Convolutional Neural Networks

Adam Cardoza

Announcements

- Midterm is on!
  - Closes Thursday at 11:59pm
  - 3 hrs, one sitting, open notes, closed internet and AI
- No quiz today
- HW 8 (SuperRes) due next week Thursday (March 13)

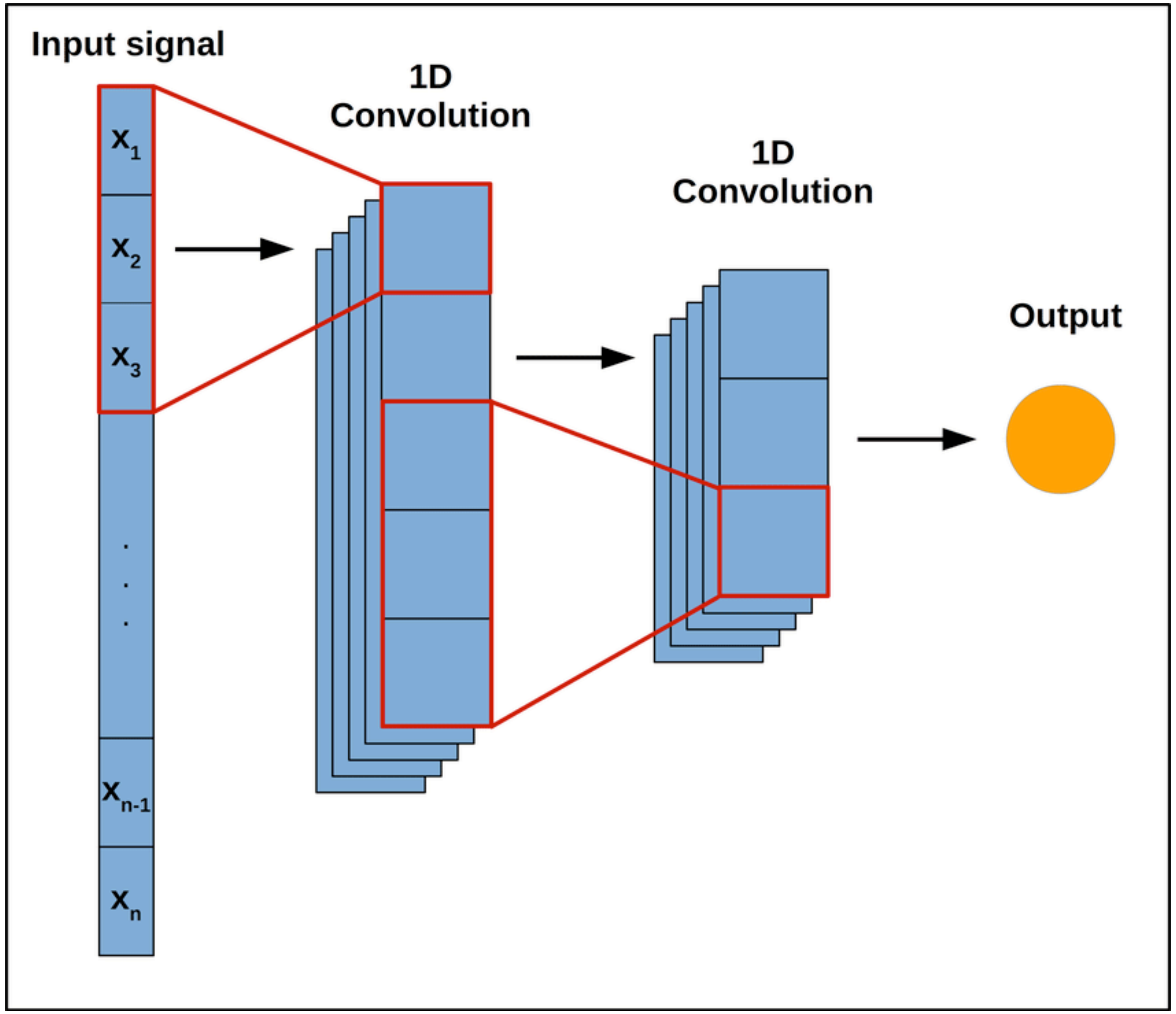
Plan for Today

- Review concepts from last time
- Building a NN with convolutional layers
- Downsampling
- Upsampling
- Application of CNN

What is a CNN?

- https://adamharley.com/nn_vis/cnn/2d.html

How does a convolutional layer work? - Main Idea



The layer is learning

how to compare/analyze inputs

Image: A Novel Deep Learning Model for the Detection and Identification of Rolling Element-Bearing Faults - Alex Shenfield, Martin Howarth, September 2020

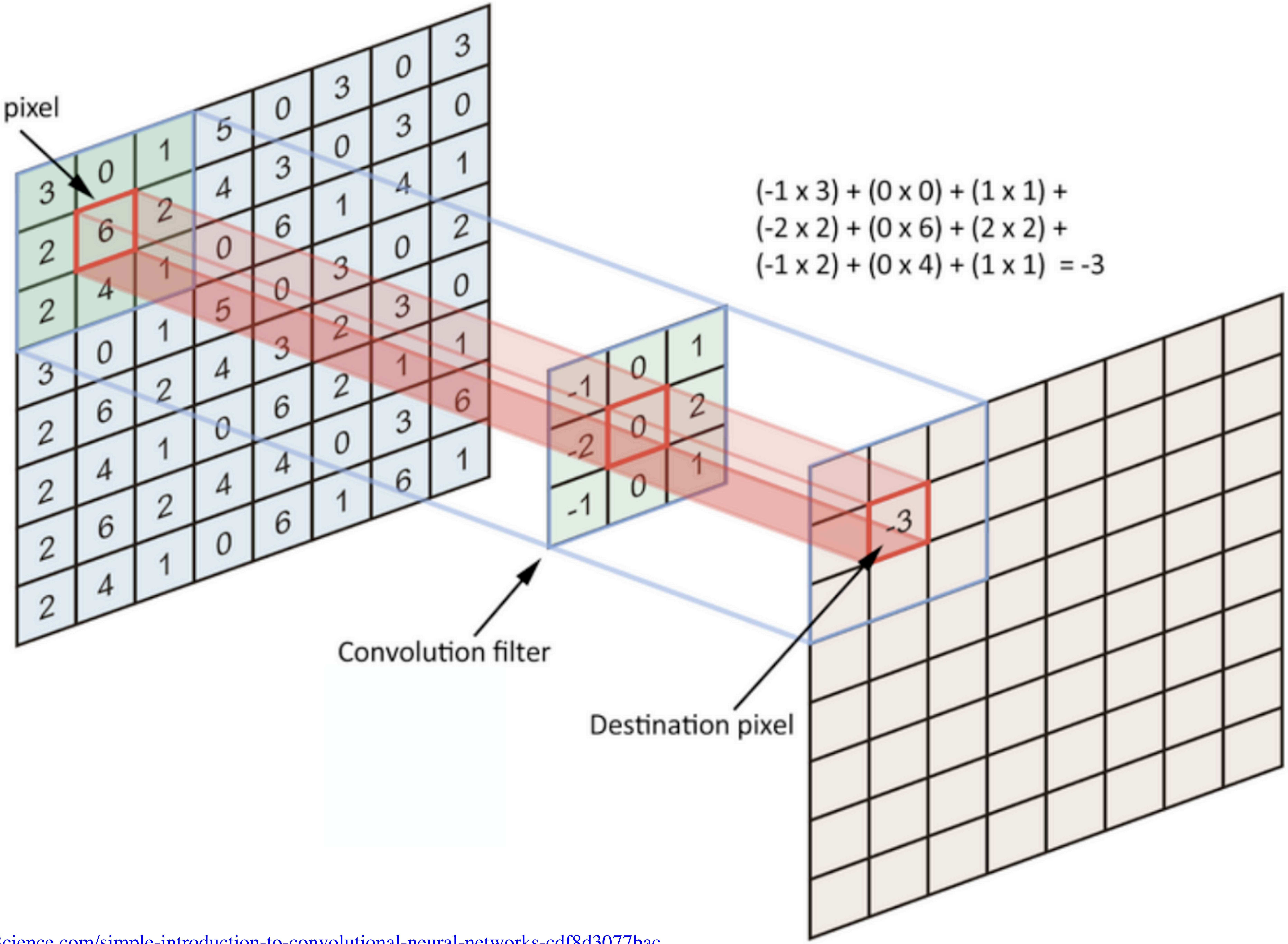


Image: TowardsDataScience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

Convolution Math (2D)

$$(I * K)_{i,j} = S_{ij} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I_{i+m,j+n} \cdot K_{m,n}$$

- I is the input
- K is the weight of the kernel (What we're learning)
- Dimension of K is the kernel size

Padding

0	0	0	0	0	0	0
0	1	2	3	4	5	0
0	6	7	8	9	0	0
0	-1	-2	-3	-4	-5	0
0	-6	-7	-8	-9	0	0
0	1	5	6	3	-4	0
0	0	0	0	0	0	0

