Convolutional Neural Networks(CNNs)

Revolutionizing Image Processing

Sina Moghimi, Autumn 2024

МФТИ

Outline

- Image Processing Fundamentals
- Convolution Operation Basics
- Kernel/Filter Concepts
- Spatial Hierarchies
- Convolutional Layer Mechanics
- Pooling Operations
- CNN Architecture
- Practical Applications

Image Processing

- Is a technique of performing operations on an image to
  - enhance
  - extract information
  - Prepare it for specific applications such as
    - **Computer vision**
    - **Medical Imaging**
    - **Photography**
    - **Etc.**
  - Consider an **image** as a **grid of numbers**, where **each cell** in that grid is called **pixel**

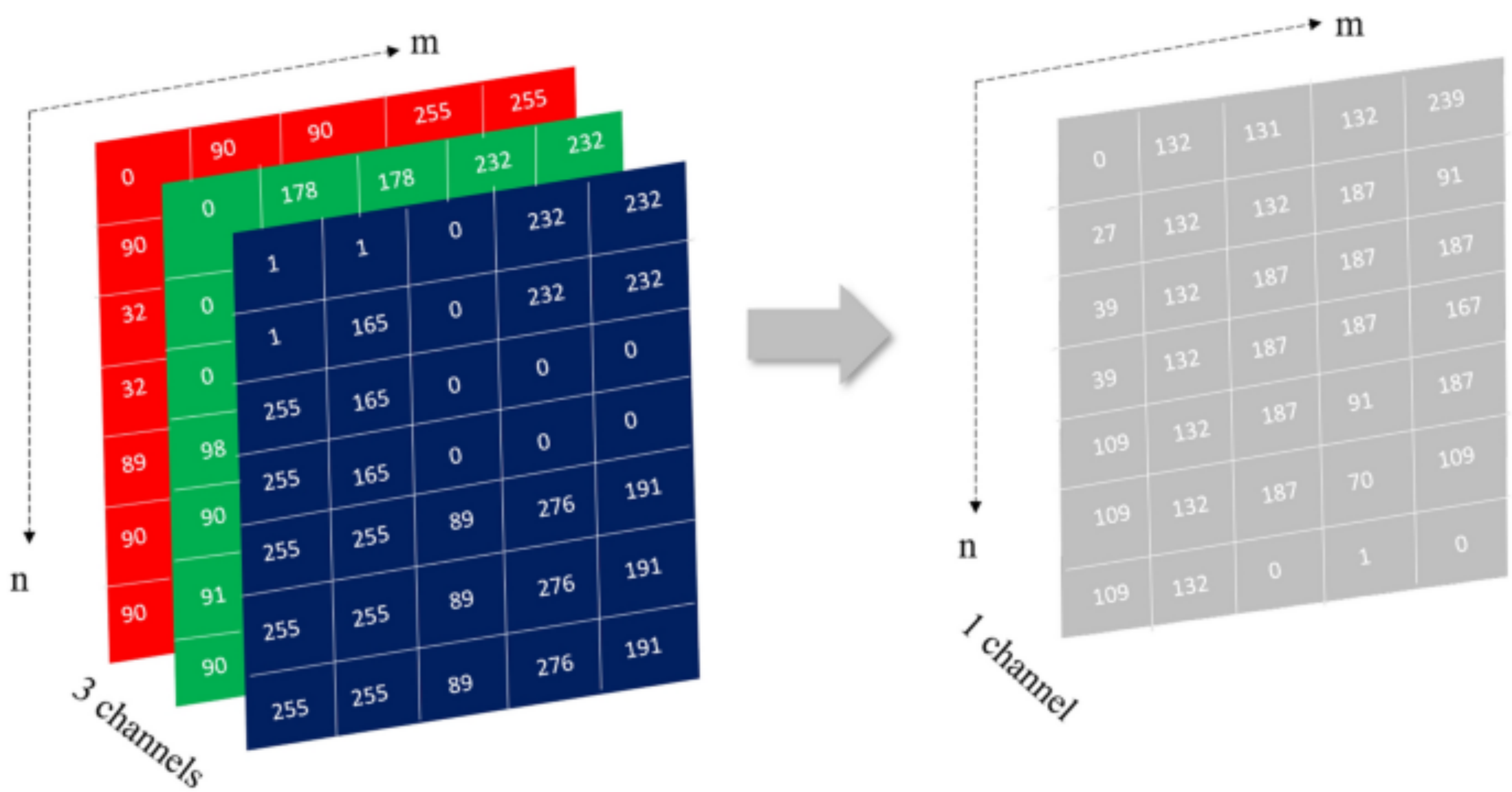
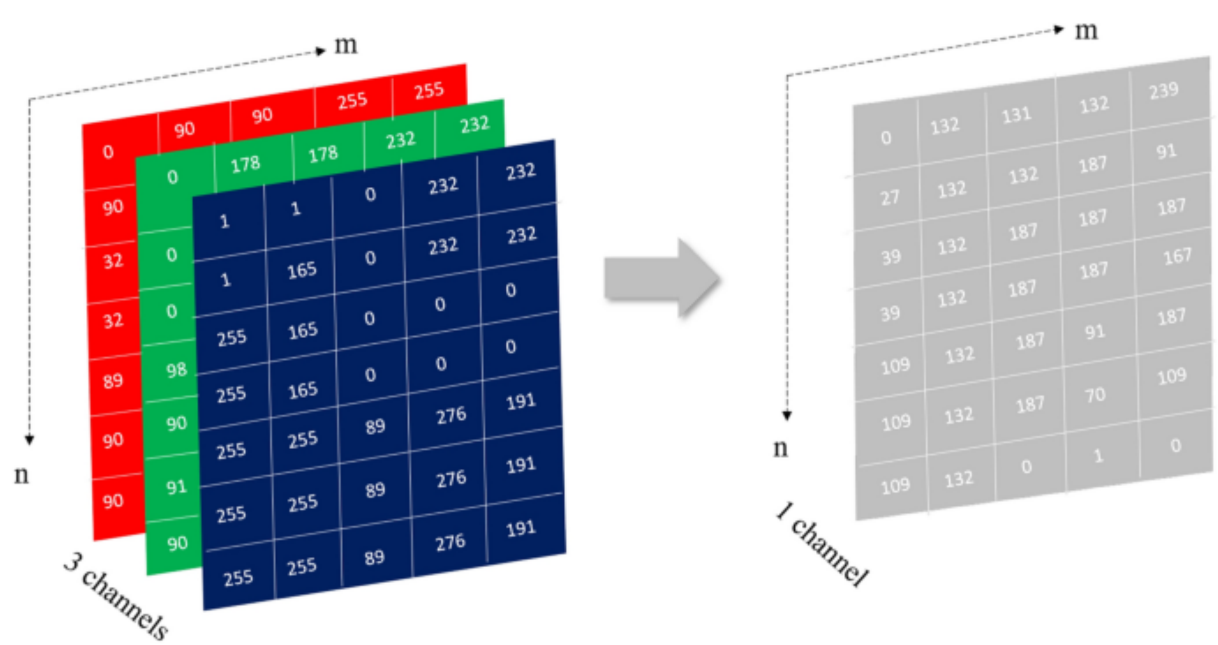


