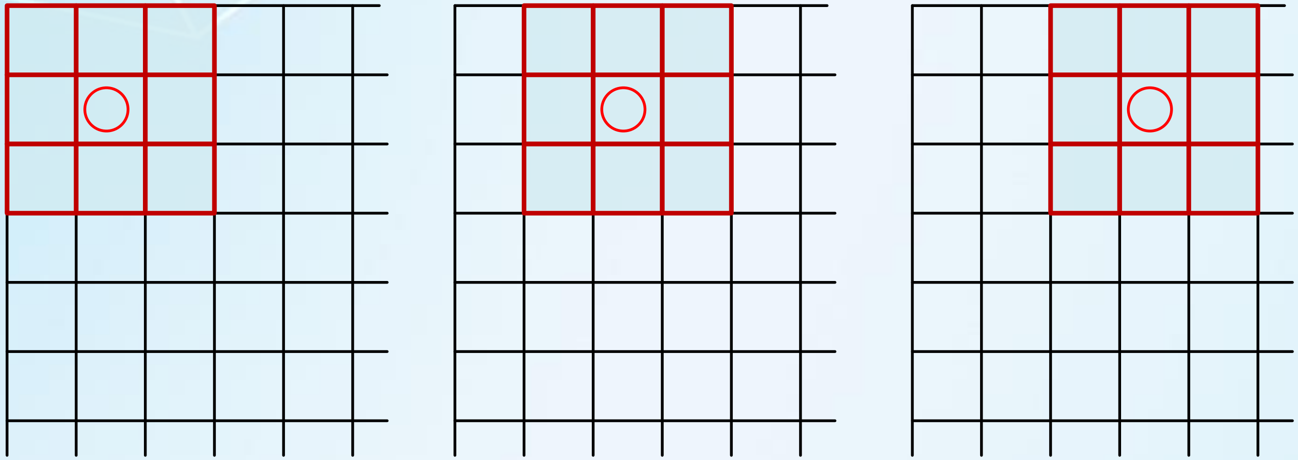
The value of the center pixel is replaced by a new value z_{new}:

$$z_{new} = w_1 z_1 + w_2 z_2 + w_3 z_3 + w_4 z_4 + w_5 z_5 + w_6 z_6 + w_7 z_7 + w_8 z_8 + w_9 z_9$$

If we set $w_i = 1/9$,

$$z_{new} = \frac{1}{9}(z_1 + z_2 + z_3 + z_4 + z_5 + z_6 + z_7 + z_8 + z_9)$$

which is the average value of the pixels in the 3×3 neigbourhood.



The mask is moved across the image, column by column, row by row.

Many useful image processing functions can be obtained by neighbourhood operations, e.g., noise reduction, edge detection and image enhancement.

Example

Input image
8 x 6

0	0	0	0	0	0
0	0	0	20	50	10
0	0	20	25	50	10
0	20	25	40	50	10
0	25	40	40	50	10
0	35	100	100	100	10
0	50	50	50	50	10
0	0	0	0	0	0

Mask
3 x 3

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

Output image
6 x 4

	2	7	18	18	
	7	17	31	29	
	14	26	38	32	
	27	47	61	46	
	33	54	64	47	
	26	43	50	36	

Pixel values have been rounded

Note: For a 3x3 mask, the outermost rows and columns are lost.

GRAY-LEVEL HISTOGRAM

The histogram plot shows the number of pixels at each gray level, i.e., number of pixels *vs* gray level.

For a digital image with gray levels in the range [0, $L - 1$], it is the discrete function

