ARTIFICIAL INTELLIGENCE

Image Processing and Computer Vision

Aims

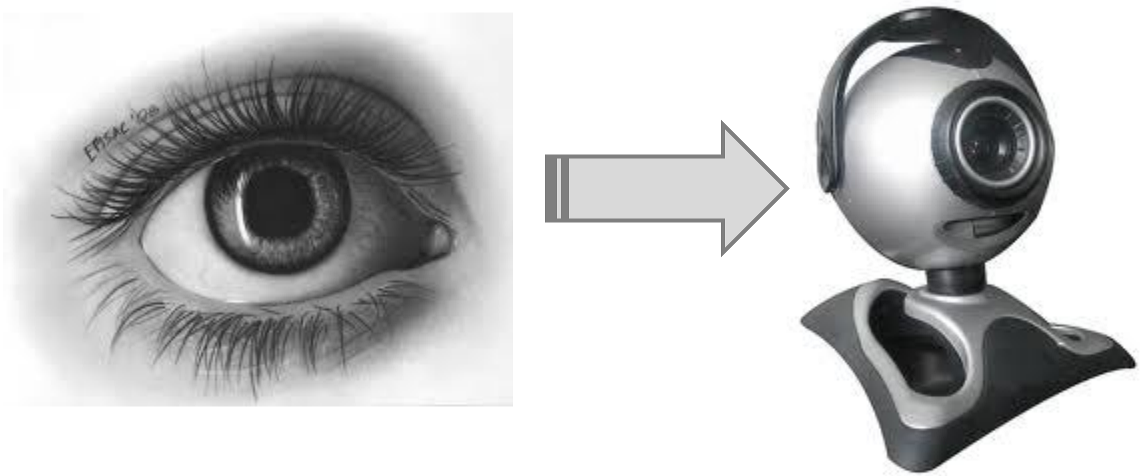
- □ Introduction to Image Processing
- □ Introduction to Computer Vision

Image Processing

- **Image Processing** is any form of signal processing for which the input is an image, such as photographs or frames of video;
- the output of image processing can be either an image or a set of characteristics or parameters related to the image.
- Most image-processing techniques involve treating the image as a 2D signal and applying standard signal-processing techniques to it.

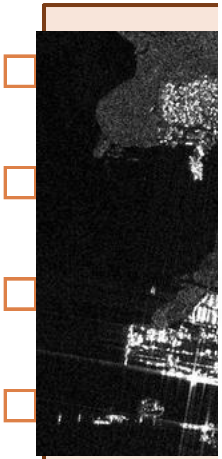
Computer Vision

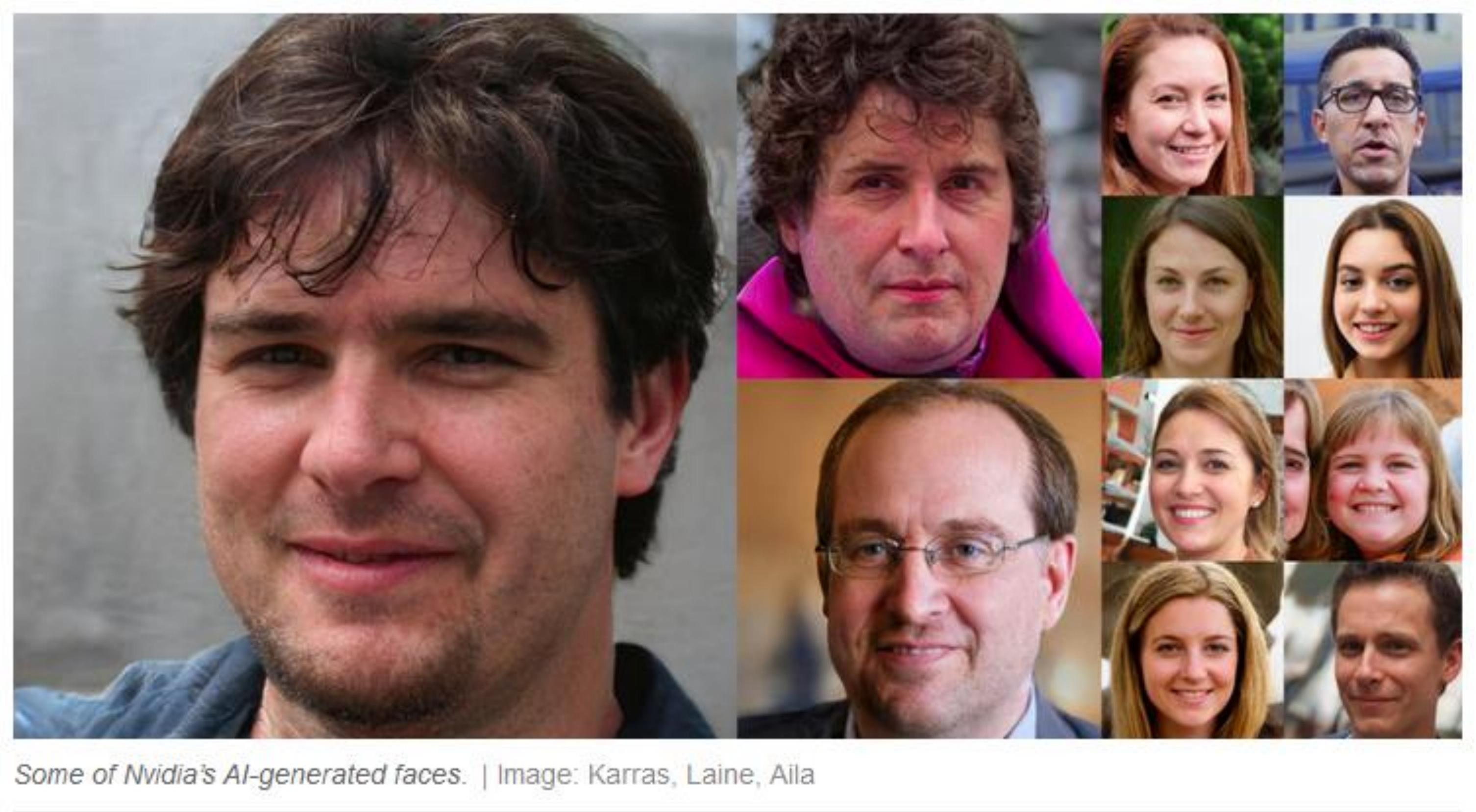
- **Computer Vision** is a field that includes methods for acquiring, processing, analyzing, and understanding images / video.



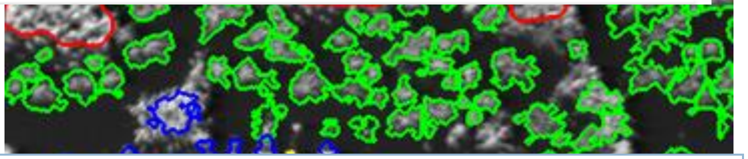
Applications

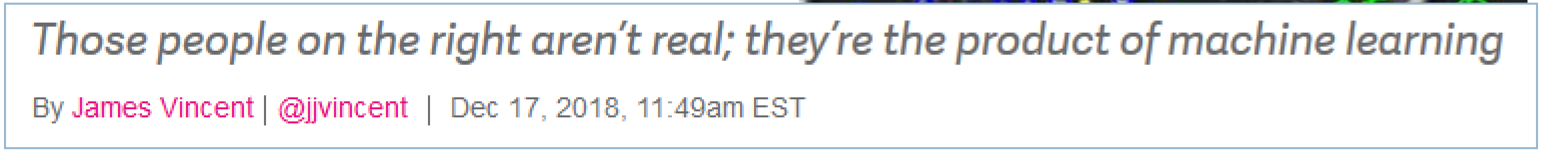
- □ Comp
- □ Face
- □
- □
- □
- □
- □ Remo
- □ Face generation





Some of Nvidia's AI-generated faces. | Image: Karras, Laine, Aila

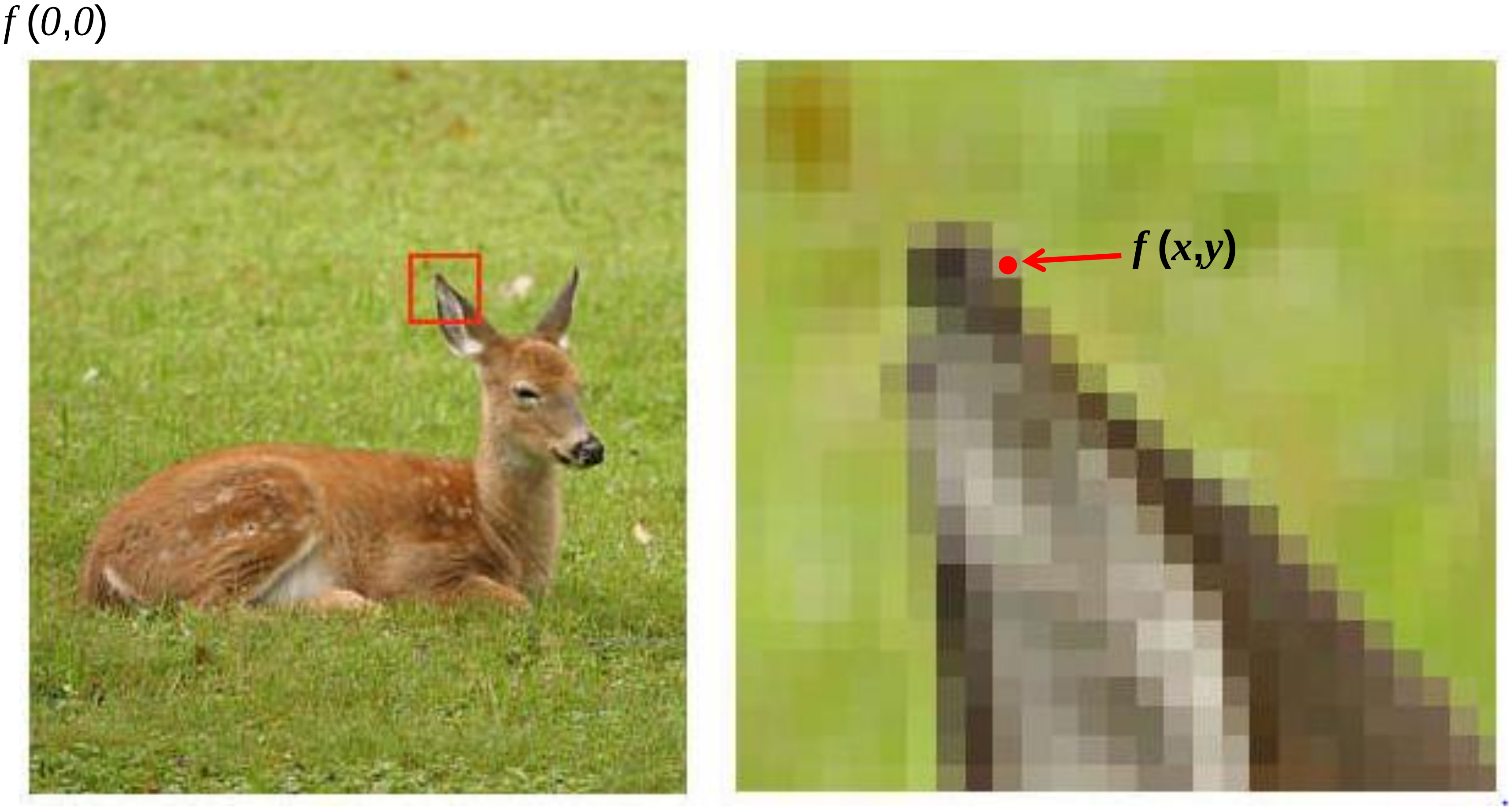




Those people on the right aren't real; they're the product of machine learning

By James Vincent | @jjvincent | Dec 17, 2018, 11:49am EST

Digital Image Representation



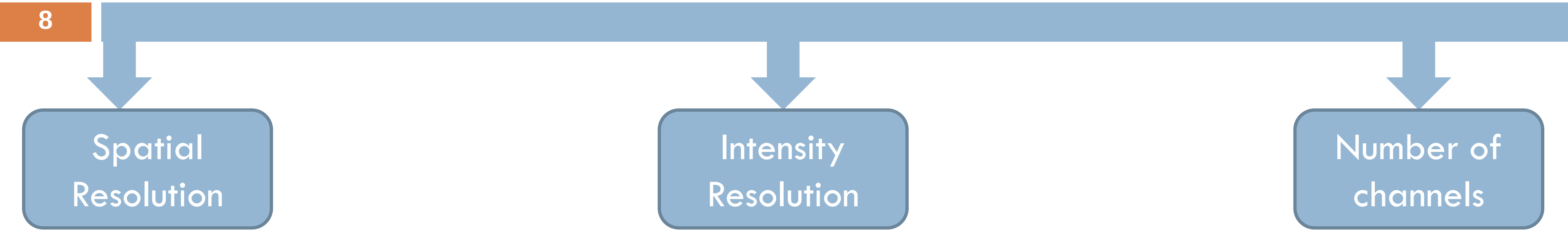
$f\,(x,y)$ is proportional to the brightness (or gray level) of the image at that point

Digital Image Representation

- A digital image is an image $f(x,y)$ that has been discretized both in spatial coordinates and brightness.
- A digital image can be considered as a matrix whose row and column indices identify a point in the image and the corresponding matrix element value identifies the gray level at that point.
- The elements of such a digital array called picture elements or pixels.

Do you know what “pixel” stands for?

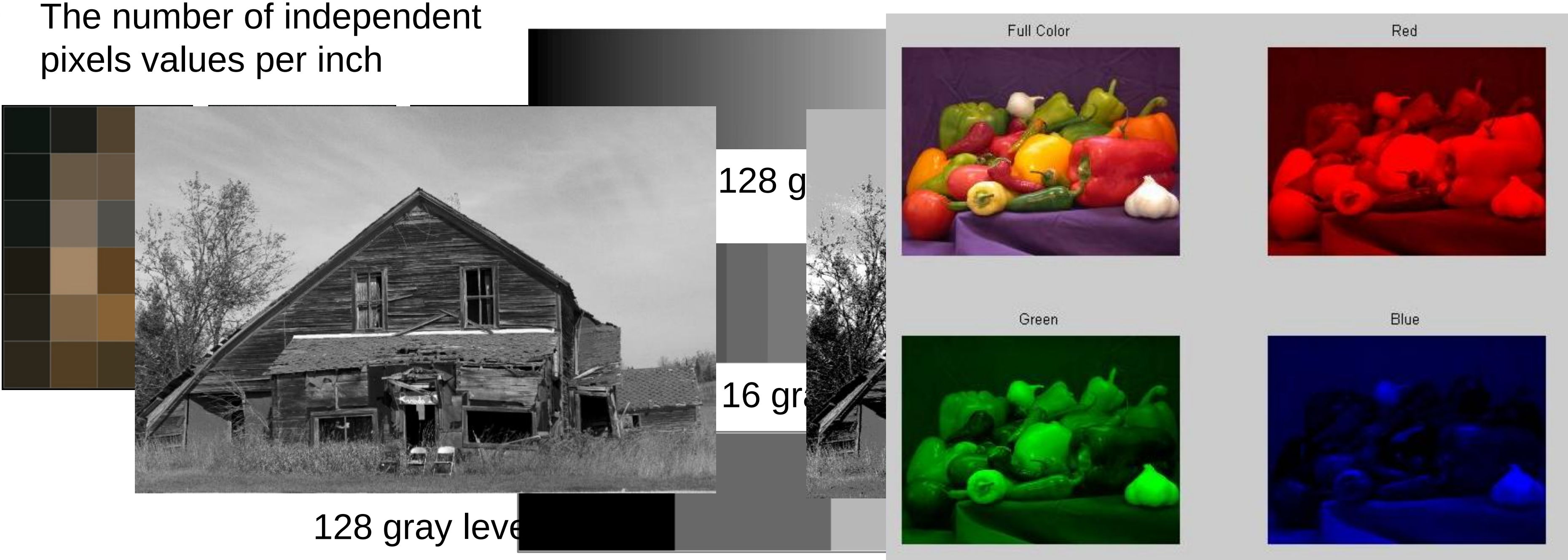
Properties of Images



- Width pixels/ width cm and height pixels / height cm
- The number of independent pixels values per inch

- Intensity bits / intensity range (per channel)

- RGB is 3 channels
- Grayscale is 1 channel



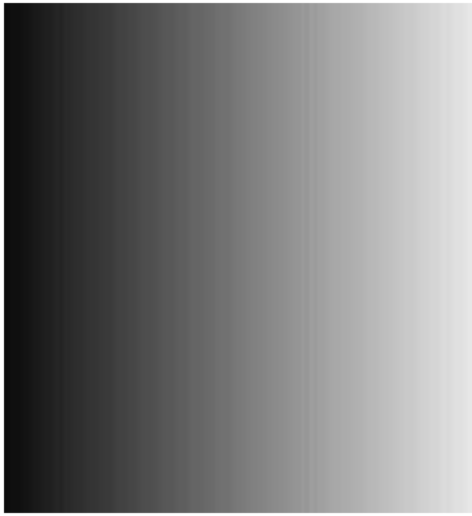
128 gray leve

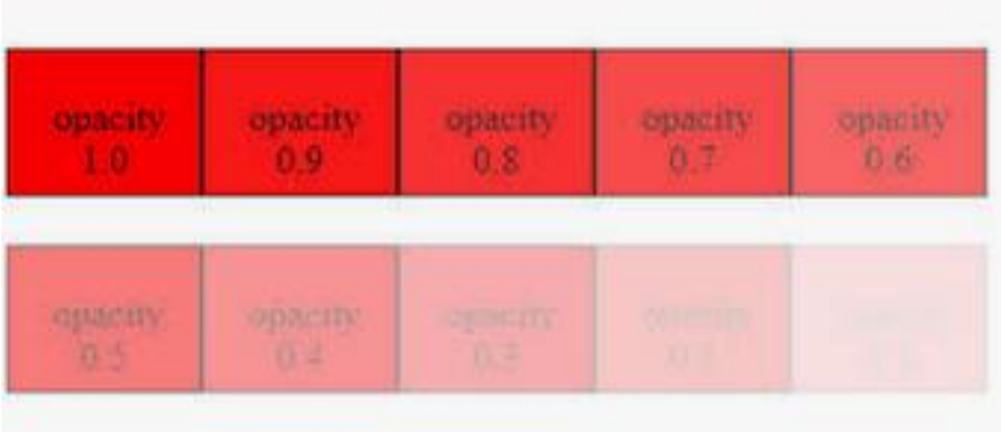
4 gray levels

What is a Digital Image?