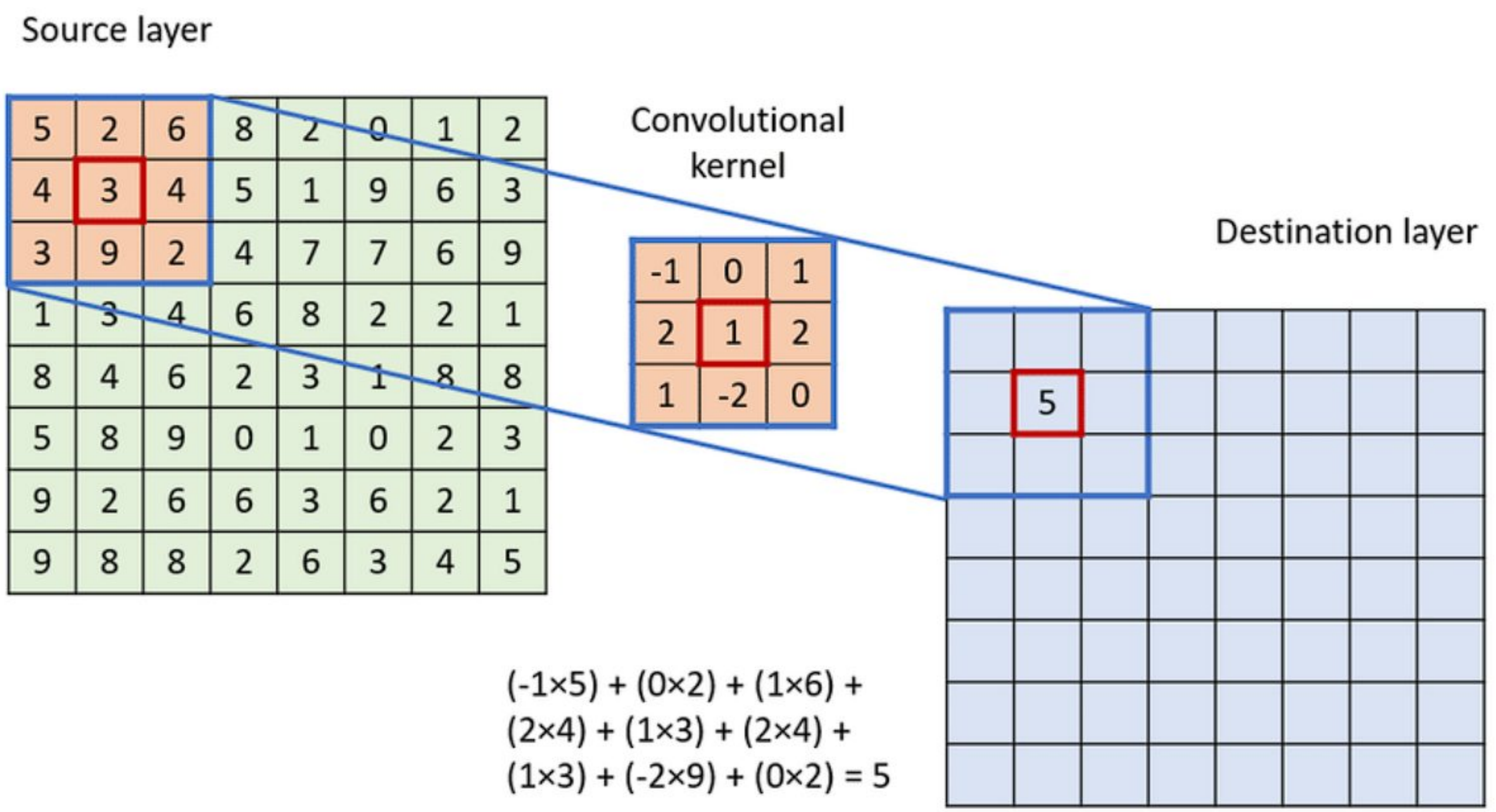
Image Processing

- A digital image is a representation of a real-world image in a computer-readable format.
  - It is a 2 dimensional array of numerical values.
  - Some definitions
    - **Resolution**: Width x Height
      - **Width =** # rows = m
      - **Height** = # columns = n
    - **GrayScale image:**
      - A single channel where each pixel's value represents its intensity, ranging from **0 (black)** to **255 (white)**
    - **Color image:**
      - Contains multiple channels, typically **Red**, **Green** and **Blue** to represent a full range of colors **(RGB)**



Convolution Operation

- **Convolution** is a fundamental mathematical operation in image processing, widely used for edge detection, filtering and feature extraction.
- Convolution operation, includes a small matrix called kernel/filter, which is applied to an image to produce a transformed version

