14. Convolutional Models

EEC E454 Introduction to Machine Learning Systems

2023 Fall, Jaeho Lee

Recap: MLPs

- **Multi-Layer Perceptrons** take the simple form:

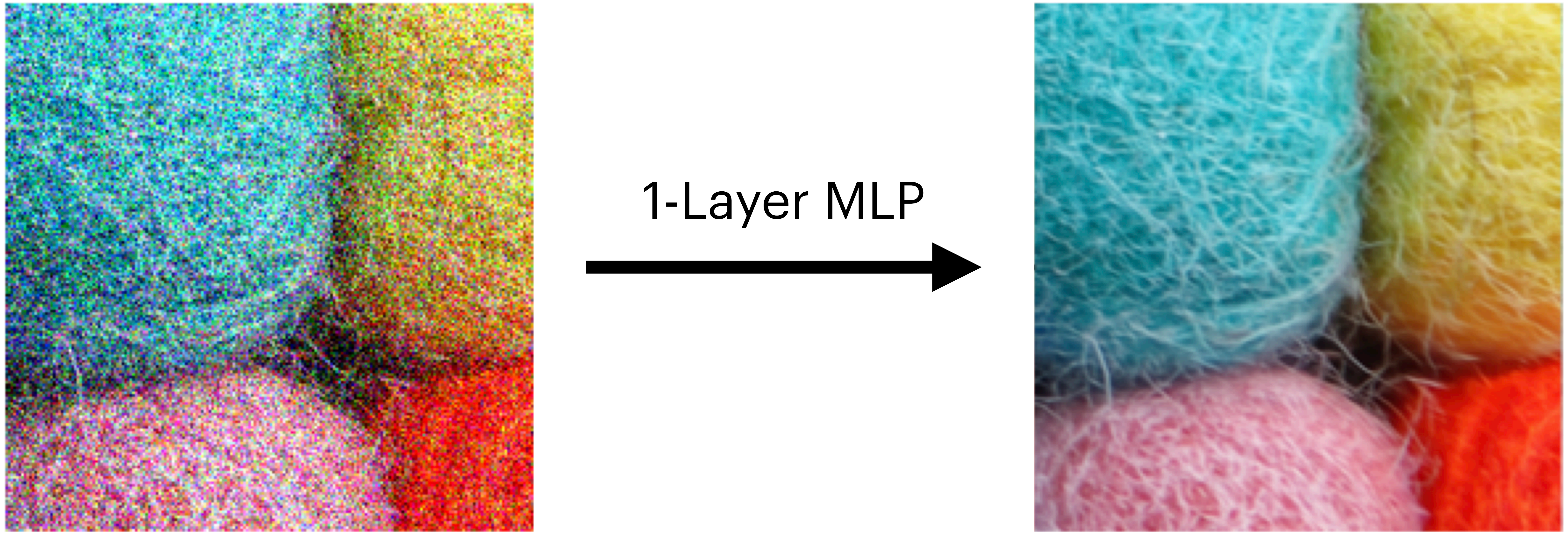
$$f(\mathbf{x}) = \mathbf{W}_L\sigma(\mathbf{W}_{L-1}\sigma(\cdots\sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)\cdots + \mathbf{b}_{L-1}) + \mathbf{b}_L$$

  - Alternatingly applies two operations:
    - A linear operation $\mathbf{x} \mapsto \mathbf{W}\mathbf{x} + \mathbf{b}$
    - A nonlinear operation (activation) $\mathbf{x} \mapsto \sigma(\mathbf{x})$

Quick Question

- Suppose that we train an image processing model (e.g., denoiser), that processes a **1080p image** into another **1080p image**.
  (1920×1080 pixels)

If we use a linear model (i.e., MLP with one layer), how many parameters do we need?



Quick Question

- Suppose that we train an image processing model (e.g., denoiser), that processes a **1080p image** into another **1080p image**.
  (1920×1080 pixels)

  If we use a linear model (i.e., MLP with one layer), how many parameters do we need?

Answer. 3.87×10^{13} weights

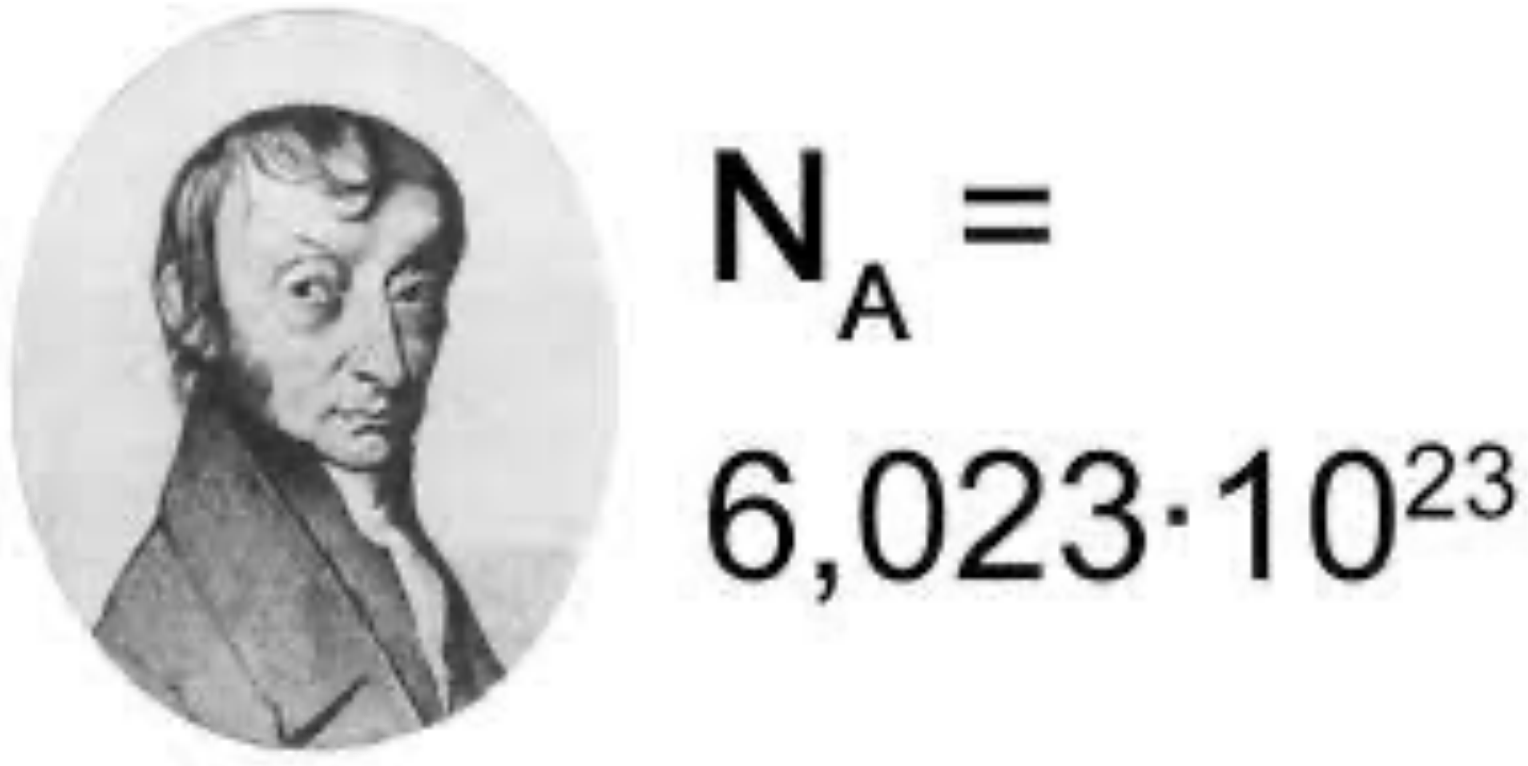
($\approx$ 1.55TB)

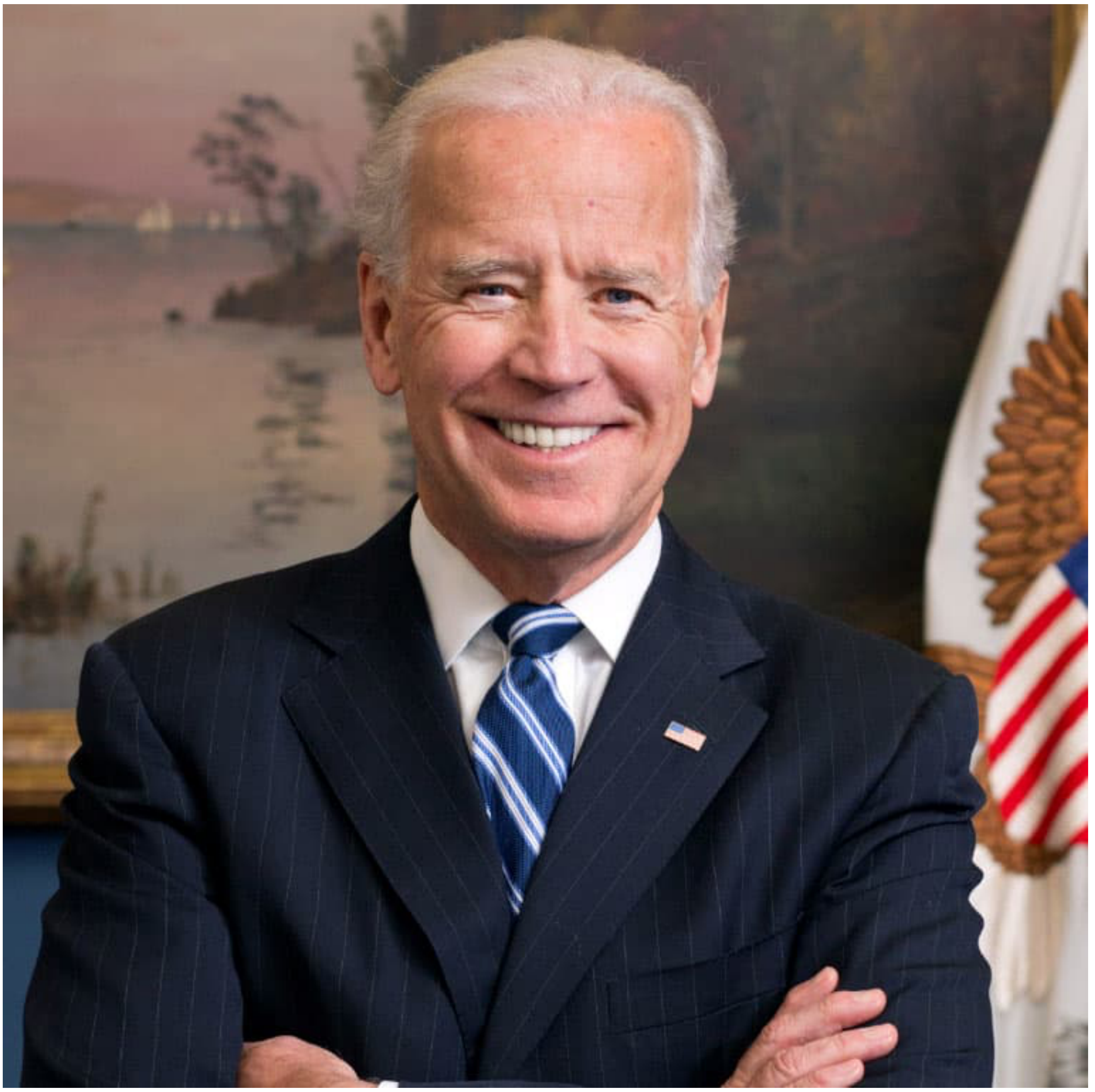
Quick Question

- **How about Compute?**

  This amounts to 7.74×10^{13} FLOPs for every image.

- Suppose that...
  - we have 10 layers,
  - train on one million images,
  - for 100 "epochs." (we'll learn later)
- Then we need $\approx 1.5 \times 10^{23}$ FLOPs.





OCTOBER 30, 2023

Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence

(b) The Secretary of Commerce, in consultation with the Secretary of State, the Secretary of Defense, the Secretary of Energy, and the Director of National Intelligence, shall define, and thereafter update as needed on a regular basis, the set of technical conditions for models and computing clusters that would be subject to the reporting requirements of subsection 4.2(a) of this section. Until such technical conditions are defined, the Secretary shall require compliance with these reporting requirements for:

(i) any model that was trained using a quantity of computing power greater than 10^{26} integer or floating-point operations, or using primarily biological sequence data and using a quantity of computing power greater than 10^{23} integer or floating-point operations; and

How do we get out of this crisis?

- Many ideas, such as…
  - Reduced precision
  - Adding zeros to weights
  - …
- By far the most powerful and clever, classic trick: **Weight Sharing**
  - The most important example: **Convolution**

Convolution—an overview

What is convolution?

- If you took *"signals & systems"*, you should be familiar:
- **Definition.** A convolution of two functions is:

$$(f * g)(t) = \int_{-\infty}^{+\infty} f(\tau) \cdot g(t - \tau) \, \mathrm{d}\tau$$

- The response of a system that has impulse response $f(t)$, when given the input signal $g(t)$.

Unit Pulse → System → $f(t)$

$g(t)$ → System → $(f * g)(t)$

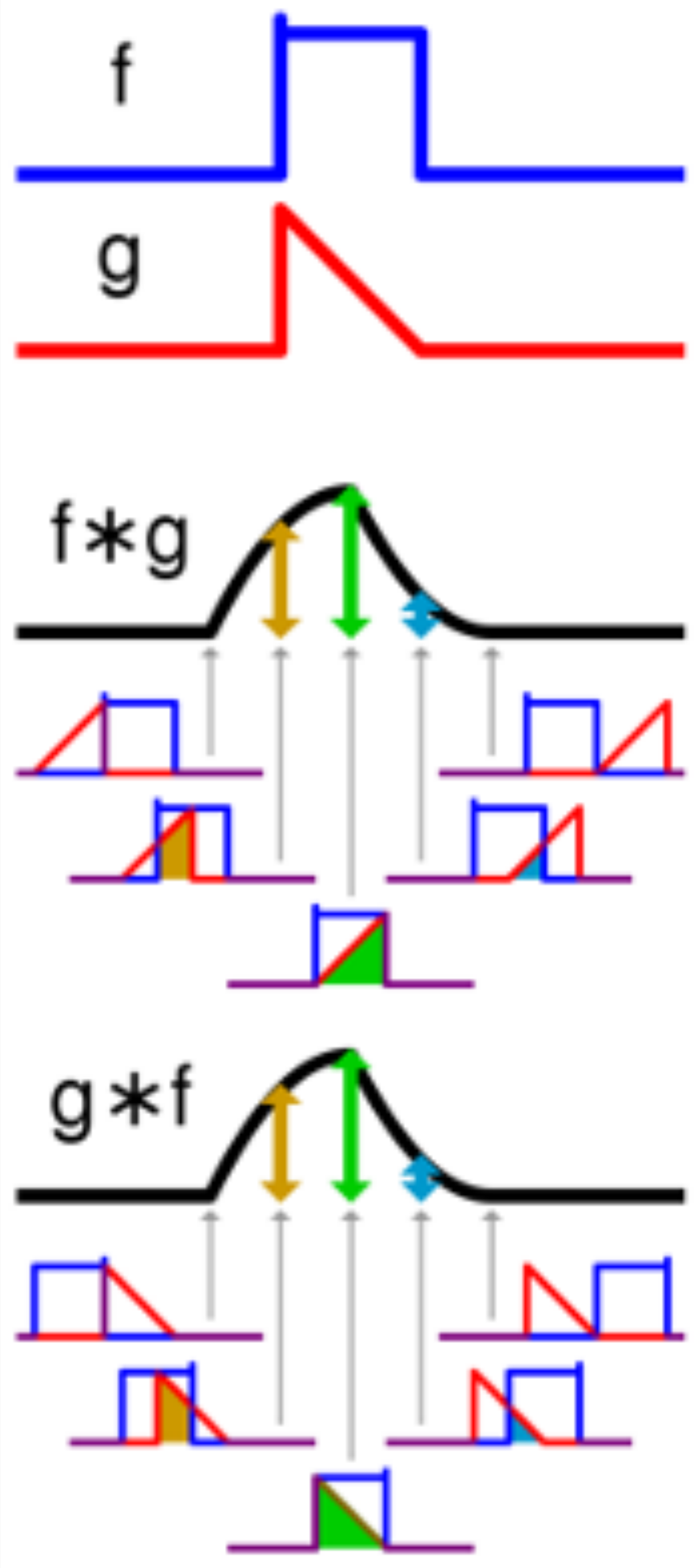


Image source: 1D convolution for neural networks, part 1: Sliding dot product, Brandon Rohrer

