Lecture 9

Convolutional neural networks

Information Systems
(Machine Learning)
Andrey Filchenkov

22.11.2018

Lecture plan

- Brief overview of ImageNet
- Earlier approaches in computer vision
- Convolutional neural networks
- 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”
- Slides are available online: **goo.gl/BspjhF**

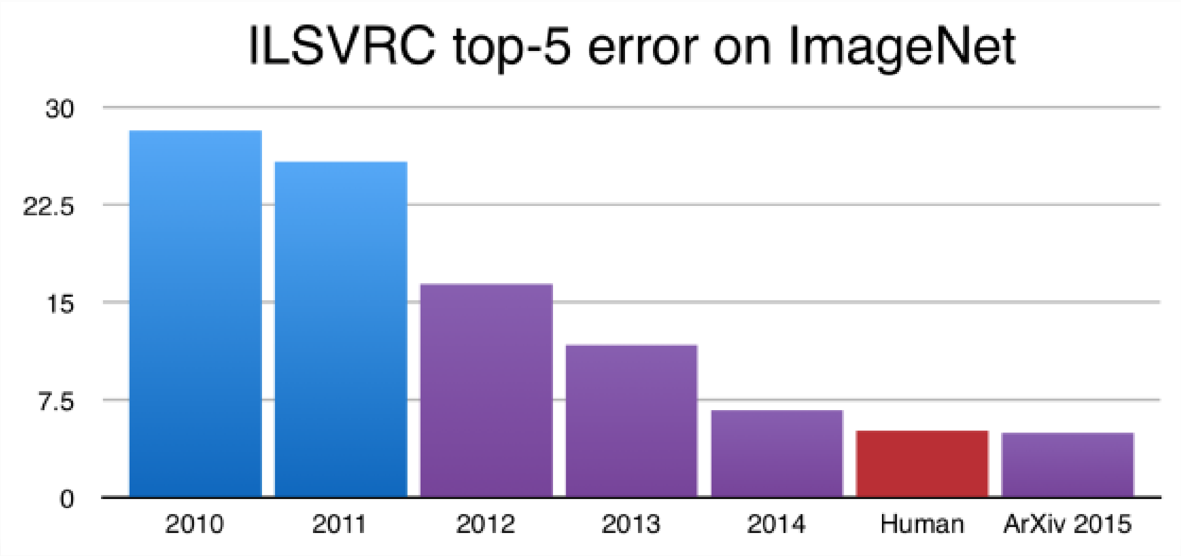
Lecture plan

- Brief overview of ImageNet
- Earlier approaches in computer vision
- Convolutional neural networks
- 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.

