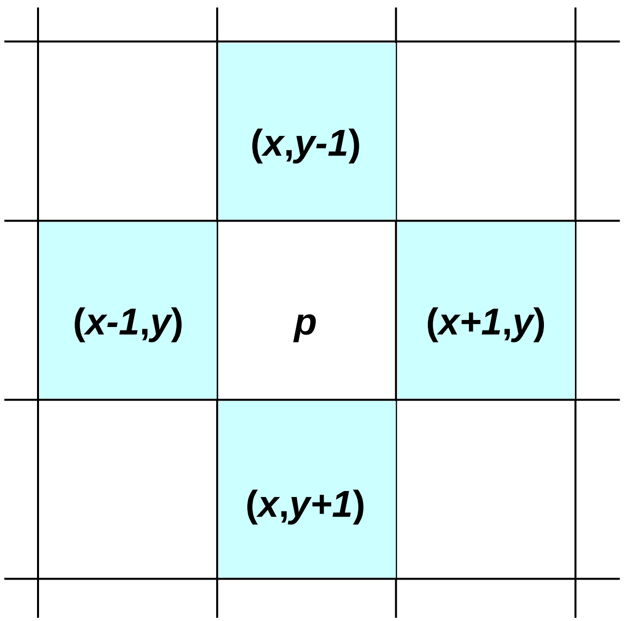
Neighbors of a Pixel

Neighborhood relation is used to tell adjacent pixels. It is useful for analyzing regions.



4-neighbors of *p*:

$$N_4(p) = \left\{ \begin{array}{c} (x-1,y) \\ (x+1,y) \\ (x,y-1) \\ (x,y+1) \end{array} \right\}$$

4-neighborhood relation considers only vertical and horizontal neighbors.

Note: $q \in N_4(p)$ implies $p \in N_4(q)$

Neighbors of a Pixel (cont.)

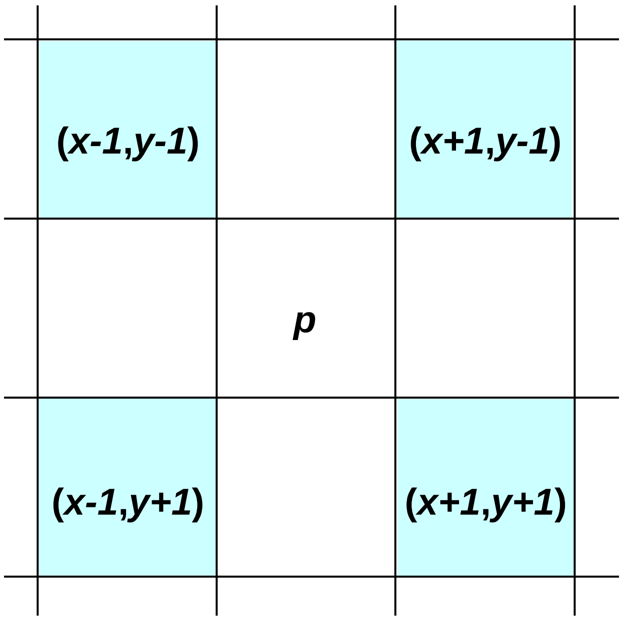
(x-1,y-1)	***(x,y-1)***	***(x+1,y-1)***
(x-1,y)	***p***	***(x+1,y)***
(x-1,y+1)	***(x,y+1)***	***(x+1,y+1)***

8-neighbors of *p*:

$$N_8(p) = \left\{ \begin{array}{c} (x-1,y-1) \\ (x,y-1) \\ (x+1,y-1) \\ (x-1,y) \\ (x+1,y) \\ (x-1,y+1) \\ (x,y+1) \\ (x+1,y+1) \end{array} \right\}$$

8-neighborhood relation considers all neighbor pixels.

Neighbors of a Pixel (cont.)



Diagonal neighbors of *p*:

$$N_D(p) = \left\{ \begin{array}{c} (x-1,y-1) \\ (x+1,y-1) \\ (x-1,y+1) \\ (x+1,y+1) \end{array} \right\}$$

Diagonal -neighborhood relation considers only diagonal neighbor pixels.

