Artificial Intelligence

Computer Vision

COMP3411/9814

Week 7, Term 3, 2025

Maryam Hashemi (m.Hashemi@unsw.edu.au)

Lecture Overview

- **Introduction**
- Image Processing
  - Histogram Equalization
  - Noise Removal
  - Edge Detection
- Computer Vision
  - Convolutional Neural Networks
    - ✓VGG16
    - ✓AlexNet
- Cognitive Vision

Introduction

What is computer vision?

What is image processing?

What is the difference?

Introduction

Computer vision is the field of AI that enables **machines to interpret and understand visual data** (images, videos) to perform tasks like object detection, image classification, and scene analysis.

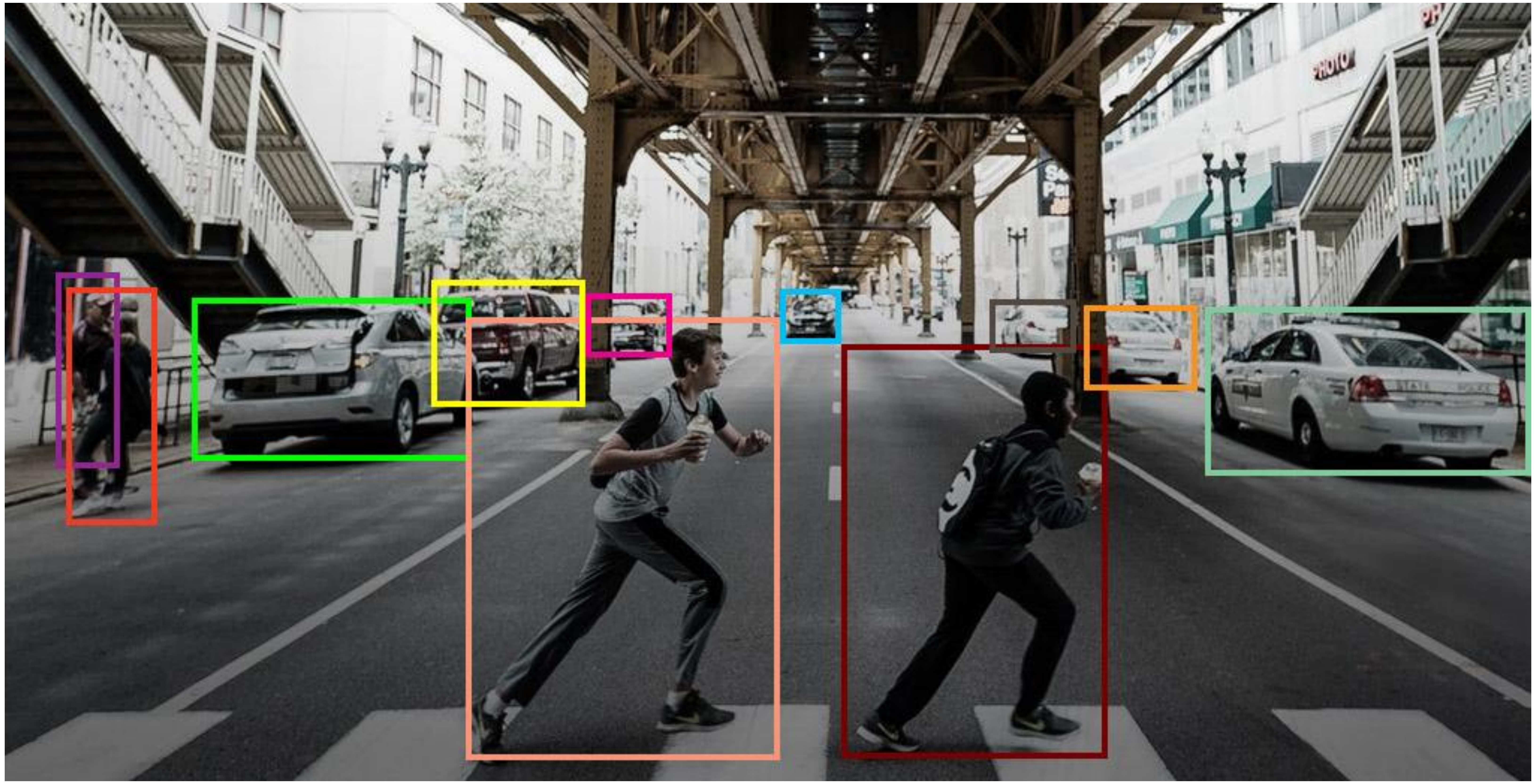
Image processing involves applying algorithms to **transform or manipulate images, focusing on enhancing or modifying the visual content** (e.g., filtering, noise reduction).

Computer vision aims to interpret and understand images, while image processing focuses on improving or transforming images.

CV is a subset of AI but image processing is a technique that might be used in AI approaches.

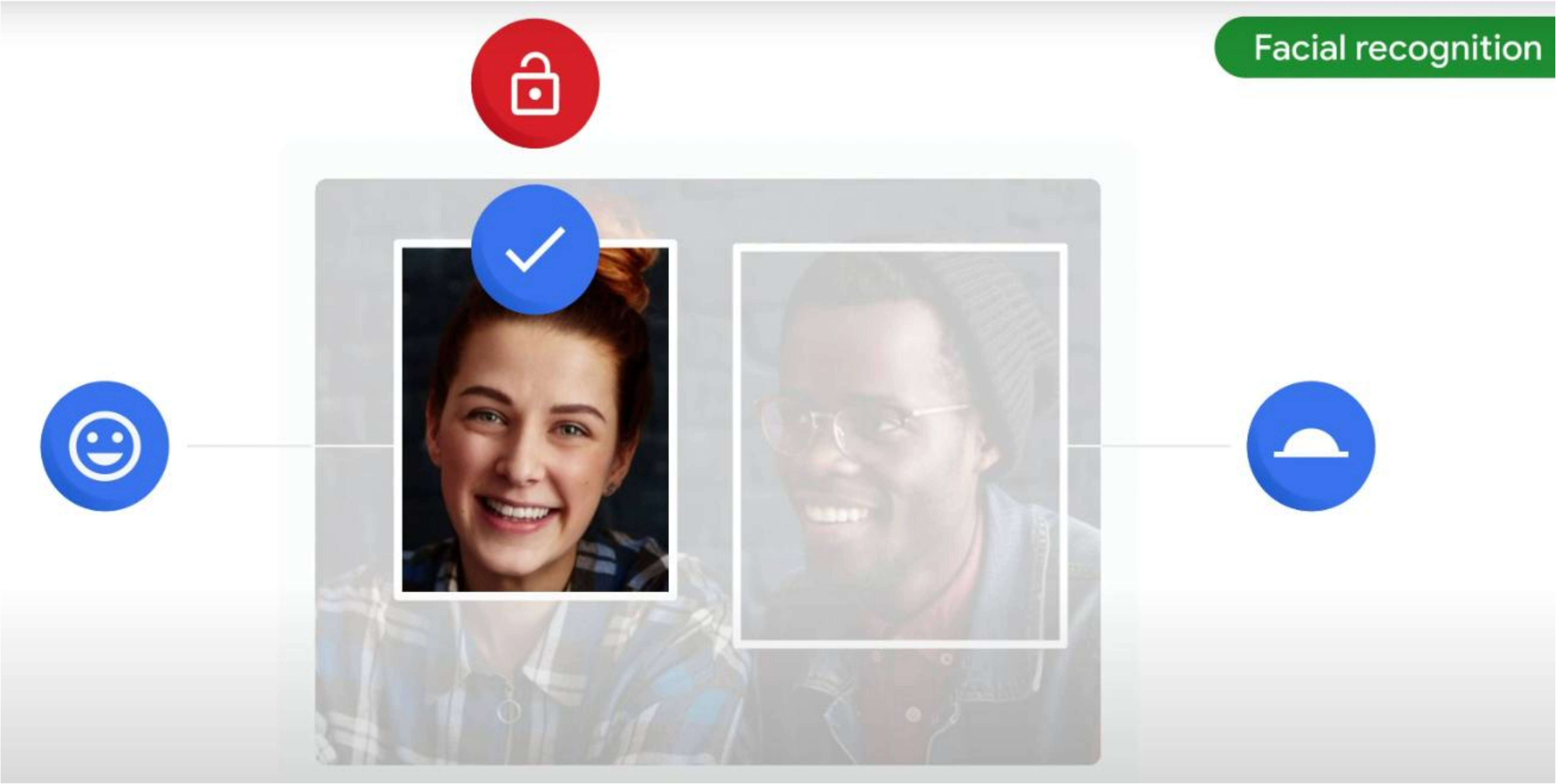
Introduction

Example of computer vision tasks: Object detection and localization



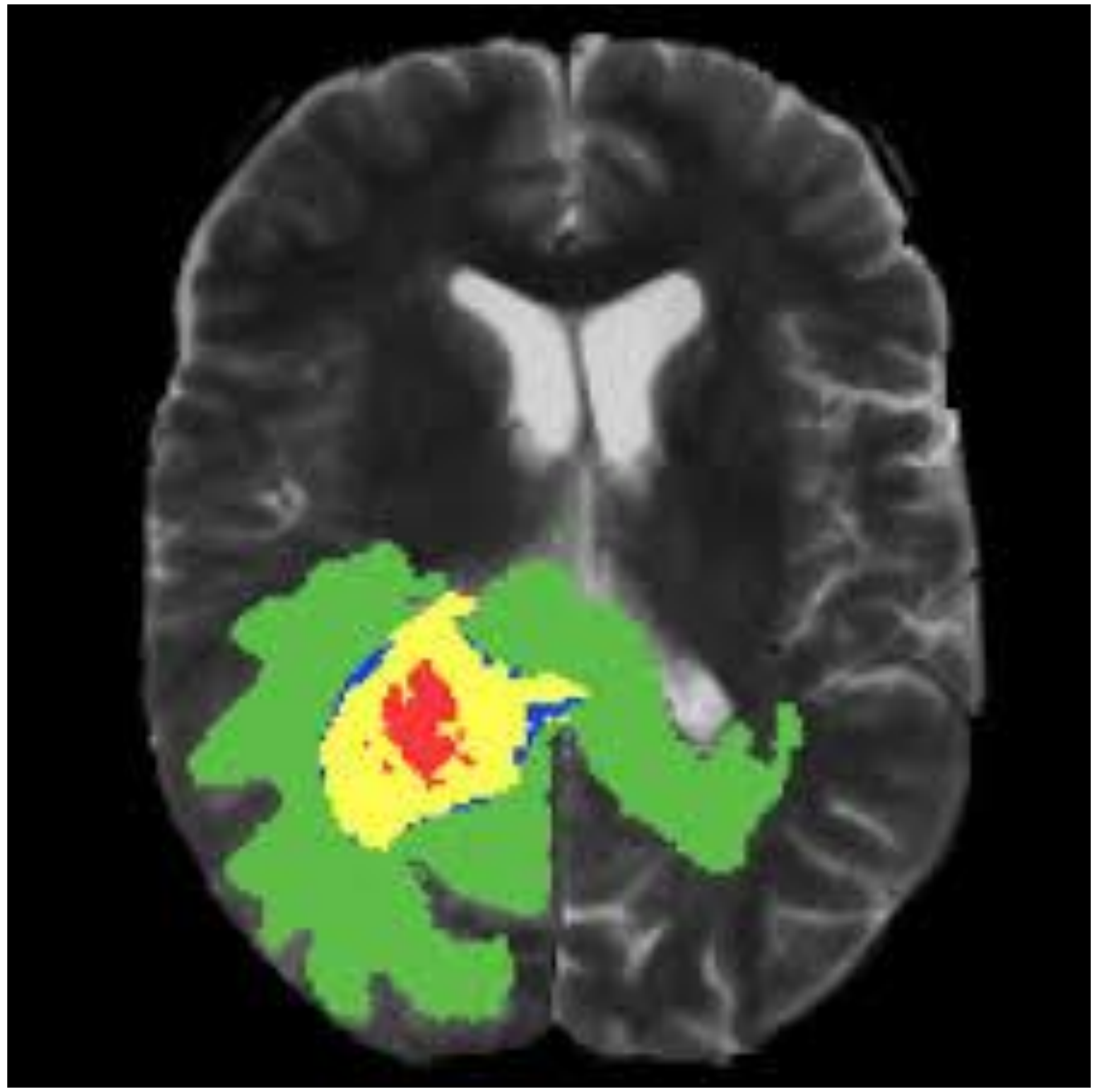
Introduction

Example of computer vision tasks: Facial Recognition



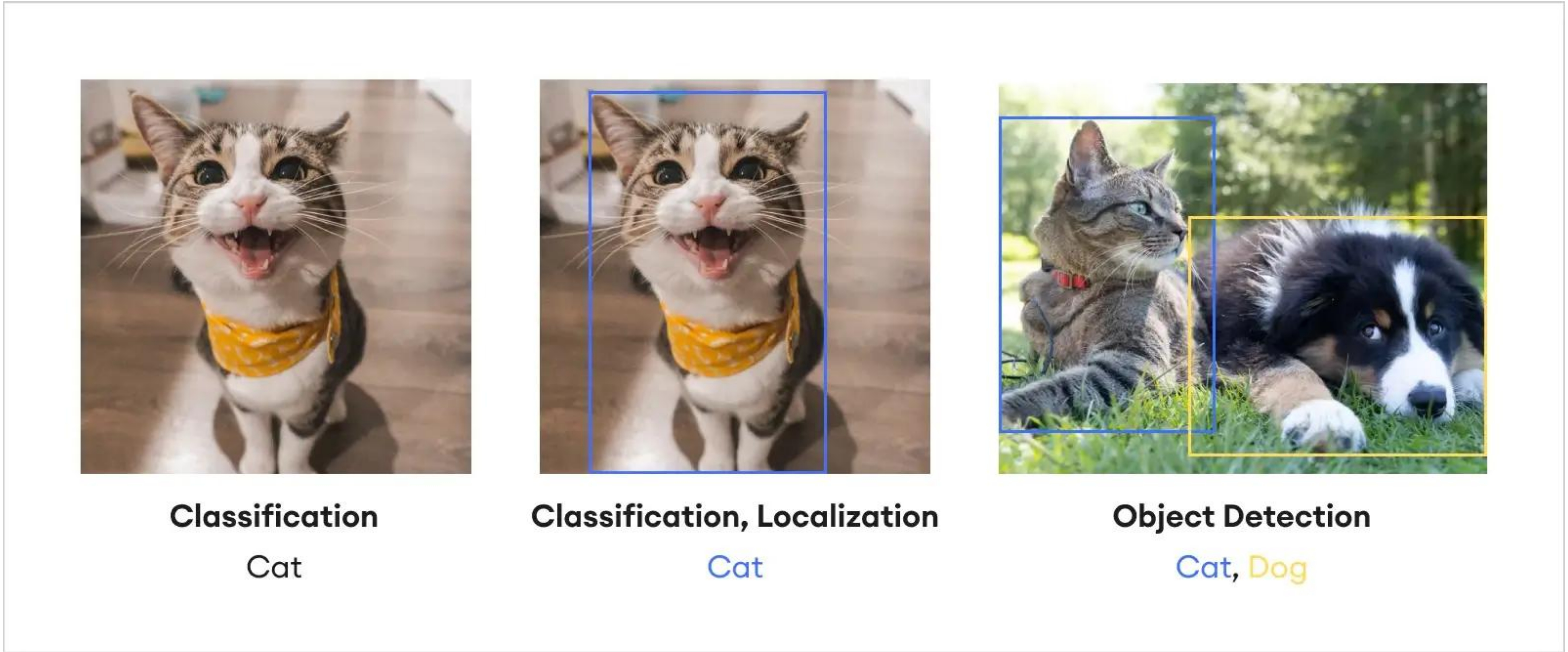
Introduction

Example of computer vision tasks: Segmentation



Introduction

Example of computer vision tasks: Classification

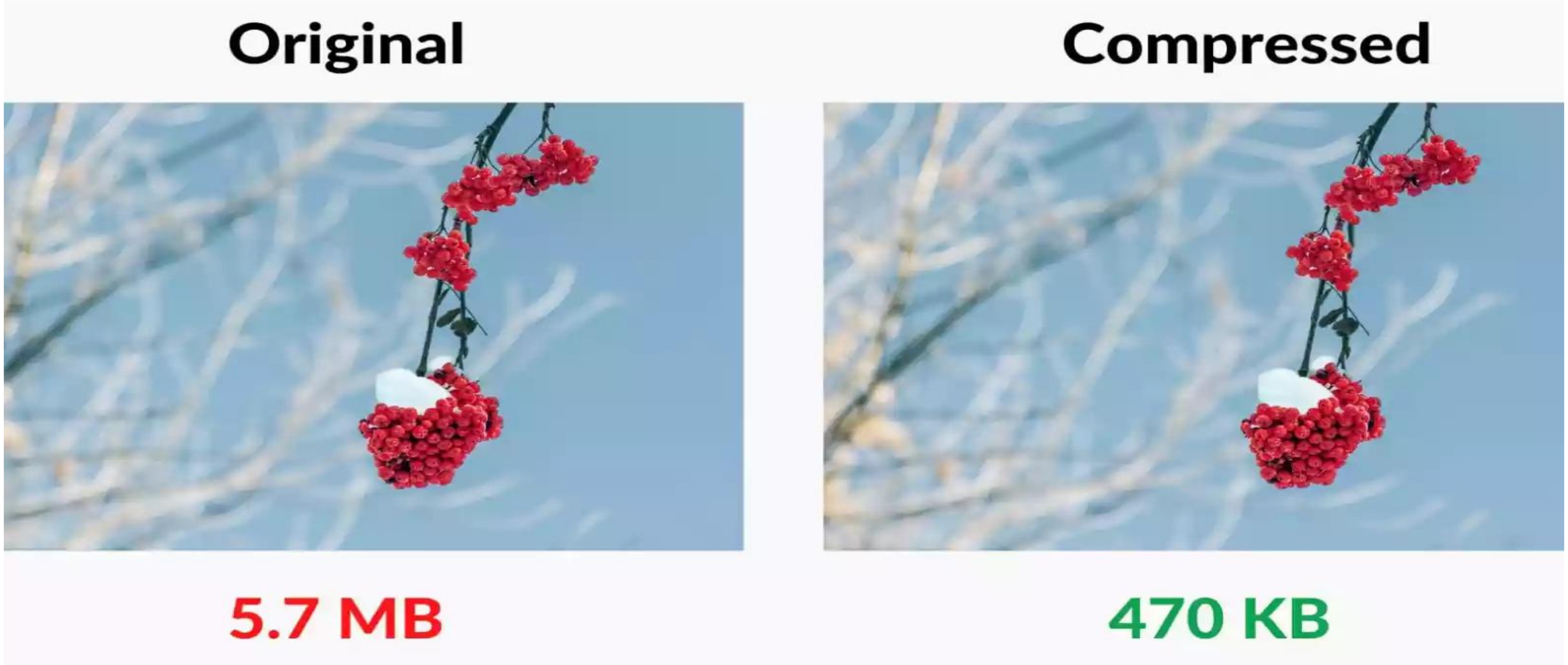


Introduction

Example of image processing tasks: Image Enhancement

Introduction

Example of image processing tasks: Image Compression



Introduction

Effortless for humans, but a difficult problem for machines:

- Variable and uncontrolled illumination
- Shadows
- Complex and hard-to-describe objects
    - ➢Objects from outdoor scenes
    - ➢Non-rigid objects
    - ➢Objects occluding other objects

Introduction

- State of computer vision → The general computer vision problem is unsolved
  - Develop a visual system as good as humans. No progress for 40 years.

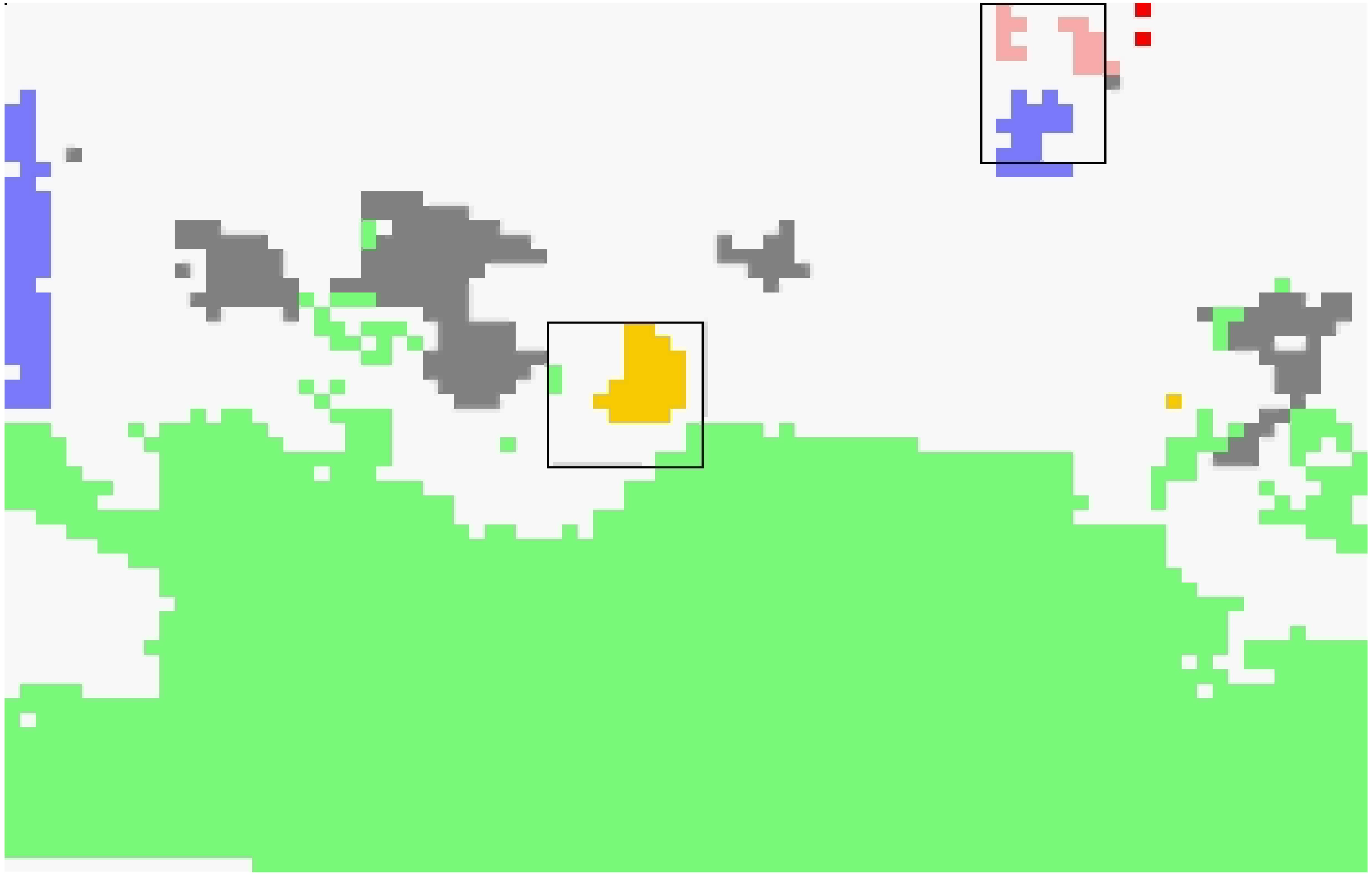
- A lot of progress in specific computer vision problems
  - e.g. face recognition used in digital cameras, surveillance, security.
  - e.g. pick and place.

