Convolution Neural Network (CNN)

Output Size After Convolution Layer

Formula for Convolution Layer Output Size

$$W_{out} = \frac{W_{in} - F + 2P}{S} + 1$$

$$H_{out} = \frac{H_{in} - F + 2P}{S} + 1$$

Where:

- W_{in} = Width of input image
- H_{in} = Height of input image
- F = Filter size (Kernel size)
- P = Padding (adding zeros around the image to preserve size)
- S = Stride (step size of filter)
- W_{out} = Width of output image
- H_{out} = Height of output image

Examples

1) Given:

Input Image Size = **32 × 32**

Filter Size = **5 × 5**

Stride = **1**

Padding = **0** (no padding)

2) Input Image Size = 32 × 32

Filter Size = 5 × 5

Stride = 1

Padding = 2

Output Size After Pooling Layer (Max/Average Pooling)

- Pooling reduces the spatial dimensions of the image. The most common type is **2x2 Max Pooling**

$$W_{out} = \frac{W_{in} - F}{S} + 1$$

$$H_{out} = \frac{H_{in} - F}{S} + 1$$

Where:

- **F** = Pooling filter size (commonly 2×2)
- **S** = Stride (commonly 2)

Max Pooling Layer

Given:

- Input Image Size = **28 × 28** (from previous Convolution output)
- Pooling Size = **2 × 2**
- Stride = **2**
- W out= (28-2)/2=(26/2)+1=14

Image size is :14*14

What is a Kernel in CNN

- A **kernel** is a small **matrix of weights** (usually of size **3x3, 5x5, or 7x7**) used to **extract features** like edges, colors, gradients, and textures from an image.
- It performs a **dot product operation** with the input image, producing an **output feature map** (also called an activation map).
- The kernel moves **across the input image** using a certain **stride** (step size).

Types of Kernels in CNN

Kernel Type	Purpose	Example
Edge Detection Kernel	Detects edges in images (horizontal, vertical)	Sobel, Prewitt, Scharr, Canny
Sharpening Kernel	Enhances edges and sharpens images	Laplacian Kernel
Blur Kernel	Blurs or smoothens the image	Gaussian Blur, Average Blur
Emboss Kernel	Highlights edges to give a 3D effect	Emboss Filter

Edge Detection Kernel (for detecting edges in images)

- When you want to detect edges like **borders, boundaries, or outlines** of objects in an image.
- Useful in tasks like **object detection, image segmentation, and feature extraction**.

Example: Sobel Kernel for Horizontal Edge Detection

Kernel Matrix (3x3):

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

- This kernel detects **horizontal edges** in an image.

Vertical Edge Detection Kernel

This kernel detects **vertical edges** in an image.

Kernel Matrix (3x3):

$$K_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Use Case:

- **Face detection** (detect facial boundaries).
- **Number plate recognition** (detect boundaries of the number plate).
- **Object recognition** (detect outlines of objects).

Blurring Kernel (for blurring or smoothing an image)

This kernel **smoothens the image** by averaging pixel values around a central pixel.

Kernel Matrix (3x3):

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

Use Case:

- **Preprocessing images** before feeding them to a CNN.
- **Reducing noise in medical images (X-ray, MRI)**.
- **Face recognition** (to remove background noise).

Sharpening Kernel (for enhancing edges and features)

- When you want to enhance edges, textures, or fine details in an image.
- Useful in object detection, face recognition, and medical imaging.
- Kernel Matrix:

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

This kernel increases the contrast around edges, making objects sharper.

Use Case:

- **Medical imaging** (highlight fractures in X-ray).
- **Object recognition** (enhance object boundaries).
- **Image enhancement** (sharpen blurry images)

Emboss Kernel (for 3D effect)

- When you want to give a 3D effect or depth to an image.
- Useful in artistic image processing, texture analysis, or stylized filters.

Kernel Matrix:

$$K = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

This kernel **highlights the edges** by giving them a **raised or embossed look**.

- **Artistic image processing.**
- **Texture analysis.**
- **3D effect generation.**

Identity Kernel (No effect, just pass the image as it is)

- When you want to simply pass the image as it is without modification.
- This kernel is useful in preprocessing pipelines when you don't want to alter the image.
- Kernel Matrix:

$$K = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The image will remain **unchanged** after applying this kernel.

