Technische Hochschule Rosenheim

Applications of & Introduction to Artificial Intelligence

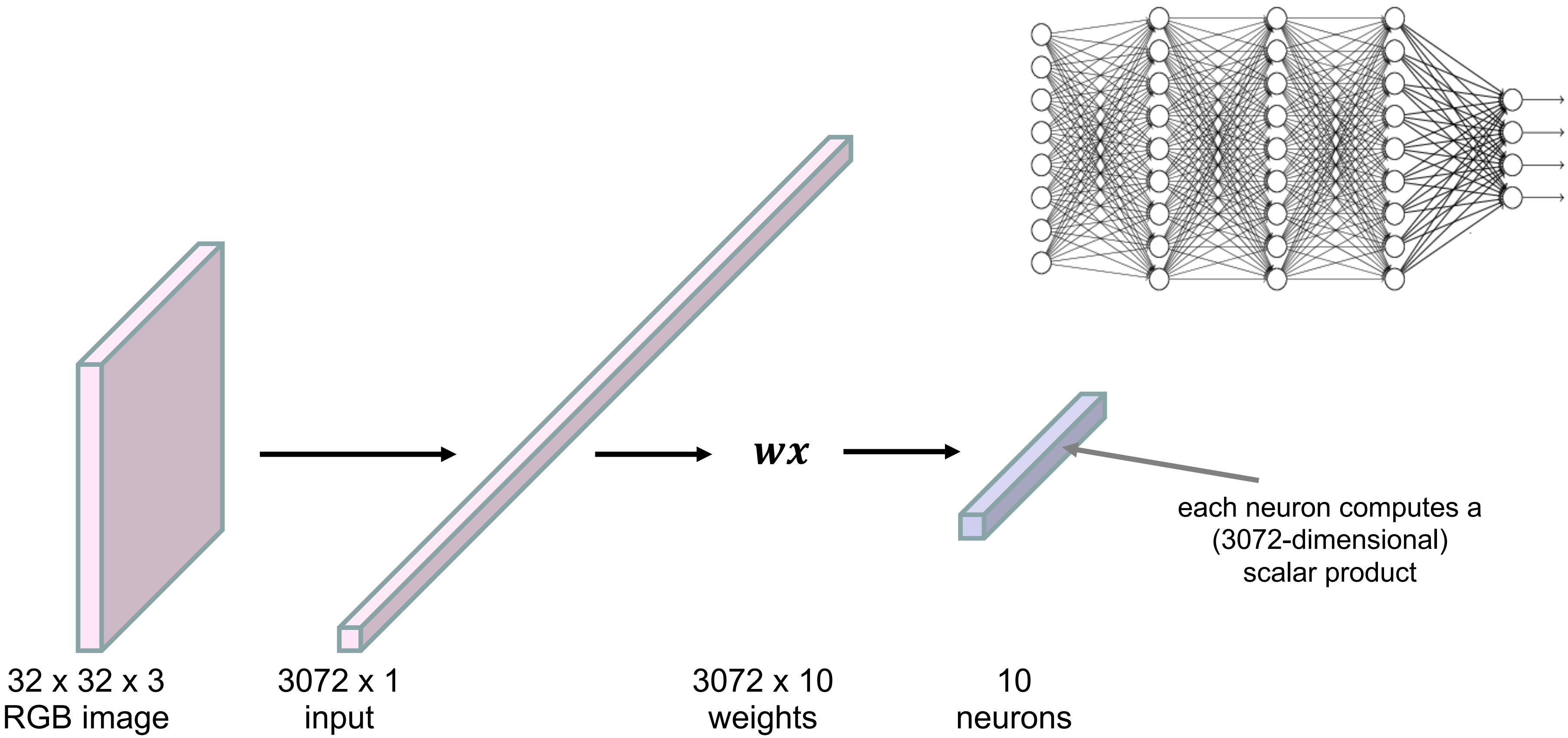
Convolutional Neural Networks

Technische Hochschule Rosenheim

Sommer 2020

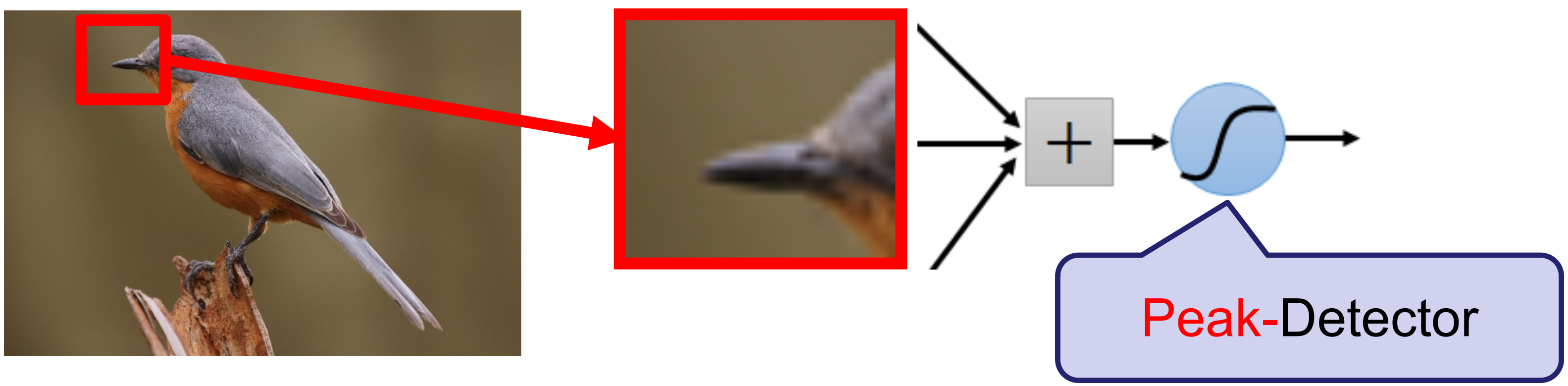
Prof. Dr. J. Schmidt

Example: Typical MLP-Structure for Images



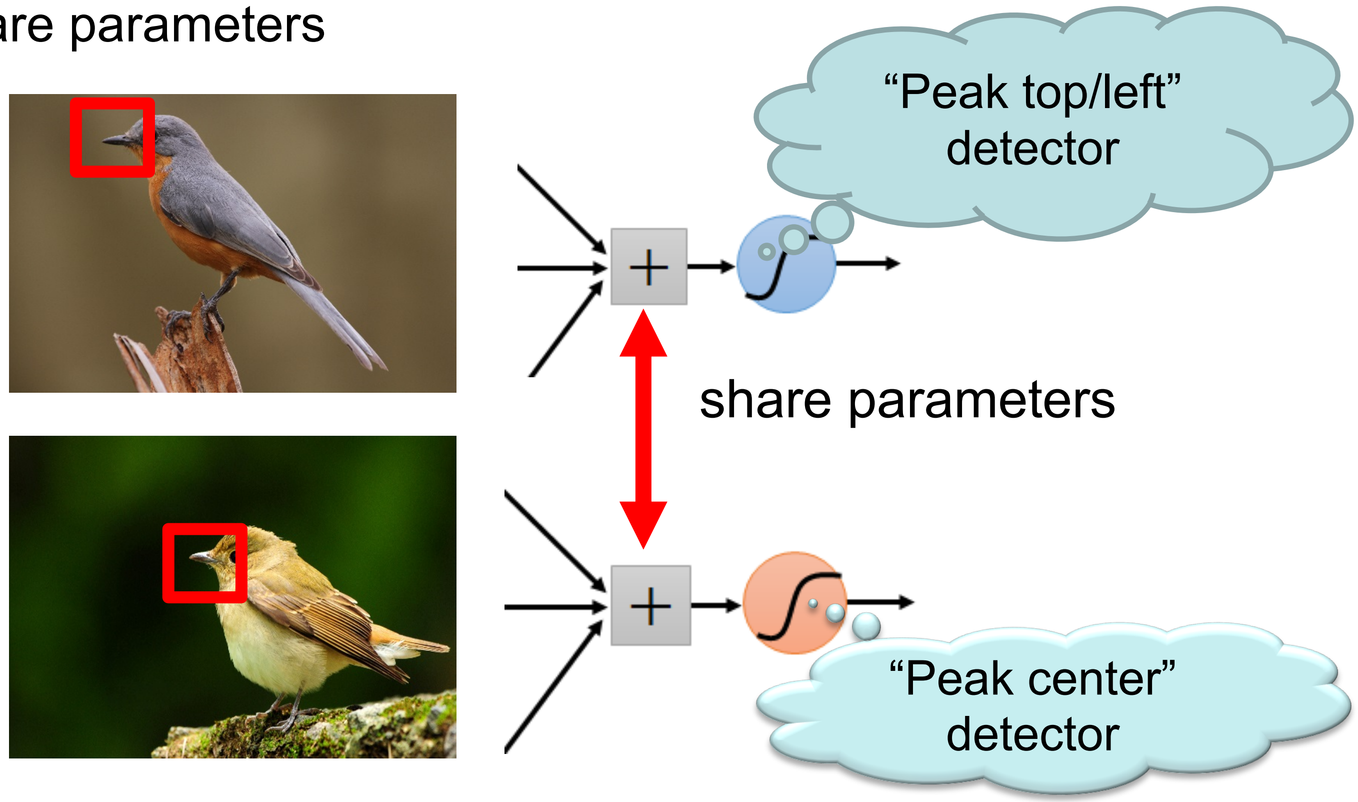