Some Basic Relationships Between Pixels

- Neighbors of a pixel
  - There are three kinds of neighbors of a pixel:
    - $N_4(p)$ 4-neighbors: the set of horizontal and vertical neighbors
    - $N_D(p)$ diagonal neighbors: the set of 4 diagonal neighbors
    - $N_8(p)$ 8-neighbors: union of 4-neighbors and diagonal neighbors

	O	
O	X	O
	O	

O		O
	X	
O		O

O	O	O
O	X	O
O	O	O

Some Basic Relationships Between Pixels

- Path:
  - The length of the path
  - Closed path
- Connectivity in a subset *S* of an image
  - Two pixels are connected if there is a path between them that lies completely within *S*.
- Connected component of *S*:
  - The set of all pixels in *S* that are connected to a given pixel in *S*.
- Region of an image
- Boundary, border or contour of a region
- Edge: a path of one or more pixels that separate two regions of significantly different gray levels.

Connectivity

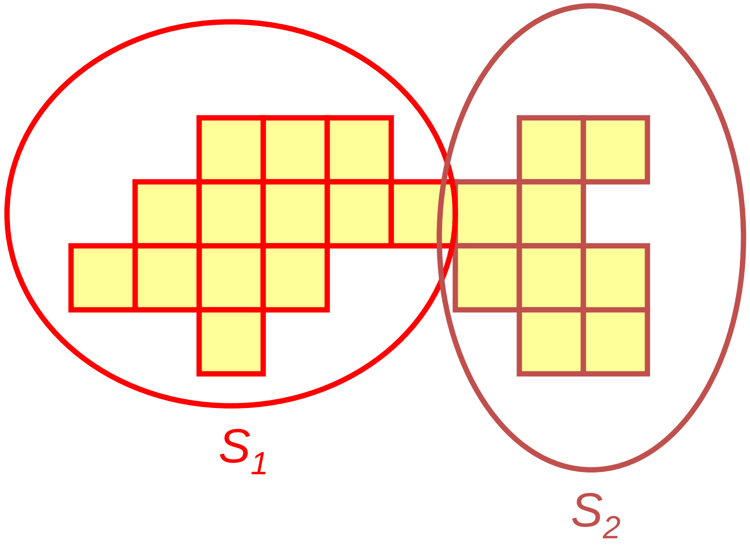
Connectivity is adapted from neighborhood relation. Two pixels are connected if they are in the same class (i.e. the same color or the same range of intensity) and they are neighbors of one another.

For p and q from the same class

- 4-connectivity: p and q are 4-connected if $q \in N_4(p)$
- 8-connectivity: p and q are 8-connected if $q \in N_8(p)$
- mixed-connectivity (m-connectivity):
  p and q are m-connected if $q \in N_4(p)$ or
  $q \in N_D(p)$ and $N_4(p) \cap N_4(q) = \varnothing$

Adjacency

A pixel *p* is *adjacent* to pixel *q* is they are connected.
Two image subsets S_1 and S_2 are adjacent if some pixel in S_1 is adjacent to some pixel in S_2



We can define type of adjacency: 4-adjacency, 8-adjacency or m-adjacency depending on type of connectivity.

Path

A ***path*** from pixel p at (x,y) to pixel q at (s,t) is a sequence of distinct pixels:

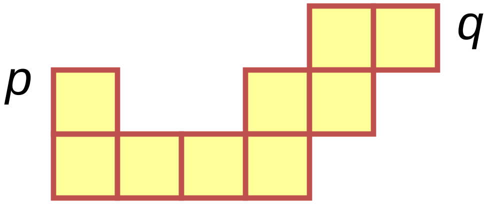
$$(x_0,y_0), (x_1,y_1), (x_2,y_2),\ldots, (x_n,y_n)$$

