

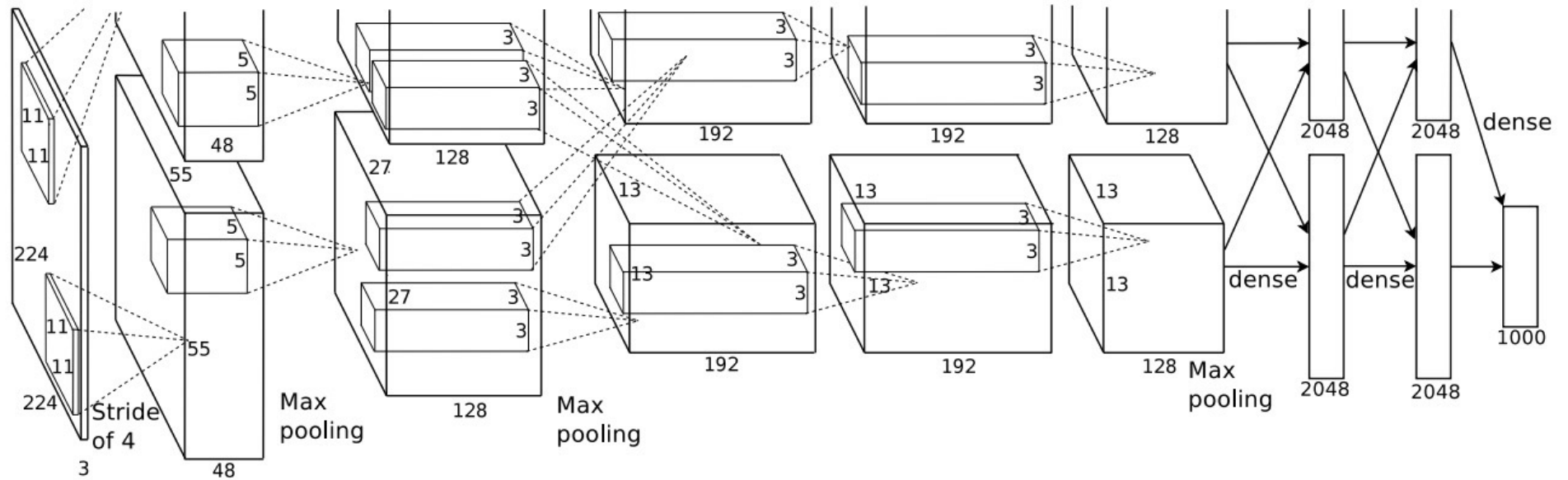
Convolutional Neural Networks

Xiaolong Wang

This Class

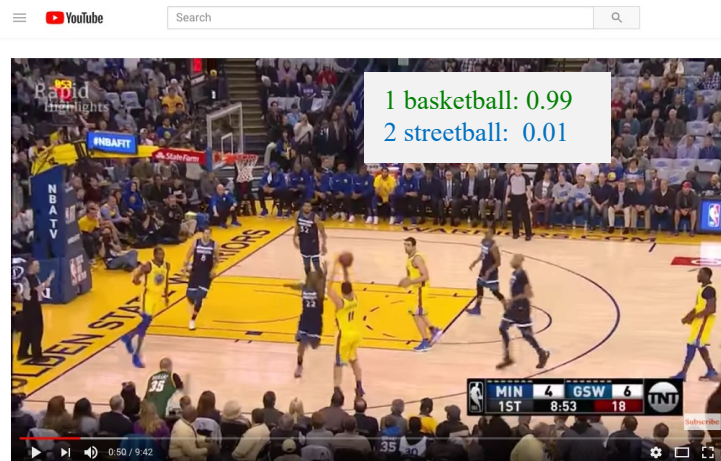
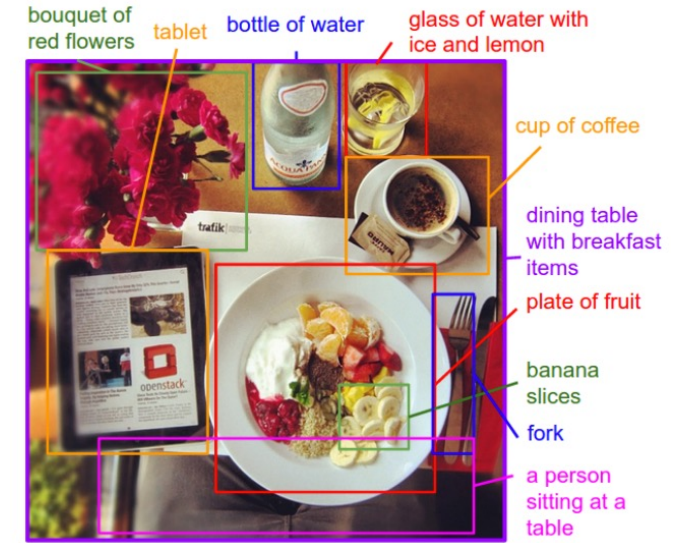
- Convolutional Operation
- Convolutional Neural Networks

Convolutional Neural Networks



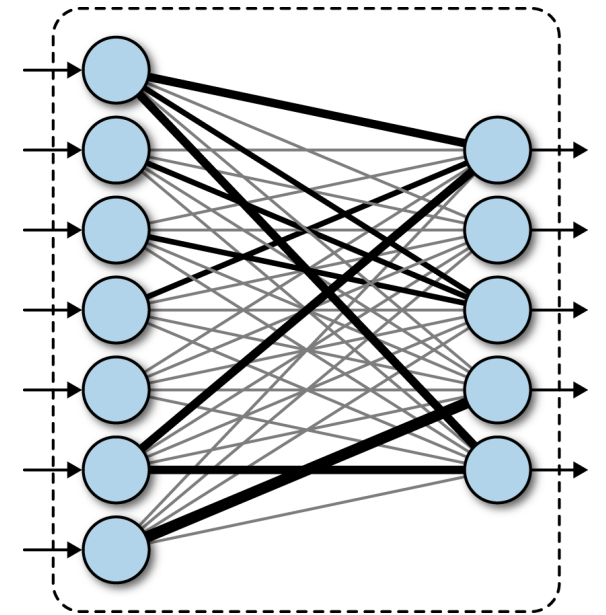
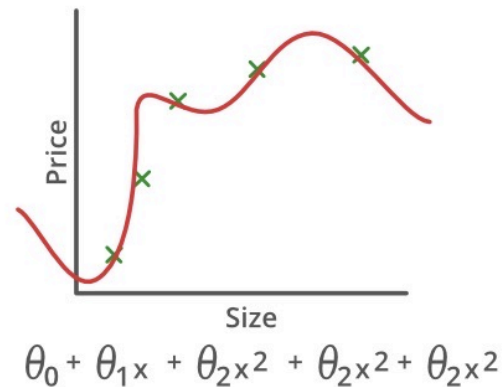
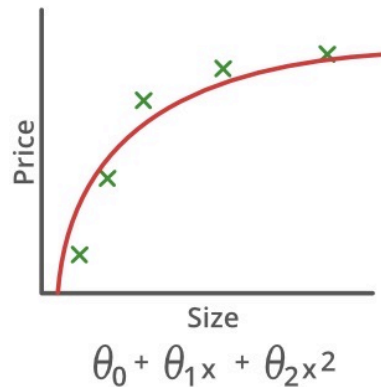
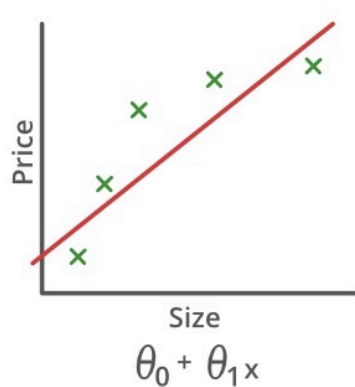
AlexNet (Krizhevsky et al. 2012)

Convolutional Neural Networks



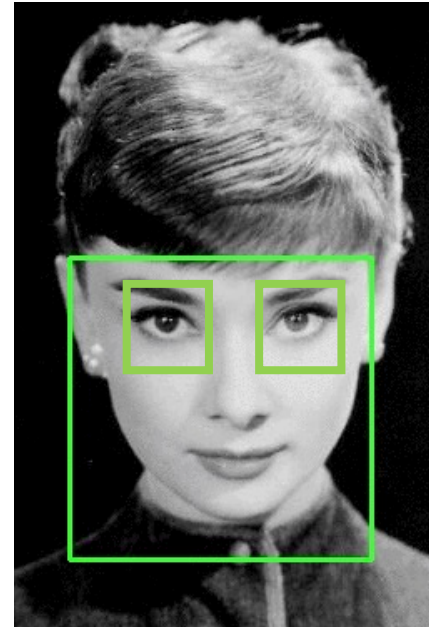
Structural Prior in Images

- Disadvantage of MLP:
 - Large number of parameters
 - First layer: $(32 \times 32 \times 3) \times 128 = 393,216$
- Why is large number of parameters bad?
 - Overfitting

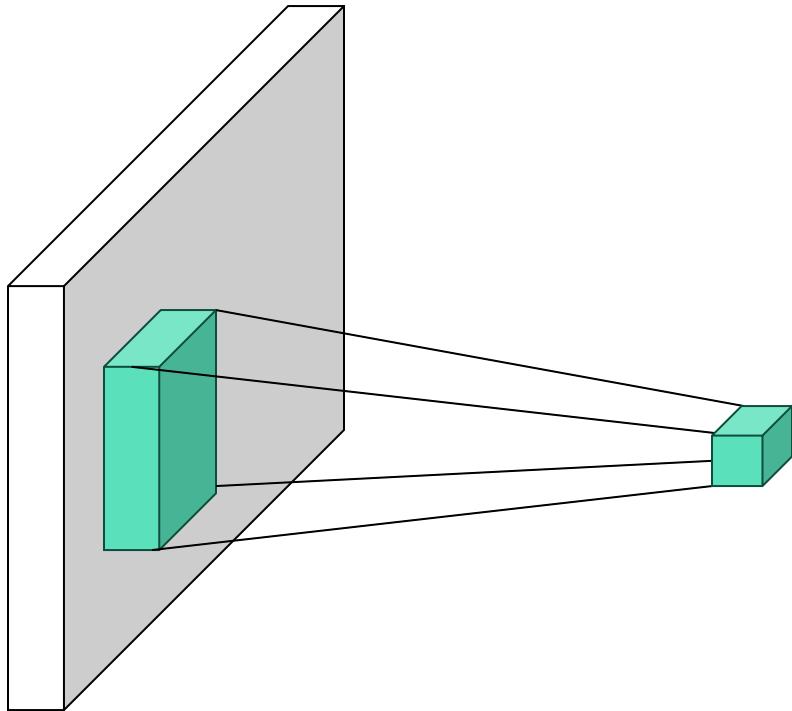


Structural Prior in Images

- ConvNets use the structural prior in images:
 - There are repetitive patterns in images
 - We should re-use and share the filter across the whole image
 - Reduce parameters, avoid overfitting