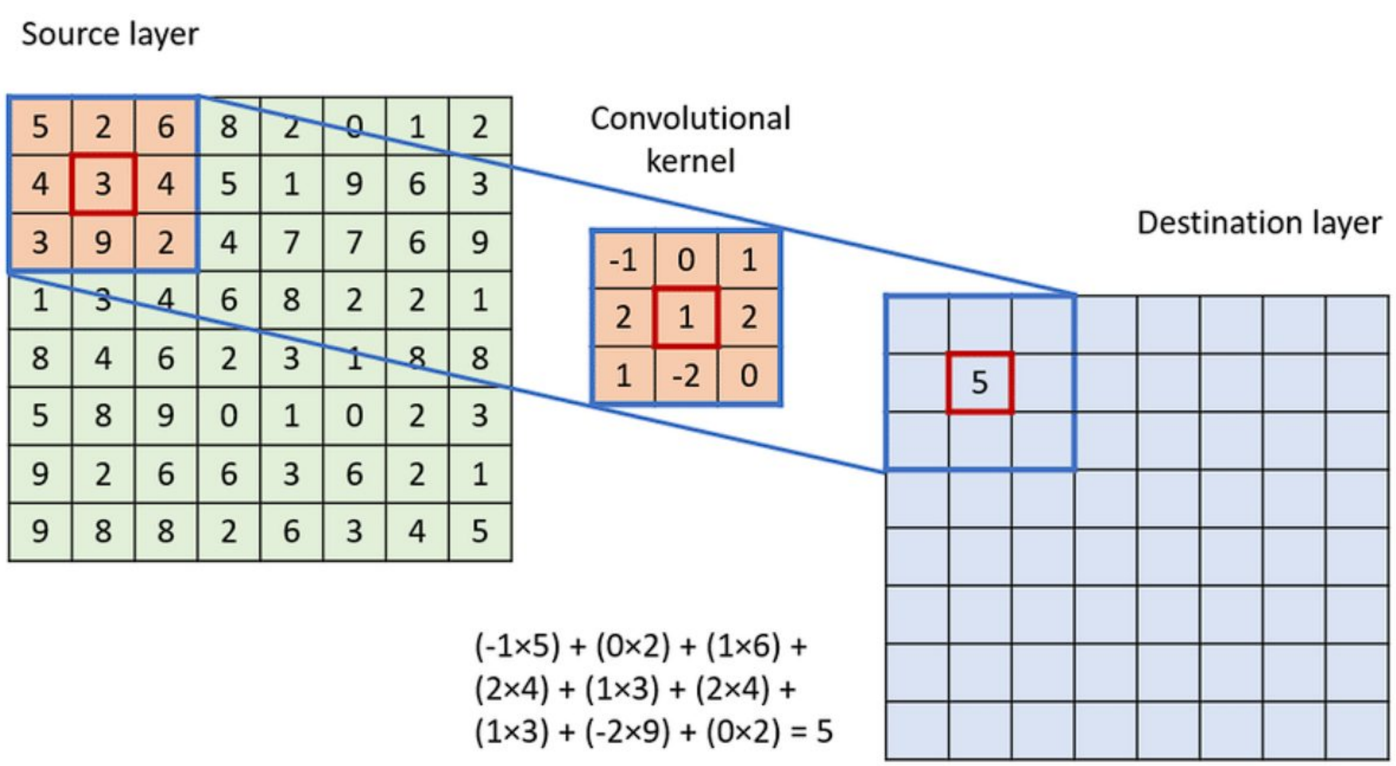


Convolution Operation

- **Mathematics**
  - Convolution in a 2D space is expressed as

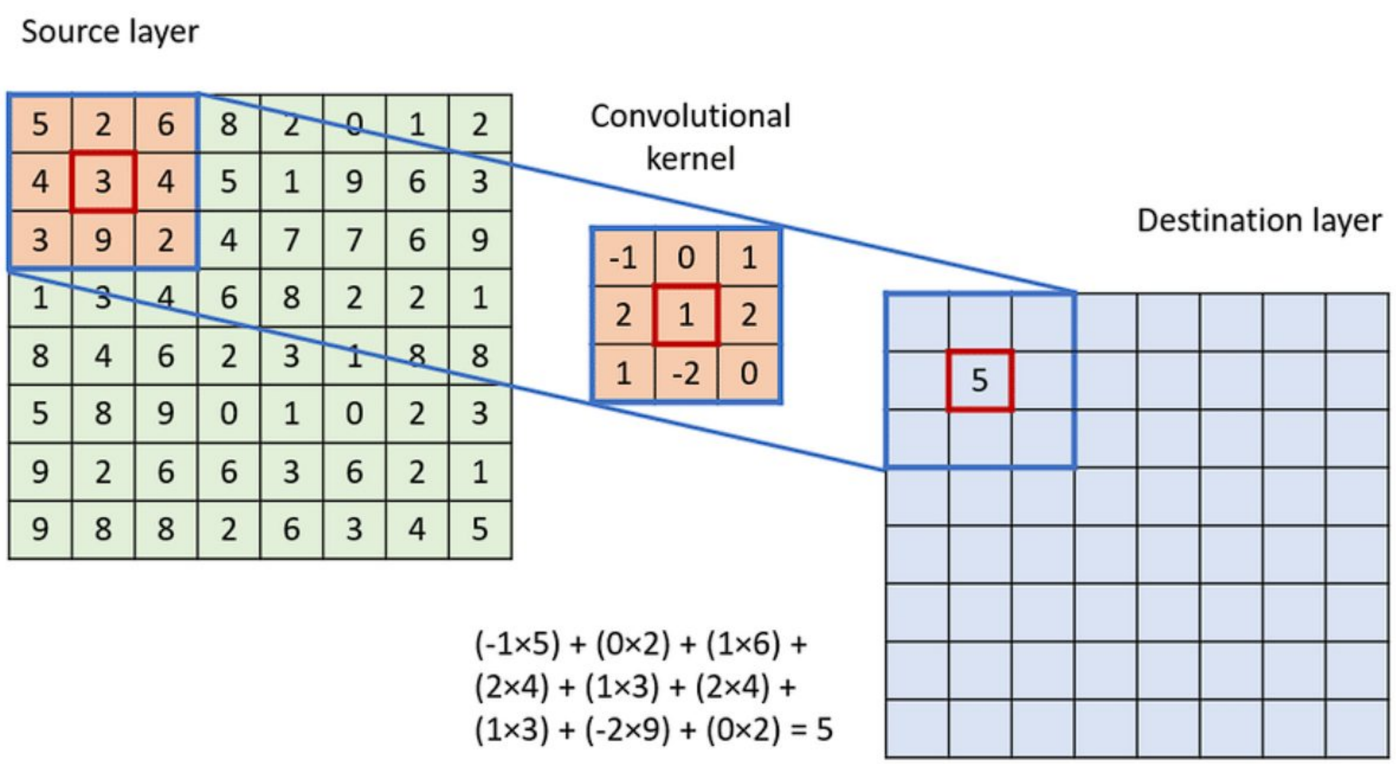
$$G(x, y) = \sum_{i=-k}^{k} \sum_{j=-j}^{k} K(i,\ j)\,.\,I(x-i,\ y-j)$$

    - $G(x, y)$ is the output pixel value at position (x, y)
    - $K(i, j)$ is the kernel matrix of size (2k + 1) x (2k + 1)
    - $I(x - i,\ y - j)$ is the input image pixel at position shifted by (i, j)



Convolution Operation

- **Mechanism**
  - Place the kernel over a section of the input image
  - Perform element-wise multiplication between the kernel and the corresponding region in the image
  - Accumulate all the resulting values to produce a single number
  - Repeat the process for every position in the image (using a sliding mechanism)