$$h(r_k) = n_k, \qquad k = 0, 1, 2, \ldots, L-1$$

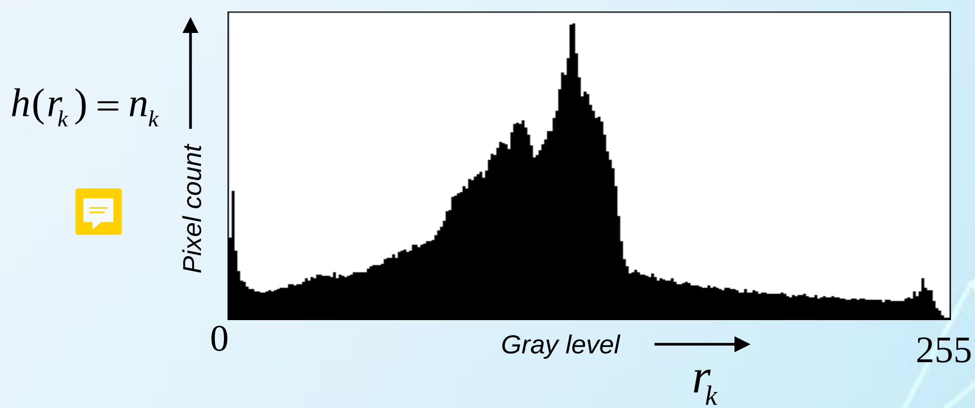
where n_k is the number of pixels with gray level r_k.

The number of pixels may be normalized by the total number of pixels in the image, N, to give

$$p(r_k) = n_k / N$$

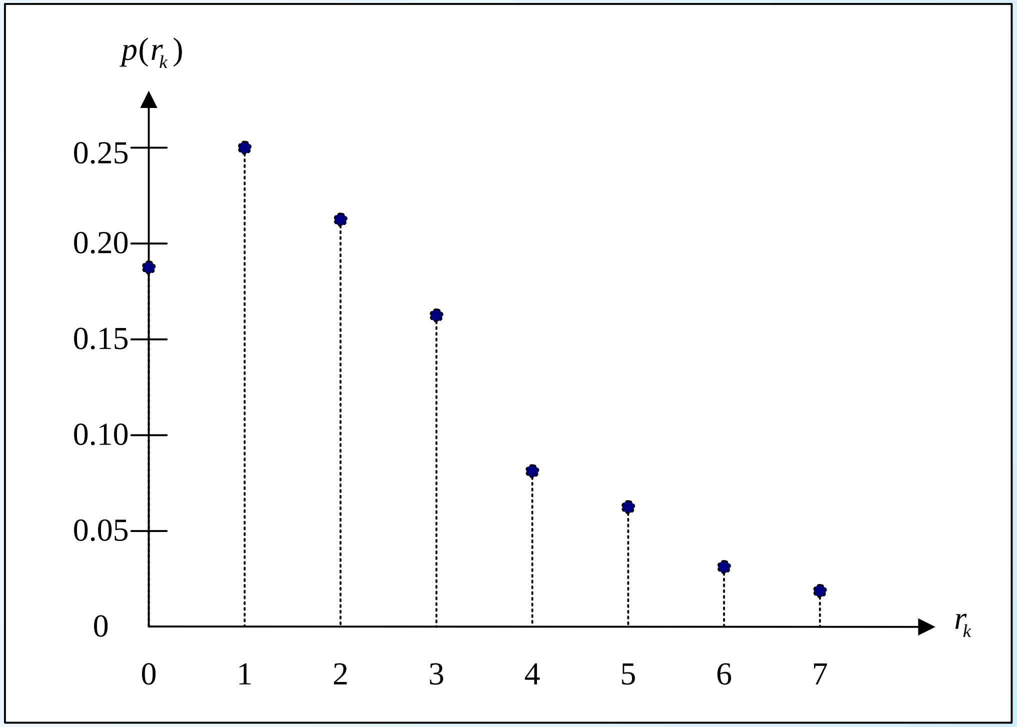
Note that:

$$\sum_{k=0}^{L-1} p(r_k) = 1$$

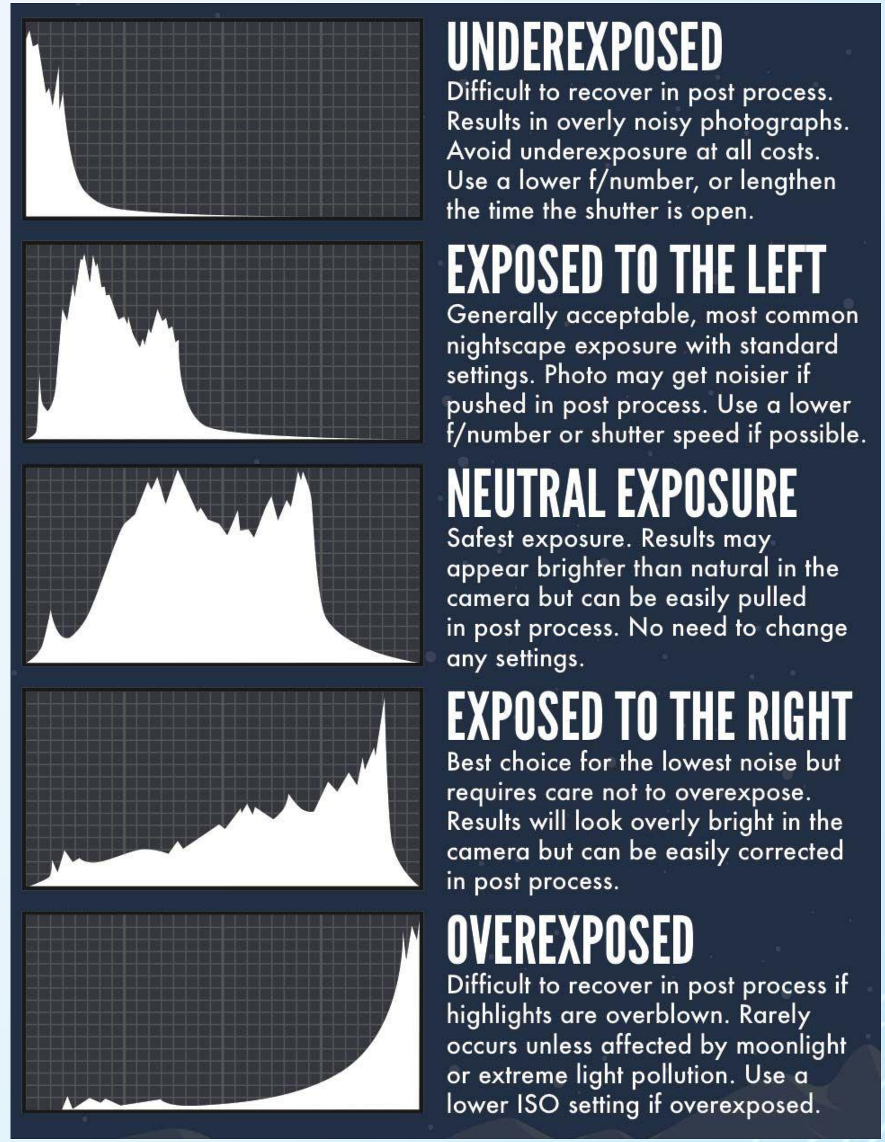


Example

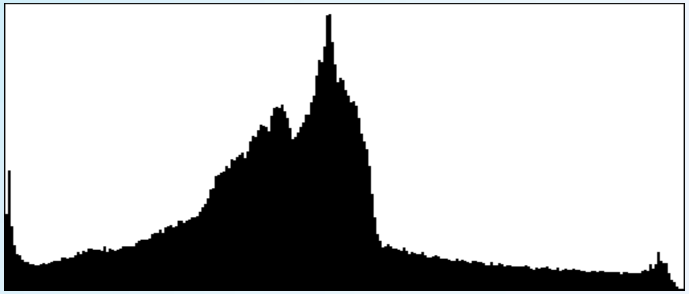
r_k	$h(r_k) = n_k$	$p(r_k) = n_k/N$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02



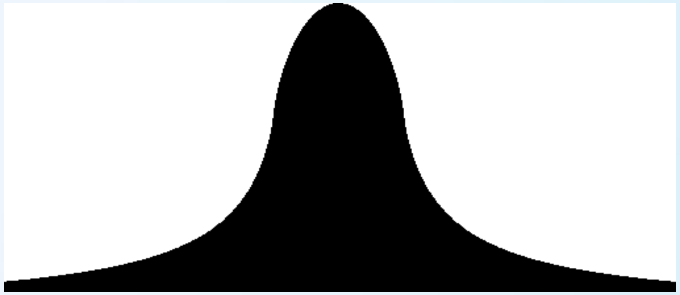
In digital photography



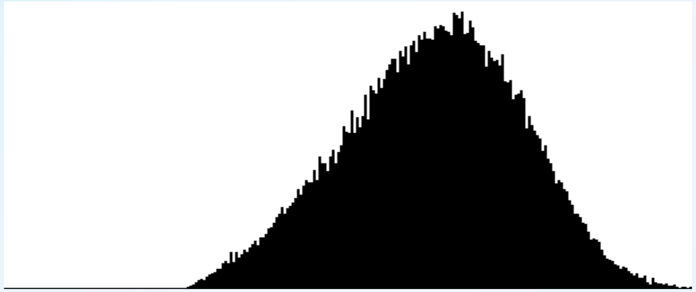
How do we describe a histogram quantitatively?