Stride

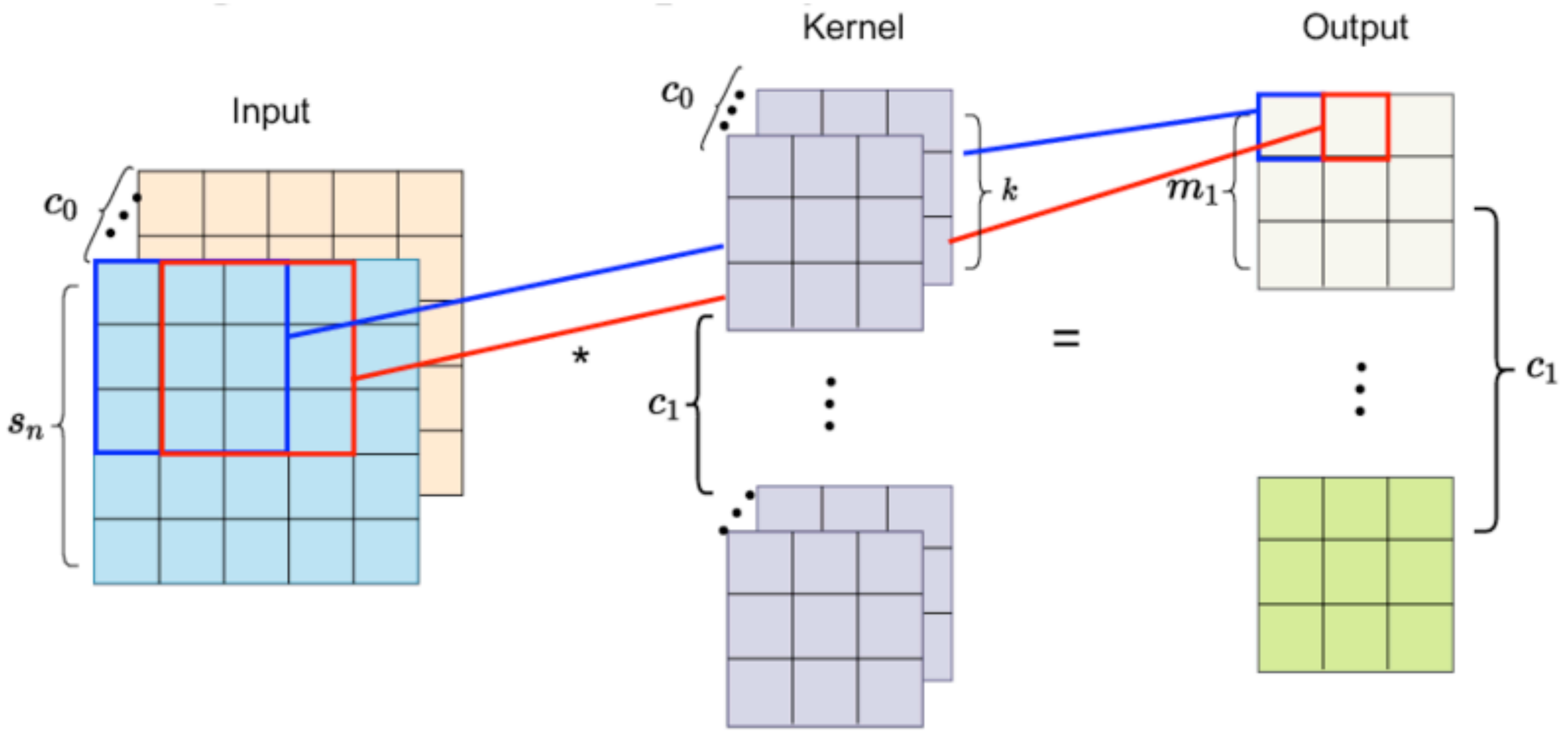
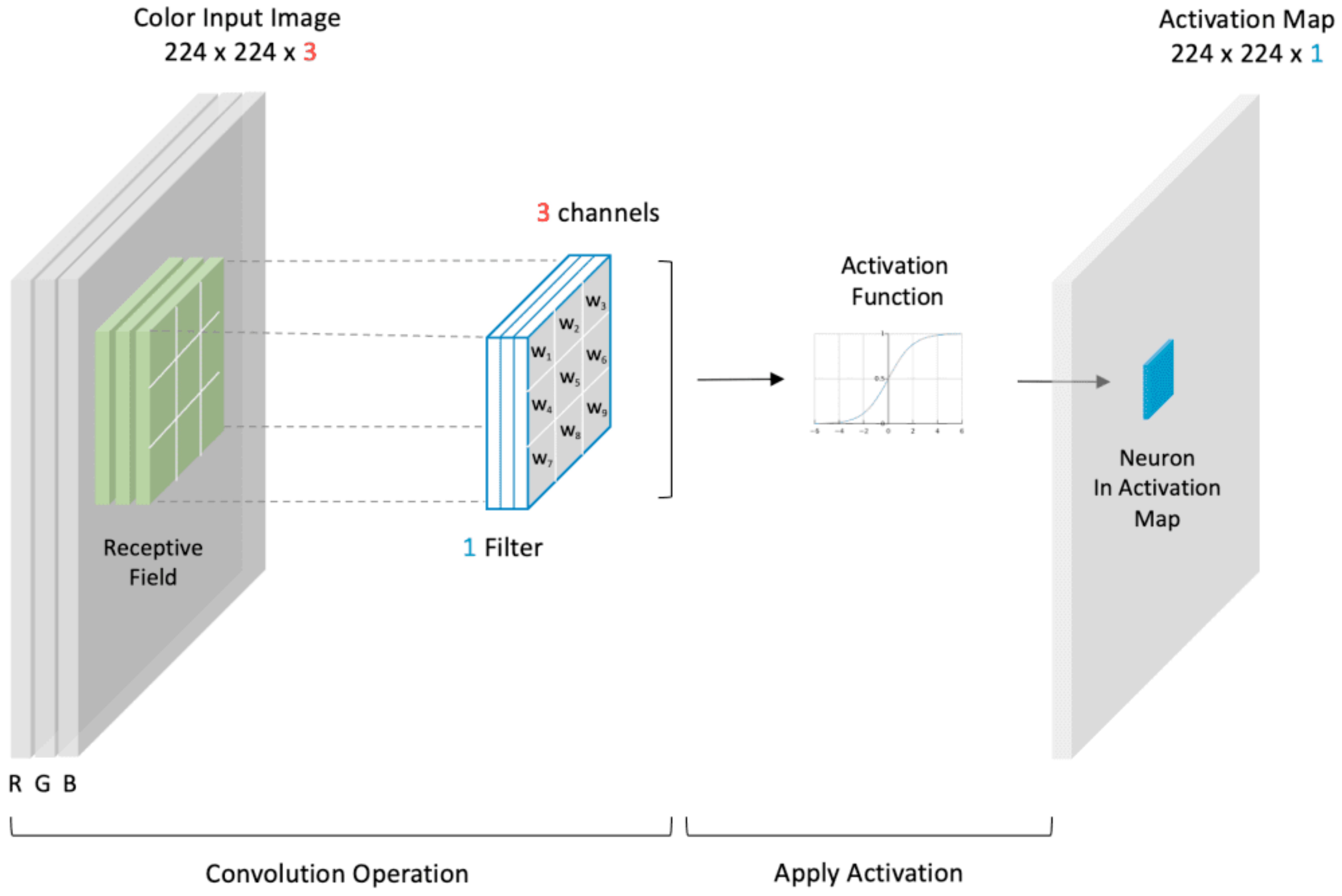


Image: Resource Allocation of Federated Learning for the Metaverse with Mobile Augmented Reality, Xinyi Zhou, Chang Liu, Jun Zhao - November 2022

Channels



[Image: Understanding Convolutional Neural Networks (CNNs) - pavlohak - https://www.oksim.ua/2024/02/01/understanding-convolutional-neural-networks-cnns/](https://www.oksim.ua/2024/02/01/understanding-convolutional-neural-networks-cnns/)

