ARTIFICIAL INTELLIGENCE

BACS2003|BACS3074|BMCS2003

CHAPTER 10 IMAGE PROCESSING AND COMPUTER VISION

OUTCOMES

1. **Introduction to Image Processing**
2. **Introduction to Computer Vision**

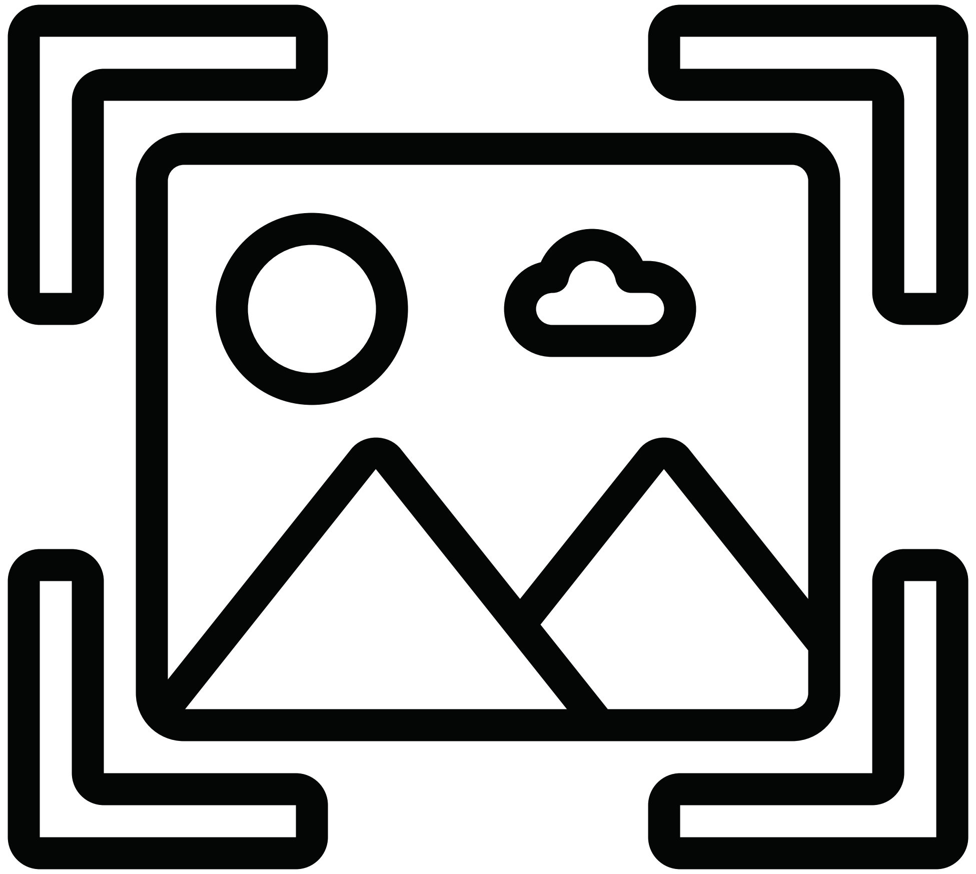


IMAGE PROCESSING

1. Image processing primarily deals with **manipulating and enhancing images to improve their visual quality or extract useful information**.
2. It involves techniques for **altering the appearance** of an image by adjusting its brightness, contrast, color balance, and other attributes.
3. Image processing techniques are often used to **remove noise, sharpen images, perform image compression, and apply various filters**.
4. The main goal of image processing is **to improve the visual quality of images** or make them more suitable for a specific application.

EXAMPLES OF IMAGE PROCESSING

1. Image denoising
2. Image sharpening
3. Color correction
4. Image compression

EXAMPLES OF IMAGE PROCESSING

1. Image denoising
2. Image sharpening
3. Color correction
4. Image compression

EXAMPLES OF IMAGE PROCESSING

1. Image denoising
2. Image sharpening
3. Color correction
4. Image compression

EXAMPLES OF IMAGE PROCESSING

1. Image denoising
2. Image sharpening
3. Color correction
4. Image compression

Original (28KB)

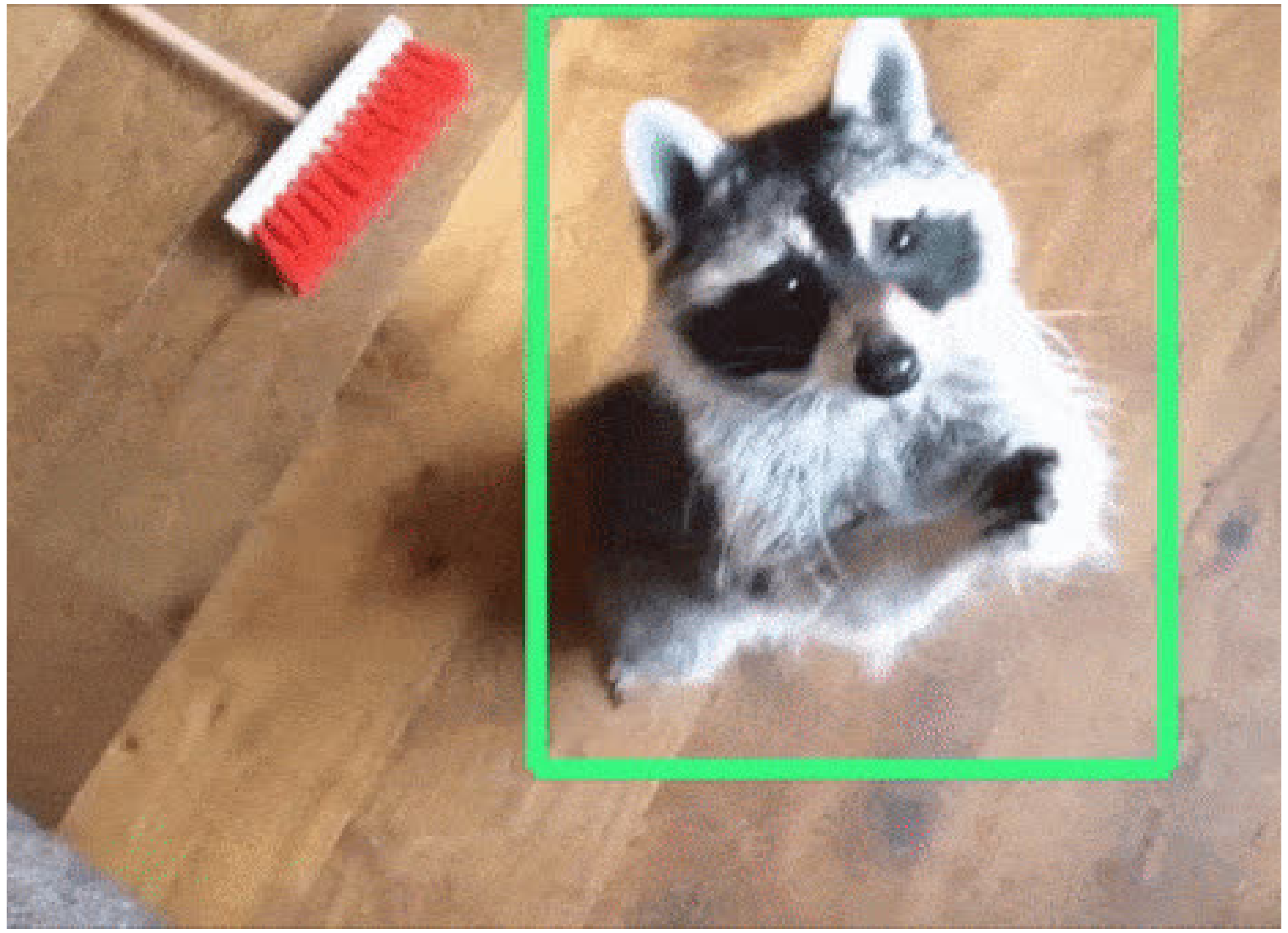
Lossy Compression (14KB, 50%)

COMPUTER VISION

1. Computer Vision is a broader field that involves **enabling computers to interpret and understand the visual world**.
2. It aims to **replicate human visual perception** using computational techniques. Computer vision algorithms analyze and interpret images or video streams to extract meaningful information from them.
3. This can include identifying objects, detecting patterns, recognizing faces, estimating depth, and even understanding the context of a scene.

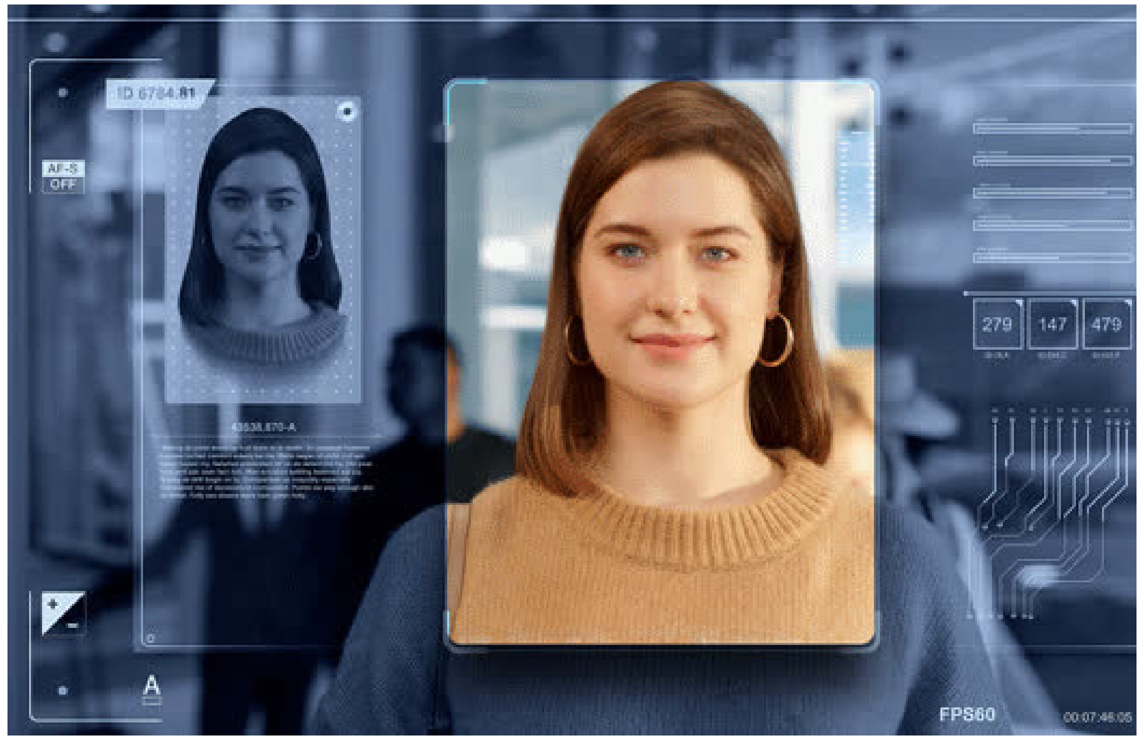
EXAMPLES OF COMPUTER VISION

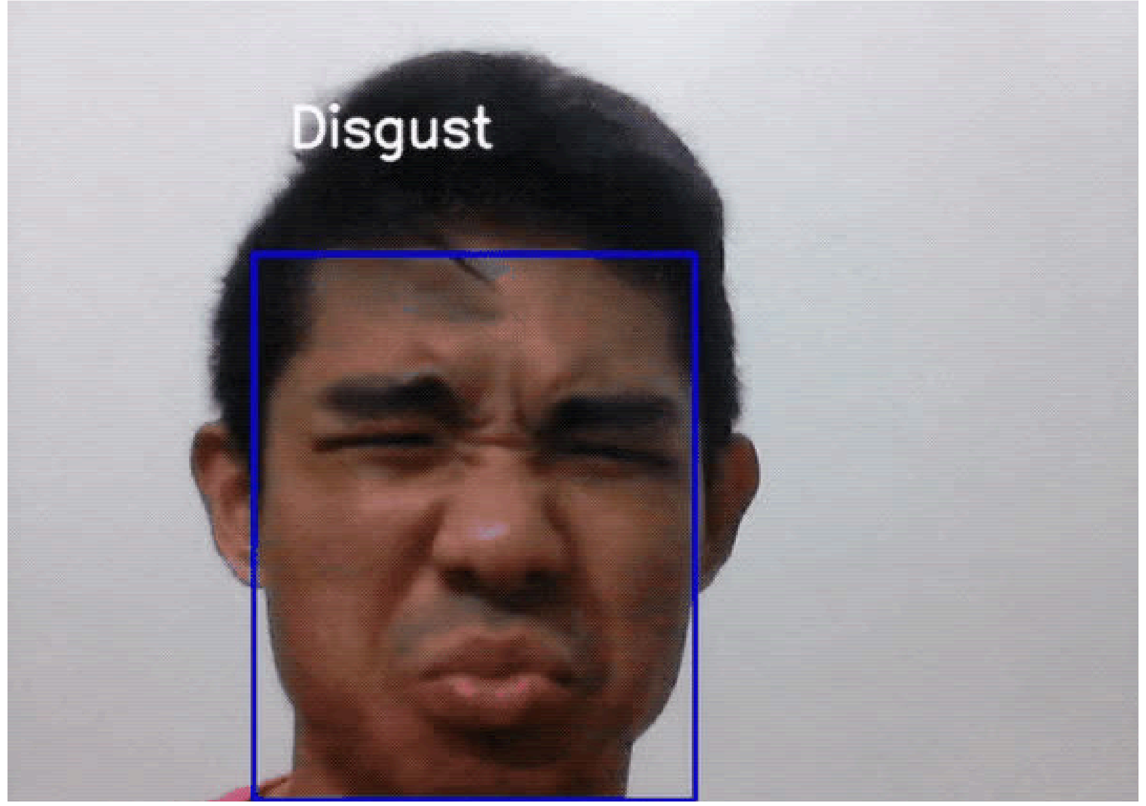
1. Object detection and recognition
2. Facial recognition
3. Optical character recognition (OCR)
4. Image segmentation
5. 3D reconstruction
6. Gesture recognition



EXAMPLES OF COMPUTER VISION

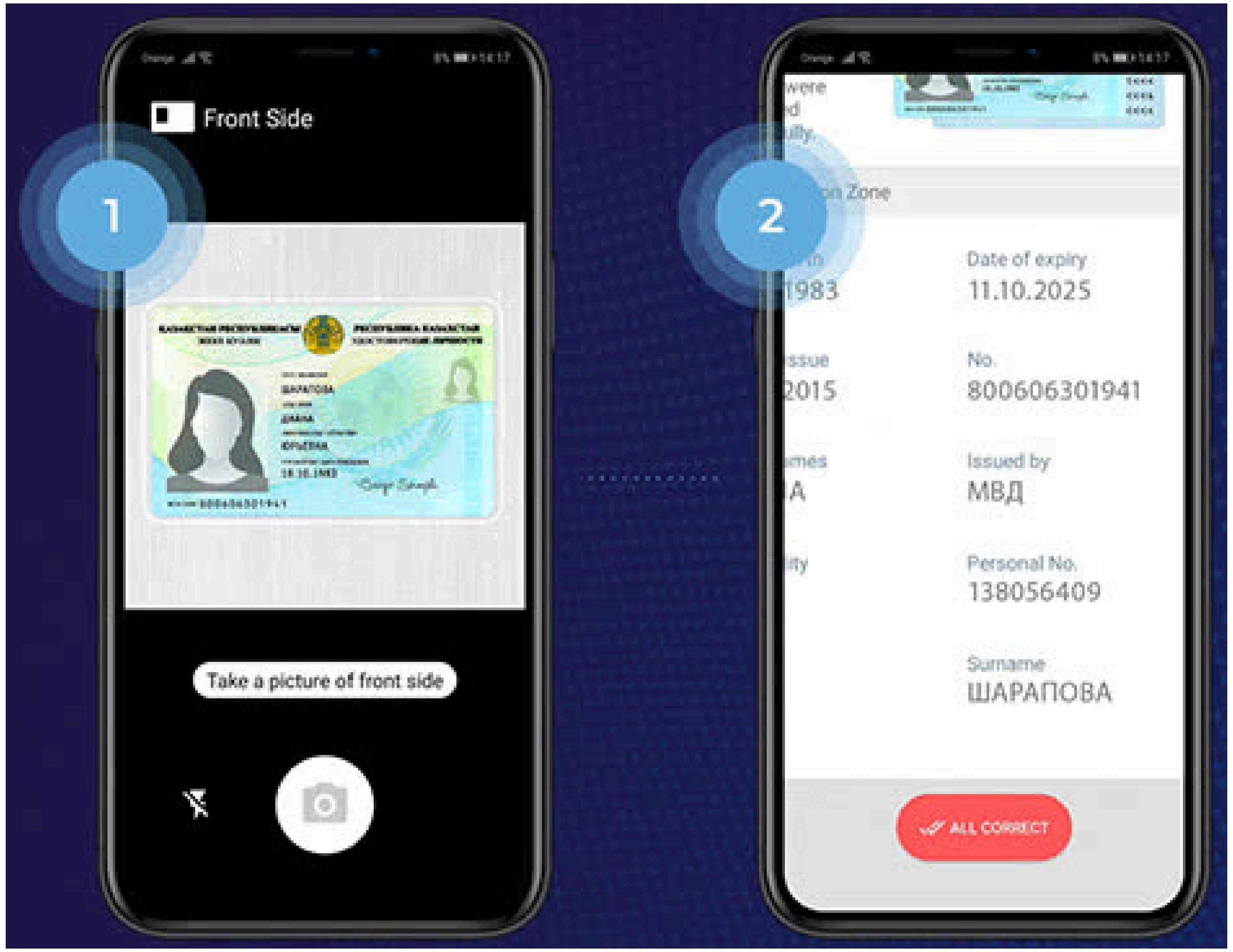
1. Object detection and recognition
2. Facial recognition
3. Optical character recognition (OCR)
4. Image segmentation
5. 3D reconstruction
6. Gesture recognition





