Convolutional neural networks

Machine Learning

Natalia Khanzhina

07.10.2019

Lecture plan

- Brief overview of ImageNet
- Earlier approaches in computer vision
- Convolutional neural networks
- Tensors
- Deconvolution and visualization of neurons
- Architecture overview
- Computer vision problems

- The presentation is prepared with materials of D. - Polykovsky and K. Khrabrov “Neural networks in machine learning”
  - S. Nayak’s “Number of Parameters and Tensor Sizes in a Convolutional Neural Network”
  - A. Karpathy’s course and blog

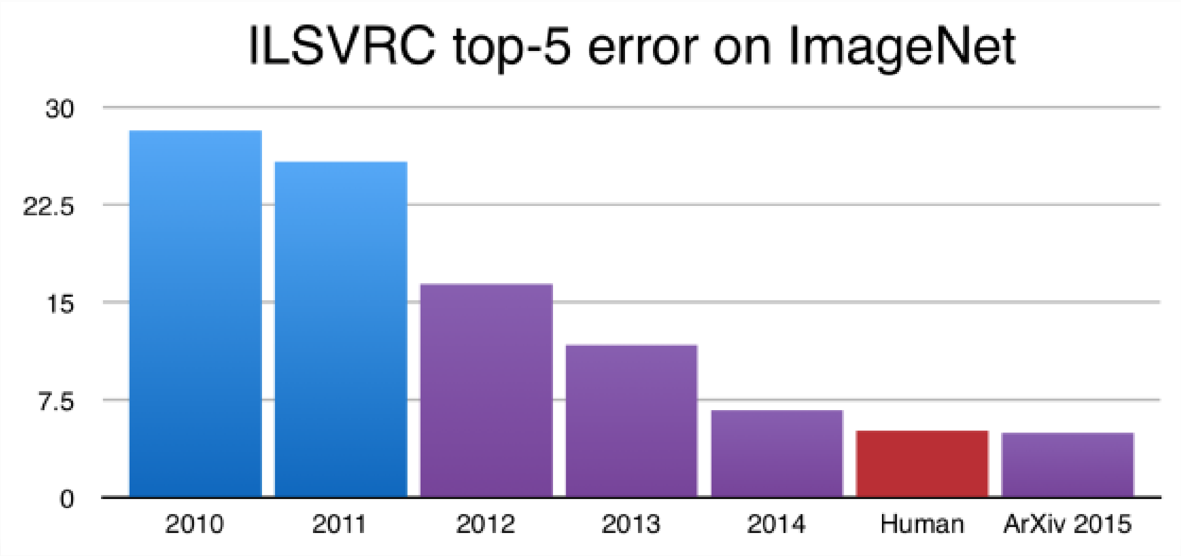
Lecture plan

- Brief overview of ImageNet
- Earlier approaches in computer vision
- Convolutional neural networks
- Tensors
- Deconvolution and visualization of neurons
- Architecture overview
- Computer vision problems

Today history (reminder)

2012 Hinton, Krizhevsky, and Sutskever suggest Dropout

2012 They win ImageNet (and two less known competitions). Deep learning era begins.

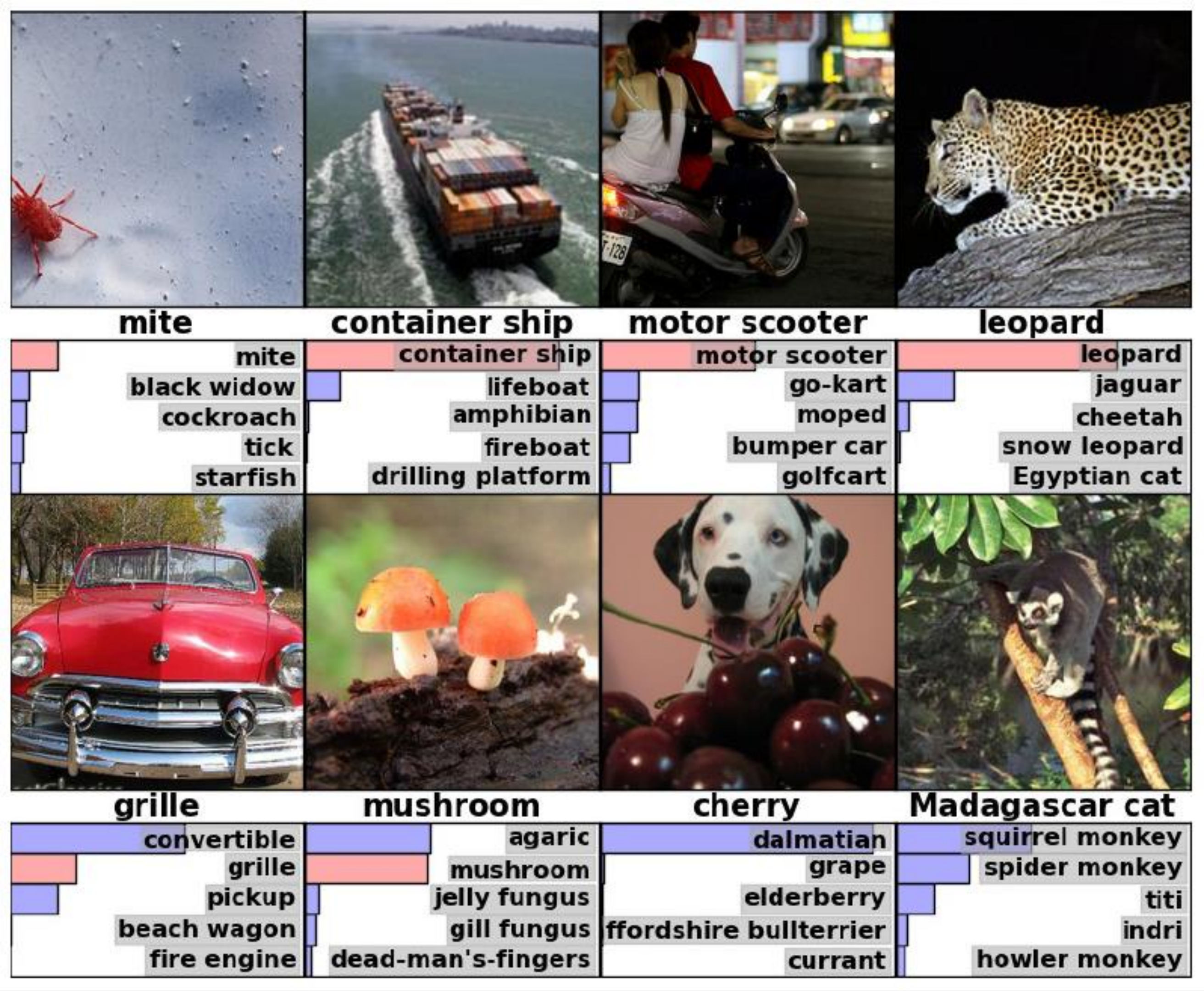


Imagenet Challenge

IMAGENET

- 1000 images per class
- 1000 classes
- Today, 14 mln images

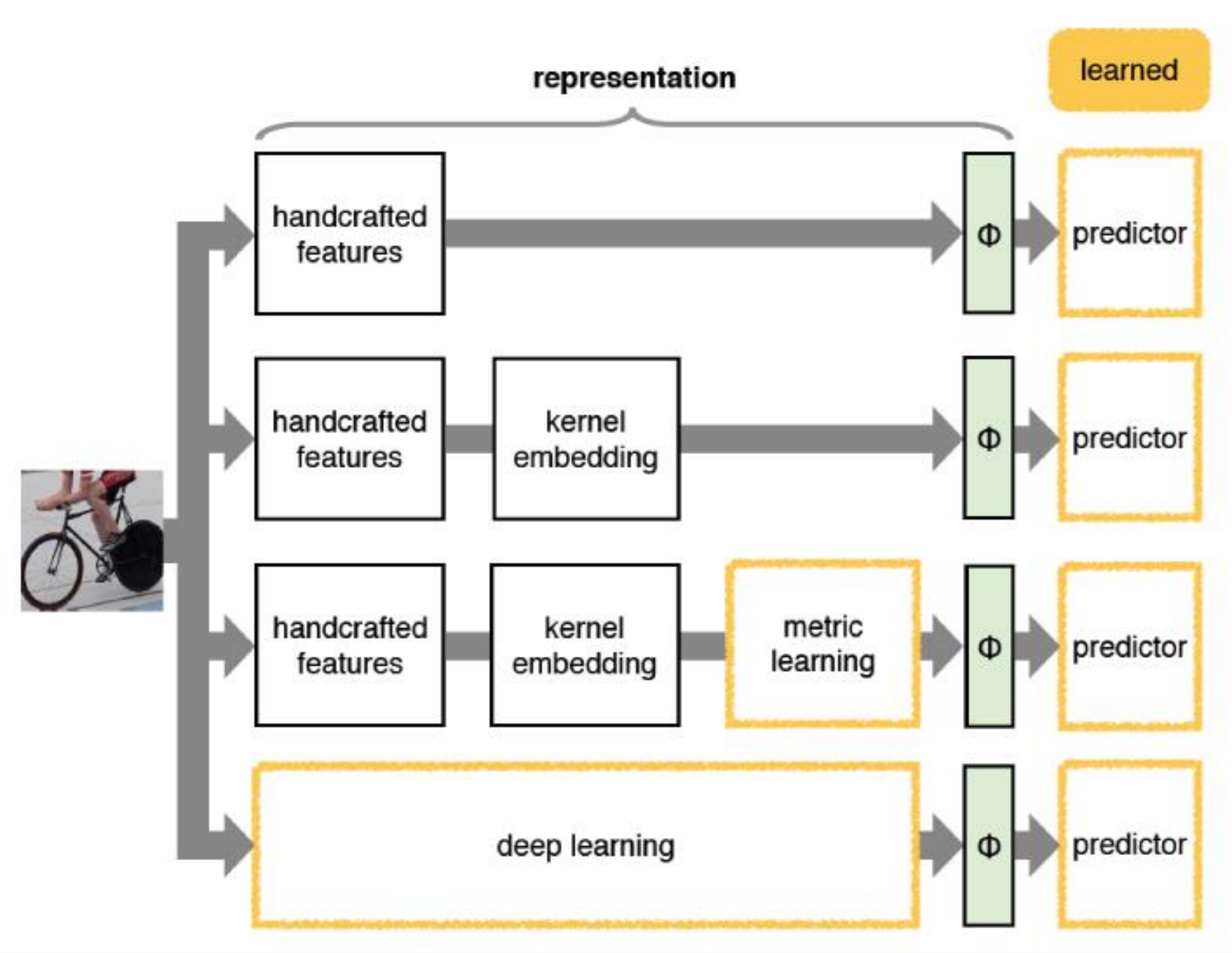
Examples of images



Lecture plan

- Brief overview of ImageNet
- Earlier approaches in computer vision
- Convolutional neural networks
- TensorsDeconvolution and visualization of neurons
- Architecture overview
- Computer vision problems

Short history of computer vision



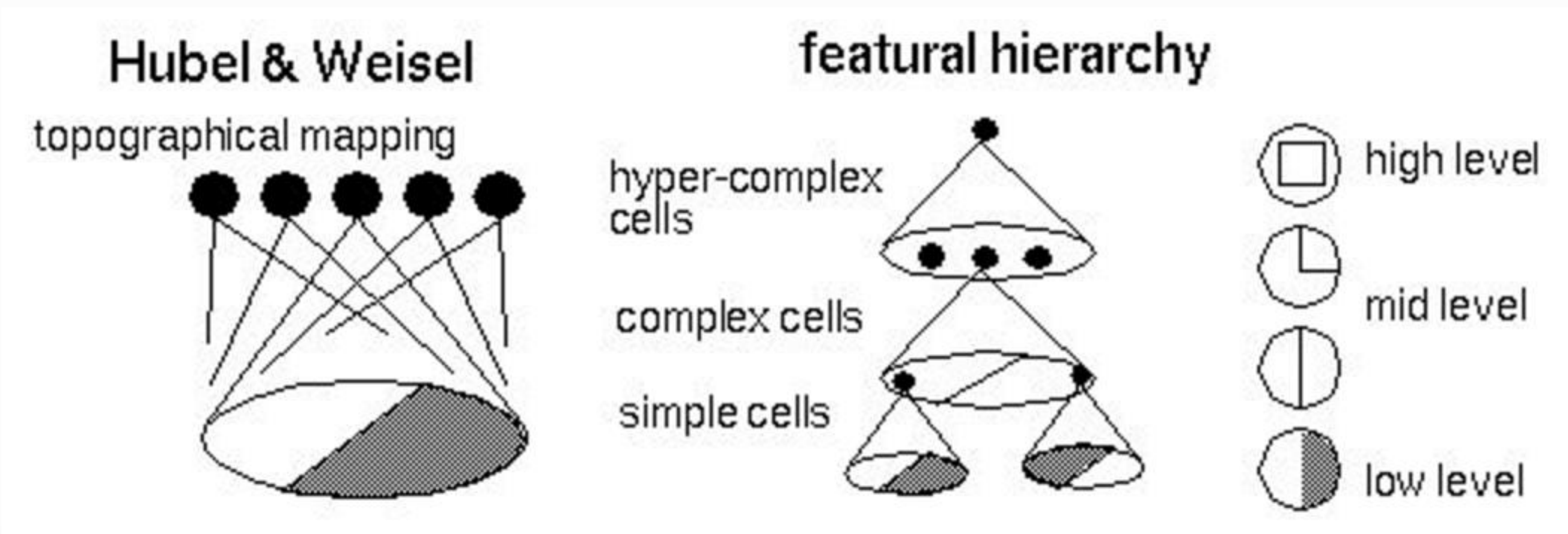
Core concepts

- **Local perception**: each neuron sees a small part of the object. Use kernels (filters) to capture 1-D or 2-D structure of objects. For instance, capture all pixel neighbors for an image.
- **Weight sharing:** use small and the same sets of kernels for all objects, this leads to reduction of number of adjusting parameters in comparison with MLP
- **Subsampling/pooling**: use dimensionality reduction for images in order to provide invariance to scale