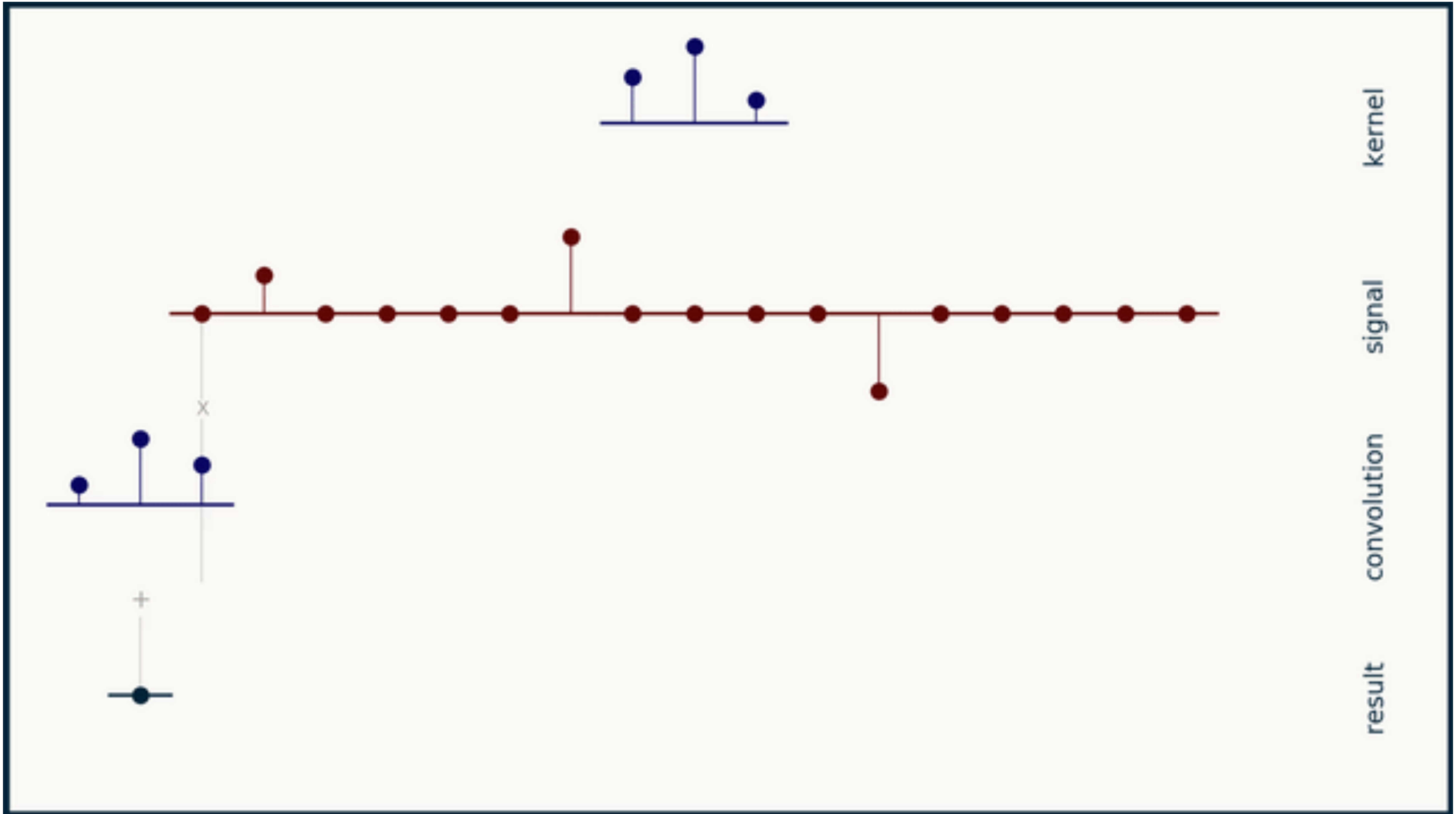


Image source: wikipedia “convolution”

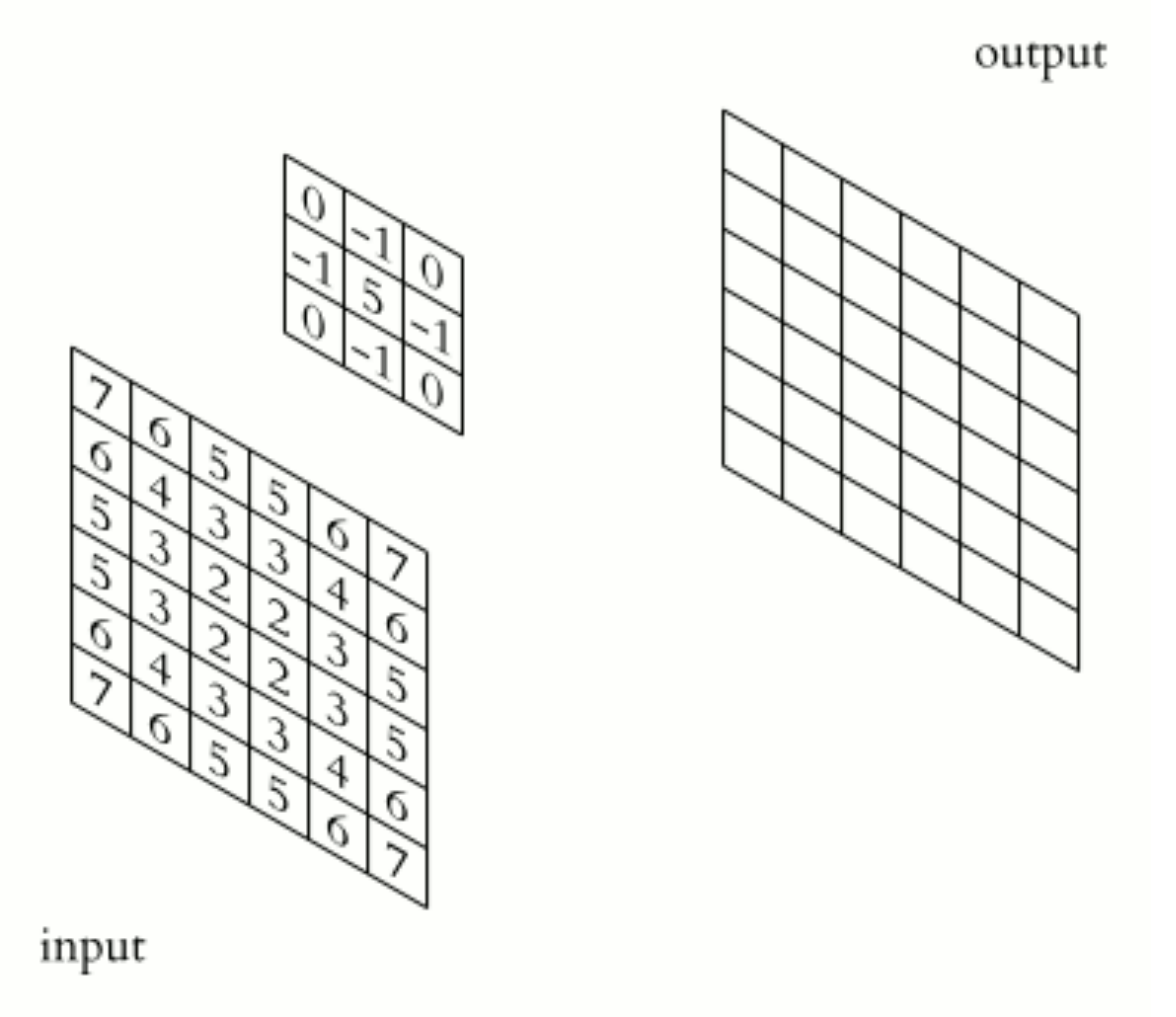
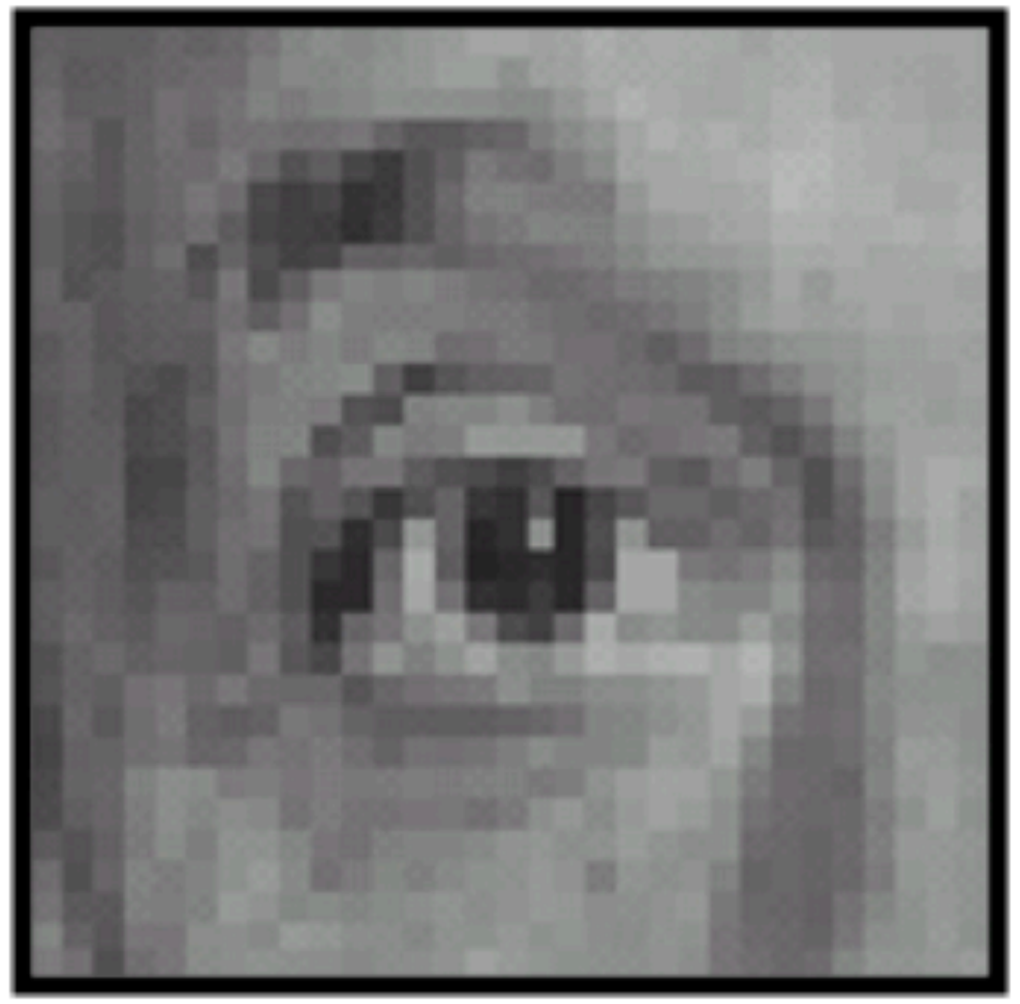


Image source: https://www.cs.cornell.edu/courses/cs1114/2013sp/sections/S06_convolution.pdf

What is convolution?

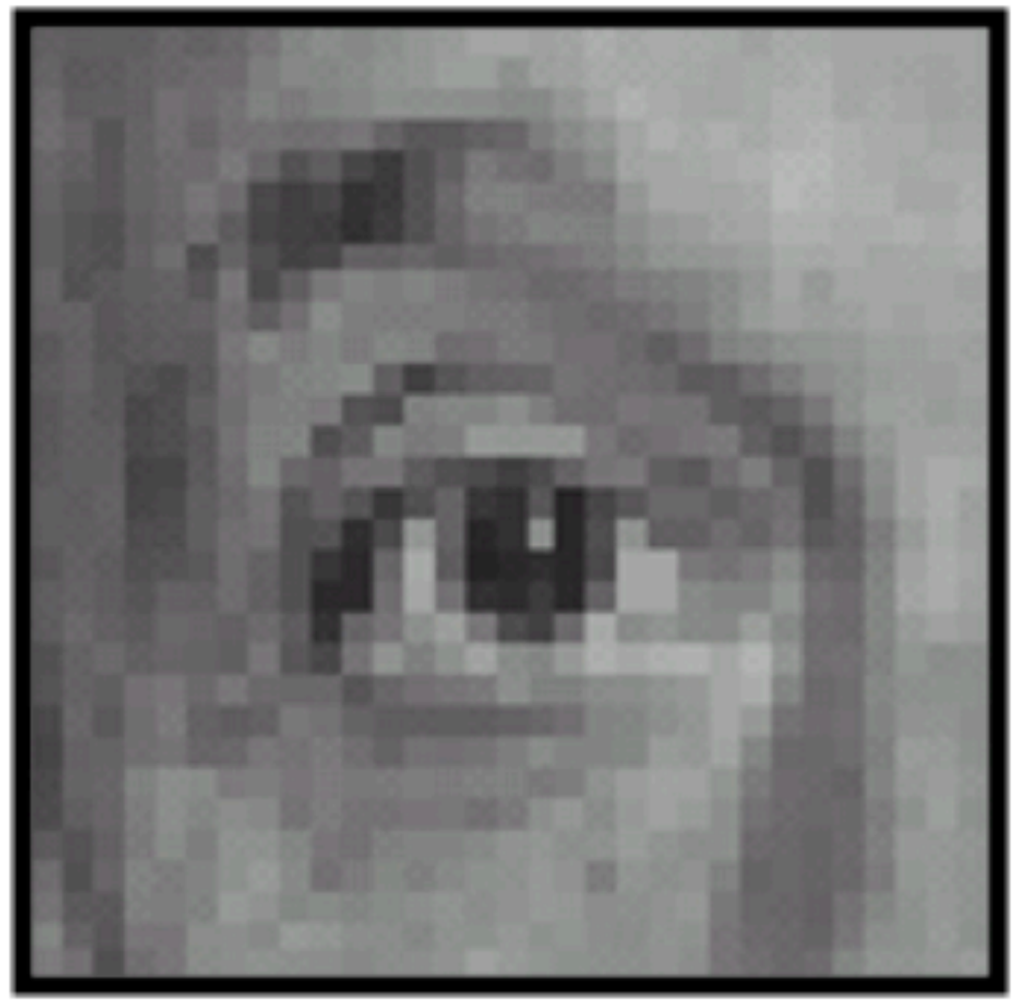
- Different "filters" can be used for different purposes.