Convolution



image

Given a 3×3 filter

Compute the response at location (k, l)

$$z(k, l) = \sum_{i, j=-1}^1 W(i, j) x(k + i, l + j)$$

Convolution

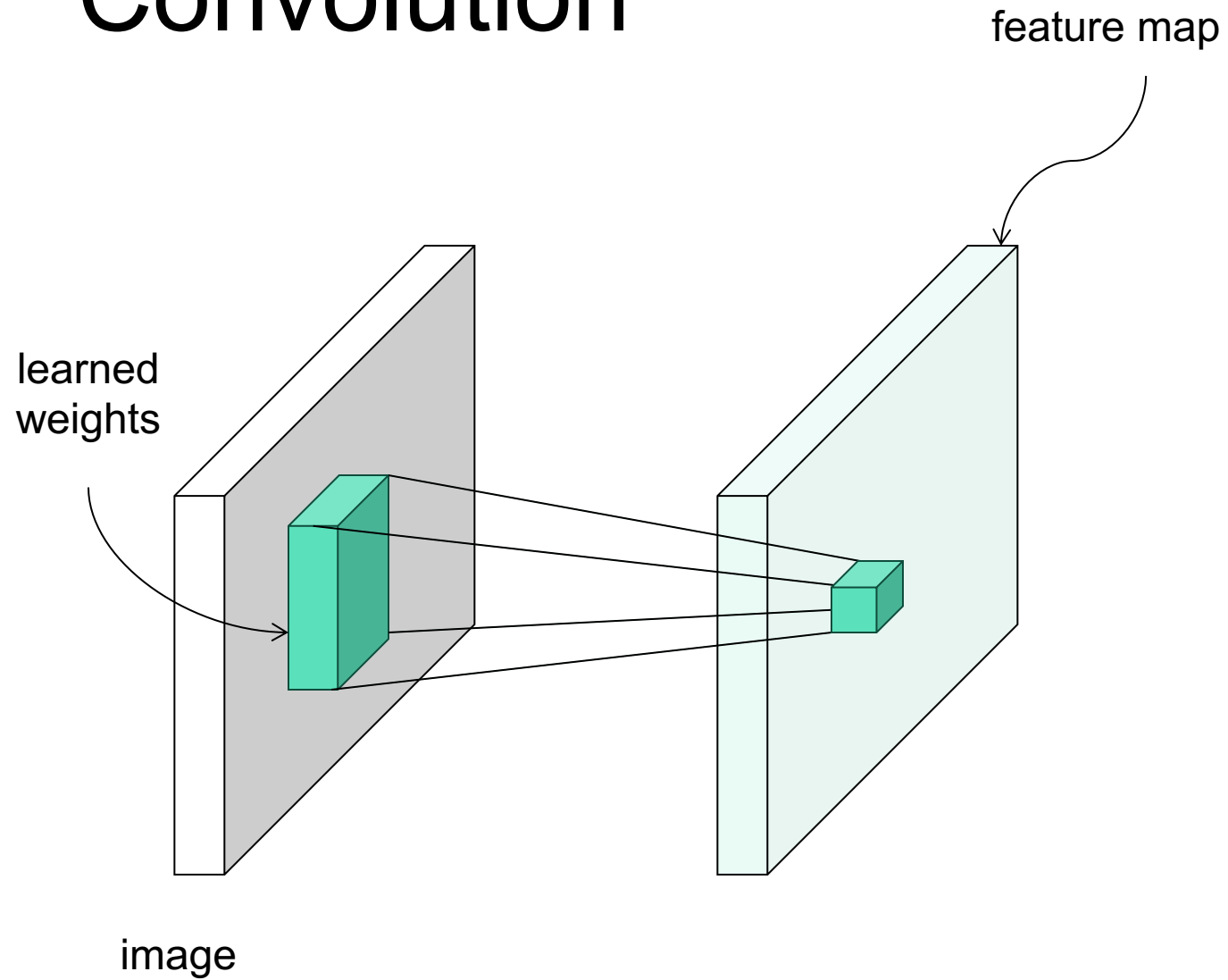
0	1	2	2	2	0	1
1	0	2	1	2	0	2
2	1	0	2	0	0	1
1	0	2	1	2	0	2
0	1	2	2	2	0	1
1	0	2	1	2	0	2
2	1	0	2	0	0	1

*

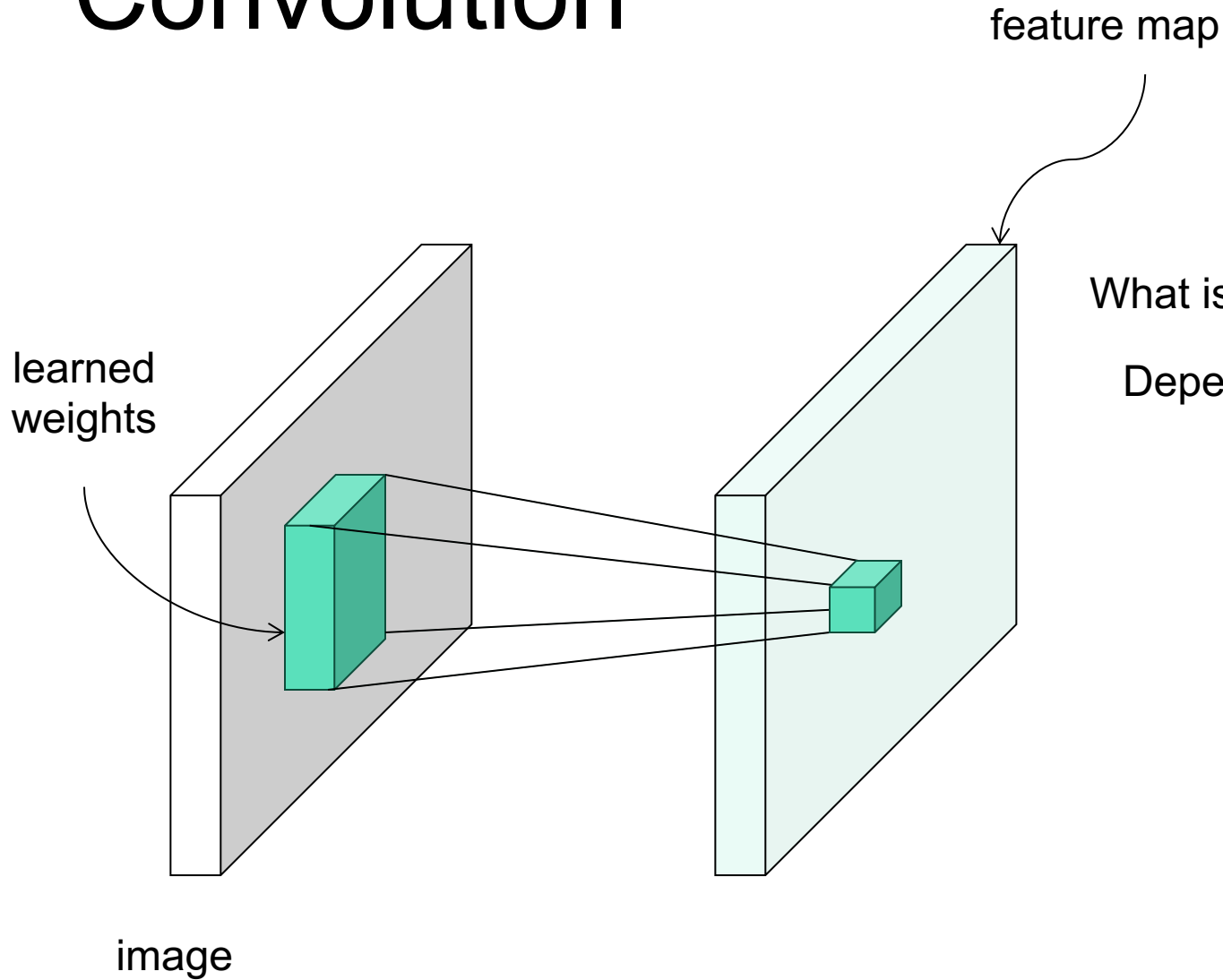
1	0	-1
0	-1	1
-1	1	0

$$\begin{aligned} & 0 \times 1 + 1 \times 0 + 2 \times (-1) \\ & + 1 \times 0 + 0 \times (-1) + 2 \times 1 \\ & + 2 \times (-1) + 1 \times 1 + 0 \times 0 \\ & = -1 \end{aligned}$$

Convolution

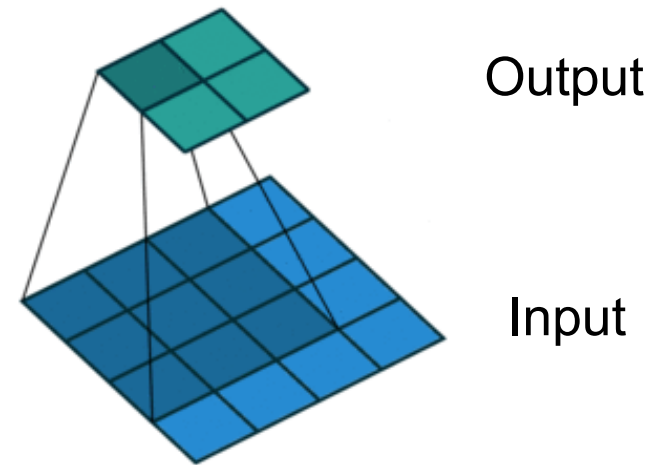


Convolution



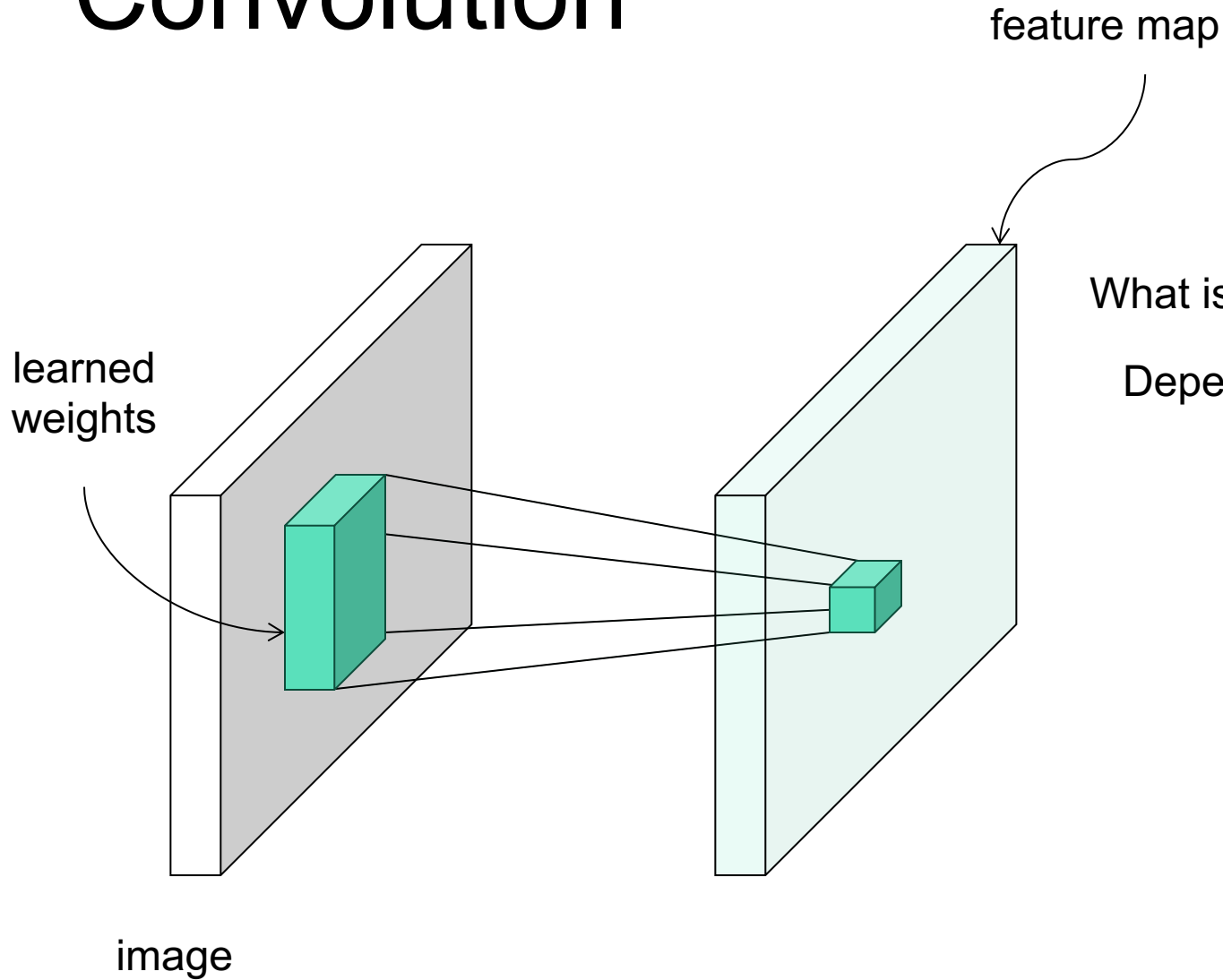
What is the feature map resolution?

