# Deep Learning

Ulf Brefeld & Soham Majumder

build: May 6, 2024

Machine Learning Group
Leuphana University of Lüneburg

# Convolutions

Treating large-dimensional inputs as *unstructured* vectors leads to intractable models

For example, a linear layer with a $256 \times 256$ RGB image as input and an image of the same size as output requires

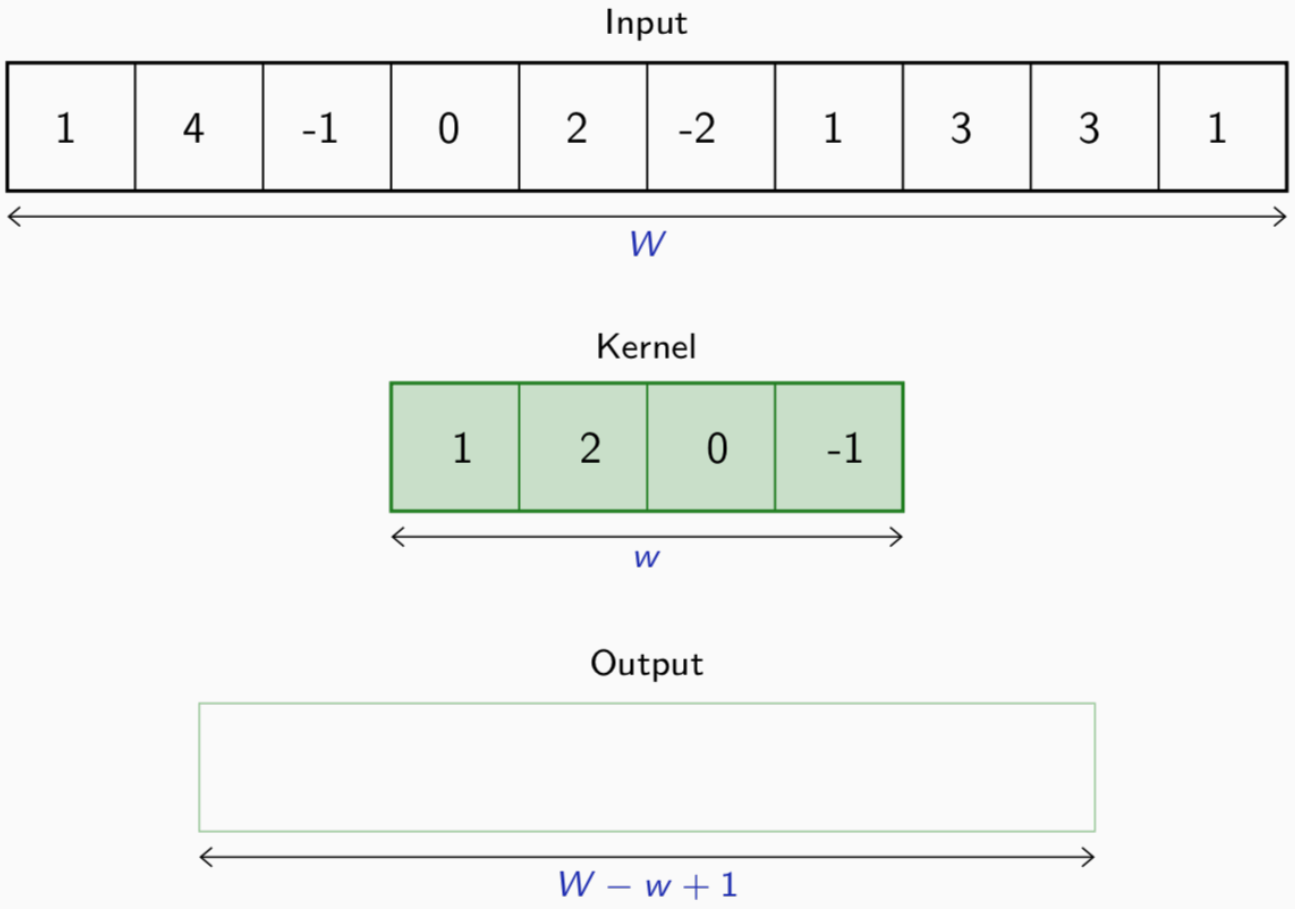
$$(256 \times 256 \times 3)^2 \approx 3.87 \times 10^{10}$$

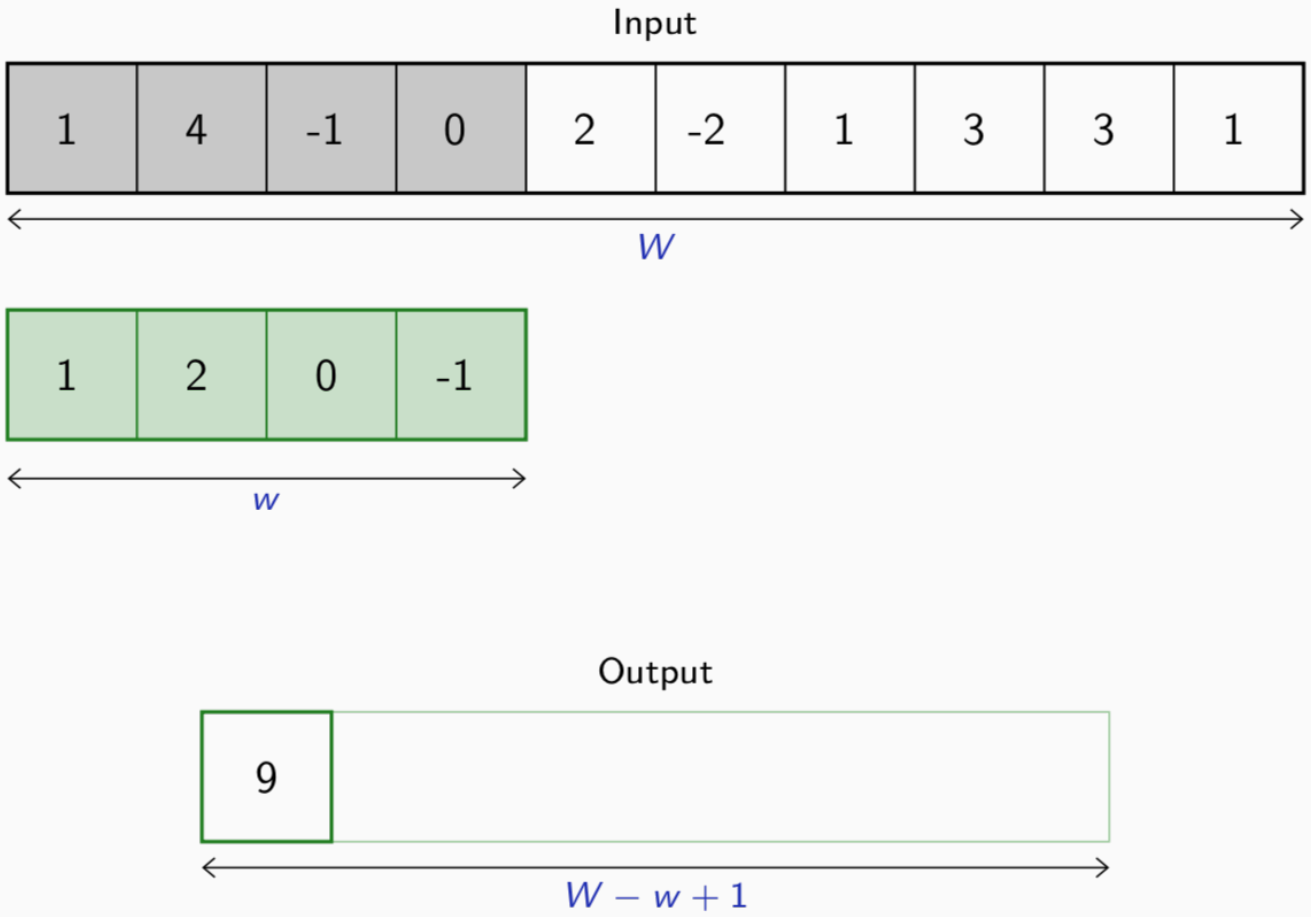
parameters, with the corresponding memory footprint ($\approx 150$GB)

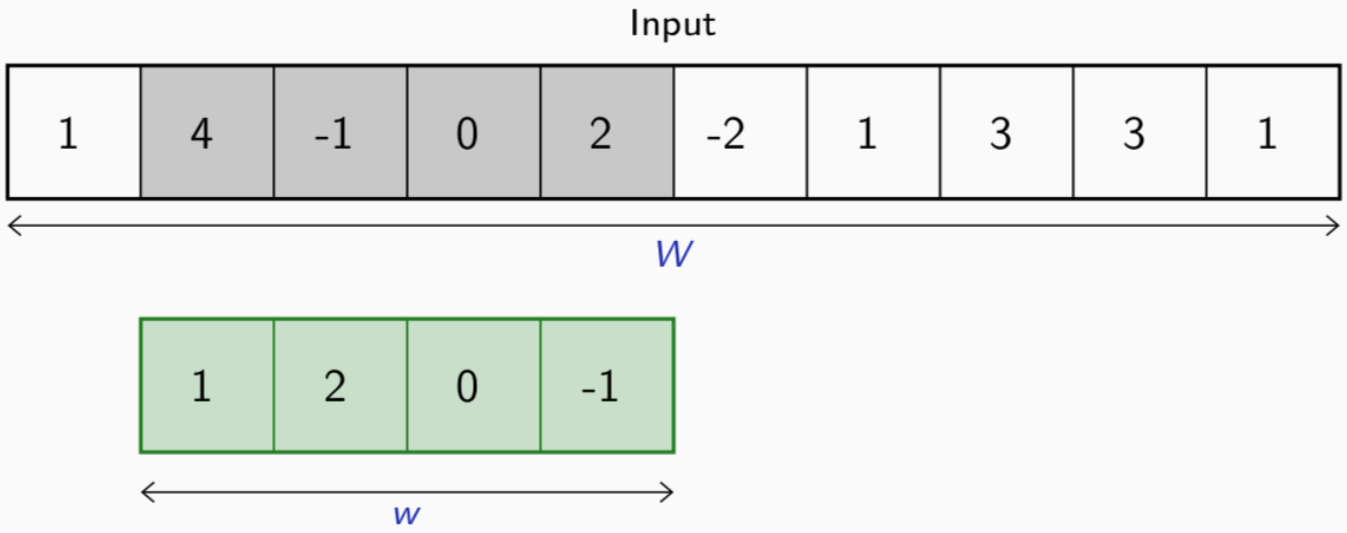
Moreover, this requirement is inconsistent with the intuition that such large signals have some *invariance in translation*.

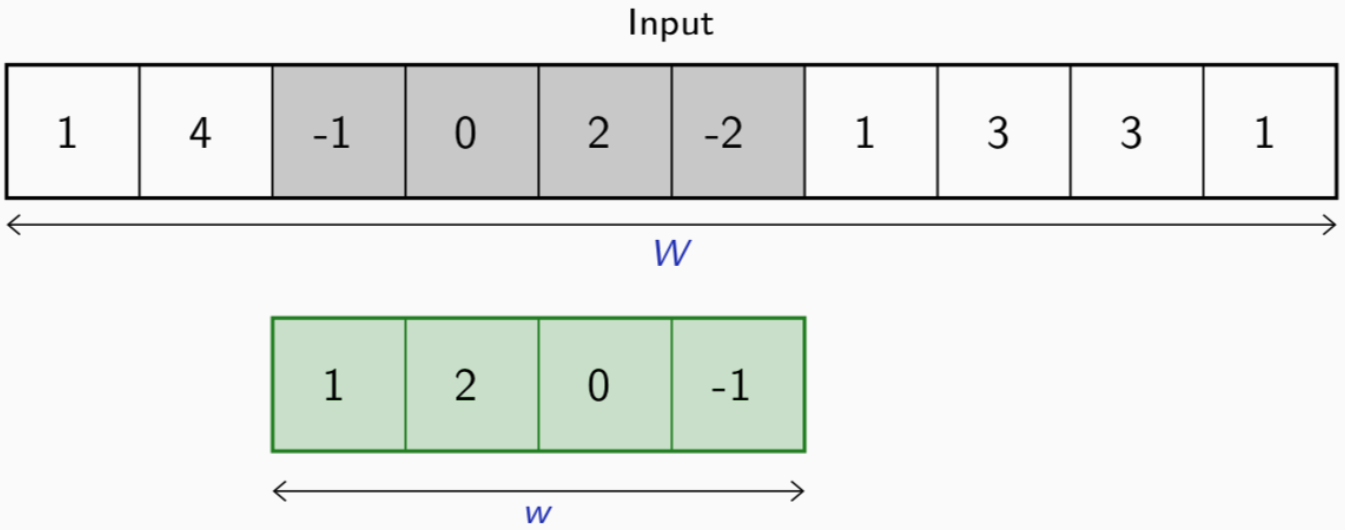
**A representation meaningful at a certain location can or should be used everywhere**

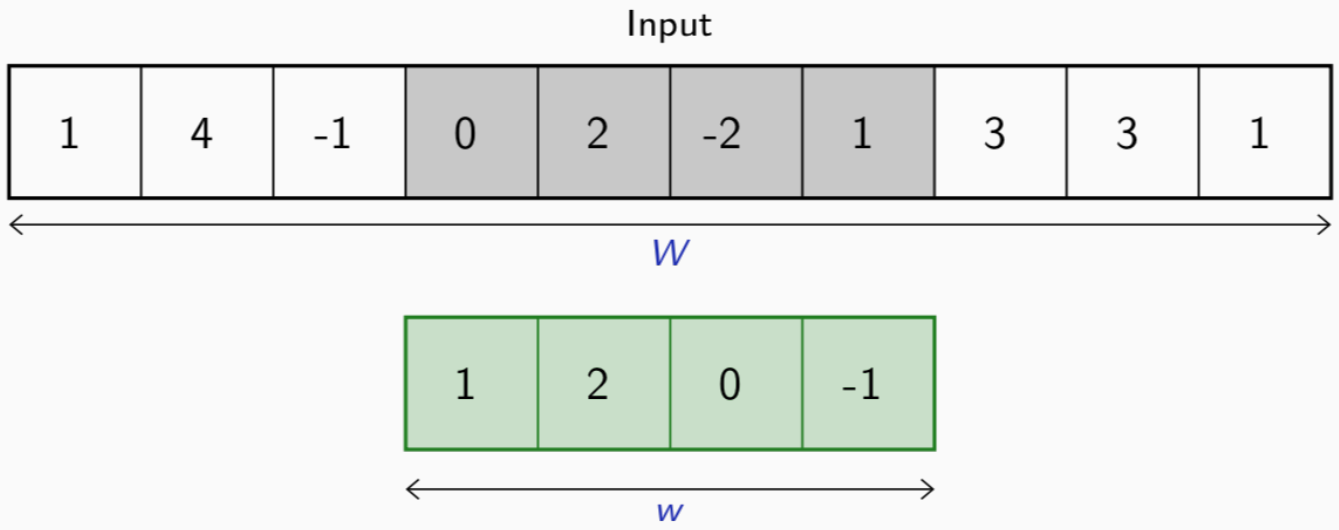
A *convolution* embodies this idea. It applies the same linear transformation locally, everywhere, and preserves the signal structure.