Systems with some impulse response



Original

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

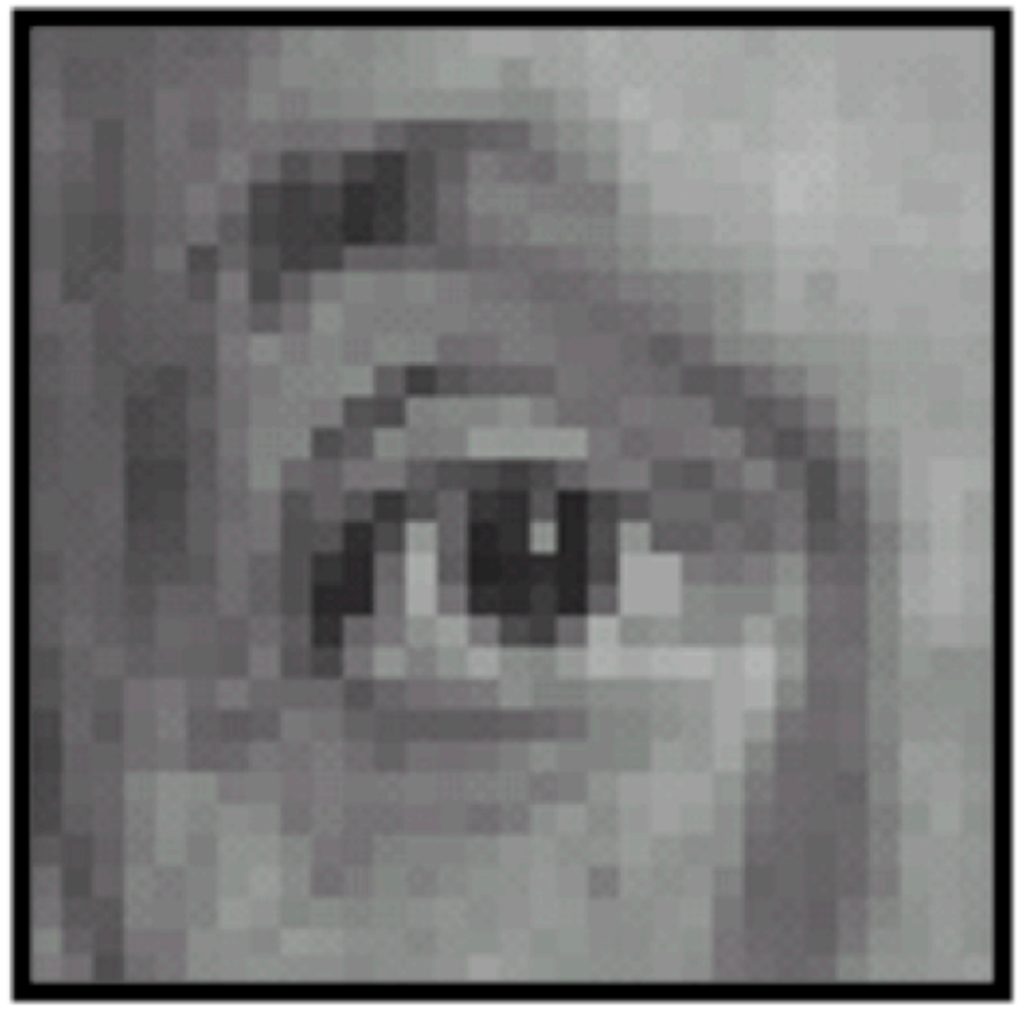


Identical image

Image source: https://www.cs.cornell.edu/courses/cs1114/2013sp/sections/S06_convolution.pdf

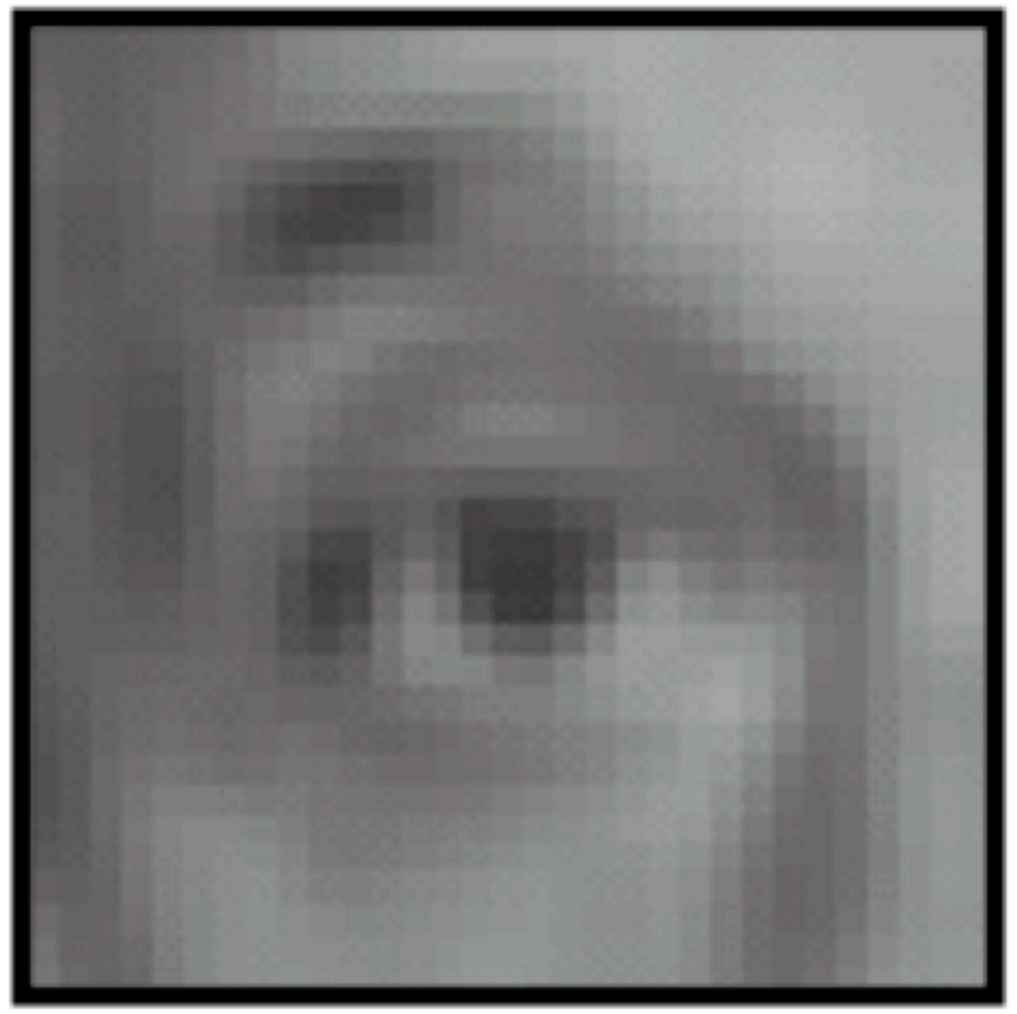
What is convolution?

- Different "filters" can be used for different purposes.



Original

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

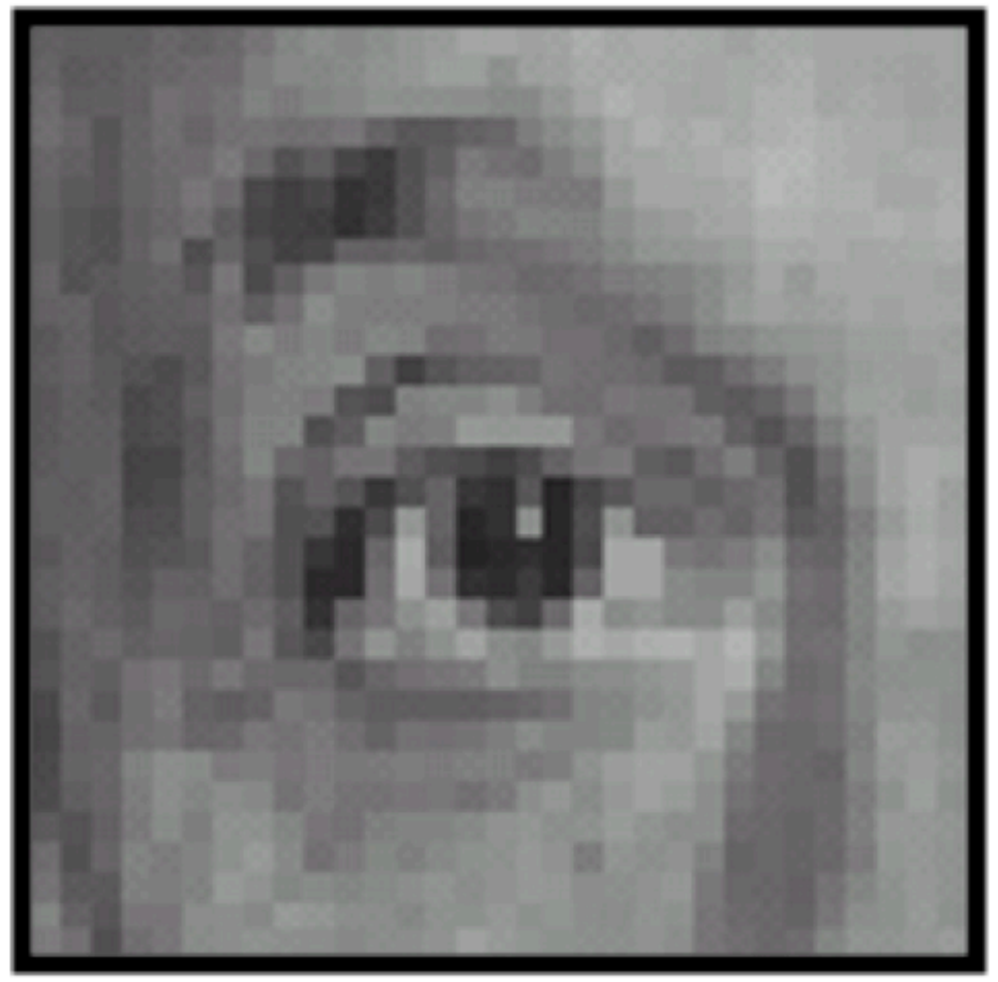


Blur (with a mean filter)

Image source: https://www.cs.cornell.edu/courses/cs1114/2013sp/sections/S06_convolution.pdf

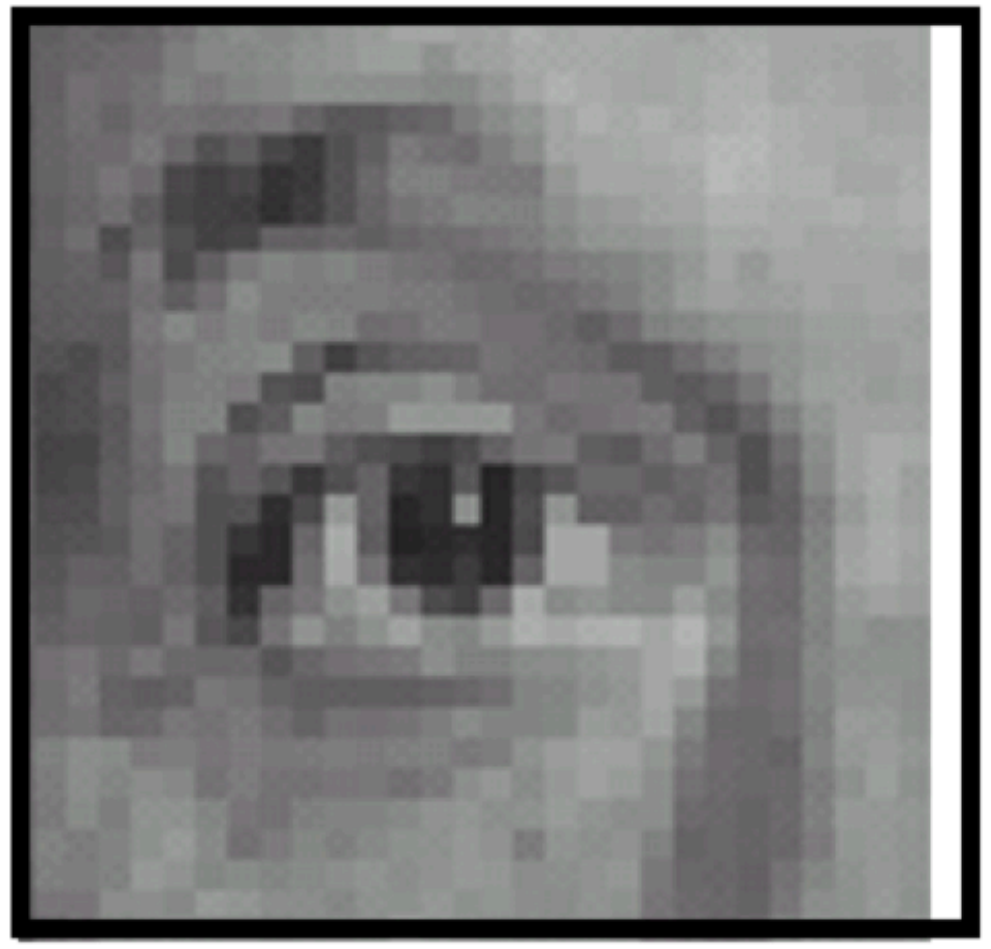
What is convolution?

- Different “filters” can be used for different purposes.



Original

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

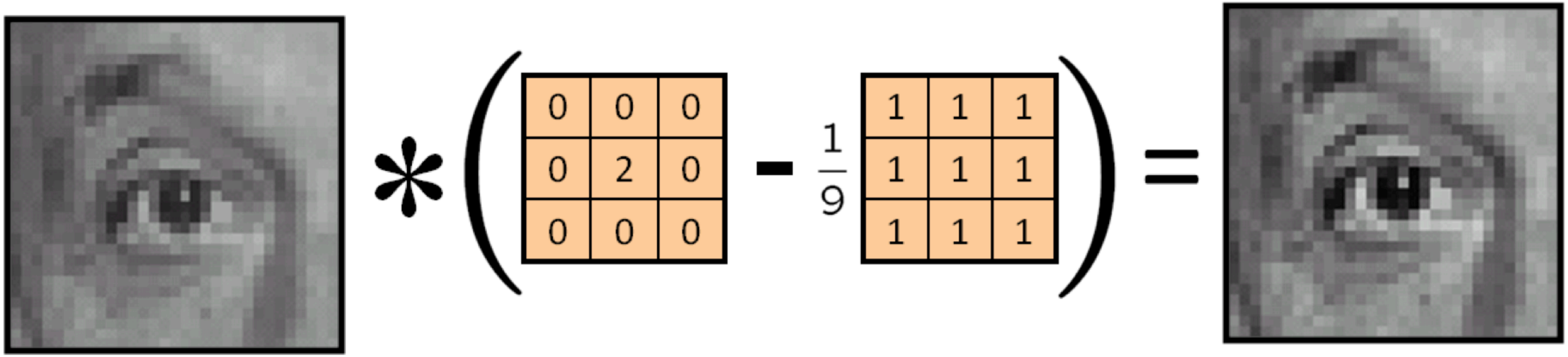


Shifted left
By 1 pixel

Image source: https://www.cs.cornell.edu/courses/cs1114/2013sp/sections/S06_convolution.pdf

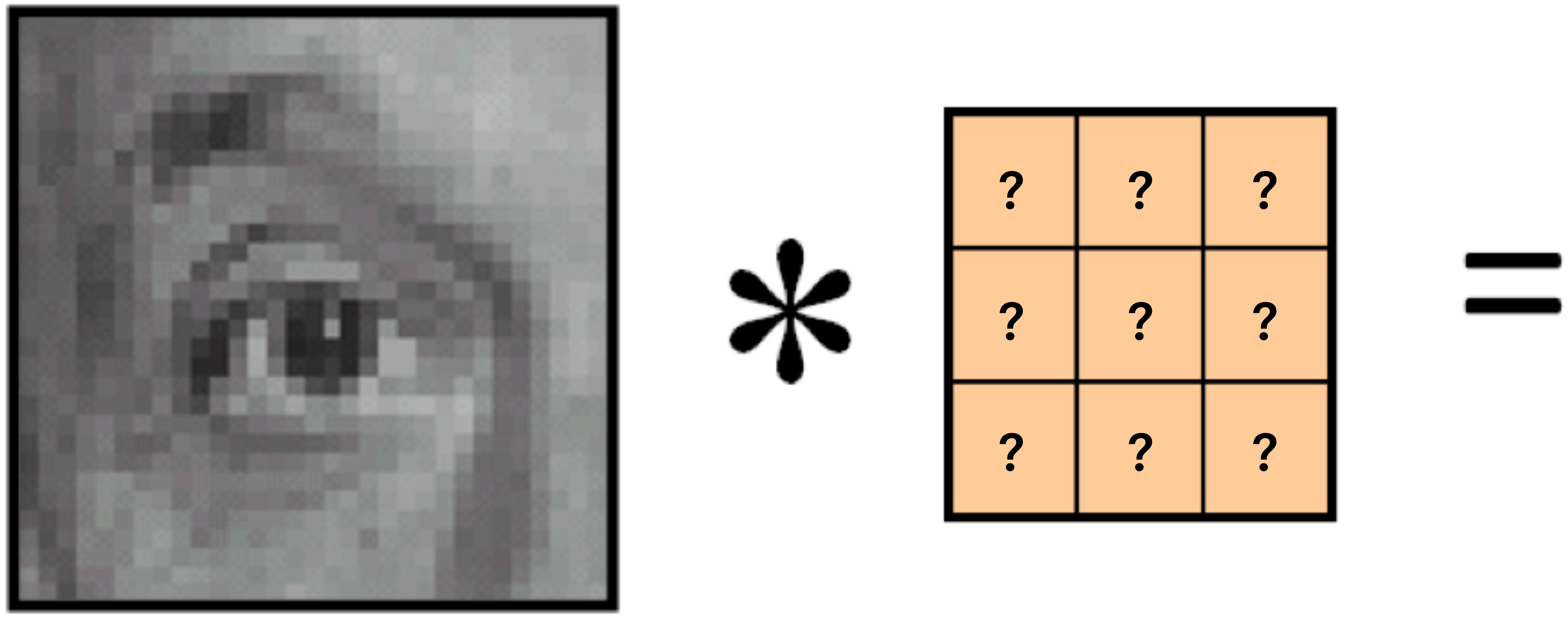
What is convolution?

- Different “filters” can be used for different purposes.