Recognition of Image Parts

- many patterns are smaller than the complete image
- for small regions: less parameters required



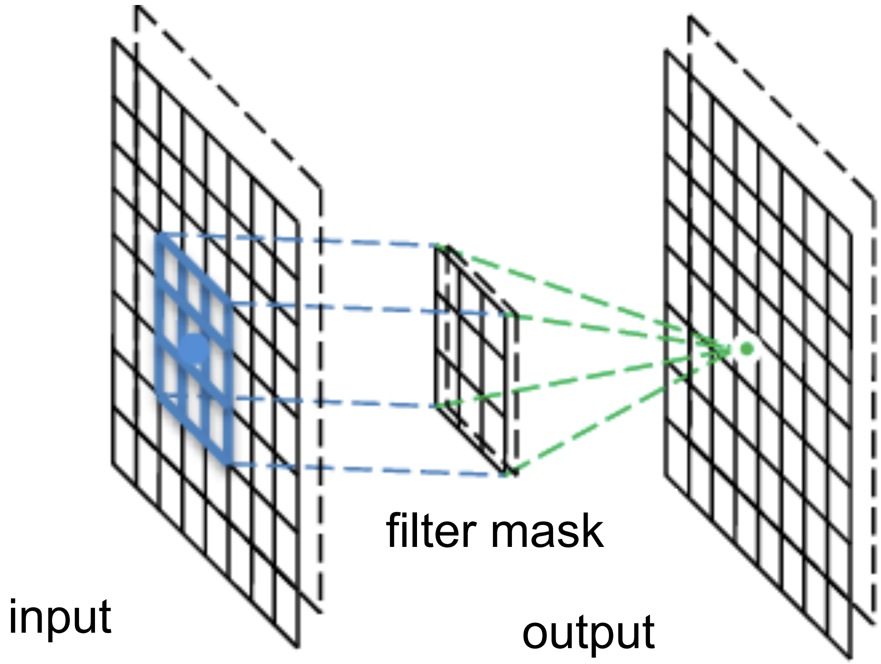
Similar Patterns

- similar patterns can be found in different image locations
- Idea: Train many small detectors that
  - move over the image
  - share parameters