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

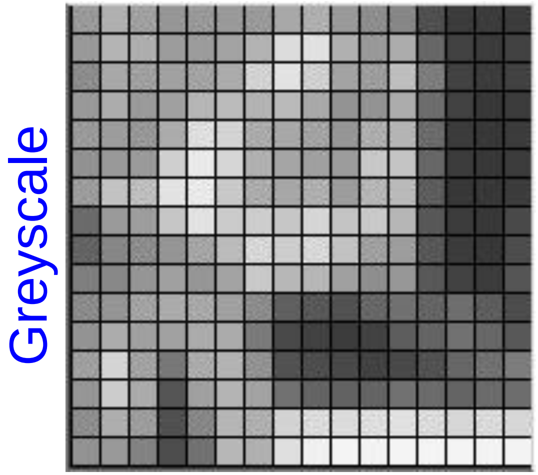
- □Pixel values typically represent gray levels, colours, opacities etc



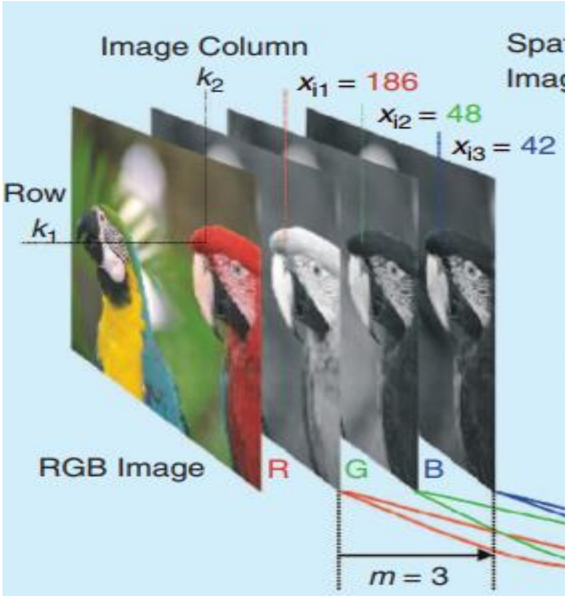


Grayscale VS Color Images

- An image has a vector at each pixel. For colour images, these vectors each have 3 components (RGB) while Grayscale images have only 1 component.



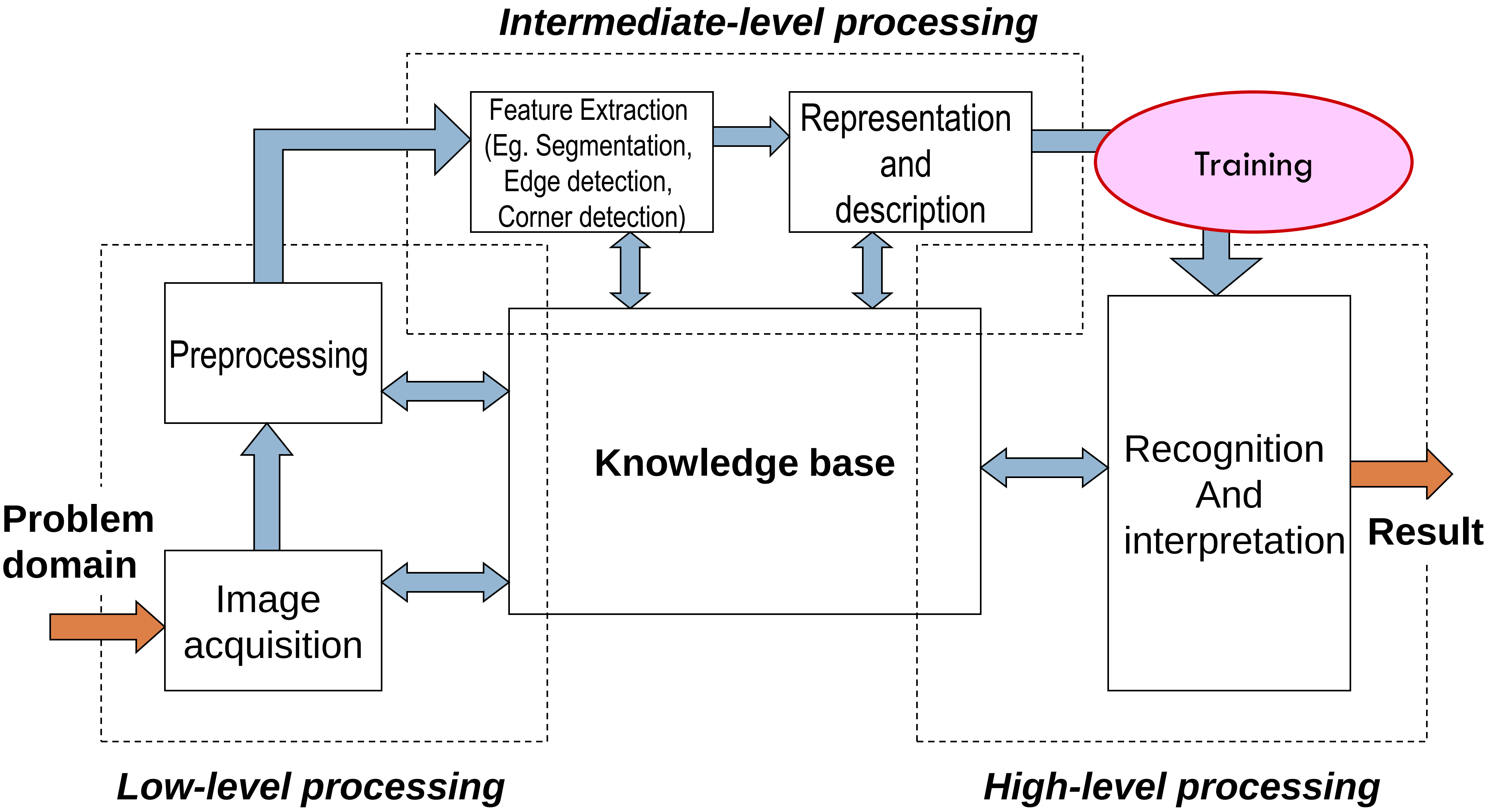
$f(x, y) \rightarrow \{0,1,\ldots, N\}$



$f(x, y) \rightarrow [\ \{0,\ldots,N\}, \{0,\ldots,N\}, \{0,\ldots,N\}\ ]$

* Normally 0 to 255

Fundamental Steps in Image Processing


Intermediate-level processing
Feature Extraction (Eg. Segmentation, Edge detection, Corner detection)
Representation and description
Training
Preprocessing
Knowledge base
Recognition And interpretation
Problem domain
Image acquisition
Result
Low-level processing
High-level processing

Fruit Recognition

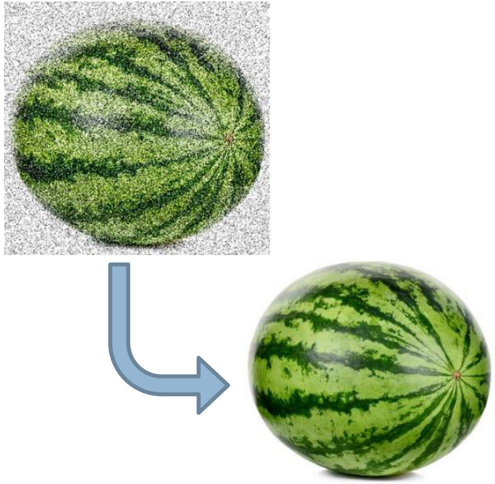
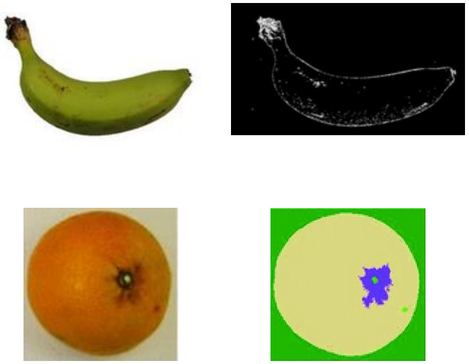


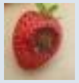
Image pre-processing



Feature extraction

Descriptor in vector
= [1.0 1.0 0.9 0.6 0.6 1.0 1.0 1.0 1.0 1.0 1.0 1.0 0.5 .... 1.0]

Statistical features

Image	Entropy	SD	Mean
	7.7	65	142
	7.5	56	109

Representation & description

Image acquisition

Knowledge base

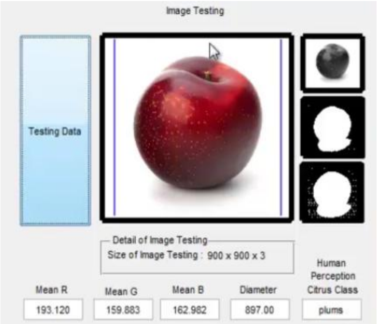


Image Testing

Testing Data

Detail of Image Testing
Size of Image Testing : 900 x 900 x 3

Mean R	Mean G	Mean B	Diameter	Human Perception Citrus Class
193.120	159.883	162.982	897.00	plums

Recognition & interpretation

Image Acquisition

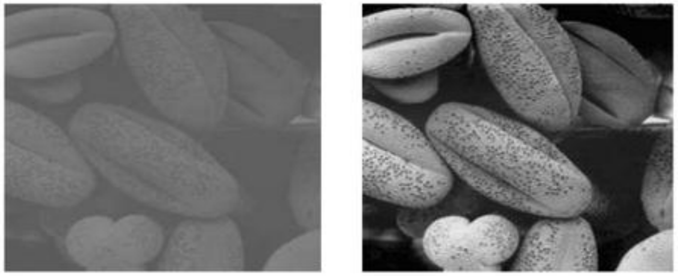
- □ Requires imaging sensor and capability to digitize the signal produced by the sensor.
- □ Example of imaging sensor: digital camera, scanner

Preprocessing

- To improve the image (to increase chance for success of other processes)

Examples

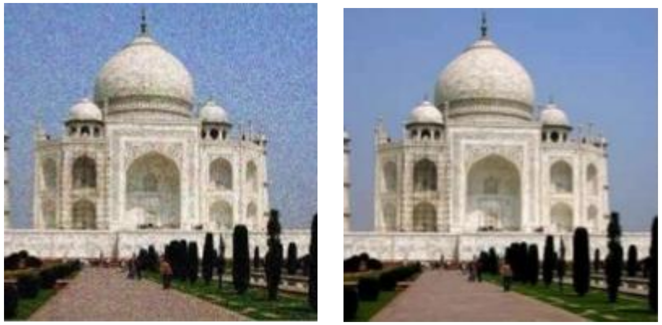
Enhancing Contrast



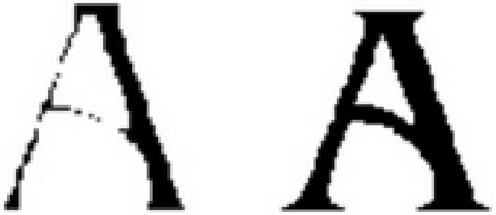
Original Image

Contrast Enhanced Image

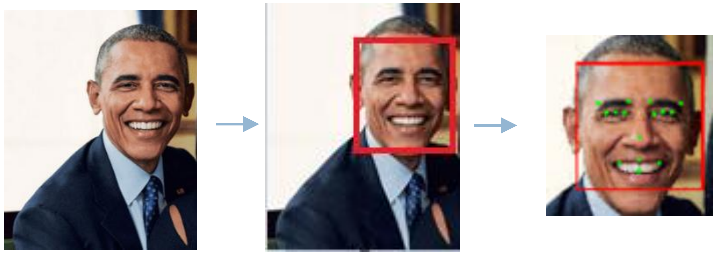
Removing noise



Morphological Operation



Isolating Regions

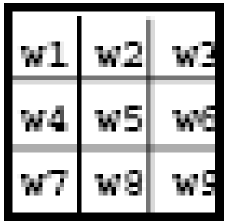


Preprocessing

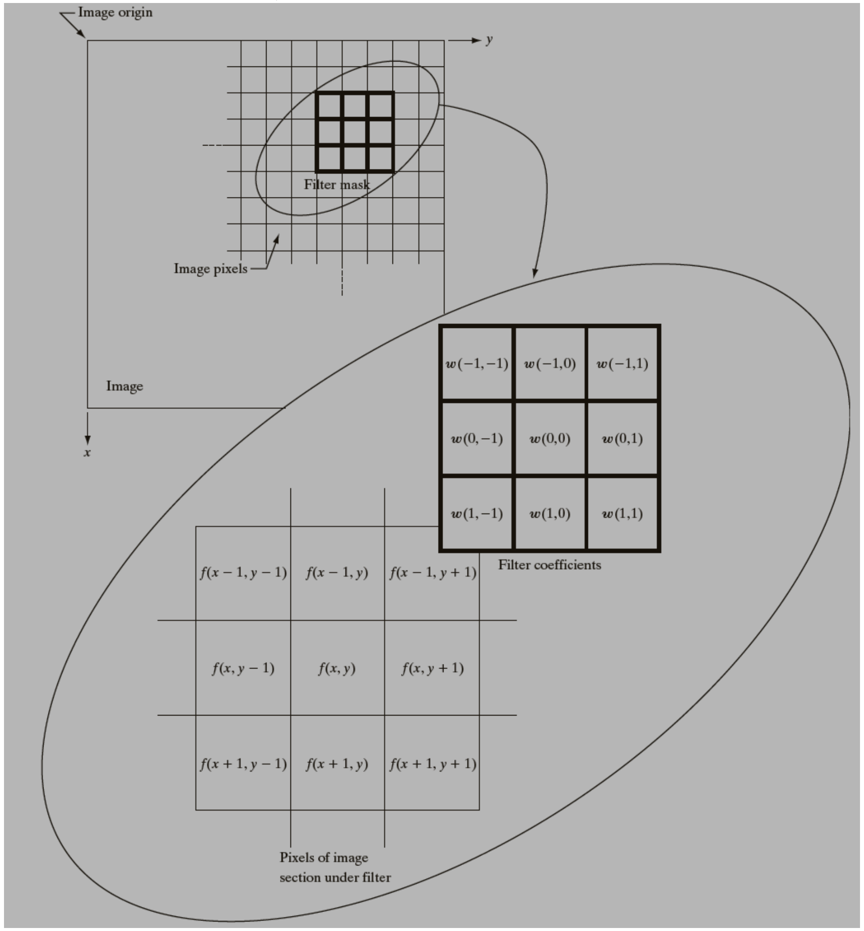
- □ Beforehand, you need to understand how image filtering (Correlation & Convolution) works.

- □ Typically linear combinations of pixel values.
  - e.g., weight pixel values and add them together.
- □ Different results can be obtained using different weights.
  - e.g., smoothing, sharpening, edge detection).