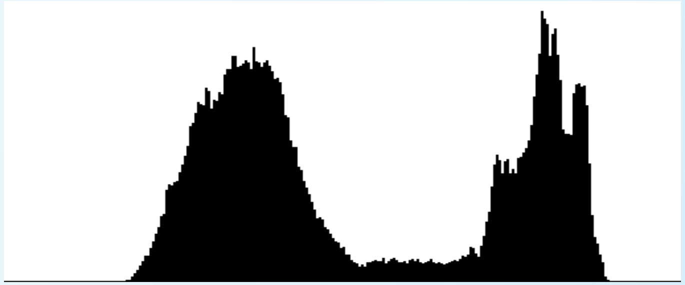
Histogram 1



Histogram 2



Histogram 3



Histogram 4

We can use statistical measures.

Statistical measures

The mean is defined as

$$m = \sum_{k=0}^{L-1} r_k p(r_k)$$

mean is the average intensity.

The nth moment of r_k about the mean is defined as

$$\mu_n = \sum_{k=0}^{L-1} (r_k - m)^n p(r_k)$$

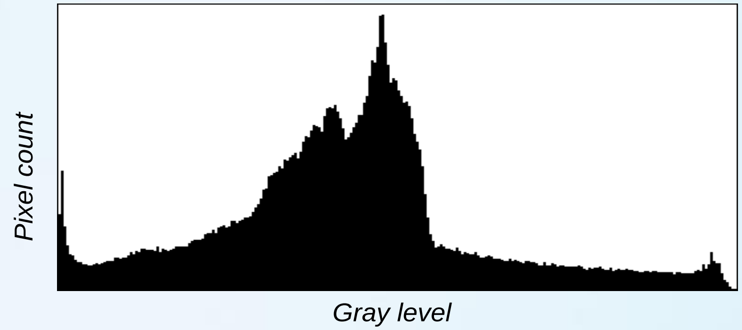
Zeroth moment μ_0 is always 1; first moment μ_1 is always 0.
Second moment μ_2 is also known as the variance, denoted by σ^2.

variance is a measure of image contrast

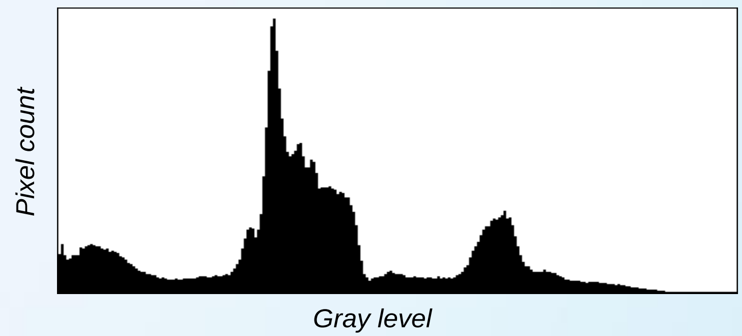
For the previous histogram

$$m = 2.09, \quad \mu_2 = 3.00, \quad \mu_3 = 4.26, \quad \mu_4 = 28.1$$

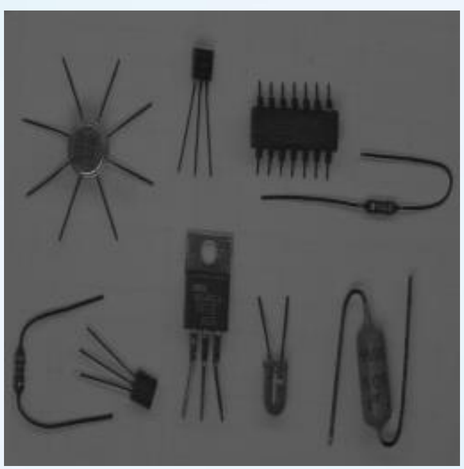
From its gray-level histogram, some characteristics of an image may be discerned, e.g., its contrast and overall intensity.