Biological inspiration

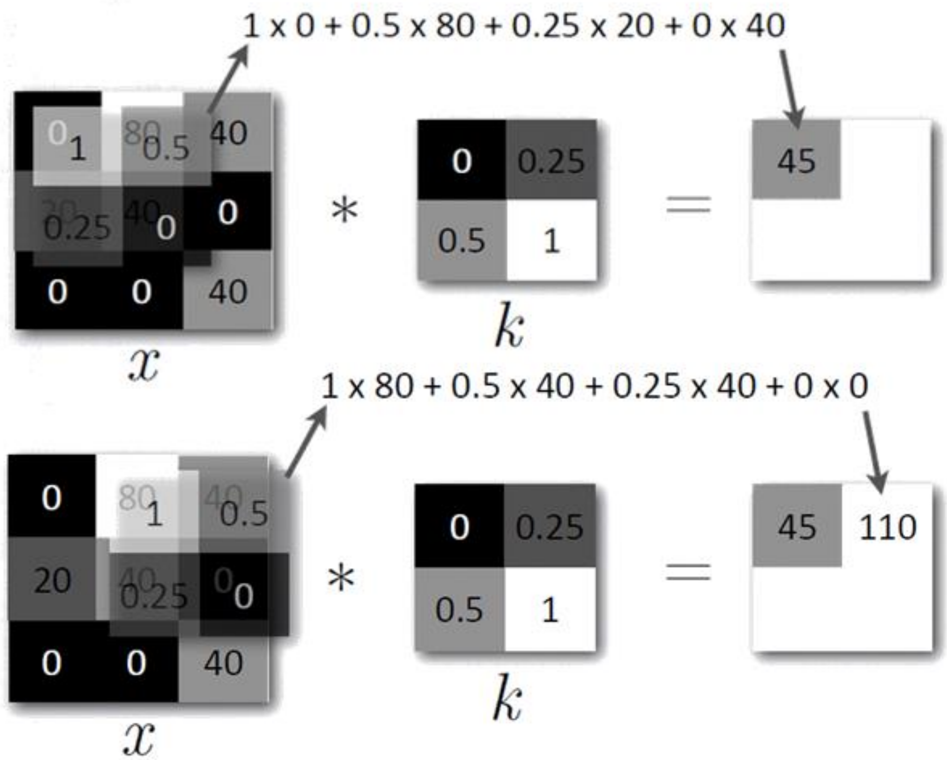
Hubel and Weisel showed that an image received by cat's retina is processed in a way that:

- neighboring neurons processes neighboring area of retina
- neurons are hierarchically structured



Discrete kernel

$$(x * k)_{ij} = \sum_{pq} x_{i+p,j+q}\, k_{r\text{-}p,r\text{-}q}$$



What kernels can do?

*

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

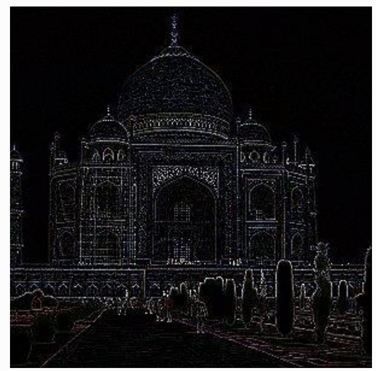
=

blur

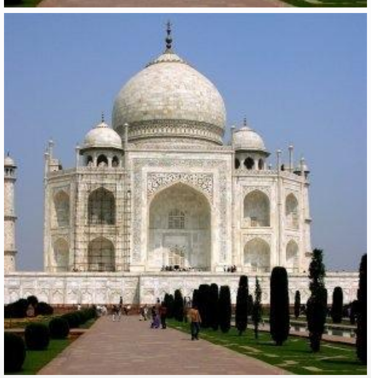
*

	0	1	0	
	1	-4	1	
	0	1	0	

=



edge detection



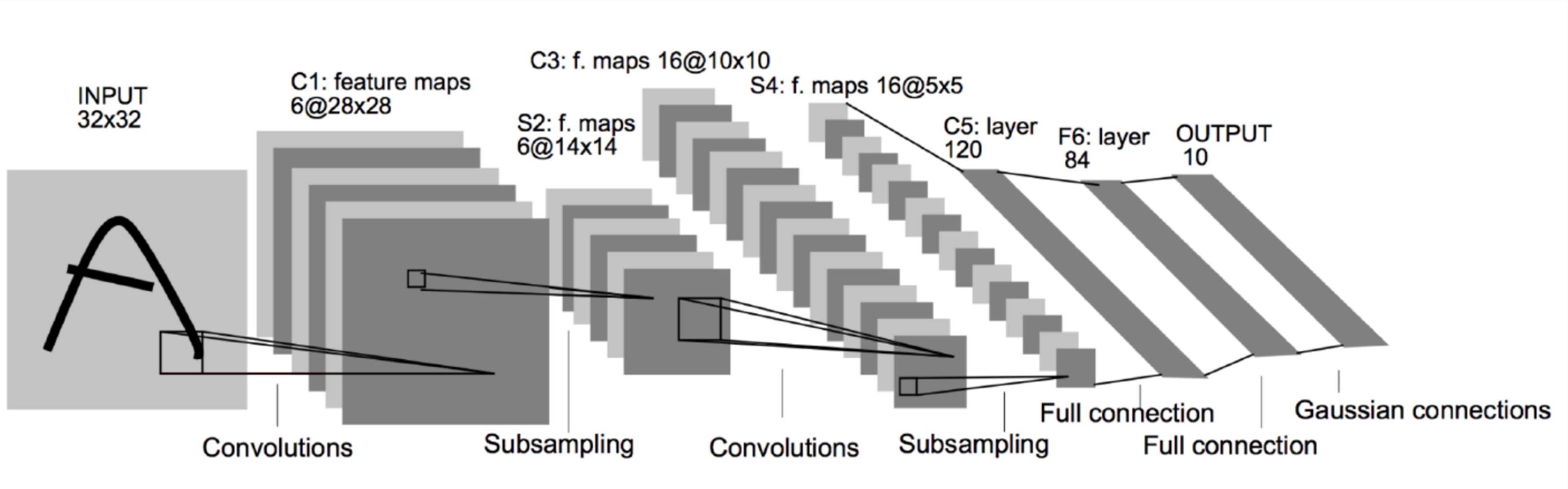
*

0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0

=

sharpen

LeNet (1998)



INPUT
32x32
C1: feature maps
6@28x28
S2: f. maps
6@14x14
C3: f. maps 16@10x10
S4: f. maps 16@5x5
C5: layer
120
F6: layer
84
OUTPUT
10
Convolutions
Subsampling
Convolutions
Subsampling
Full connection
Full connection
Gaussian connections

LeNet

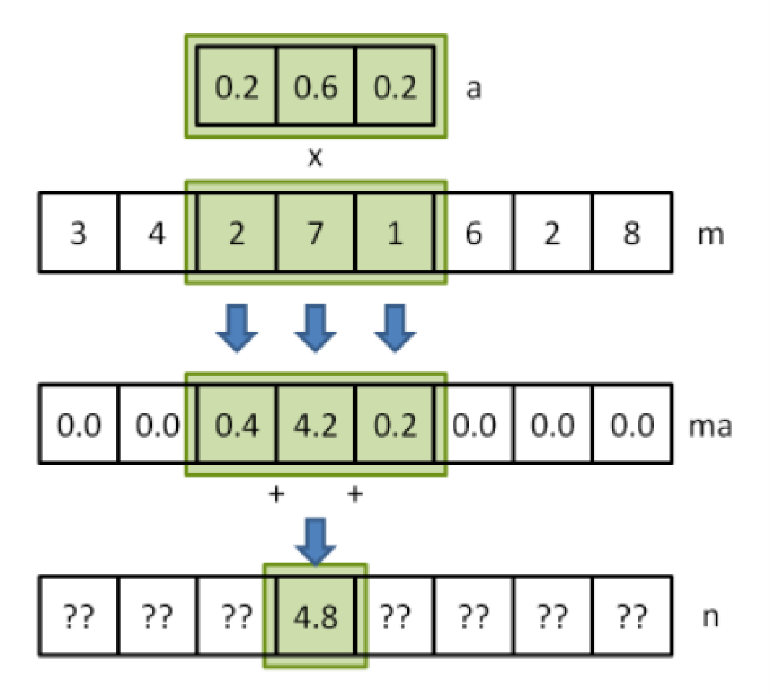
- Ideas of hierarchical structure and neighboring areas combined with backpropagation
- Not so many layers
- Good on MNIST

Lecture plan

- Brief overview of ImageNet
- Earlier approaches in computer vision
- Convolutional neural networks
- Tensors
- Deconvolution and visualization of neurons
- Architecture overview
- Computer vision problems

Convolution

Convolution of array *m* with kernel *a* is an array $ma[k] = \sum_{i=-w}^{w} m[k+i]\, a[-i]$