Filter/mask



w1	w2	w3
w4	w5	w6
w7	w8	w9



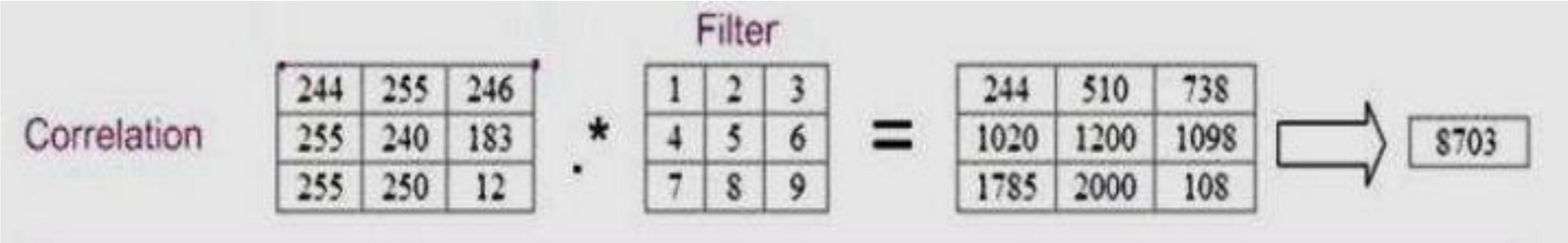
$w(-1,-1)$	$w(-1,0)$	$w(-1,1)$
$w(0,-1)$	$w(0,0)$	$w(0,1)$
$w(1,-1)$	$w(1,0)$	$w(1,1)$

Filter coefficients

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

Pixels of image section under filter

Correlation



Correlation

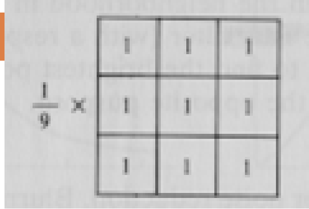
$$\begin{bmatrix} 244 & 255 & 246 \\ 255 & 240 & 183 \\ 255 & 250 & 12 \end{bmatrix} .* \overset{\text{Filter}}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}} = \begin{bmatrix} 244 & 510 & 738 \\ 1020 & 1200 & 1098 \\ 1785 & 2000 & 108 \end{bmatrix} \Rightarrow \boxed{8703}$$

244	255	246
255	240	183
255	250	12

→

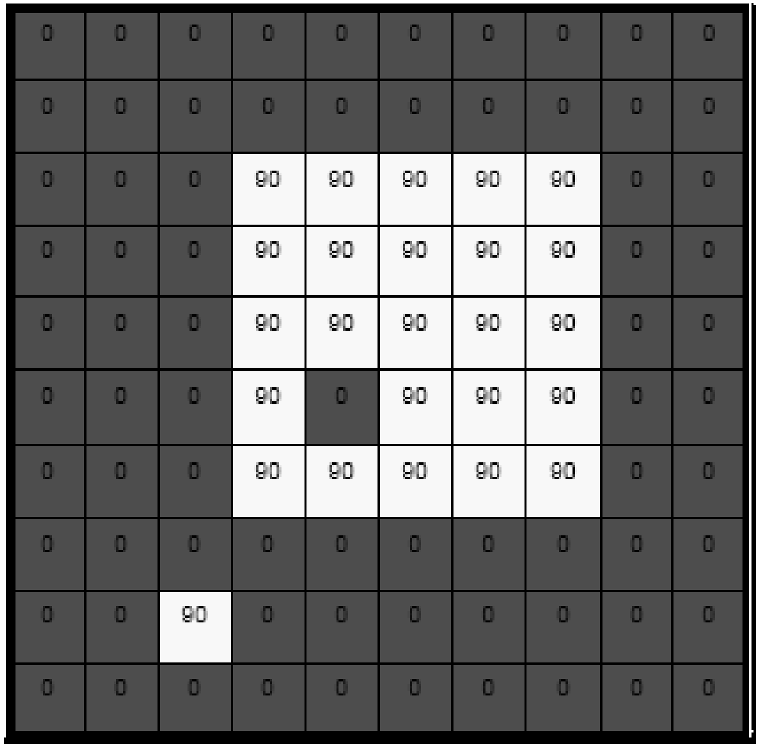
	8703	

Correlation – Example

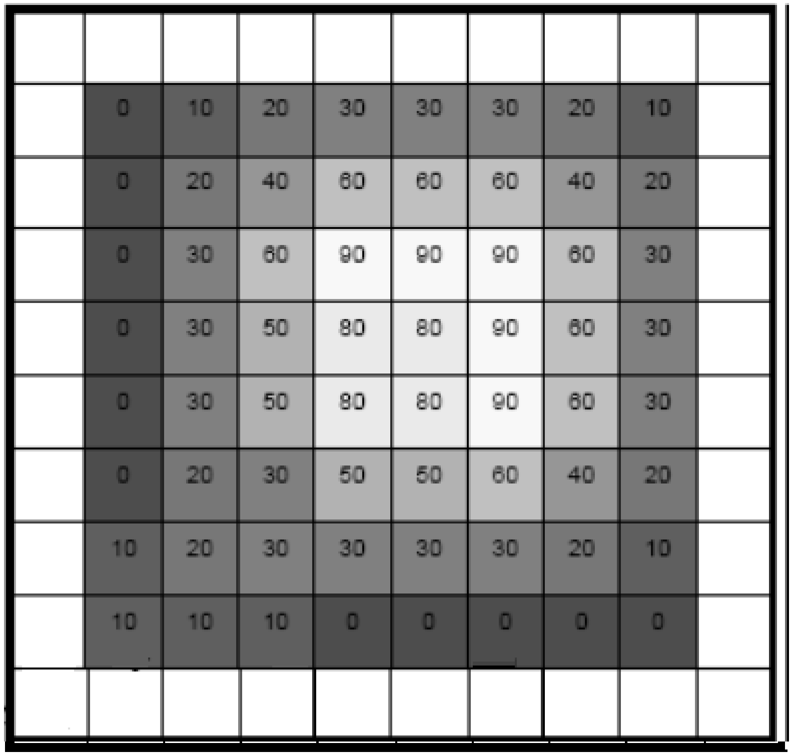


$$\frac{1}{9}\times\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

Average filter



0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0



	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

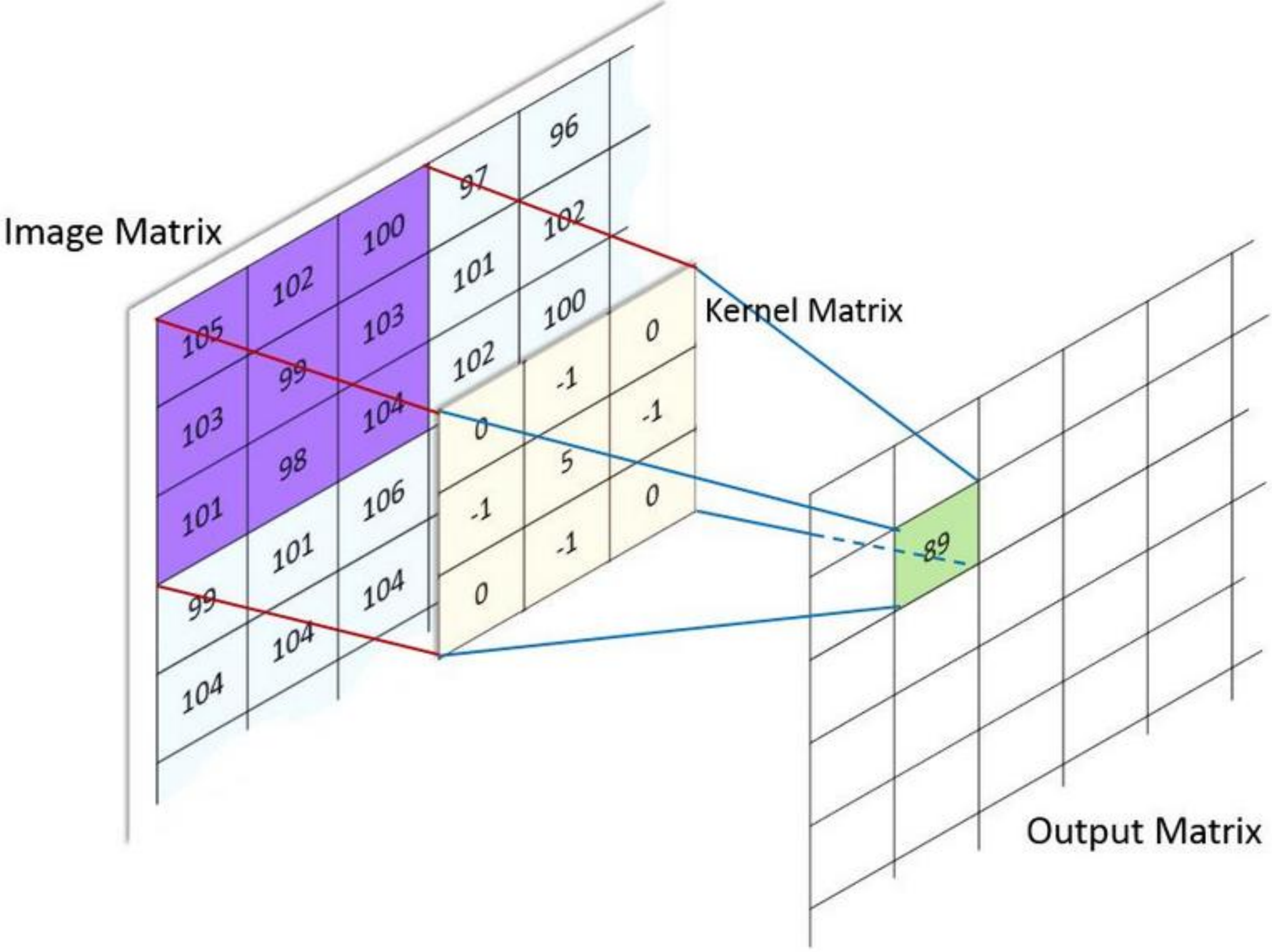
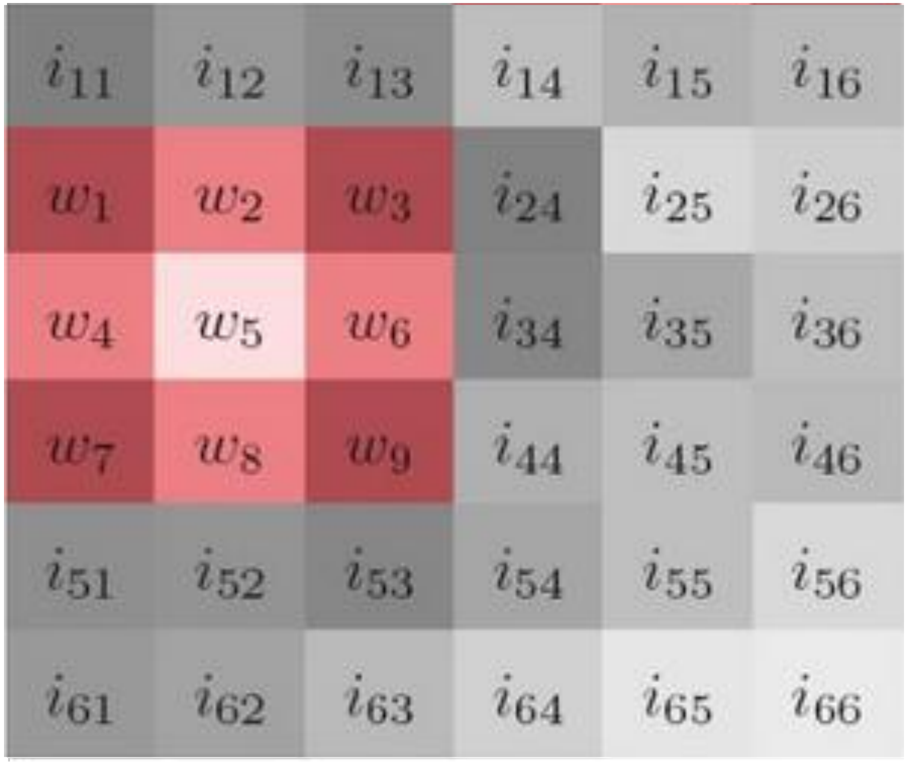


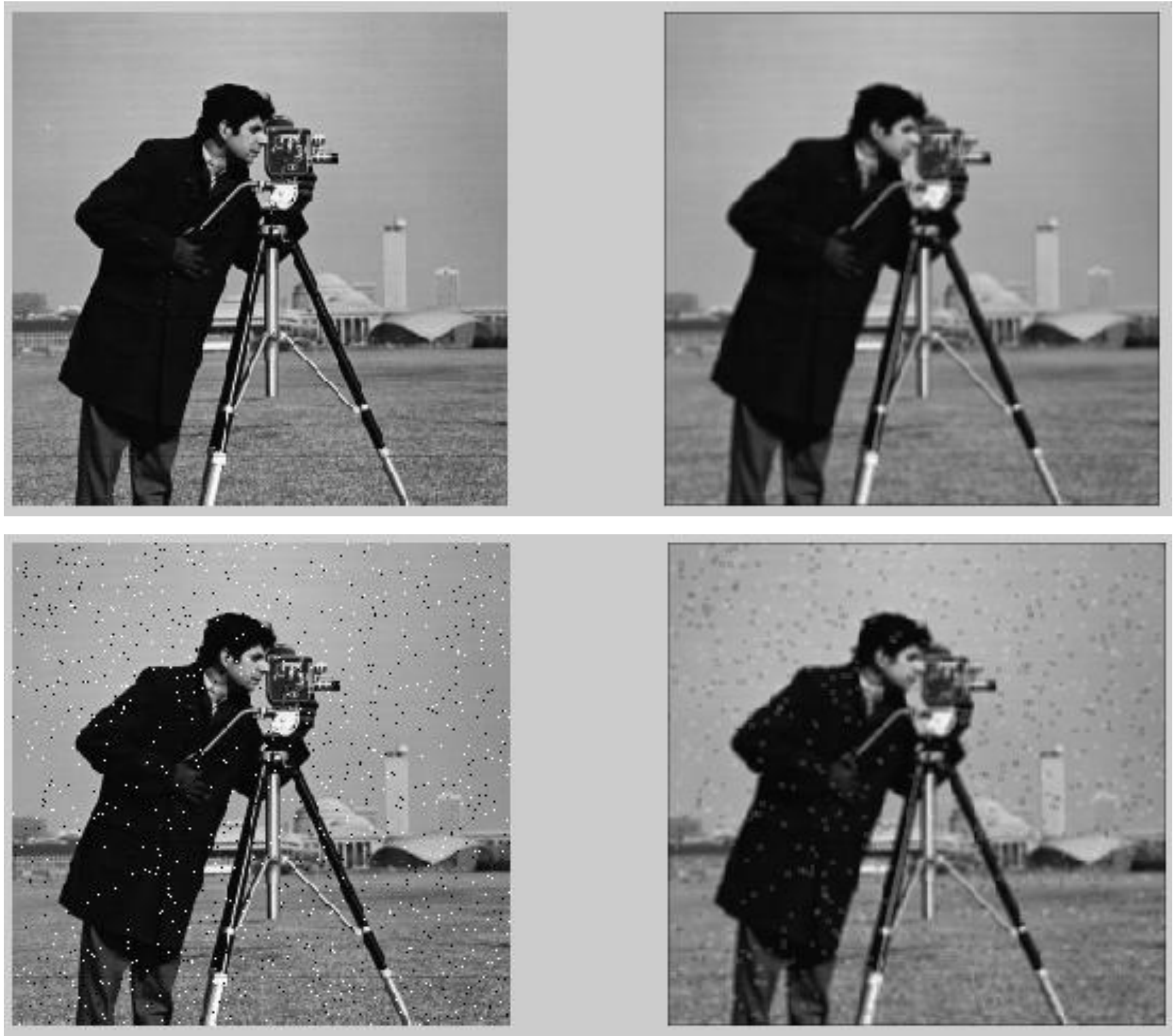
Image Matrix
105 102 100 97 96
103 99 103 101 102
101 98 104 102 100
99 101 106
104 104 104
Kernel Matrix
0 -1 0
-1 5 -1
0 -1 0
89
Output Matrix

Filtering-move filter over the image



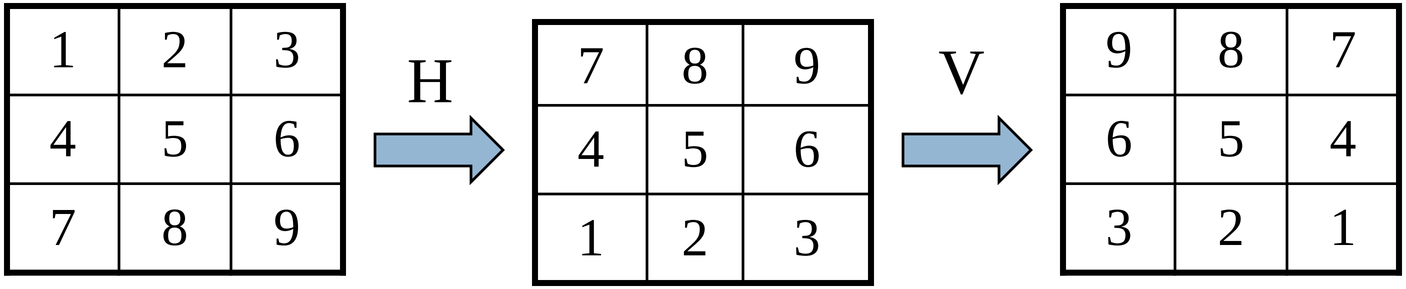
i_{11}	i_{12}	i_{13}	i_{14}	i_{15}	i_{16}
w_1	w_2	w_3	i_{24}	i_{25}	i_{26}
w_4	w_5	w_6	i_{34}	i_{35}	i_{36}
w_7	w_8	w_9	i_{44}	i_{45}	i_{46}
i_{51}	i_{52}	i_{53}	i_{54}	i_{55}	i_{56}
i_{61}	i_{62}	i_{63}	i_{64}	i_{65}	i_{66}

Correlation – Examples



Convolution

- Same as correlation except that the mask is **flipped,** both horizontally and vertically.