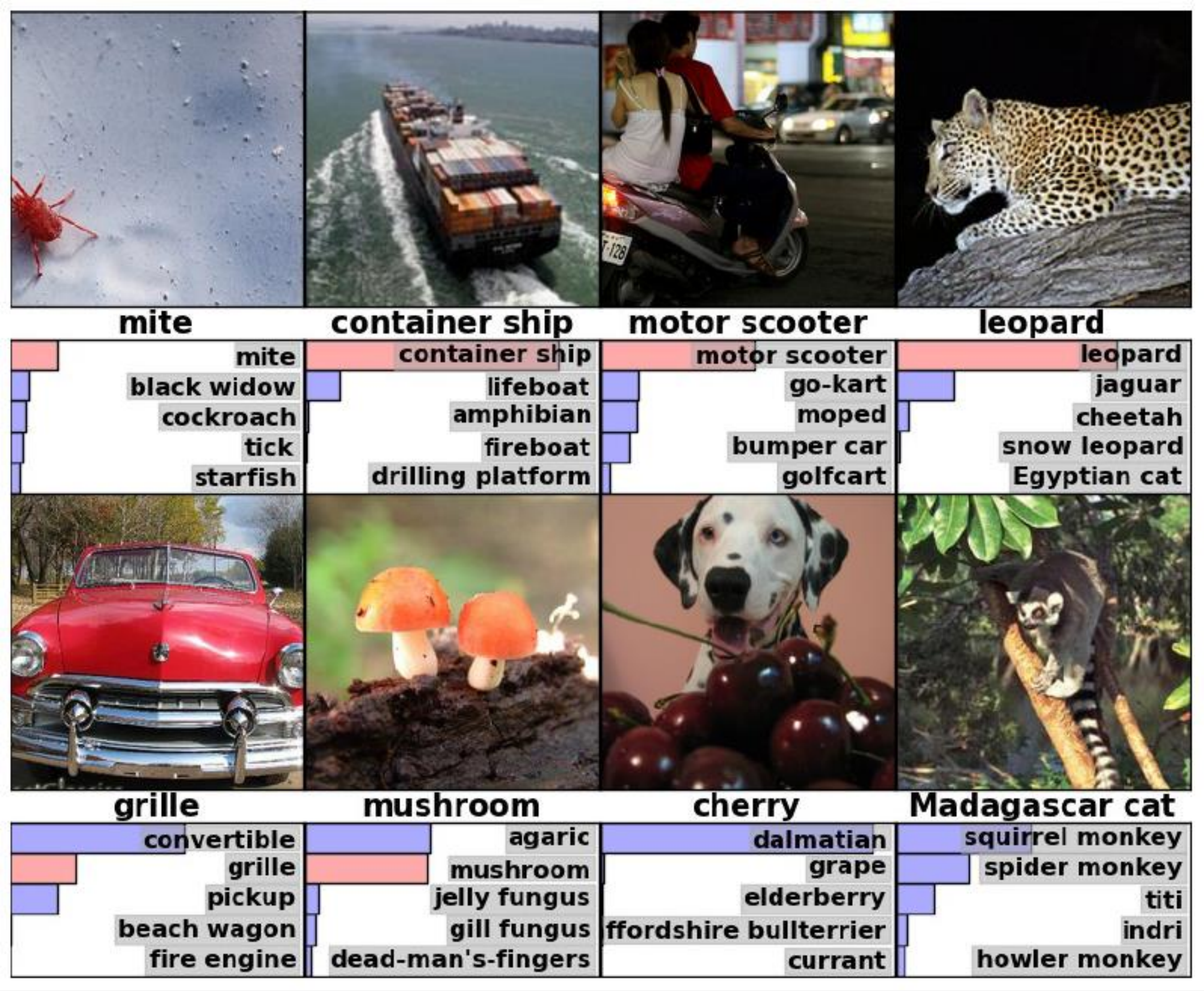


Imagenet Challenge

IMAGENET

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

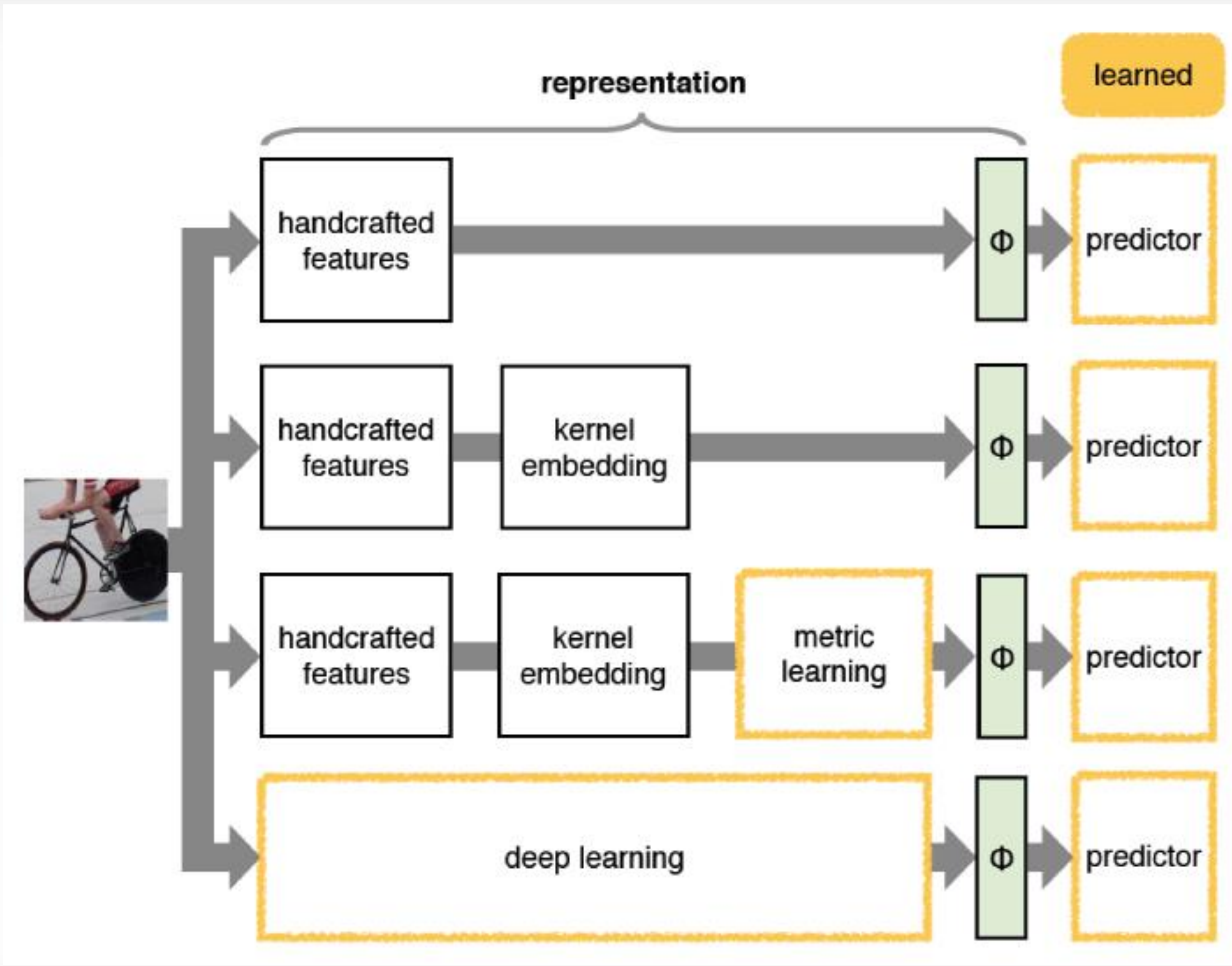
Examples of images



Lecture plan

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