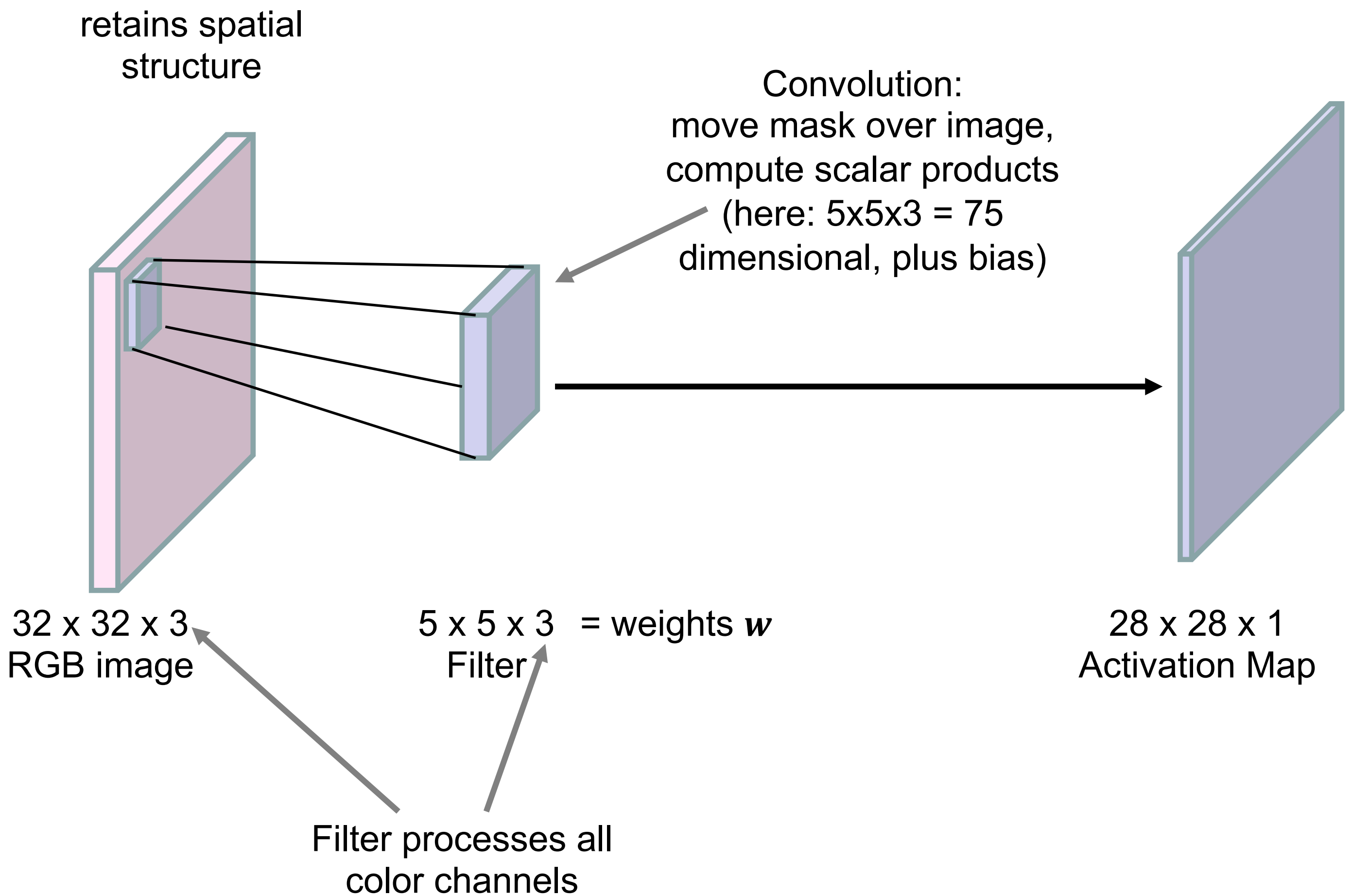


Convolutional Layer

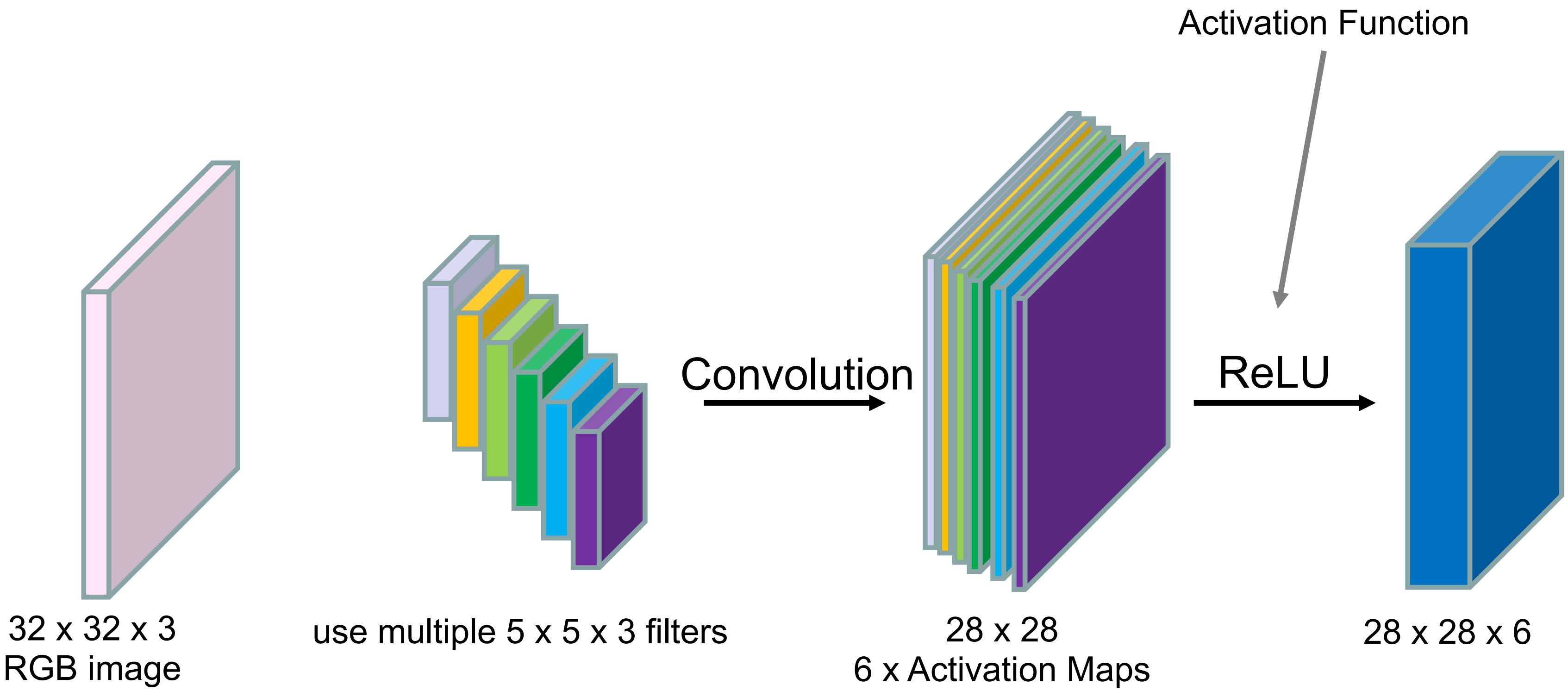
- hence: CNN – Convolutional Neuronal Network
- consists of (linear) convolution filters
- the filter masks are learned during training
- first used with backpropagation in LeNet (1989-1998):
  LeCun, Bottou, Bengio, Haffner. Gradient-Based Learning Applied to Document Recognition. Proc. of the IEEE 86(11): 2278-2324, 1998.