such that

$$(x_0,y_0) = (x,y) \text{ and } (x_n,y_n) = (s,t)$$

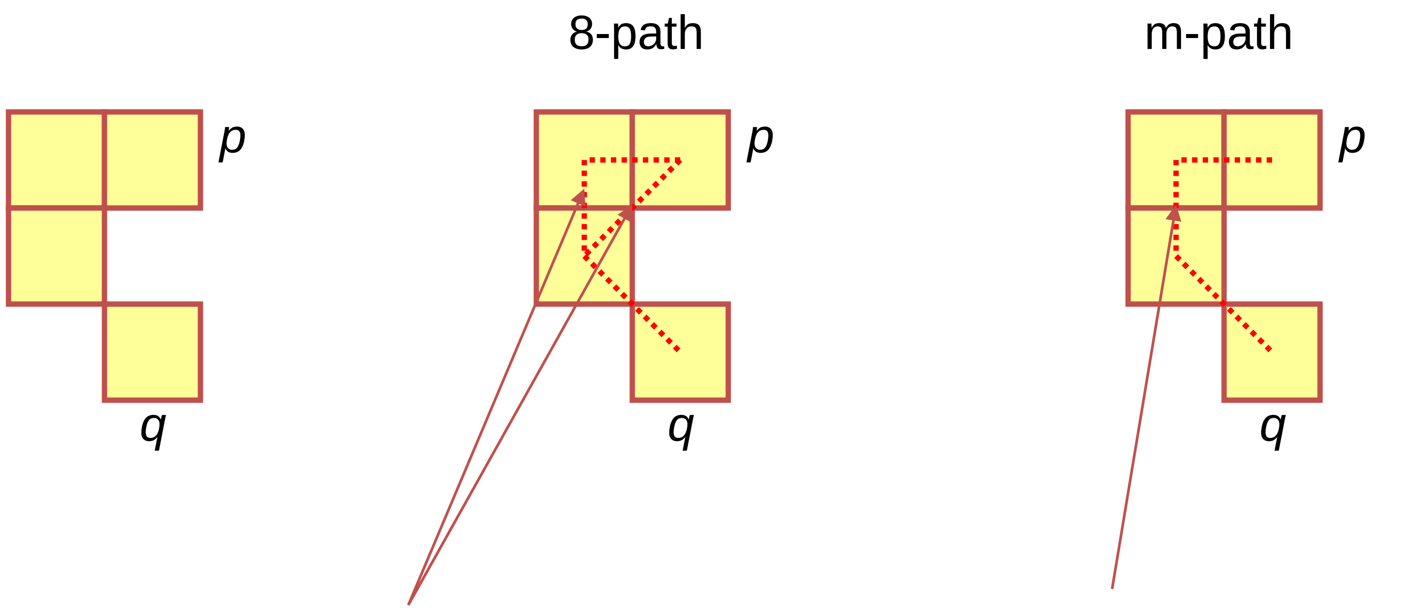
and

$$(x_i,y_i) \text{ is adjacent to } (x_{i-1},y_{i-1}), \qquad i = 1,\ldots,n$$



We can define type of path: 4-path, 8-path or m-path depending on type of adjacency.

Path (cont.)

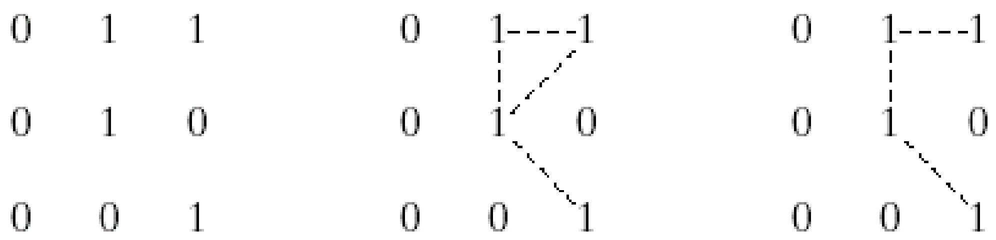


8-path from *p* to *q*
results in some ambiguity

m-path from *p* to *q*
solves this ambiguity

Some Basic Relationships Between Pixels

- An example of adjacency:



a b c

FIGURE 2.26 (a) Arrangement of pixels; (b) pixels that are 8-adjacent (shown dashed) to the center pixel; (c) *m*-adjacency.

Some Basic Relationships Between Pixels

- Adjacency:
  - Two pixels that are neighbors and have the same grey-level (or some other specified similarity criterion) are adjacent
  - Pixels can be 4-adjacent, diagonally adjacent, 8-adjacent, or m-adjacent.
- *m*-adjacency (mixed adjacency):
  - Two pixels *p* and *q* of the same value (or specified similarity) are *m*-adjacent if either
    - (i) *q* and *p* are 4-adjacent, or
    - (ii) *p* and *q* are diagonally adjacent and do not have any common 4-adjacent neighbors.
    - They cannot be both (i) and (ii).

Distance

For pixel *p*, *q*, and *z* with coordinates (x,y), (s,t) and (u,v), *D* is a ***distance function*** or ***metric*** if

- $D(p,q) \geq 0$ $\quad$ ($D(p,q) = 0$ if and only if $p = q$)
- $D(p,q) = D(q,p)$
- $D(p,z) \leq D(p,q) + D(q,z)$

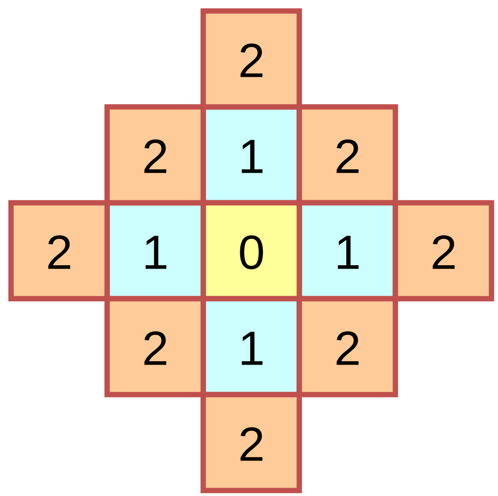
Example: Euclidean distance

$$D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$$

Distance (cont.)