Convolution properties

- Associative property
- Commutative property
- Linearity

Padding

Zero shift

0	0	**A**	**B**	**C**	0	0

Border extension

A	A	**A**	**B**	**C**	C	C

Mirror shift

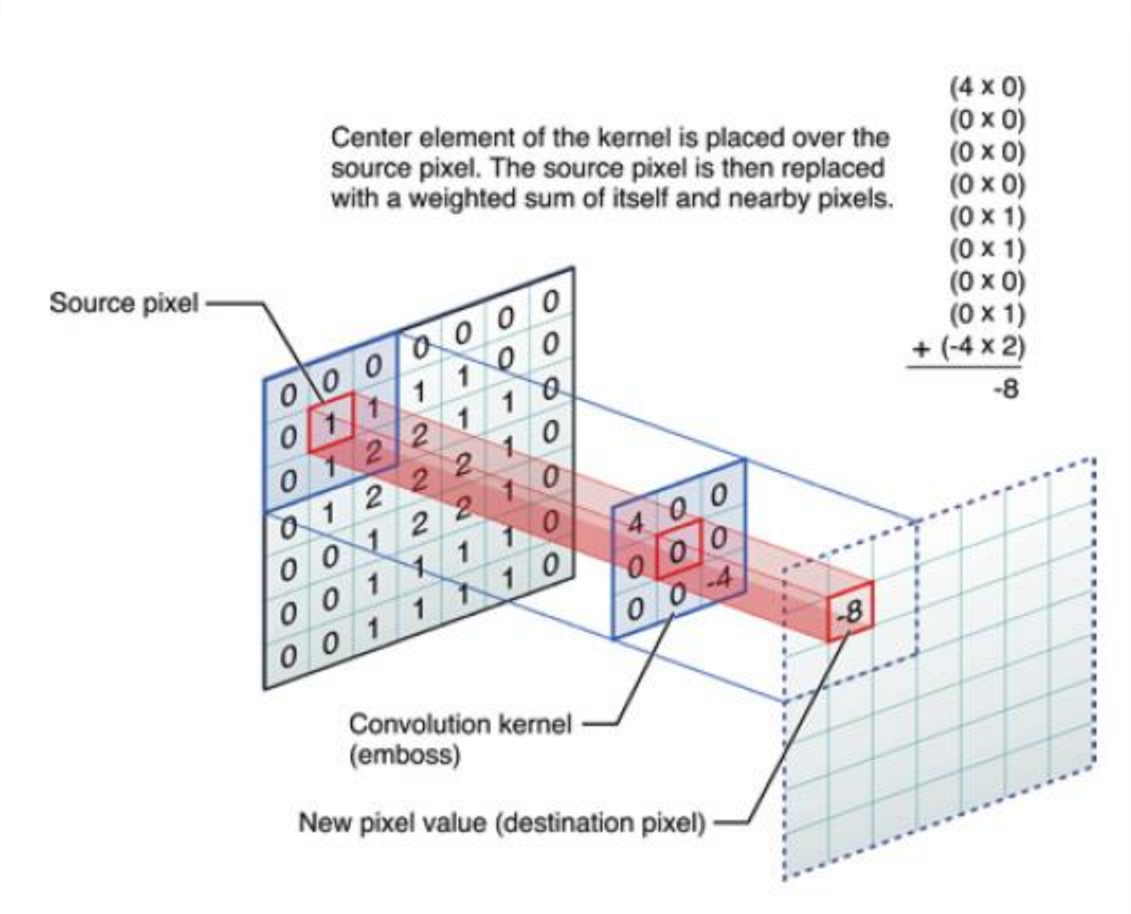
B	A	**A**	**B**	**C**	C	B

C	B	**A**	**B**	**C**	B	A

Cyclic shift

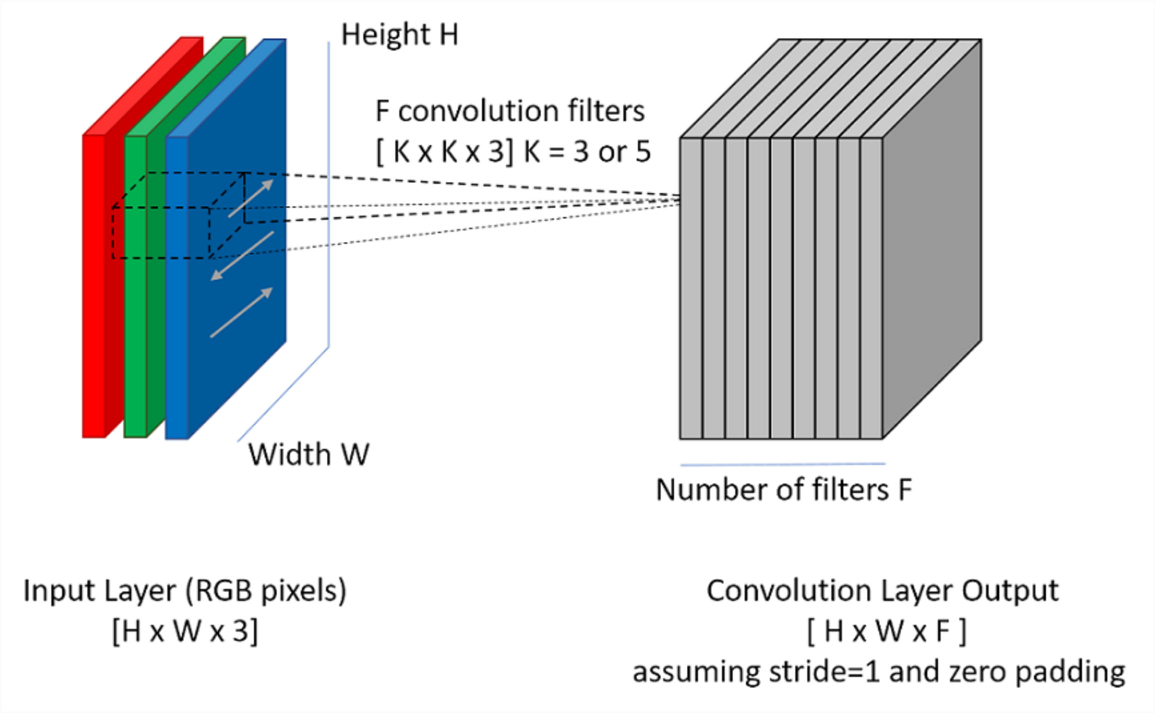
B	C	**A**	**B**	**C**	A	B

2-D convolution



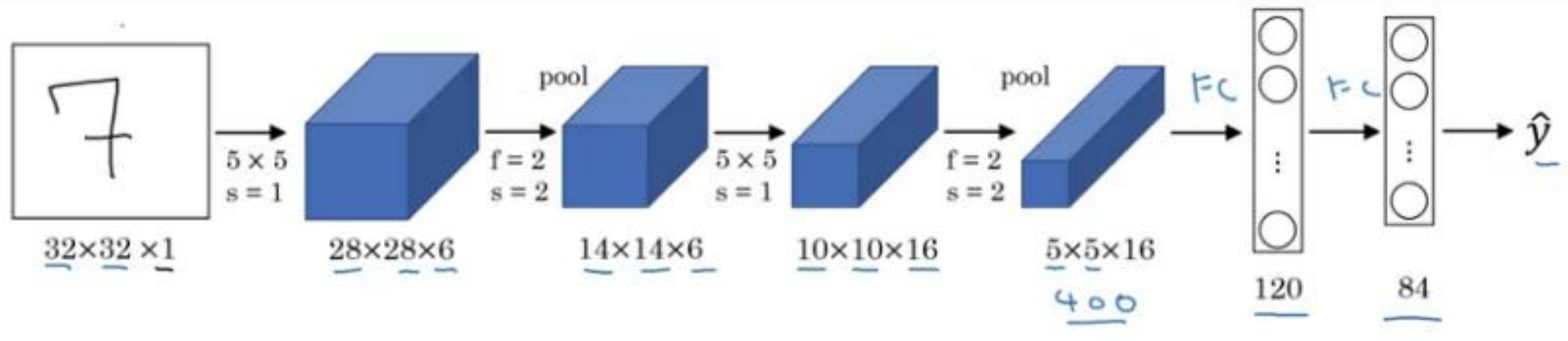
Convolutional tensors (1/2)

An image is not just one feature map, but several (typically, 3) feature maps



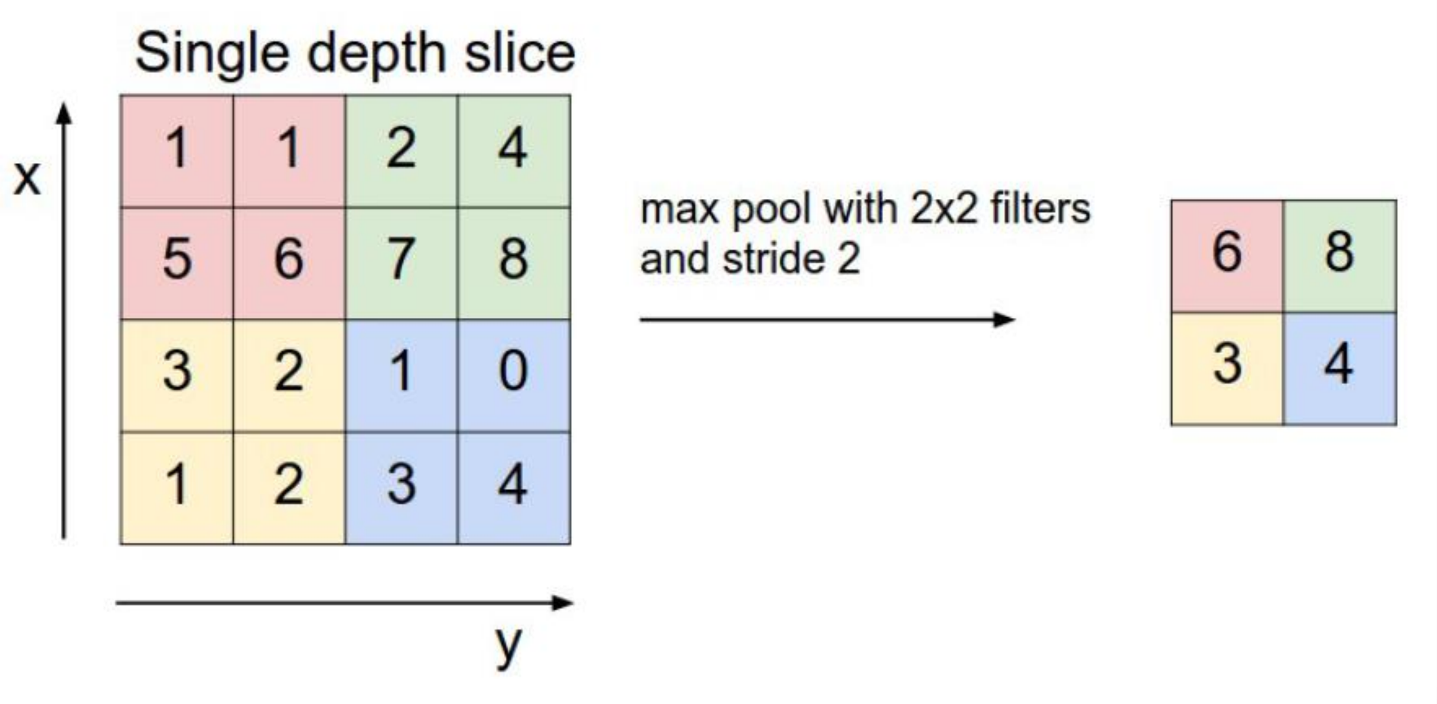
Convolutional tensors (2/2)

Tensor flow of LeNet

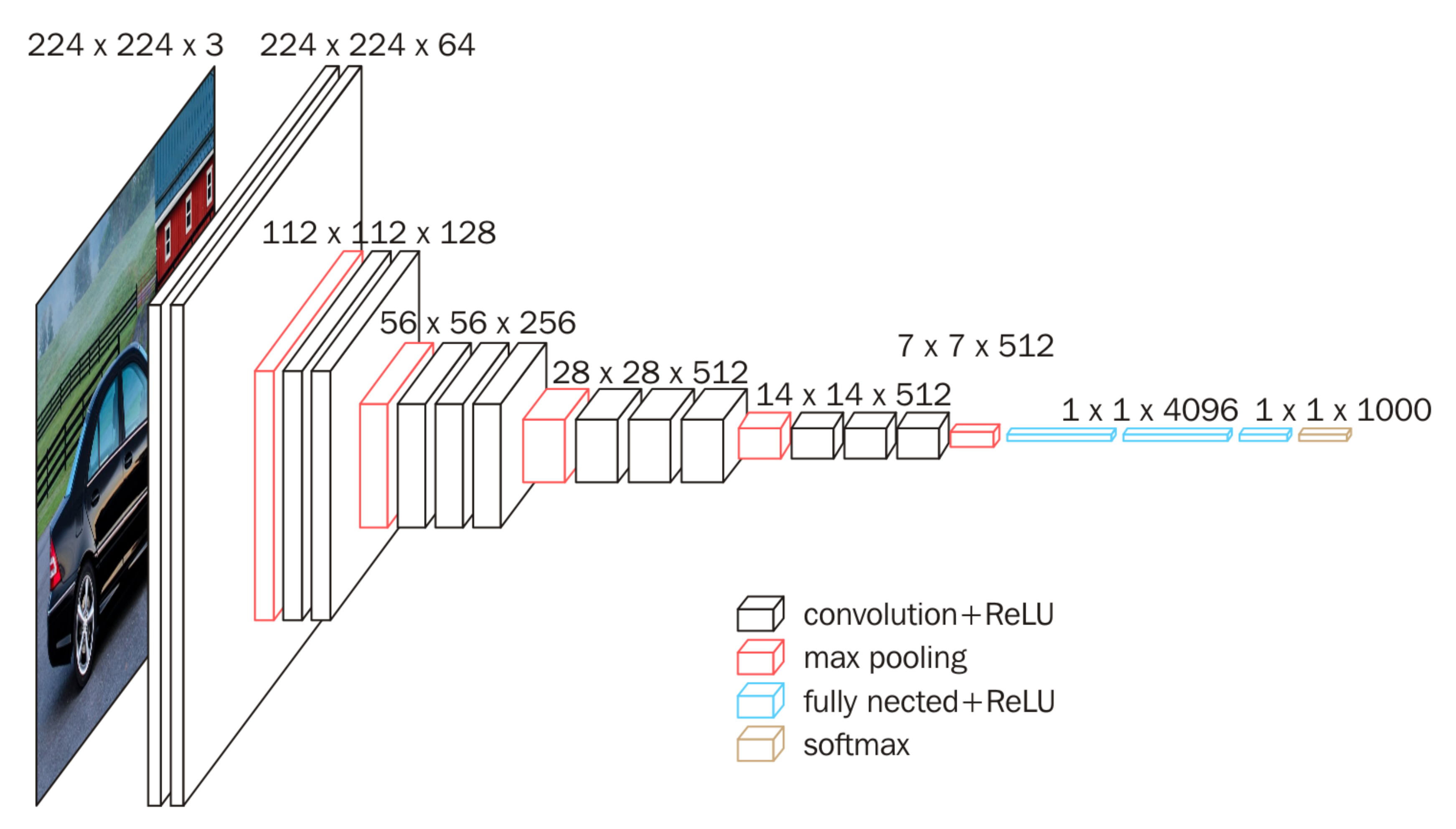


Pooling

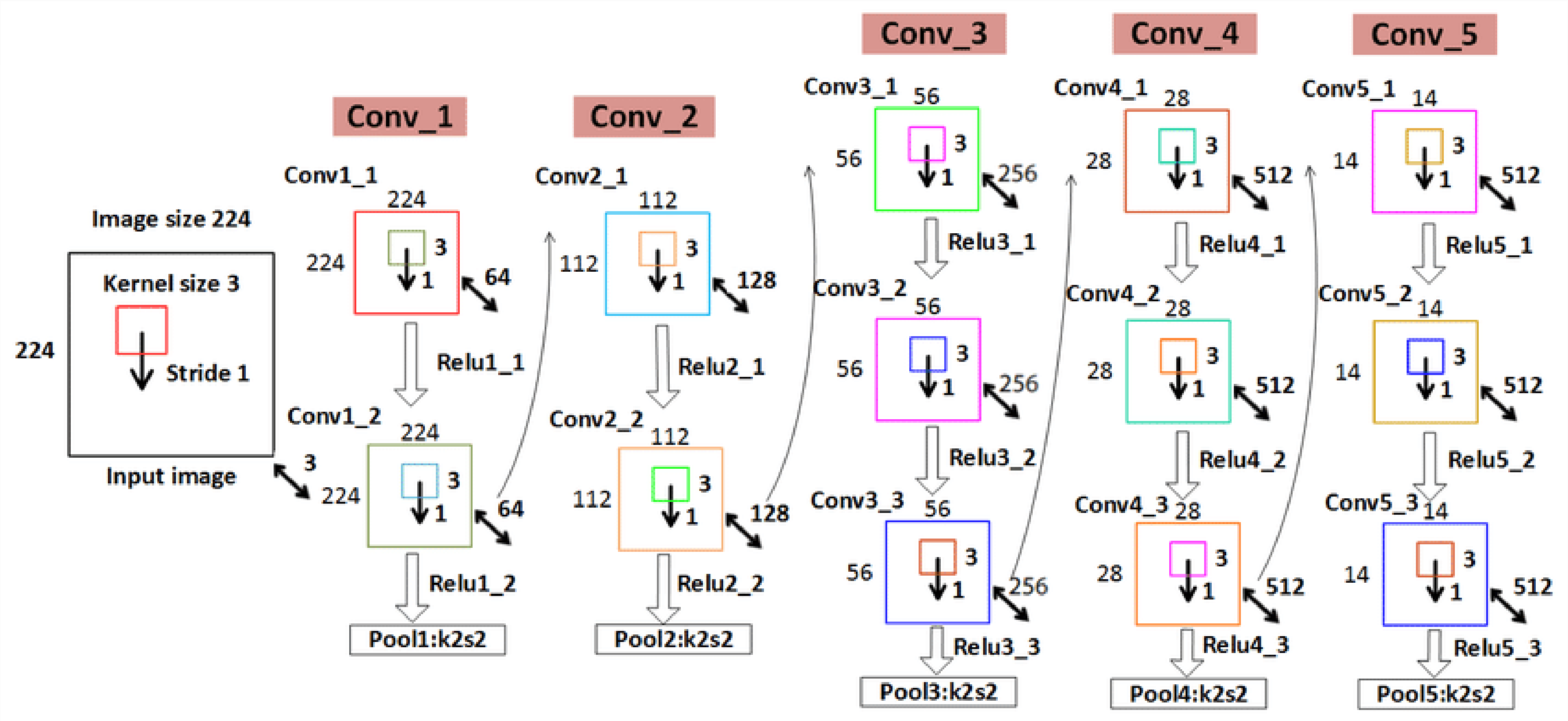
- Pooling is used to reduce dimensionality
- It also decorrelates neurons



