SSY098 - Image Analysis

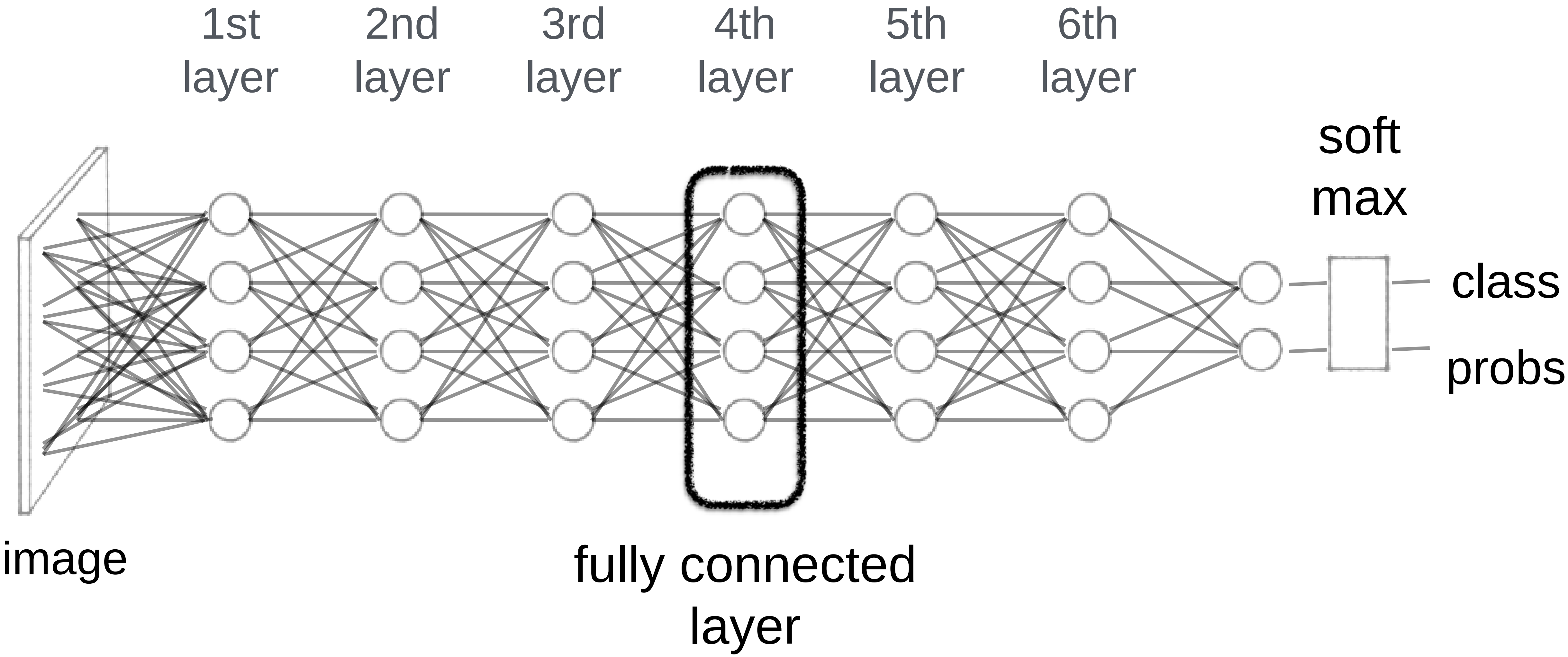
Lecture 6 - (More) Convolutional Neural Networks

Torsten Sattler
(slides adapted from Olof Enqvist)

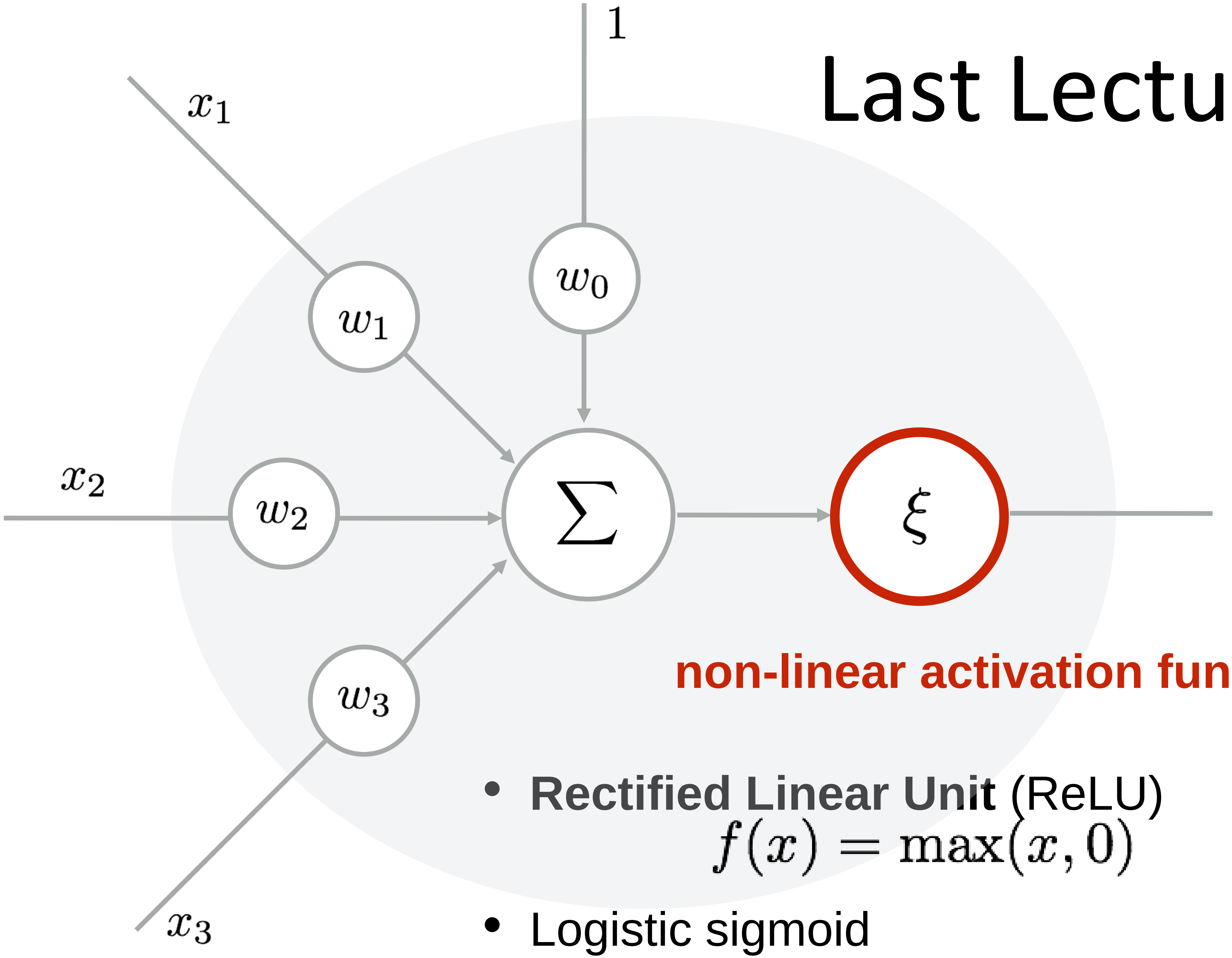
Last Lecture

Jan. 20	Introduction, Linear classifiers and filtering	
Jan. 23	Filtering, gradients, scale	Lab 1
Jan. 27	Local features	
Jan. 30	Learning a classifier	Lab 2
Feb. 3	**Convolutional neural networks**	
Feb. 6	More convolutional neural networks	
Feb. 10	Robust model fitting and RANSAC	Lab 3
Feb. 13	Image registration	
Feb. 17	Camera Geometry	Lab 4
Feb. 20	More camera geometry	
Feb. 24	Generative neural networks	
Feb. 27	Generative neural networks	
Mar. 2	TBA	
Mar. 9	TBA	

Last Lecture



Last Lecture



A neuron

non-linear activation function

- **Rectified Linear Unit** (ReLU)

$$f(x) = \max(x, 0)$$

- Logistic sigmoid
- Hyperbolic tangent (tanh)
- …

Last Lecture

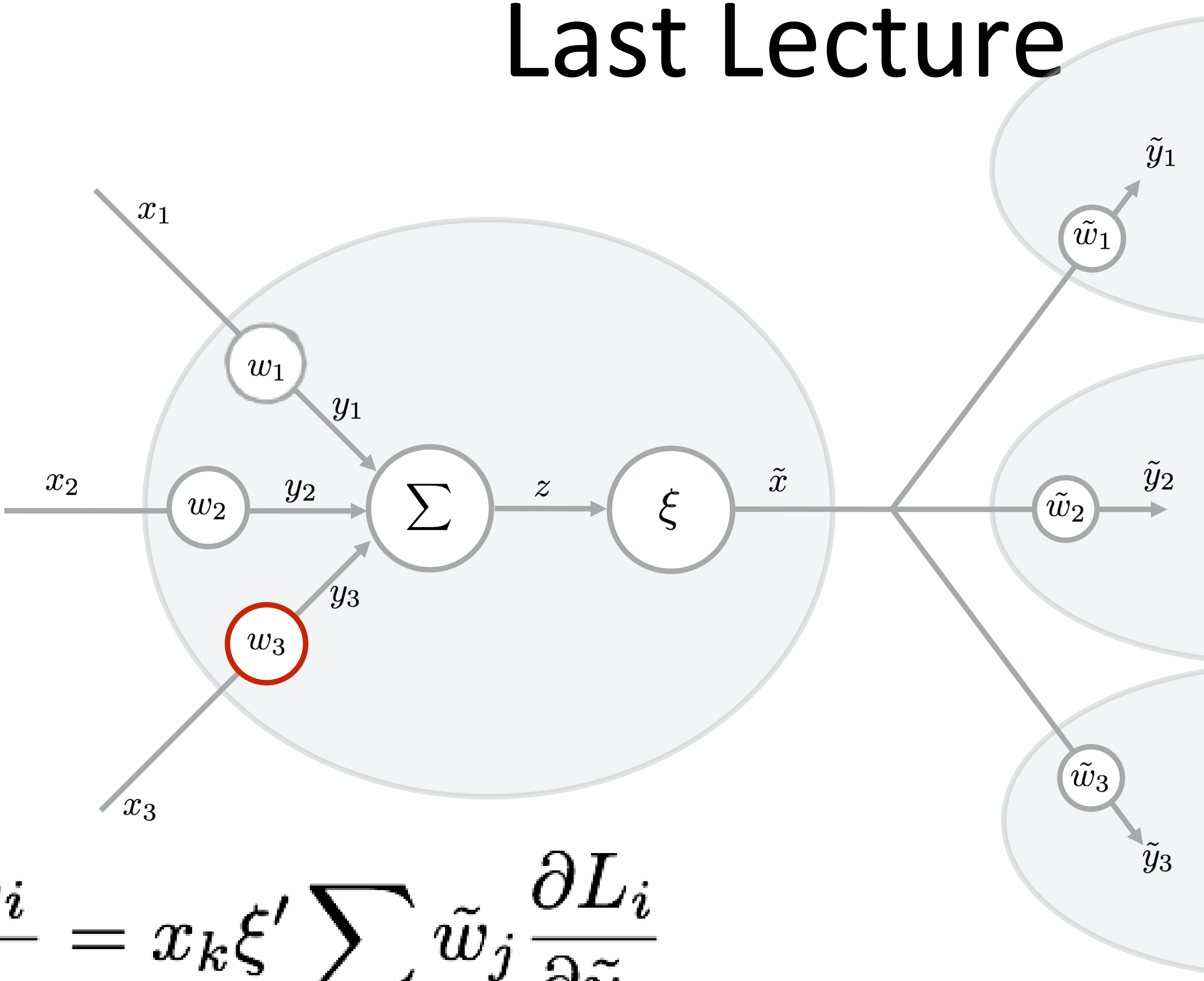
- Optimize all network parameters via gradient descent:

$$\theta^{(k+1)} = \theta^{(k)} - \mu \sum_i \nabla L_i(\theta) \approx \theta^{(k)} - \mu \nabla L_i(\theta)$$

- Compute derivative of partial loss for each parameter `wk`

Training a neural network

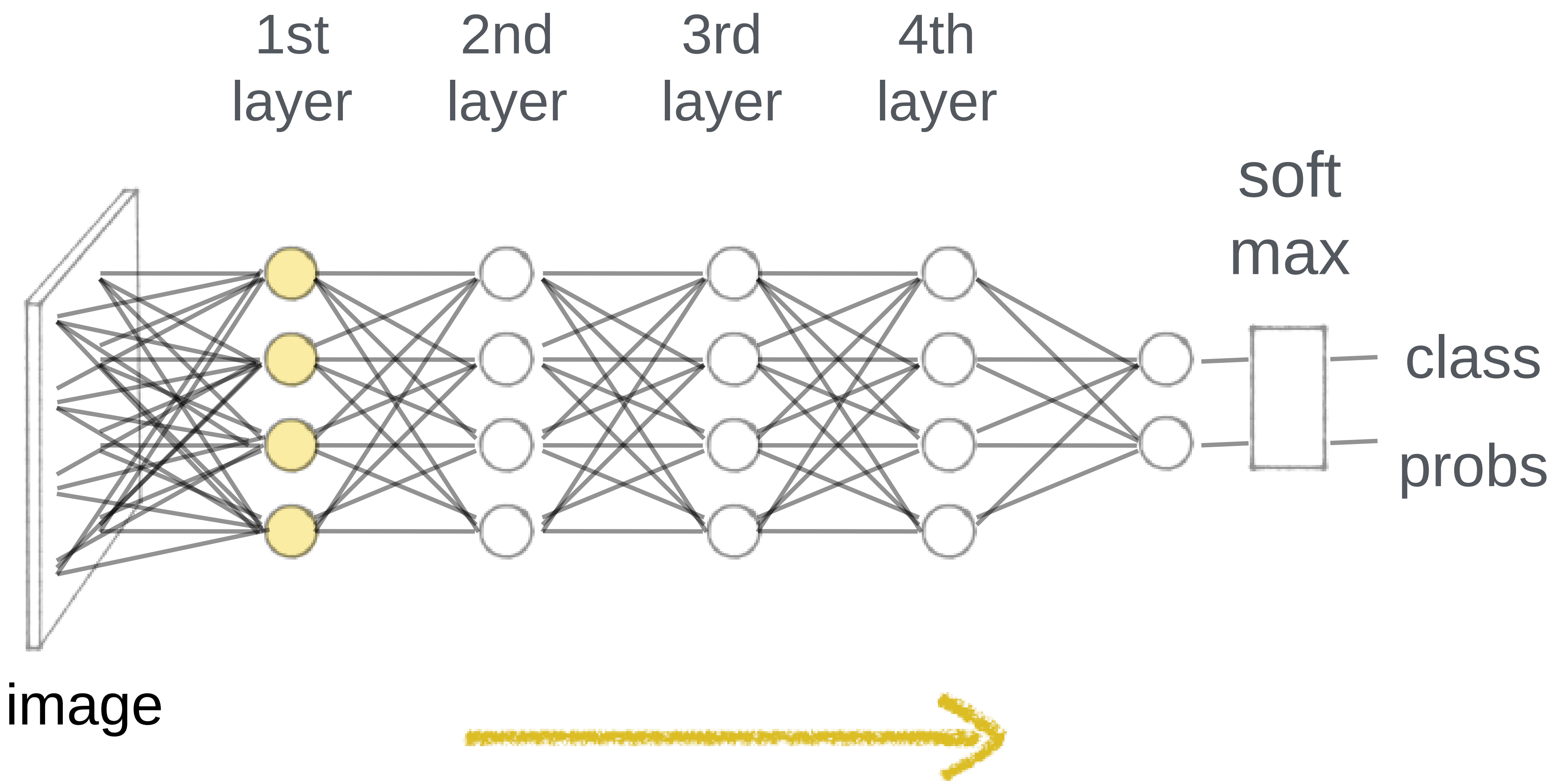
Last Lecture



$$\frac{\partial L_i}{\partial w_k} = x_k \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Backpropagation (backprop)

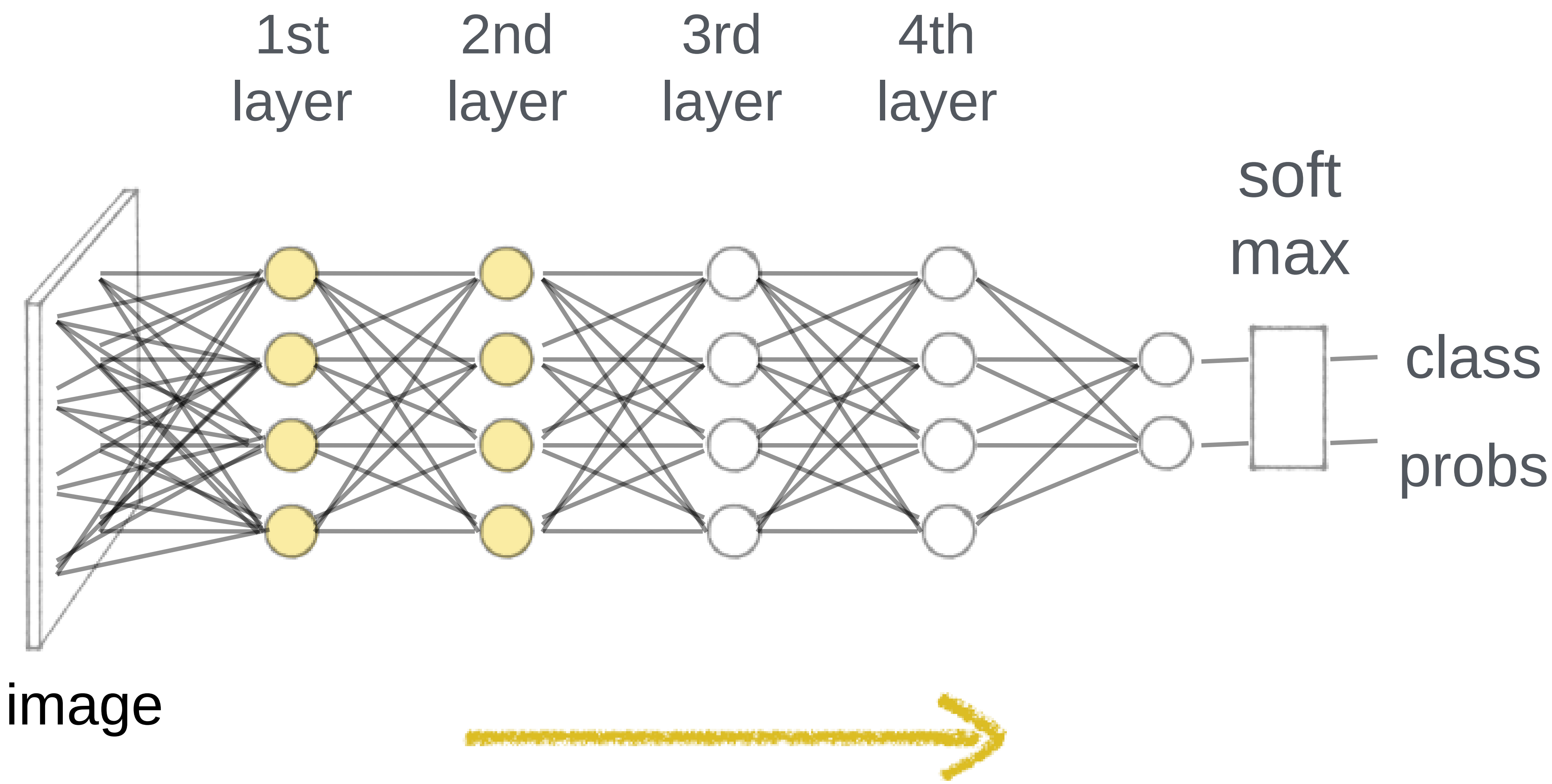
Last Lecture



$$\frac{\partial L_i}{\partial w_k} = \underline{x_k \xi'} \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Backpropagation (backprop)

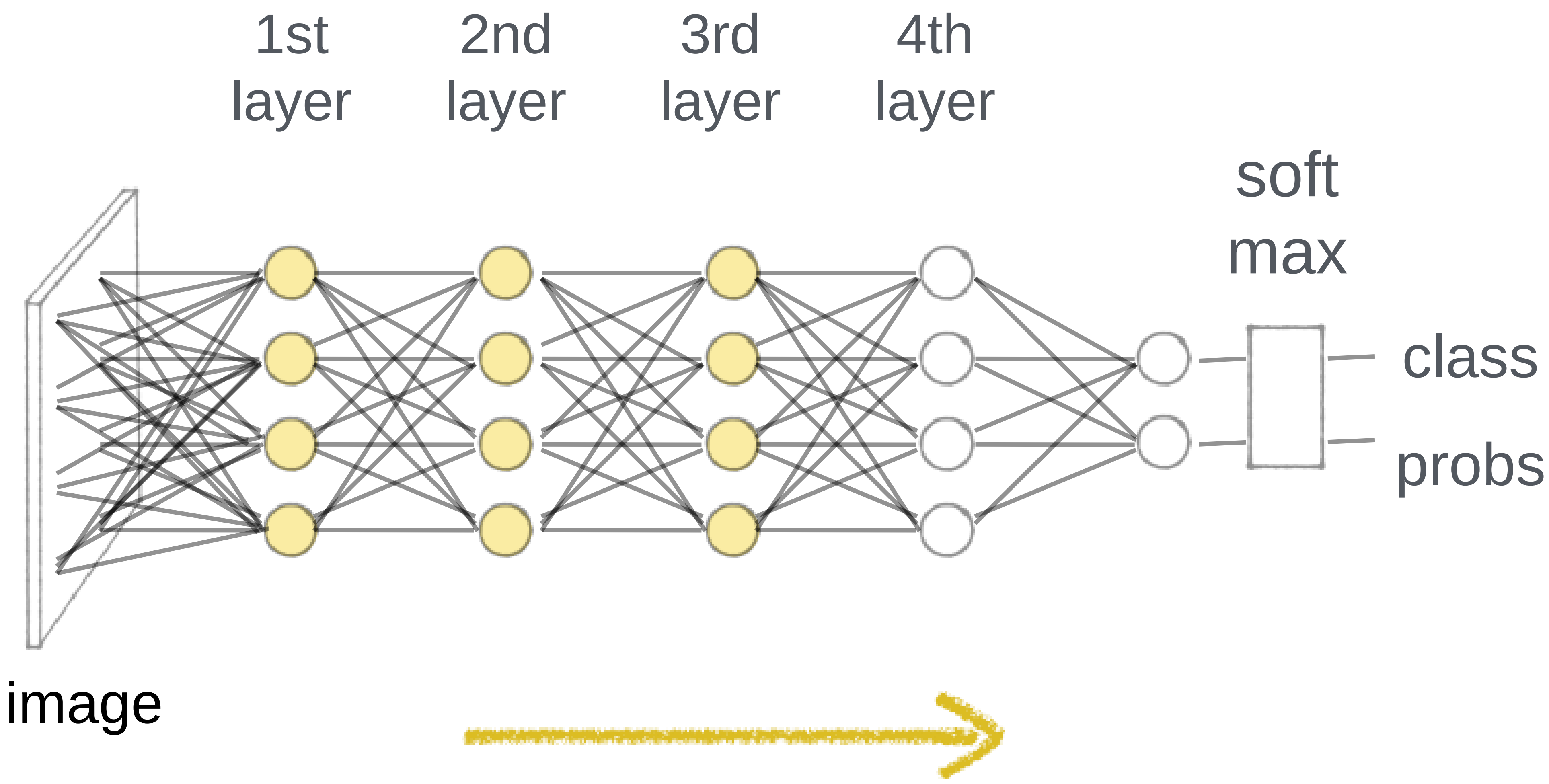
Last Lecture



$$\frac{\partial L_i}{\partial w_k} = \underline{x_k}\xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Backpropagation (backprop)

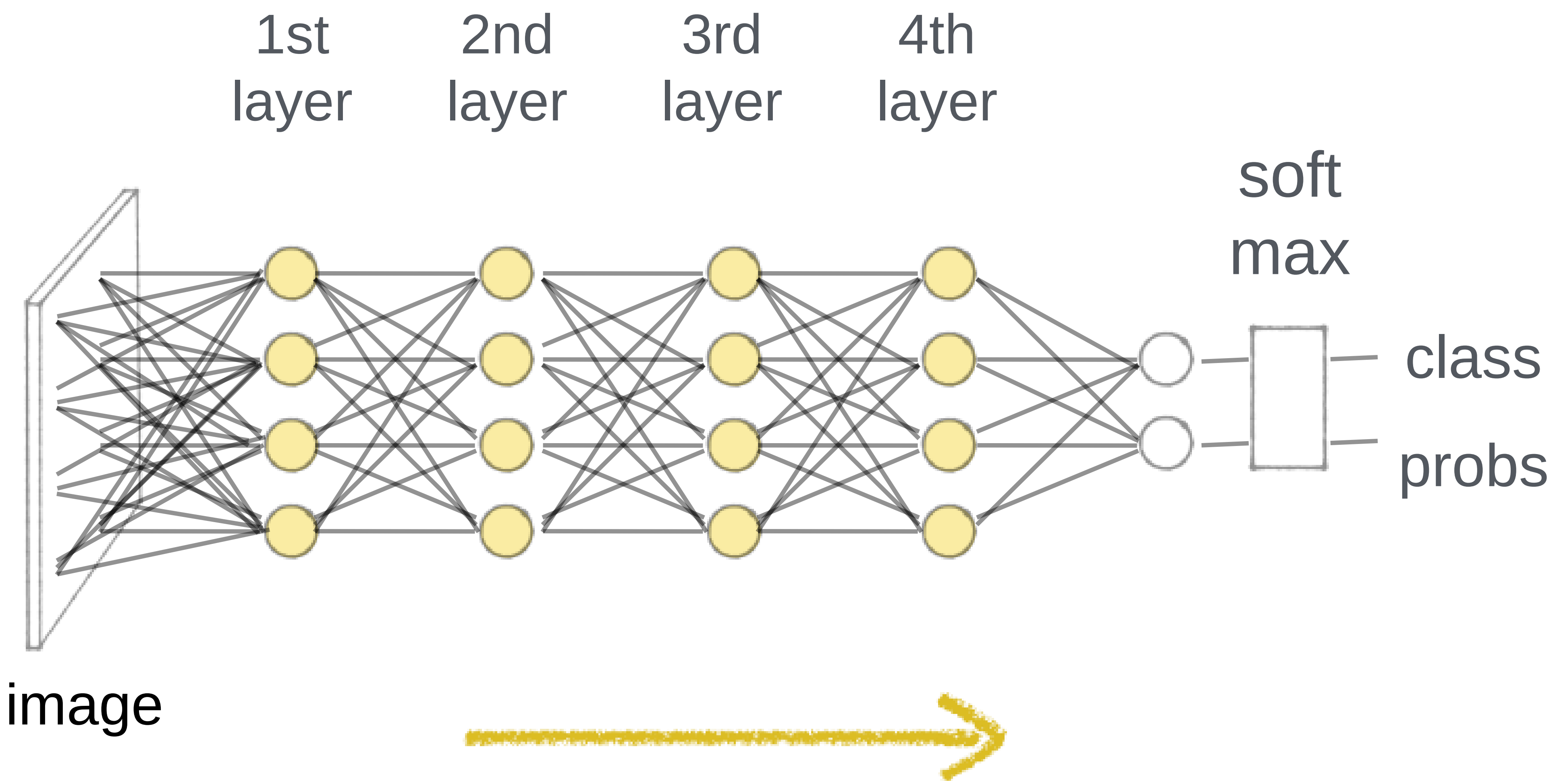
Last Lecture



$$\frac{\partial L_i}{\partial w_k} = \underline{x_k} \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Backpropagation (backprop)

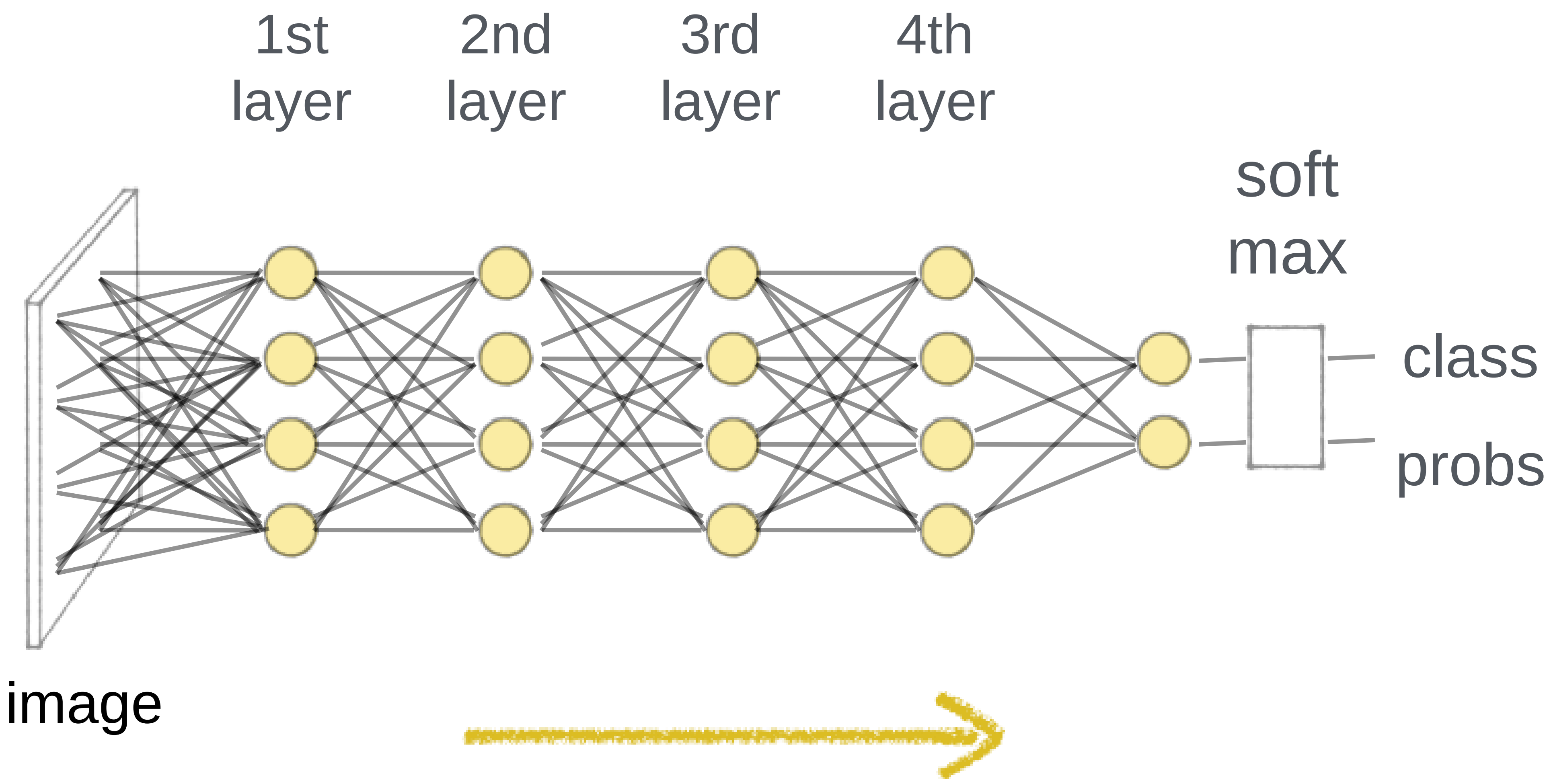
Last Lecture



$$\frac{\partial L_i}{\partial w_k} = \underline{x_k \xi'} \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Backpropagation (backprop)

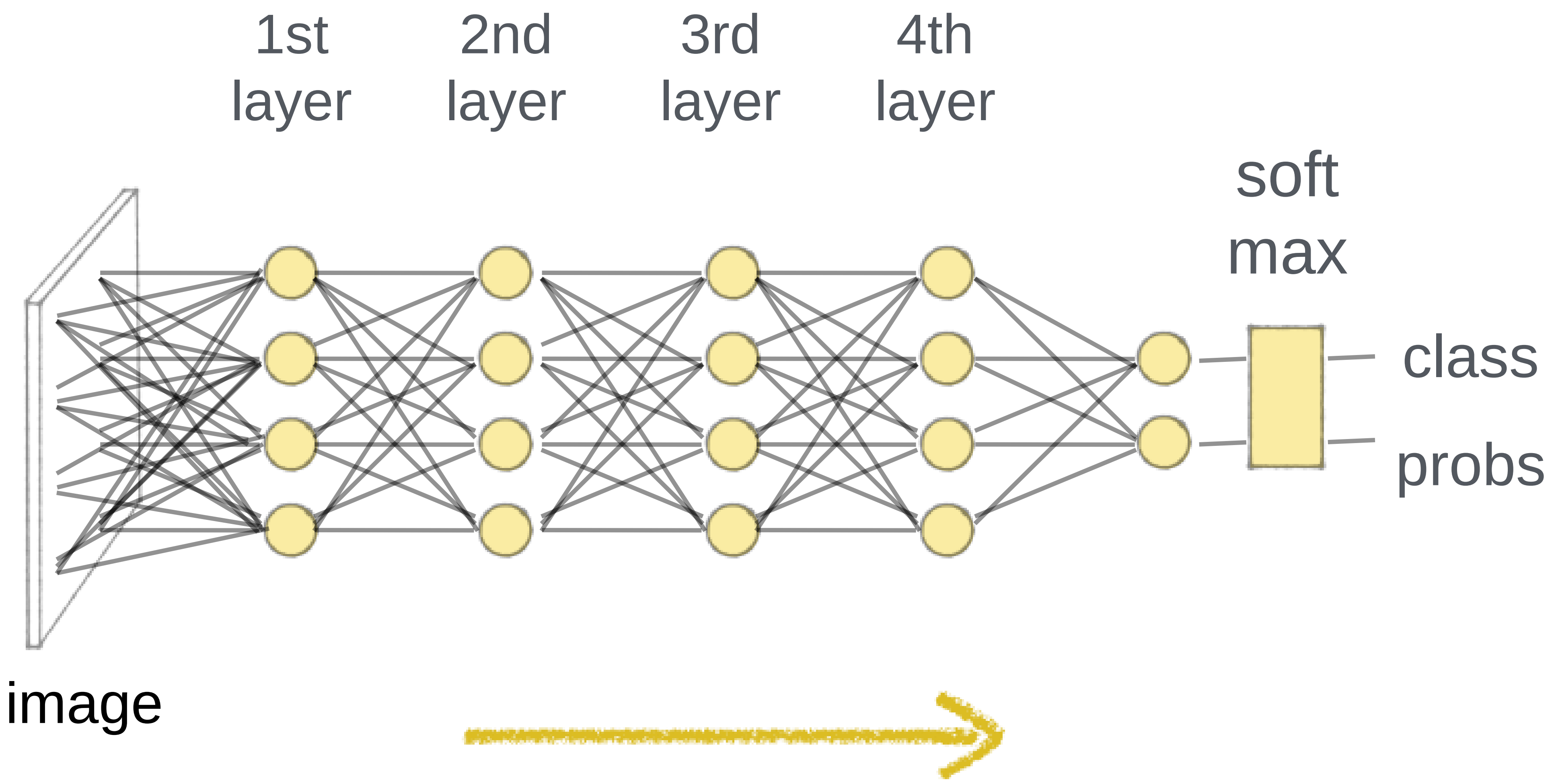
Last Lecture



$$\frac{\partial L_i}{\partial w_k} = \underline{x_k}\xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Backpropagation (backprop)

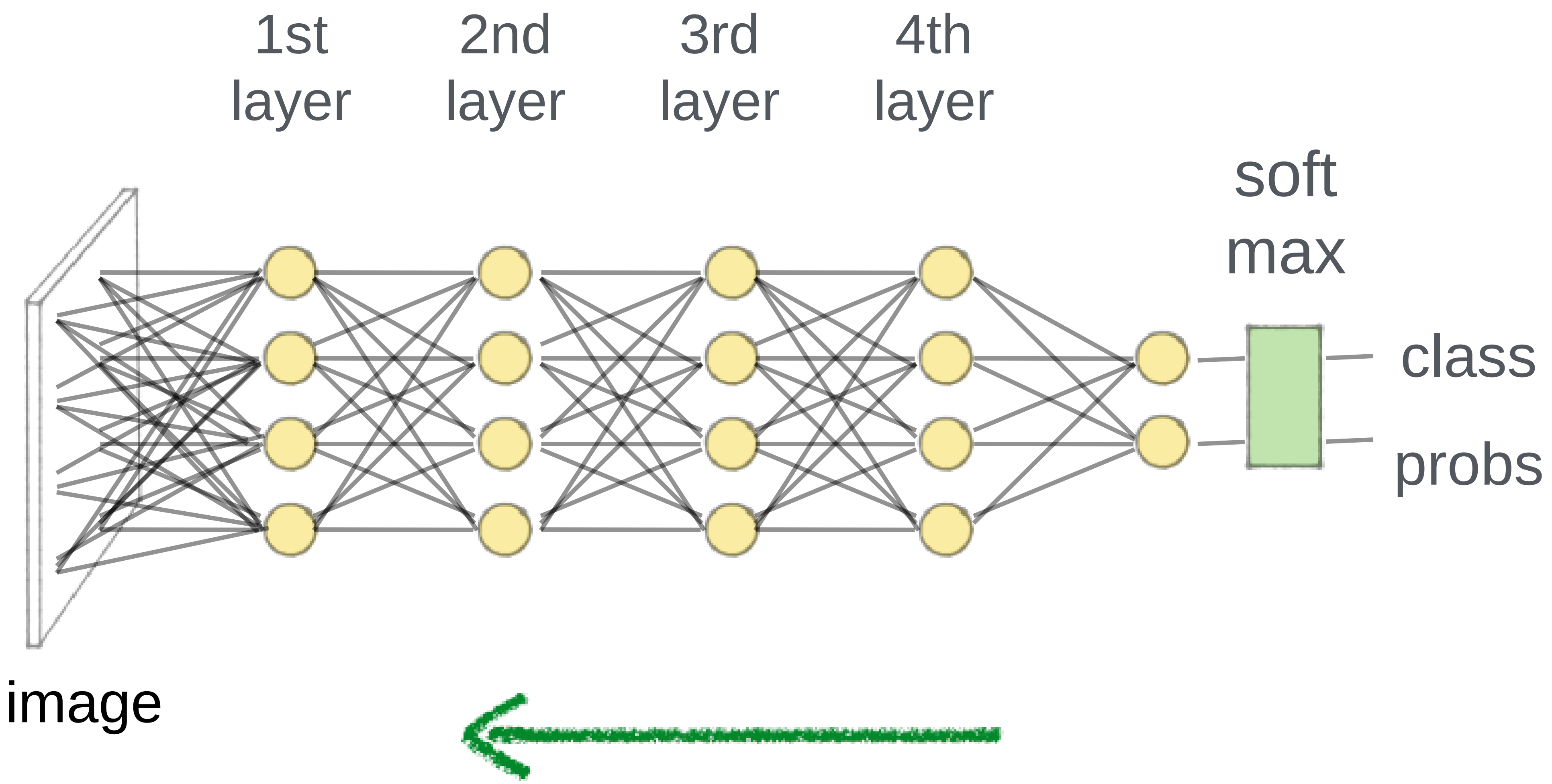
Last Lecture



Backpropagation (backprop)

$$\frac{\partial L_i}{\partial w_k} = \underline{x_k} \xi' \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

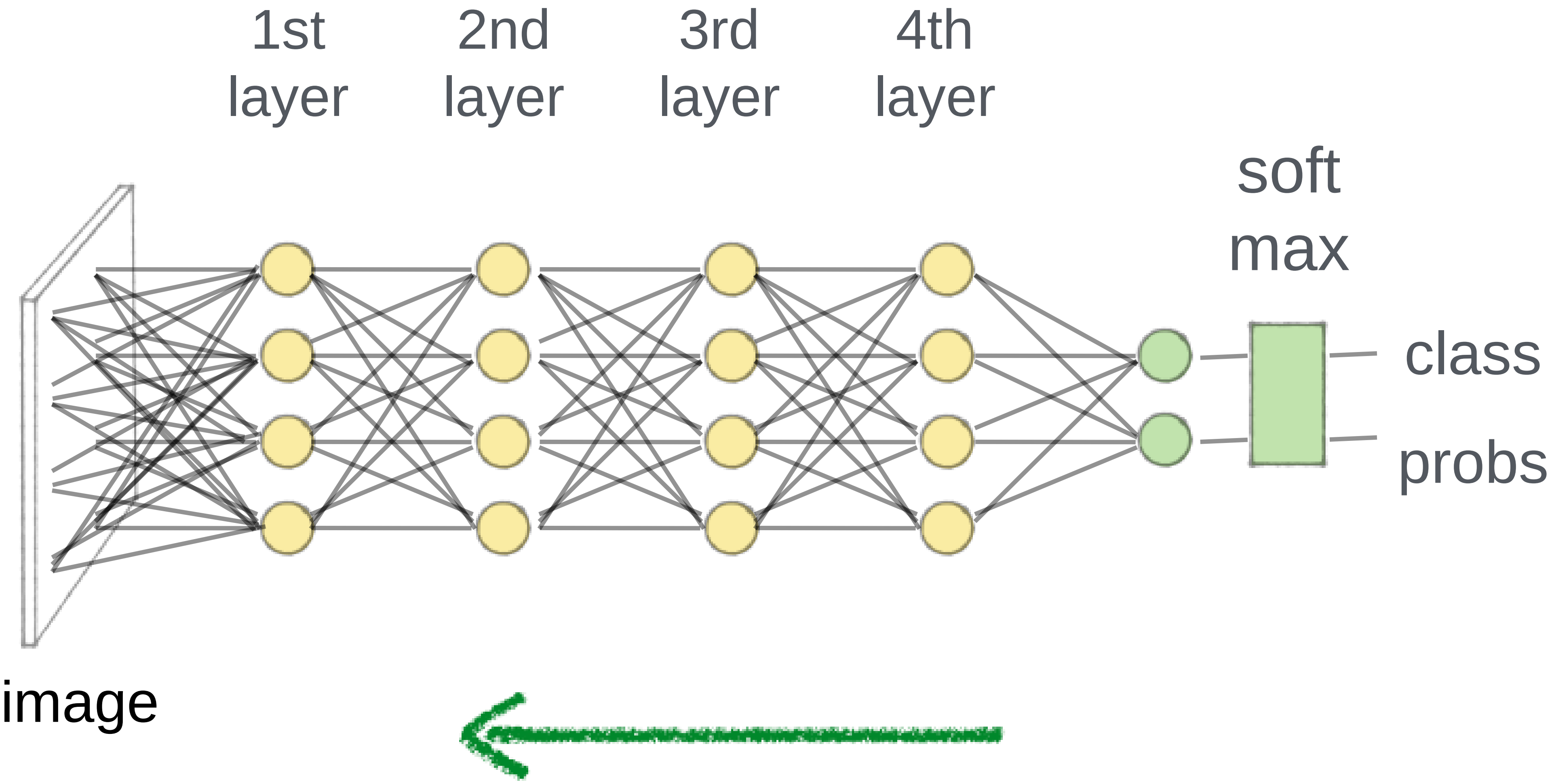
Last Lecture



Backpropagation (backprop)

$$\frac{\partial L_i}{\partial w_k} = \underline{x_k}\,\underline{\xi'} \sum_j \tilde{w}_j \frac{\partial L_i}{\underline{\partial \tilde{y}_j}}$$

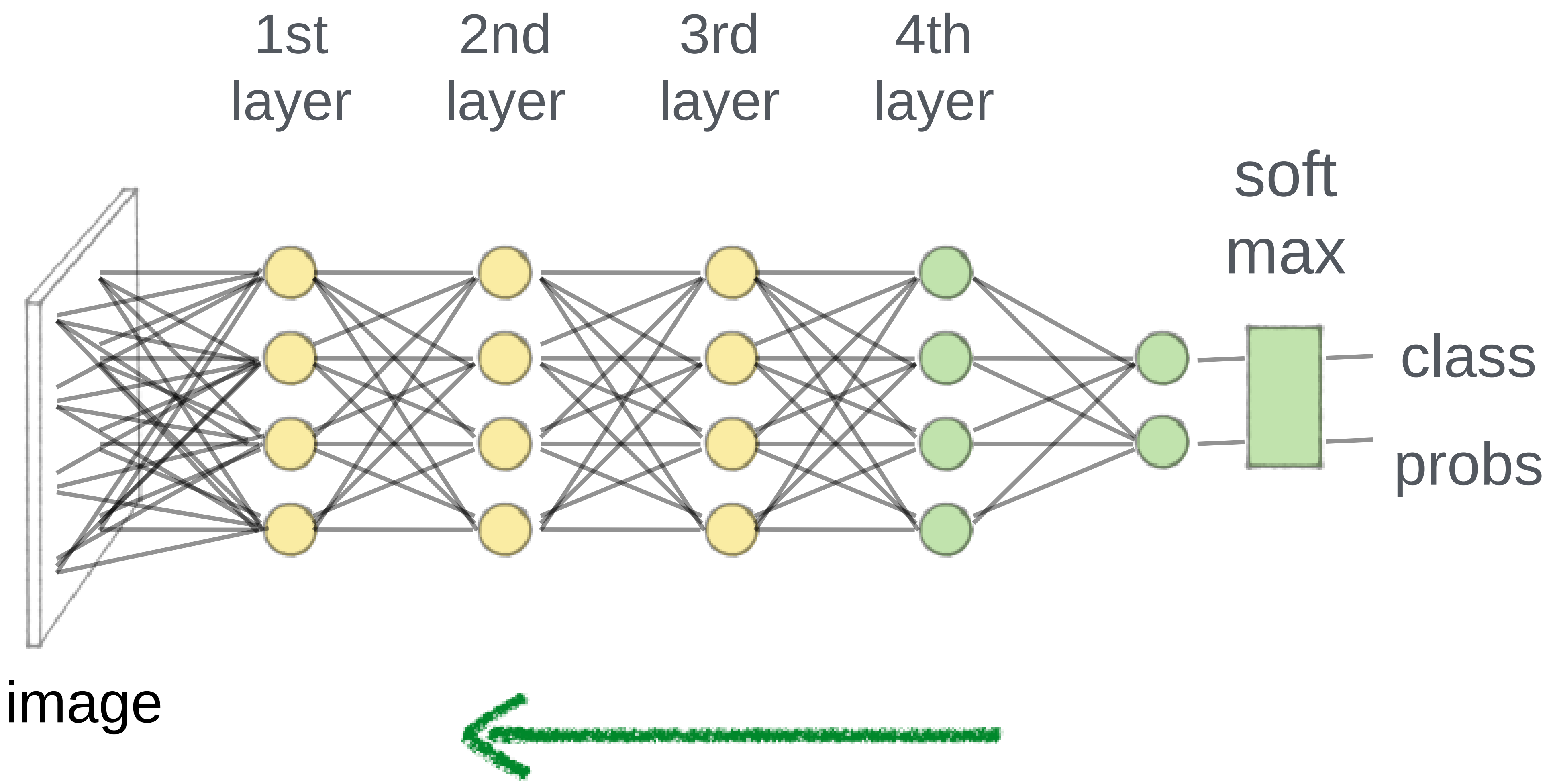
Last Lecture



Backpropagation (backprop)

$$\frac{\partial L_i}{\partial w_k} = \underline{x_k} \, \underline{\xi'} \sum_j \tilde{w}_j \frac{\partial L_i}{\underline{\partial \tilde{y}_j}}$$

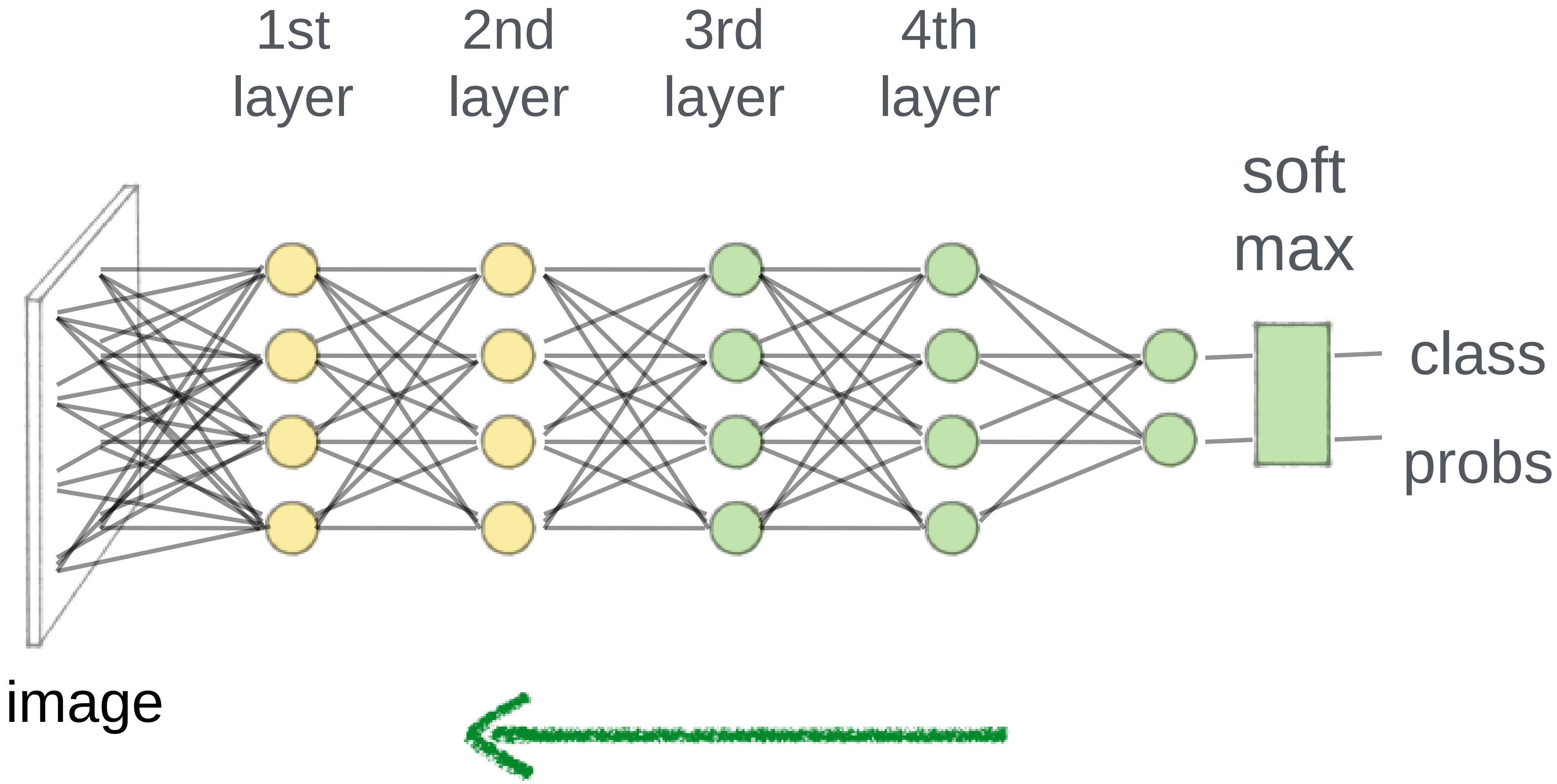
Last Lecture



Backpropagation (backprop)

$$\frac{\partial L_i}{\partial w_k} = \underline{x_k}\,\underline{\xi'} \sum_j \tilde{w}_j \frac{\partial L_i}{\underline{\partial \tilde{y}_j}}$$