N-D Convolutions

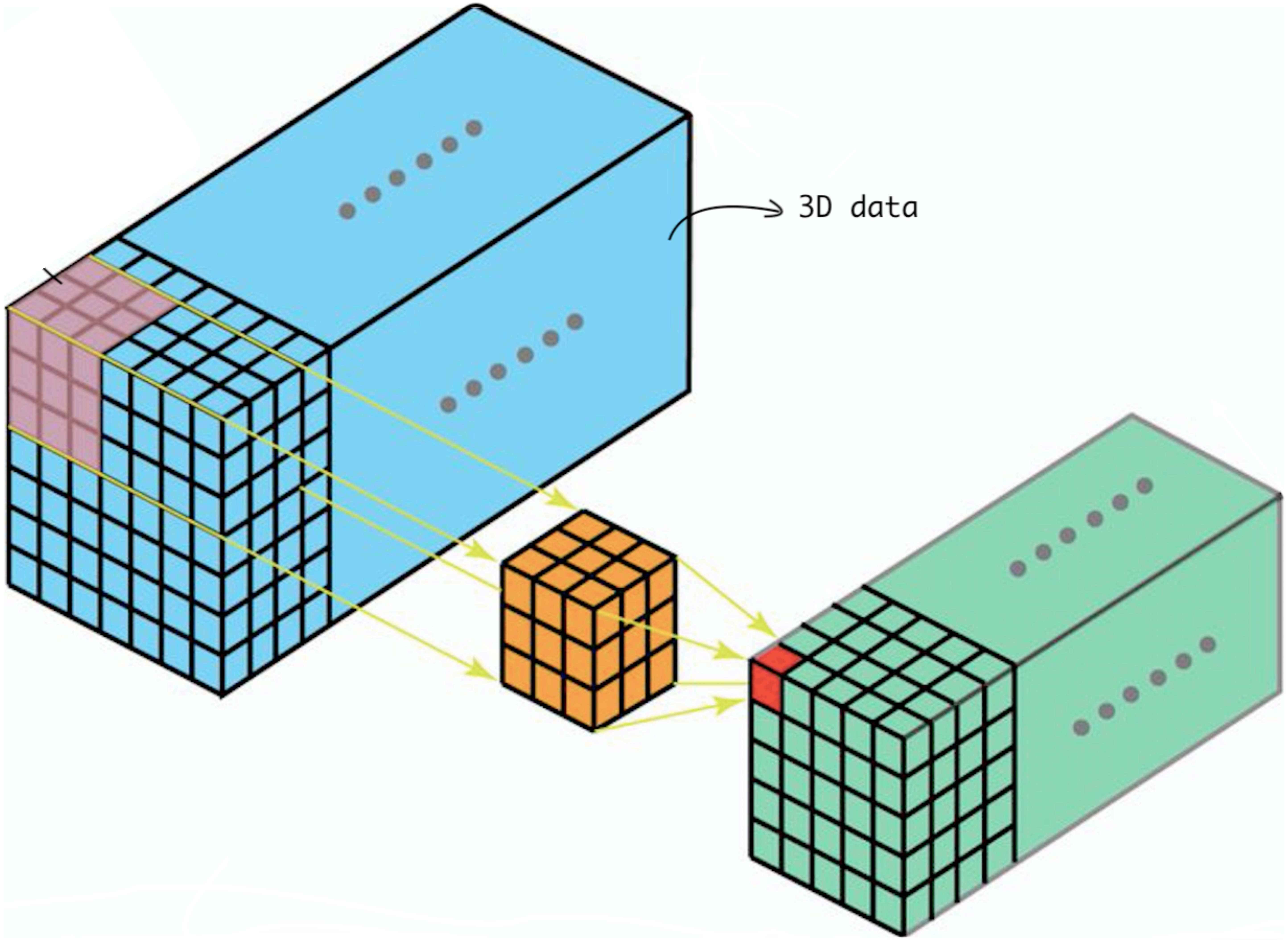


Image: Understanding 1D and 3D Convolution Neural Network - Shiva Verma - https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610?

Building a network using convolutional layers

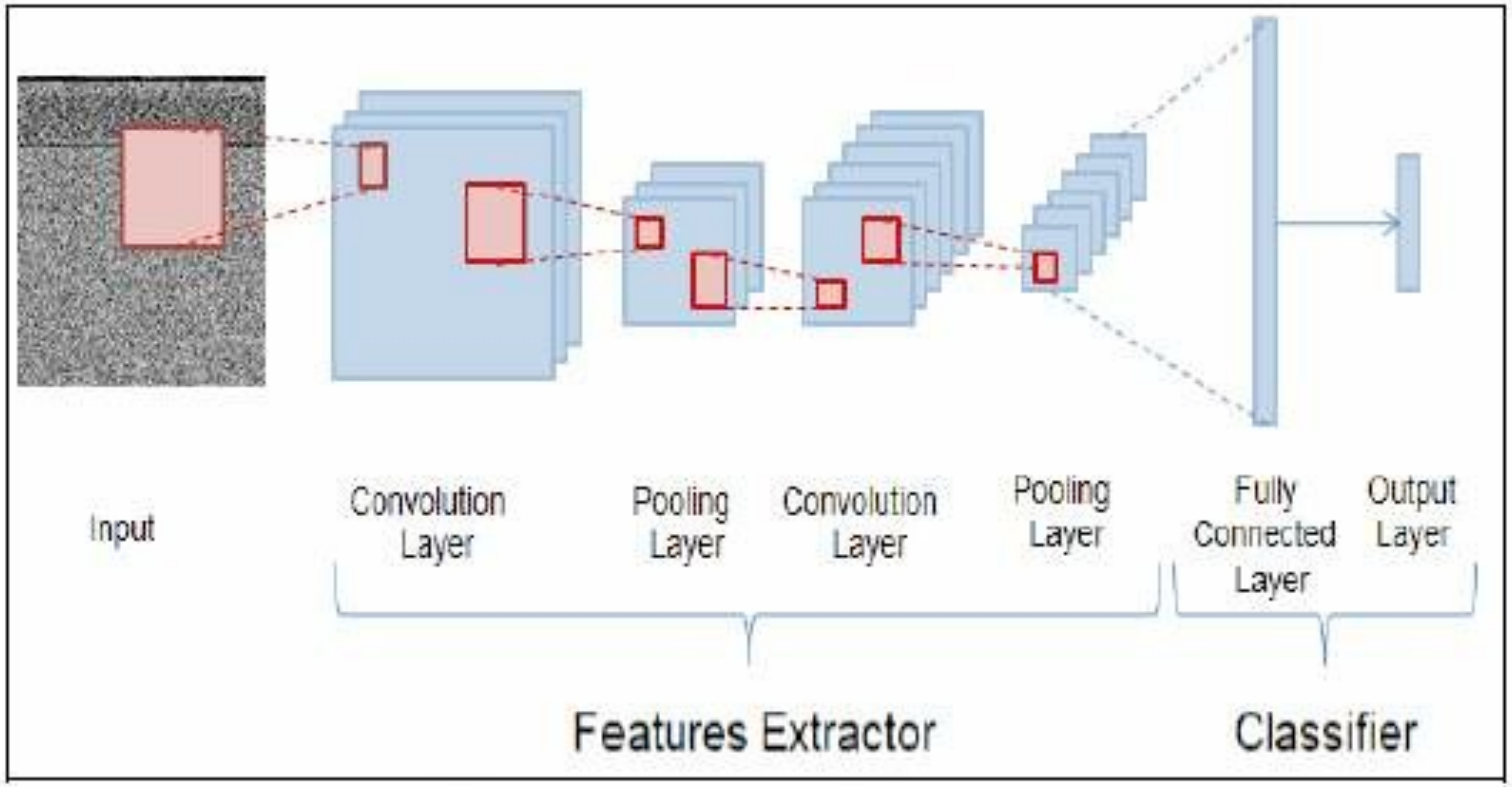
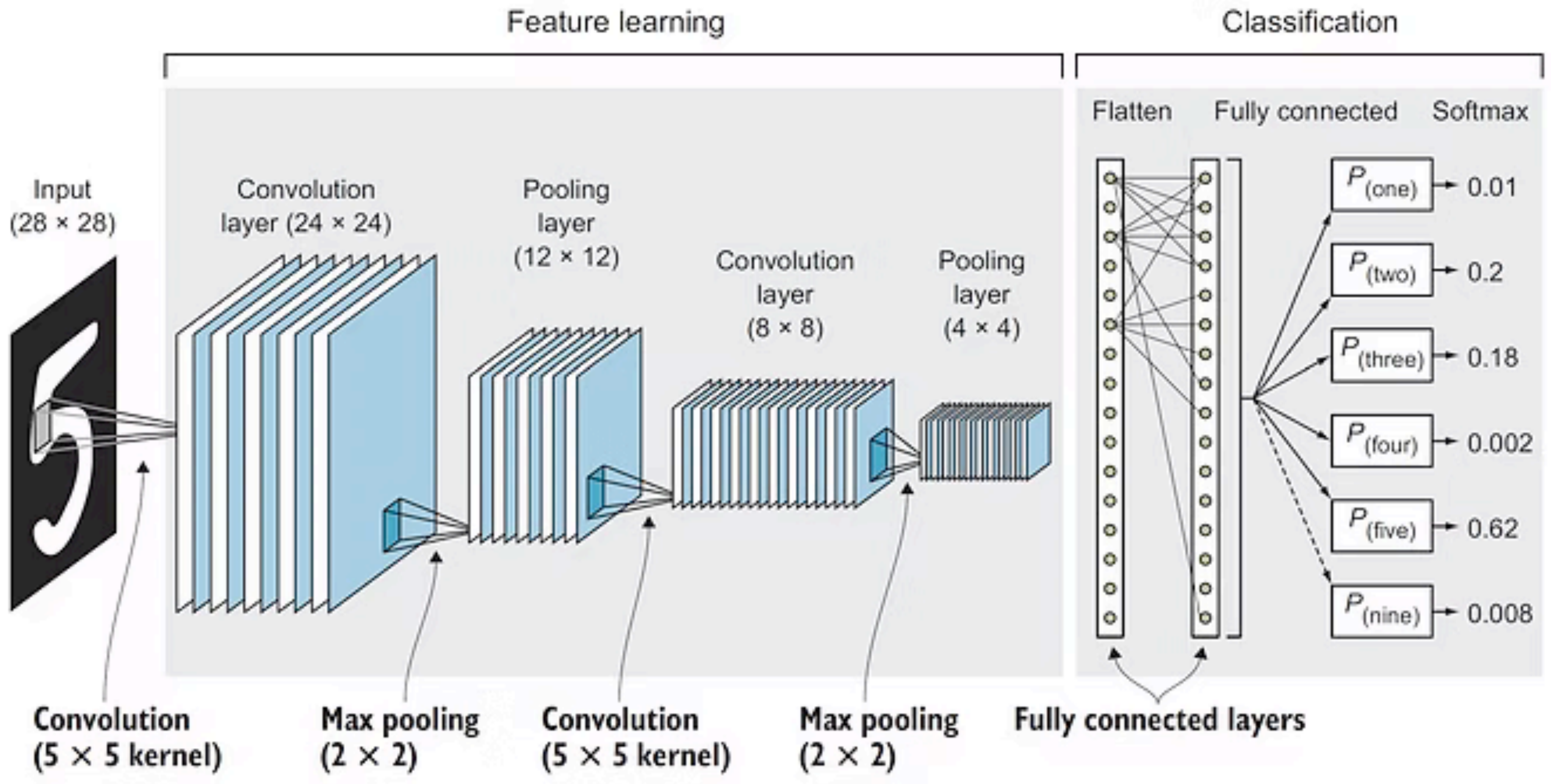