VGG-16 conceptual scheme

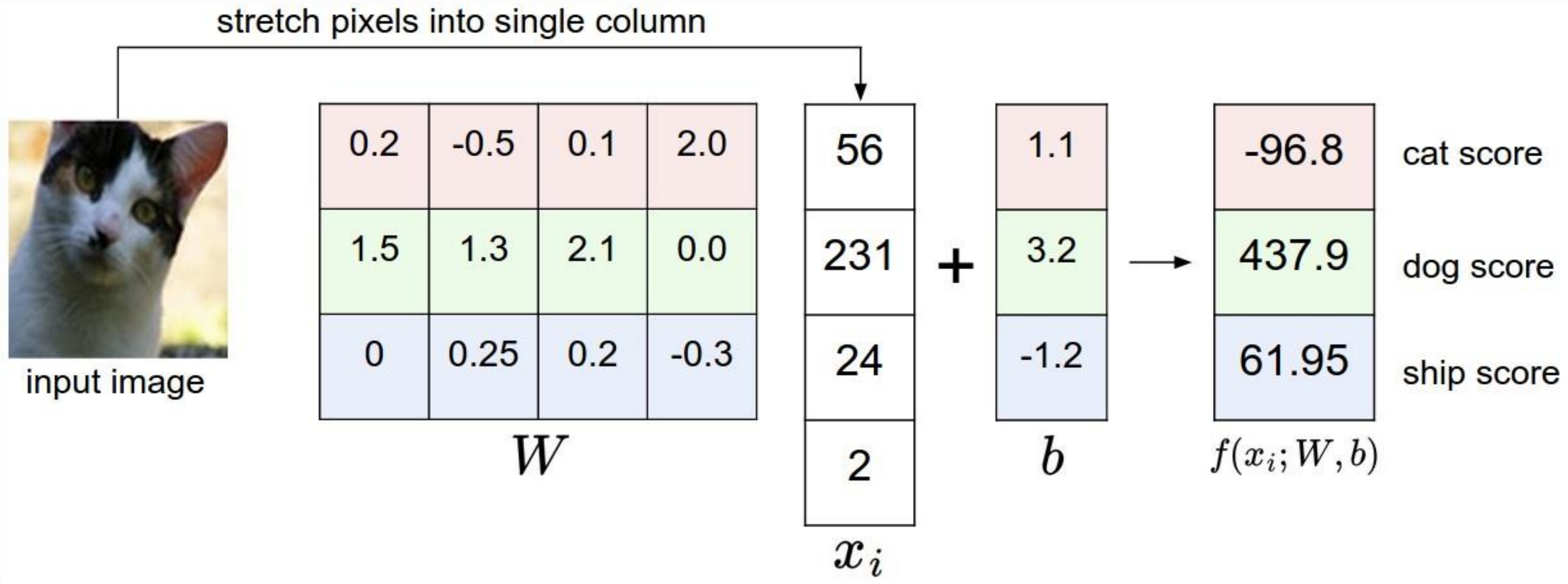


VGG-16 technical scheme



Multiclass classifier

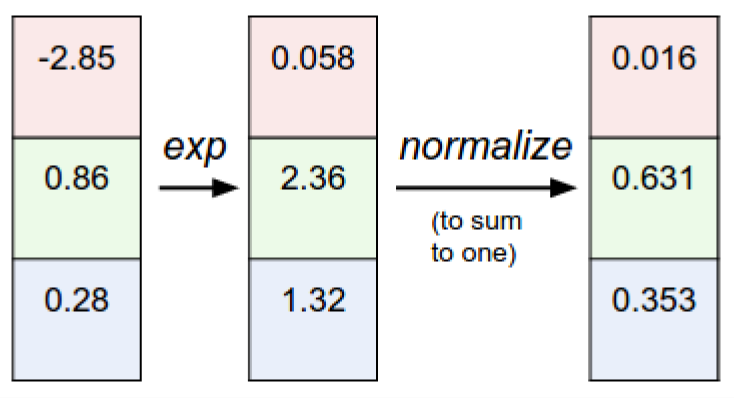
- We have *N* classes to choose from
- Thus, we have *N* neurons in output layer



Softmax layer

Instead of using scores, we can use probabilities obtained with softmax normalization:

$$p_{\text{softmax}}(s_i) = \frac{e^{s_i}}{\sum_j e^{s_j}}$$



Cross-entropy loss

As a result, we can apply more helpful cross-entropy loss:

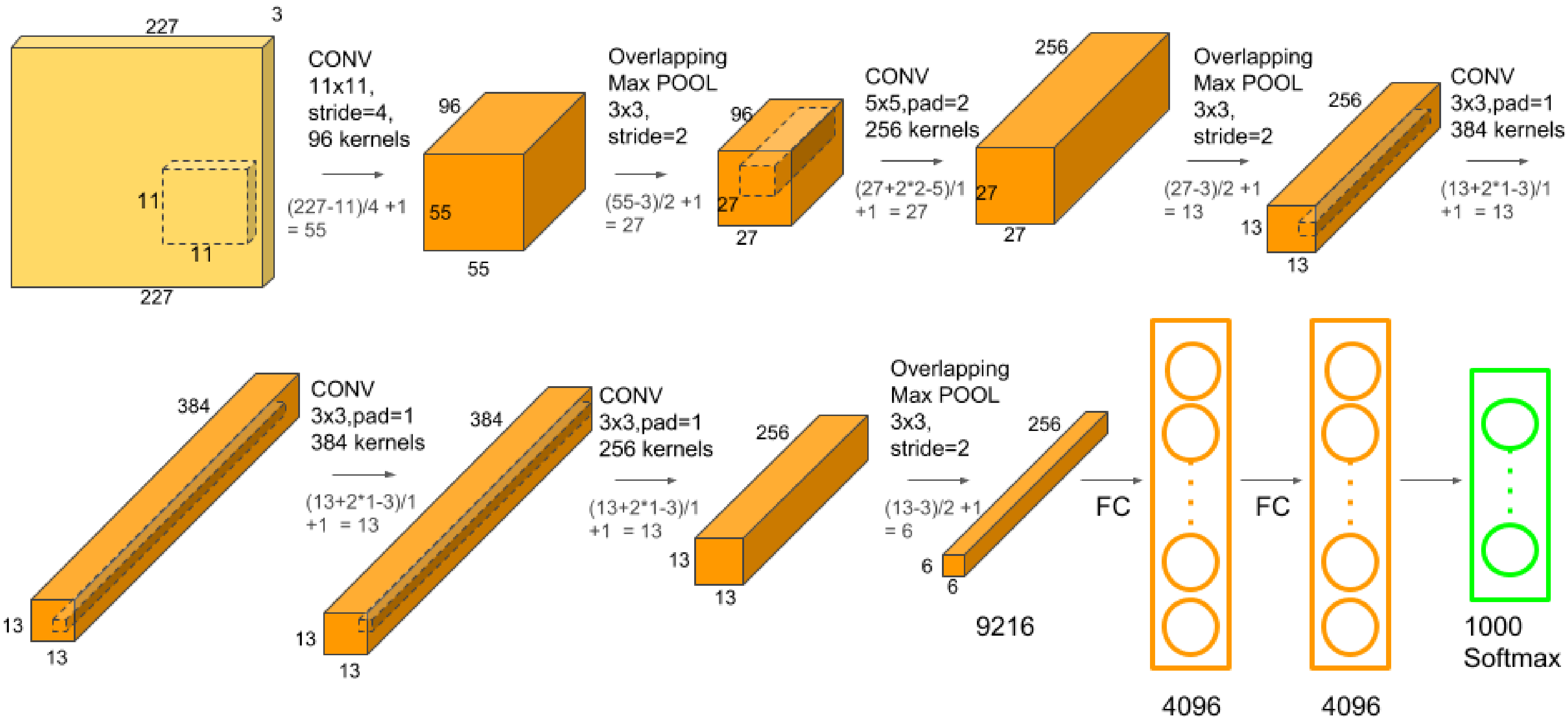
$$H\left(p_y, p_{\text{softmax}}\right) = -\sum_{x \in X} p_y(x) \log p_{\text{softmax}}(x),$$

where X are objects, $p_y(x)$ is a target distribution $(0,0,\ldots,1,\ldots,0)$, and p_{softmax} is the distribution we obtain after Softmax.

Lecture plan

- Brief overview of ImageNet
- Earlier approaches in computer vision
- Convolutional neural networks
- Tensors
- Deconvolution and visualization of neurons
- Architecture overview
- Computer vision problems

Tensors changing with LeNet



Size after convolution

$$O = \frac{I - K + 2P}{S} + 1$$

O is size (width) of output image

I is size (width) of input image

K is size (width) of kernels used in the convolution layer

N is number of kernels

S is stride

P is padding

Size after pooling

$$O = \frac{I - P_s}{S} + 1$$

O is size (width) of output image

I is size (width) of input image

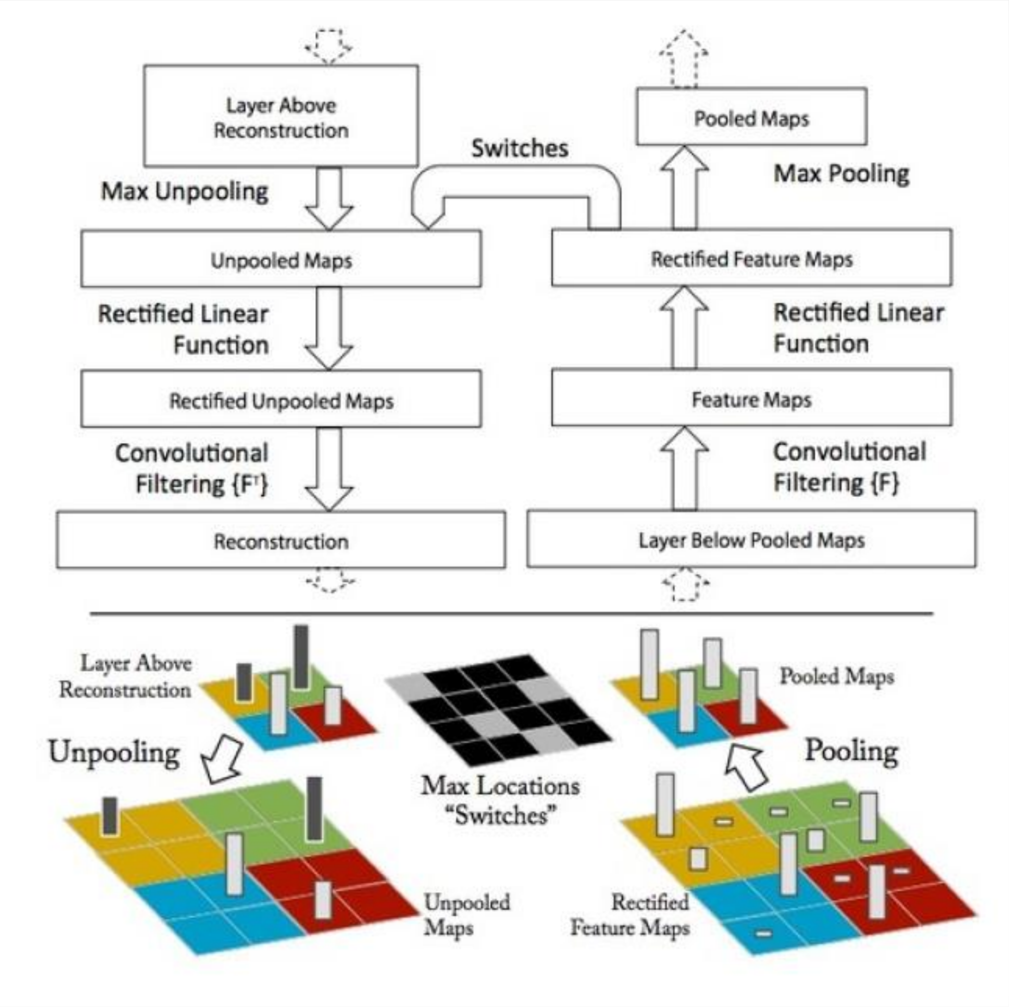
S is stride

P_s is pooling size

Lecture plan

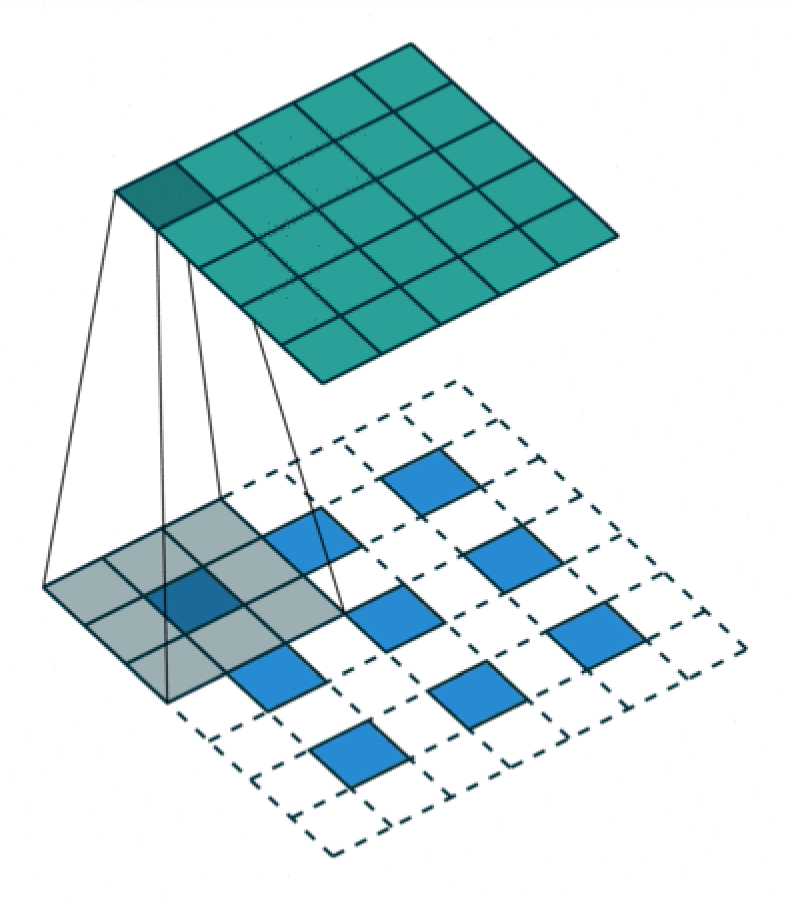
- Brief overview of ImageNet
- Earlier approaches in computer vision
- Convolutional neural networks
- Tensors
- Deconvolution and visualization of neurons
- Architecture overview
- Computer vision problems