Convolutional Layer

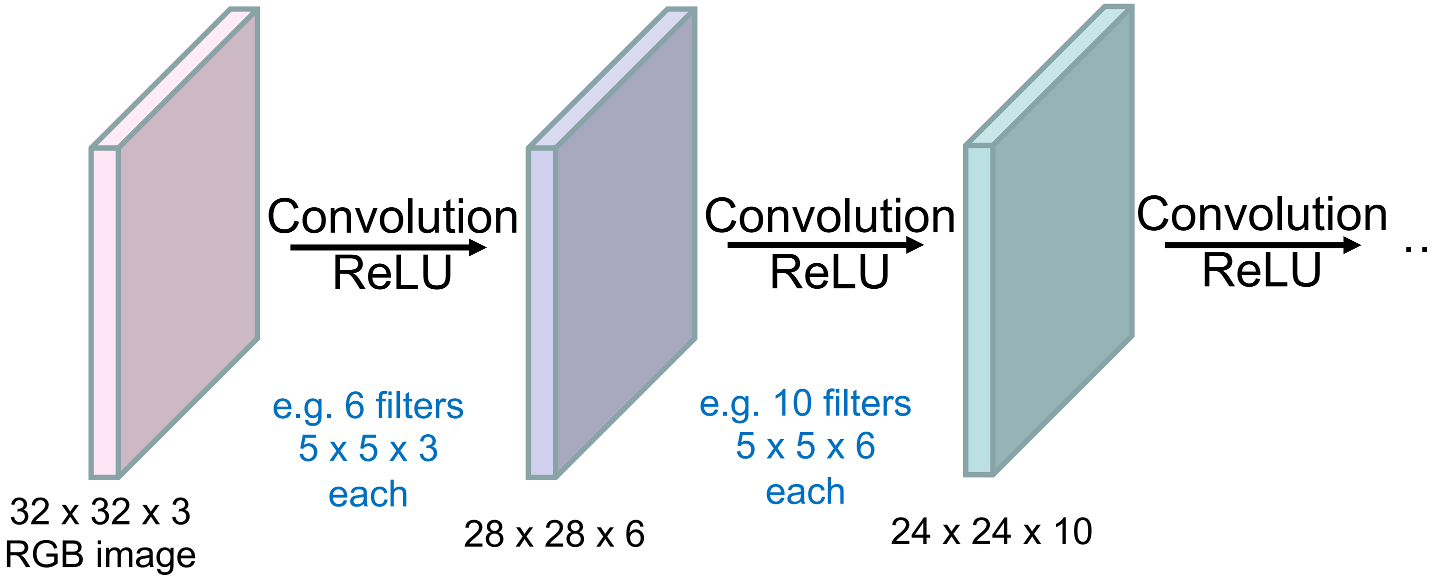


Convolutional Layer



Convolutional Layer

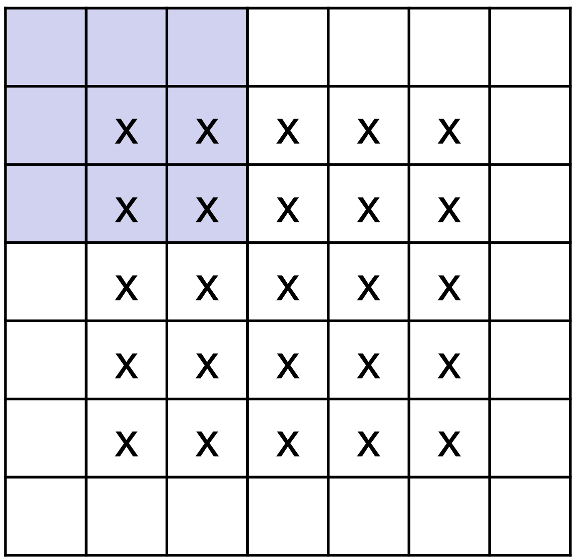
Convolution and activation are now repeated several times
Idea: Combine low-level features, combine again etc.



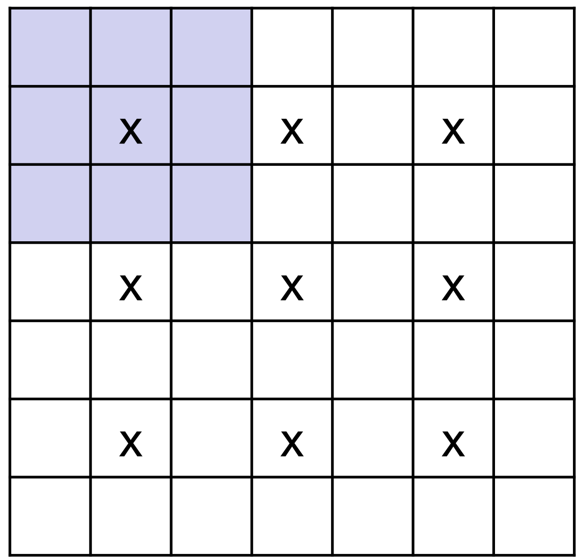
Hyperparameter – Stride

the filter mask can be moved by more than one pixel (stride)
this differs from the “normal“ convolution operation