Image: Developing an effective model for the semantic segmentation of remote sensing imagery - Aminu Aliya, Boukari Souley, Abdullahi Madaki Gamsha - Dec 2021

Downsampling

- Strided Convolutions
- Pooling (Max, Average, Global)



[Image: Convolutional neural net part 3: Downsampling - https://www.nibivid.com/post/convolutional-neural-net-part-3-downsampling](https://www.nibivid.com/post/convolutional-neural-net-part-3-downsampling)

Max Pooling

1	2	3	4
5	6	7	8
9	0	1	2
3	4	5	6

→

6	8
9	6

Upsampling (Interpolation)

- Bi-linear interpolation (Bi-cubic)
- Bilateral back projection
- Deconvolution
- Others

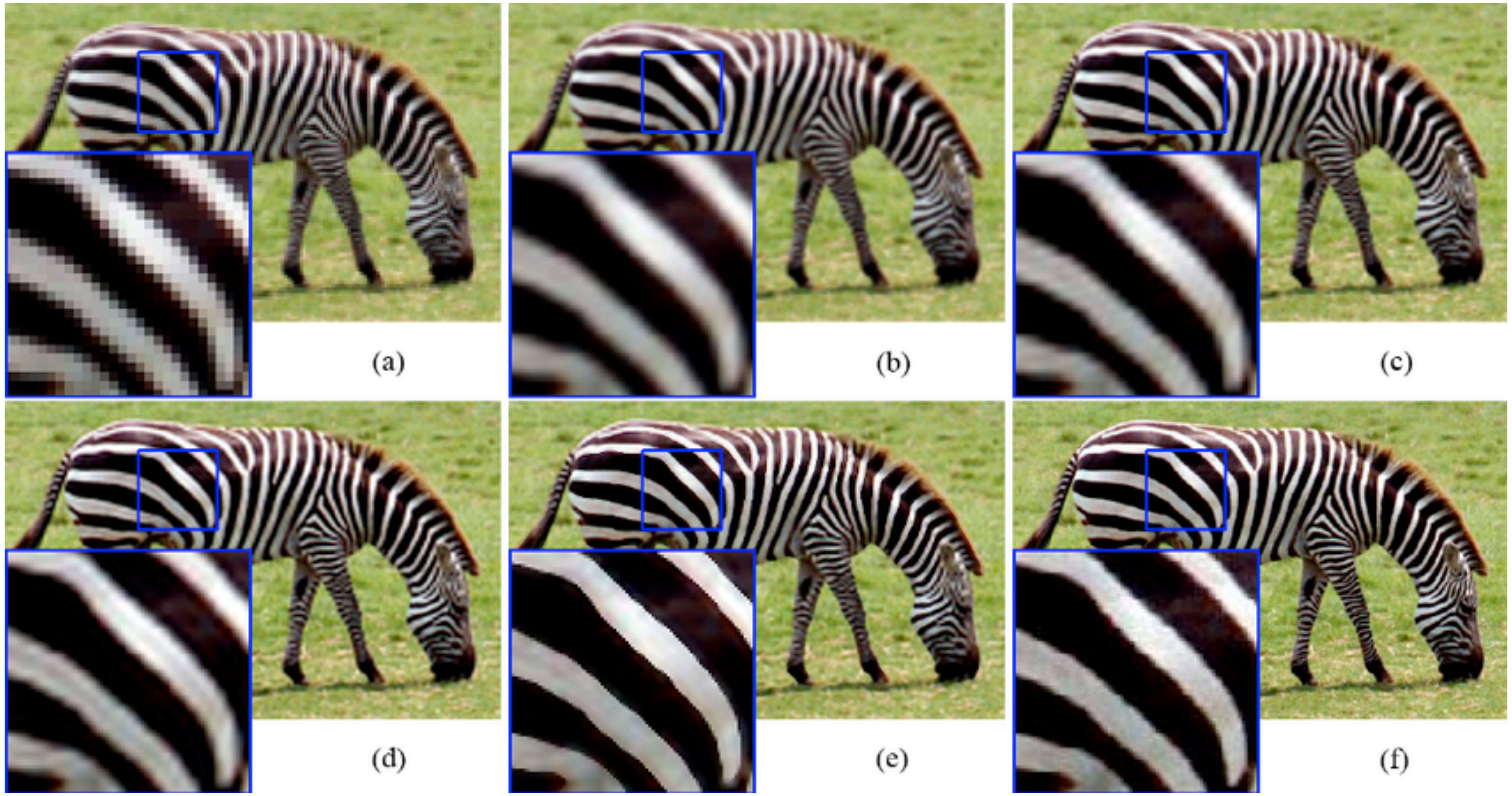


Image: Bilateral Back-Projection for Single Image Super Resolution - Mei Han, Ying Wu, Yihong Gong - Aug 2007

Upsampling - Bi-cubic interpolation

- A localized cubic interpolation
- uses the function value f, and the derivatives f_x, f_y, f_{xy} to find the coefficients of the fit.
- We'll use nn.Upsample(size, scale_factor, mode='bicubic')
  - Expects the input to be of form (batch x channels x [depth] x [height] x width)

Image: Bilateral Back-Projection for Single Image Super Resolution - Mei Han, Ying Wu, Yihong Gong - Aug 2007