1	2	3
4	5	6
7	8	9

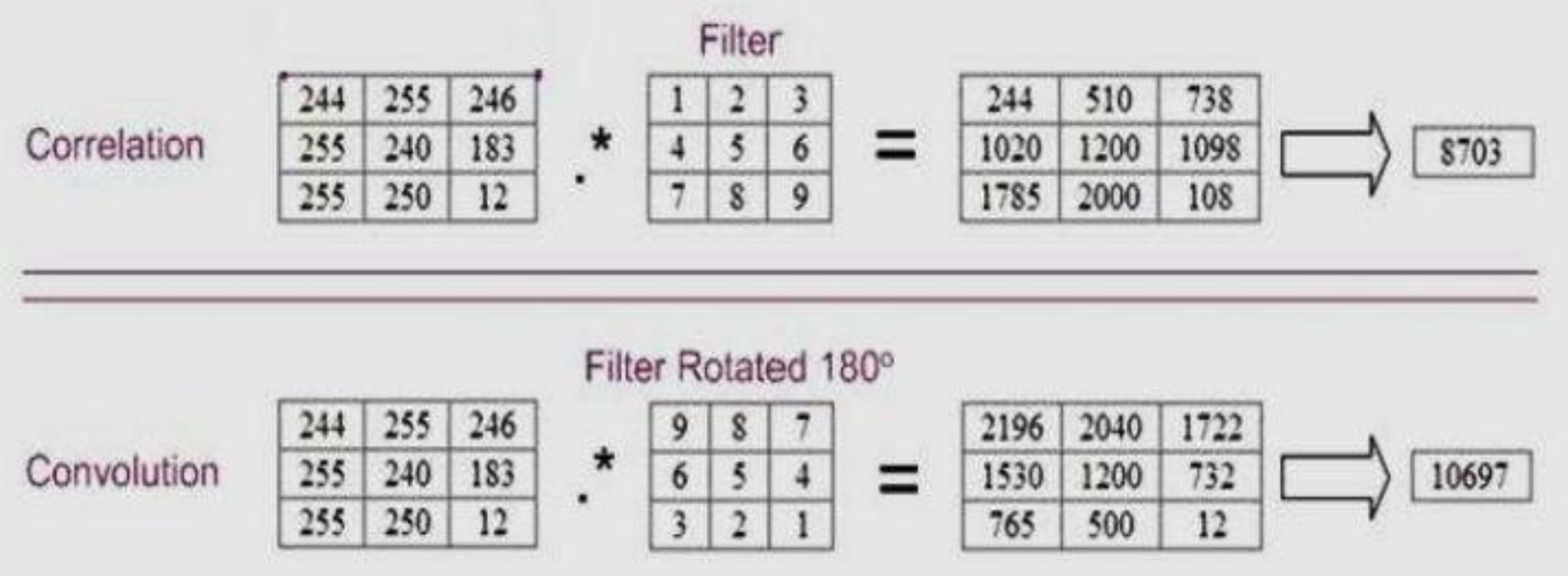
7	8	9
4	5	6
1	2	3

9	8	7
6	5	4
3	2	1

For symmetric masks, convolution is equivalent to correlation.

0	1	0
1	6	1
0	1	0

Convolution Example



Filter

Correlation

$$\begin{bmatrix} 244 & 255 & 246 \\ 255 & 240 & 183 \\ 255 & 250 & 12 \end{bmatrix} .* \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 244 & 510 & 738 \\ 1020 & 1200 & 1098 \\ 1785 & 2000 & 108 \end{bmatrix} \Rightarrow \boxed{8703}$$

Filter Rotated 180°

Convolution

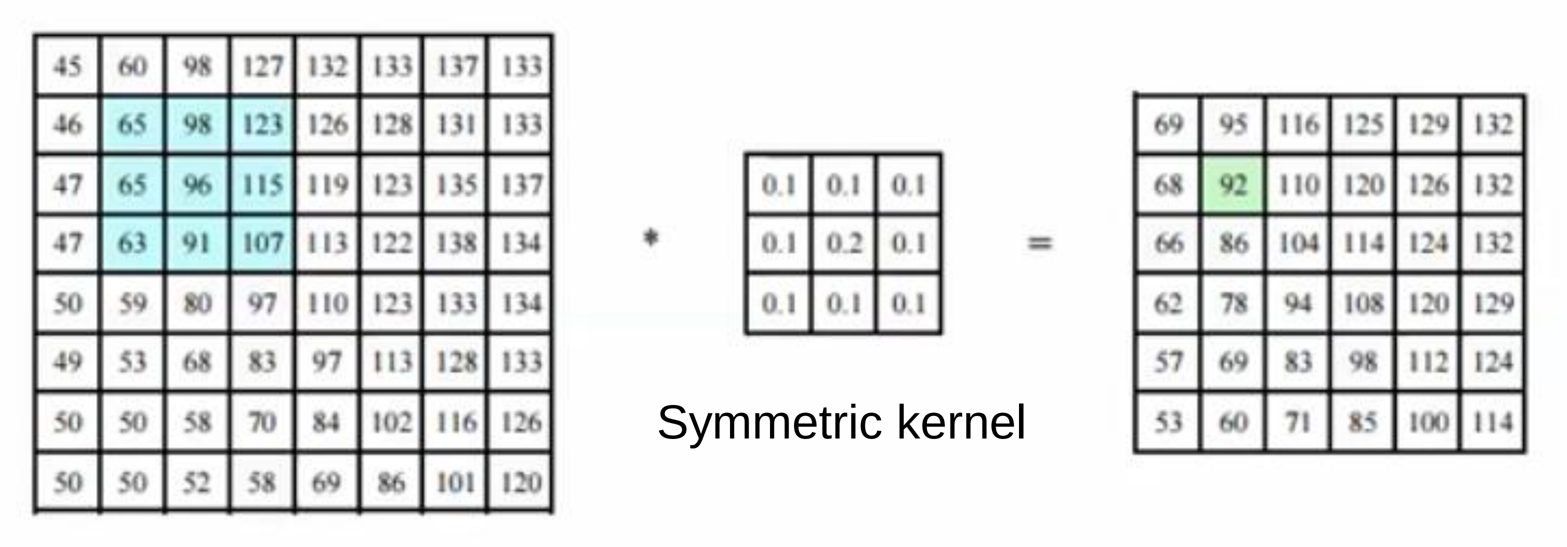
$$\begin{bmatrix} 244 & 255 & 246 \\ 255 & 240 & 183 \\ 255 & 250 & 12 \end{bmatrix} .* \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 2196 & 2040 & 1722 \\ 1530 & 1200 & 732 \\ 765 & 500 & 12 \end{bmatrix} \Rightarrow \boxed{10697}$$

244	255	246
255	240	183
255	250	12

→

	10697	

Correlation/Convolution Example



45	60	98	127	132	133	137	133
46	65	98	123	126	128	131	133
47	65	96	115	119	123	135	137
47	63	91	107	113	122	138	134
50	59	80	97	110	123	133	134
49	53	68	83	97	113	128	133
50	50	58	70	84	102	116	126
50	50	52	58	69	86	101	120

*

0.1	0.1	0.1
0.1	0.2	0.1
0.1	0.1	0.1

Symmetric kernel

=

69	95	116	125	129	132
68	92	110	120	126	132
66	86	104	114	124	132
62	78	94	108	120	129
57	69	83	98	112	124
53	60	71	85	100	114

Original image

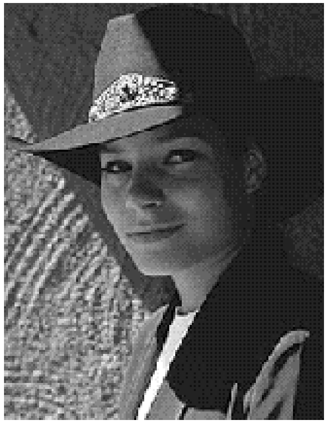
Output

Preprocessing - Image Noise Remover

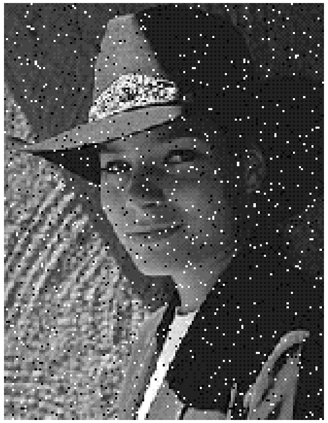
- Image Noise Remover
  - Remove unwanted signal in the image
- Importance of Image Noise Remover:
  - To recover from the Image noise that might caused by different intrinsic (i.e., sensor) and extrinsic (i.e., environment) conditions which are often not possible to avoid in practical situations.
  - To ensure the smoothness and the best performance of the later processing steps.
  - To eliminate unintended information during feature extraction.

Image noise

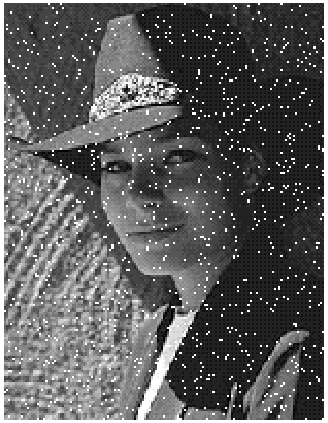
- Image processing is useful for noise reduction...
- Common types of noise:
  - **Salt and pepper noise:** contains random occurrences of black and white pixels
  - **Impulse noise:** contains random occurrences of white pixels
  - **Gaussian noise:** variations in intensity drawn from a Gaussian normal distribution



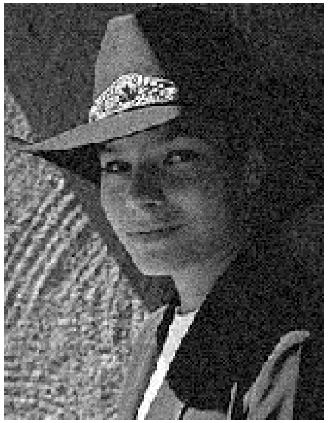
Original



Salt and pepper noise

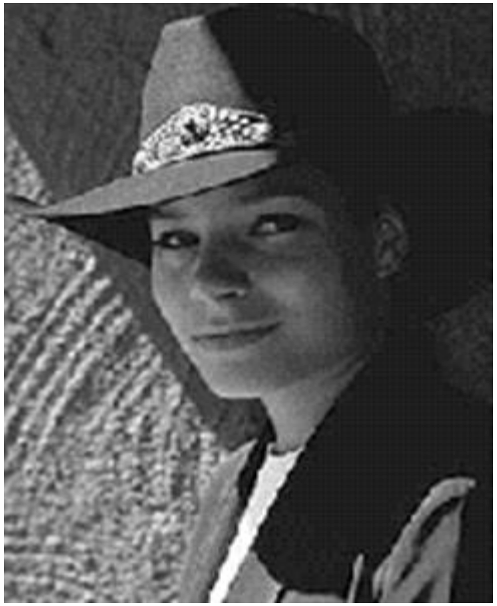


Impulse noise

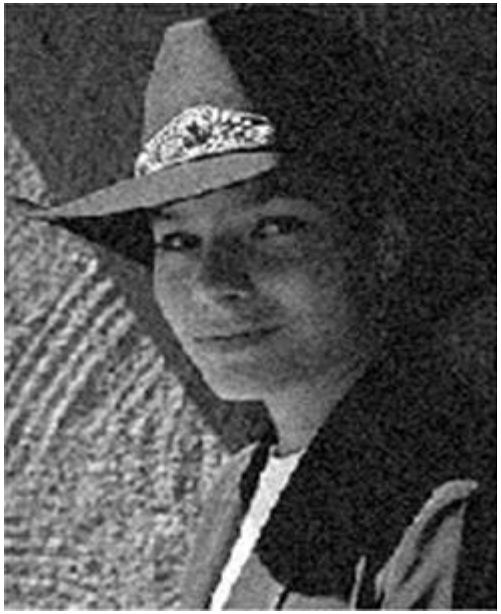


Gaussian noise

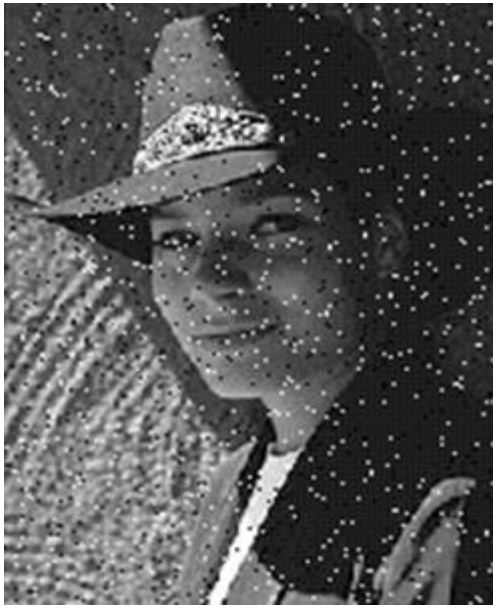
Image noise



Original image



White Gaussian noise



Salt and pepper noise

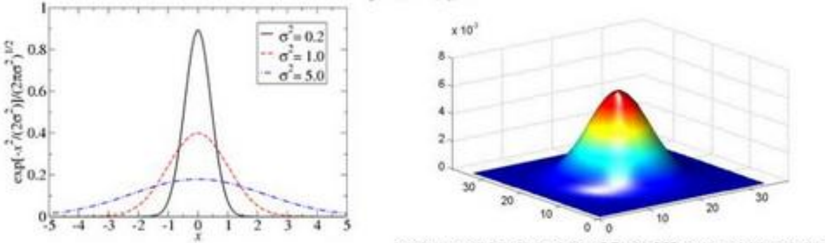
(each pixel has some chance of being switched to zero or one)

Can be recovered by filtering with different filters such as Gaussian filter, Median filter, and average filter.

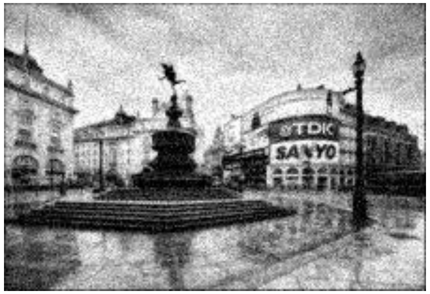
Gaussian noise – Gaussian Filtering

Filter with Gaussian Filter with different standard deviation value.

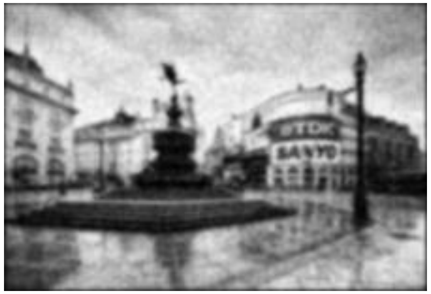
- 1D Gaussian filter: $f(x) = e^{-\frac{x^2}{2\sigma^2}}$
- 2D Gaussian filter: $f(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$



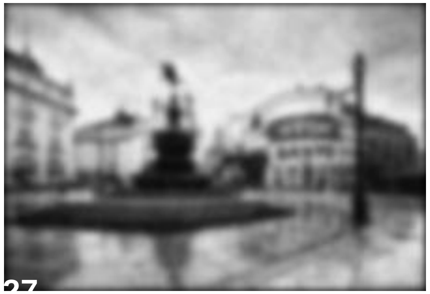
http://www-mmdb.iai.uni-bonn.de/lehre/BIT/ss03_DSP_Vorlesung/matlab_demos/



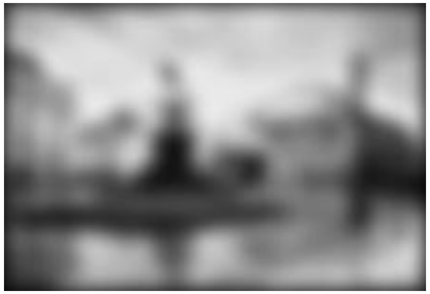
$F[x,y] + \mathcal{N}(0,5\%)$



σ = 1 pixel



σ = 2 pixels

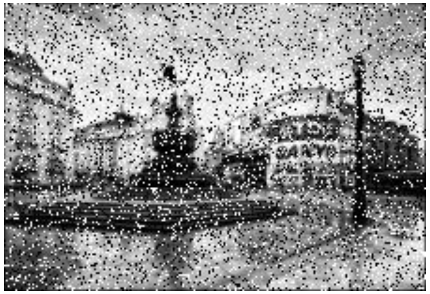


σ = 5 pixels

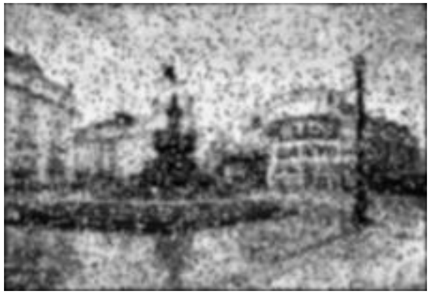
Smoothing with larger standard deviations suppresses noise, but also blurs the image

Salt & pepper noise – Gaussian Filtering

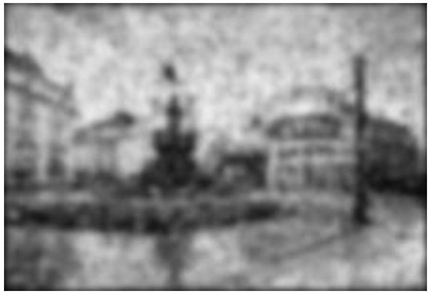
Filter with Gaussian Filter with different standard deviation value.