Deconvolution neural network

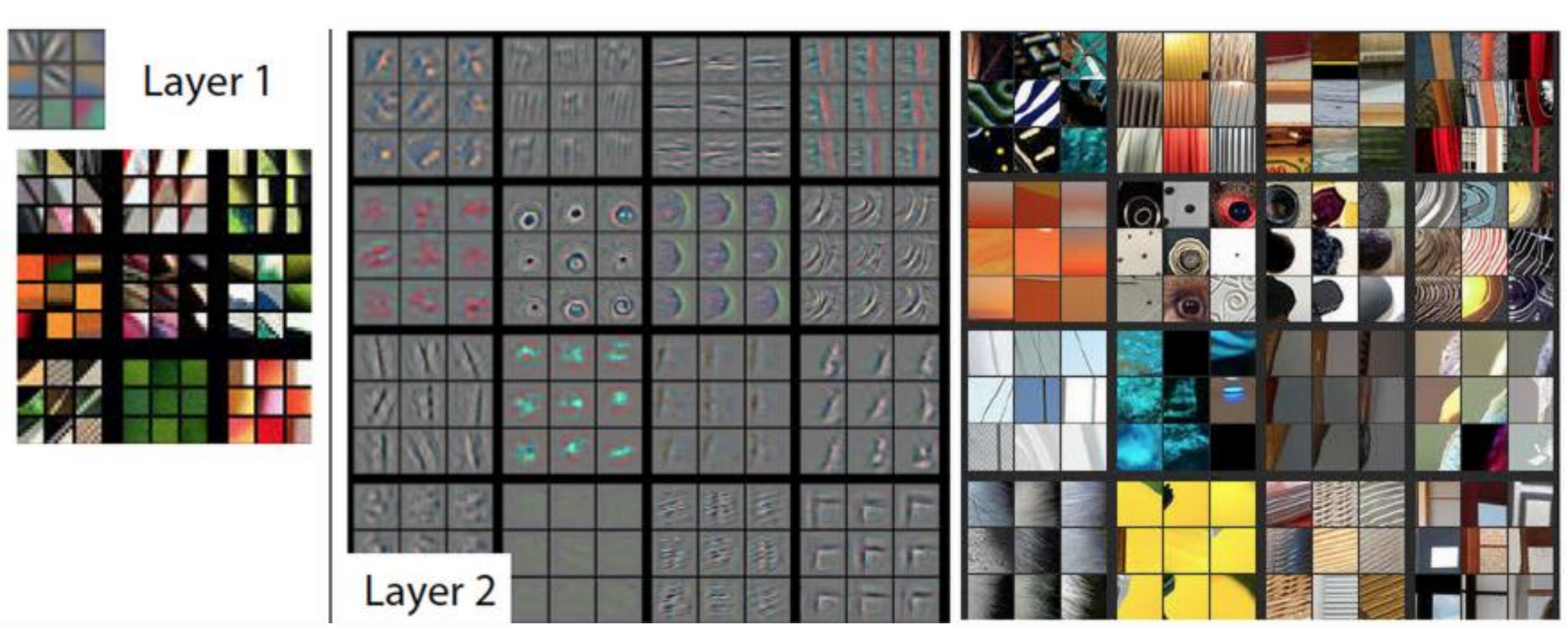


Deconvolution

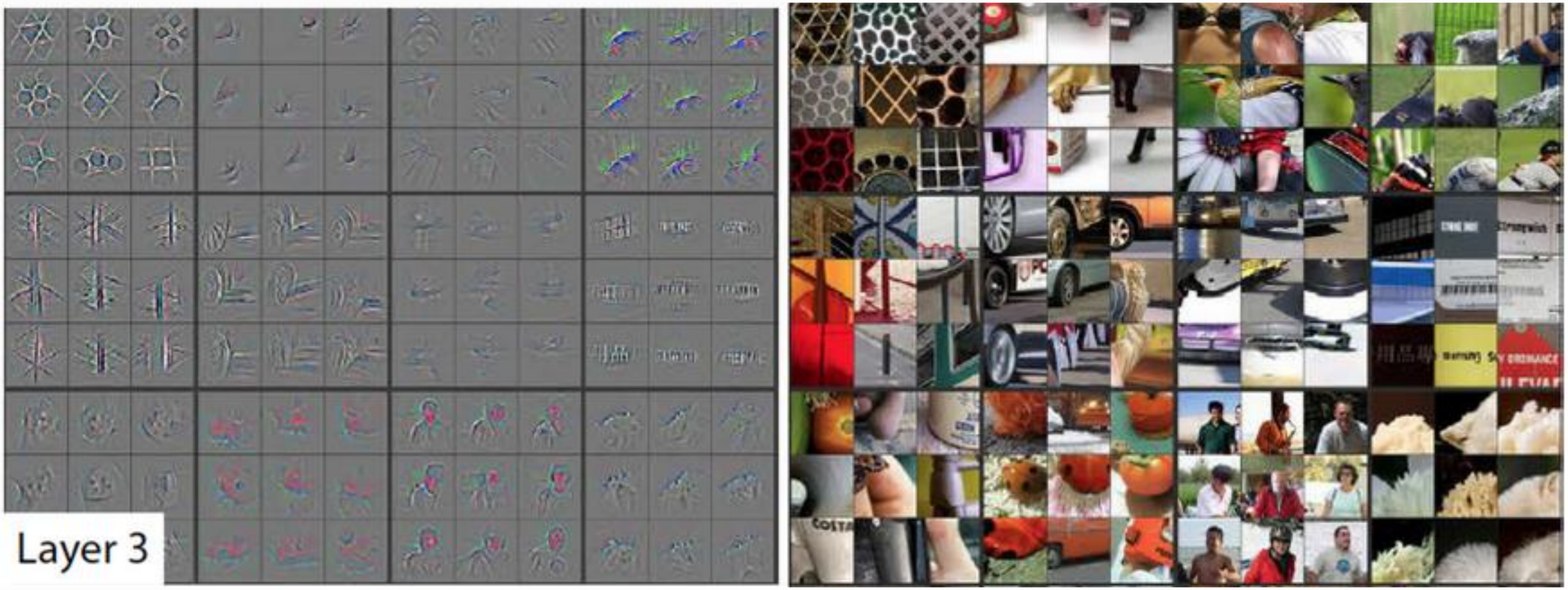
Just use the transposed kernel with strided inputs



