Convolutional neural networks

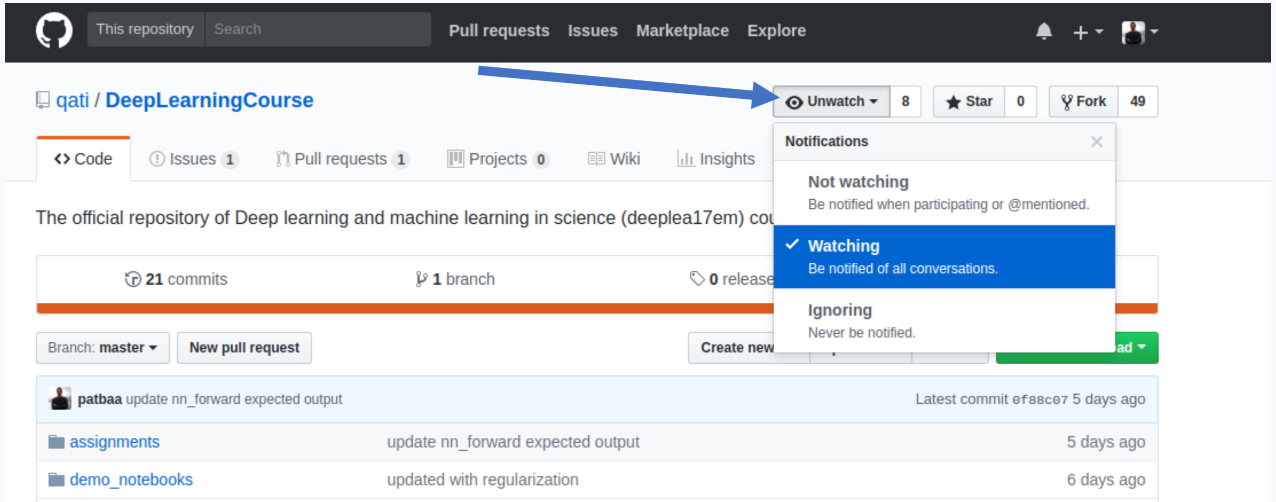
Bálint Ármin Pataki

Technical info

- Homework02 (deadline: today):
  - nn_train → update **b** too
- kaggle ELTE_phys_photoz (deadline: 2018.03.20.):
  - 7 people submitted
  - 6 outperformed baseline

Technical info

- Homework02 (deadline: today):
  - nn_train → update **b** too
- kaggle ELTE_phys_photoz (deadline: 2018.03.20.):
  - 7 people submitted
  - 6 outperformed baseline
- github watch to get e-mail notifications



- Random numbers (they are not random)

```
import numpy as np
np.random.seed(0)
np.random.randn(3)
```

```
array([ 1.76405235,  0.40015721,  0.97873798])
```

- Random numbers (they are not random)
    - rand vs randn (later will be explicitly mentioned)

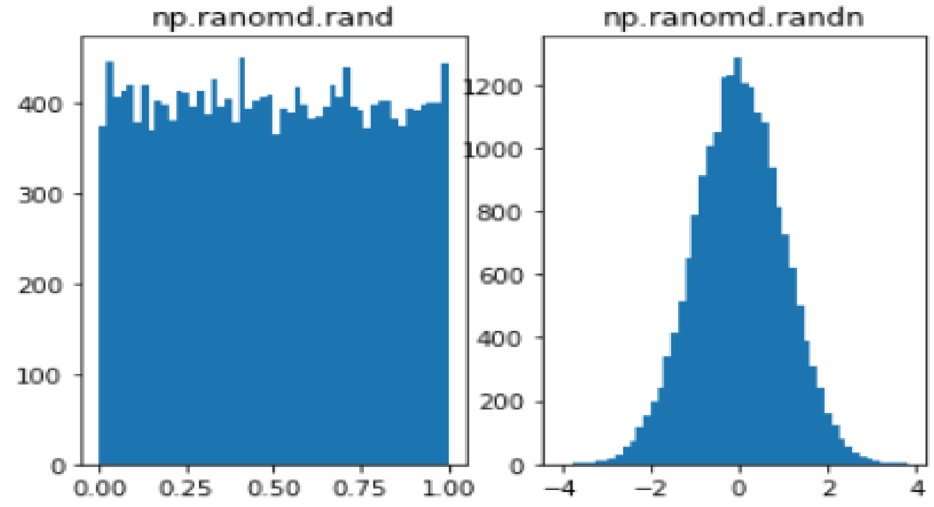
```
import numpy as np
np.random.seed(0)
np.random.randn(3)
```

```
array([ 1.76405235,  0.40015721,  0.97873798])
```

```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline

plt.subplot(1, 2, 1)
plt.hist(np.random.rand(20000), bins=50)
plt.title('np.ranomd.rand')

plt.subplot(1, 2, 2)
plt.hist(np.random.randn(20000), bins=50)
plt.title('np.ranomd.randn')
plt.show()
```



- Random numbers (they are not random)
  - rand vs randn (later will be explicitly mentioned)

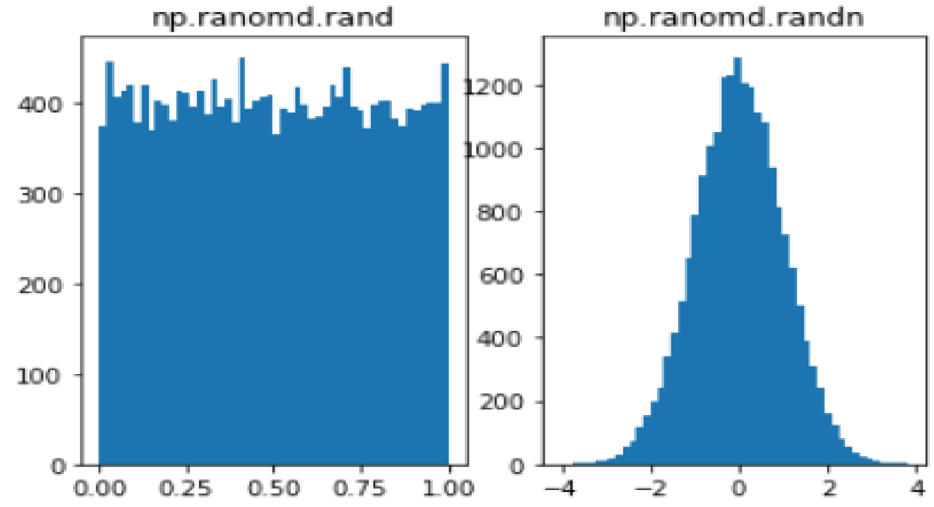
```
import numpy as np
np.random.seed(0)
np.random.randn(3)
```

```
array([ 1.76405235,  0.40015721,  0.97873798])
```

```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline

plt.subplot(1, 2, 1)
plt.hist(np.random.rand(20000), bins=50)
plt.title('np.ranomd.rand')

plt.subplot(1, 2, 2)
plt.hist(np.random.randn(20000), bins=50)
plt.title('np.ranomd.randn')
plt.show()
```



- Grading
  - Tested for different seeds
  - Tested for different inputs
  - You can correct your errors (resubmitting is allowed)

Homework - softmax

categories

samples

$$
\begin{matrix} Z_{00}, & Z_{01}, & Z_{02} \\ Z_{10}, & Z_{11}, & Z_{12} \end{matrix}
$$

Homework - softmax

categories

samples

$$Z_{00}\,,\quad Z_{01}\,,\quad Z_{02}$$
$$Z_{10}\,,\quad Z_{11}\,,\quad Z_{12}$$

categories

samples

$$\frac{e^{Z_{00}}}{e^{Z_{00}}+e^{Z_{01}}+e^{Z_{02}}},\ \frac{e^{Z_{01}}}{e^{Z_{00}}+e^{Z_{01}}+e^{Z_{02}}},\ \frac{e^{Z_{02}}}{e^{Z_{00}}+e^{Z_{01}}+e^{Z_{02}}}$$
$$\frac{e^{Z_{10}}}{e^{Z_{10}}+e^{Z_{11}}+e^{Z_{12}}},\ \frac{e^{Z_{11}}}{e^{Z_{10}}+e^{Z_{11}}+e^{Z_{12}}},\ \frac{e^{Z_{12}}}{e^{Z_{10}}+e^{Z_{11}}+e^{Z_{12}}}$$

```
np.exp(Z)/(np.exp(Z).sum(axis=1, keepdims=True)
np.exp(Z)/(np.exp(Z).sum(axis=1).reshape(Z.shape[0], 1)
```

categories

samples

$$\begin{matrix} Z_{00}, & Z_{01}, & Z_{02} \\ Z_{10}, & Z_{11}, & Z_{12} \end{matrix}$$

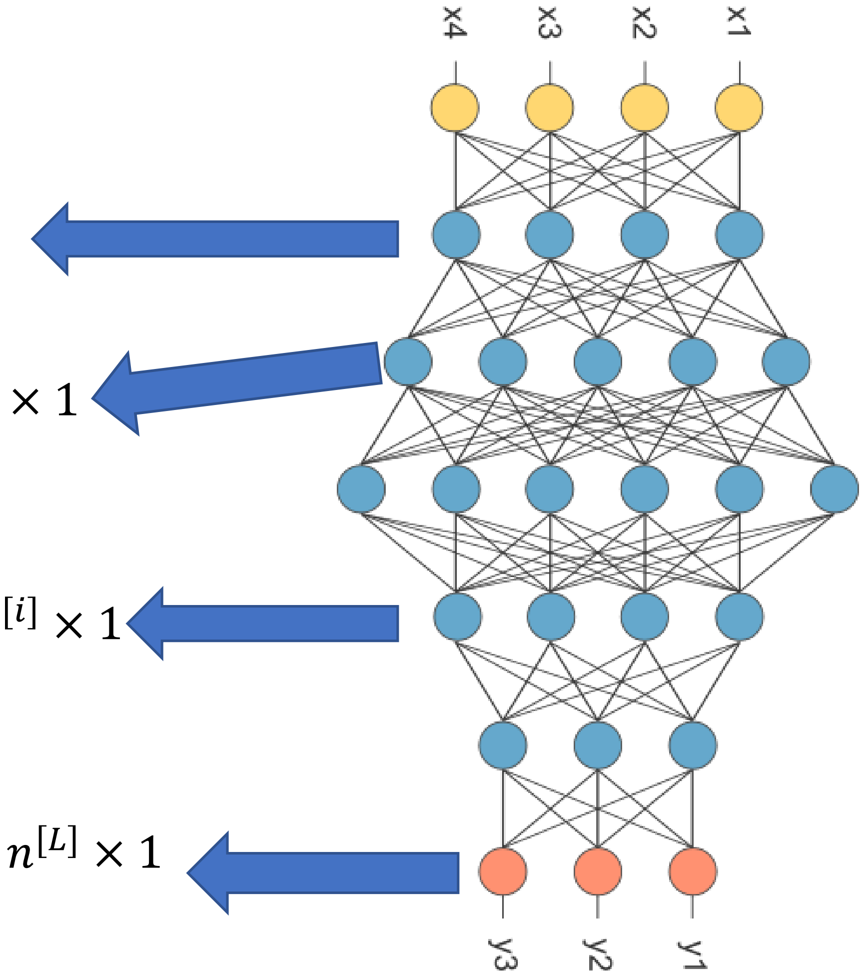
categories

samples

$$\begin{matrix} \frac{e^{Z_{00}}}{e^{Z_{00}}+e^{Z_{01}}+e^{Z_{02}}}, & \frac{e^{Z_{01}}}{e^{Z_{00}}+e^{Z_{01}}+e^{Z_{02}}}, & \frac{e^{Z_{02}}}{e^{Z_{00}}+e^{Z_{01}}+e^{Z_{02}}} \\ \frac{e^{Z_{10}}}{e^{Z_{10}}+e^{Z_{11}}+e^{Z_{12}}}, & \frac{e^{Z_{11}}}{e^{Z_{10}}+e^{Z_{11}}+e^{Z_{12}}}, & \frac{e^{Z_{12}}}{e^{Z_{10}}+e^{Z_{11}}+e^{Z_{12}}} \end{matrix}$$

L-layer neural network: reminder

$x \in \mathbb{R}^N, y \in \mathbb{R}^K$, neural network: $\mathbb{R}^N \rightarrow \mathbb{R}^K$



$$z^{[1]} = W^{[1]}x + b^{[1]}, \qquad W: n^{[1]} \times N, \qquad b: n^{[1]} \times 1$$
$$a^{[1]} = g\left(z^{[1]}\right)$$

$$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}, \qquad W: n^{[2]} \times n^{[1]}, \qquad b: n^{[2]} \times 1$$
$$a^{[2]} = g\left(z^{[2]}\right)$$

$\vdots$

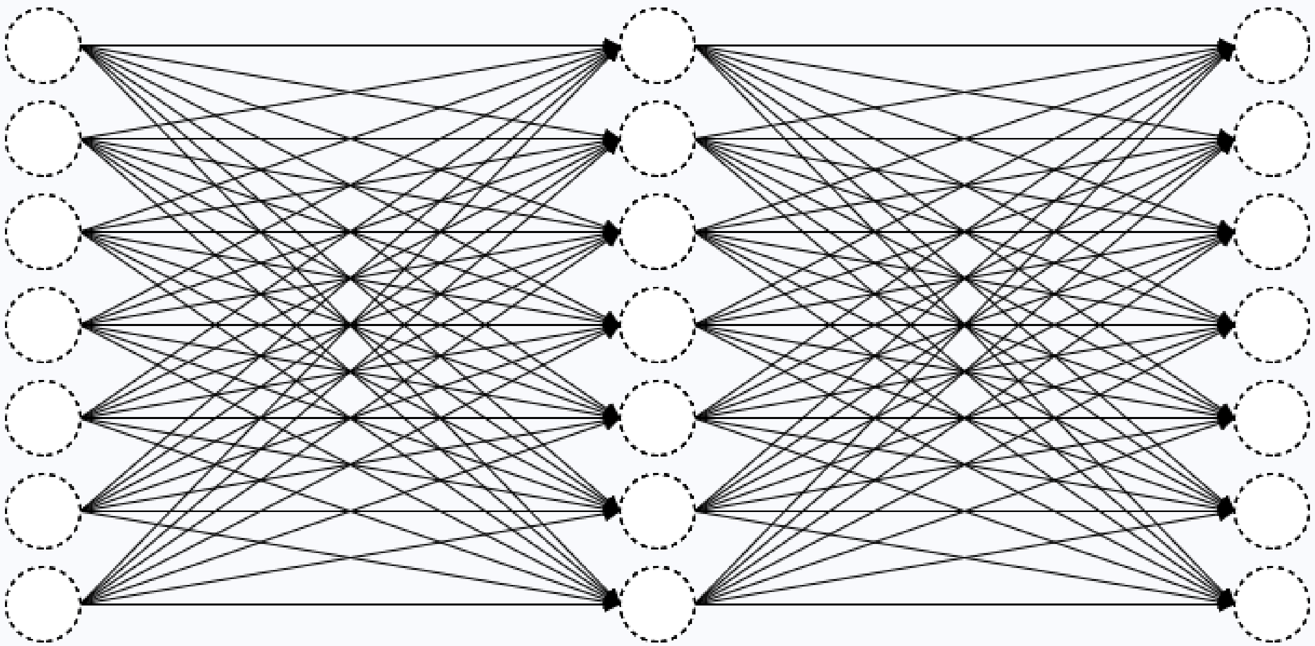
$$z^{[i]} = W^{[i]}a^{[i-1]} + b^{[i]}, \qquad W: n^{[i]} \times n^{[i-1]}, \qquad b: n^{[i]} \times 1$$
$$a^{[i]} = g\left(z^{[i]}\right)$$

$\vdots$

$$z^{[L]} = W^{[L]}a^{[L-1]} + b^{[L]}, \qquad W: n^{[L]} \times n^{[L-1]}, \qquad b: n^{[L]} \times 1$$
$$y = a^{[L]} = softmax\left(z^{[L]}\right)$$

Credit: OpenNN

Dense neural networks: problems for real world images 1.



- Exploding parameter number:
  - 200x200 pixel input → 40000 input
  - $40000^2 + 40000 \approx 1.6 \cdot 10^9$ parameters per layer
  - float32: 4 byte/number → 6.4 GB/layer
  - color images have 3 color channels (RGB) → 57.6 GB/layer

Question: How do you recognize the content of this picture?

Question: How do you recognize the content of this picture?

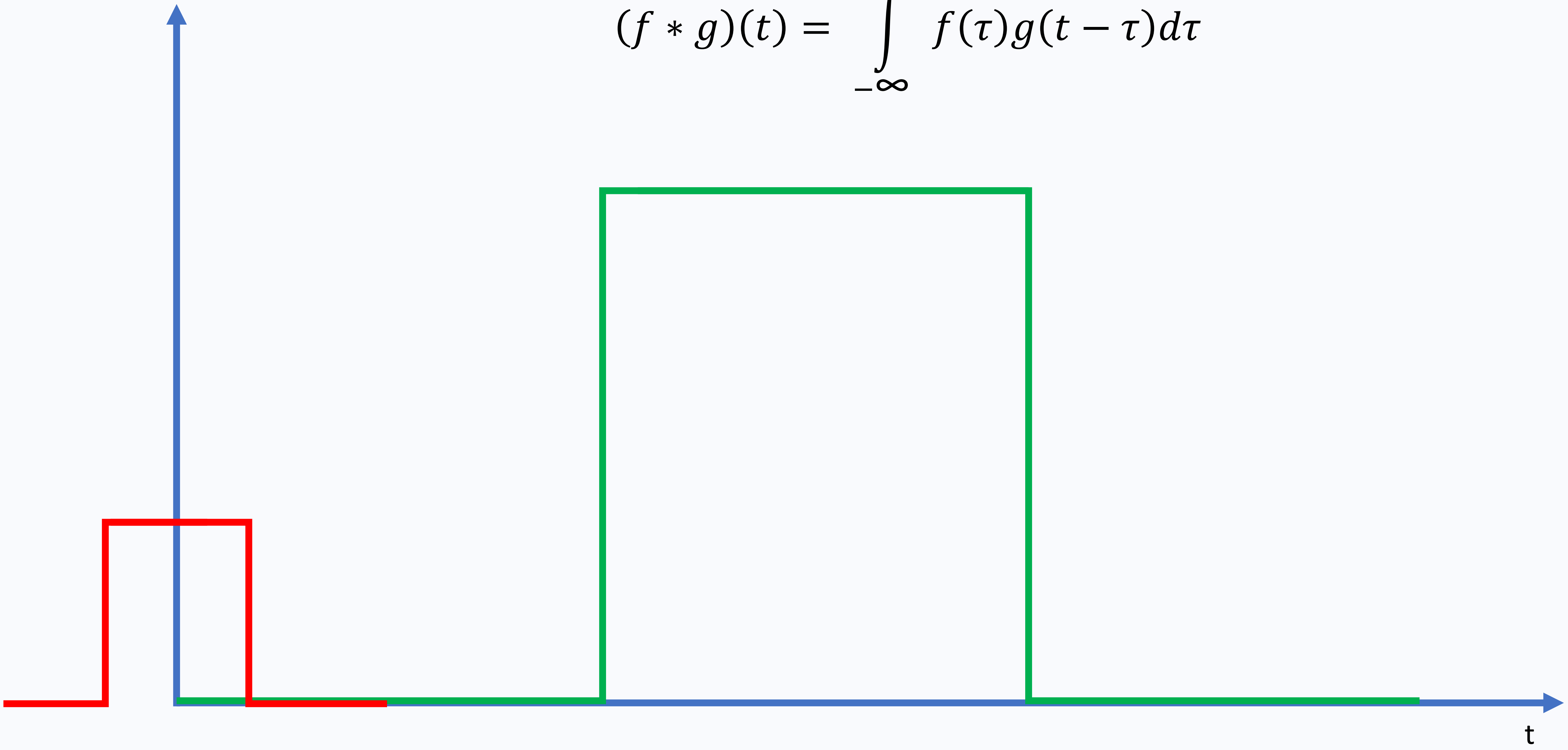
Feature locality (pixels are relevant only to their neighbours)

Question: How do you recognize the content of this picture?

Feature locality (pixels are relevant only to their neighbours)

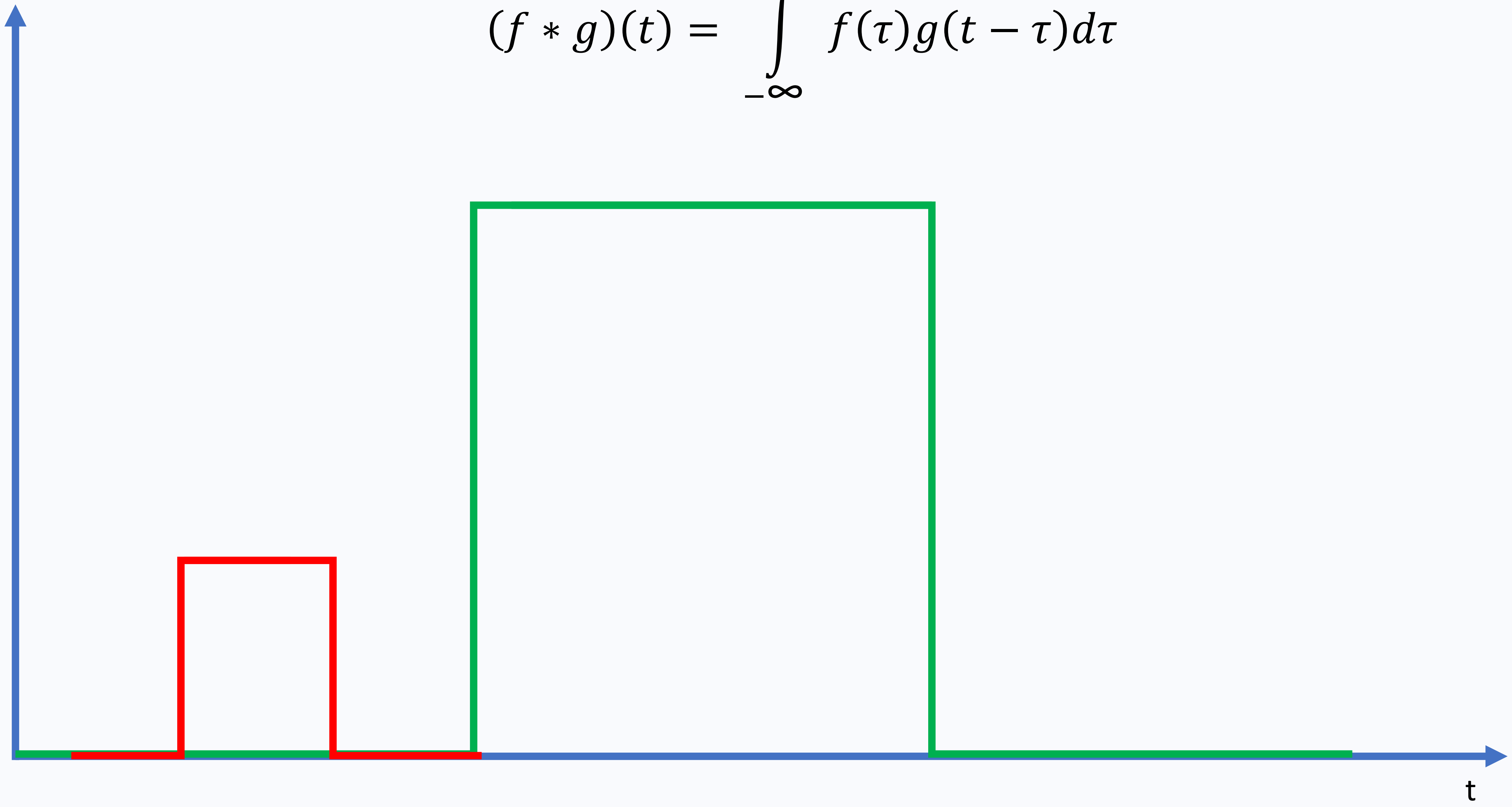
Goal: to have a „window detector" with a few parameters.