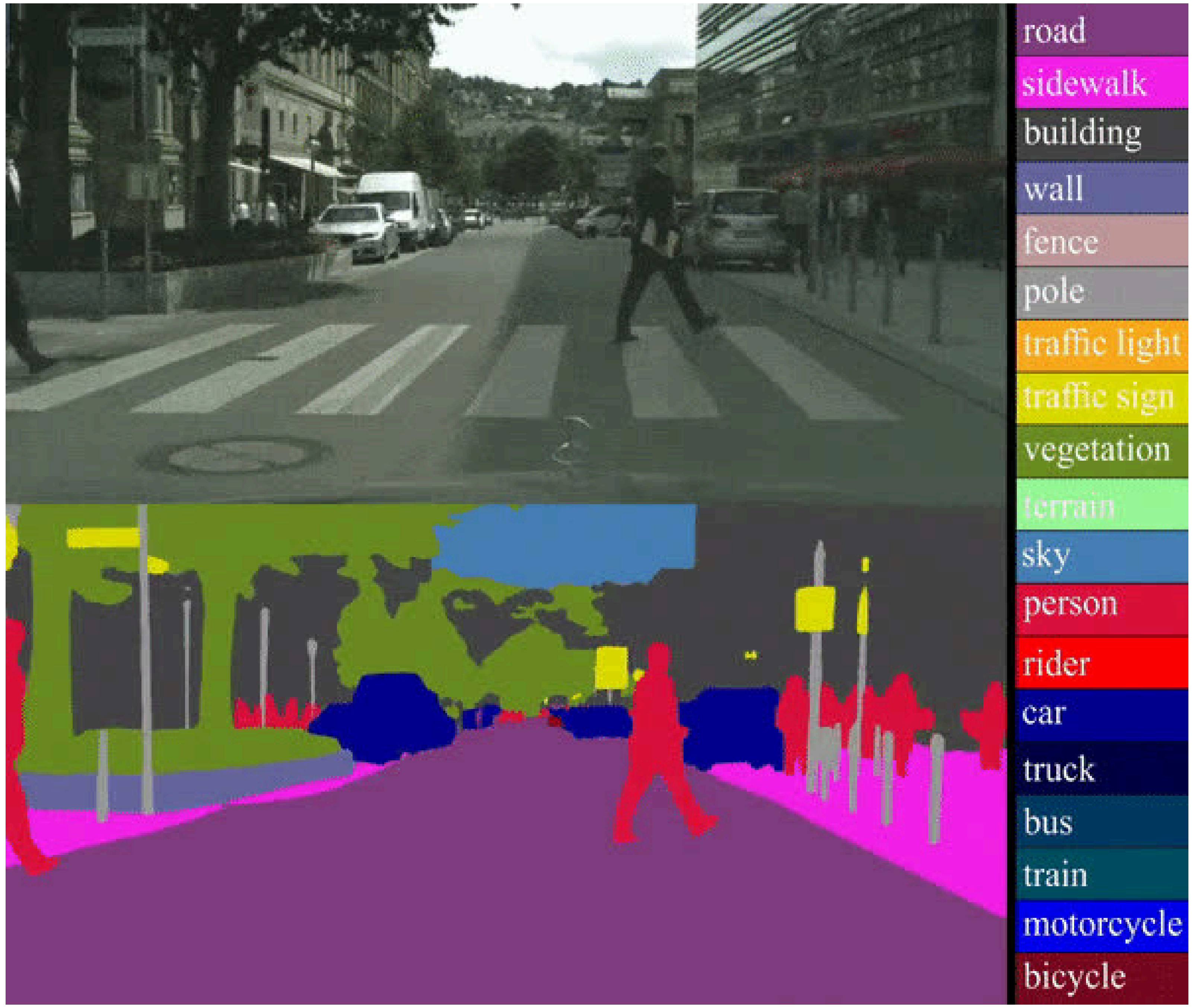
EXAMPLES OF COMPUTER VISION

1. Object detection and recognition
2. Facial recognition
3. Optical character recognition (OCR)
4. Image segmentation
5. 3D reconstruction
6. Gesture recognition



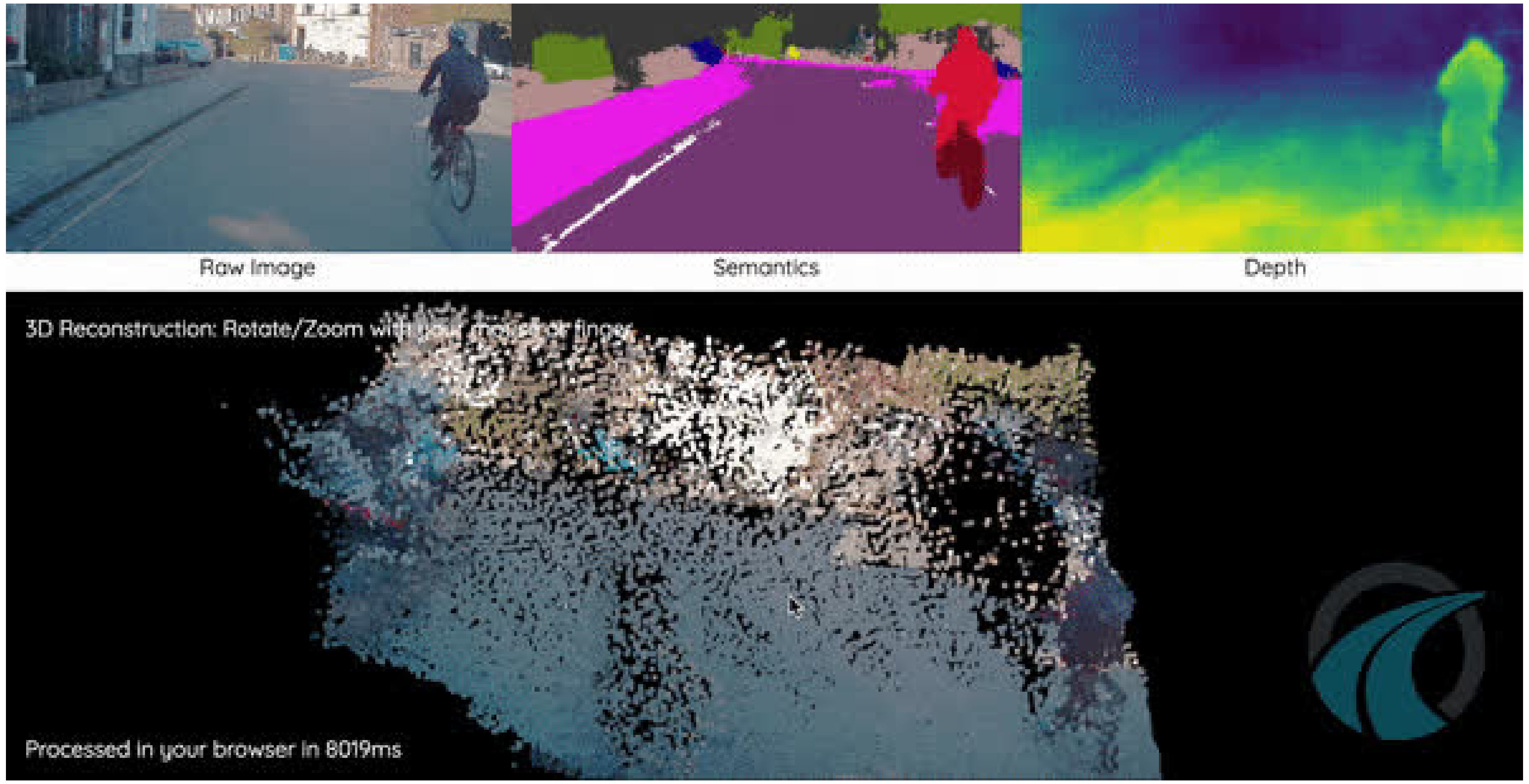
EXAMPLES OF COMPUTER VISION

1. Object detection and recognition
2. Facial recognition
3. Optical character recognition (OCR)
4. Image segmentation
5. 3D reconstruction
6. Gesture recognition



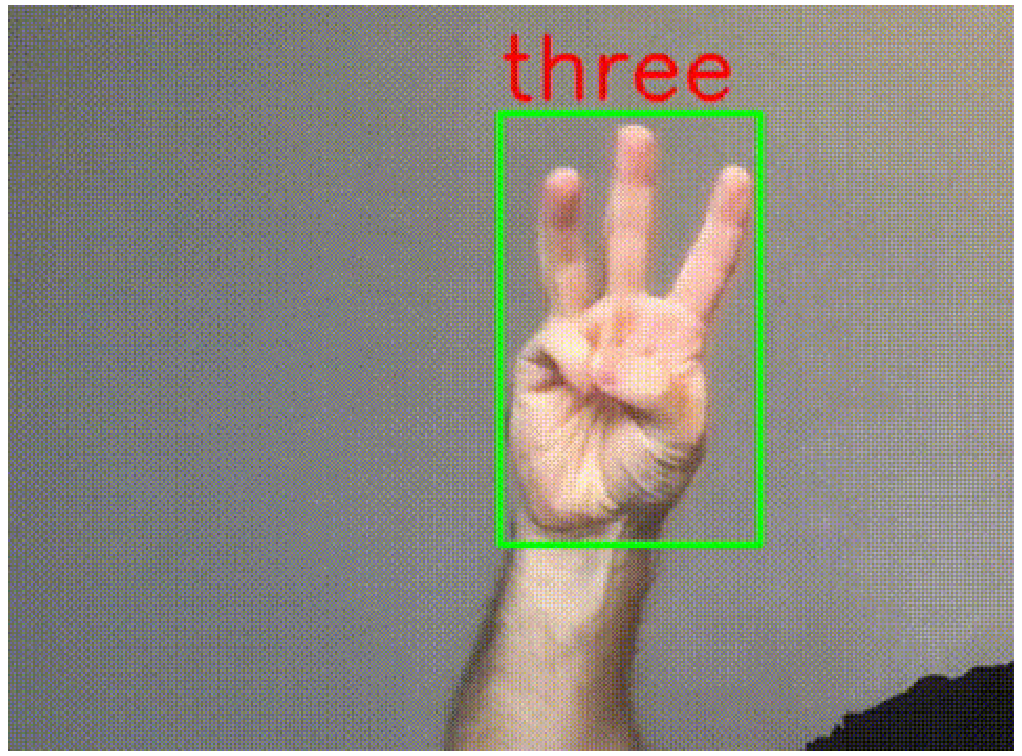
EXAMPLES OF COMPUTER VISION

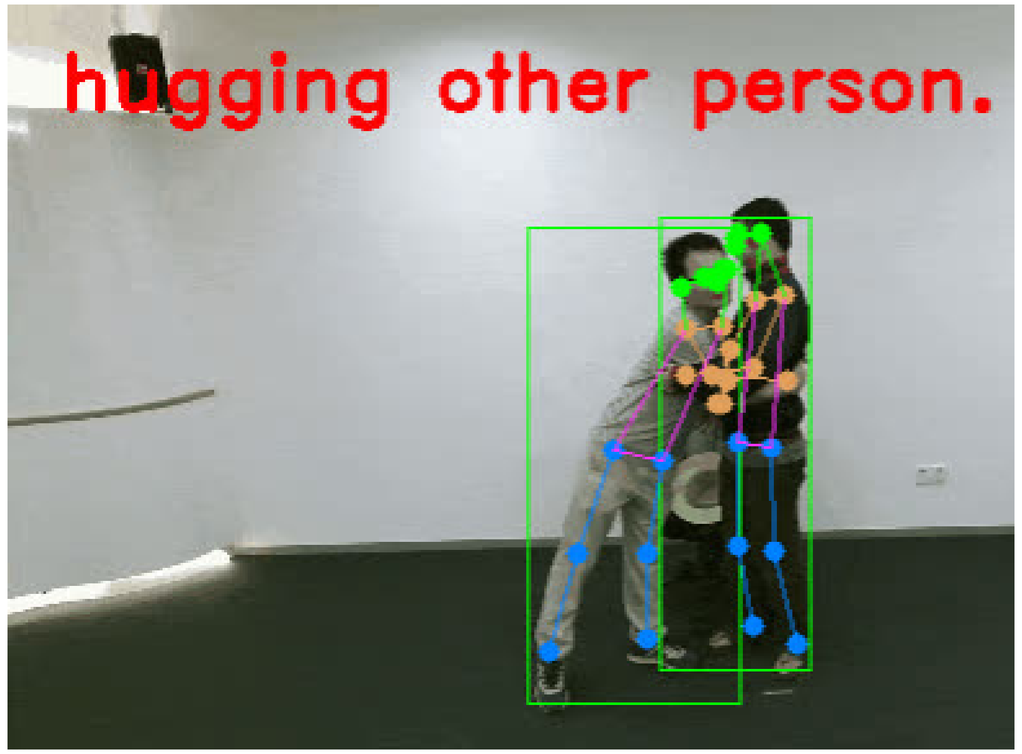
1. Object detection and recognition
2. Facial recognition
3. Optical character recognition (OCR)
4. Image segmentation
5. **3D reconstruction**
6. Gesture recognition



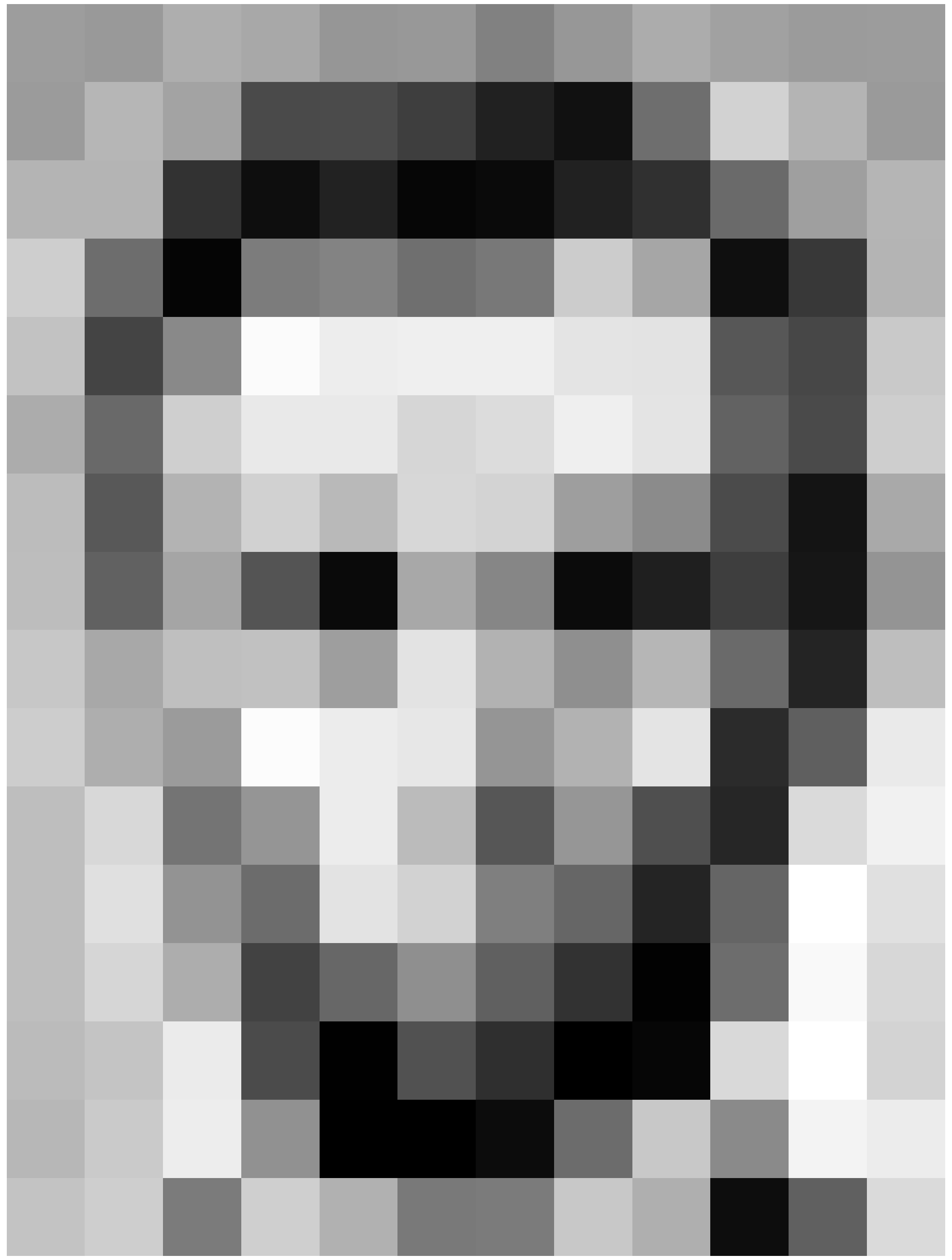
EXAMPLES OF COMPUTER VISION

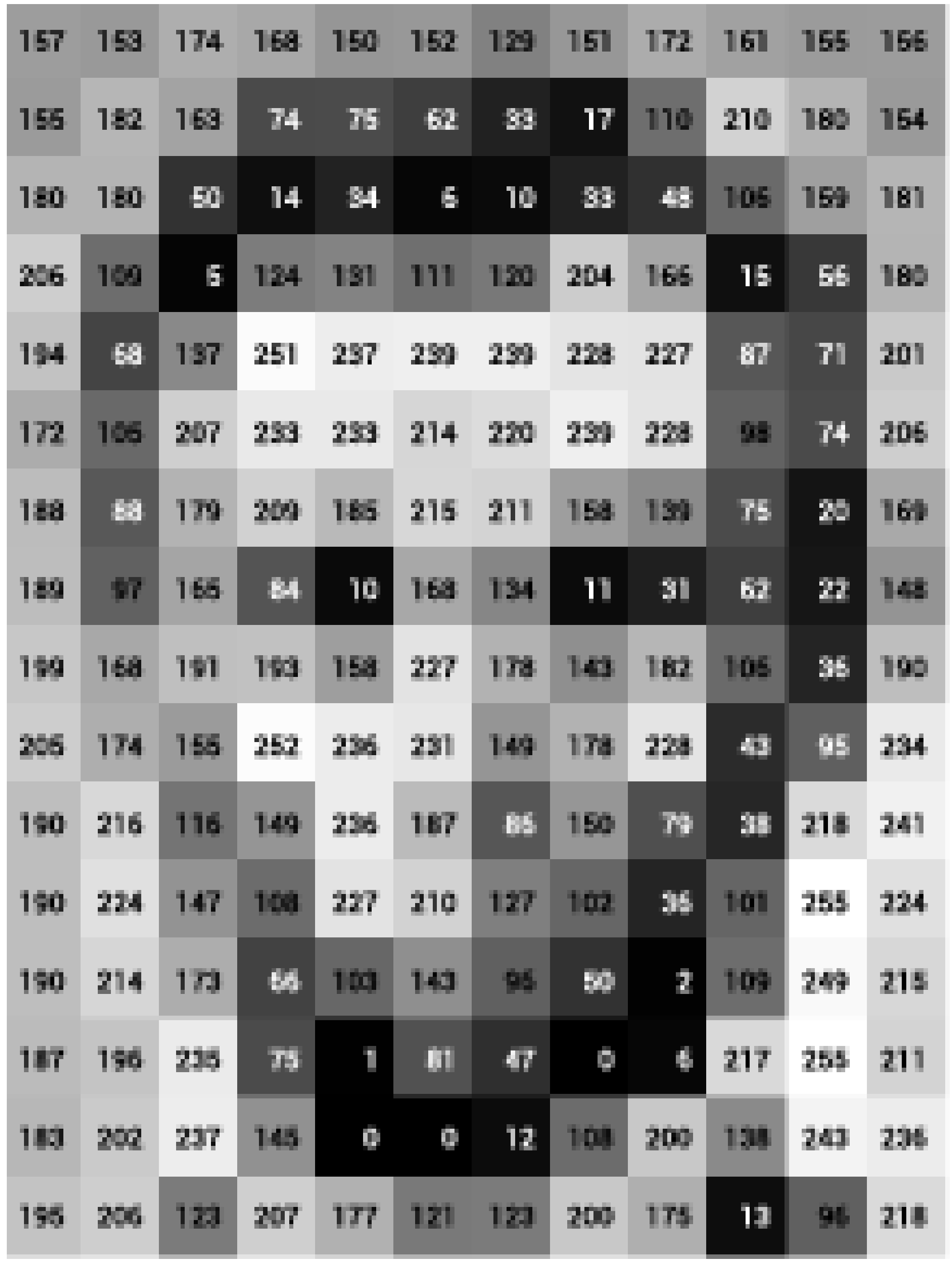
1. Object detection and recognition
2. Facial recognition
3. Optical character recognition (OCR)
4. Image segmentation
5. 3D reconstruction
6. Gesture recognition