Use Case:

- **No change** in the image.
- **Passing input directly to CNN without modification**

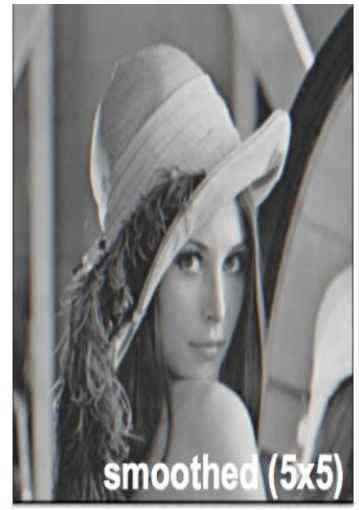
Original

Sharpen

Box Blur

Edge detection

−



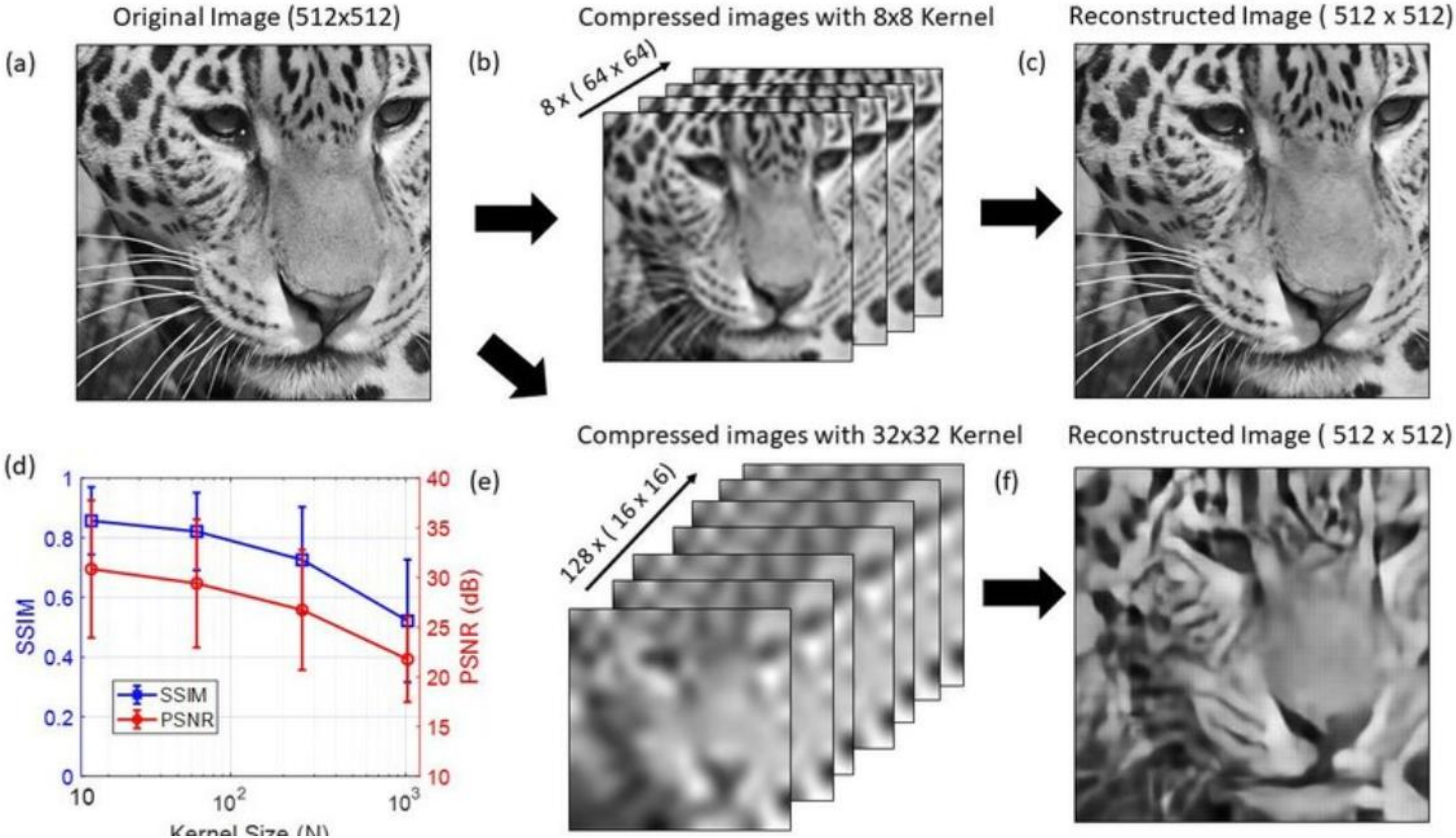
=

•0	•0	•0
•0	•1	•0
•0	•0	•0

−

$\frac{1}{9}$

•1	•1	•1
•1	•1	•1
•1	•1	•1


Original Image (512x512)
(a)
(b)
Compressed images with 8x8 Kernel
8 x (64 x 64)
(c)
Reconstructed Image (512 x 512)
(d)
SSIM
PSNR (dB)
SSIM
PSNR
Kernel Size (N)
(e)
Compressed images with 32x32 Kernel
128 x (16 x 16)
(f)
Reconstructed Image (512 x 512)