Original

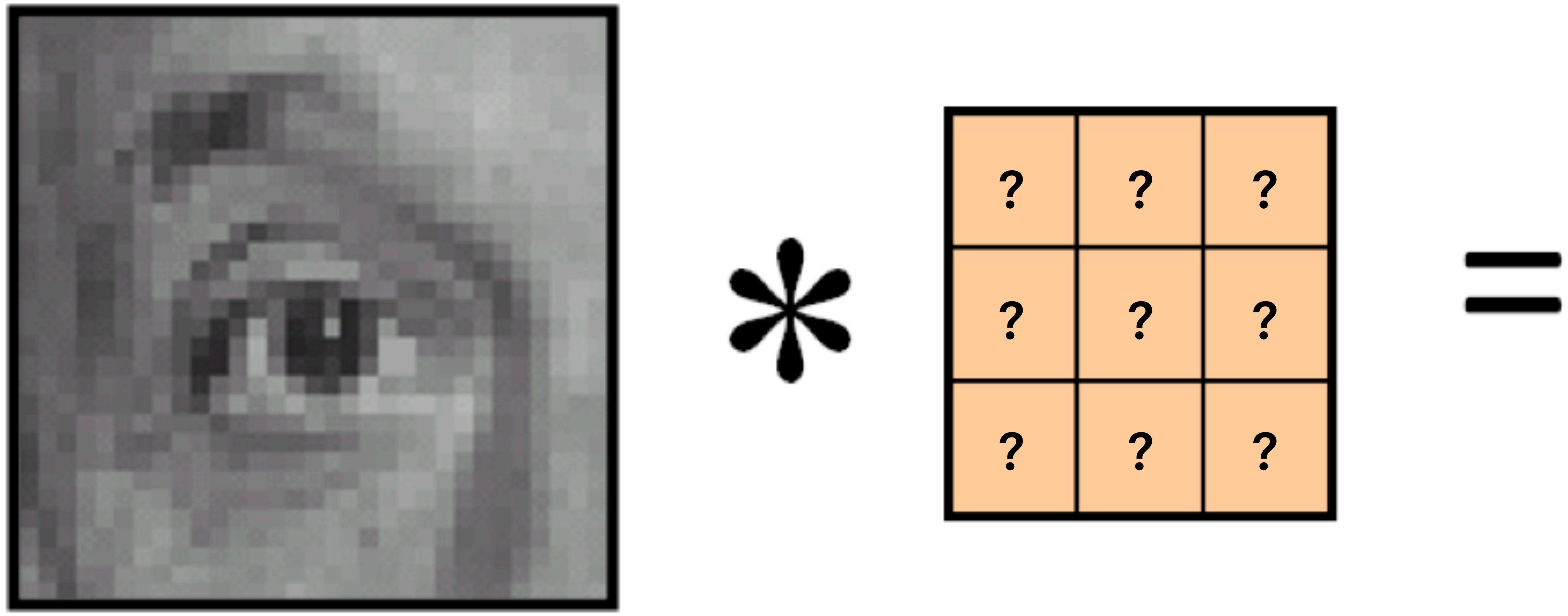
Sharpening filter
(accentuates edges)



- **Idea.** Learn the convolutional filter, not all the linear connections!

  (no need to care about “flipping” the kernel $f(\tau) \to f(t - \tau)$)

- **Properties.**
  - **Translation-Equivariance**
    We apply the same **operation** to patches in different locations.



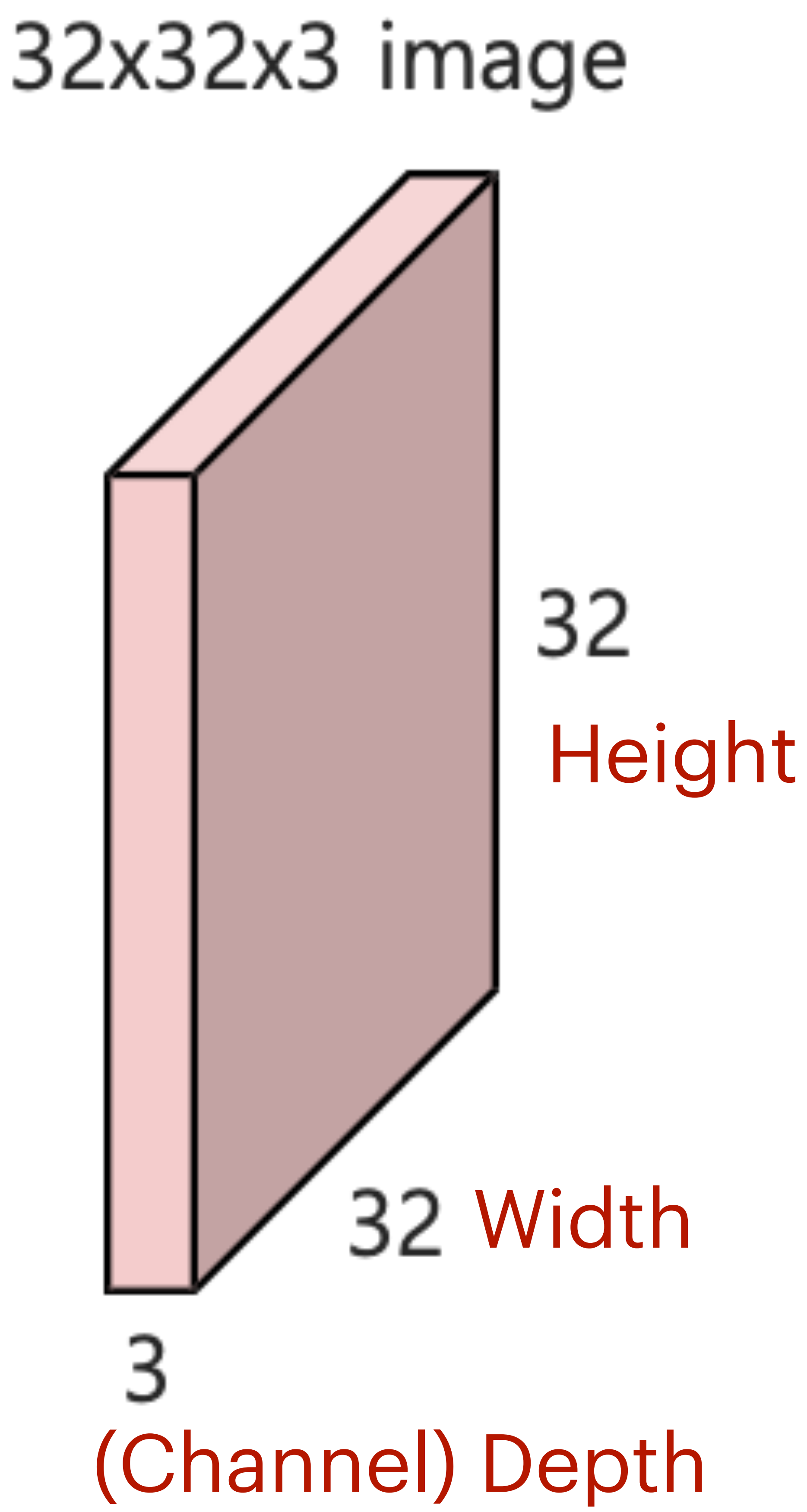
- **Local Connectivity**
  In most cases, we use 3 x 3 kernels.
  Reflects the belief that it is unlikely that far-away pixels are meaningful.
- **Parameter-Efficiency**
  For 3×3 convolution, only have 9 parameters per kernel.
  (but it is common to use many parallel kernels, leading to “channels” in hidden layers)


POOL
POOL
POOL
RELU
RELU
RELU
RELU
RELU
RELU
CONV
CONV
CONV
CONV
CONV
CONV
FC
car
truck
airplane
ship
horse

Convolution—more concretely

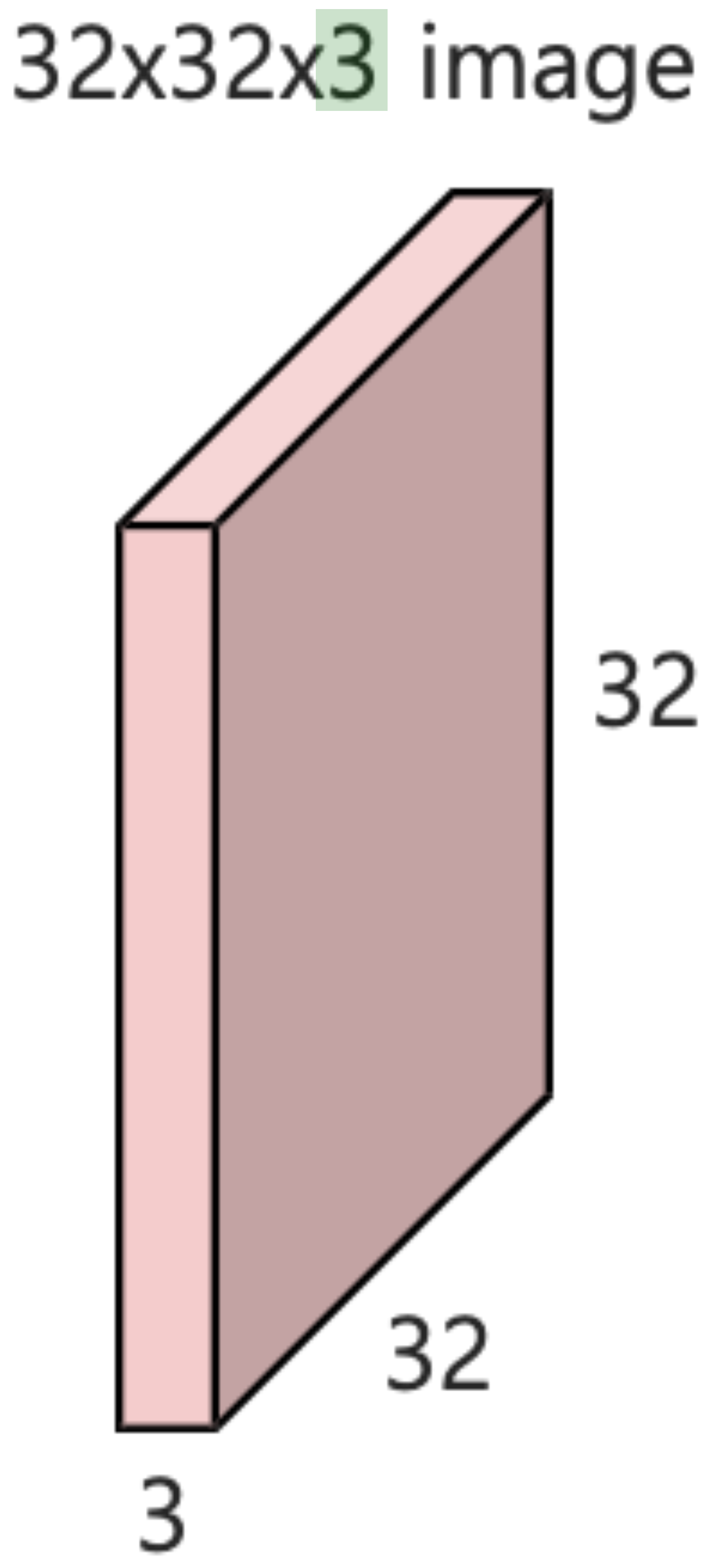
Convolutional Layer

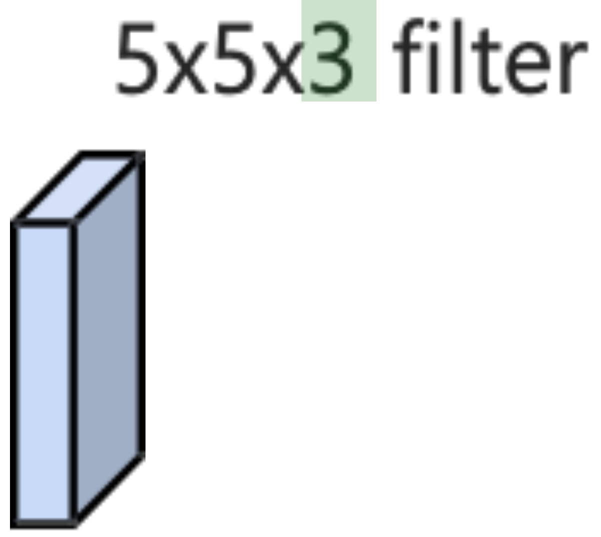
- Begin with a 32×32 image with 3 channels (RGB).



Convolutional Layer

- Convolve with conv filter. — dot product with a sliding “receptive field”



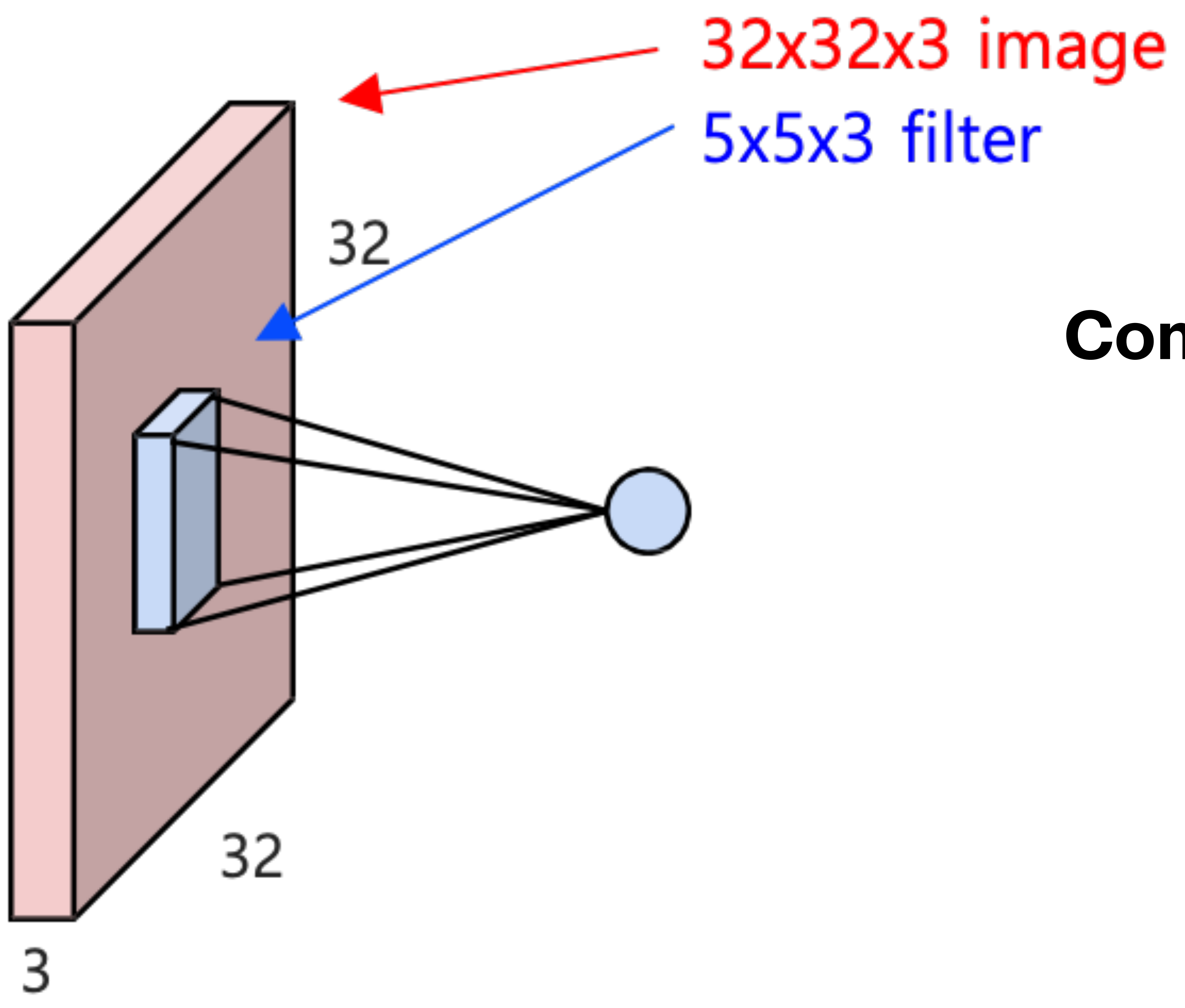


Classic Models. Filters utilize the full channel depth.