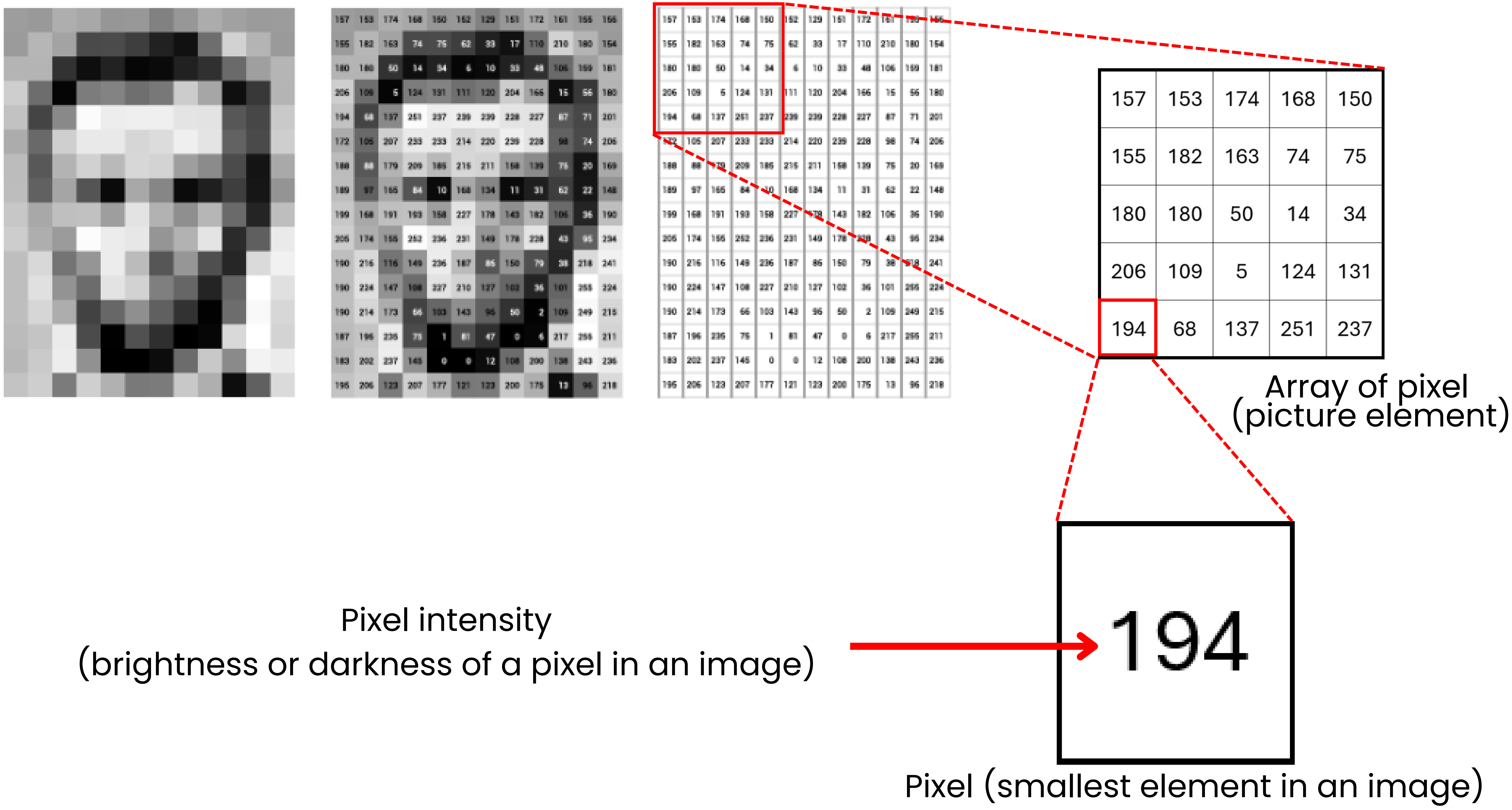
WHAT IS DIGITAL IMAGE?





157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

WHAT IS DIGITAL IMAGE?



PIXELS

- 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

Spatial resolution	Number of channels
Intensity resolution	Opacity

PROPERTIES OF IMAGES

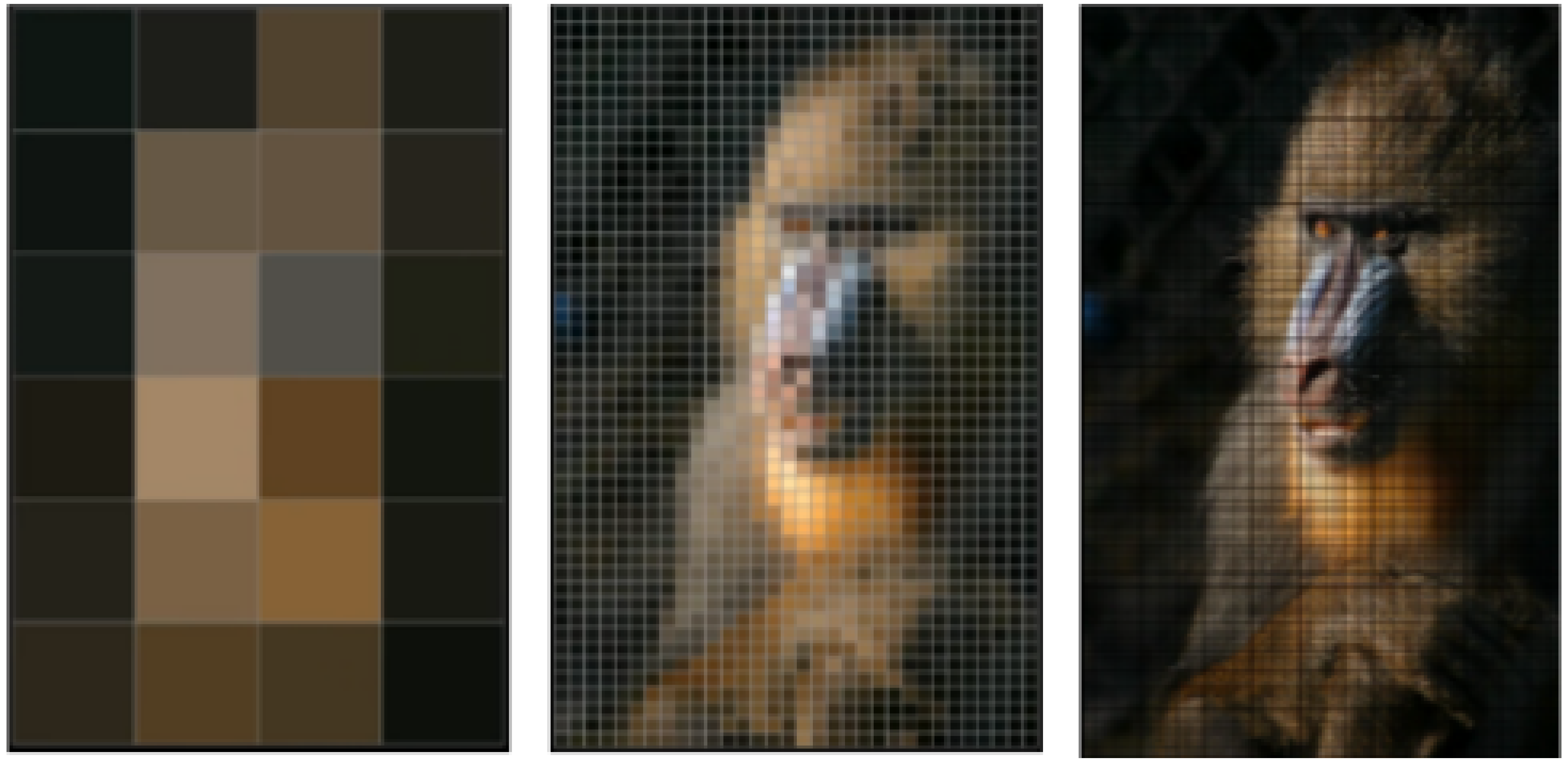
Spatial resolution

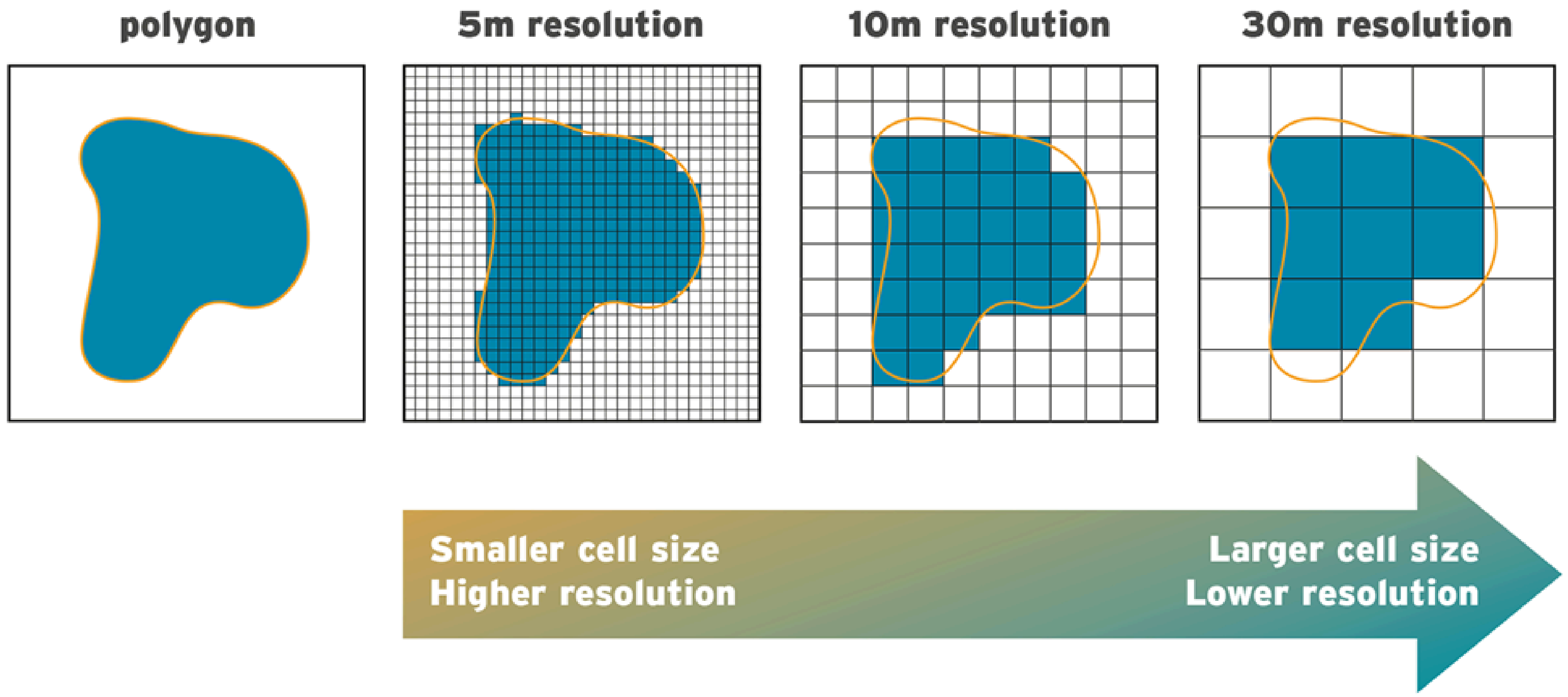
Number of channels

Intensity resolution

Opacity

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





PROPERTIES OF IMAGES

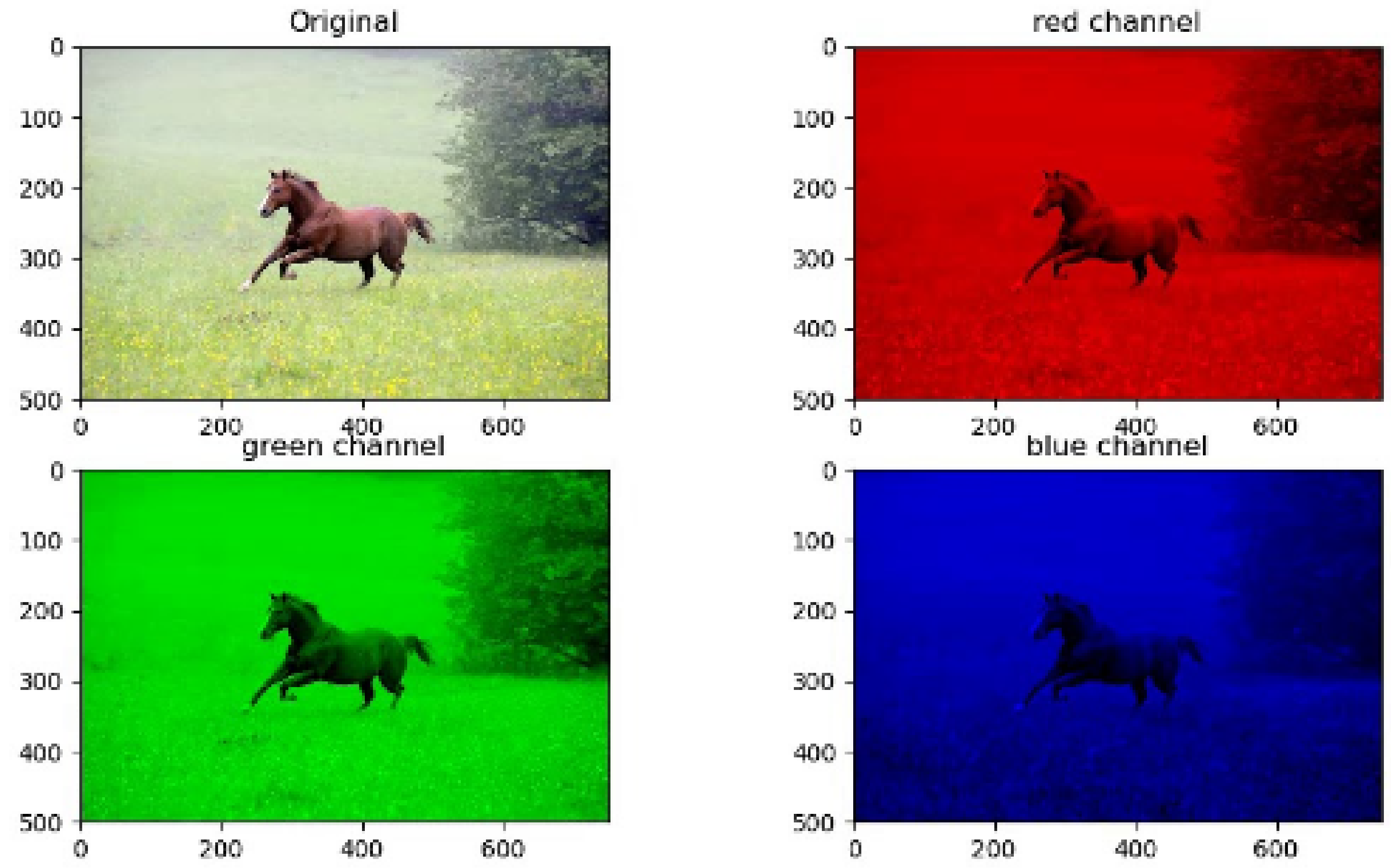
Spatial resolution

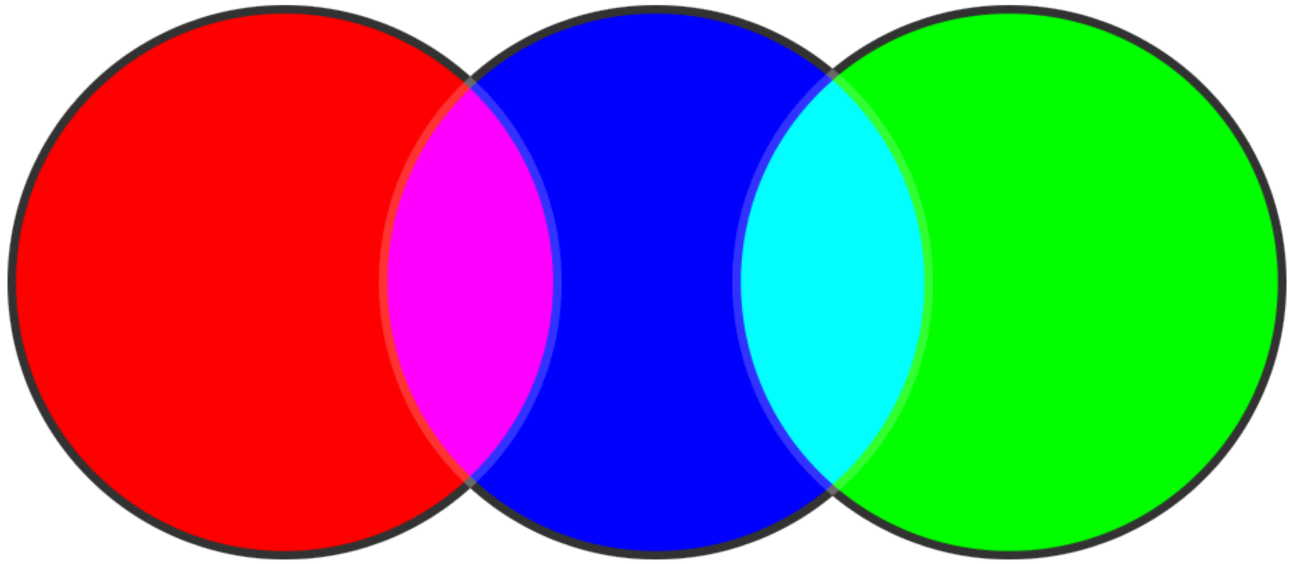
Number of channels

Intensity resolution

Opacity

number of channels refers to the number of separate and distinct components or bands of information that make up an image



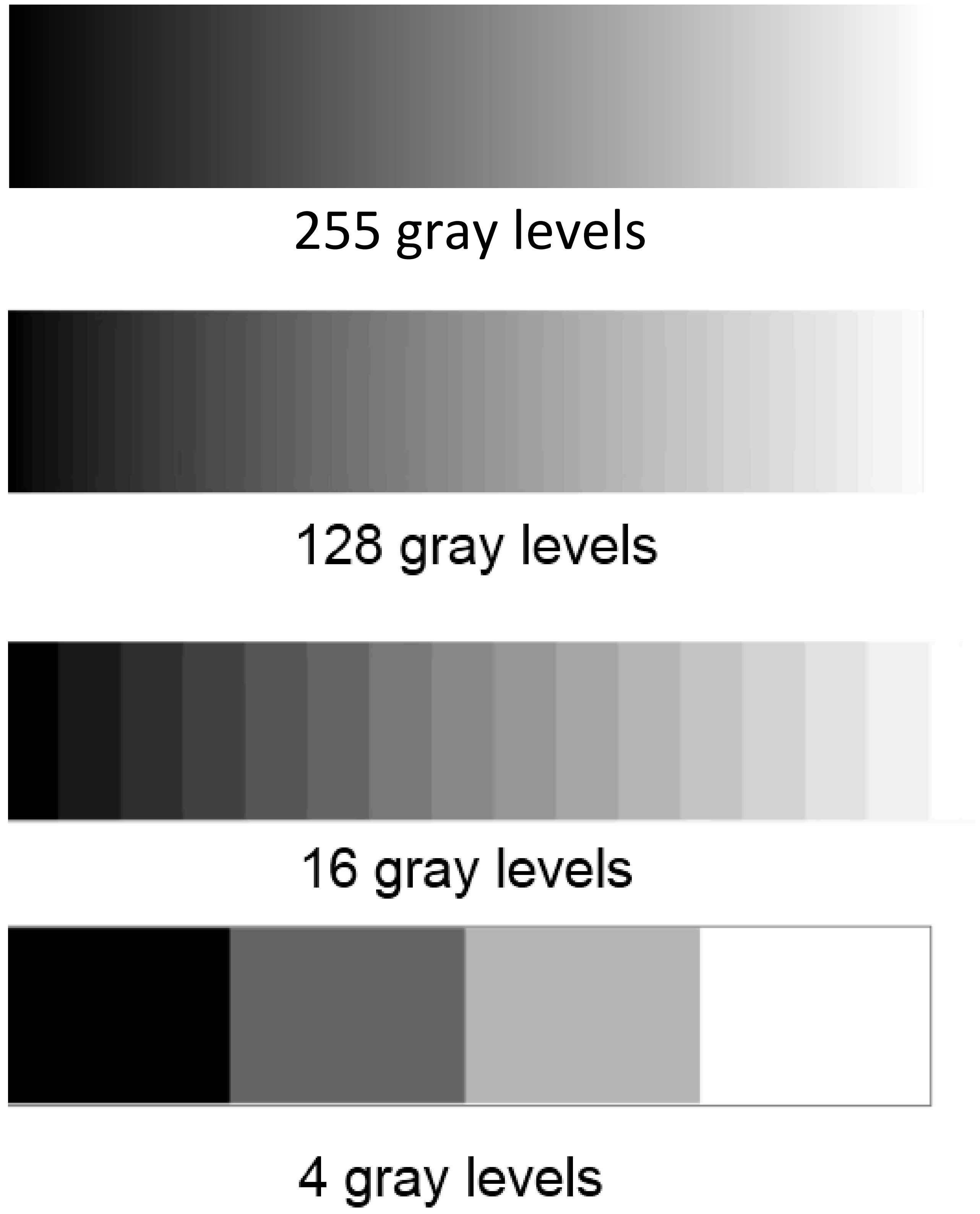


PROPERTIES OF IMAGES

Spatial resolution | Number of channels

Intensity resolution | Opacity

- ability of an imaging system to distinguish between different levels of brightness or intensity in an image
- determining the image's color depth and the range of colors and shades it can display