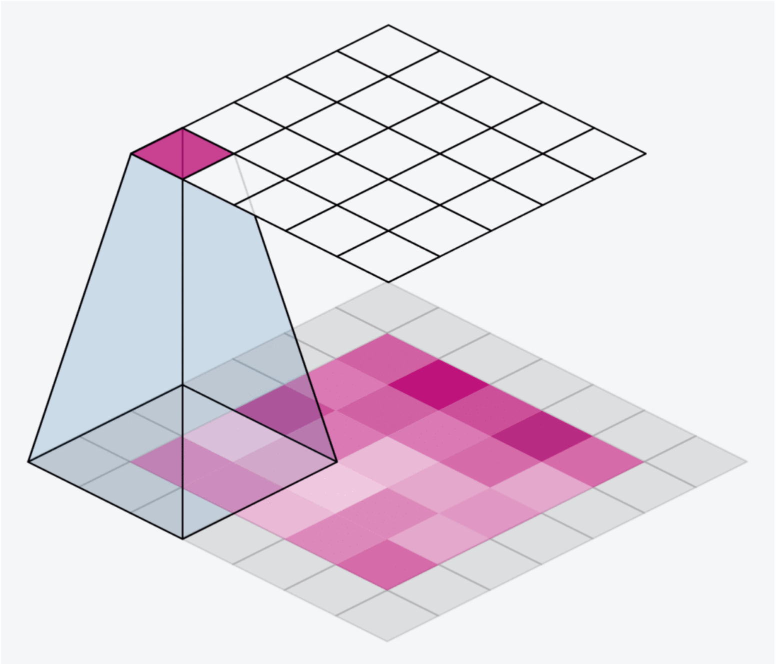


Convolution Operation

- **Sliding Window**
  - Place the kernel at top-left corner of the image (image origin)
  - Compute the dot product between the kernel and the overlapping region of the image
  - Move the kernel to next position (by step size (**stride**), e.g. 1 pixel at a time) and repeat
  - Continue until the kernel has covered the entire image
  - **Key Parameters**
    - **Stride**: # of pixel y which the kernel shift after each computation
    - **Padding:** adding extra borders to the image to ensure the kernel can cover edge regions completely
  - Input-Output Relation

$$size_{out} = \lfloor \frac{size_{in} + 2.P - K}{S} \rfloor$$

    - P: Padding
    - K: Kernel size
    - S: Stride



https://analyticsindiamag.com/developers-corner/complete-guide-to-transposed-convolutions-in-cnn-models/

Convolution Operation

- **Some Important Filters/Kernels**
  - **Smoothing Filters**
    - Reduce noise and blur the image
    - Example: Box Filter of size 3

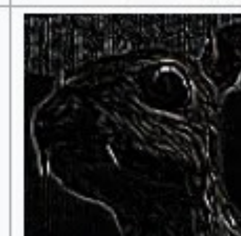
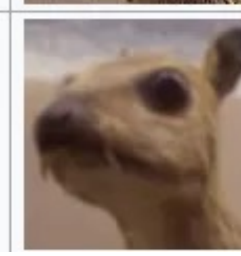
$$\frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

  - **Sharpening Filters**
    - Enhance edges and fine details by amplifying differences between adjacent pixels
    - Example:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

  - **Edge Detection Filters**
    - Highlight regions with high intensity changes (edges)
    - Example: **Sobel**, **Prewitt** and **Laplacian Filters**

Convolution Operation

Operation	Kernel ω	Image result g(x,y)
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

https://medium.com/skylar-salernos-tech-blog/image-convolution-filters-explained-c878f1056e78