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



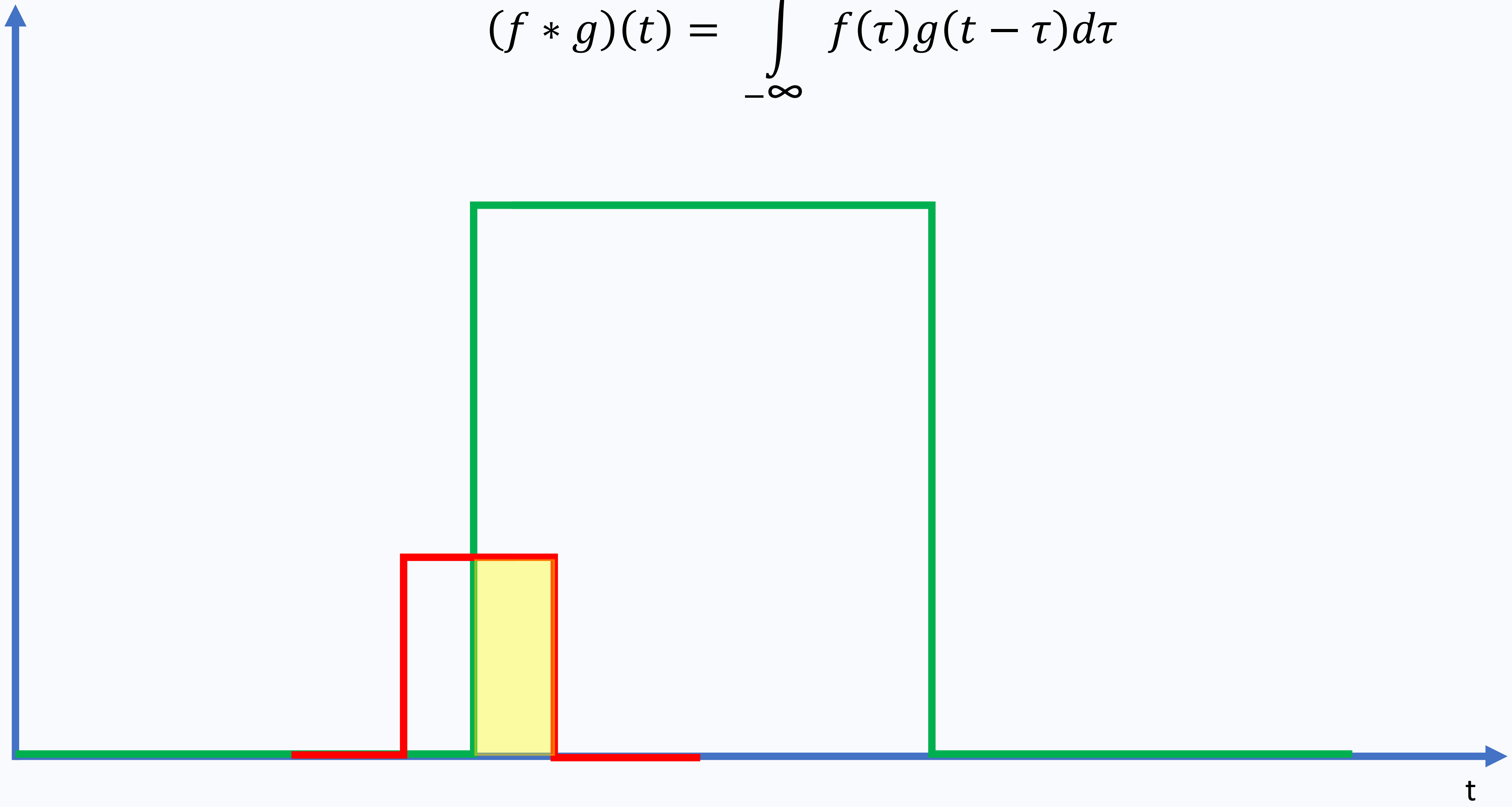
Convolution in 1D

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

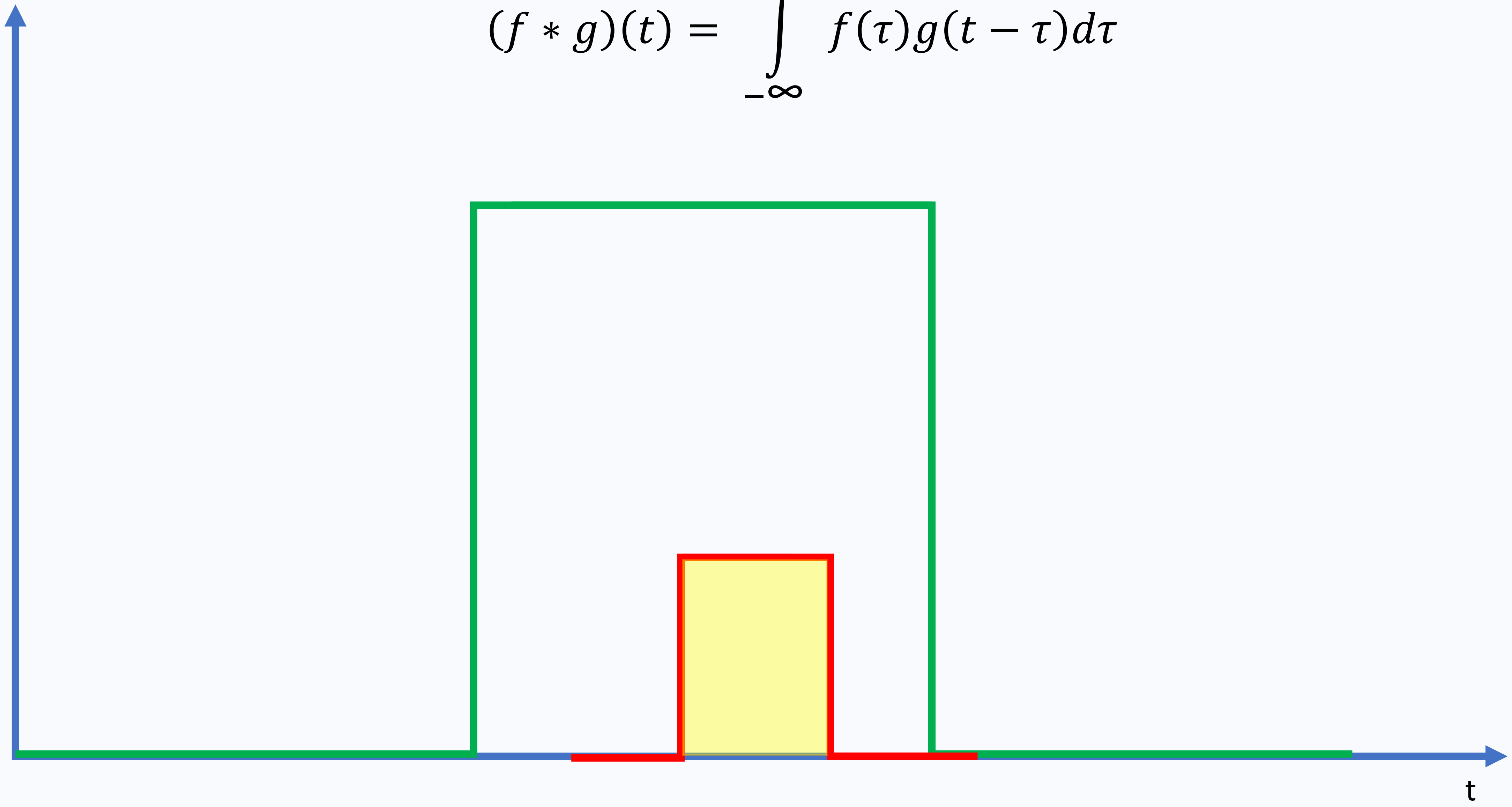


Convolution in 1D

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

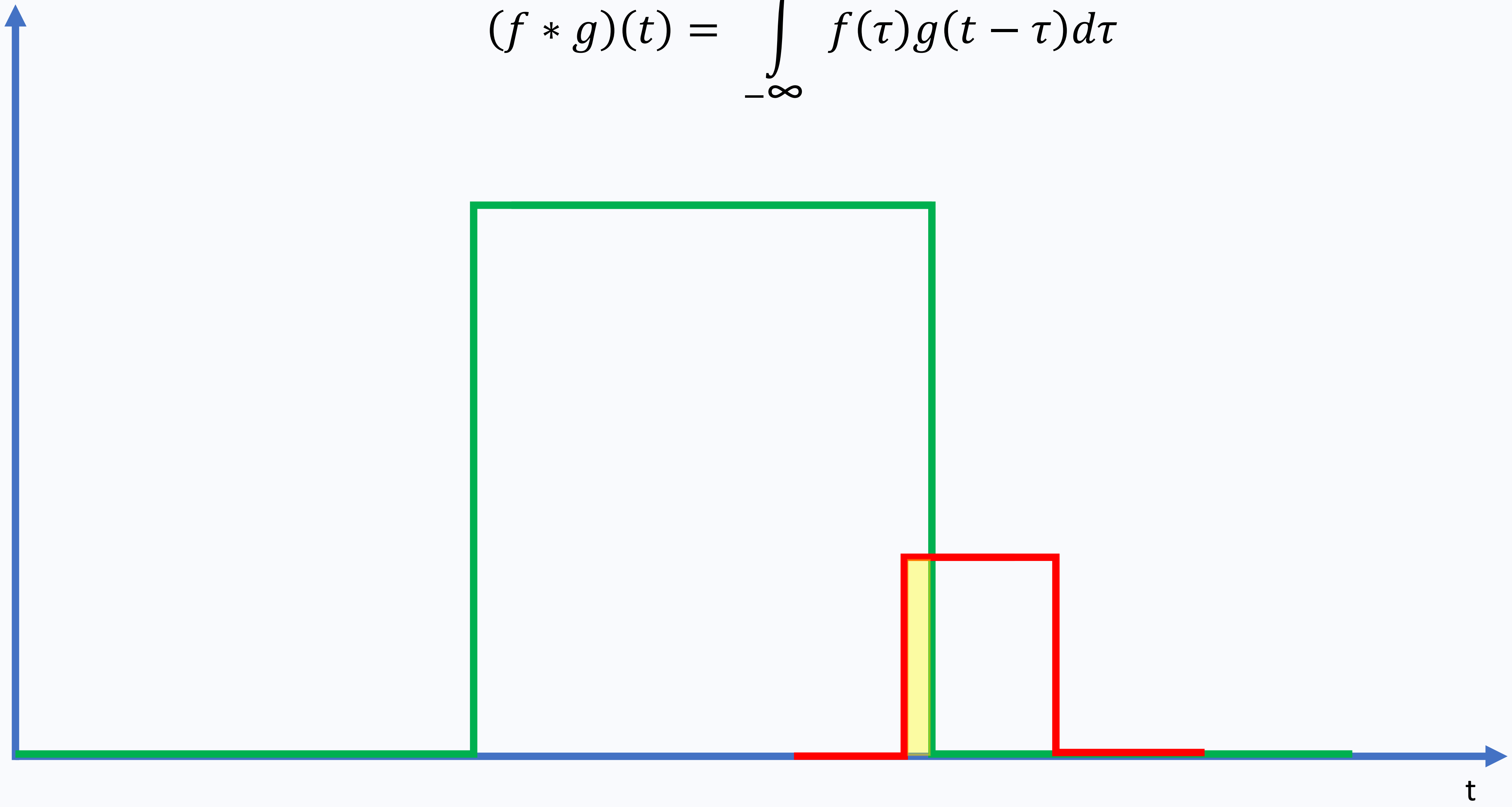


$$(f * g)(t) = \int_{-\infty}^{+\infty} f(\tau)g(t-\tau)d\tau$$

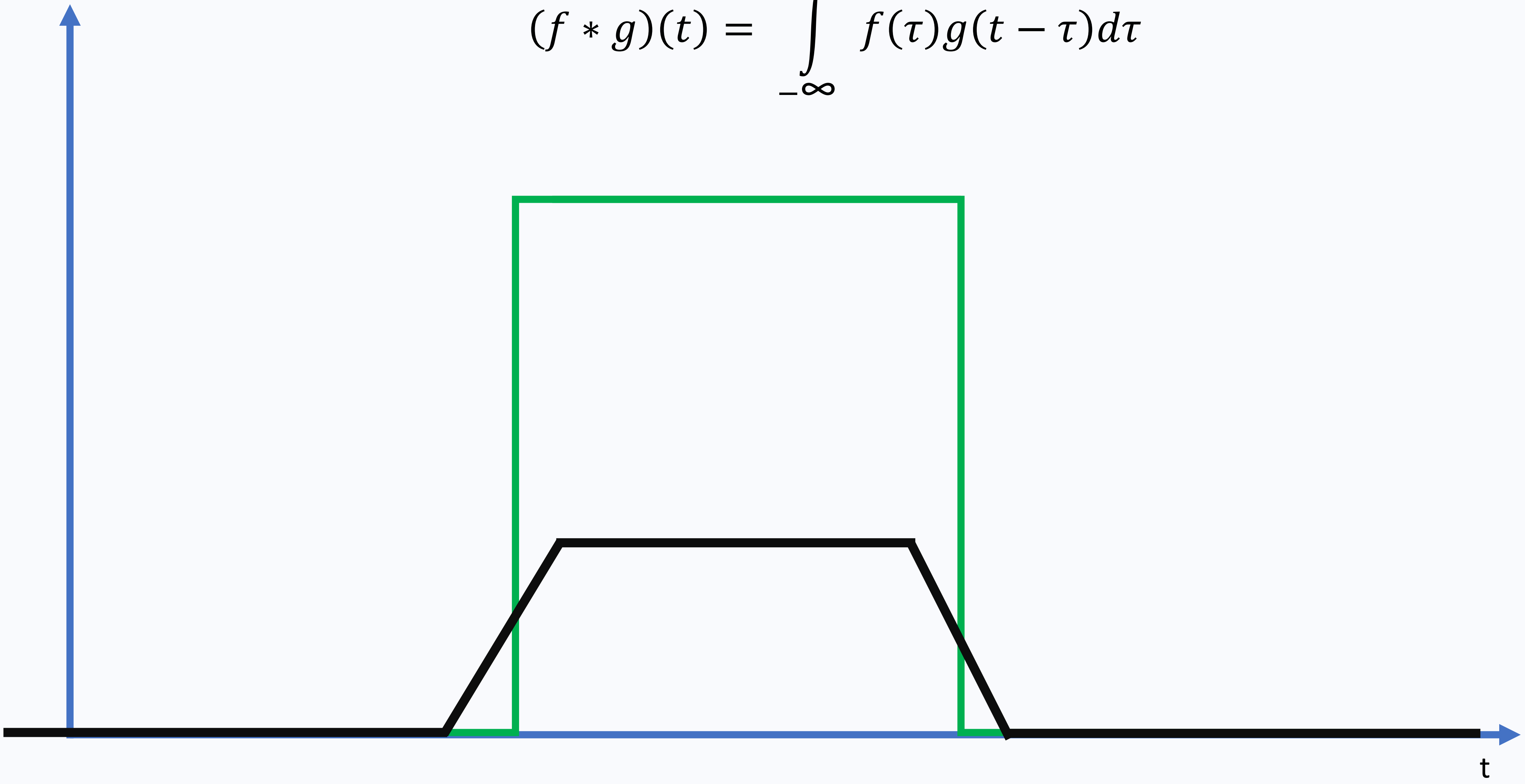


Convolution in 1D

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



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



filter

w_{00}	w_{01}	w_{02}
w_{10}	w_{11}	w_{12}
w_{20}	w_{21}	w_{22}

Image

a_{00}	a_{01}	a_{02}	a_{03}	a_{04}	a_{05}
a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
a_{20}	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}
a_{30}	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}
a_{40}	a_{41}	a_{42}	a_{43}	a_{44}	a_{45}
a_{50}	a_{51}	a_{52}	a_{53}	a_{54}	a_{55}

Convolution in deep learning (in math class it is cross-correlation)

filter

w_{00}	w_{01}	w_{02}
w_{10}	w_{11}	w_{12}
w_{20}	w_{21}	w_{22}

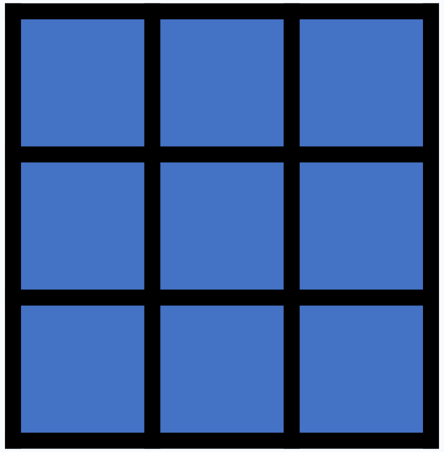
Image

a_{00}	a_{01}	a_{02}	a_{03}	a_{04}	a_{05}
a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
a_{20}	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}
a_{30}	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}
a_{40}	a_{41}	a_{42}	a_{43}	a_{44}	a_{45}
a_{50}	a_{51}	a_{52}	a_{53}	a_{54}	a_{55}

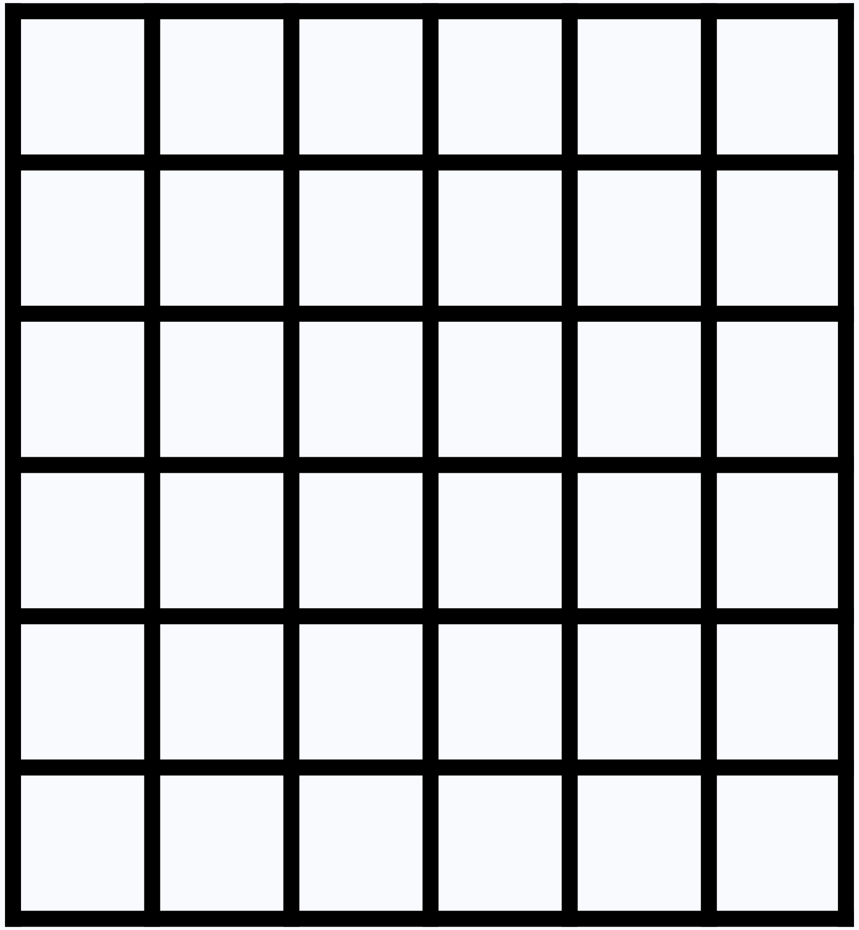
Note: Image-processing/math convolution is slightly different.
The kernel/filter is flipped around both axises before the multiplication.

$$a'_{11} = w_{00} \cdot a_{00} + w_{01} \cdot a_{01} + w_{02} \cdot a_{02} + w_{10} \cdot a_{10} + w_{11} \cdot a_{11} + \\ + w_{12} \cdot a_{12} + w_{20} \cdot a_{20} + w_{21} \cdot a_{21} + w_{22} \cdot a_{22}$$