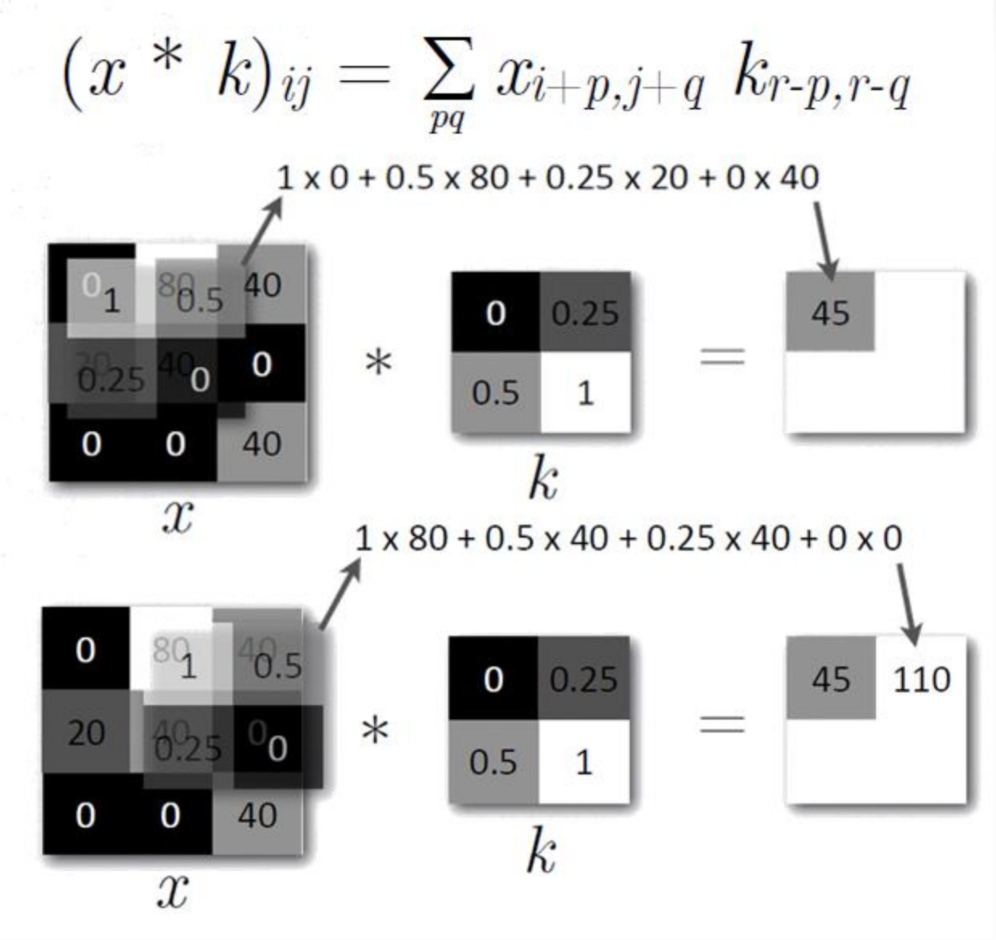
Short history of computer vision



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

Discrete kernel



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	

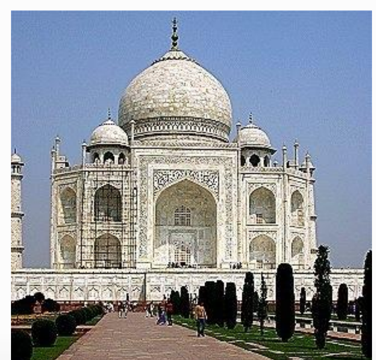
=