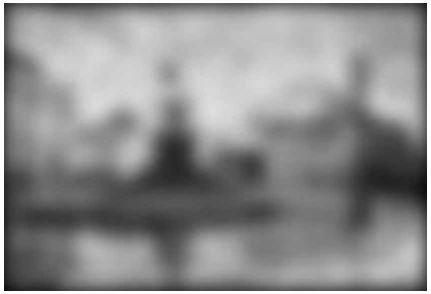
p = 10%



σ = 1 pixel



σ = 2 pixels

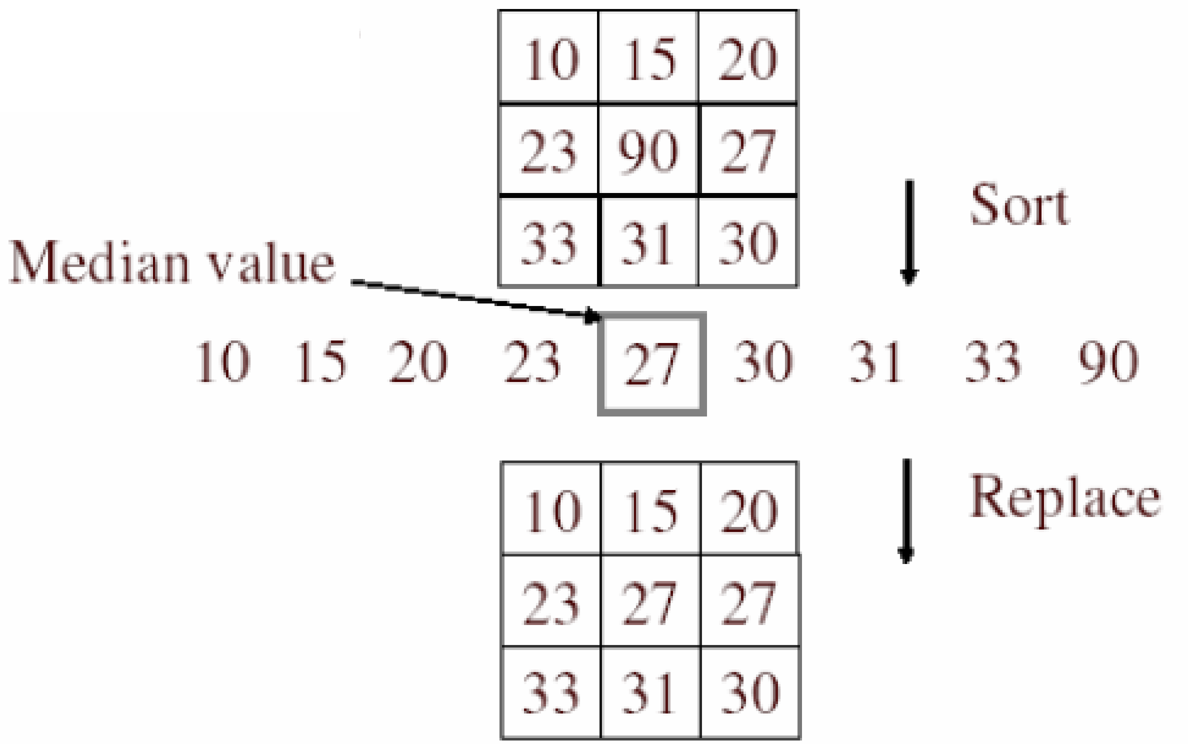


σ = 5 pixels

- What’s wrong with the results?

Alternative idea: Median filtering

- A **median filter** operates over a window by selecting the median intensity in the window



10	15	20
23	90	27
33	31	30

Sort

Median value

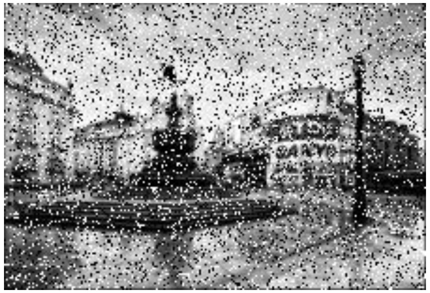
10 15 20 23 27 30 31 33 90

Replace

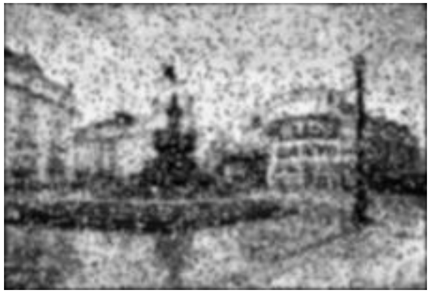
10	15	20
23	27	27
33	31	30

Source: K. Grauman

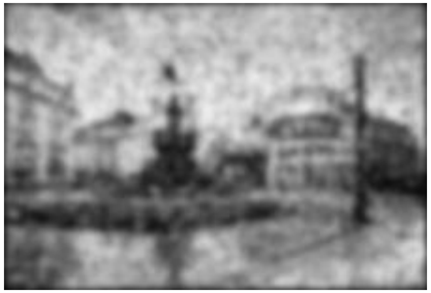
Salt & pepper noise – median filtering



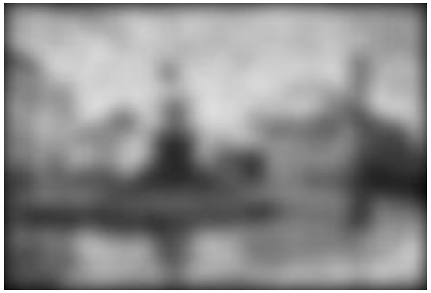
p = 10%



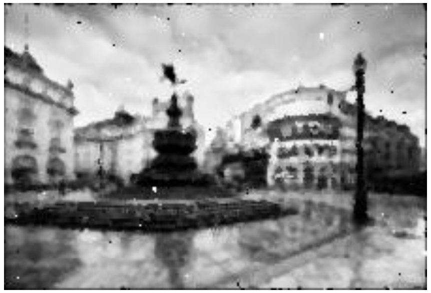
σ = 1 pixel



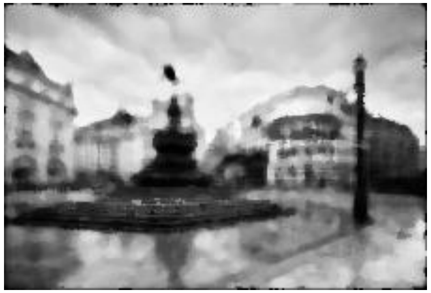
σ = 2 pixels



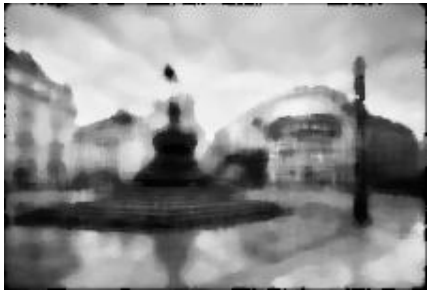
σ = 5 pixels



3x3 window

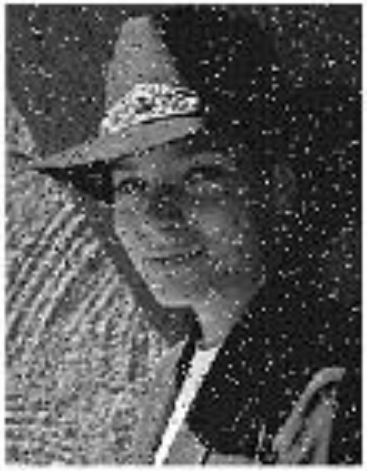


5x5 window

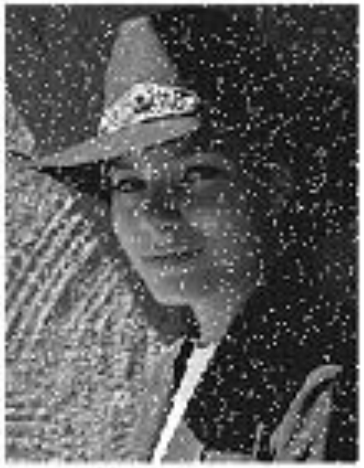


7x7 window

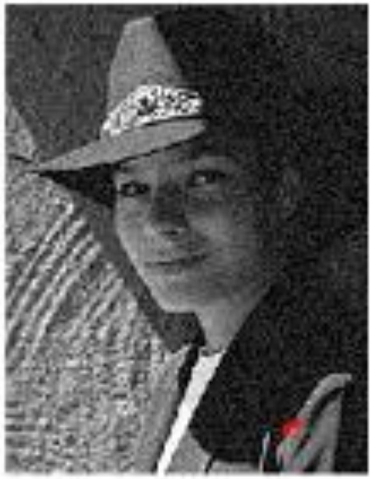
Common types of noise



Salt and pepper noise

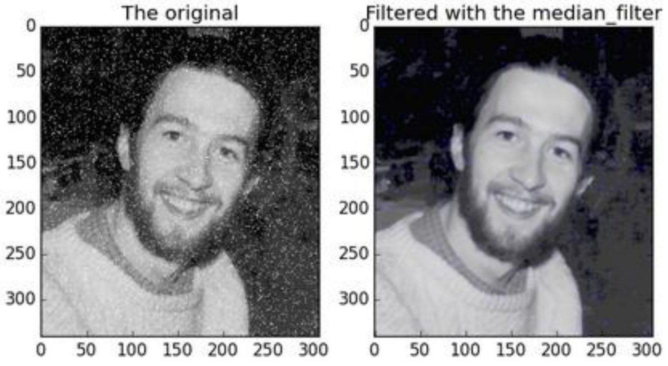


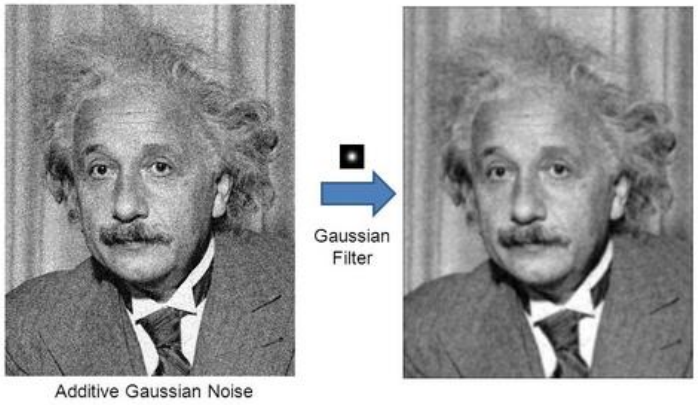
Impulse noise

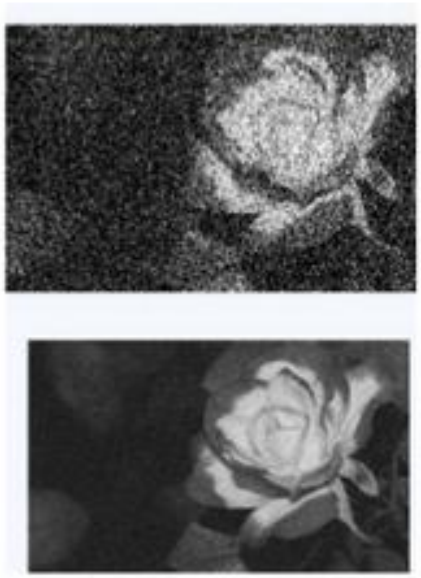


Gaussian noise

Filtering methods



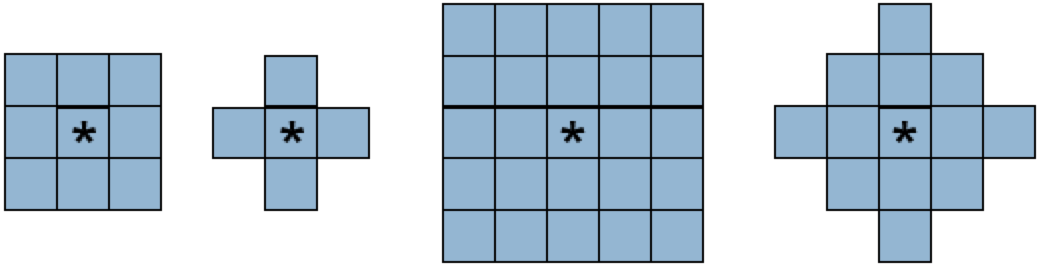




Mean Filter

Preprocessing - Morphological operation

- To simplify the objects by
  - Filling in small holes
  - Eliminating small protrusions from their boundaries
- Boundary pixels
  - Object pixels that have background neighbors
    - Various definition of neighbor
- Support by structuring element



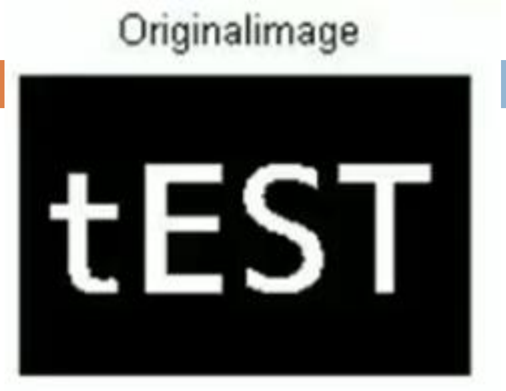
Morphology Operations

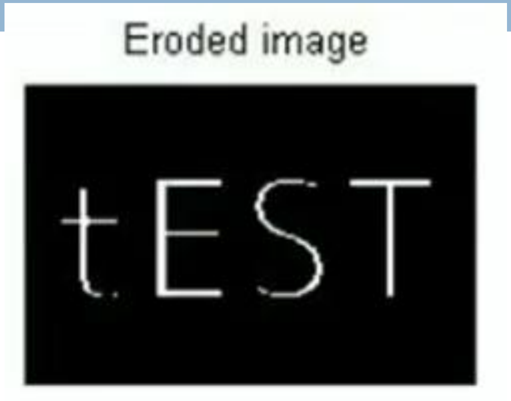
- **Erosion**
  - Elimination of boundary pixels from objects in binary images
  - R’ = $(R \ominus A)$
  - Making objects smaller, also called shrinking
- **Dilation**
  - Each background pixel that has a neighbor in the object is relabeled as an object pixel
  - R’ = $(R \oplus A)$
  - Making object bigger, also called growing

Morphology Operations

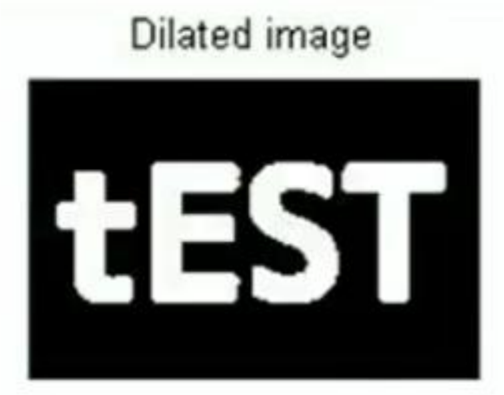
- **Opening**
  - A single erosion followed by a single dilation by the same operator
  - $R' = (R \ominus A) \oplus A$
- **Closing**
  - A single dilation followed by a single erosion by the same operator
  - $R' = (R \oplus A) \ominus A$

Morphological Operations





Erosion



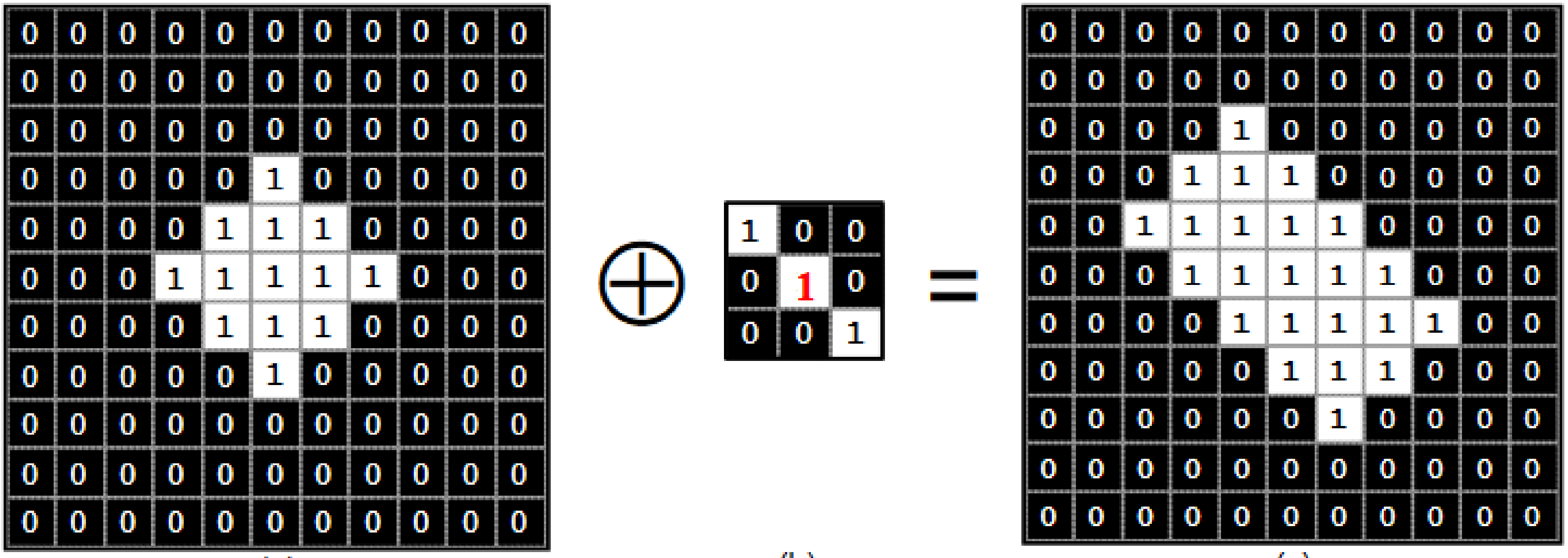
Dilation

Closing
(Dilation-> Closing)