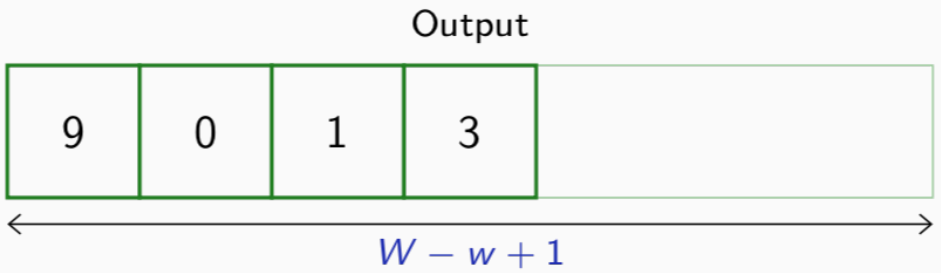

Input
1
4
-1
0
2
-2
1
3
3
1
W
Kernel
1
2
0
-1
w
Output
W − w + 1

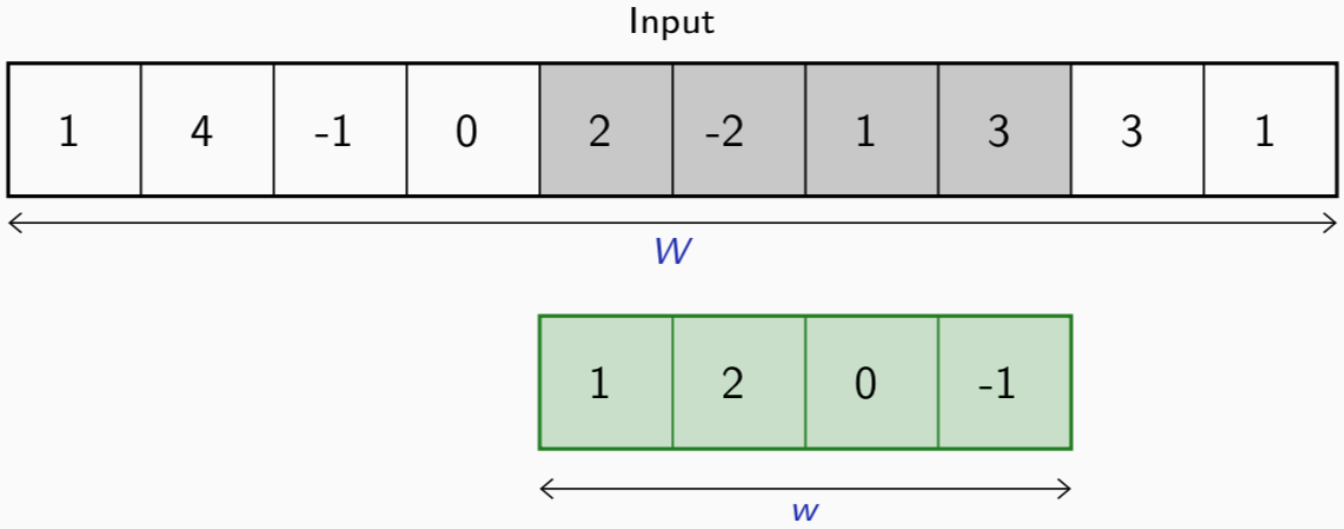

Input
1
4
-1
0
2
-2
1
3
3
1
W
1
2
0
-1
w
Output
9
W − w + 1

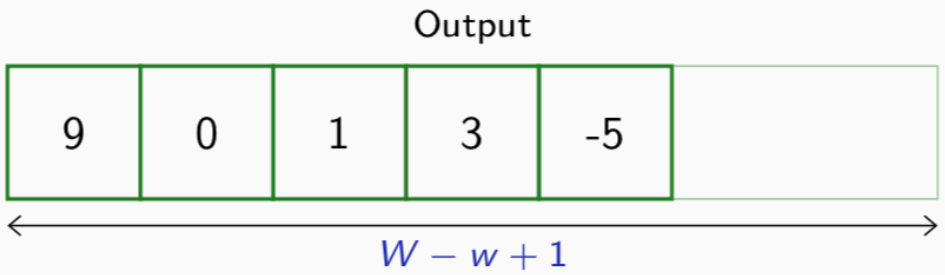

Input
1
4
-1
0
2
-2
1
3
3
1
W
1
2
0
-1
w
Output
9
0
W − w + 1

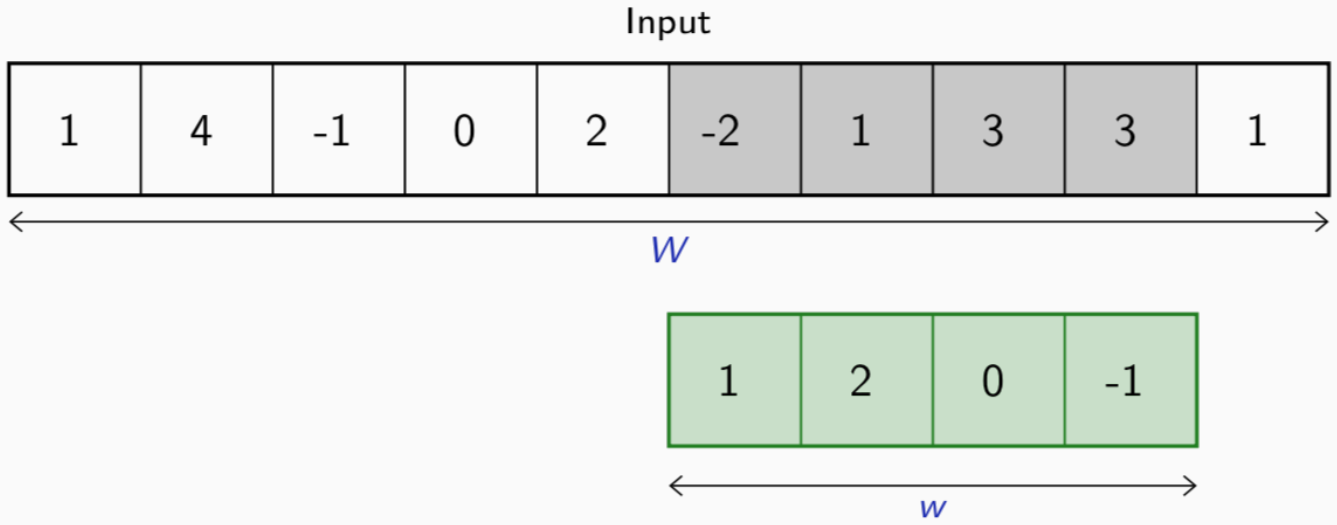

Input
1
4
-1
0
2
-2
1
3
3
1
W
1
2
0
-1
w
Output
9
0
1
W − w + 1

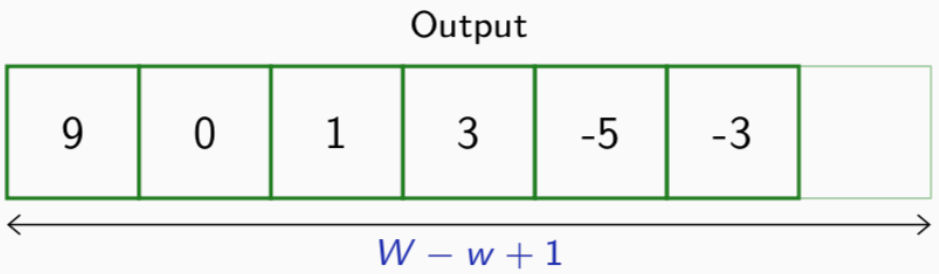

Input
1
4
-1
0
2
-2
1
3
3
1
W
1
2
0
-1
w

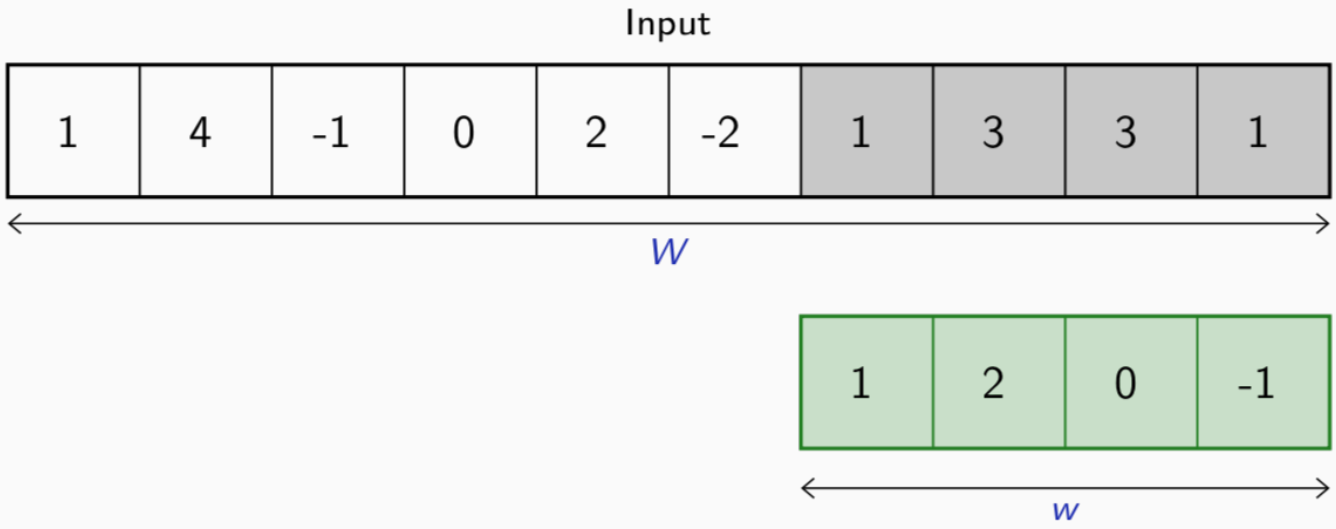

Output
9
0
1
3
W − w + 1

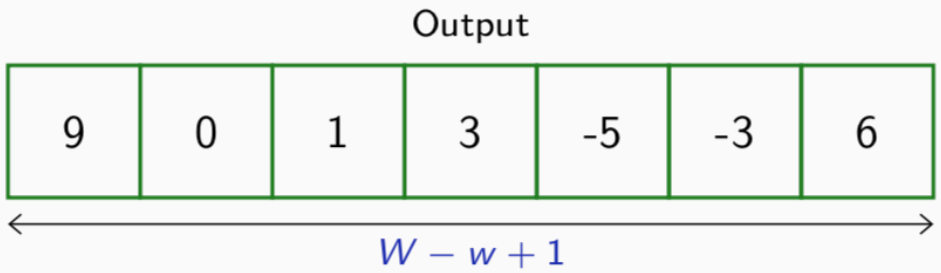

Input
1
4
-1
0
2
-2
1
3
3
1
W
1
2
0
-1
w

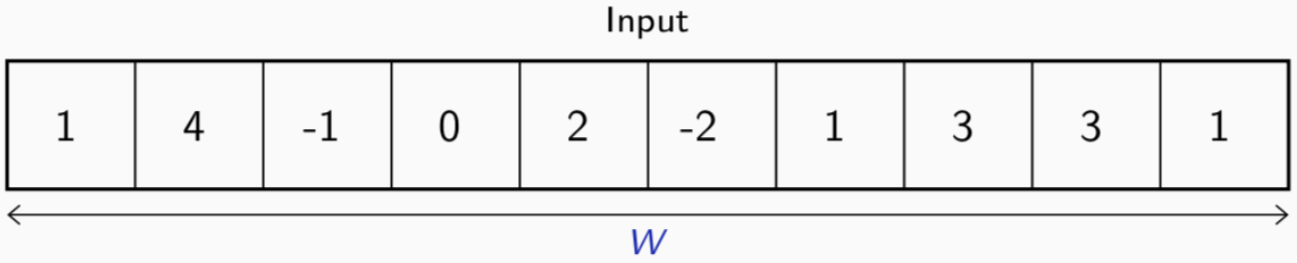

Output
9
0
1
3
-5
W − w + 1

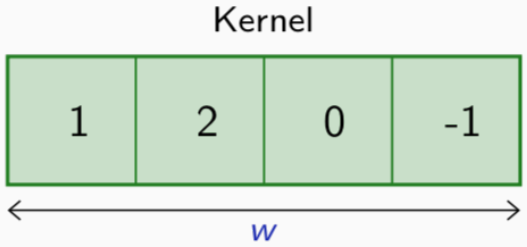

Input
1
4
-1
0
2
-2
1
3
3
1
W
1
2
0
-1
w

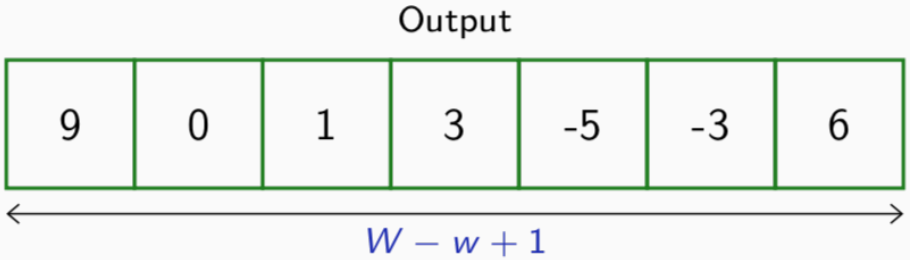

Output
9
0
1
3
-5
-3
W − w + 1


Input
1
4
-1
0
2
-2
1
3
3
1
W
1
2
0
-1
w


Output
9
0
1
3
-5
-3
6
W − w + 1


Input
1
4
-1
0
2
-2
1
3
3
1
W


Kernel
1
2
0
-1
w


Output
9
0
1
3
-5
-3
6
W − w + 1

Formally, in 1d, given

$$x = (x_1, \ldots, x_W)$$

and a 'convolution kernel' (or 'filter') of width $w$

$$u = (u_1, \ldots, u_w)$$

the convolution $x \otimes u$ is a vector of size $W - w + 1$ with

$$\begin{aligned}(x \otimes u)_i &= \sum_{j=1}^{w} x_{i-1+j} u_j \\ &= (x_i, \ldots, x_{i+w-1}) \cdot u\end{aligned}$$

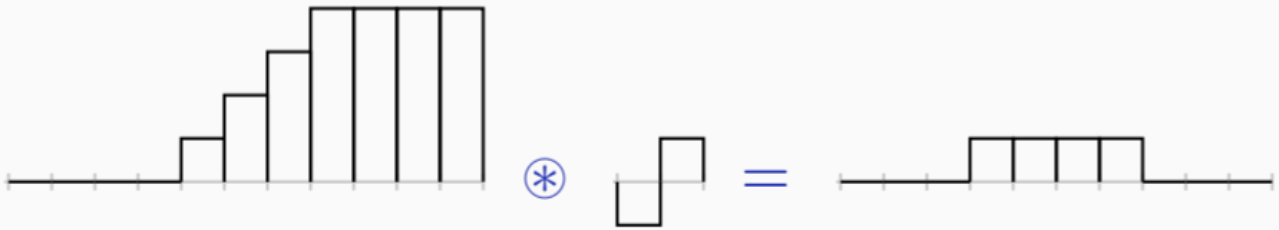
for example

$$(1, 2, 3, 4) \otimes (3, 2) = (3 + 4, 6 + 6, 9 + 8) = (7, 12, 17)$$

Convolutions can implement, e.g.,

- differential operators

$$(0,0,0,0,1,2,3,4,4,4,4) \otimes (-1,1) = (0,0,0,1,1,1,1,0,0,0)$$



- or simple template matchers

Both of these examples are indeed *invariant by translation*

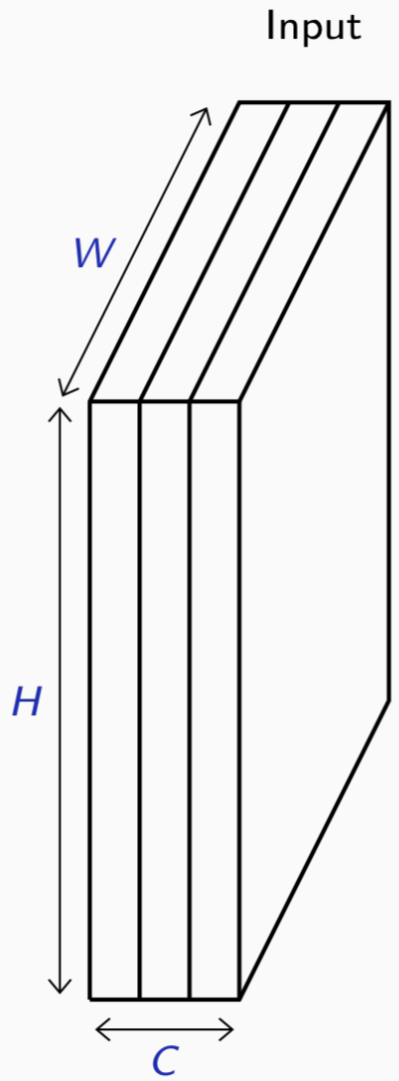
Convolutions generalize to a multi-dimensional input

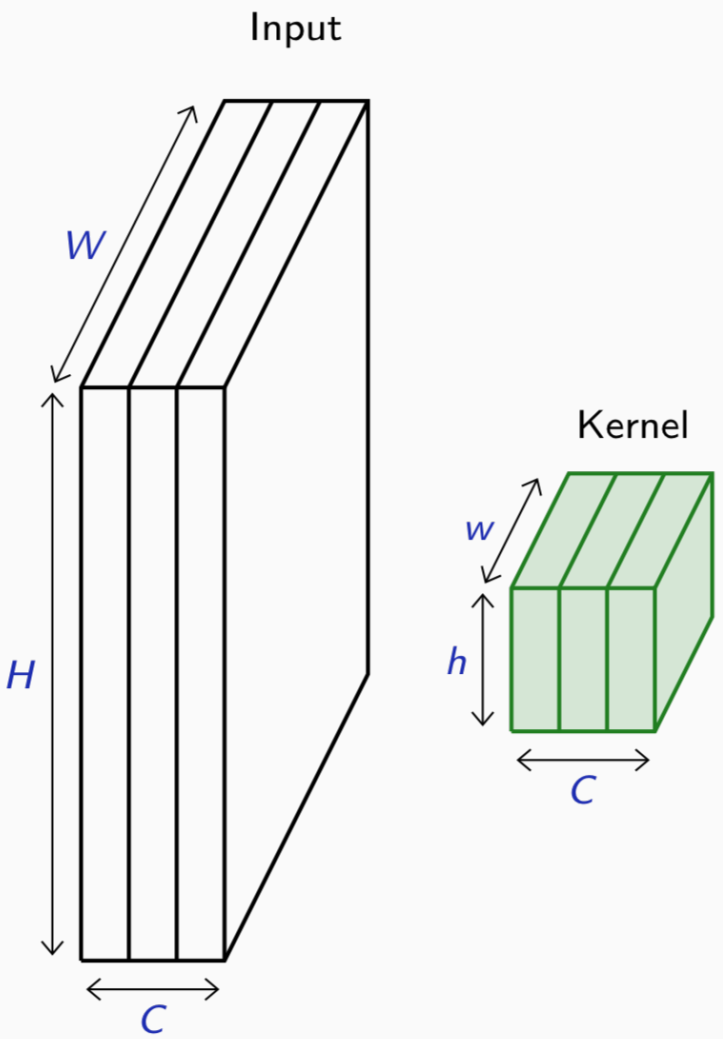
*Convolutional neural networks* process a 3d tensor as input (i.e., a multi-channel 2d signal) to output a 2d tensor. The kernel is not swiped across channels, just across rows and columns.