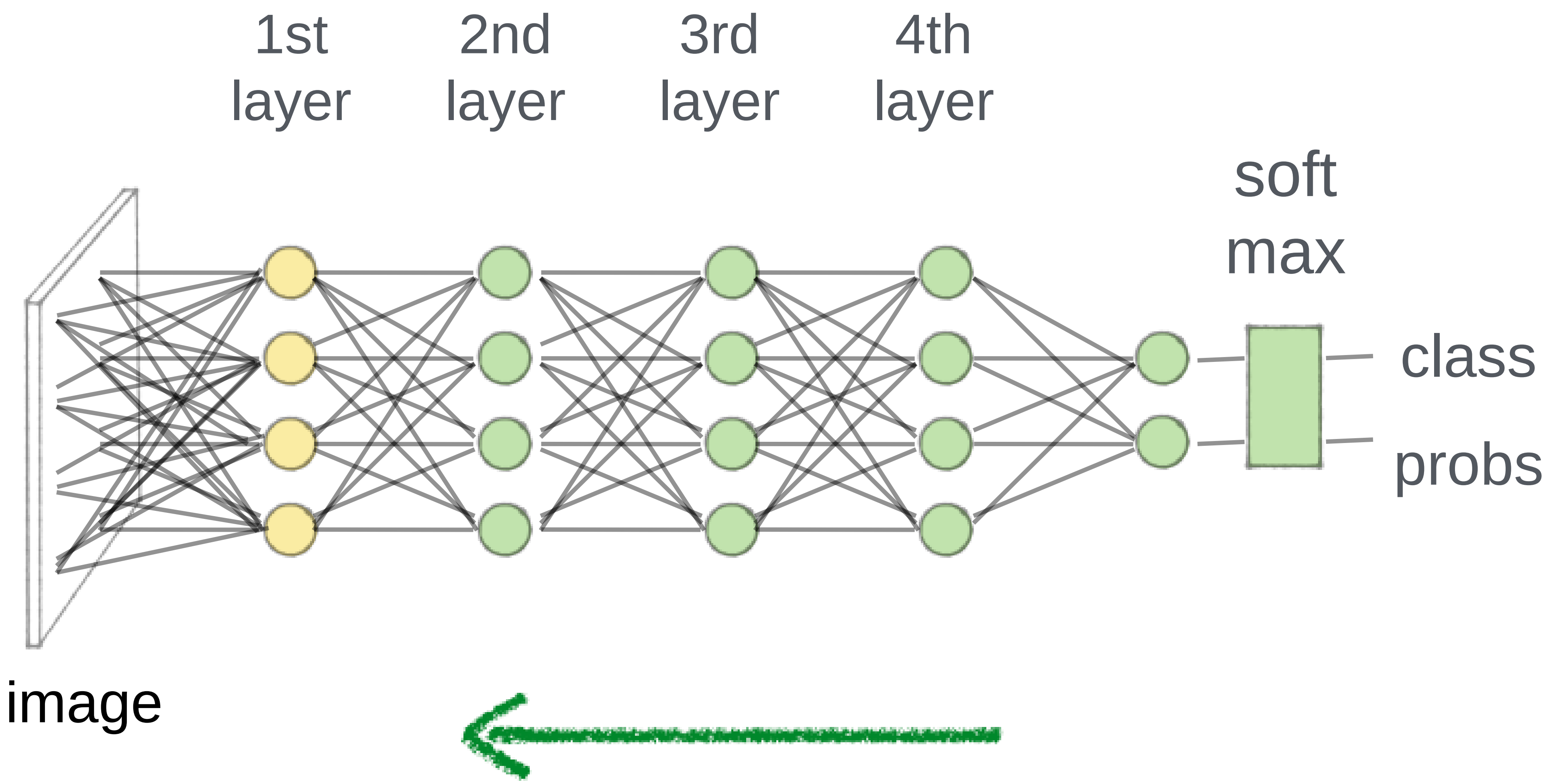
Last Lecture



$$\frac{\partial L_i}{\partial w_k} = \underline{x_k}\,\underline{\xi'}\sum_j \tilde{w}_j \frac{\partial L_i}{\underline{\partial \tilde{y}_j}}$$

Backpropagation (backprop)

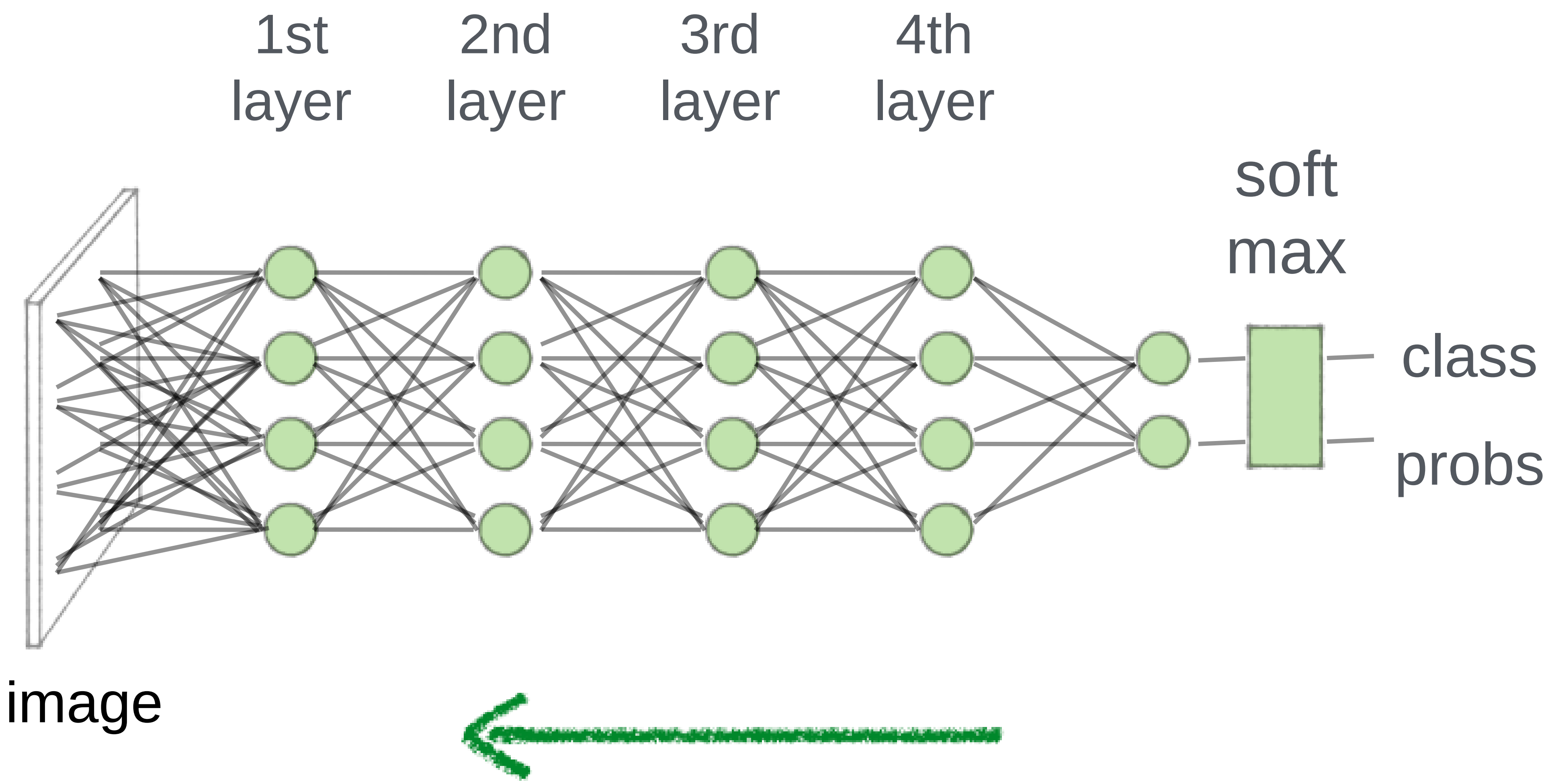
Last Lecture



Backpropagation (backprop)

$$\frac{\partial L_i}{\partial w_k} = \underline{x_k \xi'} \sum_j \tilde{w}_j \frac{\partial L_i}{\partial \tilde{y}_j}$$

Last Lecture



Backpropagation (backprop)

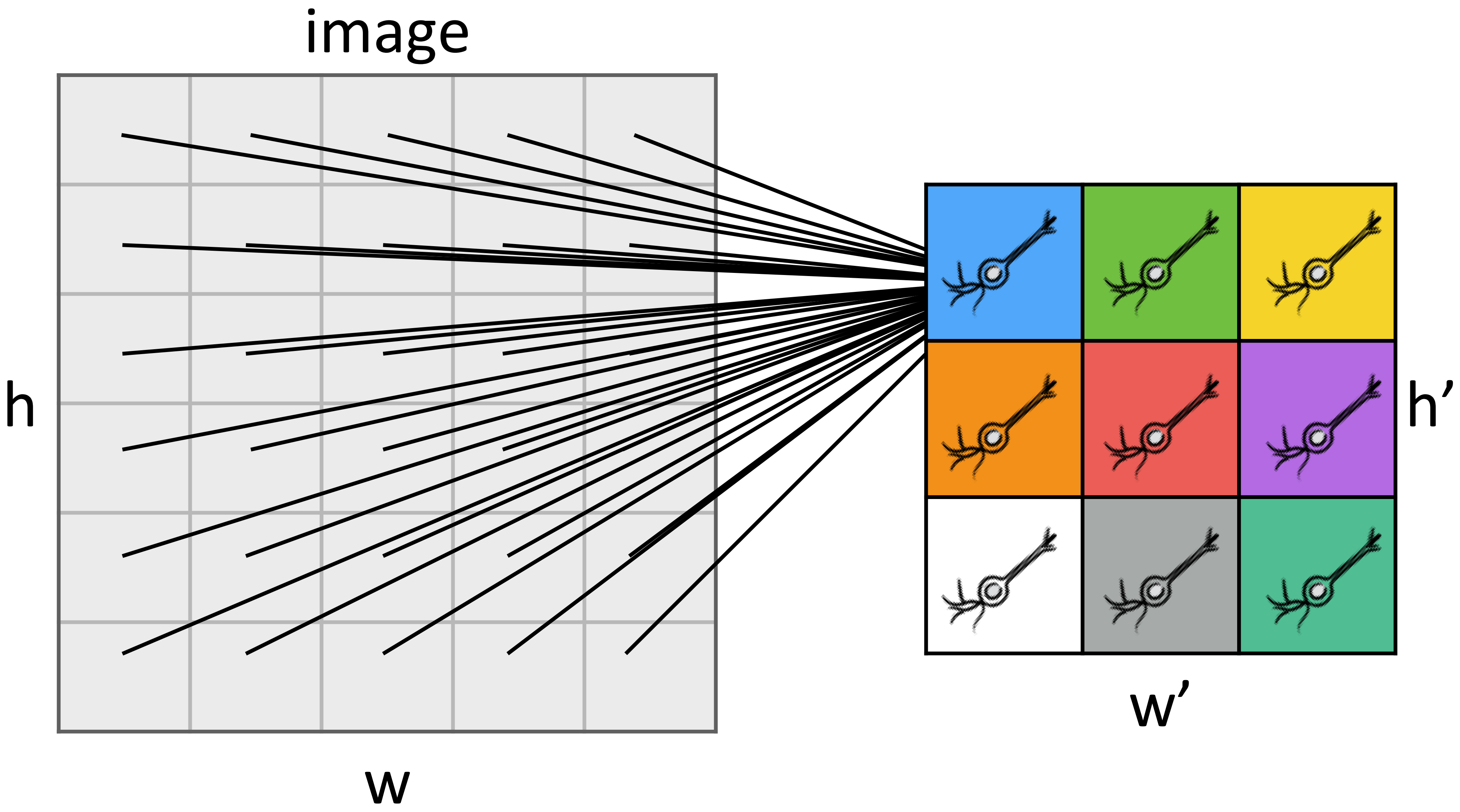
$$\frac{\partial L_i}{\partial w_k} = \underline{x_k}\,\underline{\xi'}\sum_j \tilde{w}_j \frac{\partial L_i}{\underline{\partial \tilde{y}_j}}$$

Today

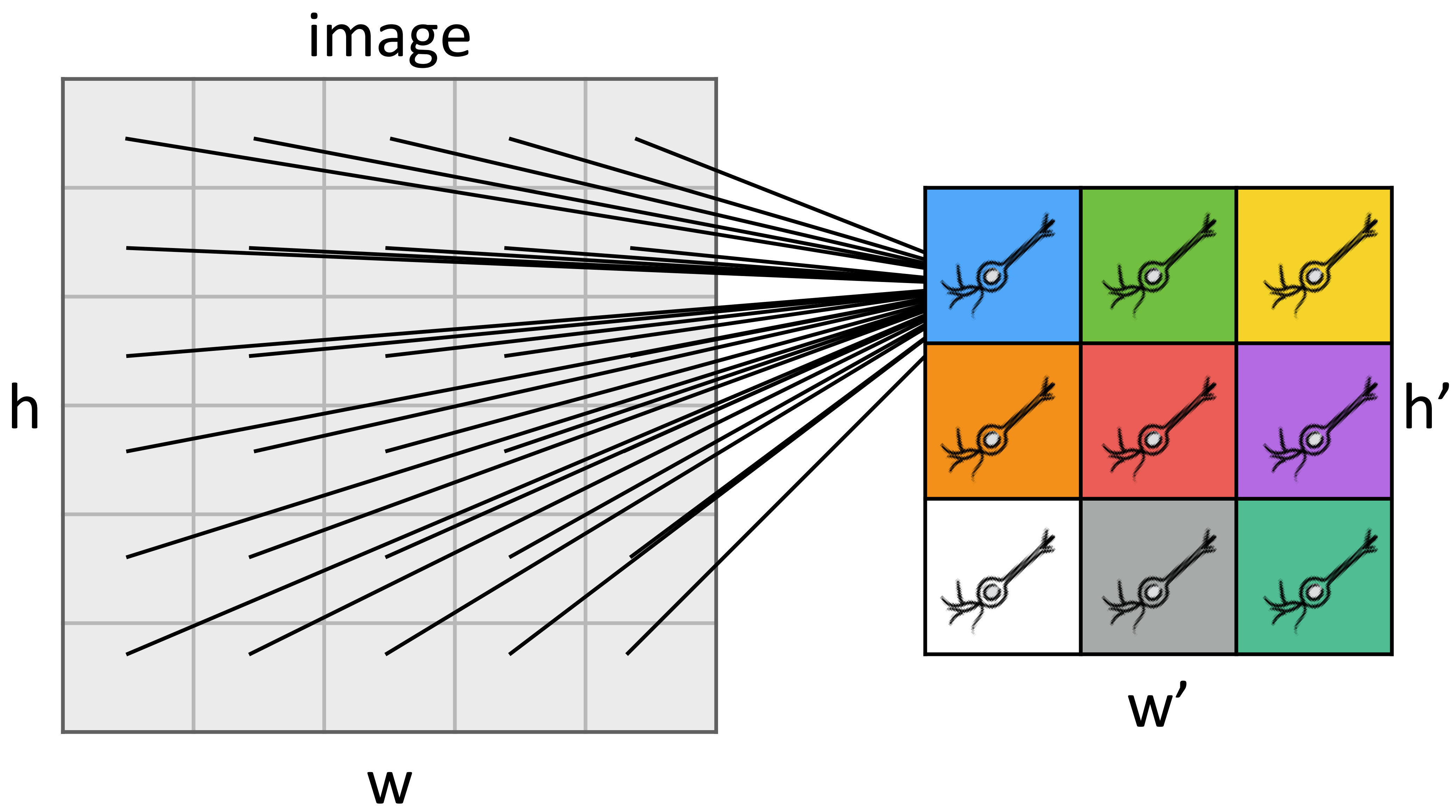
- Recap: Convolutional Neural Networks
- Fully Convolutional Neural Networks
- Convolutional Layers
- Overfitting, Part II

Convolutional Neural Networks

Fully Connected Layers

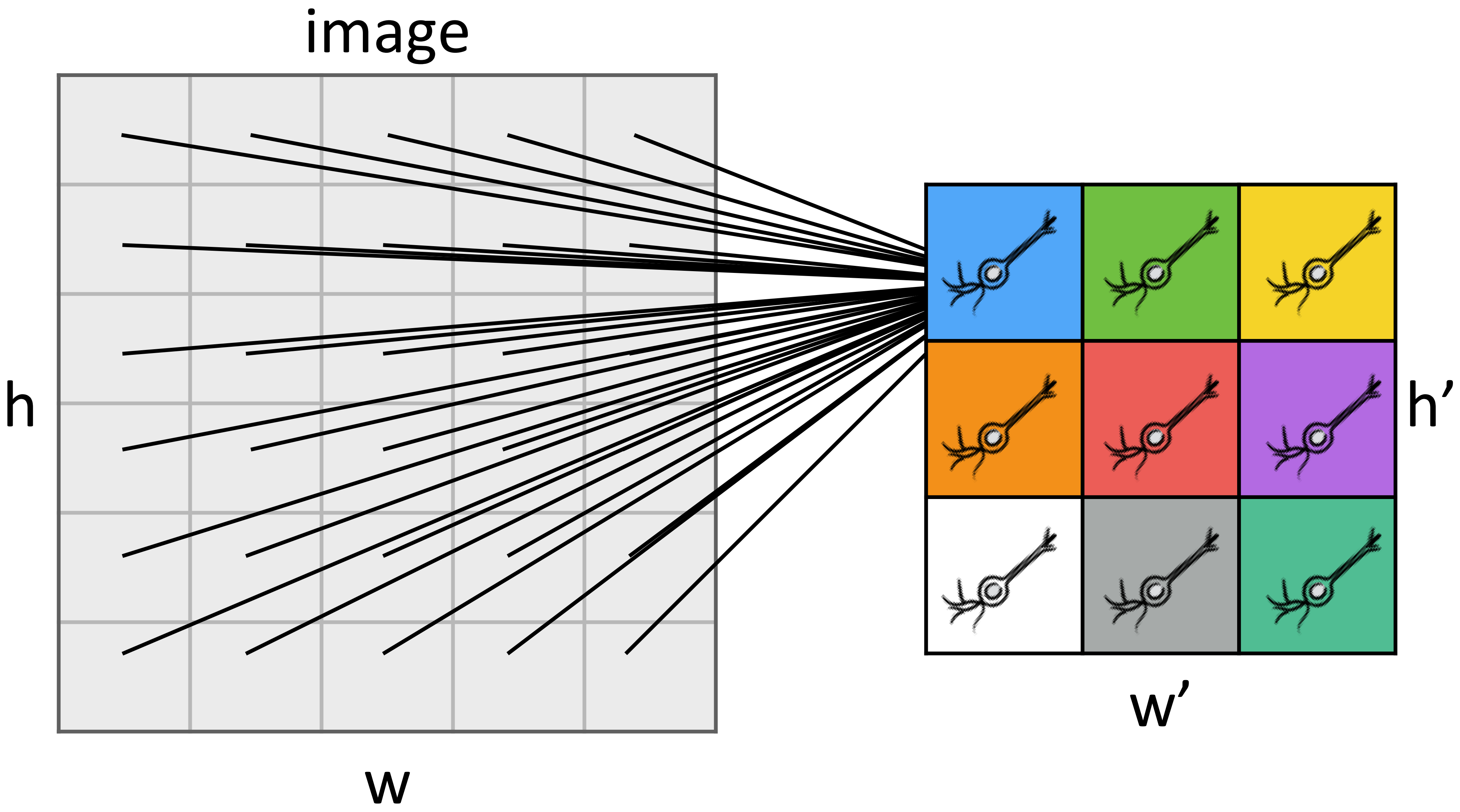


Fully Connected Layers



Number parameters: $w \cdot h \cdot w' \cdot h'$

Fully Connected Layers

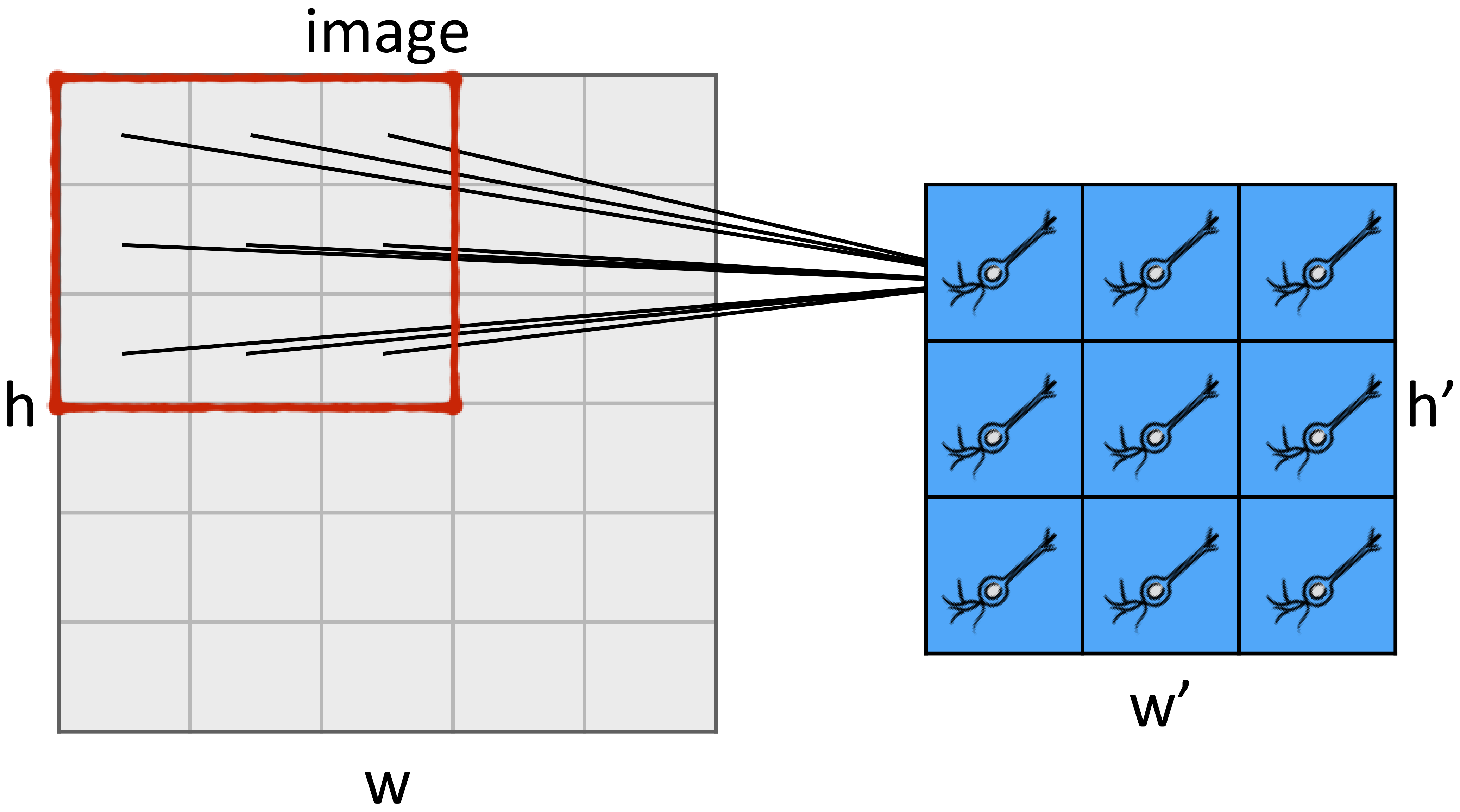


Redundancy

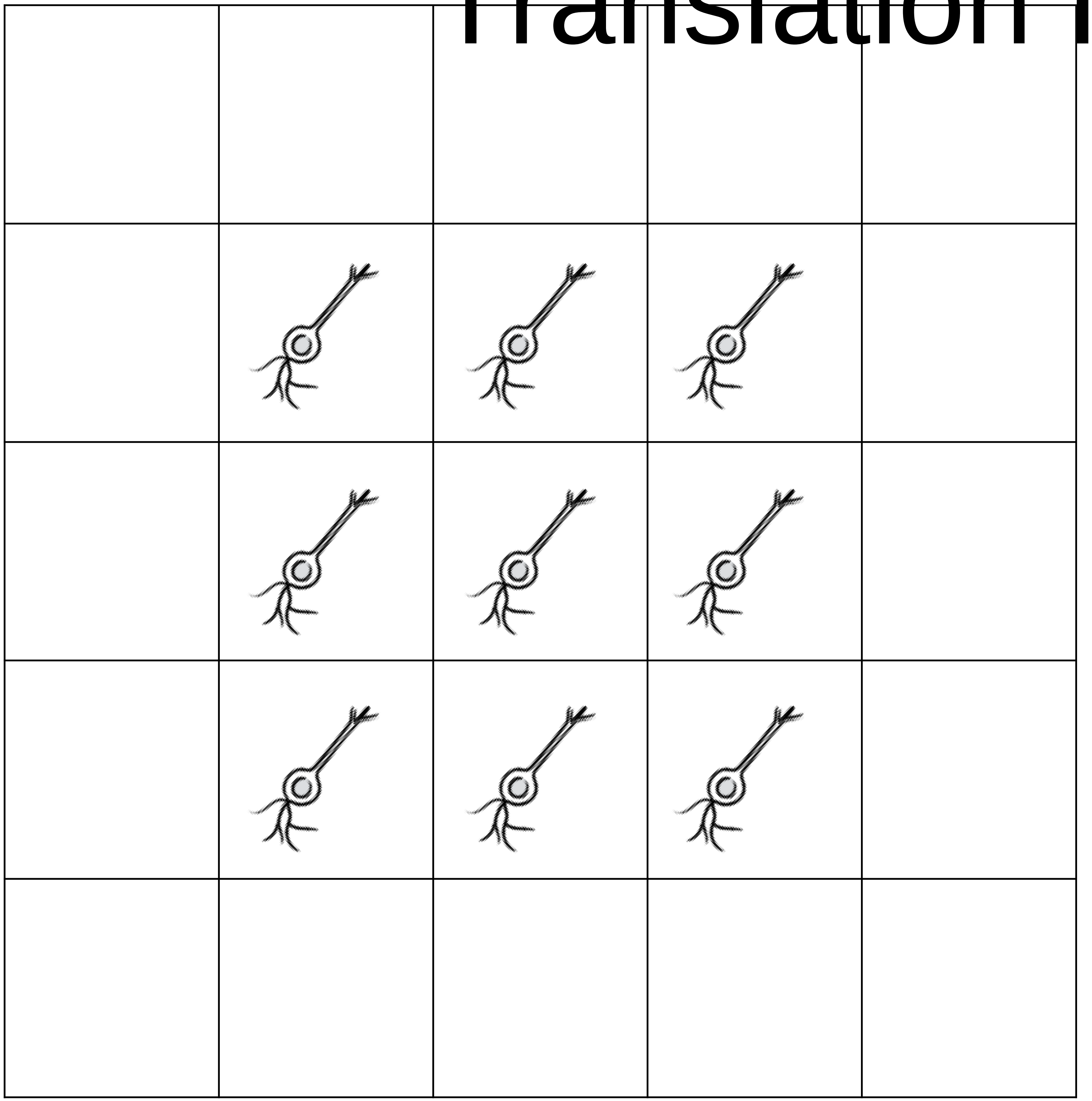
Number parameters: $w \cdot h \cdot w' \cdot h'$

But no locality

Convolutional Layers

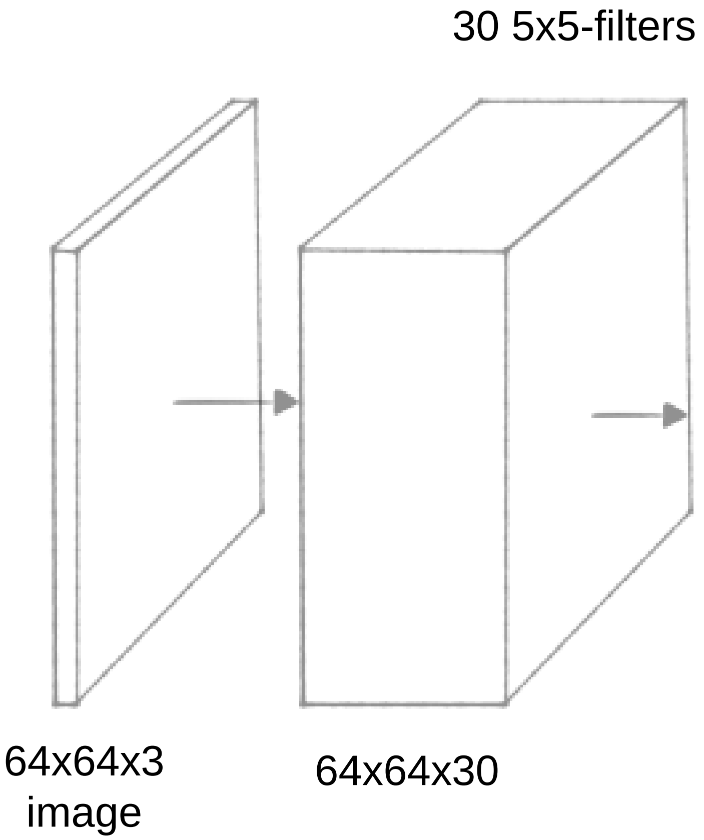


Translation Invariance



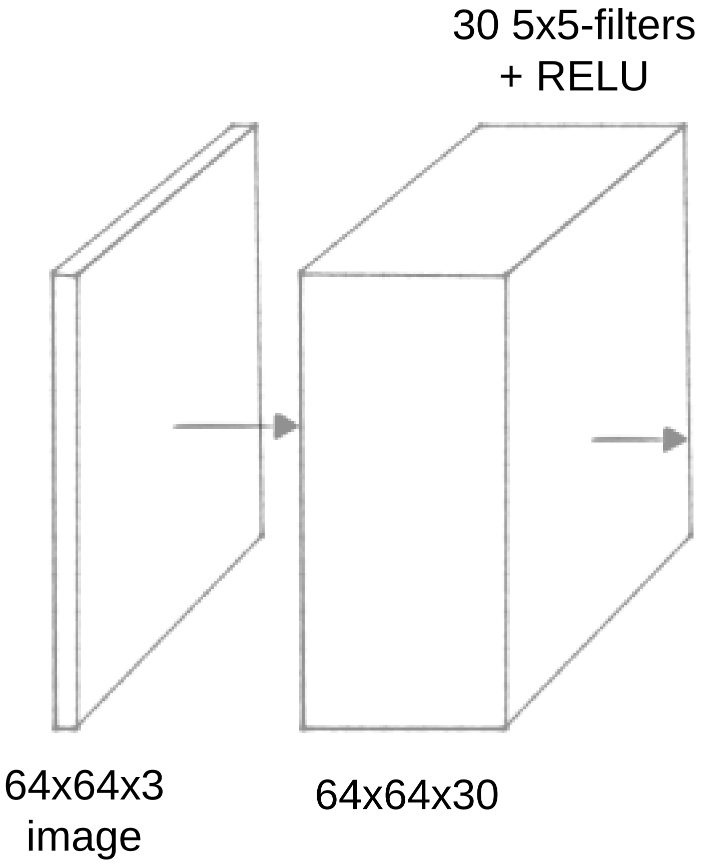
$$I \star \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Network Structure



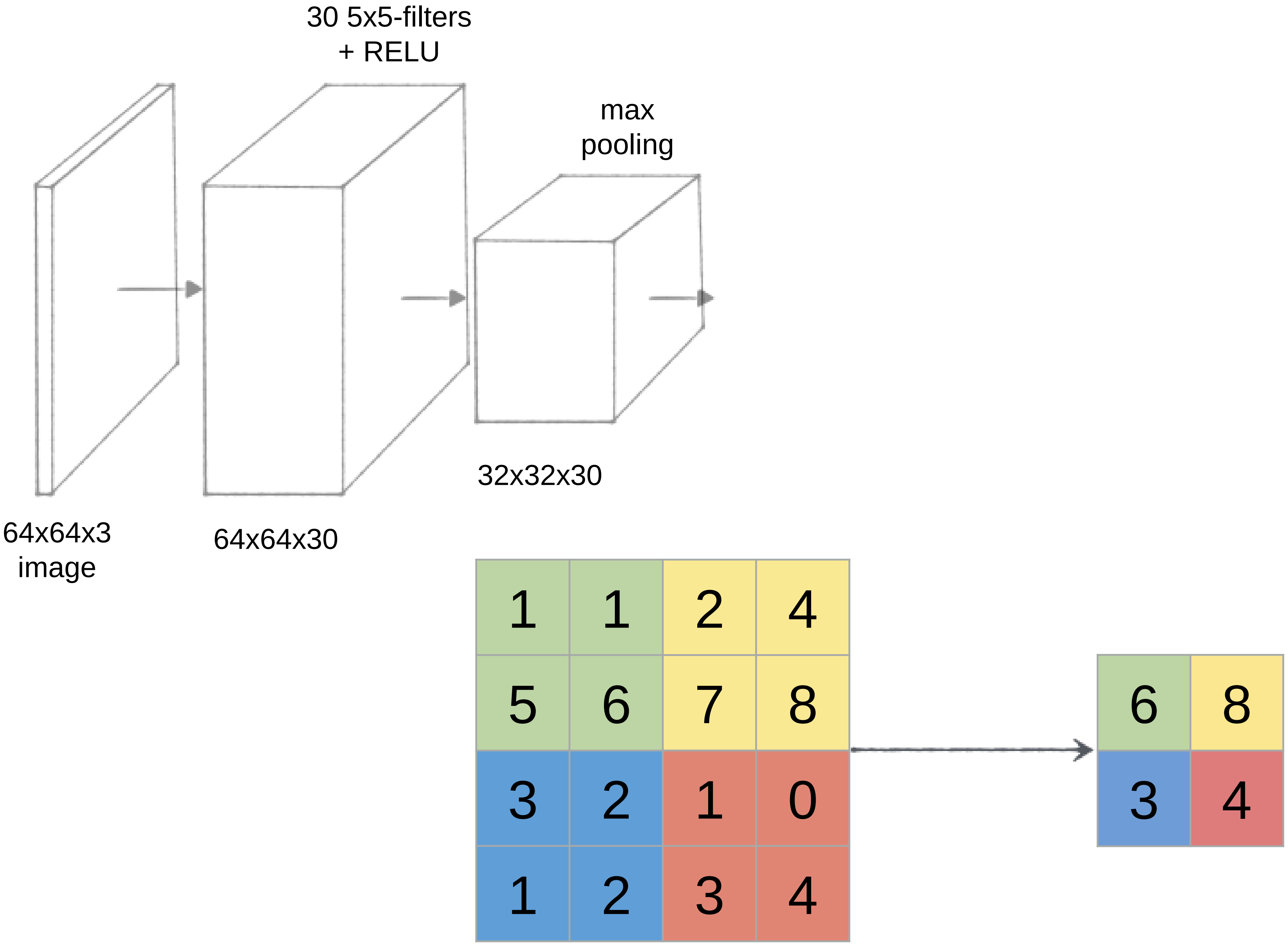
$$I \star w$$

Network Structure



$$\max\{I \star w, 0\}$$

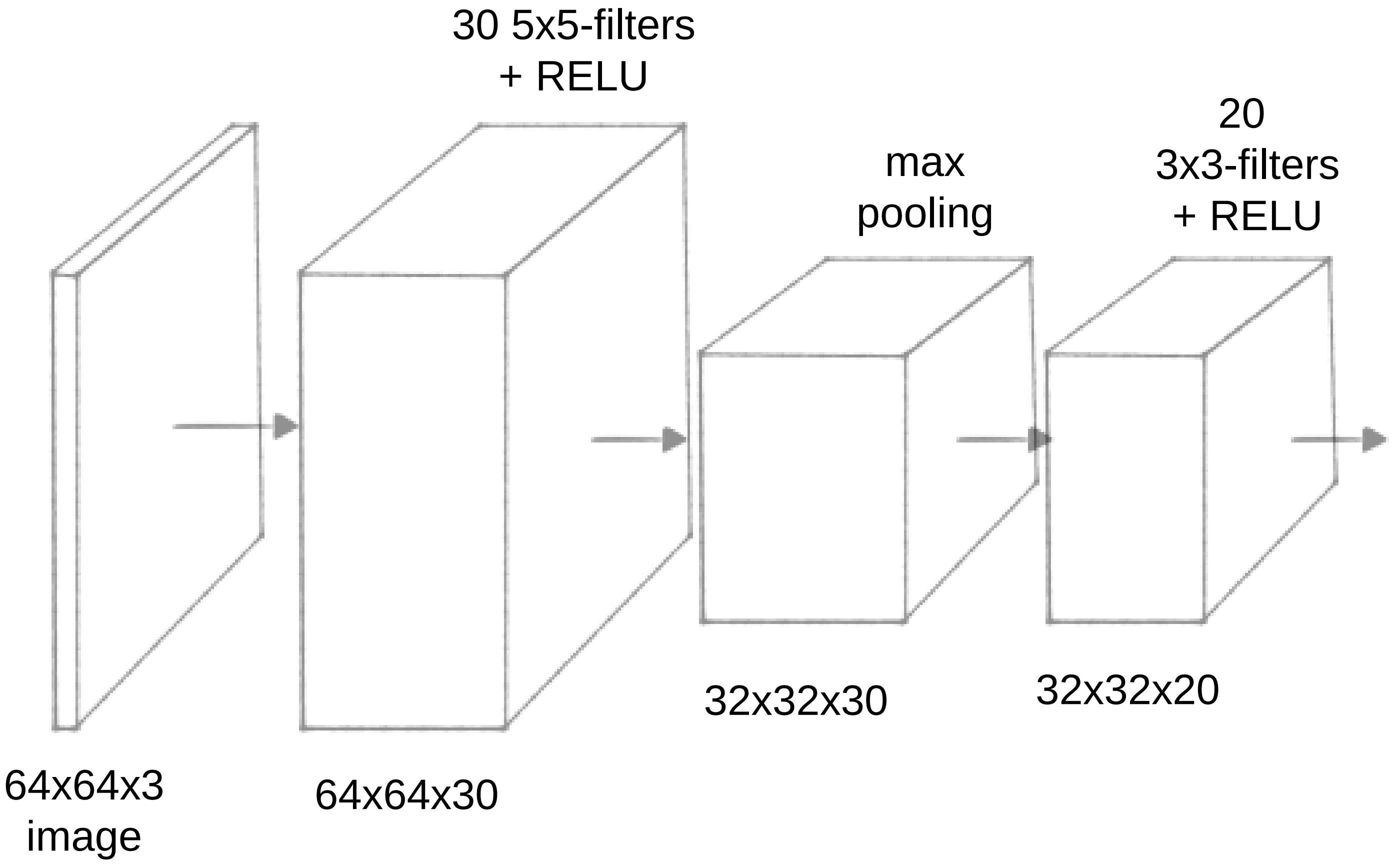
Network Structure



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

6	8
3	4

Network Structure



30 5x5-filters
+ RELU

max
pooling

20
3x3-filters
+ RELU

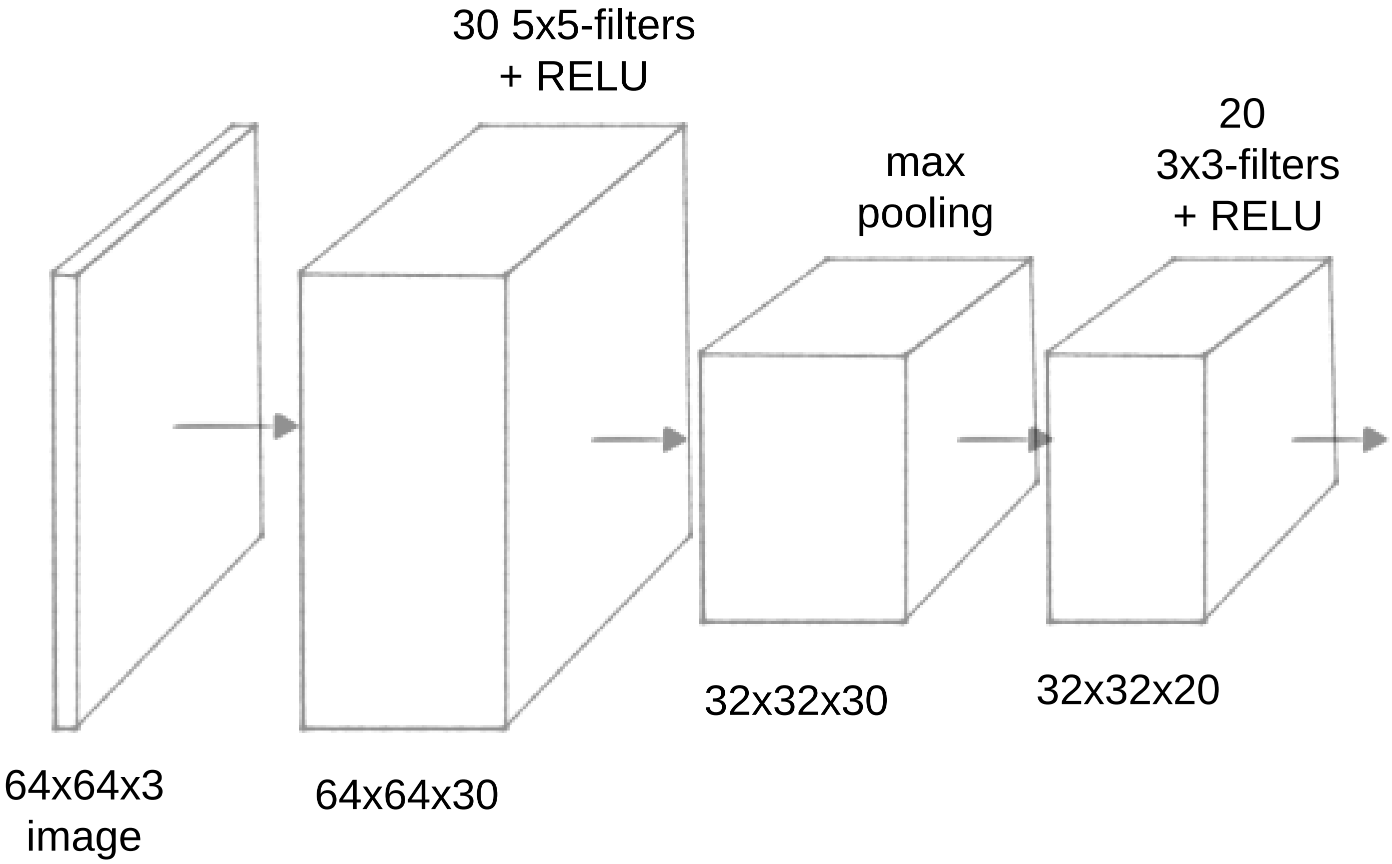
64x64x3
image

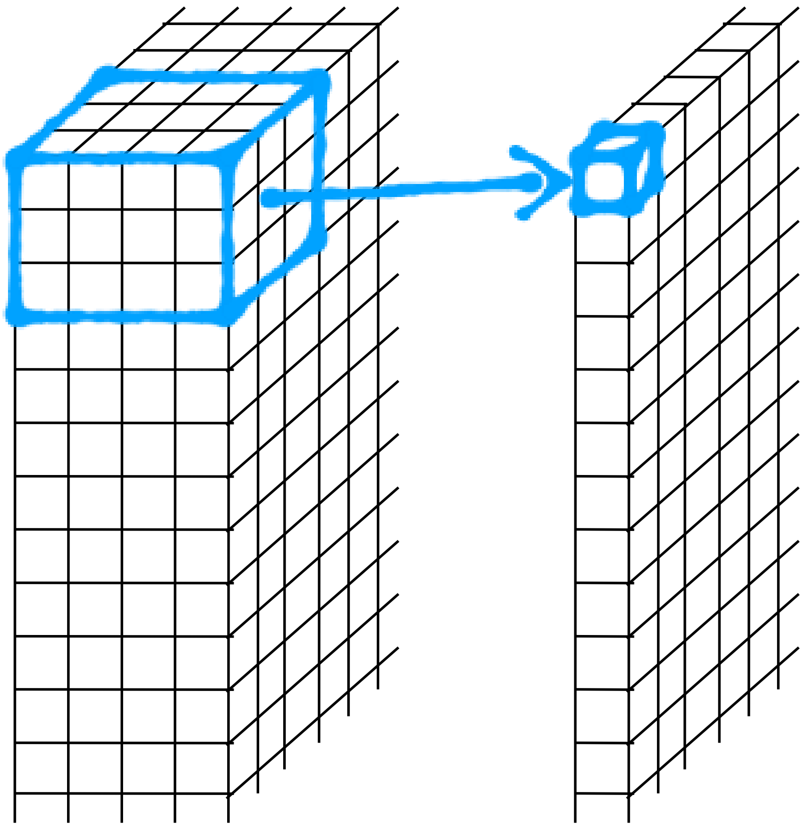
64x64x30

32x32x30

32x32x20

Network Structure

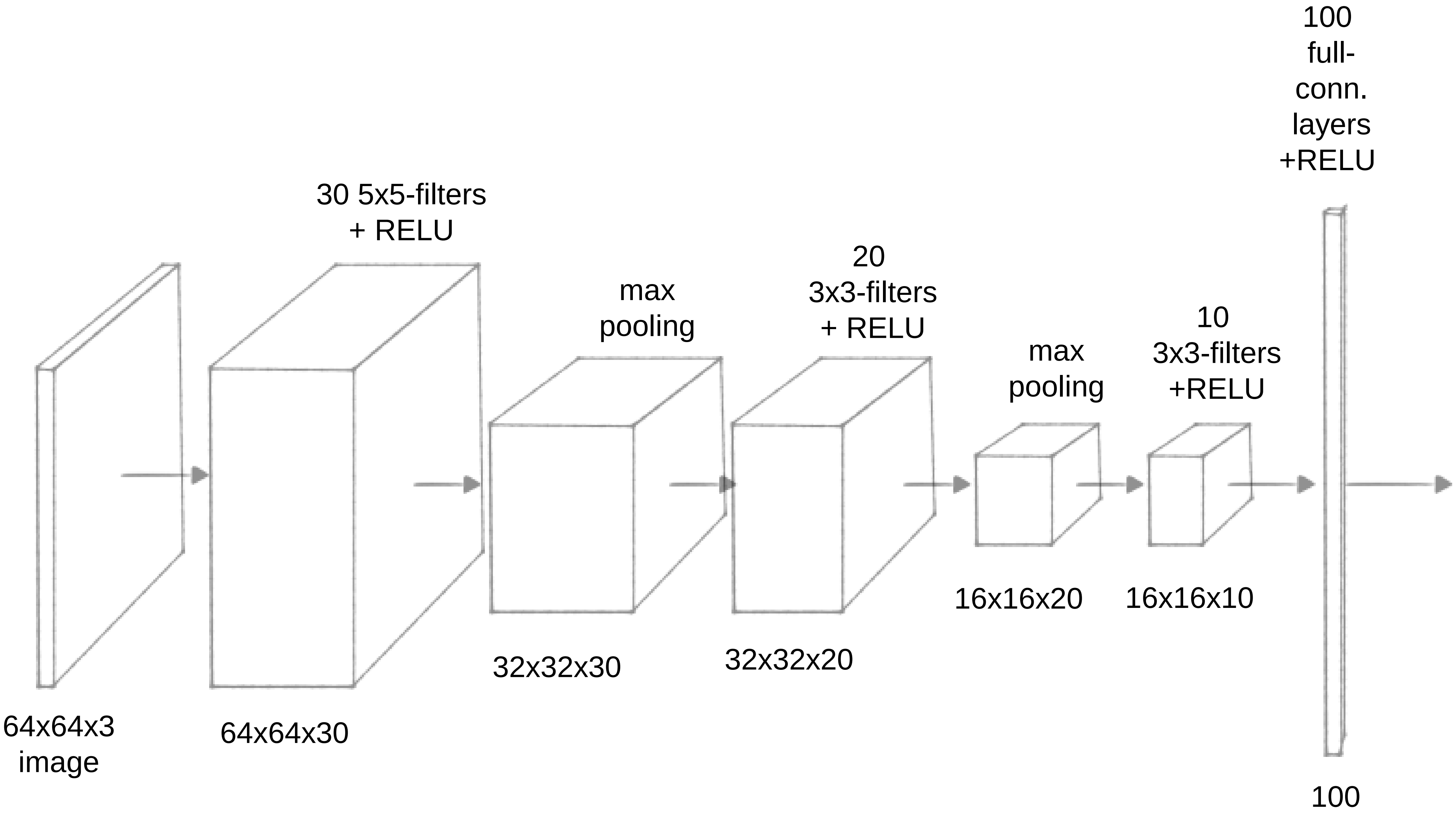




implemented as matrix-vector multiplication:

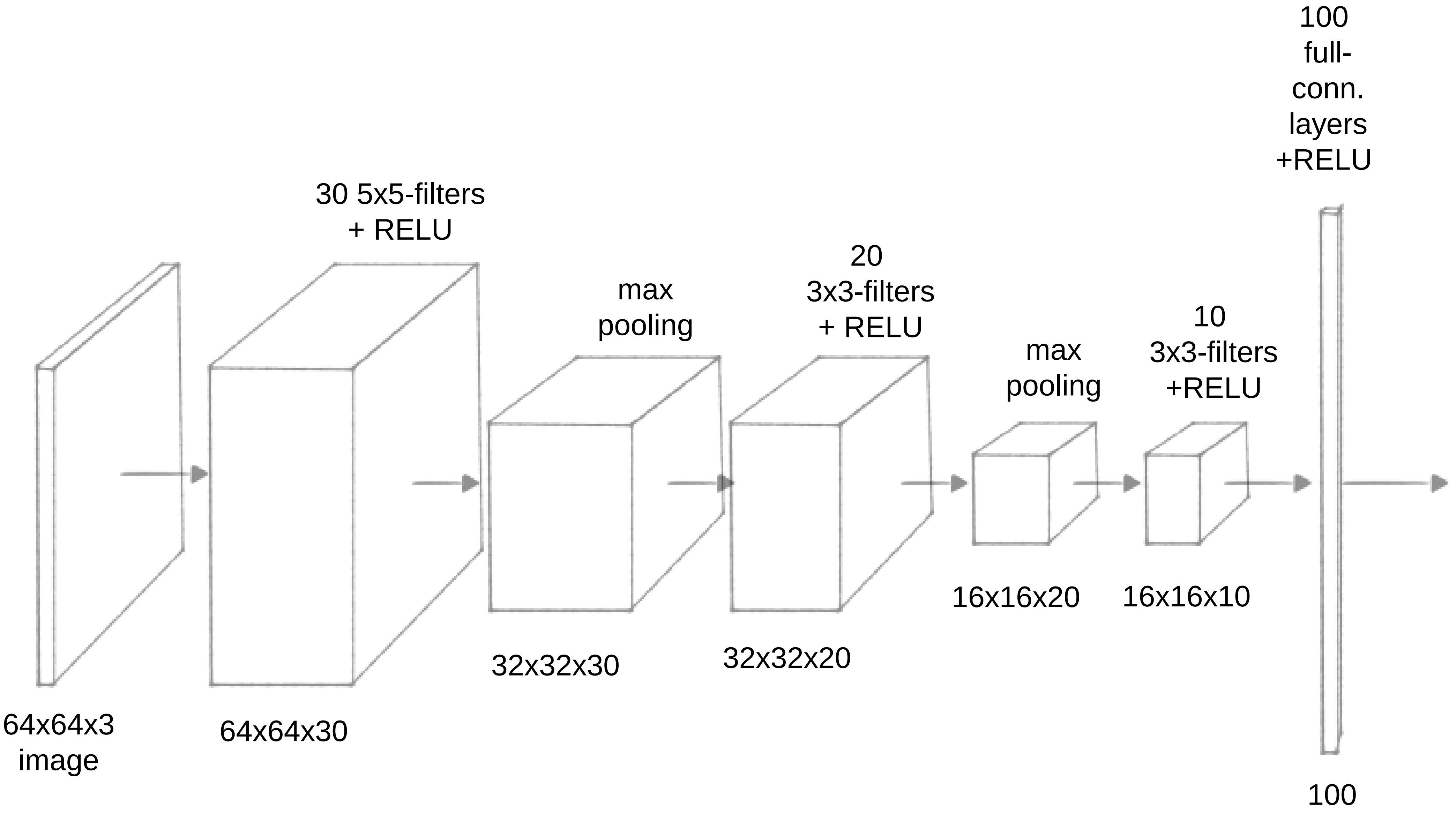
$$\mathrm{vec}(\mathcal{J}) = F\mathrm{vec}(\mathcal{I})$$

Network Structure



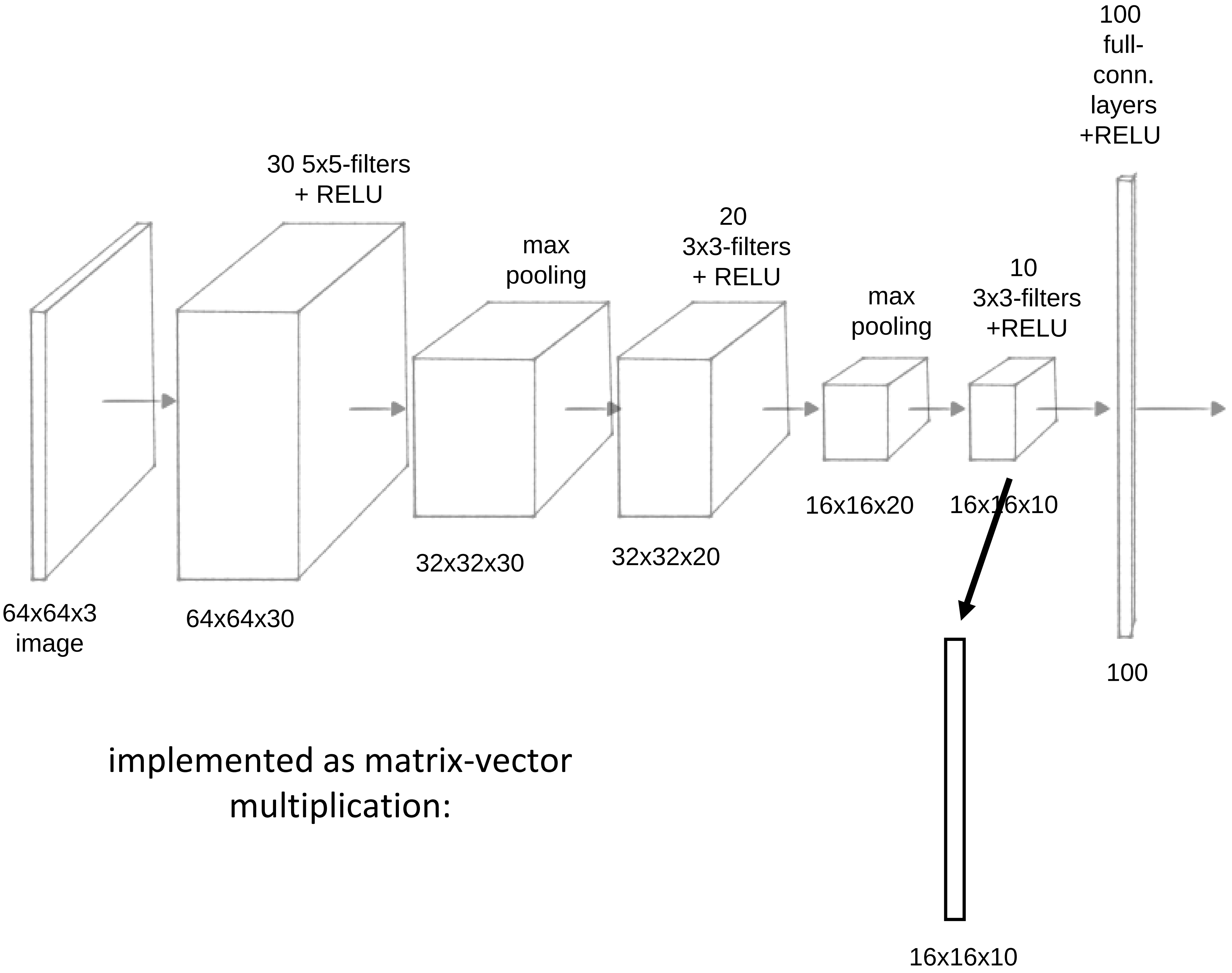
$$\sum_{u=1}^{16}\sum_{v=1}^{16}\sum_{i=1}^{10} w_{uvi}x_{uvi}$$

Network Structure



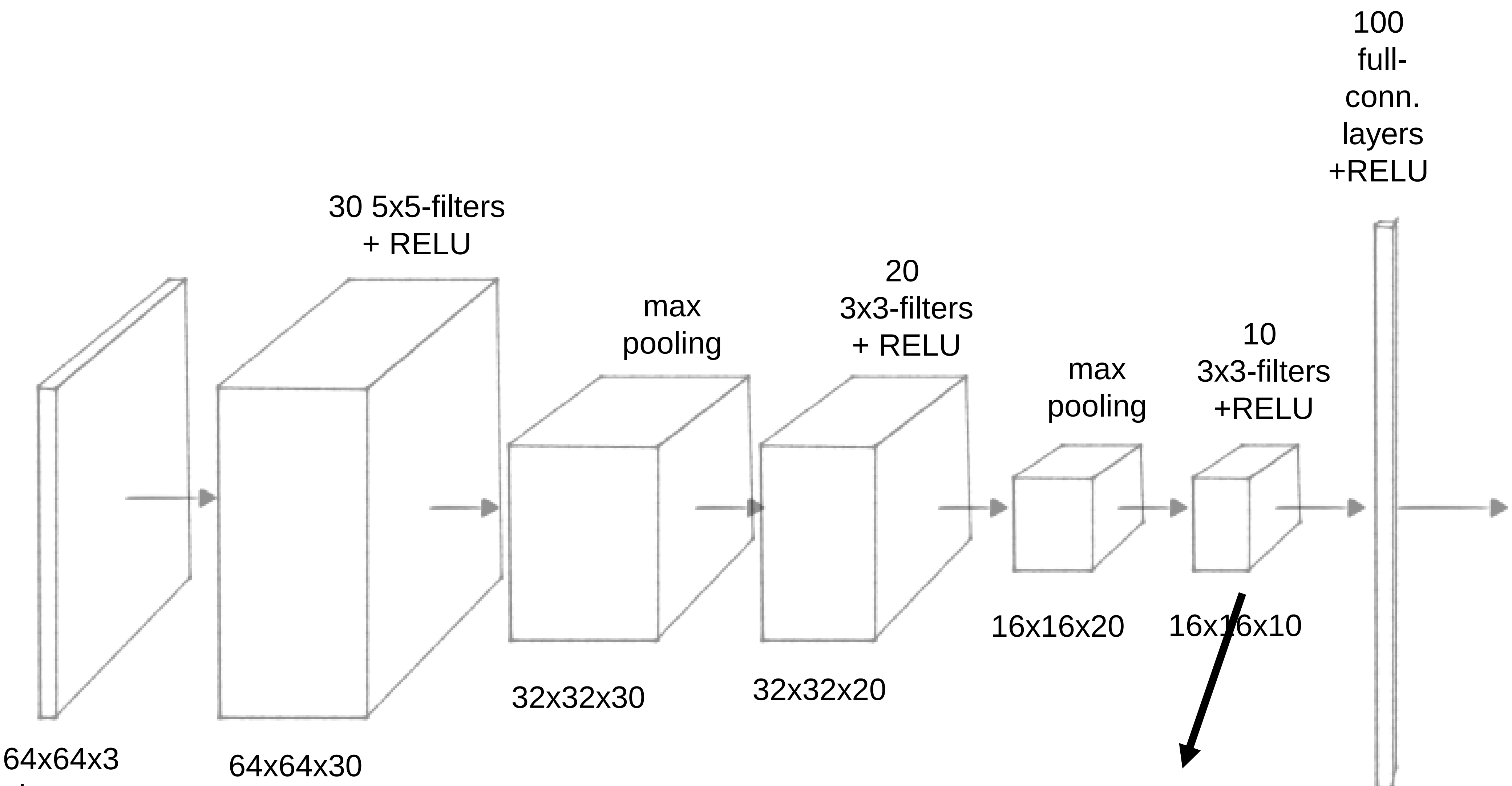
implemented as matrix-vector multiplication:

Network Structure



implemented as matrix-vector multiplication:

Network Structure



64x64x3
image

30 5x5-filters
+ RELU

64x64x30

max
pooling

32x32x30

20
3x3-filters
+ RELU

32x32x20

max
pooling

16x16x20

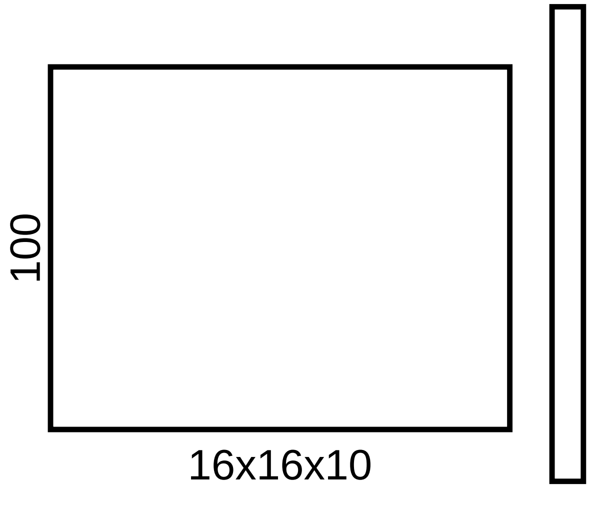
10
3x3-filters
+RELU

16x16x10

100
full-
conn.
layers
+RELU

100

implemented as matrix-vector
multiplication:

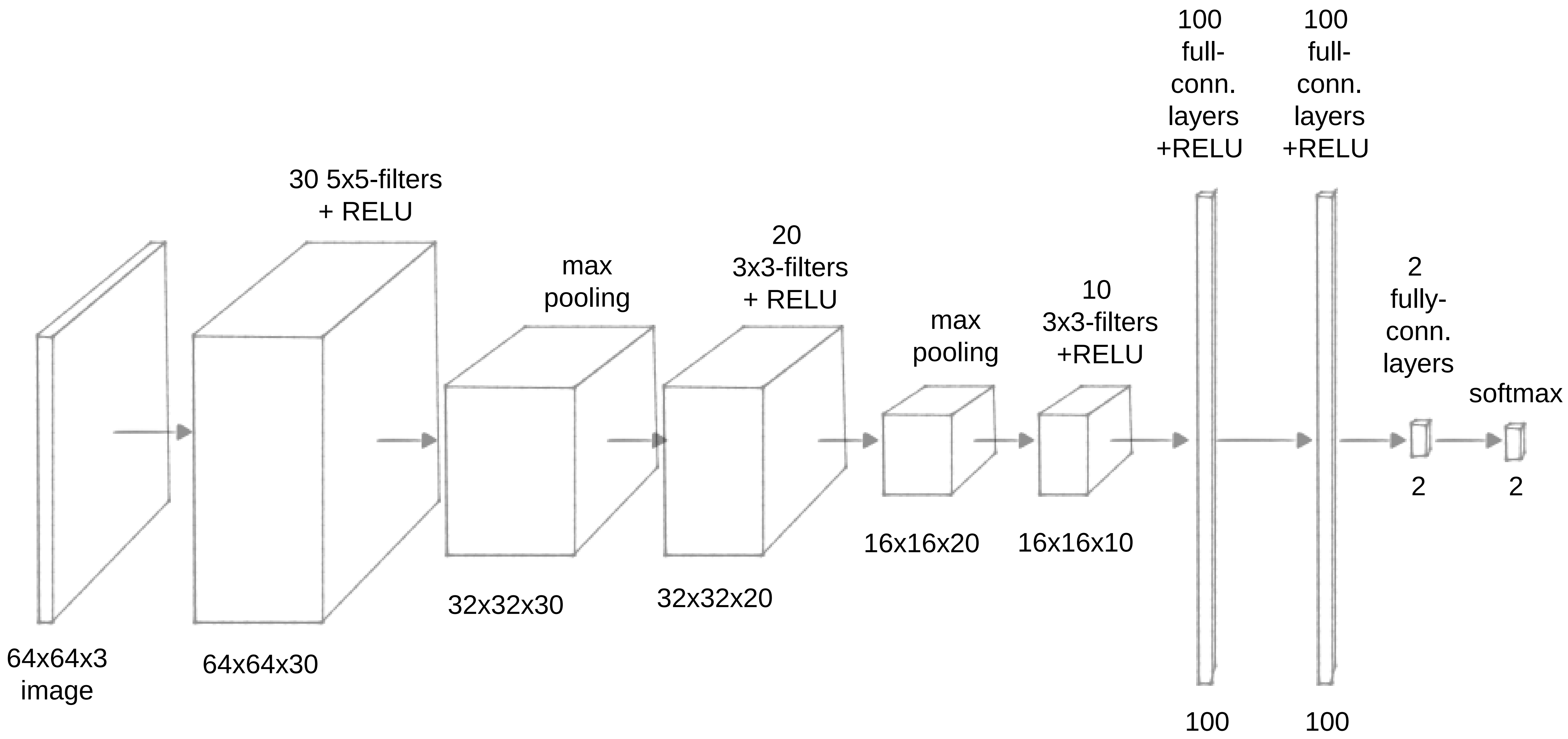


100

16x16x10

16x16x10

Network Structure

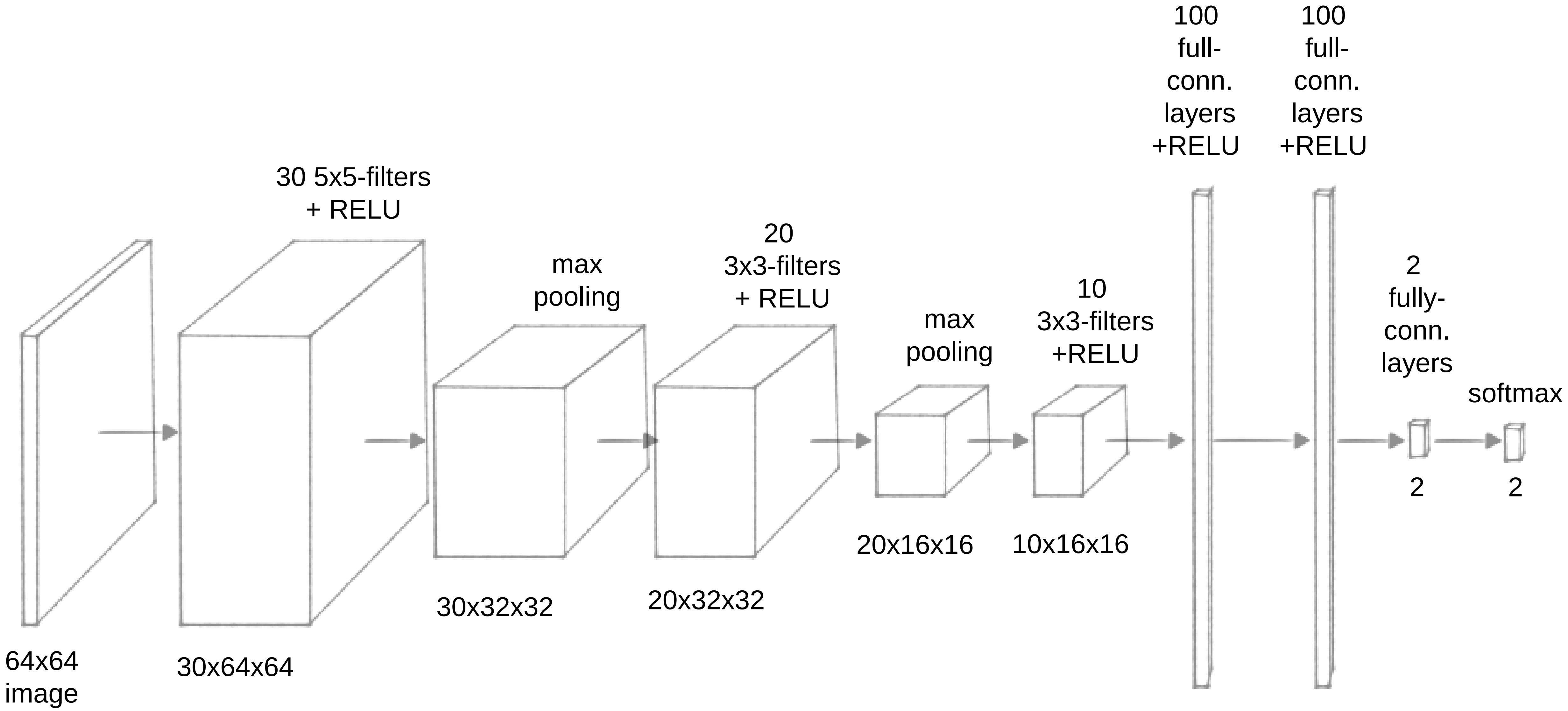


Fully Convolutional Neural Networks

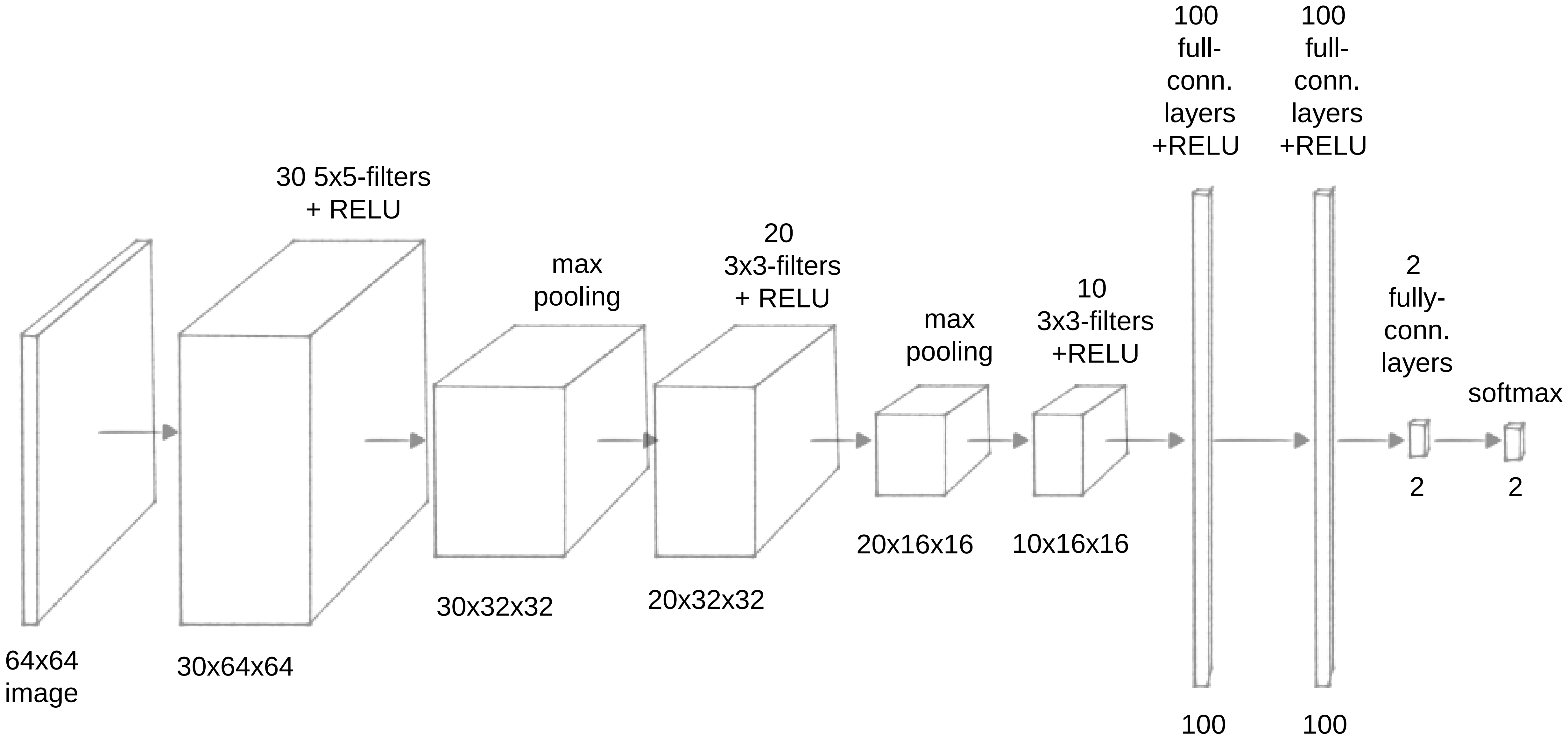
Recognition / Classification

Is there a robin in this image?

Convolutional Neural Network for Classification



Convolutional Neural Network for Classification



64x64 image

30 5x5-filters + RELU

30x64x64

max pooling

30x32x32

20 3x3-filters + RELU

20x32x32

max pooling

20x16x16

10 3x3-filters +RELU

10x16x16

100 full-conn. layers +RELU

100

100 full-conn. layers +RELU

100

2 fully-conn. layers

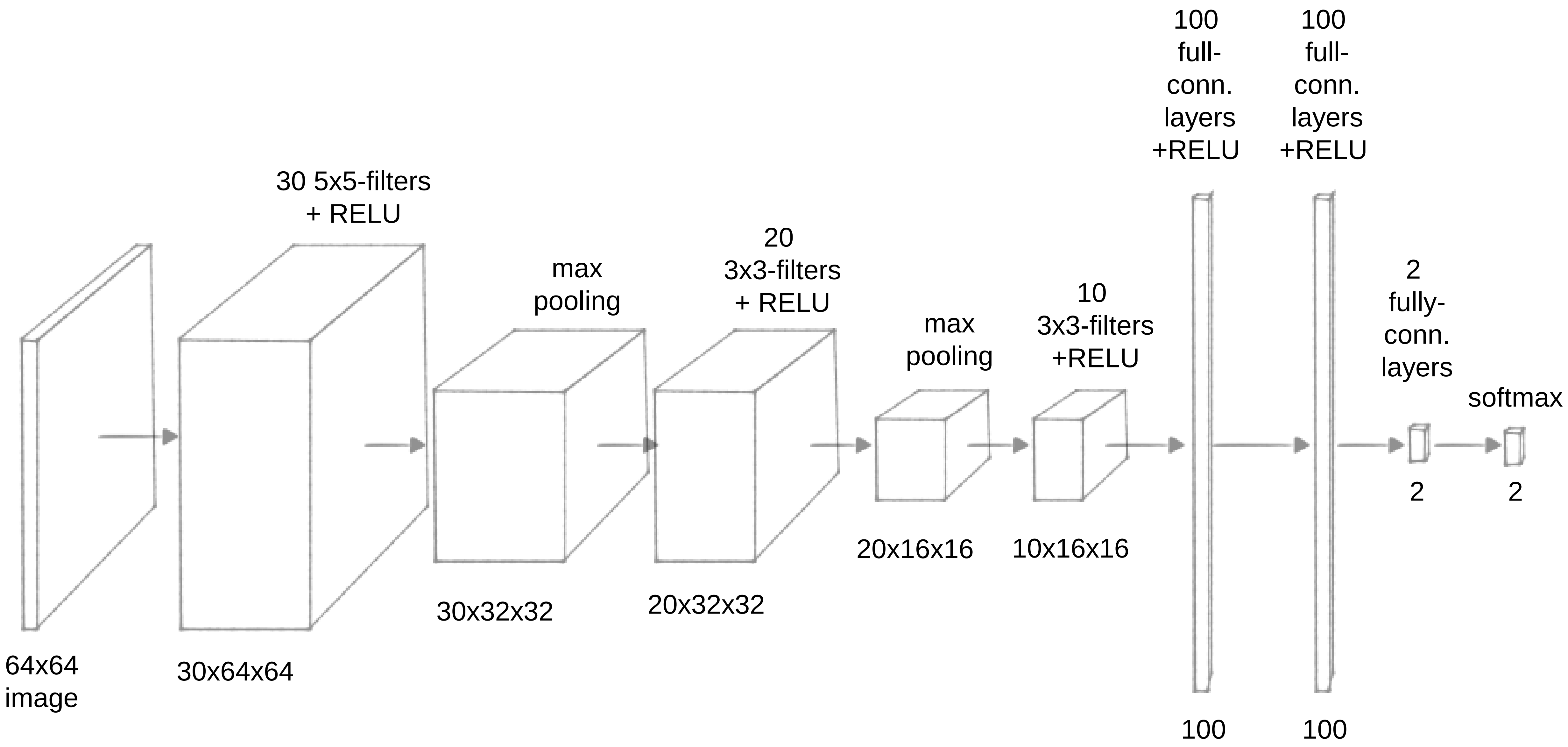
2

softmax

2

Input image size needs to be fixed during training!

Convolutional Neural Network for Classification



Input image size needs to be fixed during training!

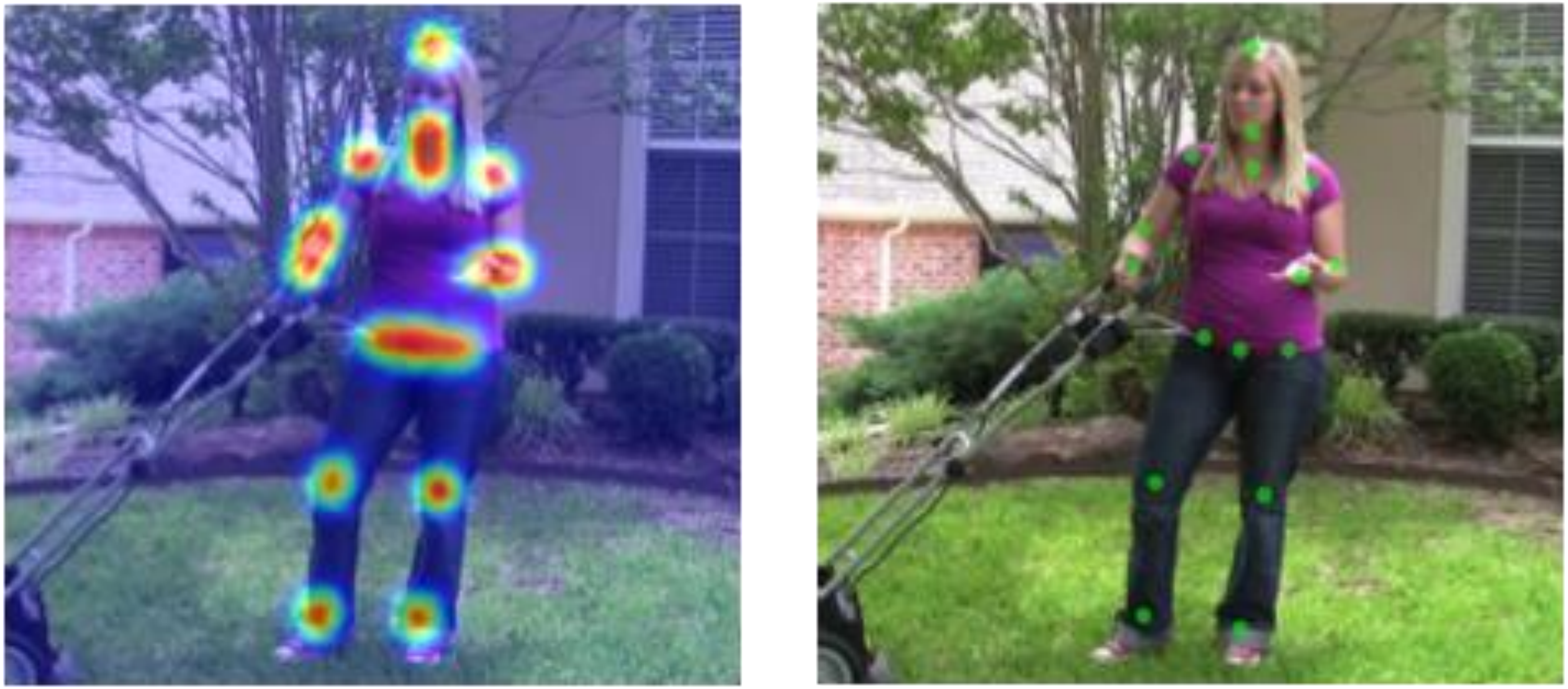
How to get a per-pixel decision?

Instance Segmentation

Semantic Segmentation

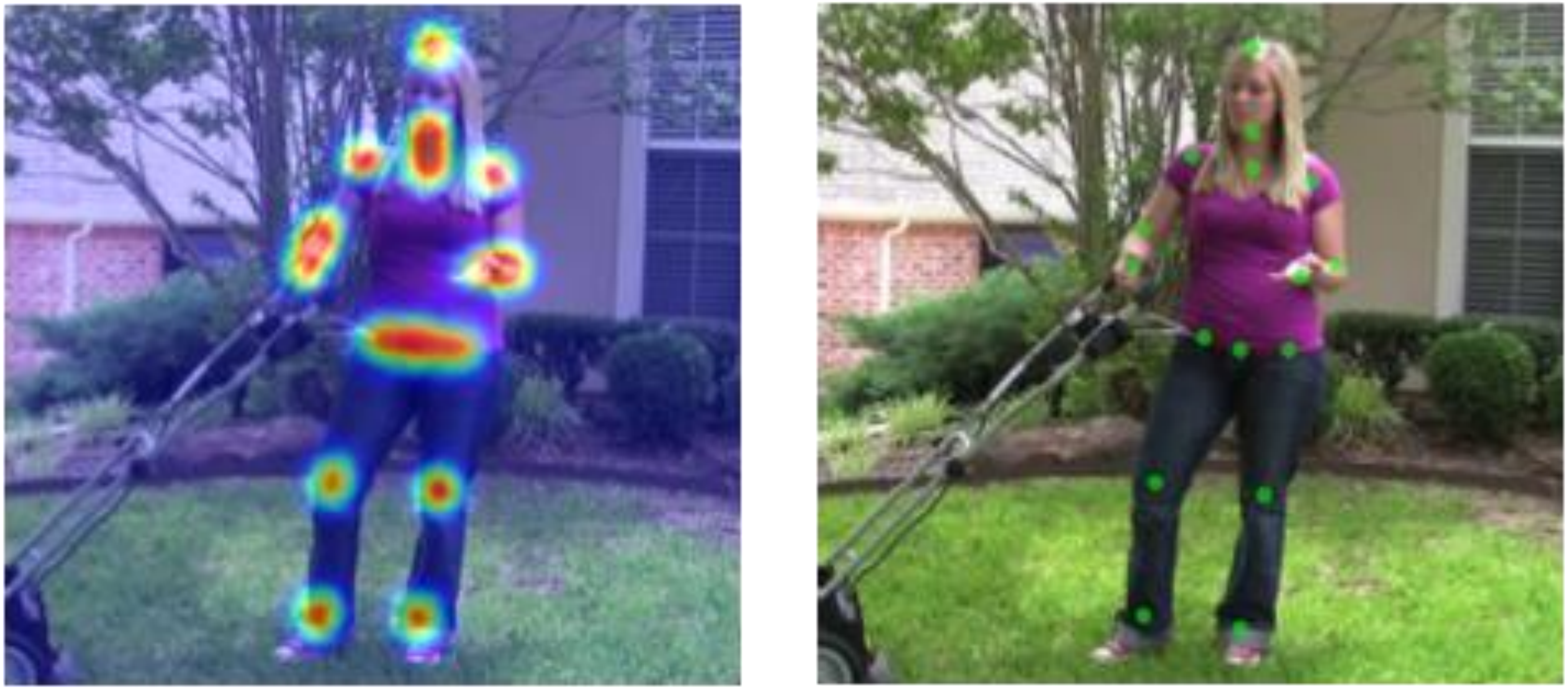
image source: Mapillary

Human Pose Estimation



Needs multiple decisions.

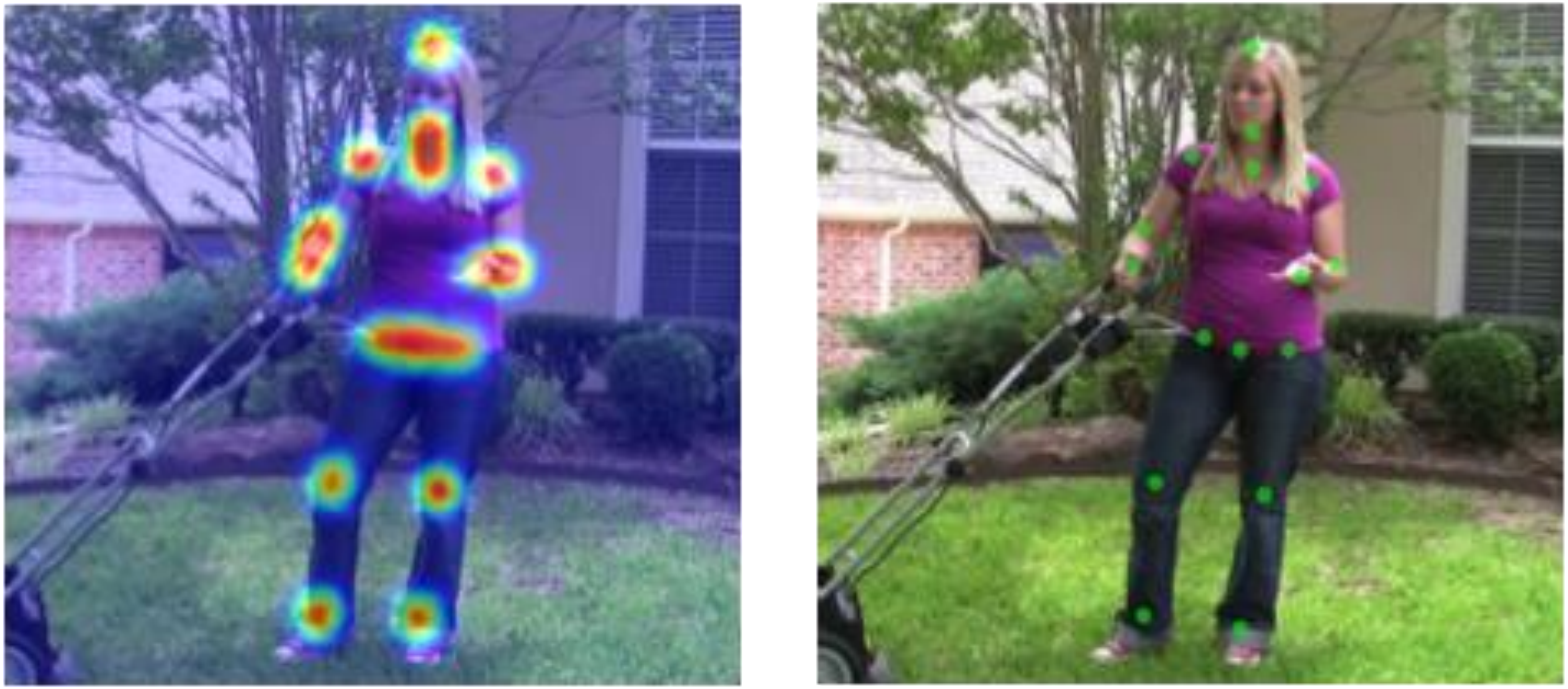
Human Pose Estimation



Needs multiple decisions.

Just apply network in sliding window fashion?

Human Pose Estimation

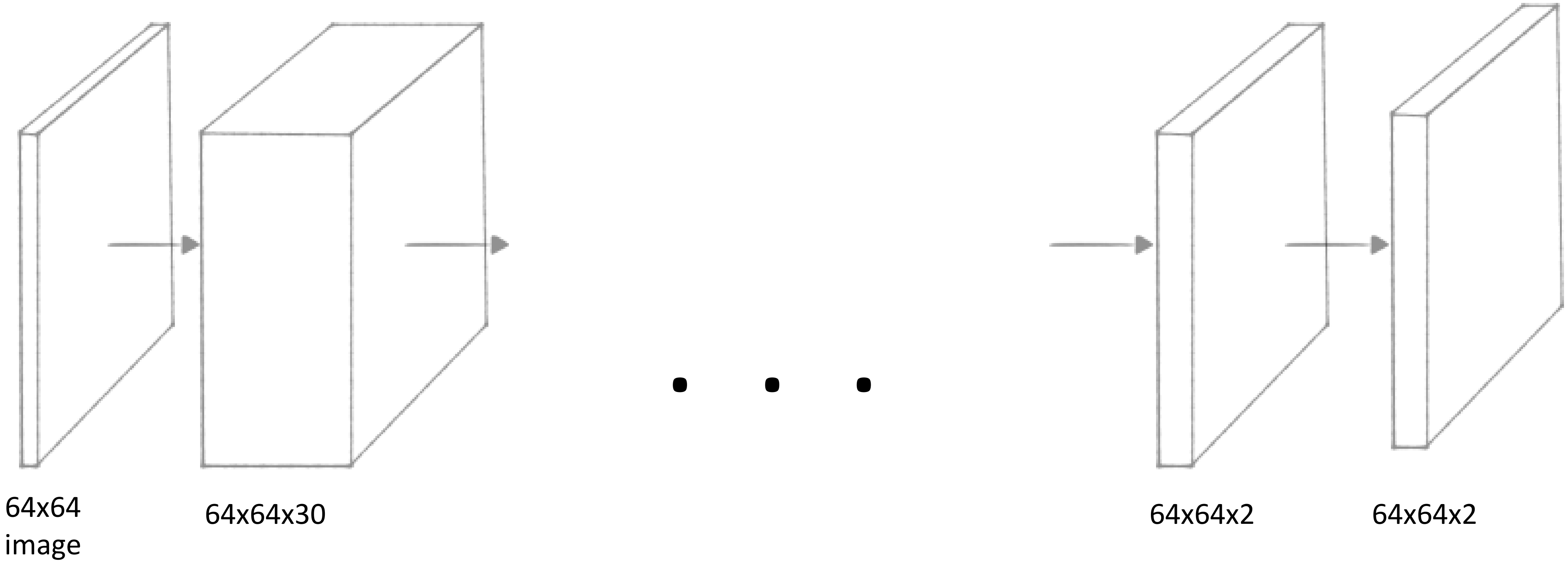


Needs multiple decisions.

Just apply network in sliding window fashion? Works, but slow!

Fully-Convolutional Networks

Faster approach:



From Fully-Connected to Fully-Convolutional

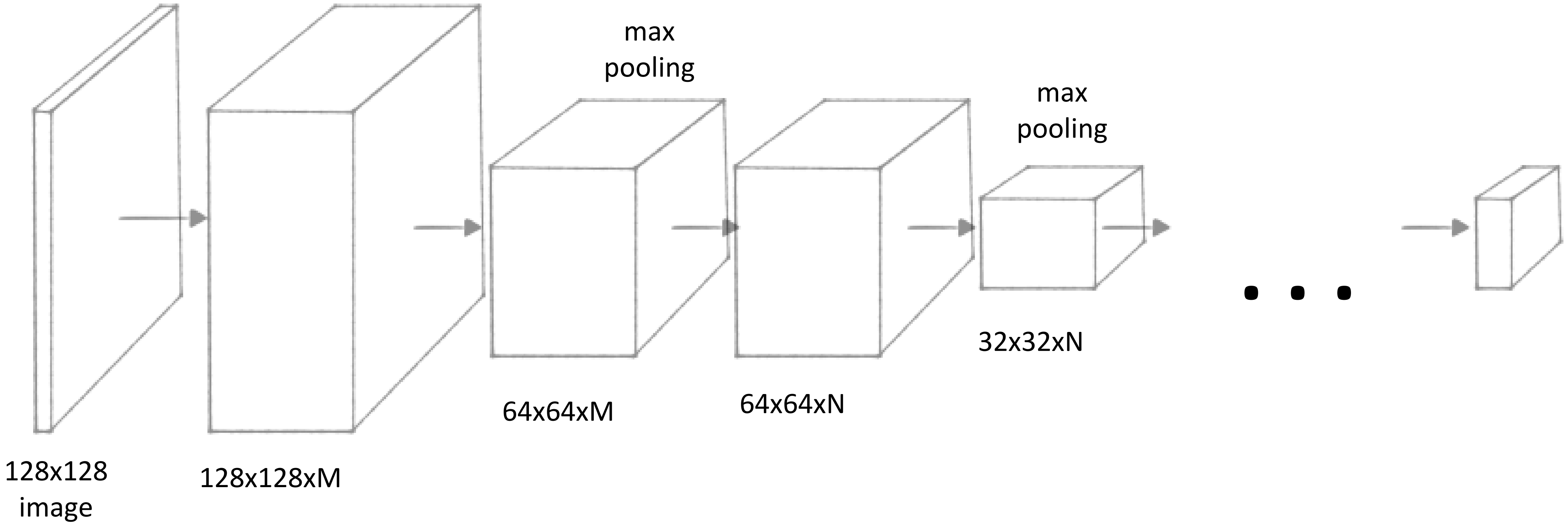
- Fully-connected layer is still a convolutional layer
- Can convert a network with fully-connected into fully-convolutional network:
  - Simply reshape matrix from fully-connected layer into a filter
  - Produces the same result on the input volume of the fully-connected layer
  - See example on black board

Fully-Convolutional Networks

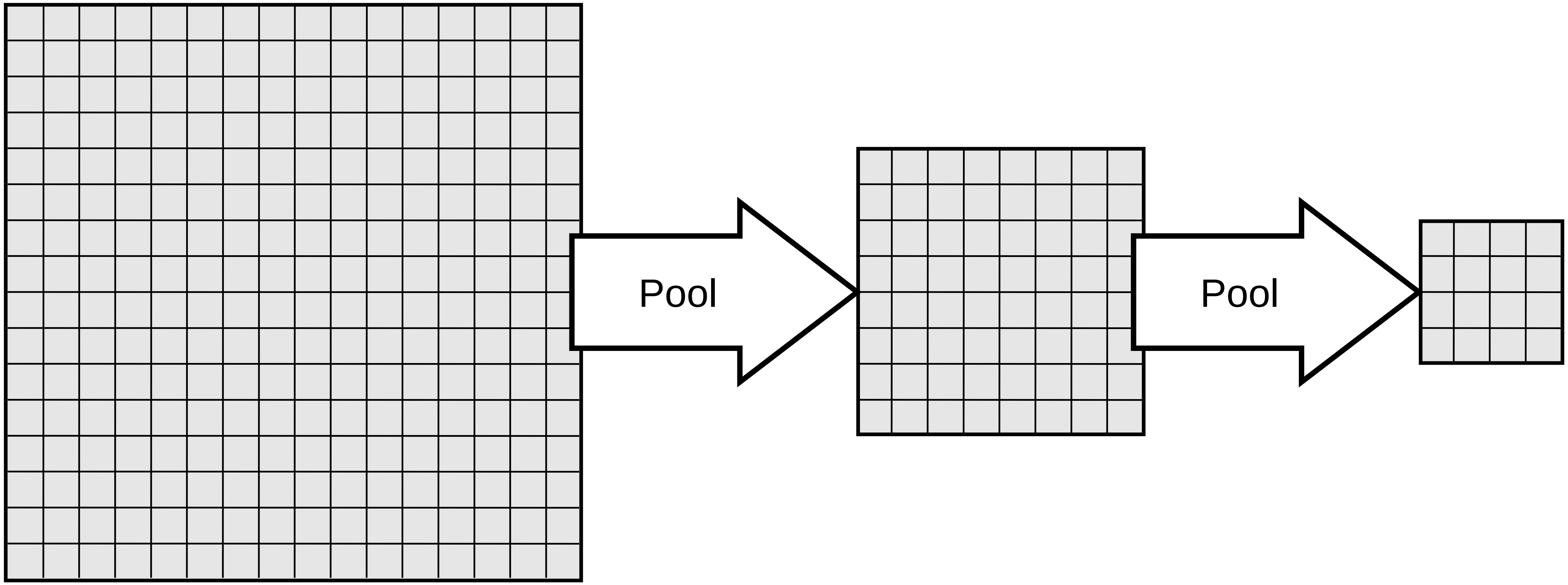
Faster approach:

But: Memory intensive at full resolution

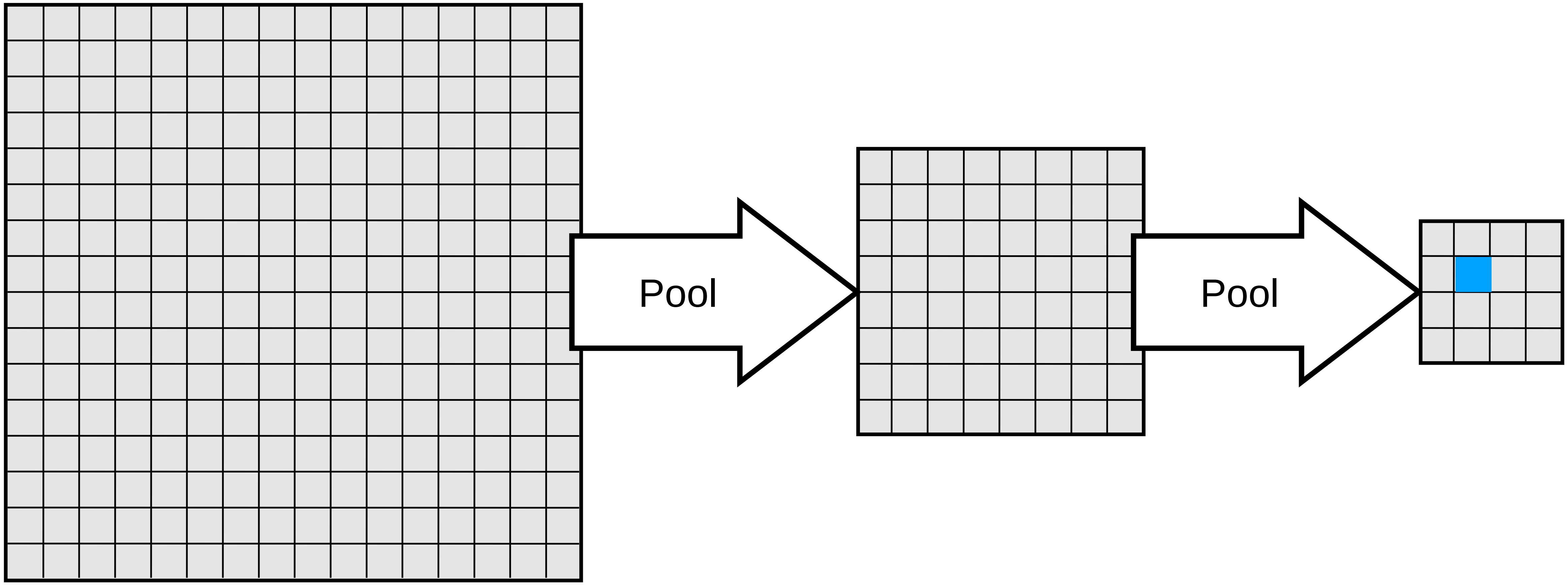
Fully-Convolutional Network for Detection