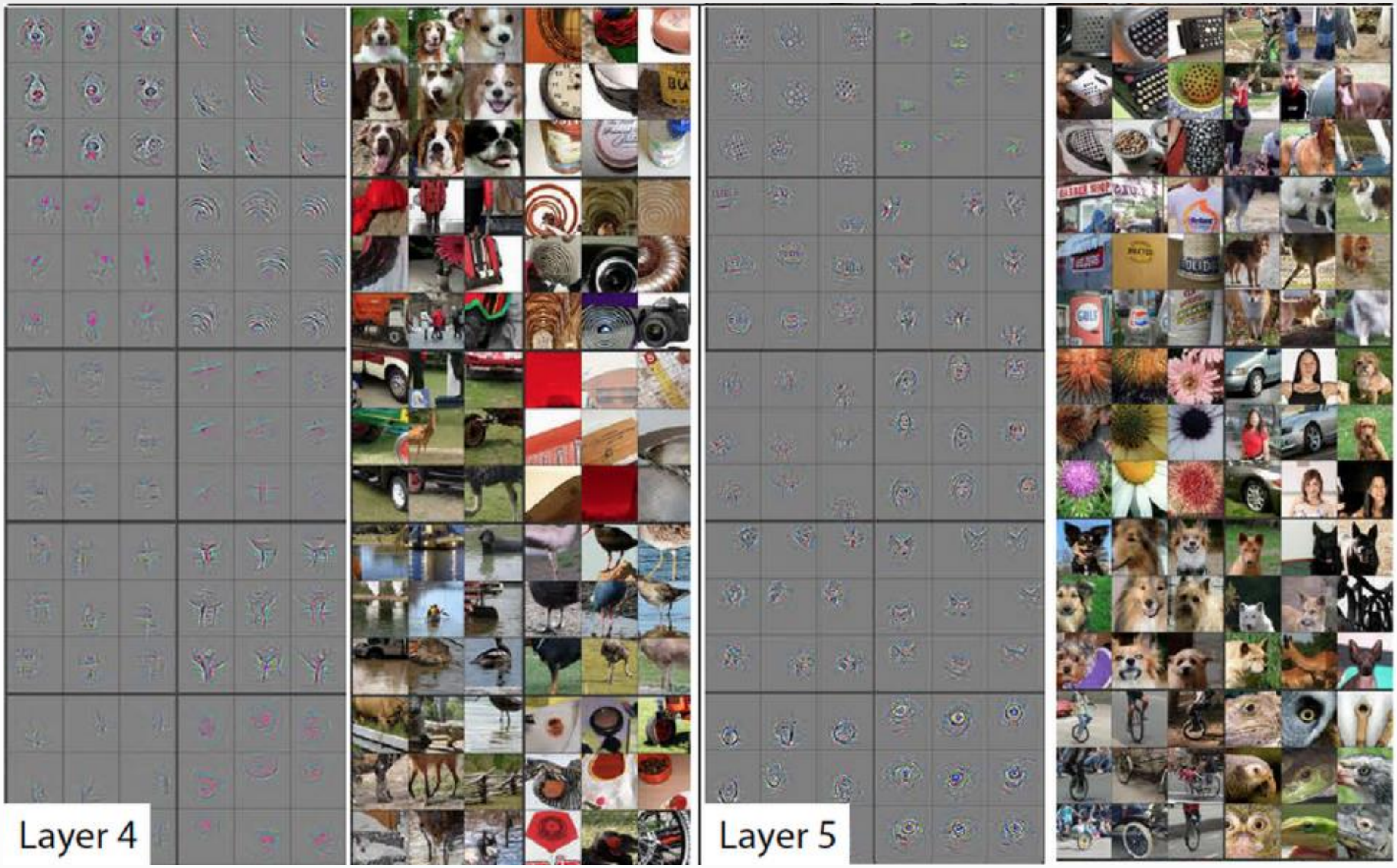
Visualization of neuron activation



Visualization of neuron activation



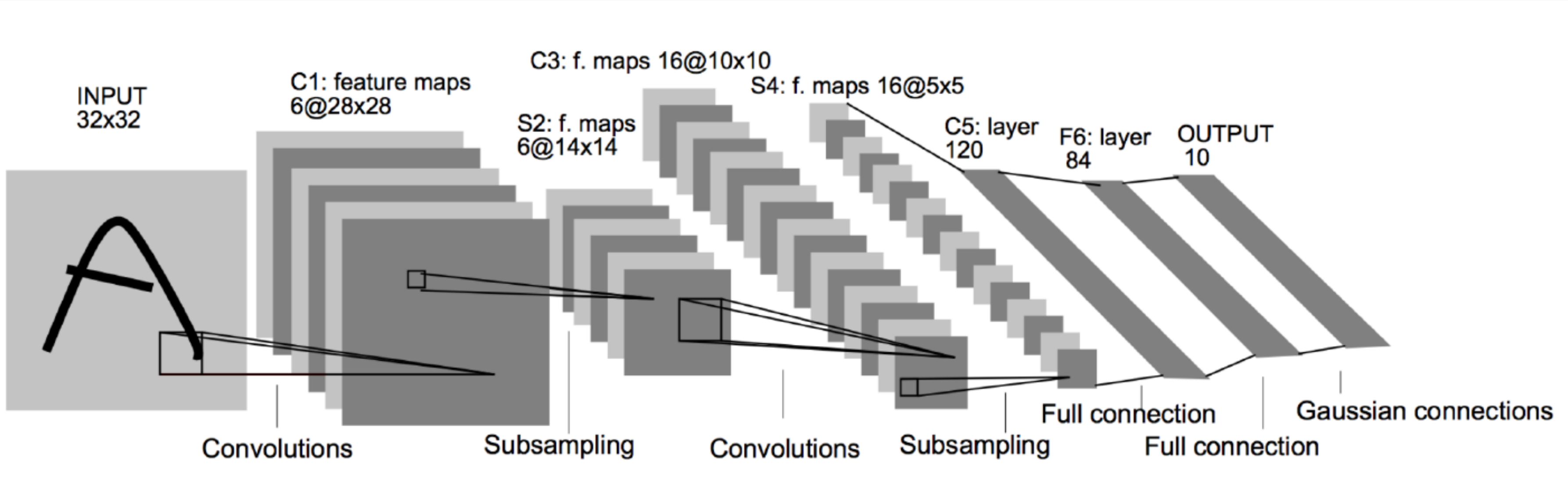
Visualization of neuron activation



Lecture plan

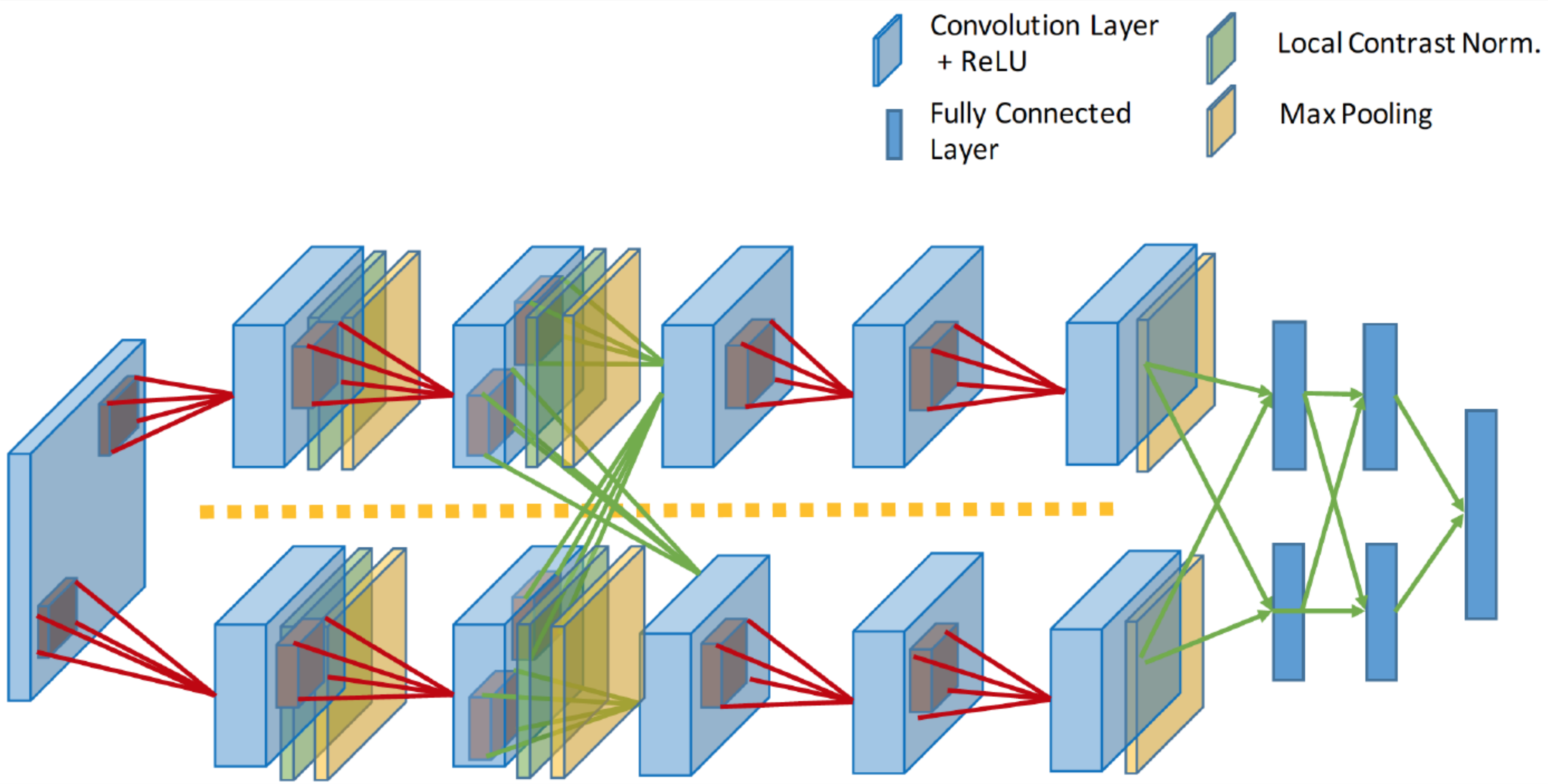
- Brief overview of ImageNet
- Earlier approaches in computer vision
- Convolutional neural networks
- Tensors
- Deconvolution and visualization of neurons
- Architecture overview
- Computer vision problems

LeNet (1998)



INPUT
32x32
C1: feature maps
6@28x28
S2: f. maps
6@14x14
C3: f. maps 16@10x10
S4: f. maps 16@5x5
C5: layer
120
F6: layer
84
OUTPUT
10
Convolutions
Subsampling
Convolutions
Subsampling
Full connection
Full connection
Gaussian connections

AlexNet (2014)



AlexNet

- Bigger version of LeNet
- Size of convolution decreases (from 11x11 to 3x3) between input and output
- Won ImageNet 2014

VGG-16 and VGG-19 (2014)

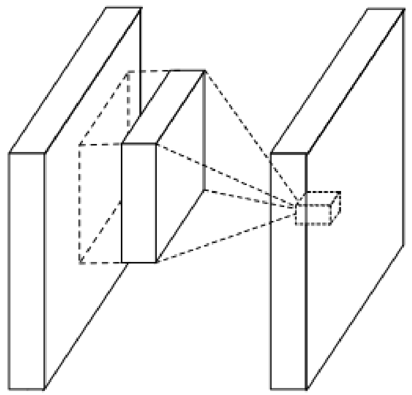
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

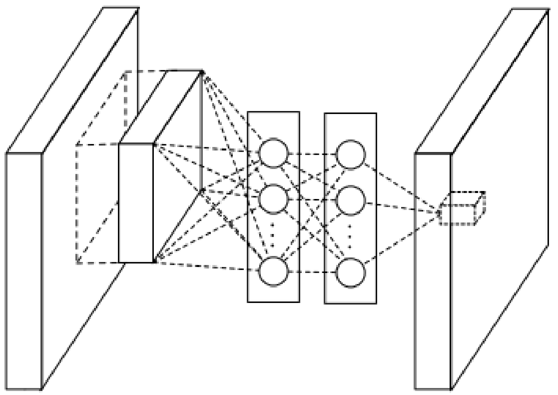
VGG-16 and VGG-19 (2014)

- Bigger version of AlexNet
- Instead of using bigger convolutions, use combinations of smaller convolutions (convolution 5x5 is 3x3 applied twice)
- 138 and 144 millions of parameters correspondingly

Network in network (2014)

- Use something more complicated, than just a convolutional layer

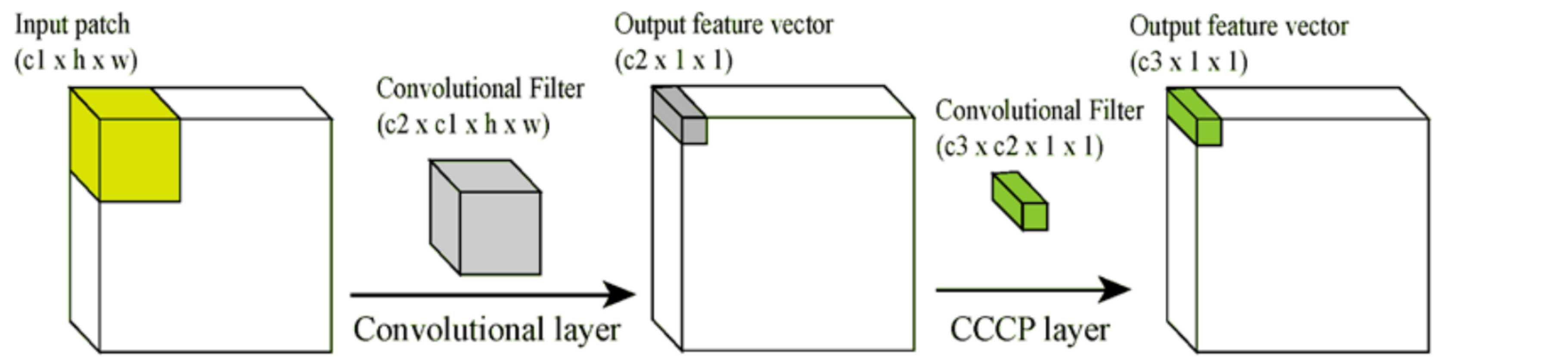

(a) Linear convolution layer


(b) Mlpconv layer

- But this is too costly!

1x1 convolution

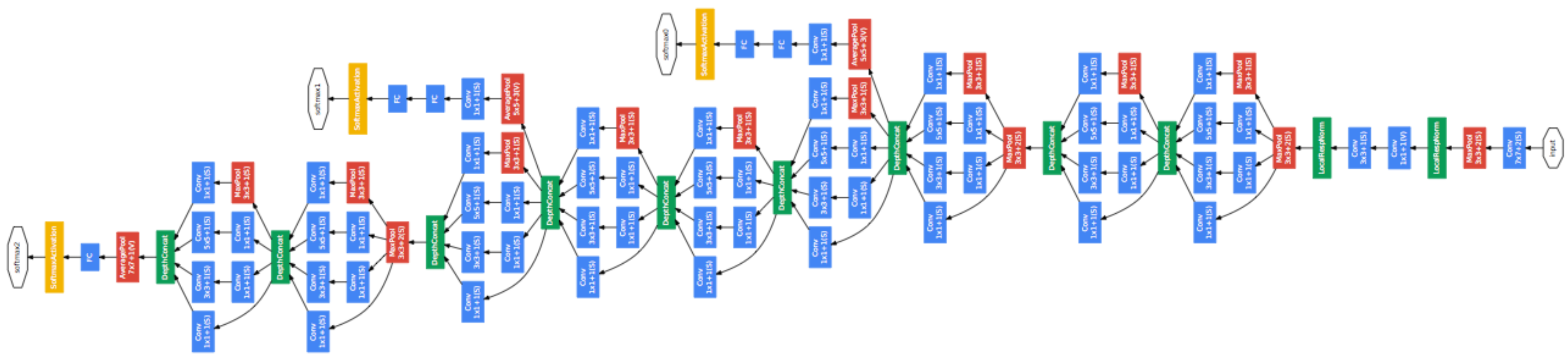
- We can use 1x1 convolution on top, which also helps to play with dimensionality



Efficient implementation of CCCP

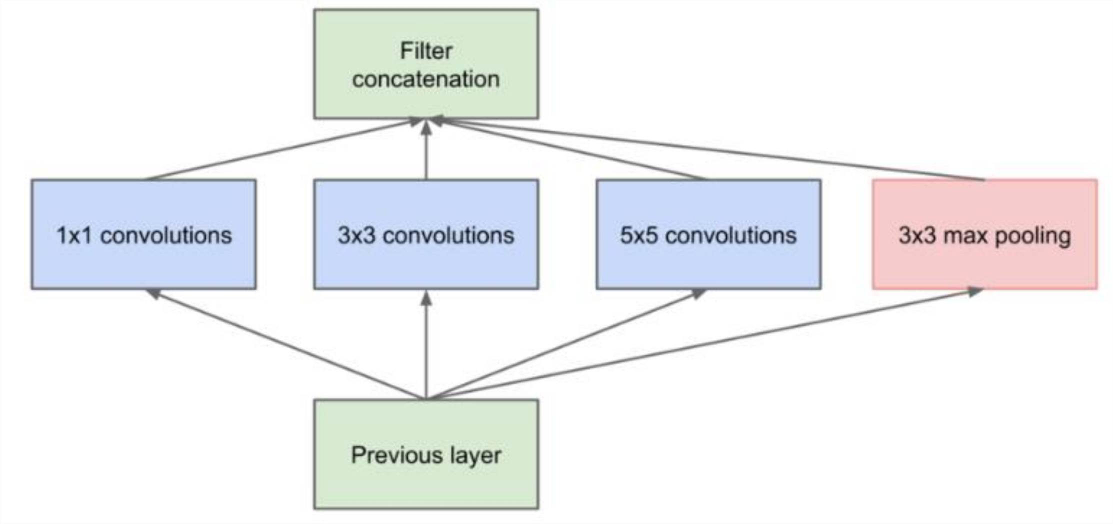
- It is called Cascaded Cross-Channel pooling (CCCP)

Inception (2014)



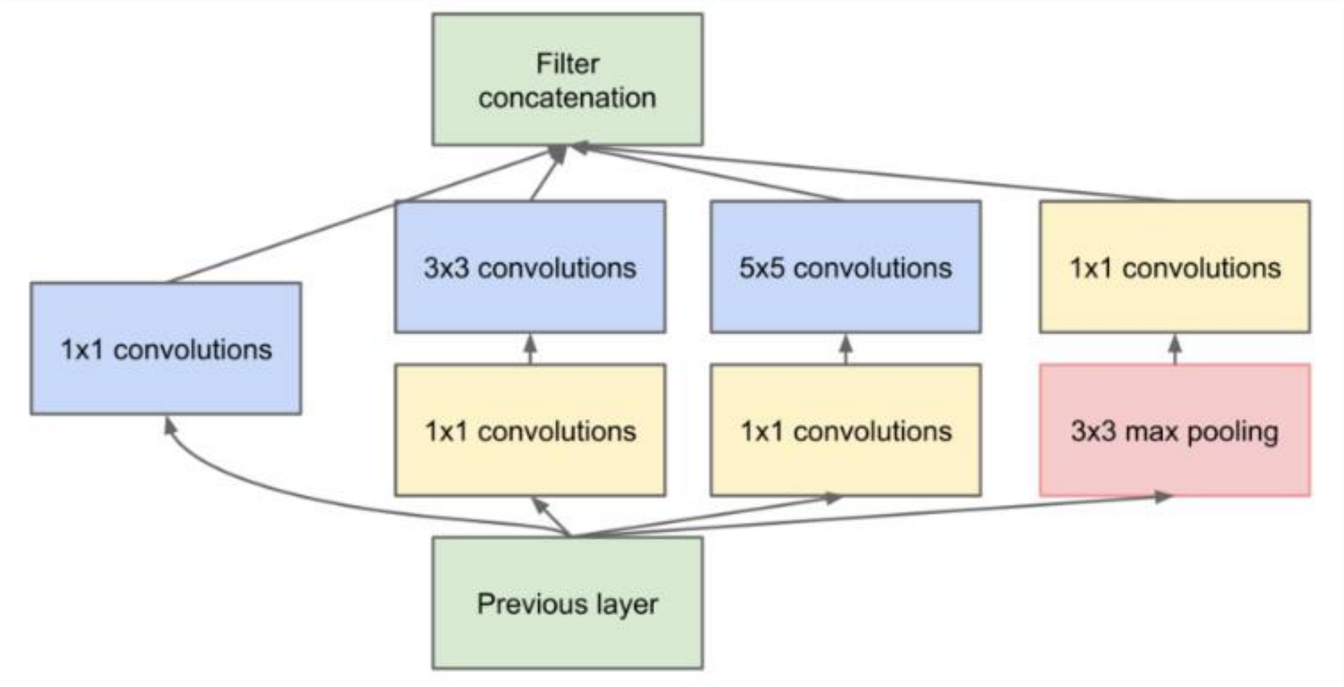
Inception module

- Idea: correlated neurons will be concentrated in small areas. To catch it, we can use 1x1 convolution.
- Also we try to search them in 3x3 and 5x5 areas