PROPERTIES OF IMAGES

Spatial resolution

Number of channels

Intensity resolution

Opacity

- ability of an imaging system to distinguish between different levels of brightness or intensity in an image
- determining the image's color depth and the range of colors and shades it can display

128 gray levels

4 gray levels

PROPERTIES OF IMAGES

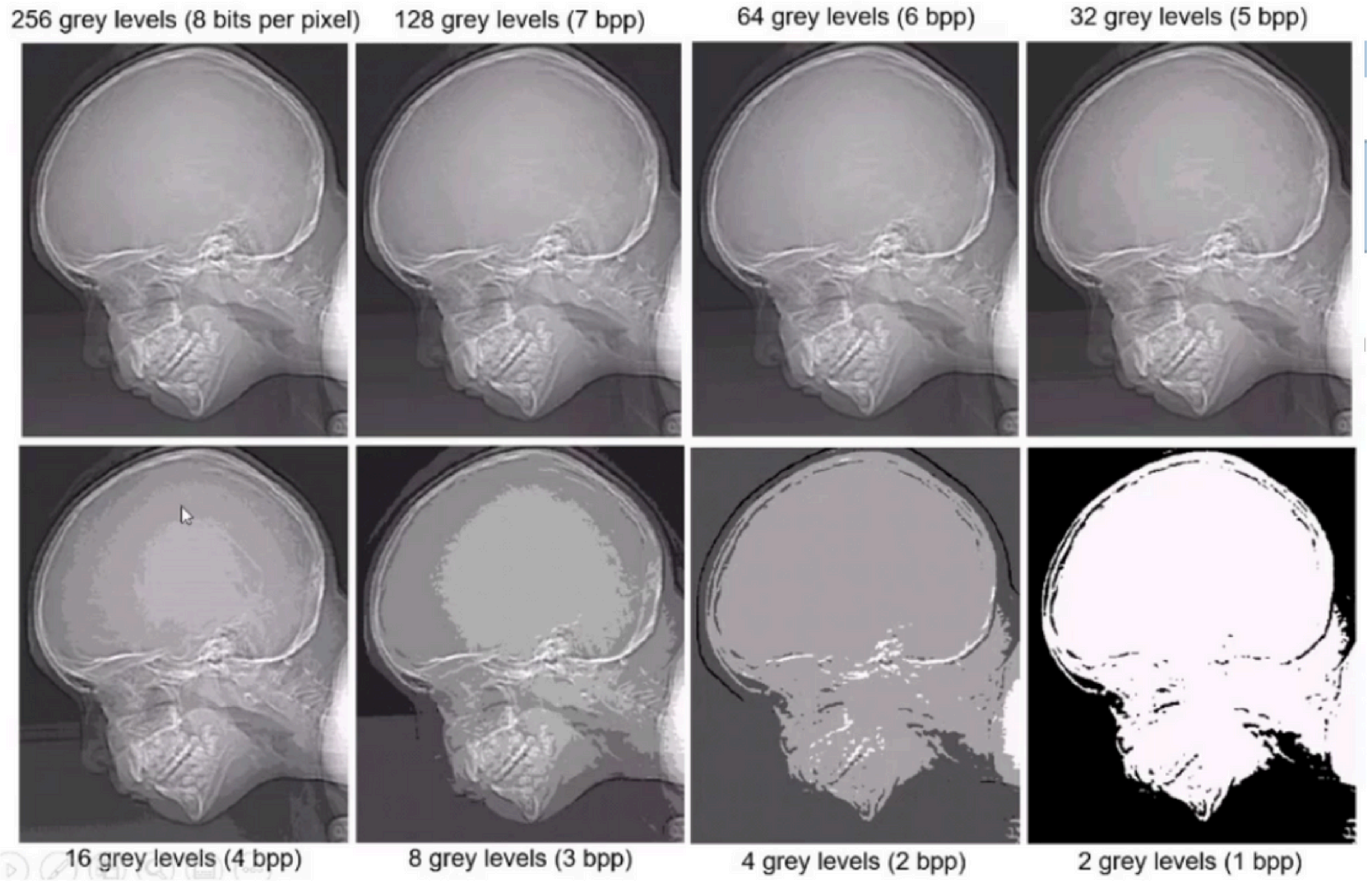
Spatial resolution

Number of channels

Intensity resolution

Opacity

- ability of an imaging system to distinguish between different levels of brightness or intensity in an image
- determining the image's color depth and the range of colors and shades it can display



PROPERTIES OF IMAGES

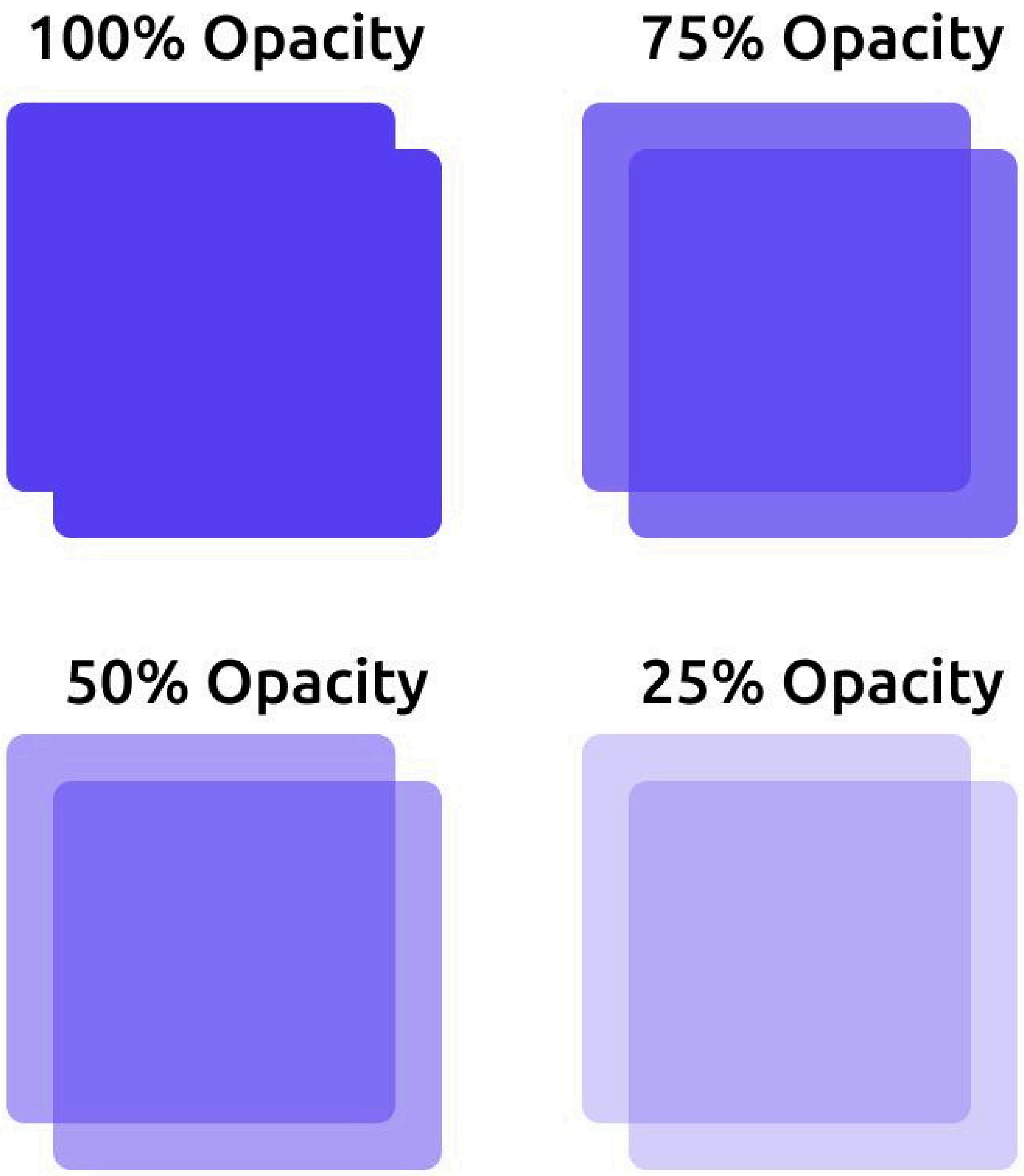
Spatial resolution

Number of channels

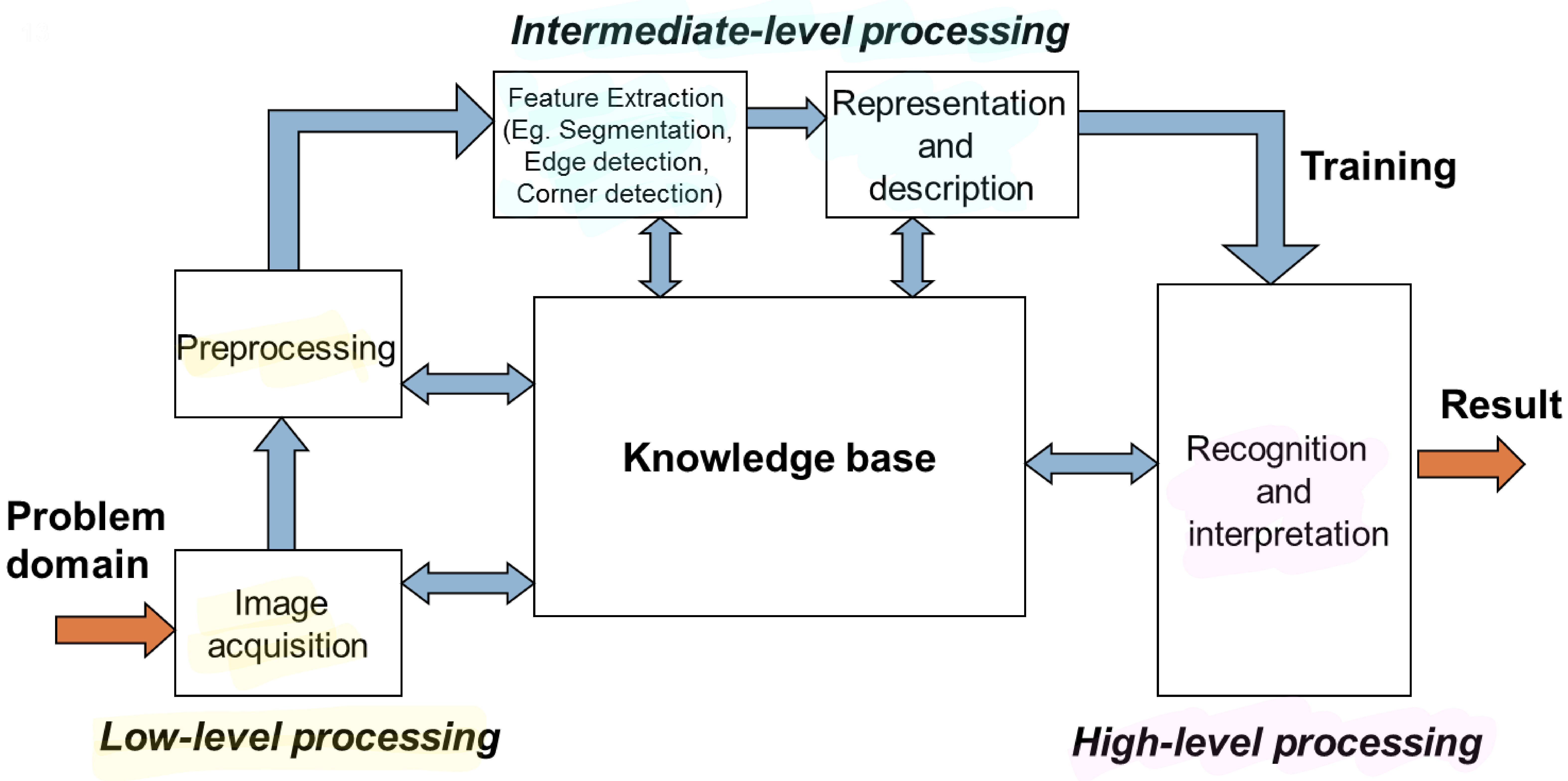
Intensity resolution

Opacity

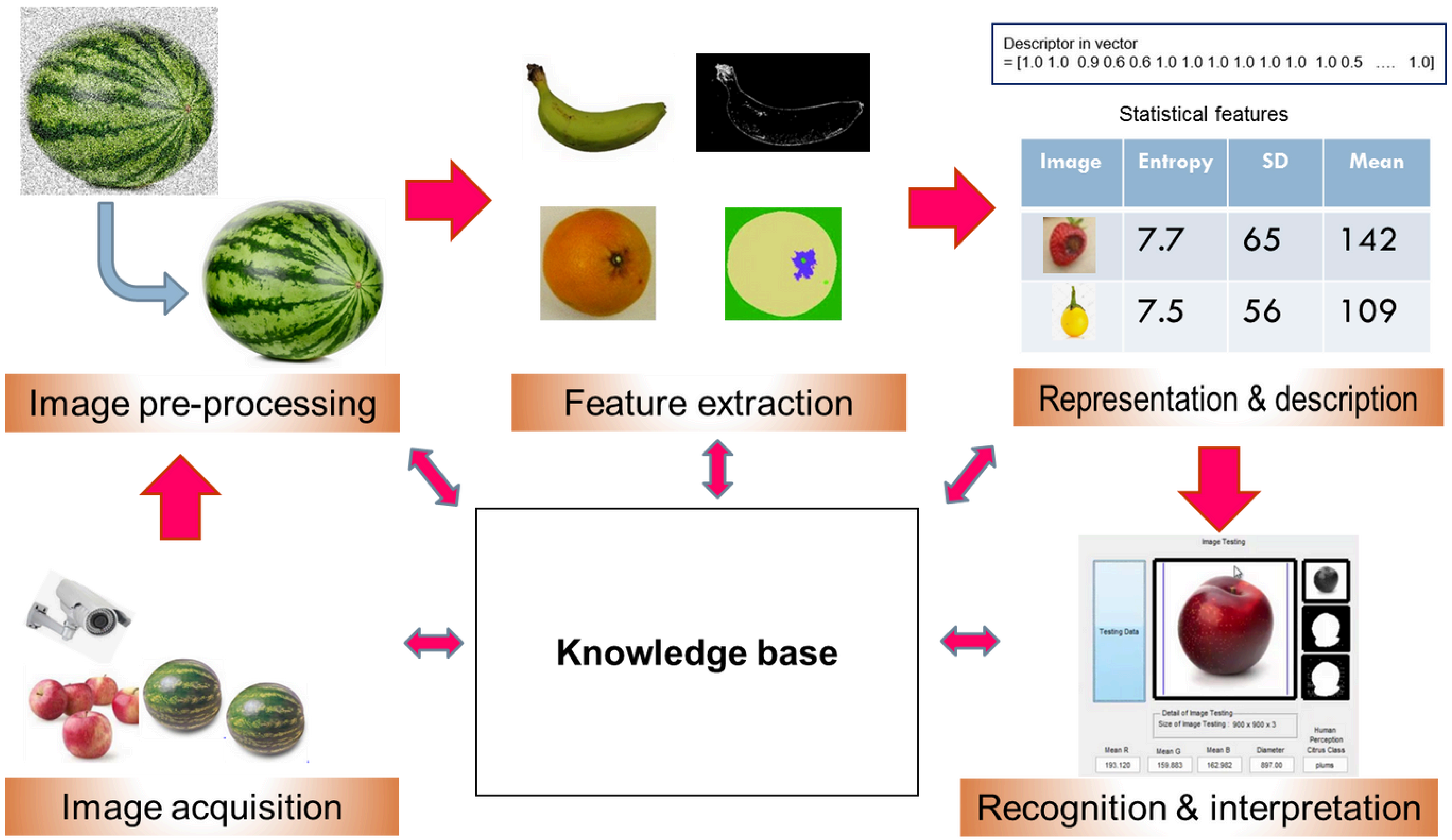
degree to which a material prevents light from passing through it.