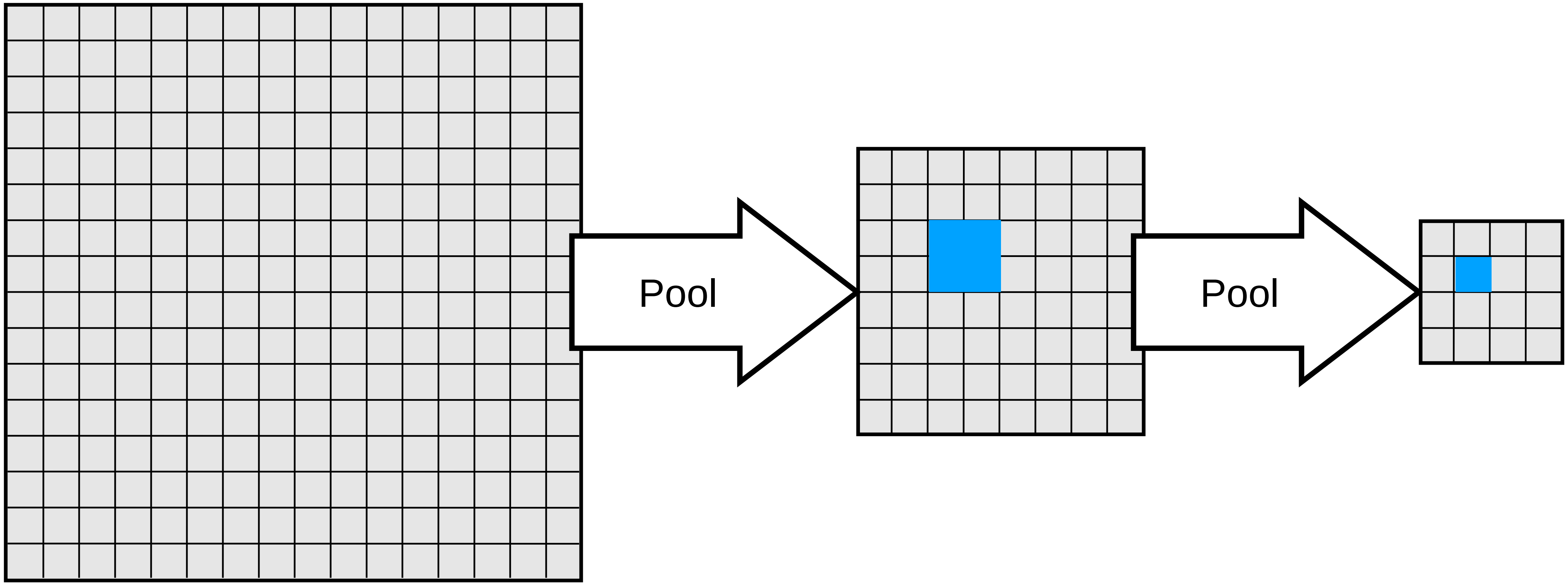
Further Benefit of (Max-)Pooling



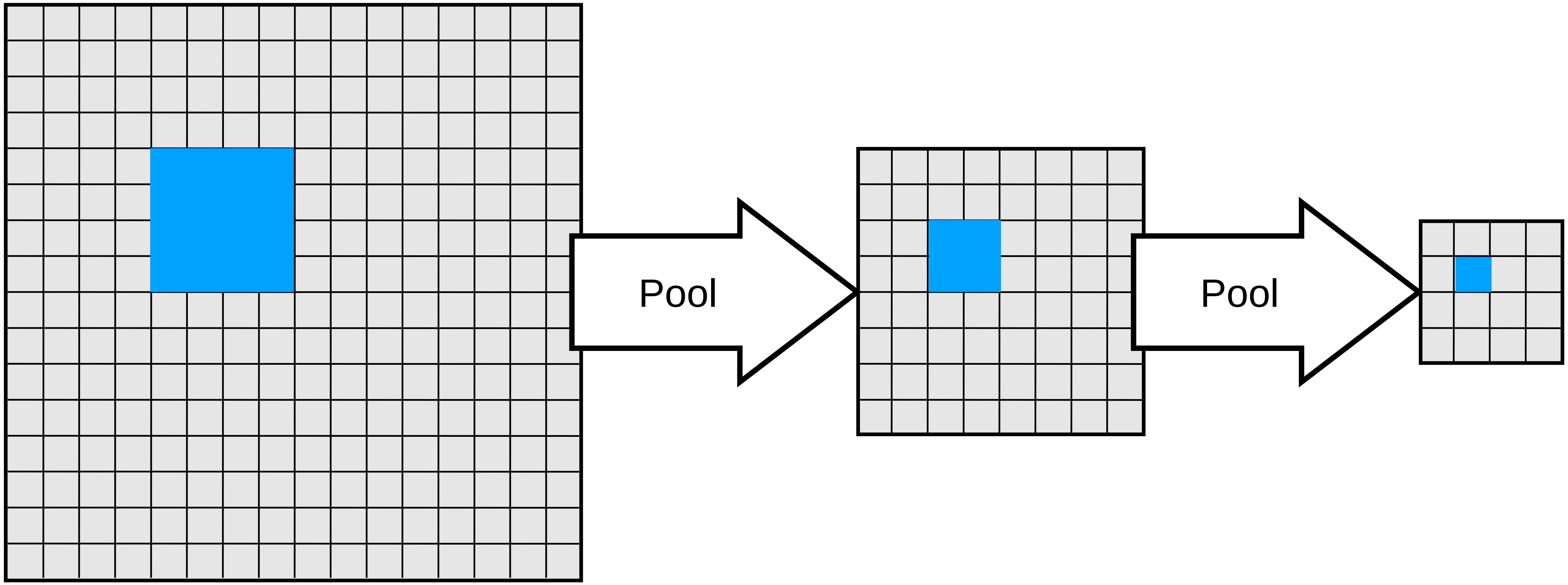
Further Benefit of (Max-)Pooling



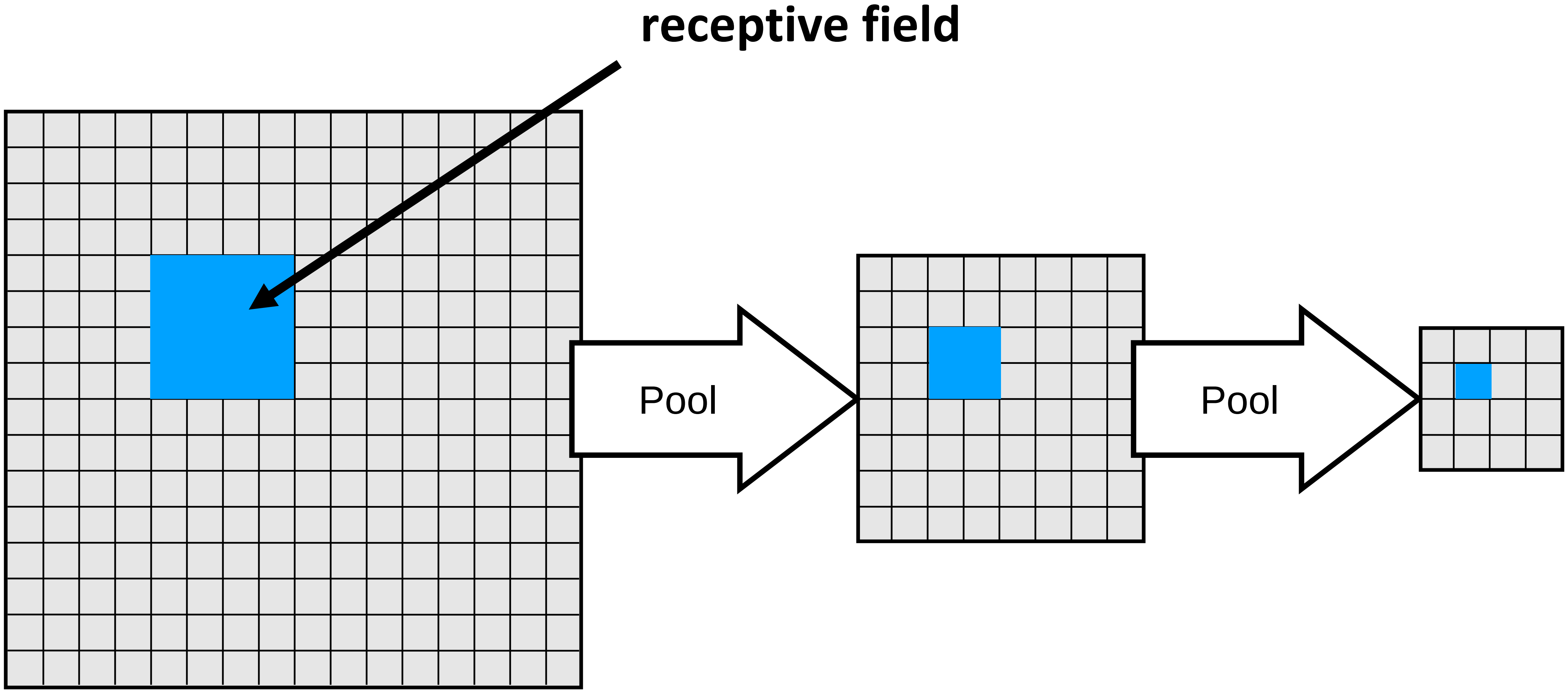
Further Benefit of (Max-)Pooling



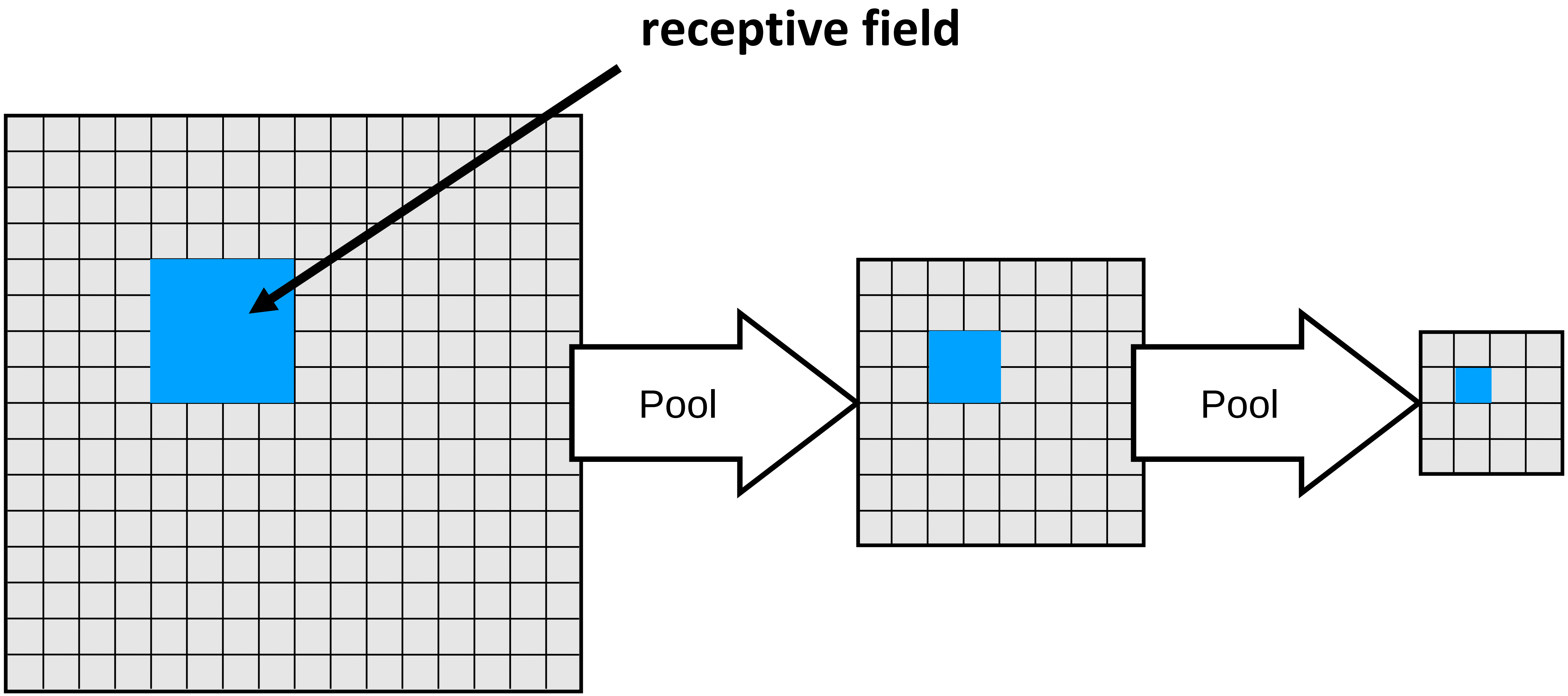
Further Benefit of (Max-)Pooling



Further Benefit of (Max-)Pooling

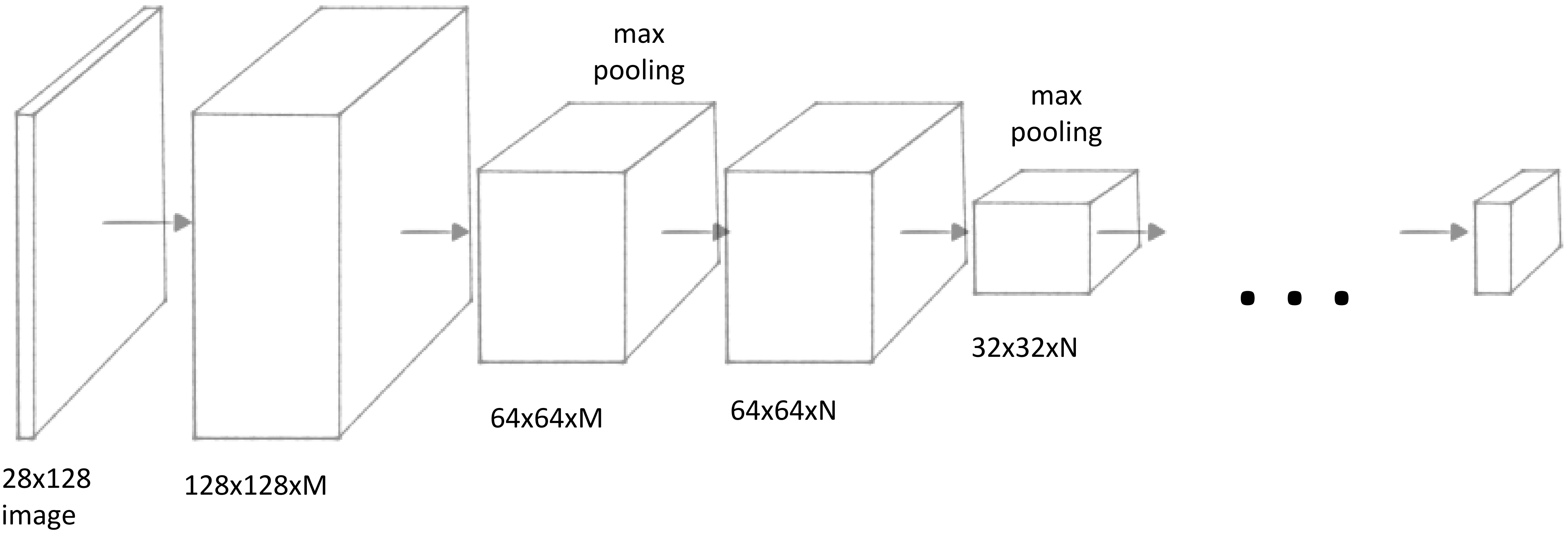


Further Benefit of (Max-)Pooling



Pooling increases receptive field size!

Fully-Convolutional Network for Detection



filters 32 64 128 256

max pooling

max pooling

128x128 image

128x128xM

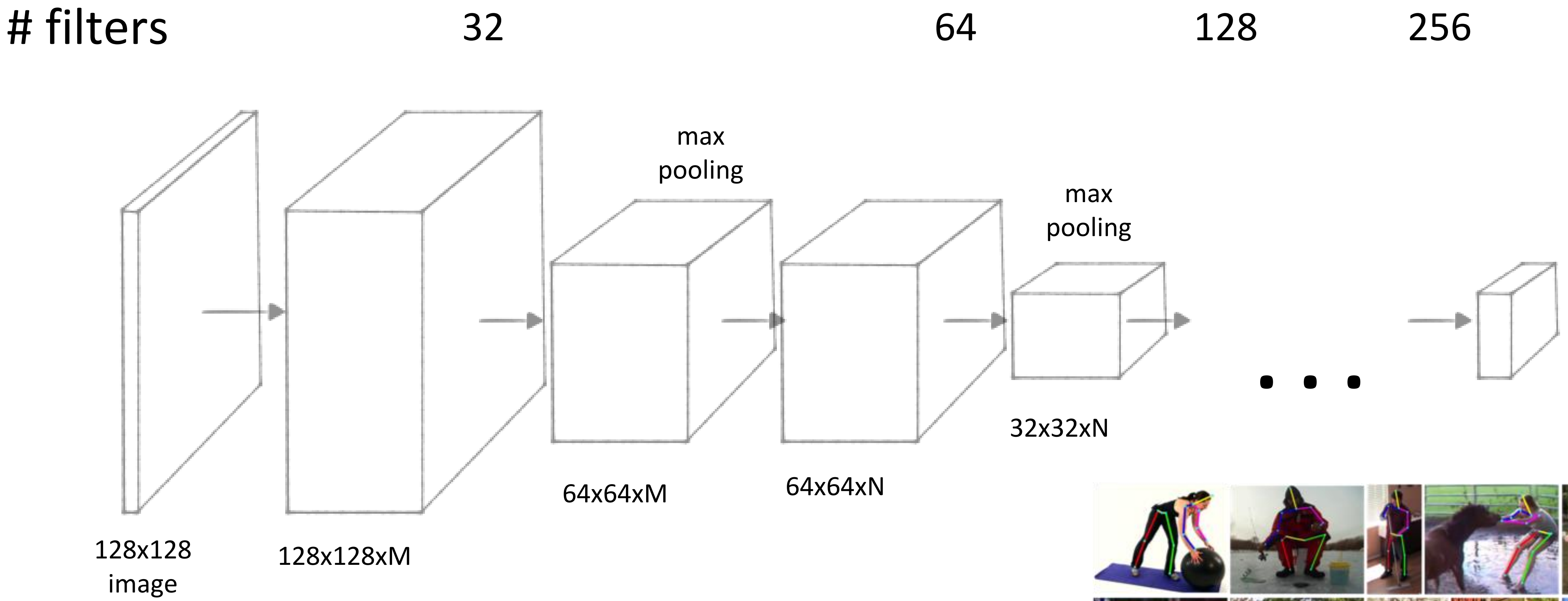
64x64xM

64x64xN

32x32xN

. . .

Fully-Convolutional Network for Detection



filters 32 64 128 256

max pooling

max pooling

128x128 image

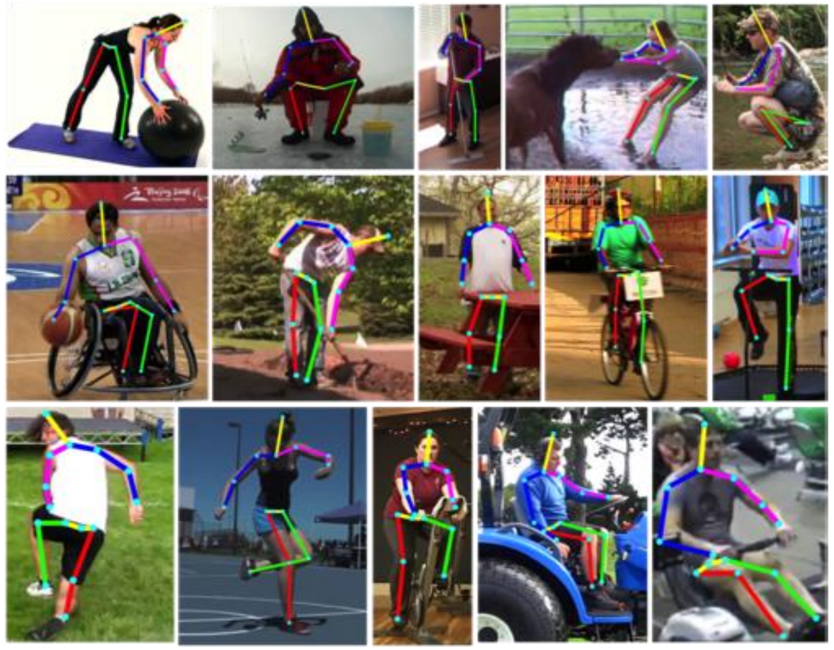
128x128xM

64x64xM

64x64xN

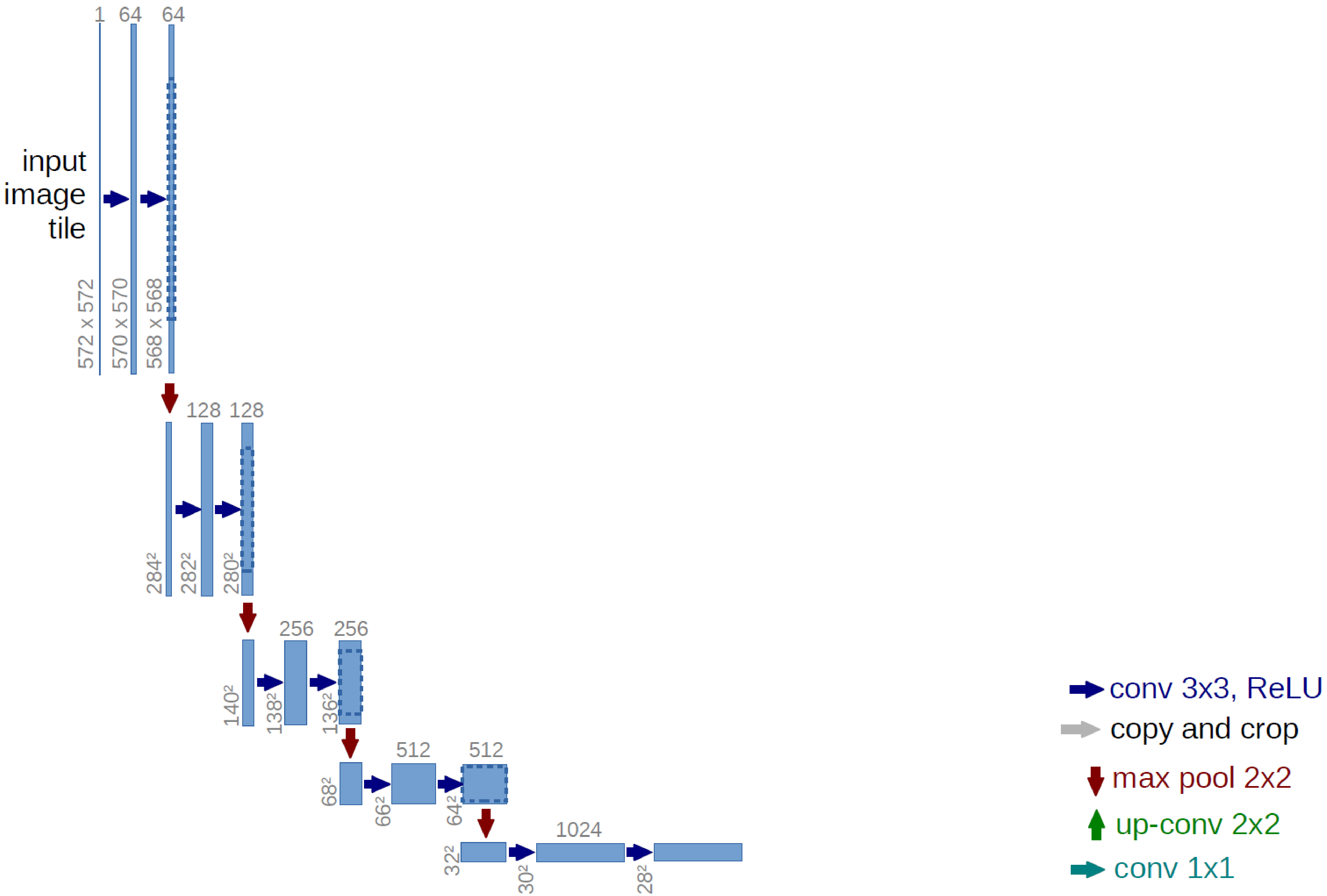
32x32xN

. . .



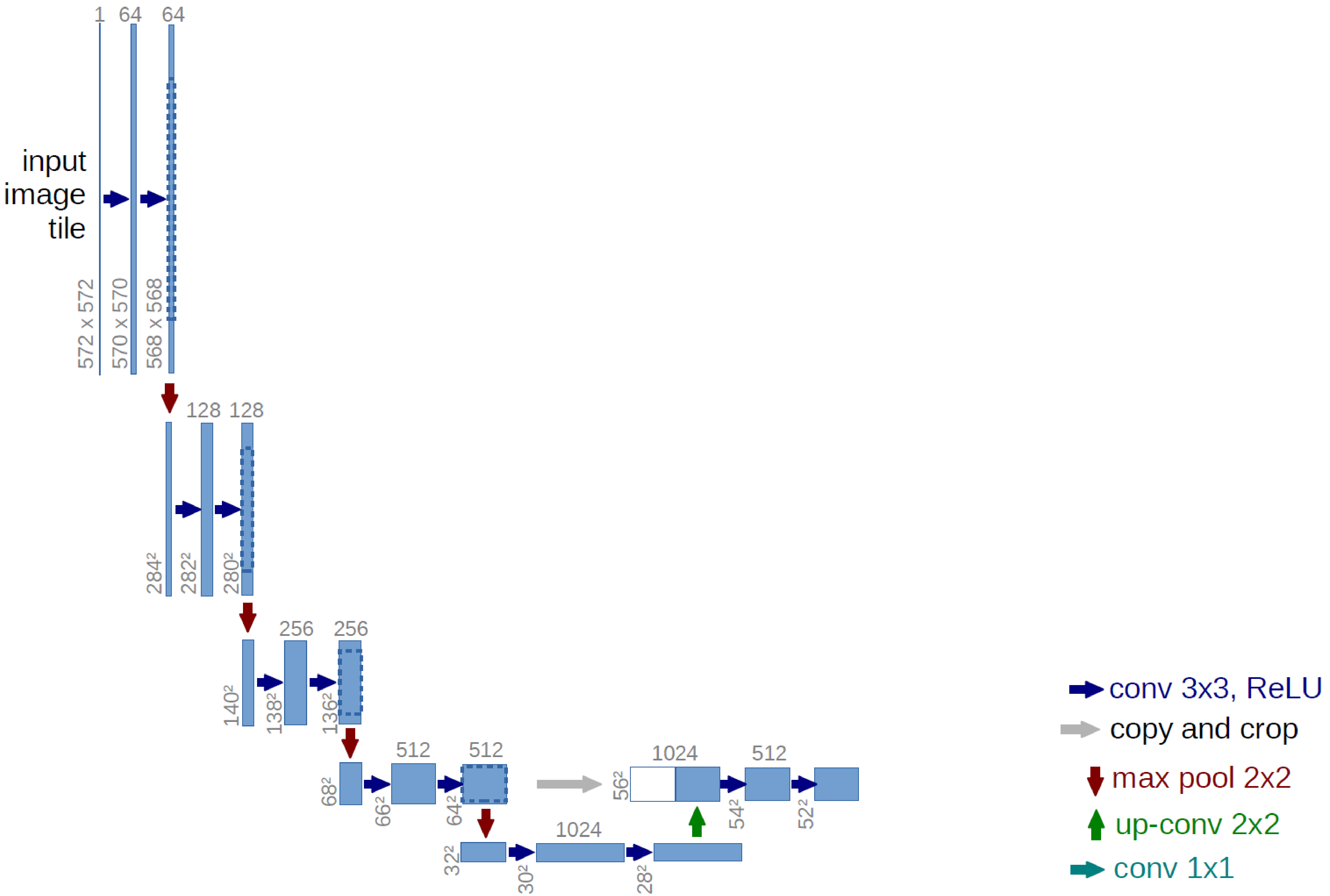
Disadvantage: Loss of resolution

U-Net: Decrease and Increase Resolution



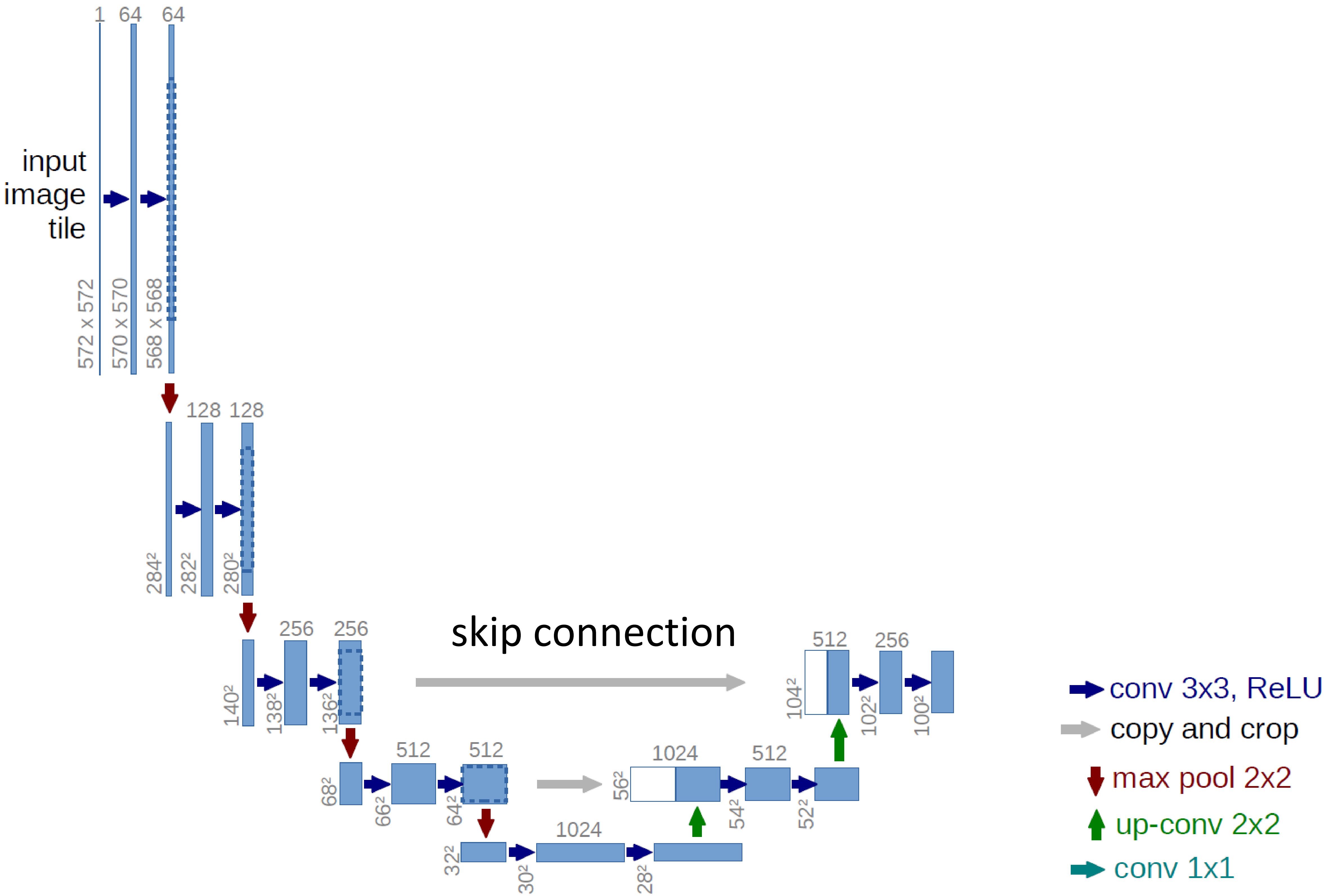
[Ronneberger et al., U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI 2015]

U-Net: Decrease and Increase Resolution



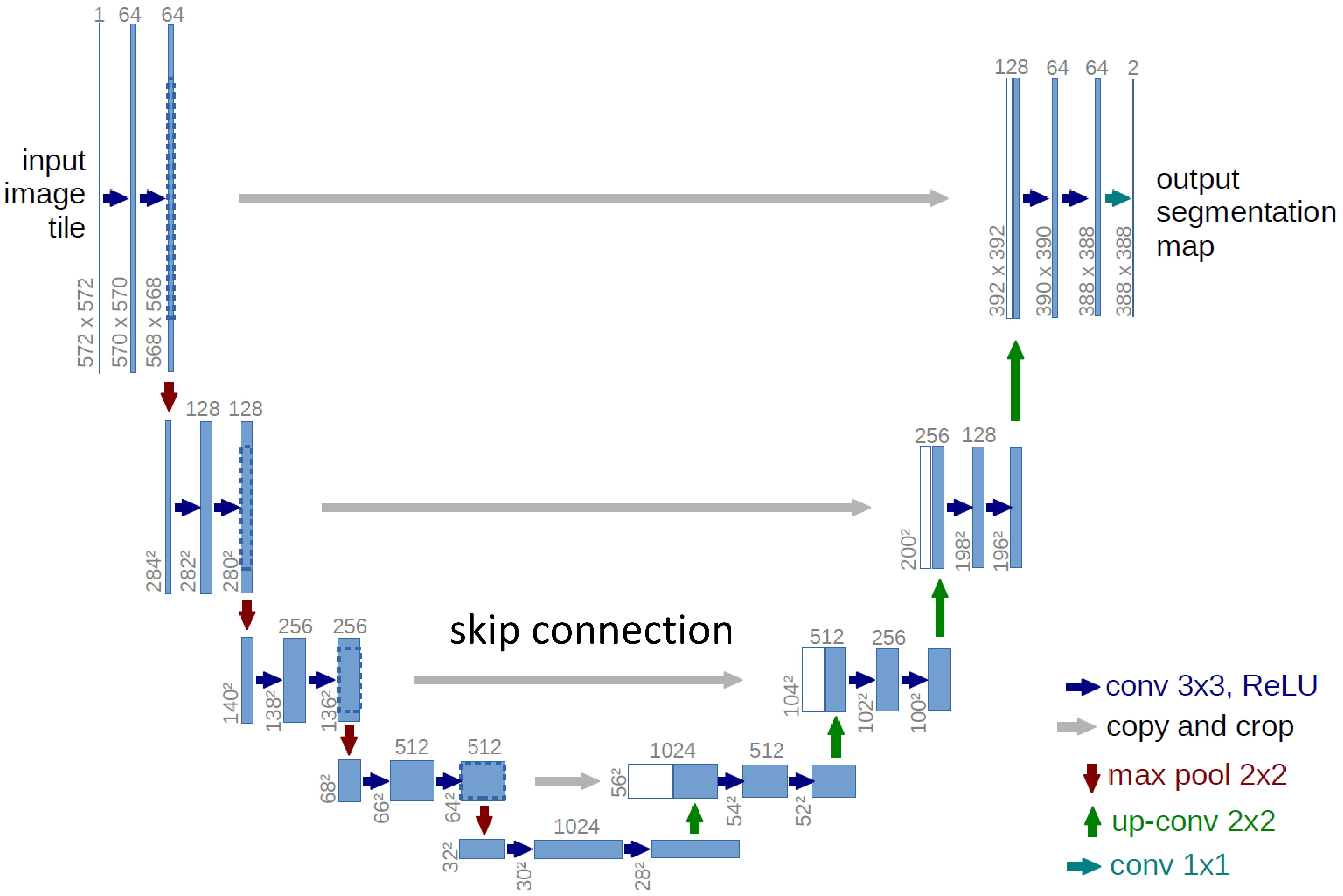
[Ronneberger et al., U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI 2015]

U-Net: Decrease and Increase Resolution



[Ronneberger et al., U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI 2015]

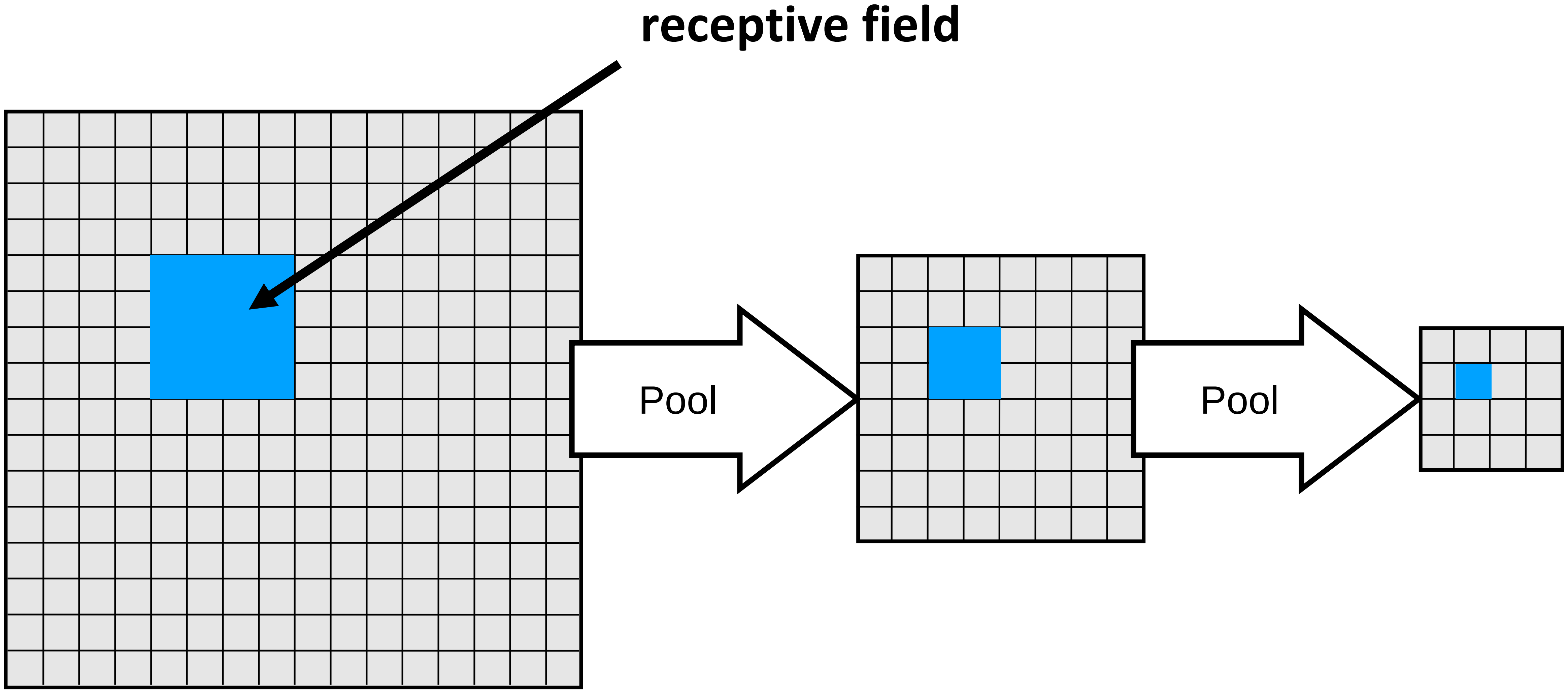
U-Net: Decrease and Increase Resolution



[Ronneberger et al., U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI 2015]

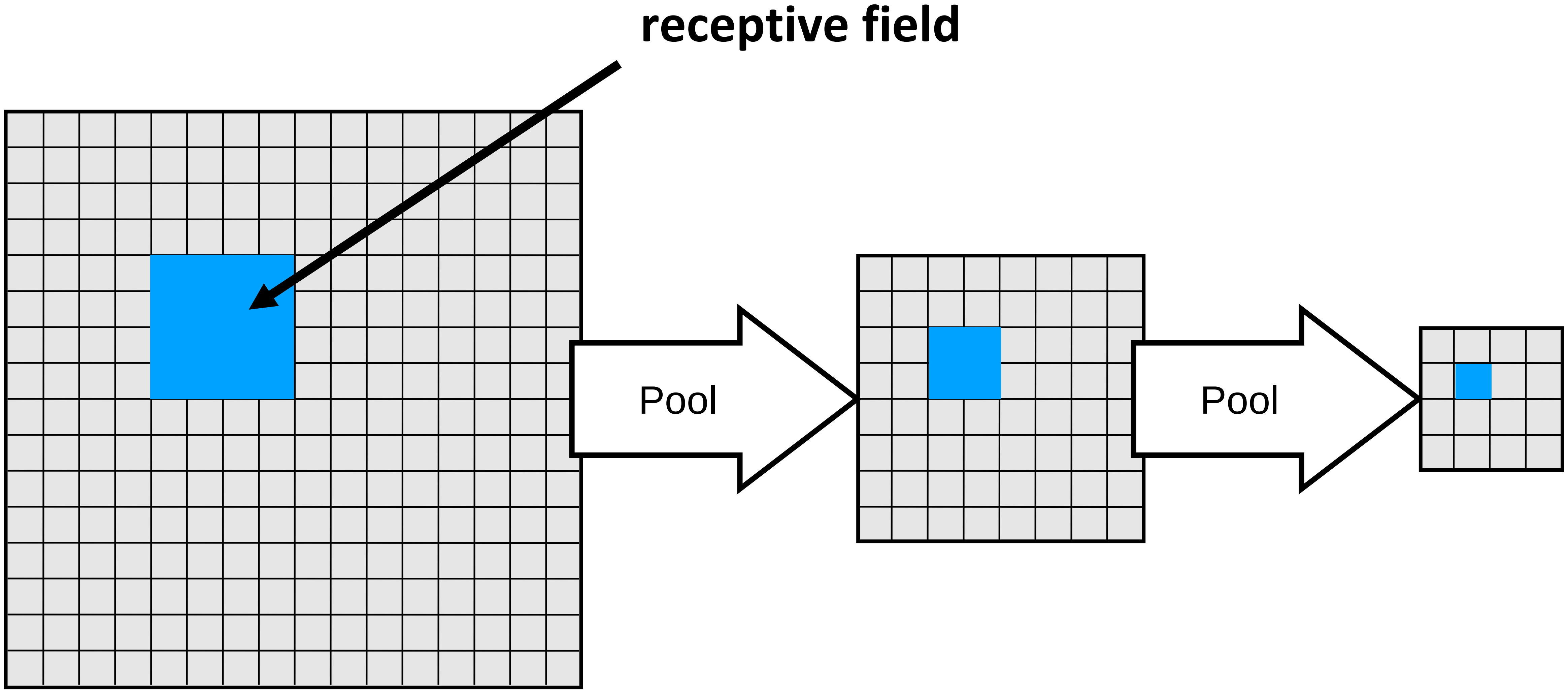
Convolutional Layers

Increasing the Receptive Field by (Max-)Pooling



Advantage: Introduces robustness to small shifts

Increasing the Receptive Field by (Max-)Pooling

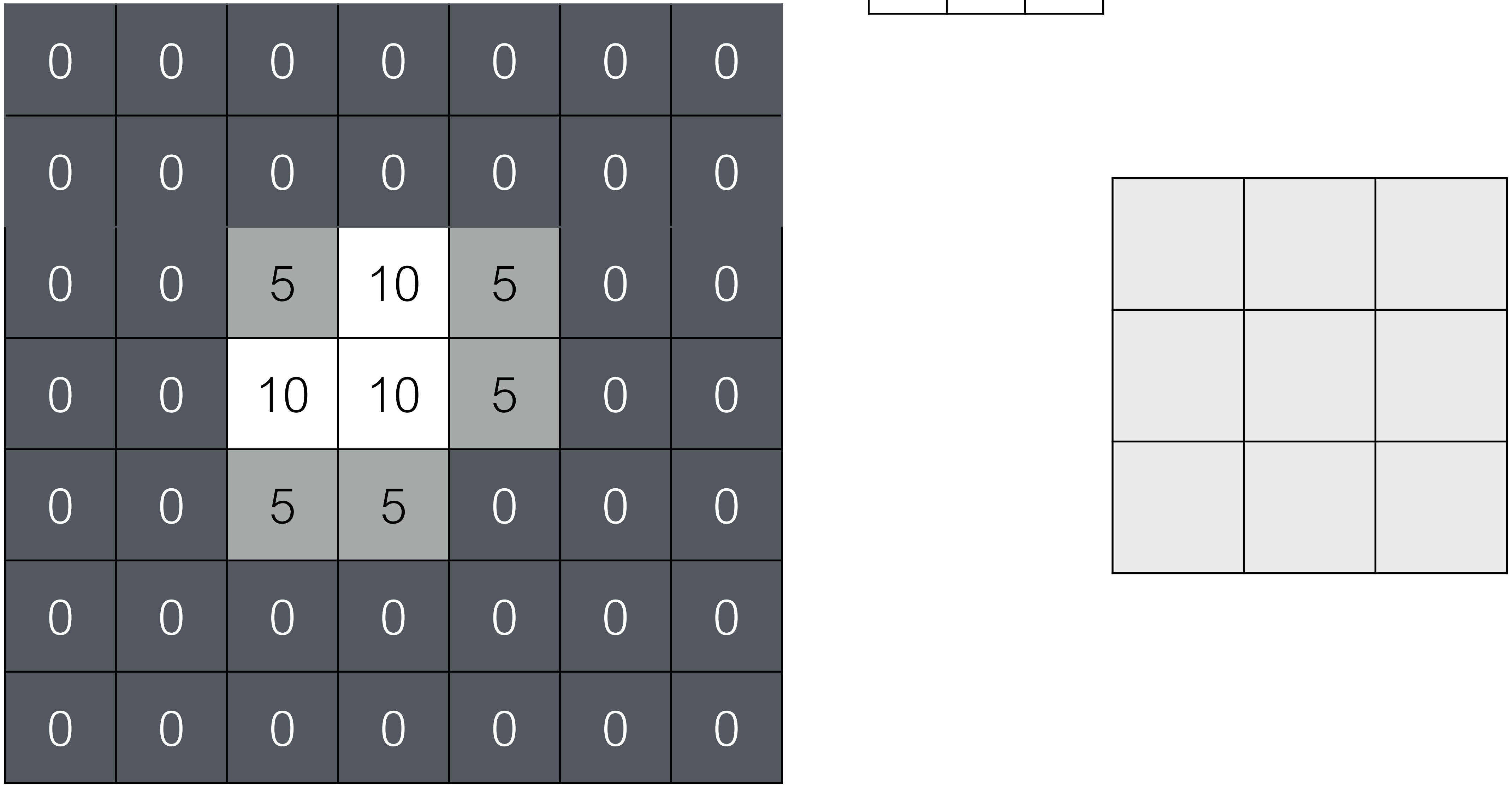


Advantage: Introduces robustness to small shifts

… but we might not always want that

Learnable Alternative: Strided Filtering

$$w = \frac{1}{5}\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 1 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$



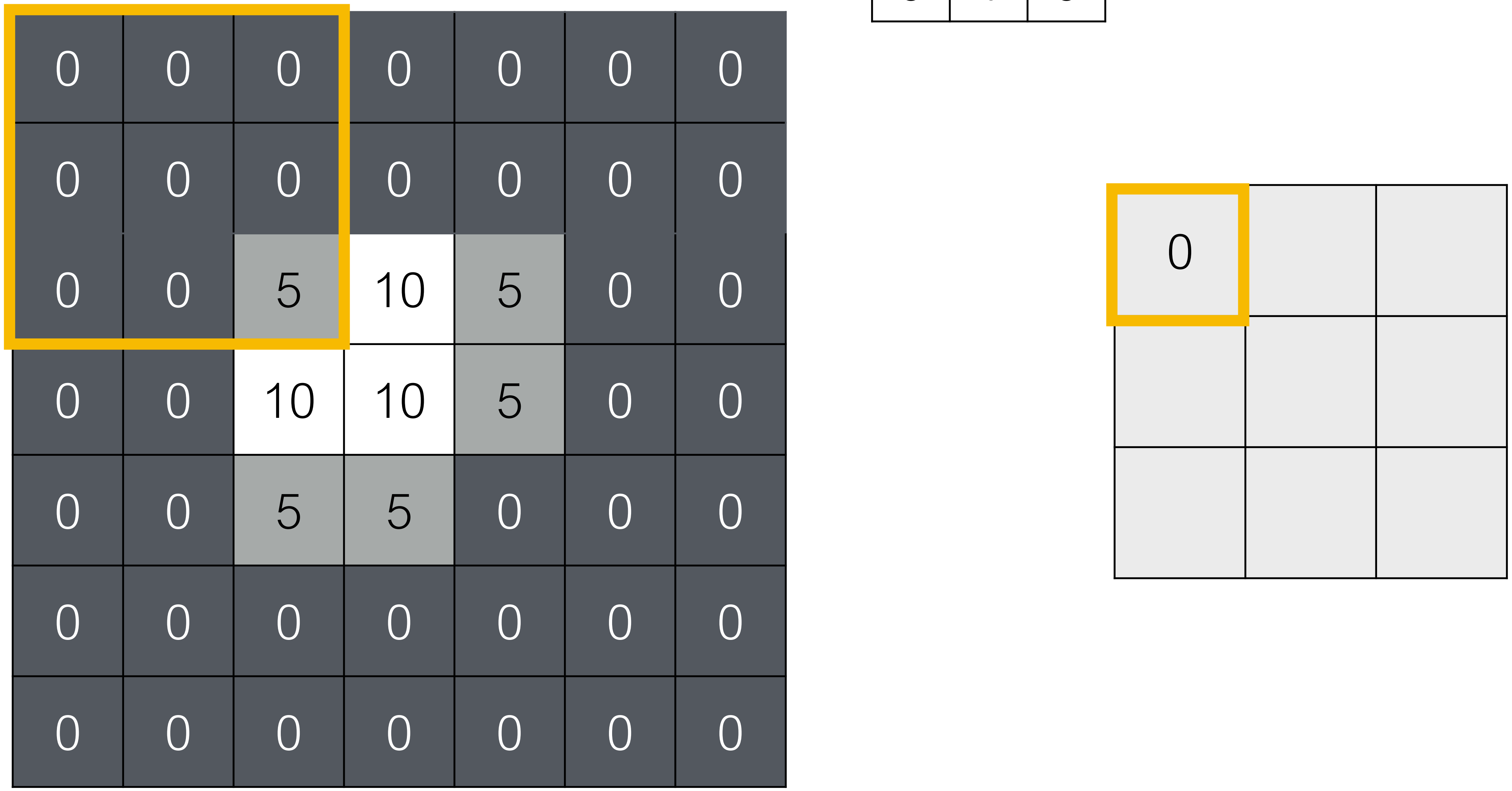
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