Deconvolution

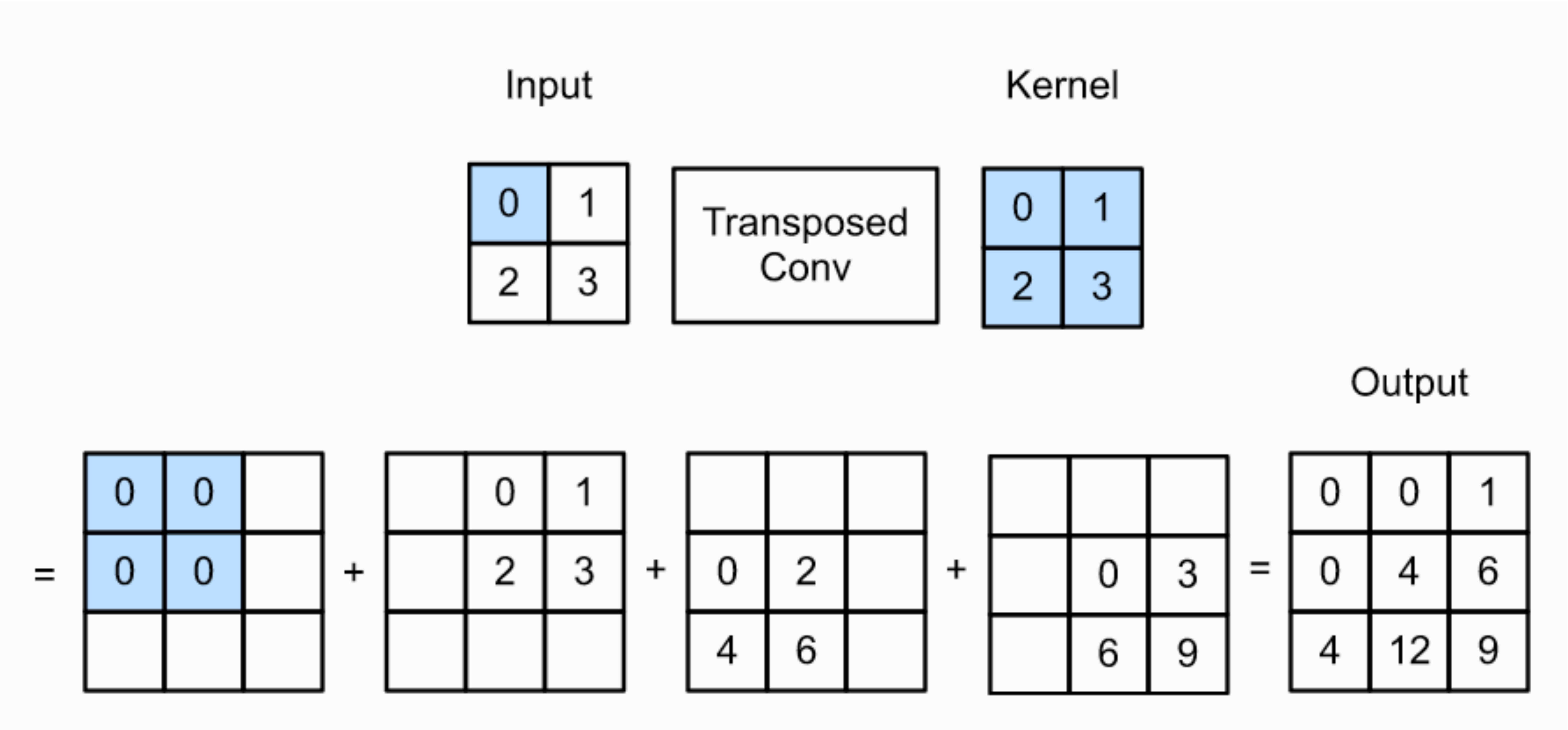


Image: Transposed Convolution vs Convolution Layer: Examples - Ajitesh Kumar - https://vitalflux.com/transposed-convolution-vs-convolution-layer-examples/#What_are_Convolutional_Transpose_Layers_Whats_their_purpose

Deconvolution

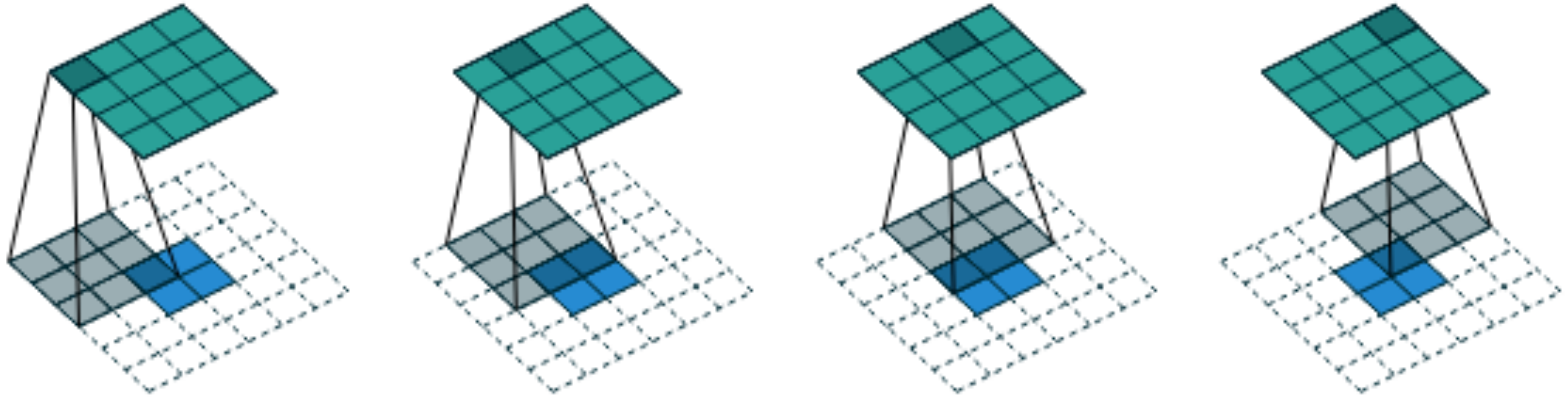


Figure 4.1: The transpose of convolving a 3×3 kernel over a 4×4 input using unit strides (i.e., $i = 4$, $k = 3$, $s = 1$ and $p = 0$). It is equivalent to convolving a 3×3 kernel over a 2×2 input padded with a 2×2 border of zeros using unit strides (i.e., $i' = 2$, $k' = k$, $s' = 1$ and $p' = 2$).

- Blue = Inputs
- Cyan = Outputs

Image: A guide to convolution arithmetic for deep learning - Vincent Dumoulin, Francesco Visin - January 2018

Deconvolution

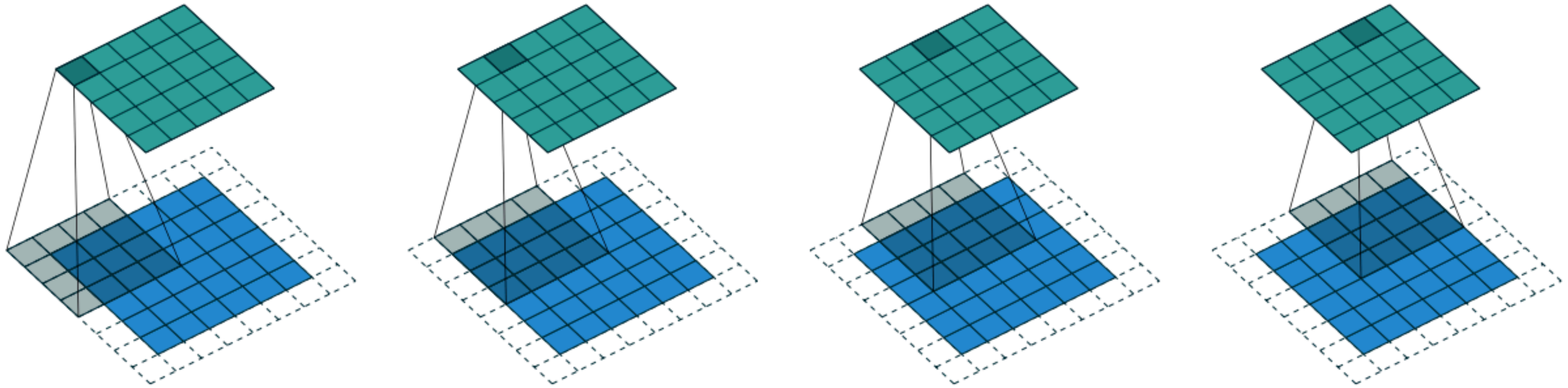


Figure 4.2: The transpose of convolving a 4×4 kernel over a 5×5 input padded with a 2×2 border of zeros using unit strides (i.e., $i = 5$, $k = 4$, $s = 1$ and $p = 2$). It is equivalent to convolving a 4×4 kernel over a 6×6 input padded with a 1×1 border of zeros using unit strides (i.e., $i' = 6$, $k' = k$, $s' = 1$ and $p' = 1$).

- Blue = Inputs
- Cyan = Outputs

Image: A guide to convolution arithmetic for deep learning - Vincent Dumoulin, Francesco Visin - January 2018

Deconvolution - Dilation

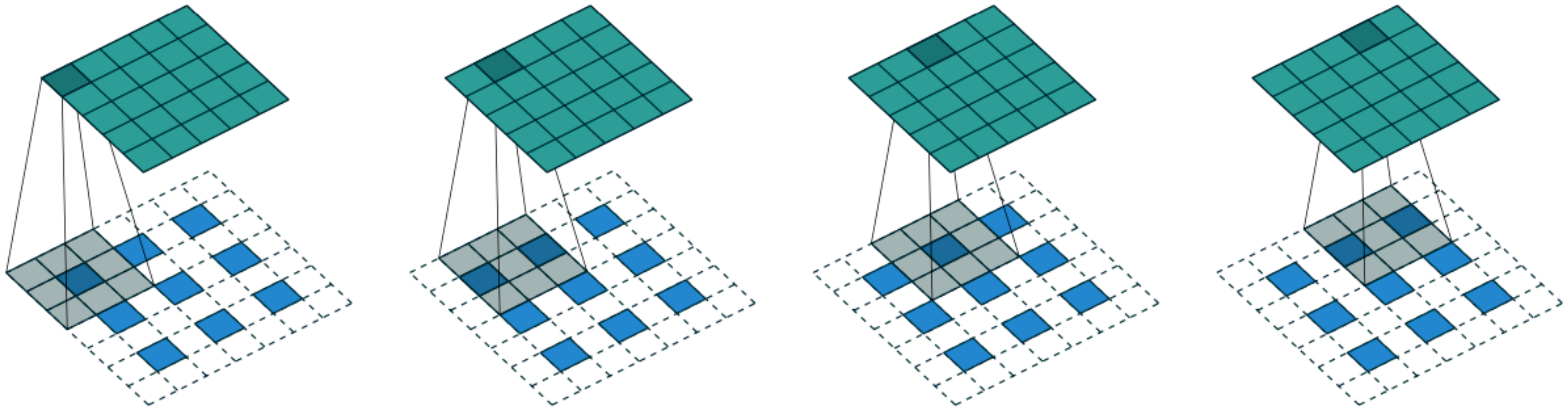


Figure 4.6: The transpose of convolving a 3×3 kernel over a 5×5 input padded with a 1×1 border of zeros using 2×2 strides (i.e., $i = 5$, $k = 3$, $s = 2$ and $p = 1$). It is equivalent to convolving a 3×3 kernel over a 3×3 input (with 1 zero inserted between inputs) padded with a 1×1 border of zeros using unit strides (i.e., $i' = 3$, $\tilde{i}' = 5$, $k' = k$, $s' = 1$ and $p' = 1$).