FUNDAMENTAL STEPS IN IMAGE PROCESSING



FUNDAMENTAL STEPS IN IMAGE PROCESSING



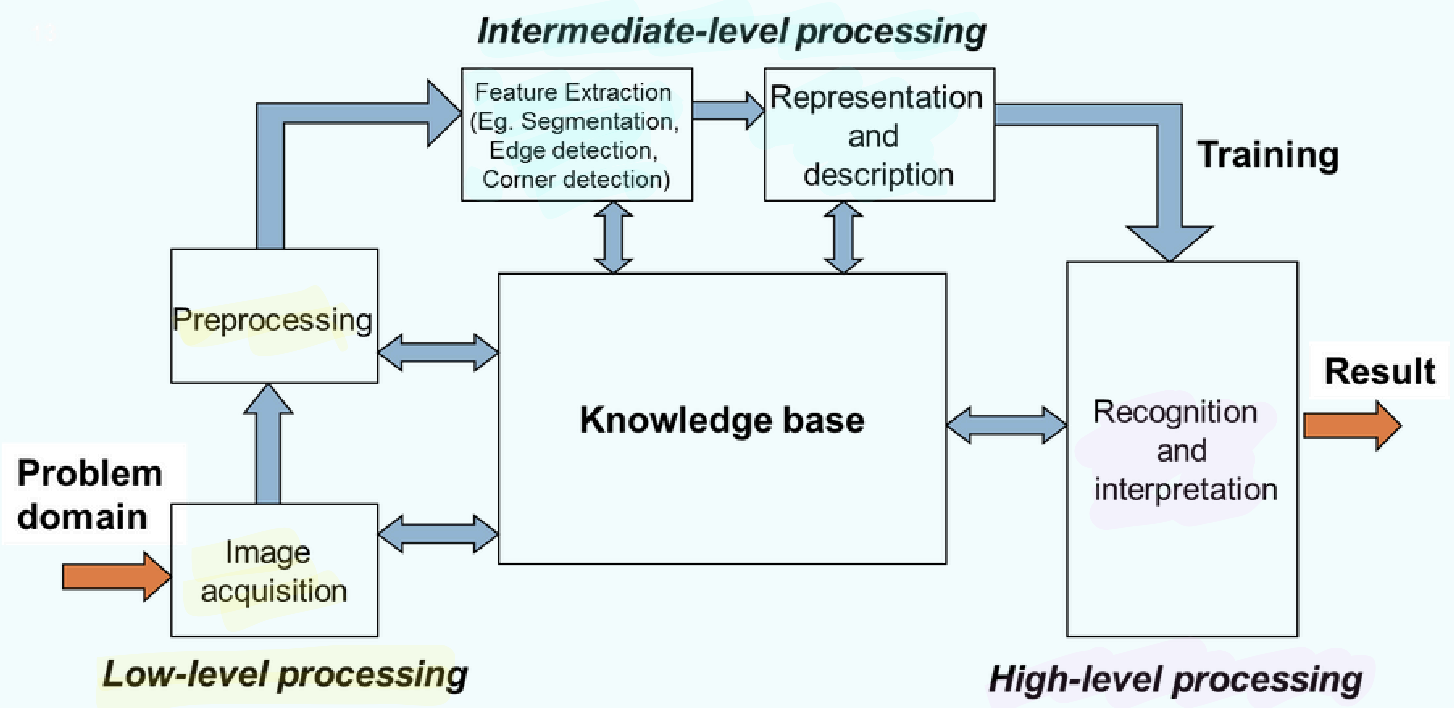


IMAGE ACQUISITION

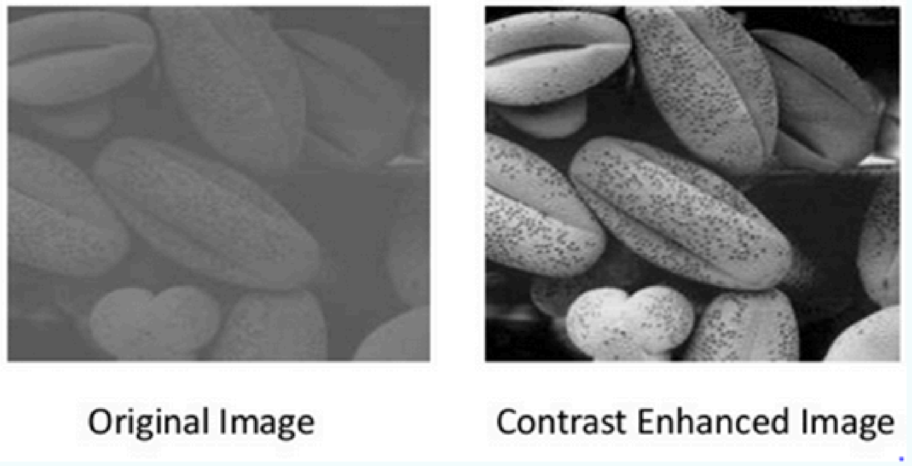
- Requires imaging sensor and capability to digitize the signal produced by the sensor.
- Example of imaging sensor: digital camera, scanner

PREPROCESSING

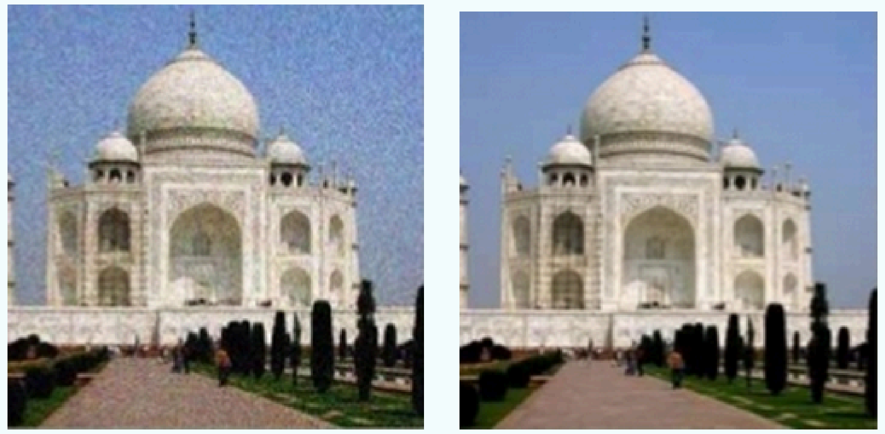
- To improve the image (to increase chance for success of other processes)

Examples

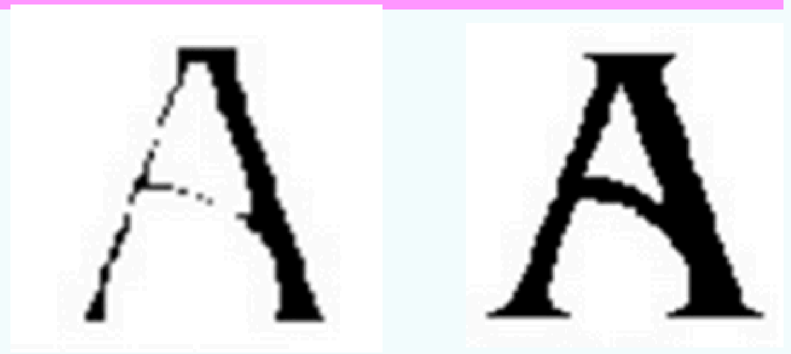
Enhancing Contrast



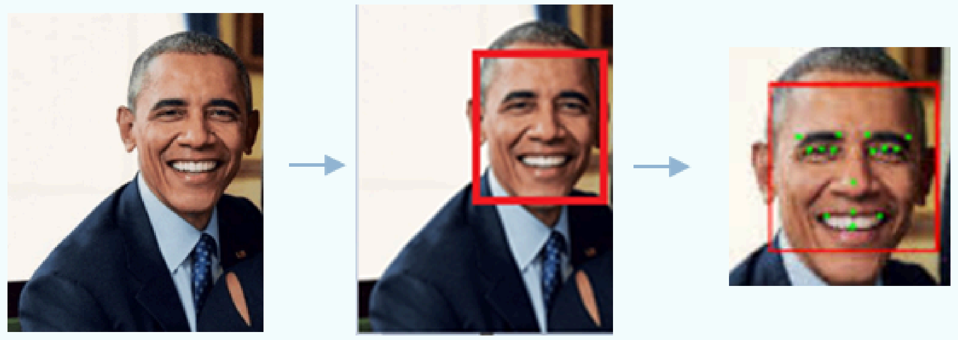
Removing noise



Morphological Operation



Isolating Regions

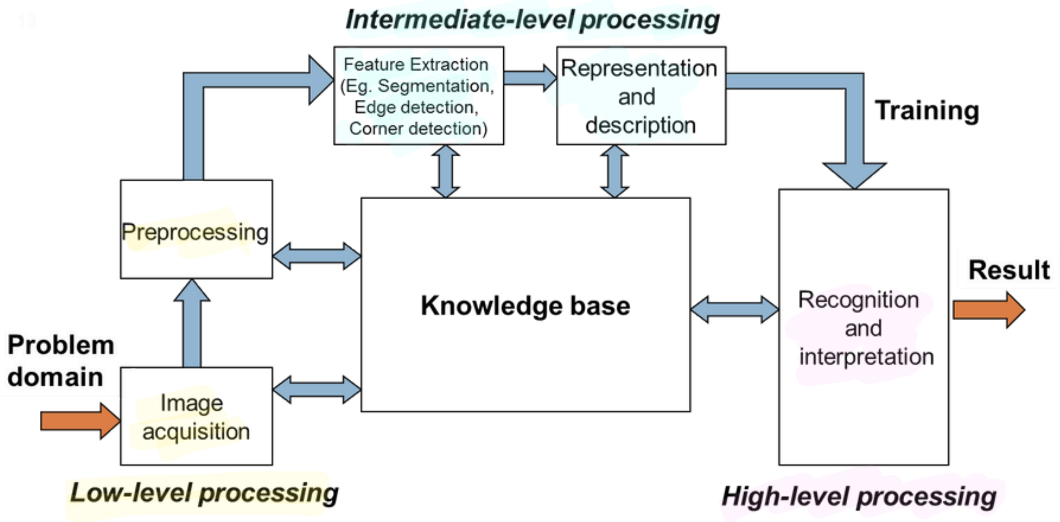


PREPROCESSING

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.

CORRELATION



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

SUBSET

KERNEL/FILTER/
MASK

CORRELATION

244	255	246
255	240	183
255	250	12

SUBSET

*

1	2	3
4	5	6
7	8	9

KERNEL/FILTER/
MASK

=

244	510	738
1020	1200	1098
1785	2000	108

$\Sigma \downarrow$

8703

CORRELATION

244	255	246
255	240	183
255	250	12

SUBSET

*

1	2	3
4	5	6
7	8	9

KERNEL/FILTER/
MASK

=

244	510	738
1020	1200	1098
1785	2000	108

$\Sigma \Downarrow$

8703 $\Rightarrow$

	8703	

EXAMPLE

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

KERNEL/FILTER/ MASK

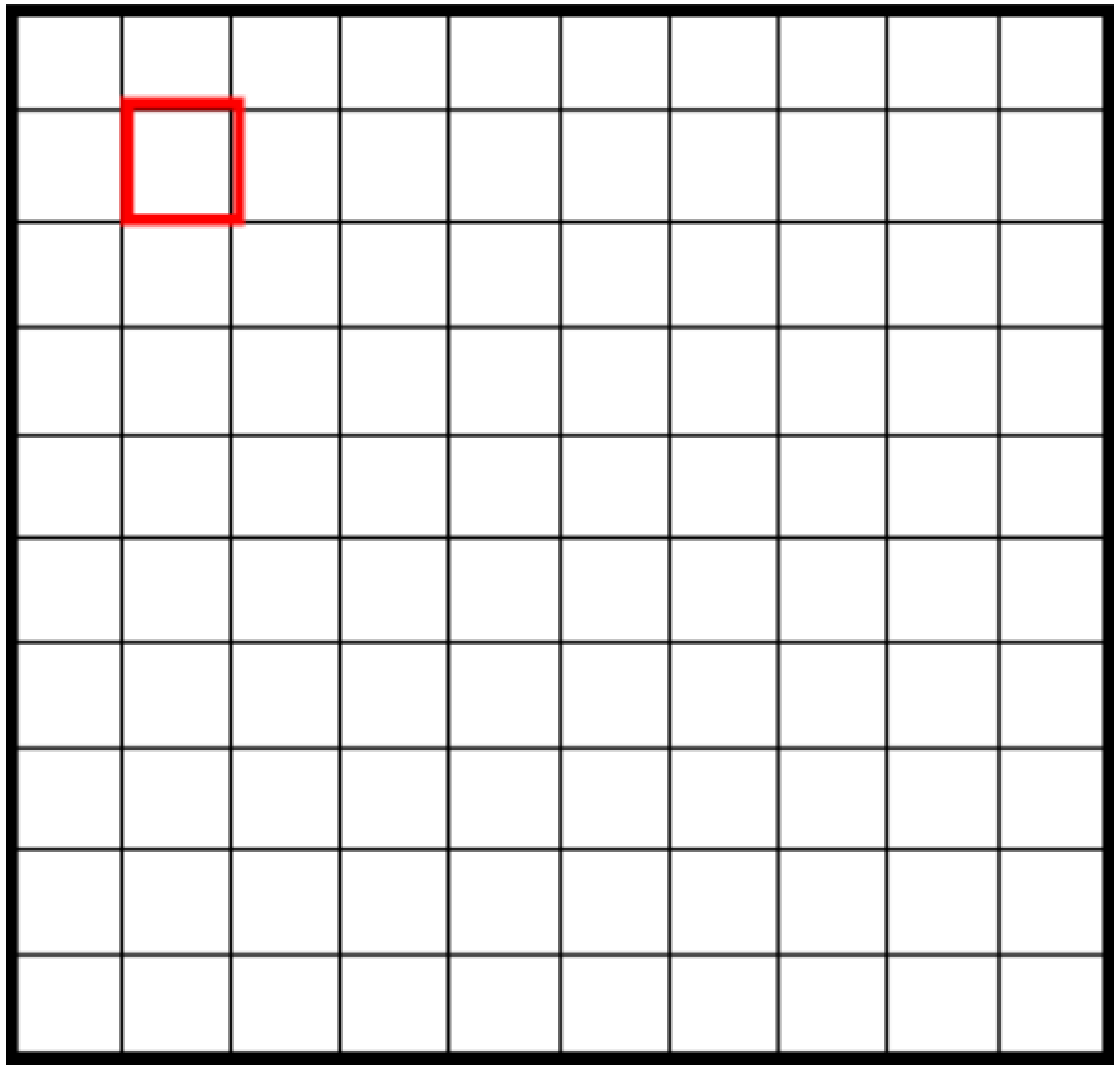
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

EXAMPLE

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

KERNEL/FILTER/ MASK

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



EXAMPLE

$$\frac{1}{9} * \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

KERNEL/FILTER/
MASK

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							

EXAMPLE

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

KERNEL/FILTER/
MASK

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						

EXAMPLE

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

KERNEL/FILTER/
MASK

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					

EXAMPLE

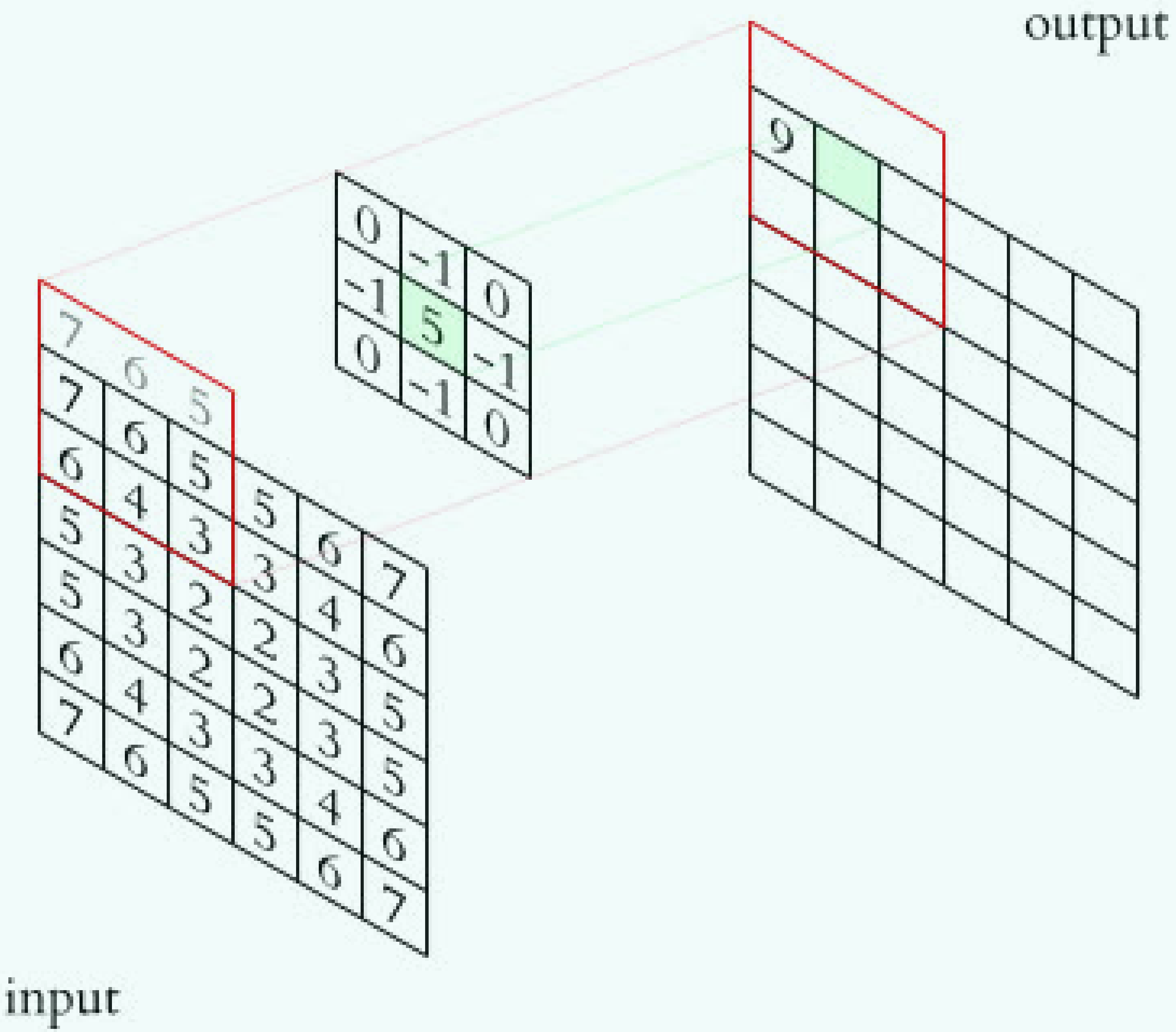
$$\frac{1}{9} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

KERNEL/FILTER/
MASK

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

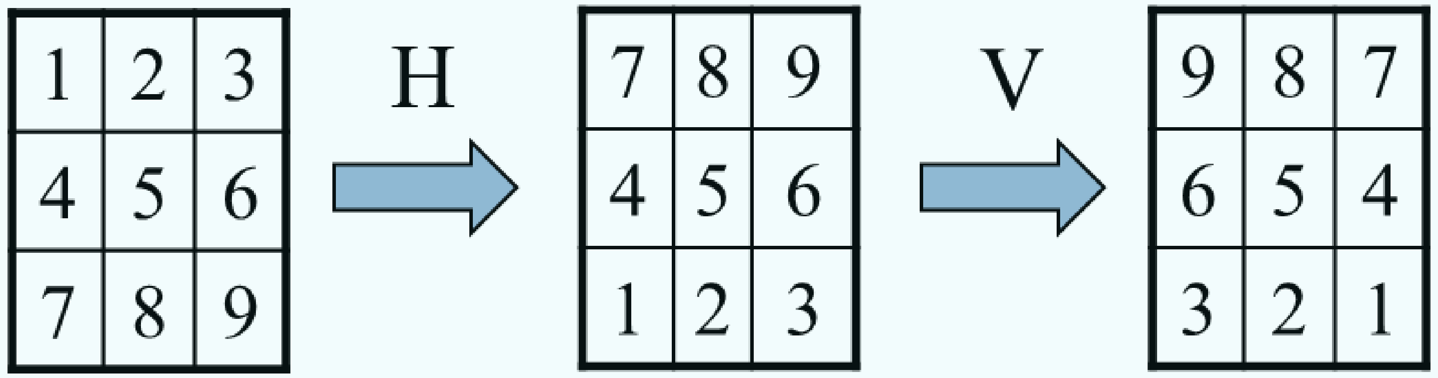
	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