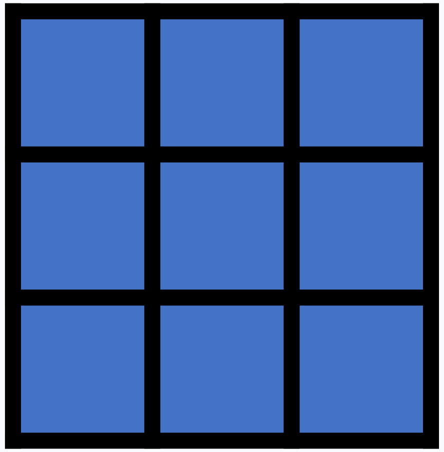
filter



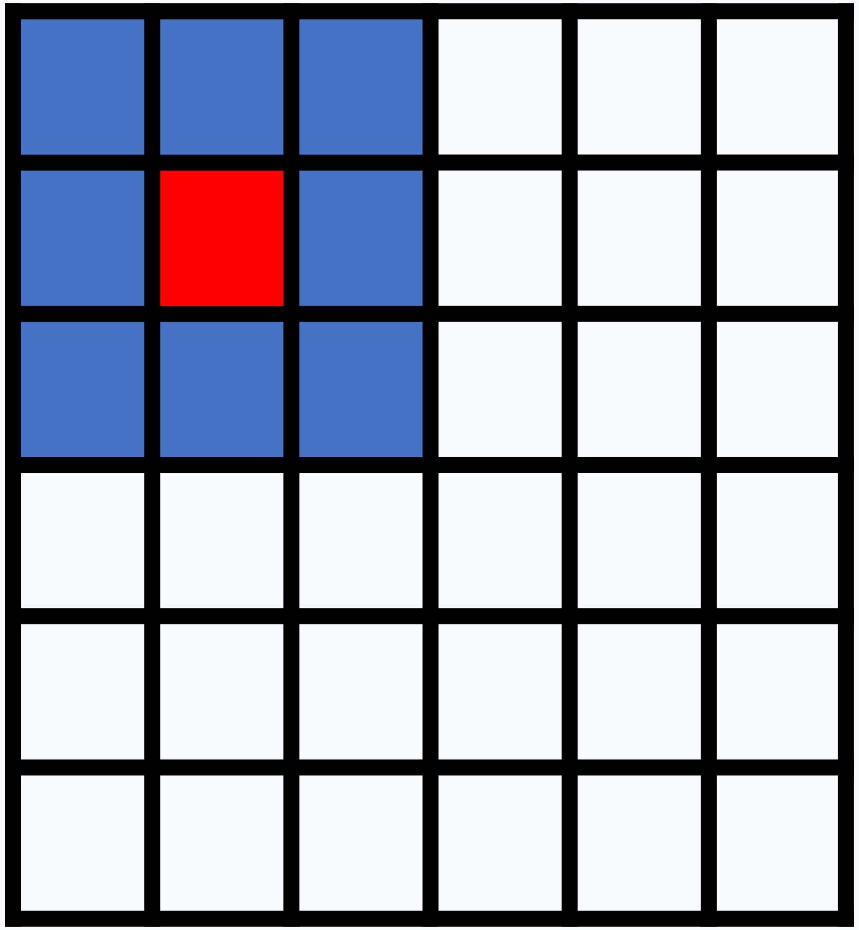
Image



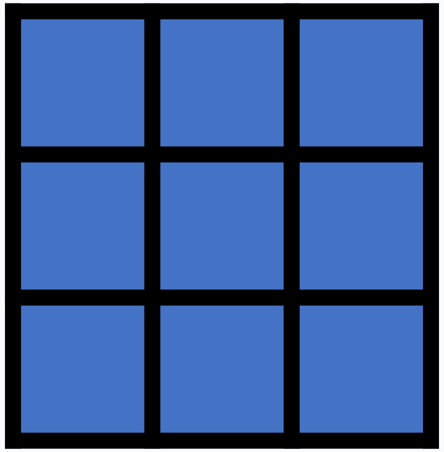
filter



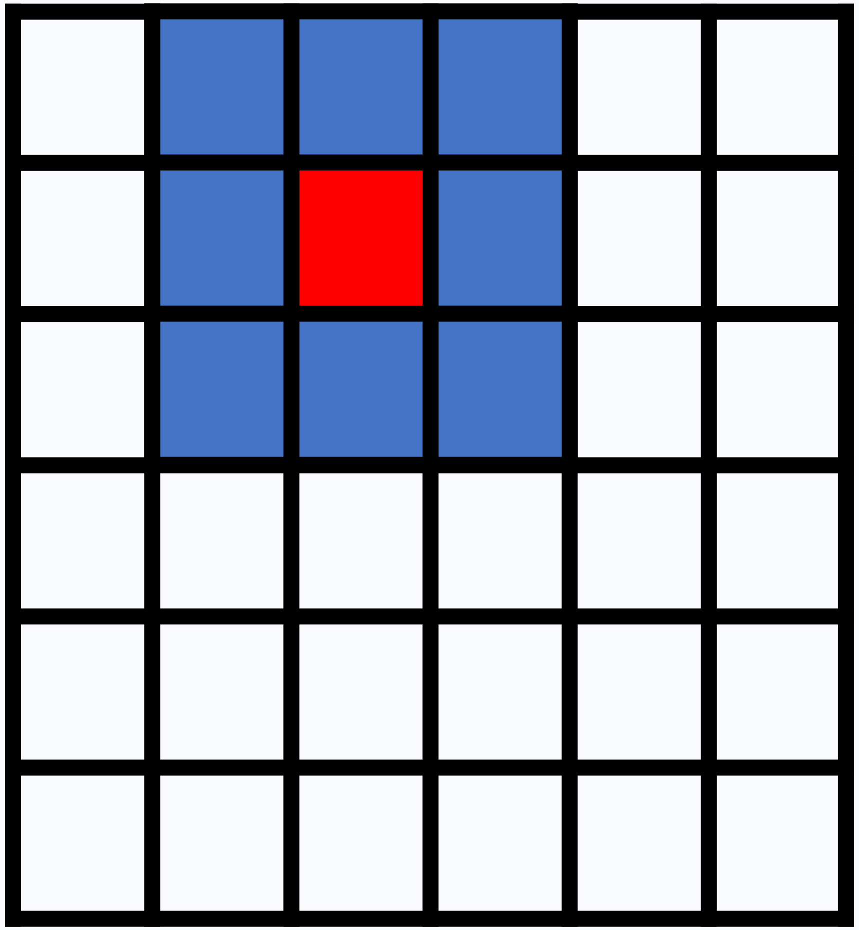
Image

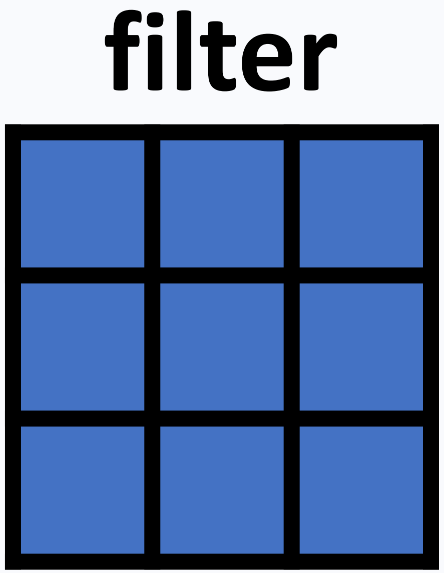


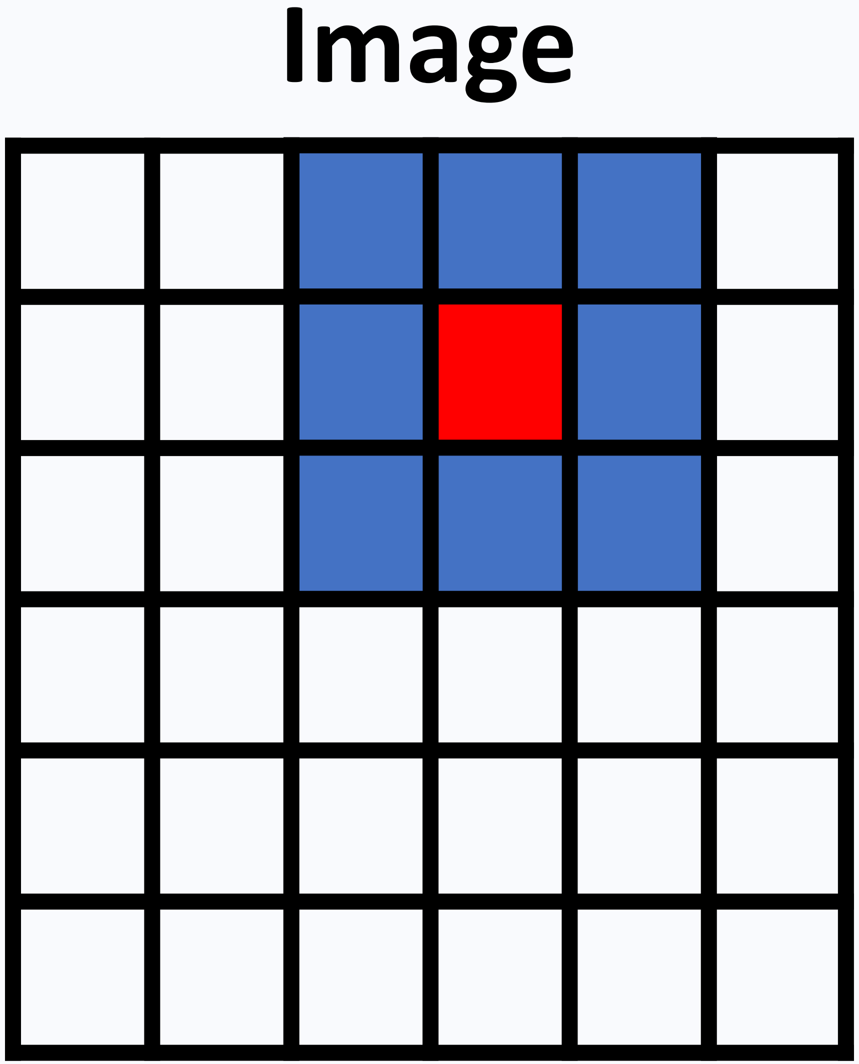
filter



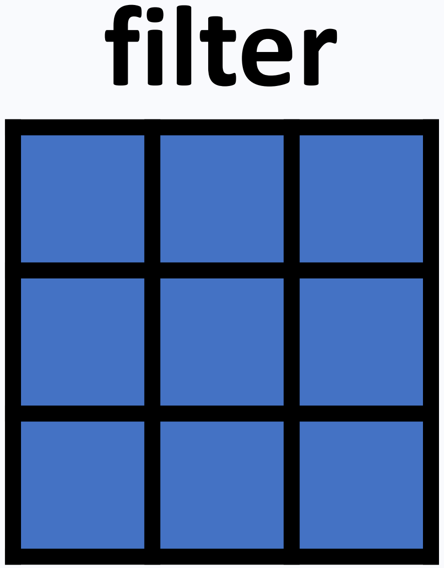
Image

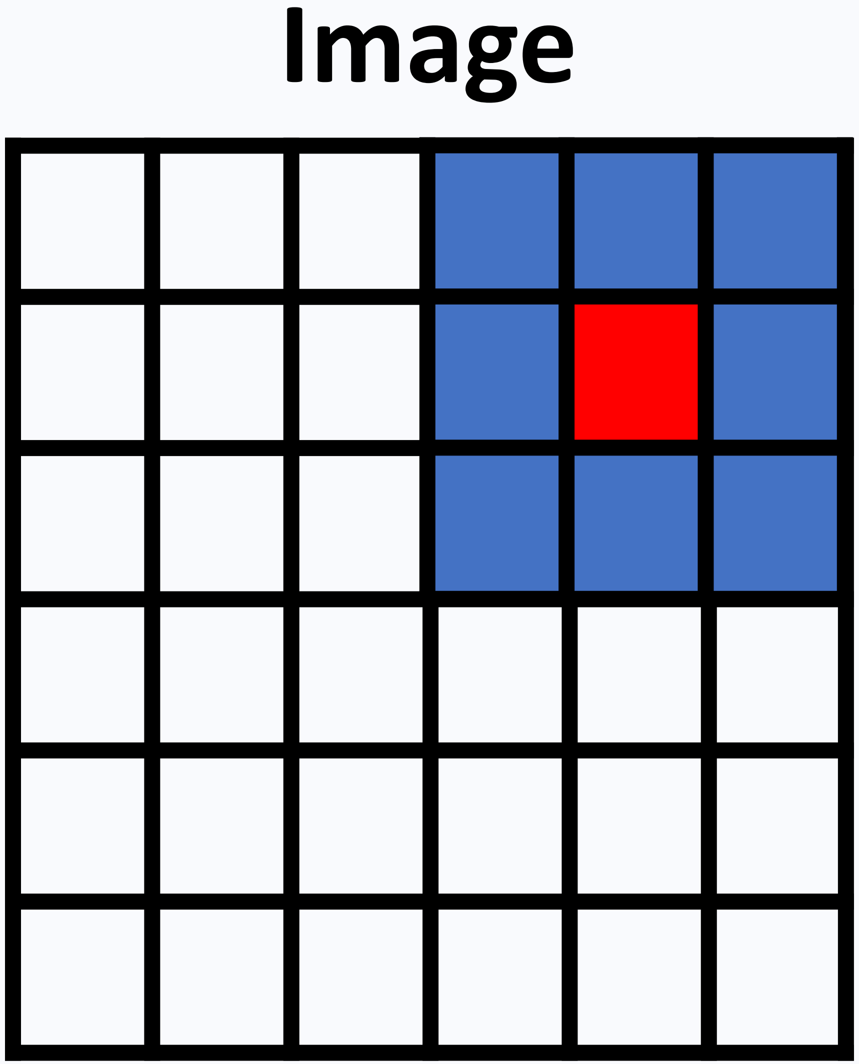



filter



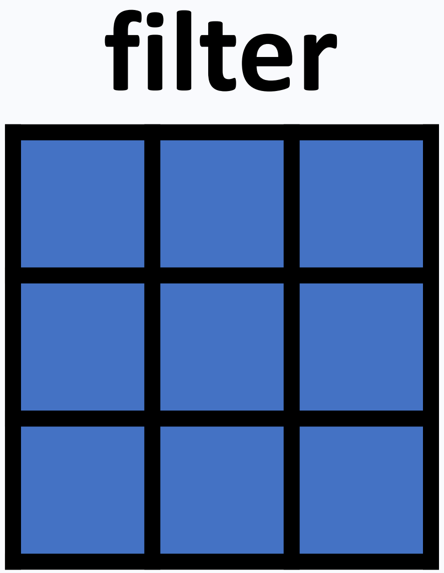
Image

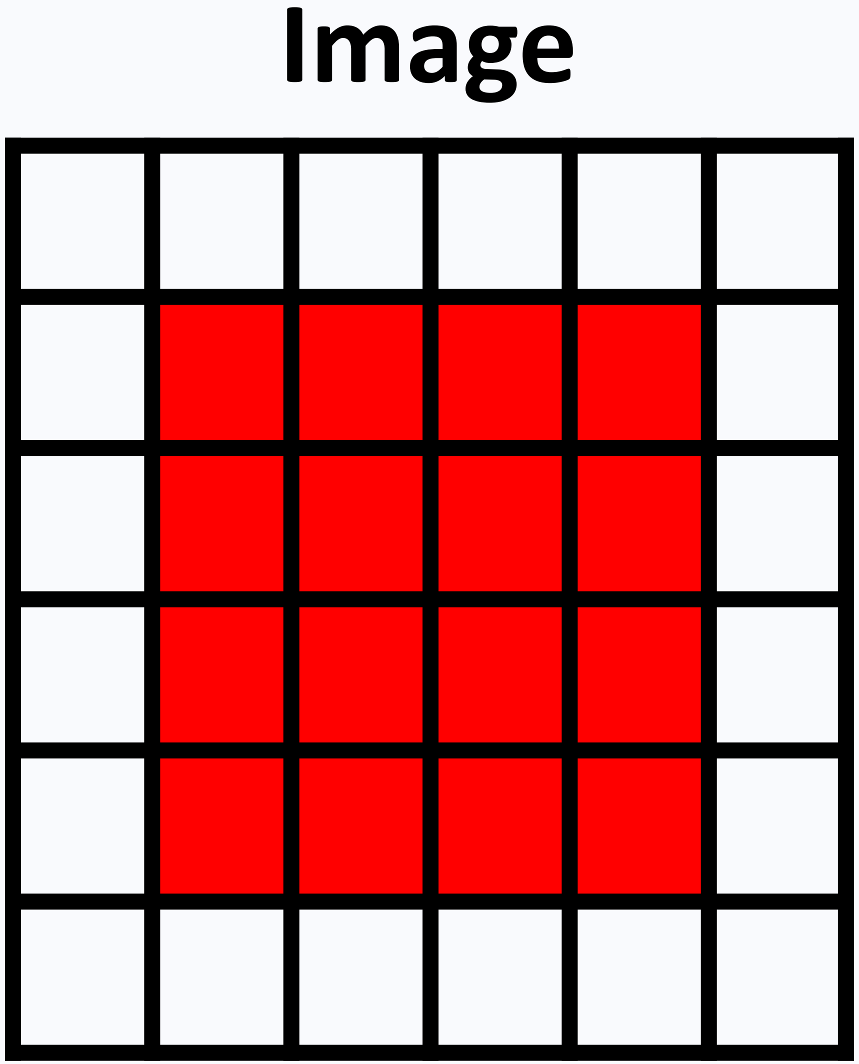

filter

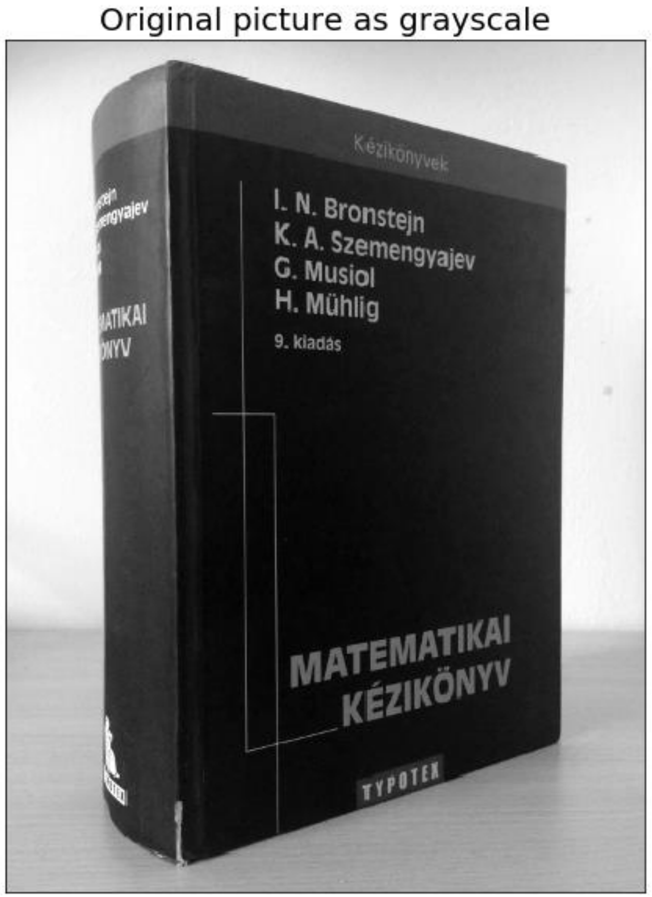


Image

Convolution for images







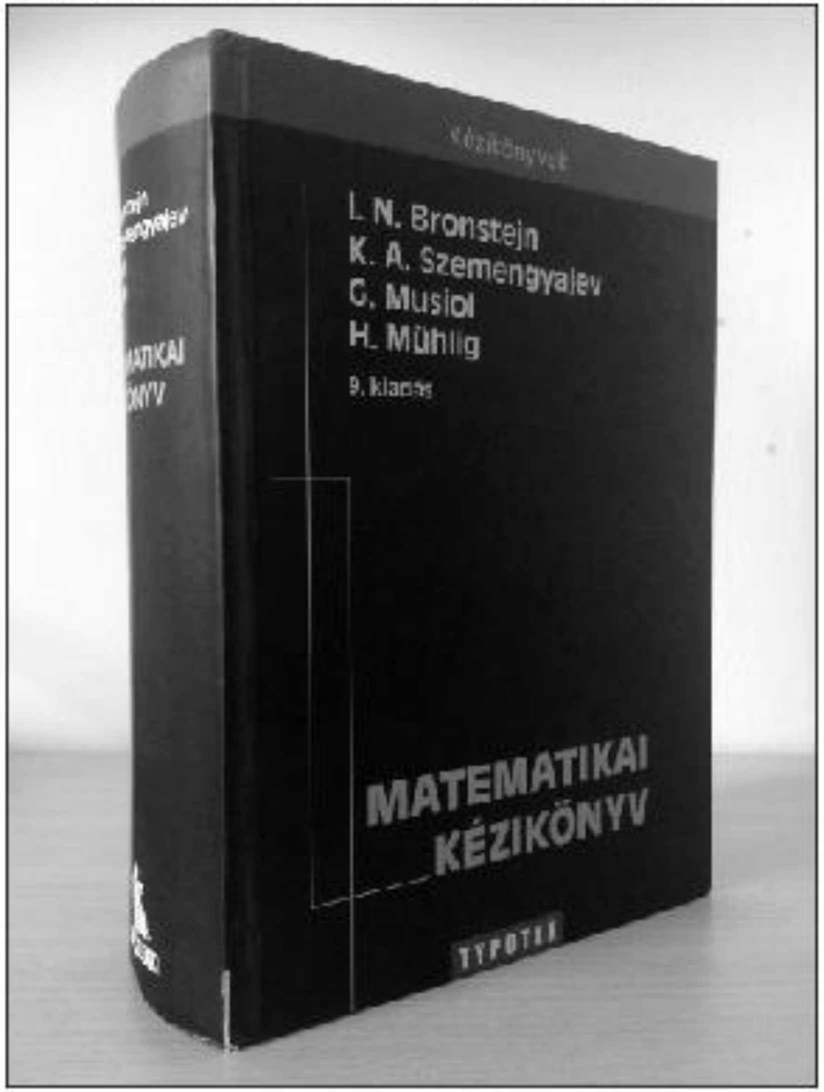
*

filter

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

Question: what do you expect?

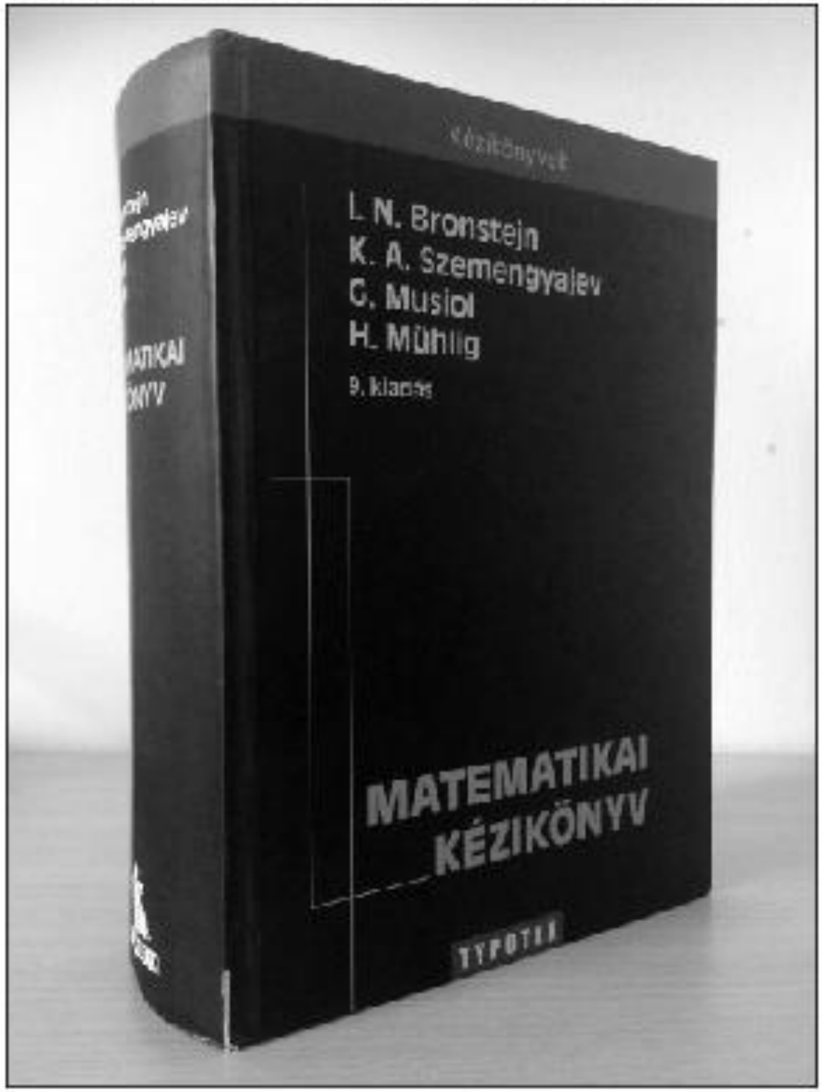
Original picture as grayscale



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

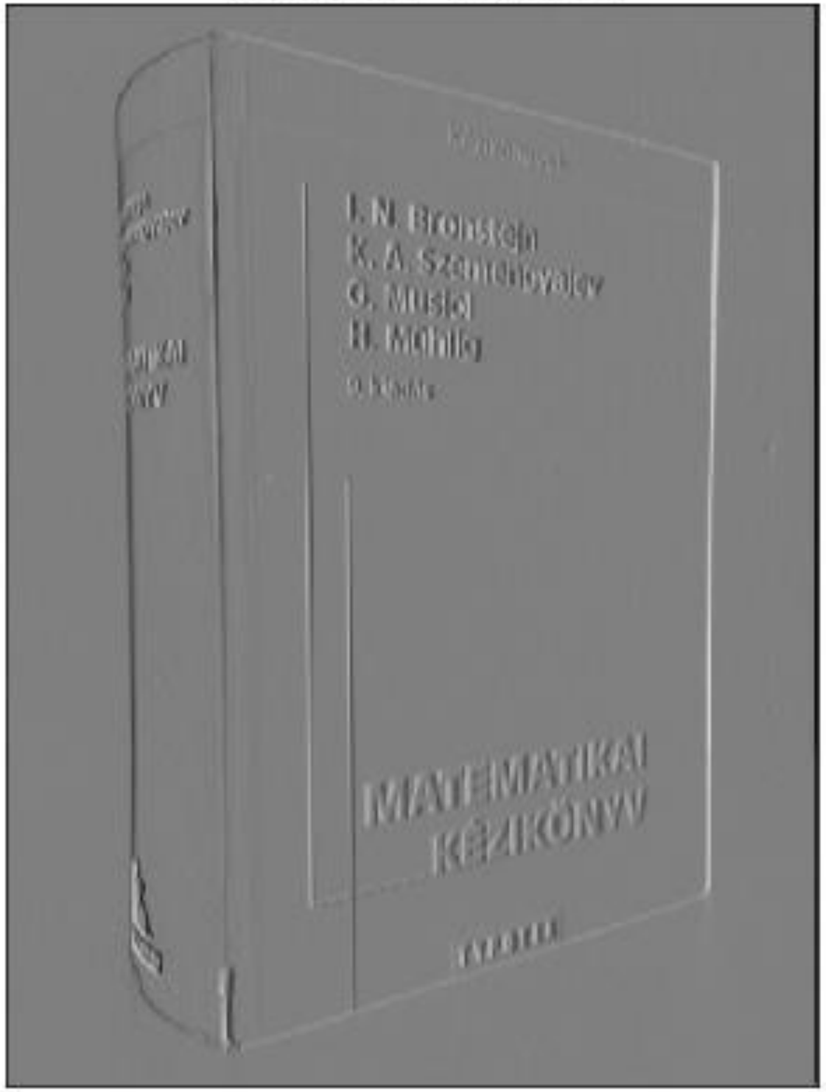
Horizontal edges

Original picture as grayscale

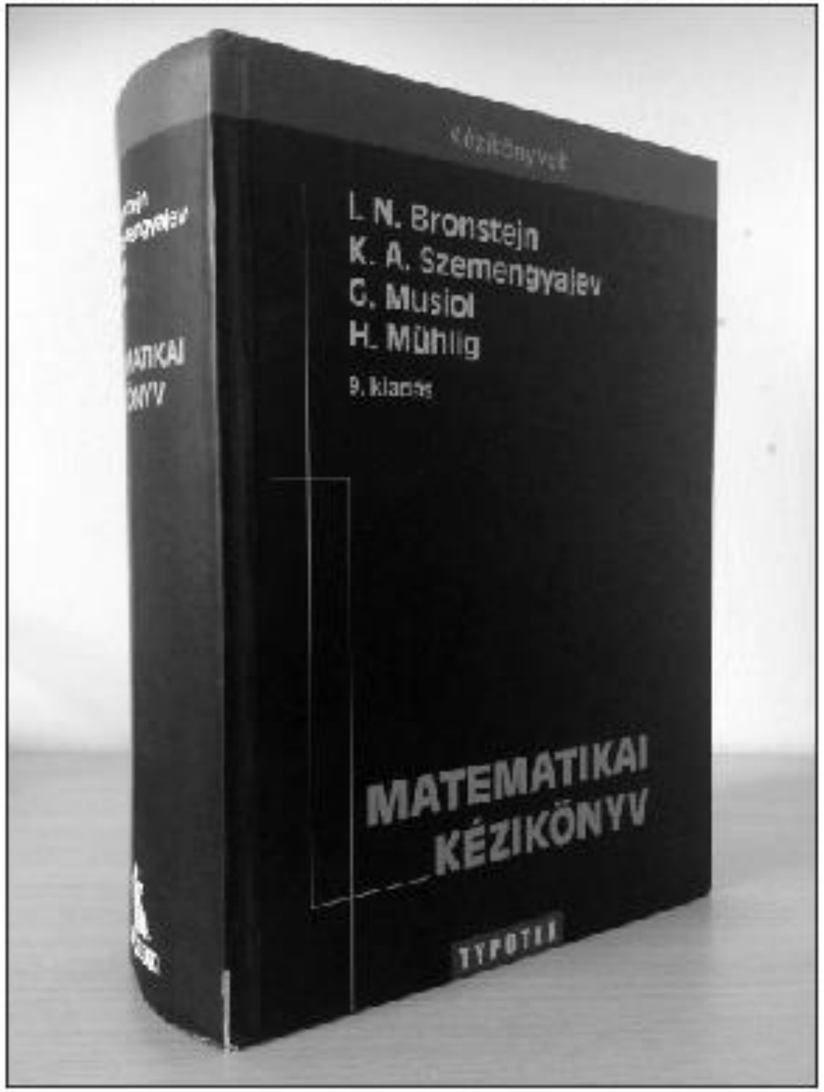


$$* \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} =$$

Vertical edges

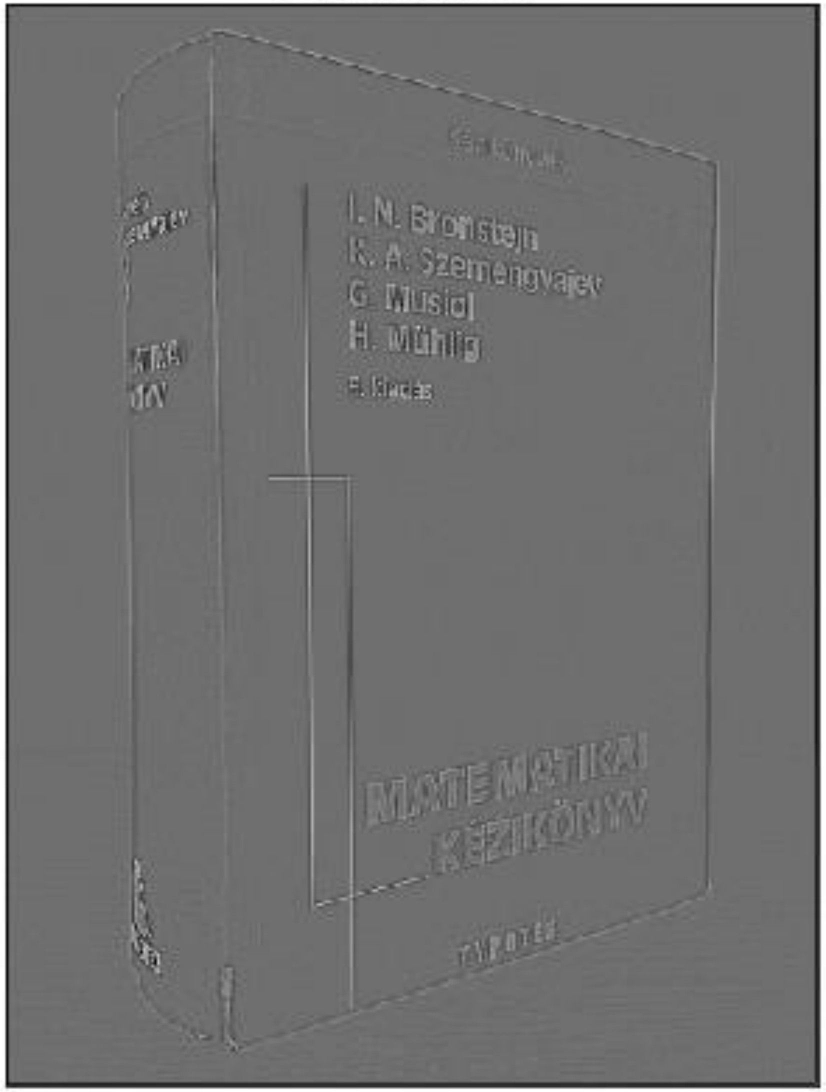


Original picture as grayscale

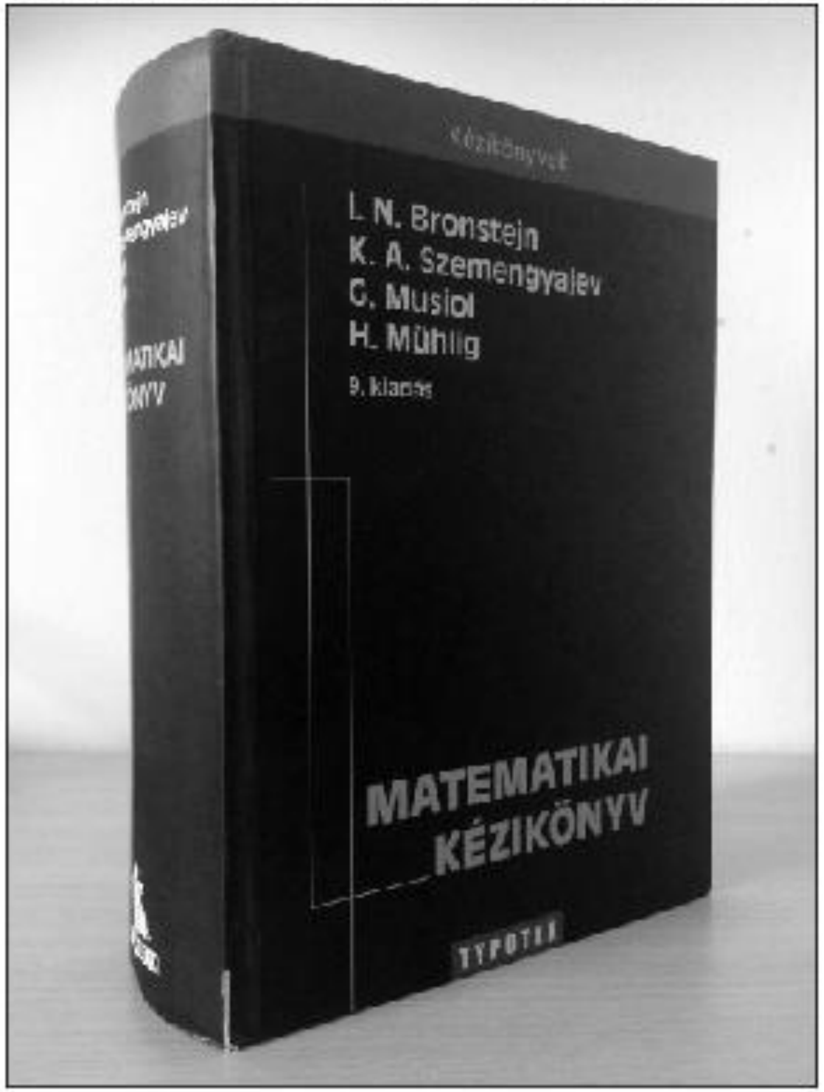


$$* \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} =$$

All edges



Original picture as grayscale

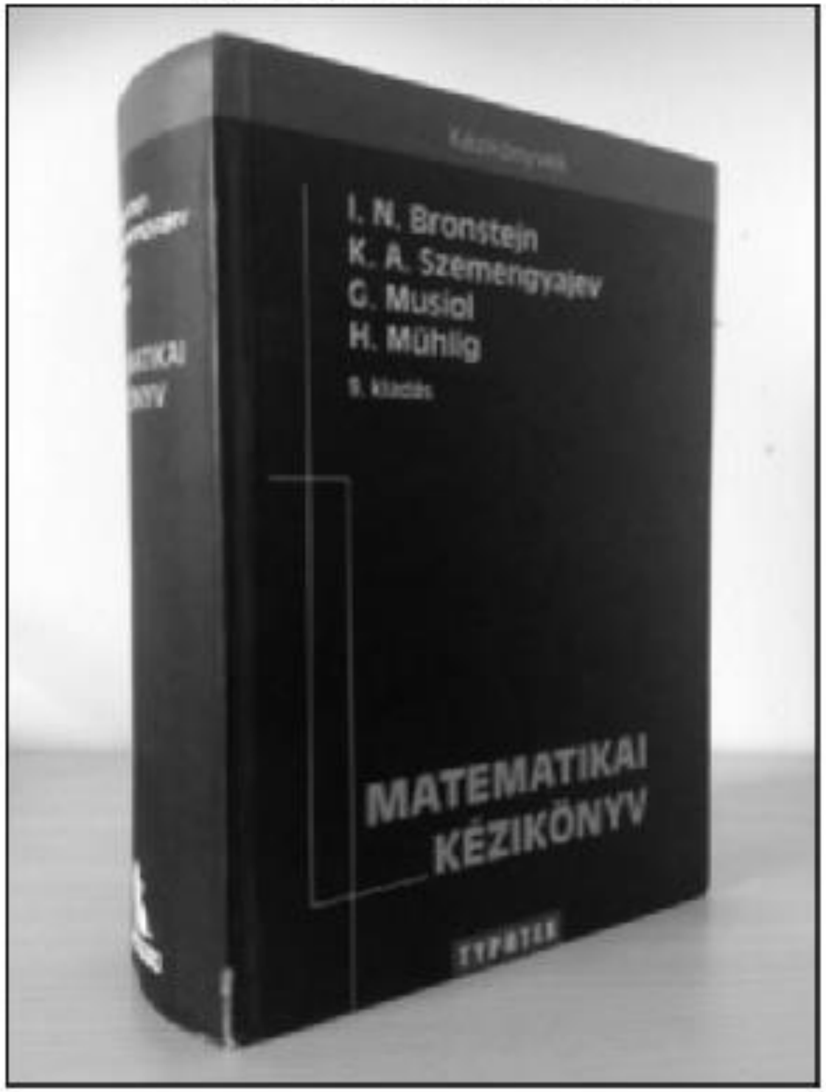


*

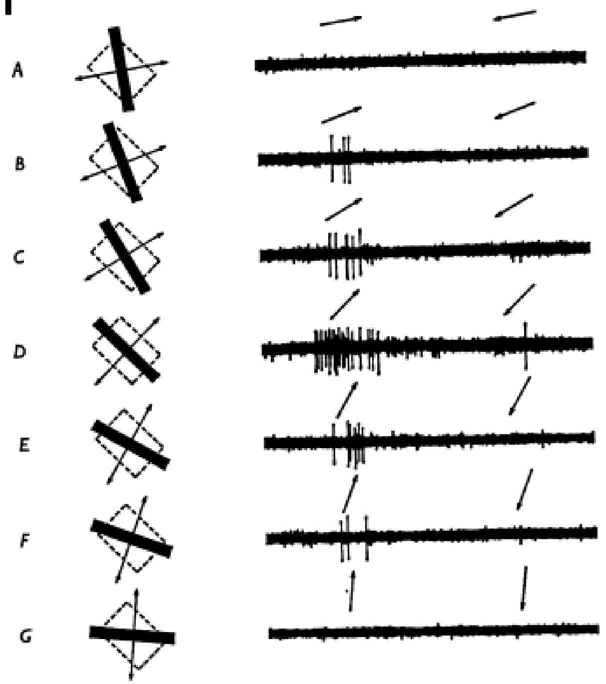
1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	/19

=

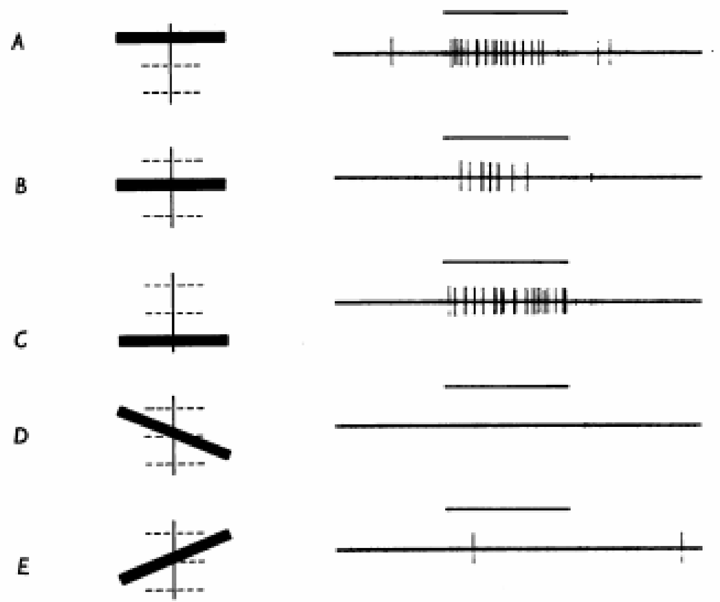
Smoothed picture



- Hubel, Wiesel, Sperry
- cat's &monkey's vision
- Electrodes to the brain
- 1981 Nobel prize



Text-fig. 2. Responses of a complex cell in right striate cortex (layer IV A) t various orientations of a moving black bar. Receptive field in the left eye indicate by the interrupted rectangles; it was approximately $\frac{3}{8} \times \frac{3}{8}°$ in size, and was situate $4°$ below and to the left of the point of fixation. Ocular-dominance group 4. Duratio of each record, 2 sec. Background intensity $1{\cdot}3\ \log_{10}\ cd/m^2$, dark bars $0{\cdot}0$ log cd/m



Text-fig. 7. Cell activated only by left (contralateral) eye over a field approximately $5 \times 5°$, situated $10°$ above and to the left of the area centralis. The cell responded best to a black horizontal rectangle, $\frac{1}{3} \times 6°$, placed anywhere in the receptive field (*A–C*). Tilting the stimulus rendered it ineffective (*D–E*). The black bar was introduced against a light background during periods of 1 sec, indicated by the upper line in each record. Luminance of white background, $1{\cdot}0\ \log_{10}\ cd/m^2$; luminance of black part, $0{\cdot}0\ \log_{10}\ cd/m^2$. A lesion, made while recording from the cell, was found in layer 2 of apical segment of post-lateral gyrus.