- Blue = Inputs
- Cyan = Outputs

Image: A guide to convolution arithmetic for deep learning - Vincent Dumoulin, Francesco Visin - January 2018

Typical Applications

- Image/video recognition
- Super-resolution tasks (denoising and interpolating)
- Recommender systems
- Natural Language processing

Common Architectures - VGG-16

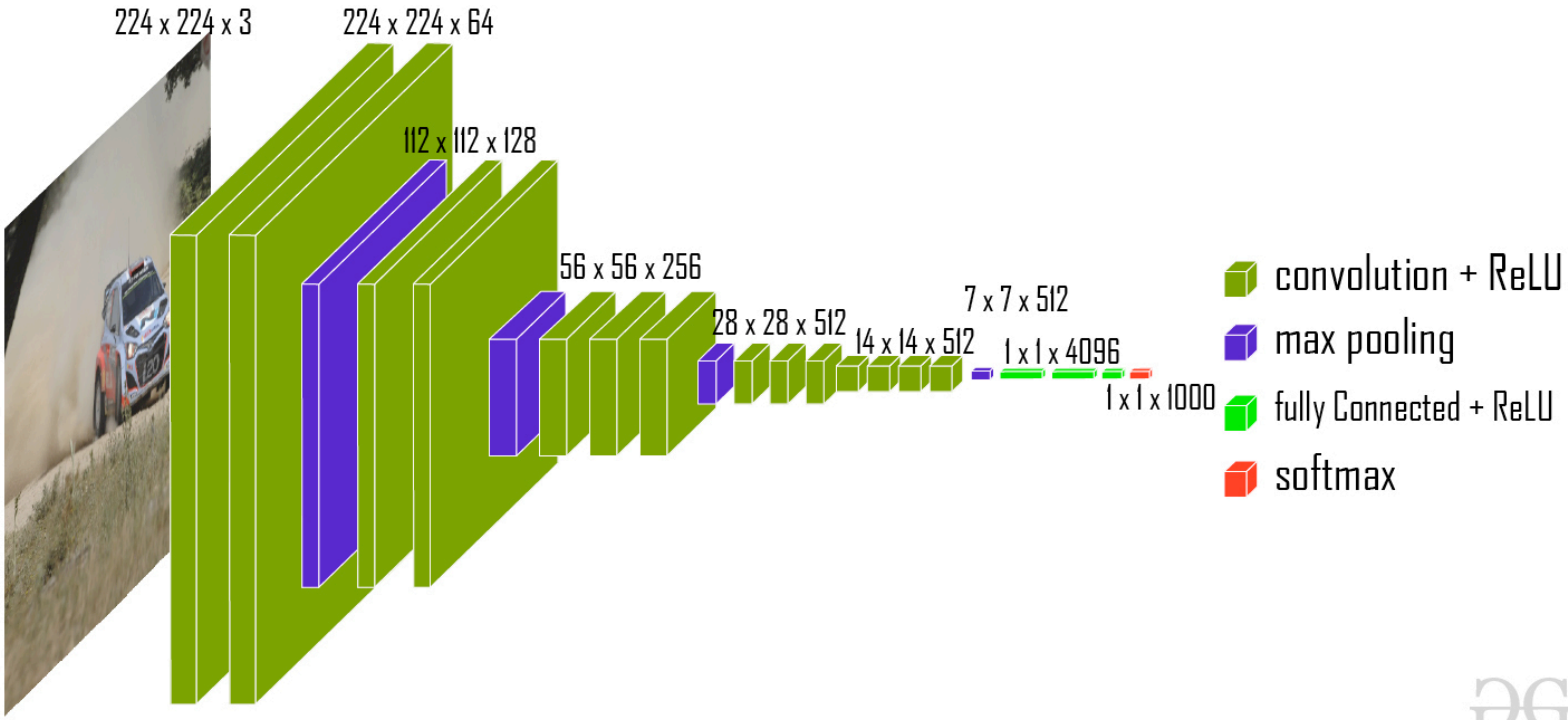


Image: VGG - 16 | CNN Model - https://www.geeksforgeeks.org/vgg-16-cnn-model/

Common Architectures - U-Net

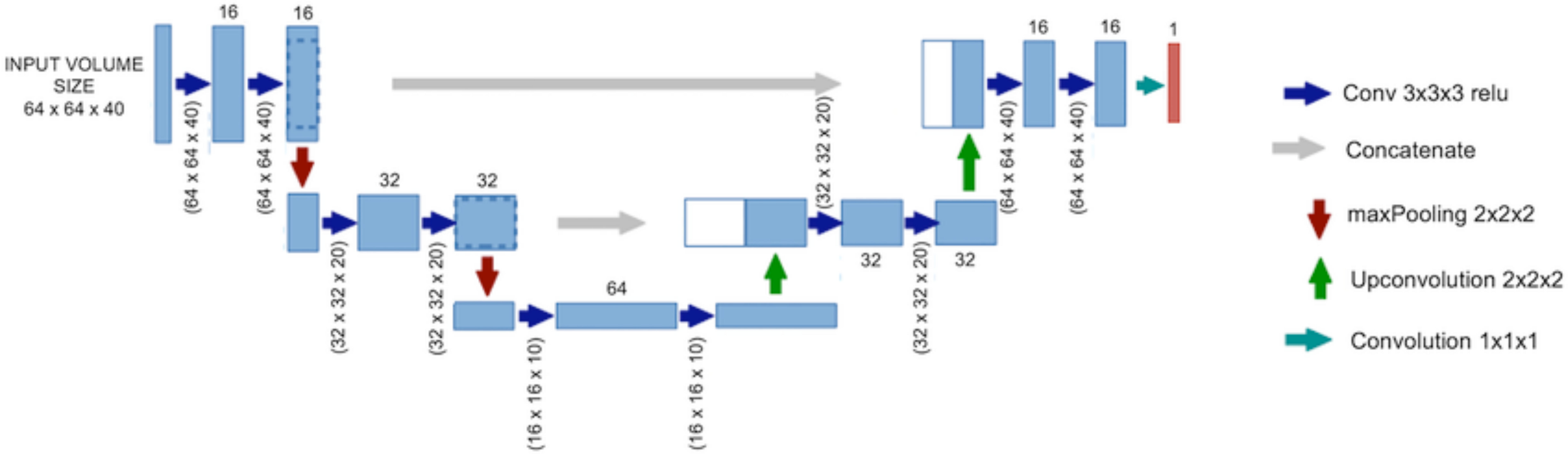


Image: Automatic lesion detection and segmentation of F-FET PET in gliomas: A full 3D U-Net convolutional neural network study - Paul Blanc-Durand, Axel Van Der Gucht, Niklaus Schaefer ...