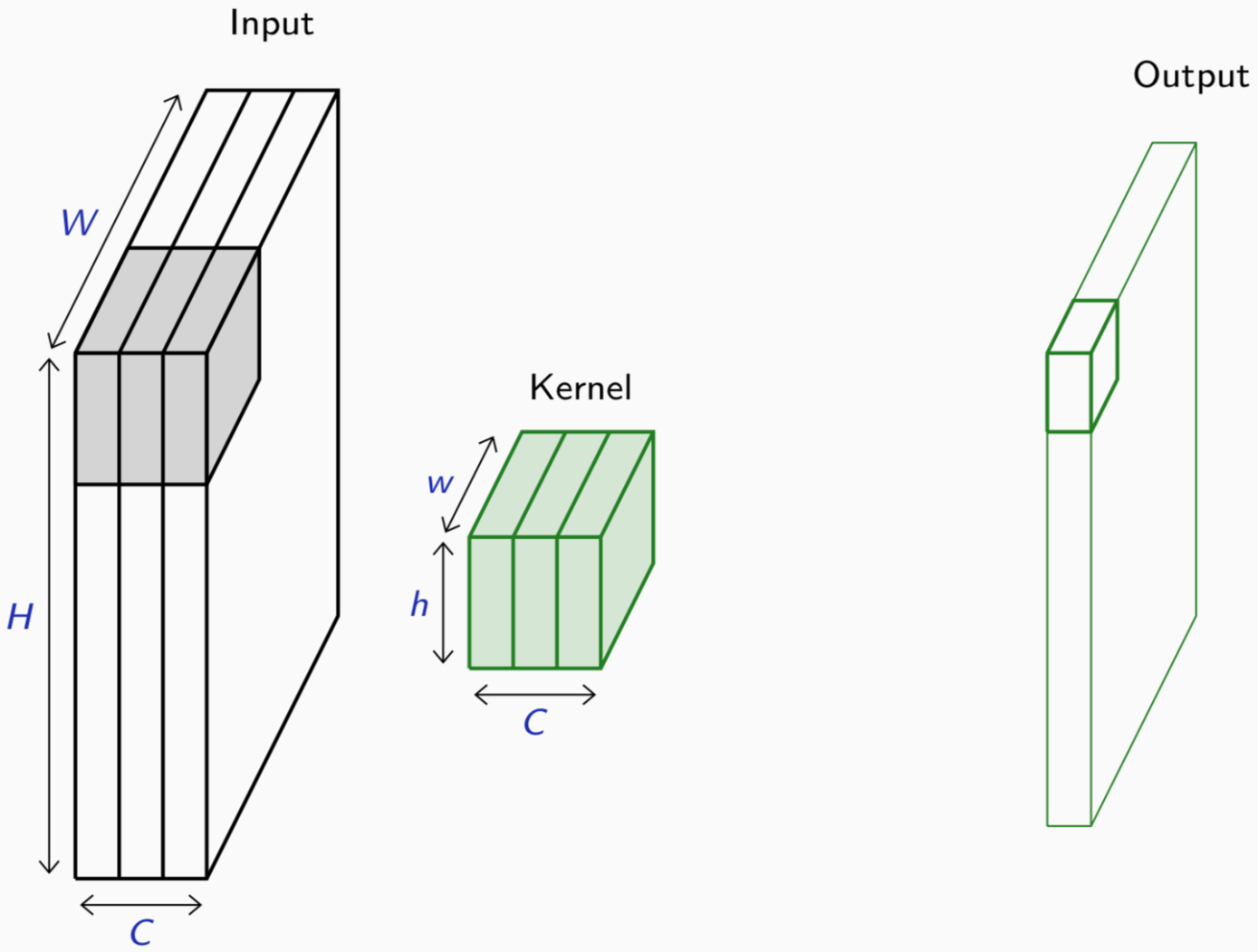
In this case, if the input tensor is of size $C \times H \times W$, and the kernel is $C \times h \times w$, the output is $(H - h + 1) \times (W - w + 1)$.[1]

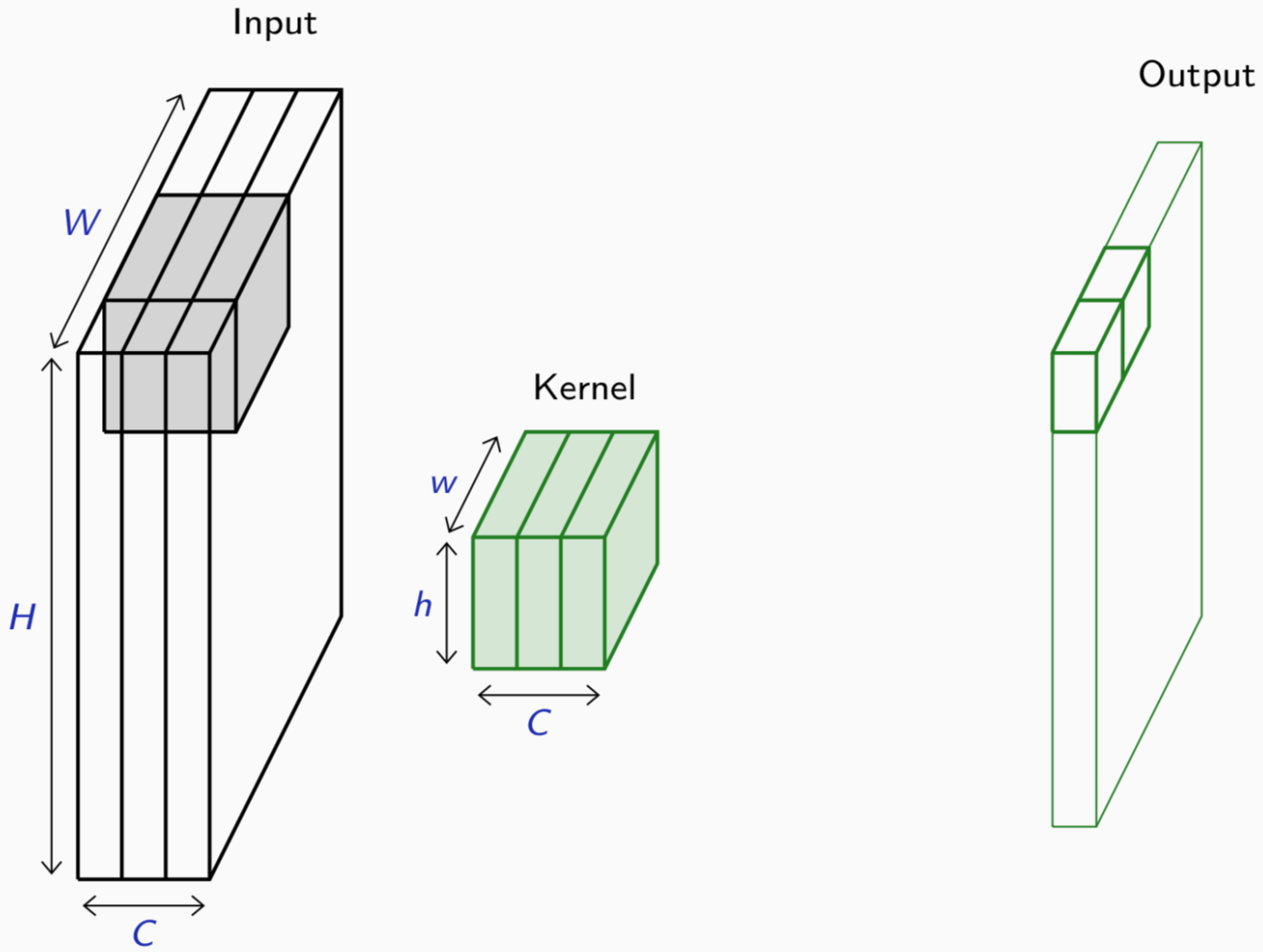
In a standard convolution layer, $D$ such convolutions are combined to generate a $D \times (H - h + 1) \times (W - w + 1)$ output.

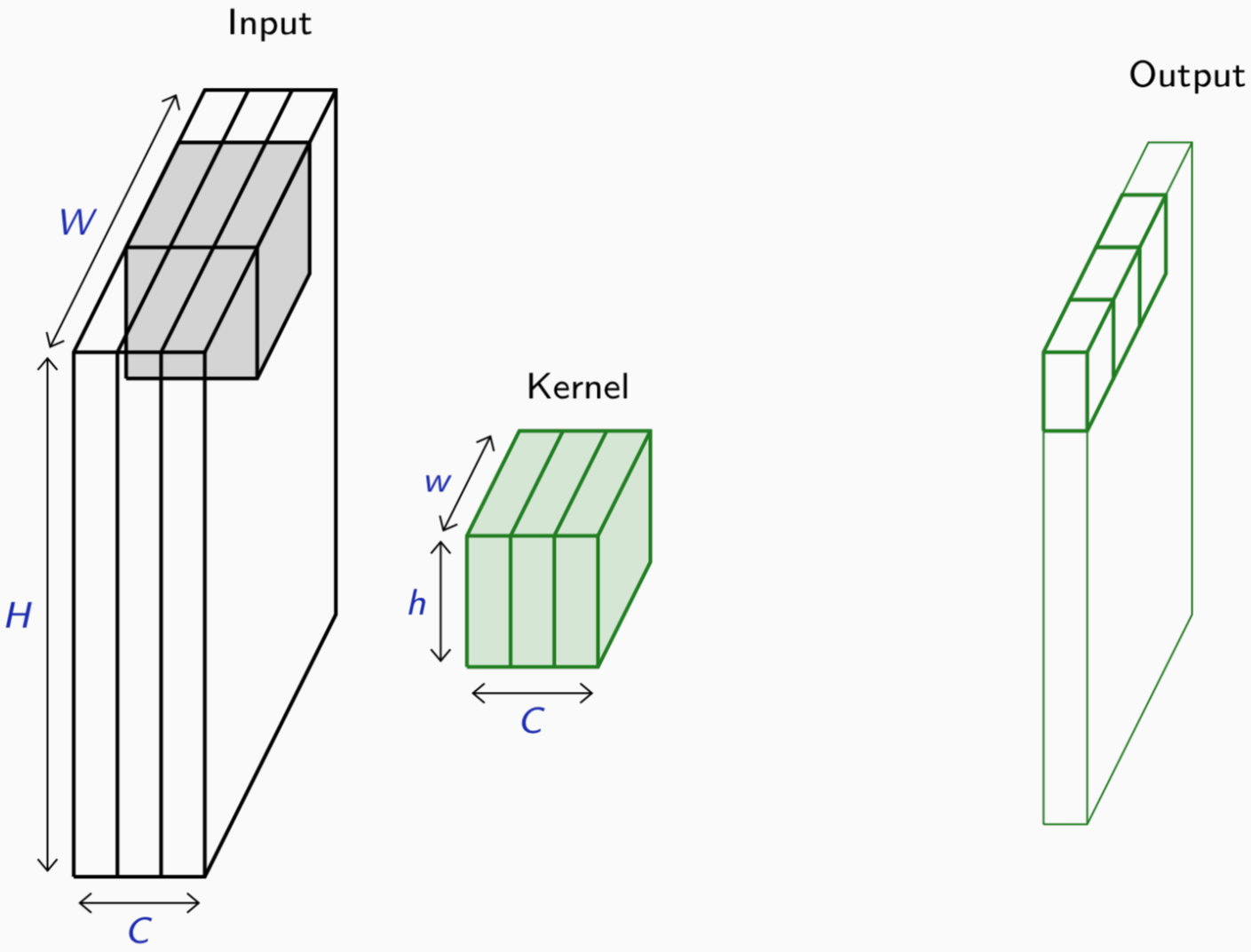
[1] We say '2d signal' even though it has $C$ channels, since it is a feature vector indexed by a 2d location without structure on the feature indexes.