D_4-distance (*city-block distance or* Manhattan) is defined as

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



Pixels with $D_4(p) = 1$ is 4-neighbors of p.

Distance (cont.)

D_8-distance (*chessboard distance*) is defined as

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

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

Pixels with $D_8(p) = 1$ is 8-neighbors of *p*.

Zooming and Shrinking Digital Images

- Zooming: increasing the number of pixels in an image so that the image appears larger
  - Nearest neighbor interpolation
    - For example: pixel replication--to repeat rows and columns of an image
  - Bilinear interpolation
    - Smoother
  - Higher order interpolation
- Image shrinking: subsampling

Zooming and Shrinking Digital Images

Nearest neighbor
Interpolation
(Pixel replication)

Bilinear
interpolation

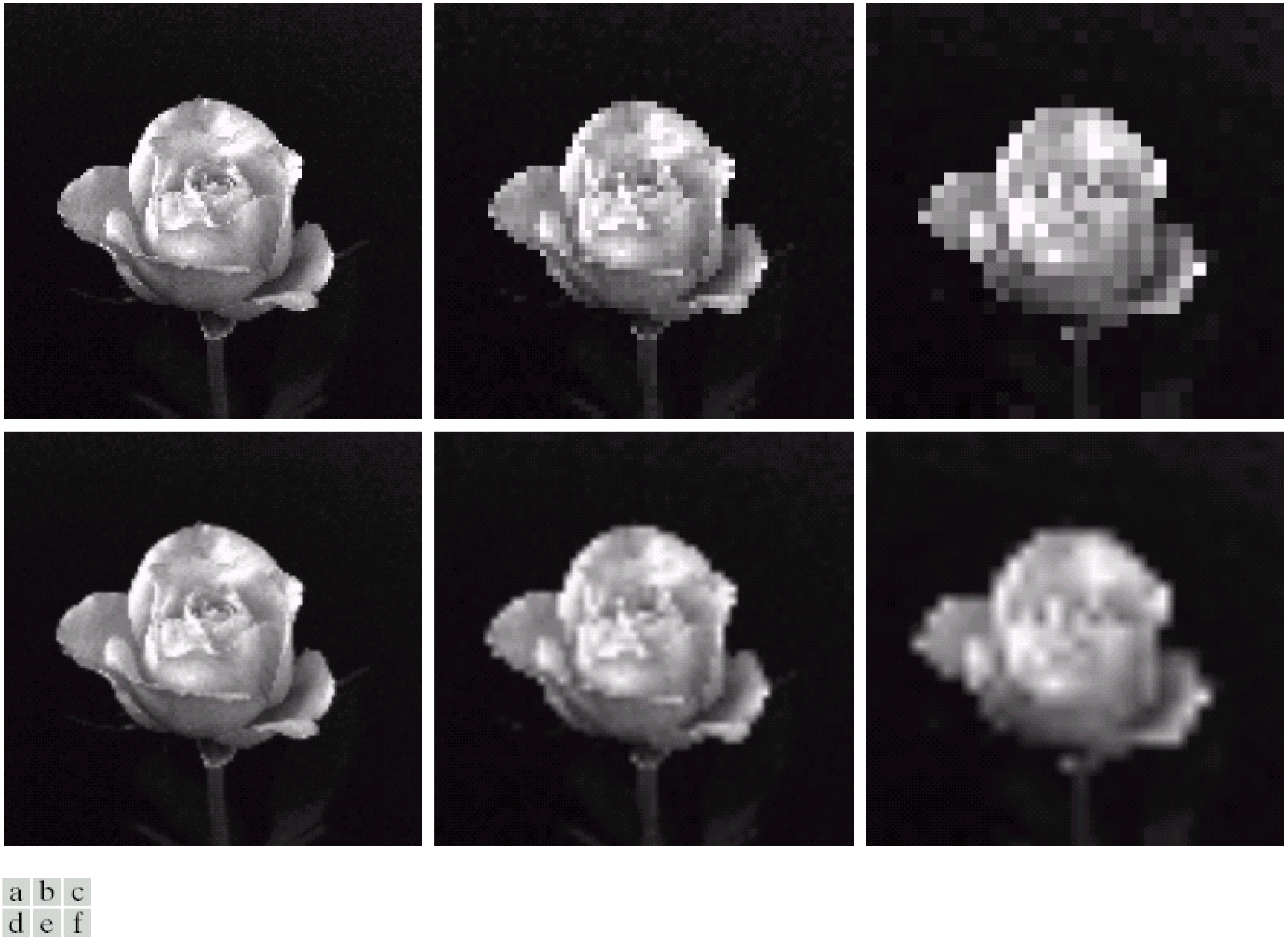


FIGURE 2.25 Top row: images zoomed from 128 × 128, 64 × 64, and 32 × 32 pixels to 1024 × 1024 pixels, using nearest neighbor gray-level interpolation. Bottom row: same sequence, but using bilinear interpolation.

Face Recognition

Face Expression

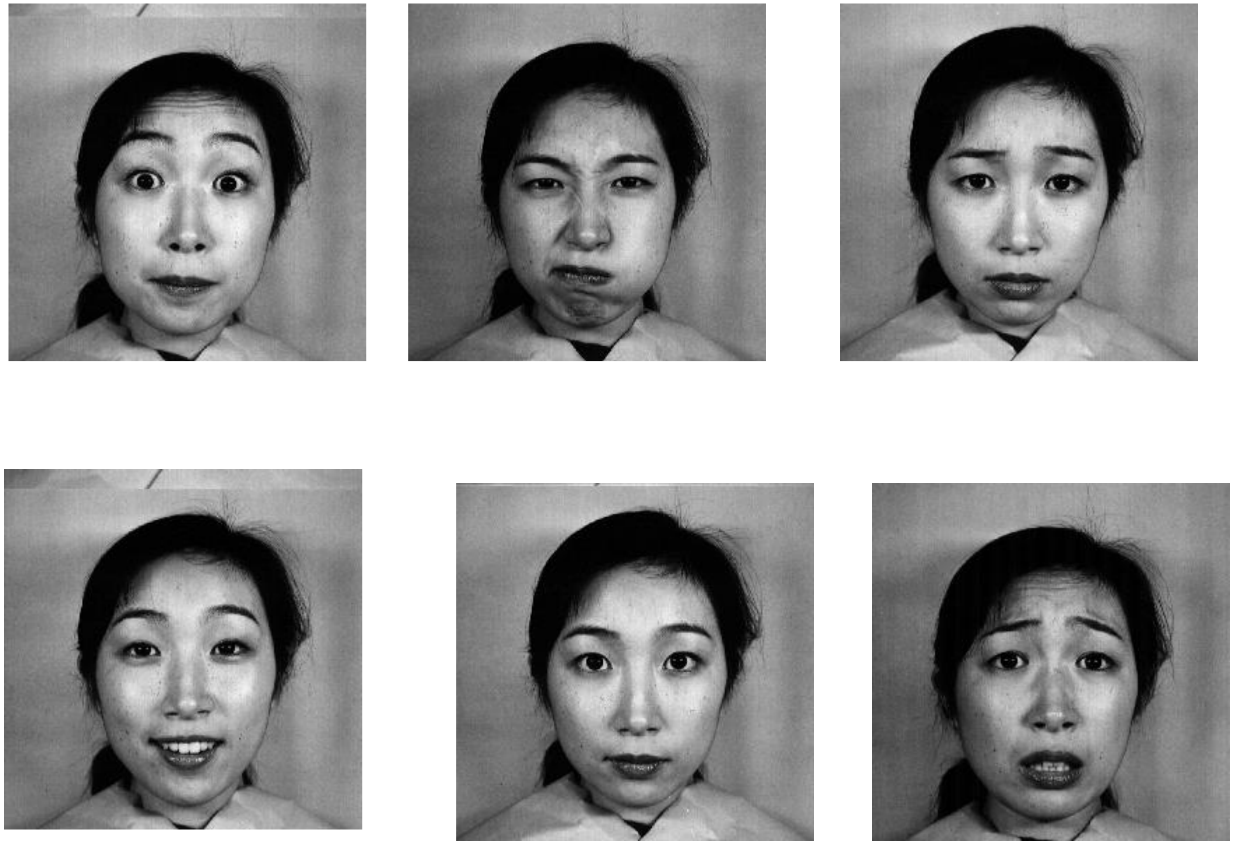
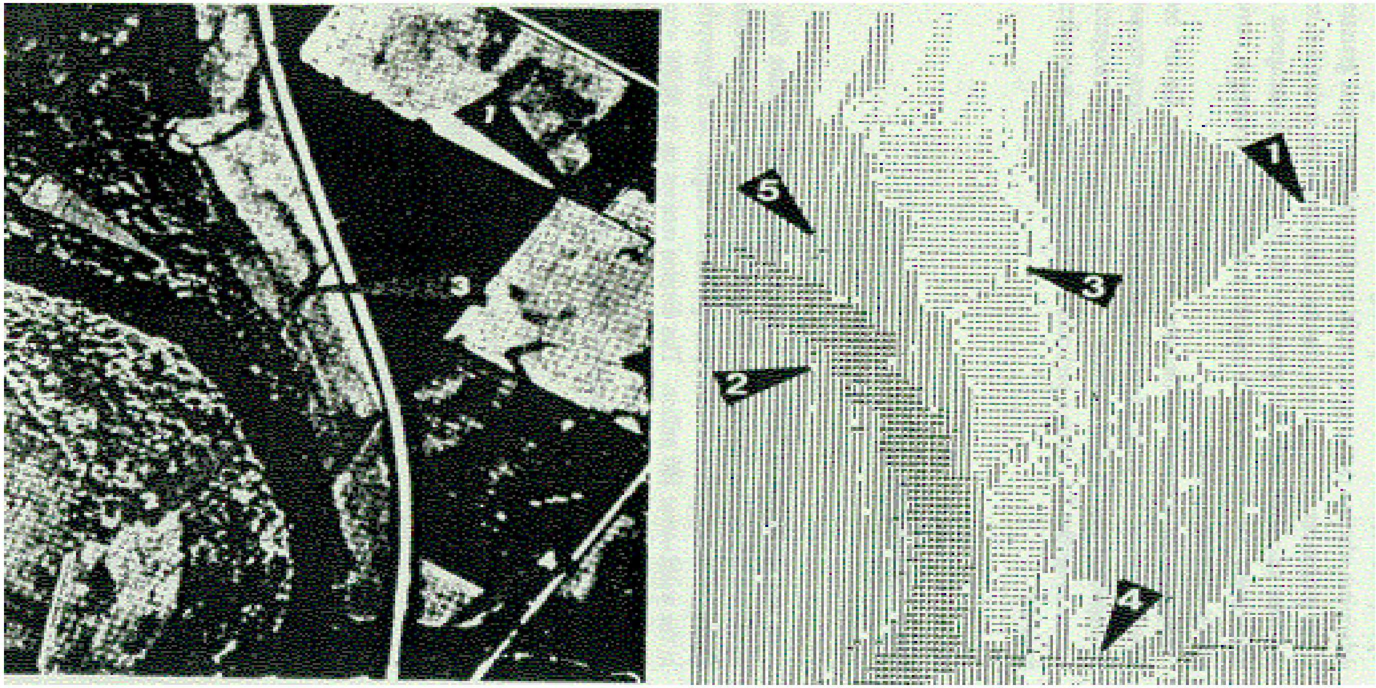