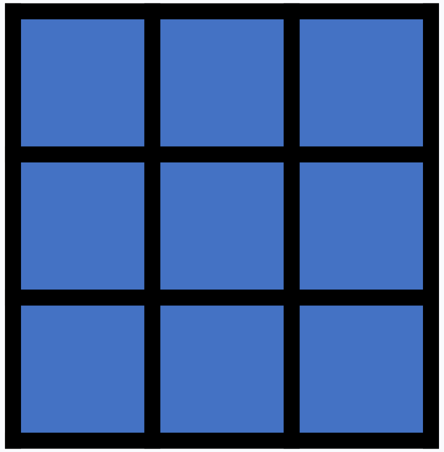
[Hubel, Wiesel: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex, 1961]

[Hubel, Wiesel: Receptive fields and functional architecture of monkey striate cortex, 1968]

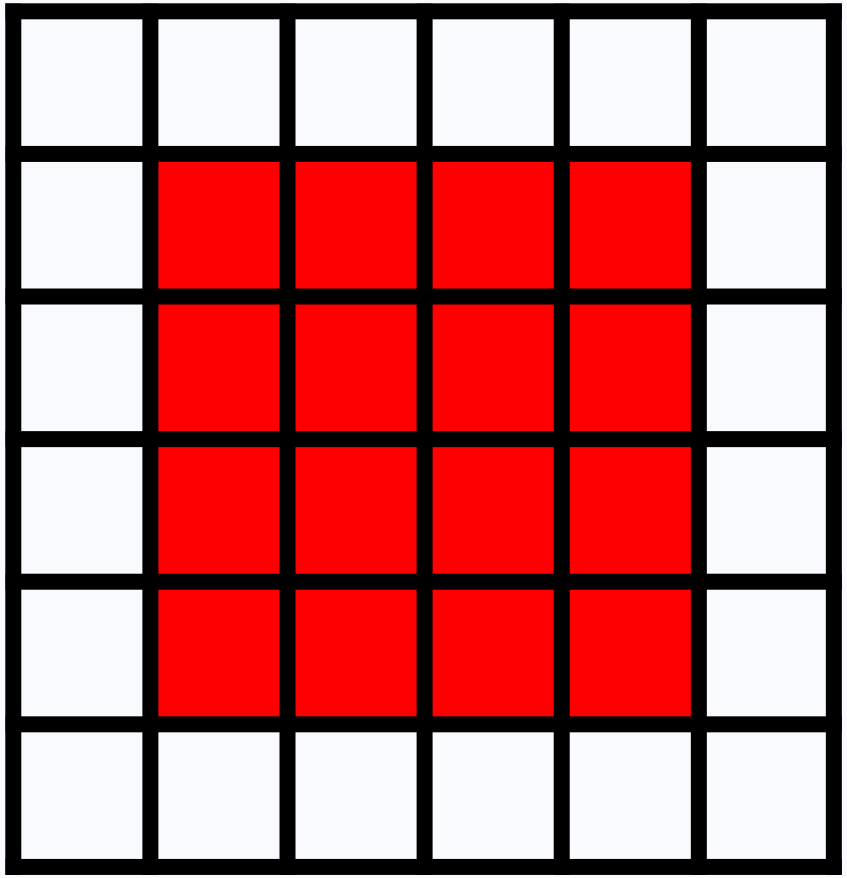
Demo convoluton notebook

https://github.com/qati/DeepLearningCourse/tree/master/demo_notebooks/lecture_05

filter

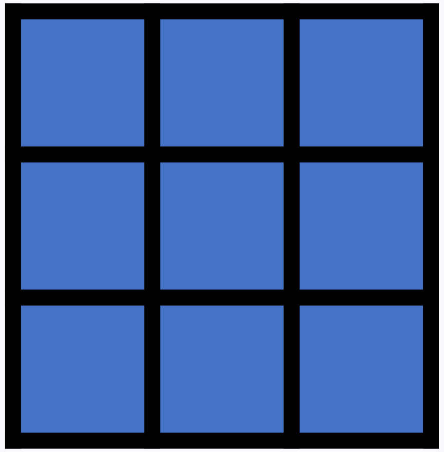


Image

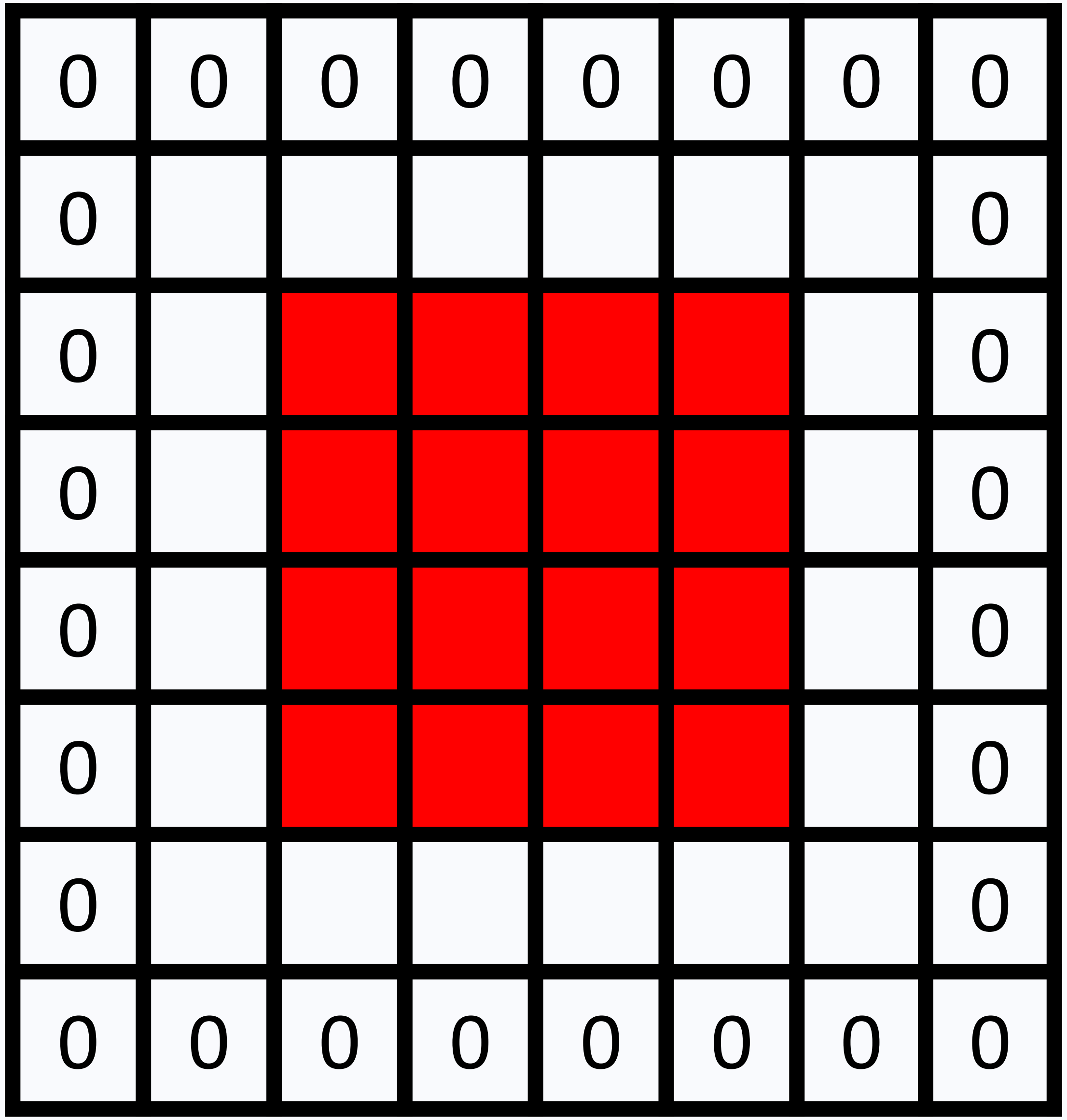


Padding: 1

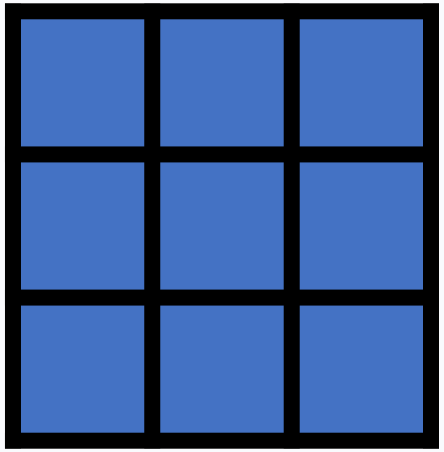
filter



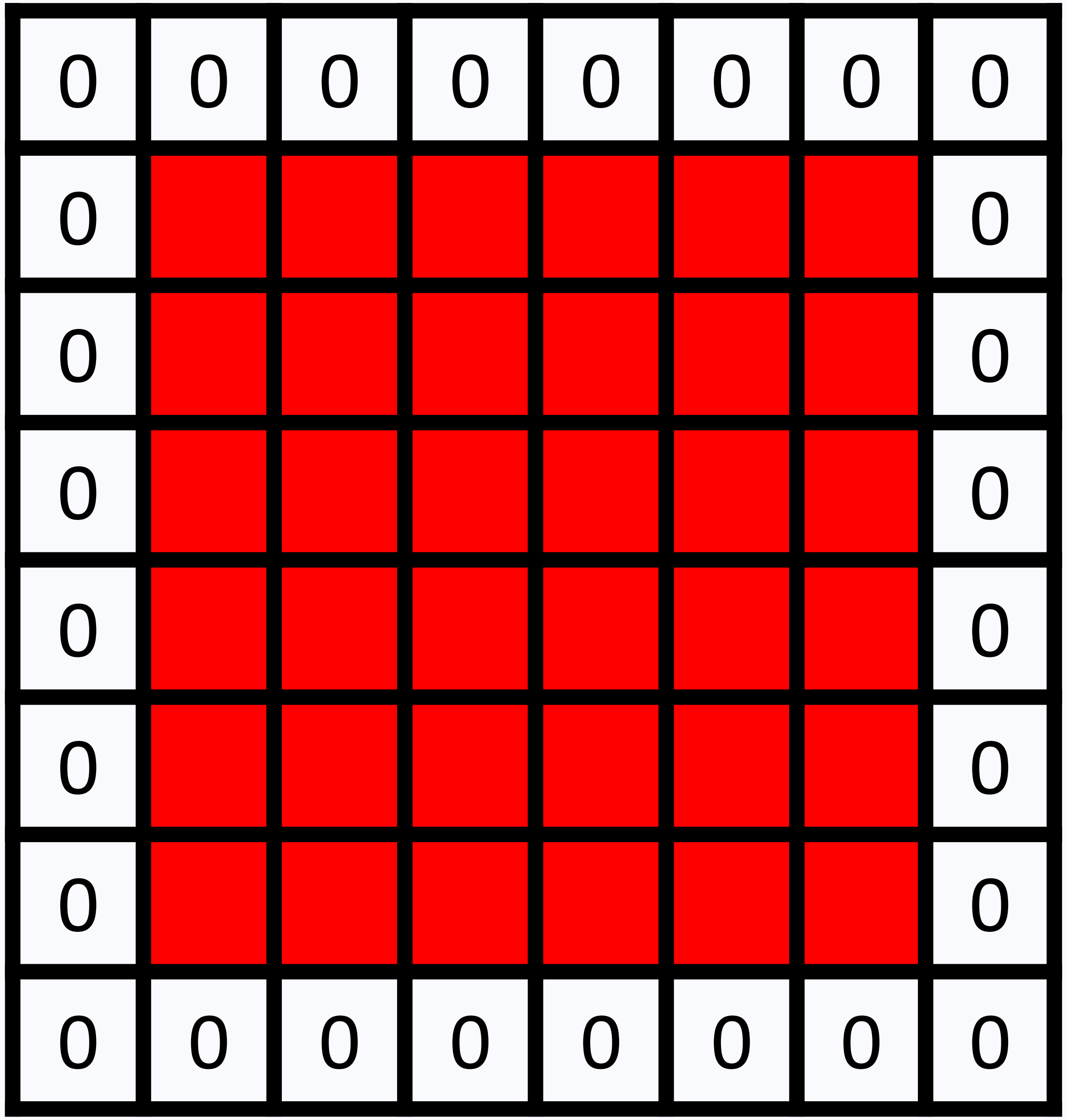
Image

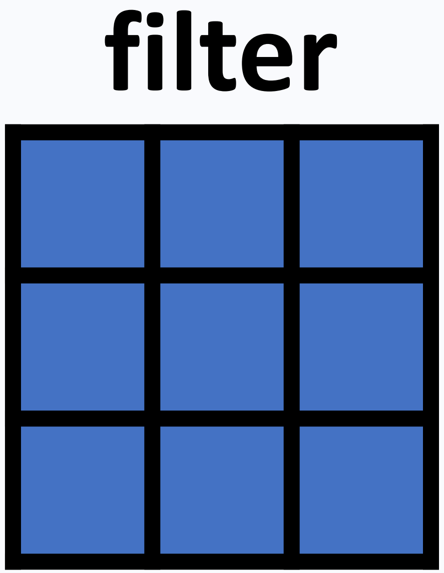


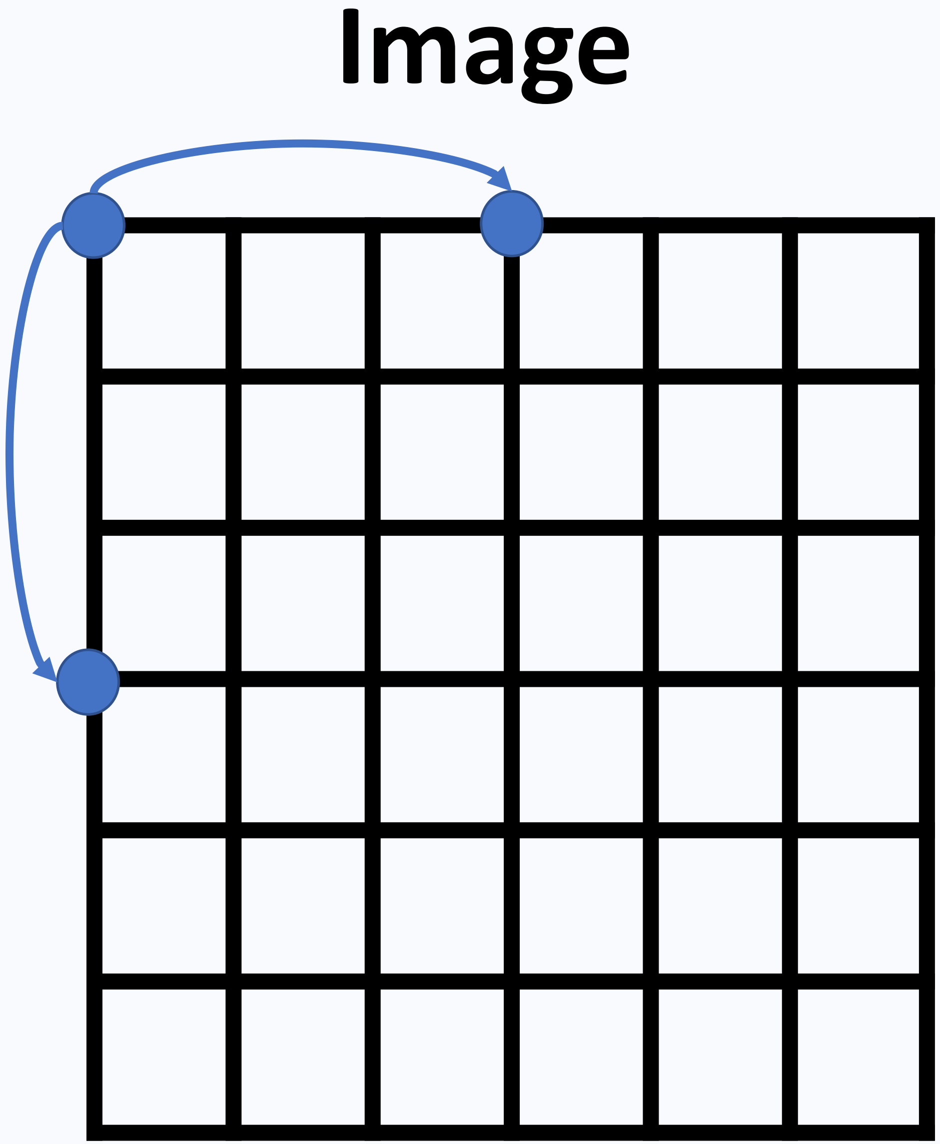
filter



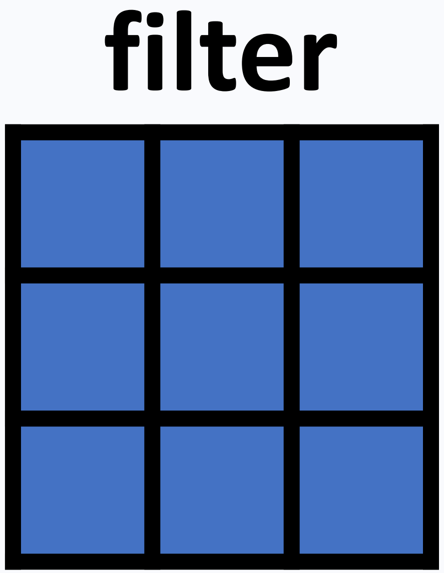
Image

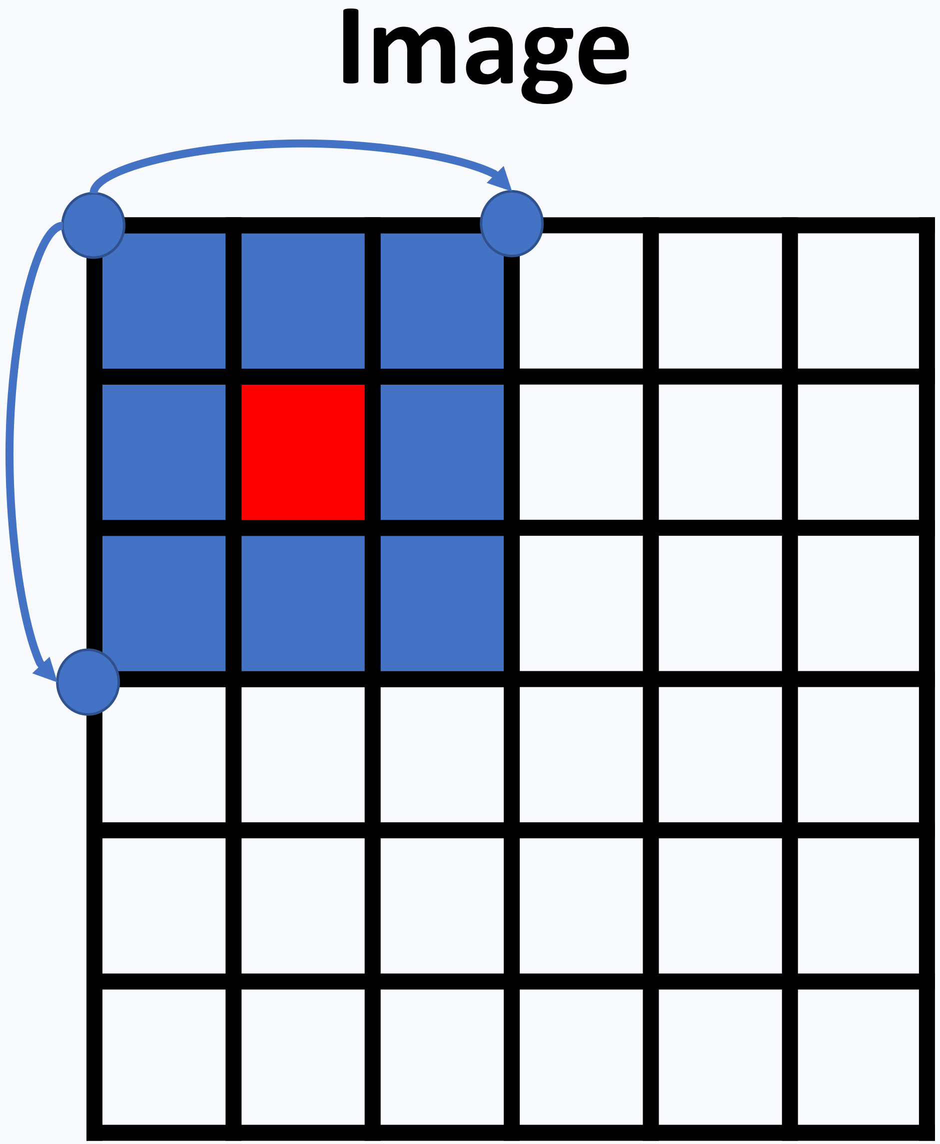



filter

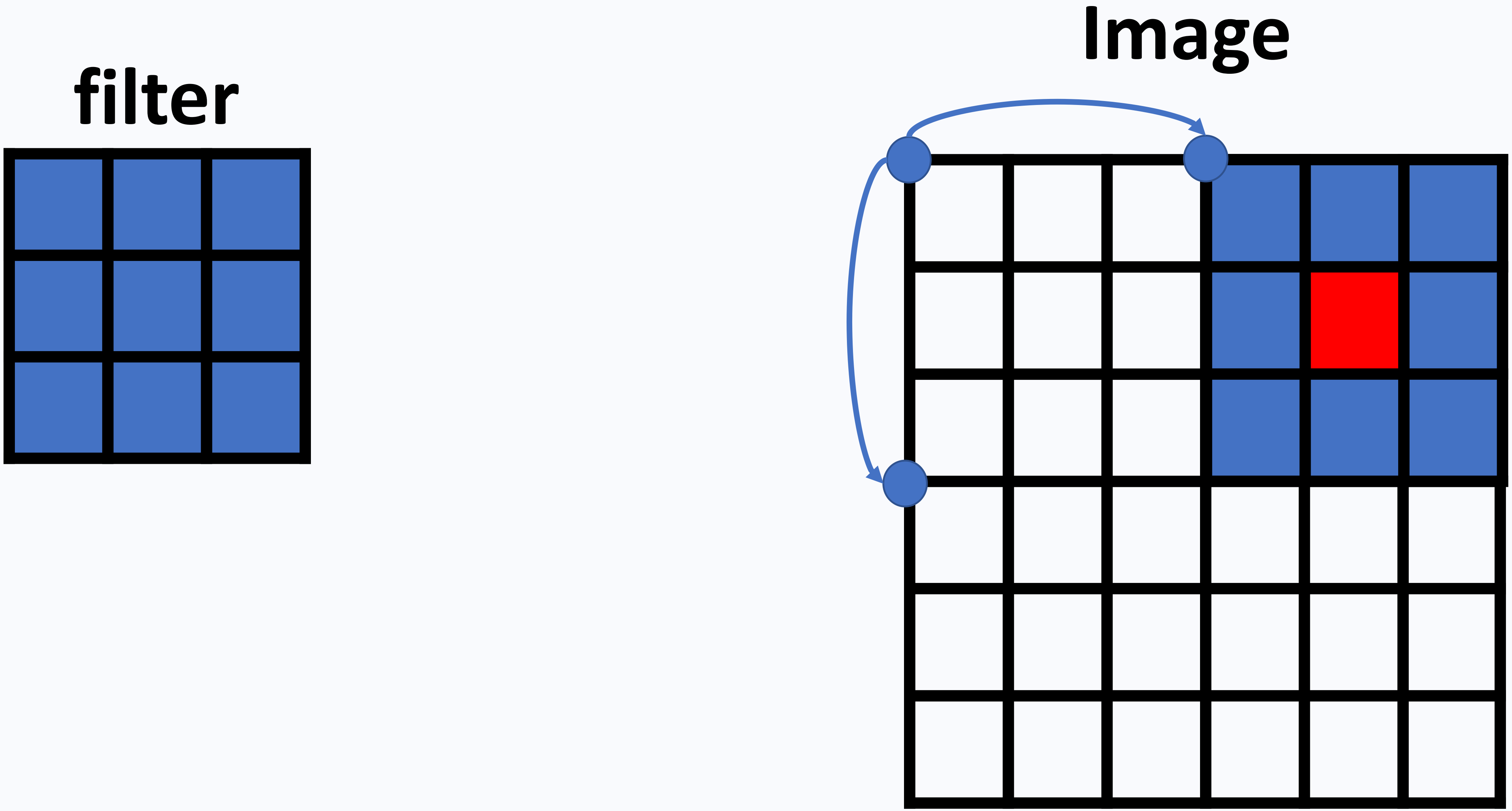


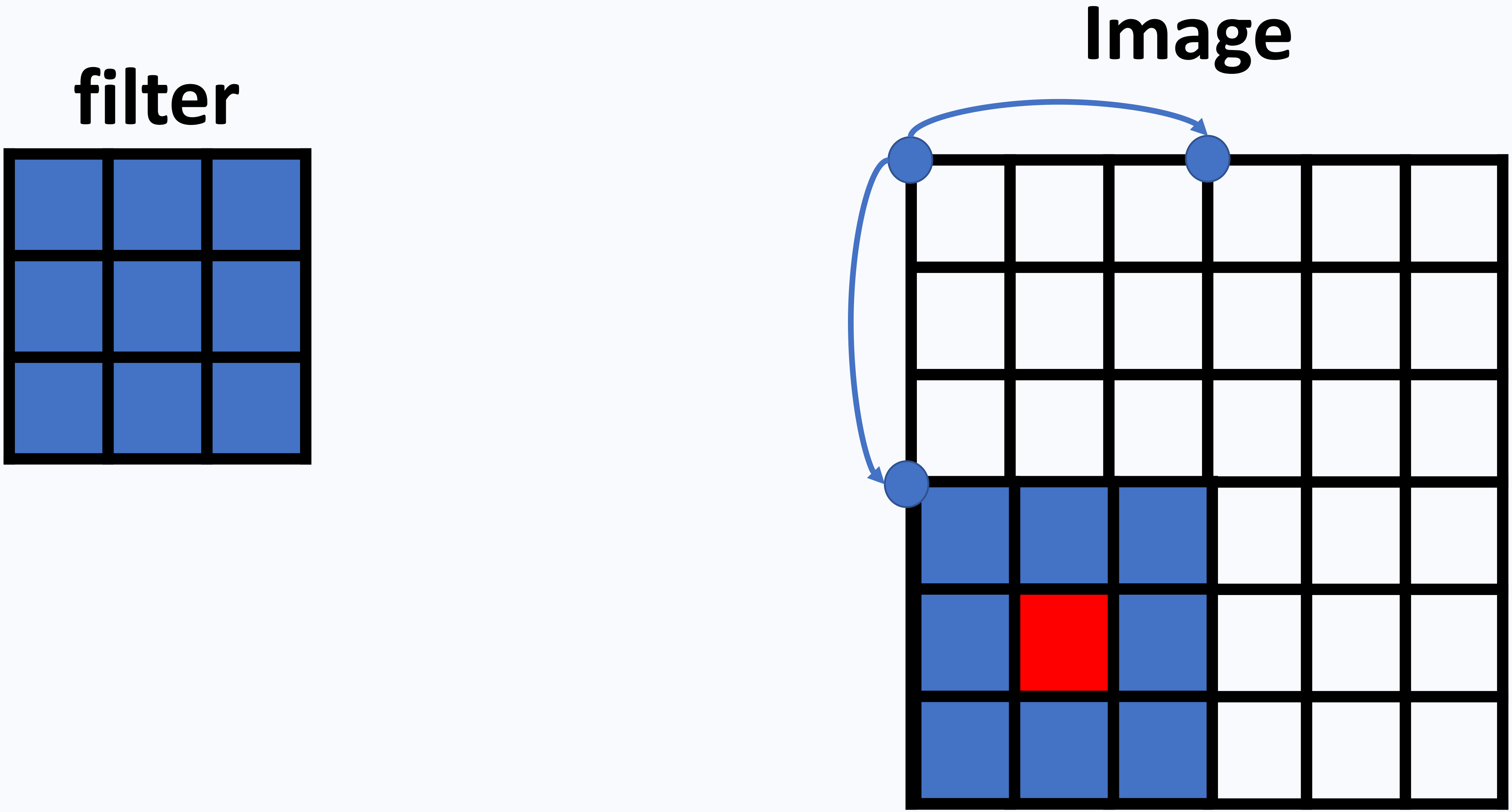
Image

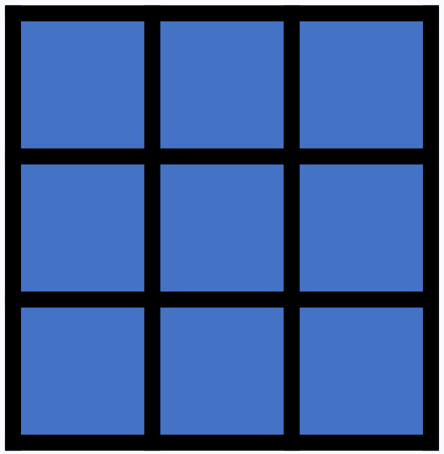

filter