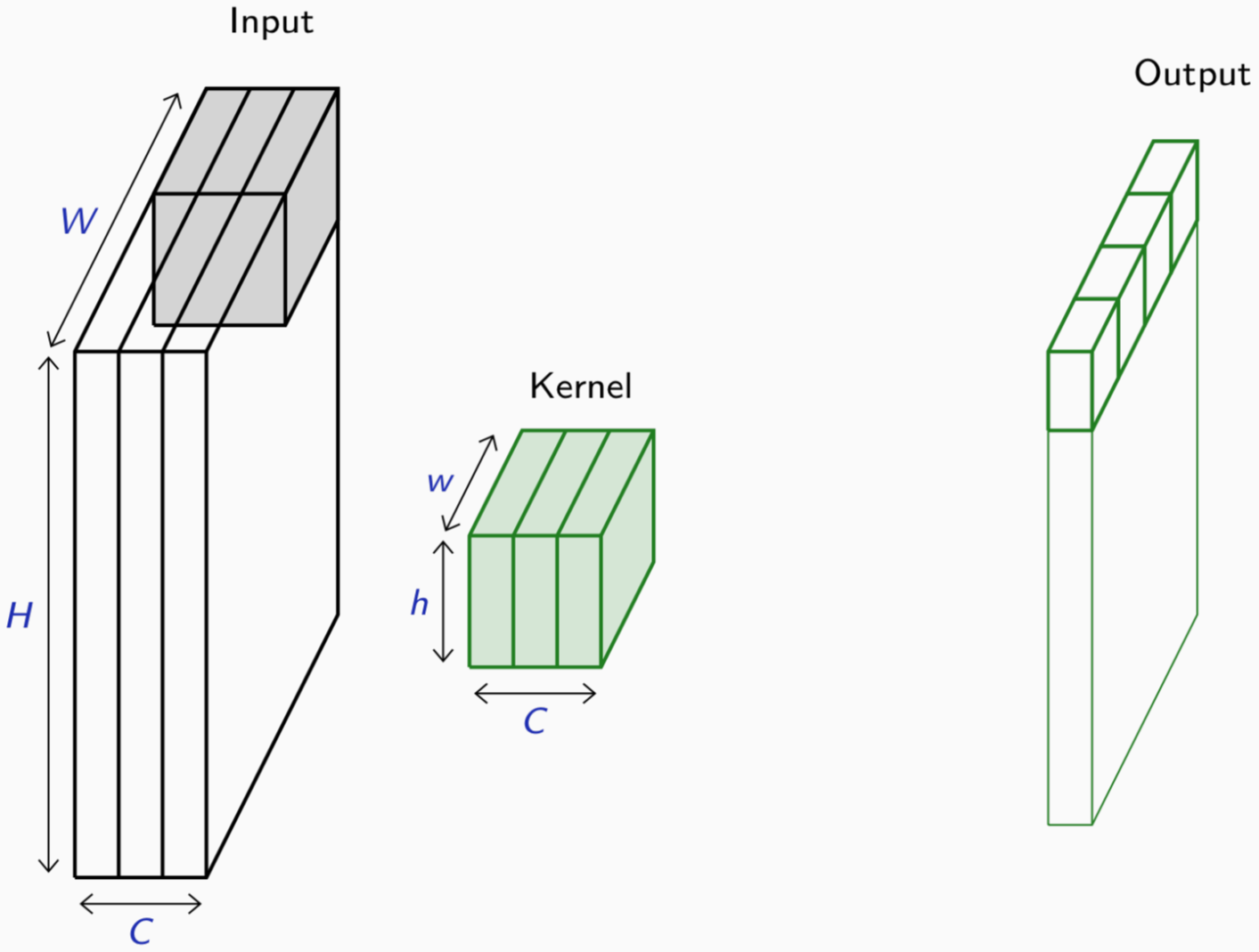

Input
W
H
C

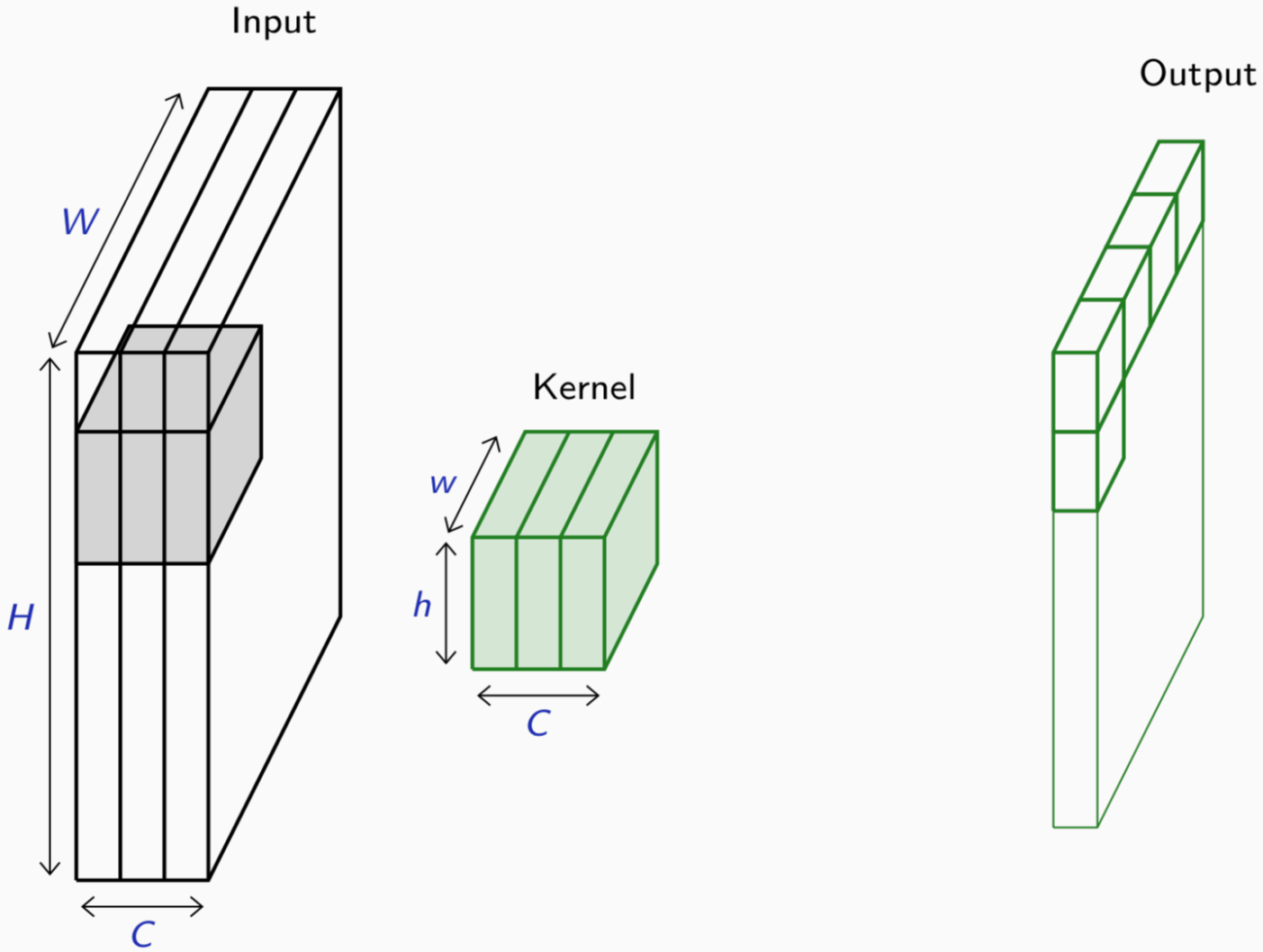

Input
W
H
C
Kernel
w
h
C

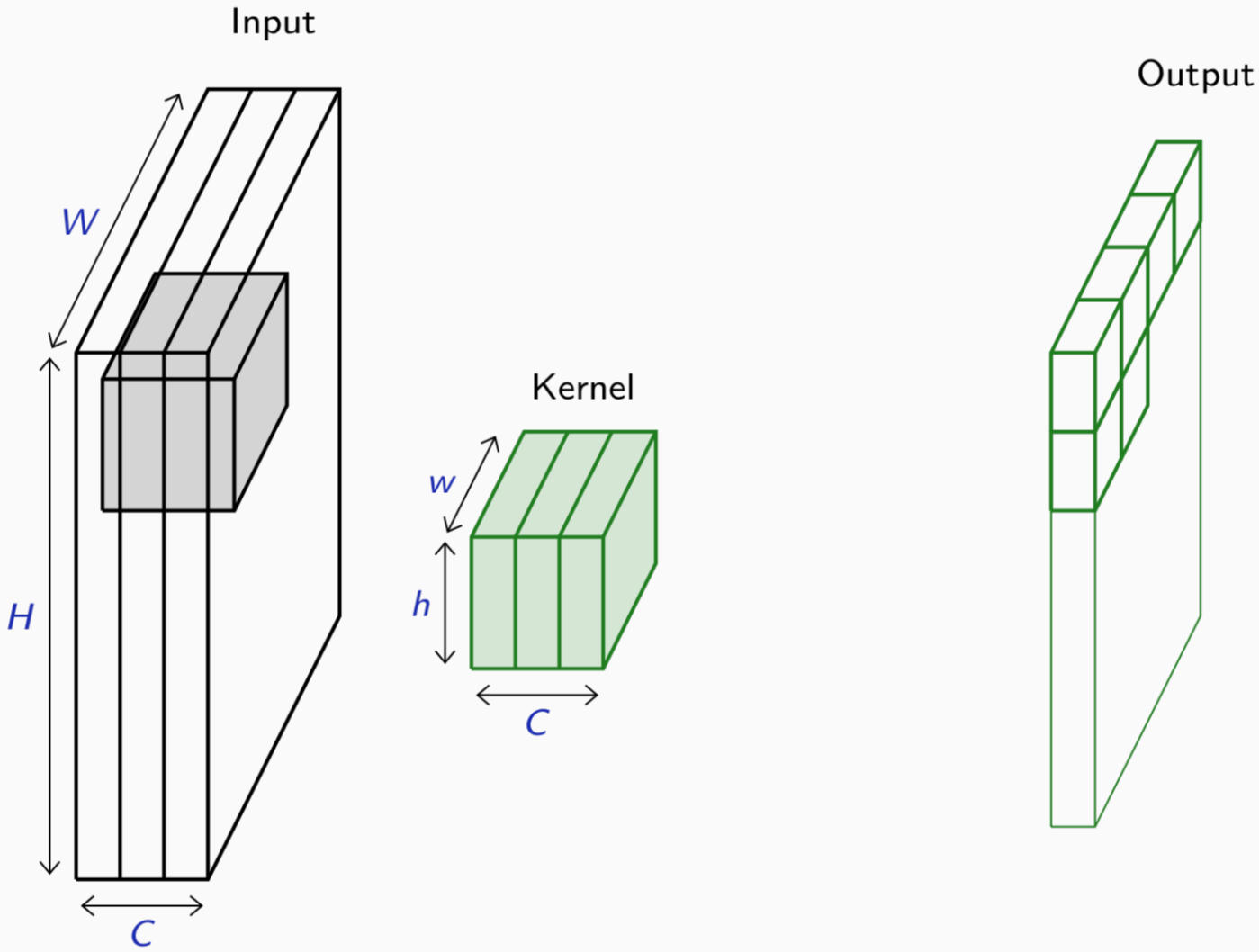

Input
W
H
C
Kernel
w
h
C
Output

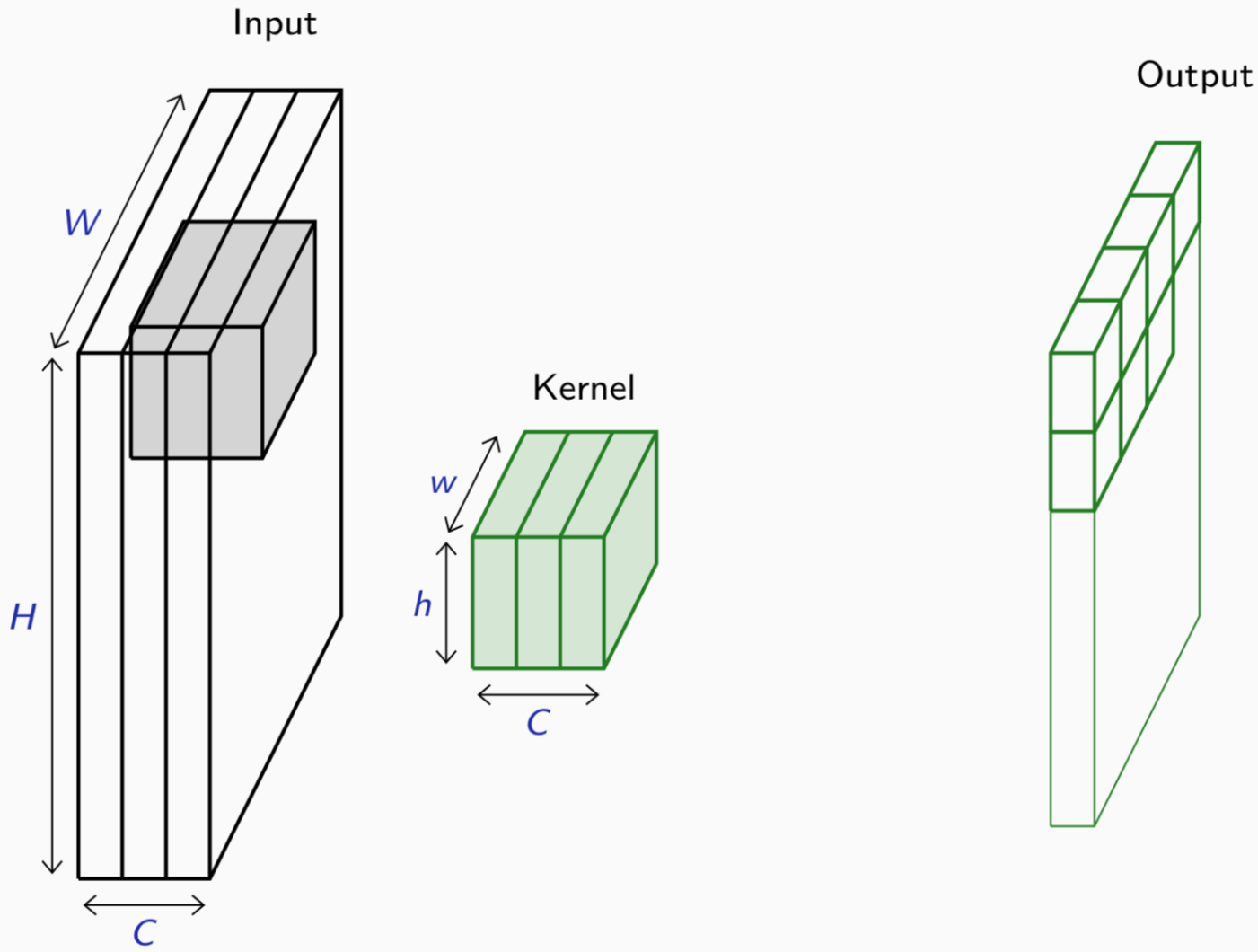

Input
W
H
C
Kernel
w
h
C
Output

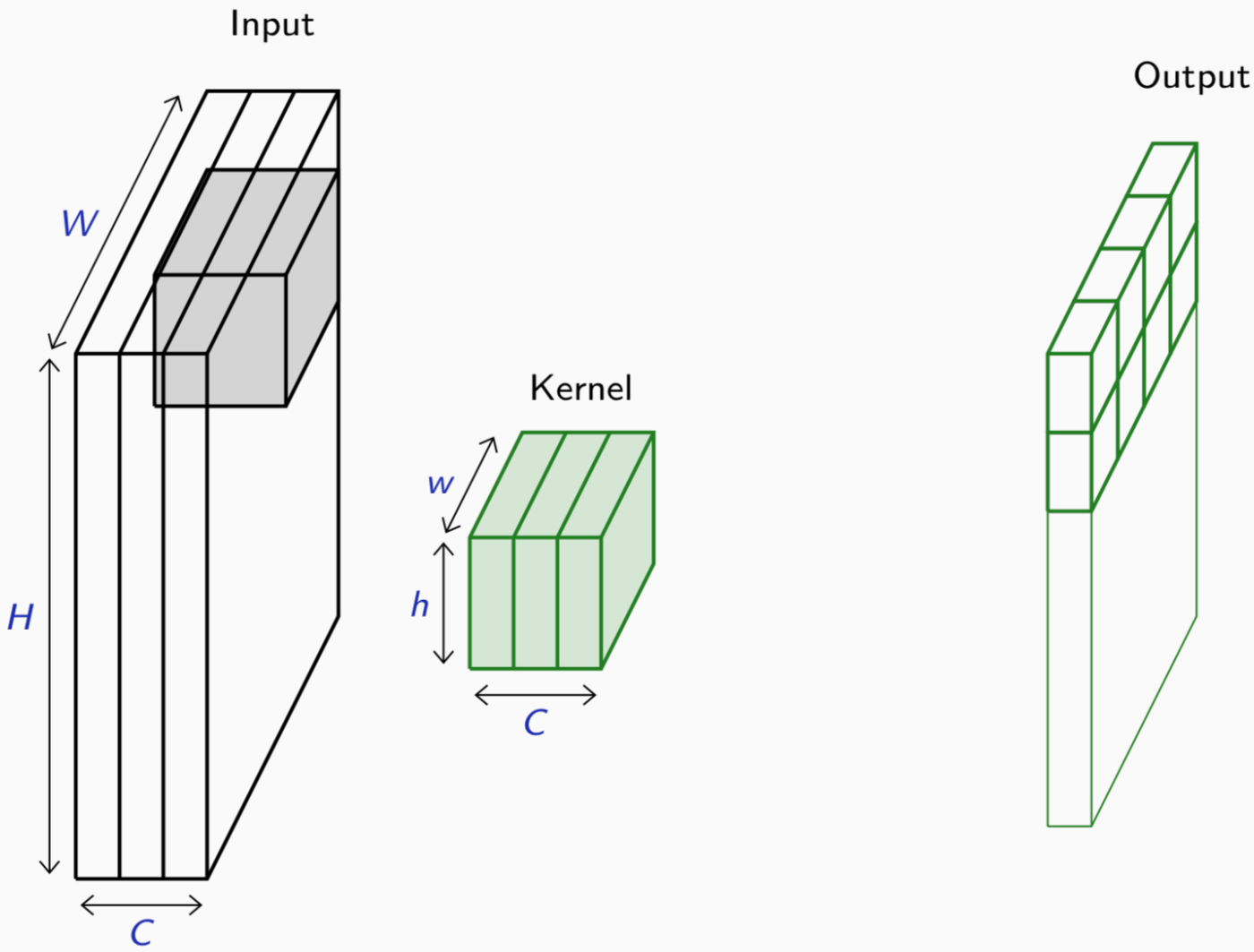

Input
W
H
C
Kernel
w
h
C
Output

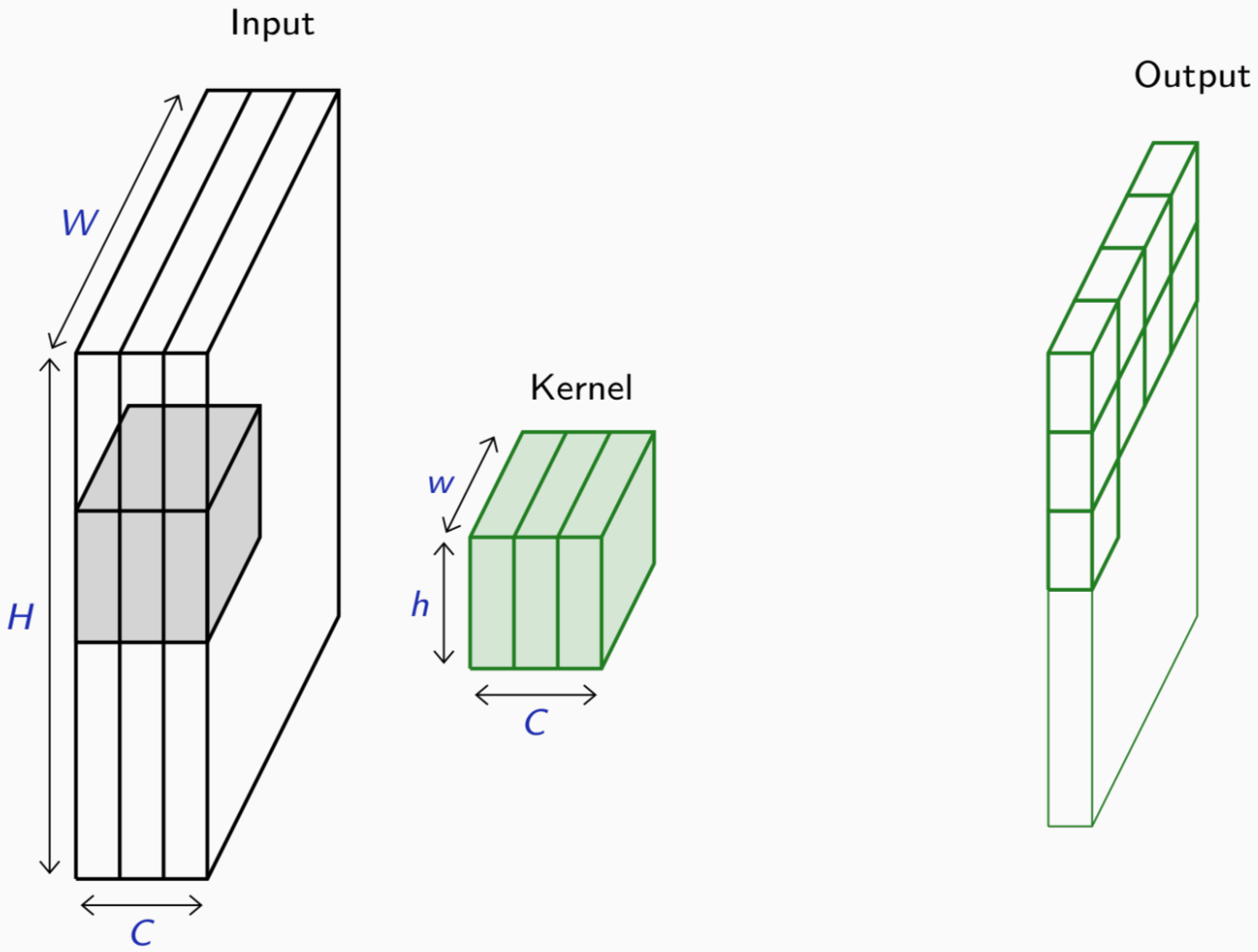

Input
W
H
C
Kernel
w
h
C
Output

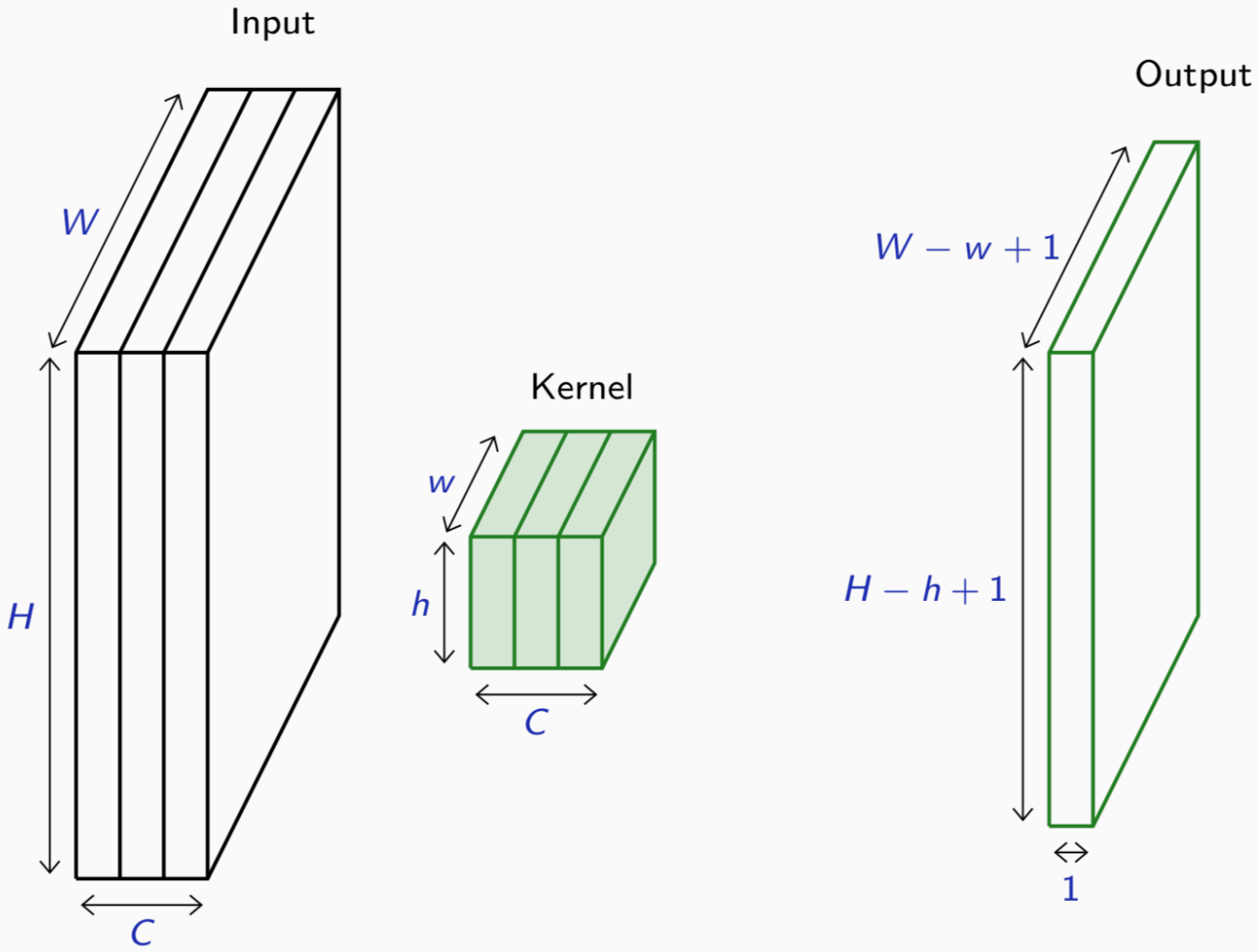

Input
W
H
C
Kernel
w
h
C
Output

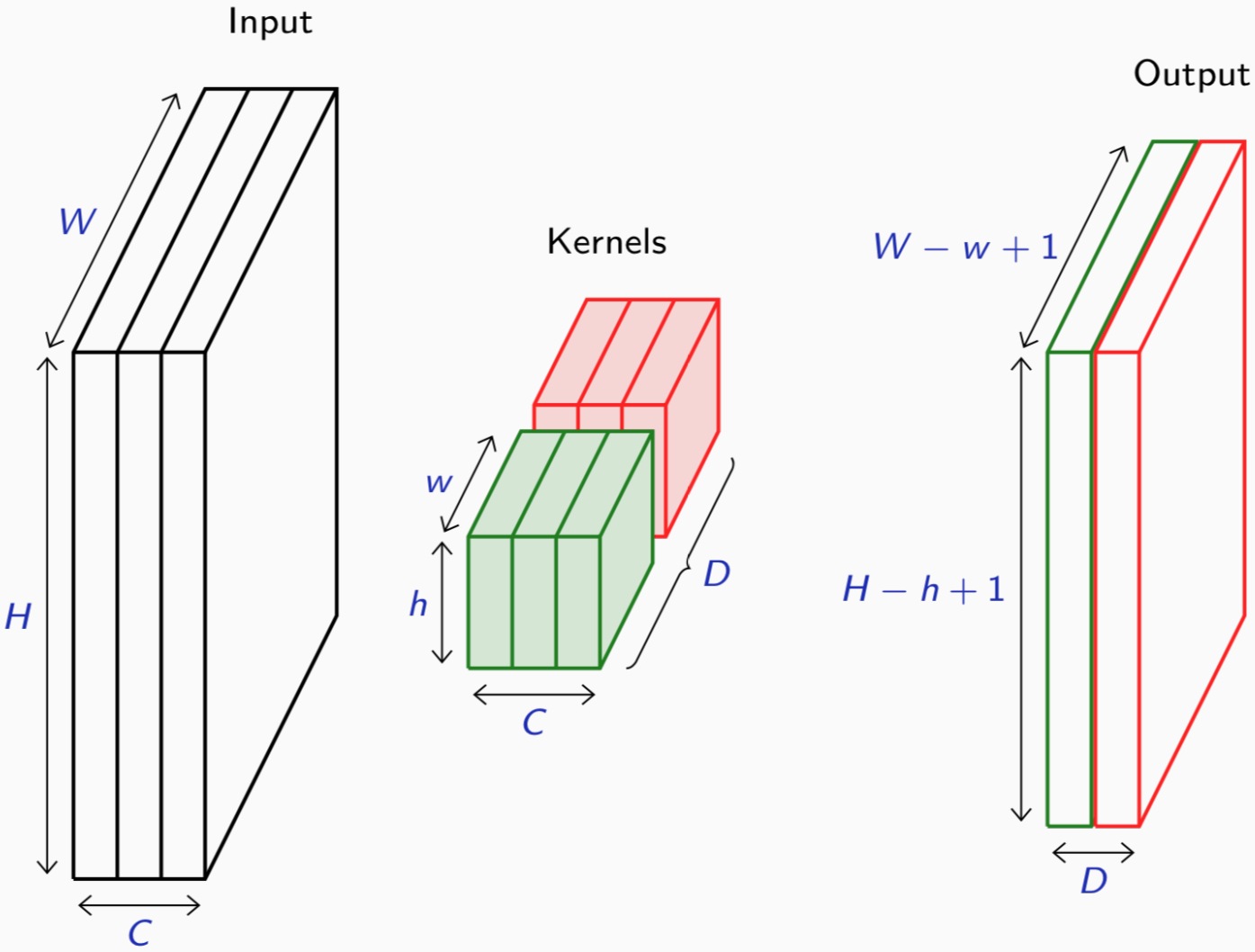

Input
W
H
C
Kernel
w
h
C
Output


Input
W
H
C
Kernel
w
h
C
Output


Input
W
H
C
Kernel
w
h
C
Output


Input
W
H
C
Kernel
w
h
C
Output


Input
W
H
C
Kernel
w
h
C
Output
W − w + 1
H − h + 1
1


Input
W
H
C
Kernels
w
h
D
C
Output
W − w + 1
H − h + 1
D

Note that a convolution preserves the signal structure: a 1d signal is converted into a 1d signal, a 2d signal into a 2d

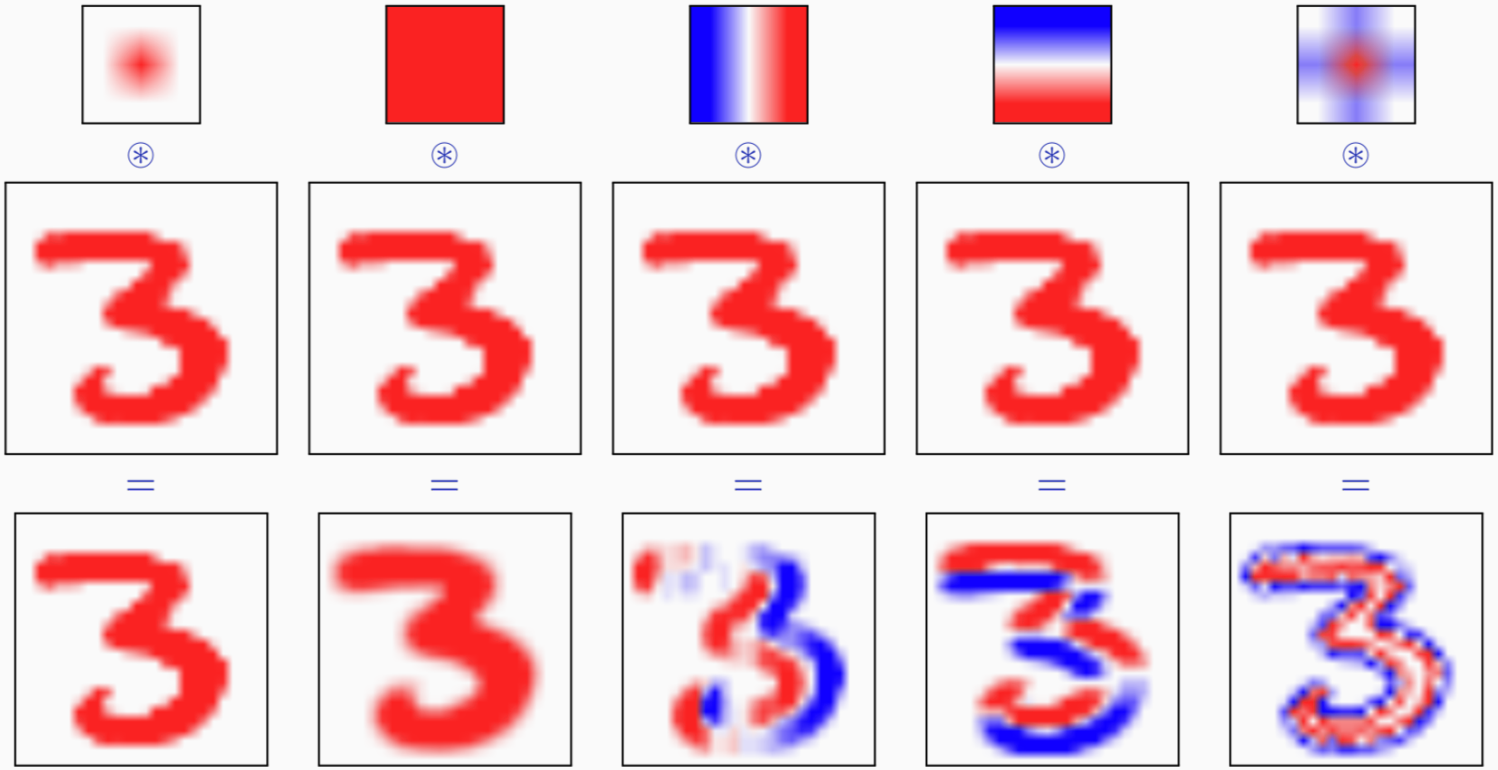
Neighboring parts of the input signal influence neighboring parts of the output signal