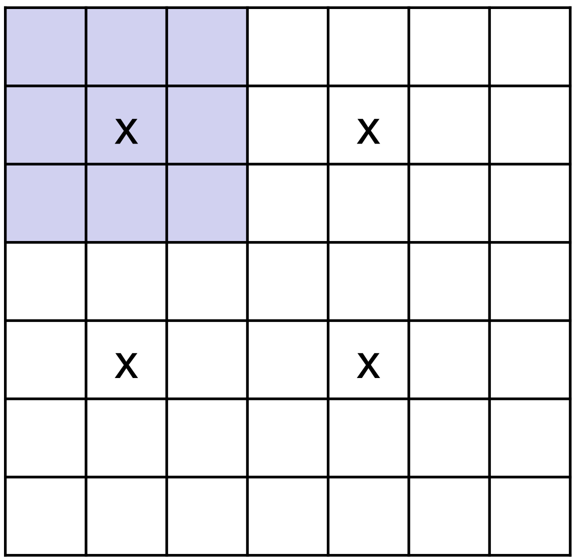
Example: 7x7 image with 3x3 filter



Stride 1
Output: 5x5

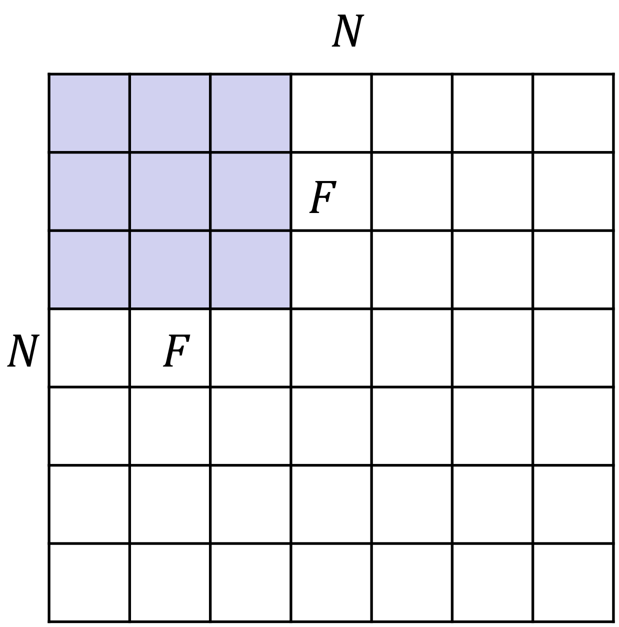


Stride 2
Output : 3x3



Stride 3
asymmetric border –
stride does not match

Hyperparameter – Stride



Stride S

Size of output: $\frac{N-F}{S}+1$

If result is integer:
Stride and filter size match

Example $N=7, F=3$:

$S=1$: $\frac{7-3}{1}+1=5$

$S=2$: $\frac{7-3}{2}+1=3$

$S=3$: $\frac{7-3}{3}+1=2{,}33$

Hyperparameter – Pad

- Problem: Input size for a layer is getting smaller and smaller
- Solution: Padding of border
  - with zeros (Zero-Padding)
  - with copies of the border pixels

0	0	0	0	0	0	0	0	0
0								0
0								0
0								0
0								0
0								0
0								0
0								0
0	0	0	0	0	0	0	0	0

For filter size $F \times F$

$\frac{F-1}{2}$ values are lost at the border

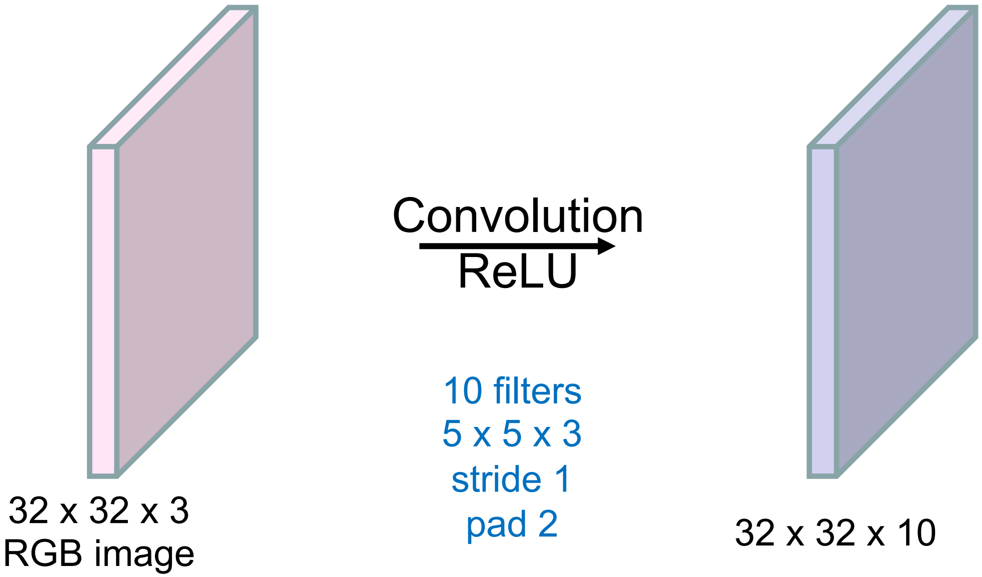
Examples:

$F = 3$: Padding with 1

$F = 5$: Padding with 2

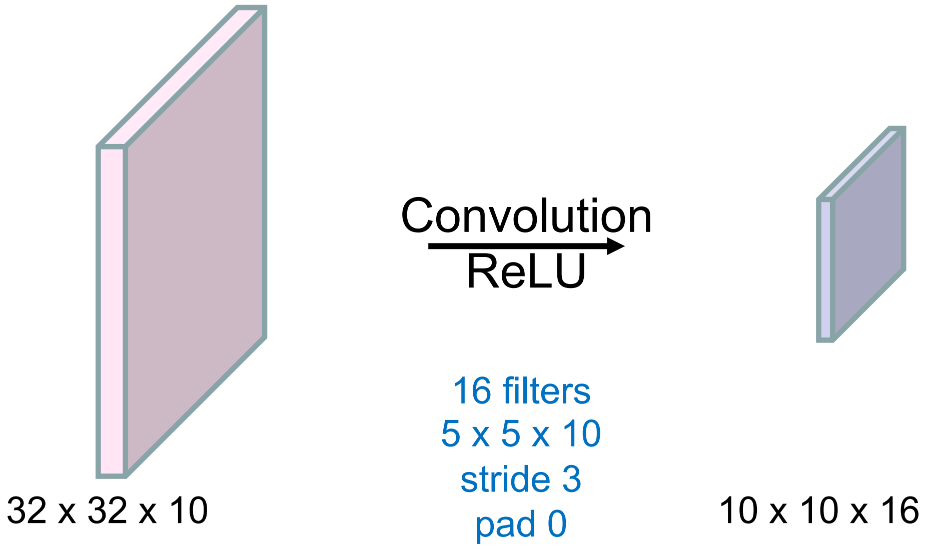
$F = 7$: Padding with 3

Example



Number of parameters for this layer:
each filter has $5 \cdot 5 \cdot 3 + 1 = 76$ parameters (+1 because of bias)
10 filters, total: $76 \cdot 10 = 760$ parameters