Depends on *padding* and *stride*:



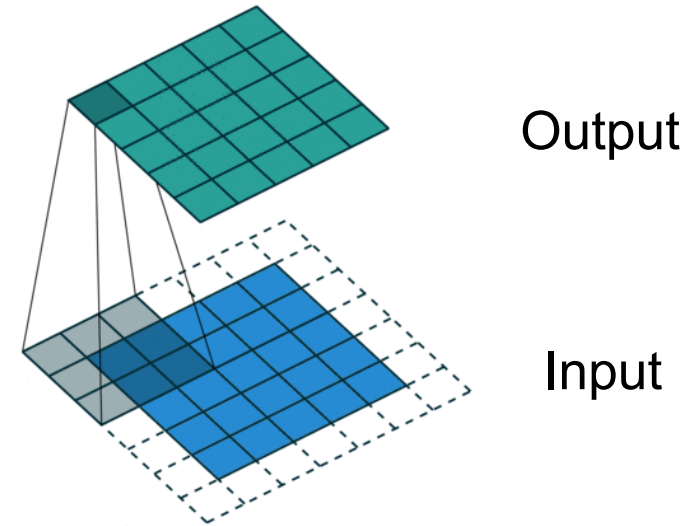
No padding, stride 1

Convolution



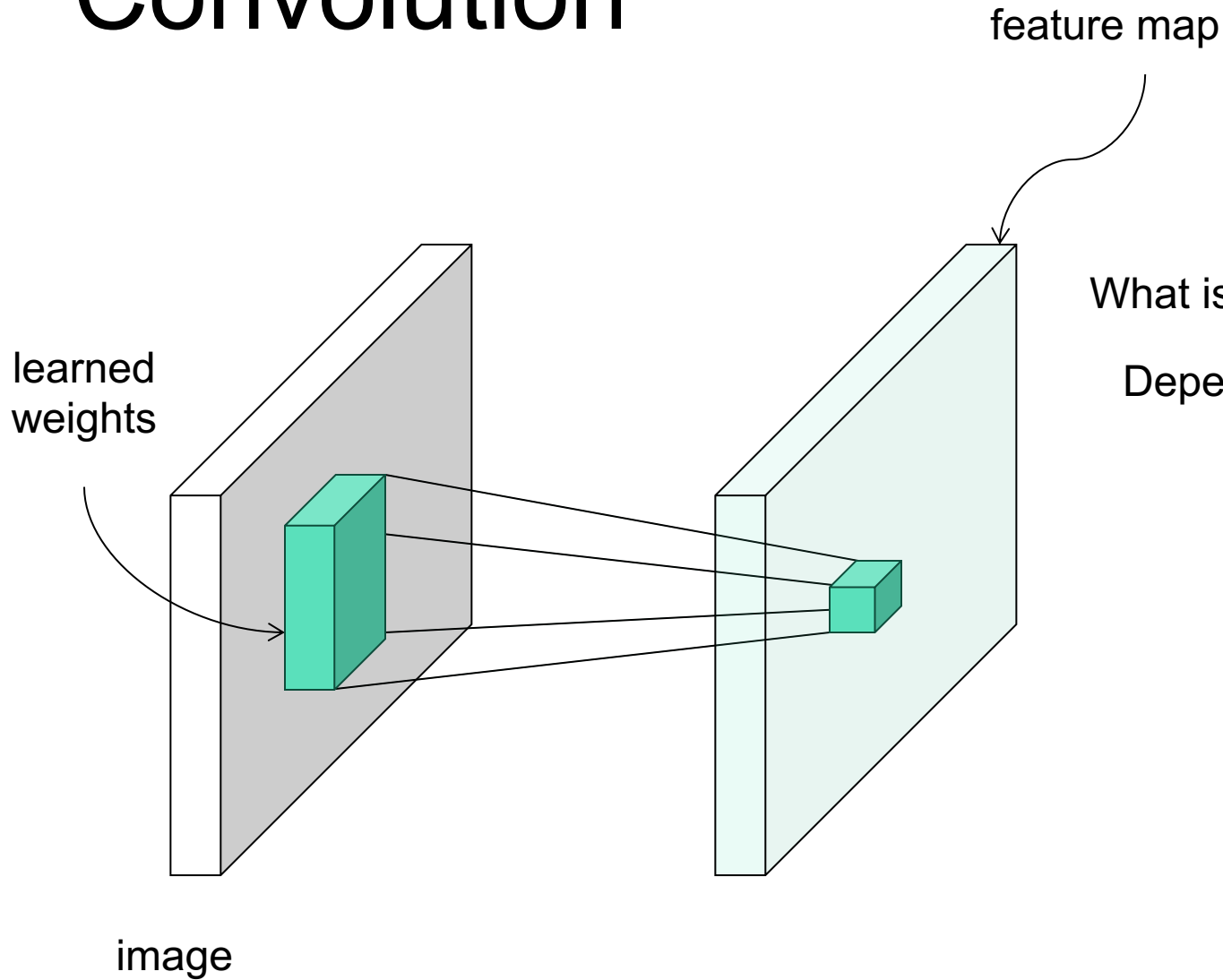
What is the feature map resolution?

Depends on *padding* and *stride*:



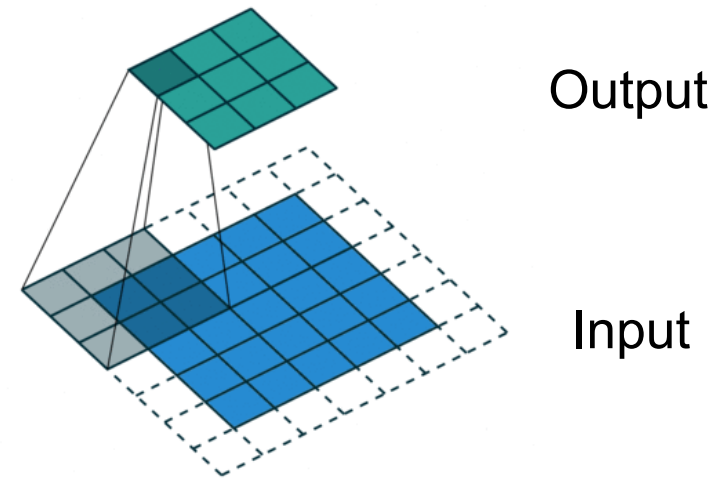
With padding, stride 1

Convolution



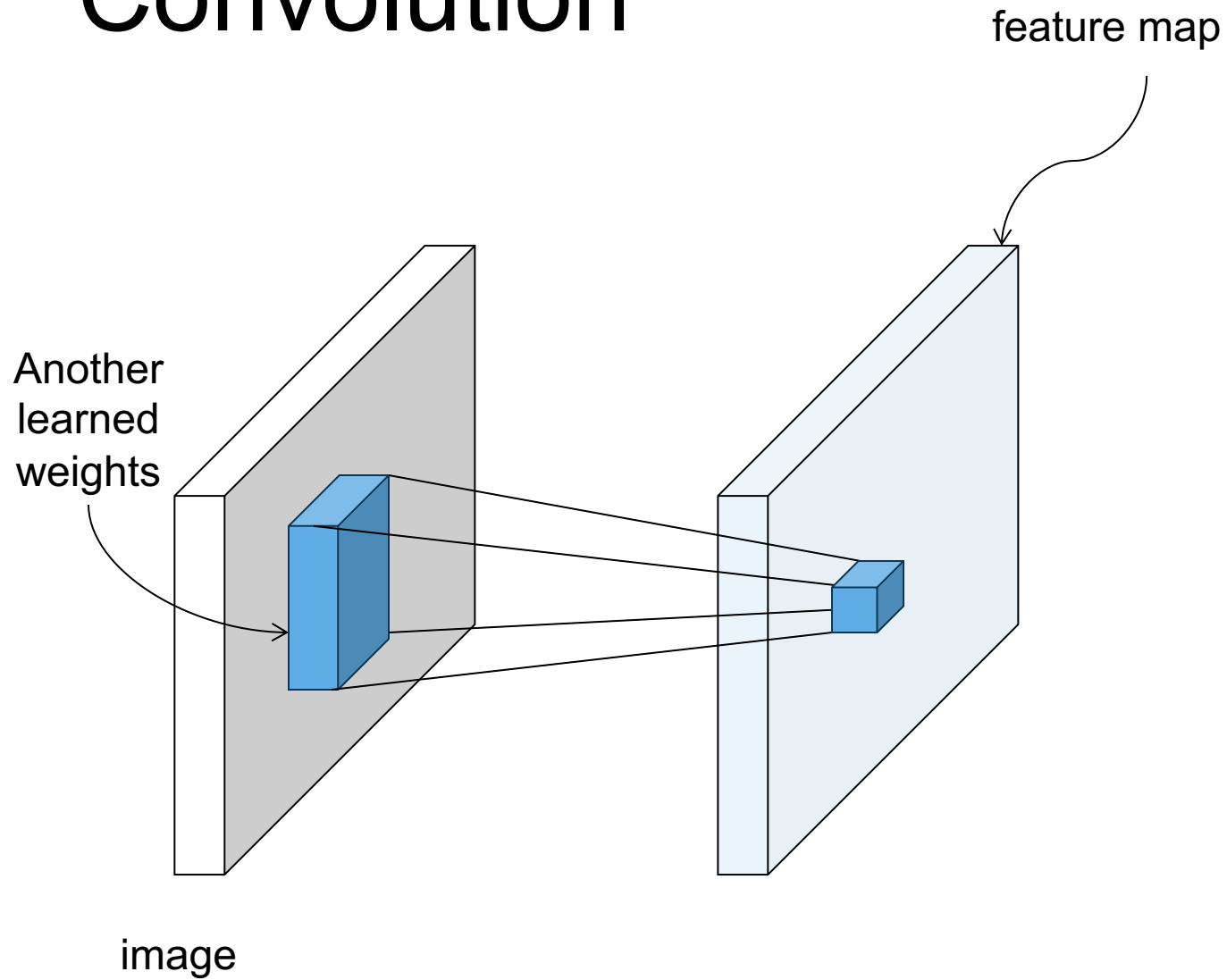
What is the feature map resolution?