Introduction

Computer vision in action

Introduction

What the robot sees

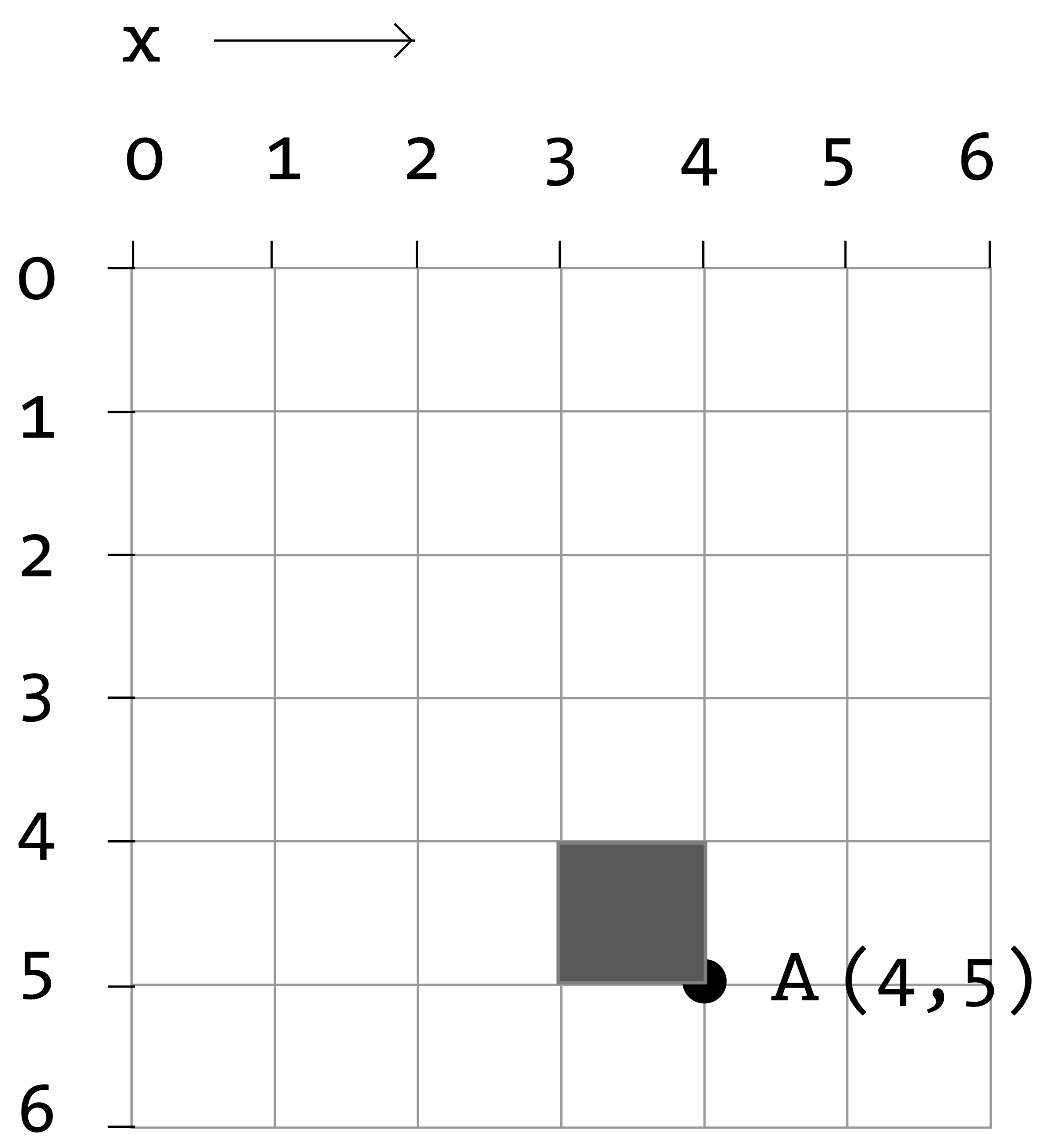


Lecture Overview

- Introduction
- **Image Processing**
  - Histogram Equalization
  - Noise Removal
  - Edge Detection
- Computer Vision
  - Convolutional Neural Networks
    - ✓ VGG16
    - ✓ AlexNet
- Cognitive Vision

Image Processing

- A computer sees digital images as a matrix, where each cell represents a pixel.
- Each pixel (or cell) has a value that indicates the intensity or color of that pixel.



Example: A(4, 5)= 120

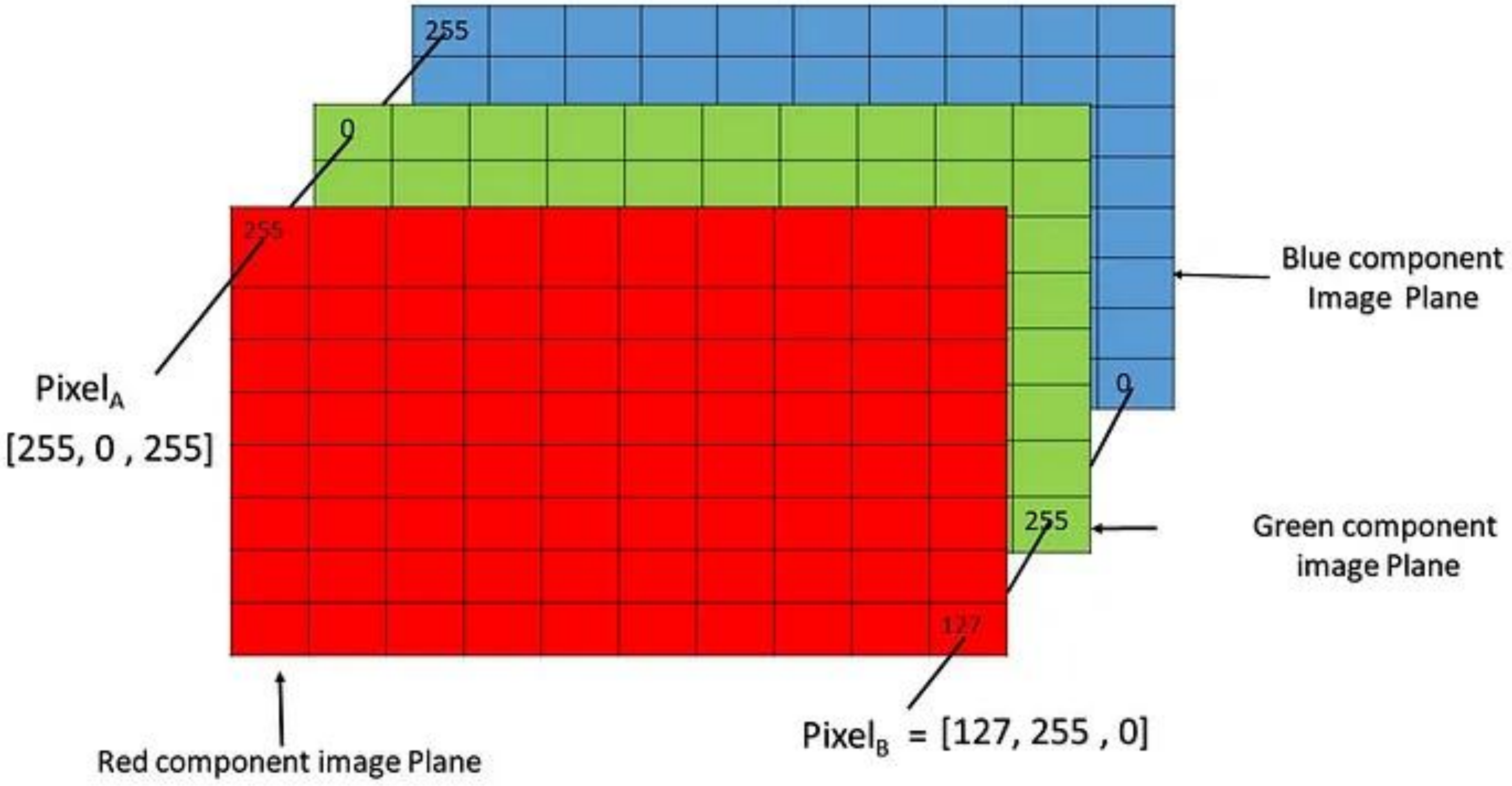


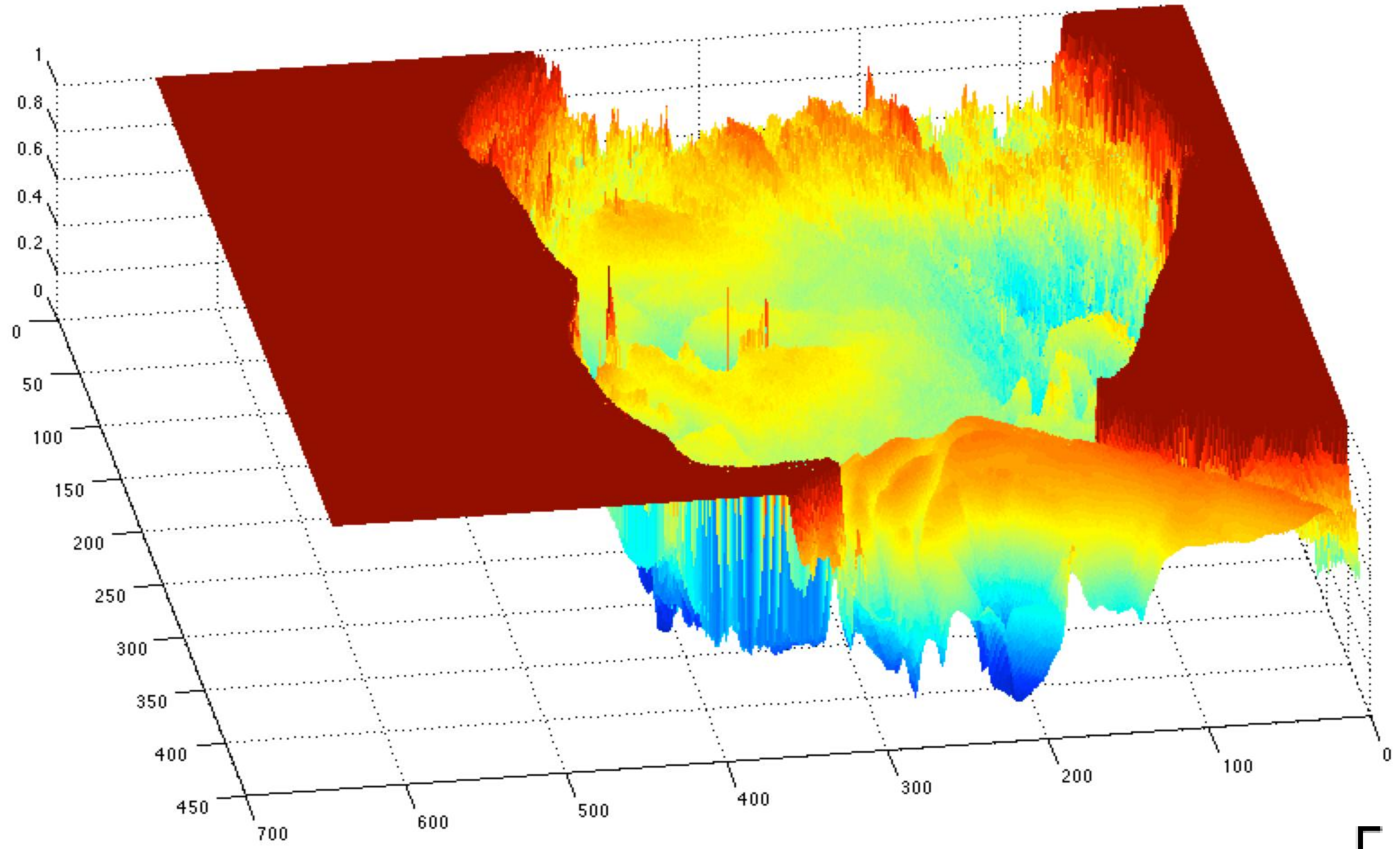
Image Processing

- Aims to enhance relevant information in the image.
- Usually is used to prepare images for further analysis and interpretation.
- Image represented as n x m array I(x,y) → image intensity array.
- Cells are called pixels. Each number represent light intensity.
- We will discuss two important techniques: **histogram equalization**, noise removal, and edge detection.

Image Processing: Histogram Equalization

- A (grayscale) image is a 2D function. In the provided example, we have 450 * 700 pixels, and each pixel can have a value between [0, 1] (or [0, 255] if we are using 8-bit presentation).

$$f(\boldsymbol{x})$$



domain $\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

Image Processing: Histogram Equalization

- An image histogram is a plot of the gray-level frequencies (i.e., the number of pixels in the image that have that gray level).

0	0	1	0	2	0
1	0	7	7	7	0
0	7	0	0	7	0
1	0	0	7	2	0
0	0	7	1	0	1
1	0	7	7	7	0

frequencies

f(0)= 18

f(1)= 6

f(2)= 2

f(3)=f(4)=f(5)=f(6) = 0

f(7) = 10

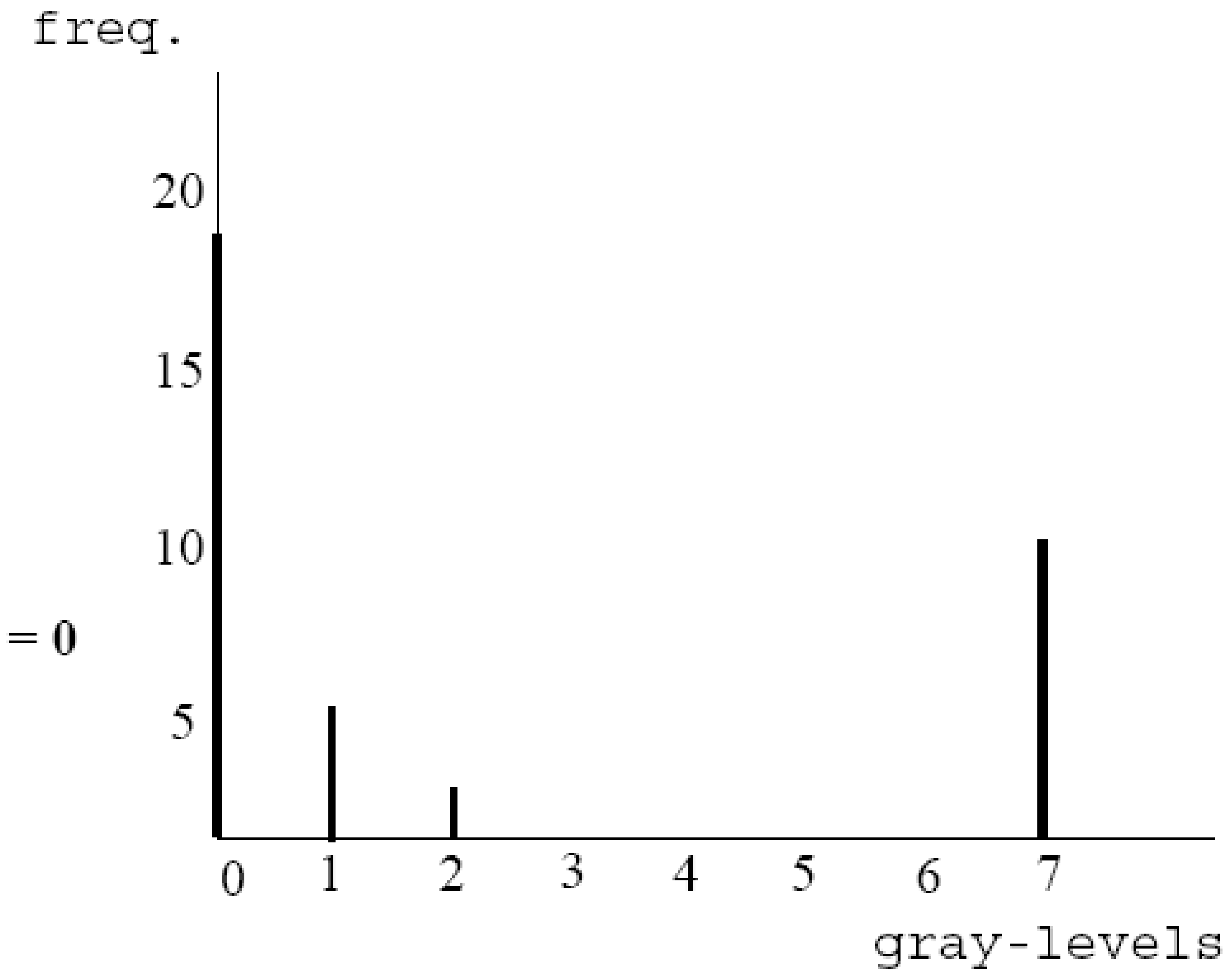


Image Processing: Histogram Equalization

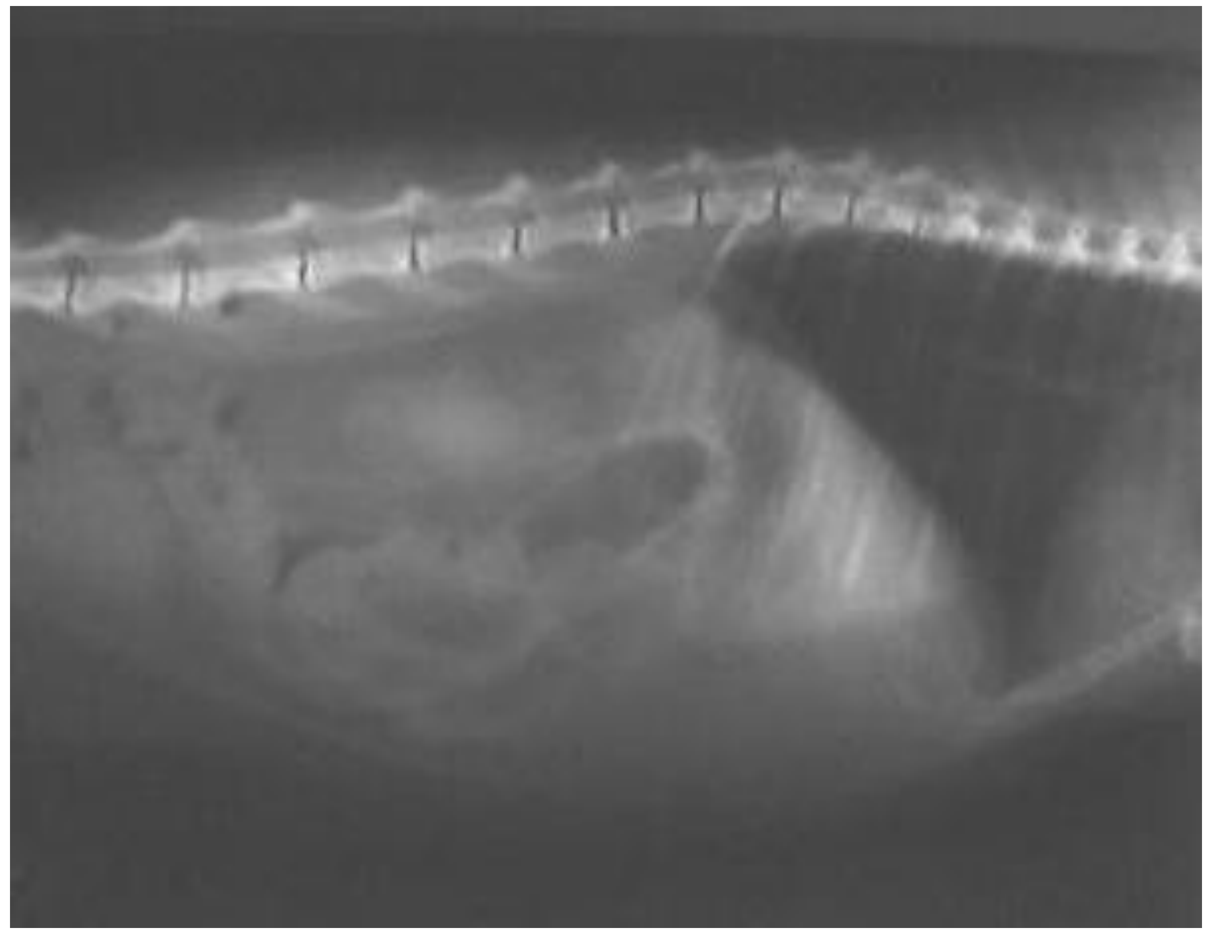
- To represent the histogram as probabilities (i.e., a normalized histogram), divide the frequency of each gray level by the total number of pixels.

- L is the total number of gray levels (typically 256), n_i is the number of pixels with value i, and n is the total number of pixels.