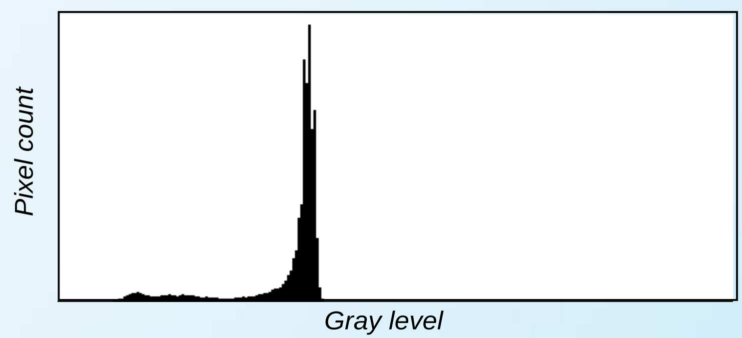


$$m = 109; \quad \sigma = 48$$

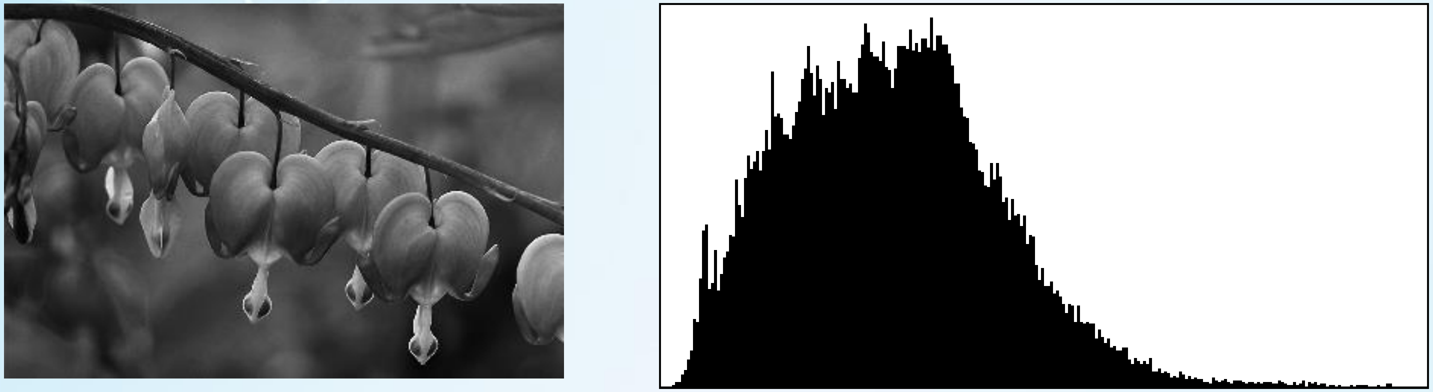


$$m = 98; \quad \sigma = 48$$





$$m = 86; \quad \sigma = 18$$

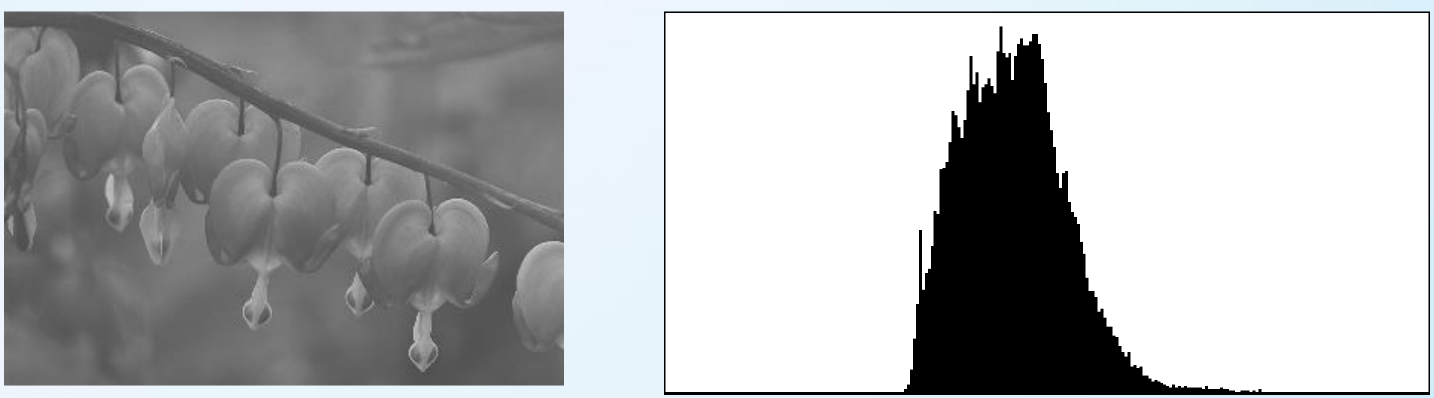


dark image

$m = 76; \quad \sigma = 36$

bright image

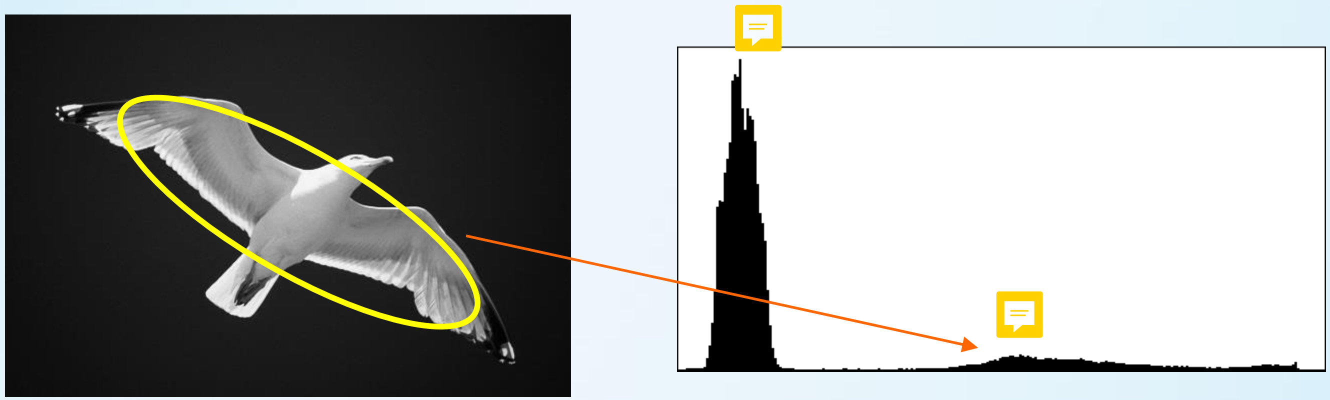
$m = 169; \quad \sigma = 29$