EXAMPLE

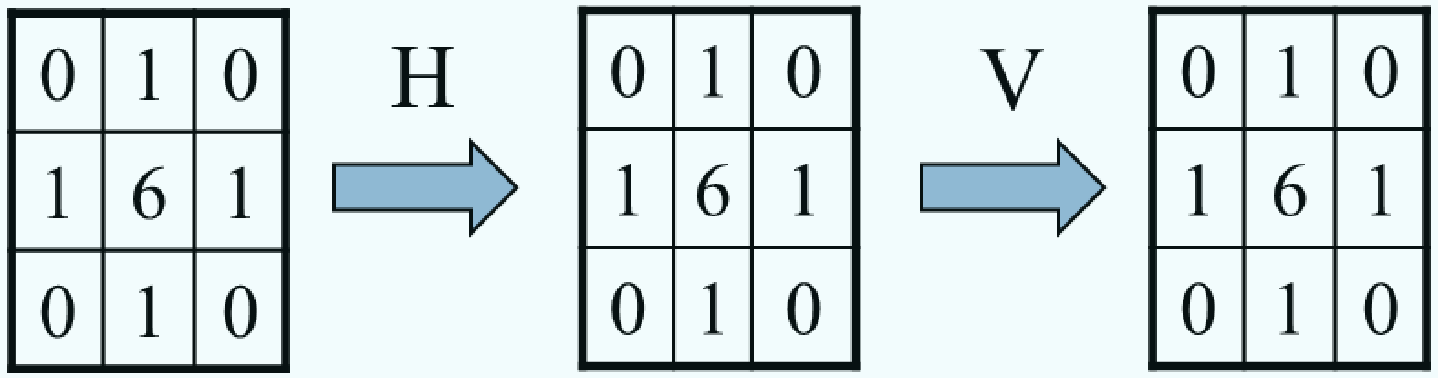


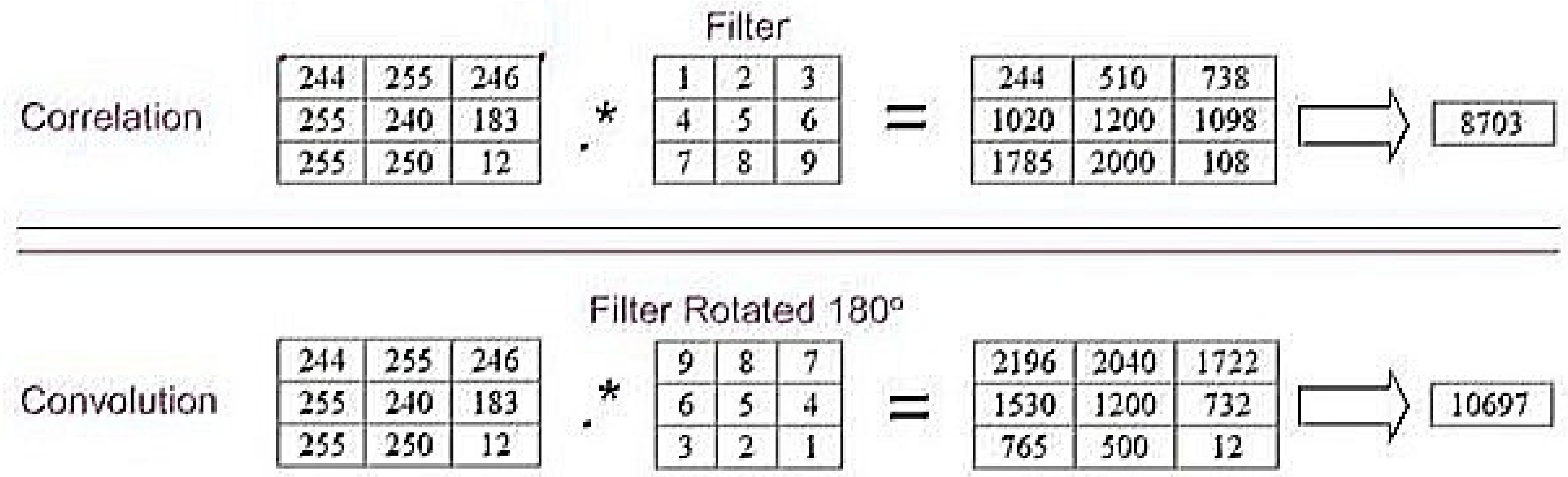
CONVOLUTION

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



For symmetric masks, convolution is equivalent to correlation.



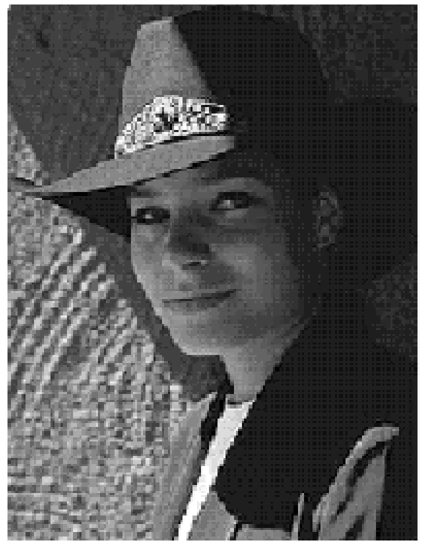


PREPROCESSING - IMAGE NOISE REMOVER

- **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.

PREPROCESSING - IMAGE NOISE REMOVER

- 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

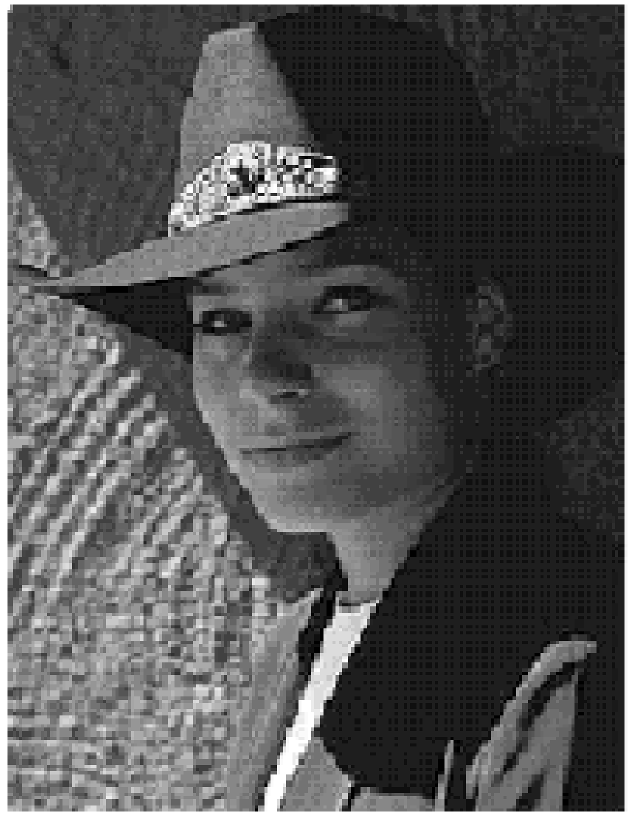

Original


Salt and pepper noise


Impulse noise


Gaussian noise

PREPROCESSING - IMAGE NOISE REMOVER



Original

Salt and pepper noise

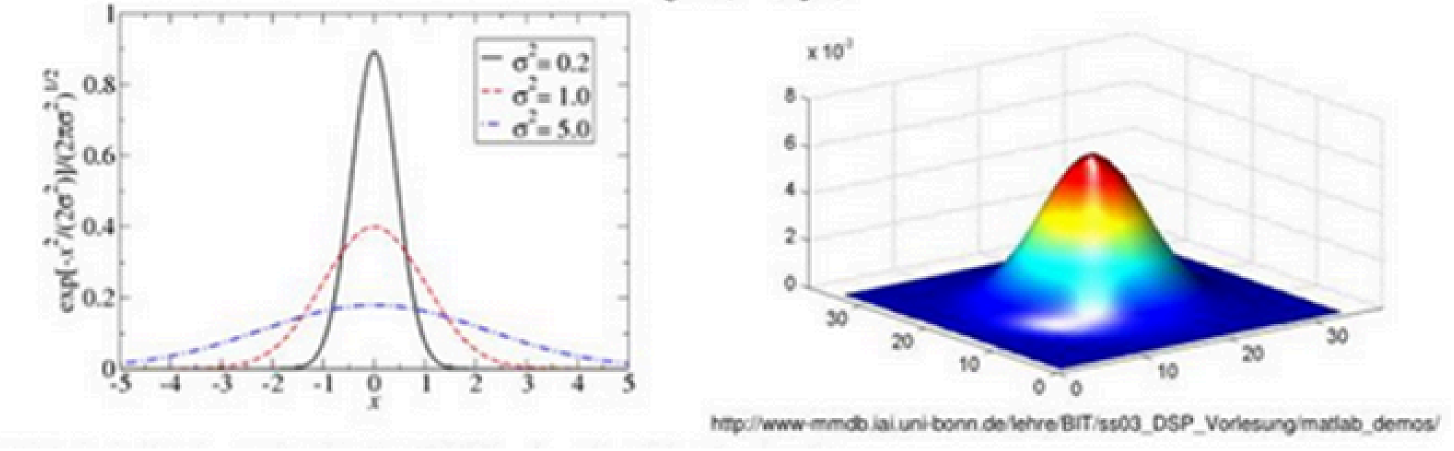
Impulse noise

Gaussian noise

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

GAUSSIAN NOISE – GAUSSIAN FILTERING

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



Filter with Gaussian Filter with different standard deviation value.

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

σ = 1 pixel

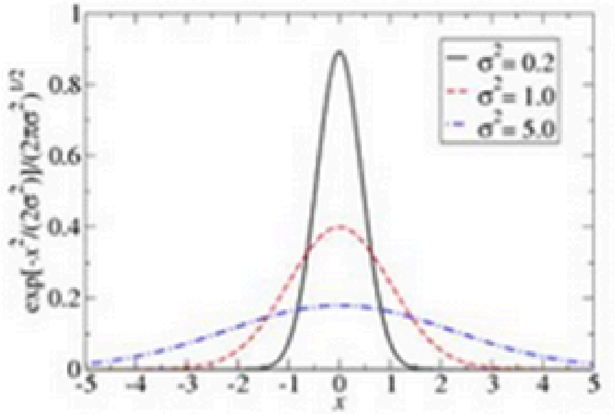
σ = 2 pixels

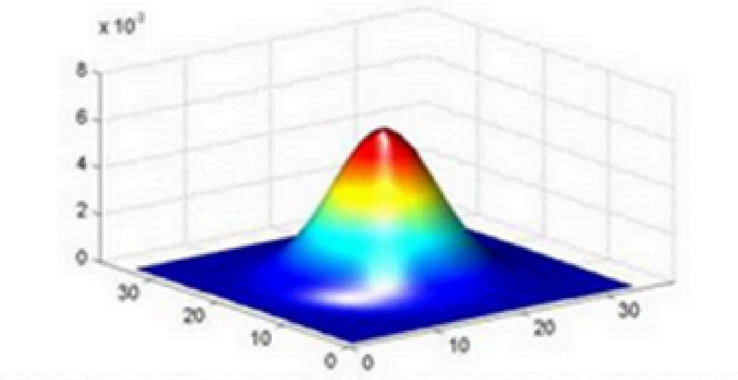
σ = 5 pixels

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

SALT & PEPPER NOISE – GAUSSIAN FILTERING

- 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/

Filter with Gaussian Filter with different standard deviation value.

p = 10%

σ = 1 pixel

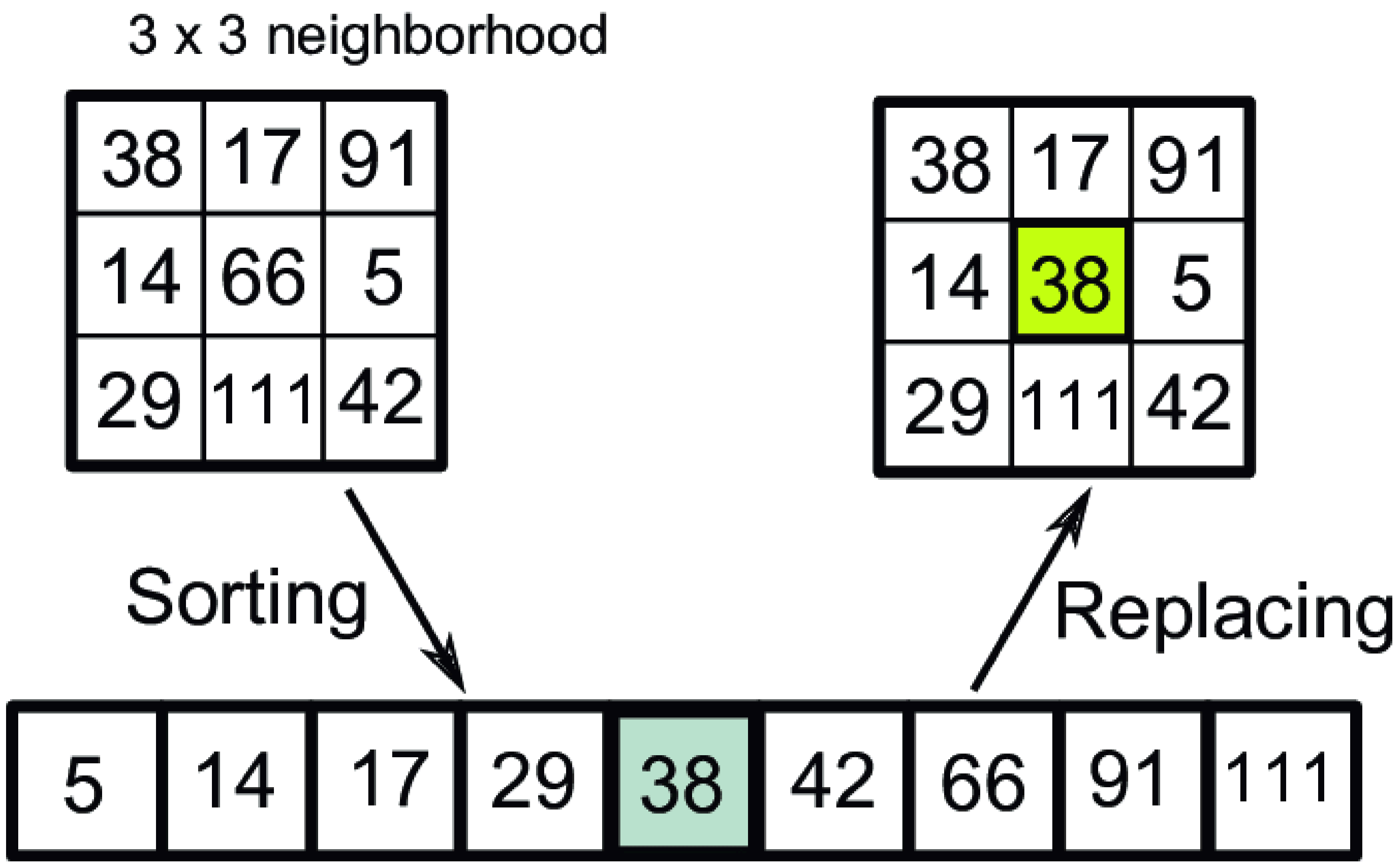
σ = 2 pixels

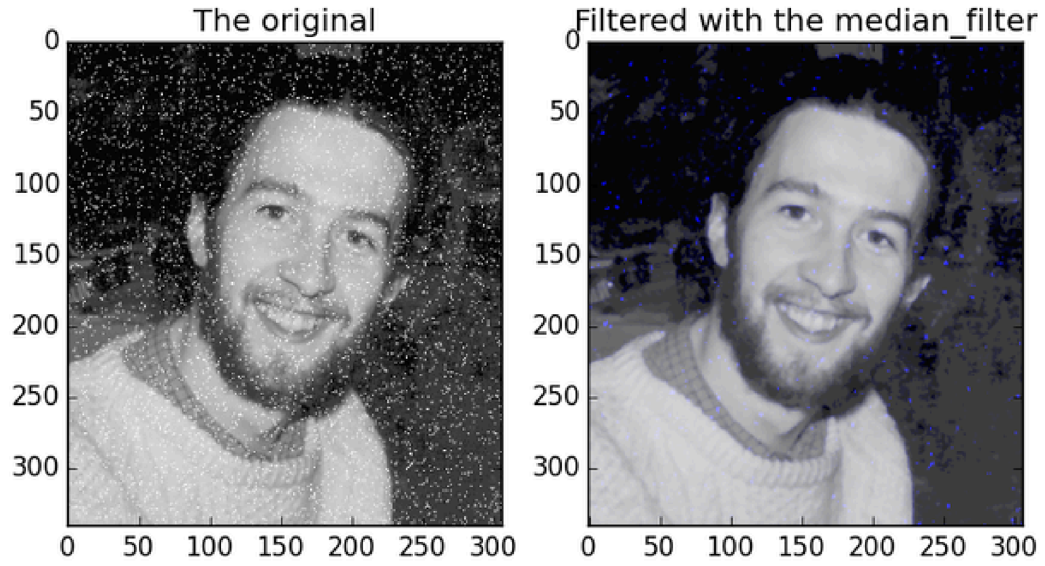
σ = 5 pixels

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





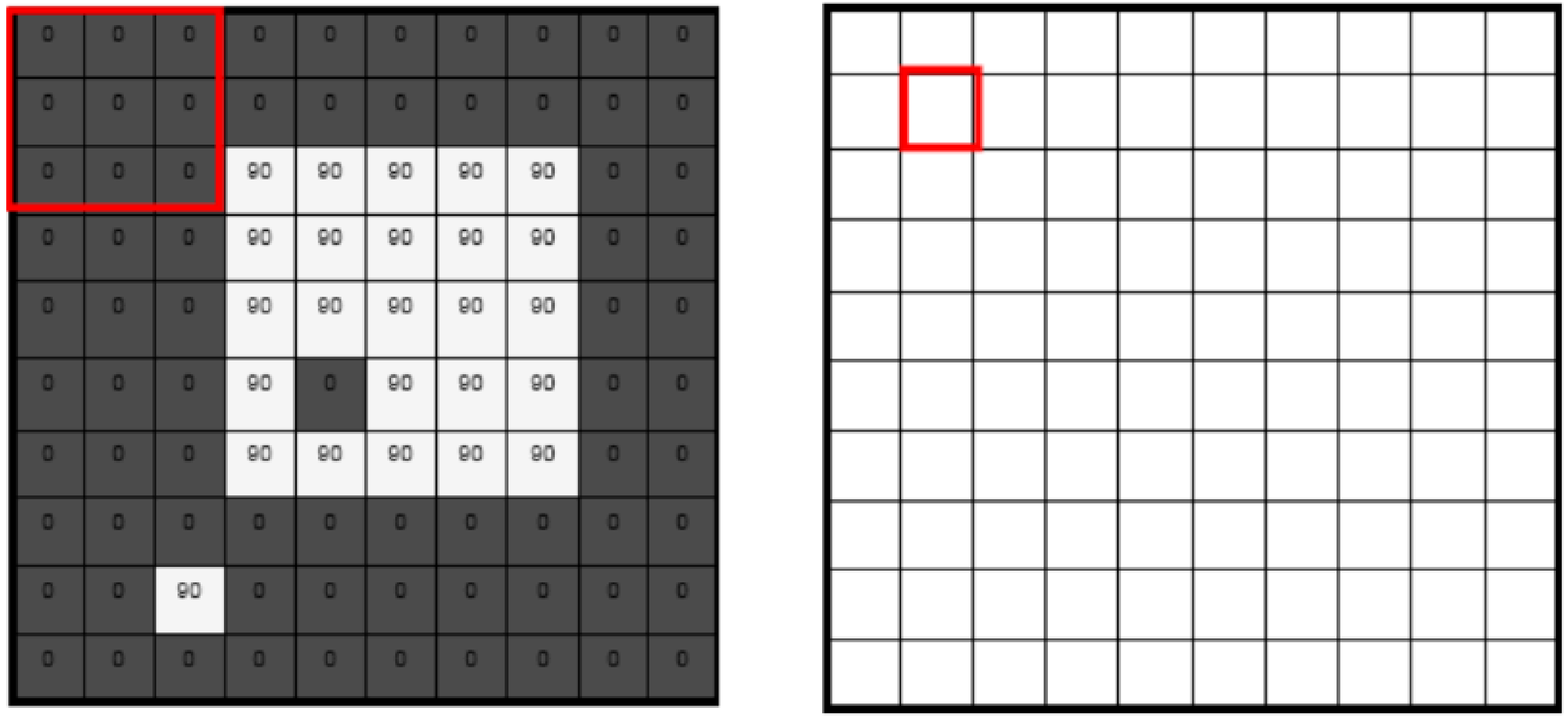
AVERAGE FILTER

Replacing each pixel in an image with the average value of its neighboring pixels.