A 3d convolution can be used if the channel index has some metric meaning, such as time for a series of grayscale video frames

We usually refer to one of the channels generated by a convolution layer as an **activation map**

The sub-area of an input map that influences a component of the output is called the **receptive field** of the latter.

In the context of convolutional networks, a standard linear layer is called a **fully connected layer** since every input influences every output.

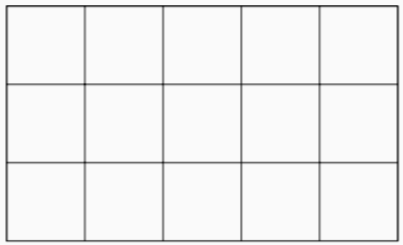


# Padding and stride

Convolutions have two standard parameters:

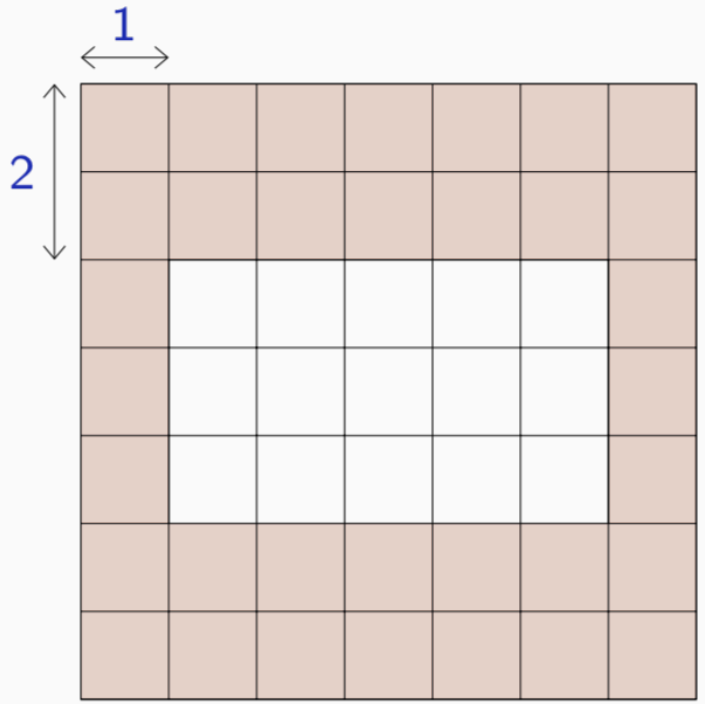
- the **padding** specifies the size of a zeroed frame added around the input
- the **stride** specifies a step size when moving the kernel across the signal

A convolution with an input of $C \times 3 \times 5$

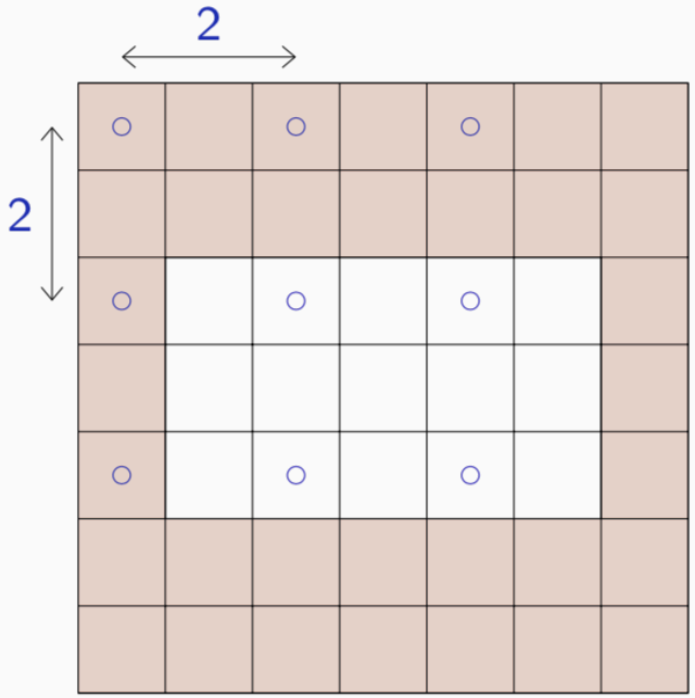


Input

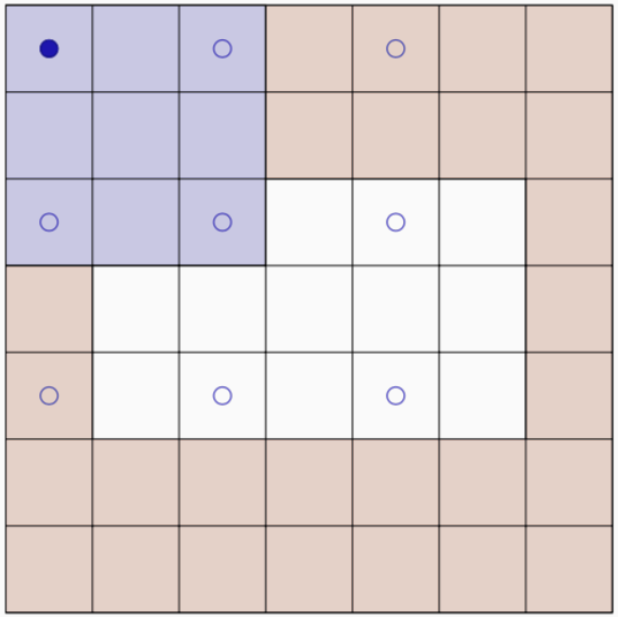
A convolution with an input of $C \times 3 \times 5$, a padding of $(2, 1)$