Opening
(Erosion -> Dilation)

Dilation - Example

- □ R’ = R ⊕ A



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

(a)

⊕

1	0	0
0	1	0
0	0	1

(b)

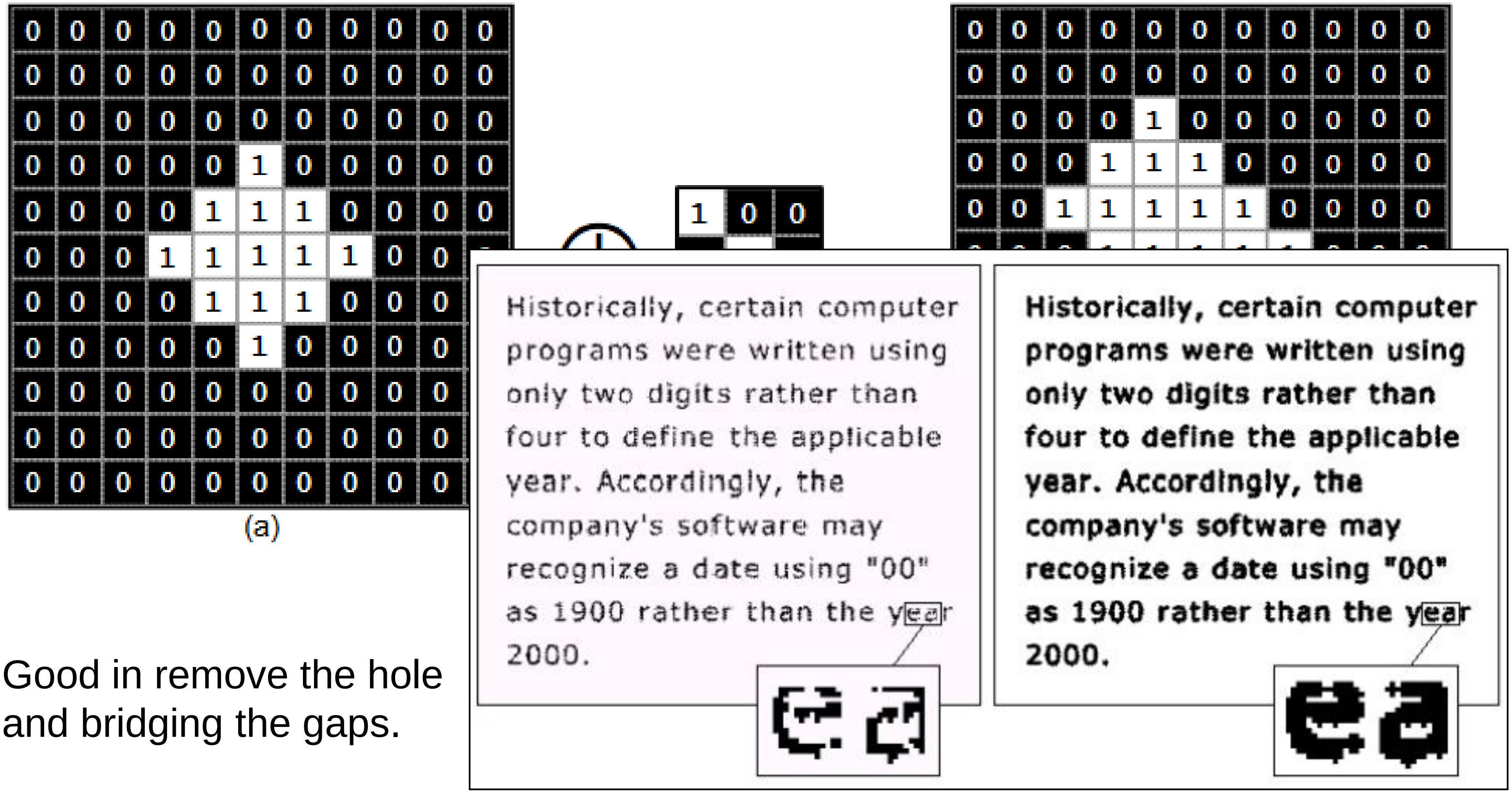
=

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

(c)

Dilation - Example

- R’ = R ⊕ A



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

(a)

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

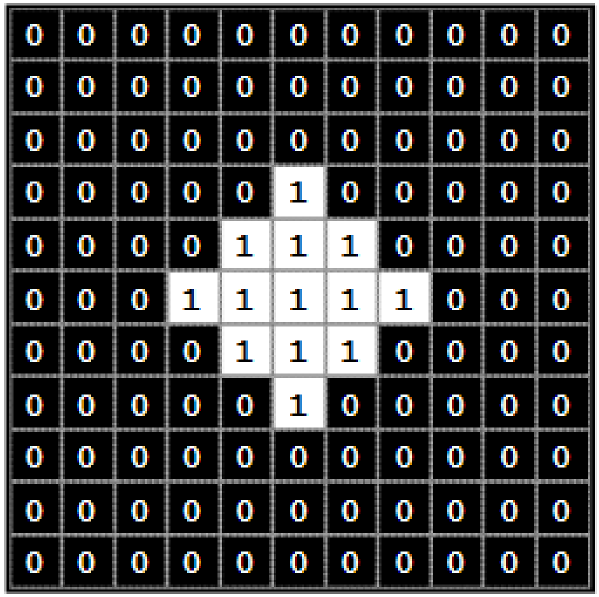
Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

Good in remove the hole and bridging the gaps.

Erosion - Example

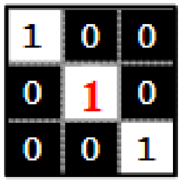
- R’ = R Θ A



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

(a)

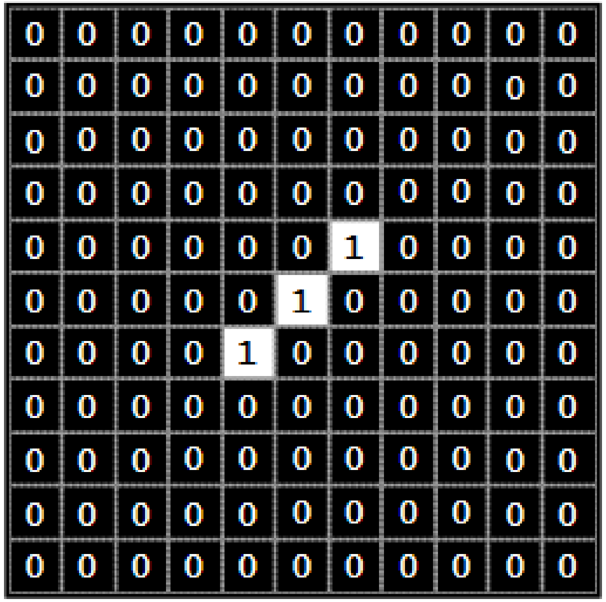
Θ



1	0	0
0	1	0
0	0	1

(b)

=

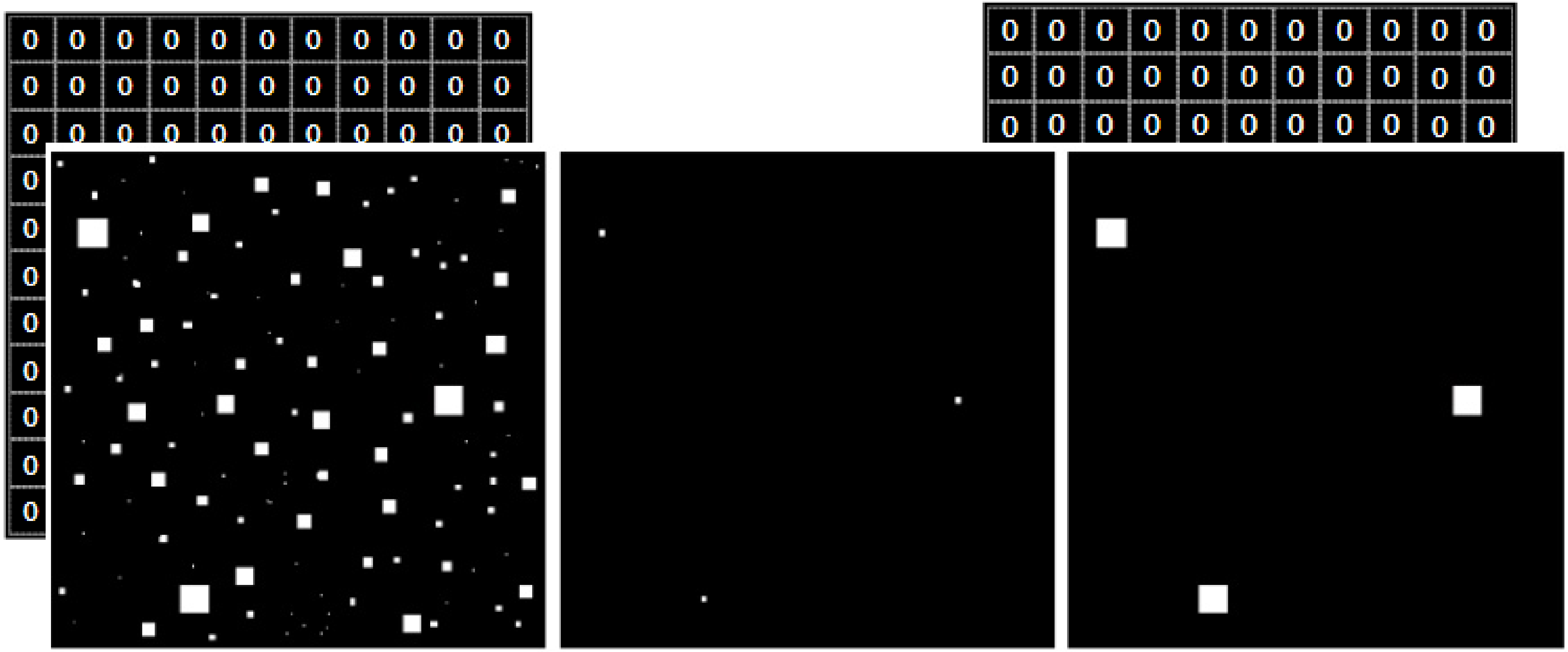


0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

(c)

Erosion - Example

- R’ = R Θ A



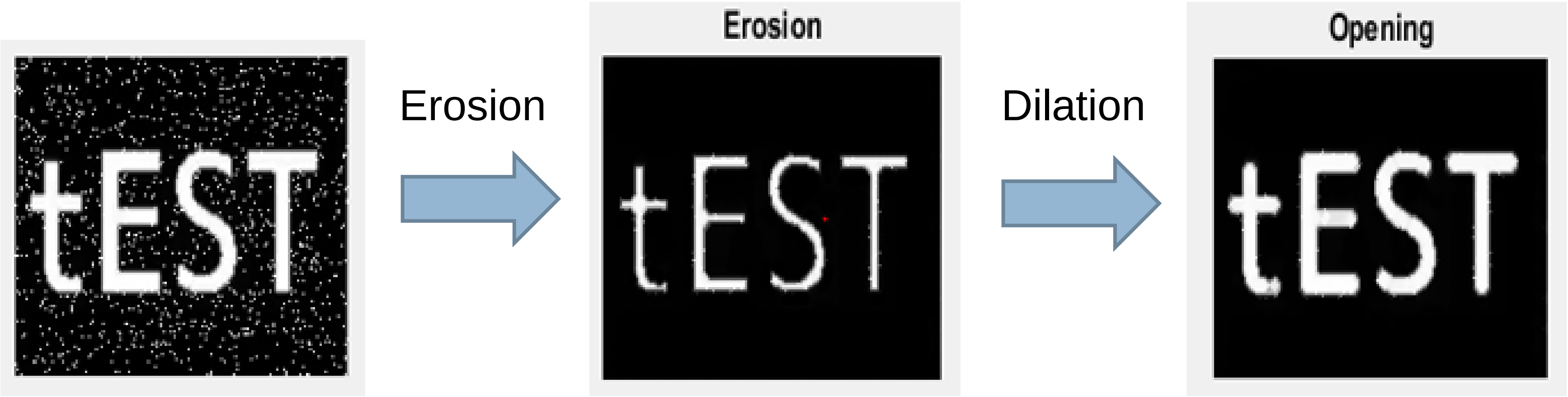
Original

Erosion

Dilation

Good in removing irrelevant details.

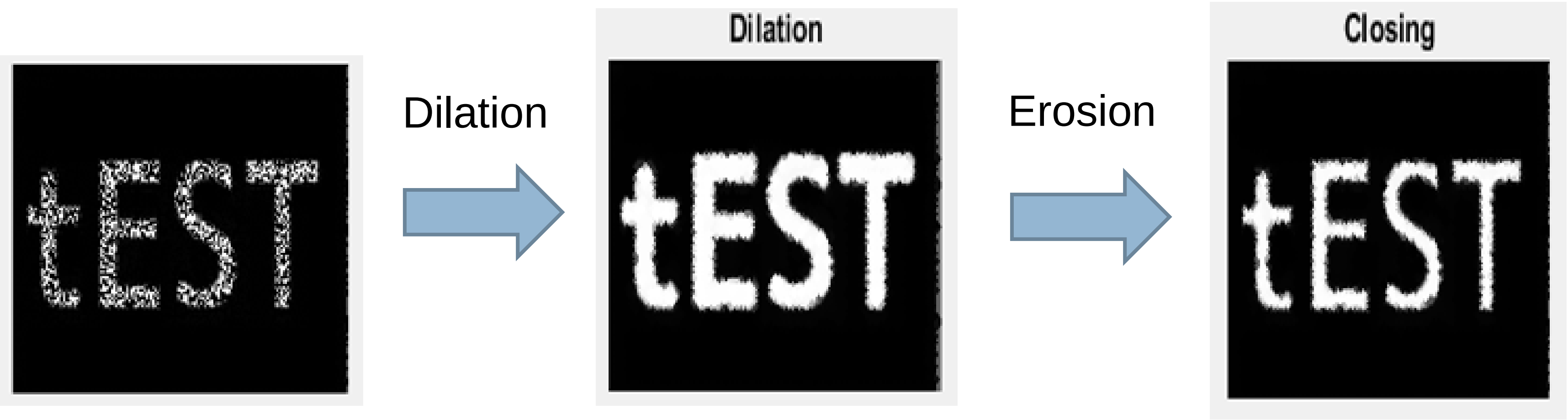
Opening



Erosion

Dilation

Closing



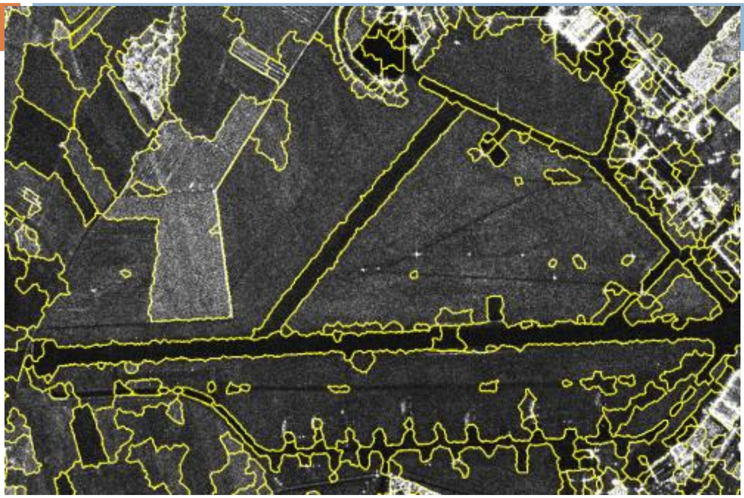
Dilation

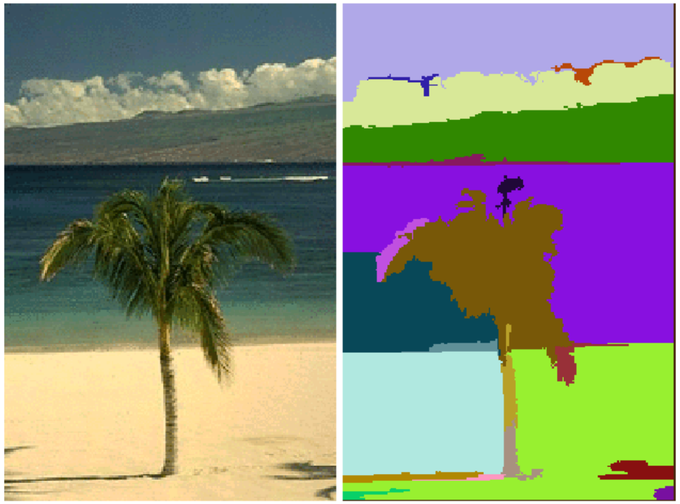
Erosion

Segmentation

- Image segmentation is the process of partitioning a digital image into a set of non-overlapping regions that together cover the entire image.
- All pixels in a region are similar with respect to some characteristics, such as color, intensity, or texture.
- Adjacent regions are significantly different with respect to the same characteristics.
- Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images.