Recall:

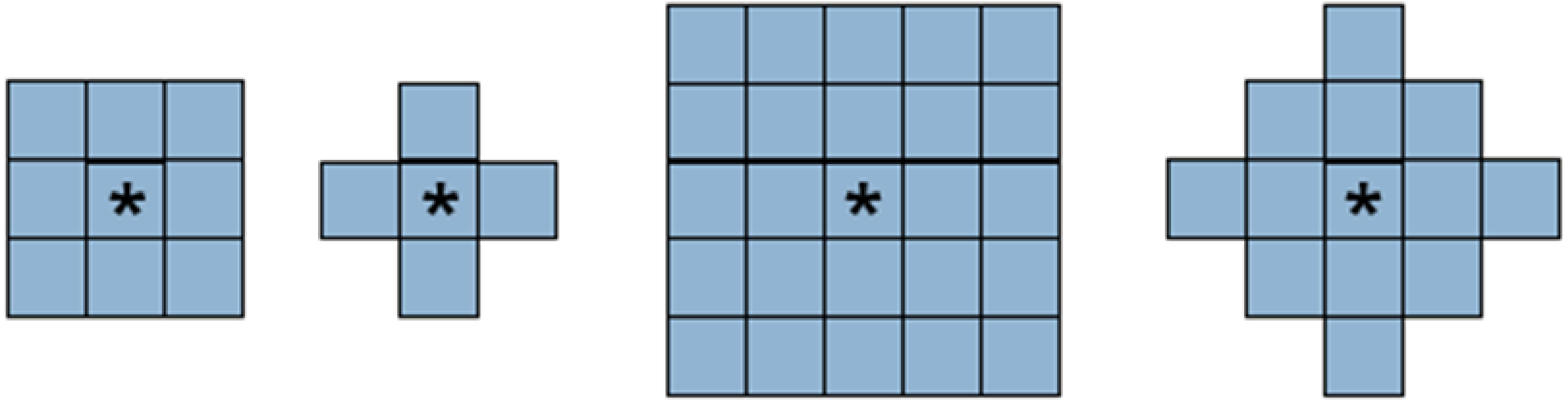
$$\frac{1}{9} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

KERNEL/FILTER/
MASK



PREPROCESSING - MORPHOLOGICAL OPERATION

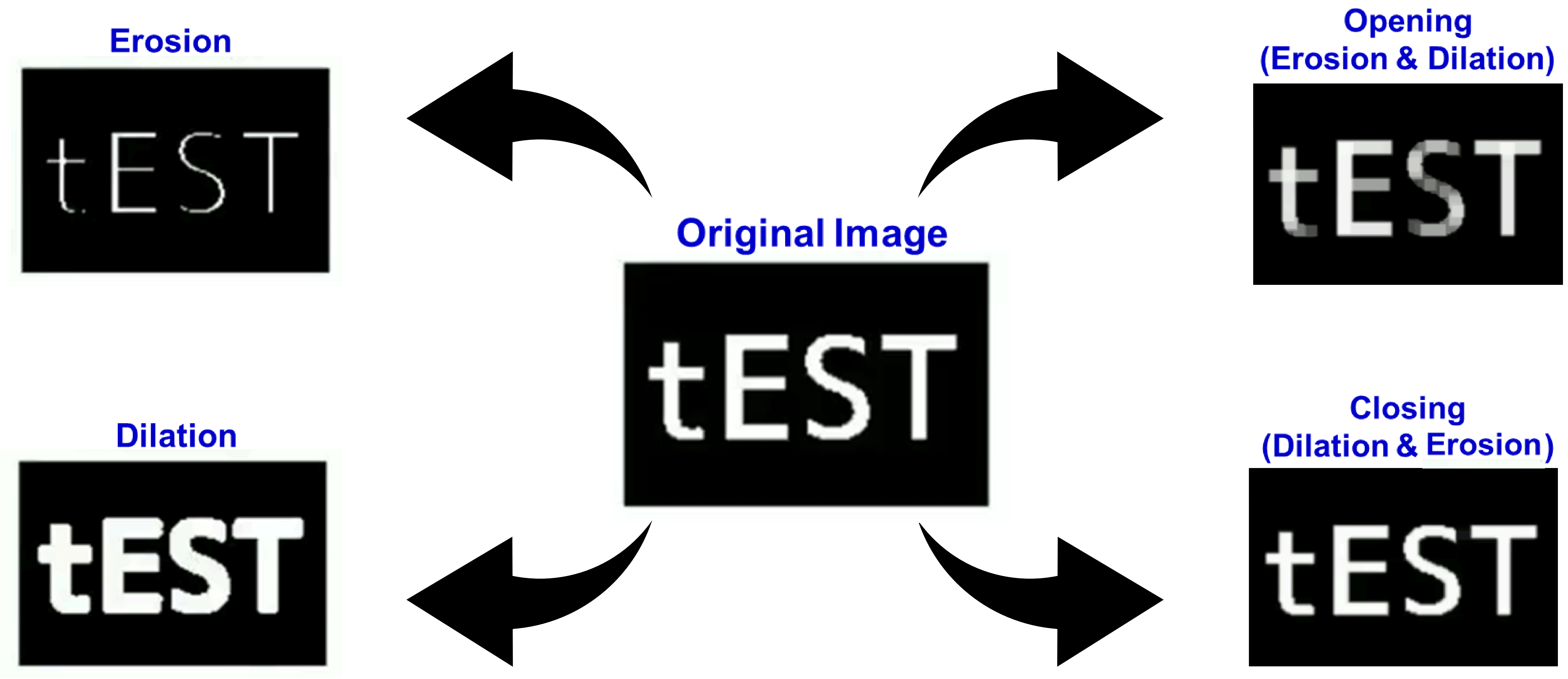
- To simplify the objects by
  - Filling in small holes
  - Eliminating small protrusions from their boundaries
- Boundary pixels
  - Object pixels that have background neighbors
    - Various definition of neighbor
- Support by structuring element



PREPROCESSING - MORPHOLOGICAL OPERATION

- **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
- **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
  - $R' = (R \oplus A) \ominus A$

PREPROCESSING - MORPHOLOGICAL OPERATION



DILATION - EXAMPLE

$$R' = R \oplus A$$

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)

$\oplus$

1	0	0
0	**1**	0
0	0	1

(b)

=

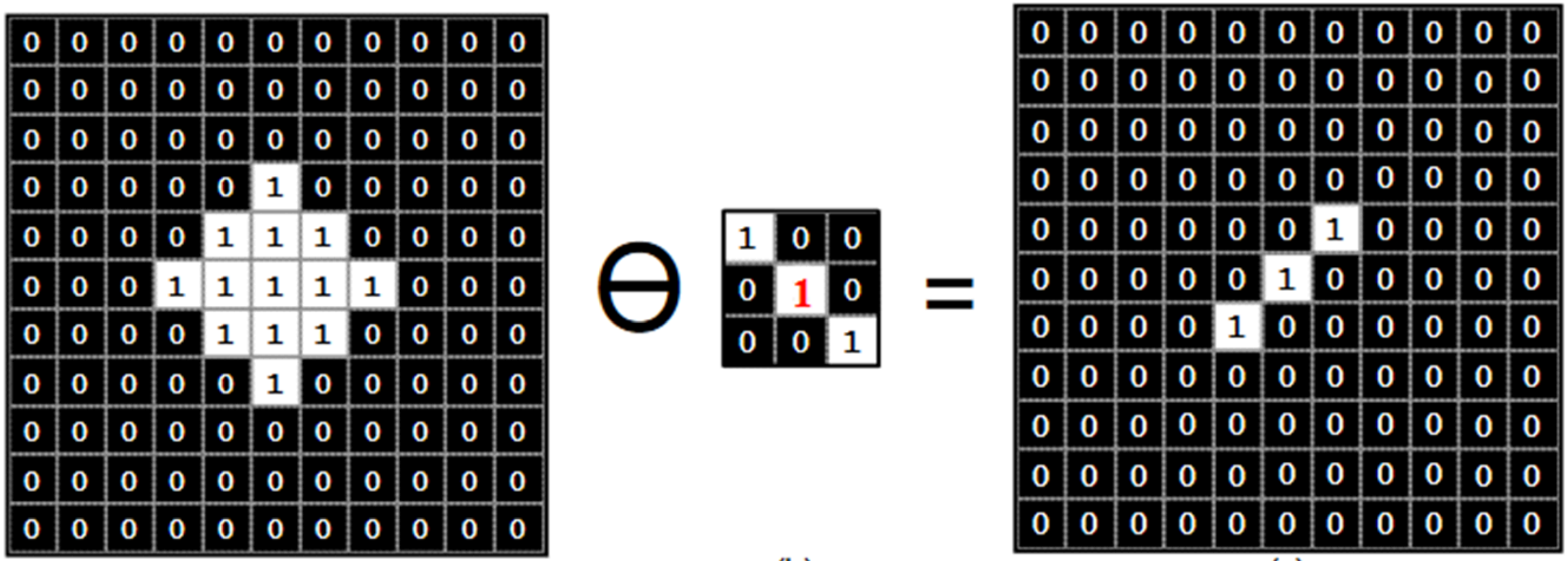
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	1	0	0	0
0	0	0	0	1	1	1	1	1	0	0
0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

(c)

Good in remove the hole and bridging the gaps.

EROSION - EXAMPLE

$R' = R \ominus A$



(a) (b) (c)

Good in removing irrelevant details.

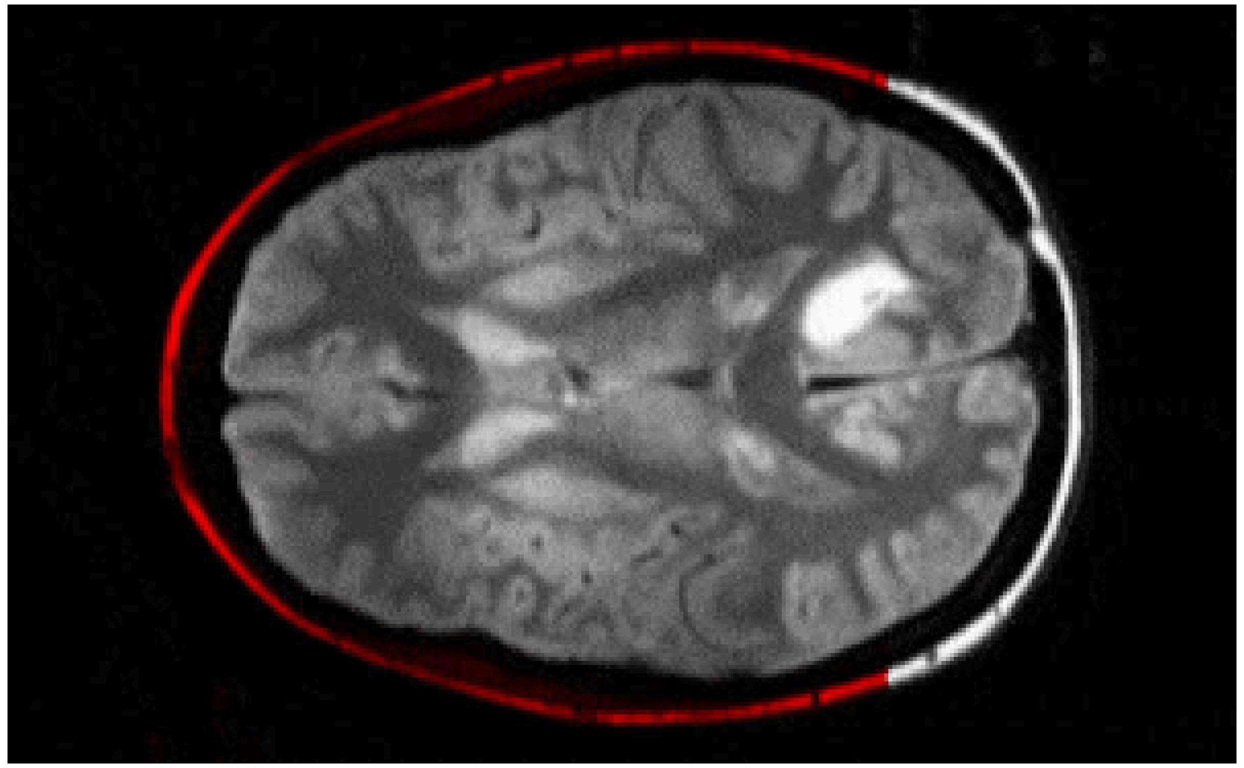
OPENING | CLOSING

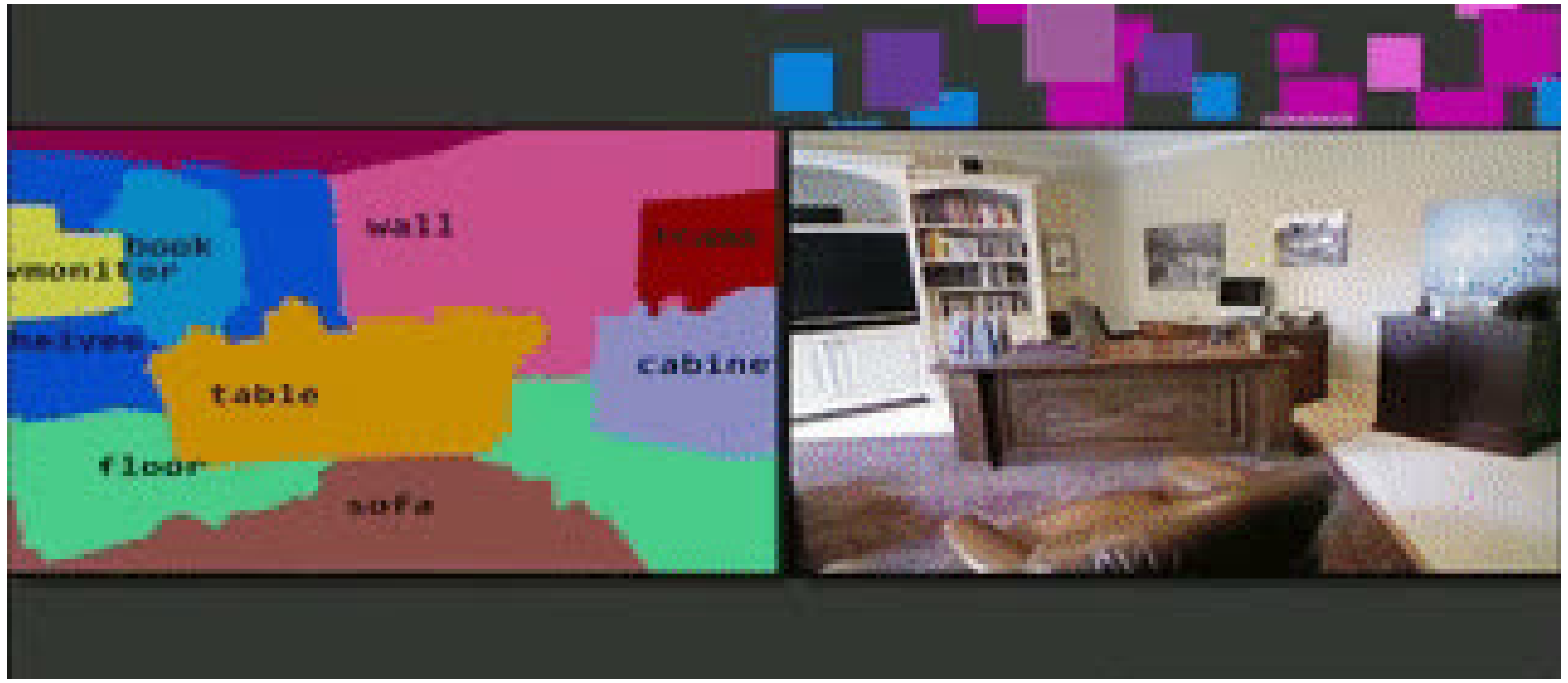
Opening

Closing

FEATURE EXTRACTION: SEGMENTATION

- Image segmentation is the process of <u>partitioning</u> 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.