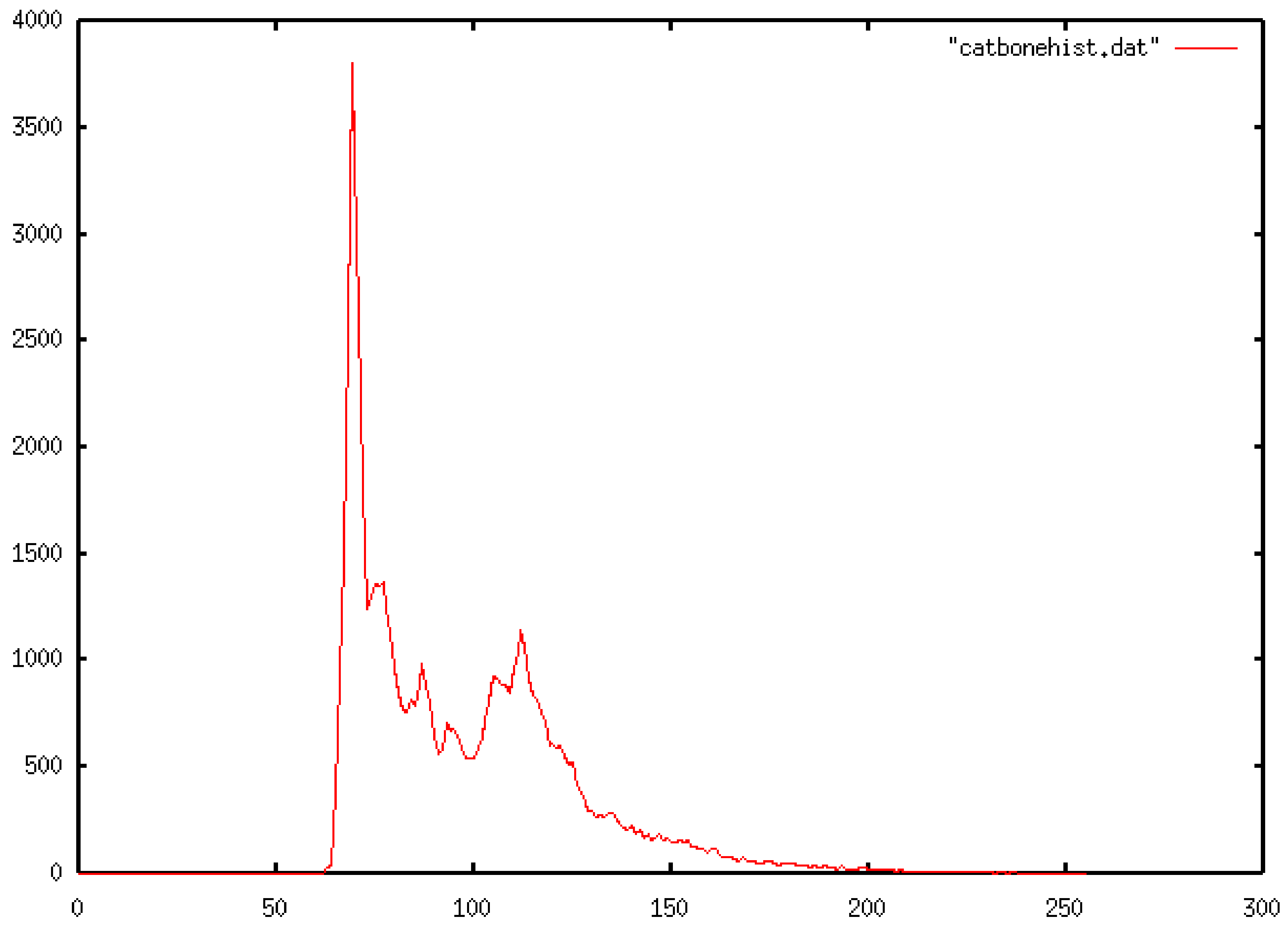
$$p_x(i) = \frac{n_i}{n},\ 0 \le i < L-1$$

$$P(0)= \frac{f(0)}{36} = \frac{1}{2}\ ,\ P(1)= \frac{f(1)}{36} = \frac{1}{6}\ ,\ P(2)= \frac{f(2)}{36} = \frac{1}{18}\ ,\ P(3)= P(4)= P(5)= P(6)= 0,\ P(7)= \frac{f(7)}{36} = \frac{5}{18}$$

Image Processing: Histogram Equalization



Example:

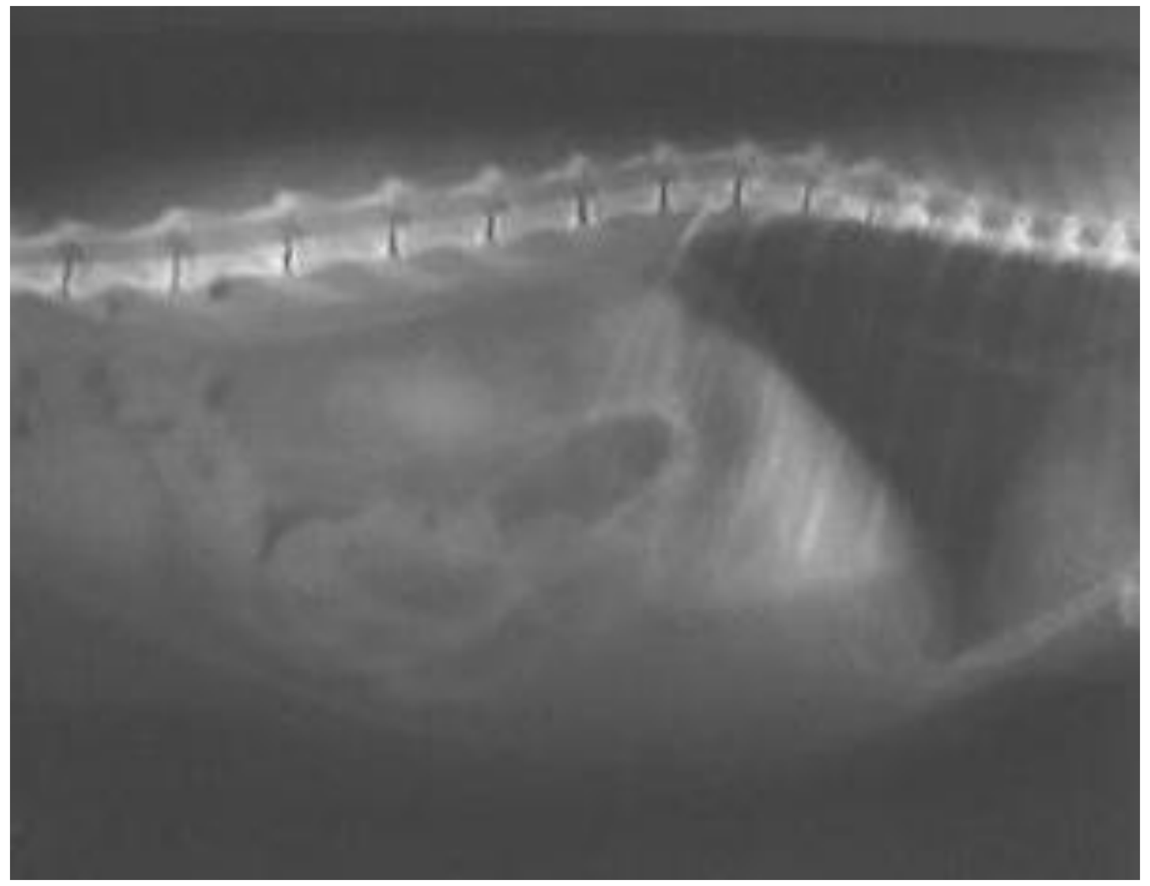


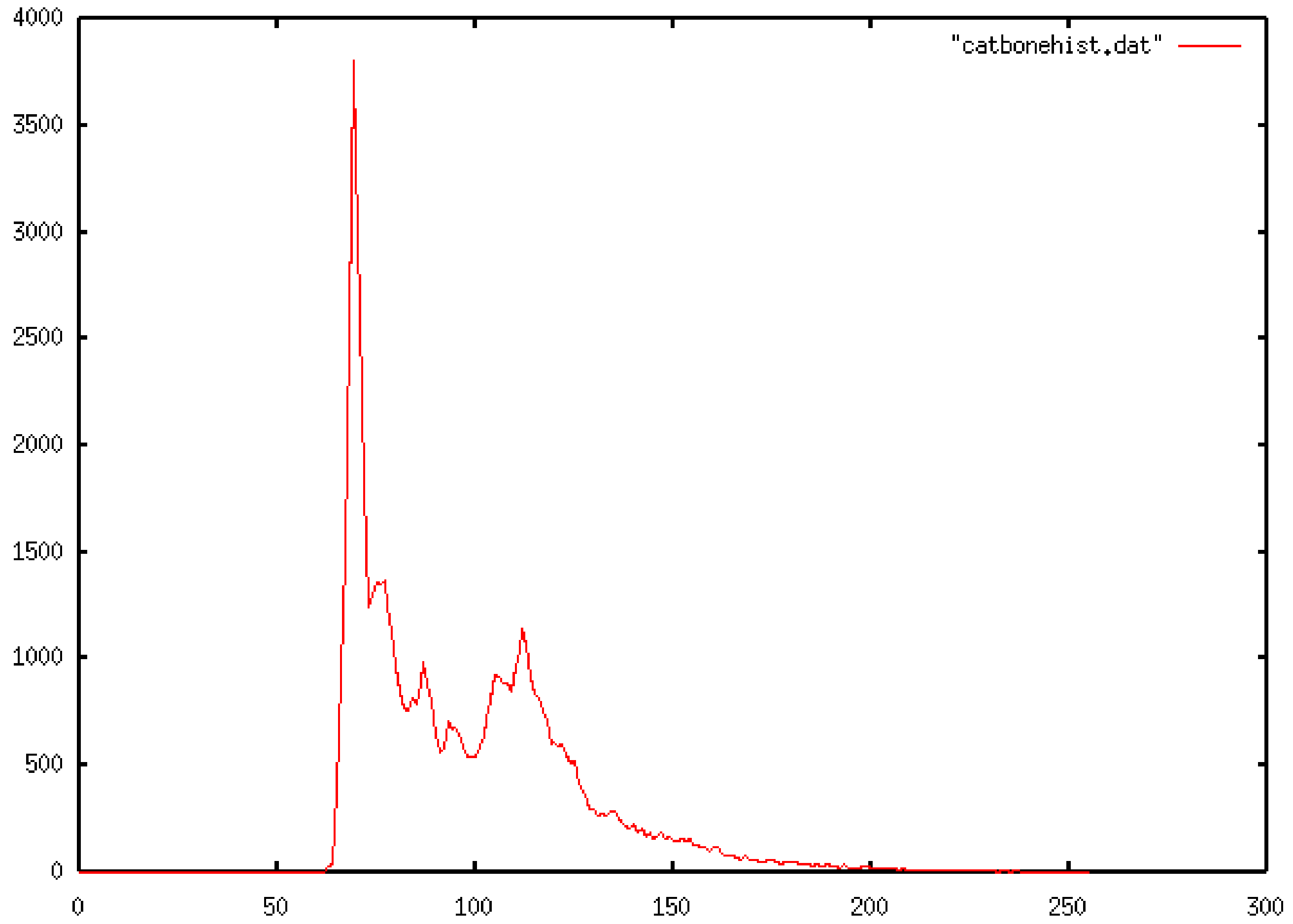
Histogram

Question: Could different images have exactly the same histogram?

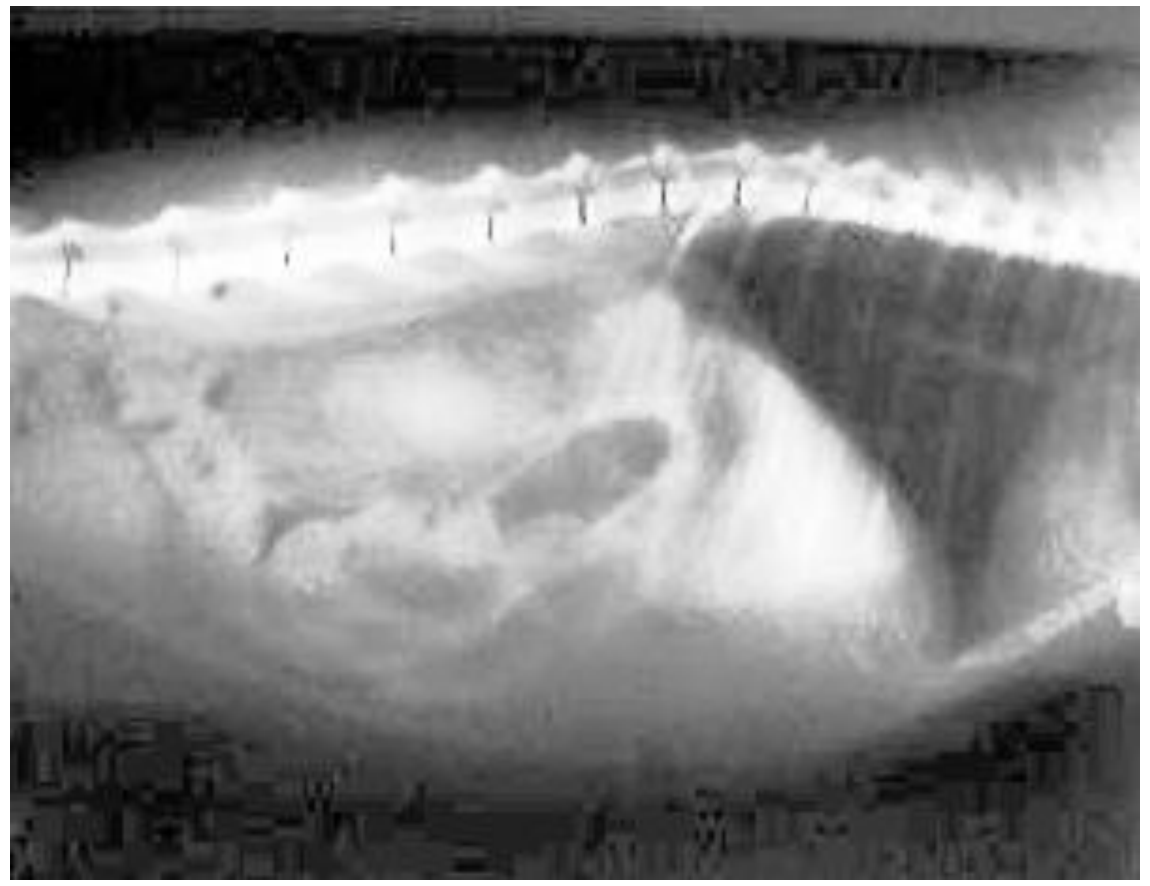
Image Processing: Histogram Equalization

Low contrast





High contrast



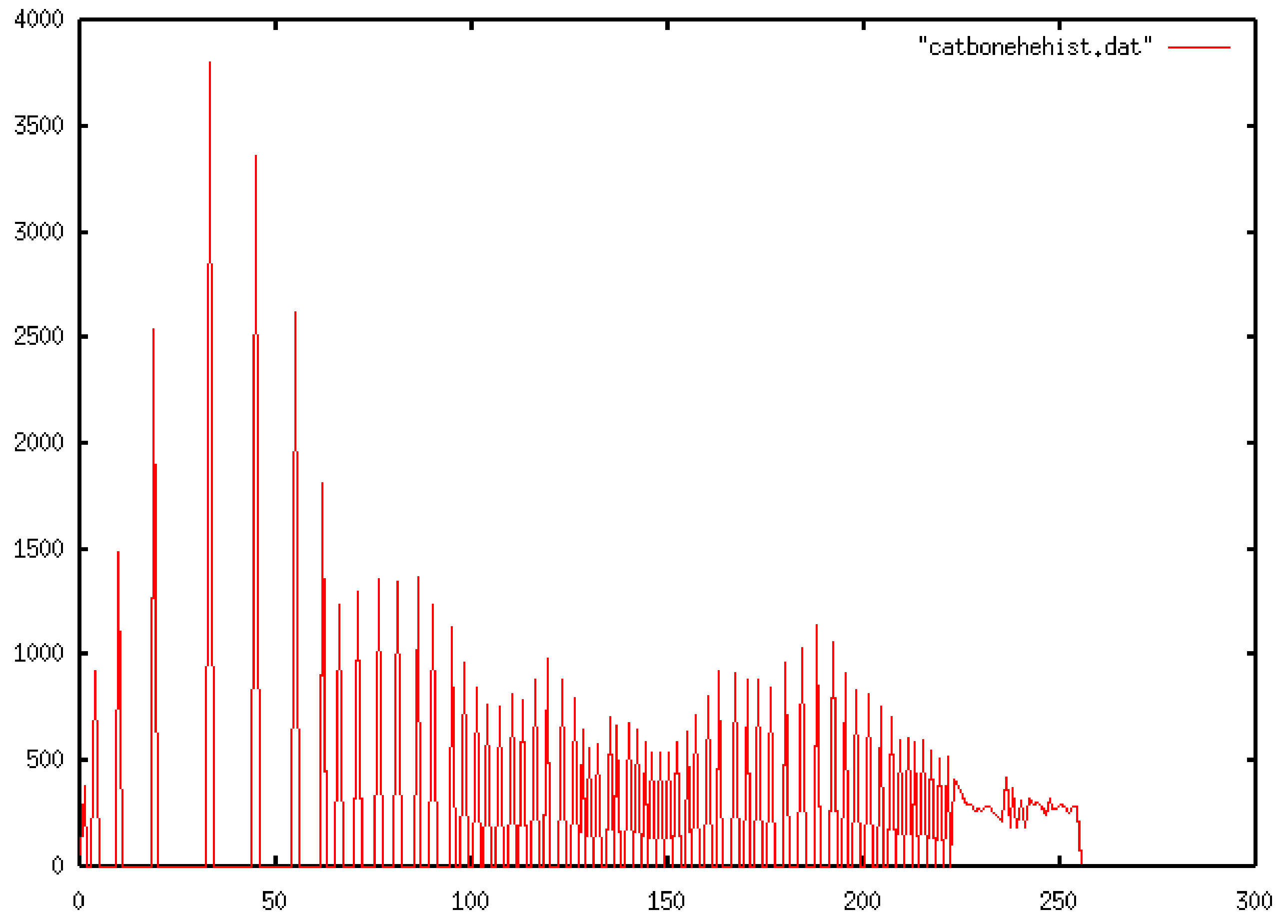


Image Processing: Histogram Equalization

- In histogram equalization, the main idea is to redistribute the gray-level values uniformly.

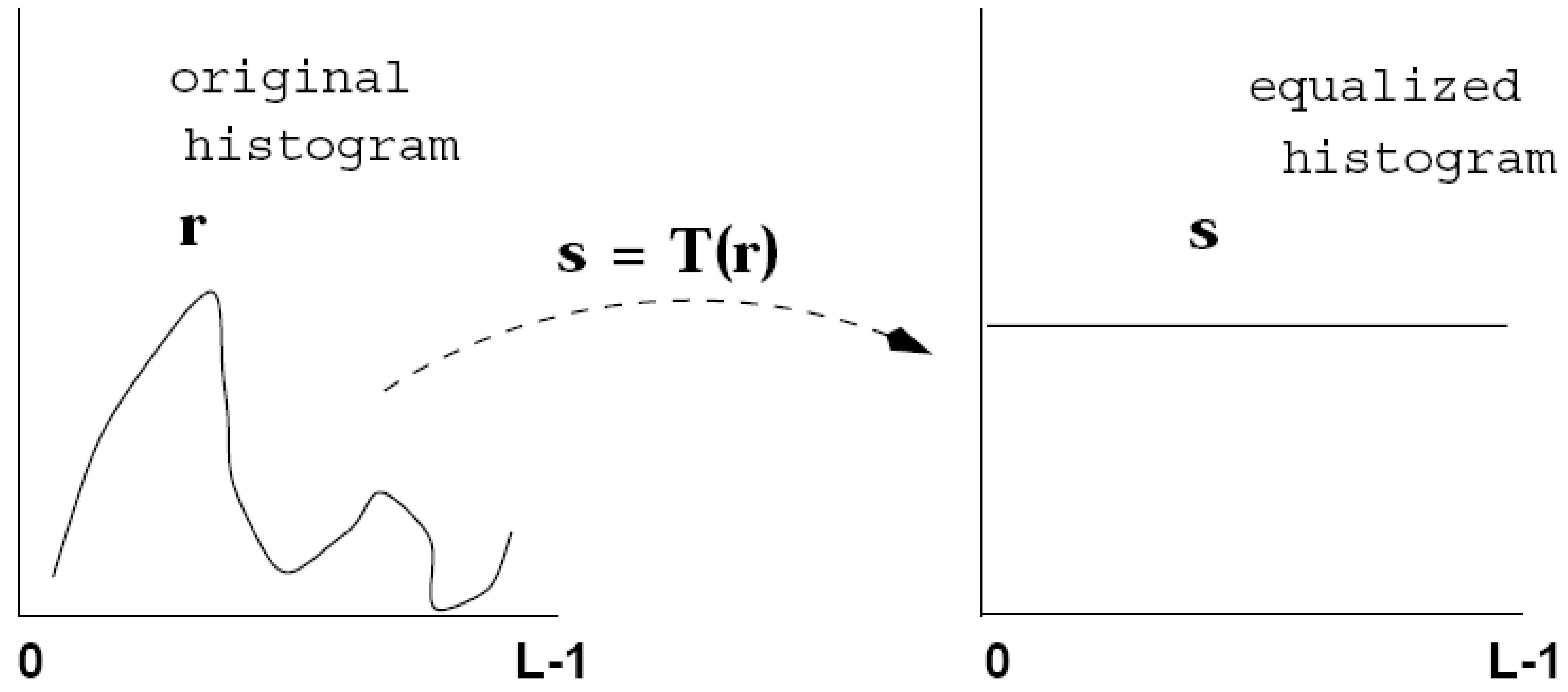


Image Processing: Histogram Equalization

We would like to create a transformation $T : [\, 0 \,,\, L - 1 \,] \rightarrow [\, 0 \,,\, L - 1 \,]$ to produce a new image Y from the input image X, with a flat histogram.
Such an image would have a linearized cumulative distribution function (CDF) across the value range.
Cumulative Distribution Function (CDF) of pixels in image X and Y:

$$cdf_X(i) = \sum_{j=0}^{i} p_x(j)$$

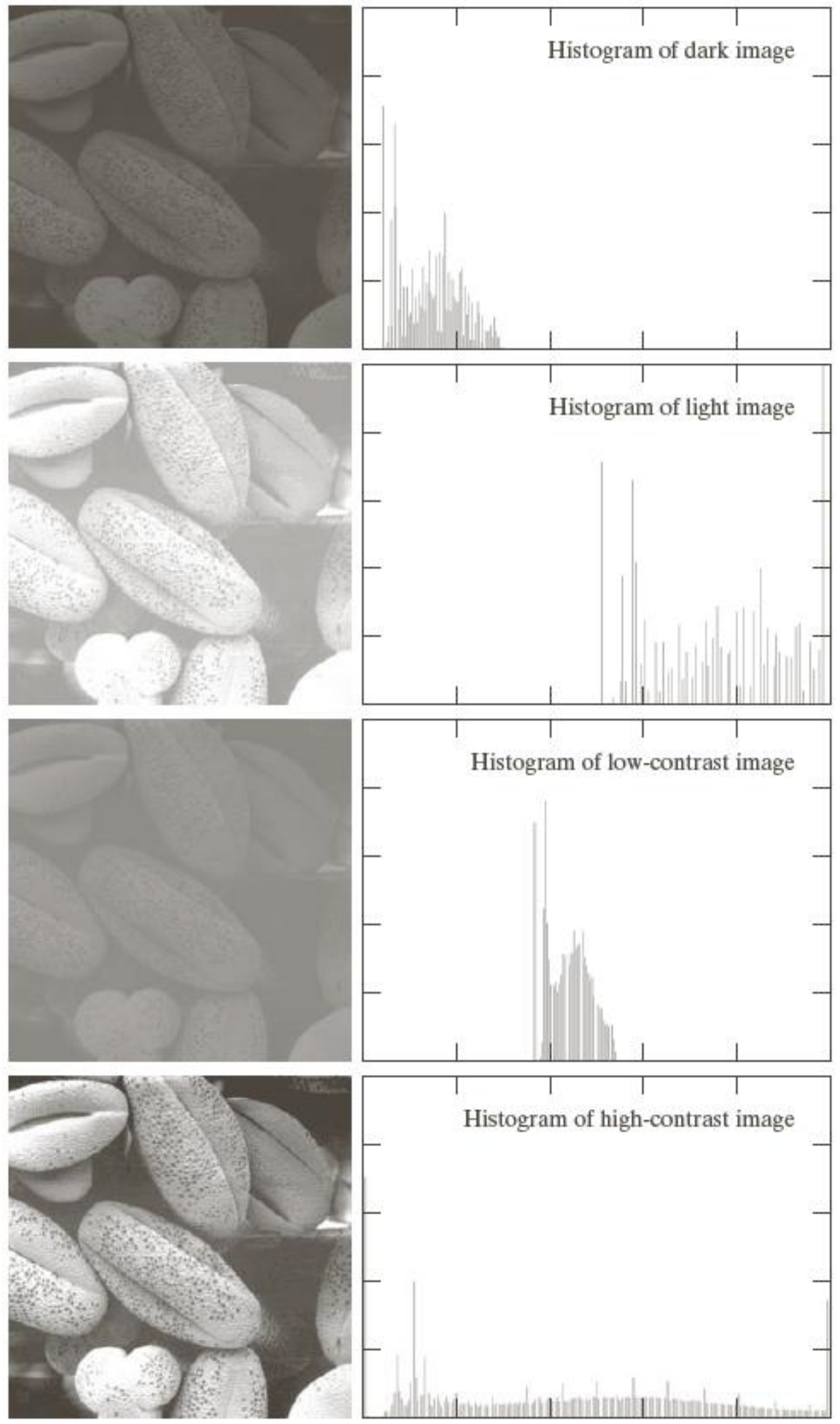
$$cdf_Y(\, i \,) = (\, i + 1 \,)K, \quad \text{for } 0 \leq i < L \text{ for some constant } K.$$