I

$I \star w$

Learnable Alternative: Strided Filtering

$$w = \frac{1}{5}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



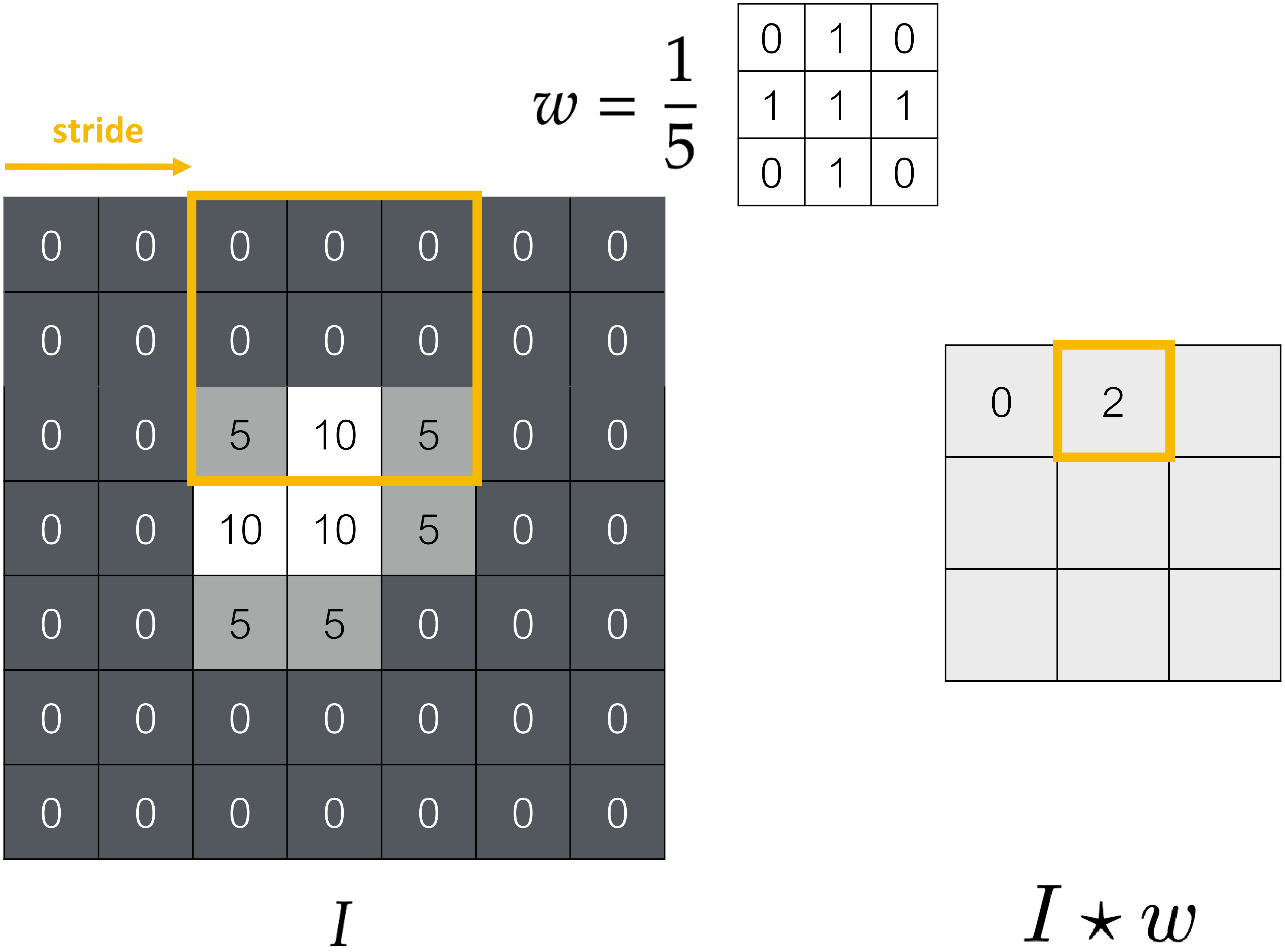
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

I

0		

$I \star w$

Learnable Alternative: Strided Filtering



$$w = \frac{1}{5}\begin{bmatrix}0 & 1 & 0\\ 1 & 1 & 1\\ 0 & 1 & 0\end{bmatrix}$$

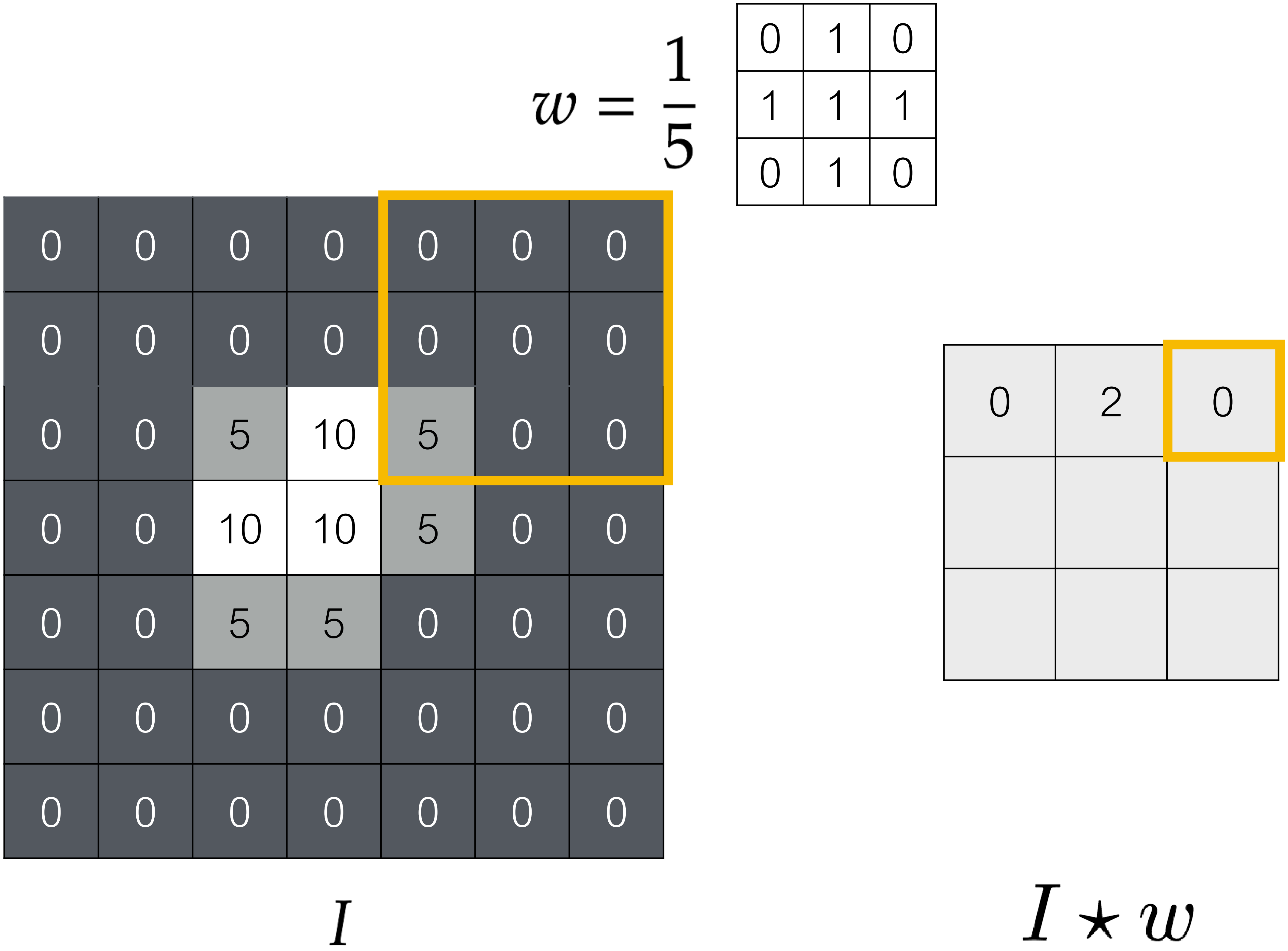
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

I

0	2	

$I \star w$

Learnable Alternative: Strided Filtering



$$w = \frac{1}{5}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

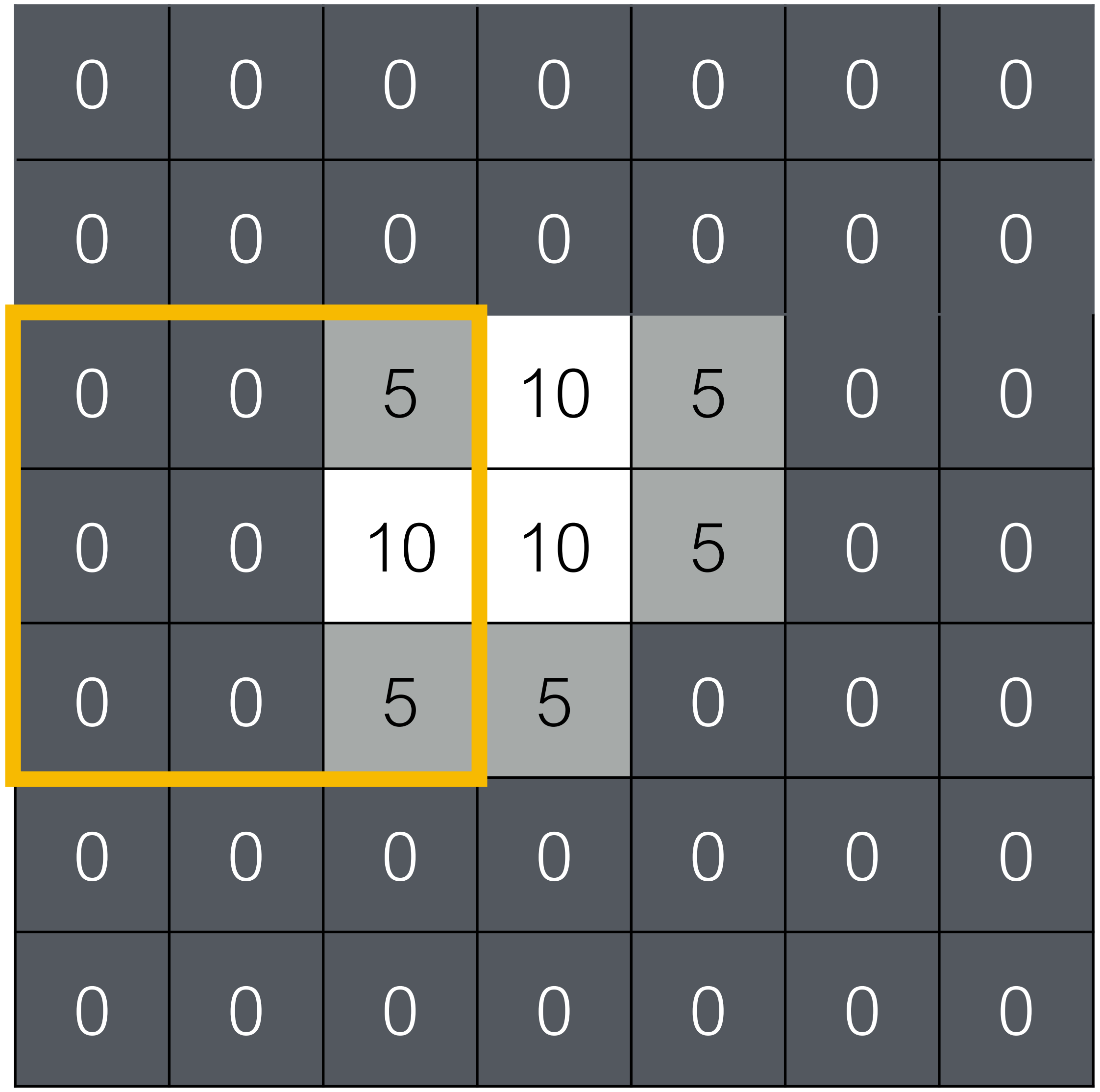
I

0	2	0

$I \star w$

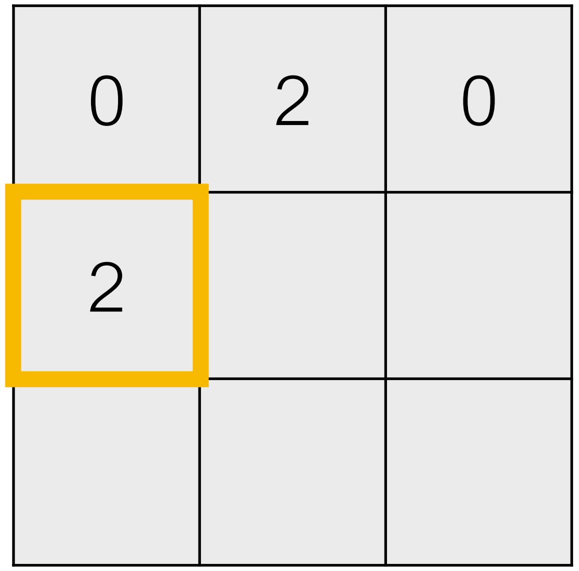
Learnable Alternative: Strided Filtering

$$w = \frac{1}{5}\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 1 & 1 \\ \hline 0 & 1 & 0 \\ \hline\end{array}$$



0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

I

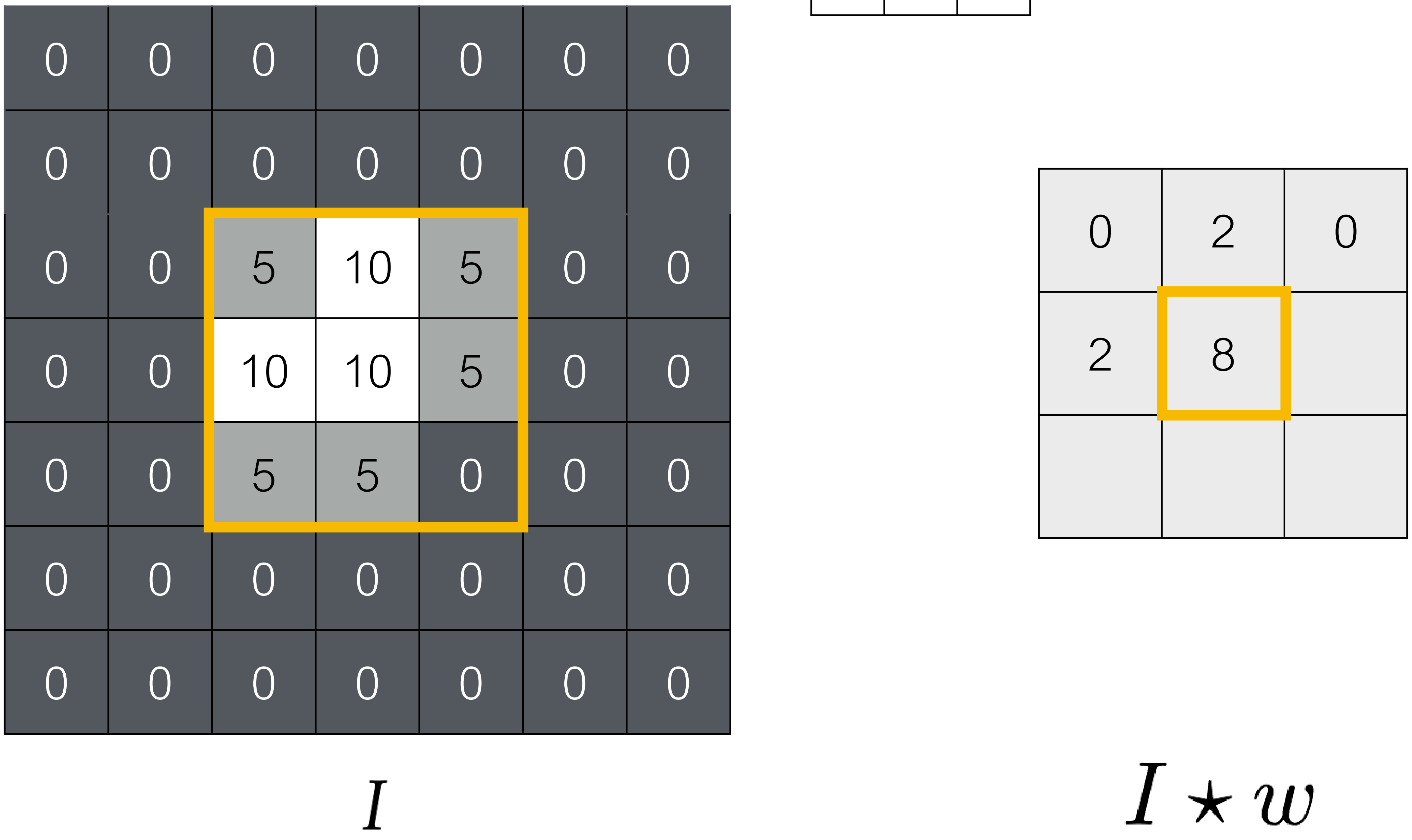


0	2	0
2		

$I \star w$

Learnable Alternative: Strided Filtering

$$w = \frac{1}{5}\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 1 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$



0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

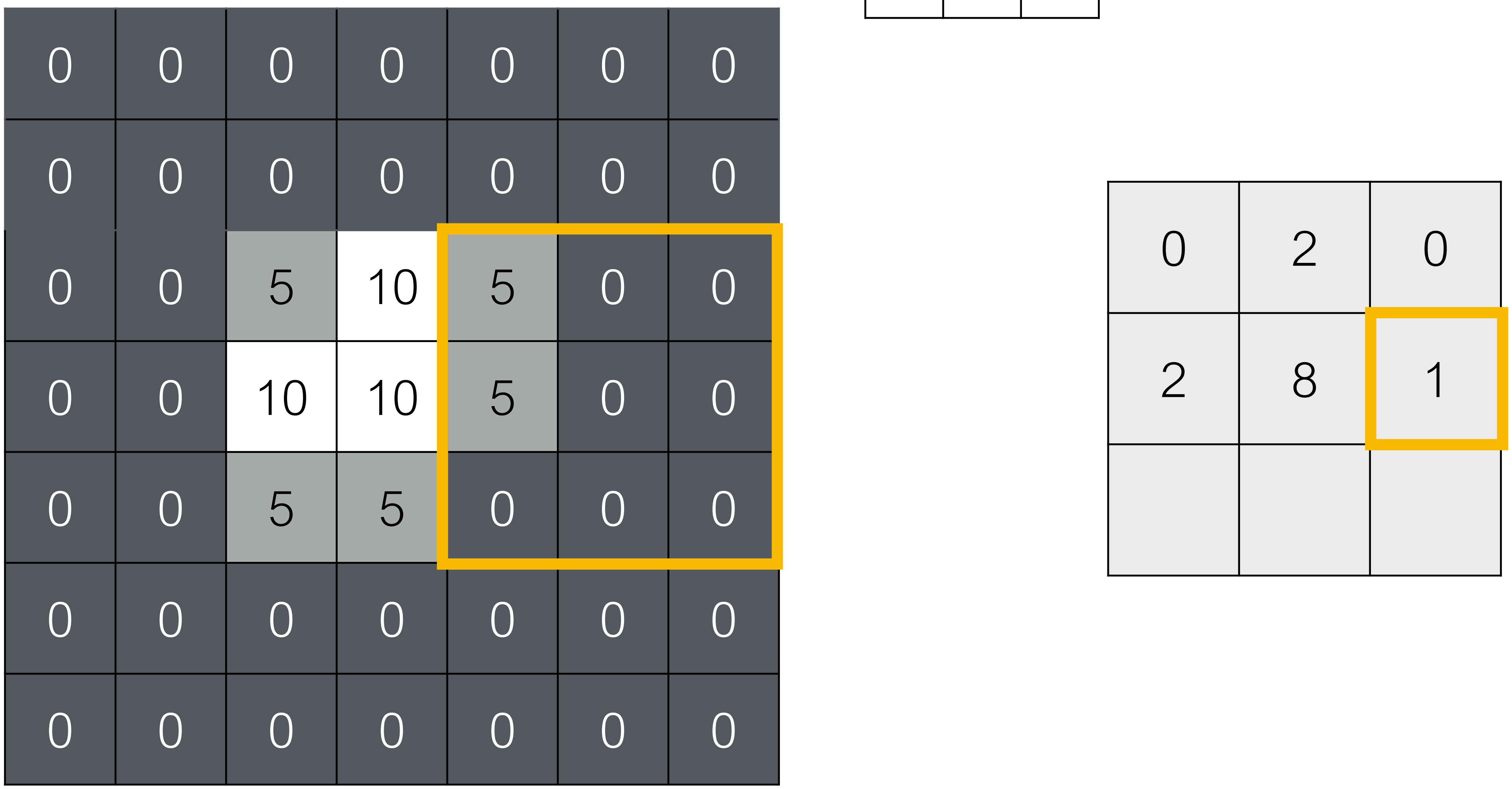
I

0	2	0
2	8	

$I \star w$

Learnable Alternative: Strided Filtering

$$w = \frac{1}{5}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

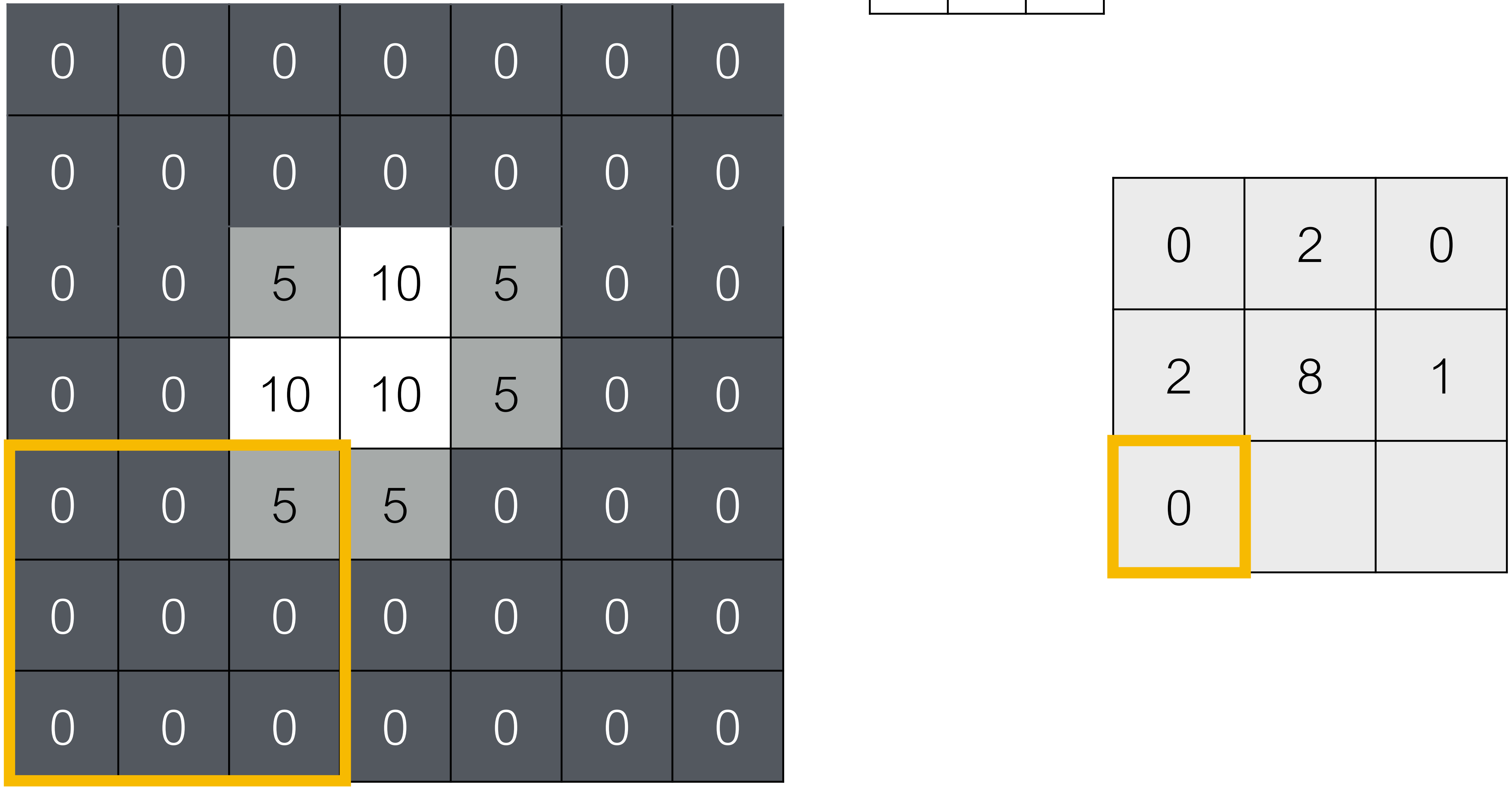
I

0	2	0
2	8	1

$I \star w$

Learnable Alternative: Strided Filtering

$$w = \frac{1}{5}\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 1 & 1 \\ \hline 0 & 1 & 0 \\ \hline\end{array}$$



0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

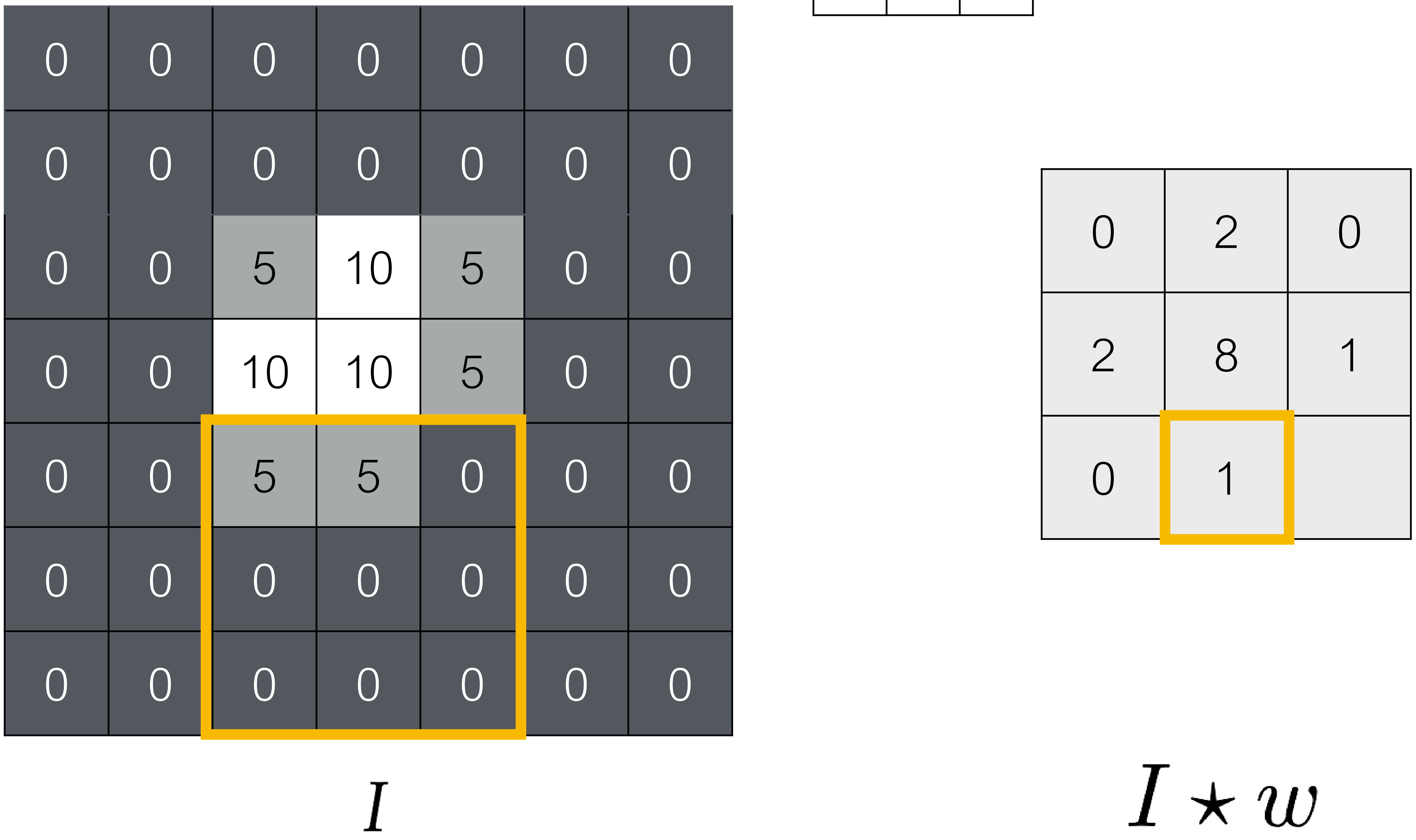
I

0	2	0
2	8	1
0		

$I \star w$

Learnable Alternative: Strided Filtering

$$w = \frac{1}{5}\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 1 & 1 \\ \hline 0 & 1 & 0 \\ \hline\end{array}$$



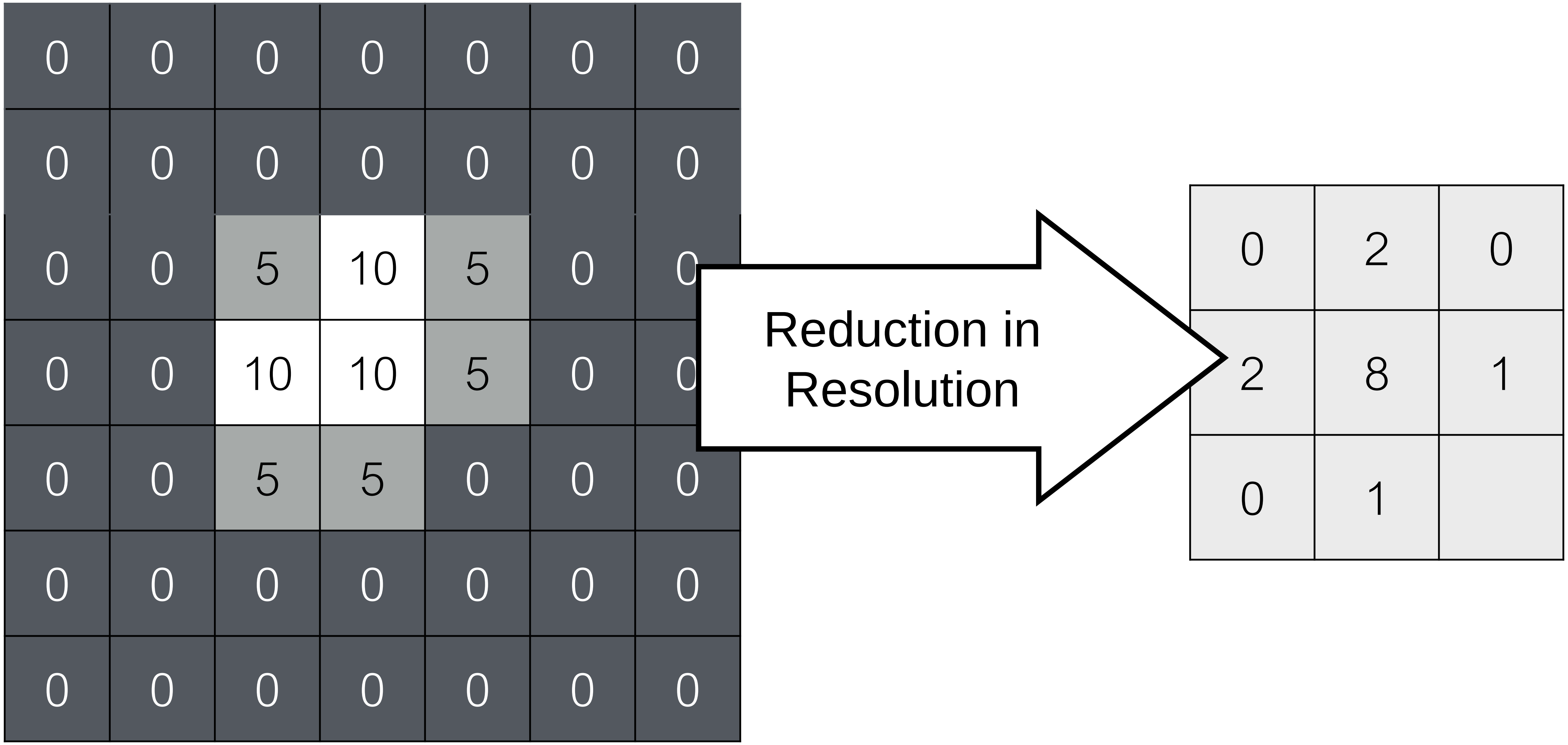
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

I

0	2	0
2	8	1
0	1	

$I \star w$

Strided Convolutions

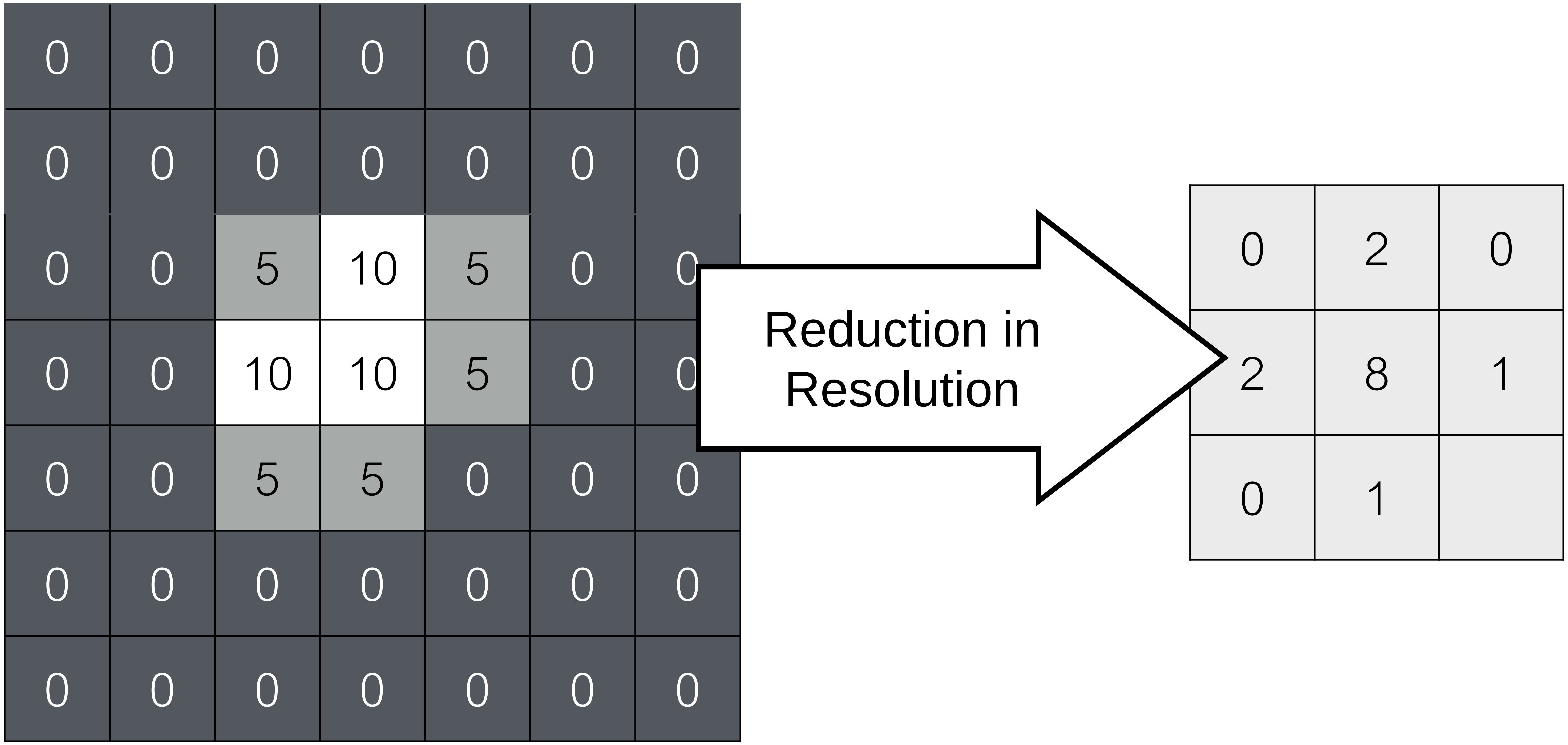


0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Reduction in Resolution

0	2	0
2	8	1
0	1	

Strided Convolutions



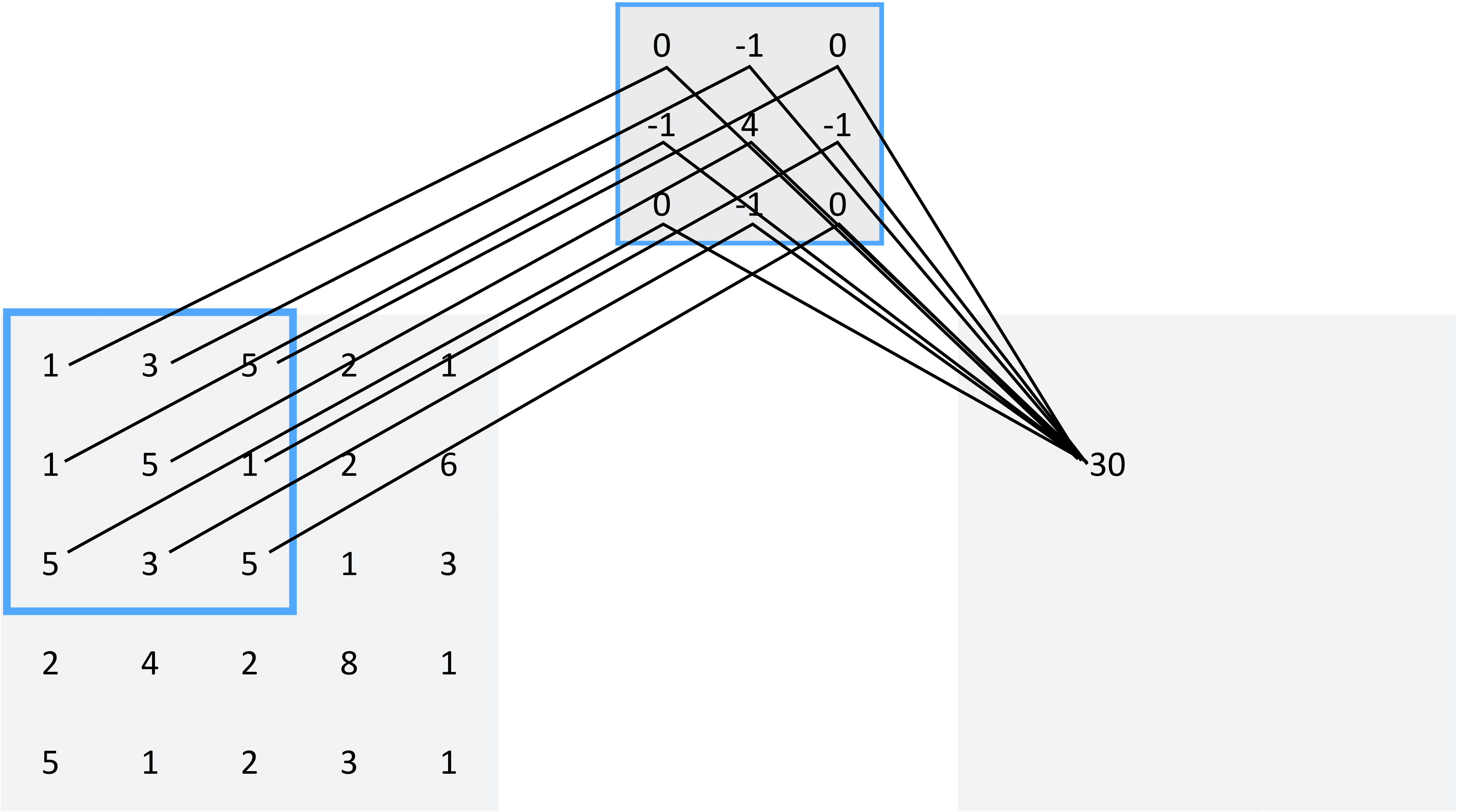
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Reduction in Resolution

0	2	0
2	8	1
0	1	

What if we want to **increase** resolution?