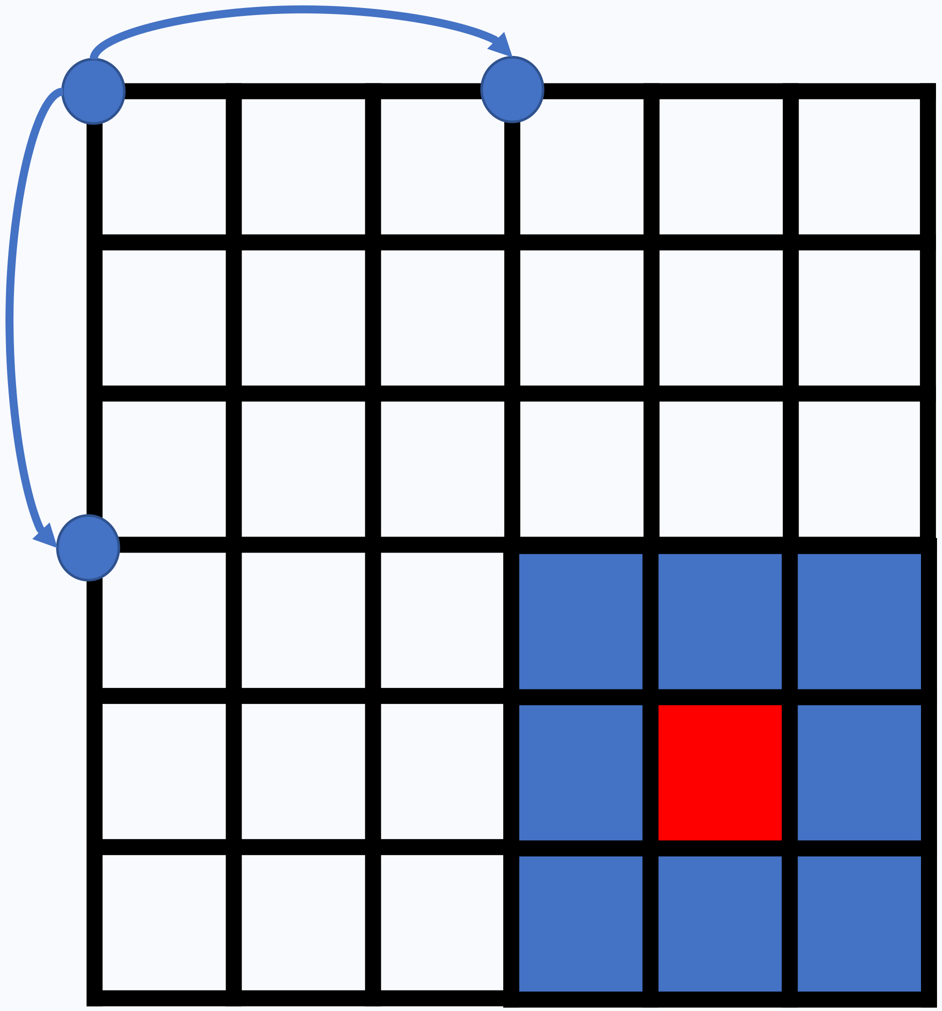


Image


filter
Image

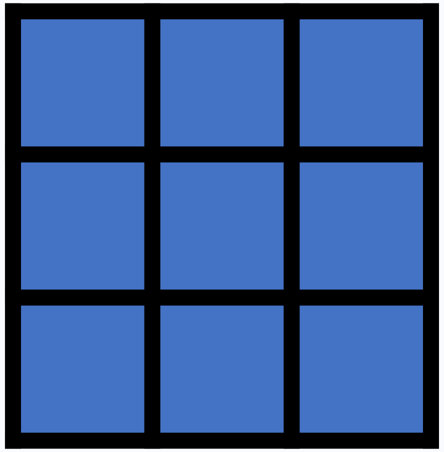

filter
Image


filter

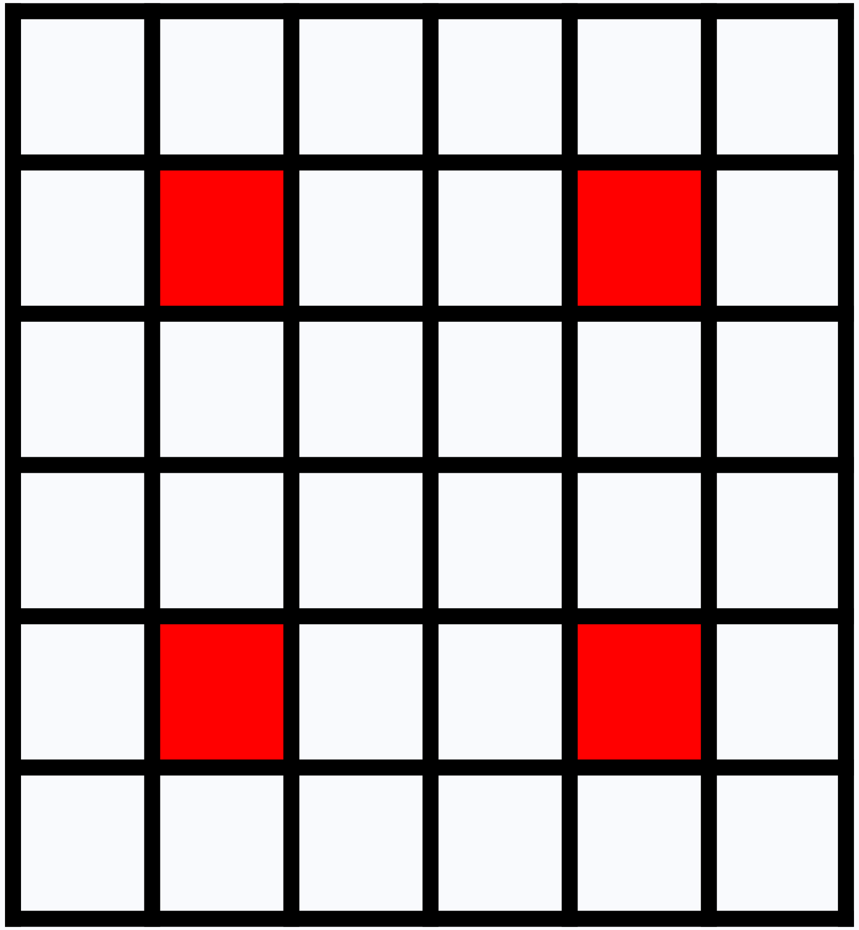


Image

filter



Image



stride: s

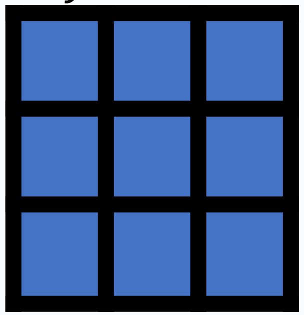
padding: p

filter: $f \times f$

image: $n \times n$

resul

$f = 3$



stride: s

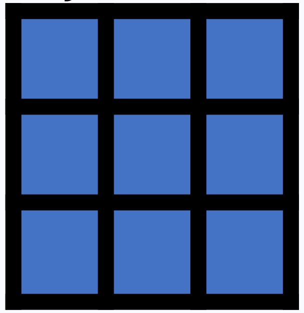
padding: p

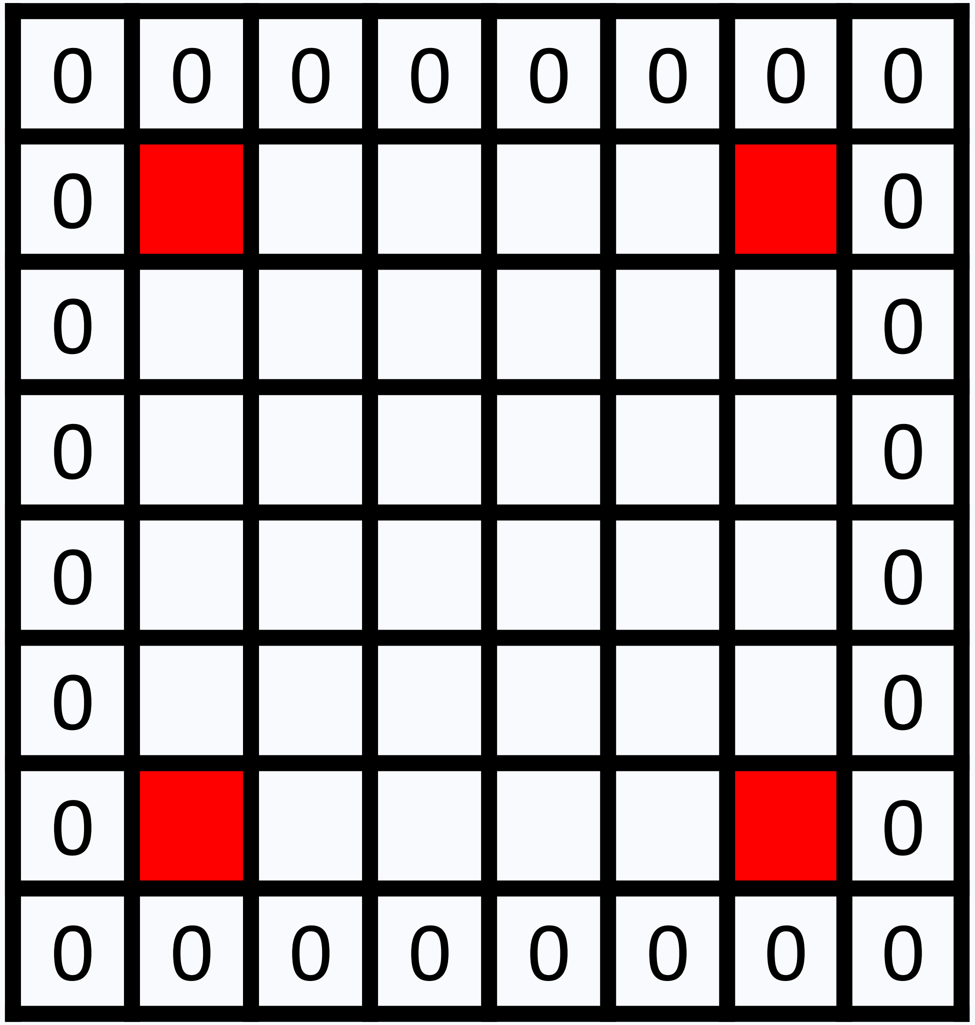
filter: f × f

image: n × n

resul

$f = 3$





$n = 6$

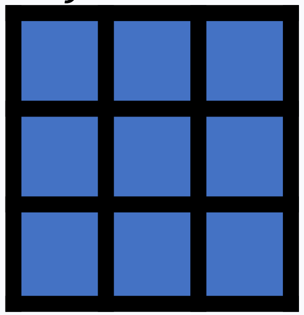
stride: s

padding: p

filter: f × f

image: n × n

resul



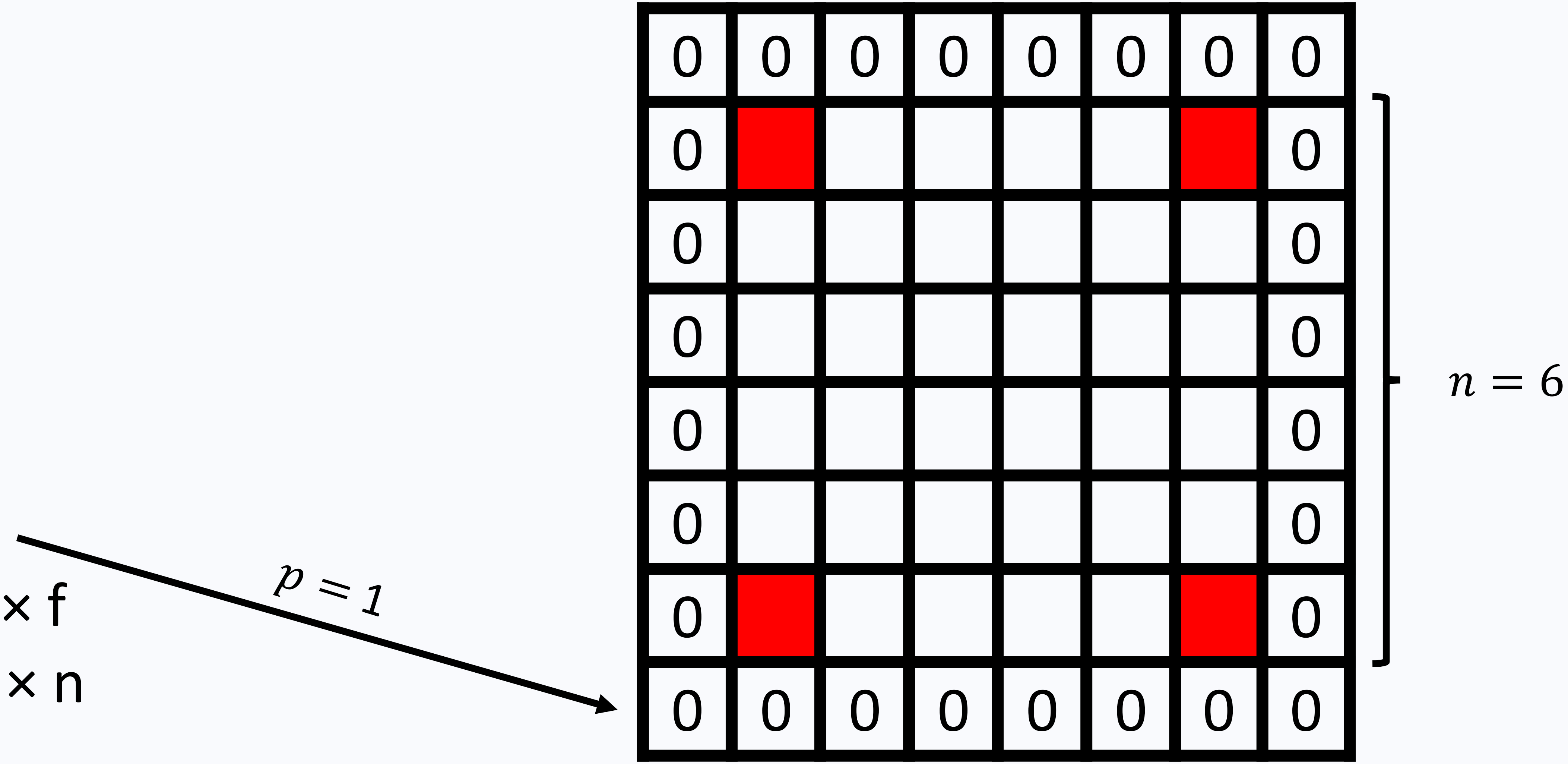
stride: s

padding: p

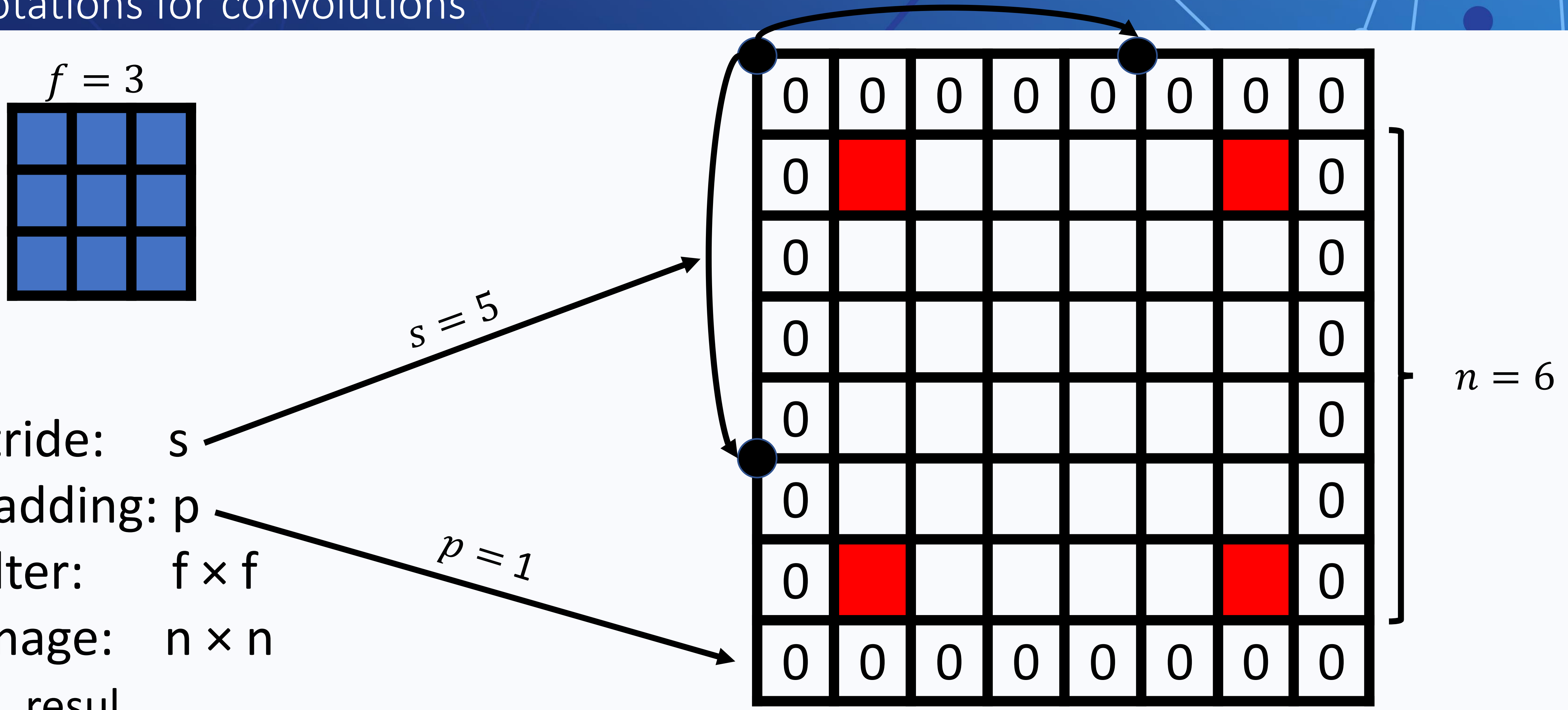
filter: f × f

image: n × n

resul

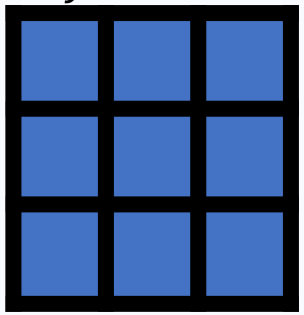


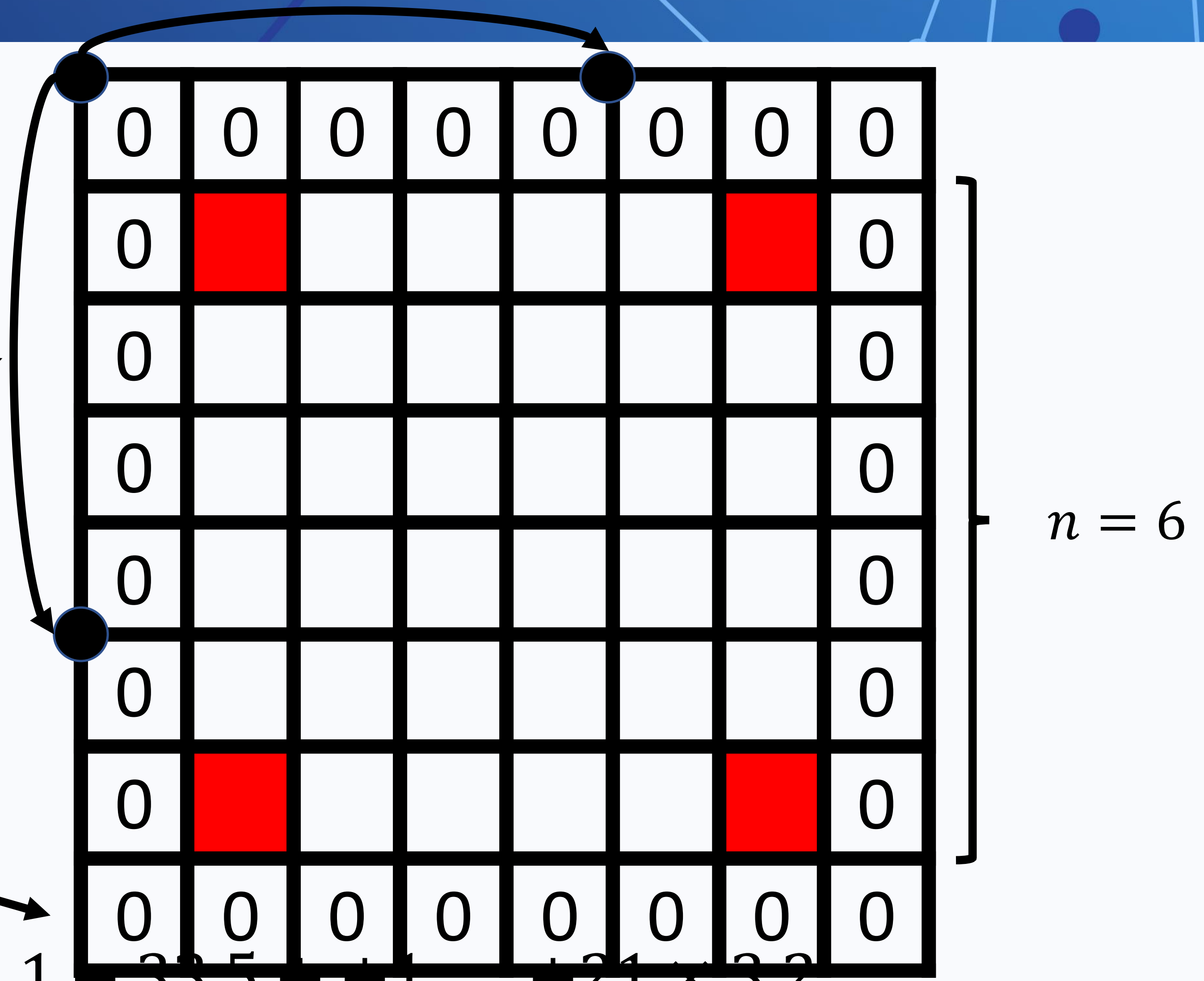
Notations for convolutions



stride: s
padding: p
filter: f × f
image: n × n
resul

$f = 3$