So, the transfer function can be defined as:

$$\mathbf{T(\, i \,) = round\left(\frac{cdf_X\,(\, i \,) - cdfmin}{1 - cdfmin} \times (L-1)\right)} \xrightarrow{\text{if } \mathrm{cdf_{min}}=0} \mathbf{round(cdf_x(i) \times (L-1))}$$

Image Processing: Histogram Equalization

original images and histograms



equalized images and histograms

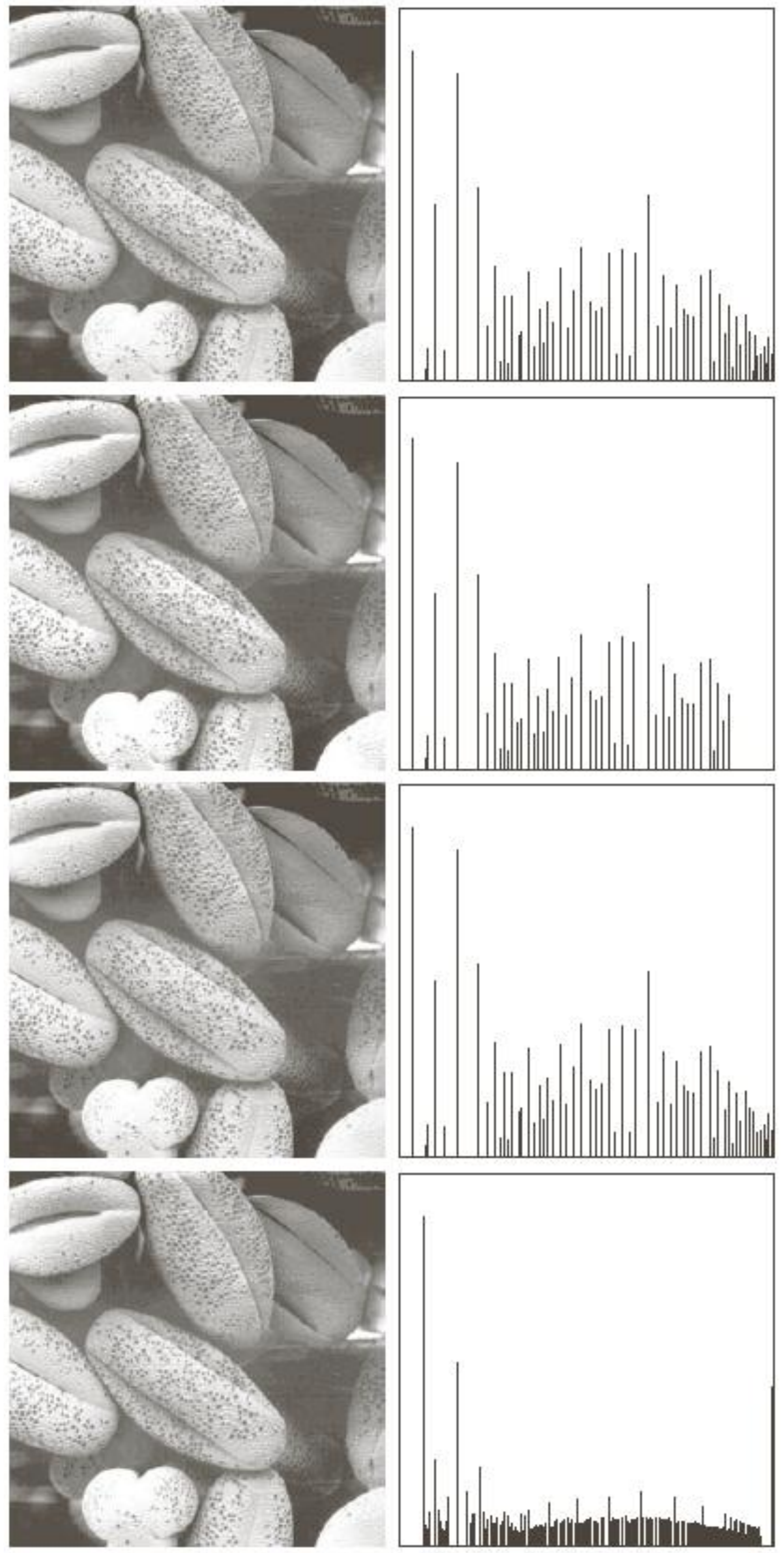


Image Processing

- Aims to enhance relevant information in the image.
- Usually is used to prepare images for further analysis and interpretation.
- Image represented as n x m array I(x,y) → image intensity array.
- Cells are called pixels. Each number represent light intensity.
- We will discuss three important techniques: histogram equalization, **noise removal**, and edge detection.

Image Processing: Noise Removal

- Real images always contain noise.

Image Processing: Noise Removal

- Real images always contain noise.

- **Smoothing** is a process used to reduce noise and sharp transitions in an image.
  - Smoothing tries to remove isolated bright and dark regions.
  - Smoothing filters work by making a pixel's intensity closer to the average or median intensity of its surroundings, suppressing isolated extreme values.

- Averaging + sliding → convolution.

- Has side-effect of blurring image.

Image Processing: Noise Removal

- It can use a threshold.
- Larger rectangles achieve more smoothing.
- Broad lines are thickened and thin lines eliminated
- In the example, ε = 3, i.e., 0 if sum ≤ 3, 1 otherwise.

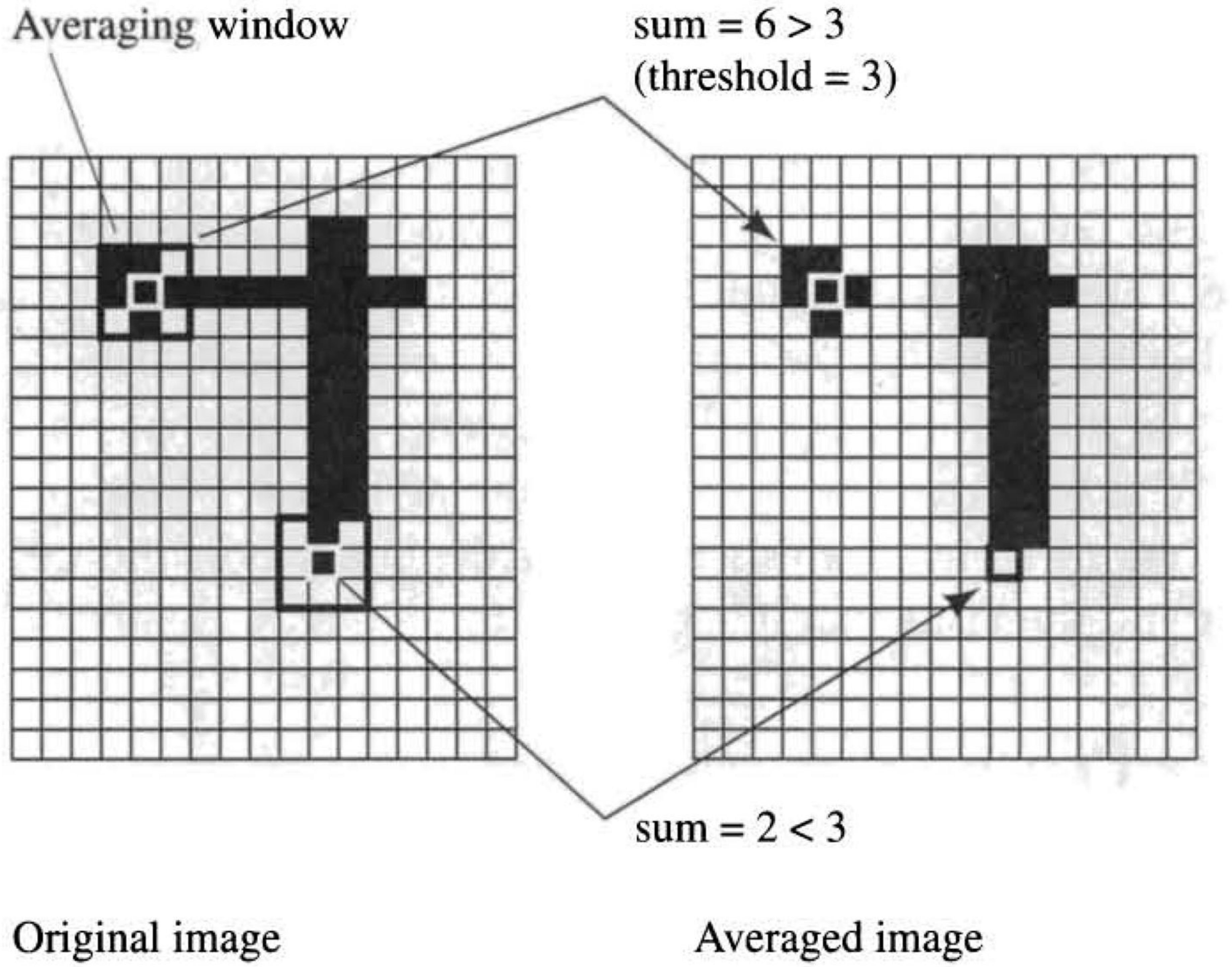


Image Processing: Averaging

- Given a simple 4 x 4 picture matrix:

$$\begin{matrix} 9 & 9 & 9 & 3 \\ 9 & 9 & 3 & 3 \\ 9 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \end{matrix}$$

- Smooth this matrix using an averaging technique and a 3 x 3 pixel window.

Image Processing: Averaging

- There are four 3 x 3 pixel windows in the matrix.

- Replace middle value in each window by average of all the values in the window.

$$\begin{matrix} \boxed{\begin{matrix} 9 & 9 & 9 \\ 9 & 9 & 3 \\ 9 & 3 & 3 \end{matrix}} & \begin{matrix} 3 \\ 3 \\ 3 \end{matrix} \\ \begin{matrix} 3 & 3 & 3 \end{matrix} & 3 \end{matrix} \longrightarrow \begin{matrix} 9 & 9 & 9 & 3 \\ 9 & 7 & 5 & 3 \\ 9 & 5 & 4 & 3 \\ 3 & 3 & 3 & 3 \end{matrix}$$

Image Processing

- Aims to enhance relevant information in the image.
- Usually is used to prepare images for further analysis and interpretation.
- Image represented as n x m array I(x,y) → image intensity array.
- Cells are called pixels. Each number represent light intensity.
- We will discuss three important techniques: histogram equalization, noise removal, and **edge detection**.

Image Processing: Edge Detection

- Edges are used to build a line drawing.
- Outlines can be compared with object models.
- Edges are parts of the image with markedly different property values (e.g., intensity)

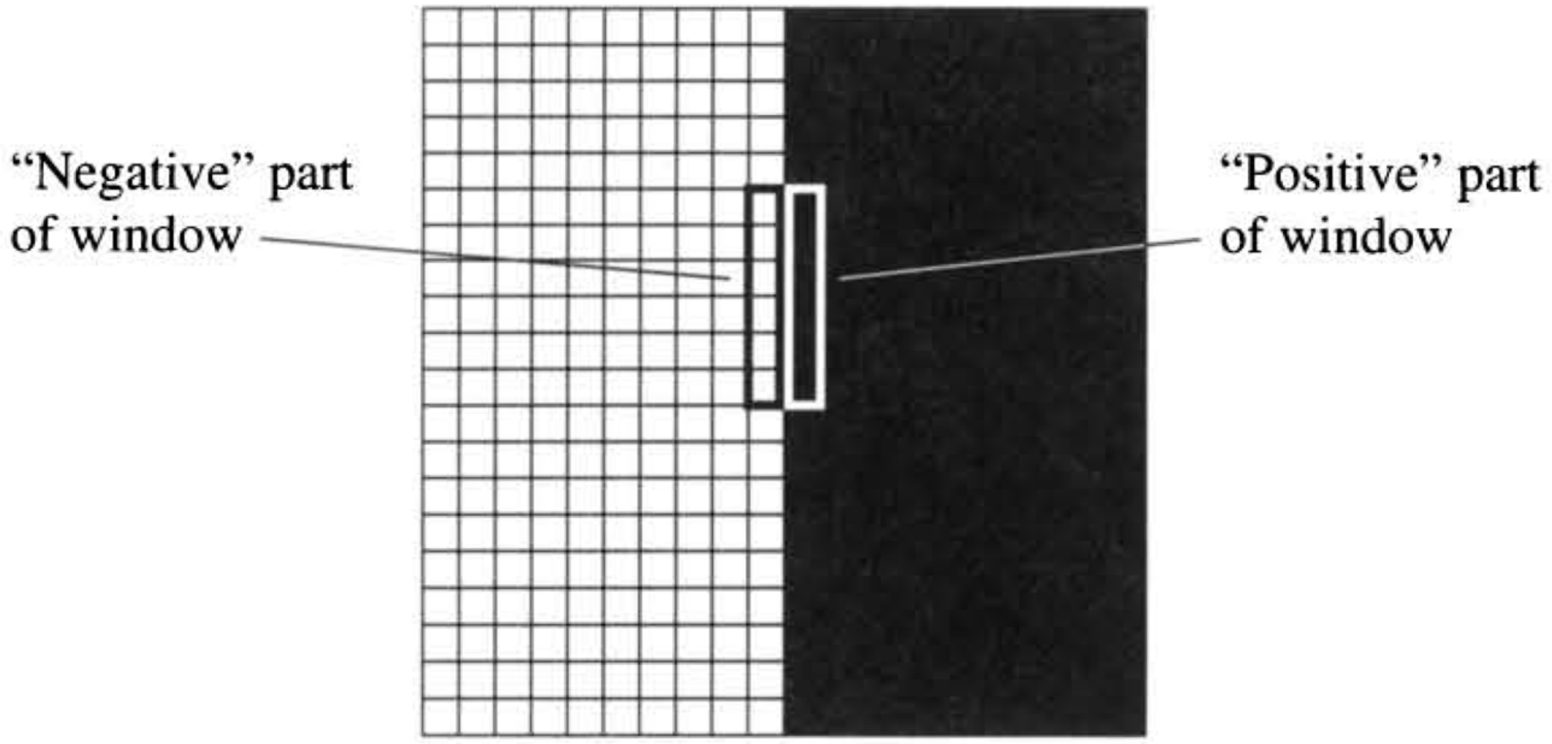


Figure 6.9

Edge Enhancement

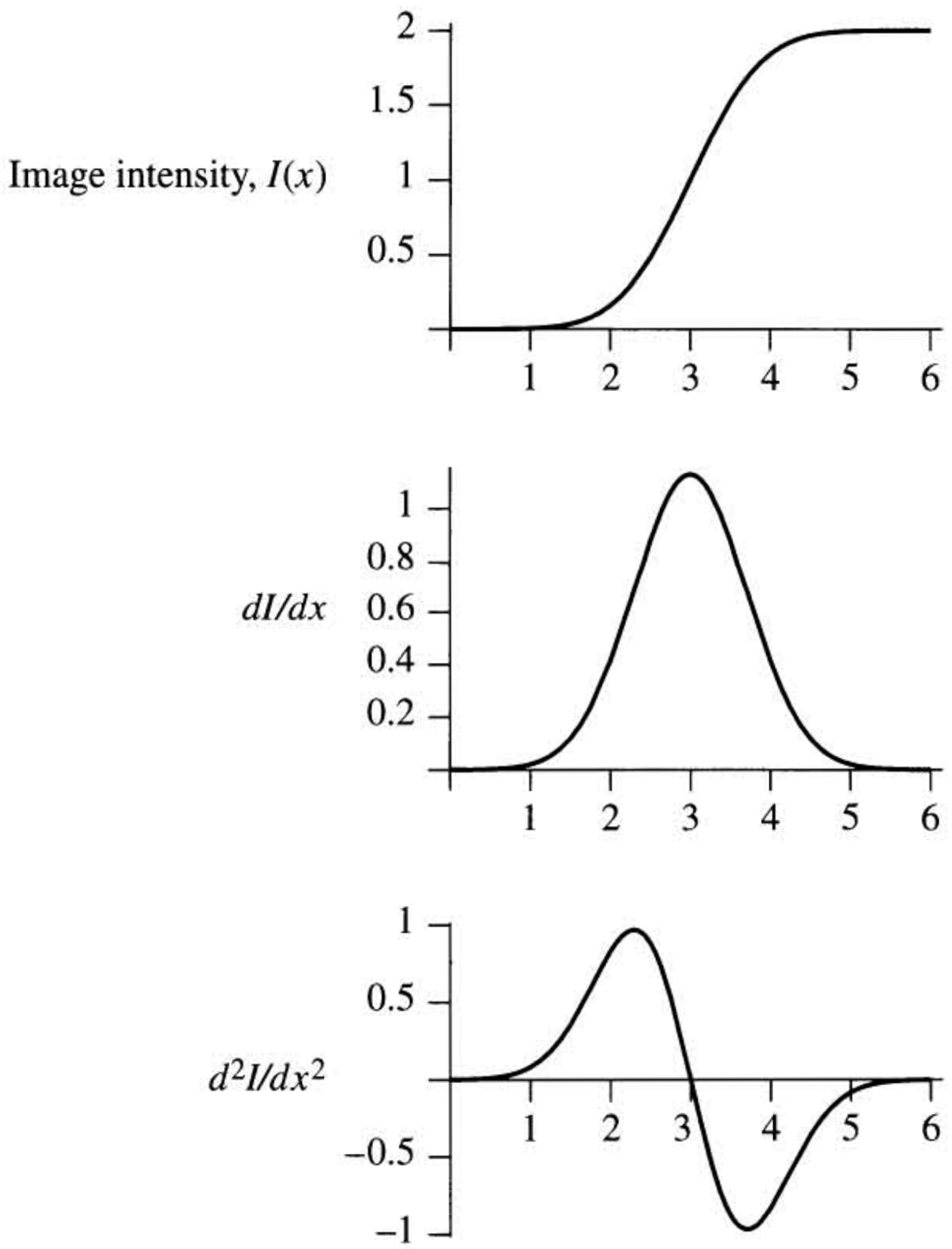


Figure 6.10

Taking Derivatives of Image Intensity

Image Processing: Edge Detection

Horizontal intensity profile through the center of the image, including the isolated noise point is shown here.

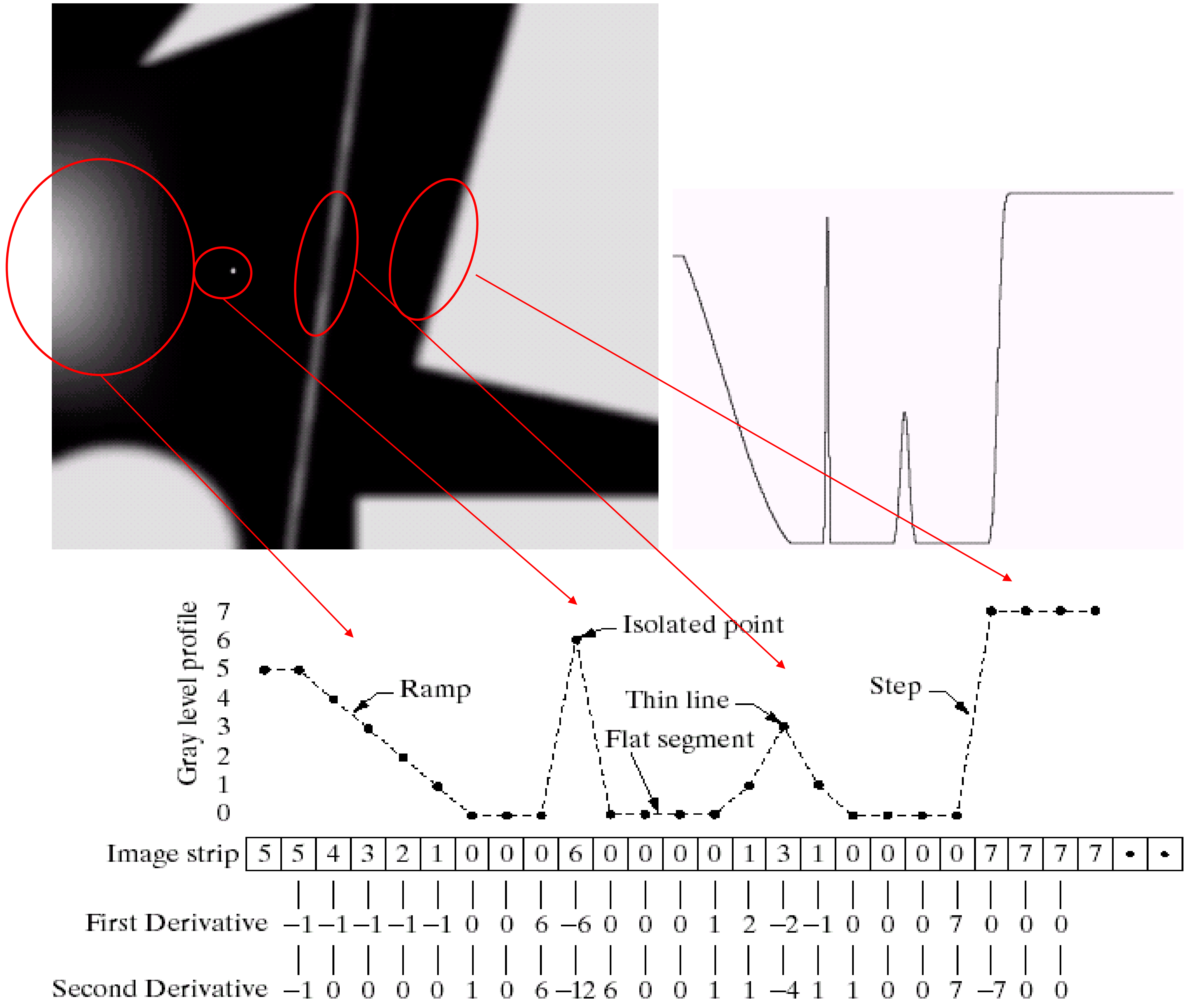


Image Processing: Edge Detection

Image Processing: Edge Detection

- The **Laplacian filter** is a spatial filter primarily used for edge detection.
- It computes the second derivatives of an image, highlighting regions where there are sudden changes in pixel intensity, which typically correspond to edges.
- If the pixel is part of a flat region, the result is close to zero.