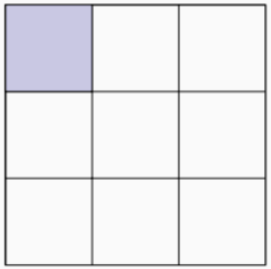
A convolution with an input of $C \times 3 \times 5$, a padding of $(2, 1)$, a stride of $(2, 2)$



A convolution with an input of $C \times 3 \times 5$, a padding of $(2, 1)$, a stride of $(2, 2)$, and a kernel of size $C \times 3 \times 3$

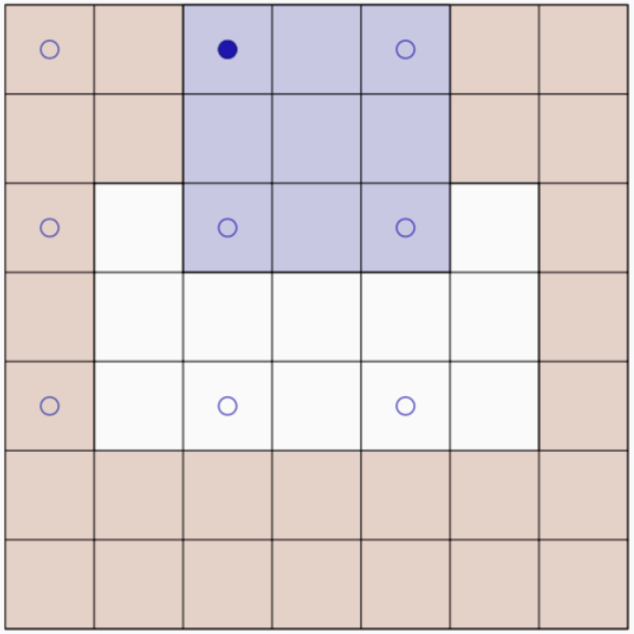


Input

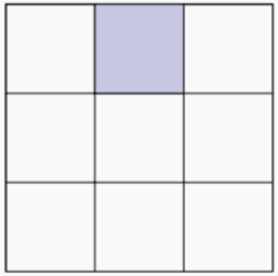


Output

A convolution with an input of $C \times 3 \times 5$, a padding of $(2, 1)$, a stride of $(2, 2)$, and a kernel of size $C \times 3 \times 3$

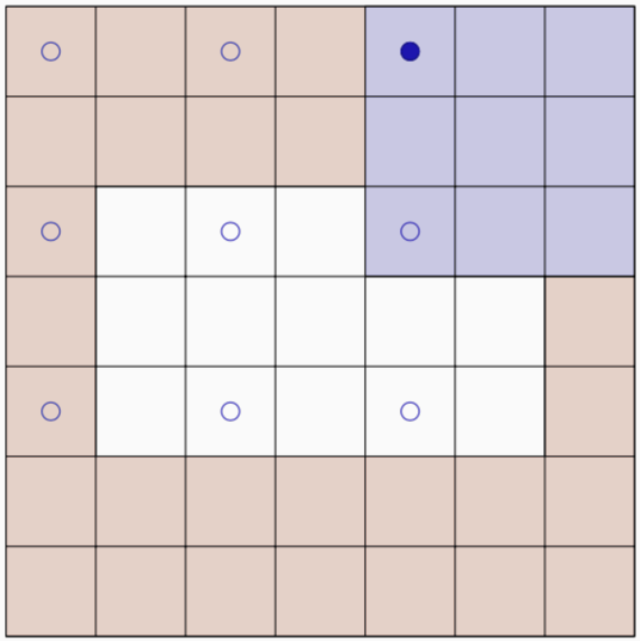


Input

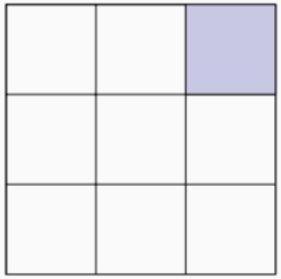


Output

A convolution with an input of $C \times 3 \times 5$, a padding of $(2, 1)$, a stride of $(2, 2)$, and a kernel of size $C \times 3 \times 3$

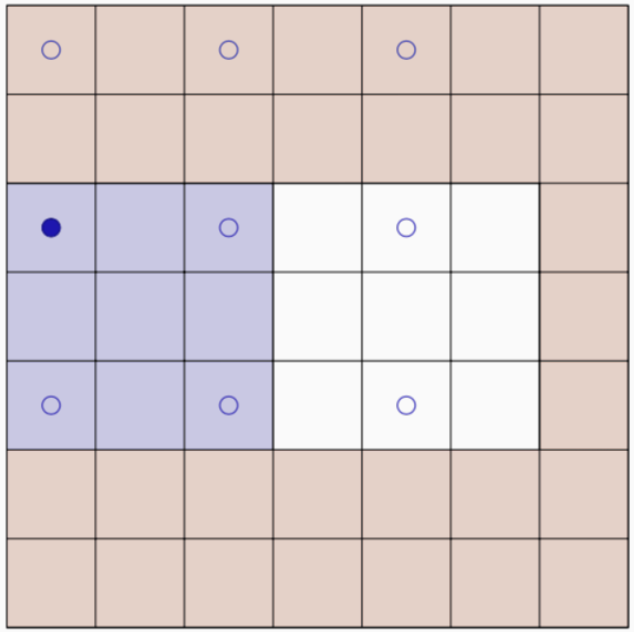


Input

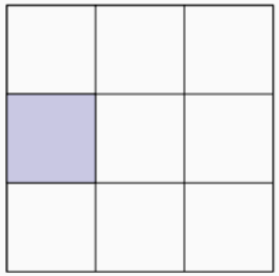


Output

A convolution with an input of $C \times 3 \times 5$, a padding of $(2, 1)$, a stride of $(2, 2)$, and a kernel of size $C \times 3 \times 3$

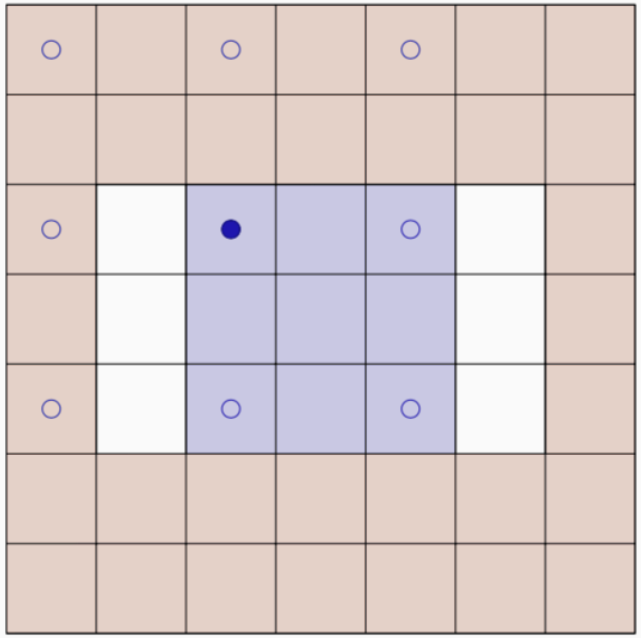


Input

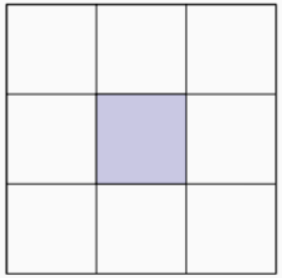


Output

A convolution with an input of $C \times 3 \times 5$, a padding of $(2, 1)$, a stride of $(2, 2)$, and a kernel of size $C \times 3 \times 3$

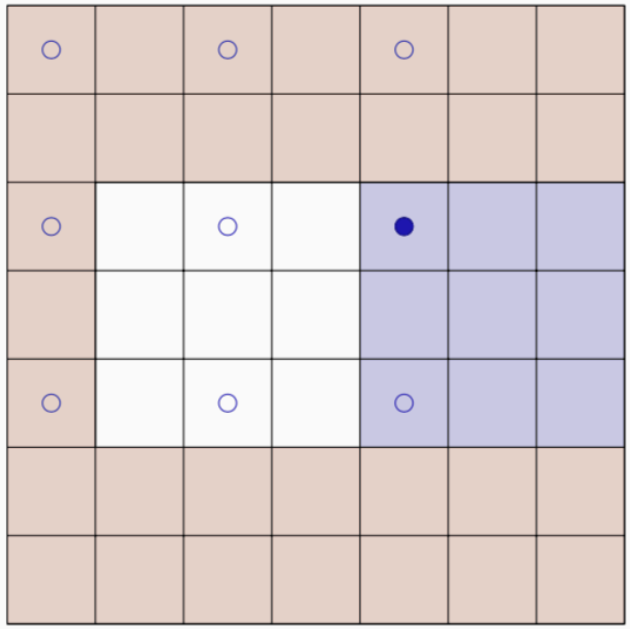


Input

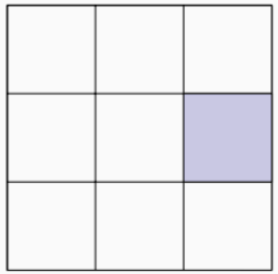


Output

A convolution with an input of $C \times 3 \times 5$, a padding of $(2, 1)$, a stride of $(2, 2)$, and a kernel of size $C \times 3 \times 3$

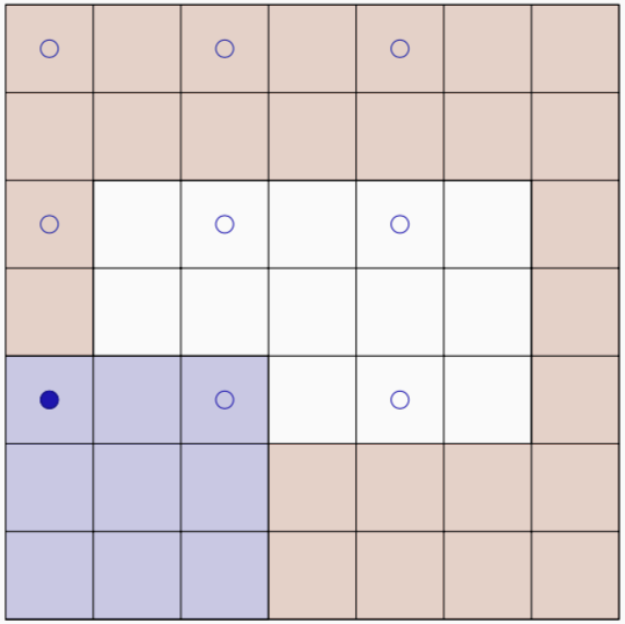


Input

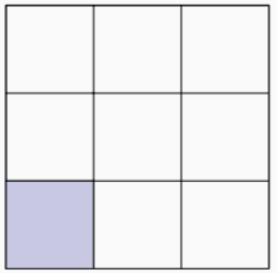


Output

A convolution with an input of $C \times 3 \times 5$, a padding of $(2, 1)$, a stride of $(2, 2)$, and a kernel of size $C \times 3 \times 3$

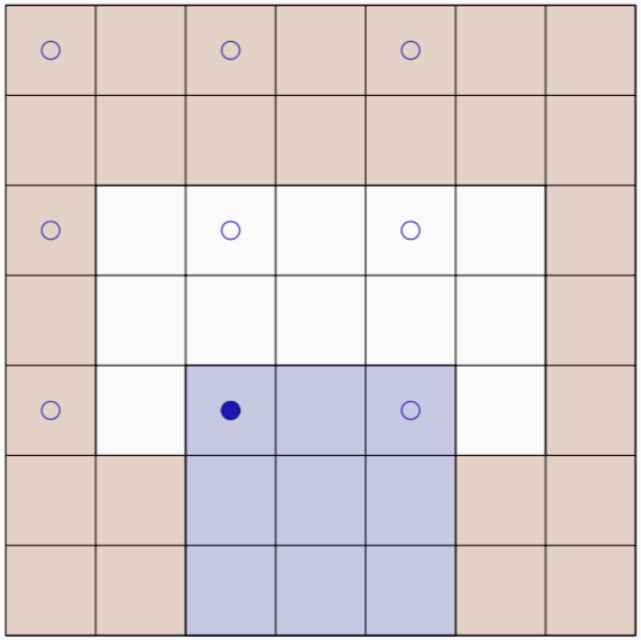


Input

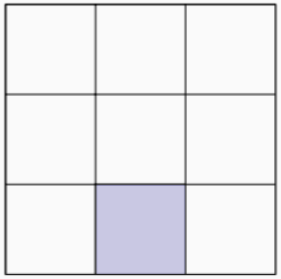


Output

A convolution with an input of $C \times 3 \times 5$, a padding of $(2, 1)$, a stride of $(2, 2)$, and a kernel of size $C \times 3 \times 3$

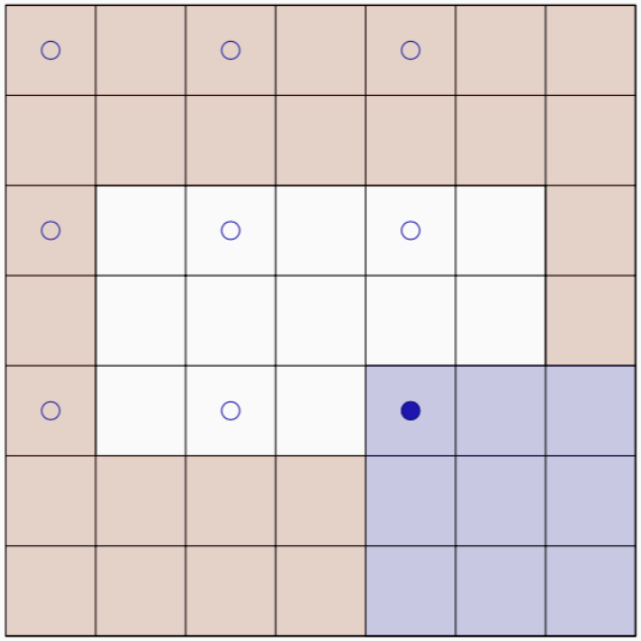


Input

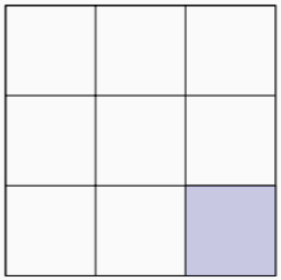


Output

A convolution with an input of $C \times 3 \times 5$, a padding of $(2, 1)$, a stride of $(2, 2)$, and a kernel of size $C \times 3 \times 3$

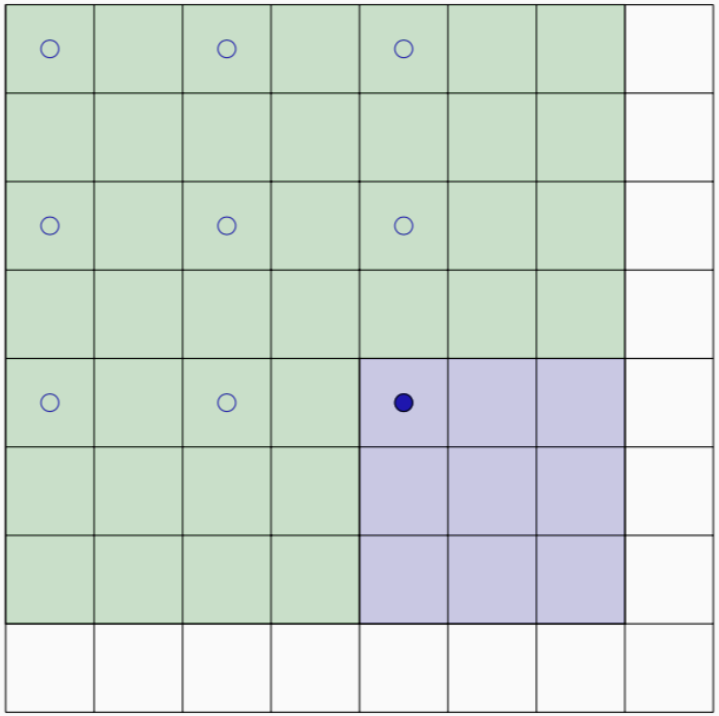


Input



Output

A convolution with a stride $> 1$ may not cover the input map completely and ignore some of the input values