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

=



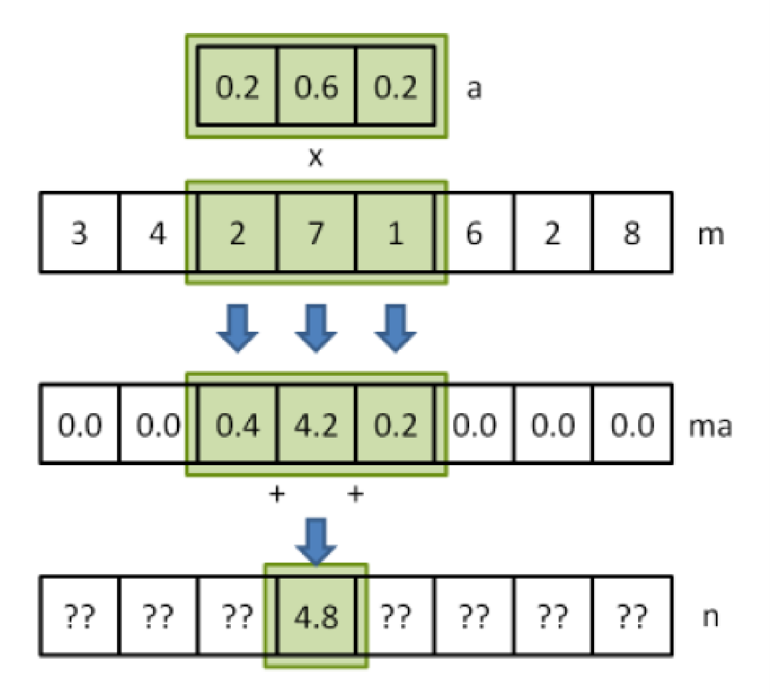
sharpen

Lecture plan

- Brief overview of ImageNet
- Earlier approaches in computer vision
- Convolutional neural networks
- 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