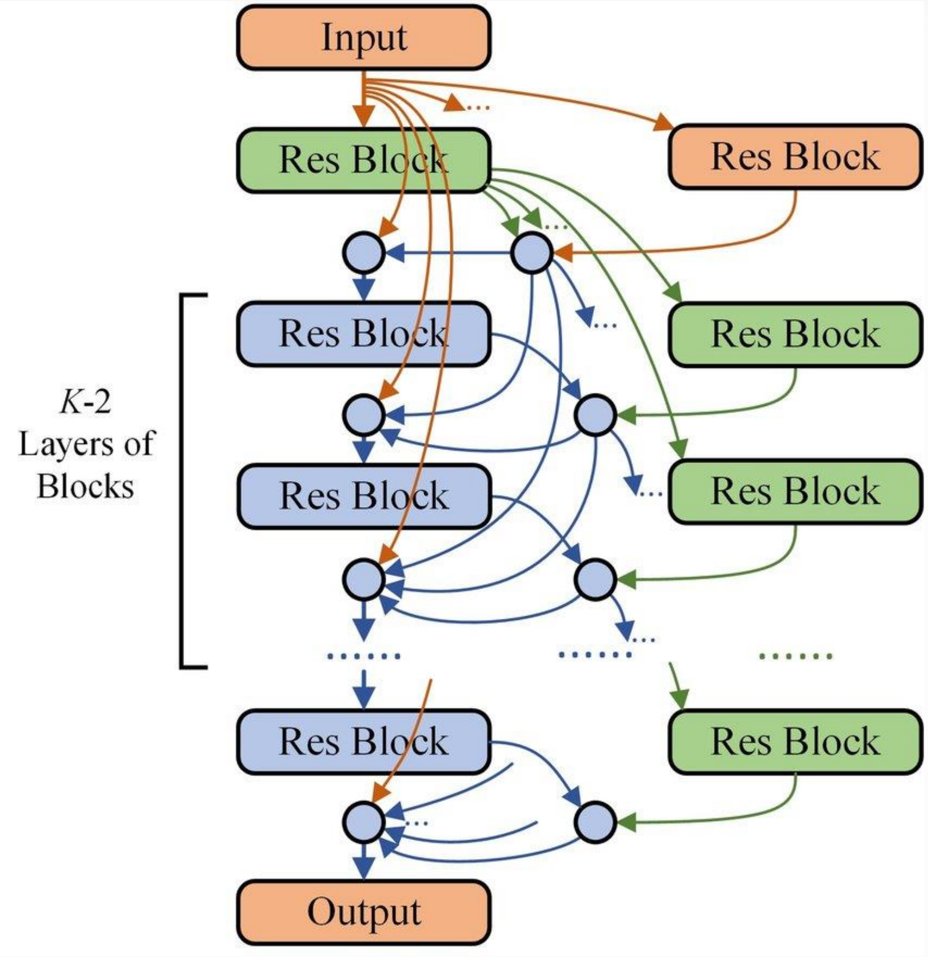


Inception module with dimensionality reduction

- We can also try to make tesnors to be of the same size



ResNet (2015)



Input
Res Block
Res Block
Res Block
Res Block
K-2
Layers of
Blocks
Res Block
Res Block
Res Block
Res Block
Output

Skip layers

- Additional layers not always help
- Adding skip layers may help

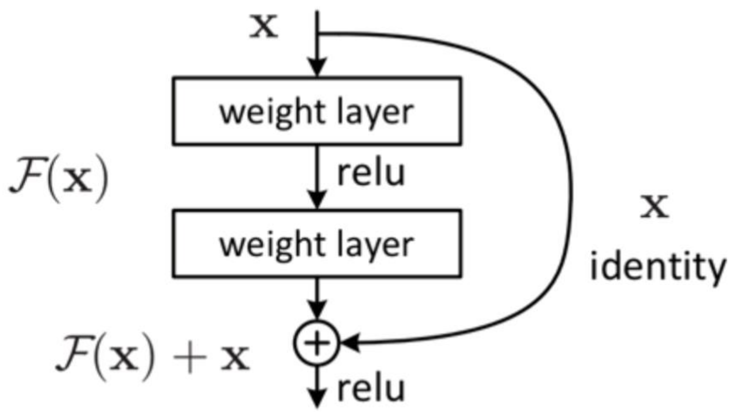


Figure 2. Residual learning: a building block.

$\mathcal{H}(\mathbf{x})$ is the true function we want to learn

Let's pretend we want to learn

$$\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$$

instead.

The original function is then

$$\mathcal{F}(\mathbf{x}) + \mathbf{x}$$

Skip layers

- Additional layers not always help
- Adding skip layers may help

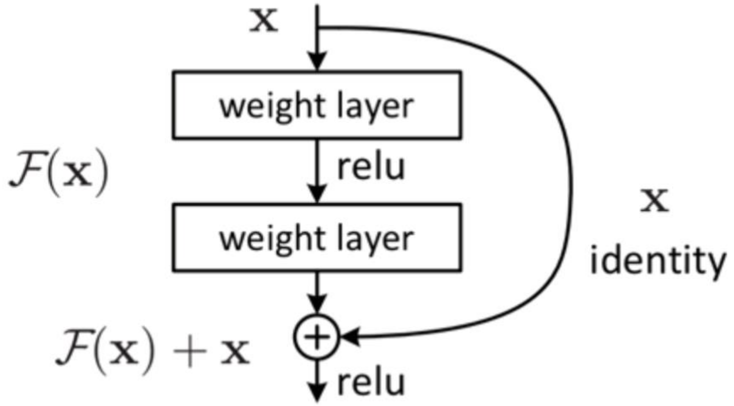


Figure 2. Residual learning: a building block.

$\mathcal{H}(\mathbf{x})$ is the true function we want to learn

Let's pretend we want to learn

$$\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$$

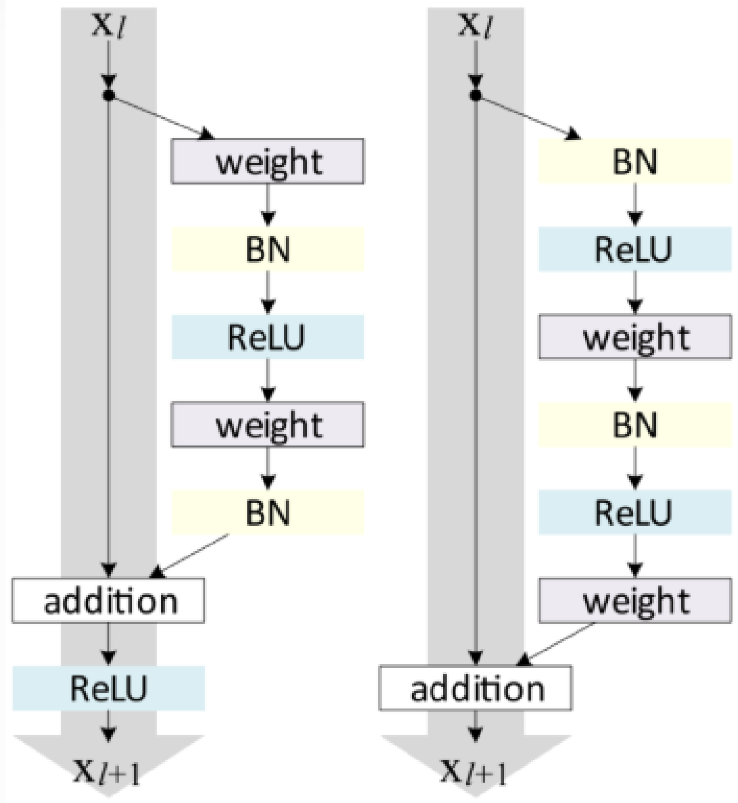
instead.