stride: s

padding: p

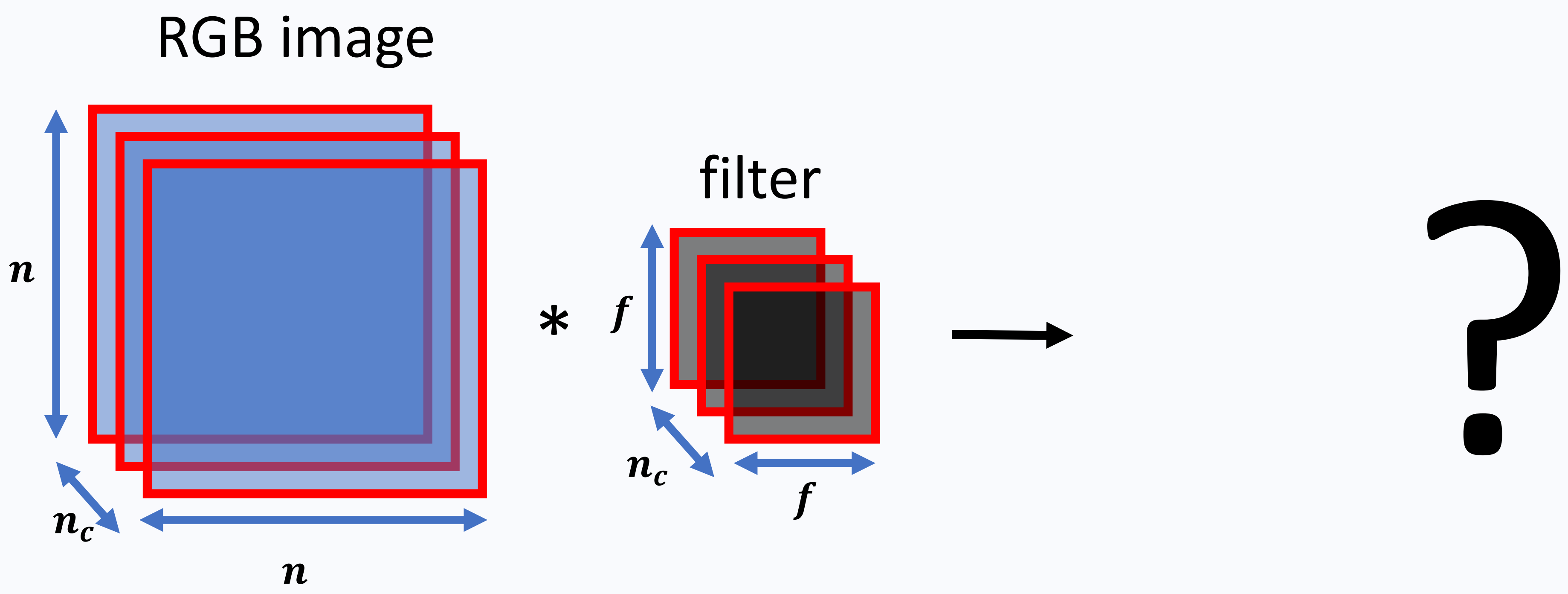
filter: f × f

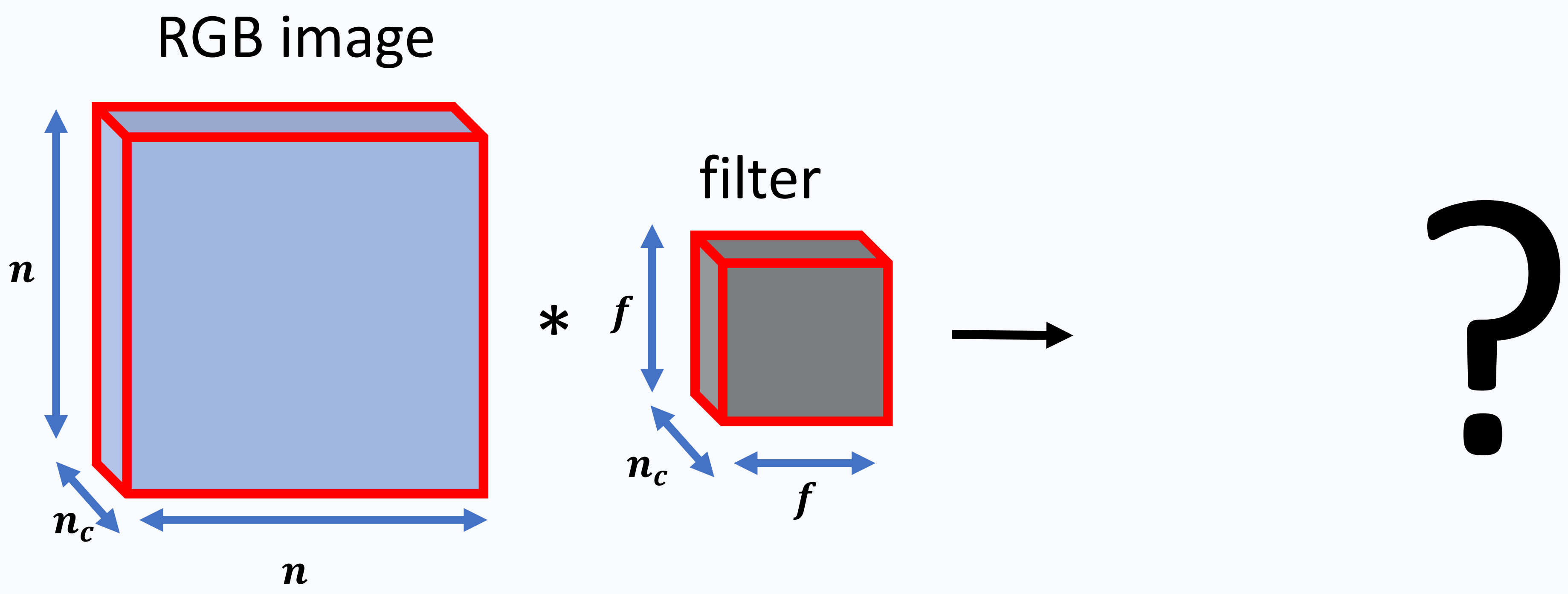
image: n × n

resul +2∗1−3 5 + +1 × + 6 + 2 ∗ 1 − 33 5 + +1 = +21 × 3 2

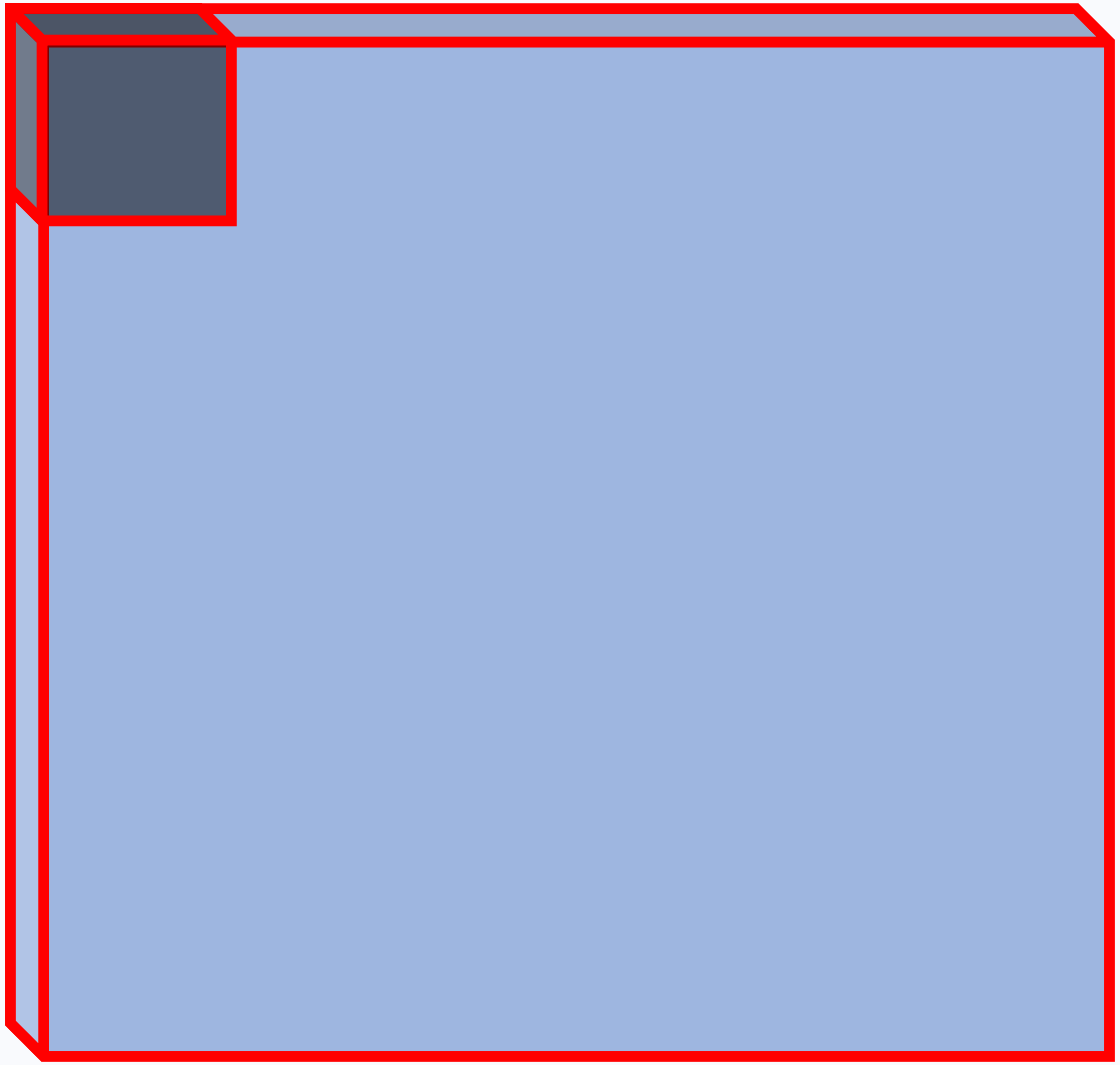
+1 1 − 3 5 2 ∗ 1 − 3 5 6 + 2 ∗ 1 − 3 5 + 1 1 − 3 5 + 1 1 − 3 5 2 ∗ 1 − 3 5

$$= 6 + 2 * 1 - 3\ 5\ 6 + 2 * 1 - 3\ 5\ + 1 \times \left\lfloor \frac{6+2*1-3}{5} + 1 \right\rfloor = 2 \times 2$$

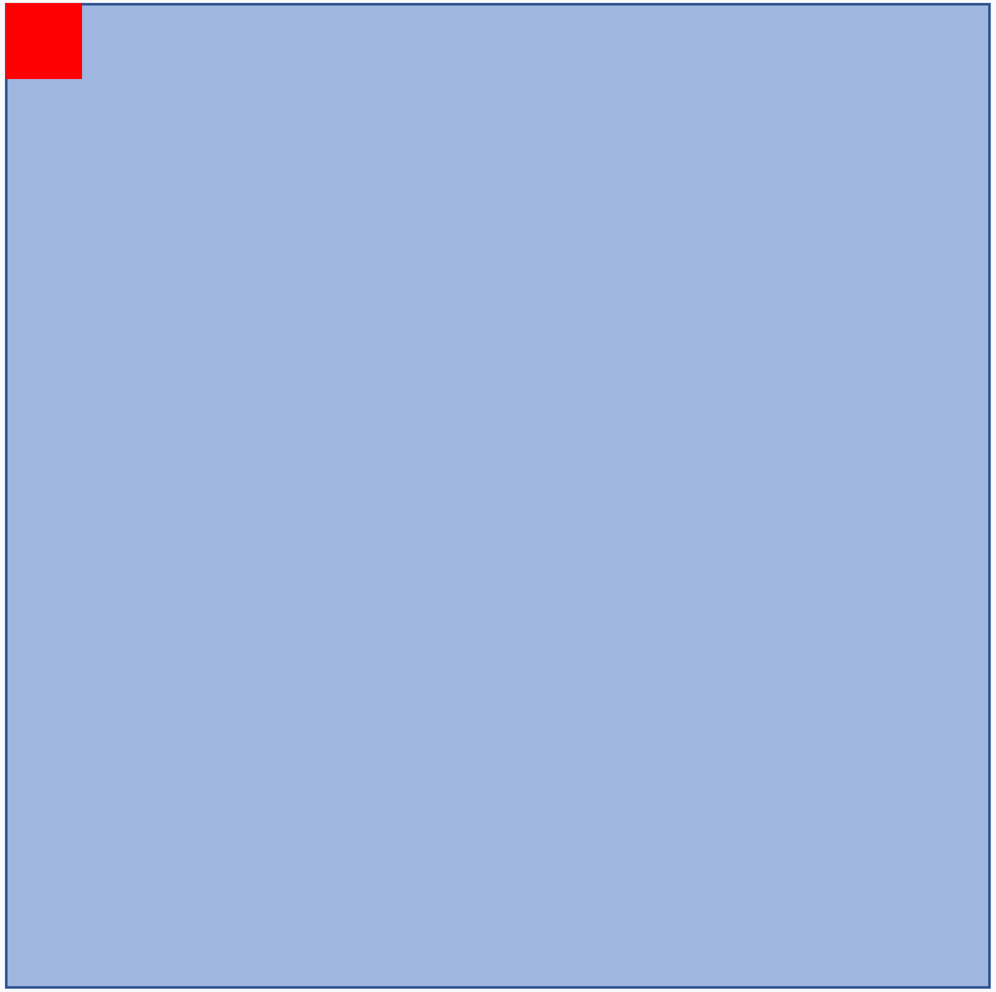

RGB image
n
n_c
n
$*$
filter
f
n_c
f
?


RGB image
n
n_c
n
$*$
filter
f
n_c
f
?

Convolution over volume

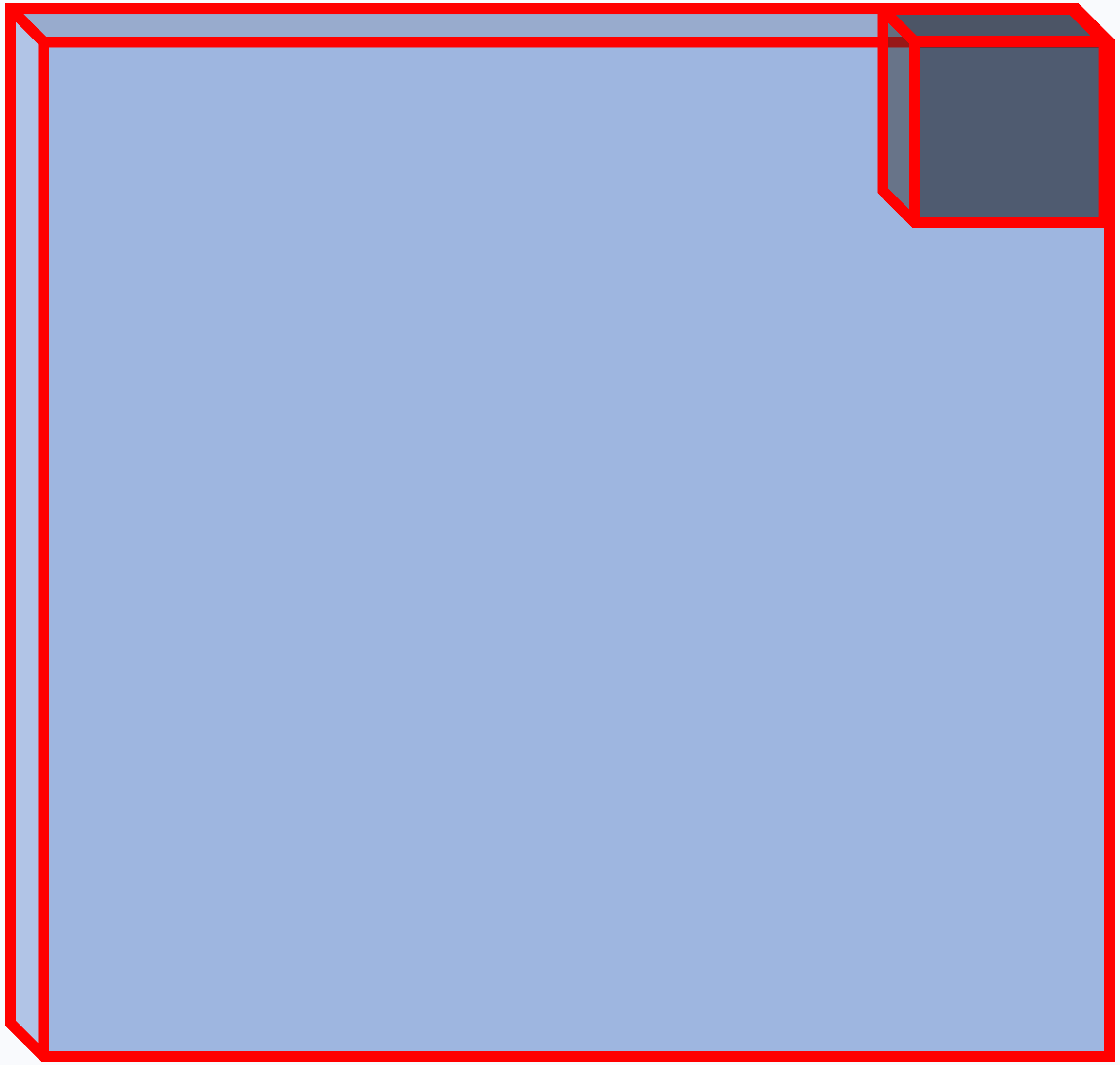


n_c = 3 (RGB)

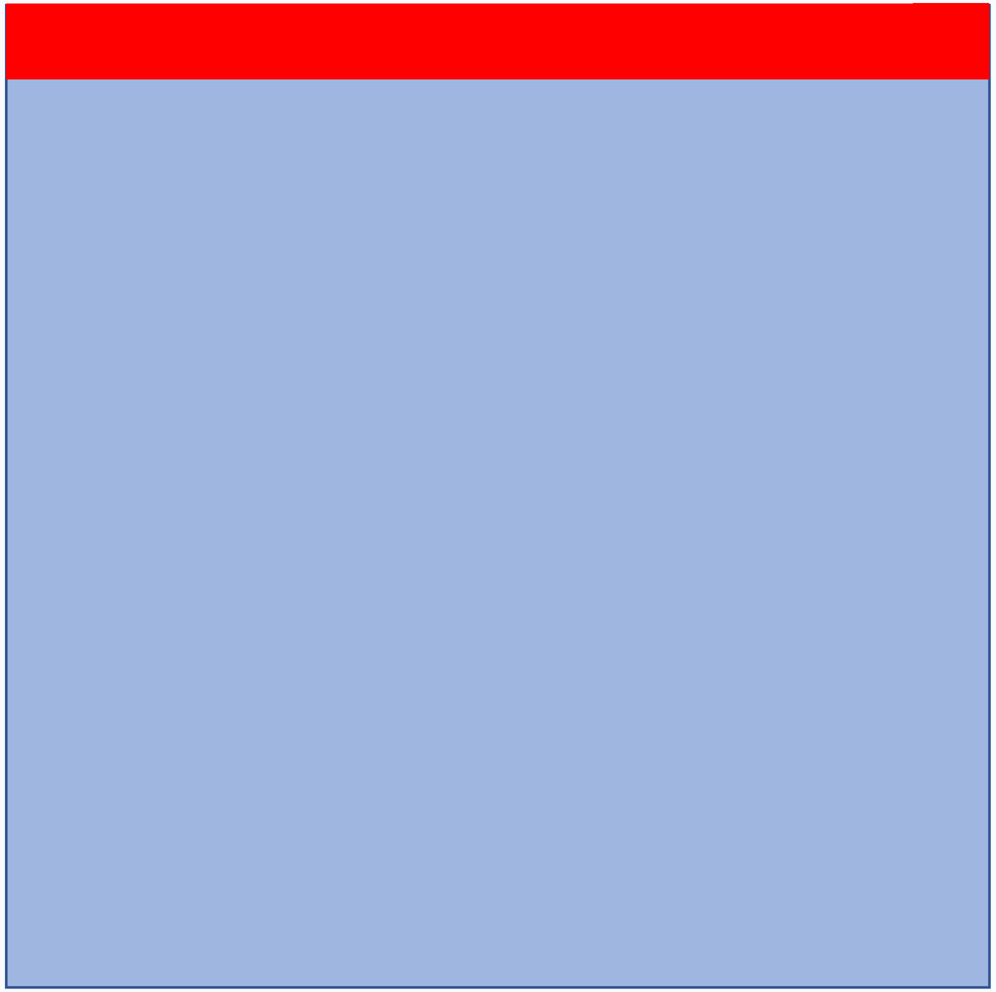


n_c = 1

Convolution over volume

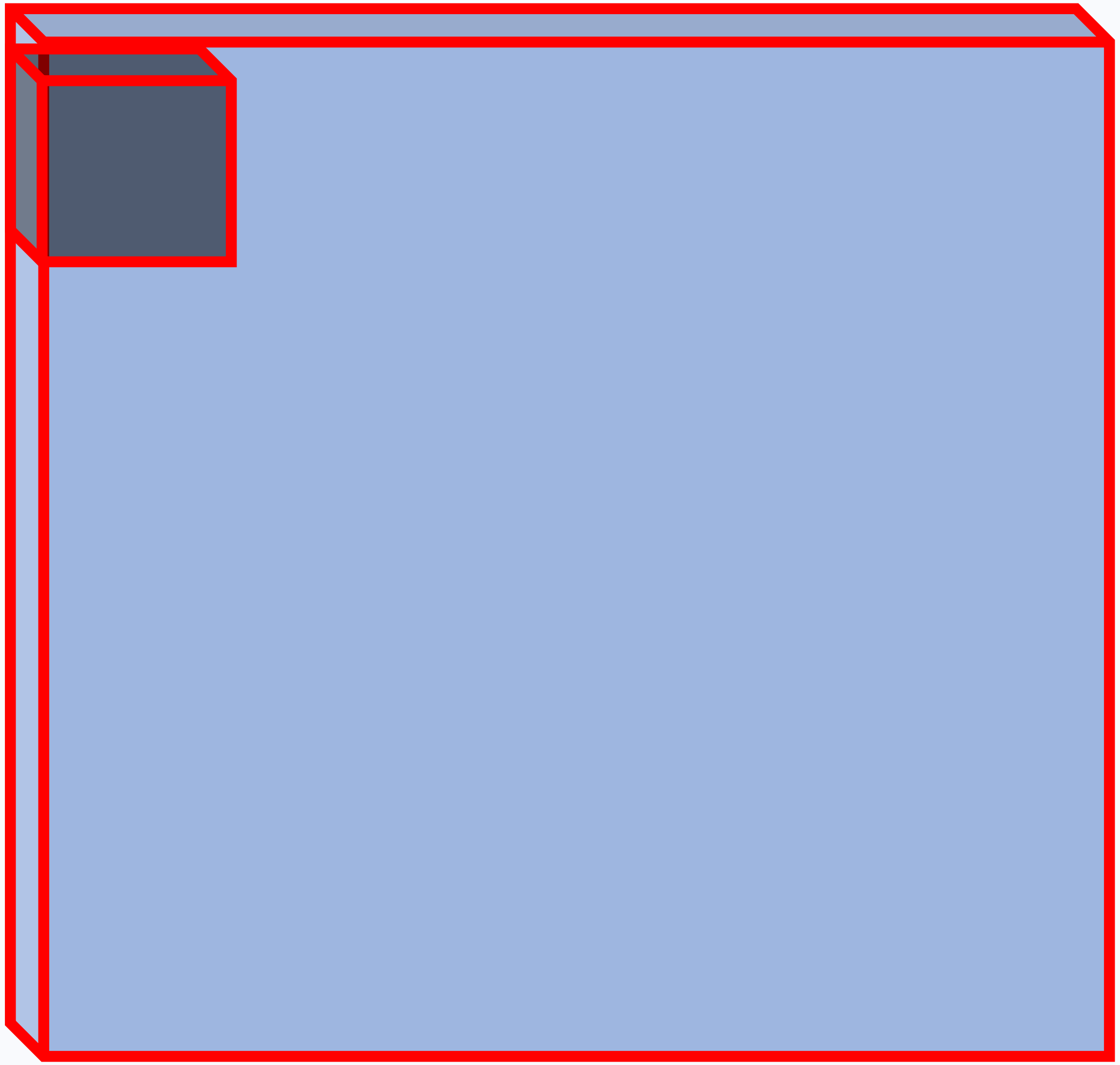


n_c = 3 (RGB)

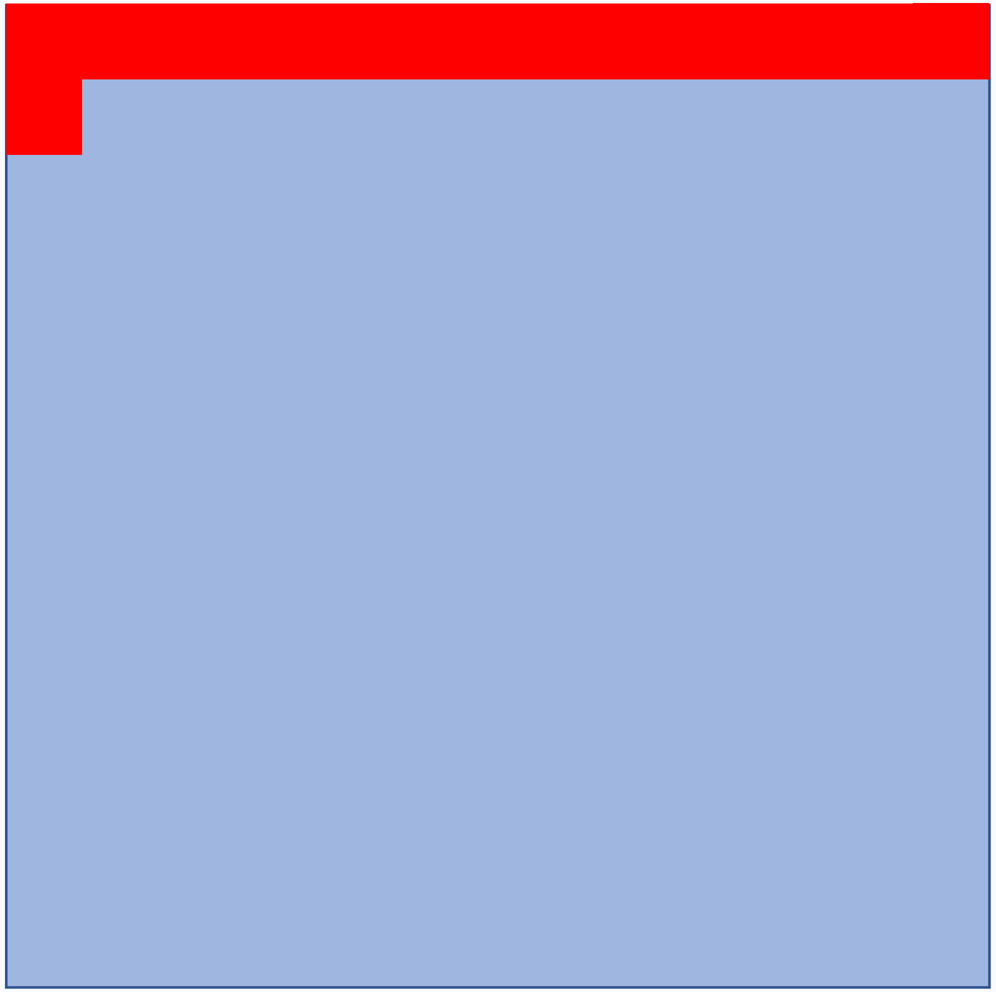


n_c = 1

Convolution over volume

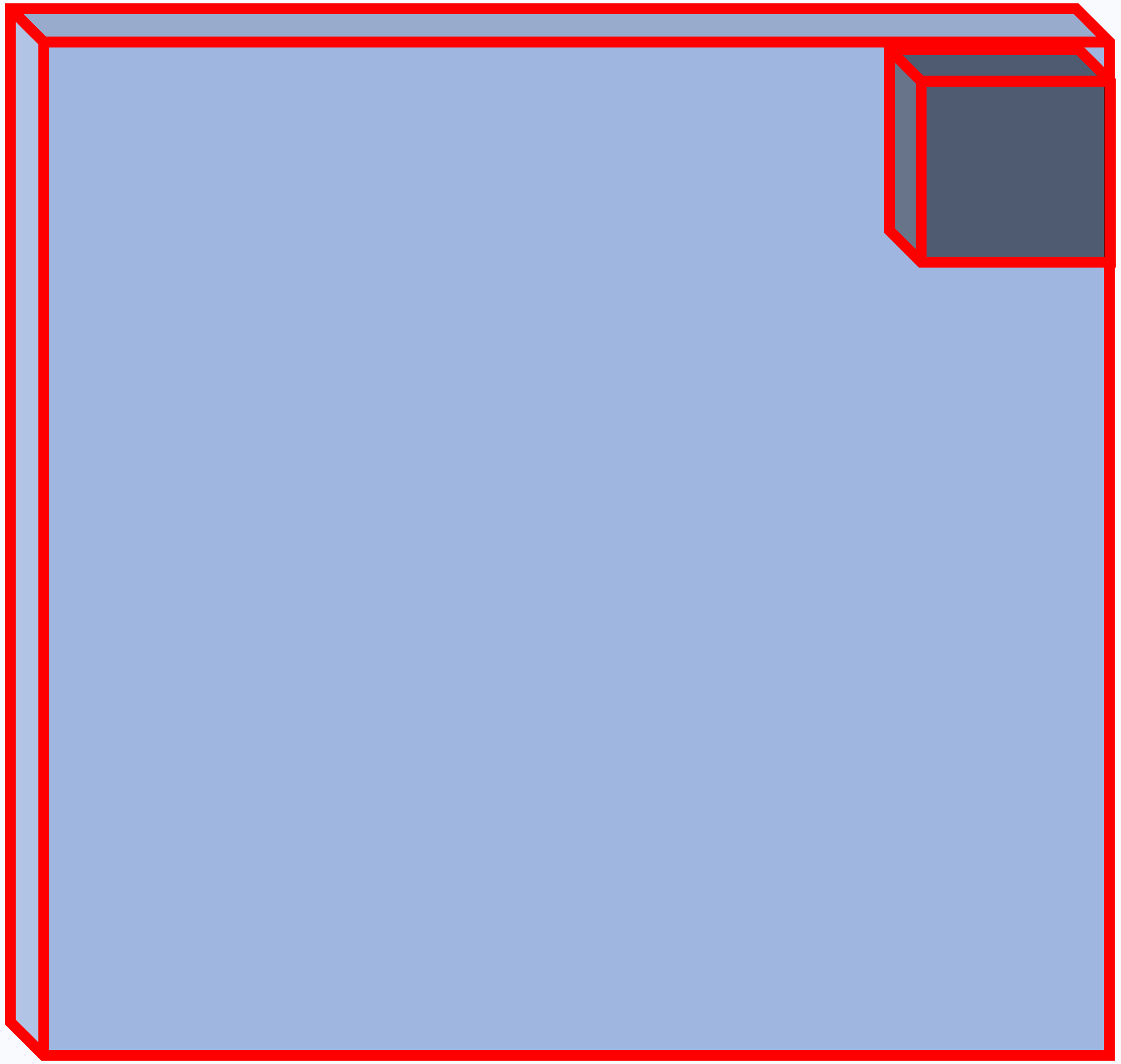


n_c = 3 (RGB)