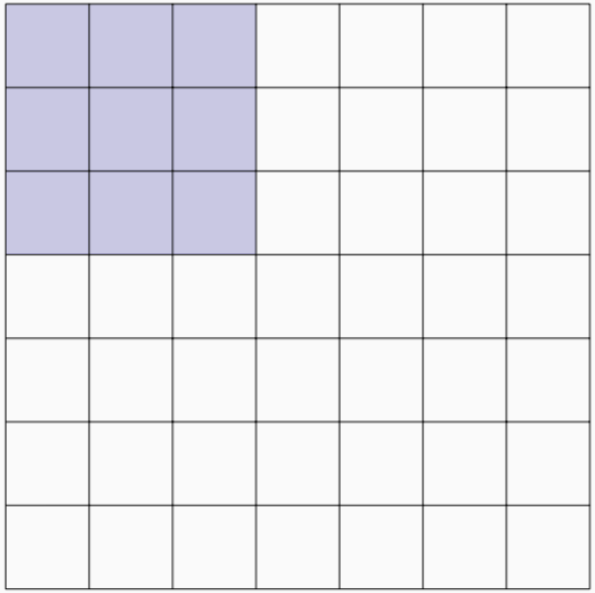


# Dilated convolutions

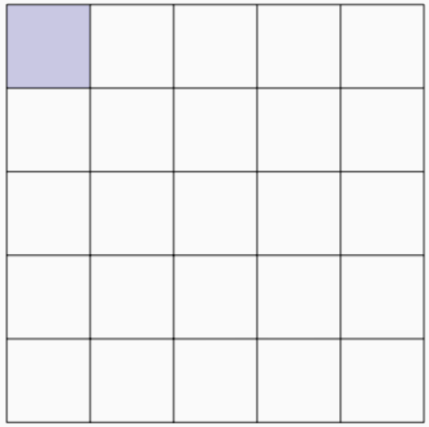
A third paramerter of convolution operations is the **dilation**, which modulates the expansion of the filter support (Yu and Koltun, 2015).

It is $1$ for standard convolutions, but can be greater, in which case the resulting operation can be envisioned as a convolution with a regularly sparsified filter.

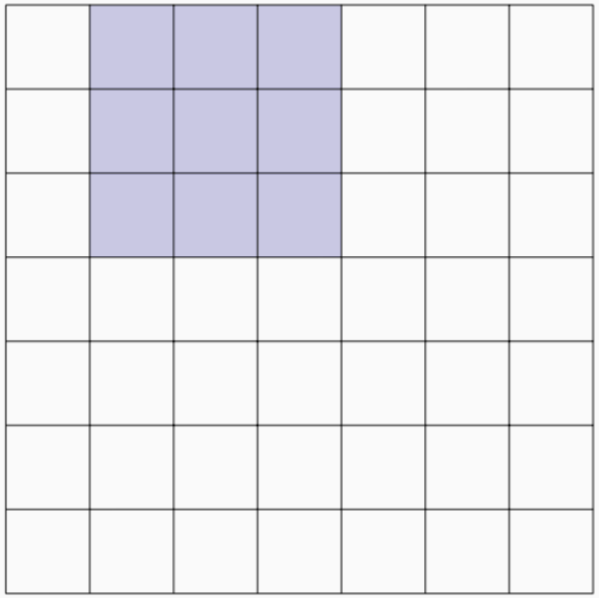
This notion comes from signal processing, where it is referred to as *algorithme à trous*, which is why it is sometimes referred to as *convolution à trous*.