Depends on *padding* and *stride*:

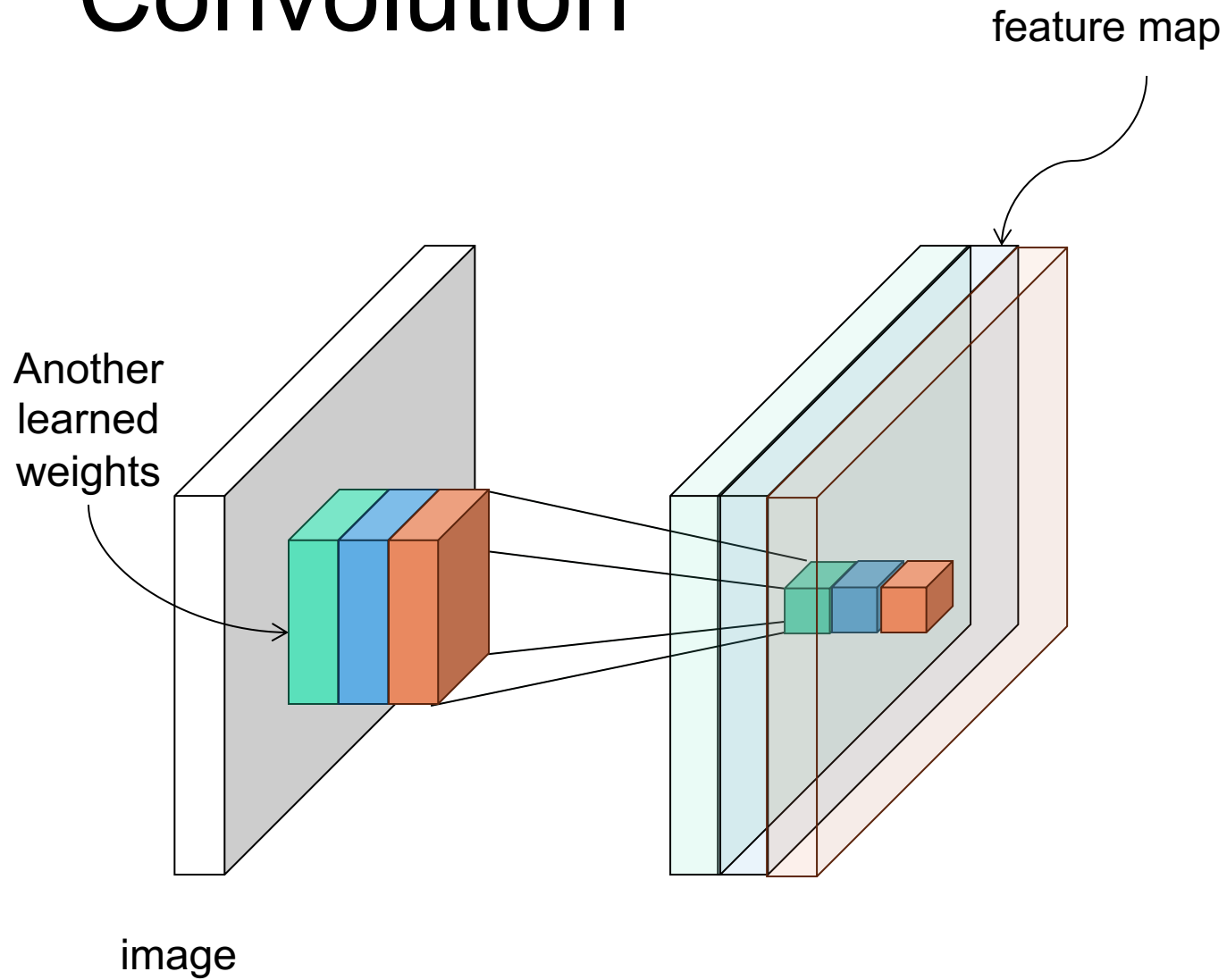


With padding, stride 2

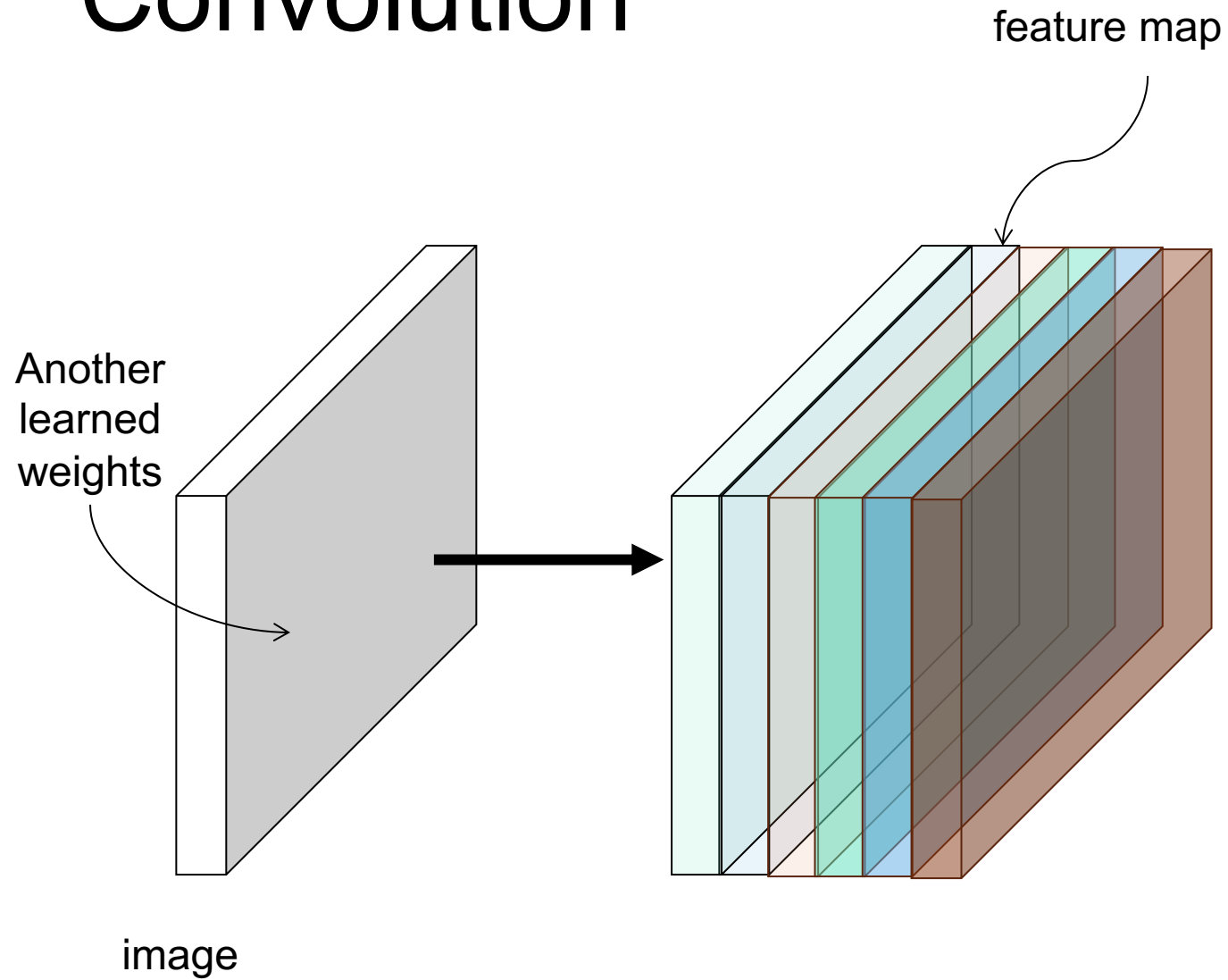
Convolution



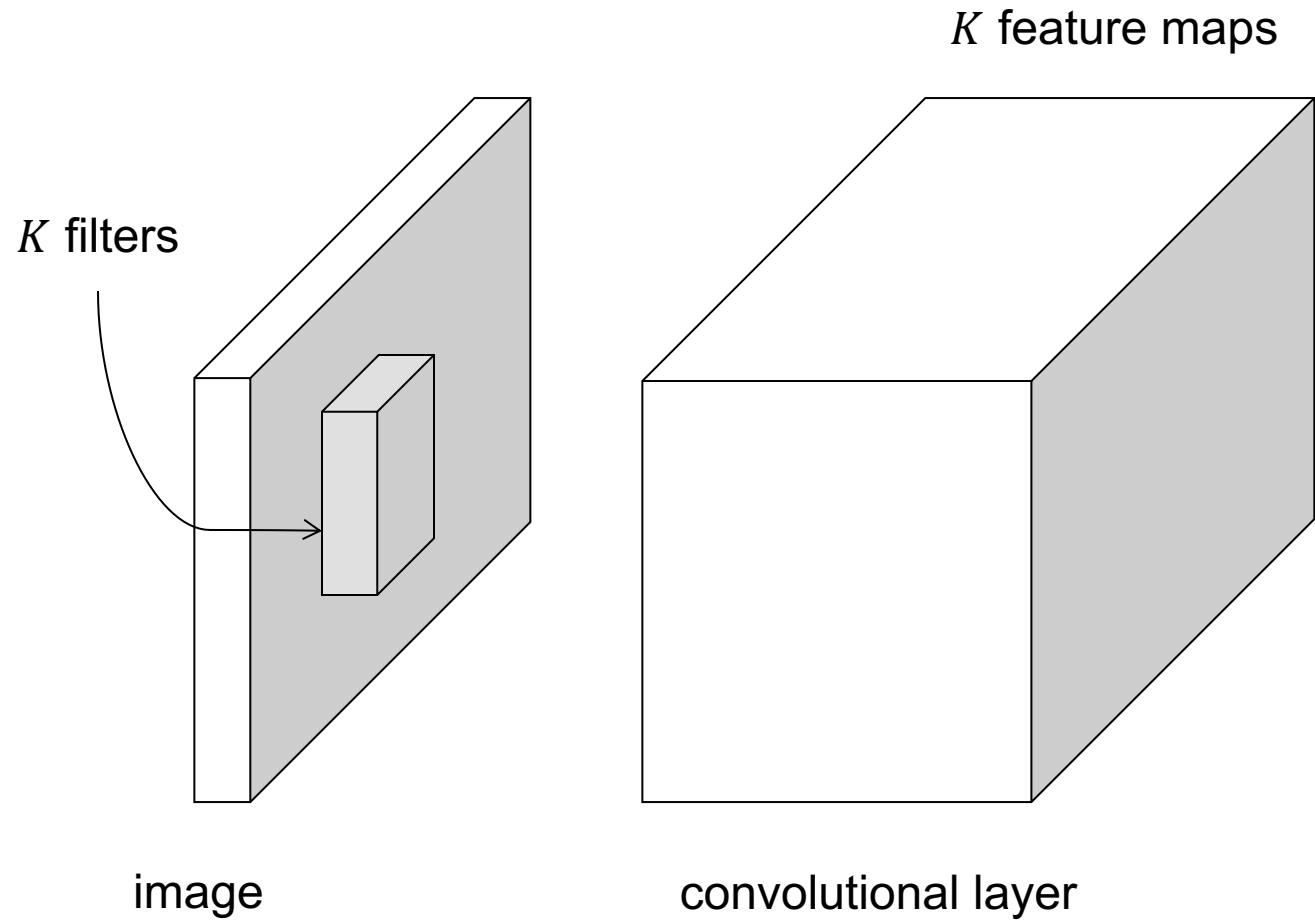
Convolution



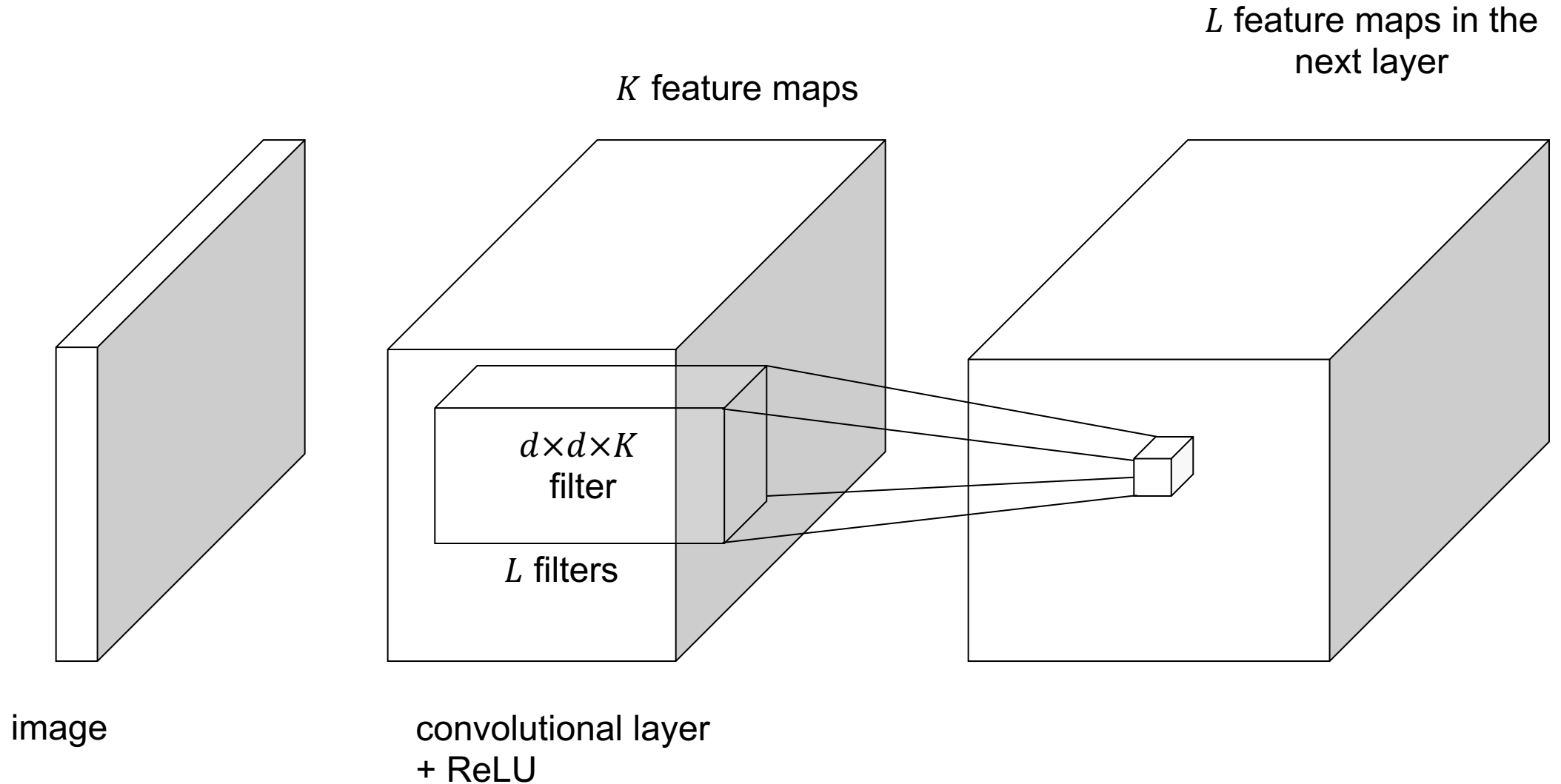
Convolution



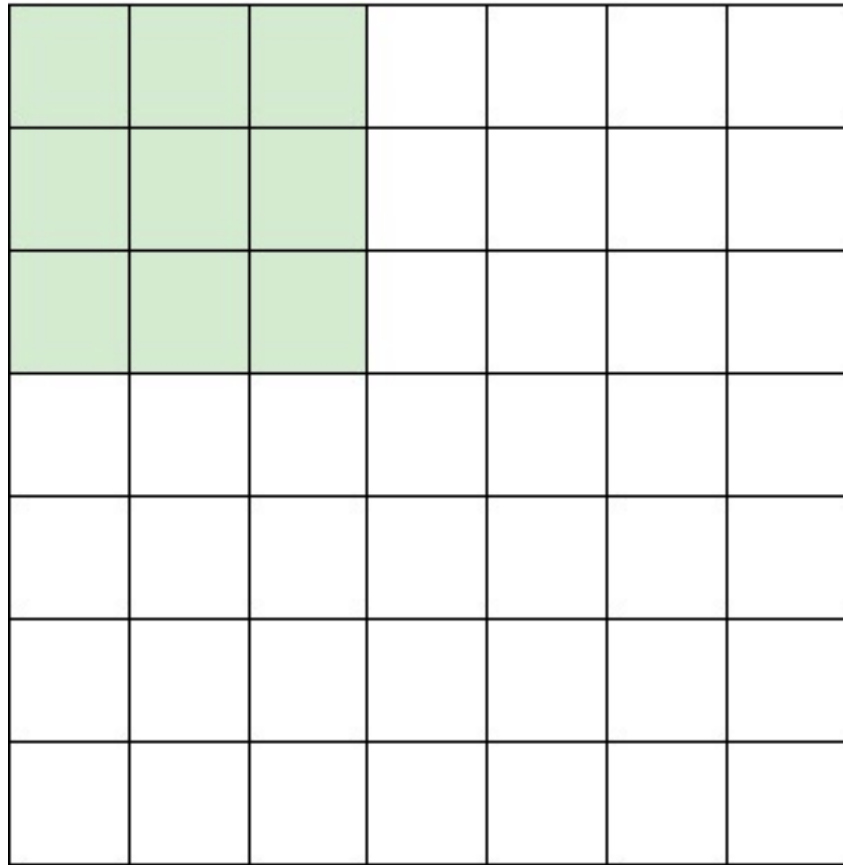
Convolution



Convolution



More examples on stride, padding, filter size



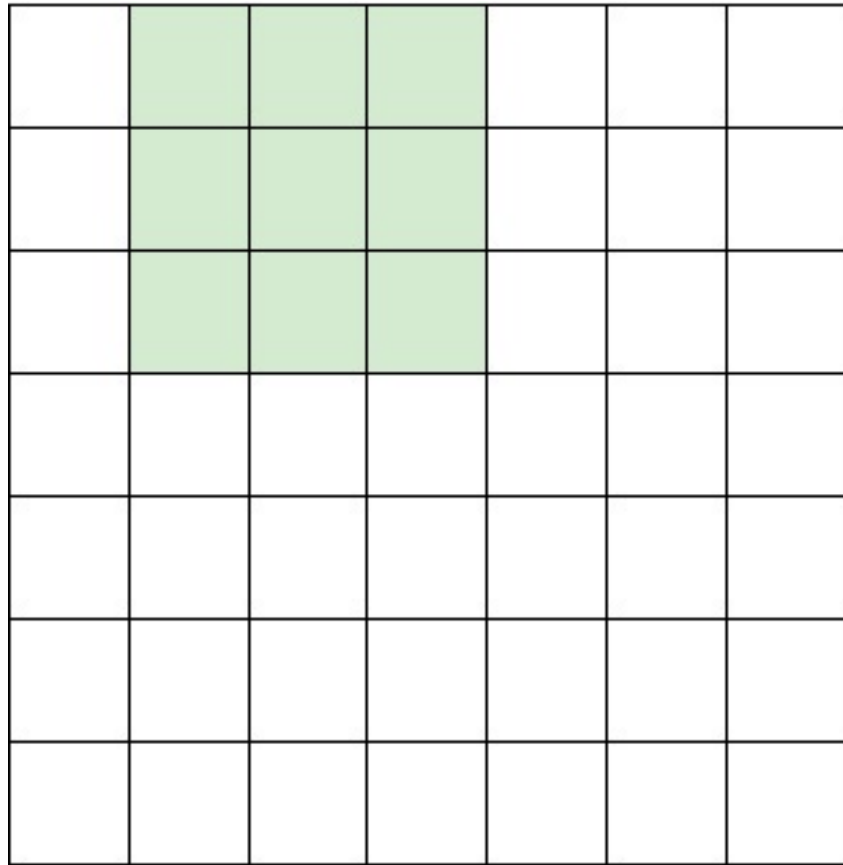