Modern. For efficiency, we apply depth-1 convolution for each input channel (called “depthwise convolution”)

Convolutional Layer

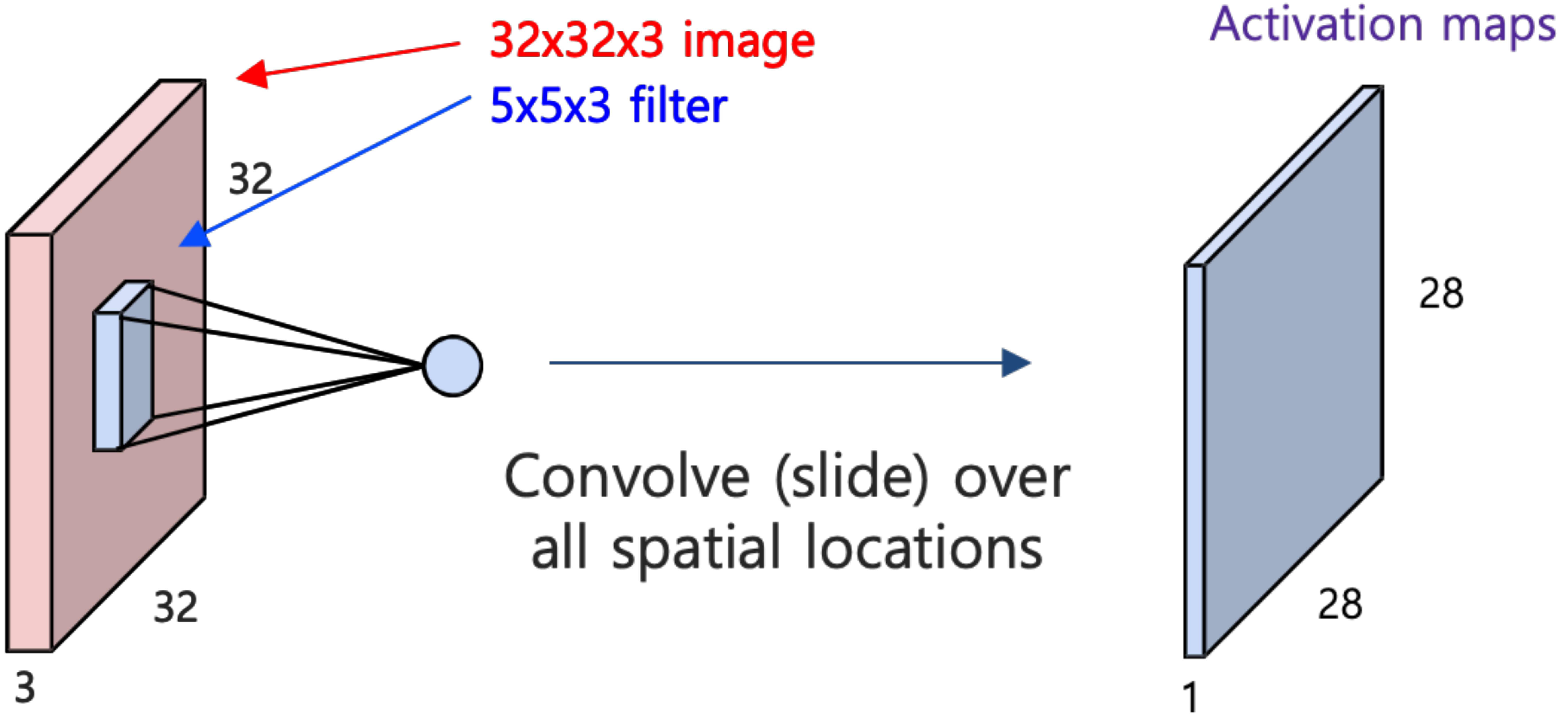
- Convolving generates a single entry of the layer output

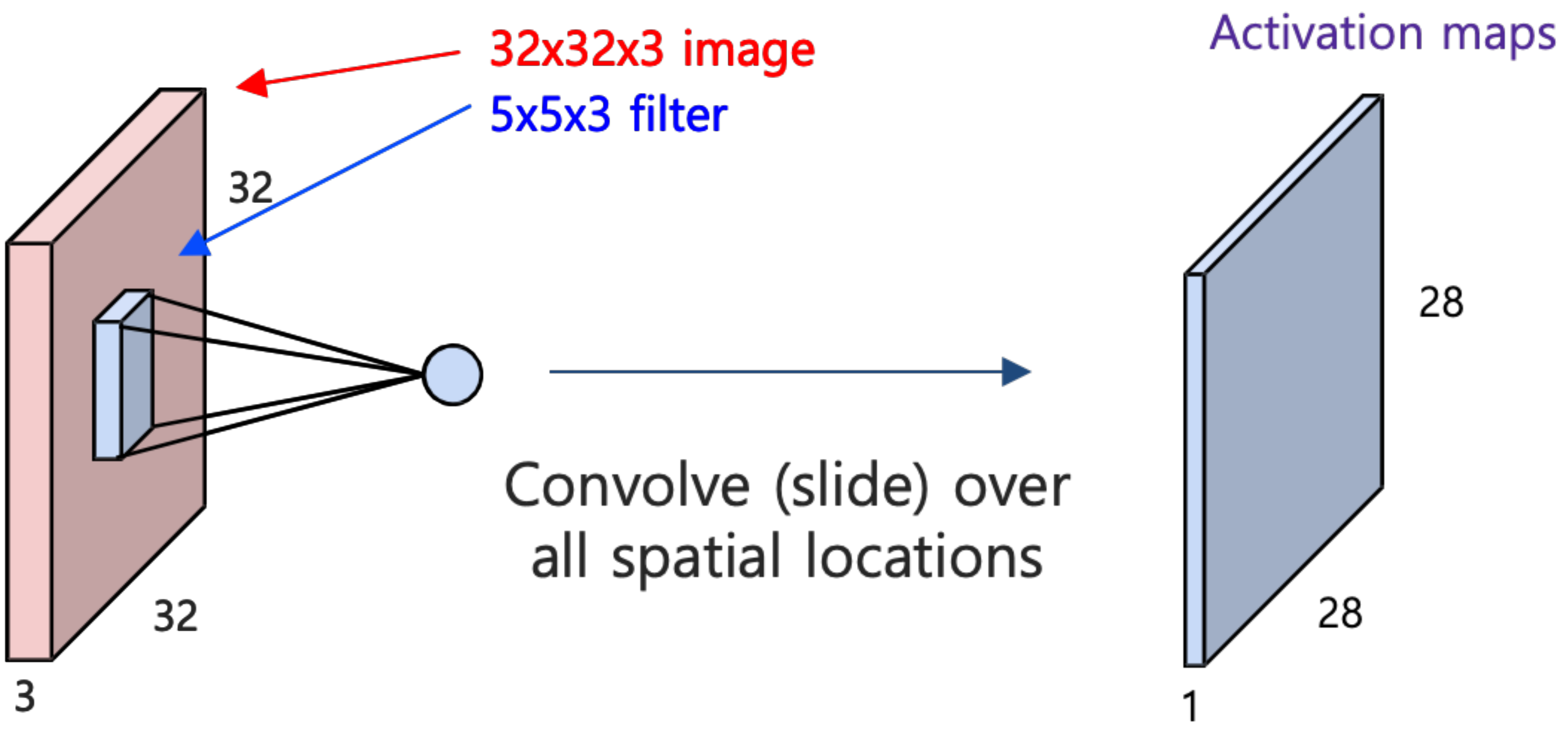


Compute. Dot product of two tensors with $5 \times 5 \times 3 = 75$ dimensions
(+ bias addition)

Convolutional Layer

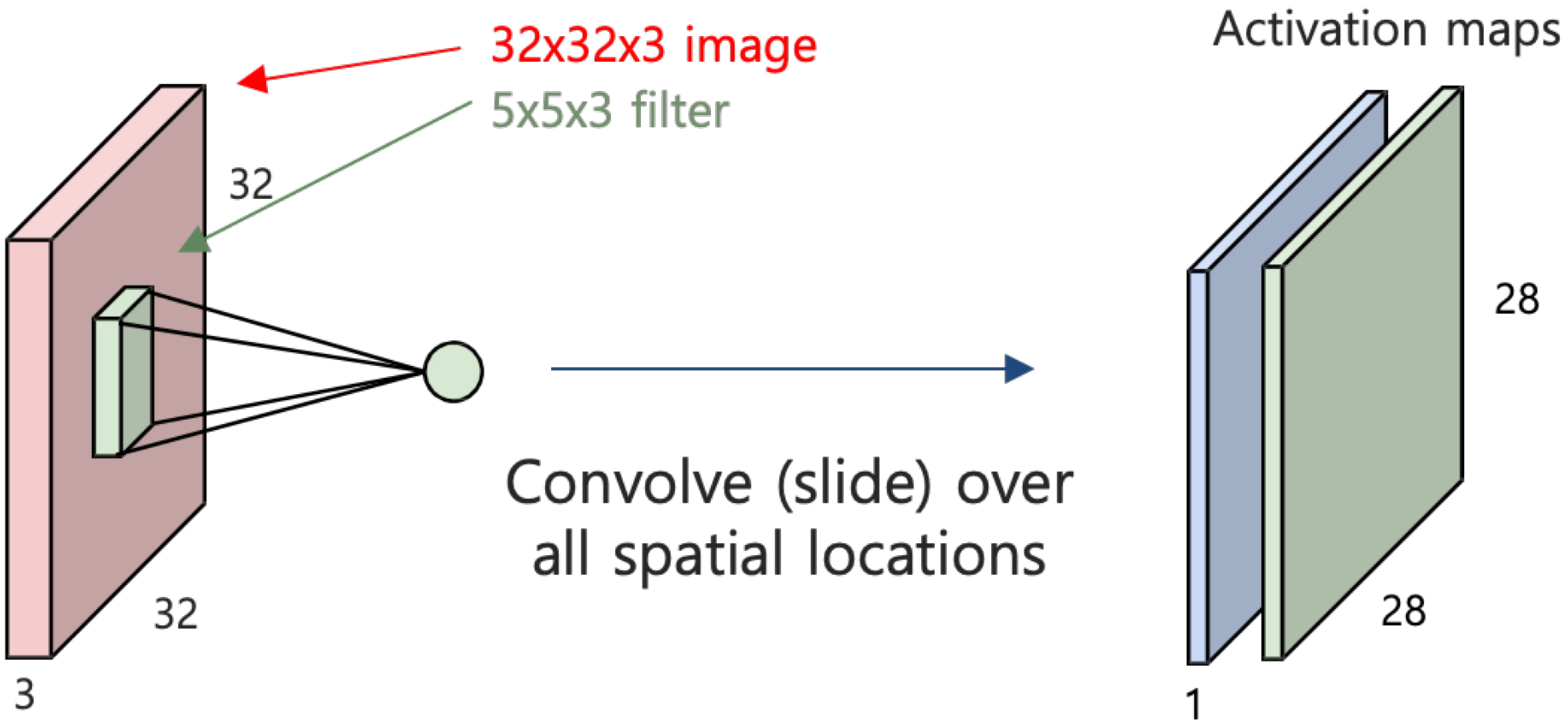
- Convolving generates a single entry of the layer output




32x32x3 image
5x5x3 filter
32
32
3
Convolve (slide) over
all spatial locations
Activation maps
28
28
1

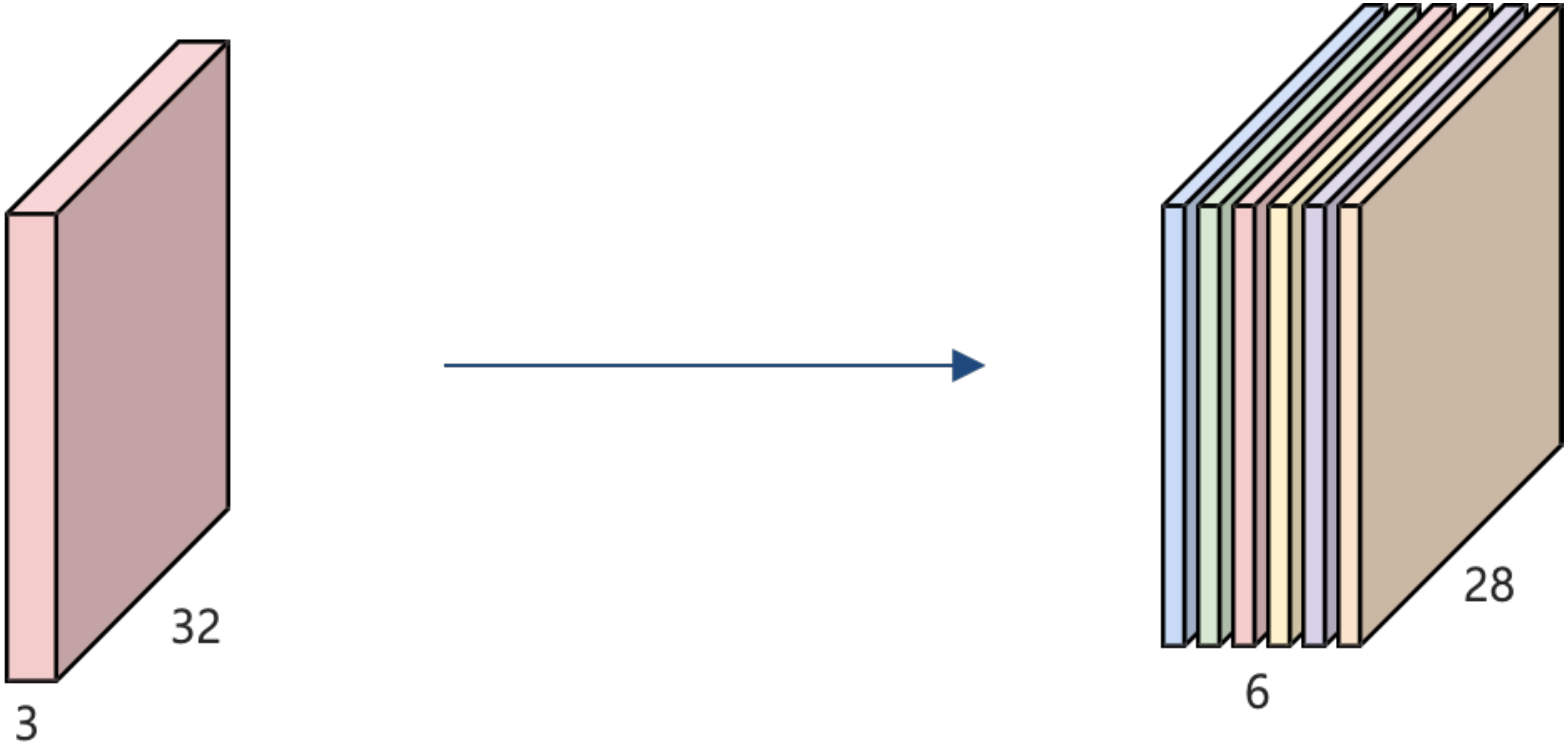
Convolutional Layer

- Do it for the second filter



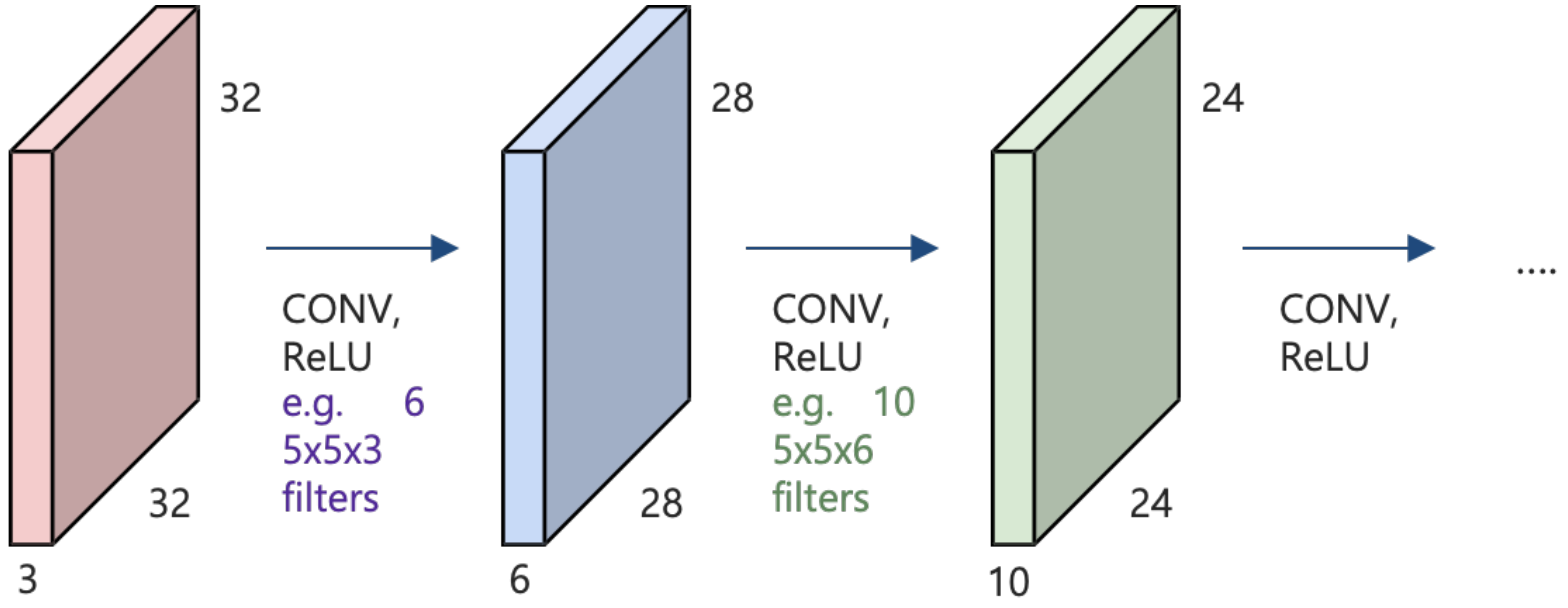
Convolutional Layer

- Generate the pre-activation with **multiple depth** (called “channels”)