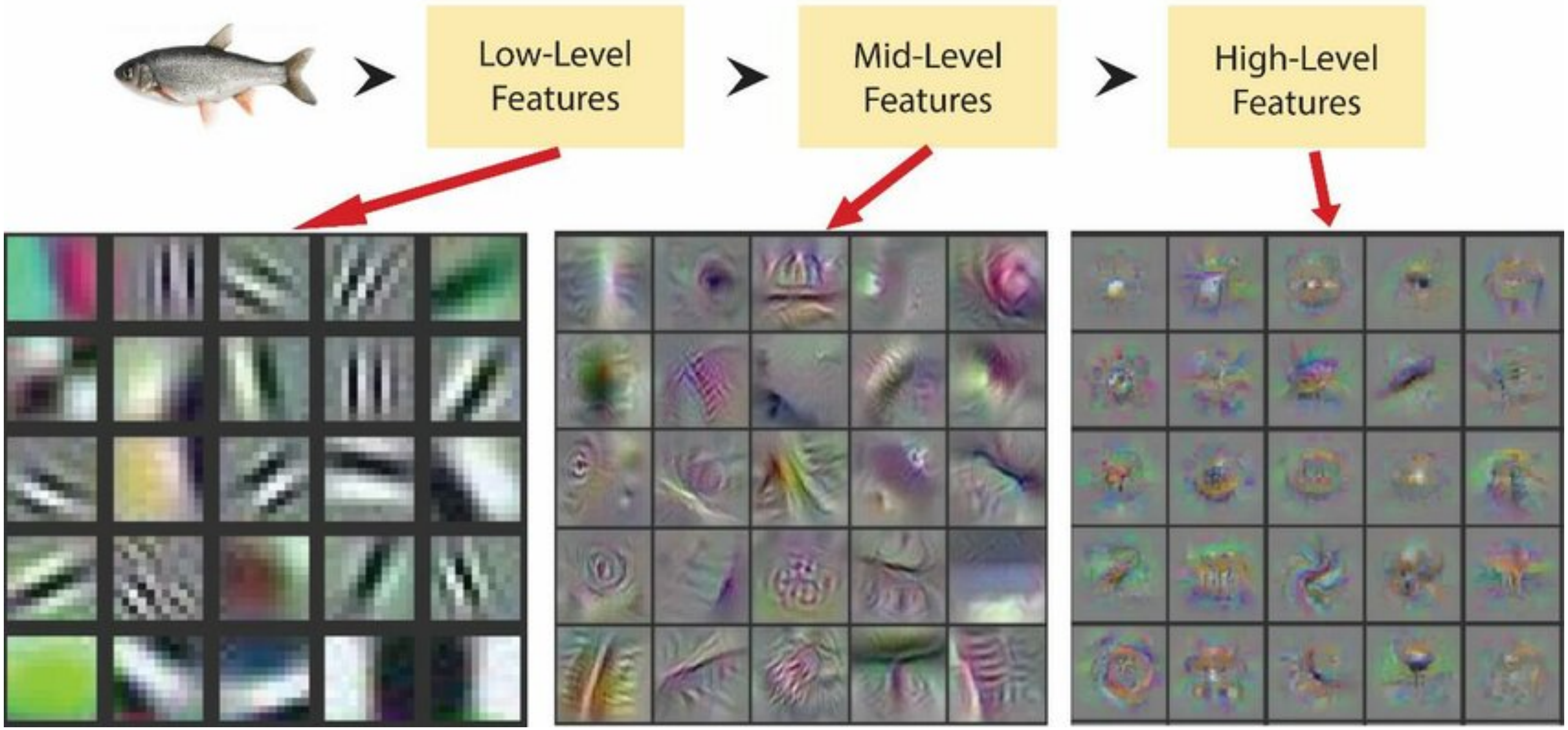
Feature Extraction

- Use kernels to identify patterns, shapes and textures
  - **Common Techniques**
    - **Harris Corner Detector:** Finds corners by calculating intensity variations in all directions using a kernel
    - **Gabor Filters:** Detect specific frequency and orientation patterns
    - **CNN Filters:** In deep learning, convolutional layers learn filters automatically during training to detect hierarchical features (e.g., edges, shapes, complex patterns)

Spatial Hierarchies

- **Layered Feature Learning**
  - **Low Layers:**
    - detect basic, small-scale patterns (e.g., edges, corners, or textures)
  - **Middle layers**
    - combine these patterns to form more complex structures (e.g., shapes or object parts)
  - **Higher layers**
    - interpret these structures into meaningful abstractions (e.g., entire objects)

Spatial Hierarchies



Automatic fish species classification in underwater videos: Exploiting pretrained deep neural network models to compensate for limited labelled data

Pooling Operations

- Pooling operations are used to reduce the spatial dimensions of feature maps while retaining the most important informations
- Pooling Operators, have a window size and extract a number from the corresponding window in the image
- Famous Pooling Operators
    - **Max Pooling**
        - It extracts the max number in that window
    - **Average Pooling**
        - It averages all the values in that window
- **Dimension Reduction**
    - Considering we have an image of size H x W, the output image would be of size $H_{out} \times W_{out}$ considering window size **K**