Example



Number of parameters for this layer:
each filter has $5 \cdot 5 \cdot 10 + 1 = 251$ parameters (+1 because of bias)
16 filters, total: $251 \cdot 16 = 4016$ parameters

Hyperparameters – Convolution

- Number K and size F of filters
- Stride S
- Size of padding P
- typical values:
  - K = power of 2, e.g. 32, 64, 128, 512
  - $F = 3, \;\; S = 1, P = 1$
  - $F = 5, \;\; S = 1, P = 2$
  - $F = 5, \;\; S = 2, P =$ matching
  - $F = 1, \;\; S = 1, P = 0$
- transforms a layer of size $W{\times}H{\times}D$ into a layer of size $W'{\times}H'{\times}D'$:

$$W' = \frac{W-F+2P}{S} + 1, \;\; H' = \frac{H-F+2P}{S} + 1, \; D' = K$$

- Number of weights: $(F \cdot F \cdot D) \cdot K + K$

Pooling

- scaling does not change the object
- objective: smaller-sized layers

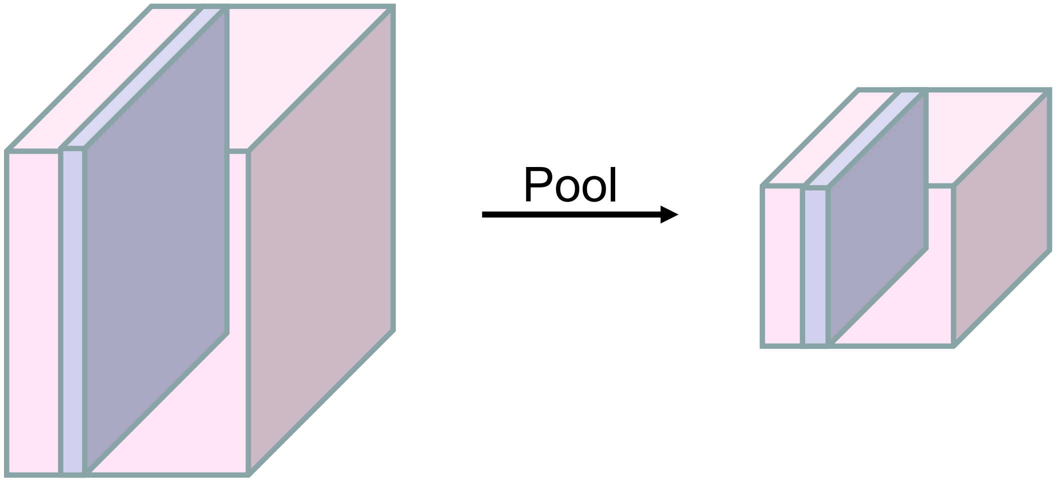
Bird

Scaling
(subsampling)

Bird

Pooling

each activation map is processed separately



200 x 200 x 16

100 x 100 x 16

Pooling

MAX-Pooling: Use the largest element within a windows of size $F \times F$

Average-Pooling: Use the mean value of all elements within a windows of size

Example: MAX-Pooling using 2x2 windows and stride $S = 2$

$$\begin{bmatrix} 2 & 1 & 2 & 4 \\ 7 & 3 & 1 & 5 \\ 6 & 7 & 1 & 8 \\ 9 & 3 & 4 & 2 \end{bmatrix} \xrightarrow{\text{Pool}} \begin{bmatrix} 7 & 5 \\ 9 & 8 \end{bmatrix}$$

Hyperparameters – Pooling

Technische Hochschule **Rosenheim**

- Size F of windows
- Stride S
- Typical values:
  - $F = 2, \quad S = 2$
  - $F = 3, \ S = 2$

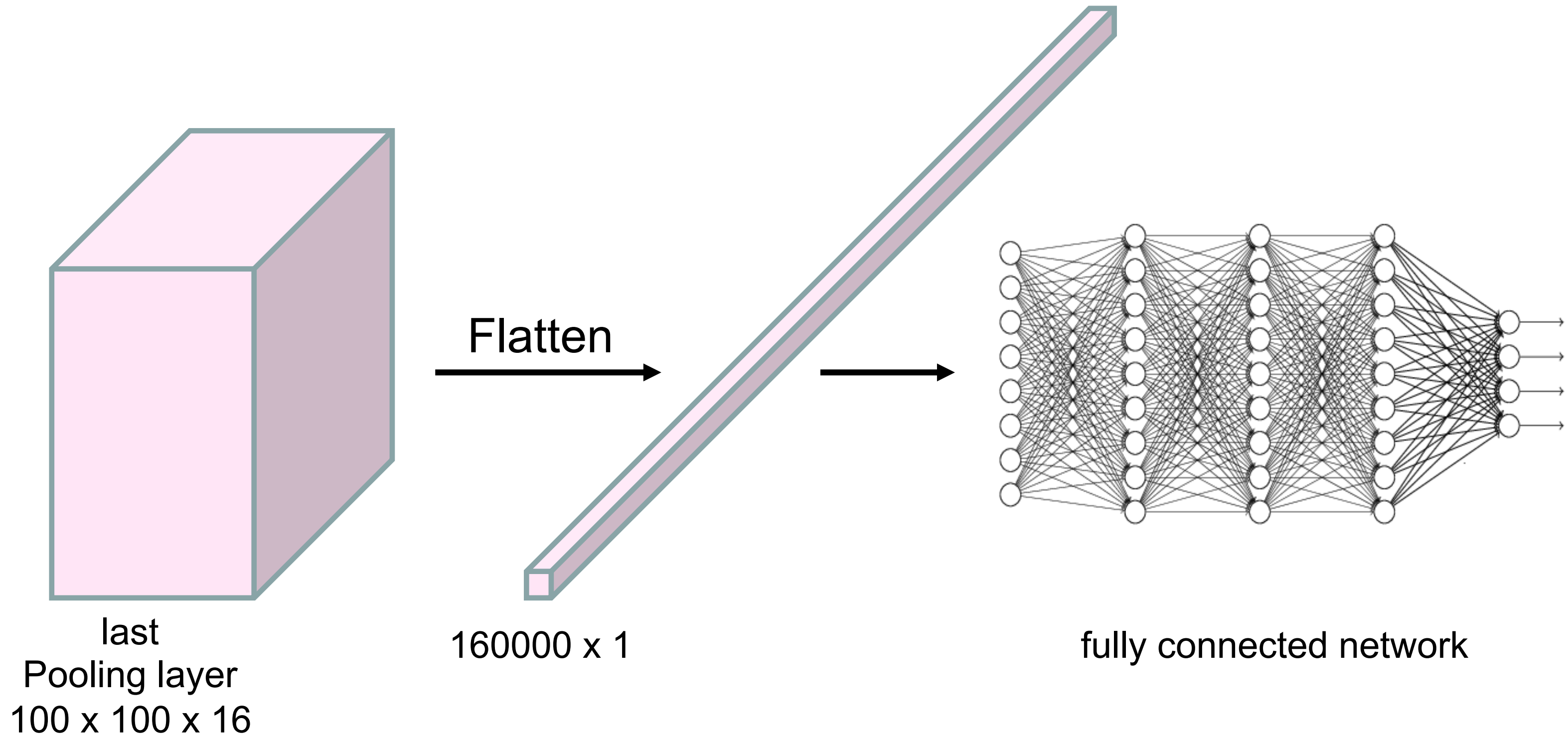
- transforms a layer of size $W \times H \times D$ into a layer of size $W' \times H' \times D'$:

$$W' = \frac{W-F}{S} + 1, \; H' = \frac{H-F}{S} + 1, \; D' = D$$

- Number of weights: none

Fully Connected Layers / Flatten

- at the end: fully connected layers as before (MLP)
  → Flattening



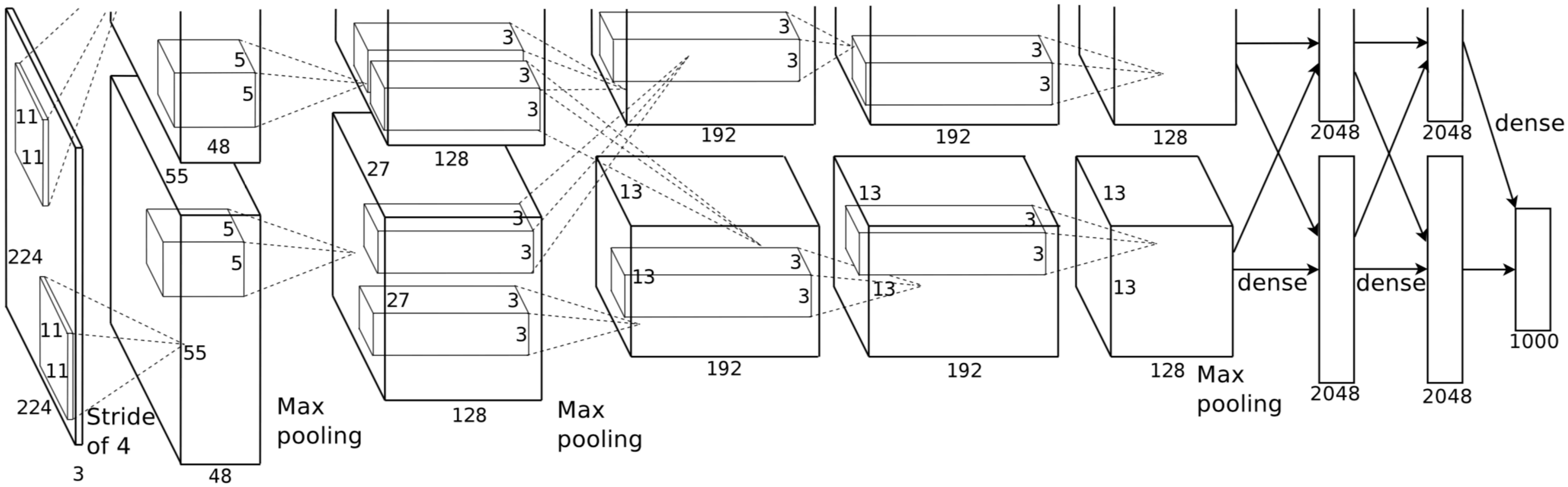
Typical Architecture

$$\text{Input image} \rightarrow \underbrace{\underbrace{\begin{matrix}\text{convolution}\\ \text{ReLU}\end{matrix}}_{n \text{ times}} \rightarrow \text{pool}}_{m \text{ times}} \rightarrow \underbrace{\begin{matrix}\text{fully connected}\\ \text{ReLU}\end{matrix}}_{k \text{ layers}} \rightarrow \begin{matrix}\text{output}\\ \text{Softmax}\end{matrix}$$

- n ca. 3, up to ca. 5
- m large
- $0 \leq k \leq 2$
- General tendency:
  - use smaller filter sizes and deeper architecture
  - away from pooling/fully connected layers towards pure convolutional layers

Technische Hochschule Rosenheim

Example: Alexnet



ImageNet Classification Challenge 2012

- 1000 classes
- 1.2 million training images
- 50,000 validation images
- 150,000 test images

Network:

- 650,000 neurons
- 60 million parameters
- used CNN with ReLU on GPU for the first time

Pre-Processing:

- Scale/Crop images to 256 x 256
  (training uses random crops of size 224x224 from these)
- Subtract mean RGB image

Krizhevsky, Sutskever, Hinton: ImageNet Classification with Deep Convolutional Neural Networks. Commun. ACM 60(6):84-90, 2017.