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\frac{\partial^2 f}{\partial x^2} \approx f(x+1, y) - 2f(x, y) + f(x-1, y)$$

$$\frac{\partial^2 f}{\partial y^2} \approx f(x, y+1) - 2f(x, y) + f(x, y-1)$$

Image Processing: Edge Detection

$$\nabla^2 f(x,y) \approx f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

- We can write the kernel as follow:

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

- There are two common Laplacian filter:

Less sensitive

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

More sensitive, detects more edges

Image Processing: Edge Detection

- Averaging and edge enhancement can be combined.
- For instance, using a Laplacian filter.

(A) Original MR image

(B) Laplacian results

(C) Extraction of the zero crossing of the Laplacian (object edges)

Image Processing: Edge Detection

The **Canny edge detector** is an edge detection operator and can be broken down to different steps:

1. Apply **Gaussian filter** to smooth the image in order to remove the noise

2. Find the intensity gradients of the image and direction of each pixel.

3. Apply **gradient magnitude thresholding** and suppress edges that are not the strongest.

4. Apply double threshold to determine which edges are **strong**, **weak**, or **non-edges.**

Image Processing: Edge Detection

- **Gaussian filter** is a **filter** that reduces noise and detail in an image.
- Instead of averaging all pixels equally (as in a mean filter), the **Gaussian filter** gives **more weight to pixels near the center** of the window and **less to those farther away**.
- The 2D Gaussian function is:

$$G(\mathrm{x}, \mathrm{y}) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Where (x, y) is pixel coordinates relative to the center of the filter, and σ is the standard deviation.

Example (3×3 Gaussian kernel, σ≈1)

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Image Processing: Edge Detection

$$\begin{cases} G_x : \text{Gradient in the x-direction (vertical edges)} \\ G_y : \text{Gradient in the y-direction (horizontal edges)} \end{cases}$$

Edges in an image are places where **intensity changes rapidly** — that means **high derivatives**.

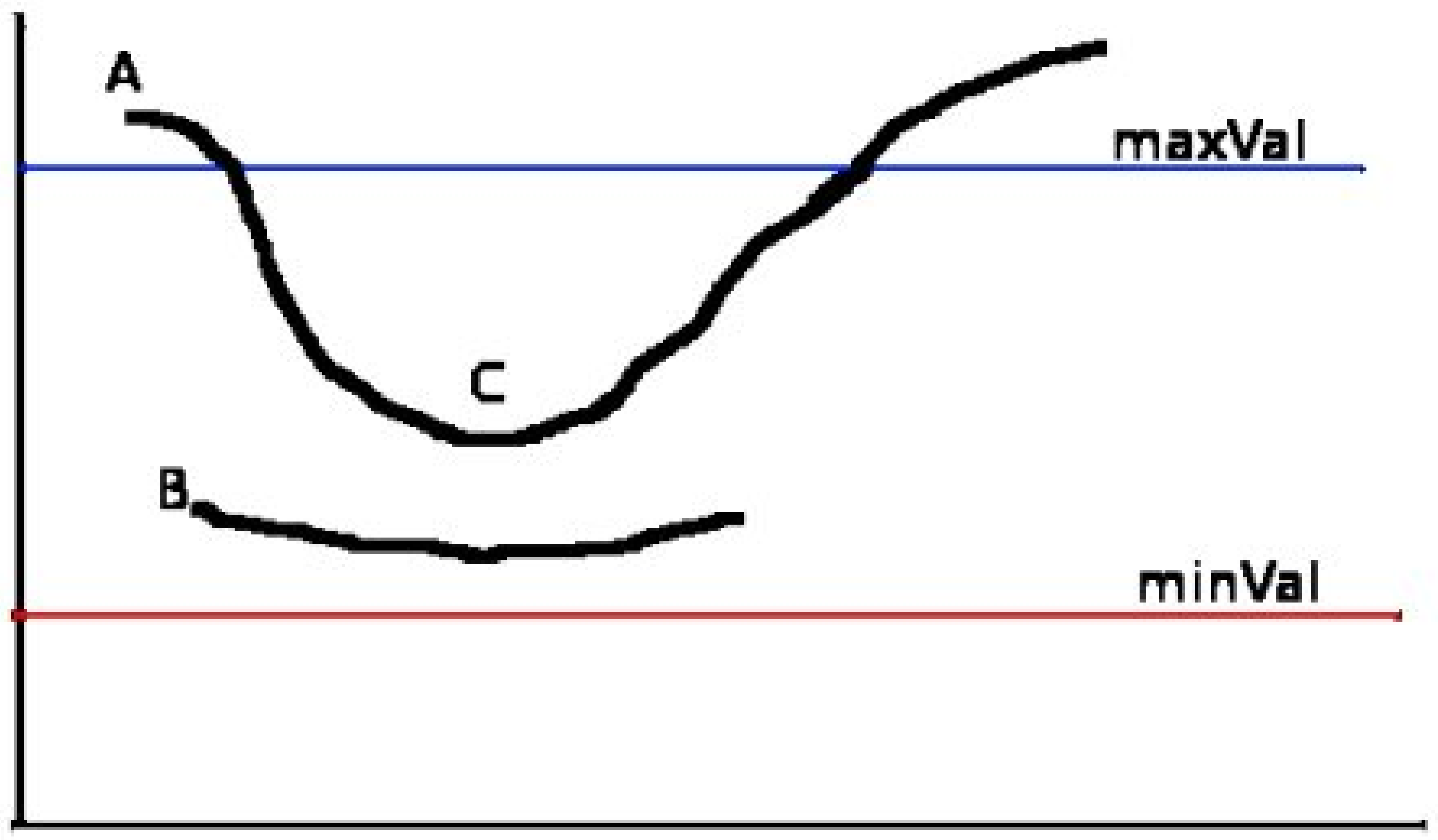
- **Horizontal edges:** Large values in G_y
- **Vertical edges** : Large values in G_x

Then we compute the **magnitude and direction of the gradient** at each pixel:

- Gradient Magnitude= $\sqrt{G_x^2+G_y^2}$
- Gradient Direction= $\arctan(G_y/G_x)$

Image Processing: Edge Detection

- Two thresholds are called minVal and maxVal.
- Any edges with intensity gradient more than maxVal are sure to be edges
- Those below minVal are sure to be non-edges, so discarded.
- Those who lie between these two thresholds if they are connected to "sure-edge" pixels, they are considered to be part of edges.



https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html

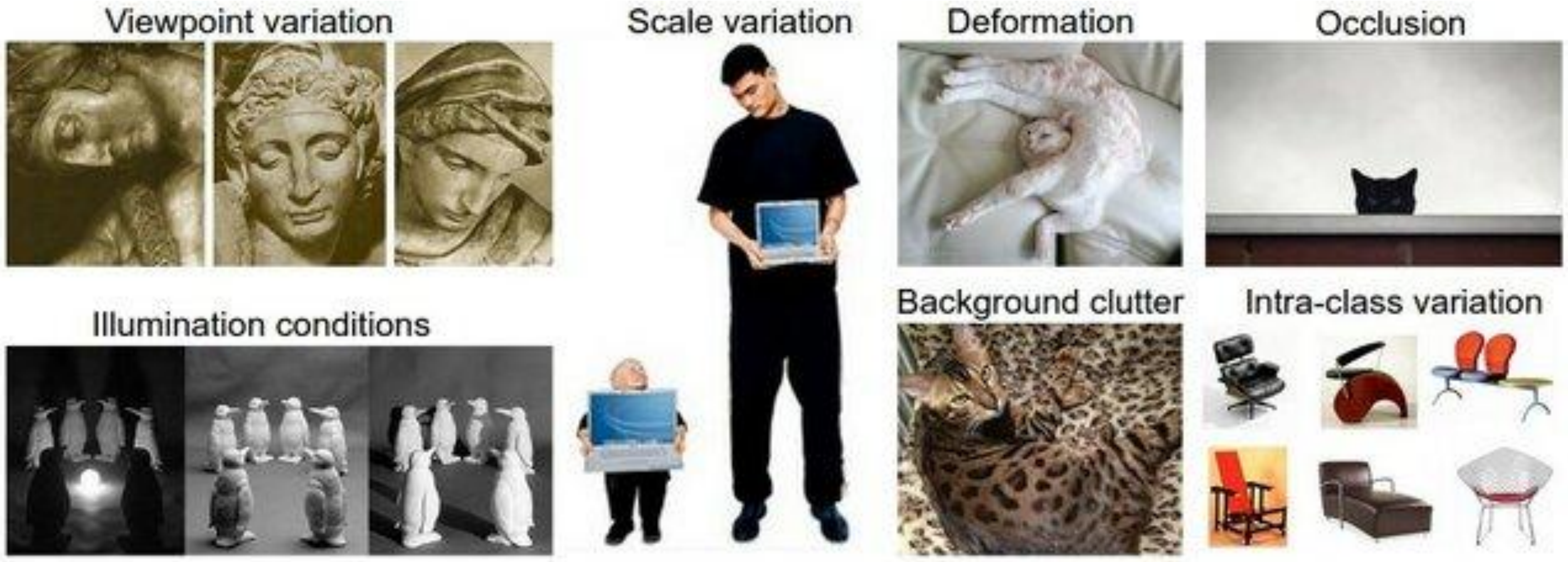
Lecture Overview

- Introduction
- Image Processing
  - ➢ Histogram Equalization
  - ➢ Noise Removal
  - ➢ Edge Detection
- **Computer Vision**
  - ➢ Convolutional Neural Networks
    - ✓ VGG16
    - ✓ AlexNet
- Cognitive Vision

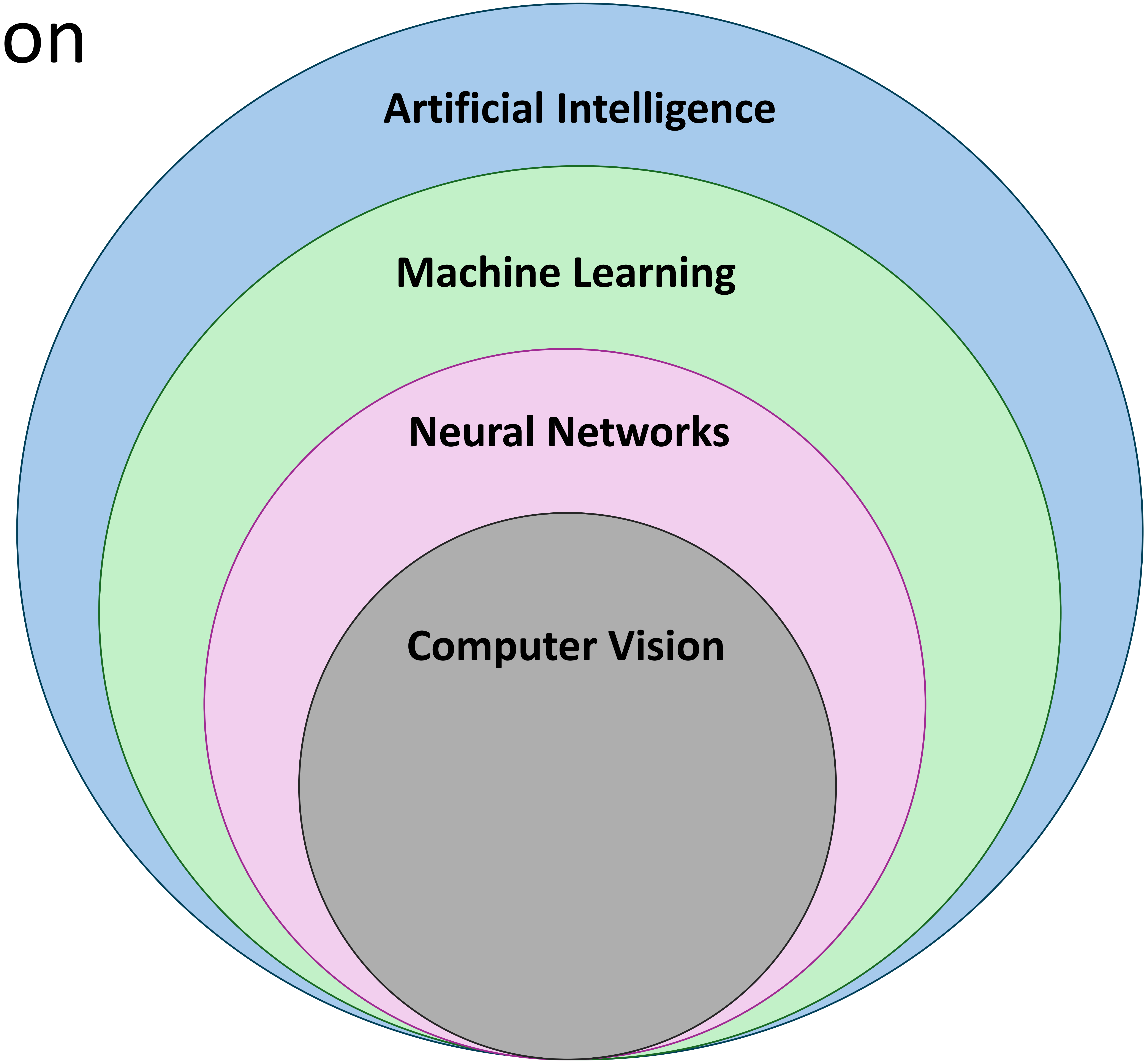
Computer Vision

Challenges in CV

- Variations in viewpoint
- Illumination condition
- Occlusion and deformation
- Background clutter



Computer Vision

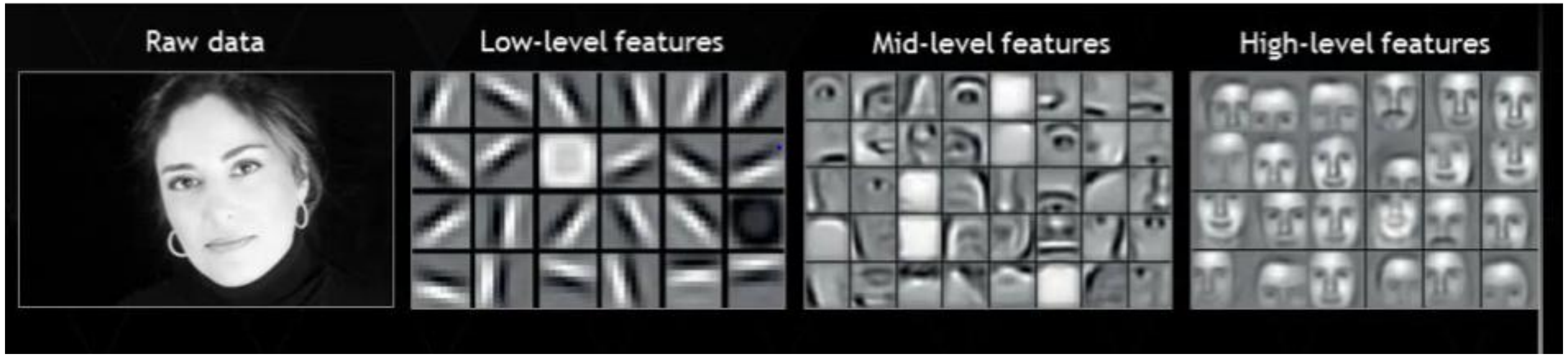


A comparative view of AI, machine learning, neural networks, and computer vision.

Computer Vision: CNN

Convolutions + Neural Networks = Convolutional Neural Networks (CNNs)

- **Convolutional neural networks (CNNs)** is a class of artificial neural networks dominant in various computer vision tasks.
- Convolutional neural network is composed of multiple building blocks, such as convolution layers, pooling layers, and fully connected layers, and is designed to automatically and adaptively learn spatial hierarchies of features.



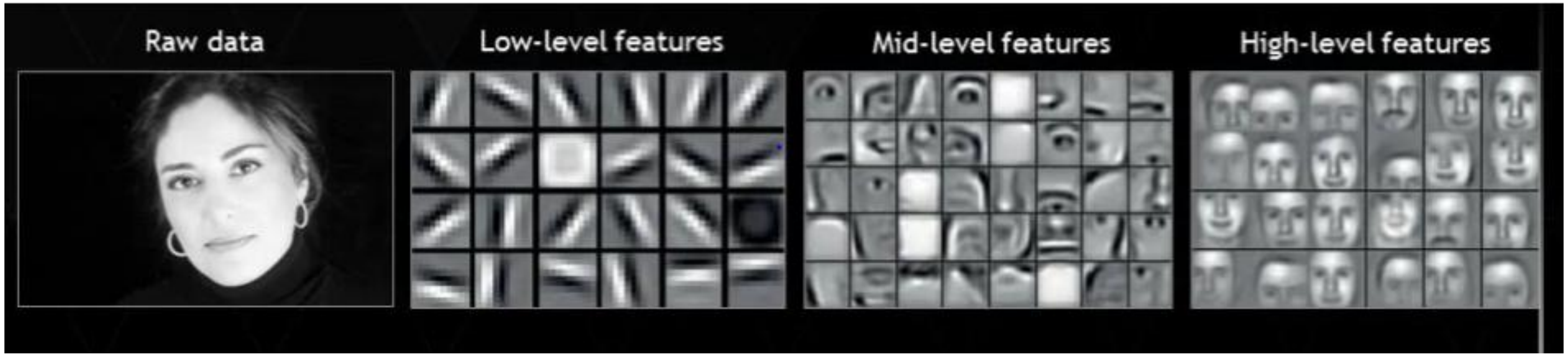
https://www.analyticsvidhya.com/blog/2017/04/comparison-between-deep-learning-machine-learning/

Computer Vision: CNN

Convolutions + Neural Networks = Convolutional Neural Networks (CNNs)

- CNNs will try to learn **low-level features** such as edges and lines in early layers, then parts of objects and then high-level representation of an object in subsequent layers.

- CNNs can be interpreted as gradually transforming the images into a representation in which the classes are separable by a linear classifier.



https://www.analyticsvidhya.com/blog/2017/04/comparison-between-deep-learning-machine-learning/

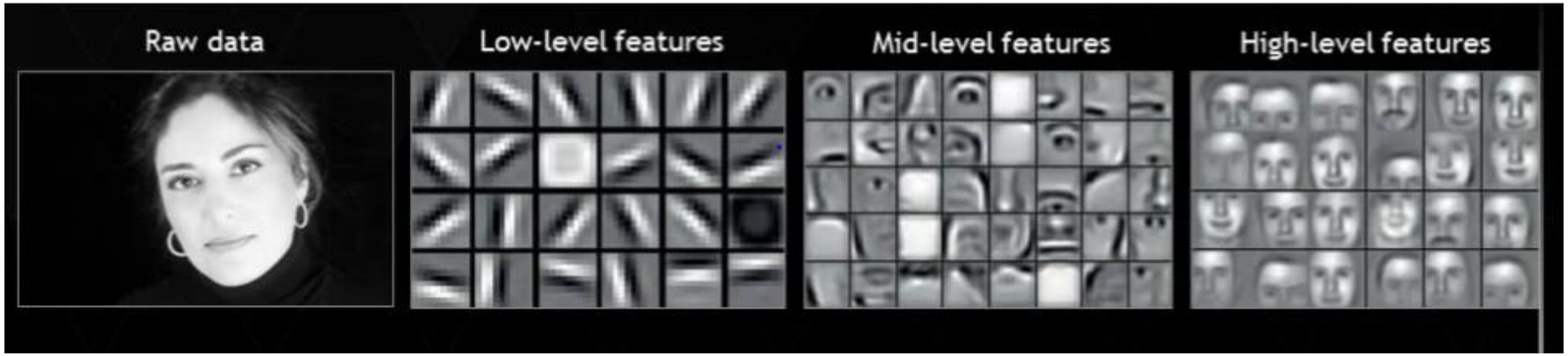
Computer Vision: CNN

Convolutions + Neural Networks = Convolutional Neural Networks (CNNs)

- CNNs are used for image recognition, object detection, and classification tasks.

- CNNs excel at feature extraction, learning complex and abstract features from input data.

- **Parameter sharing** in CNNs reduces computational and memory requirements, which makes them efficient.



https://www.analyticsvidhya.com/blog/2017/04/comparison-between-deep-learning-machine-learning/

Computer Vision: CNN

Convolution is a mathematical operator.