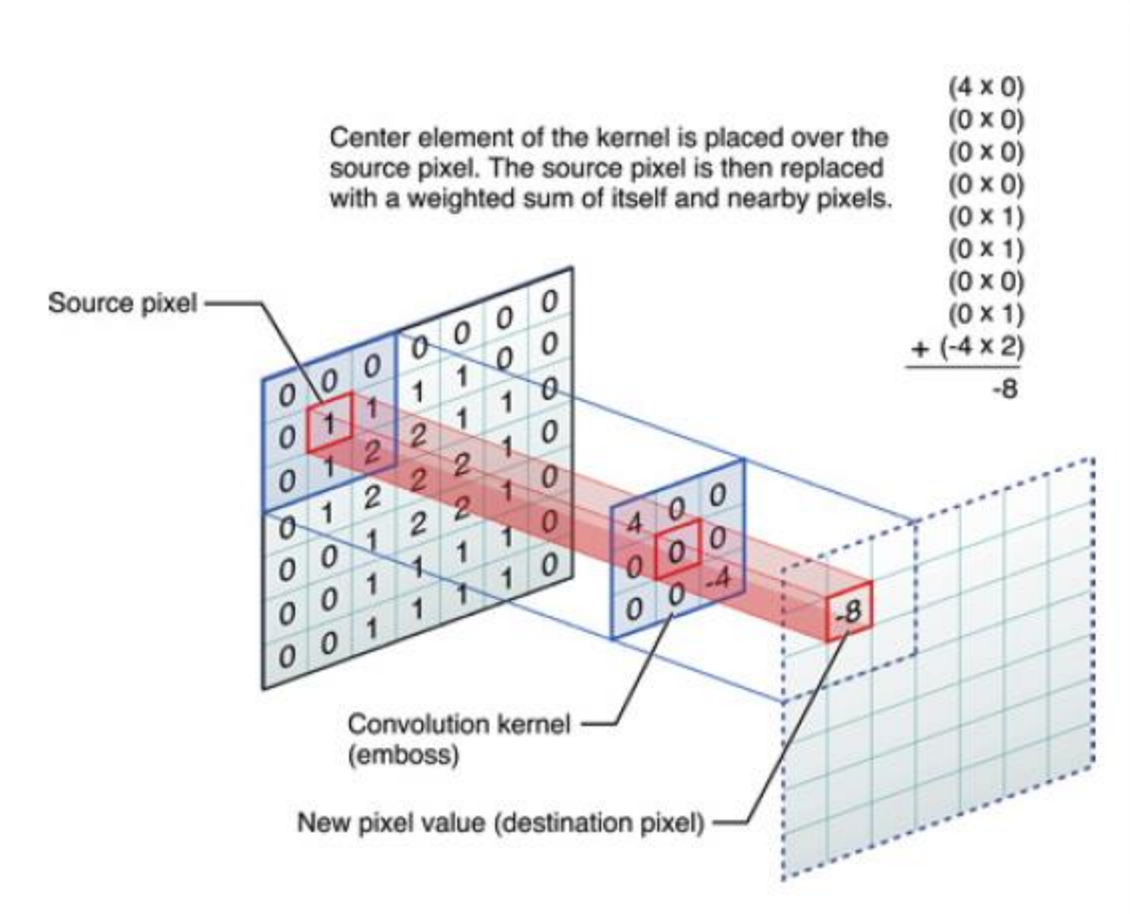
B	A	**A**	**B**	**C**	C	B

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