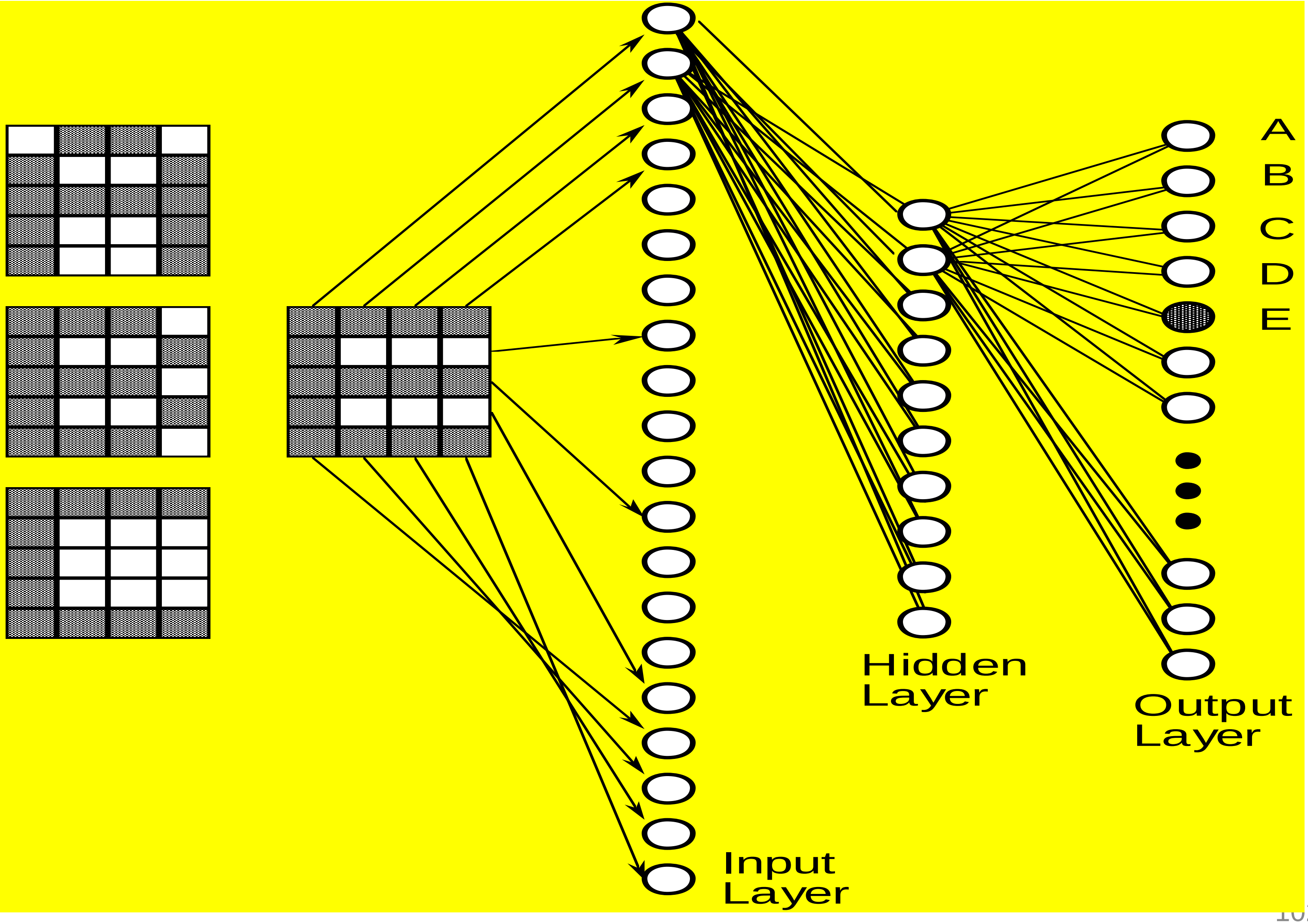
Figure 1 Different Facial Expressions of same person [23]