Input: 7×7 feature map

Filter: 3×3 , stride=1

7

7

More examples on stride, padding, filter size



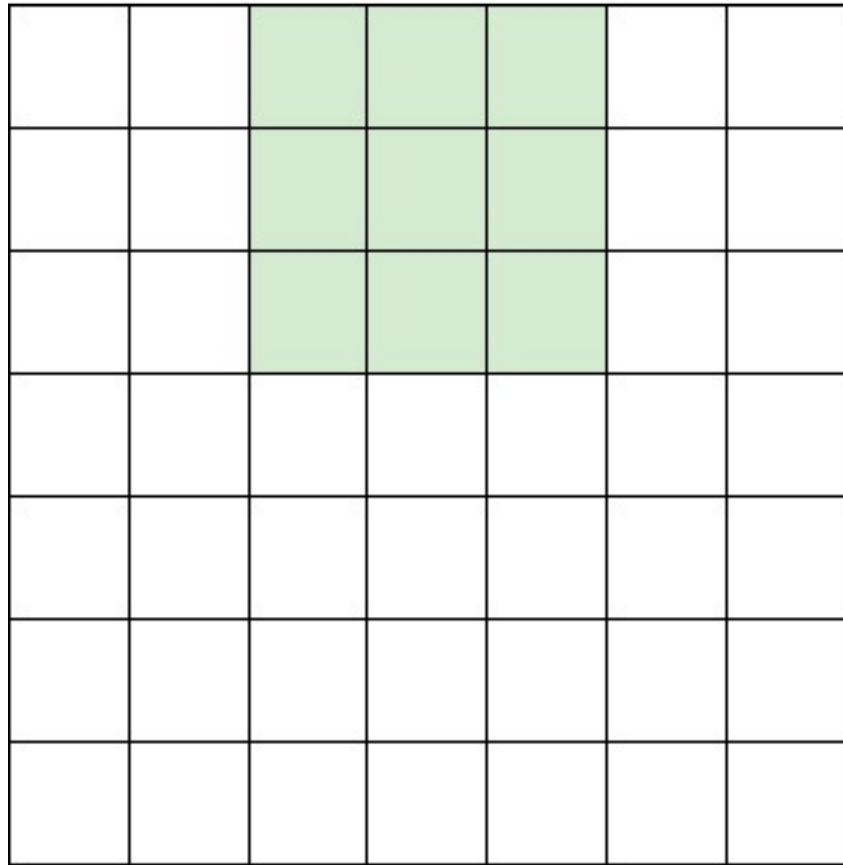
Input: 7×7 feature map

Filter: 3×3 , stride=1

7

7

More examples on stride, padding, filter size



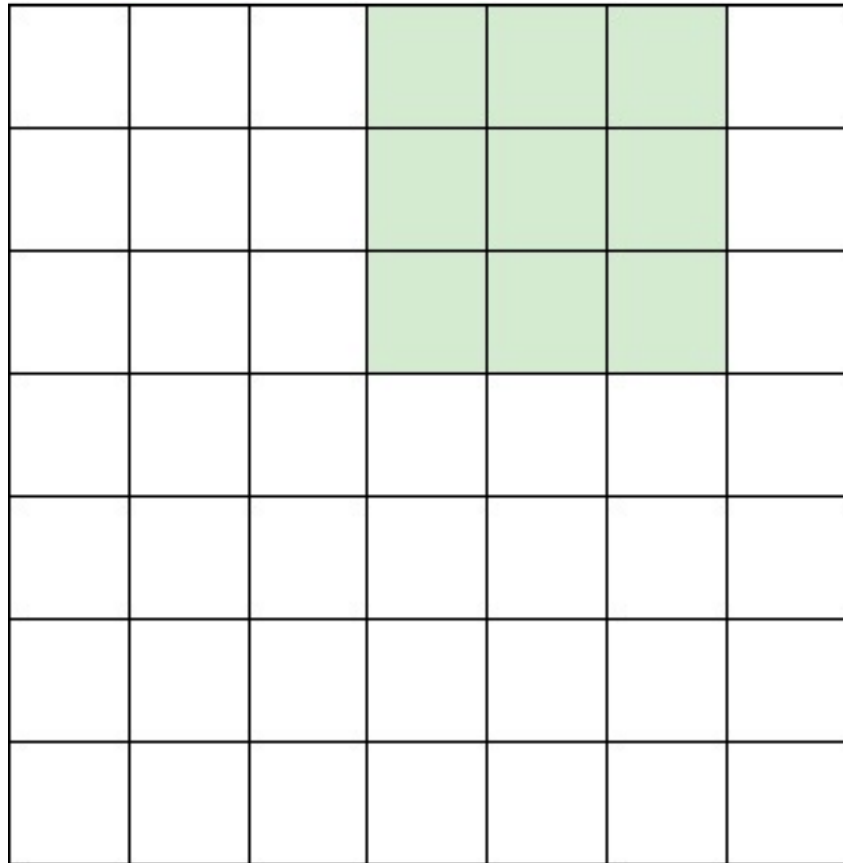
Input: 7×7 feature map

Filter: 3×3 , stride=1

7

7

More examples on stride, padding, filter size