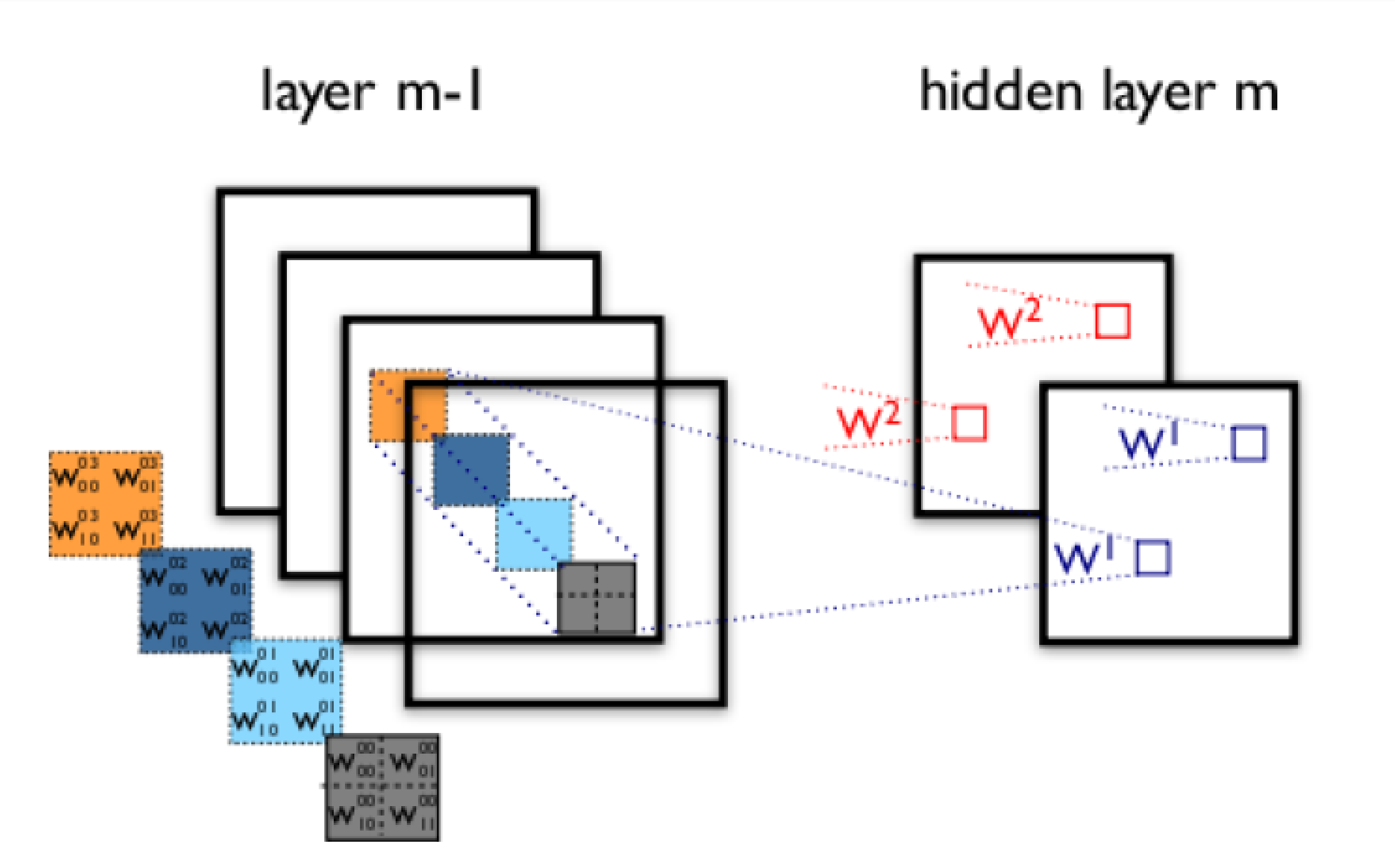
Cyclic shift

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