Continuous convolution

$$s(t) = (x * w)(t) = \int x\,(a) w(t-a)\,da$$

Discrete convolution

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x\,(a) w(t-a)$$

Two-dimensional convolution

$$S(j,k) = (K * I)(j,k) = \sum_m \sum_n K(m,n)\, I(j+m, k+n)$$

Computer Vision: CNN

Key Components:

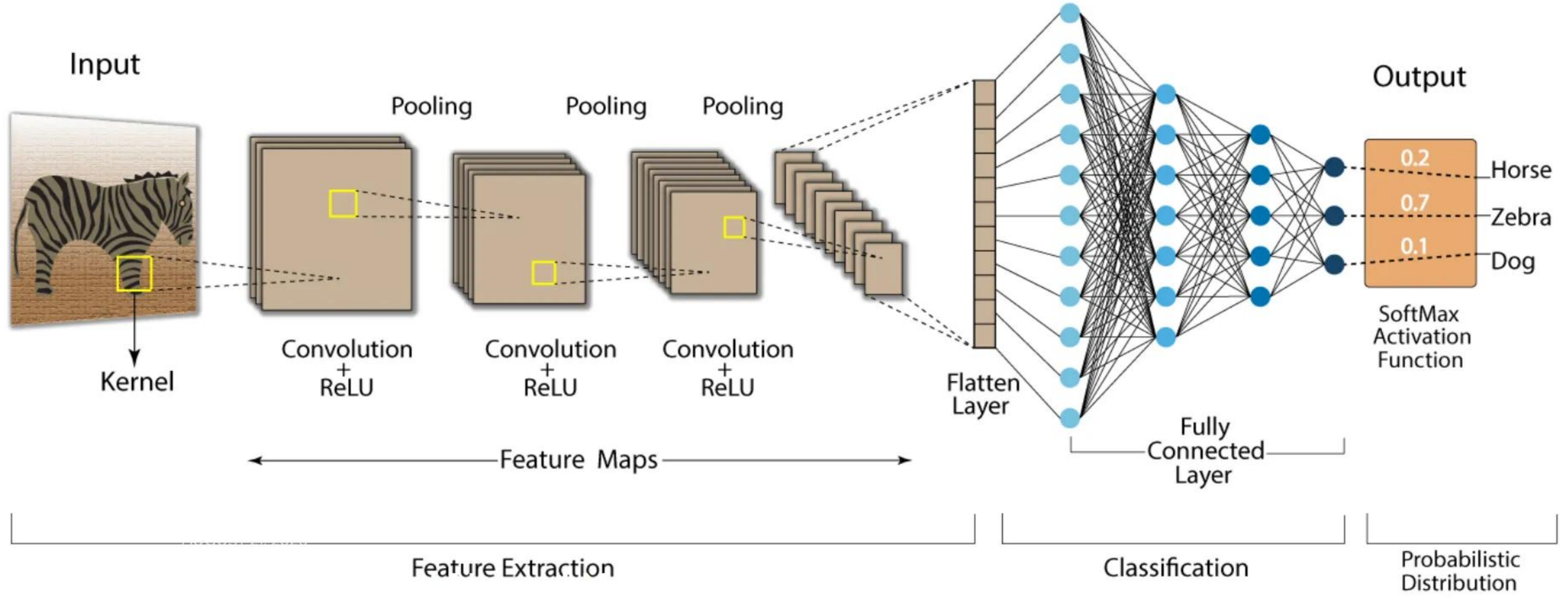
1. Convolutional Layers – Apply filters (kernels) to detect features such as edges, textures, and patterns.

2. Pooling Layers – Reduce spatial dimensions while retaining important features.

3. Flattening layer – convert feature maps into a one-dimensional vector

4. Fully Connected Layers – Connect extracted features to output layers for classification or regression.

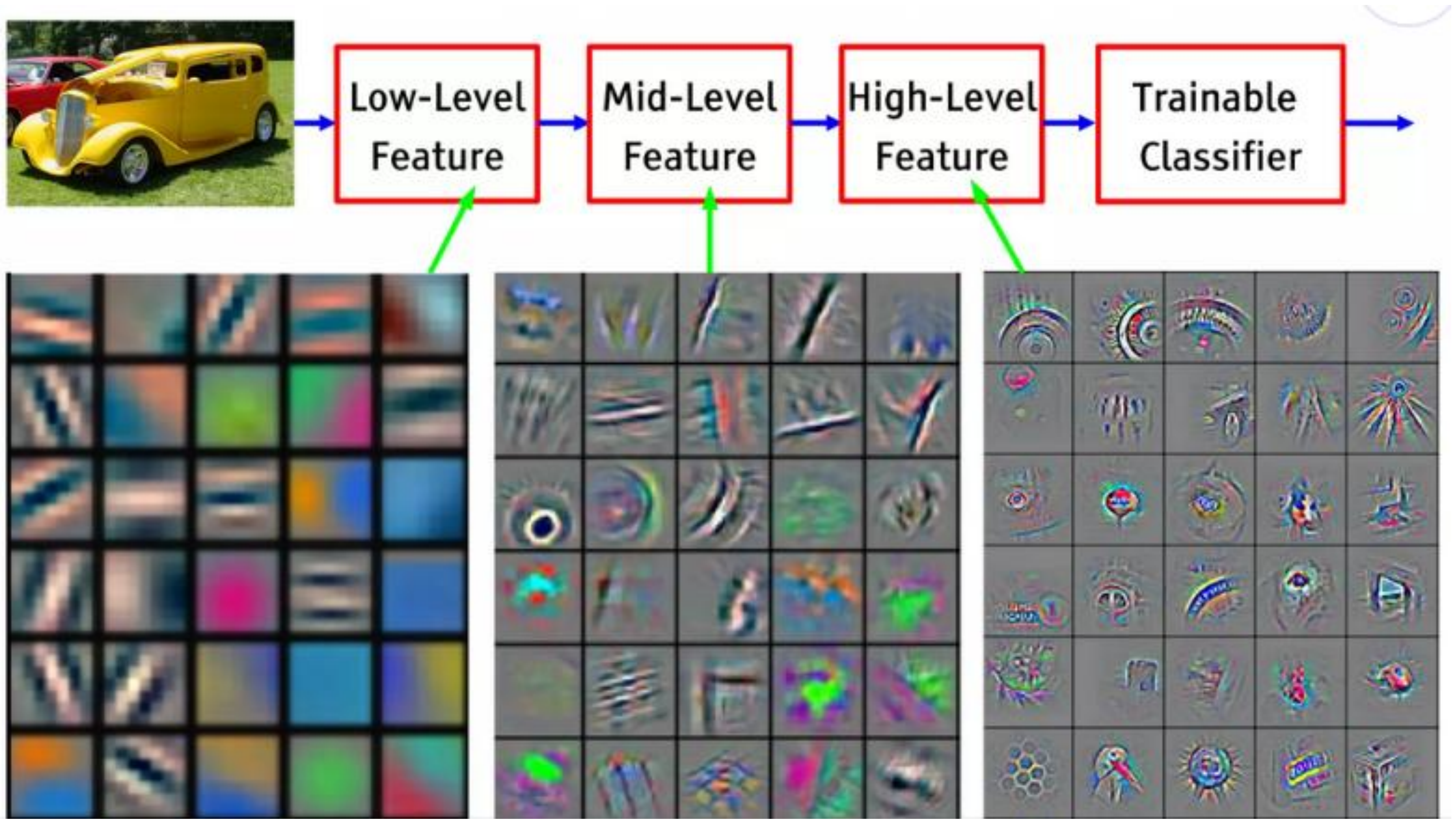
5. Output layer – choose between classes.



Sourced from nafizshahriar.medium.com on convolutional neural networks and deep learning.

Computer Vision: CNN

- Convolution helps us extract important features from an image (e.g. edges)



Feature Visualization of convolutional network trained on ImageNet from [Zeiler and Fergus 2013]. Source: https://www.slideshare.net/slideshow/yann-le-cun/27570713

Computer Vision: CNN

Convolutional Layers

- Convolution is a mathematical operation which produces a filtered version of the original image.

- This operation is called convolution because it involves “sliding” the filter over the image, element-wise multiplying the values of the filter, and summing the results.

- This process is repeated for every pixel in the image.

1. Multiply the 1st neighbor pixel by the 1st value in the Kernel

125	213	98	203	202	170
104	145	161	204	201	157
72	8	209	202	194	144
73	9	202	201	194	156
81	15	189	185	181	144
15	189	185	194	227	158

Original Image

1	1	1
1	1	1
1	1	1

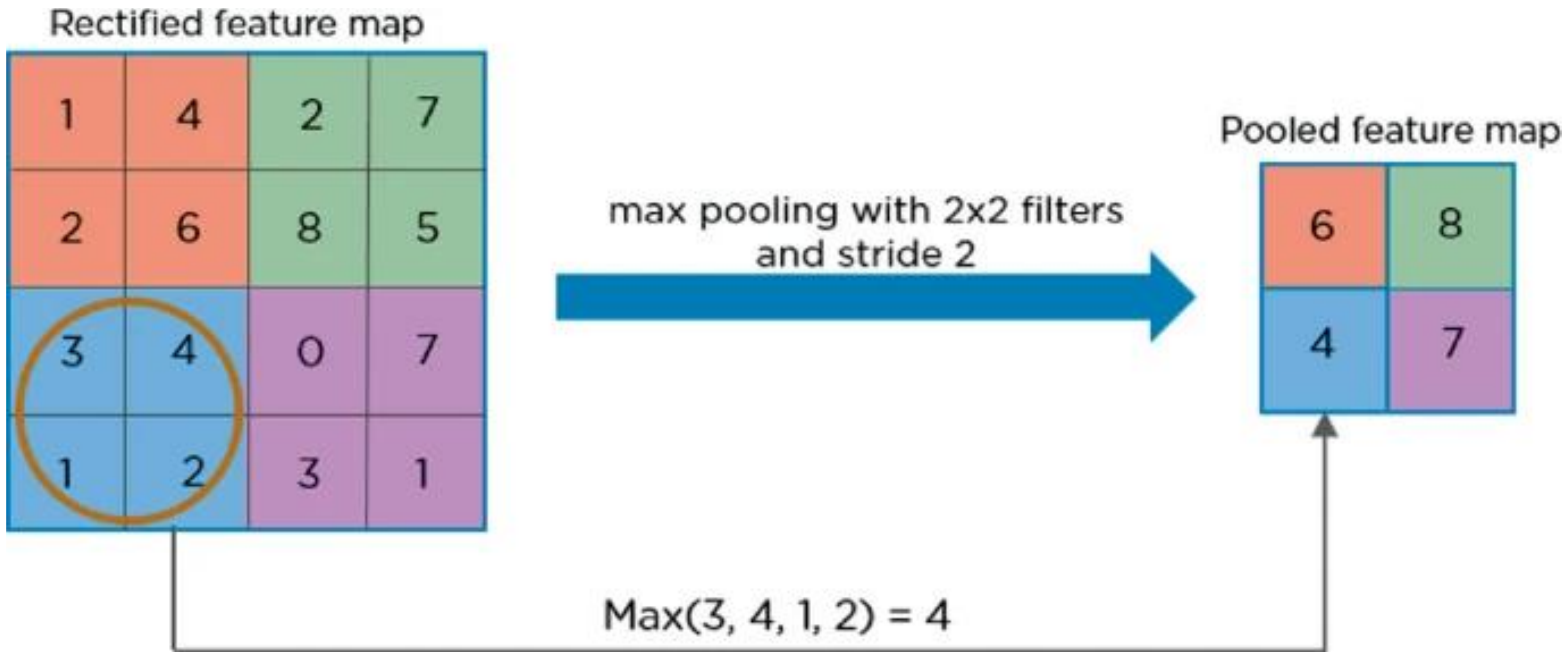
3x3 Box
Blur Kernel

https://medium.com/@ChanakaDev/deep-learning-for-computer-vision-8117d7fc4d76

Computer Vision: CNN

Pooling Layers :

- Pooling is a **down-sampling** operation that reduces the dimensionality of the feature map.
- The pooling layer uses various filters to identify different parts of the image like edges, corners, body, feathers, eyes, and beak.

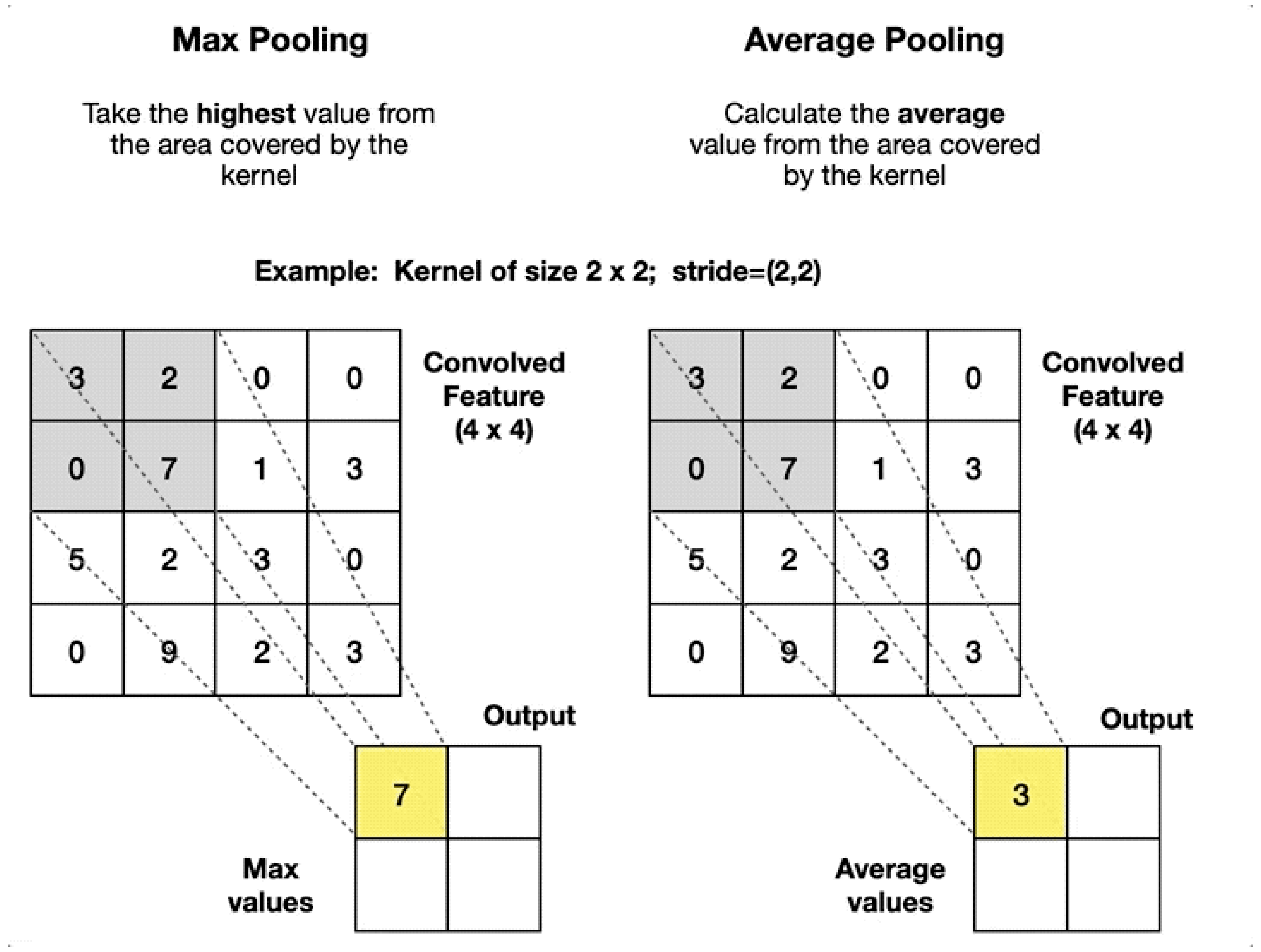


Das, V., Das, D., Hazra, R. (2024). Vernacular Language Handwriting Recognition Using Deep Learning Techniques. In: Gabbouj, M., Pandey, S.S., Garg, H.K., Hazra, R. (eds) Emerging Electronics and Automation. E2A 2022. Lecture Notes in Electrical Engineering, vol 1088. Springer, Singapore. https://doi.org/10.1007/978-981-99-6855-8_46

Computer Vision: CNN

Pooling Layers :

- Pooling is a down-sampling operation that reduces the dimensionality of the feature map.
- The pooling layer uses various filters to identify different parts of the image like edges, corners, body, feathers, eyes, and beak.

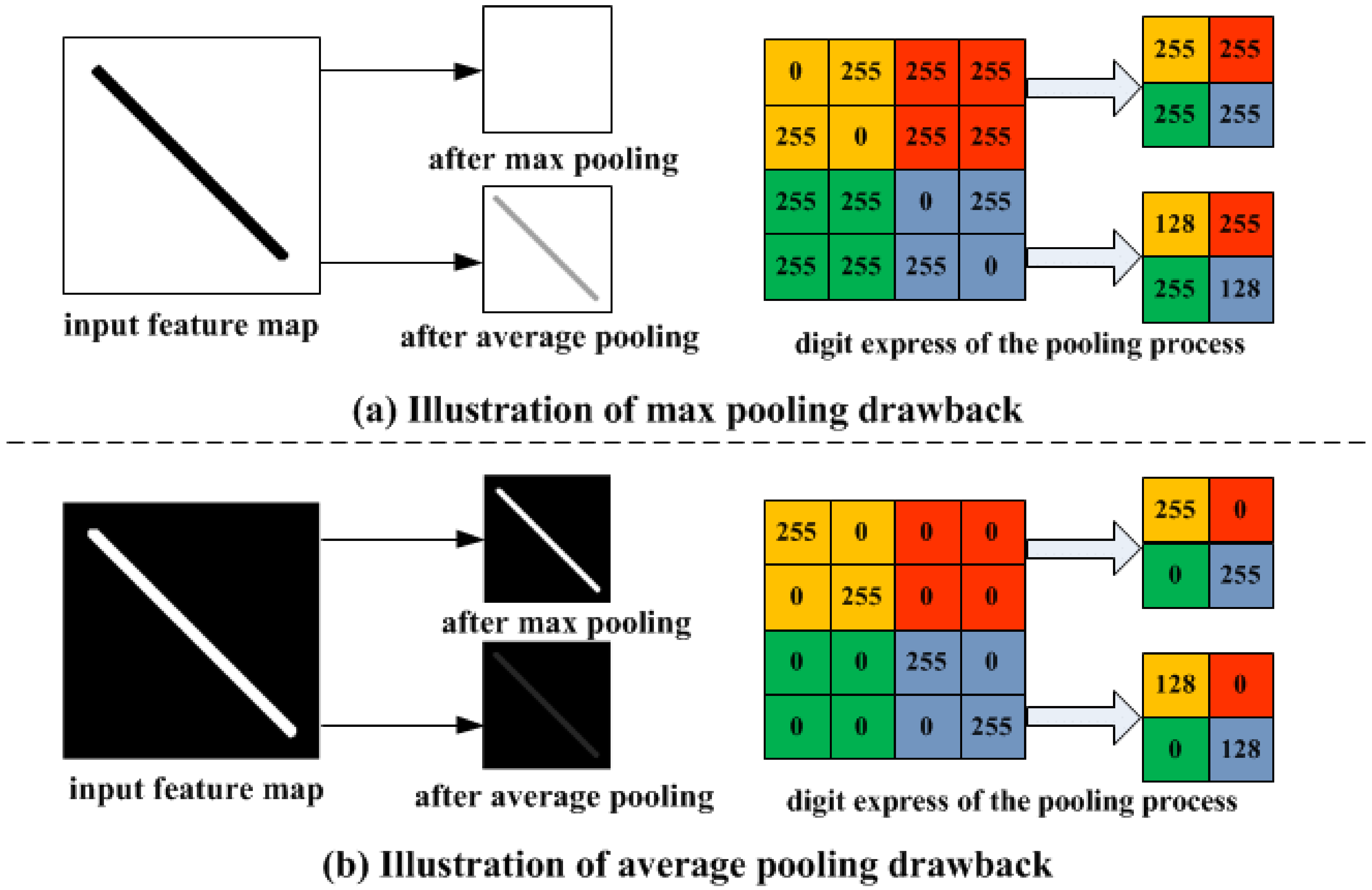


Medium: Intro to Pooling layers in CNN https://pub.towardsai.net/introduction-to-pooling-layers-in-cnn-dafe61eabe34

Computer Vision: CNN

Pooling Layers :

- Pooling is a down-sampling operation that reduces the dimensionality of the feature map.
- The pooling layer uses filters to identify edges, corners, body, feathers, etc.

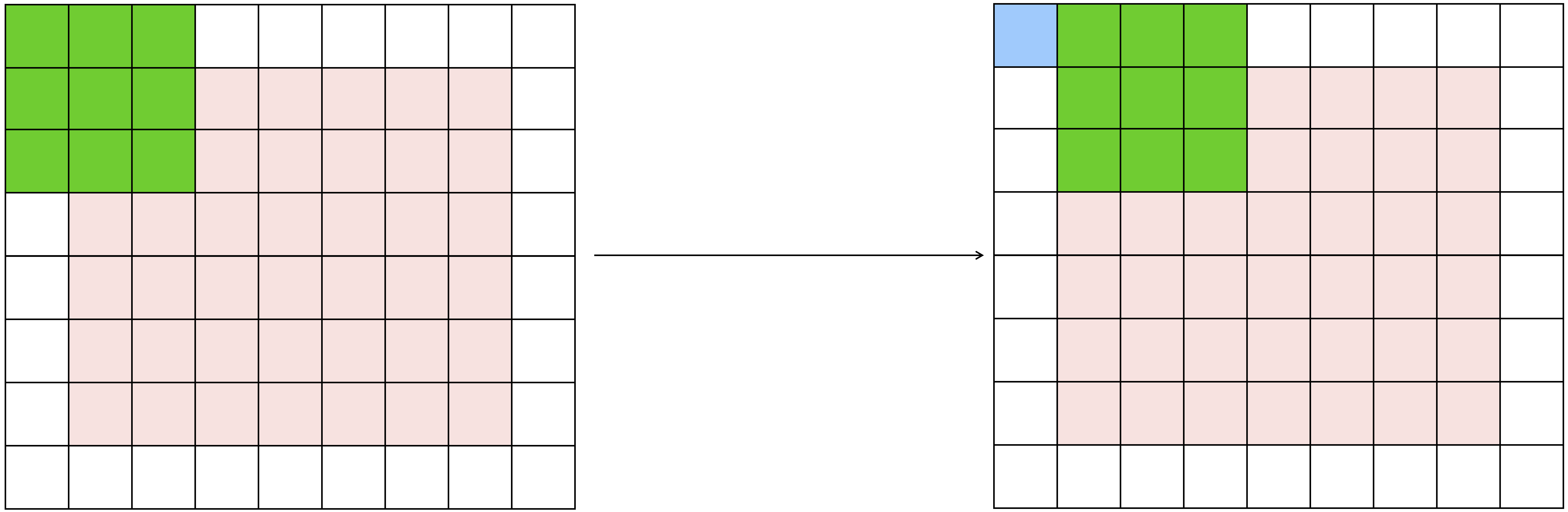


Yu, Dingjun, Hanli Wang, Peiqiu Chen and Zhihua Wei. “Mixed Pooling for Convolutional Neural Networks.” Rough Sets and Knowledge Technology (2014).

Computer Vision: CNN

Padding :

Sometimes, we expand the size of the input (or the previous layer) but treat the newly added units as having a fixed value (typically, zero) in order to **save the information in the edges**.