Classification of Remotely Sensed Data

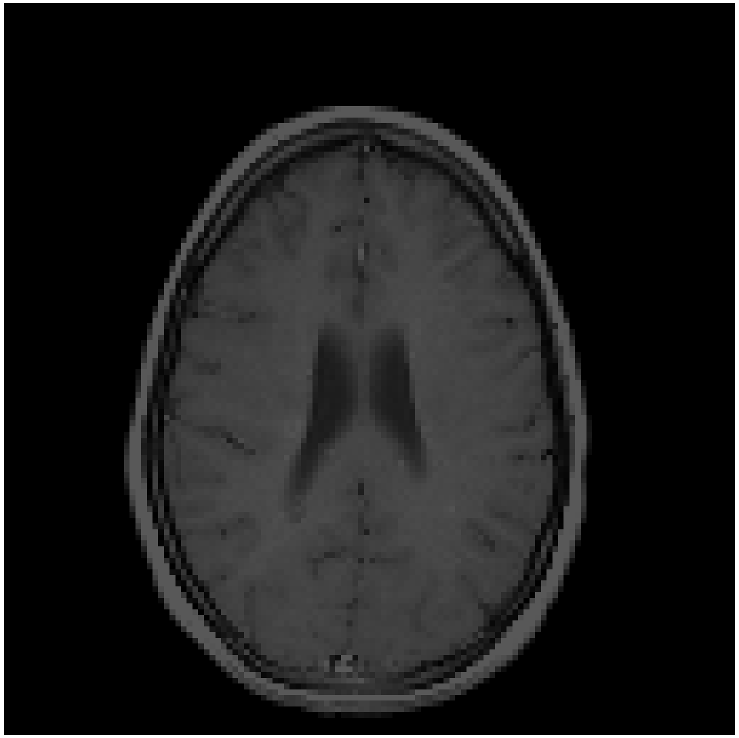


Fingerprint recognition

Optical Character Recognition



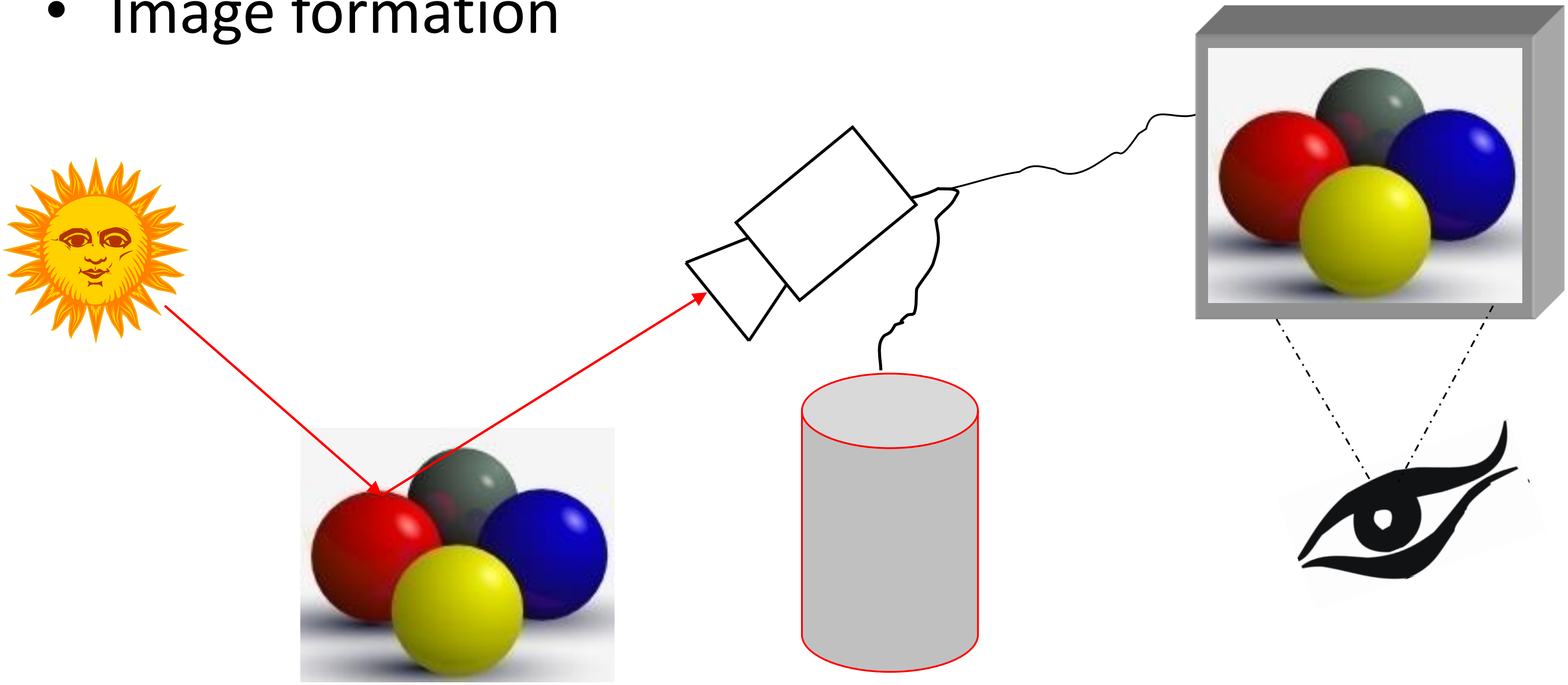
MRI Image



Signature recognition

- Each person's signature is different.
- There are structural similarities which are difficult to quantify.
- One company has manufactured a machine which recognizes signatures to within a high level of accuracy.
  - **Makes forgery even more difficult.**

- Image formation



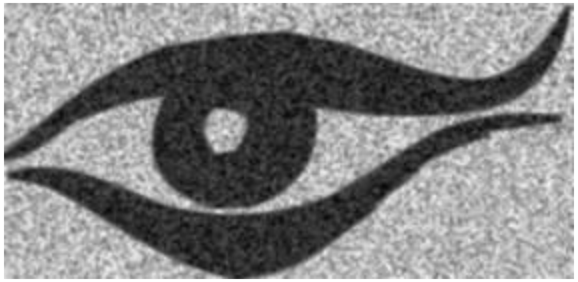
Result of Image processing…

- Image processing theory and practices

Why this is possible?
How ?
Theory
Practice

And much more …

- Edge detection and image segmentation



How ?
Theory
Practice