Convolutional Layer

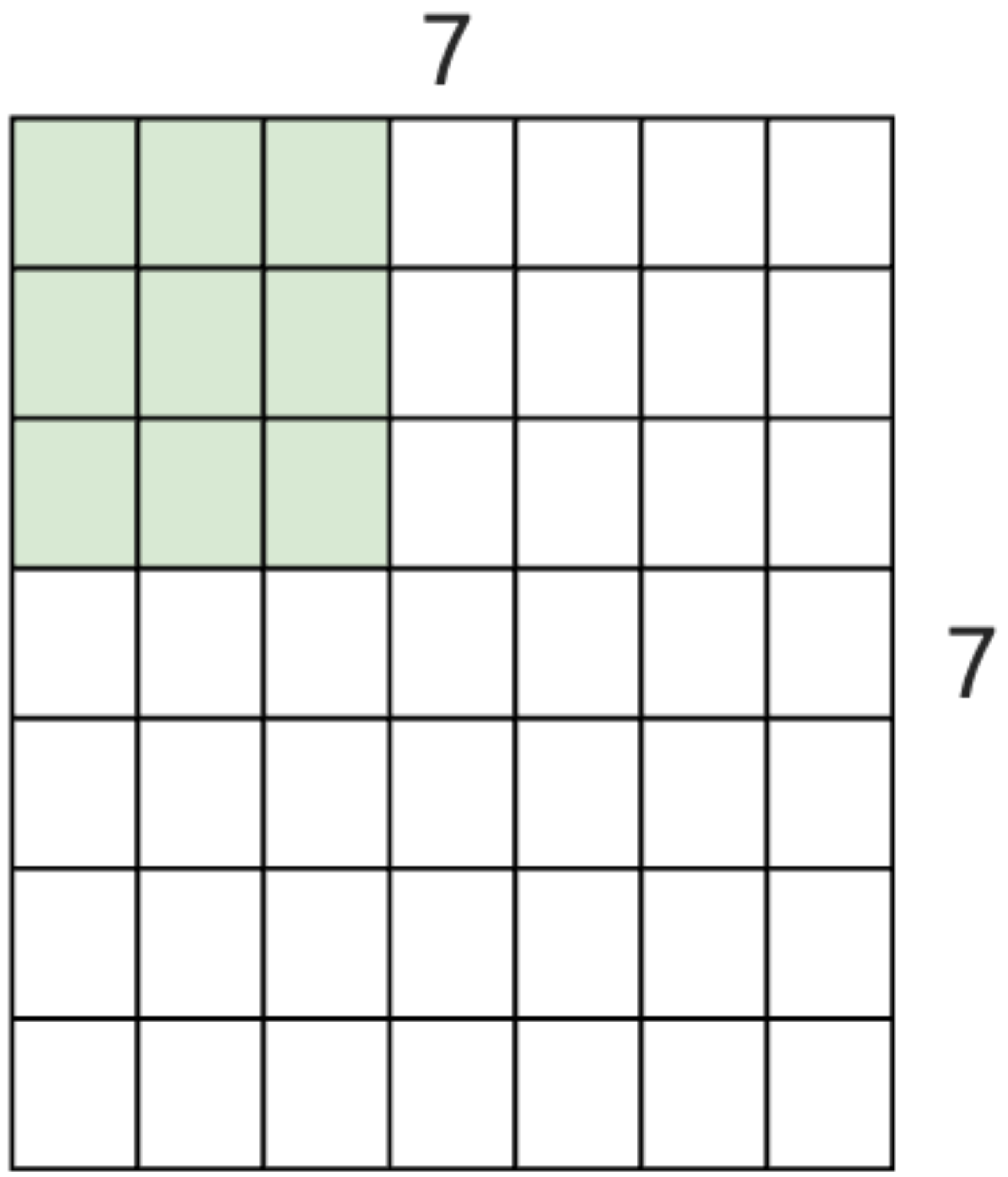
- Stack the layers, with activation functions in between!



Convolution—Spatial Dimension

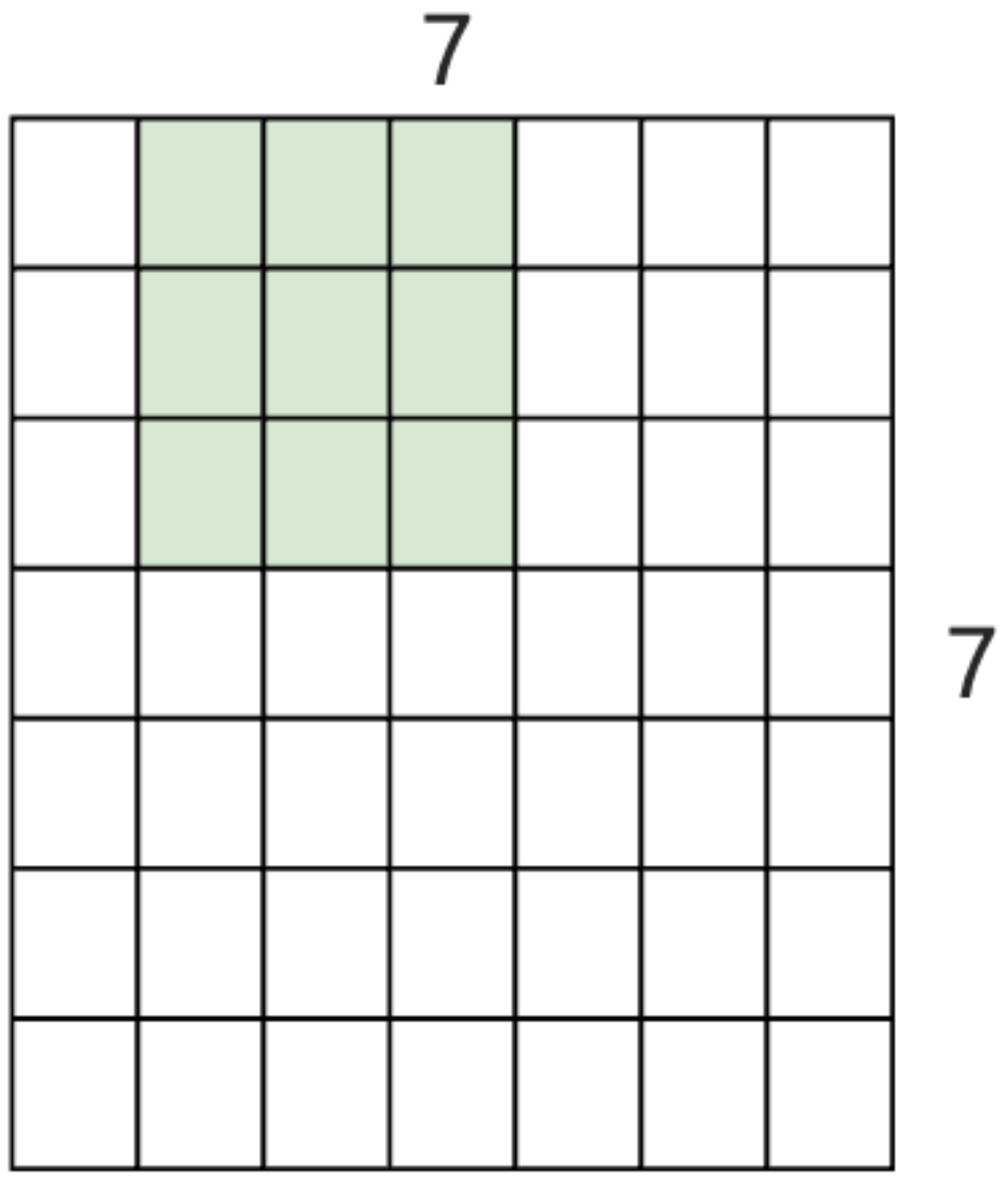
Spatial Dimension

- Consider a 7×7 image, with 3×3 filters.



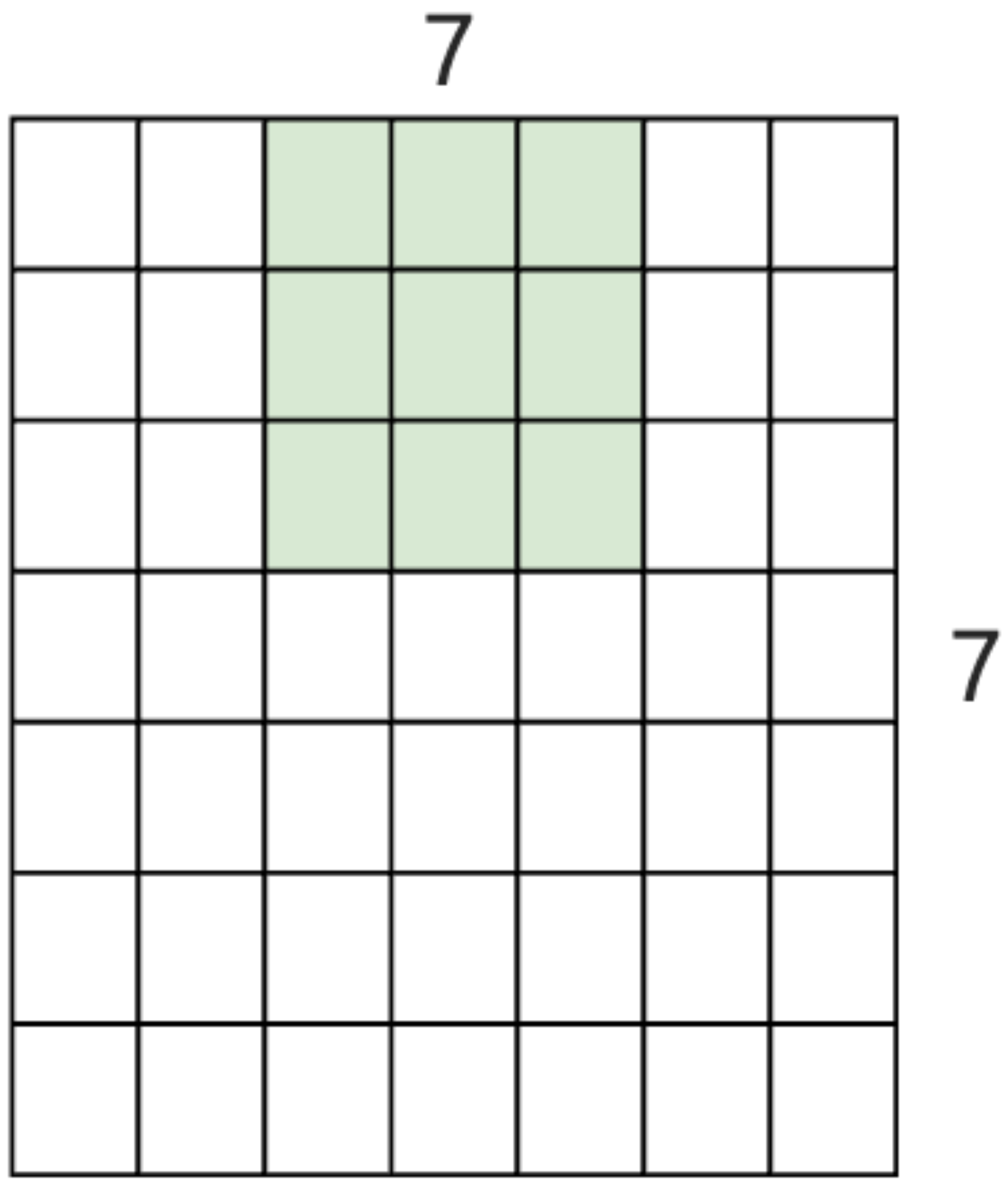
Spatial Dimension

- Consider a 7×7 image, with 3×3 filters.



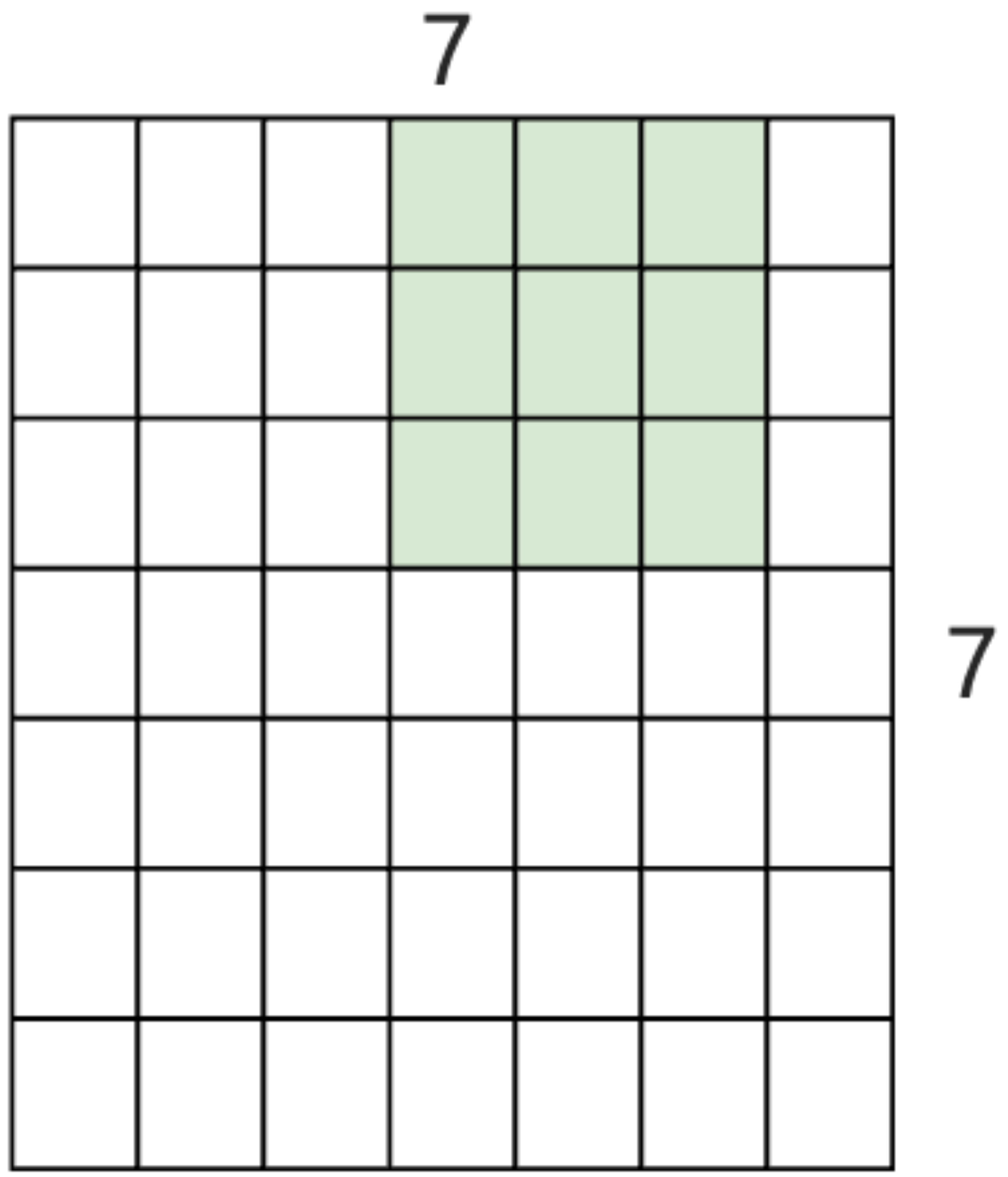
Spatial Dimension

- Consider a 7×7 image, with 3×3 filters.