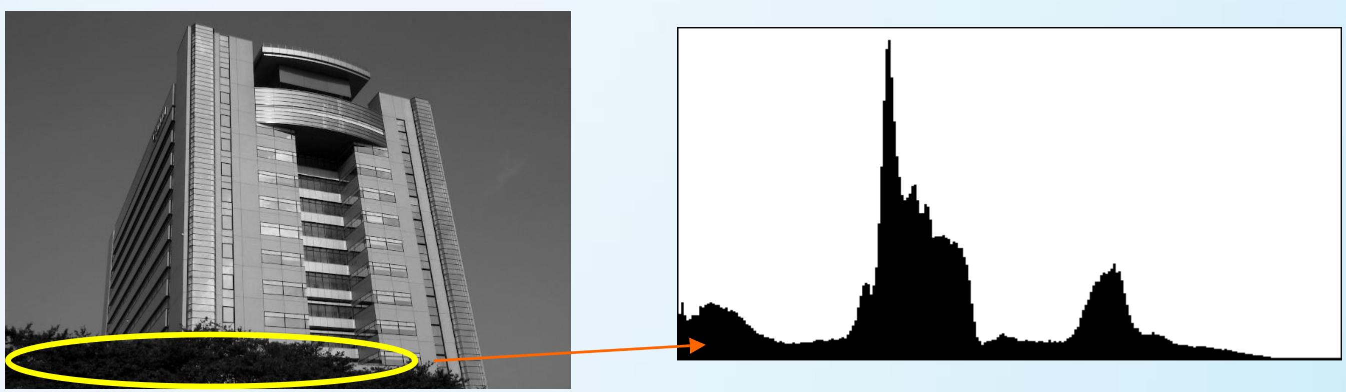


low-contrast image

$m = 116; \quad \sigma = 18$

It may be possible to relate a histogram mode to its corresponding image region.





The histogram provides a global view of the gray-level distribution; thus two images with the same or similar histogram may be markedly different in appearance.