Engineering Modeling Applications

- Super-resolution
-

Example from literature

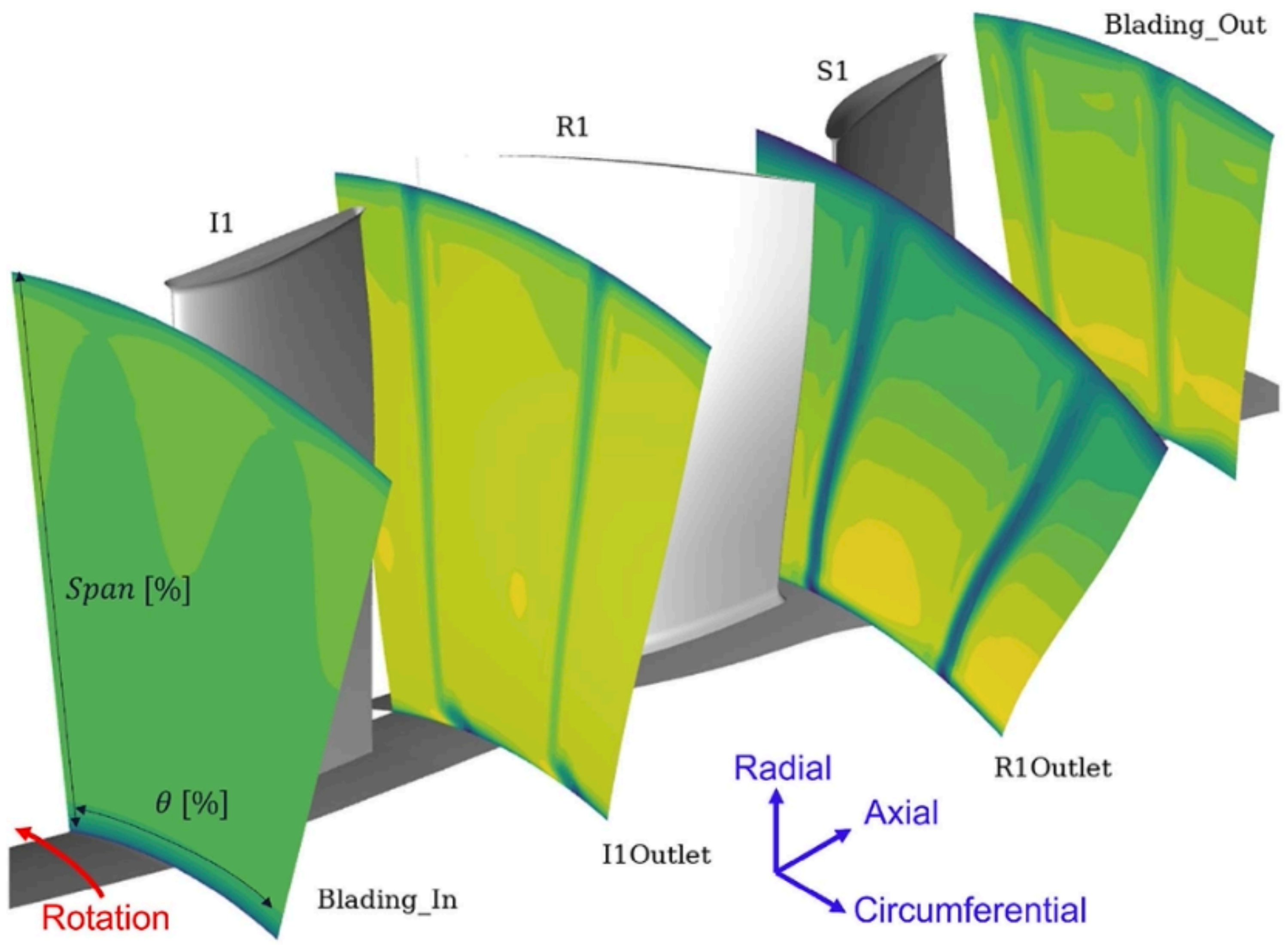


Fig. 2. Overview of the CFD domain and locations selected for post-processing, with corresponding axial velocity contours.

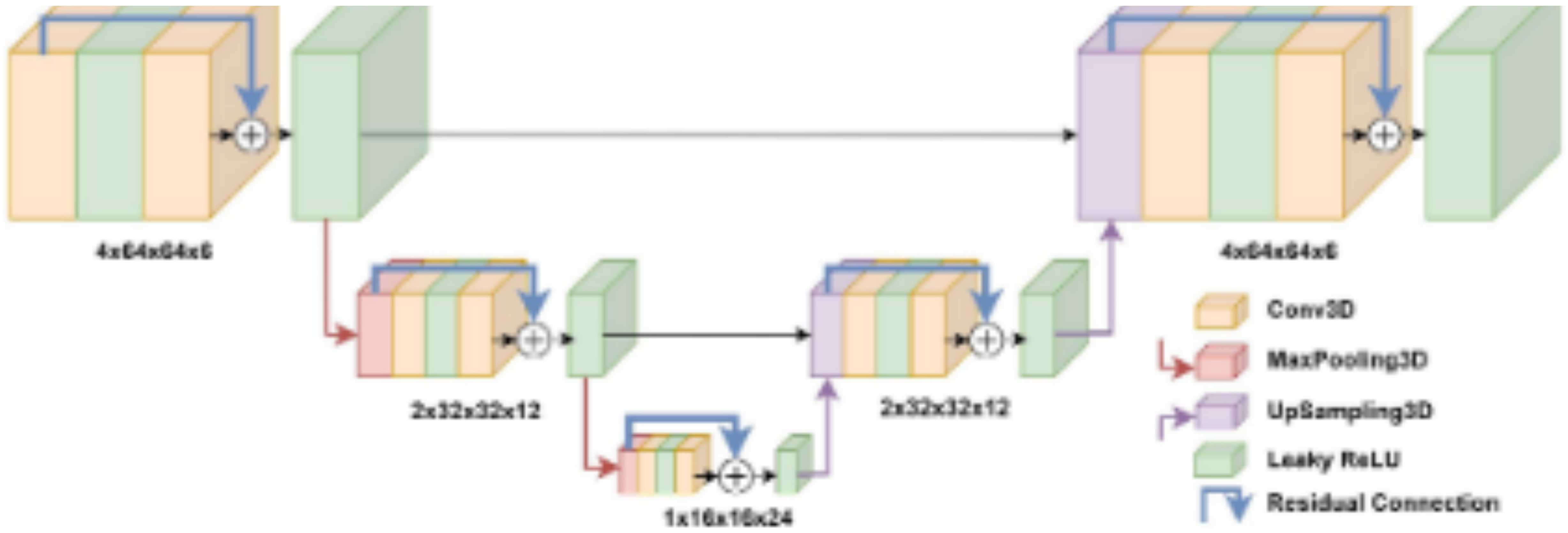


Fig. 9. C(NN)FD architecture overview.

[Images: C(NN)FD - A Deep Learning Framework for Turbomachinery CFD Analysis]

PyTorch Implementation

Physics of Fluids ARTICLE scitation.org/journal/phf

Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels

Cite as: Phys. Fluids **33**, 073603 (2021); doi: 10.1063/5.0054312
Submitted: 16 April 2021 · Accepted: 1 June 2021 ·
Published Online: 7 July 2021

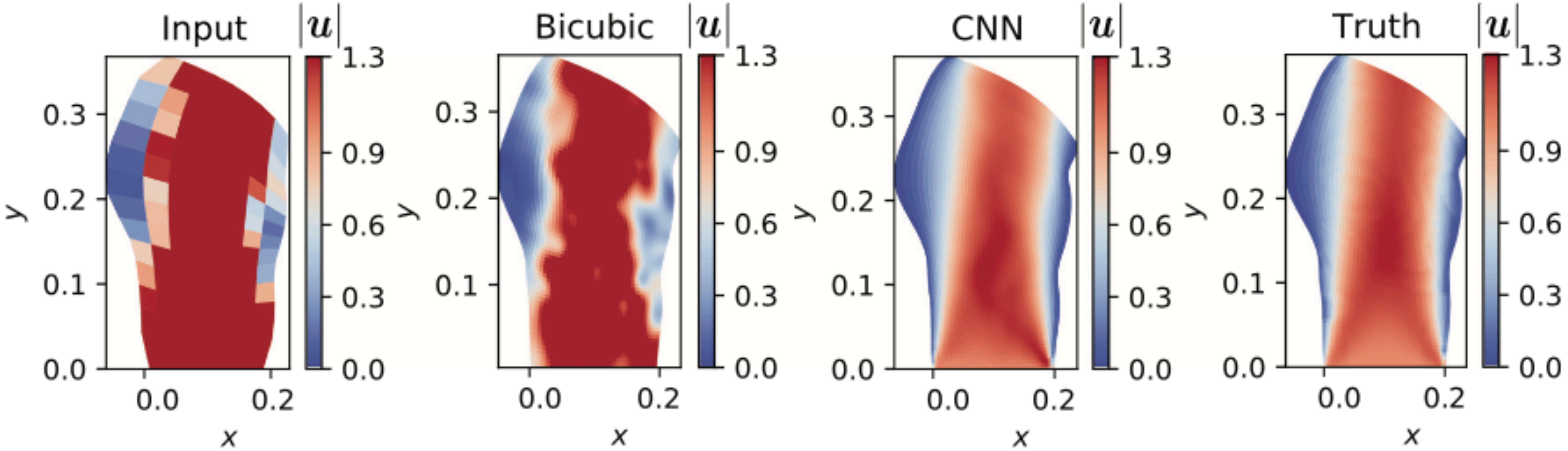


FIG. 3. The super-resolved results of the LR input with the 100% Gaussian noise ($c = 1.0$). The relative errors of the bicubic-SR and CNN-SR fields are 0.520 and 0.067, respectively.