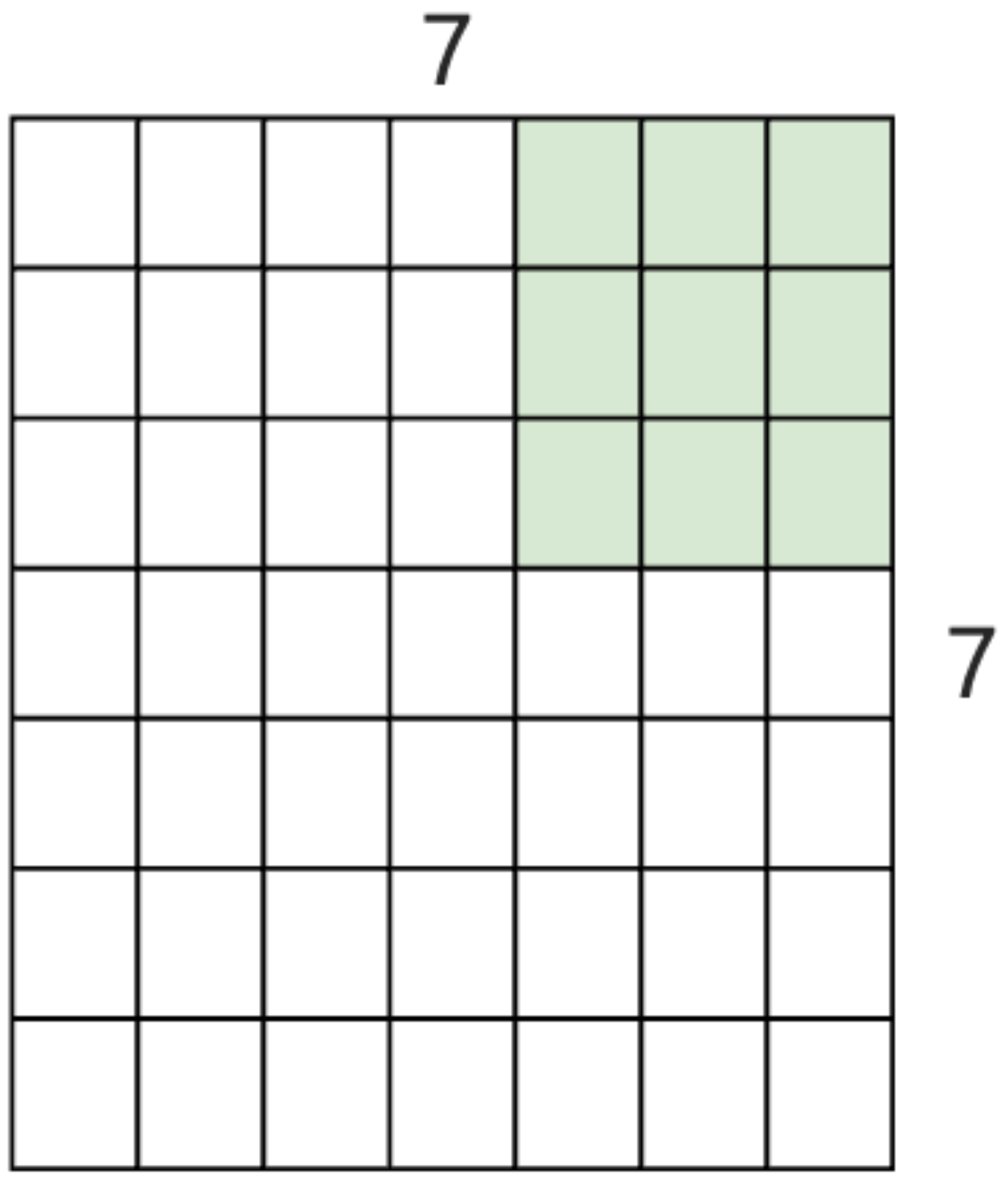
Spatial Dimension

- Consider a 7×7 image, with 3×3 filters.



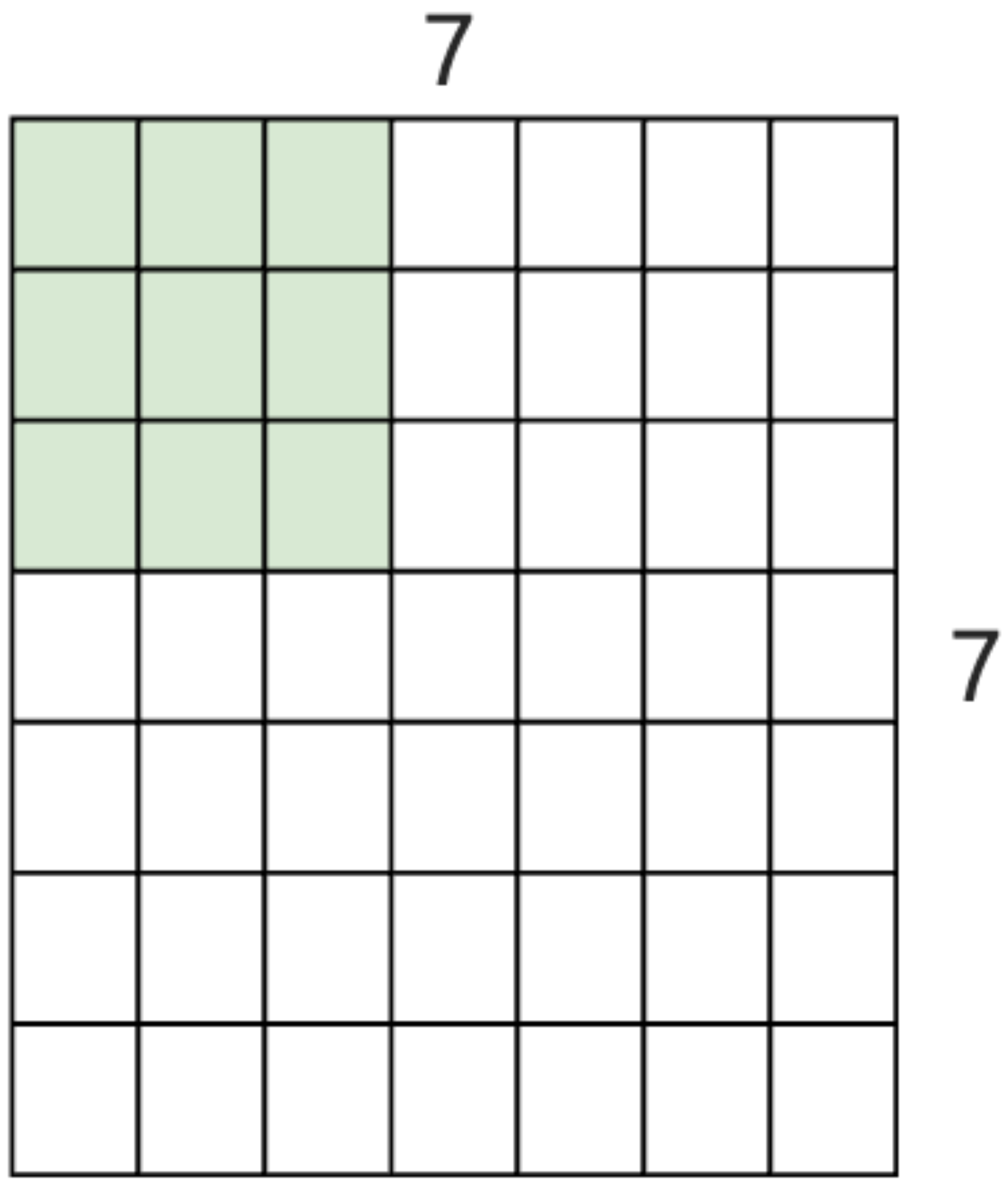
Spatial Dimension

- Consider a 7×7 image, with 3×3 filters. $\Rightarrow 5 \times 5$ output



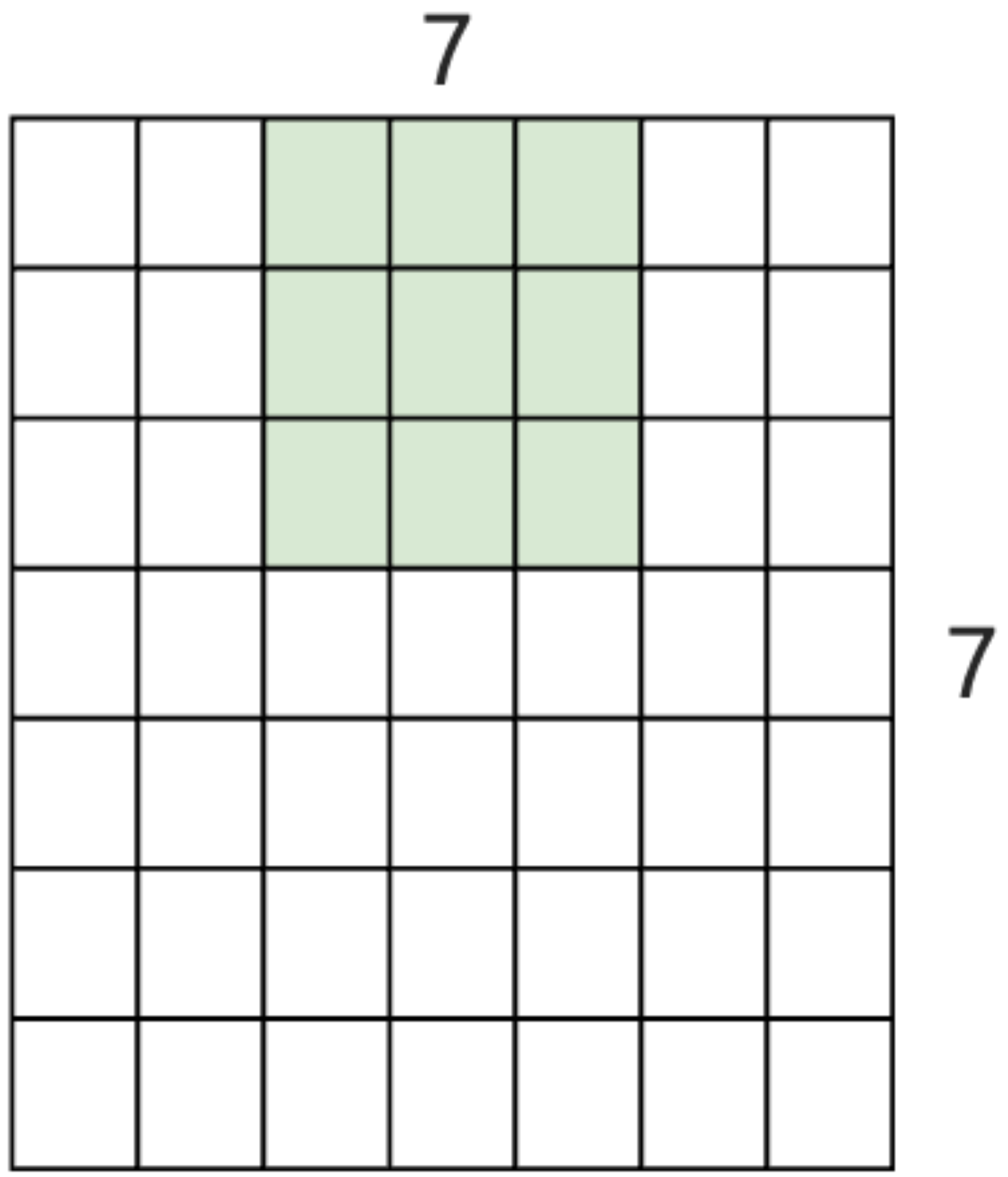
Spatial Dimension

- It is common to apply “strides”—with stride 2...



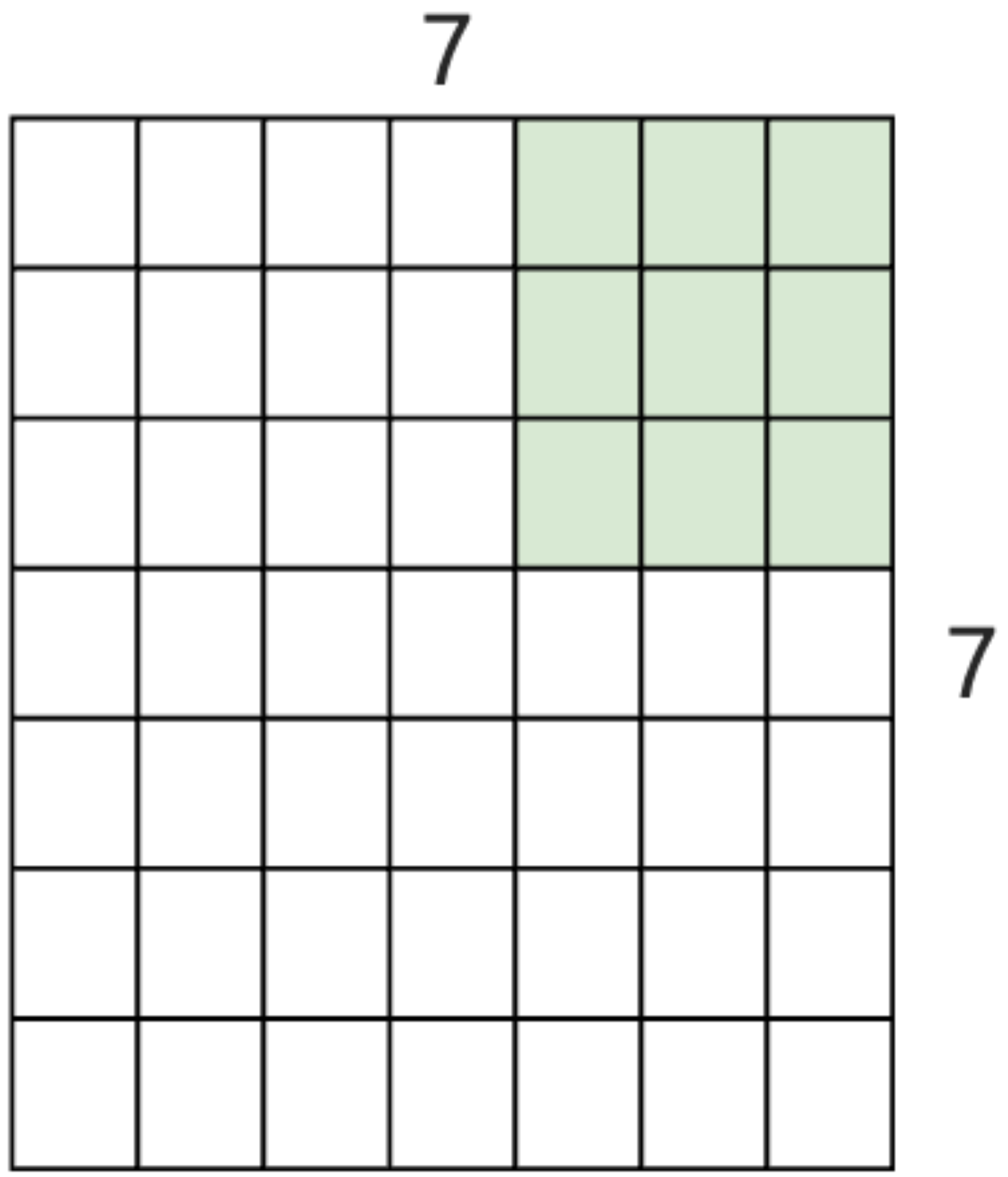
Spatial Dimension

- It is common to apply “strides”—with stride 2…



Spatial Dimension

- It is common to apply “strides”—with stride 2... $\Rightarrow 3 \times 3$ output



Note. The stride 3 does not fit for this case, and thus cannot be used.

Output Size. (Image length - Filter length) / Stride + 1.