Different View on Strided Convolutions

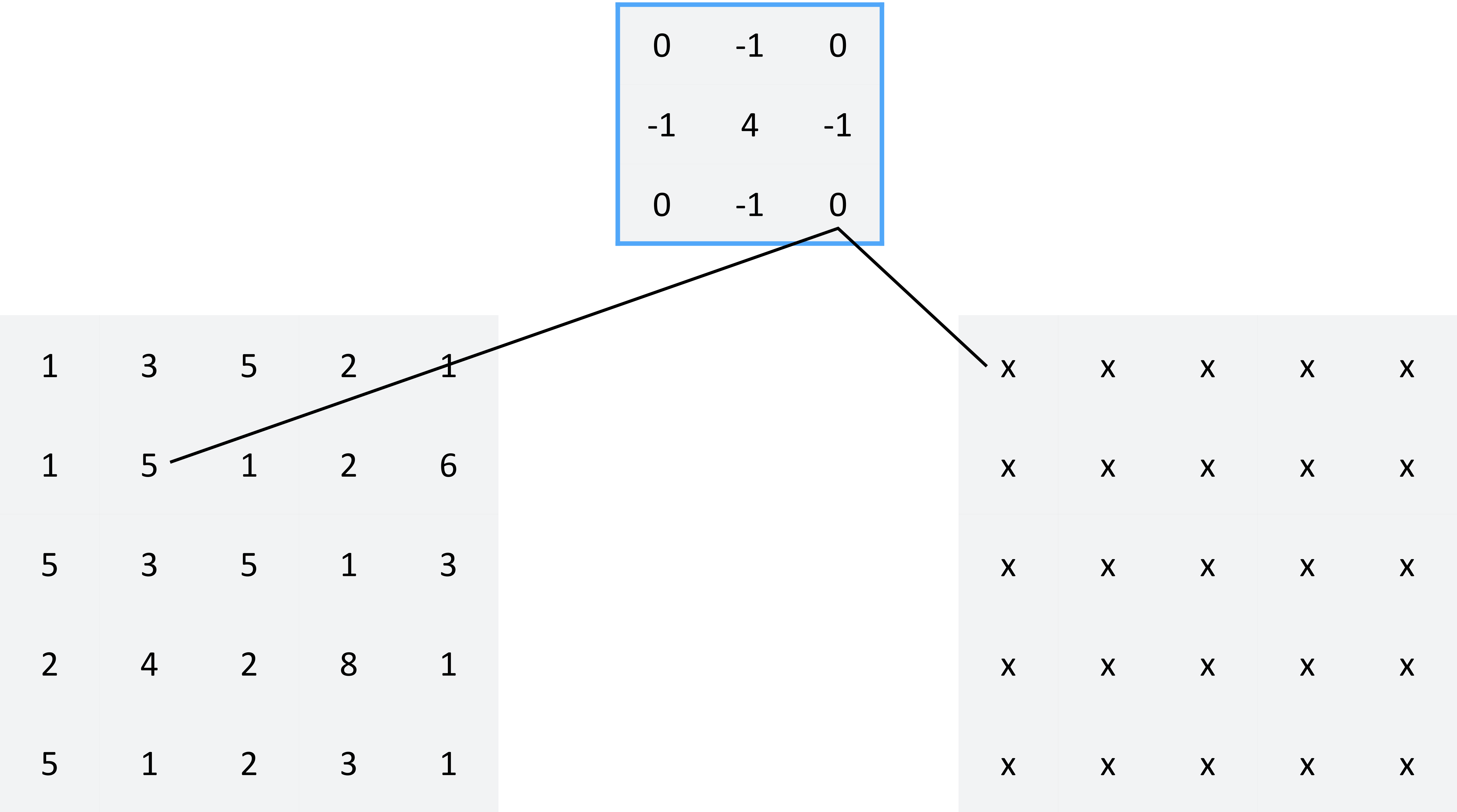


0	-1	0
-1	4	-1
0	-1	0

1	3	5	2	1
1	5	1	2	6
5	3	5	1	3
2	4	2	8	1
5	1	2	3	1

Standard view: Multiple pixels contribute to one pixel

Different View on Strided Convolutions



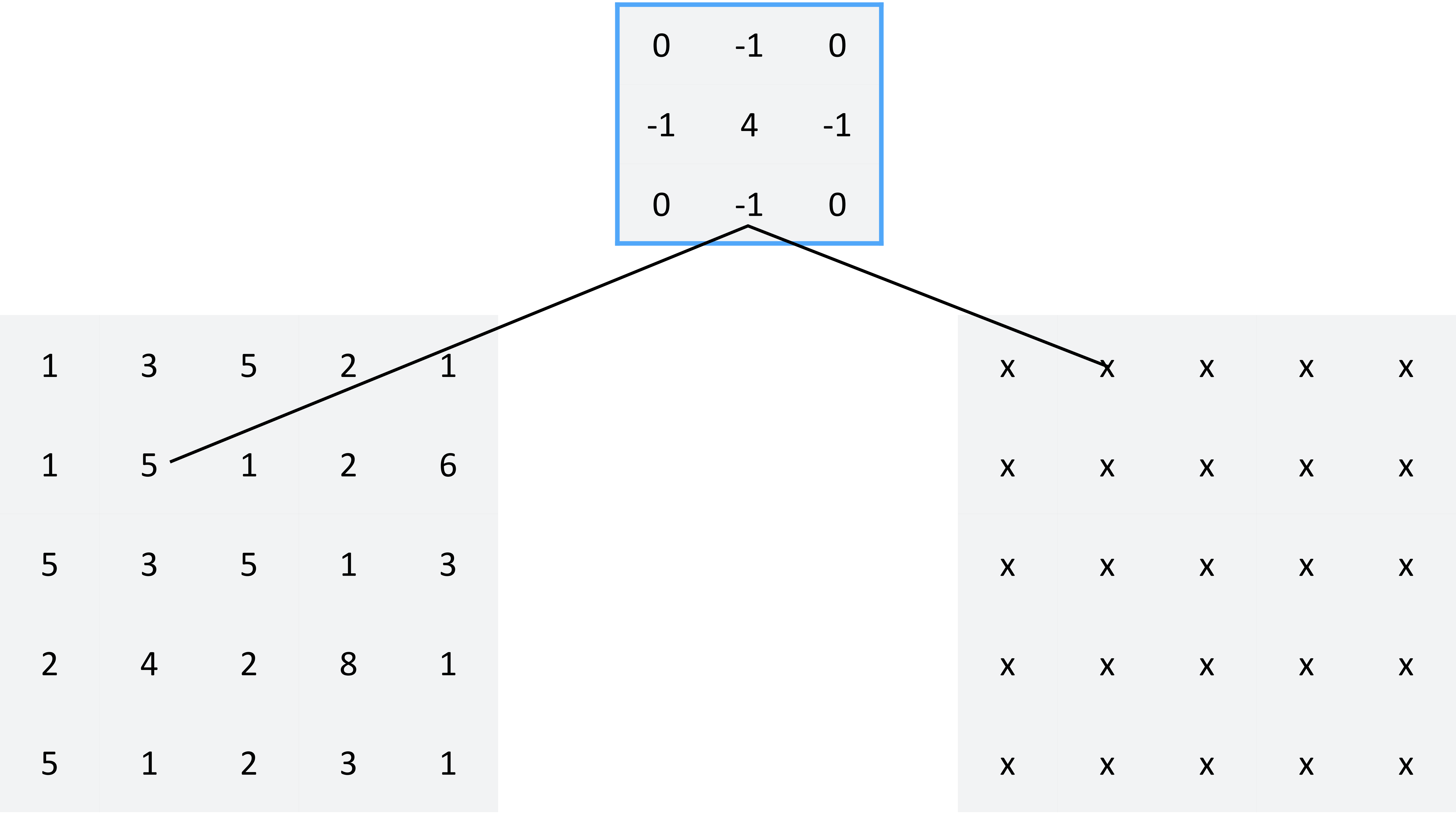
0	-1	0
-1	4	-1
0	-1	0

1	3	5	2	1
1	5	1	2	6
5	3	5	1	3
2	4	2	8	1
5	1	2	3	1

x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x

Or: One pixel contributes to many pixels

Different View on Strided Convolutions



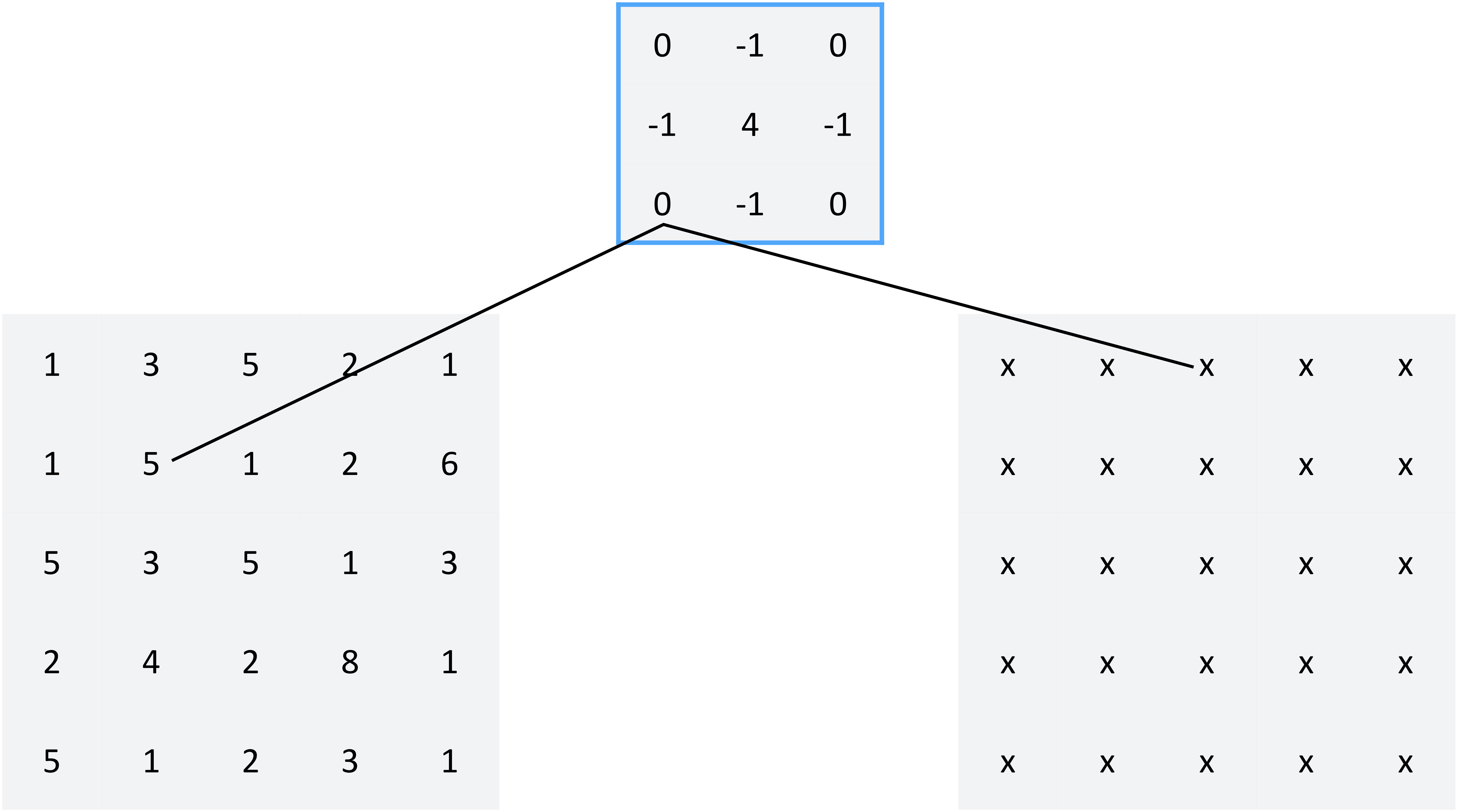
0	-1	0
-1	4	-1
0	-1	0

1	3	5	2	1
1	5	1	2	6
5	3	5	1	3
2	4	2	8	1
5	1	2	3	1

x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x

Or: One pixel contributes to many pixels

Different View on Strided Convolutions



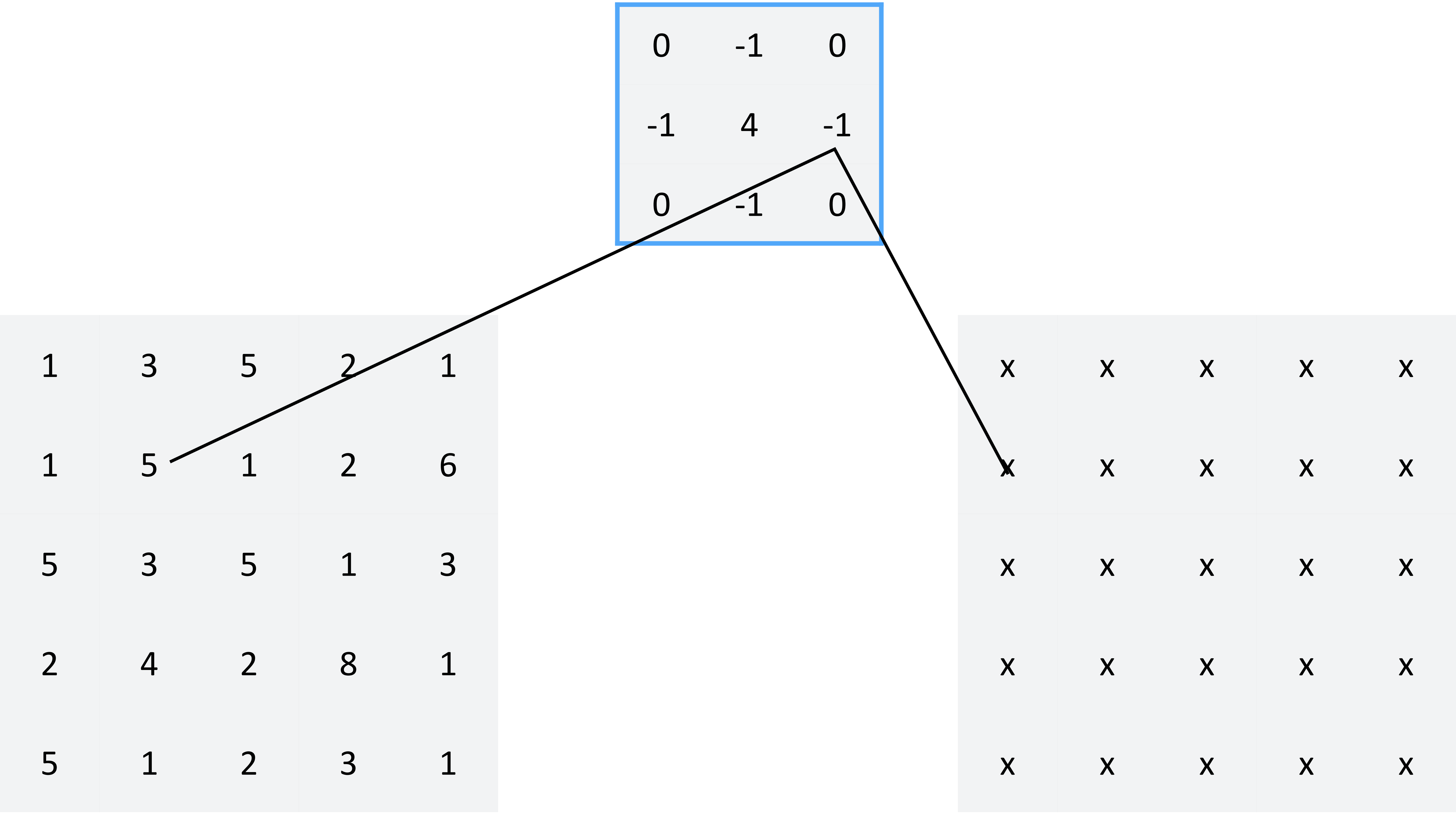
0	-1	0
-1	4	-1
0	-1	0

1	3	5	2	1
1	5	1	2	6
5	3	5	1	3
2	4	2	8	1
5	1	2	3	1

x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x

Or: One pixel contributes to many pixels

Different View on Strided Convolutions



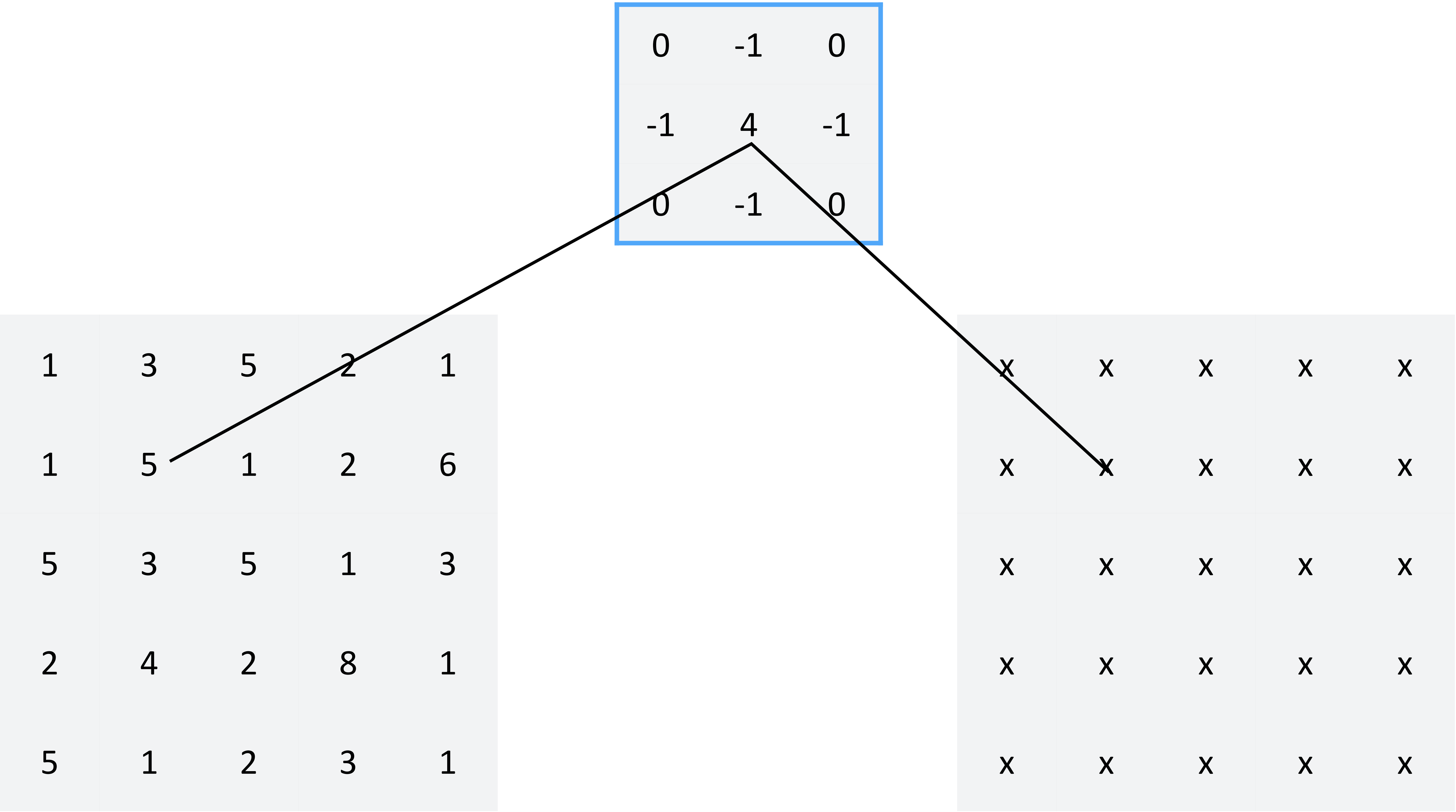
0	-1	0
-1	4	-1
0	-1	0

1	3	5	2	1
1	5	1	2	6
5	3	5	1	3
2	4	2	8	1
5	1	2	3	1

x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x

Or: One pixel contributes to many pixels

Different View on Strided Convolutions



0	-1	0
-1	4	-1
0	-1	0

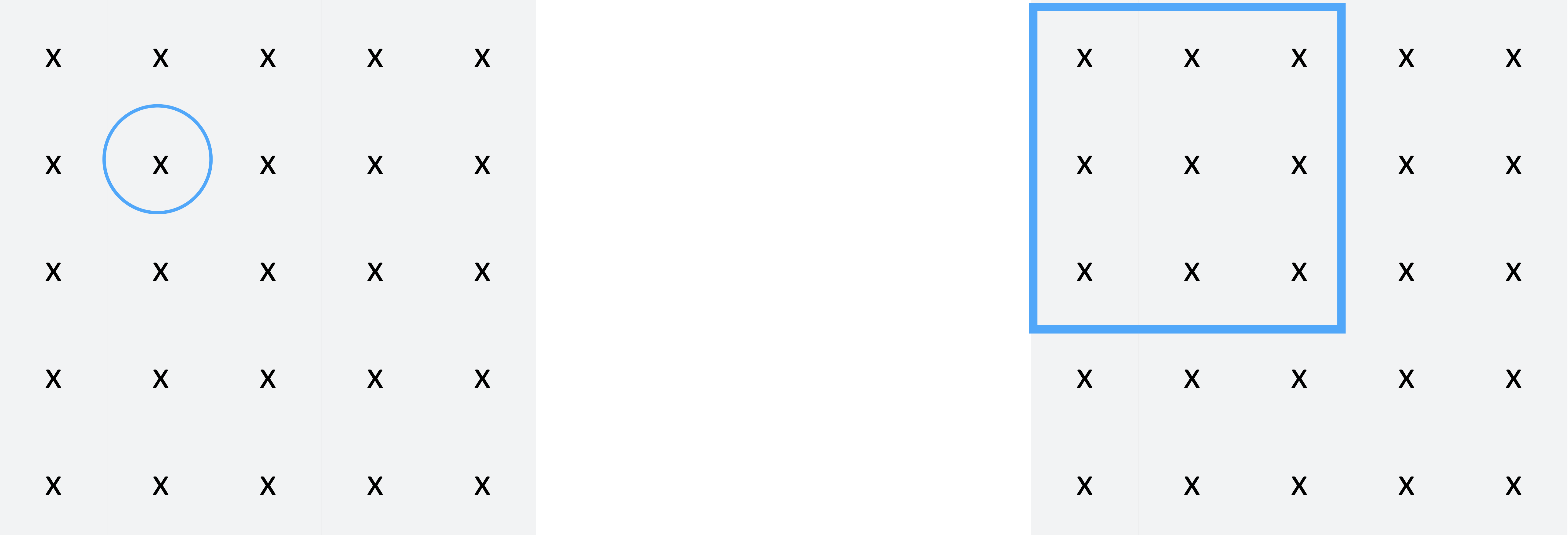
1	3	5	2	1
1	5	1	2	6
5	3	5	1	3
2	4	2	8	1
5	1	2	3	1

x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x

Or: One pixel contributes to many pixels

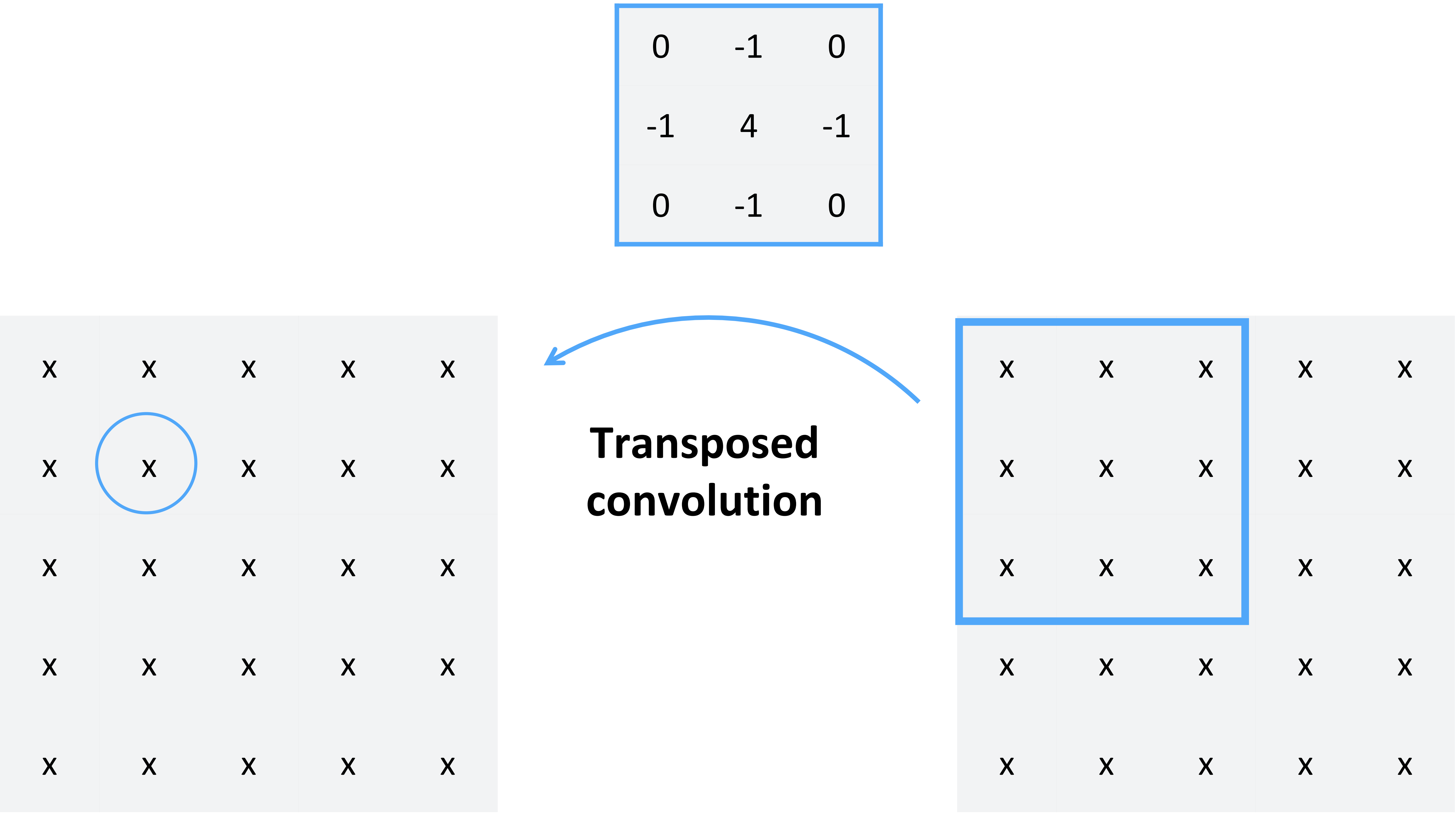
Different View on Strided Convolutions

0	-1	0
-1	4	-1
0	-1	0



Or: One pixel contributes to many pixels

Different View on Strided Convolutions

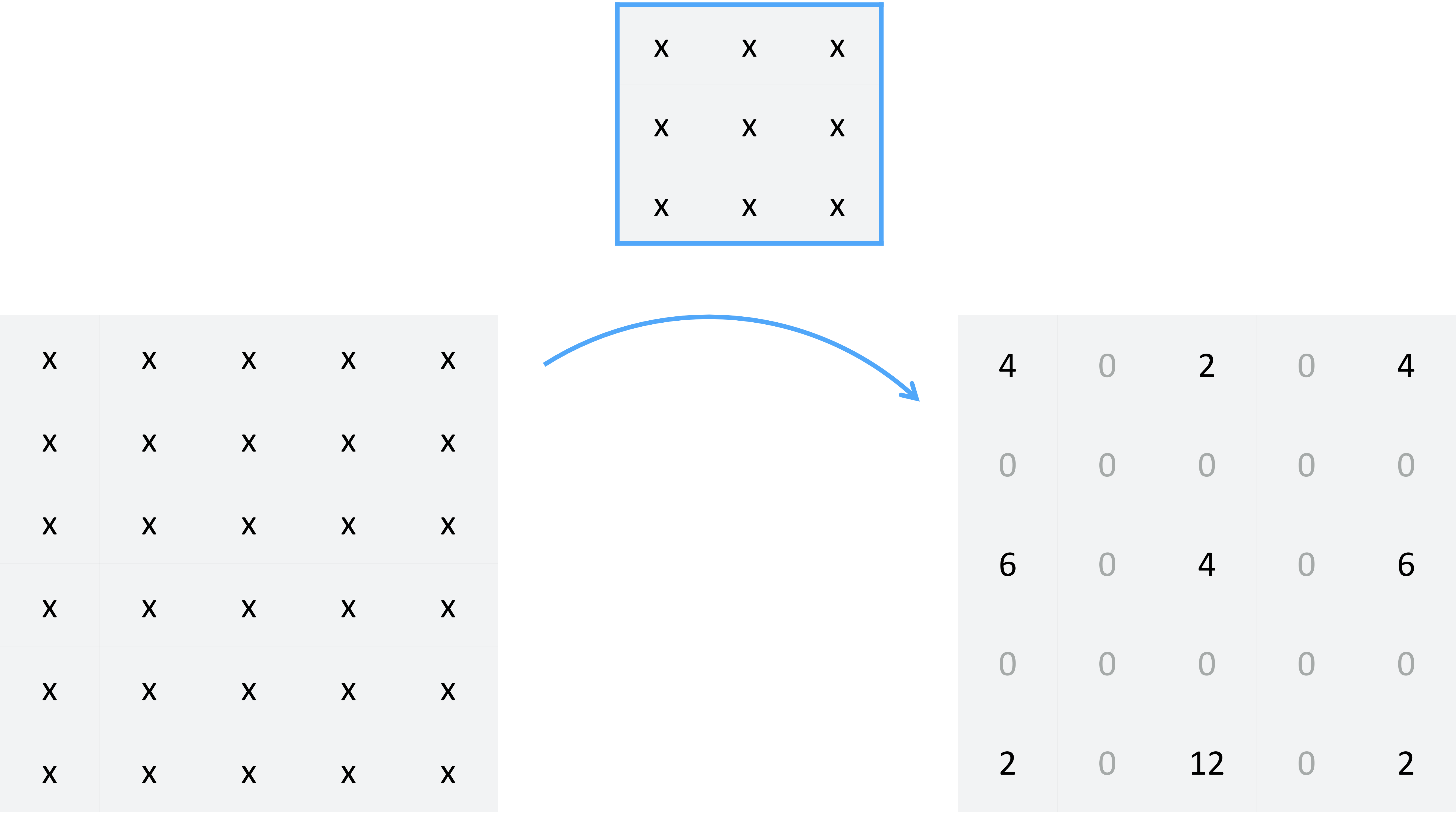


0	-1	0
-1	4	-1
0	-1	0

Transposed convolution

Or: One pixel contributes to many pixels

What’s the point?



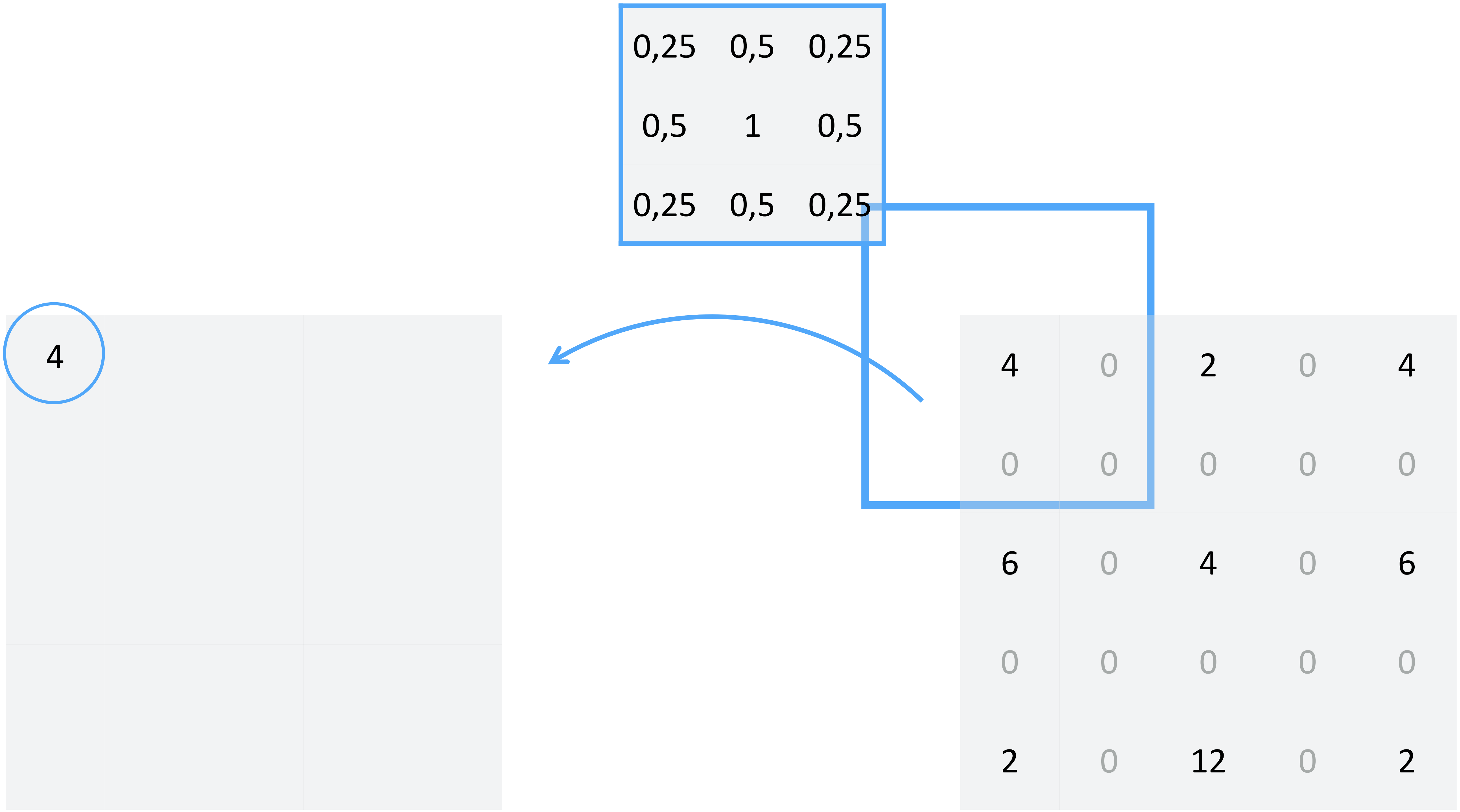
x	x	x
x	x	x
x	x	x

x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x

4	0	2	0	4
0	0	0	0	0
6	0	4	0	6
0	0	0	0	0
2	0	12	0	2

Strided filtering reduces resolution

What’s the point?



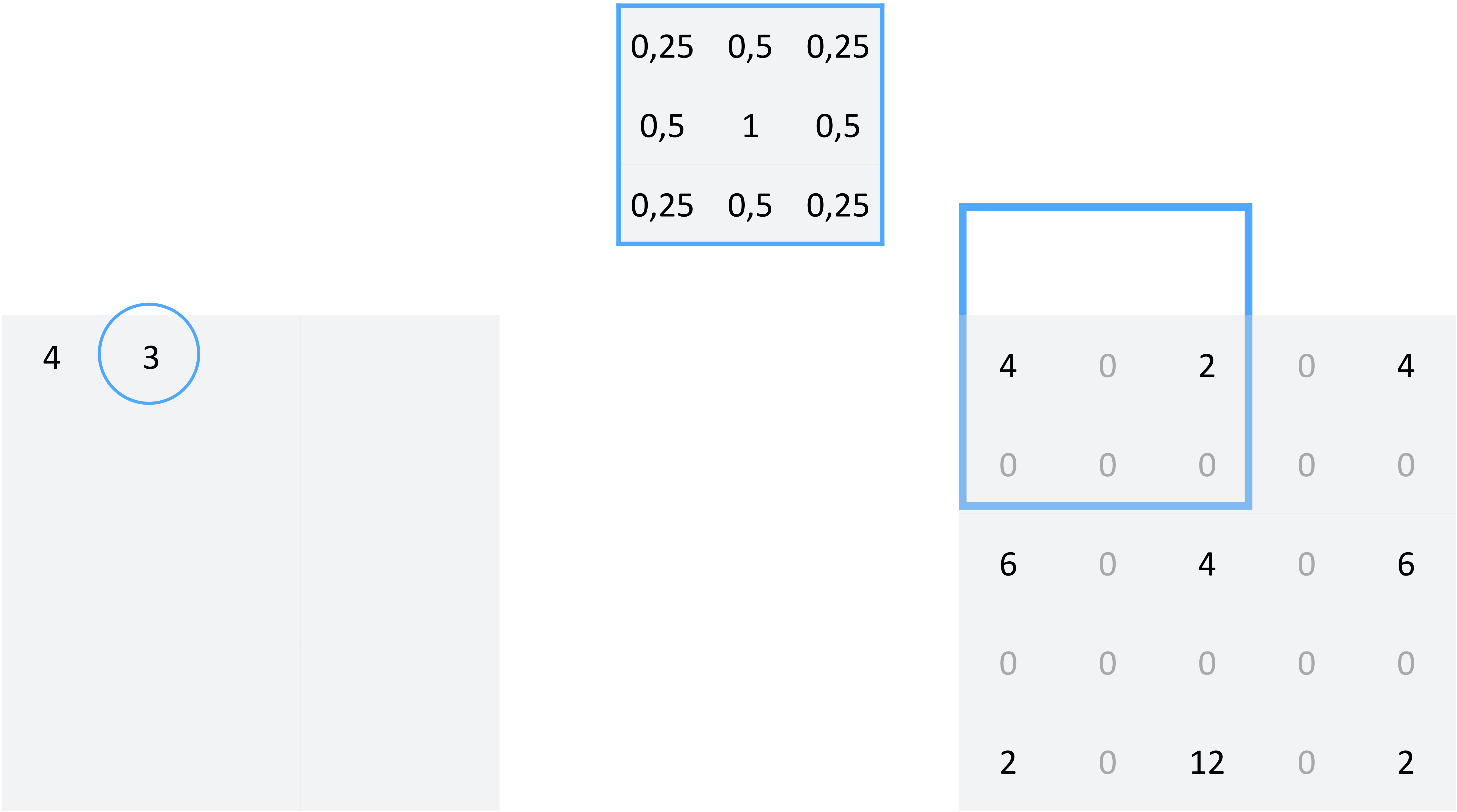
0,25	0,5	0,25
0,5	1	0,5
0,25	0,5	0,25

4				

4	0	2	0	4
0	0	0	0	0
6	0	4	0	6
0	0	0	0	0
2	0	12	0	2

Transposed strided filtering increases resolution

What's the point?



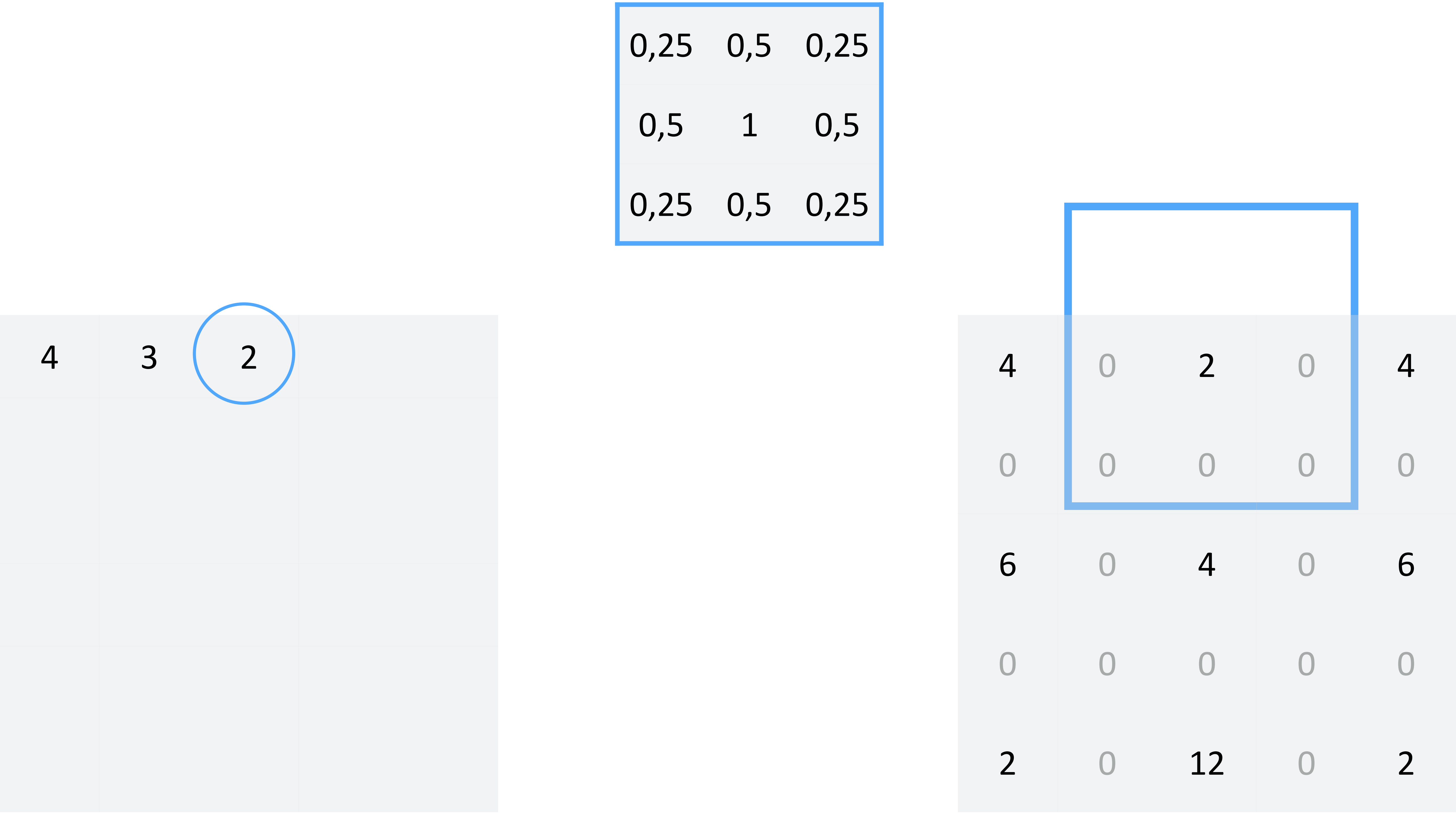
0,25	0,5	0,25
0,5	1	0,5
0,25	0,5	0,25

4	3			

4	0	2	0	4
0	0	0	0	0
6	0	4	0	6
0	0	0	0	0
2	0	12	0	2

Transposed strided filtering increases resolution

What's the point?



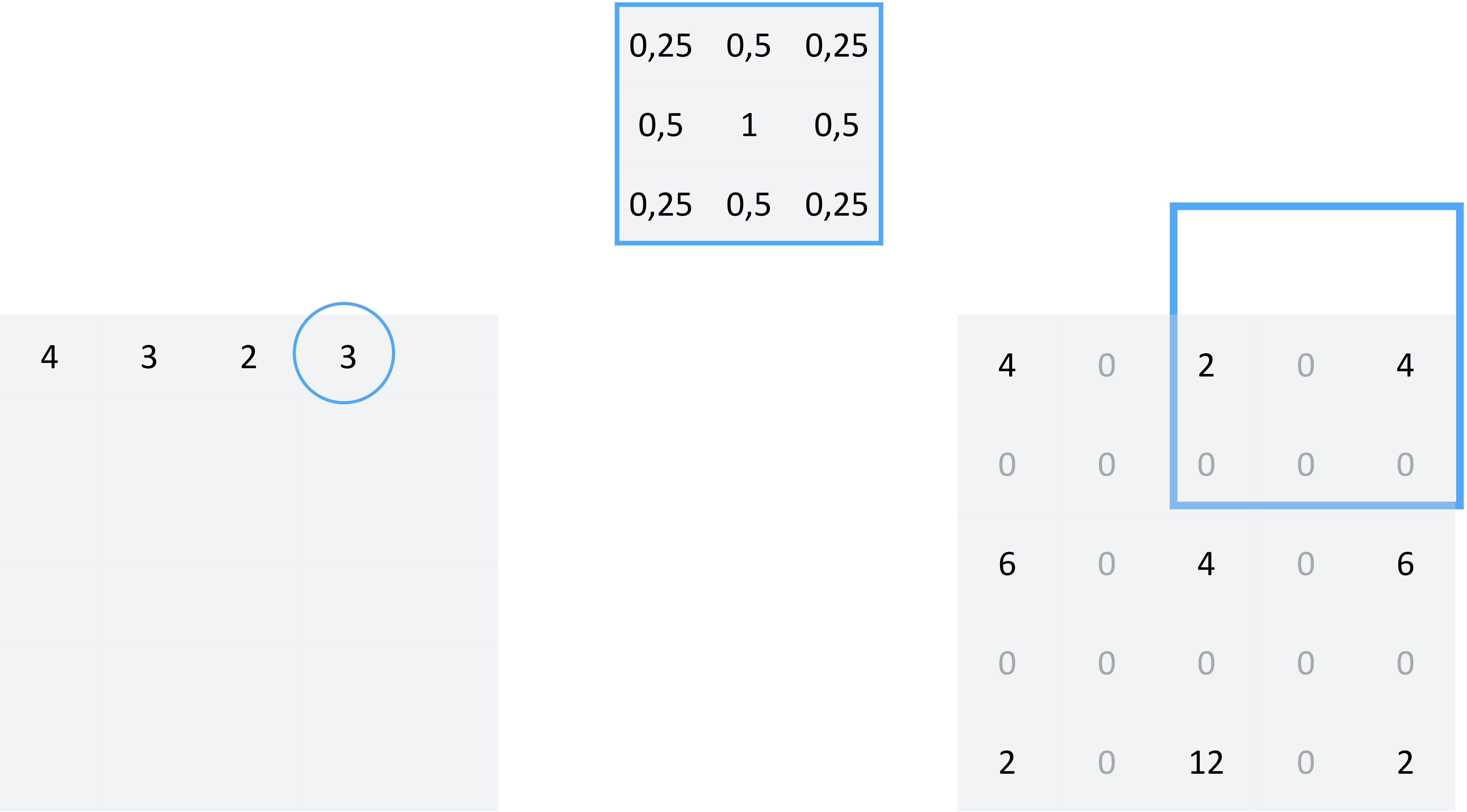
0,25	0,5	0,25
0,5	1	0,5
0,25	0,5	0,25

4	3	2		

4	0	2	0	4
0	0	0	0	0
6	0	4	0	6
0	0	0	0	0
2	0	12	0	2

Transposed strided filtering increases resolution

What's the point?



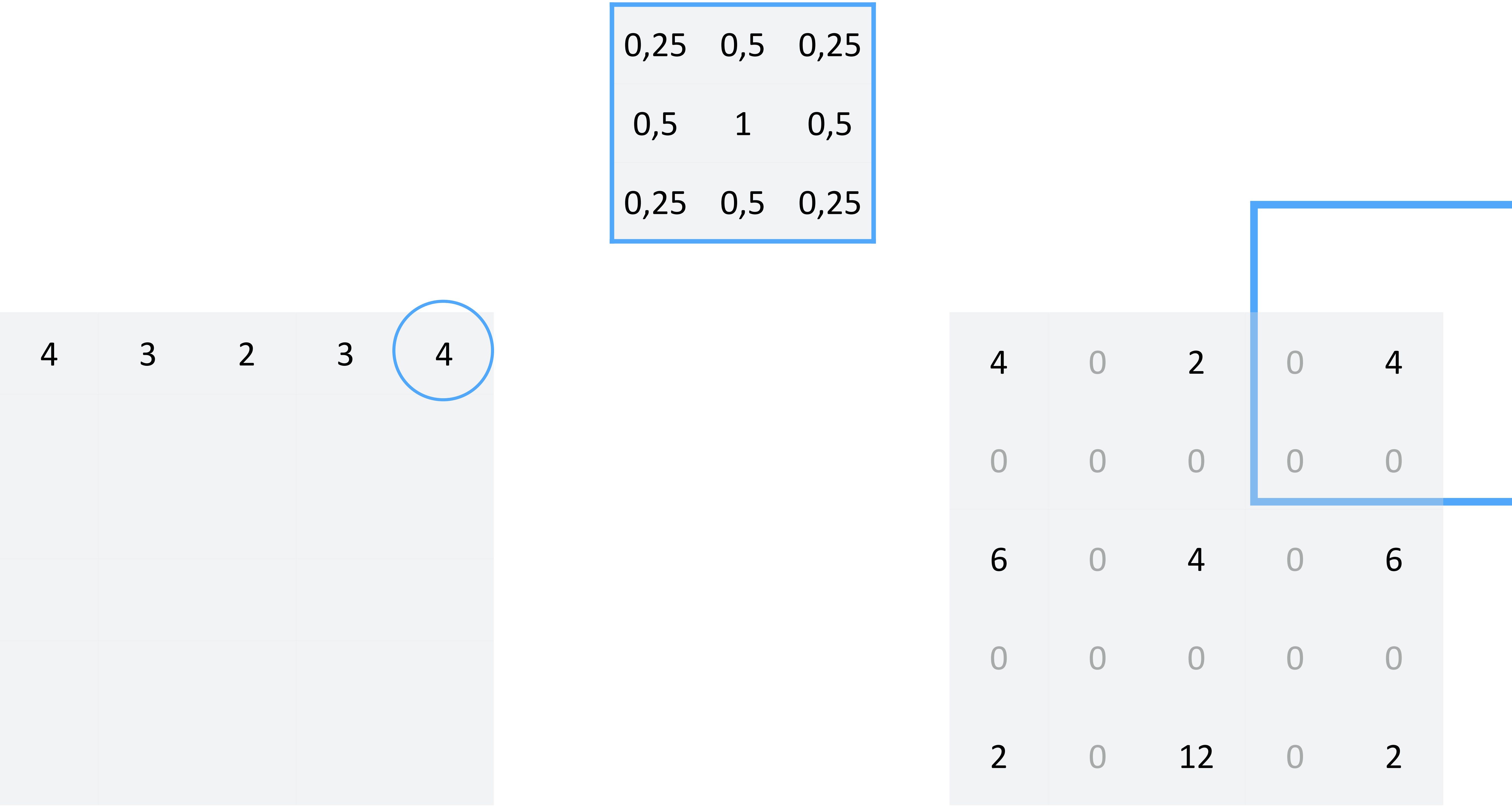
0,25	0,5	0,25
0,5	1	0,5
0,25	0,5	0,25

4	3	2	3	

4	0	2	0	4
0	0	0	0	0
6	0	4	0	6
0	0	0	0	0
2	0	12	0	2

Transposed strided filtering increases resolution

What’s the point?



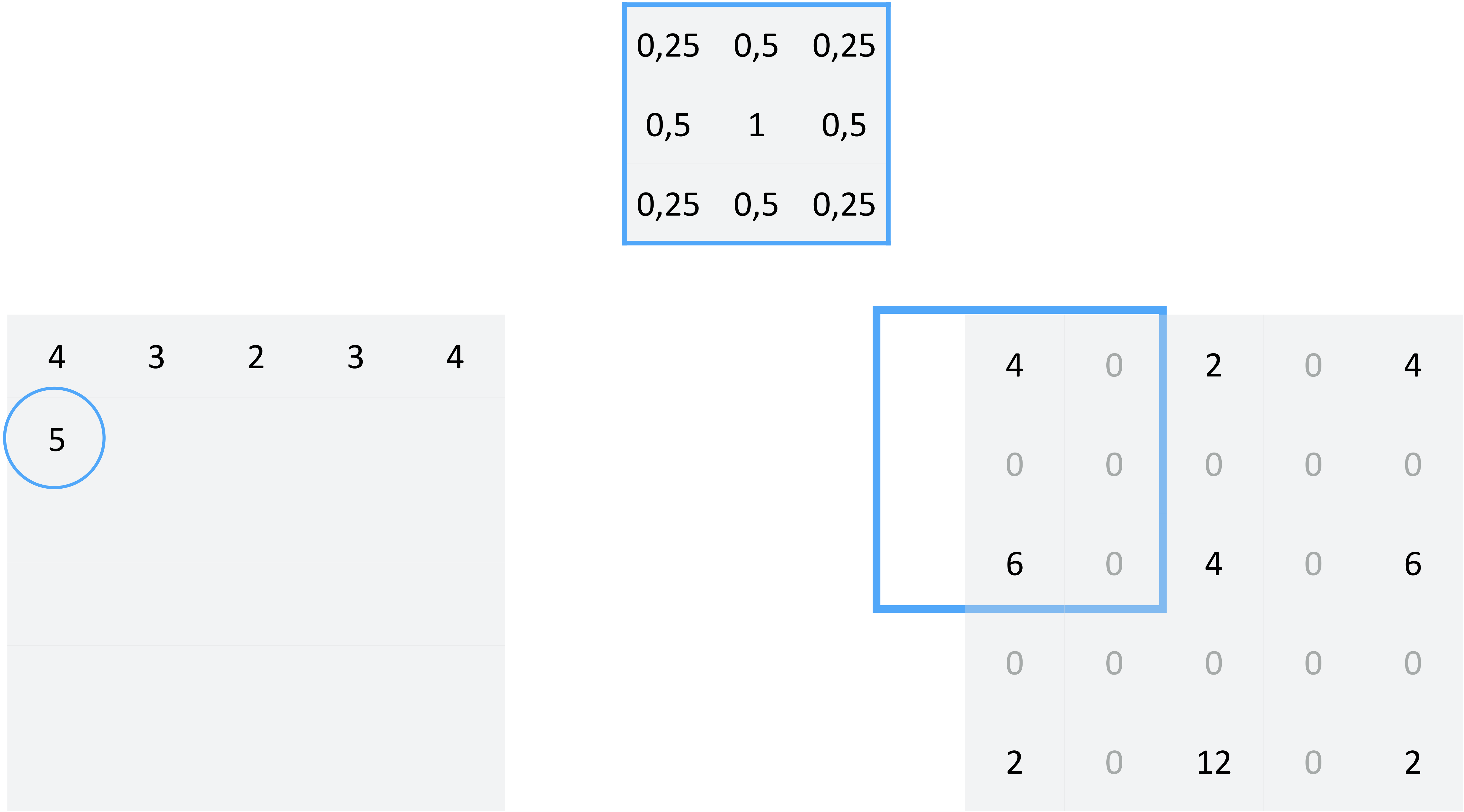
0,25	0,5	0,25
0,5	1	0,5
0,25	0,5	0,25

4	3	2	3	4

4	0	2	0	4
0	0	0	0	0
6	0	4	0	6
0	0	0	0	0
2	0	12	0	2

Transposed strided filtering increases resolution

What’s the point?



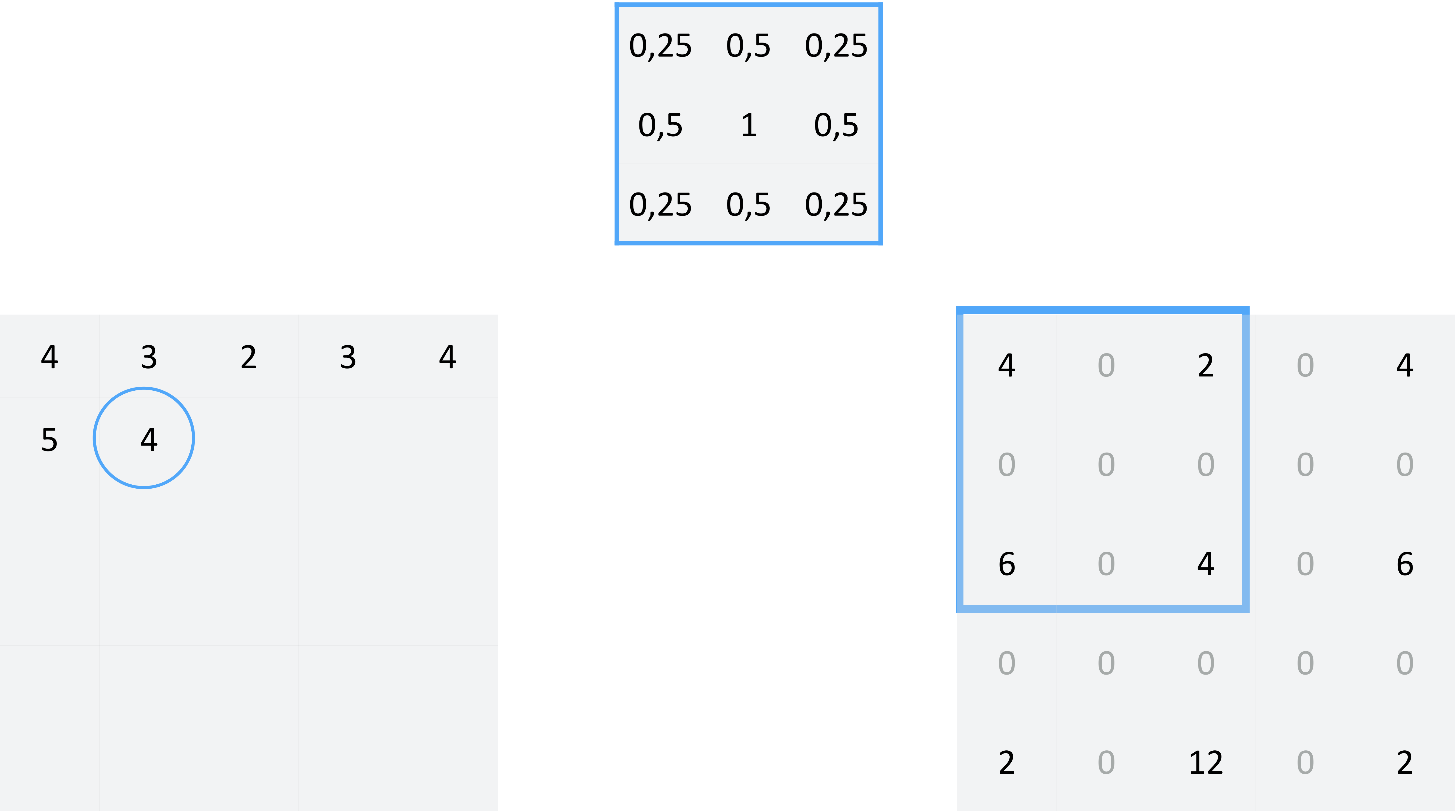
0,25	0,5	0,25
0,5	1	0,5
0,25	0,5	0,25

4	3	2	3	4
5				

4	0	2	0	4
0	0	0	0	0
6	0	4	0	6
0	0	0	0	0
2	0	12	0	2

Transposed strided filtering increases resolution

What’s the point?



0,25	0,5	0,25
0,5	1	0,5
0,25	0,5	0,25

4	3	2	3	4
5	4			

4	0	2	0	4
0	0	0	0	0
6	0	4	0	6
0	0	0	0	0
2	0	12	0	2

Transposed strided filtering increases resolution

What’s the point?