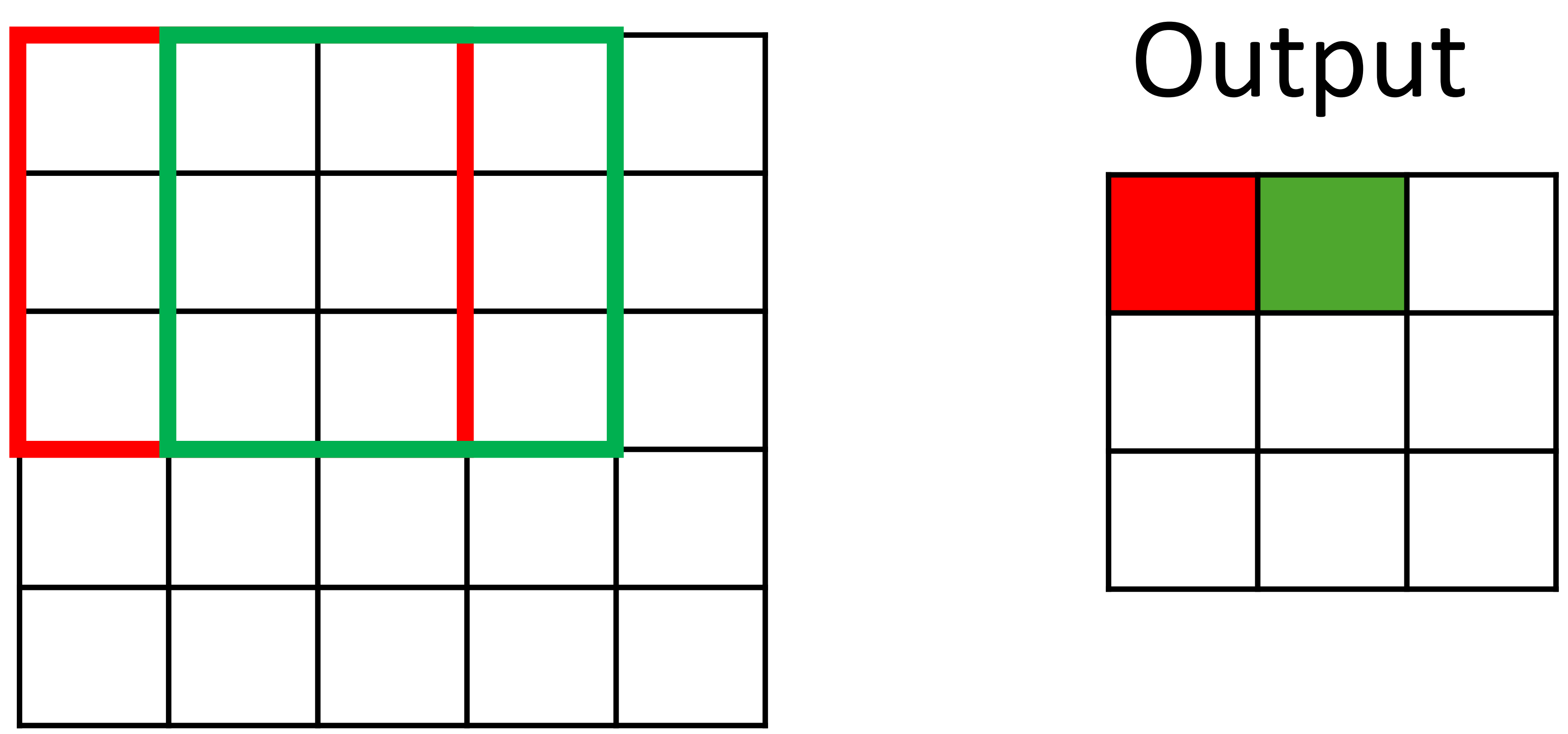


Computer Vision: CNN

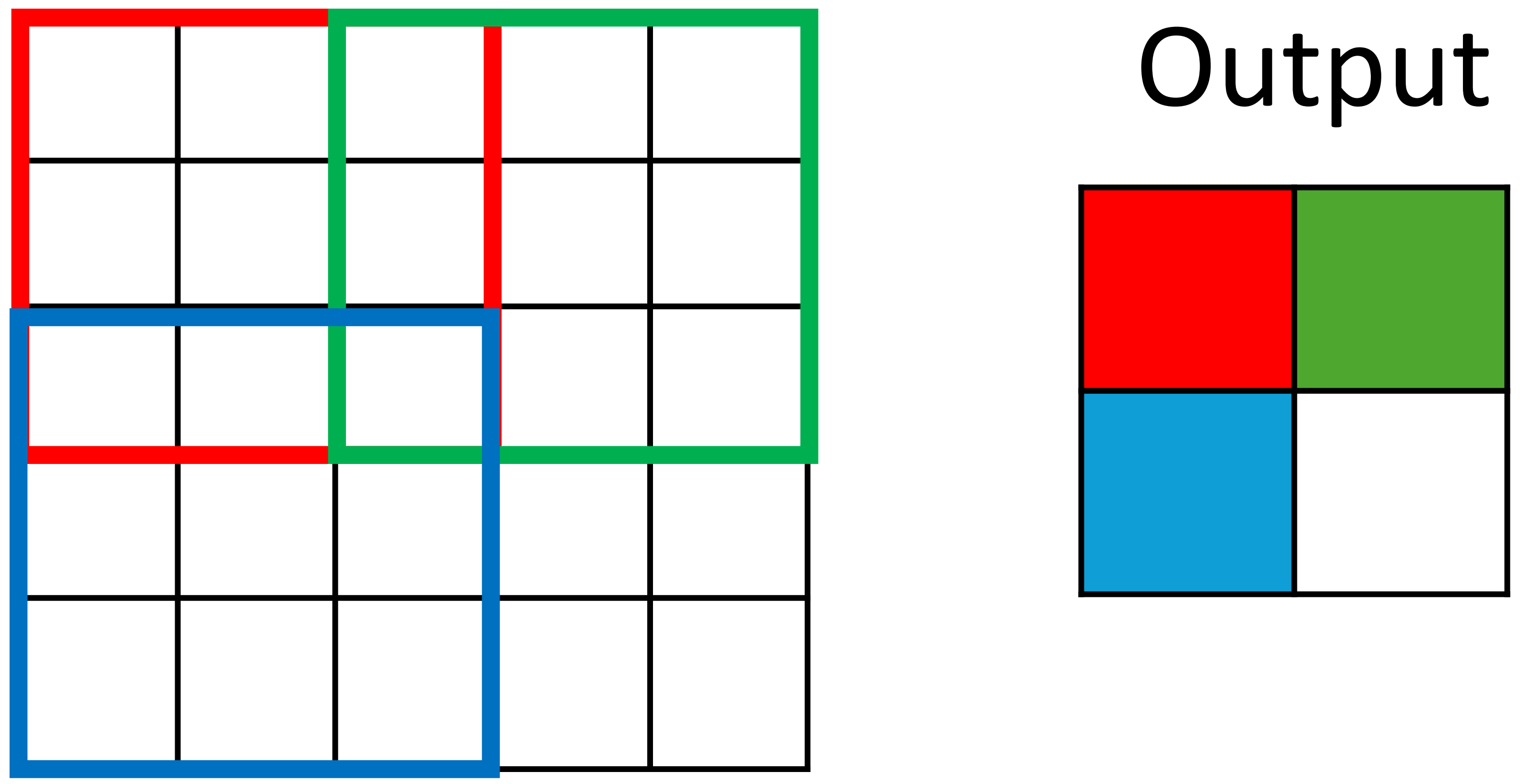
Stride refers to the step size by which a convolutional filter moves across an input image.

- A **stride of 1** means the filter moves one pixel at a time, leading to a highly detailed feature map with minimal size reduction.
- A **stride of 2** means the filter moves two pixels at a time, reducing the spatial dimensions more aggressively.
- Larger stride values result in **smaller output feature maps** and reduce computational cost, but may lose fine-grained details.

Convolution with stride =1



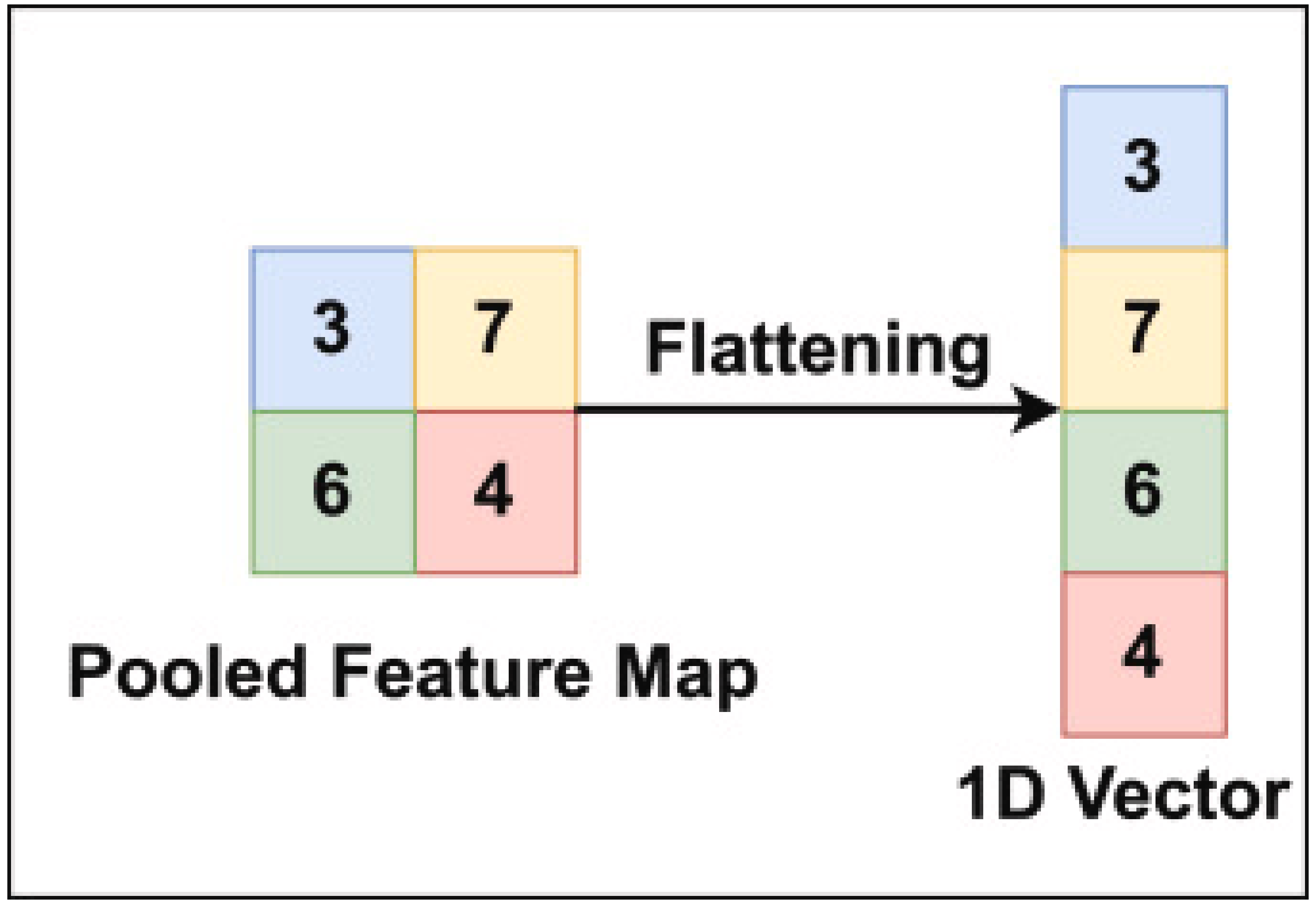
Convolution with stride =2



Computer Vision: CNN

Flattening Layer :

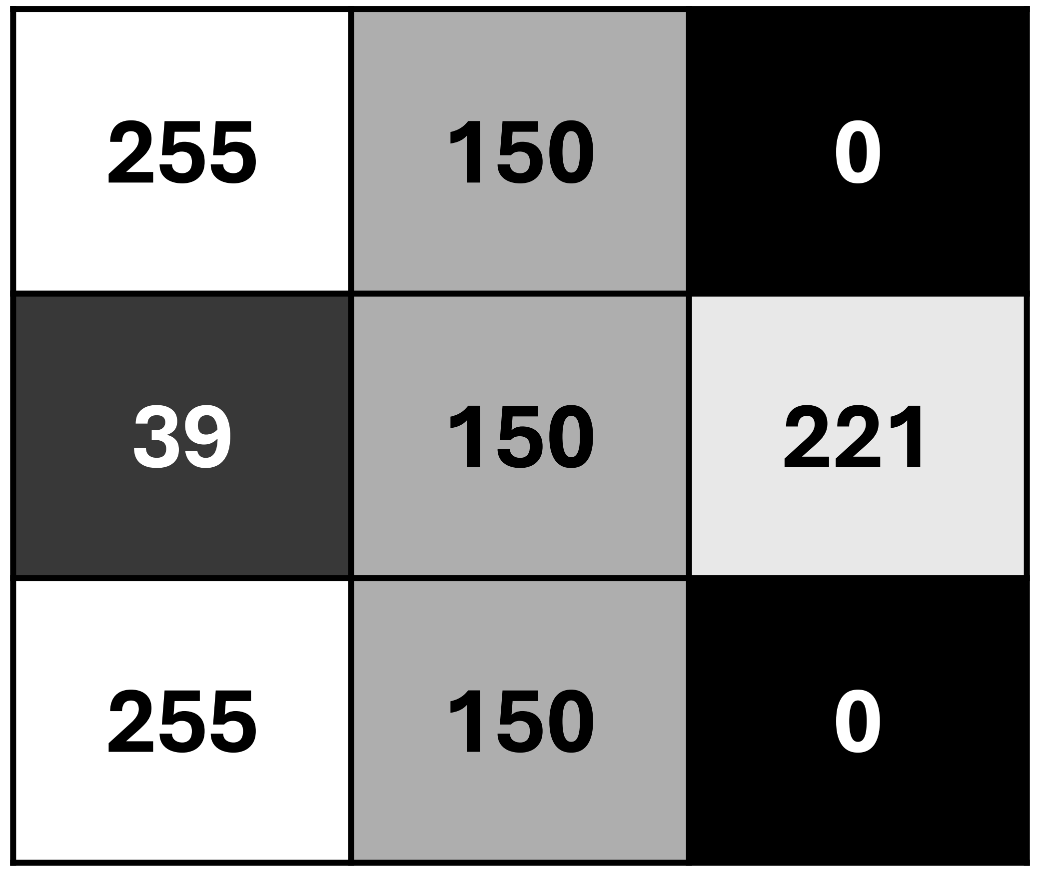
- After the convolution and pooling operations, the feature maps still exist in a multi-dimensional format.
- Flattening converts these feature maps into a one-dimensional vector. It prepares the data to be passed into fully connected layers for classification or regression tasks.



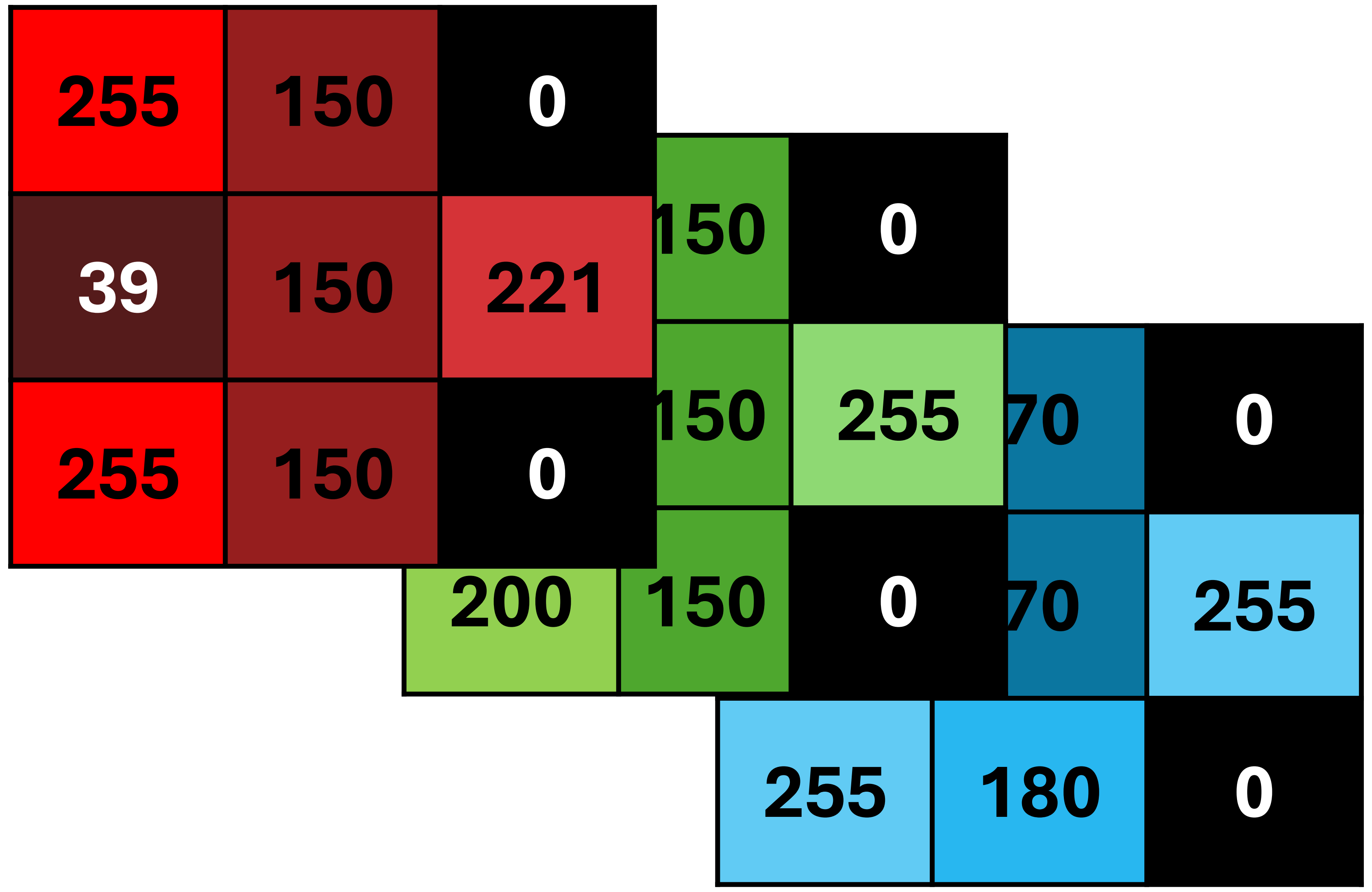
Computer Vision: CNN

- For an RGB image, there are typically separate kernels for each colour channel because different features might be more visible or relevant in one channel compared to the others.
- Each filter produces a separate feature map.
- The output normally has multiple channels, where each channel is a feature map corresponding to a particular kernel.

Grayscale

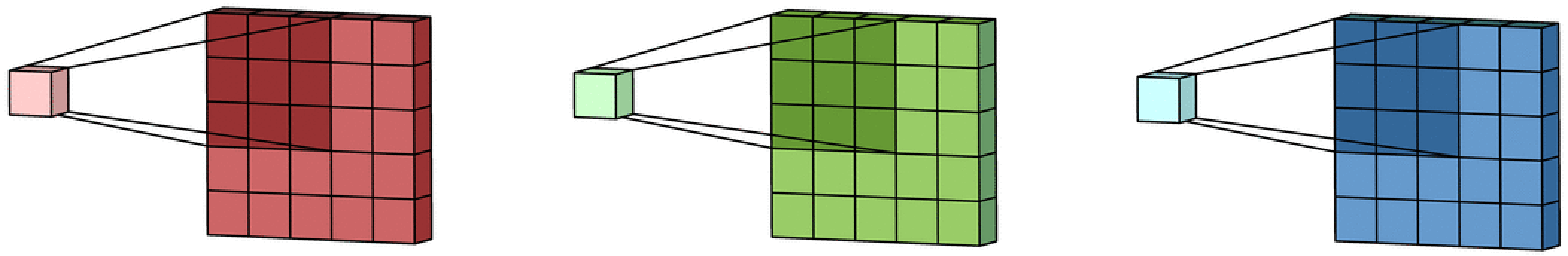


RGB



Computer Vision: CNN

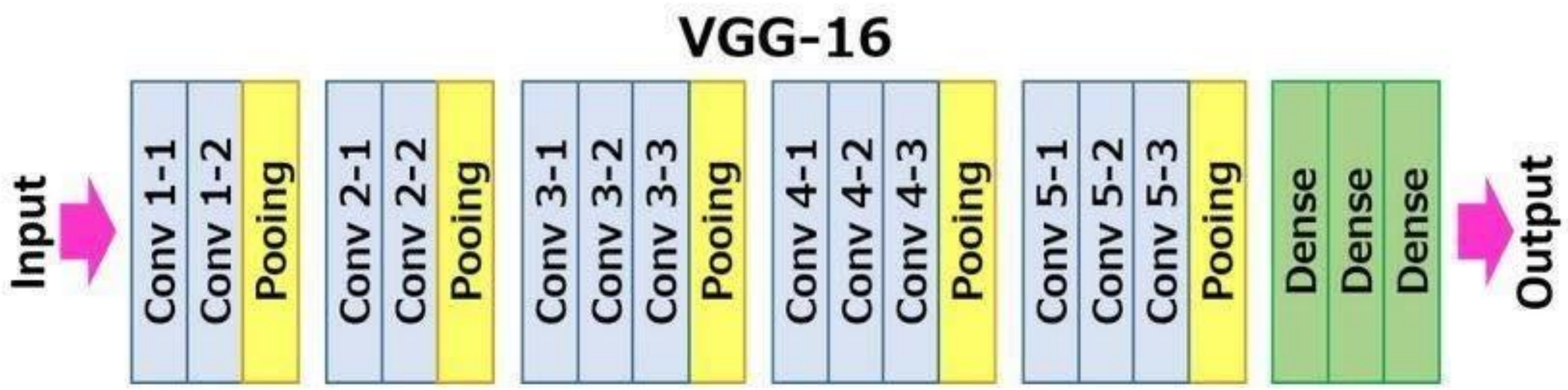
- For an RGB image, there are typically separate kernels for each colour channel because different features might be more visible or relevant in one channel compared to the others.
- Each filter produces a separate feature map.
- The output normally has multiple channels, where each channel is a feature map corresponding to a particular kernel.