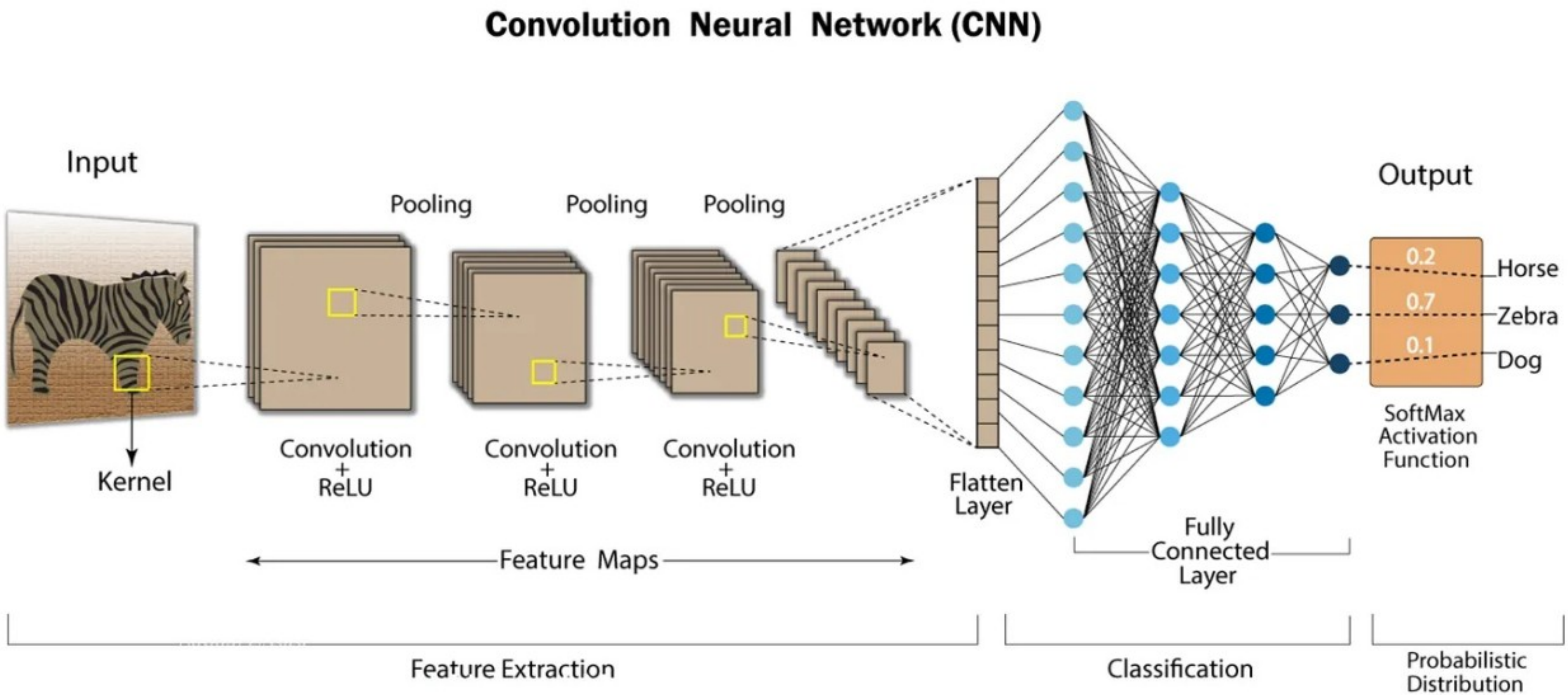
$$H_{out} = \frac{H_{in} - K}{S} + 1, \quad W_{out} = \frac{W_{in} - K}{S} + 1$$

МФТИ

Pooling Operations

Feature	Max Pooling	Average Pooling
Focus	Retains the strongest feature	Provides smoother outputs
Purpose	Highlights key activations	Retains broader context
Use Case	Edge or pattern detection	Smoothing or noise reduction

CNN Architecture



https://www.linkedin.com/pulse/what-convolutional-neural-network-cnn-deep-learning-nafiz-shahriar/

Applications of CNN

- Image Classification
- Object Detection
- Segmentation
- Face Recognition
- Medical Image Analysis
- etc.