The original function is then

$$\mathcal{F}(\mathbf{x}) + \mathbf{x}$$

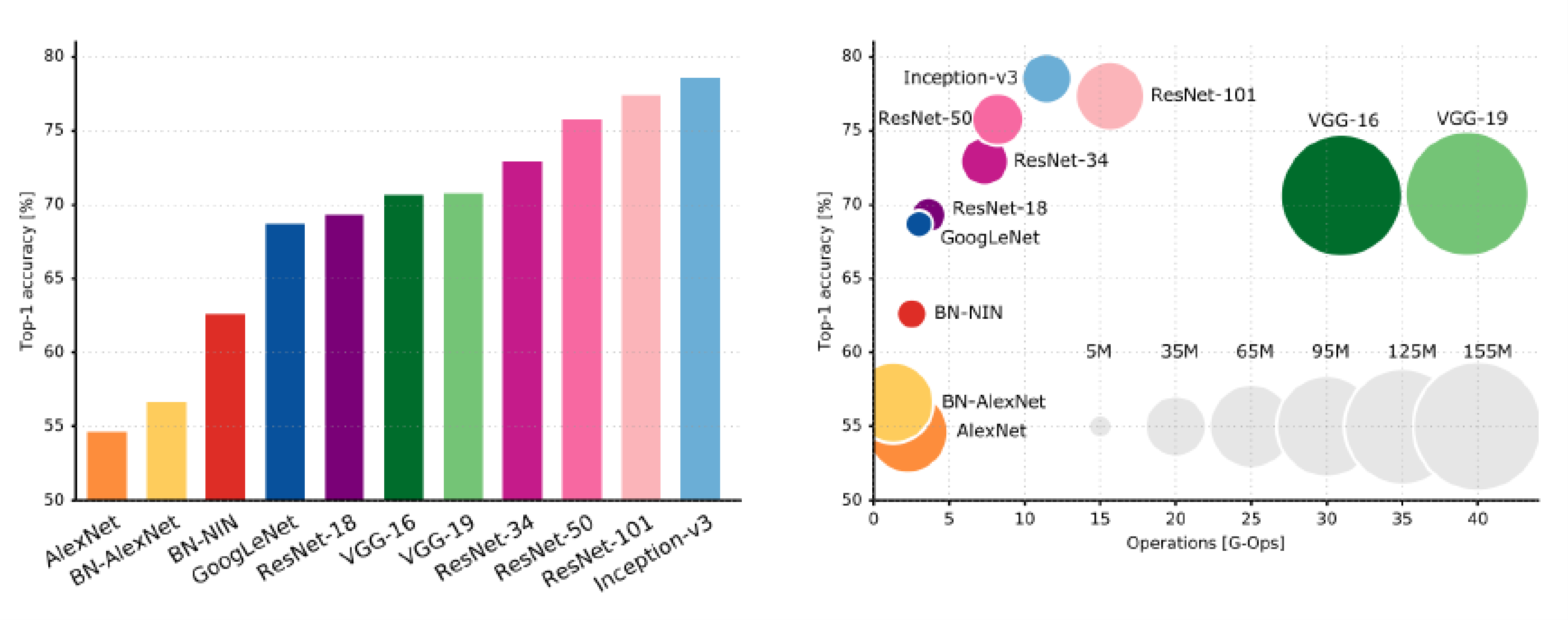
Better skip layers

- Idea is just to add original input after ReLU is applied



(a) original (b) proposed

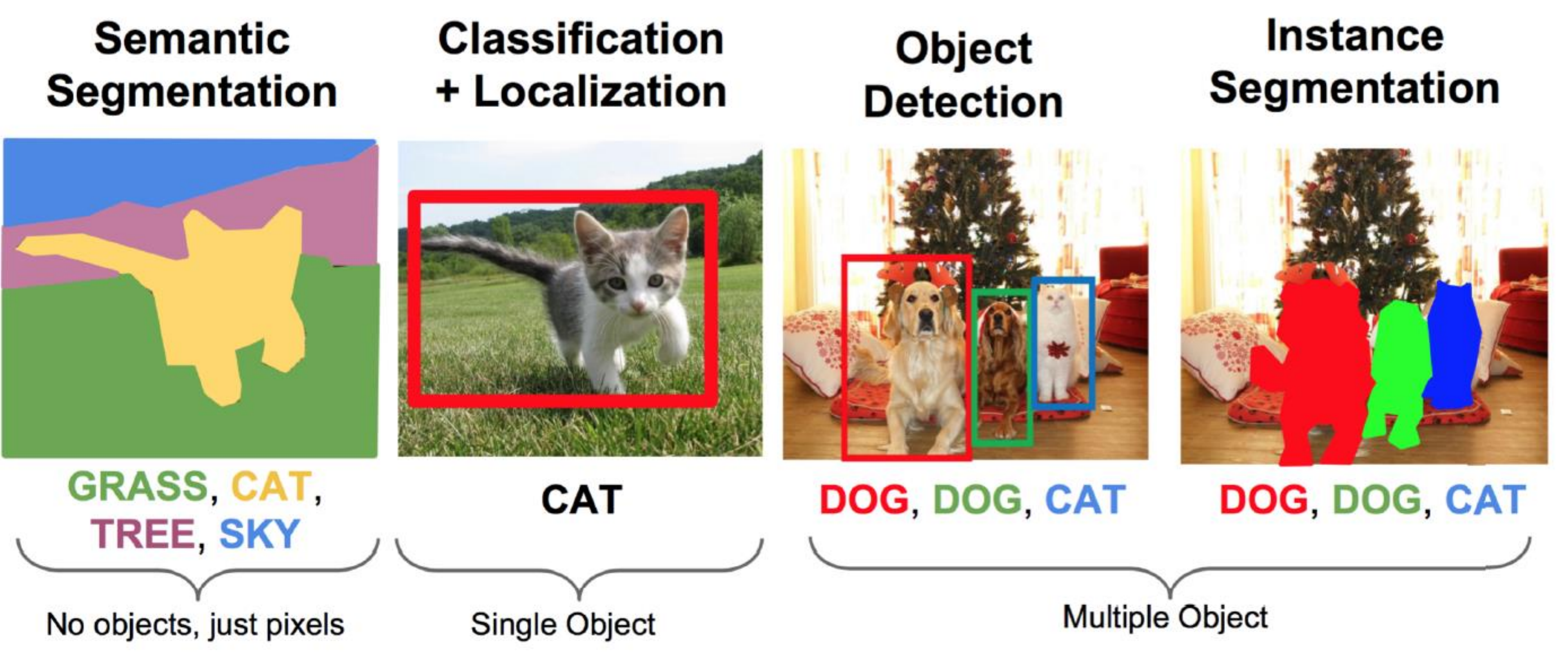
Network comparison



Lecture plan

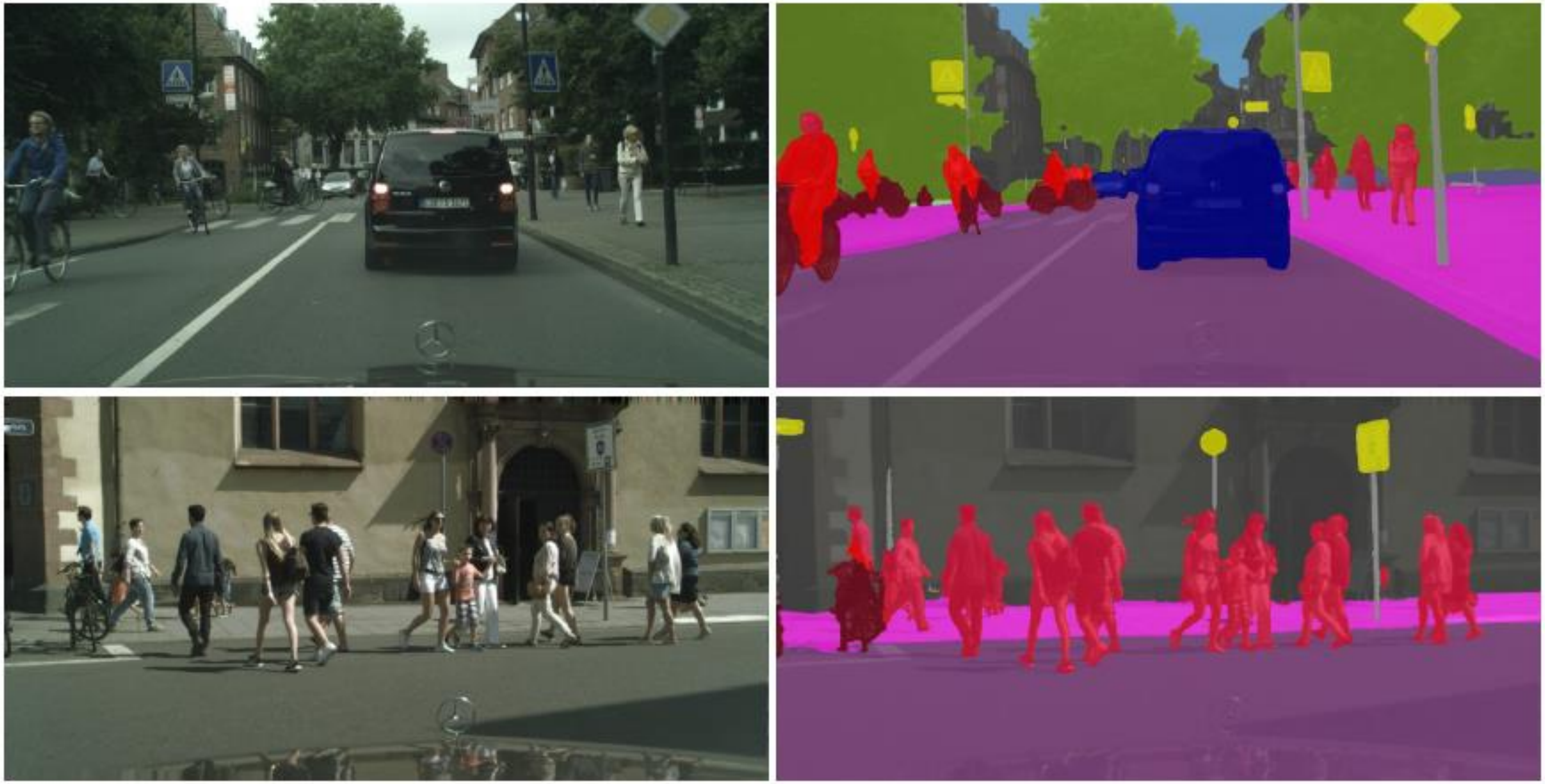
- Brief overview of ImageNet
- Earlier approaches in computer vision
- Convolutional neural networks
- Tensors
- Deconvolution and visualization of neurons
- Architecture overview
- Computer vision problems

CV tasks

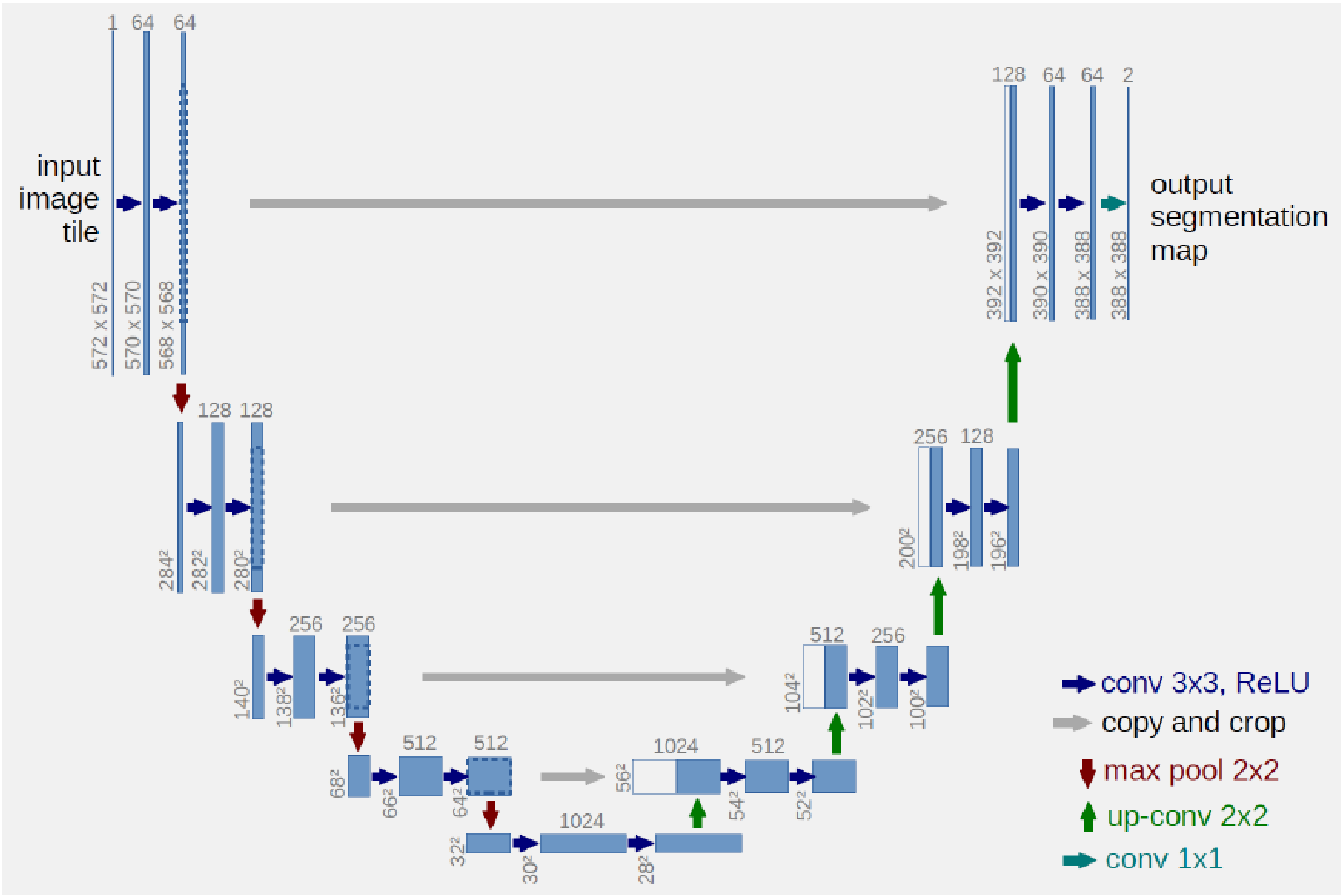


Semantic Segmentation
Classification + Localization
Object Detection
Instance Segmentation
GRASS, CAT, TREE, SKY
CAT
DOG, DOG, CAT
DOG, DOG, CAT
No objects, just pixels
Single Object
Multiple Object

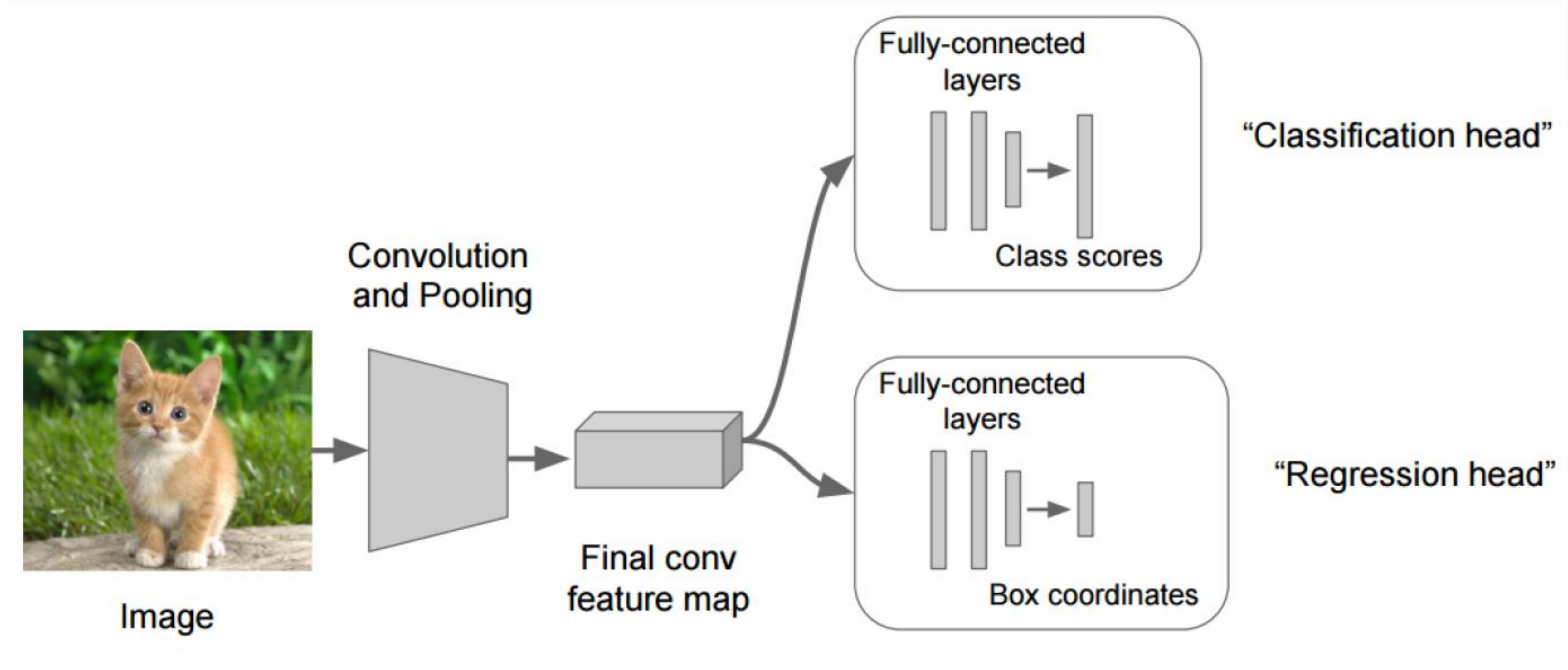
Semantic segmentation



Semantic segmentation via U-Net

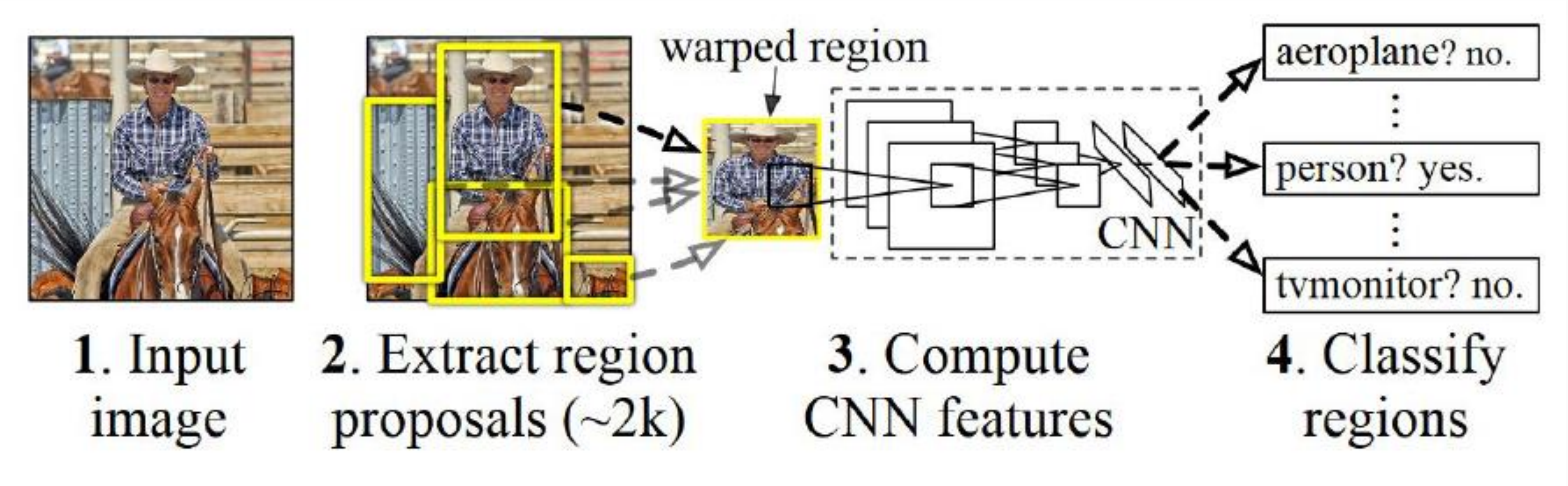


Object localization



Object detection. Pascal VOC

Detection via R-CNN



Detection via YOLO