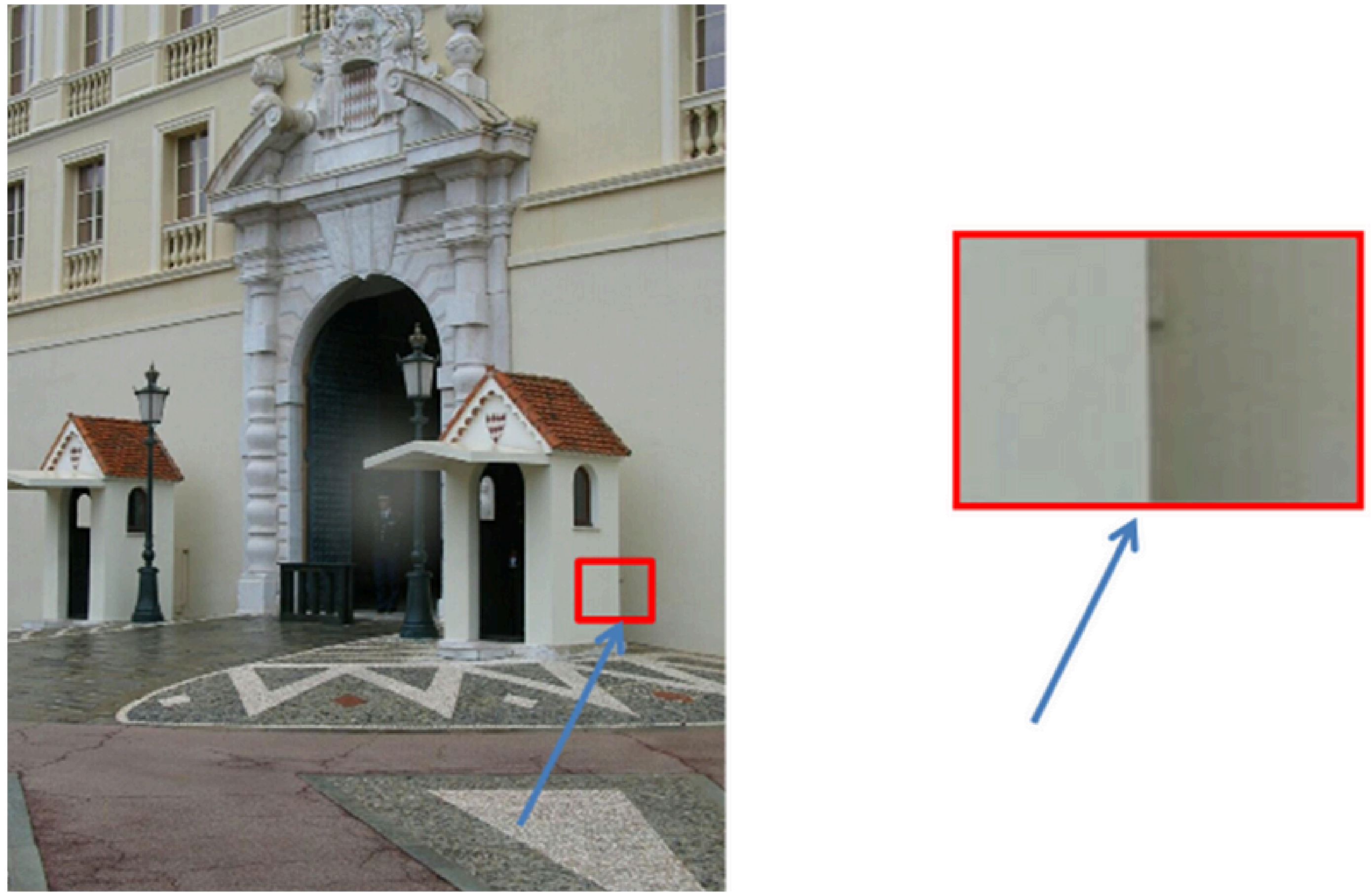


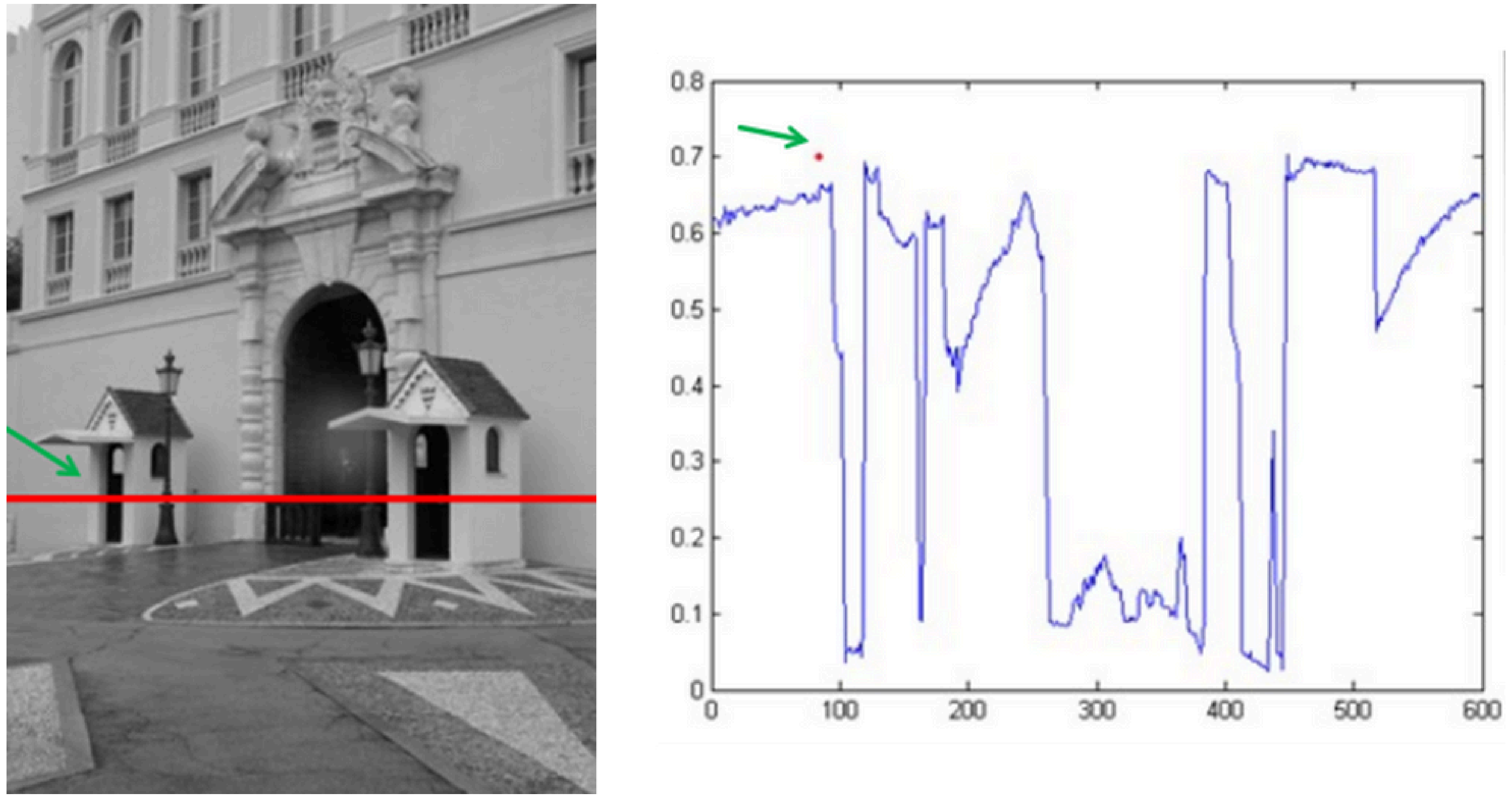
EDGE DETECTION

1. Identify the boundaries of objects within images.
2. It works by detecting discontinuities in brightness, which are areas where the image intensity changes sharply or has discontinuities

EDGE DETECTION

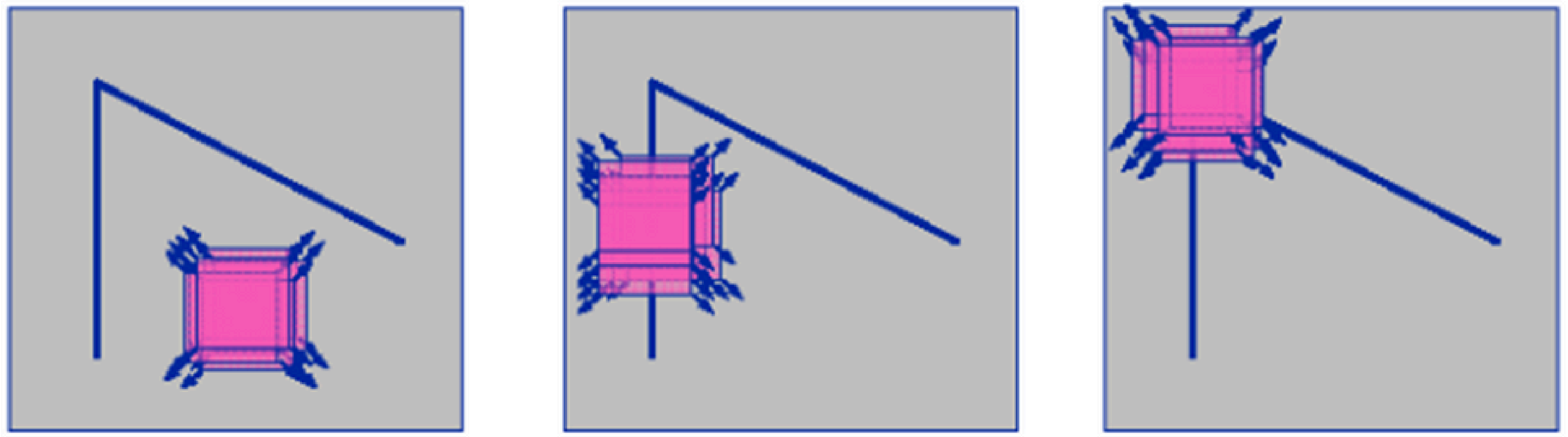
1. Edges are caused by a variety of factors
2. An edge is a place of rapid change in the image intensity function
3. Edge detection algorithms: a) Canny b) Prewitts c) Sobel





CORNER DETECTION

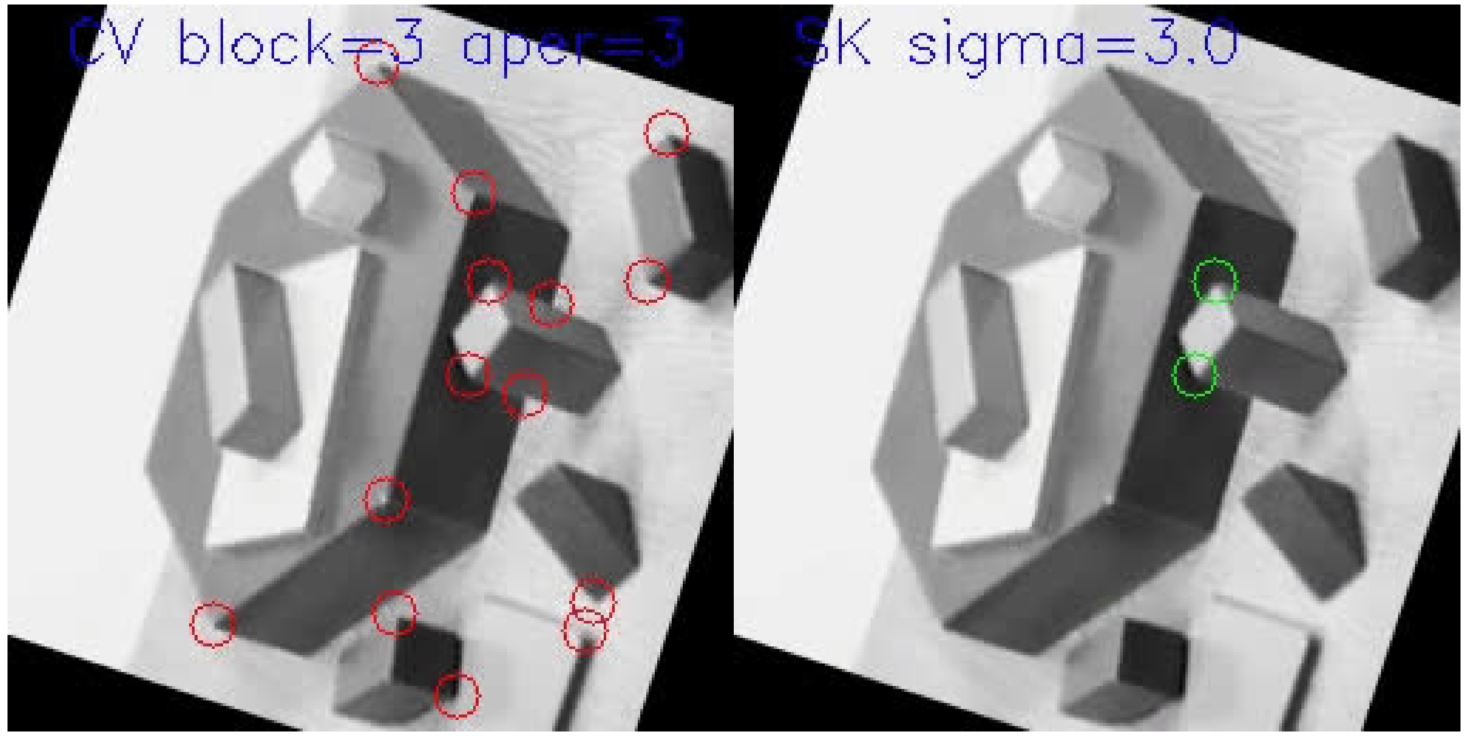
1. Shifting the window in any direction should yield a large change in appearance.
2. Of course, Harris corner detector gives a mathematical approach for determining which case holds.



"flat" region: no change in all directions

"edge": no change along the edge direction

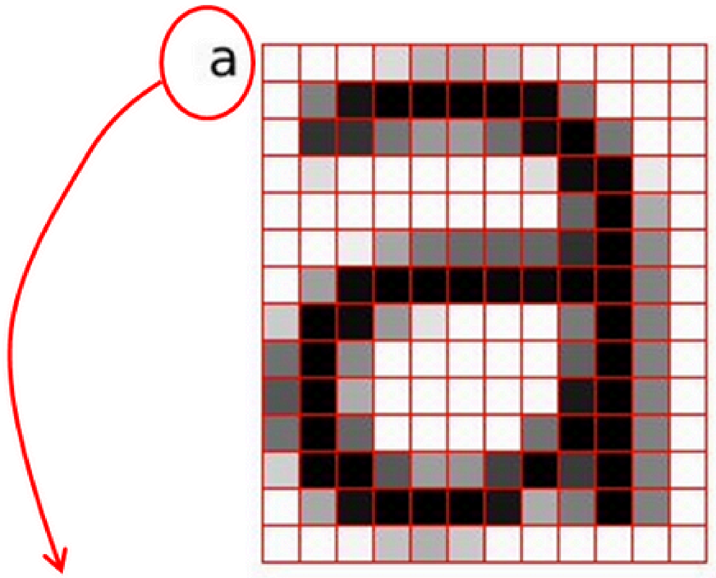
"corner": significant change in all directions



REPRESENTATION AND DESCRIPTION

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

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:

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

- 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.

IMAGE RECOGNITION PROCESS

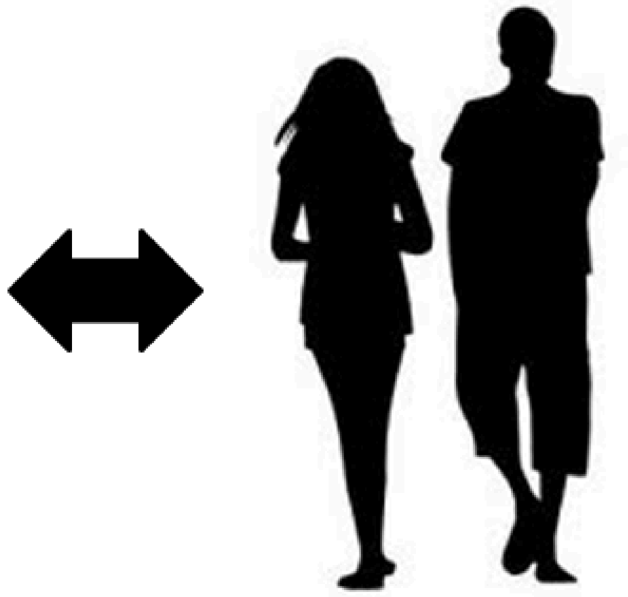
height

```
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

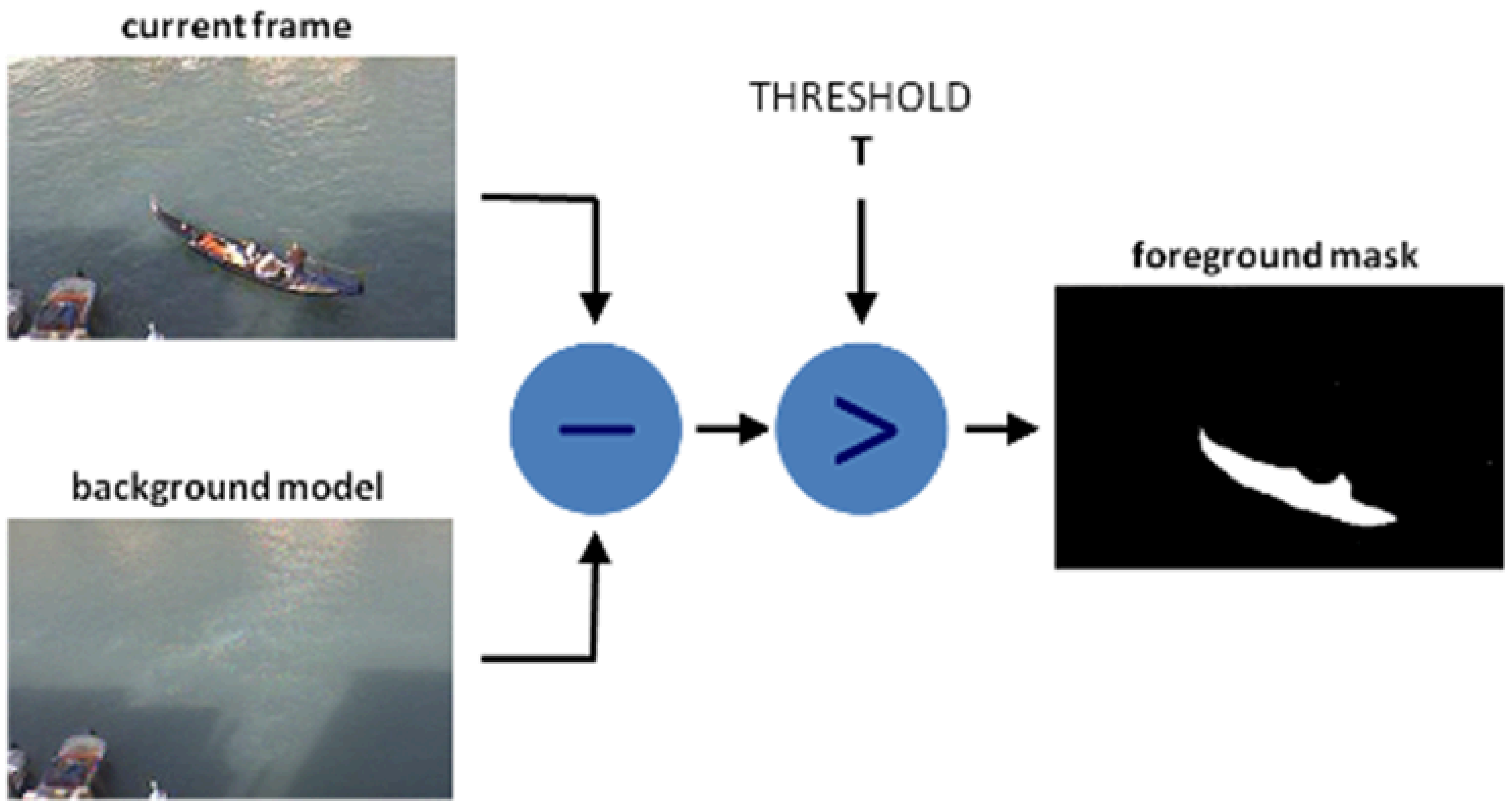
Recognition & Interpretation
(these usually involve classification as well)

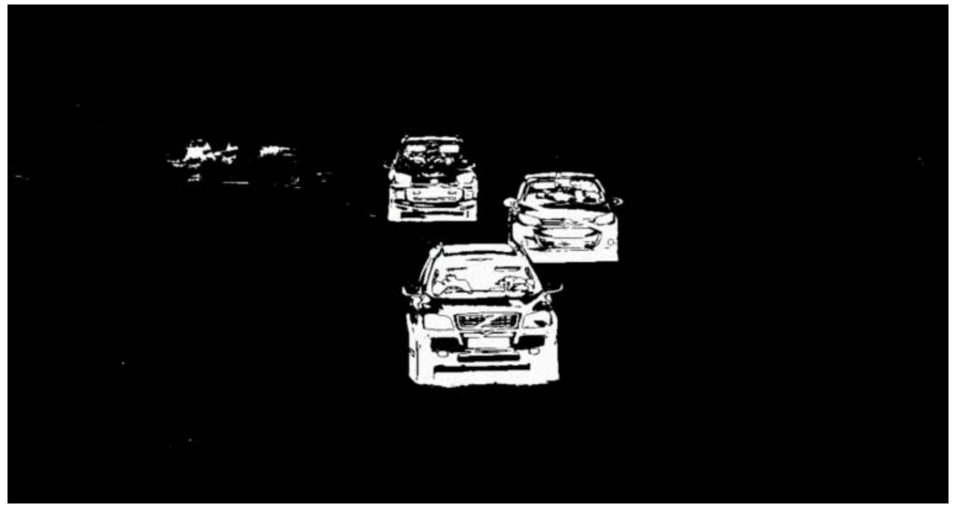
CHALLENGES OF IMAGE PROCESSING

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**

BACKGROUND SUBTRACTION

1. 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).
2. Generally an image's regions of interest are objects (humans, cars, text etc.) in its foreground.





MOTION TRACKING

A 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

EXAMPLE OF STEPS

1. Video acquisition
2. Video slicing
3. Preprocessing (E.g. background subtraction, image enhancement, etc.)
4. Feature extraction (E.g. motion features, edge, corner, etc.)
5. Perform Machine Learning (Training)
6. Perform classification / recognition / interpretation

THE END

NEXT LECTURE

Expert Systems