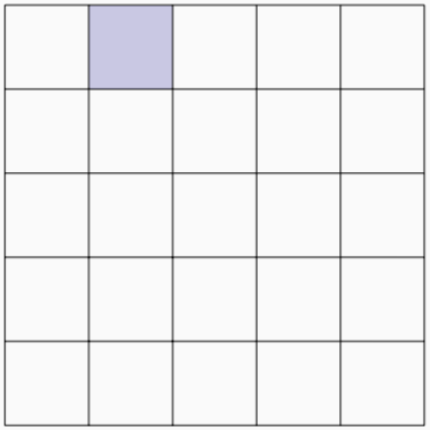
Input



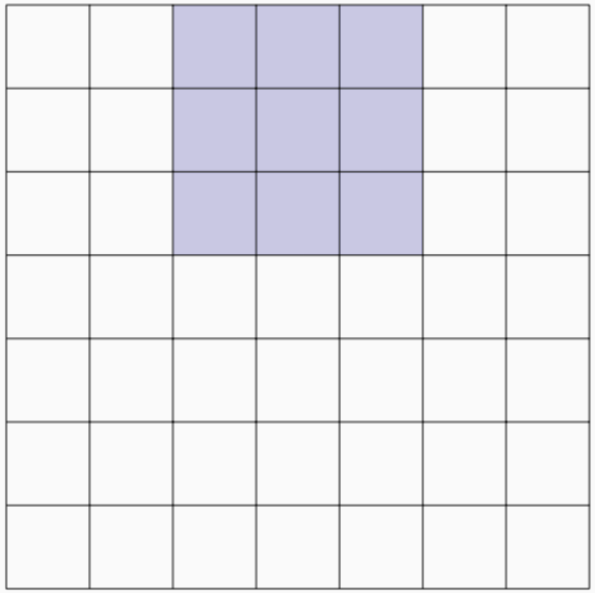
Output



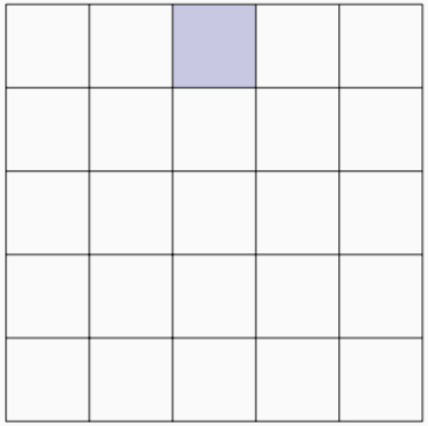
Input



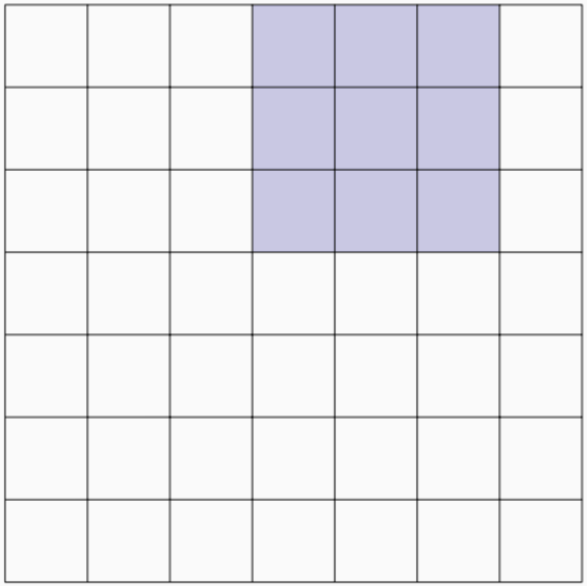
Output



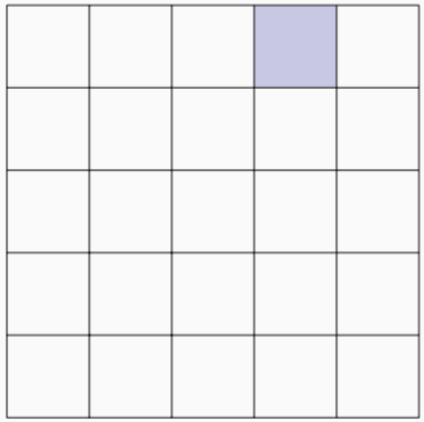
Input



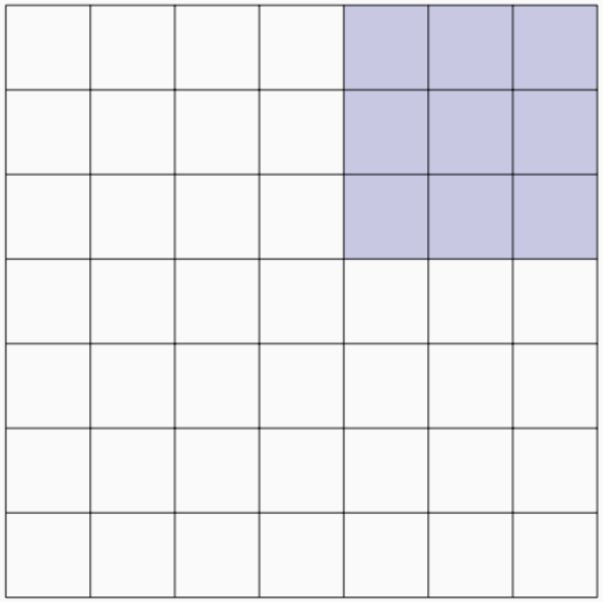
Output



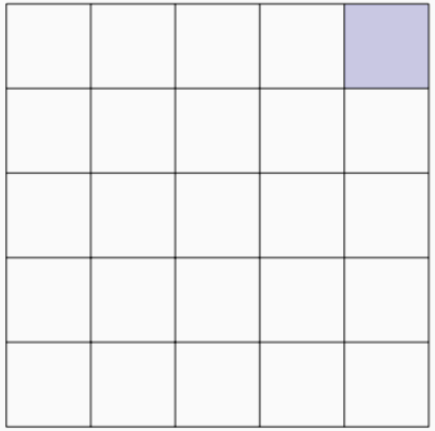
Input



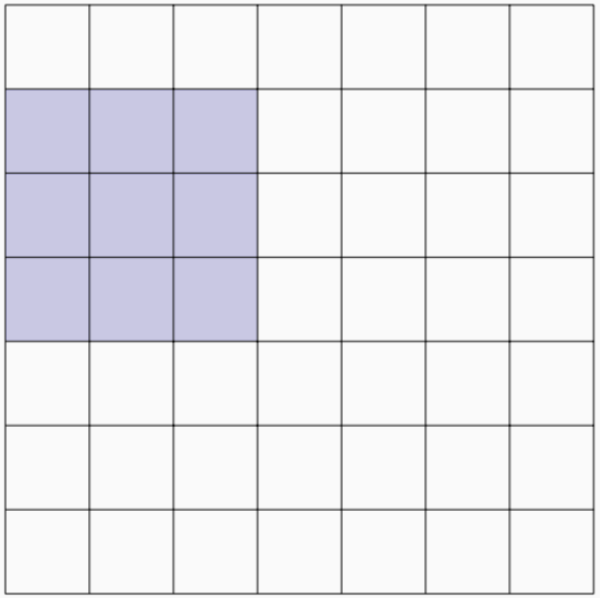
Output



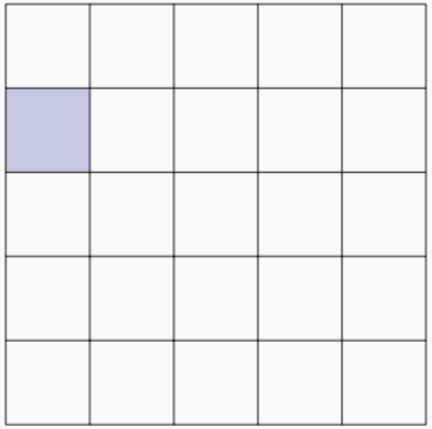
Input



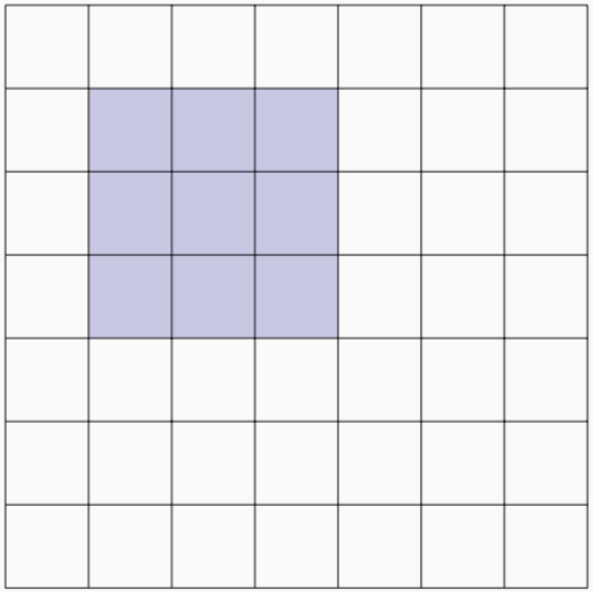
Output



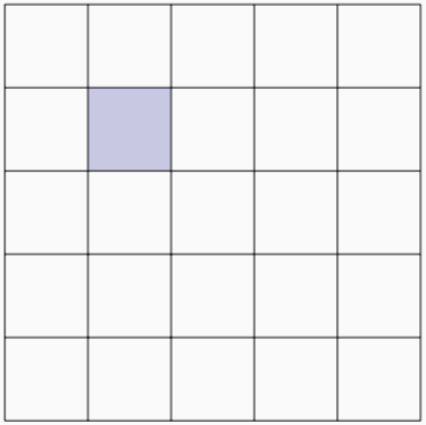
Input



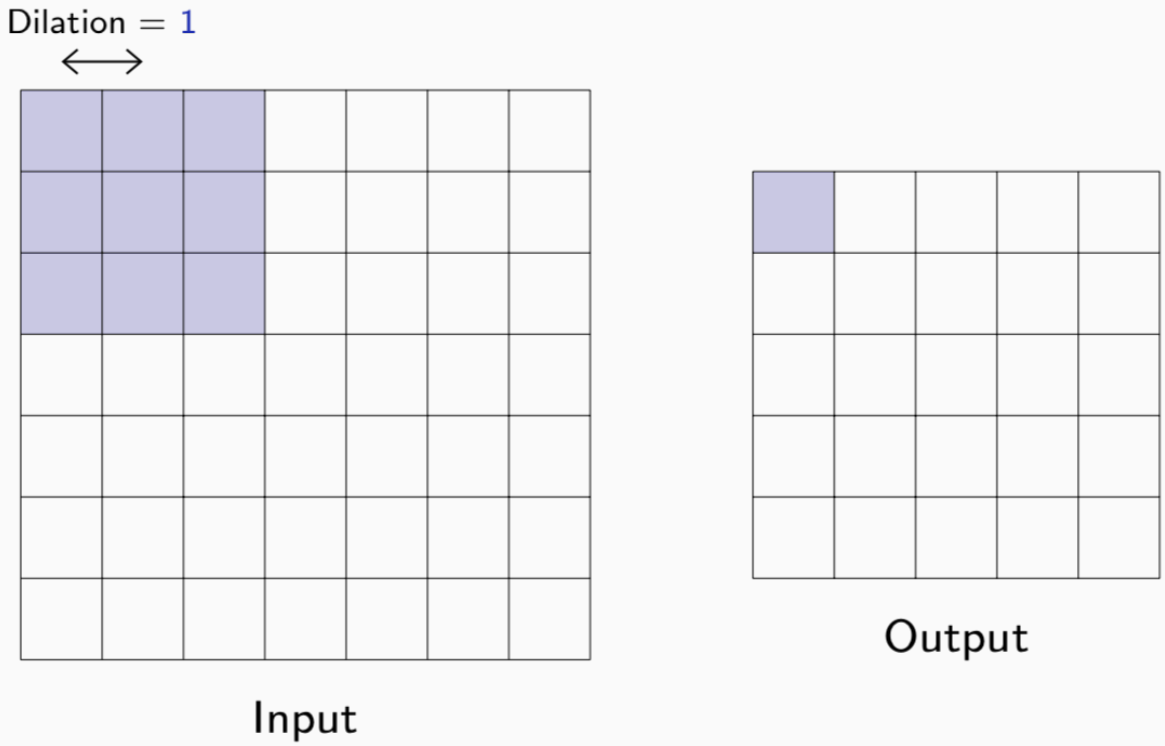
Output

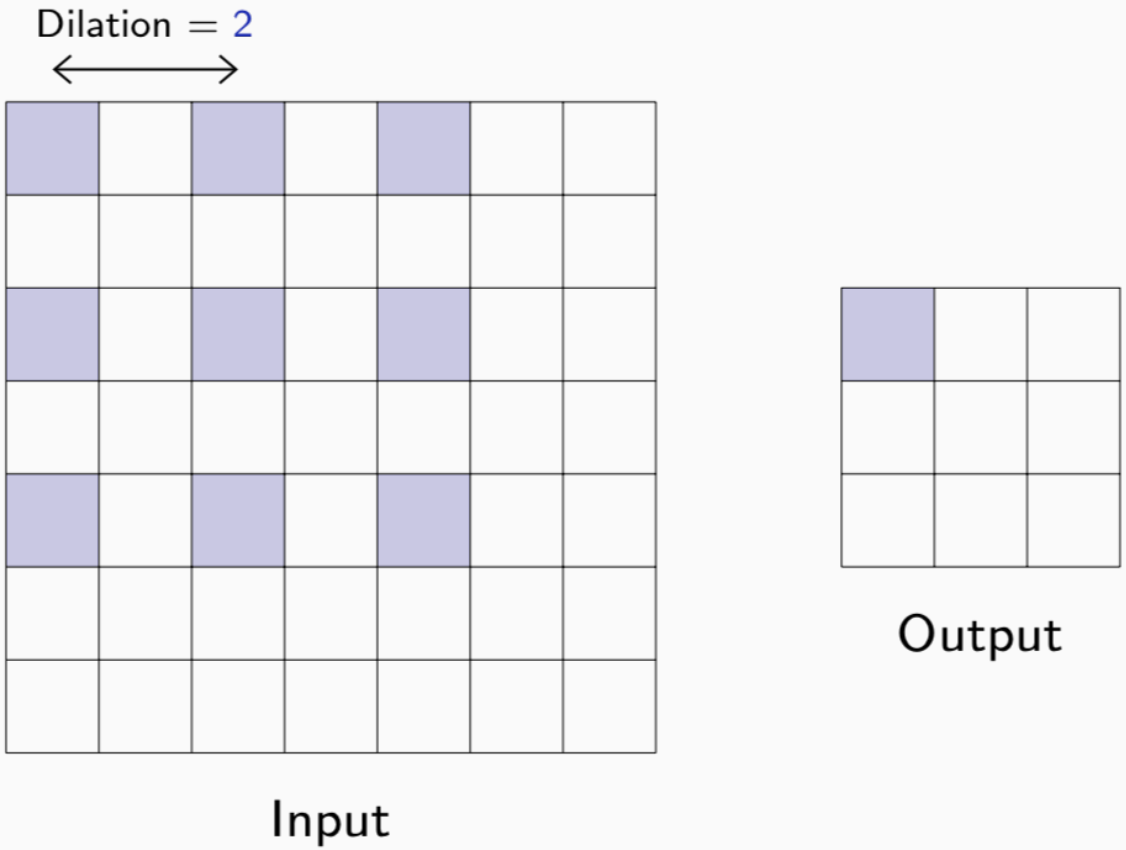


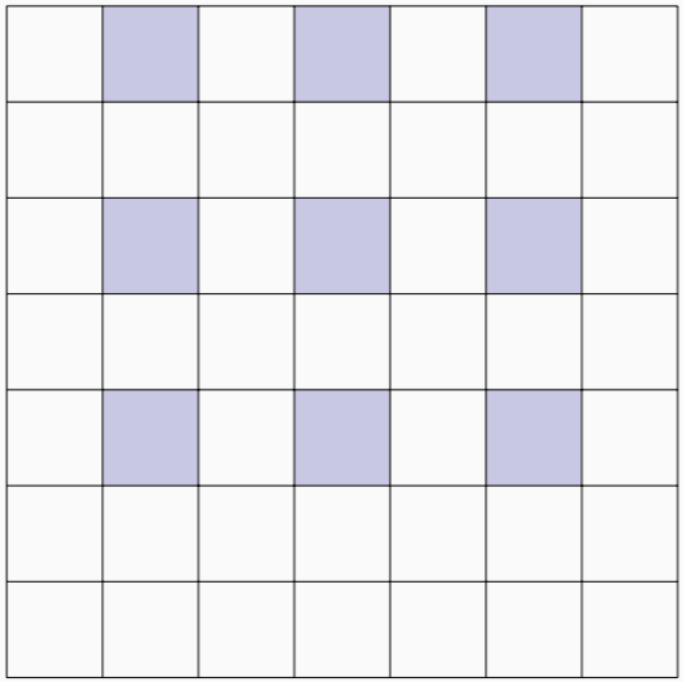
Input



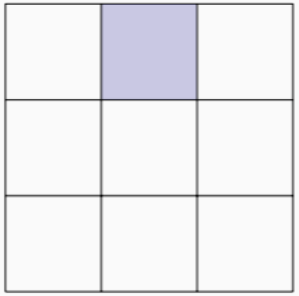
Output


Dilation = 1
Input
Output

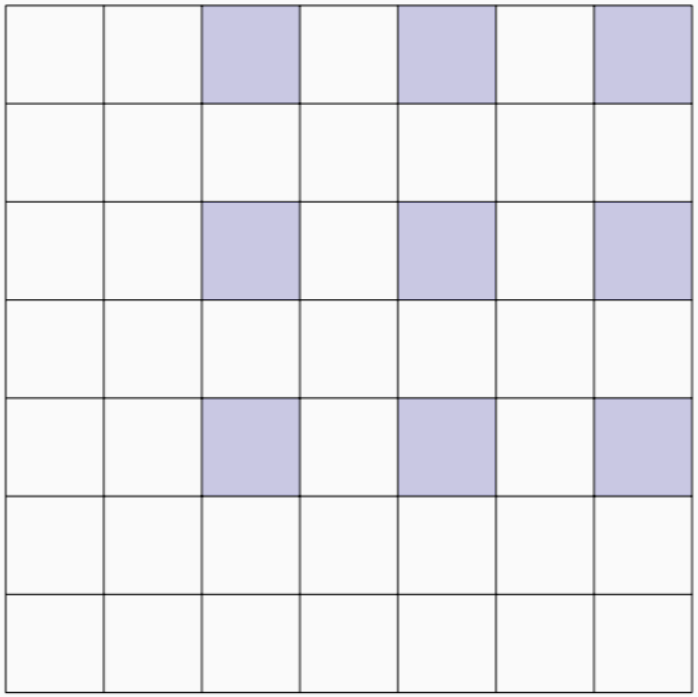

Dilation = 2
Input
Output



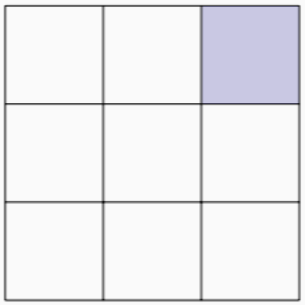
Input



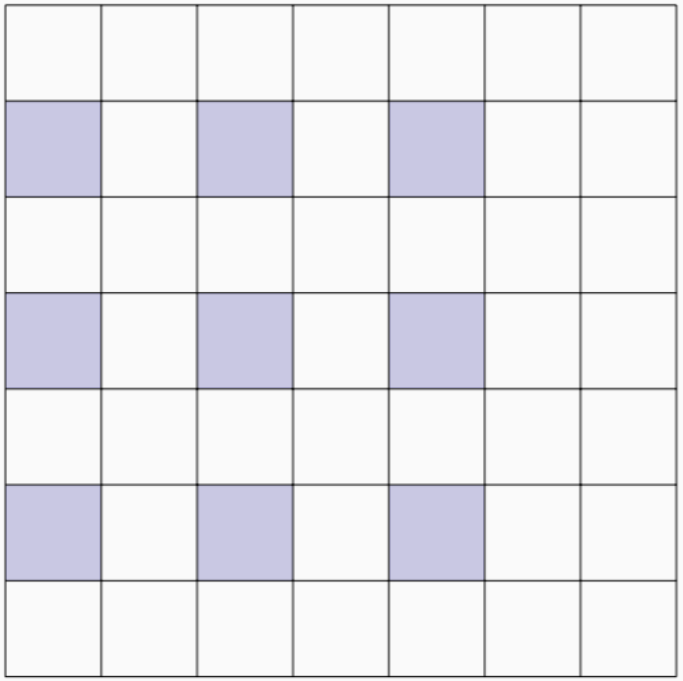
Output



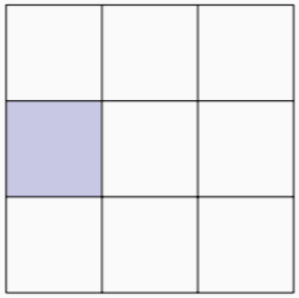
Input



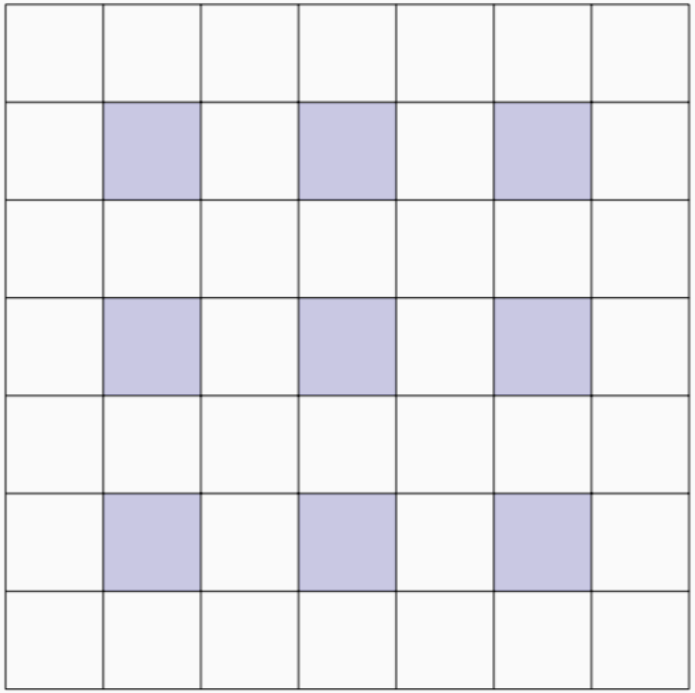
Output



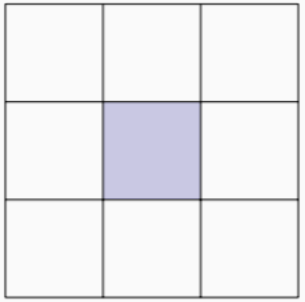
Input



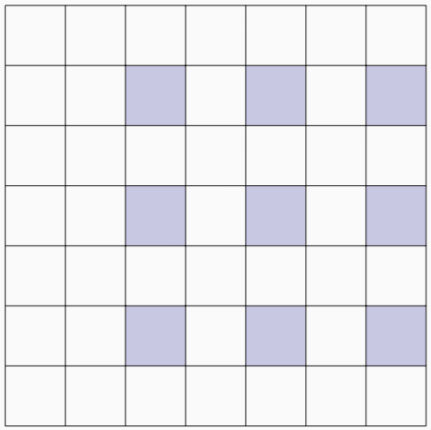
Output



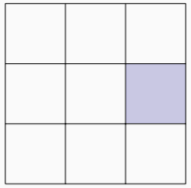
Input



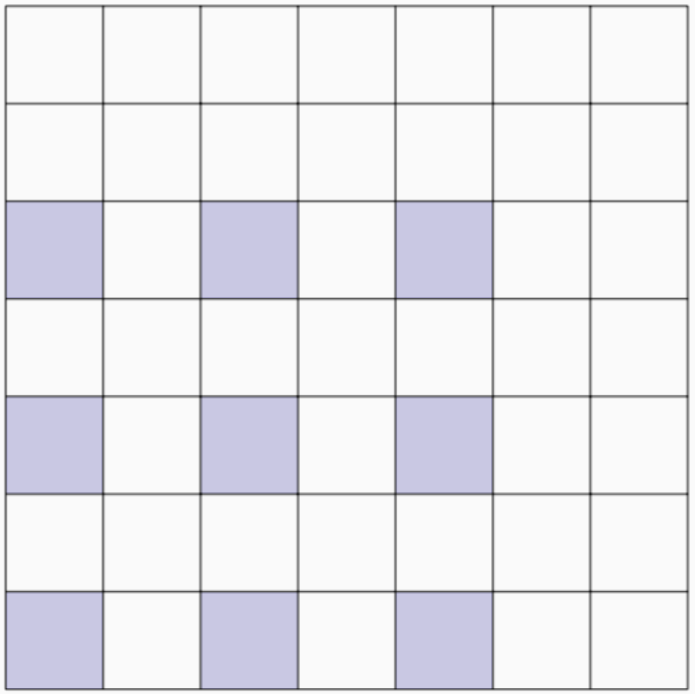
Output



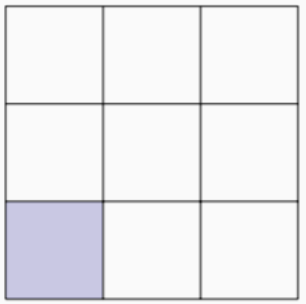
Input



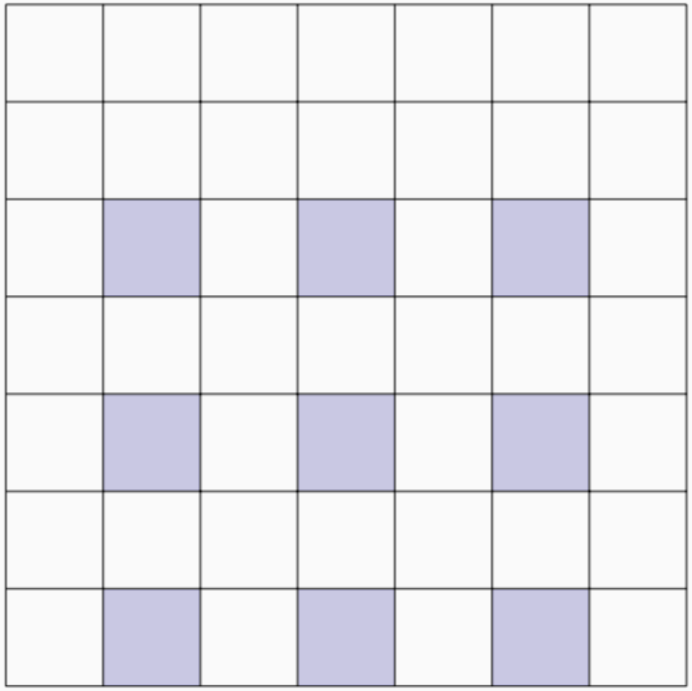
Output



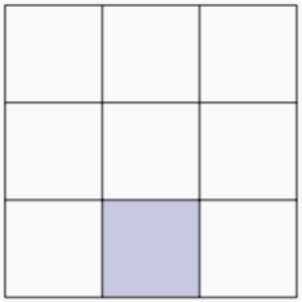
Input



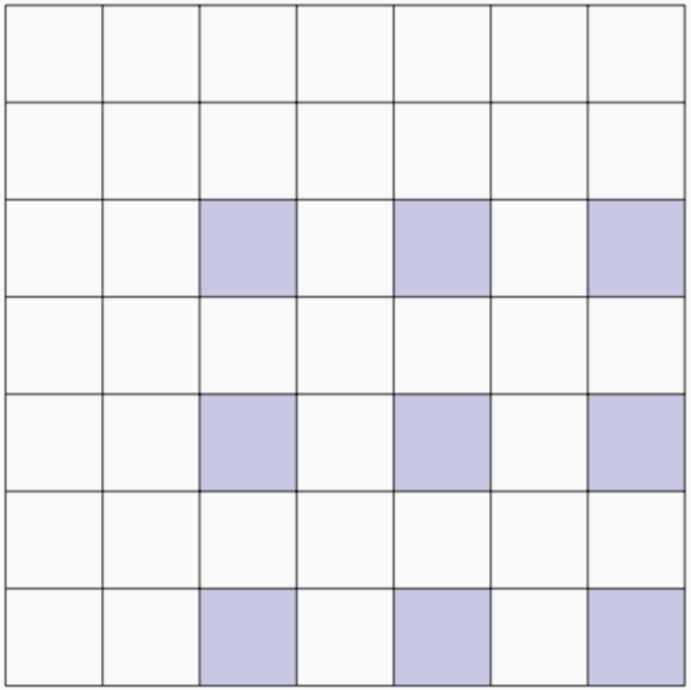
Output



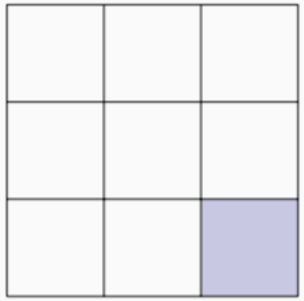
Input



Output



Input



Output

A convolution with a 1d kernel of size $k$ and dilation $d$ can be interpreted as a convolution with a filter of size $1 + (k - 1)d$ with only $k$ non-zero coefficients.

For with $k = 3$ and $d = 4$, the difference between the input map size and the output map size is $1 + (3 - 1)4 - 1 = 8$.

Having a dilation greater than one increases the units' receptive field size without increasing the number of parameters.

Convolutions with stride or dilation strictly greater than one reduce the activation map size, for instance to make a final classification decision.

Such networks have the advantage of simplicity:

- non-linear operations are only in the activation function,
- joint operations that combine multiple activations to a single one are only in linear layers

# References

F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. CoRR, abs/1511.07122v3, 2015.