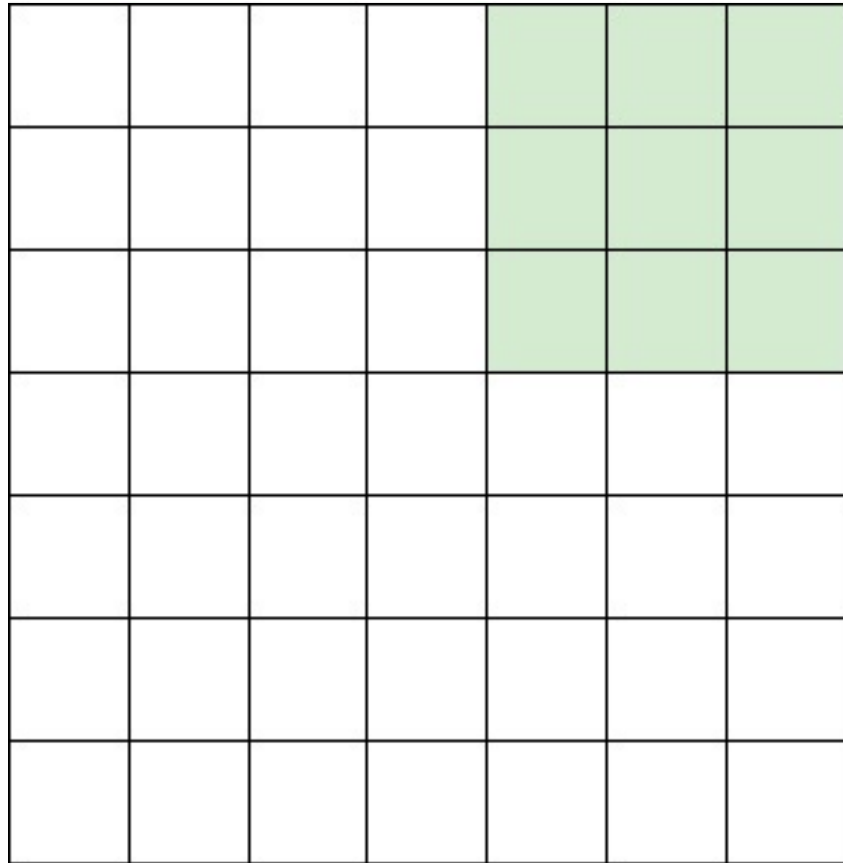
7

7

Input: 7×7 feature map

Filter: 3×3 , stride=1

More examples on stride, padding, filter size

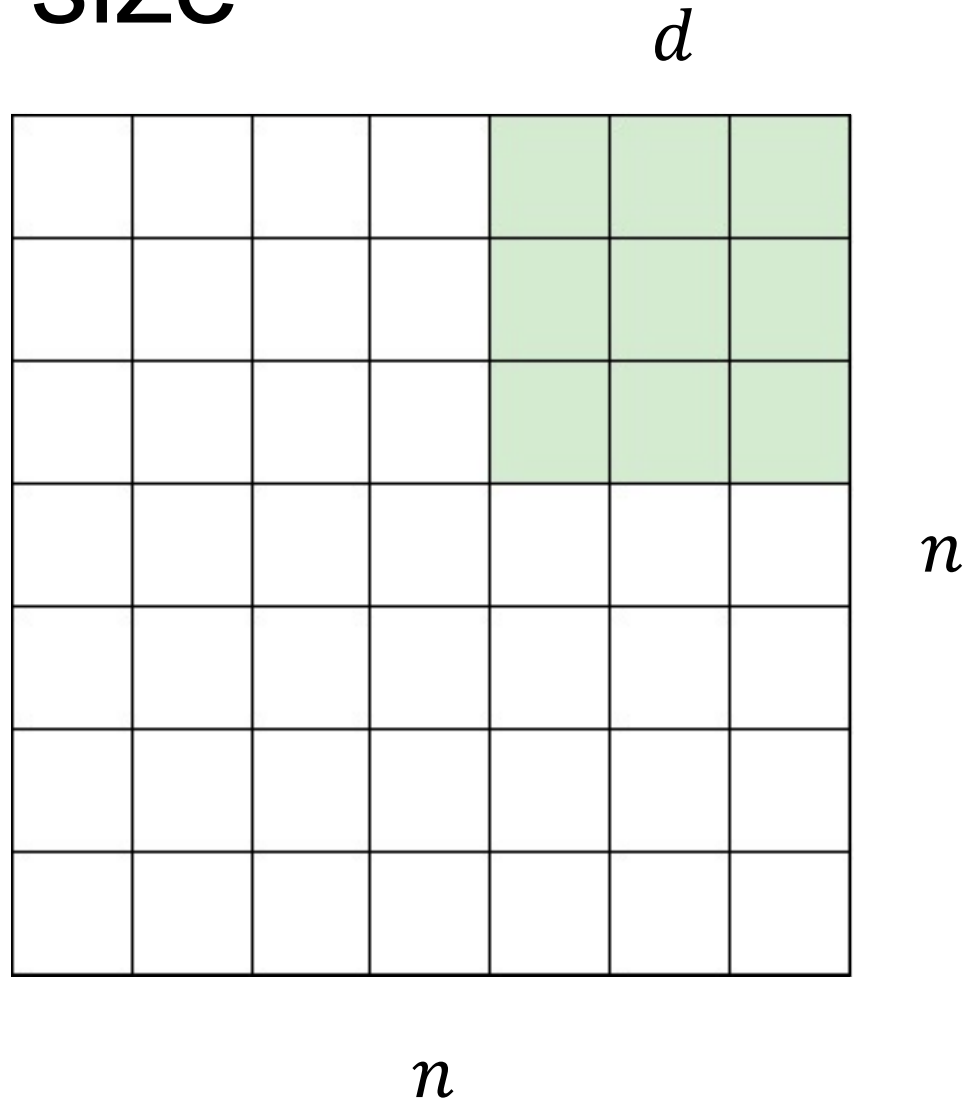


Input: 7×7 feature map

Filter: 3×3 , stride=1

Output: 5×5 feature map

More examples on stride, padding, filter size



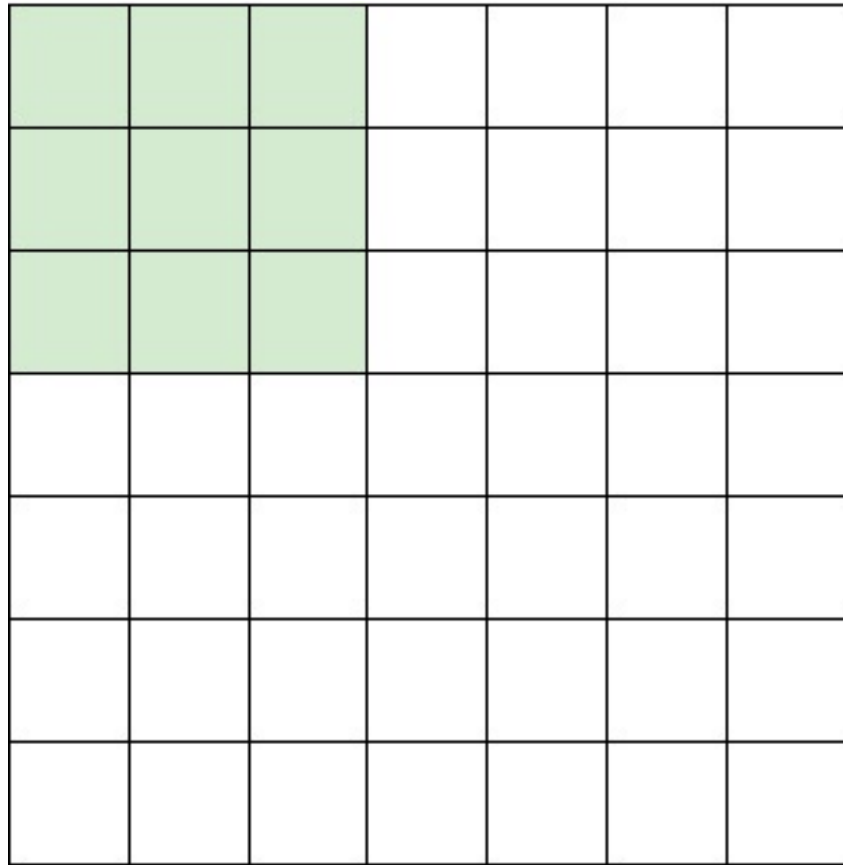
Input: $n \times n$ feature map

Filter: $d \times d$, stride=1

Output size:

$$\frac{n - d}{stride} + 1$$

More examples on stride, padding, filter size



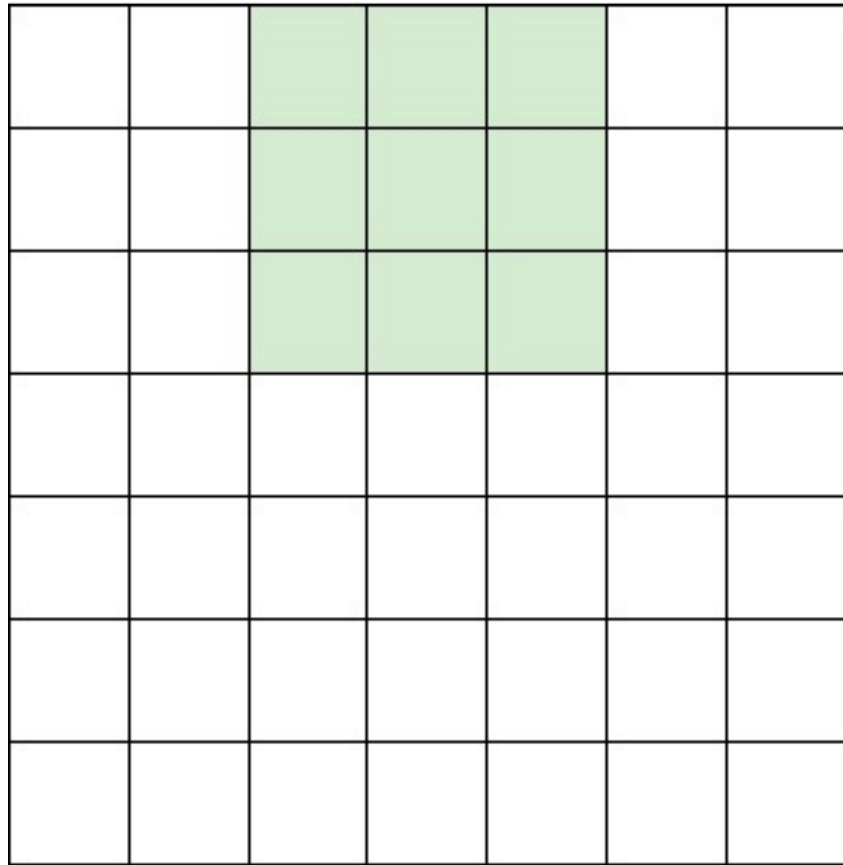
Input: 7×7 feature map

Filter: 3×3 , stride=2

7

7

More examples on stride, padding, filter size



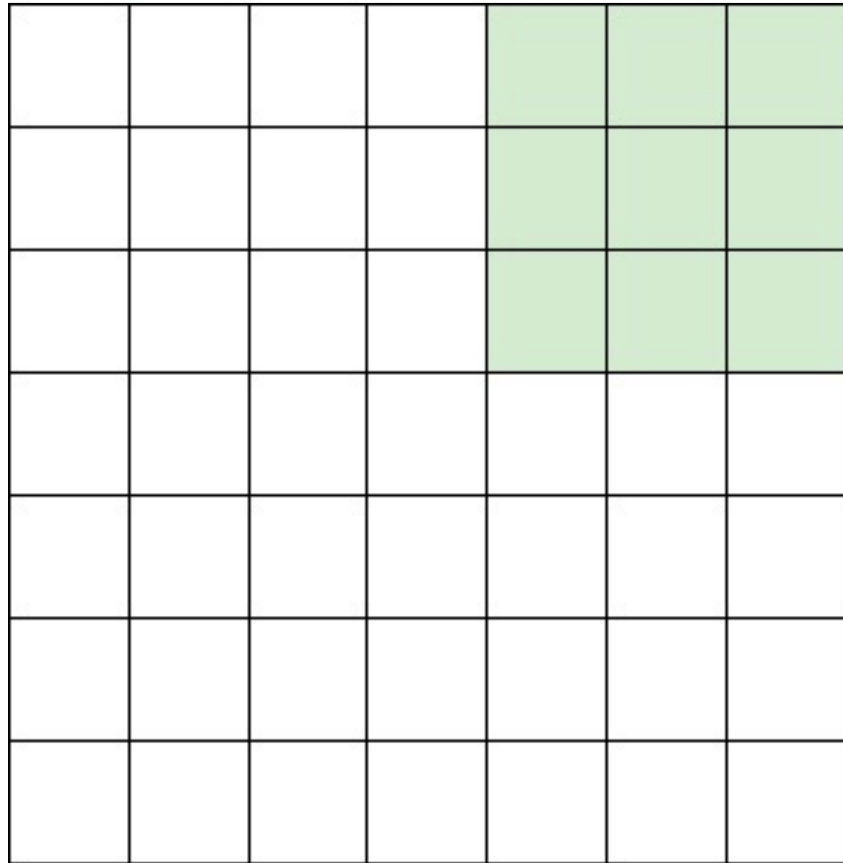
Input: 7×7 feature map

Filter: 3×3 , stride=2

7

7

More examples on stride, padding, filter size



Input: 7×7 feature map

Filter: 3×3 , stride=2

Output: 3×3 feature map

$$\frac{n - d}{stride} + 1 = \frac{7 - 3}{2} + 1 = 3$$

More examples on stride, padding, filter size

0	0	0	0	0	0			
0								
0								
0								
0								

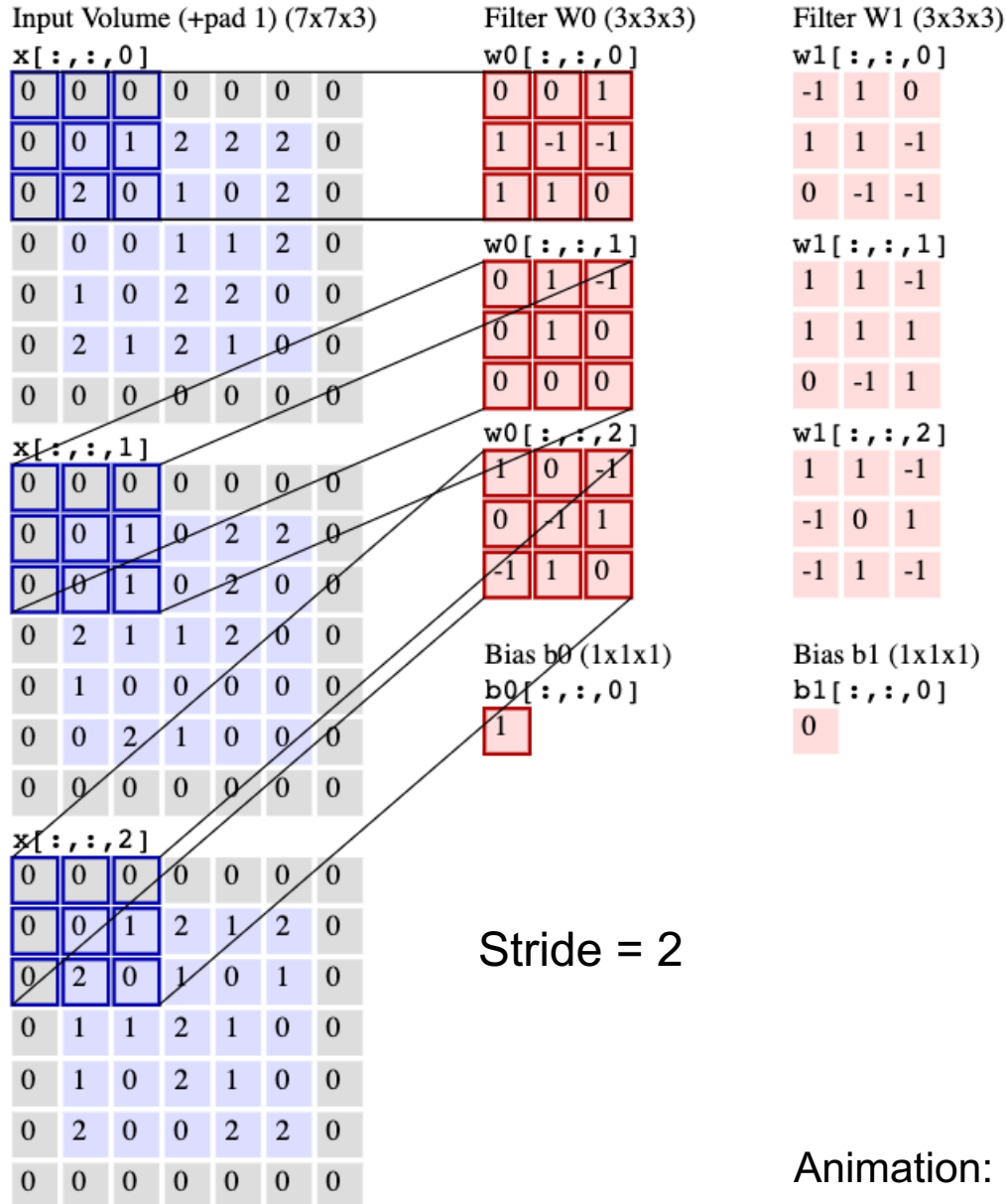
Input: 7×7 feature map

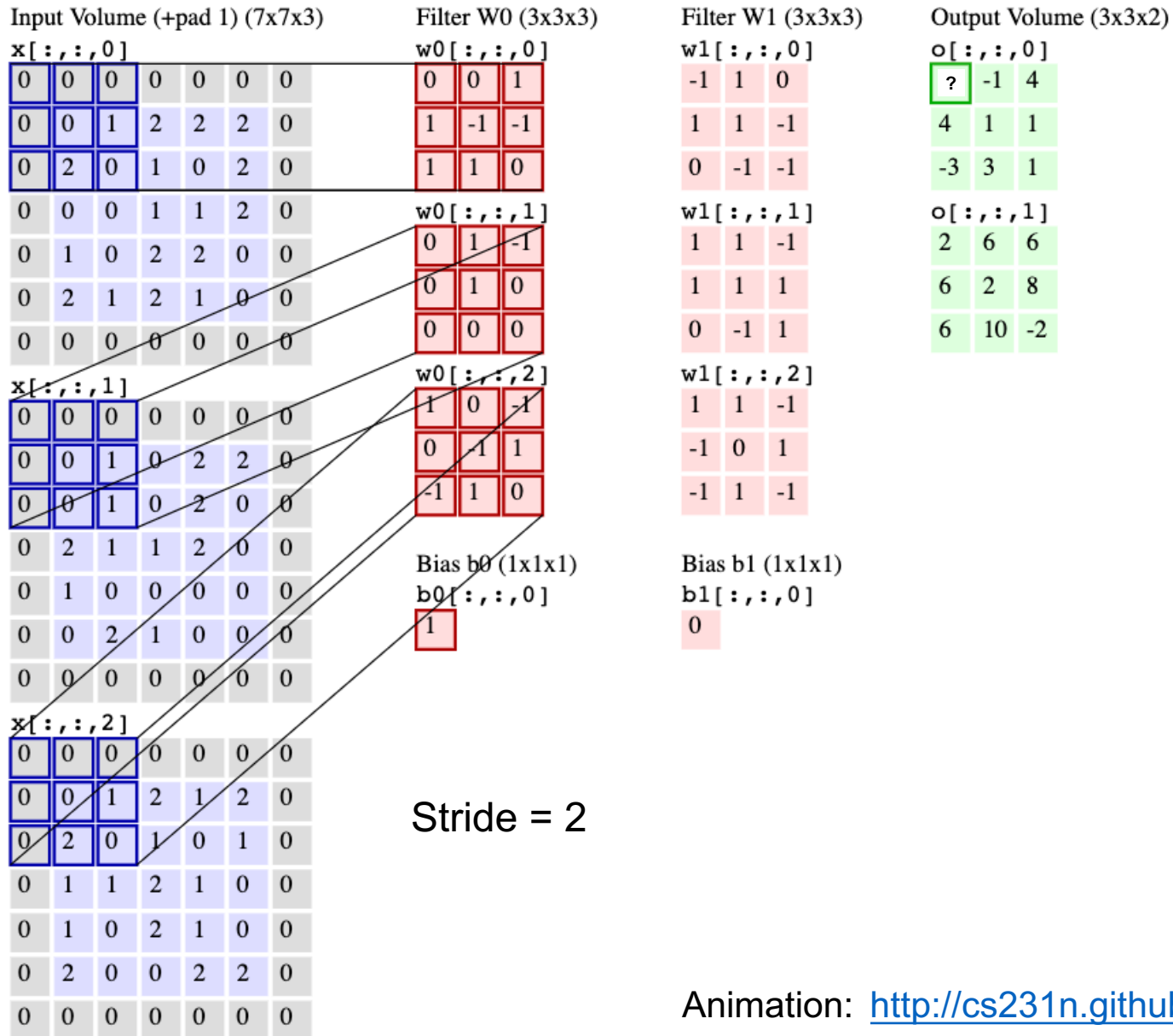