Transfer-Learning

- Problem:
  Huge data sets required for training so many weights

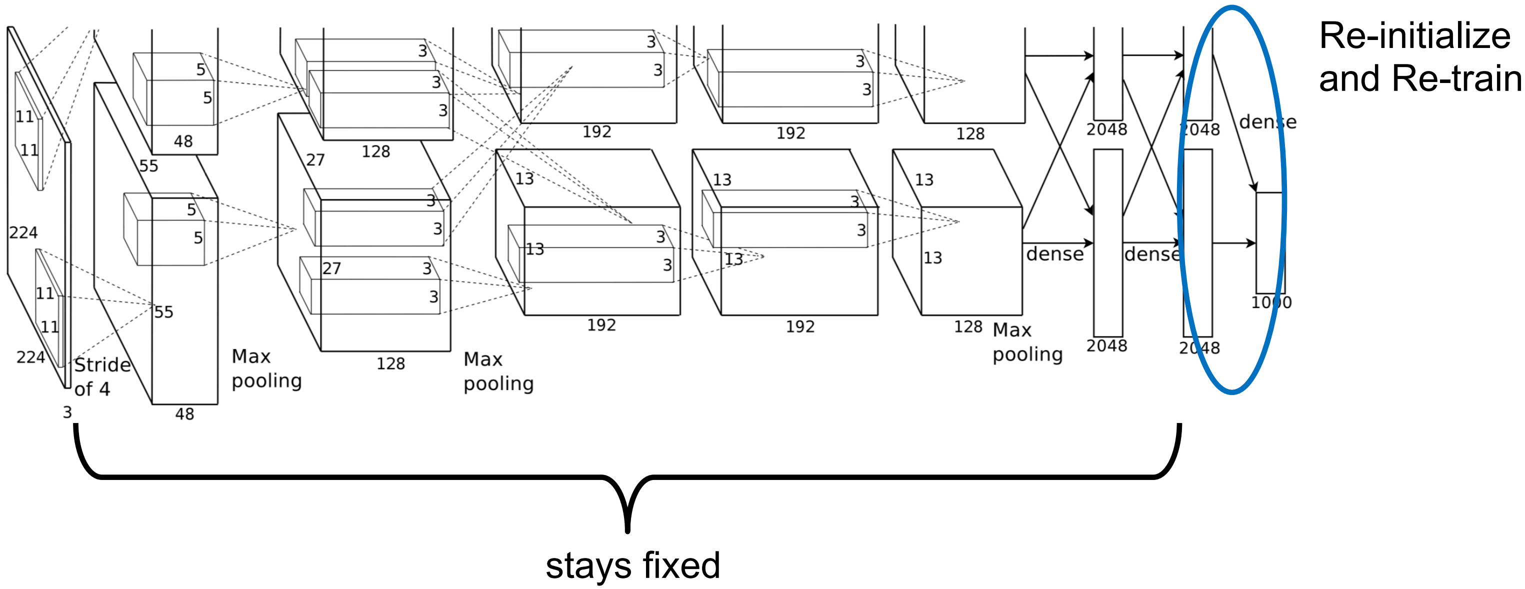
- What if you do not have 1,000,000 images?
  - Get a CNN that was trained on similar data
  - use transfer-learning on that CNN

- Trained models available from, e.g.,
  https://github.com/BVLC/caffe/wiki/Model-Zoo
  https://github.com/tensorflow/models

- Transfer-Learning is the rule, not the exception

Transfer-Learning

small data set:



instead of re-training the last layer of weights also: use CNN as feature extractor:

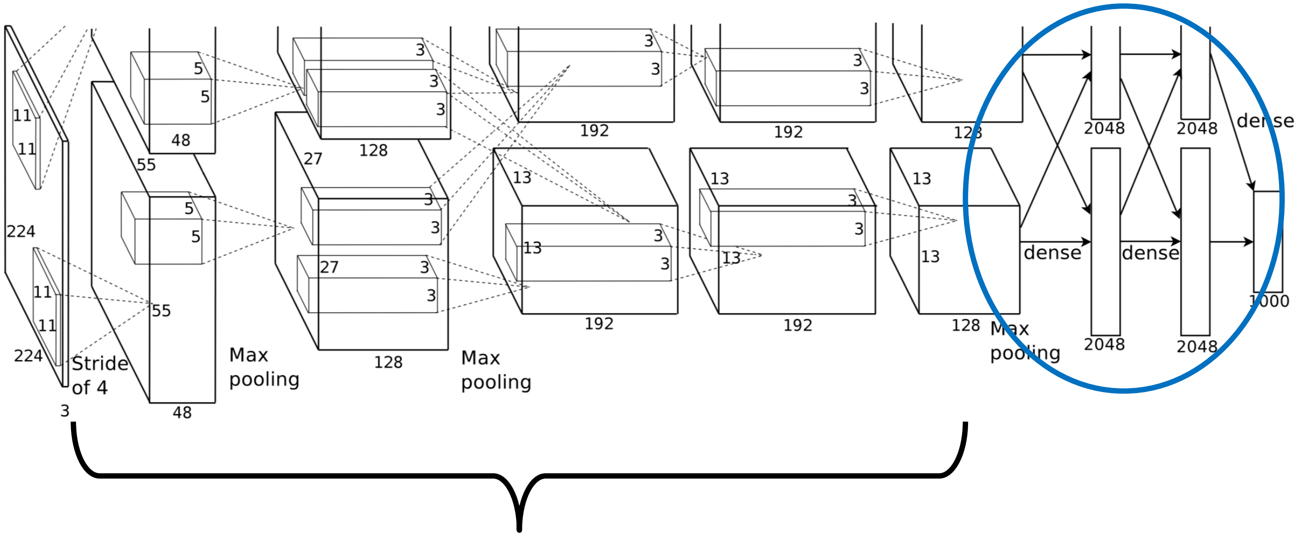
- remove output layer
- the output of the layer before produces features (here: 4096)
- use these features to train a classifier (e.g. SVM)

Advantage: overfitting is not an issue for small data sets

Transfer-Learning

large data set:

re-train



stays fixed

- the larger the data set, the more layers can be re-trained
- chose learning rate considerably smaller than that of the original network (e.g. 1/10)

Object Detection/-localization

- R-CNN – Region Proposal CNN (Girschick 2014)
  https://arxiv.org/abs/1311.2524

- SSD – Single Shot MultiBox Detector (Liu, 2016)
  https://arxiv.org/abs/1512.02325

- YOLO – You Only Look Once (Redmon 2016)
  https://pjreddie.com/darknet/yolo/

Sources

- Goodfellow, Bengio, Courville: *Deep Learning*, MIT Press, 2017. http://www.deeplearningbook.org/
- Li, Johnson, Yeung: *CS231n: Convolutional Neural Networks for Visual Recognition*. Vorlesung Stanford University, 2018. http://cs231n.stanford.edu/
- Li: Deep Learning and Its Applications. Lecture University of Waterloo, 2017. https://cs.uwaterloo.ca/~mli/cs898-2017.html
- Original research articles as stated on the slides