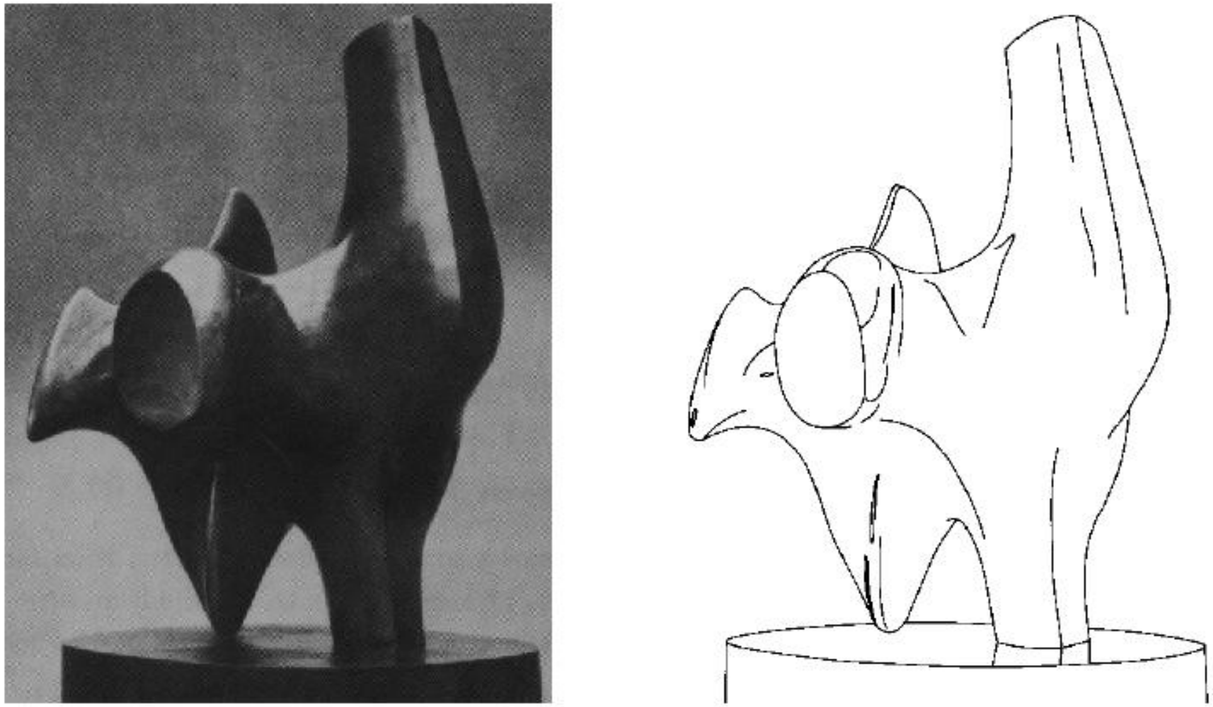
Segmentation (Examples)





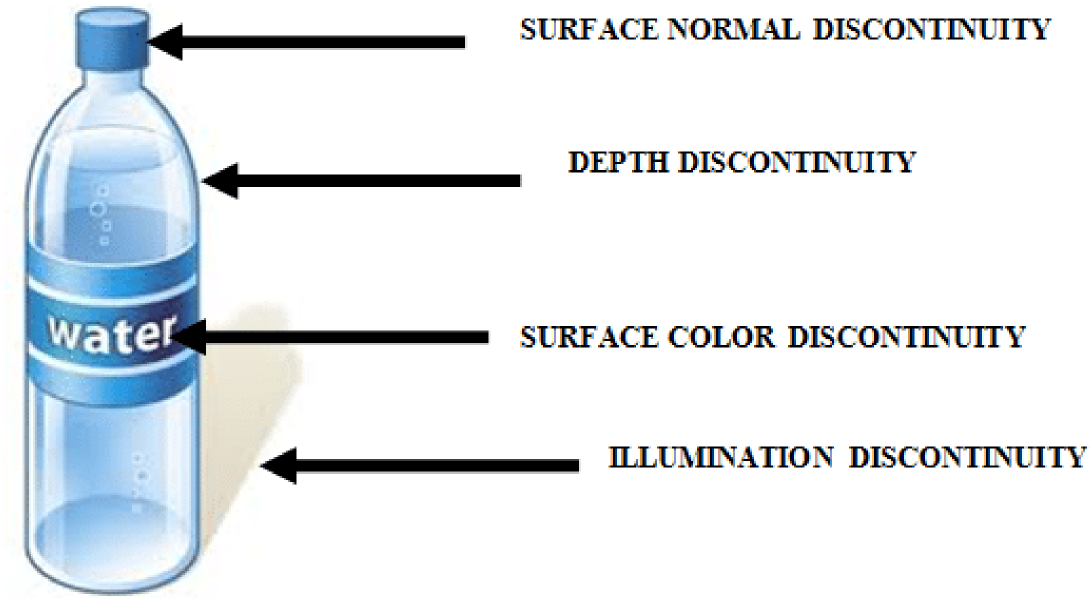
□ The output of segmentation stage usually is raw pixel data, constituting either the boundary of a region, or all the points in the region itself.

Edge detection



- Convert a 2D image into a set of curves
  - Extracts salient features of the scene
  - More compact than pixels

Edge detection - Origin of Edges

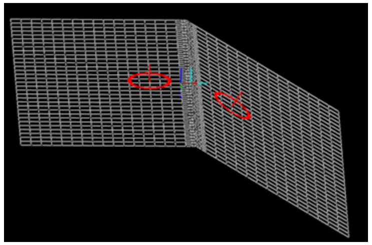


SURFACE NORMAL DISCONTINUITY

DEPTH DISCONTINUITY

SURFACE COLOR DISCONTINUITY

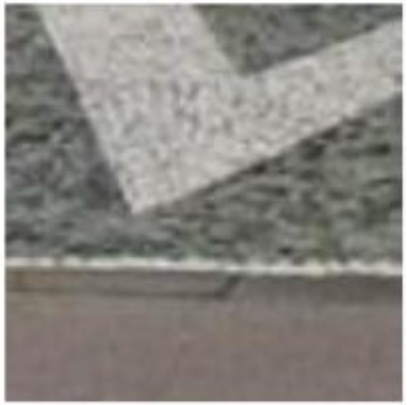
ILLUMINATION DISCONTINUITY



- □ Edges are caused by a variety of factors
- □ An edge is a place of rapid change in the image intensity function

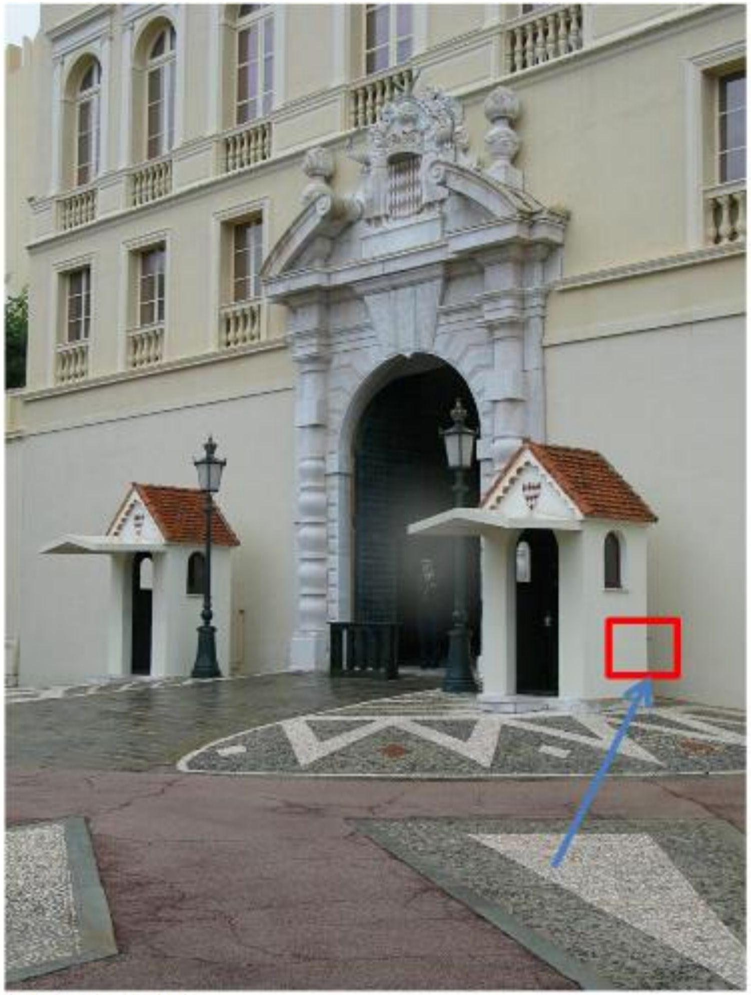
Edge detection - Characterizing edges

Source: Hays, Brown

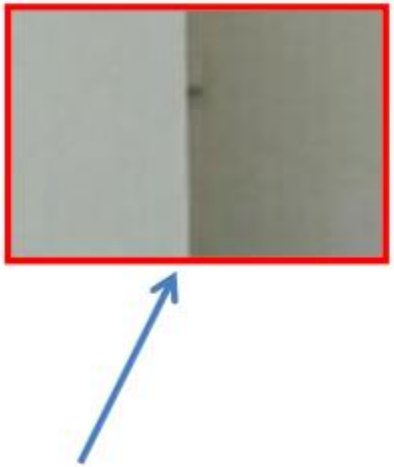


Source: D. Hoiem

Edge detection - Characterizing edges

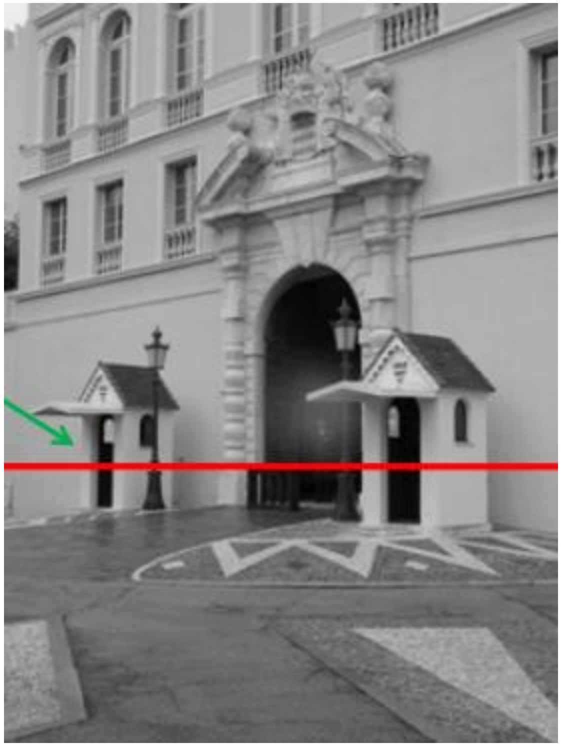


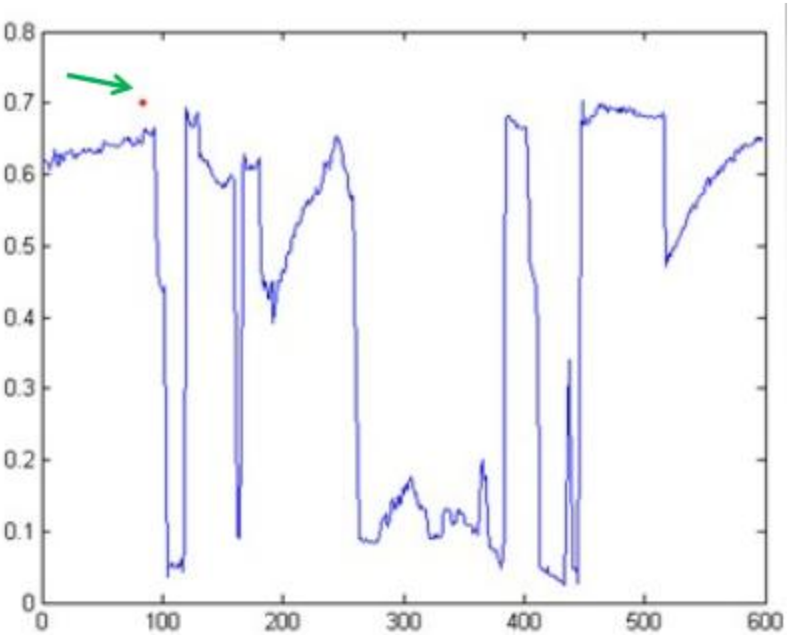
Source: Hays, Brown



Source: D. Hoiem

Edge detection - Characterizing edges





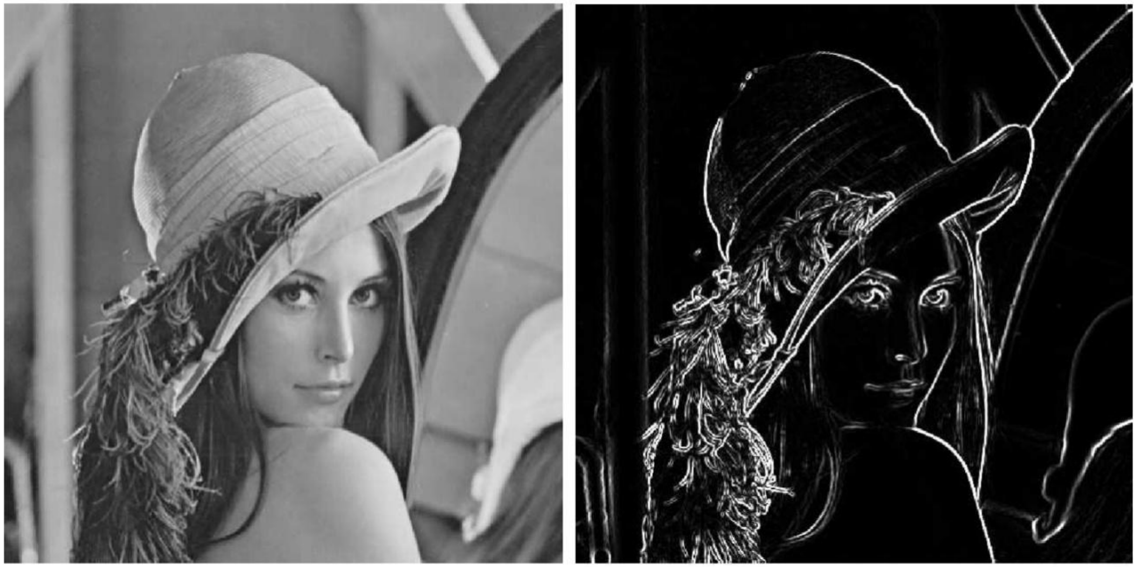
Edge detection - Characterizing edges

- □ So, at each point convolve with 1st derivative filter / mask

$$G_x = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \qquad G_y = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

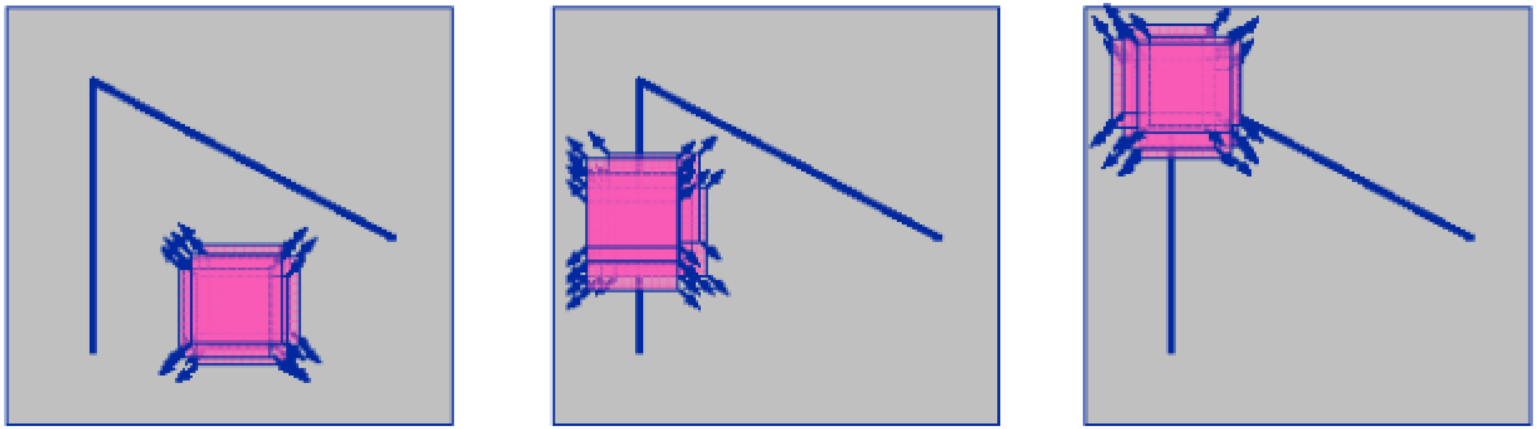
Edge detection algorithms:

1) Canny
2) Prewitts
3) Sobel



Corner detection

- Shifting the window in any direction should yield a large change in appearance.