Filter: 3×3 , stride=2

Padding: 1

Output: 4×4 feature map

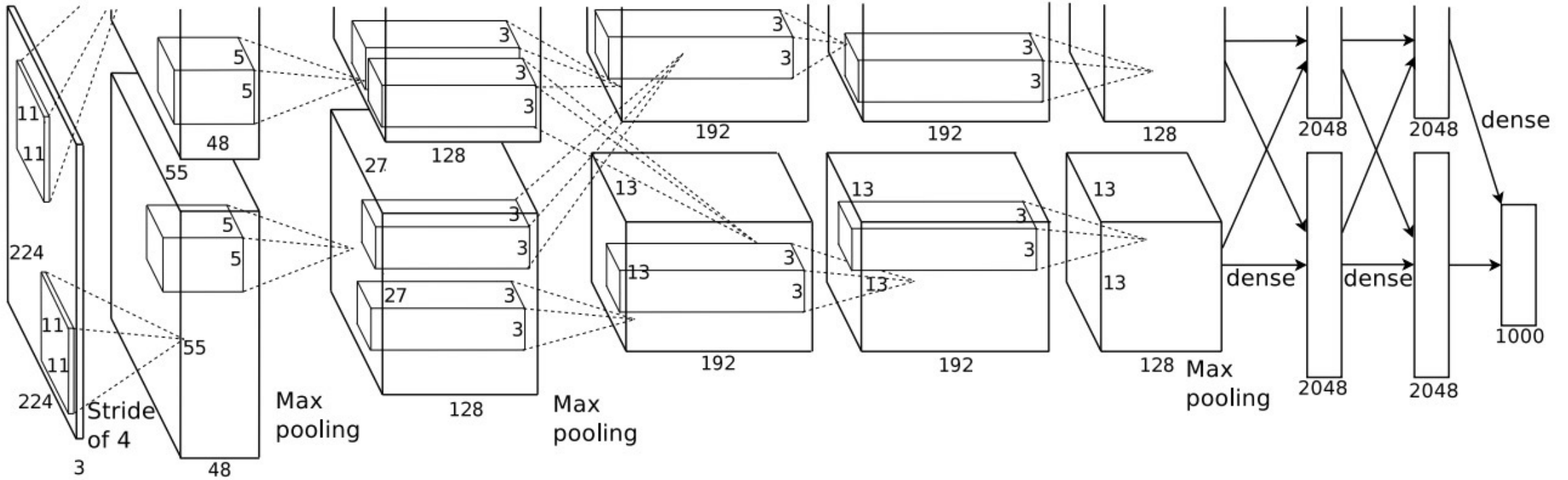
$$\frac{n + 2 \times pad - d}{stride} + 1 = \frac{7 + 2 - 3}{2} + 1 = 4$$



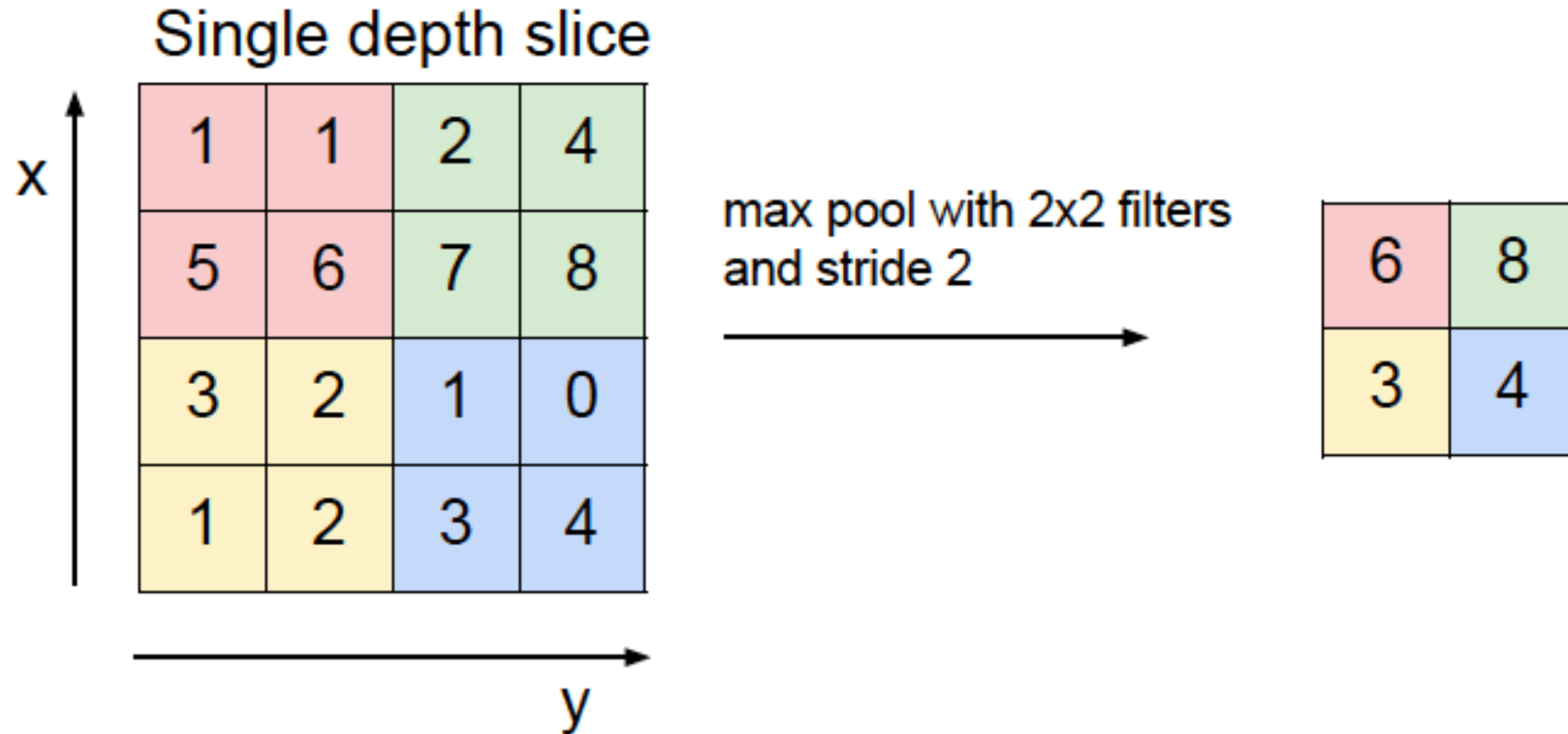


Max Pooling

Max Pooling

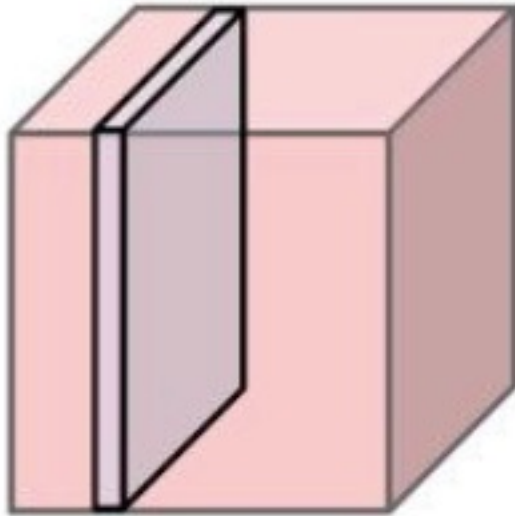


Max Pooling



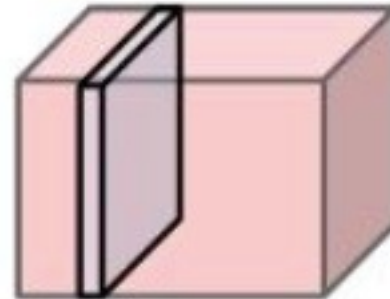
Max Pooling

224x224x64



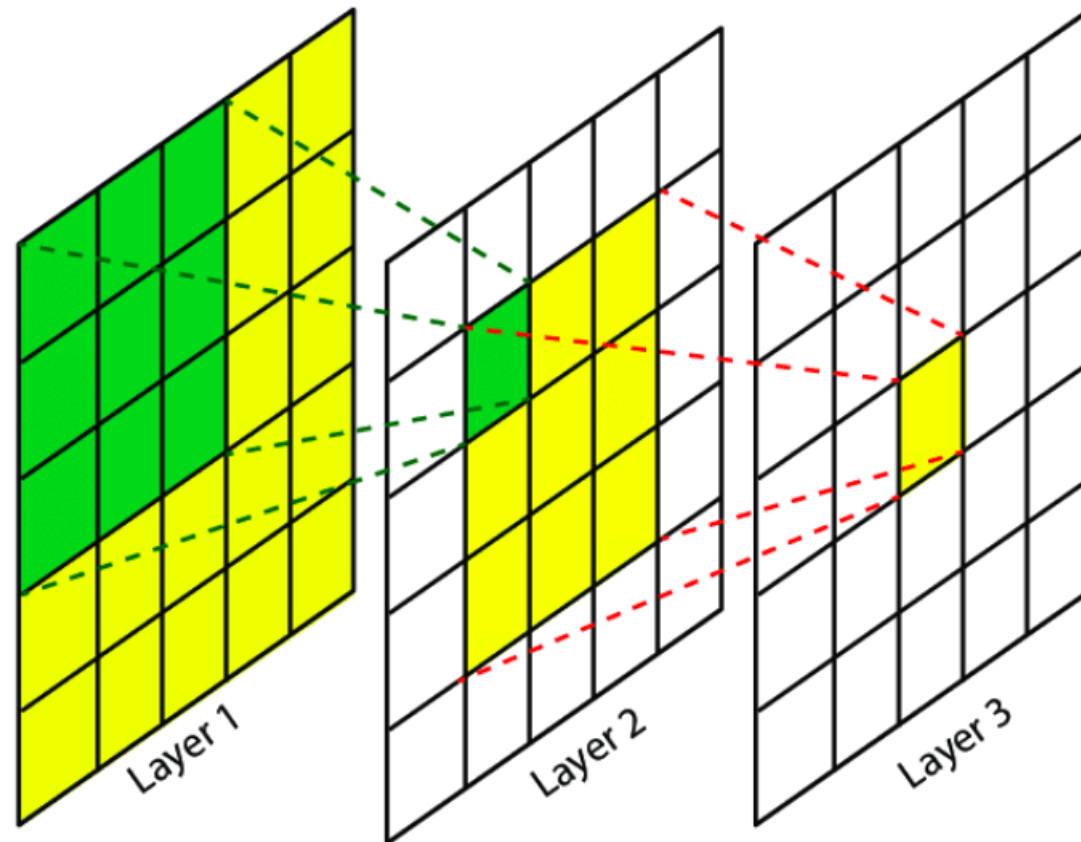
pool
→

112x112x64



Why Max Pooling

- Save computation and memory
- Increase receptive field



This Class

- Convolutional Operation
- Back-propagation for Convolutional Neural Networks
- Max Pooling

Next Class

More elements in training Convolutional Neural Networks