https://blog.devgenius.io/convolutional-neural-networks-98e2067072a9

Computer Vision: VGG16

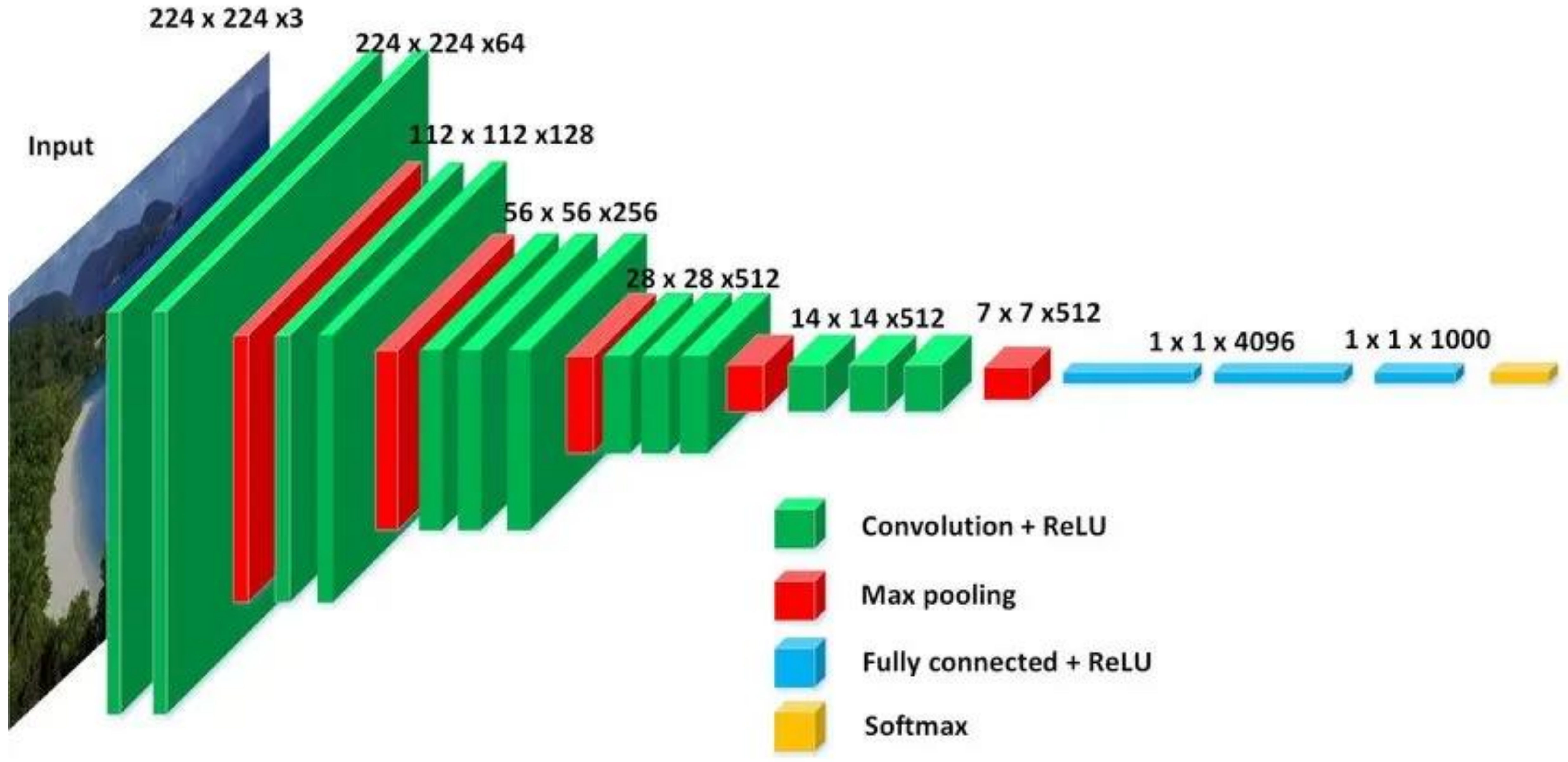
- **VGG16** is a CNN architecture developed by the Visual Geometry Group at the University of Oxford (Simonyan and Zisserman, 2014).
- It is a **16-layer** deep model, consisting of **13 convolutional layers**, 3 fully connected layers, and 5 pooling layers.
- The architecture is characterized by the use of small 3x3 convolution kernels and ReLU activation functions.



https://smuhabdullah.medium.com/understanding-vgg16-a-powerful-deep-learning-model-for-image-recognition-d40b074fd01c

Computer Vision: VGG16

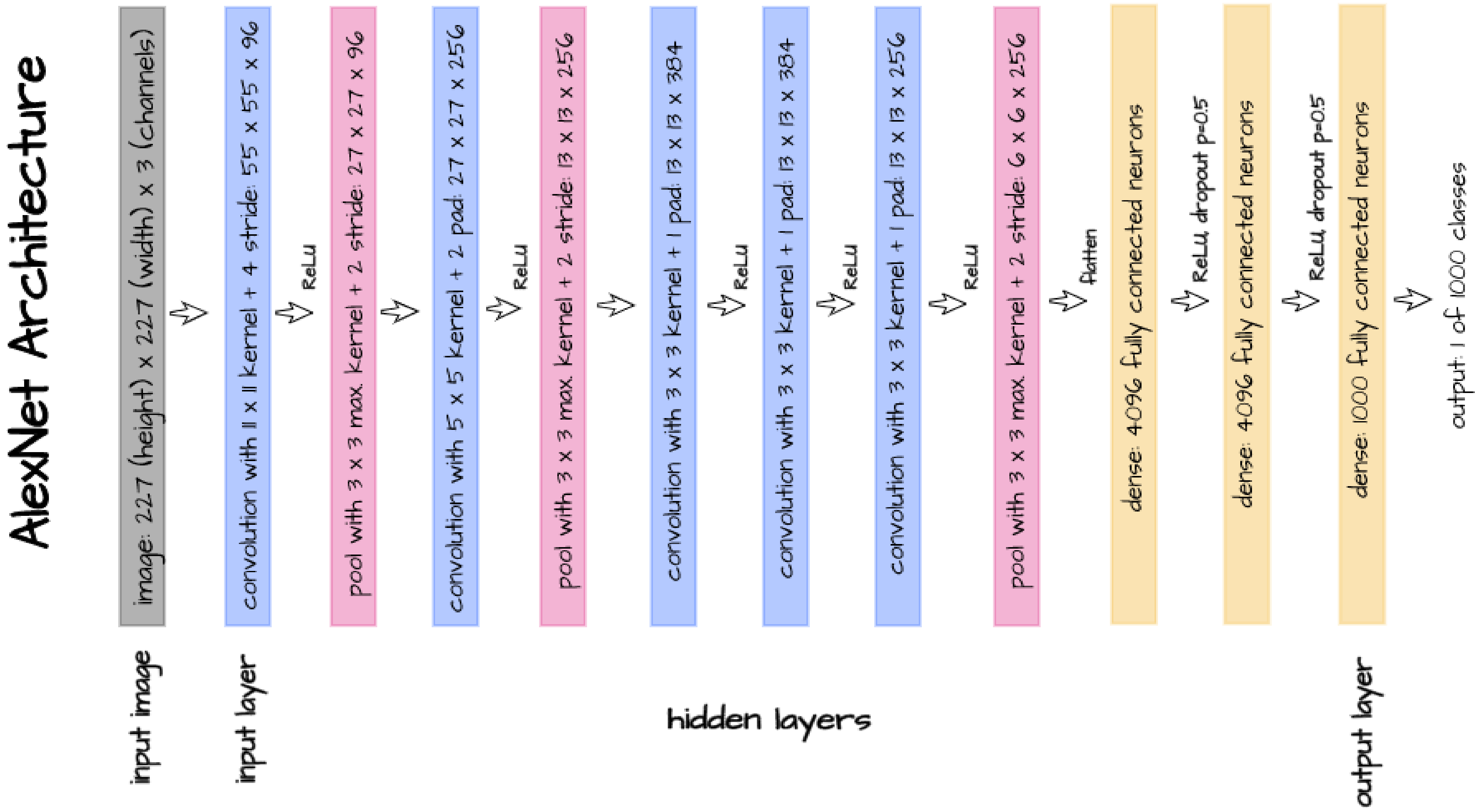
- It is a 16-layer deep model, consisting of 13 convolutional layers, 3 fully connected layers, and 5 pooling layers.
- The architecture is characterized by the use of small 3x3 convolution kernels and ReLU activation functions.



https://smuhabdullah.medium.com/understanding-vgg16-a-powerful-deep-learning-model-for-image-recognition-d40b074fd01c

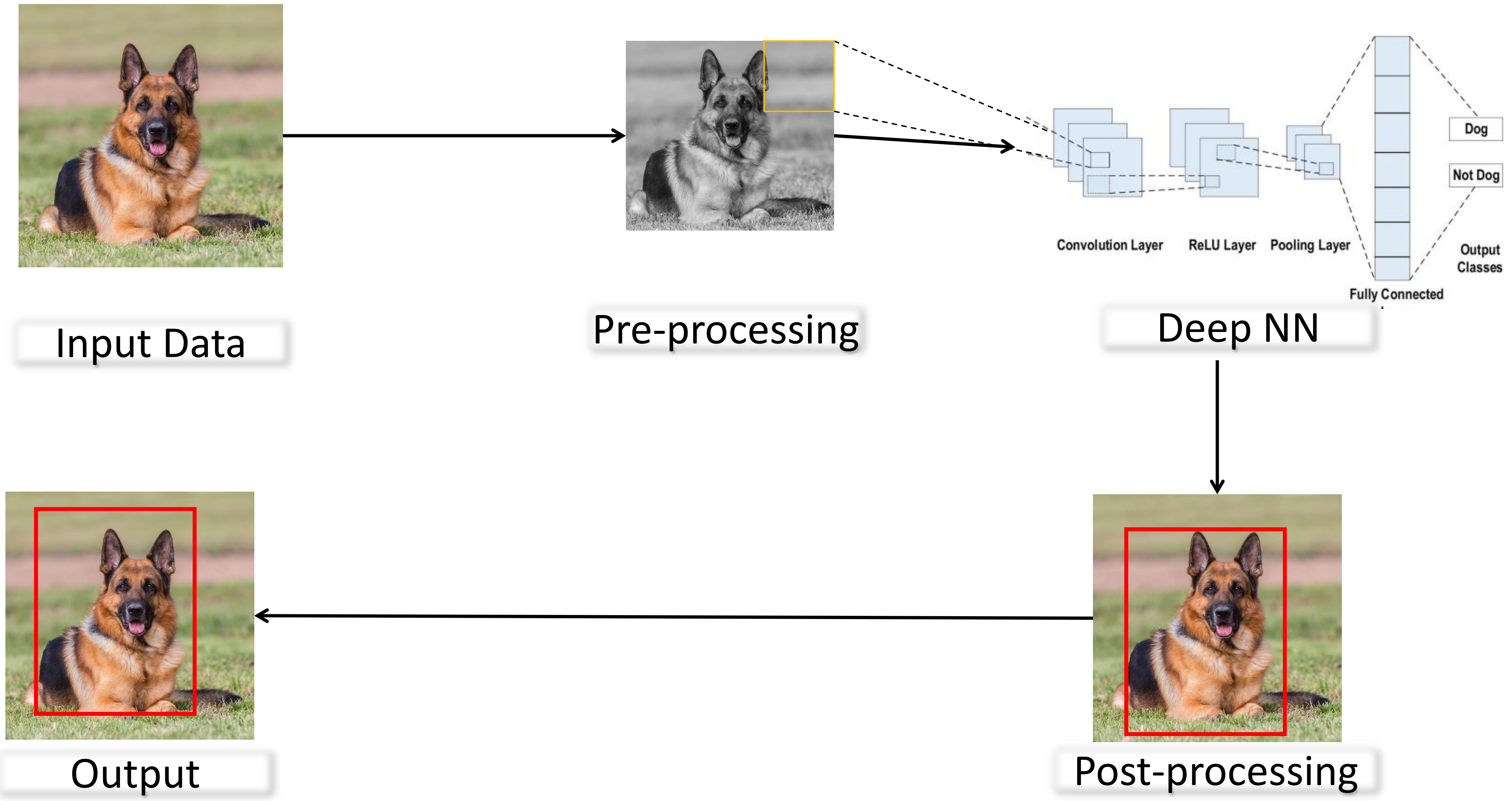
Computer Vision: AlexNet

- **AlexNet** is a CNN architecture developed for image classification tasks, notably achieving prominence through its performance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC).
- It has **8 layers (5 conv + 3 fully connected)** and has been trained through GPU parallelization.



https://blog.paperspace.com/understanding-interpreting-convolutional-neural-network-architectures/

Computer Vision: Pipeline



Lecture Overview

- Introduction
- Image Processing
  - Histogram Equalization
  - Noise Removal
  - Edge Detection
- Computer Vision
  - Convolutional Neural Networks
    - ✓VGG16
    - ✓AlexNet
- **Cognitive Vision**

Cognitive Vision: Principles

Is perception only about reconstructing a 3D world from images, or if there's more to it — such as reasoning, learning, and predicting?

- Computer vision → 3D descriptions: Traditional computer vision focuses on building 3D models of the world — identifying objects, actions, shapes, etc.
- Labels → symbolic systems: These labels (like “car”, “person”, “running”) are then fed into systems that make decisions or reason symbolically.

Cognitive Vision: Principles

- In many AI systems, perception is treated as a **black box** that just turns raw images into labels.
- The system uses mostly static data (like images or single video frames) and doesn't really "**understand**" .
- Moving from raw pixel data to **high-level symbolic understanding** (e.g., "*John is picking up the red ball to throw it*") is not easy.
- There's often no causal link —system **doesn't understand why things happen**, or how the present relates to the past.
- Because of this, such systems struggle to **anticipate the future.**

Cognitive Vision: Principles

Is perception only an inference process?

- Signal analysis is not enough to understand a scene.
- Additional knowledge through inference – as we look at the world, we think about it.

Cognitive vision continuously exchanges information between perception and reasoning.

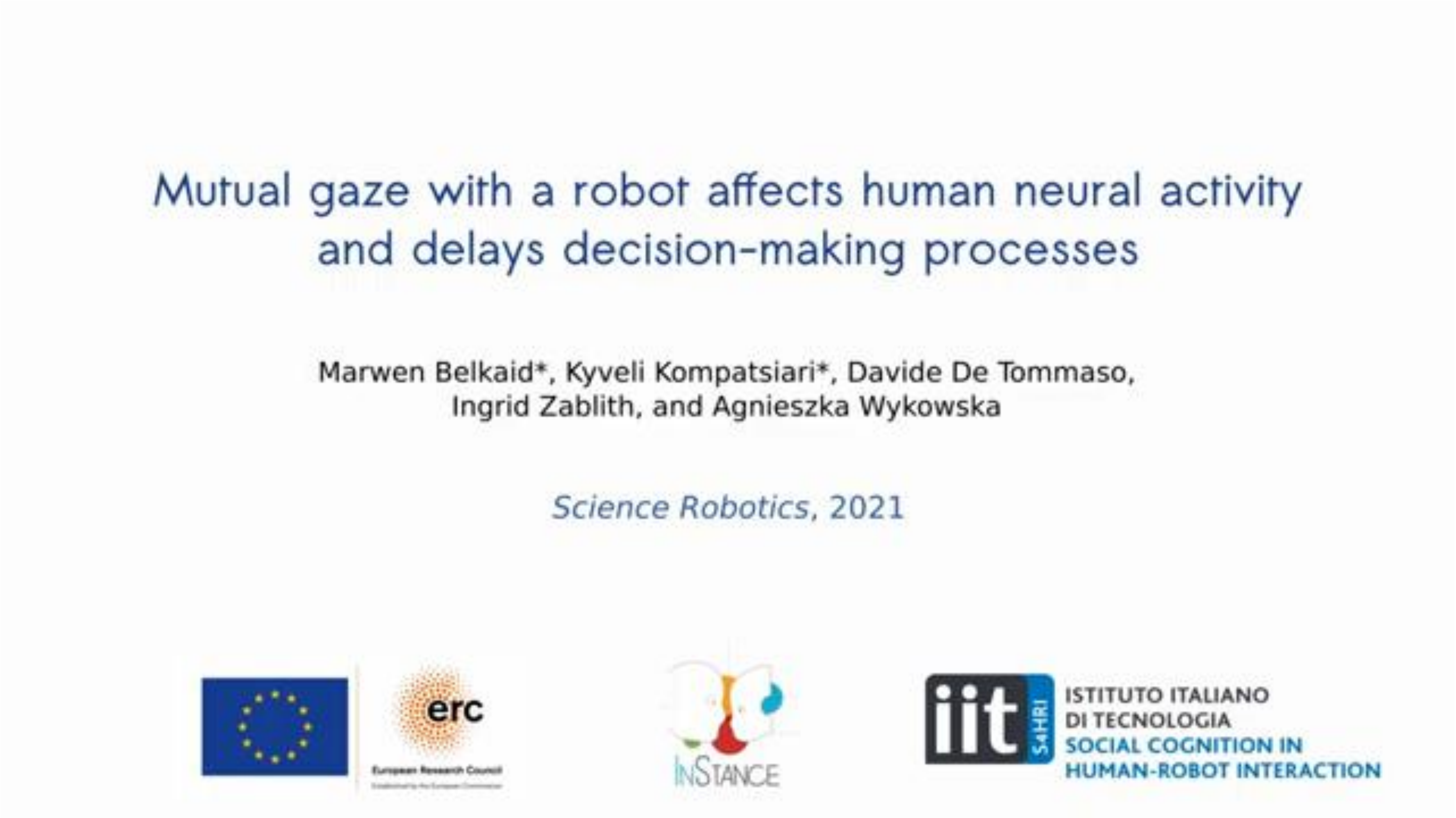
Actions driven by perceptual expectations – how should I act to see my hand close to the object vs. how should I act to reach the object?

Cognitive Vision: Vision and Reasoning Interaction

- **Cognitive vision** instead of just extracting features (like edges or shapes), it aims to **understand and reason** about what’s happening in the scene
  - The system **predicts** what it expects to see or sense.
  - Then it **explores** to check if its predictions are correct.
- This **prediction–exploration loop** helps the system go from just labeling things to truly interacting with and learning about the world, like humans do.
- So, **cognitive vision** is about combining **perception with reasoning, learning, and action**, enabling systems to adapt, anticipate the future, and deal with uncertainty.

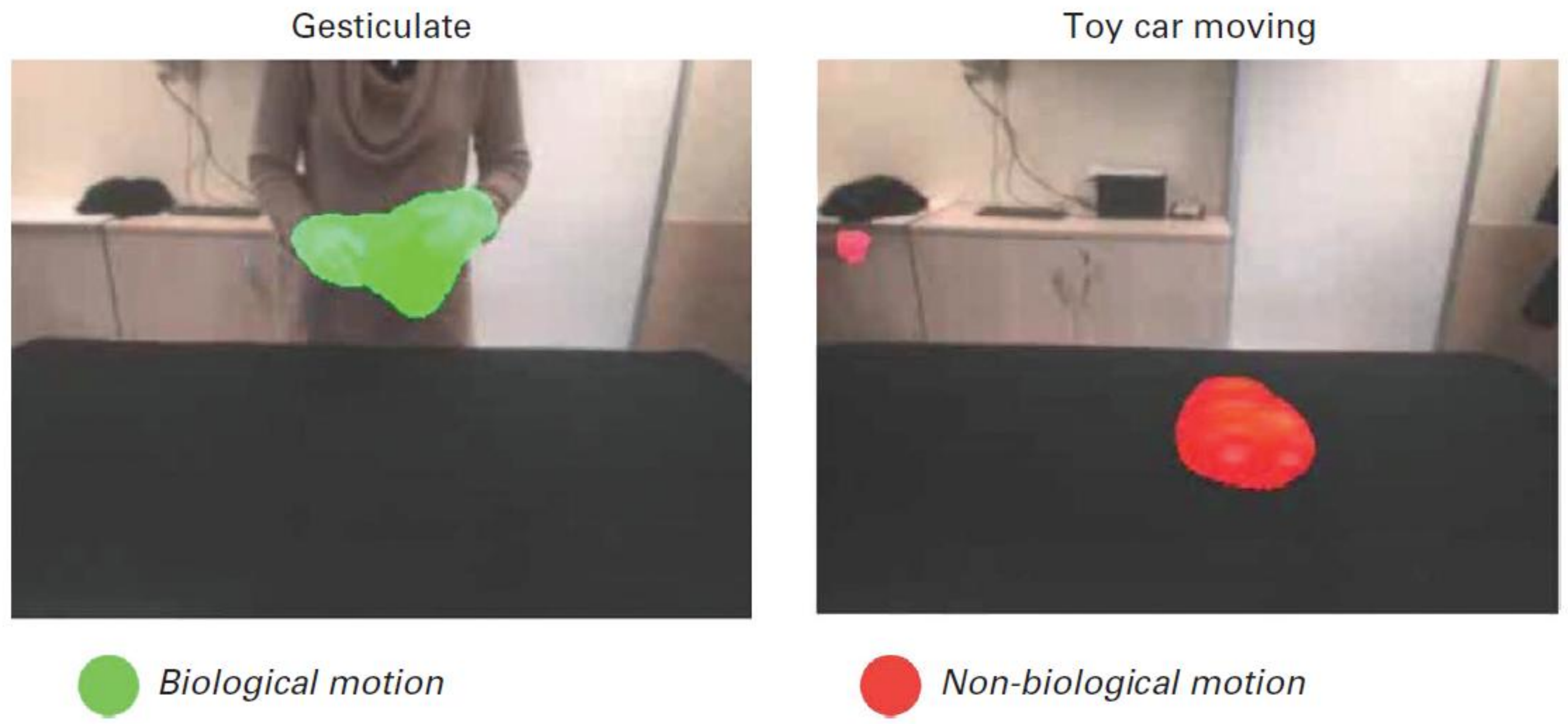
Cognitive Vision: Vision and Reasoning Interaction

- Cognitive vision to support human-robot interaction.
- iCub’s behavior driven only by the direction of the subject’s gaze making explicit intention to reach for the left or right hand.



Cognitive Vision: Vision and Reasoning Interaction

- Cognitive vision can be used for signature of biological motion.
- Angular velocity and curvature of the trajectory are features that CV systems might analyze to identify biological motions.
    - Hand during drawing or writing.
    - Knee or ankle during walking.
- Visually measured independently of its shape and color.



Cognitive Vision: Vision and Reasoning Interaction

- Cognitive Vision is not just about identifying **"what"** is in the scene and **"where"** it is.

- Most current vision systems treat perception as a **standalone task**—just detect and classify. Cognitive vision **challenges** this narrow view.

- Cognitive vision **combines V** and **R** to drive **action**. It's a **multi-modal system**—vision works together with other senses and cognitive functions.

    - Questions beyond what, where → why, how, who.

    - Also how synthesize visual information to anticipate action effects.

References

- OpenCV: real-time optimized Computer Vision library. https://opencv.org/
- Nilsson, N. J. (1998). *Artificial intelligence: a new synthesis*. Morgan Kaufmann. Chapter 6.
- Aloimonos, Y., & Sandini, G. Principles of Cognitive Vision. In Cangelosi, A., & Asada, M. (Eds.). (2022). *Cognitive robotics*. MIT Press. Chapter 14.
- Davies, E. R. (2018). Computer vision: principles, algorithms, applications, learning. Academic Press.
- Semantic Segmentation of Vegetation in Remote Sensing Imagery Using Deep Learning. September 2022. DOI: 10.48550/arXiv.2209.14364. Alexandru Munteanu, Marian Neagul.