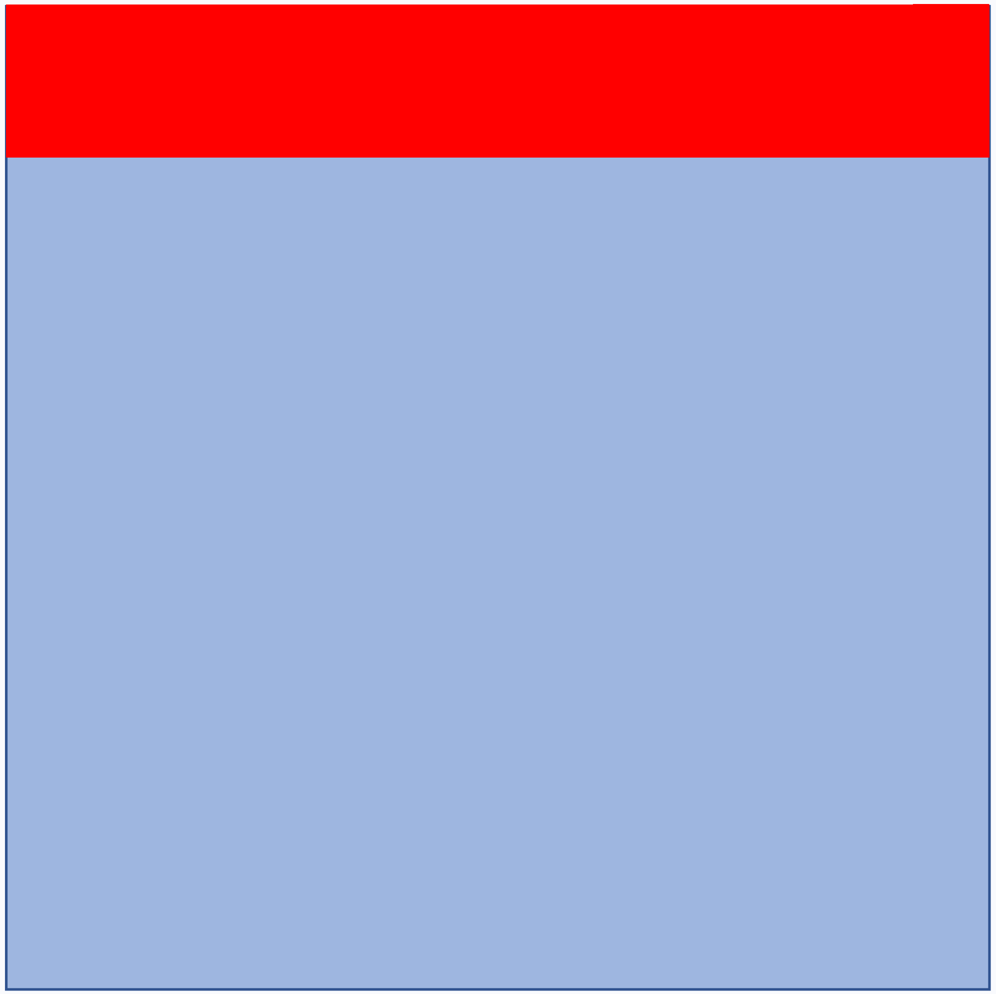


n_c = 1

Convolution over volume

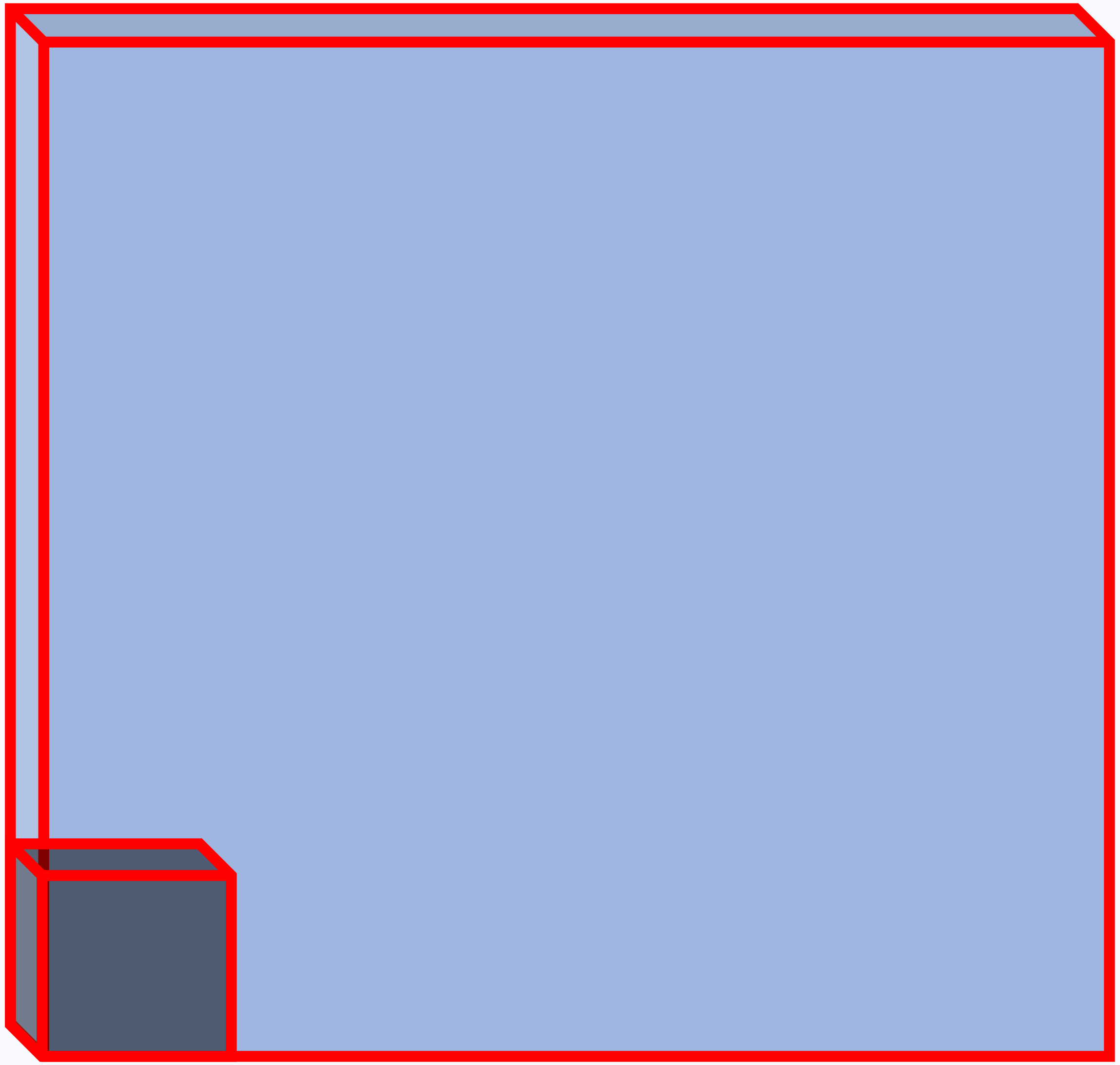


n_c = 3 (RGB)

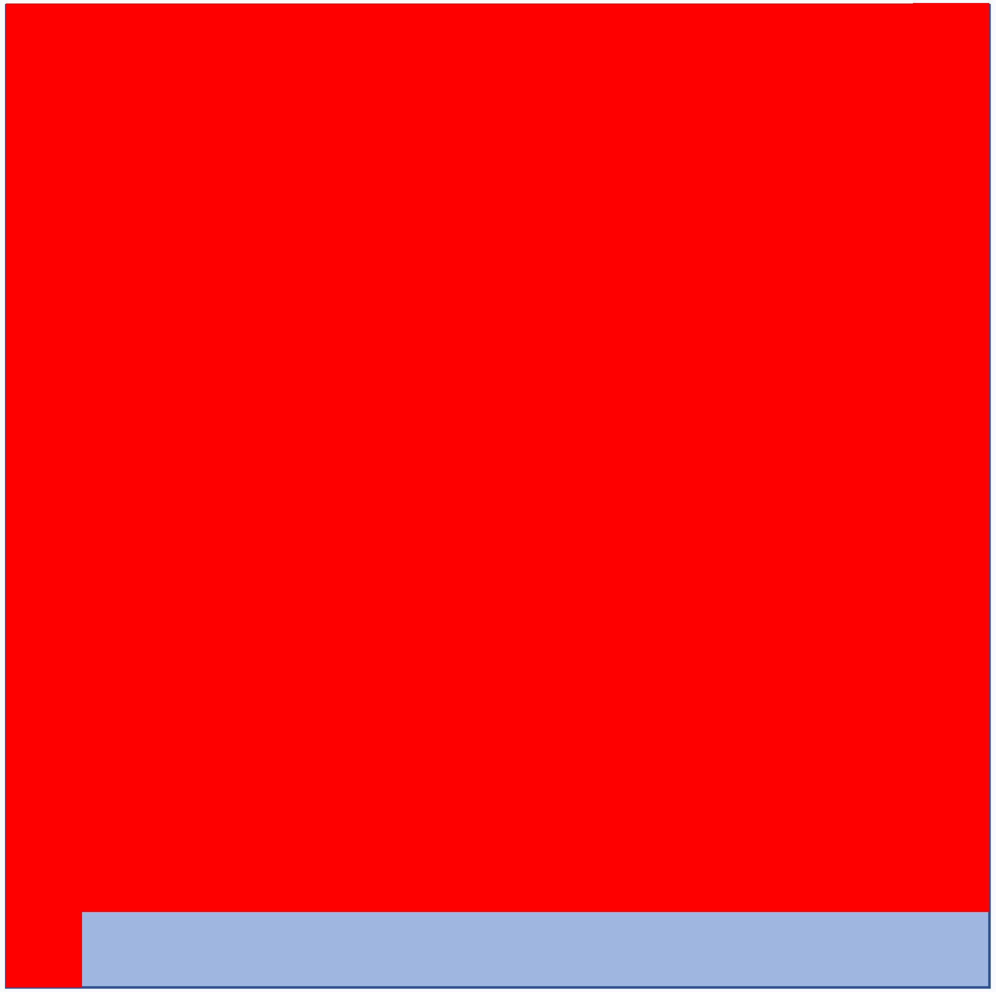


n_c = 1

Convolution over volume

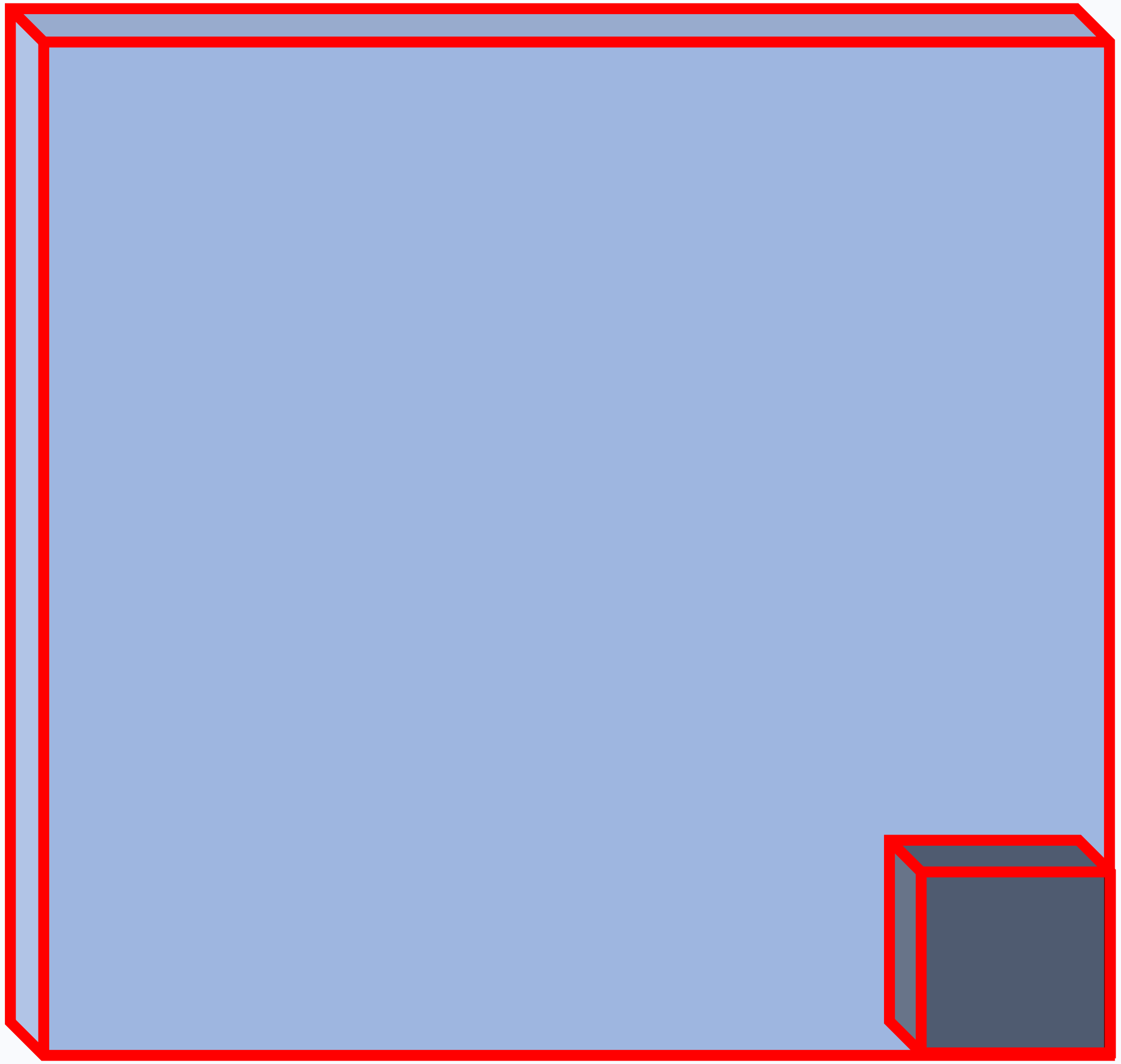


n_c = 3 (RGB)

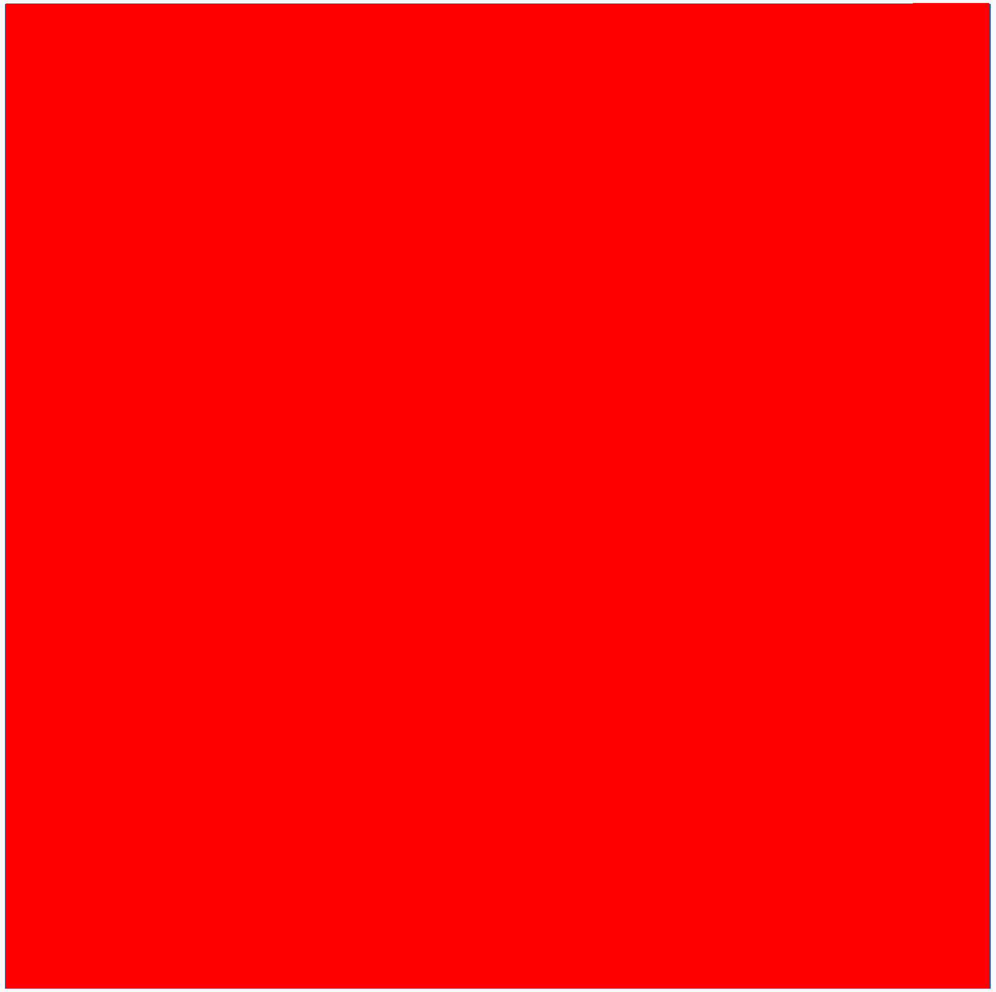


n_c = 1

Convolution over volume

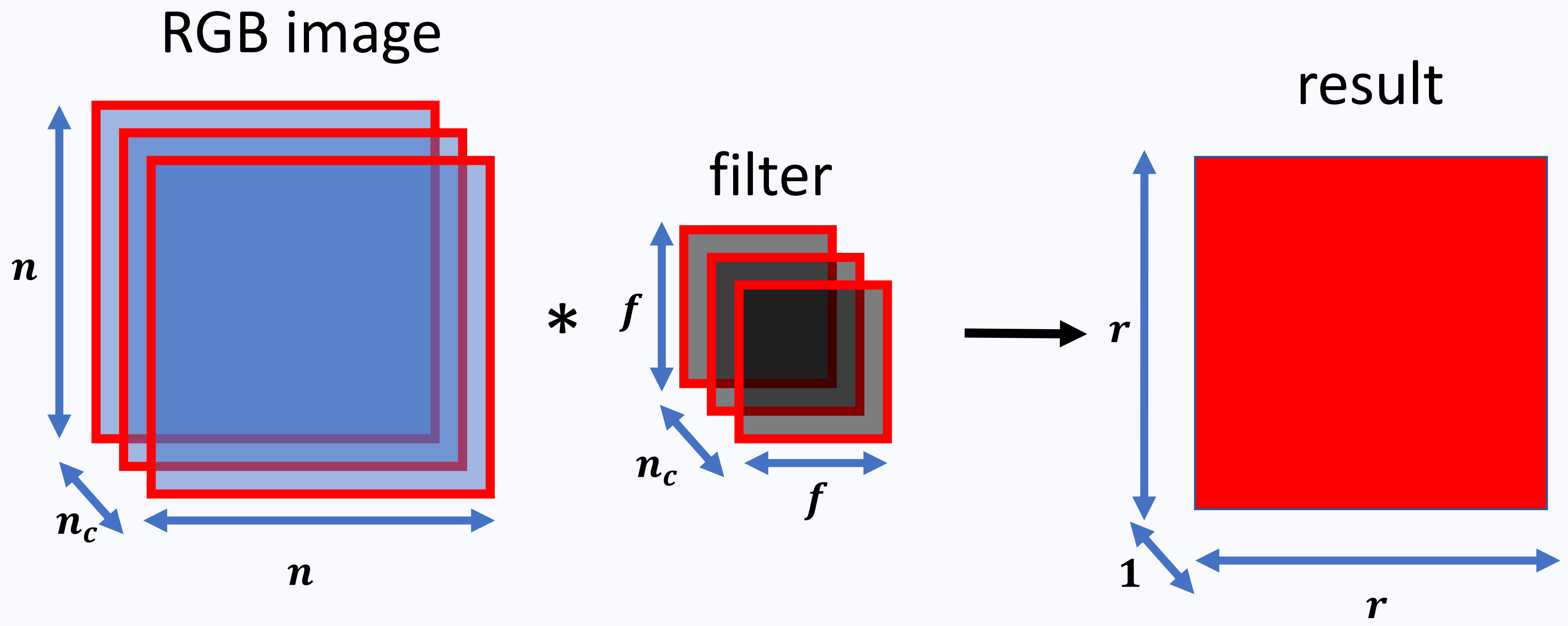


n_c = 3 (RGB)



n_c = 1

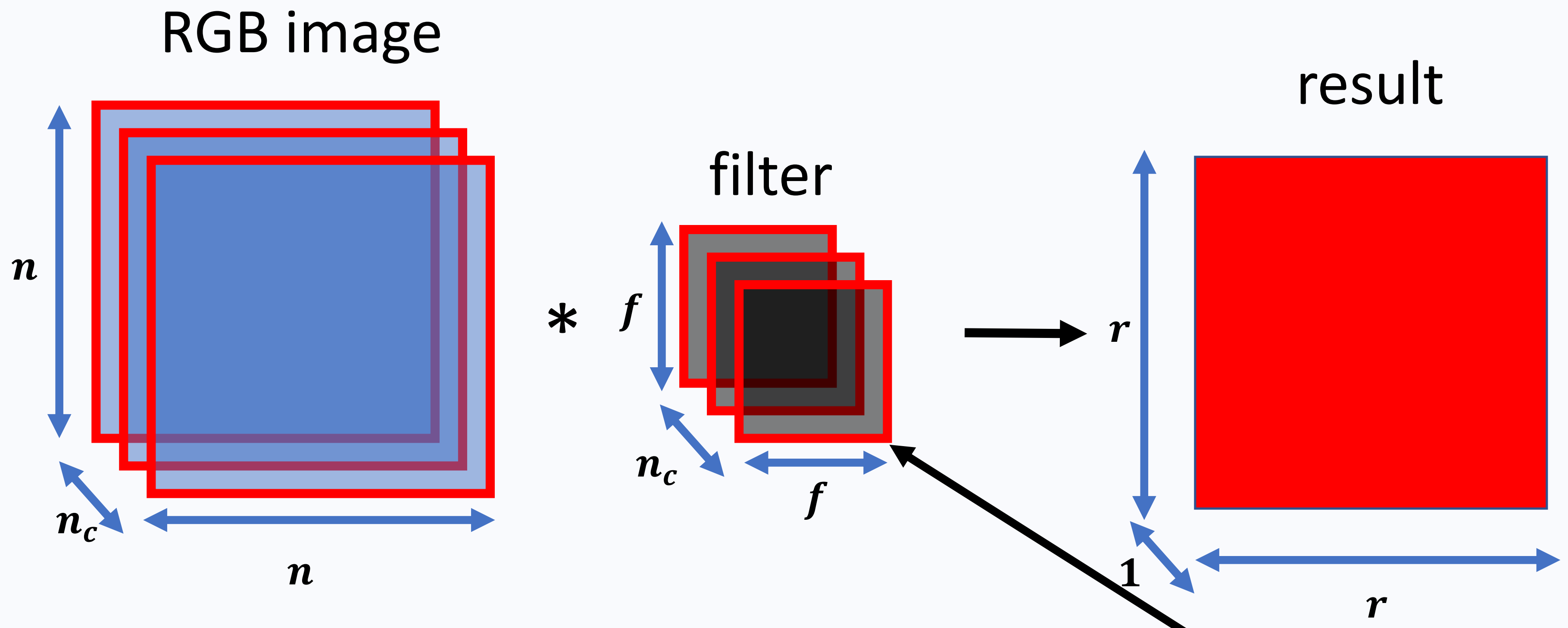
Convolution over volume



$$r = \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

from previous slide

Convolution over volume



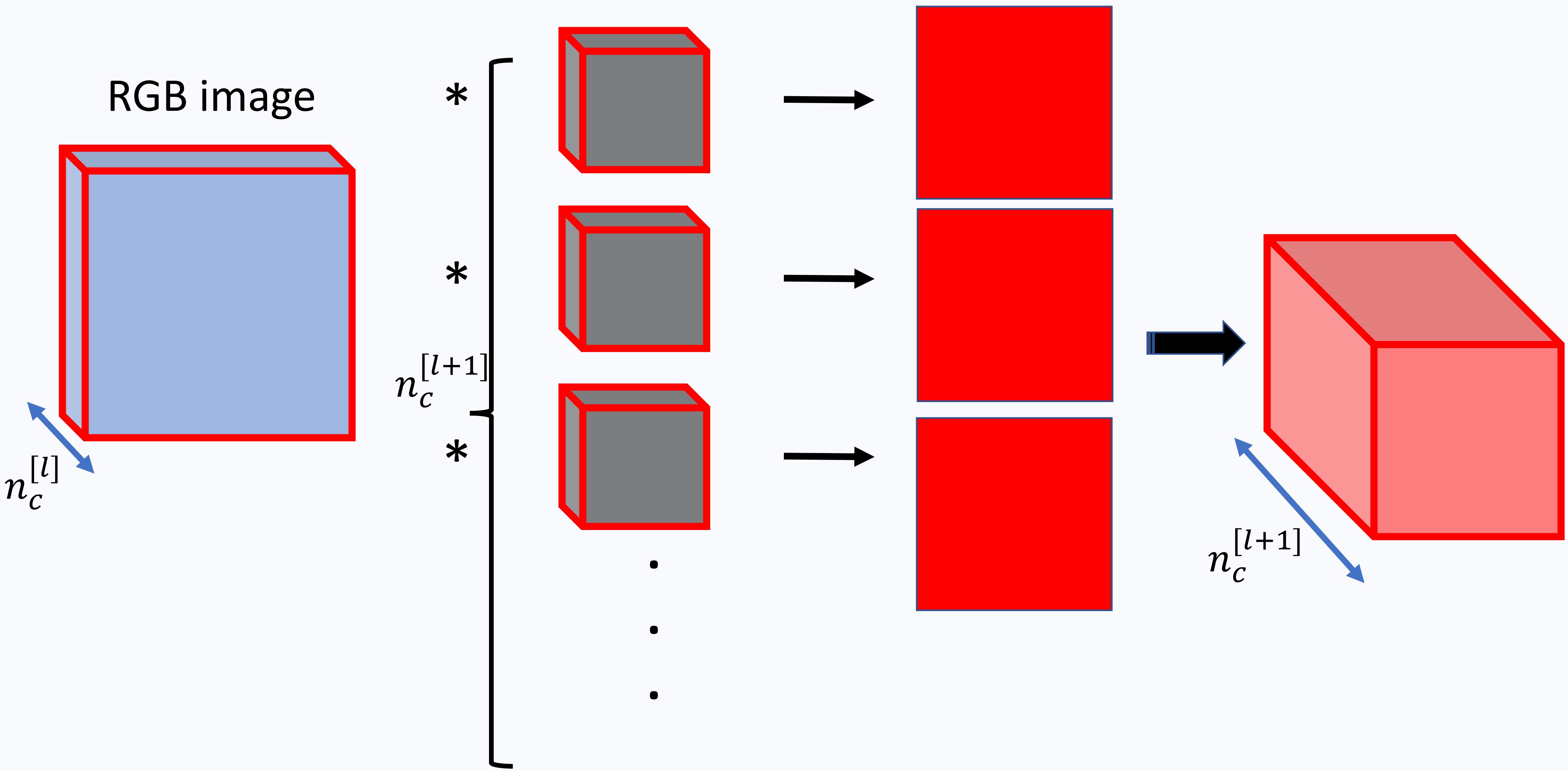
$$r = \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

from previous slide

$f * f * n_c$

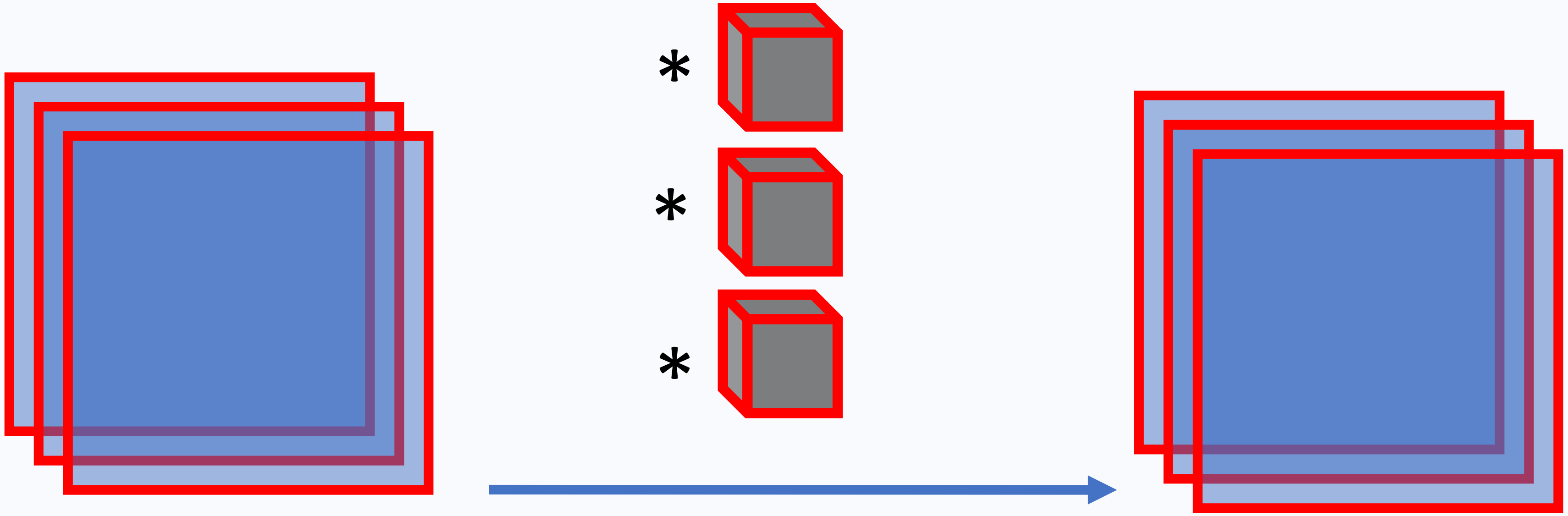
different parameters

Convolution over volume



1 layer of convolution in neural networks

$\#channels\ in\ layer\ l: n_c^{[l]}$



$n_c^{[0]} = 3$

$n_c^{[1]} = 3$

Difference from the previous: 1 bias per filter

Each filter has a parameter number of: $f \cdot f \cdot n_c + 1$

Input: 200 x 200 pixel RGB image, in the first layer we want 200 x 200 x 3 neurons

How many parameters does it have?