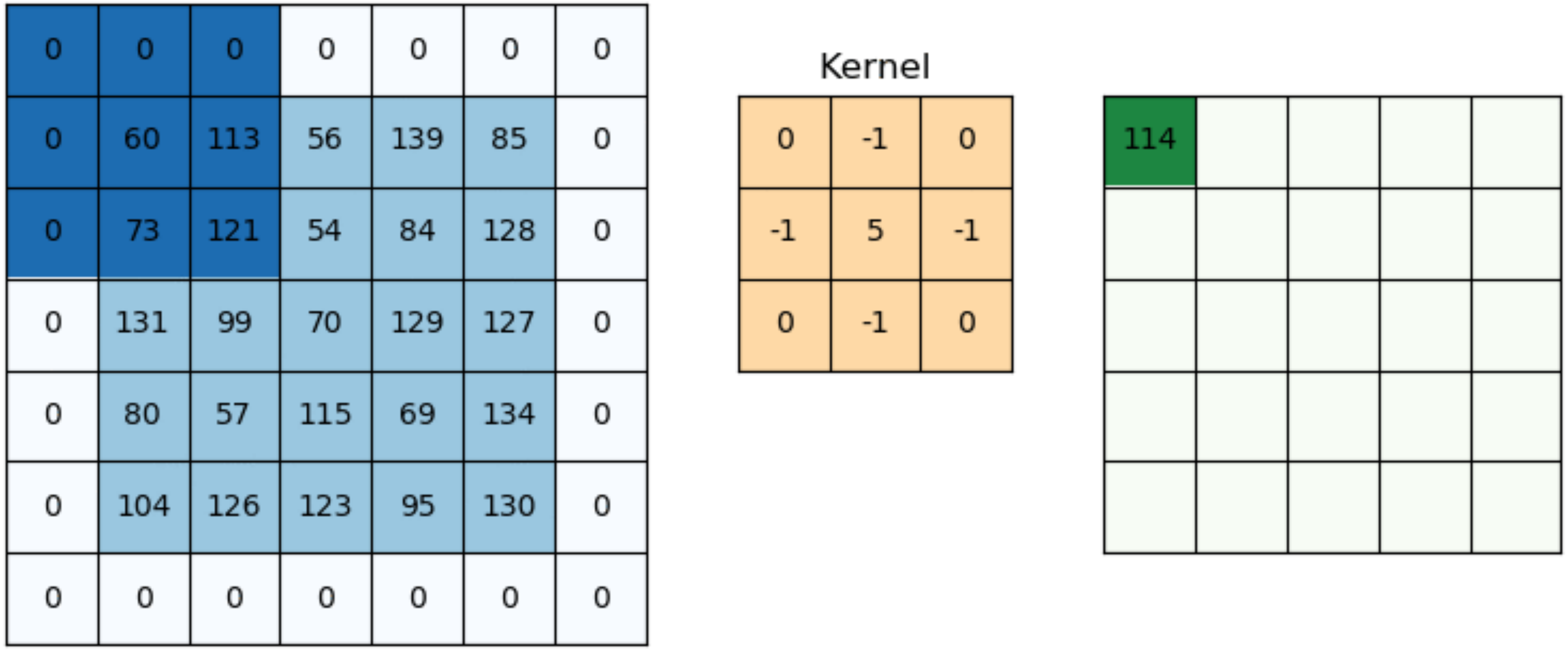
Stride 1: $(7-3)/1+1=5$

Stride 2: $(7-3)/2+1=3$

Stride 4: $(7-3)/4+1=2$

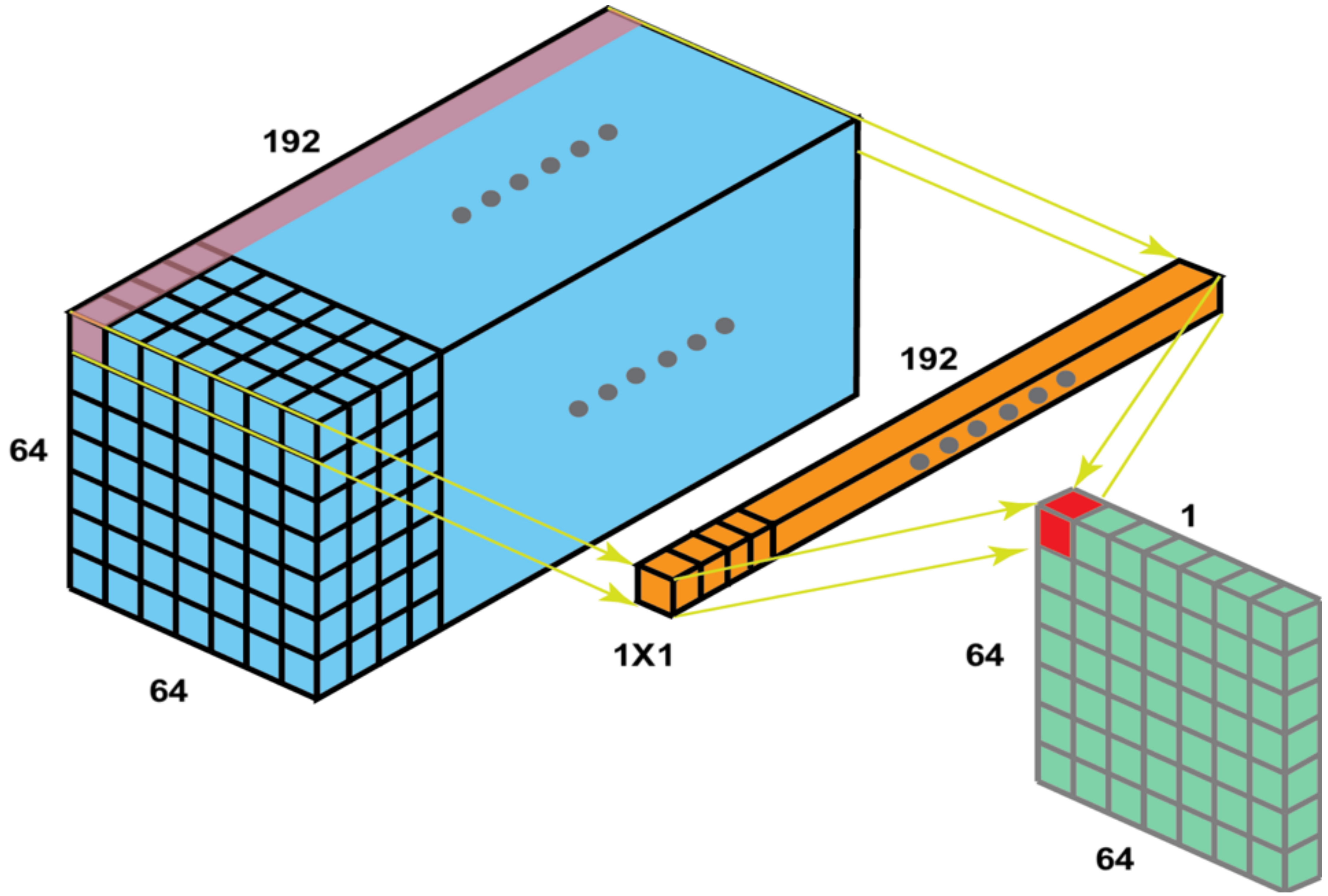
Spatial Dimension

- It is common to apply “zero-paddings”
  - Image size does not reduce, and thus can use more layers.



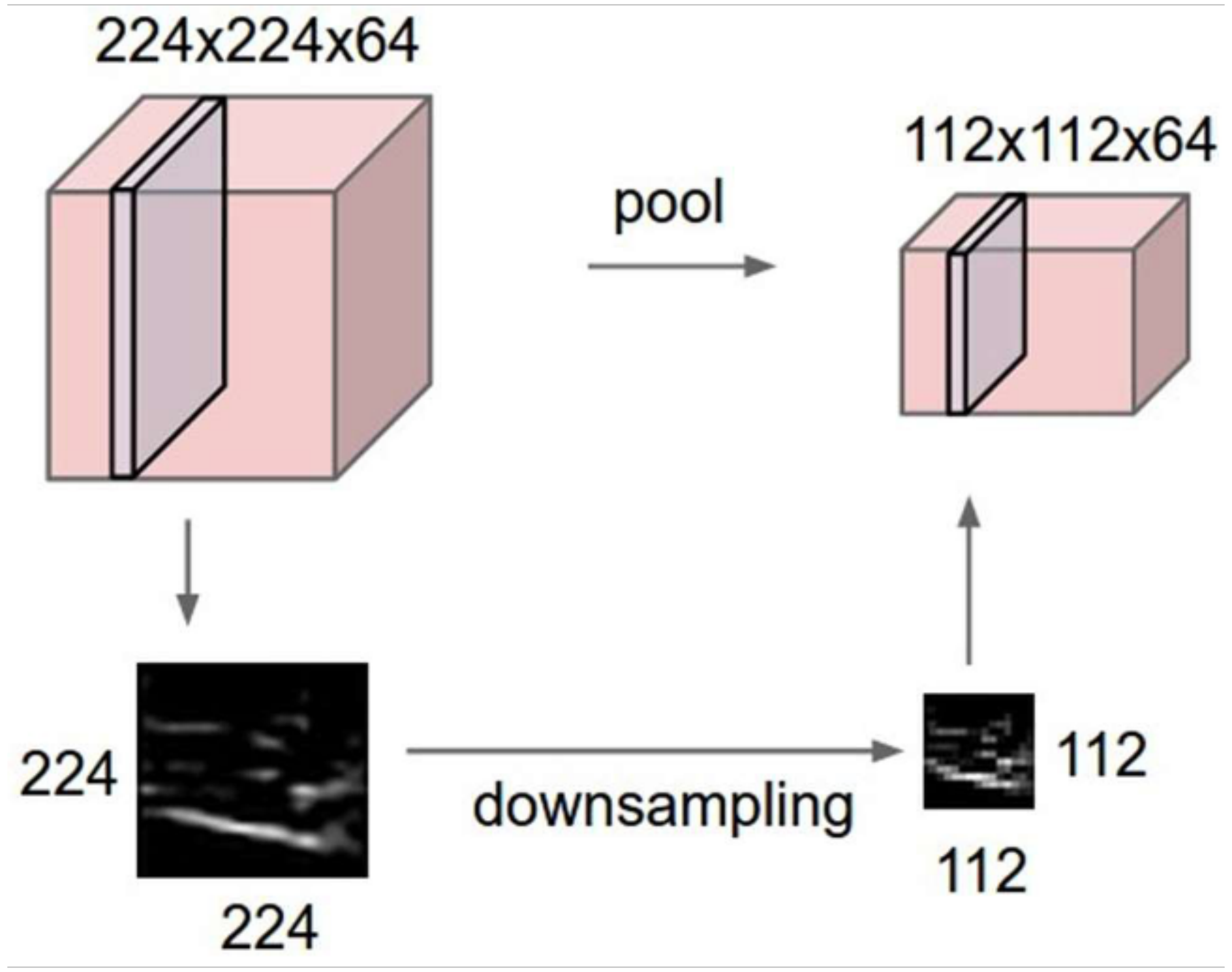
Spatial Dimension

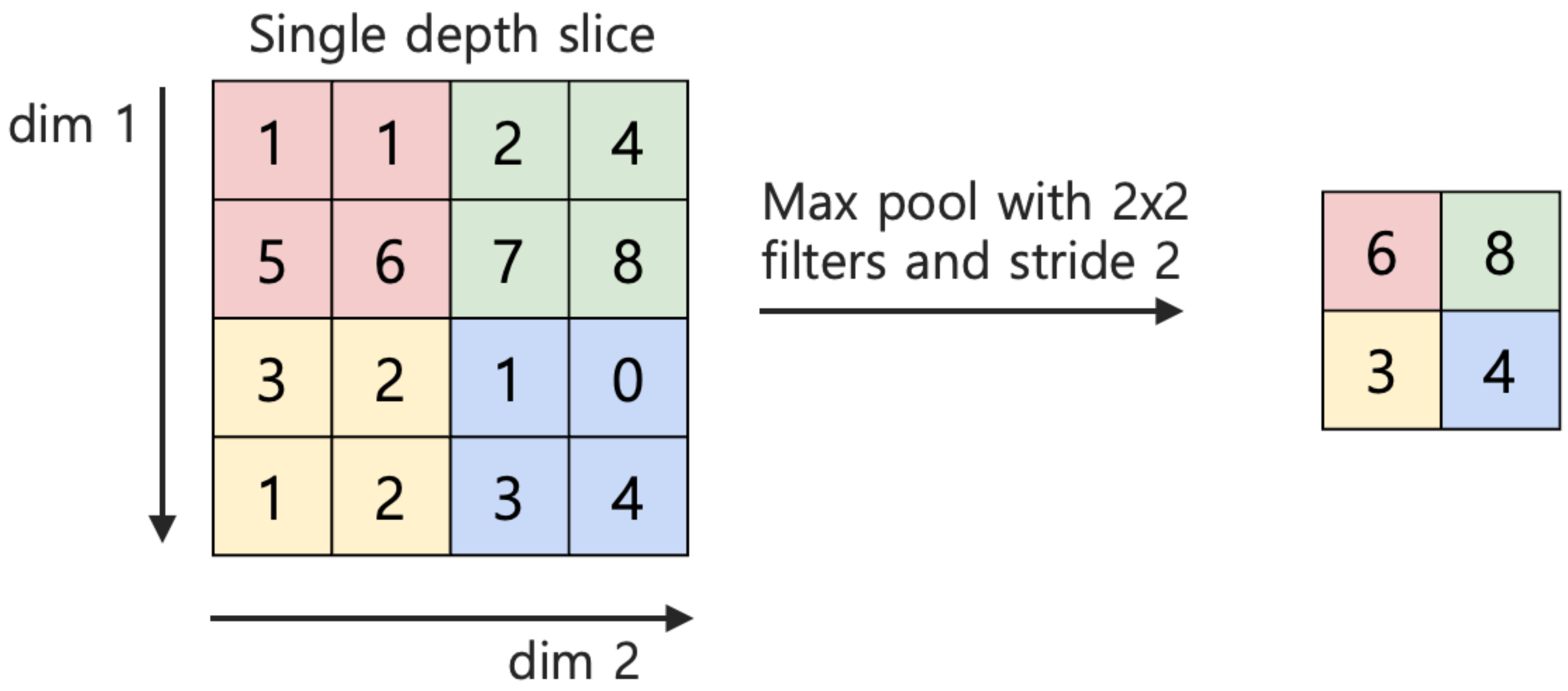
- It is common to use "1×1 convolution"
  - Increase or decrease the number of channels via linear combination—often used together with depthwise conv.



Pooling Layer

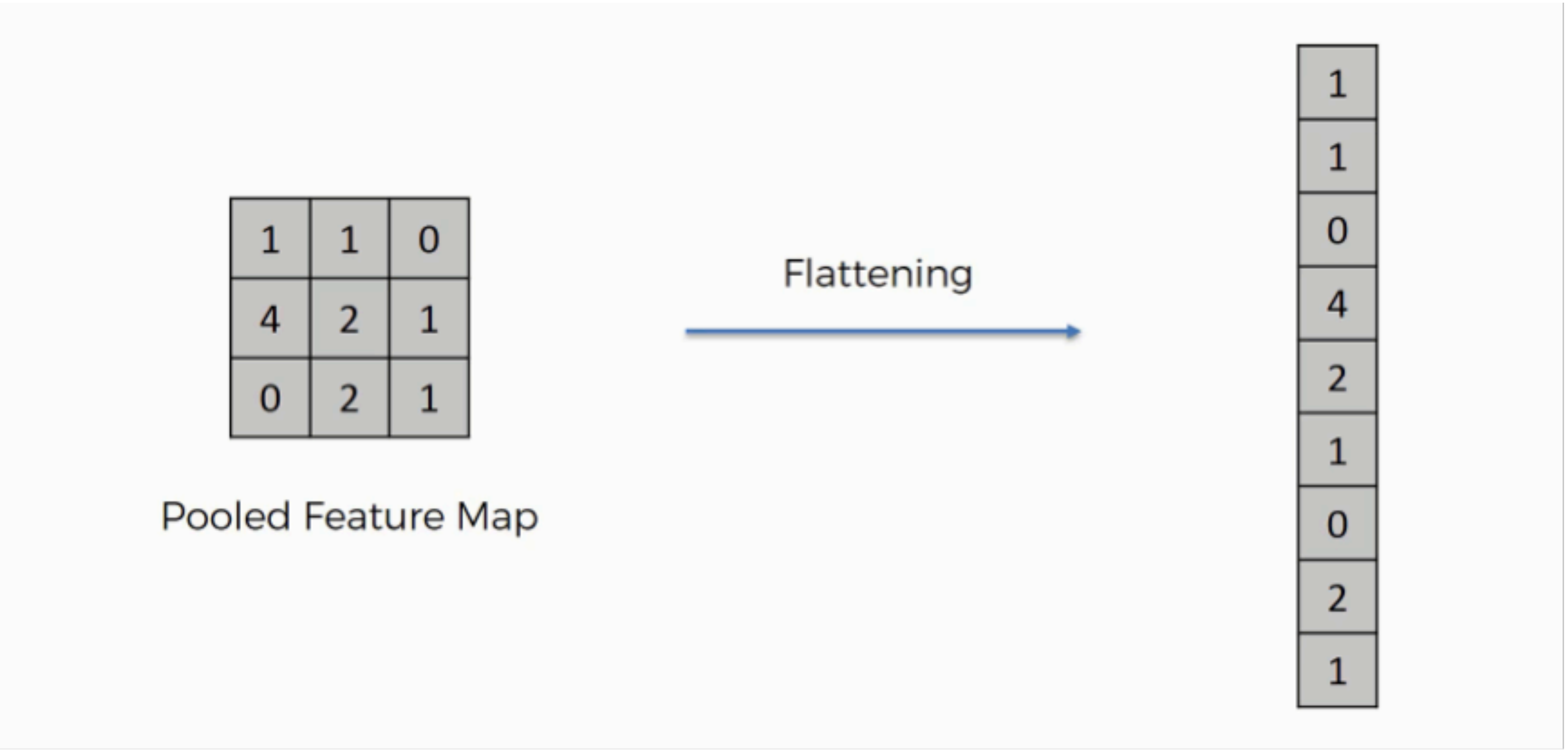
- Reduces the spatial dimension by taking **max**/mean/else of pixels.
  - Gets smaller resolution, without losing information.
    (e.g., the activation represents a specific feature)





Final Layer — Fully-Connected

- In the final layer, it is common to "flatten" the features.
    - Then, we perform linear classification/regression.



Additional Remarks

- Convolutional layer can be applied on images of any size.
  - For segmentation-like cases (no FC layer), a model trained on 178×178 image can be used on 256×256 images.

Cheers

- *Next up.* GD and Backprop