Image: Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels

Loss Function

$$\mathscr{R}(u,p) = 0 = \begin{cases} \nabla \cdot u \\ (u \cdot \nabla)u + \frac{1}{\rho}\nabla p - \nu \nabla^2 u \end{cases}$$

- u is the velocity
- p is the pressure
- ν is the viscosity

Coordinate Transformation

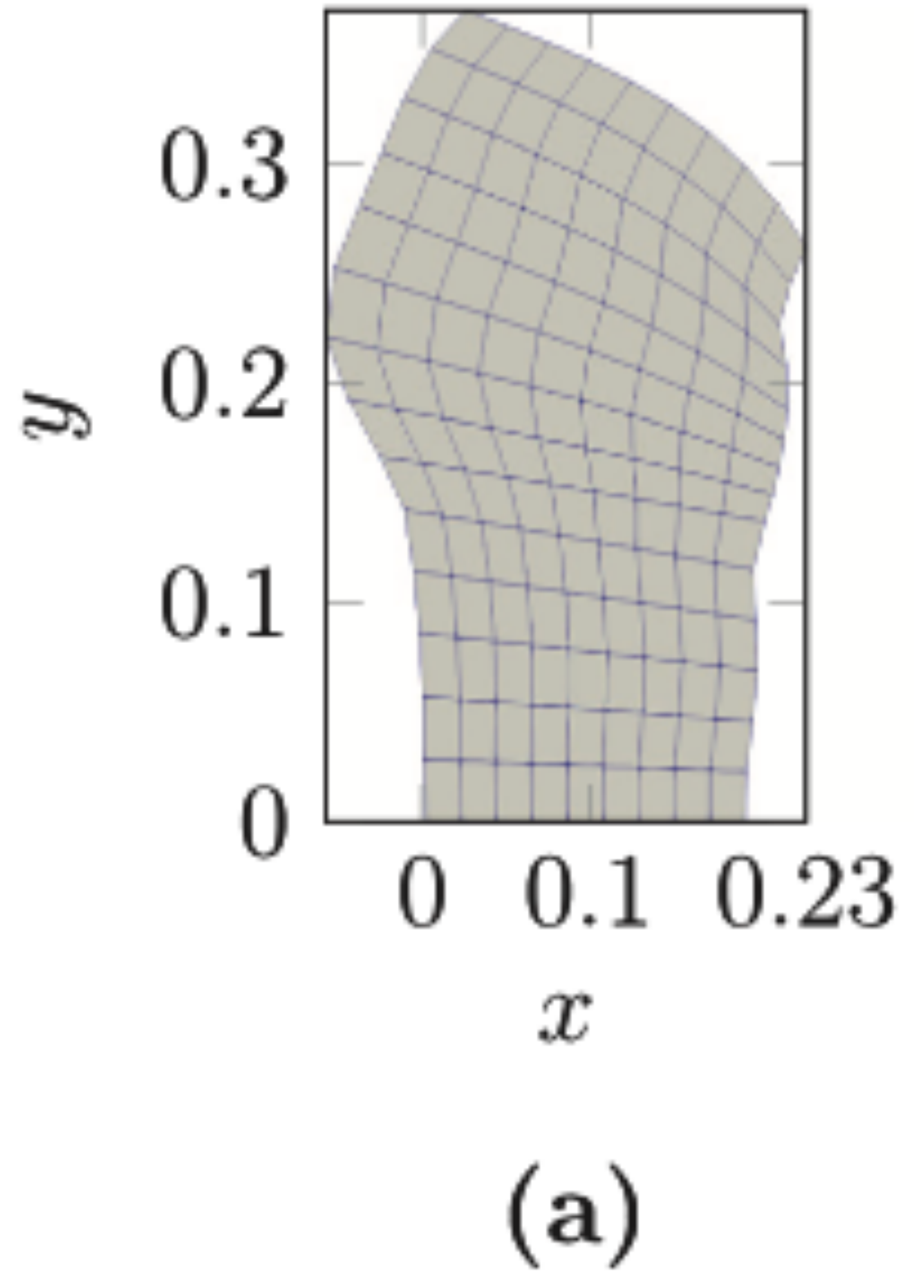


FIG. 2. (a) Coarse mesh and (b) fine mesh [the low-resolution mesh (126 cells and high-resolution mesh (3773 cells)]. The LR input data are refined by 30×.

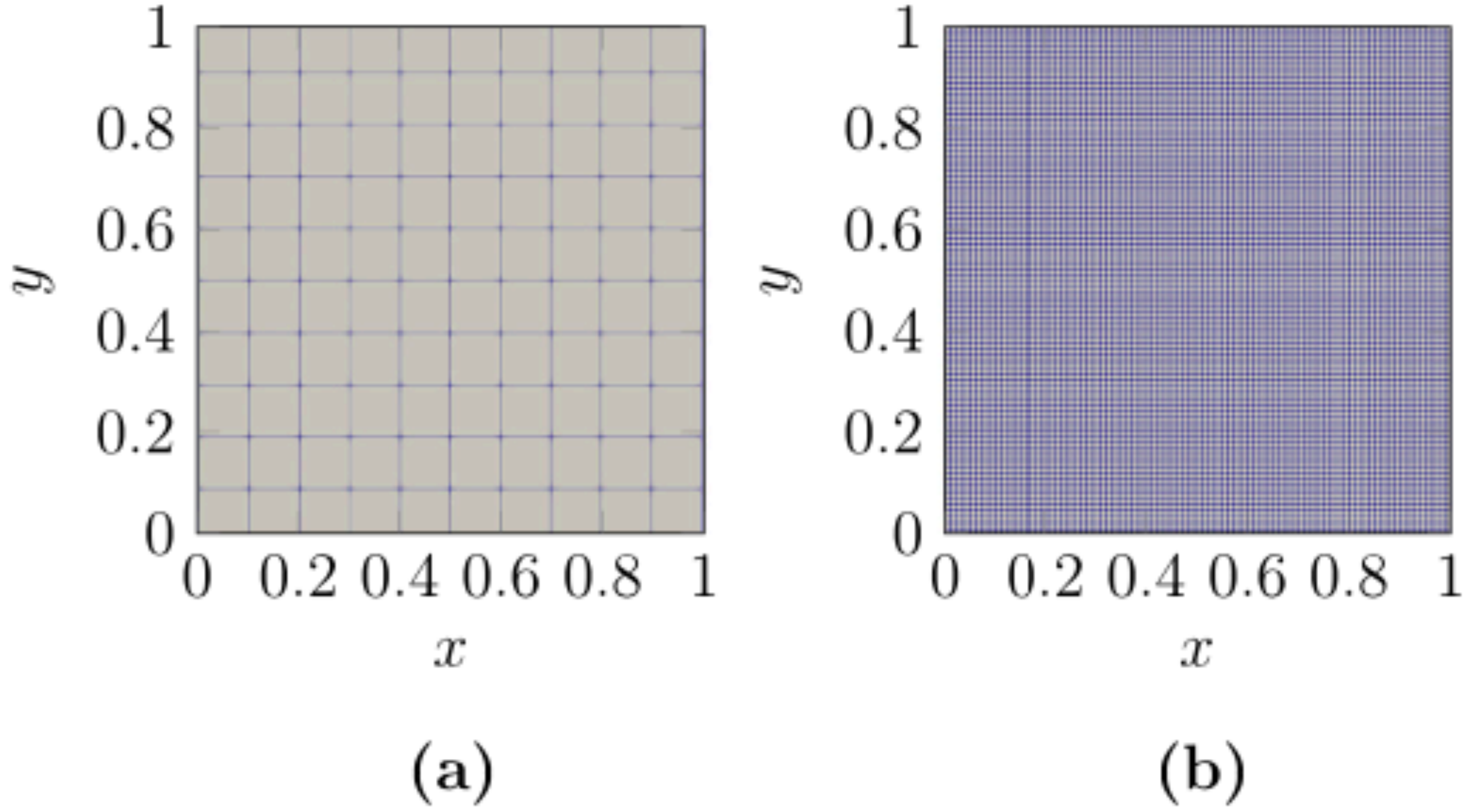


FIG. 9. (a) Coarse mesh and (b) fine mesh [the low-resolution input mesh (10×10) and the high-resolution output mesh (200×200)]. The LR data will be refined by 400×.

Coordinate Transformation

$$\frac{\partial}{\partial x} = \frac{1}{J}\left[\left(\frac{\partial}{\partial \xi}\right)\left(\frac{\partial y}{\partial \eta}\right) - \left(\frac{\partial}{\partial \eta}\right)\left(\frac{\partial y}{\partial \xi}\right)\right], \quad (10a)$$

$$\frac{\partial}{\partial y} = \frac{1}{J}\left[\left(\frac{\partial}{\partial \eta}\right)\left(\frac{\partial x}{\partial \xi}\right) - \left(\frac{\partial}{\partial \xi}\right)\left(\frac{\partial x}{\partial \eta}\right)\right], \quad (10b)$$

How you convert derivatives