Fully connected layer

- weights: $(200 \cdot 200 \cdot 3)^2$
- bias: $200 \cdot 200 \cdot 3$
- $Total \approx 1.4 \cdot 10^{10}$

Input: 200 x 200 pixel RGB image, in the first layer we want 200 x 200 x 3 neurons

How many parameters does it have?

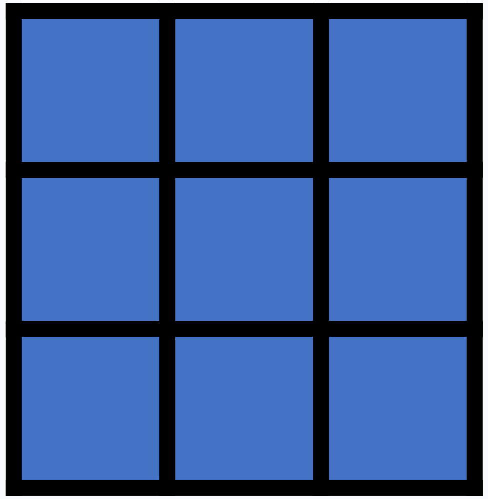
Fully connected layer

- weights: $(200 \cdot 200 \cdot 3)^2$
- bias: $200 \cdot 200 \cdot 3$
- $Total \approx 1.4 \cdot 10^{10}$

Convolutional layer (f = 3)

- Weights per filter: $f \cdot f \cdot n_c^{[0]} + 1$
- Number of filters: $n_c^{[1]}$
- $Total = 84$

'filter'



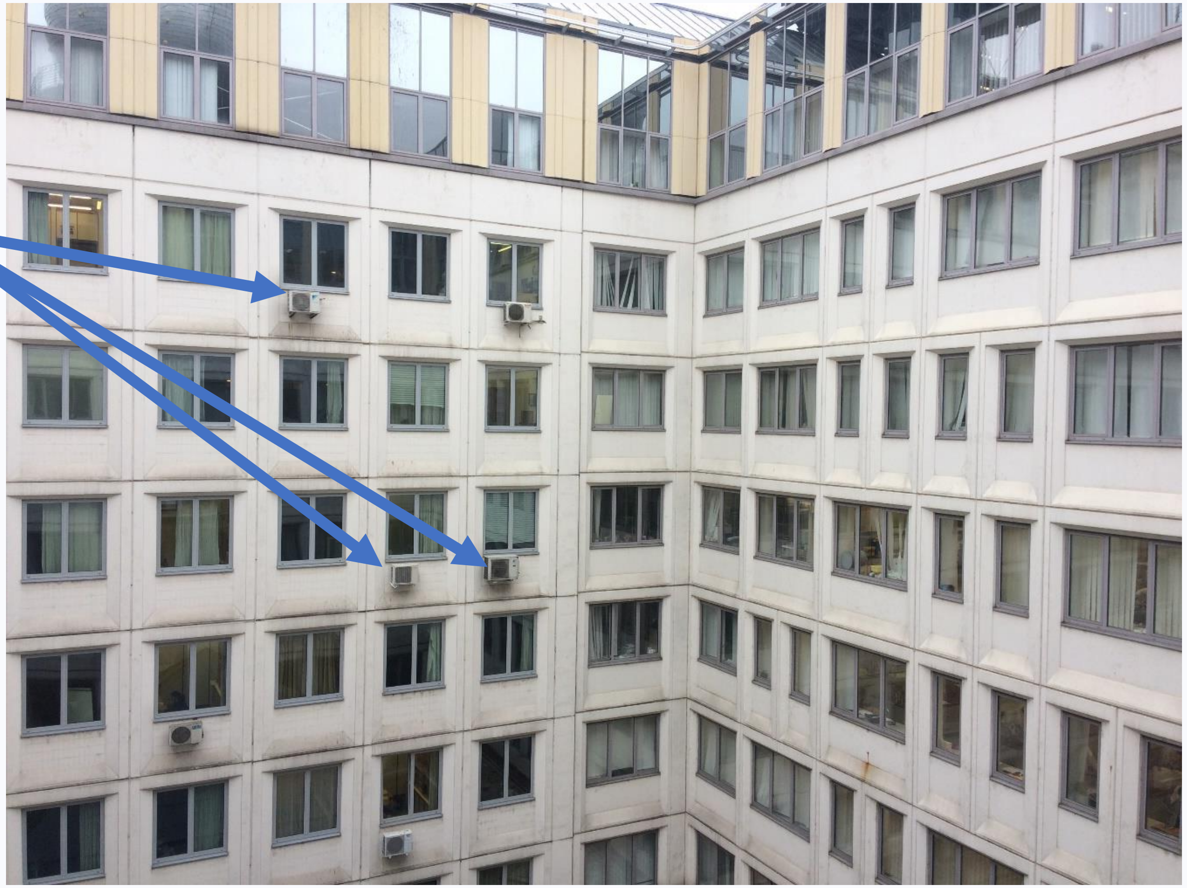
Image

a_{00}	a_{01}	a_{02}	a_{03}	a_{04}	a_{05}
a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
a_{20}	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}
a_{30}	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}
a_{40}	a_{41}	a_{42}	a_{43}	a_{44}	a_{45}
a_{50}	a_{51}	a_{52}	a_{53}	a_{54}	a_{55}

$a'_{11} = \max(a_{00}, a_{01}, a_{02}, a_{10}, a_{11}, a_{12}, a_{20}, a_{21}, a_{22})$

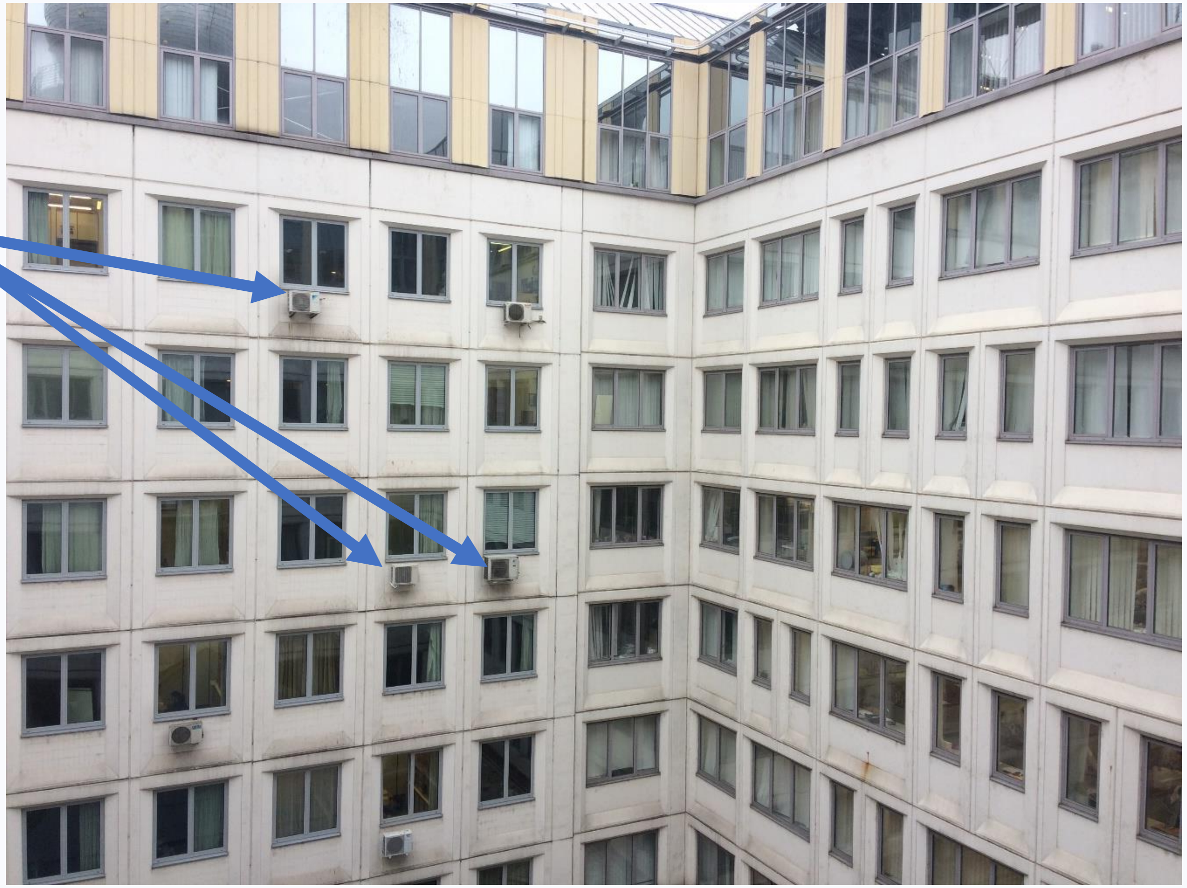
Default stride = pool size (f)

- Objects are not position dependents



- Objects are not position dependents

✓ A convolution filter



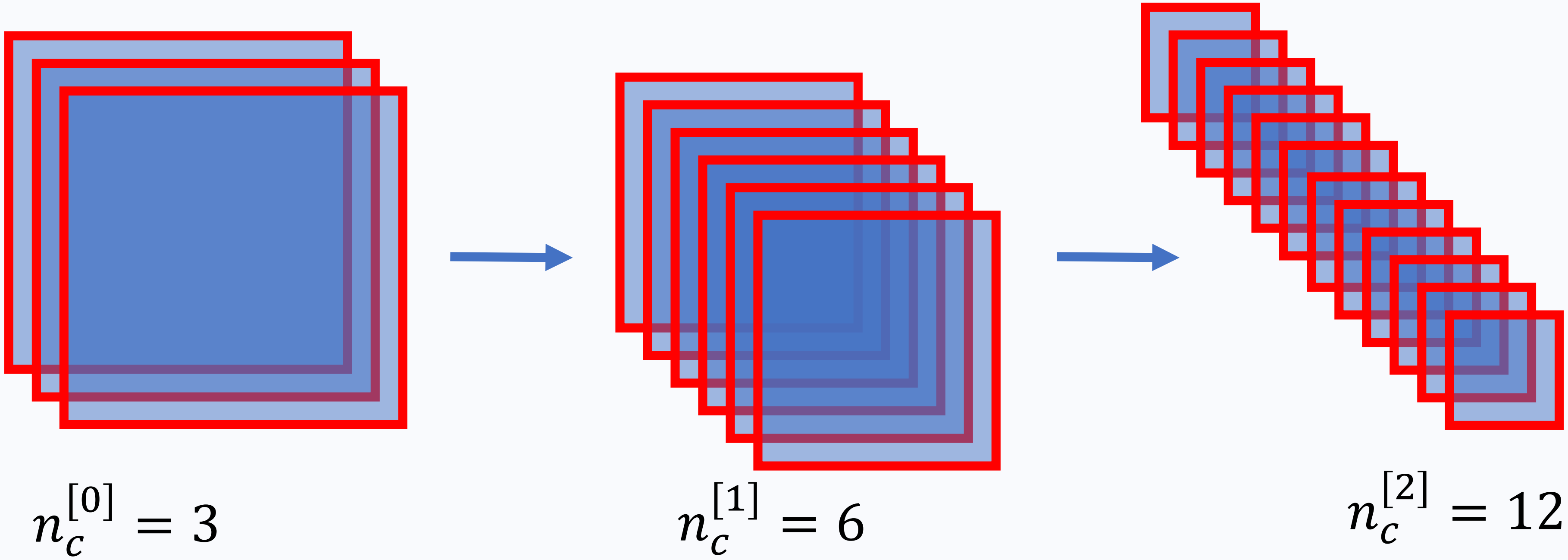
- Objects are not position dependents

✓ A convolution filter

✓ Maxpooling: best value from a region (exact position doesn't matter for image classification)

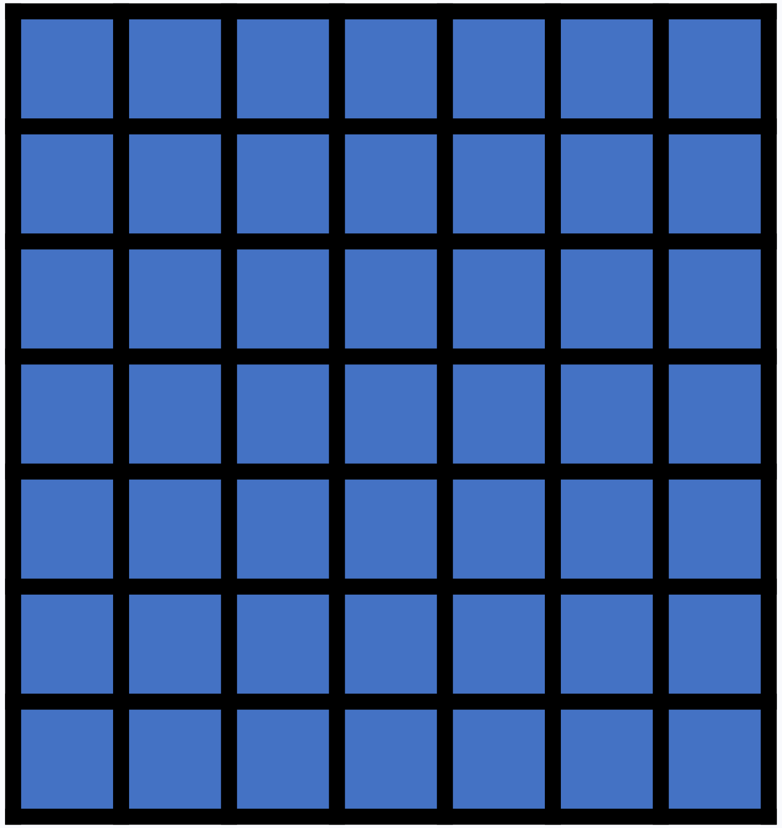
Convolution in neural networks - representation

$\#channels\ in\ layer\ l: n_c^{[l]}$



Three 3 x 3 convolution after each other

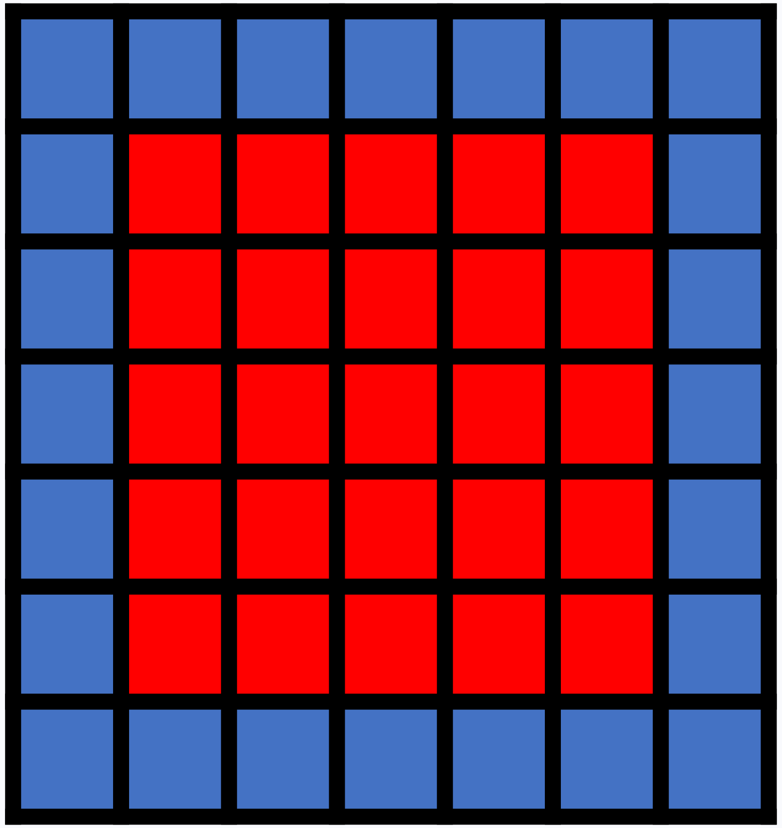
No padding, stride=1, f=3



Three 3 x 3 convolution after each other

No padding, stride=1, f=3

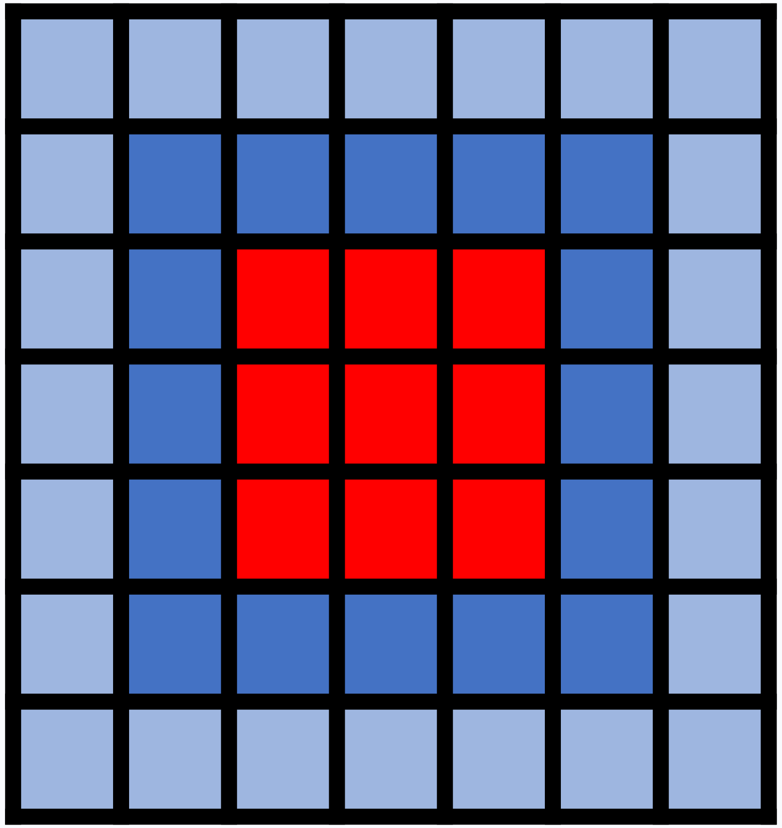
After 1 layer:



Three 3 x 3 convolution after each other

No padding, stride=1, f=3

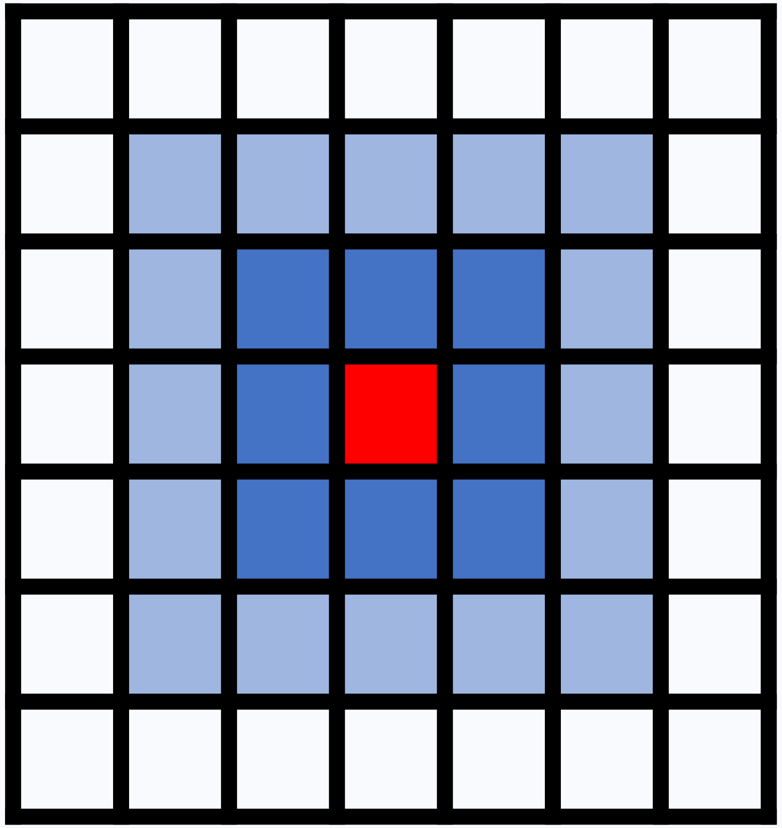
After 2 layer:



Three 3 x 3 convolution after each other

No padding, stride=1, f=3

After 3 layer:



Small **f** vs large **f** – receptive field

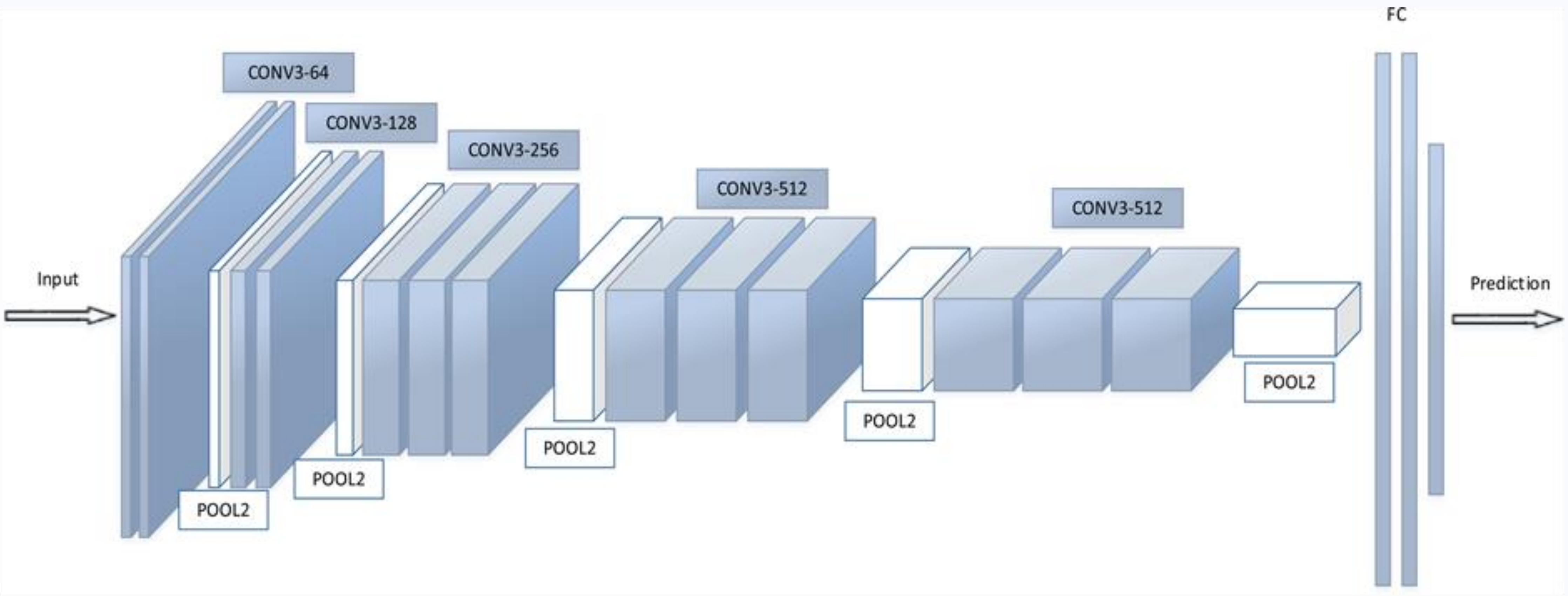
After three 3 x 3 convolution each neuron can see 7 x 7 field from the input

#parameters in three 3 x 3 conv: $3 \cdot (3 \cdot 3 + 1) = 30$

#parameters in one 7 x 7 conv: $1 \cdot (7 \cdot 7 + 1) = 50$

3 convolutions → more 'non-linearity'

VGG16



[http://file.scirp.org/Html/4-7800353_65406.htm]

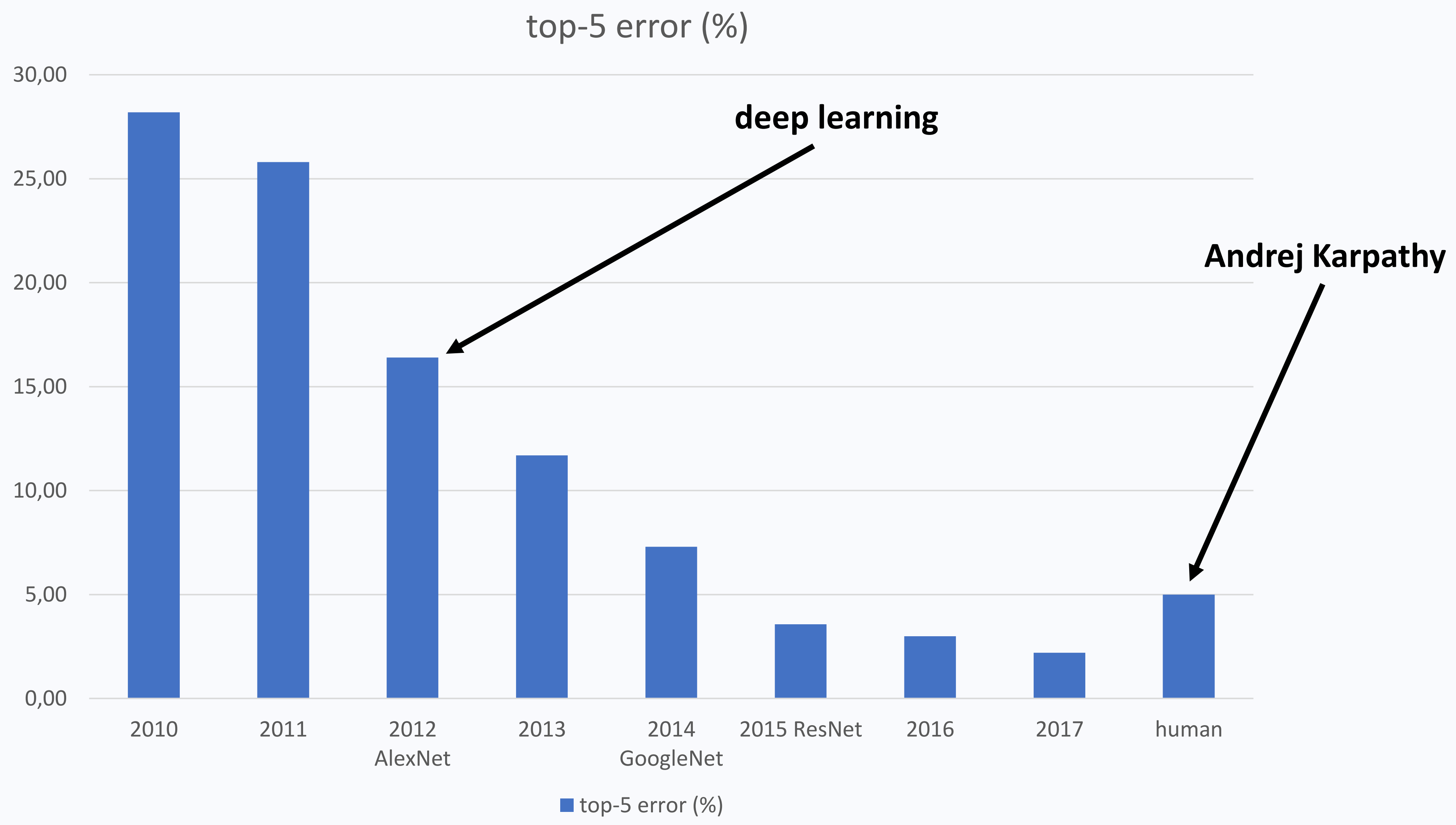
Keras notebook

https://github.com/qati/DeepLearningCourse/tree/master/demo_notebooks/lecture_05

ImageNet Large Scale Visual Recognition Challange

- 2010 –
- 1.2M images (100K test set)
- 1000 categories
- 'Image classification world cup',
- top-5 error (still not that easy…)

ImageNet Large Scale Visual Recognition Challange



What about you?

https://cs.stanford.edu/people/karpathy/ilsvrc/