"flat" region: no change in all directions

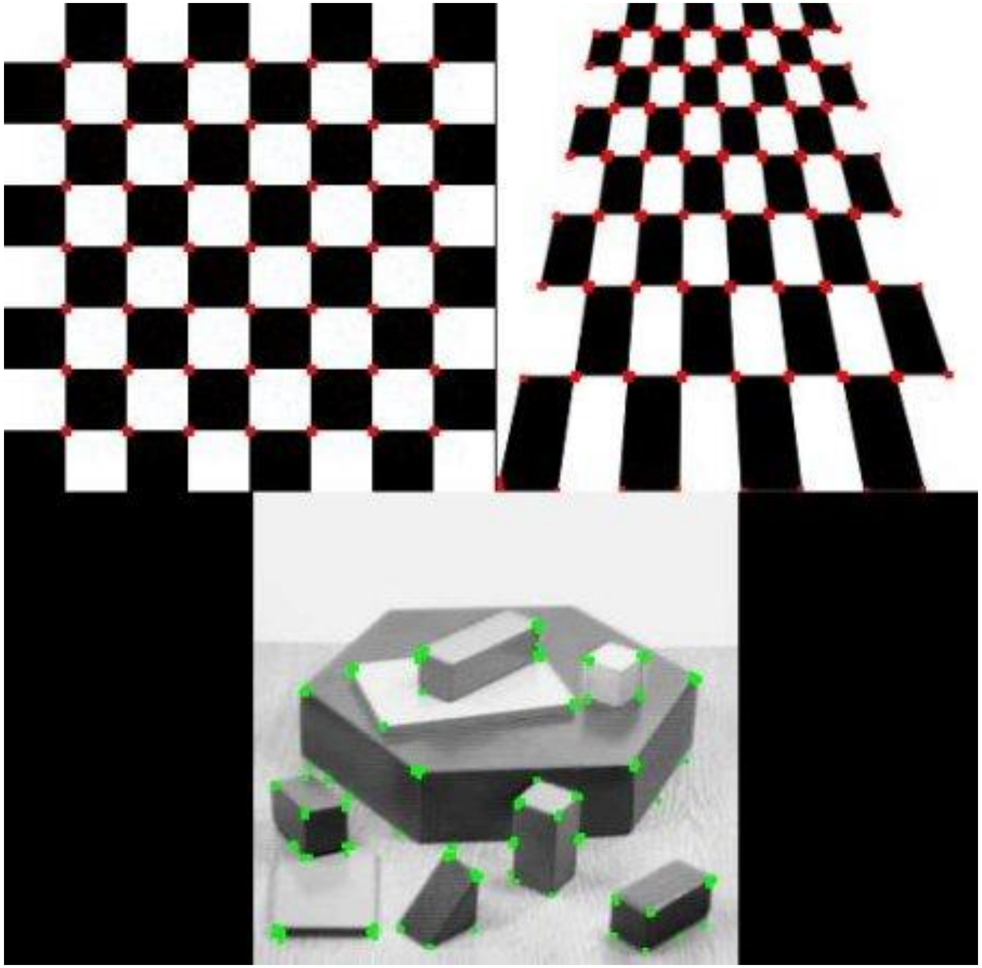
"edge": no change along the edge direction

"corner": significant change in all directions

Of course, Harris corner detector gives a mathematical approach for determining which case holds.

Corner detection

- At each point convolve with

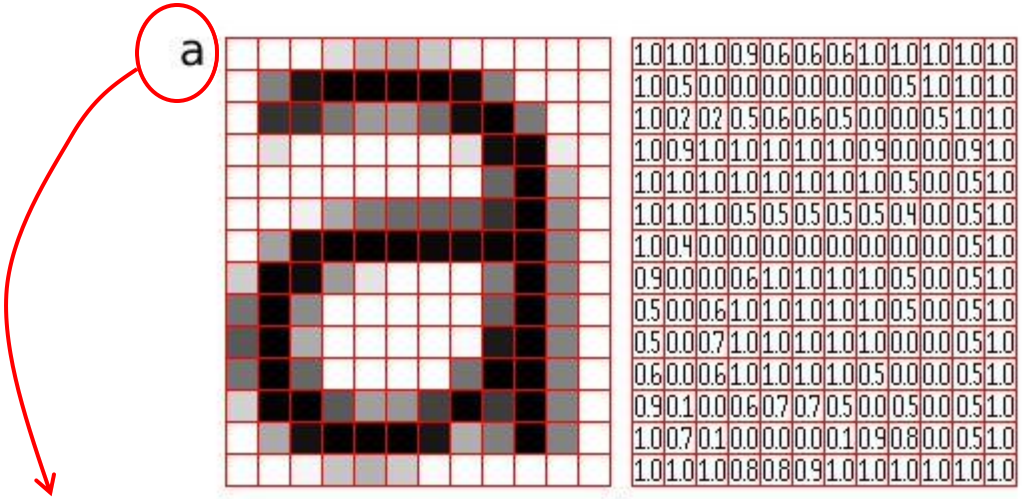


Representation and Description

- Representation is to transform raw data into a form suitable for subsequent computer processing.
- Description (feature selection) extracting features that result in some quantitative information, to differentiate one class/object from another.
- E.g. descriptor for size and shape of boat will differentiate it from ship

Representation & Description

Representation of letter “a”



1.0	1.0	1.0	0.9	0.6	0.6	0.6	1.0	1.0	1.0	1.0	1.0
1.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0	1.0	1.0
1.0	0.2	0.2	0.5	0.6	0.6	0.5	0.0	0.0	0.5	1.0	1.0
1.0	0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.0	0.0	0.9	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
1.0	1.0	1.0	0.5	0.5	0.5	0.5	0.5	0.4	0.0	0.5	1.0
1.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0
0.9	0.0	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.6	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.7	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.5	1.0
0.6	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.0	0.5	1.0
0.9	0.1	0.0	0.6	0.7	0.7	0.5	0.0	0.5	0.0	0.5	1.0
1.0	0.7	0.1	0.0	0.0	0.0	0.1	0.9	0.8	0.0	0.5	1.0
1.0	1.0	1.0	0.8	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0

A rasterized form of the letter 'a' magnified 16 times using pixel doubling

Descriptor in vector
= [1.0 1.0 0.9 0.6 0.6 1.0 1.0 1.0 1.0 1.0 1.0 1.0 0.5 …. 1.0]

Vectors

- □ Represented by bold lower case letters, such as **x**, ***y***, and ***z*** and take the form:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \mathrm{M} \\ x_n \end{bmatrix} \quad \text{OR} \quad \mathbf{x} = [x_1, x_2, \ldots, x_\mathrm{n}]$$

- □ Where each component x_i, represents *i*th descriptor and n is the number of descriptor.

Recognition

- determining whether or not the image data contains some specific object, feature, or activity
  - **Object recognition** – to recognize one or several pre-specified objects or object classes, usually together with their 2D positions in the image or 3D poses in the scene.
  - **Identification** – an individual instance of an object is recognized. Examples include identification of a specific person's face or fingerprint
  - **Detection** – the image data are scanned for a specific condition. Examples include detection of a vehicle in an automatic road toll system.

Interpretation

- □ Interpretation involves assigning meaning to an ensemble of recognized image. E.g. a string of five numbers can be interpreted as ZIP code.

Classification

- **classification** is the problem of identifying which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known.
- Each segmented object can be classified to one of a set of meaningful classes.
- For example, an image of ocean may contain classes such as ships, small boats, water body, etc.
- Expert systems, semantic networks, neural-network-based system can perform such task quite efficiently.

Recap – Image Recognition Process

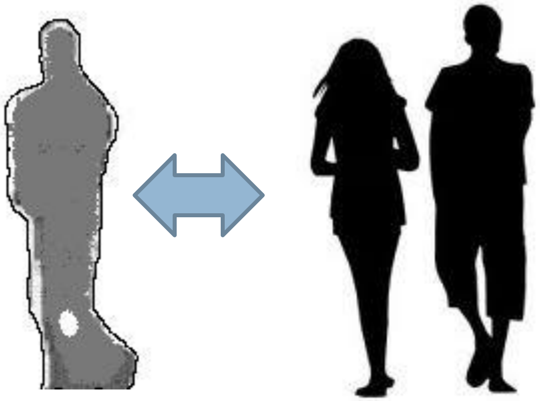
height

```
11111110000011111111
11111009999900111111
11110099999999990011
11100999999999900011
11009999999999990011
10099999999999990011
10099999999999999011
10099999999999999011
10099999999999999011
10999999999999990011
10999999999999990011
10099999999999900111
11009999999999001111
11100999999990011111
11111009999990011111
11111008888880011111
11100888888888800111
10066666666666666001|
```

0 - boundary

1 – unwanted background

0<x<1 other regions



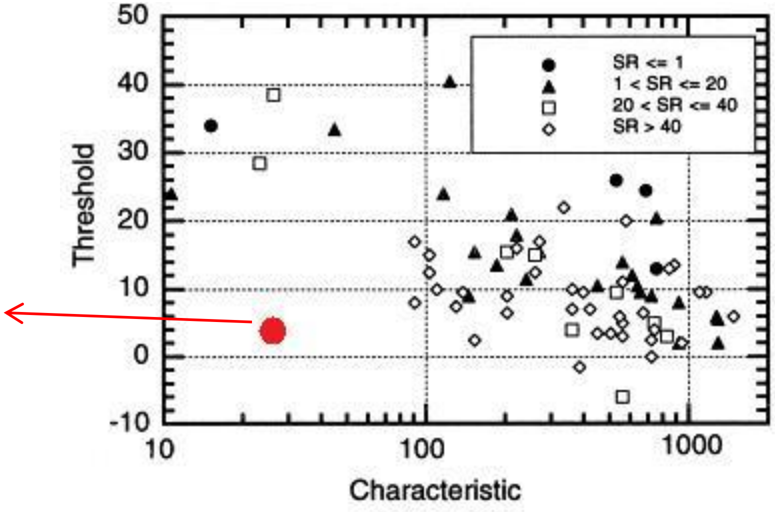
recognized as a human

Interpreted as a man

Representation & Description

Recognition & Interpretation (these usually involve classification as well)

Recap – Image Recognition Process



Abnormal activity

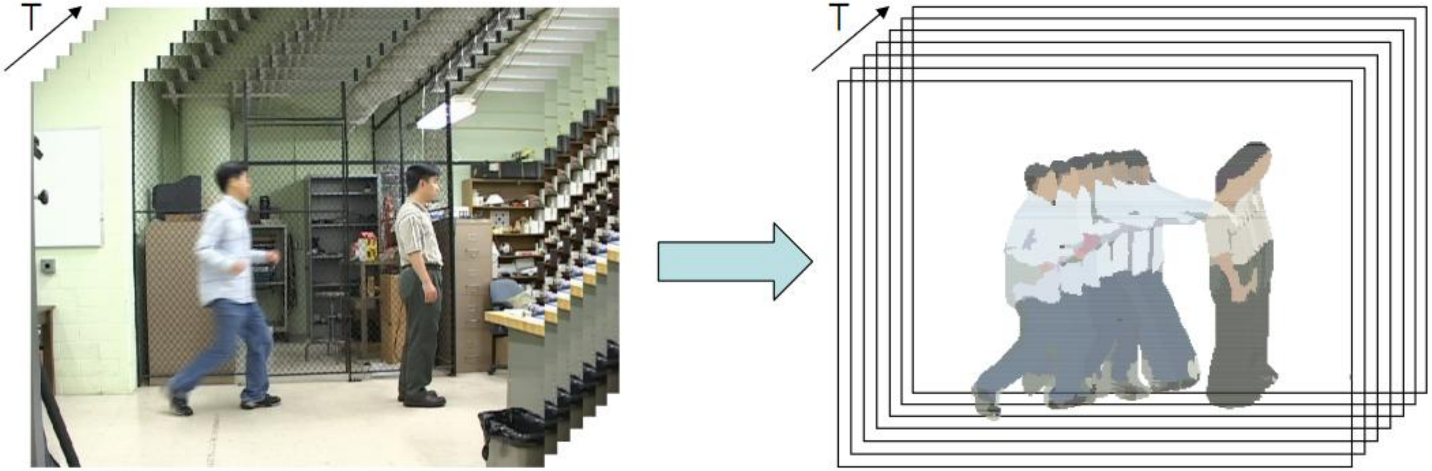
Classification – abnormality detection

Challenges of Image Processing

1. The ability to extract pertinent information from a background of irrelevant details.
2. The capability to learn from examples and to generalize this knowledge so that it will apply in new and different circumstances.
3. Ability to make inferences from incomplete information

Computer Vision

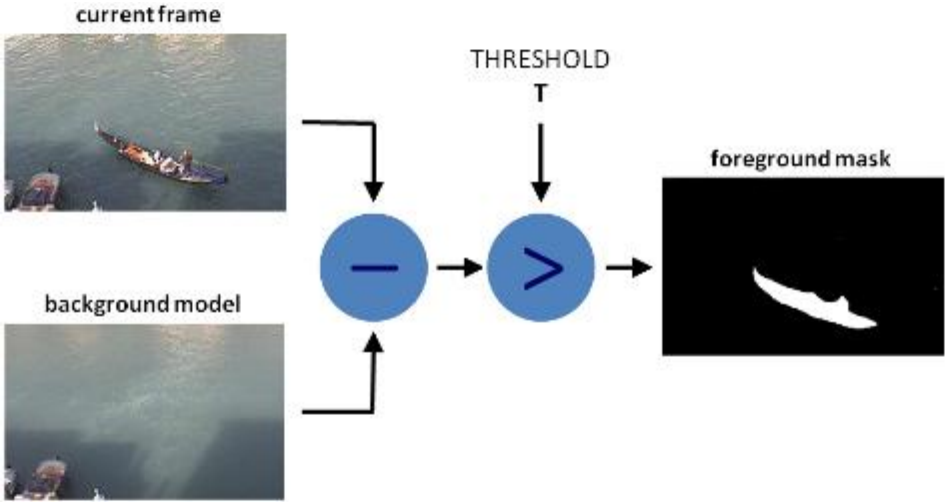
- □ Dealing with video
- □ Video is visual multimedia source that combines a sequence of images/frames to form a moving picture.
- □ Temporal information is available. A single image only provide spatial information.



[J. K. Aggarwal and M. S. Ryoo]

Computer Vision

- Background subtraction
  - also known as Foreground Detection, is a technique in the fields of image processing and computer vision wherein an image's foreground is extracted for further processing (object recognition etc.).
  - Generally an image's regions of interest are objects (humans, cars, text etc.) in its foreground.

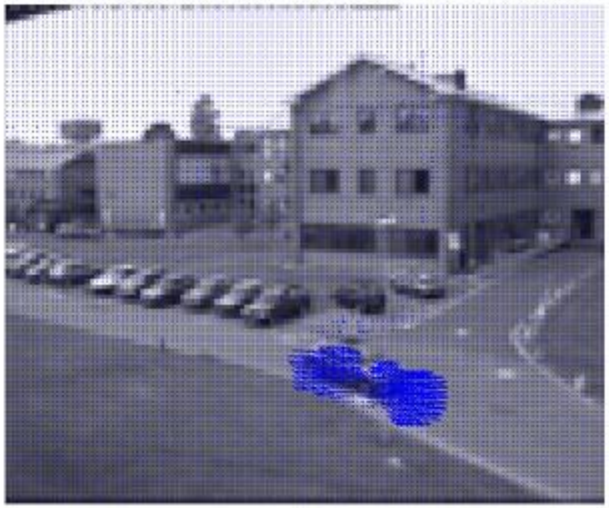


Computer Vision

- Motion tracking
  - is the process of locating a moving object (or multiple objects) over time using a camera. It has a variety of uses, some of which are: human-computer interaction, security and surveillance, video communication and compression, augmented reality, traffic control, medical imaging and video editing.

Frame t

Frame $t + dt$



Techniques:

1) Optical Flow
2) Kalman Filter
3) Particle Filter

Computer Vision

- Example of steps involved in Computer Vision:
  - Video acquisition
  - Video slicing
  - Preprocessing (E.g. background subtraction, image enhancement, etc.)
  - Feature extraction (E.g. motion features, edge, corner, etc.)
  - Perform Machine Learning (Training)
  - Perform classification / recognition / interpretation