0,25	0,5	0,25
0,5	1	0,5
0,25	0,5	0,25

4	3	2	3	4
5	4	3	4	5
6	5	4	5	6
4	6	8	6	4
2	7	12	7	2

4	0	2	0	4
0	0	0	0	0
6	0	4	0	6
0	0	0	0	0
2	0	12	0	2

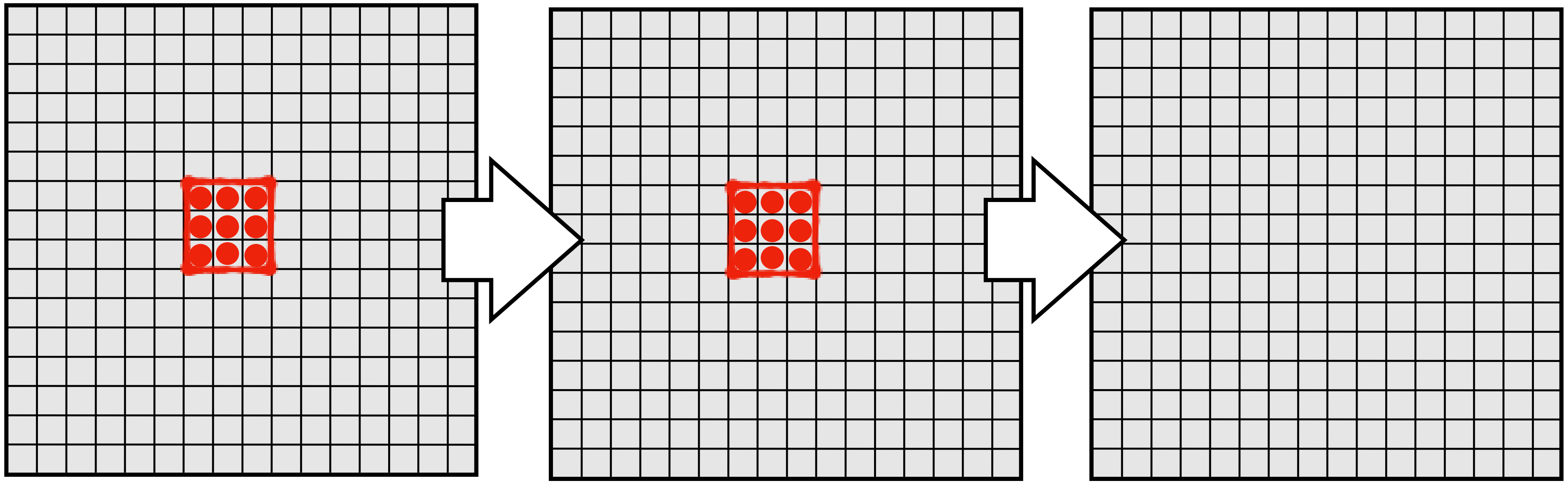
Transposed strided filtering increases resolution

Transposed Convolutions

- Not to be confused with matrix transpose
- Parameters are learned, not computed from the downsampling filtering stage
- Also known as:
  - fractionally strided convolutions
  - deconvolution (misleading, since transposed convolutions do not undo effect of convolution)

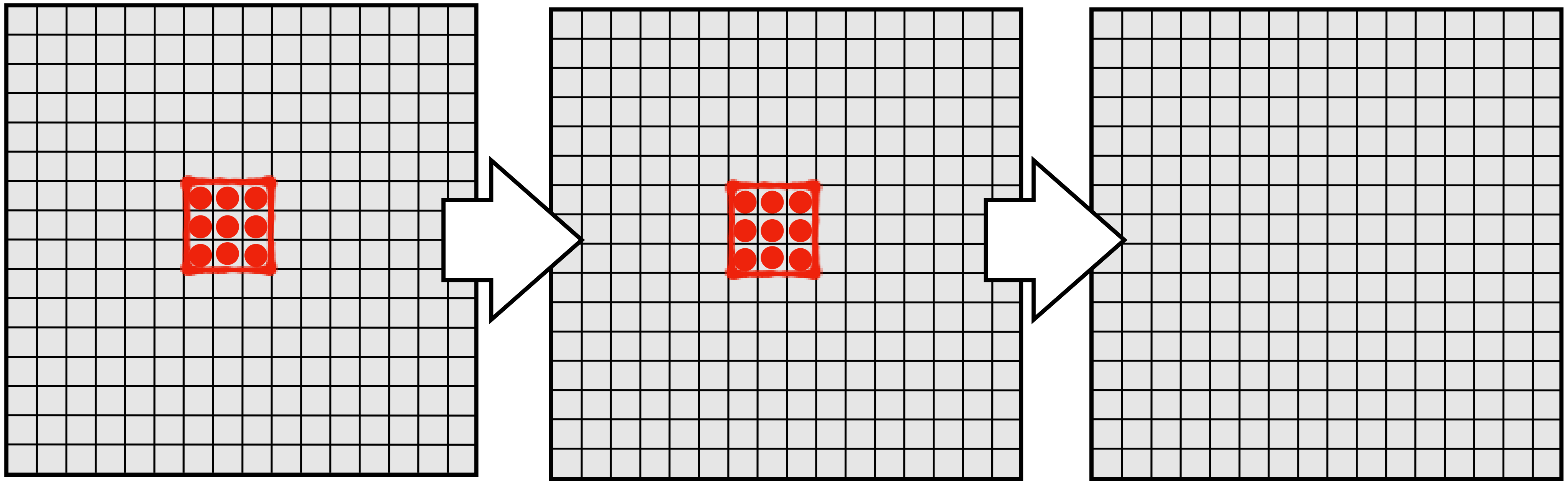
Receptive Field of Convolutions

Standard convolution:

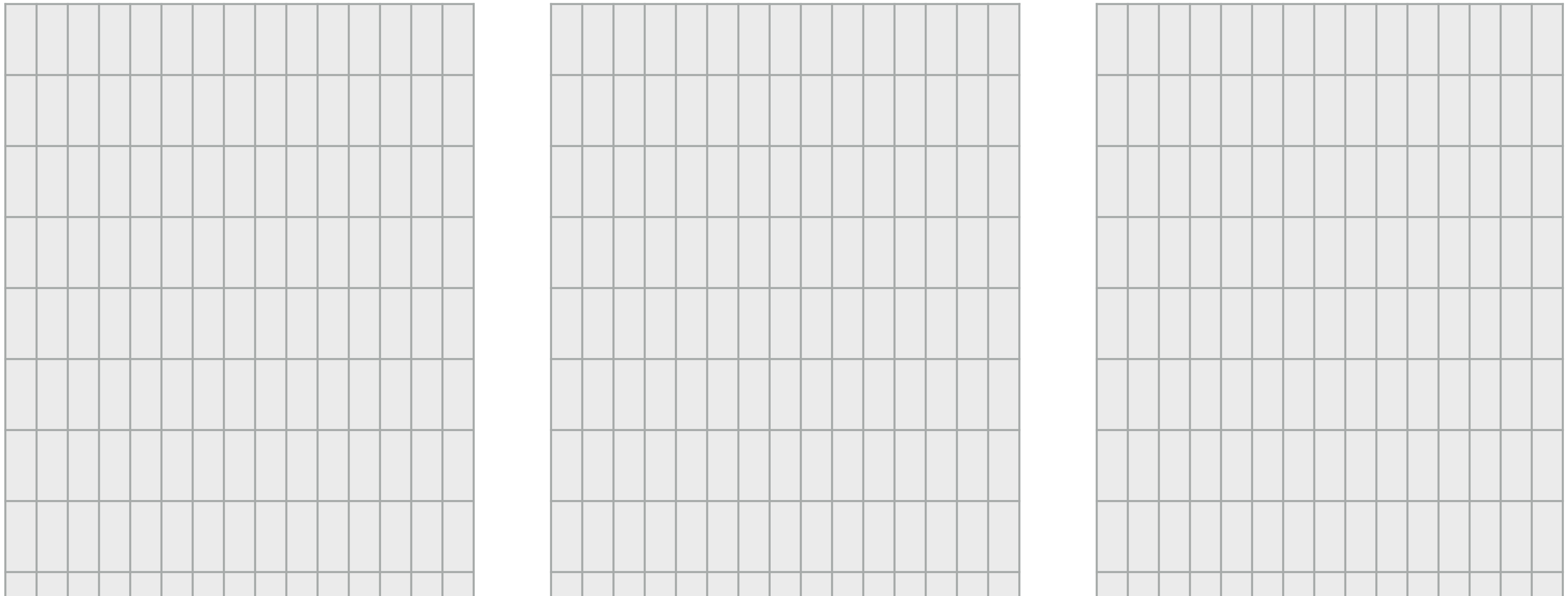


Receptive Field of Convolutions

Standard convolution:

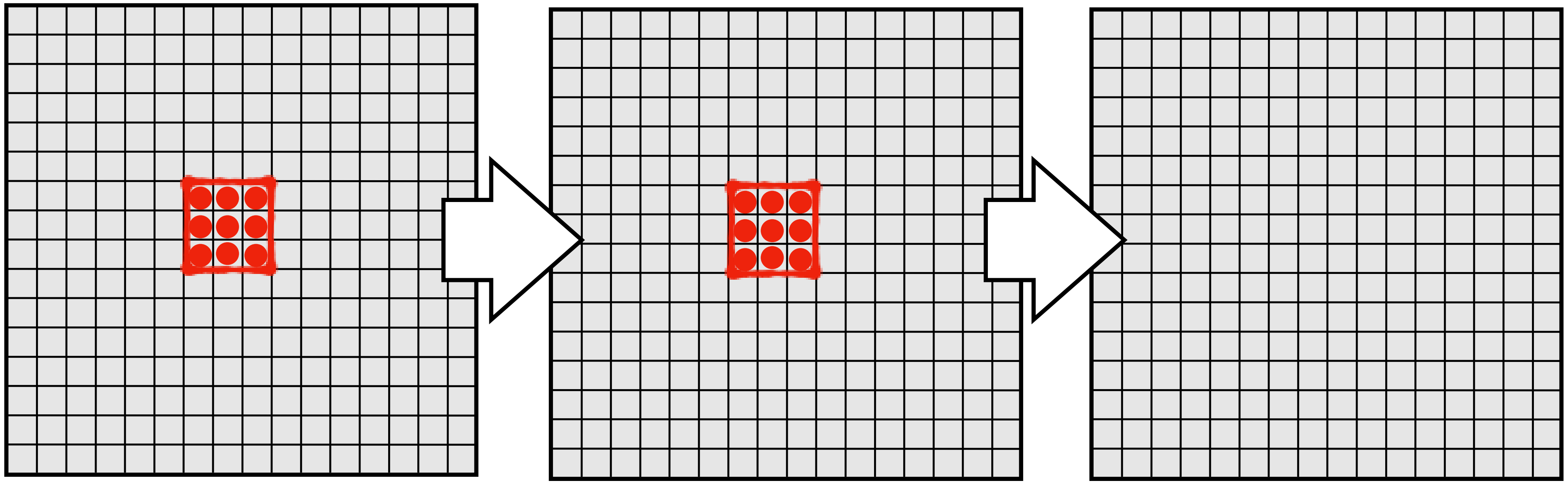


Receptive field:

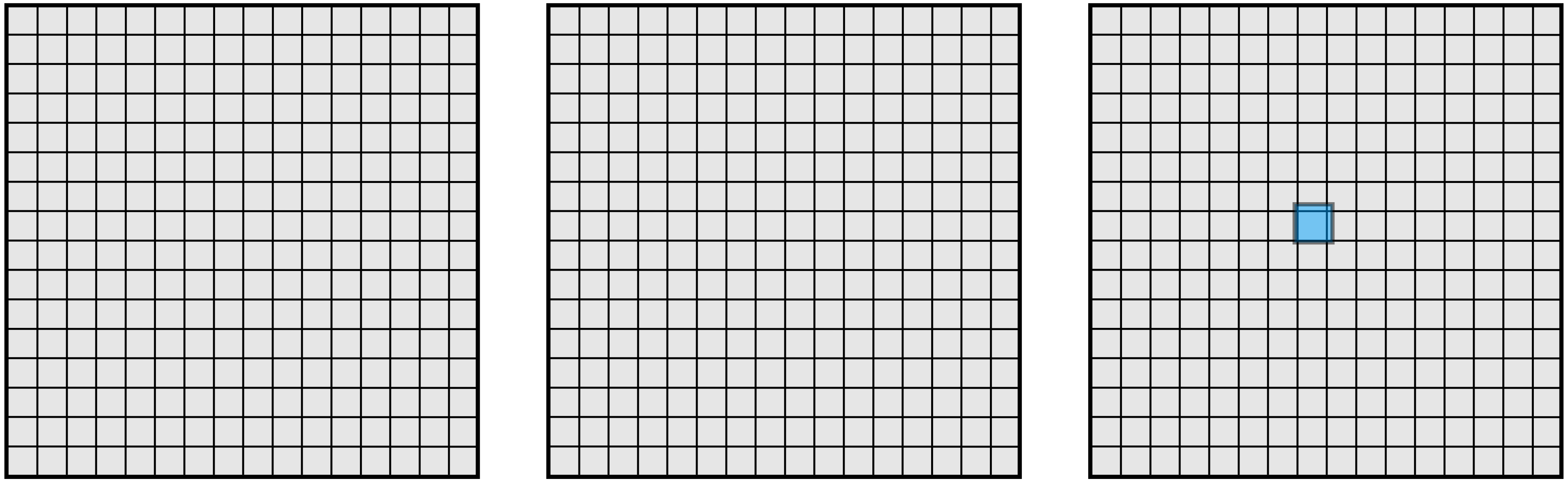


Receptive Field of Convolutions

Standard convolution:

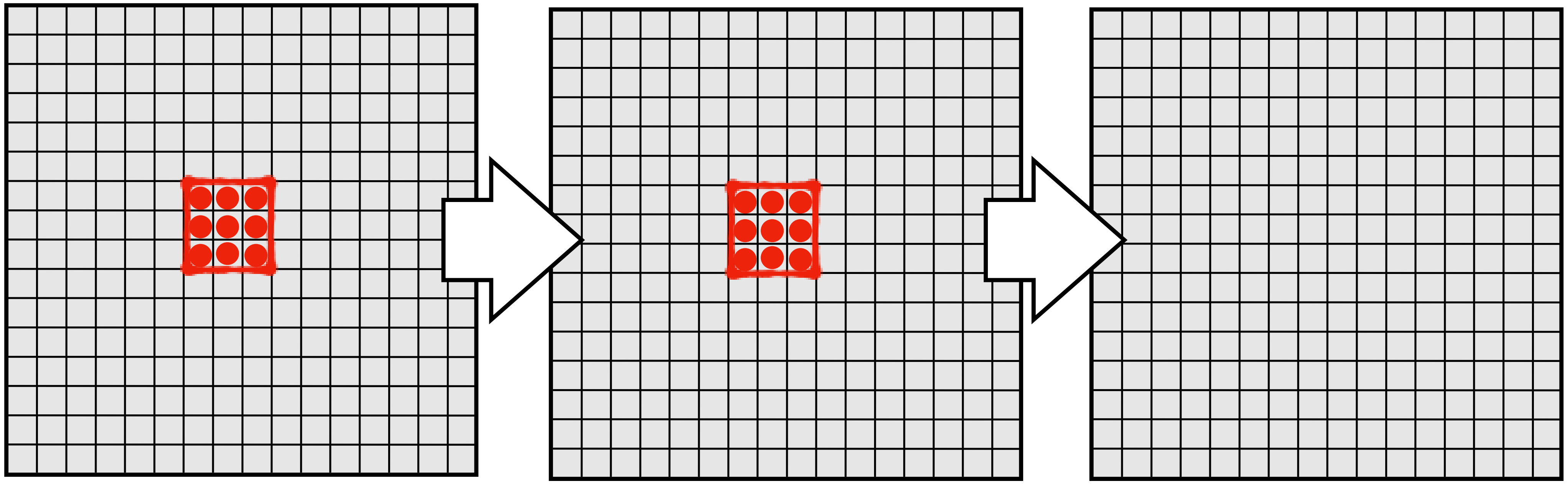


Receptive field:

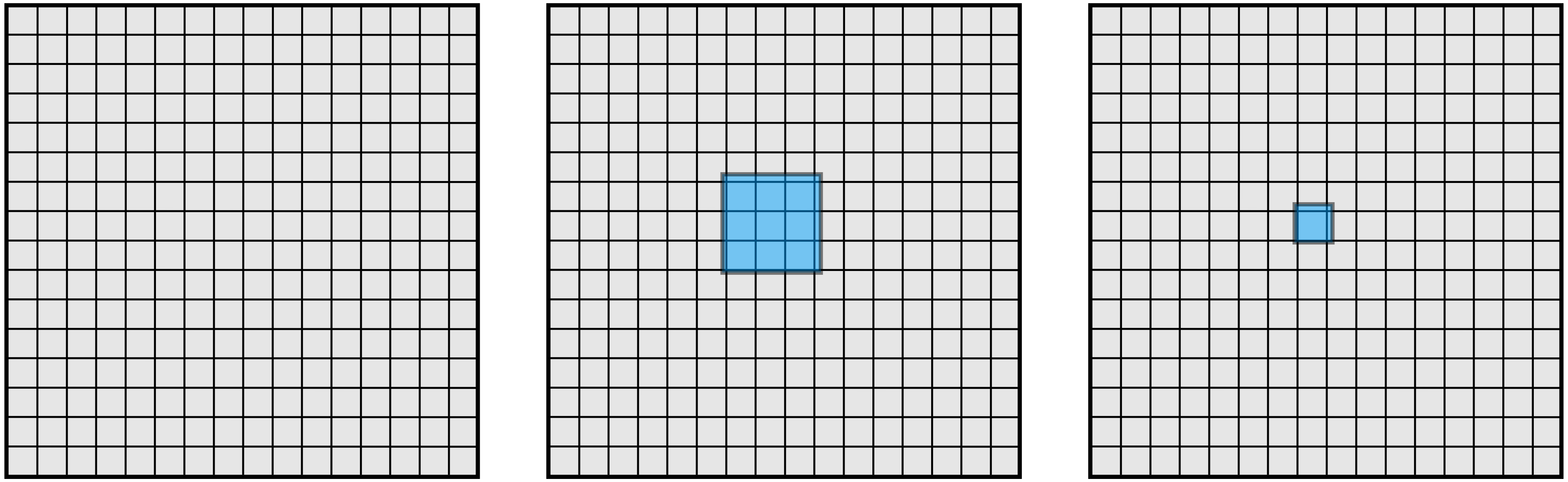


Receptive Field of Convolutions

Standard convolution:

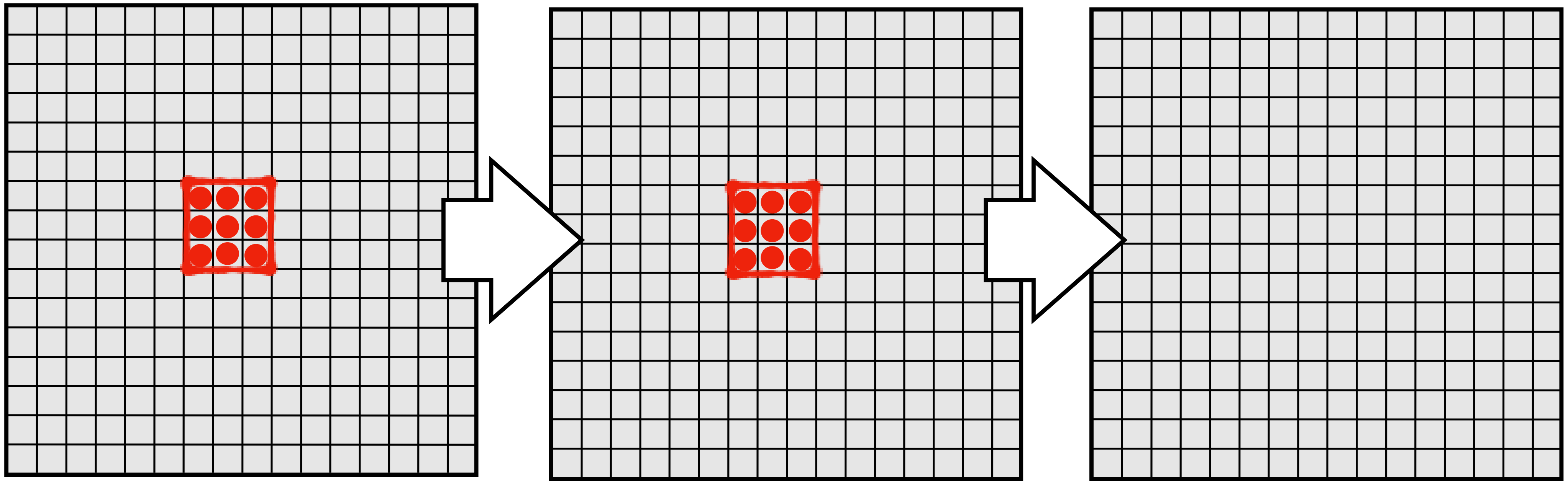


Receptive field:

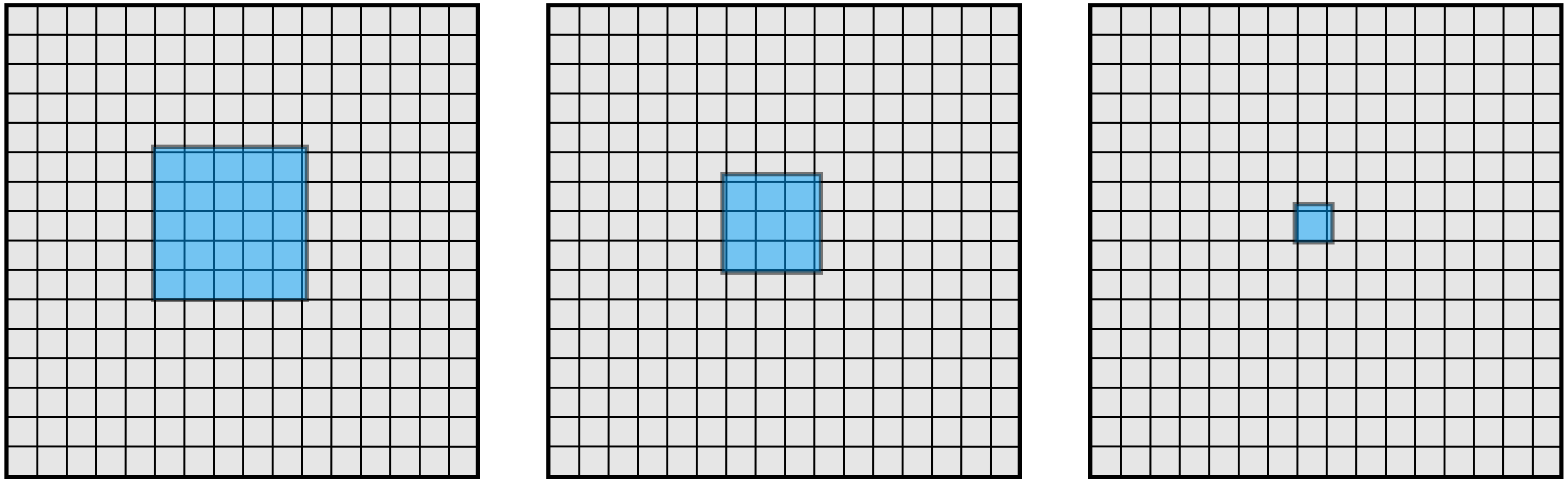


Receptive Field of Convolutions

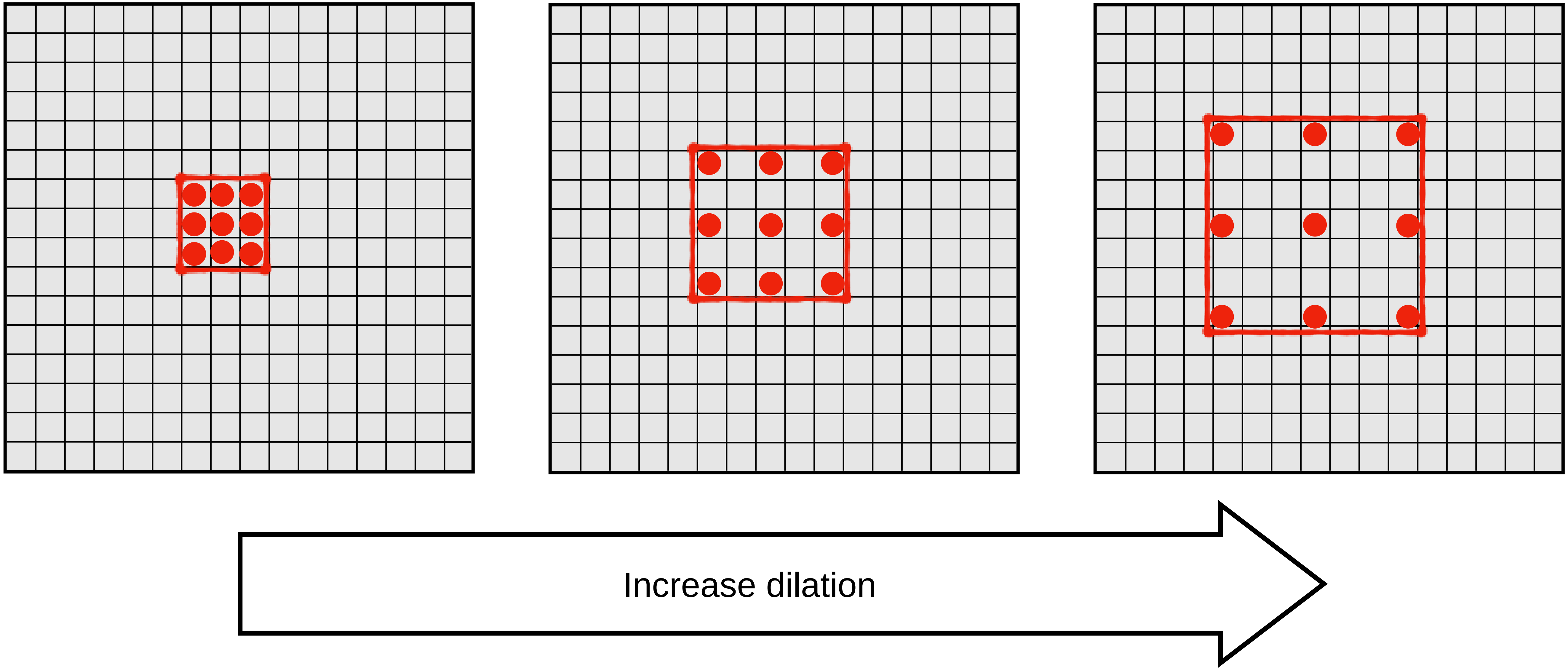
Standard convolution:



Receptive field:



Dilated Convolutions

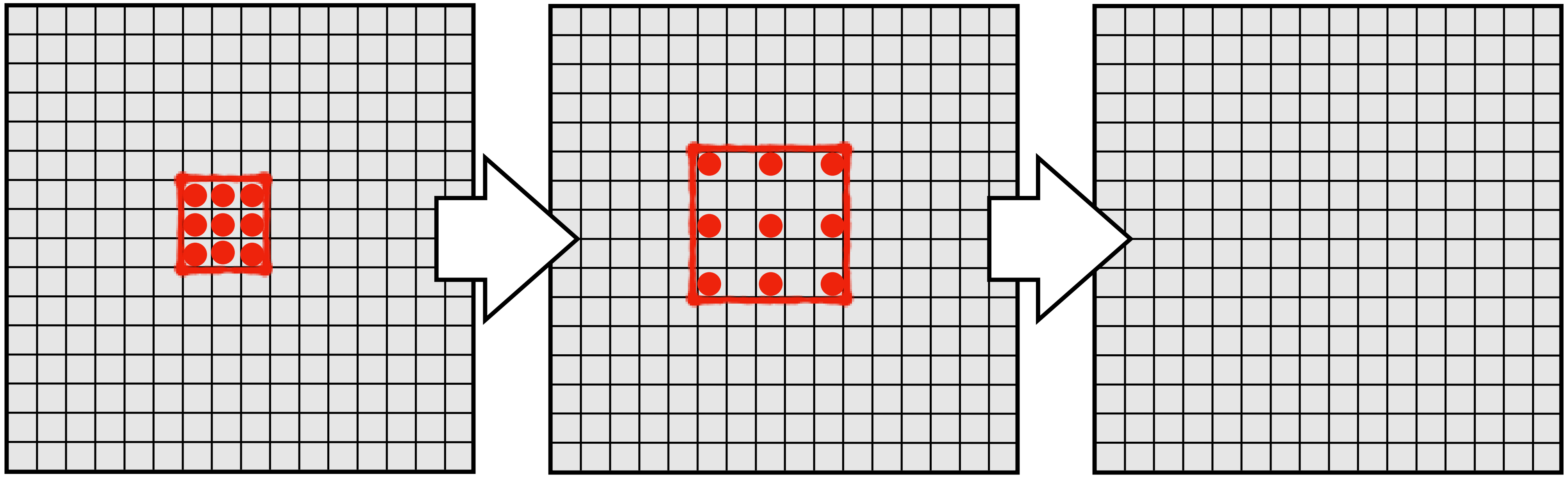


Increase dilation

[Yu & Koltun, Multi-Scale Context Aggregation by Dilated Convolutions, ICLR 2016]

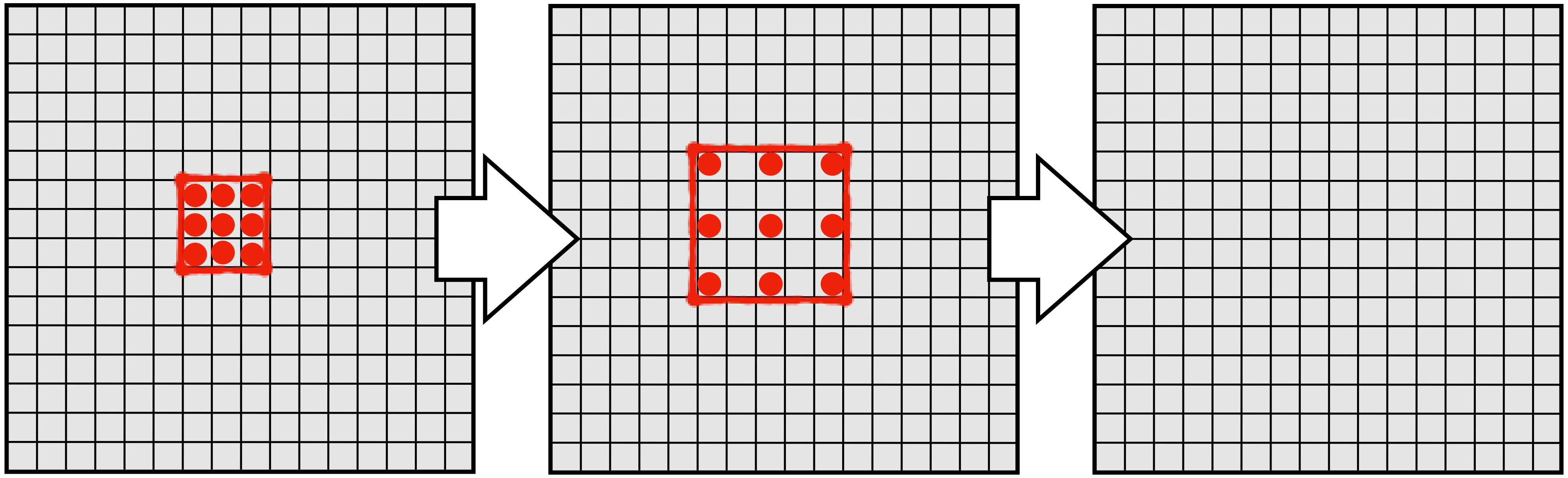
Receptive Field of Convolutions

Standard convolution:

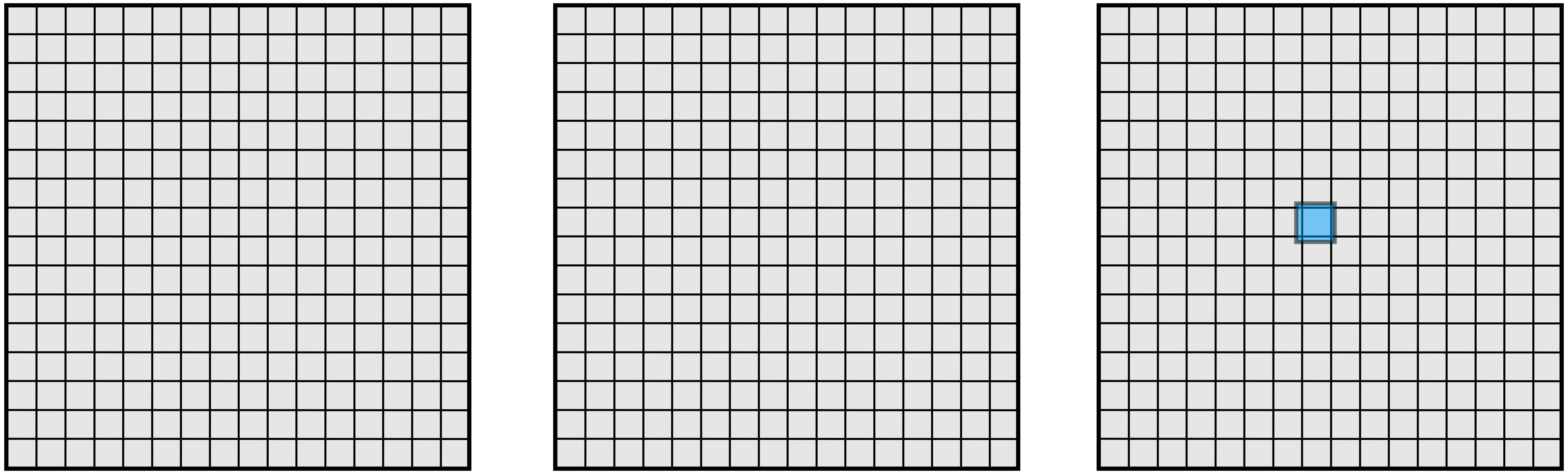


Receptive Field of Convolutions

Standard convolution:

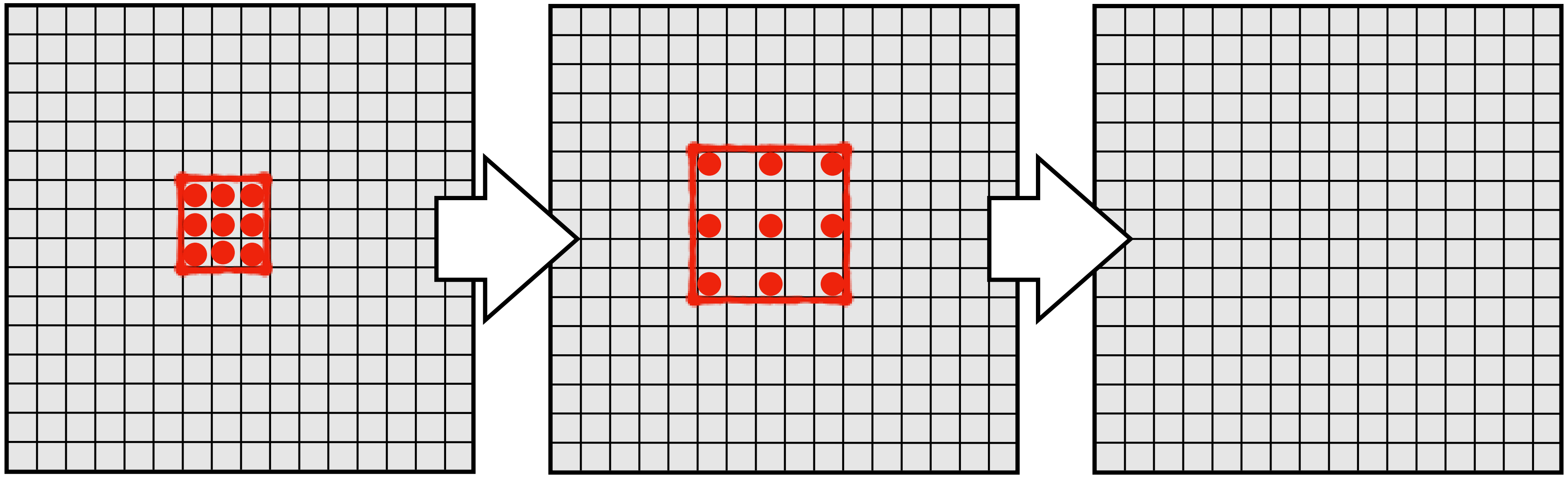


Receptive field:

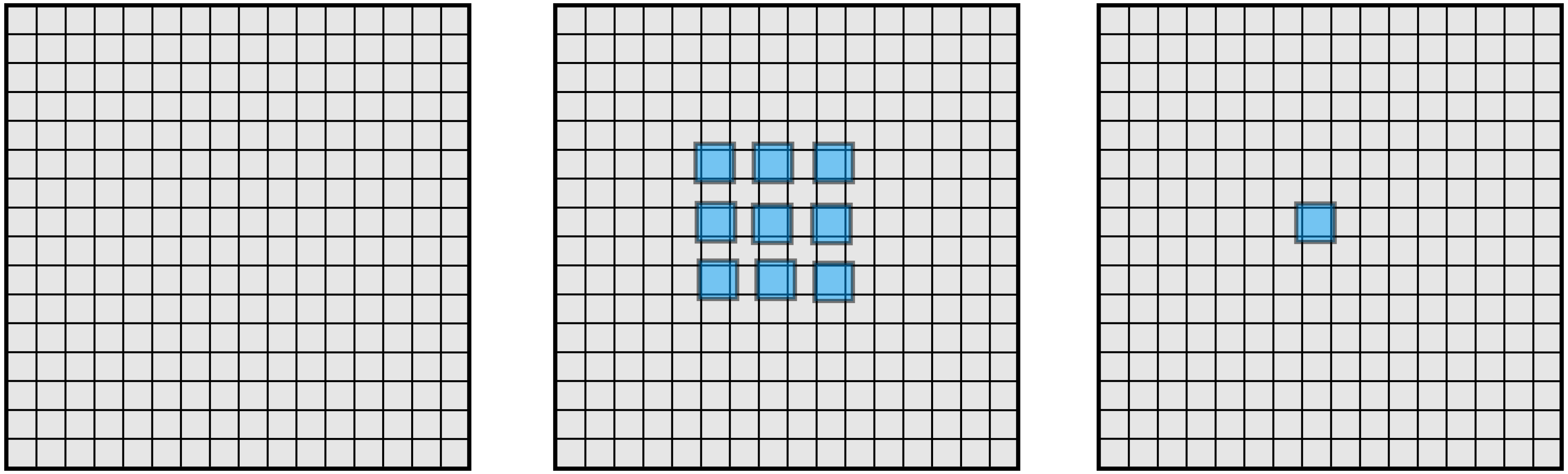


Receptive Field of Convolutions

Standard convolution:

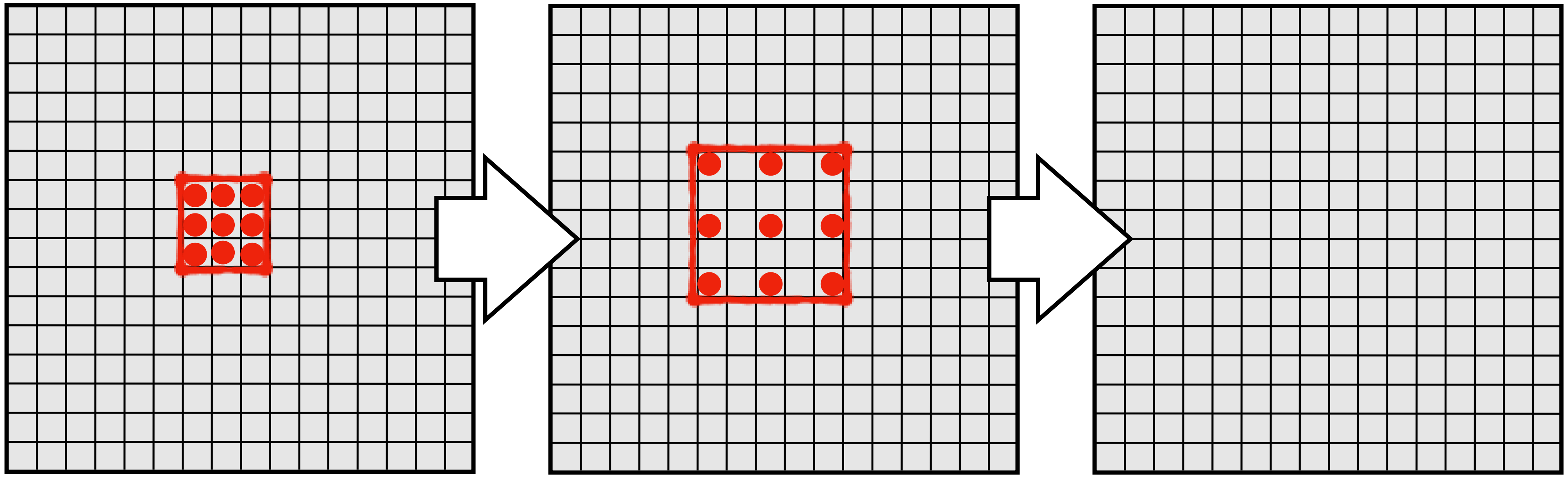


Receptive field:

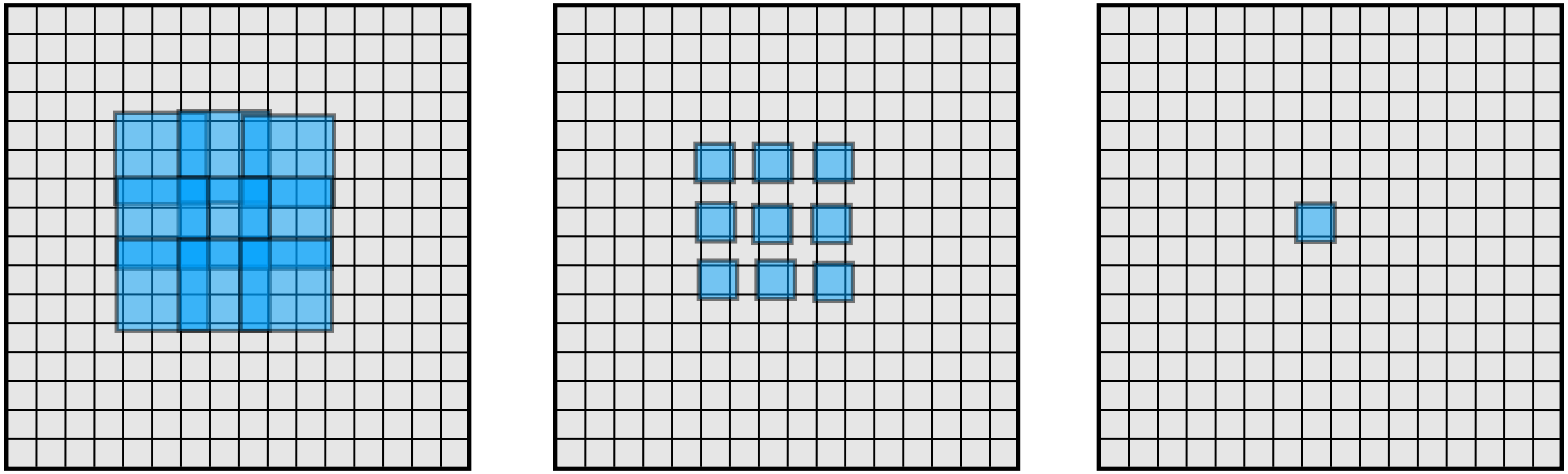


Receptive Field of Convolutions

Standard convolution:

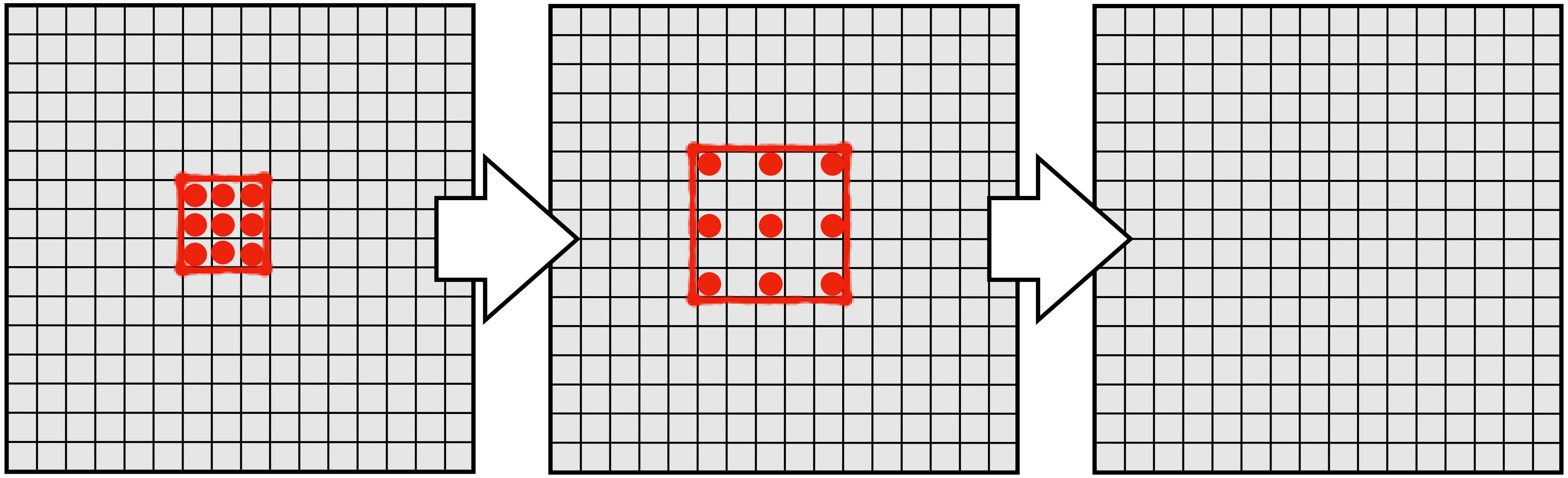


Receptive field:

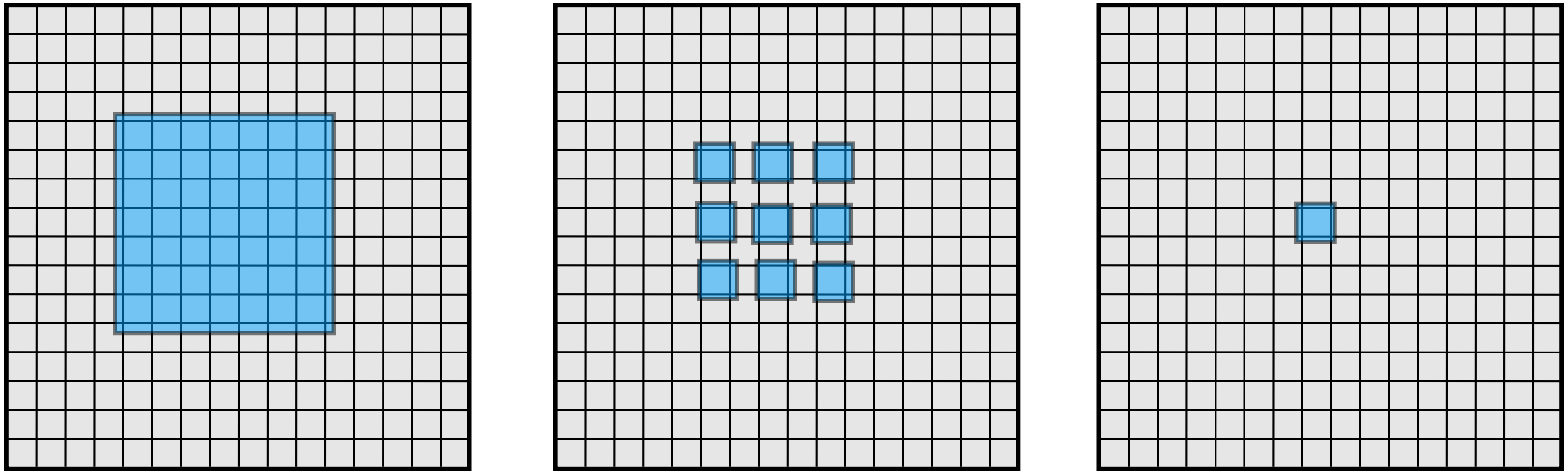


Receptive Field of Convolutions

Standard convolution:



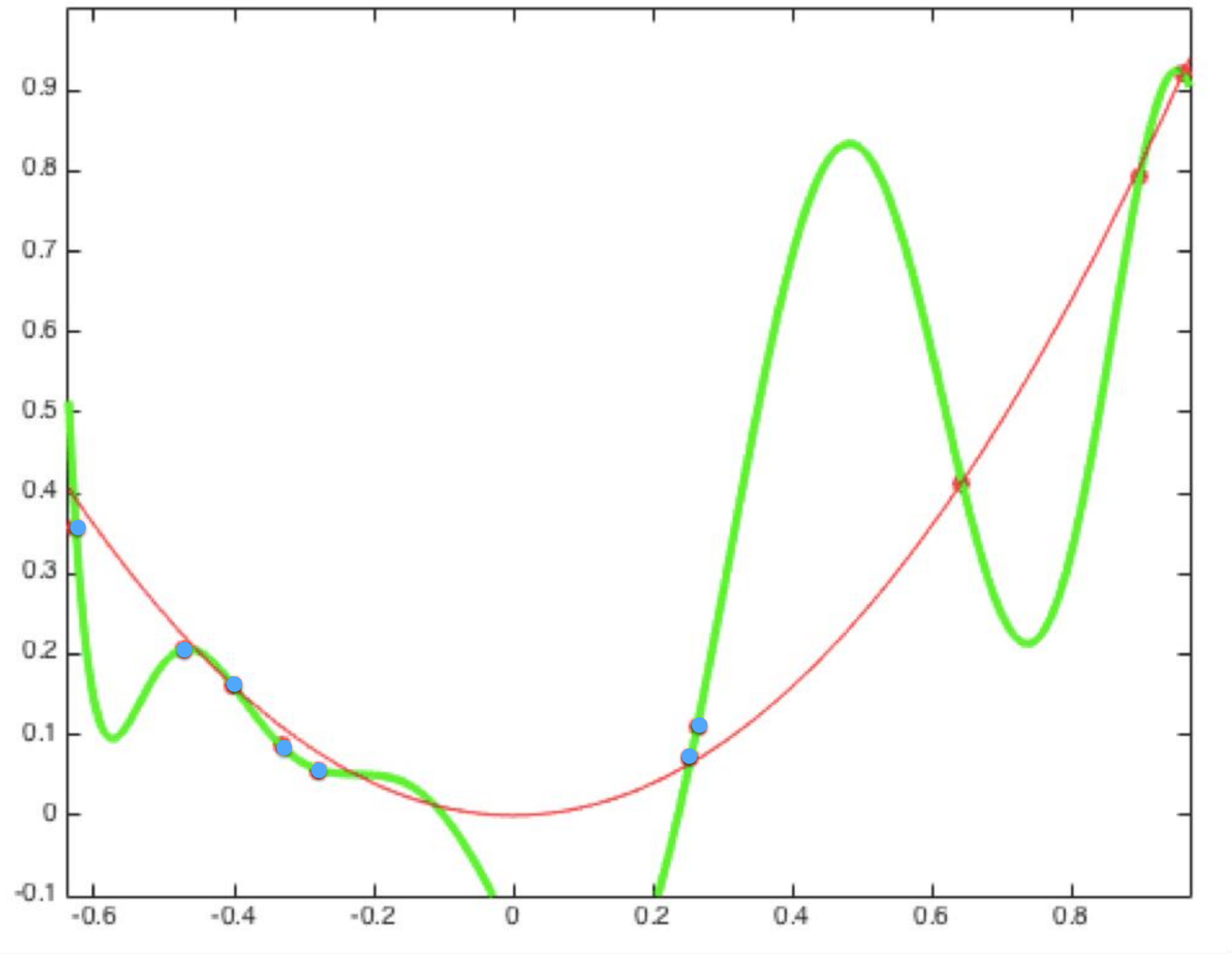
Receptive field:



Overfitting, Part II

Overfitting

10 data points - Fitting 10th degree polynomial



$$y = x^2$$

Overfitting

VGG-16 architecture

[Simonyan & Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015]

Overfitting

VGG-16 architecture

Conv
64 3x3

Conv
64 3x3

max
pooling

conv 1

[Simonyan & Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015]

Overfitting

VGG-16 architecture

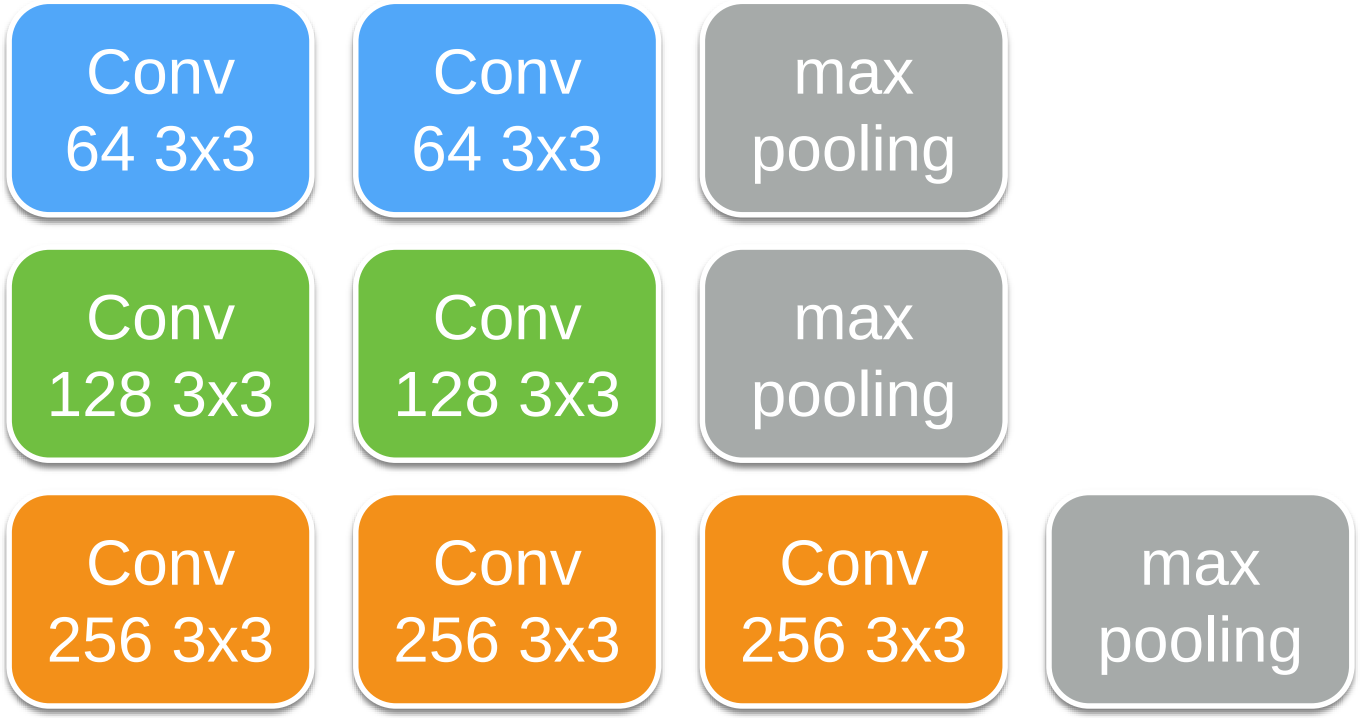
Conv 64 3x3 | Conv 64 3x3 | max pooling — **conv 1**

Conv 128 3x3 | Conv 128 3x3 | max pooling — **conv 2**

[Simonyan & Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015]

Overfitting

VGG-16 architecture



Conv 64 3x3 | Conv 64 3x3 | max pooling — **conv 1**

Conv 128 3x3 | Conv 128 3x3 | max pooling — **conv 2**

Conv 256 3x3 | Conv 256 3x3 | Conv 256 3x3 | max pooling — **conv 3**

[Simonyan & Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015]

Overfitting

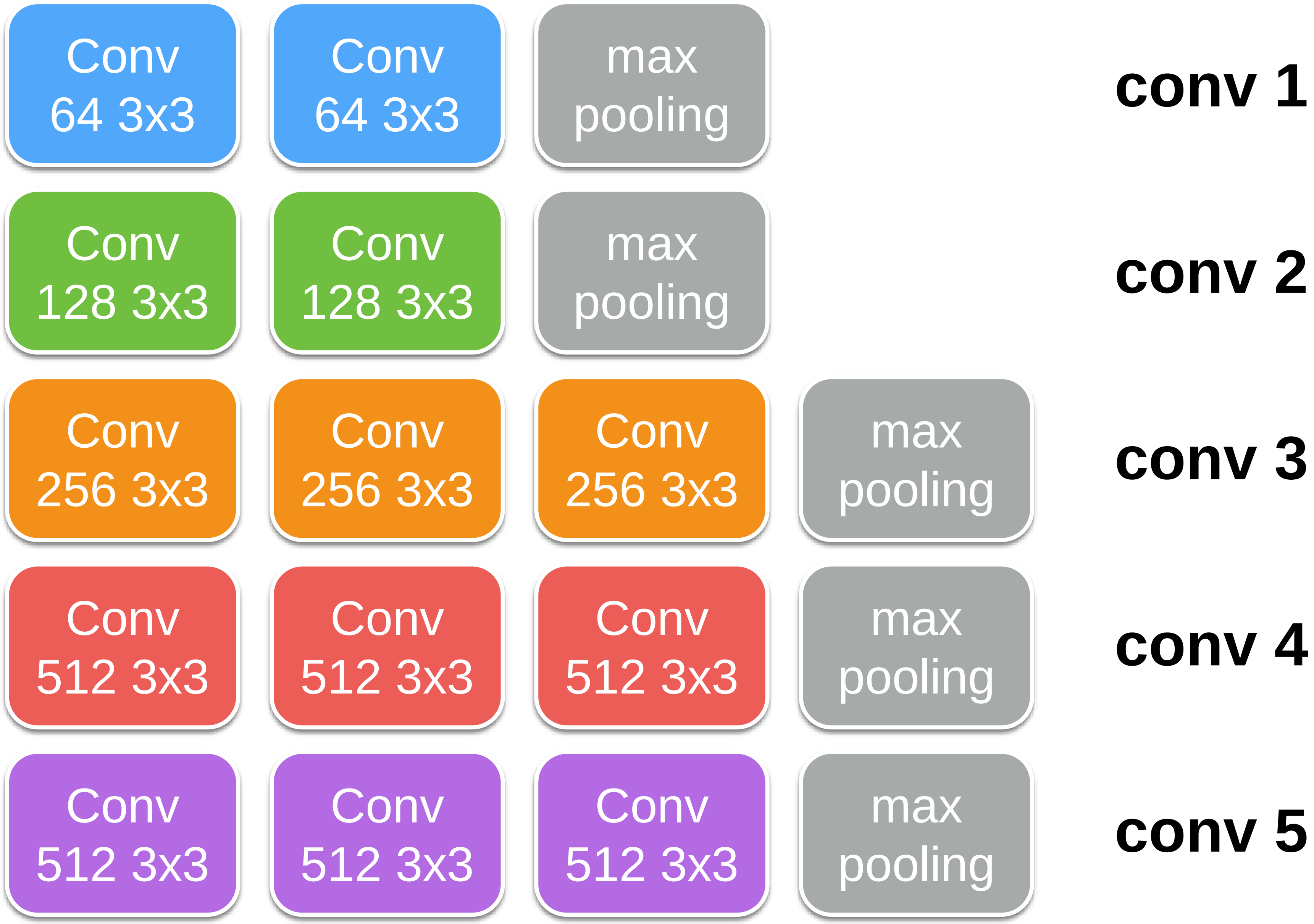
VGG-16 architecture

Conv 64 3x3	Conv 64 3x3	max pooling		**conv 1**
Conv 128 3x3	Conv 128 3x3	max pooling		**conv 2**
Conv 256 3x3	Conv 256 3x3	Conv 256 3x3	max pooling	**conv 3**
Conv 512 3x3	Conv 512 3x3	Conv 512 3x3	max pooling	**conv 4**

[Simonyan & Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015]

Overfitting

VGG-16 architecture



Conv 64 3x3	Conv 64 3x3	max pooling		**conv 1**
Conv 128 3x3	Conv 128 3x3	max pooling		**conv 2**
Conv 256 3x3	Conv 256 3x3	Conv 256 3x3	max pooling	**conv 3**
Conv 512 3x3	Conv 512 3x3	Conv 512 3x3	max pooling	**conv 4**
Conv 512 3x3	Conv 512 3x3	Conv 512 3x3	max pooling	**conv 5**

[Simonyan & Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015]

Overfitting

VGG-16 architecture

Conv 64 3x3	Conv 64 3x3	max pooling		**conv 1**
Conv 128 3x3	Conv 128 3x3	max pooling		**conv 2**
Conv 256 3x3	Conv 256 3x3	Conv 256 3x3	max pooling	**conv 3**
Conv 512 3x3	Conv 512 3x3	Conv 512 3x3	max pooling	**conv 4**
Conv 512 3x3	Conv 512 3x3	Conv 512 3x3	max pooling	**conv 5**
FC 4096	FC 4096	FC 1000	soft max	

[Simonyan & Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015]

Overfitting

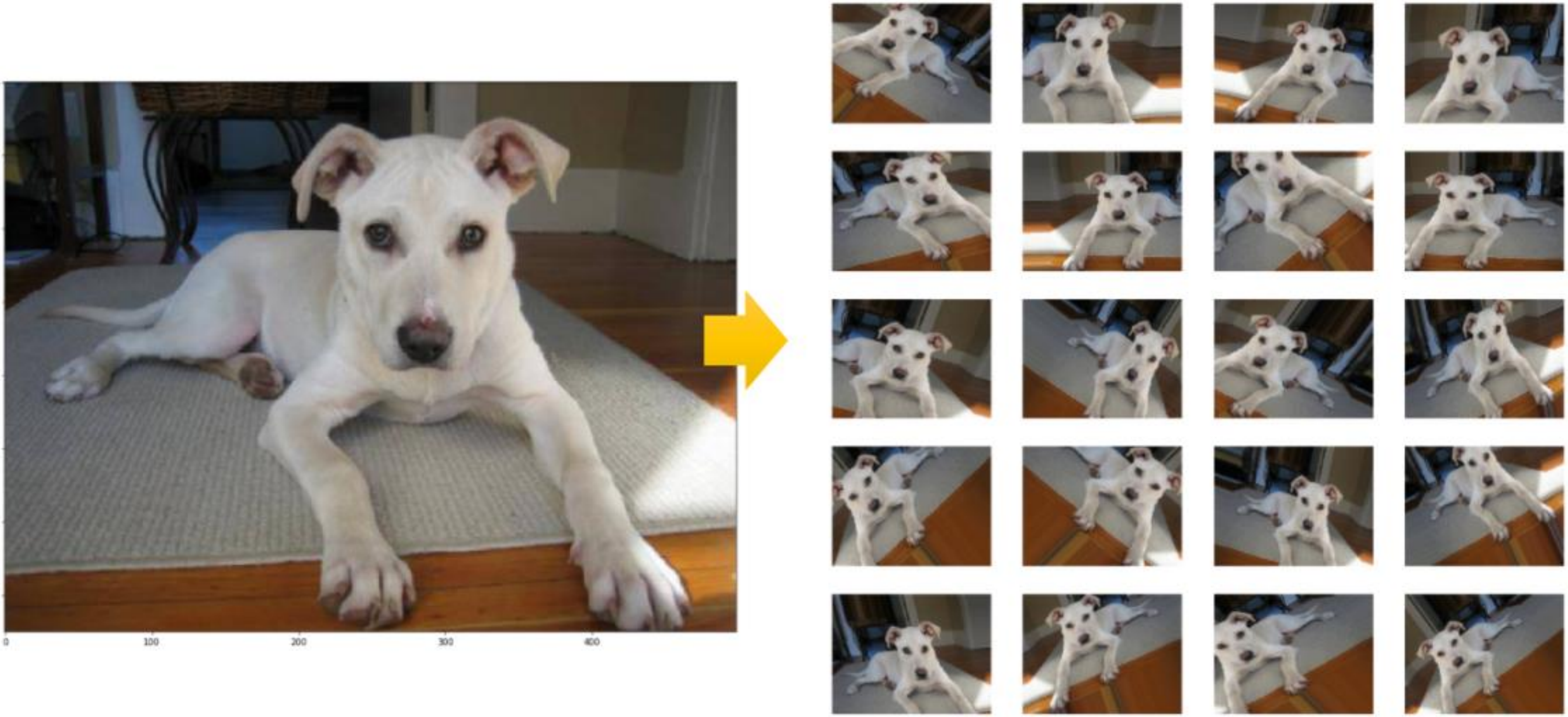
VGG-16 architecture

Conv 64 3x3	Conv 64 3x3	max pooling		**conv 1**
Conv 128 3x3	Conv 128 3x3	max pooling		**conv 2**
Conv 256 3x3	Conv 256 3x3	Conv 256 3x3	max pooling	**conv 3**
Conv 512 3x3	Conv 512 3x3	Conv 512 3x3	max pooling	**conv 4**
Conv 512 3x3	Conv 512 3x3	Conv 512 3x3	max pooling	**conv 5**
FC 4096	FC 4096	FC 1000	soft max	

138M parameters!

[Simonyan & Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015]

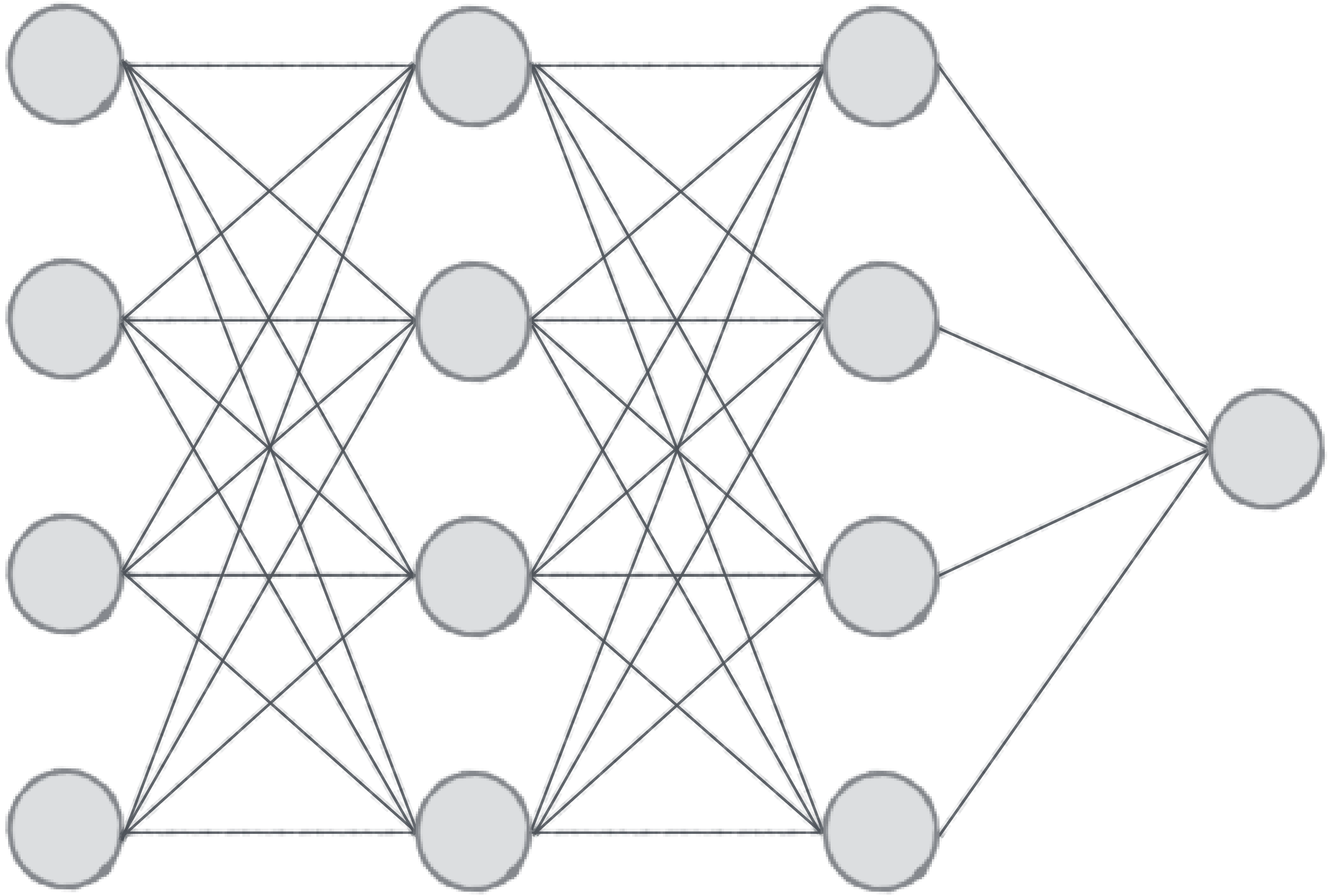
Data Augmentation



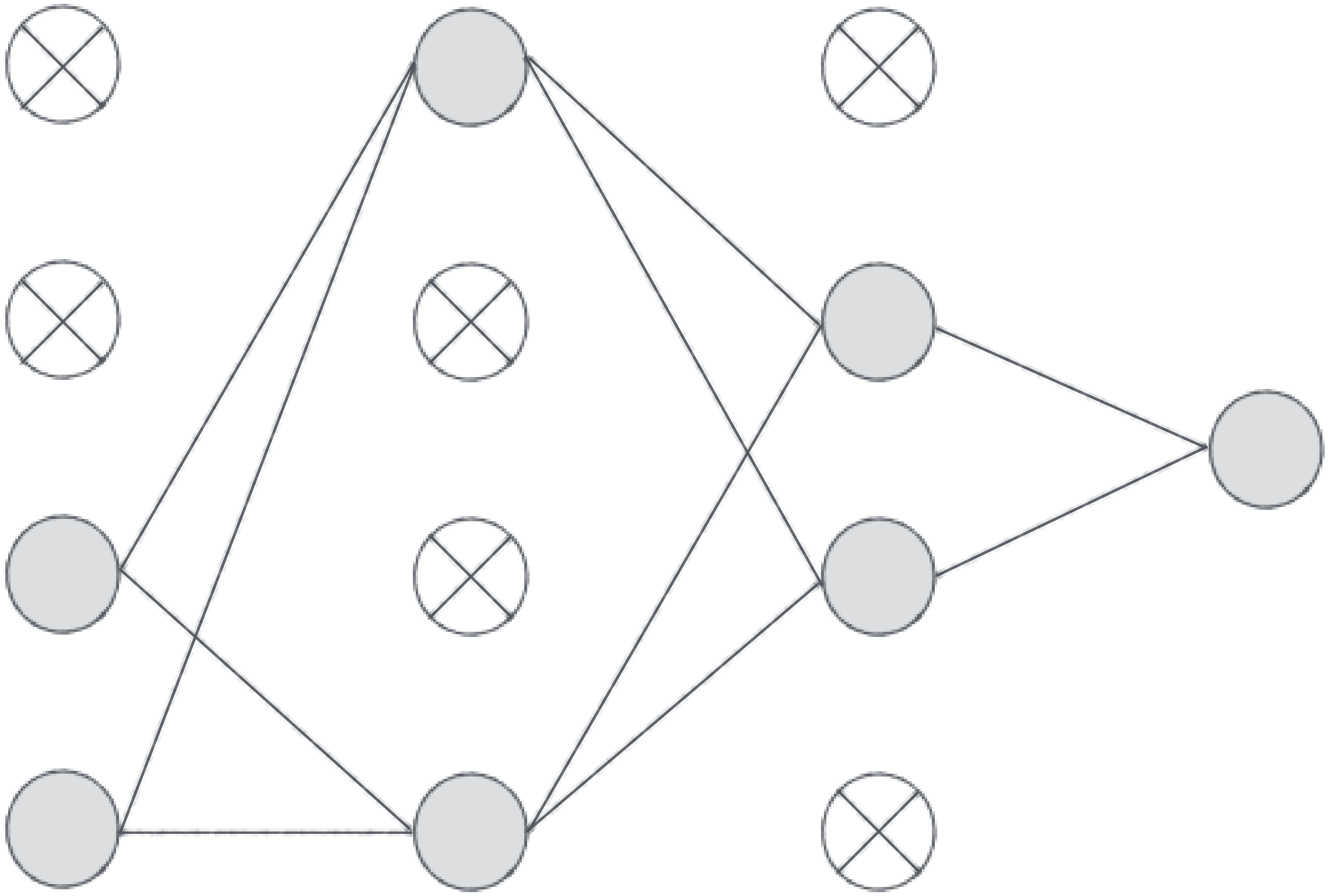
Regularization

$$L(w_1, \ldots, w_n) + c \sum_i w_i^2$$

Dropout



Dropout



Randomly disable neurons for each minibatch

Dropout

Transfer Learning

IMAGENET

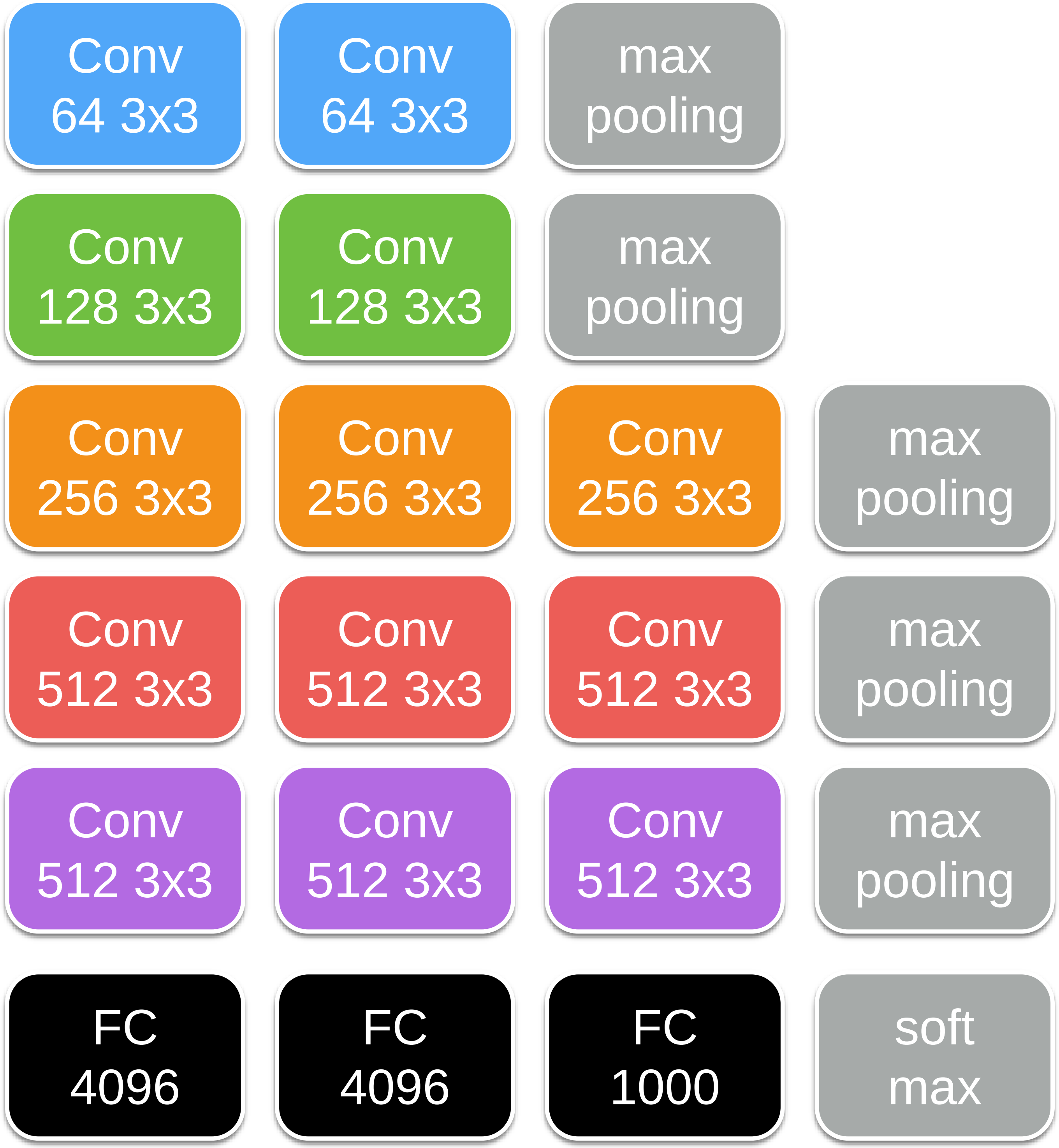
Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575.* [web]

3

14 million images of 1000 classes

Transfer Learning

VGG-16 architecture



Conv 64 3x3	Conv 64 3x3	max pooling	
Conv 128 3x3	Conv 128 3x3	max pooling	
Conv 256 3x3	Conv 256 3x3	Conv 256 3x3	max pooling
Conv 512 3x3	Conv 512 3x3	Conv 512 3x3	max pooling
Conv 512 3x3	Conv 512 3x3	Conv 512 3x3	max pooling
FC 4096	FC 4096	FC 1000	soft max

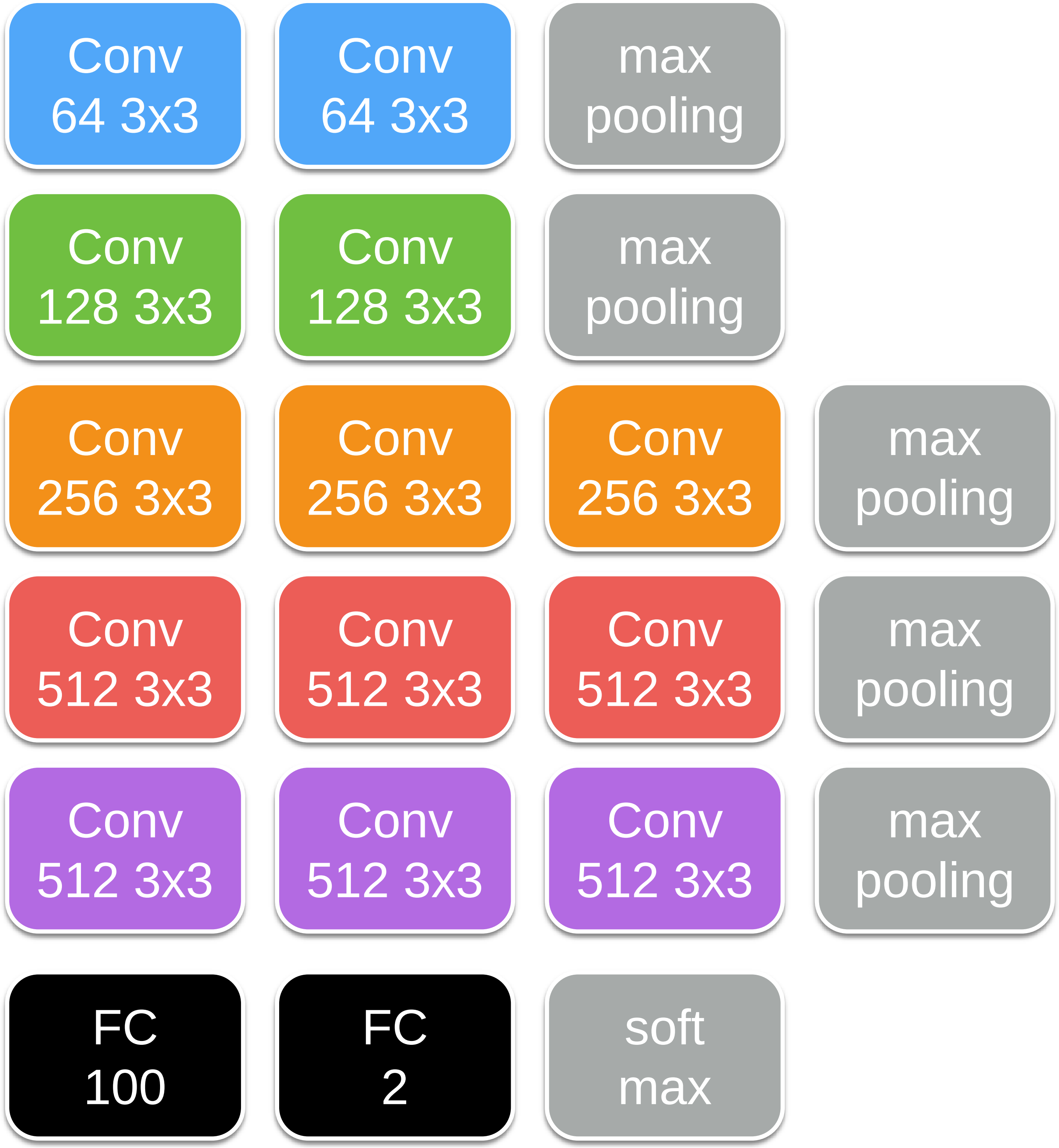
Transfer Learning

VGG-16 architecture

Conv 64 3x3	Conv 64 3x3	max pooling	
Conv 128 3x3	Conv 128 3x3	max pooling	
Conv 256 3x3	Conv 256 3x3	Conv 256 3x3	max pooling
Conv 512 3x3	Conv 512 3x3	Conv 512 3x3	max pooling
Conv 512 3x3	Conv 512 3x3	Conv 512 3x3	max pooling

Transfer Learning

VGG-16 architecture



Conv 64 3x3 | Conv 64 3x3 | max pooling

Conv 128 3x3 | Conv 128 3x3 | max pooling

Conv 256 3x3 | Conv 256 3x3 | Conv 256 3x3 | max pooling

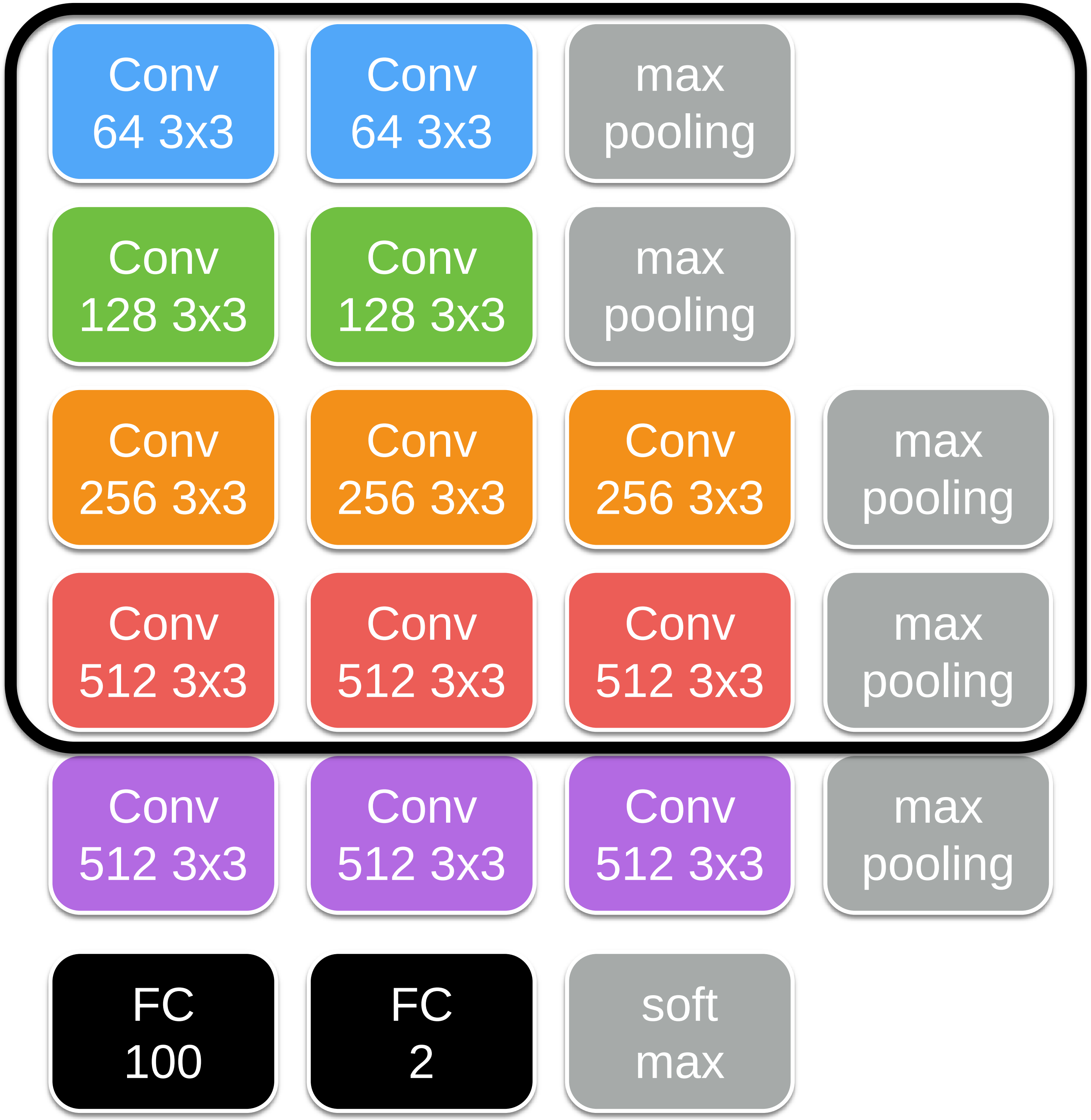
Conv 512 3x3 | Conv 512 3x3 | Conv 512 3x3 | max pooling

Conv 512 3x3 | Conv 512 3x3 | Conv 512 3x3 | max pooling

FC 100 | FC 2 | soft max

Transfer Learning

VGG-16 architecture



Conv 64 3x3 | Conv 64 3x3 | max pooling

Conv 128 3x3 | Conv 128 3x3 | max pooling

Conv 256 3x3 | Conv 256 3x3 | Conv 256 3x3 | max pooling

Conv 512 3x3 | Conv 512 3x3 | Conv 512 3x3 | max pooling

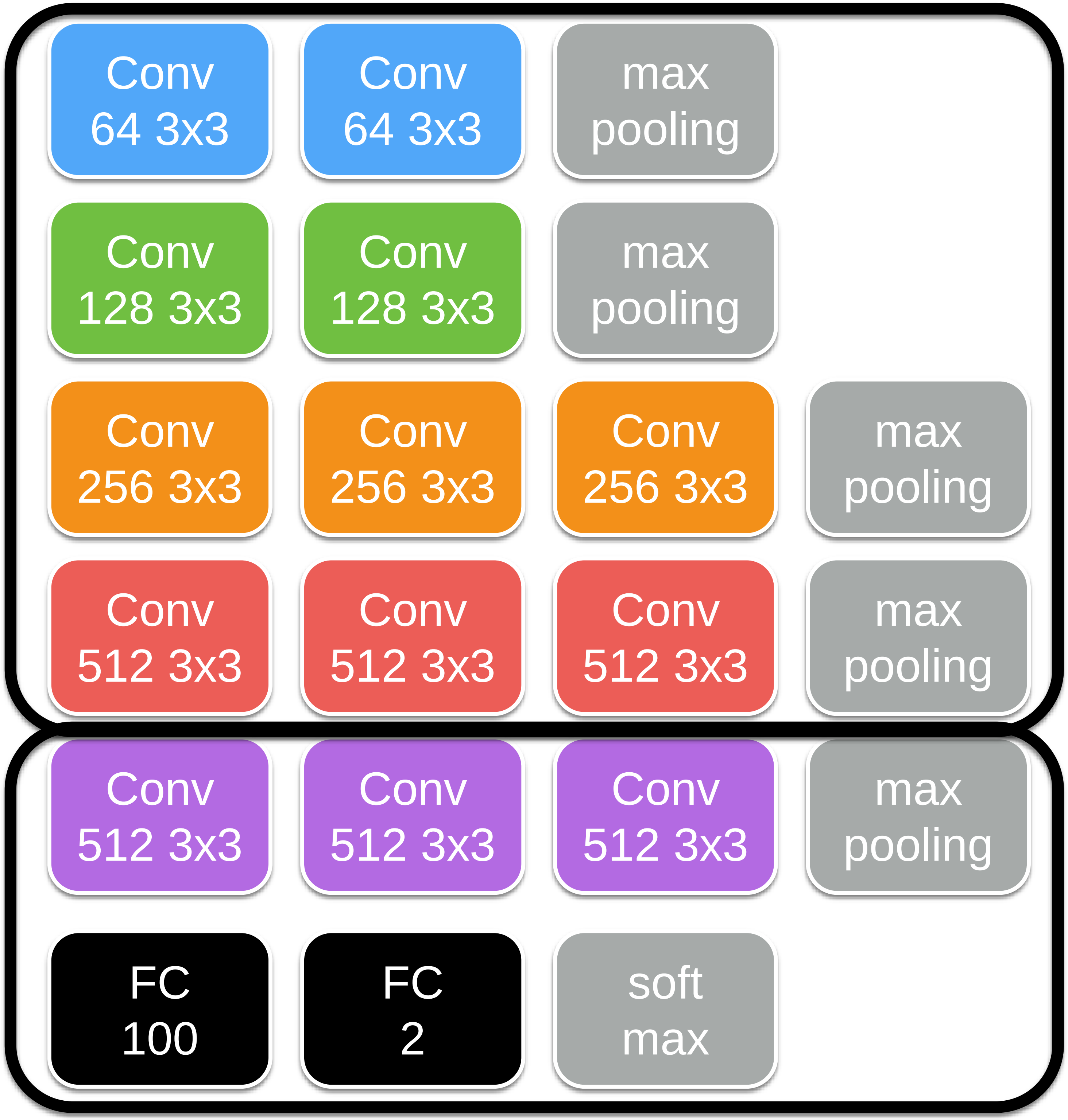
Conv 512 3x3 | Conv 512 3x3 | Conv 512 3x3 | max pooling

FC 100 | FC 2 | soft max

Fix weights

Transfer Learning

VGG-16 architecture



Conv 64 3x3

Conv 64 3x3

max pooling

Conv 128 3x3

Conv 128 3x3

max pooling

Conv 256 3x3

Conv 256 3x3

Conv 256 3x3

max pooling

Conv 512 3x3

Conv 512 3x3

Conv 512 3x3

max pooling

Conv 512 3x3

Conv 512 3x3

Conv 512 3x3

max pooling

FC 100

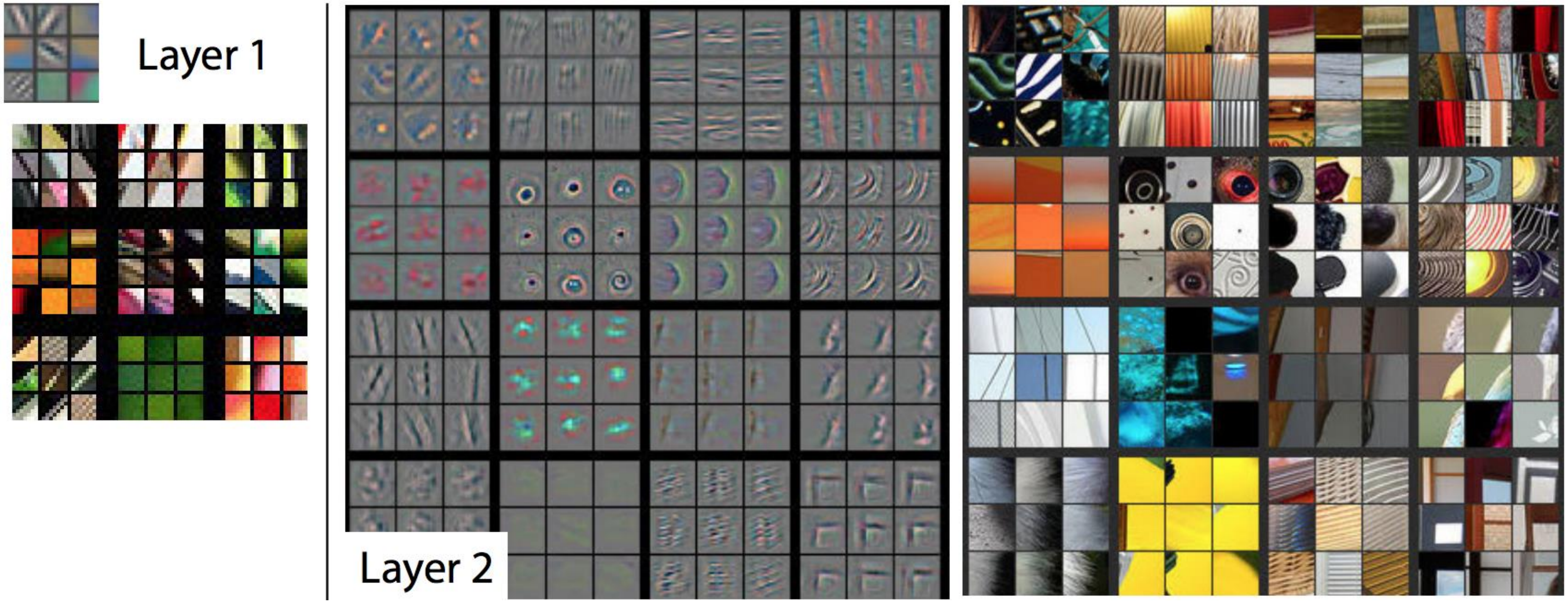
FC 2

soft max

Fix weights

Train on limited data

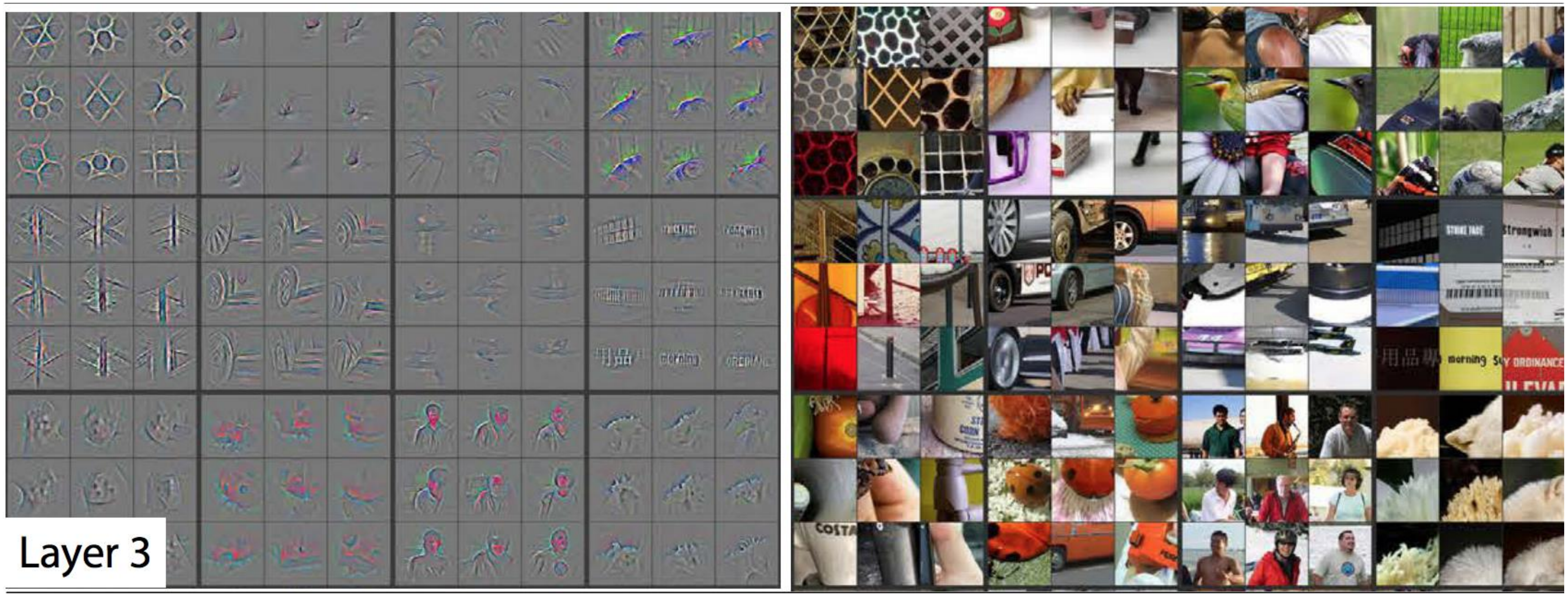
Transfer Learning



Low-level features: corners, edges, …

[Zeiler & Fergus, Visualizing and Understanding Convolutional Networks, ECCV 2014]

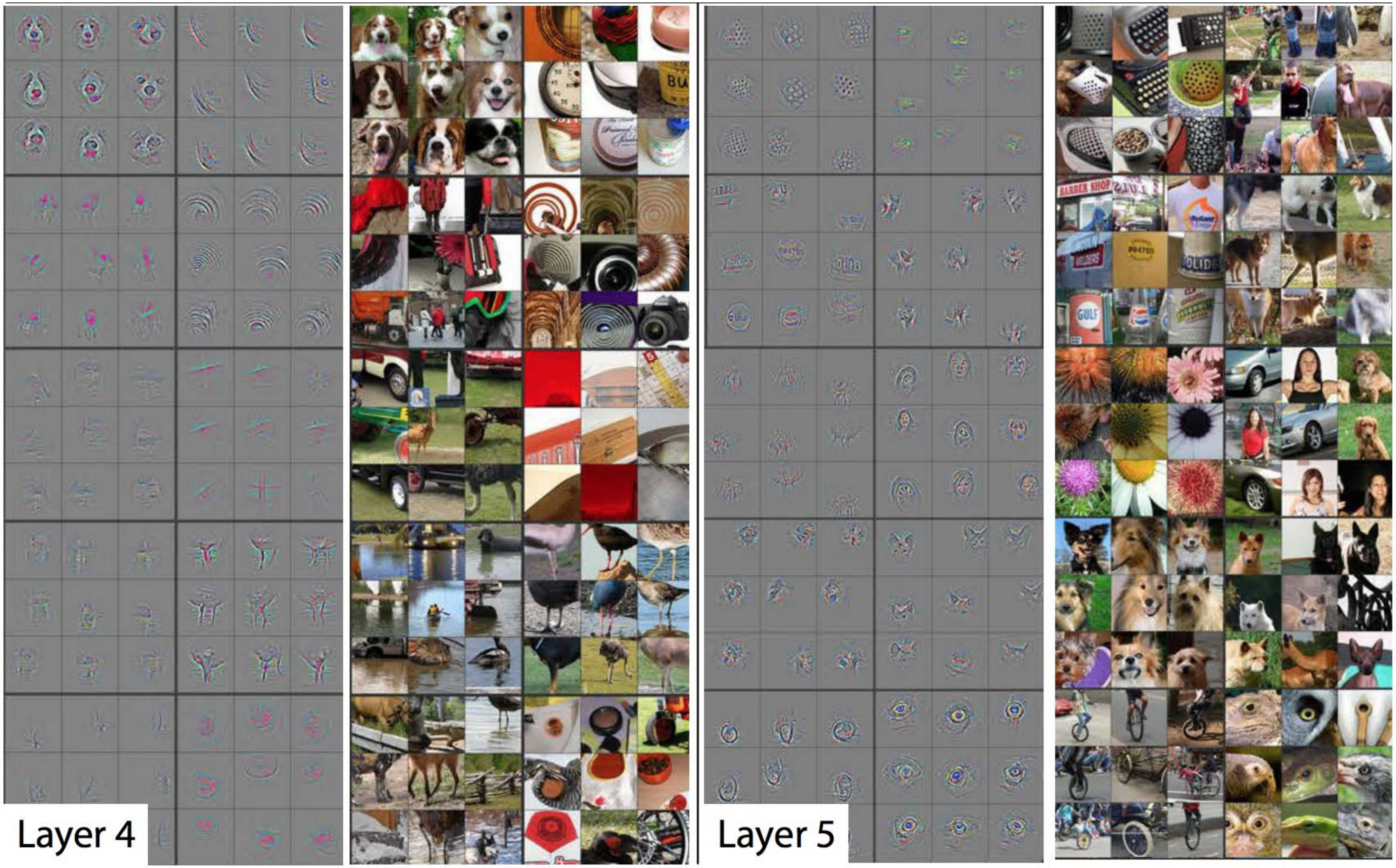
Transfer Learning



Mid-level features

[Zeiler & Fergus, Visualizing and Understanding Convolutional Networks, ECCV 2014]

Transfer Learning



Higher-level features: object parts & objects

[Zeiler & Fergus, Visualizing and Understanding Convolutional Networks, ECCV 2014]

Lessons Learned

- Main lessons from this lecture
  - Convolutional neural networks
  - Layers: Fully connected, convolutional, activation functions, max pooling, strides & dilated filters
  - Fully convolutional networks

  - Handling overfitting

- Next lecture: Robust Model Fitting

Next Lecture

Jan. 20	Introduction, Linear classifiers and filtering	
Jan. 23	Filtering, gradients, scale	Lab 1
Jan. 27	Local features	
Jan. 30	Learning a classifier	Lab 2
Feb. 3	Convolutional neural networks	
Feb. 6	More convolutional neural networks	
Feb. 10	**Robust model fitting and RANSAC**	Lab 3
Feb. 13	Image registration	
Feb. 17	Camera Geometry	Lab 4
Feb. 20	More camera geometry	
Feb. 24	Generative neural networks	
Feb. 27	Generative neural networks	
Mar. 2	TBA	
Mar. 9	TBA	