Image 1

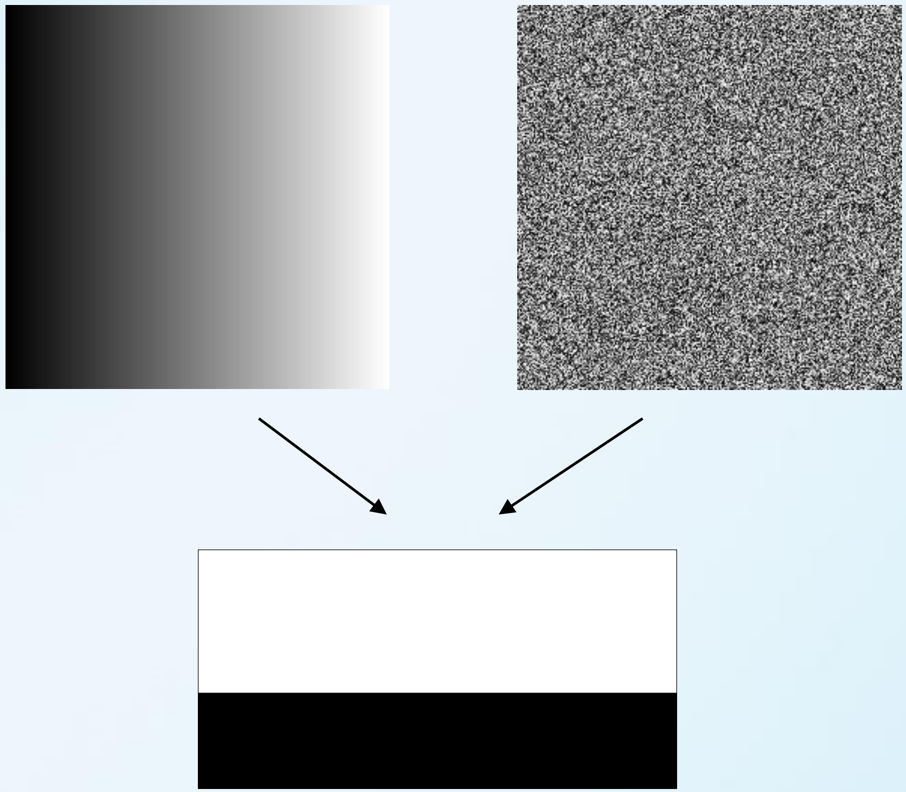
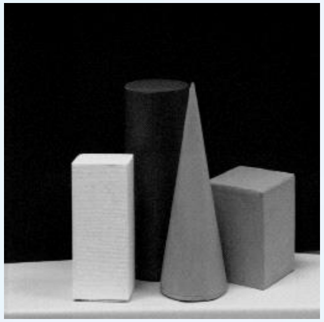


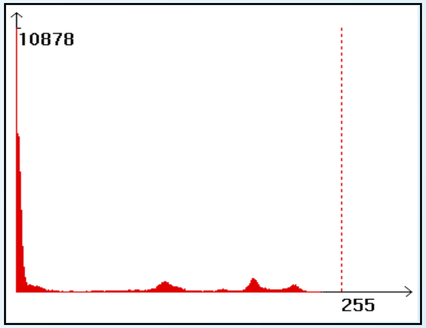
Image 2

Histogram

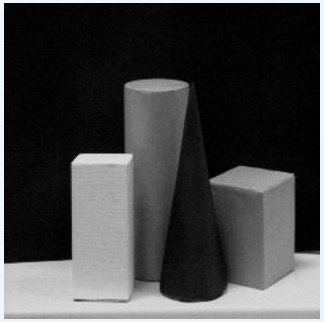
For colour images, each pixel has three components: red, green and blue. Thus, for a colour image, we have three histograms, one for each of the three components.

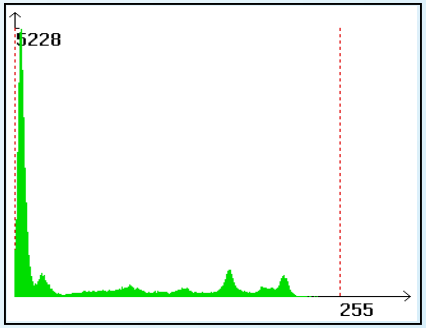
Red





Green





Blue