2-D convolution

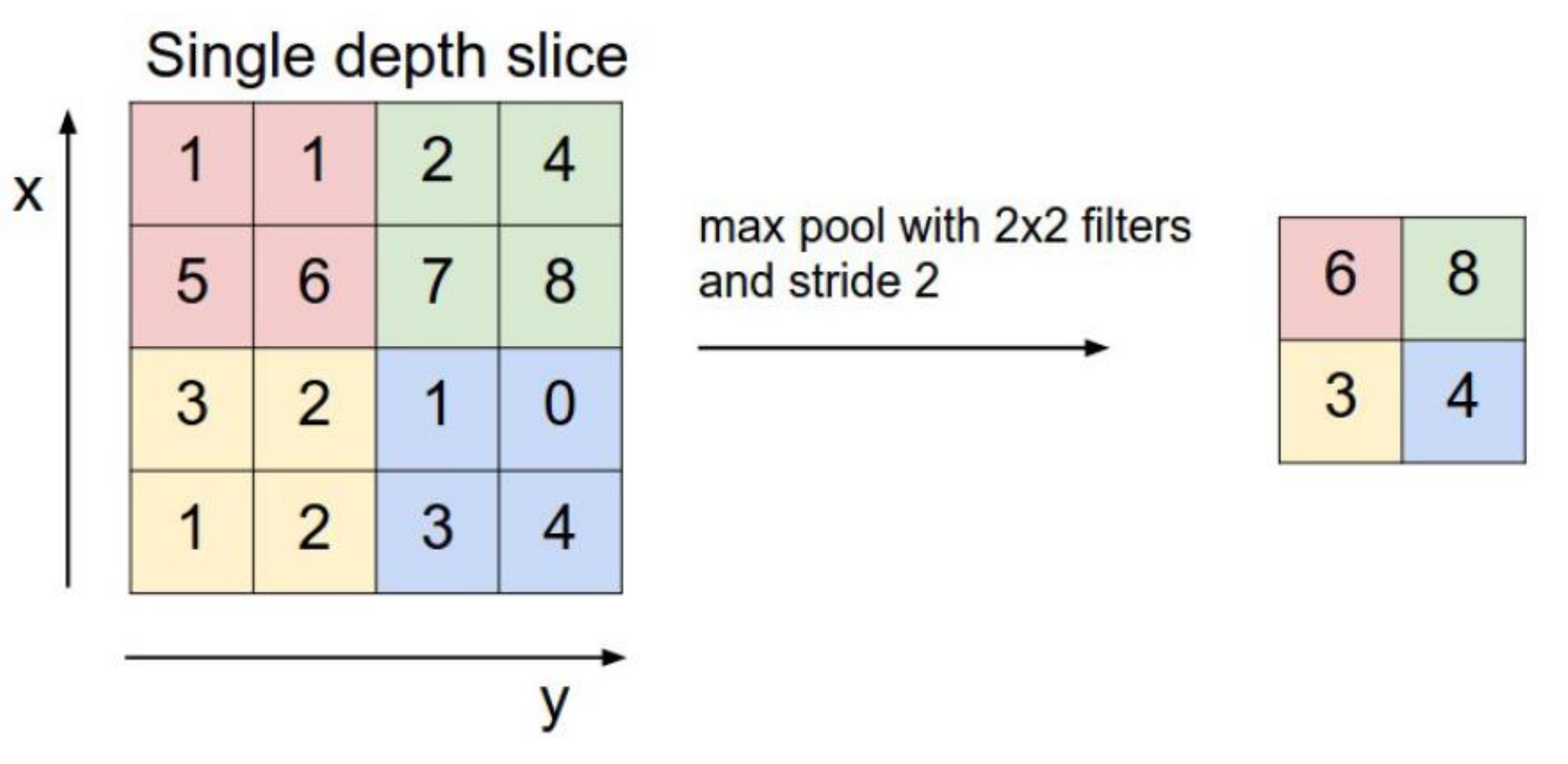


Convolutional tensors

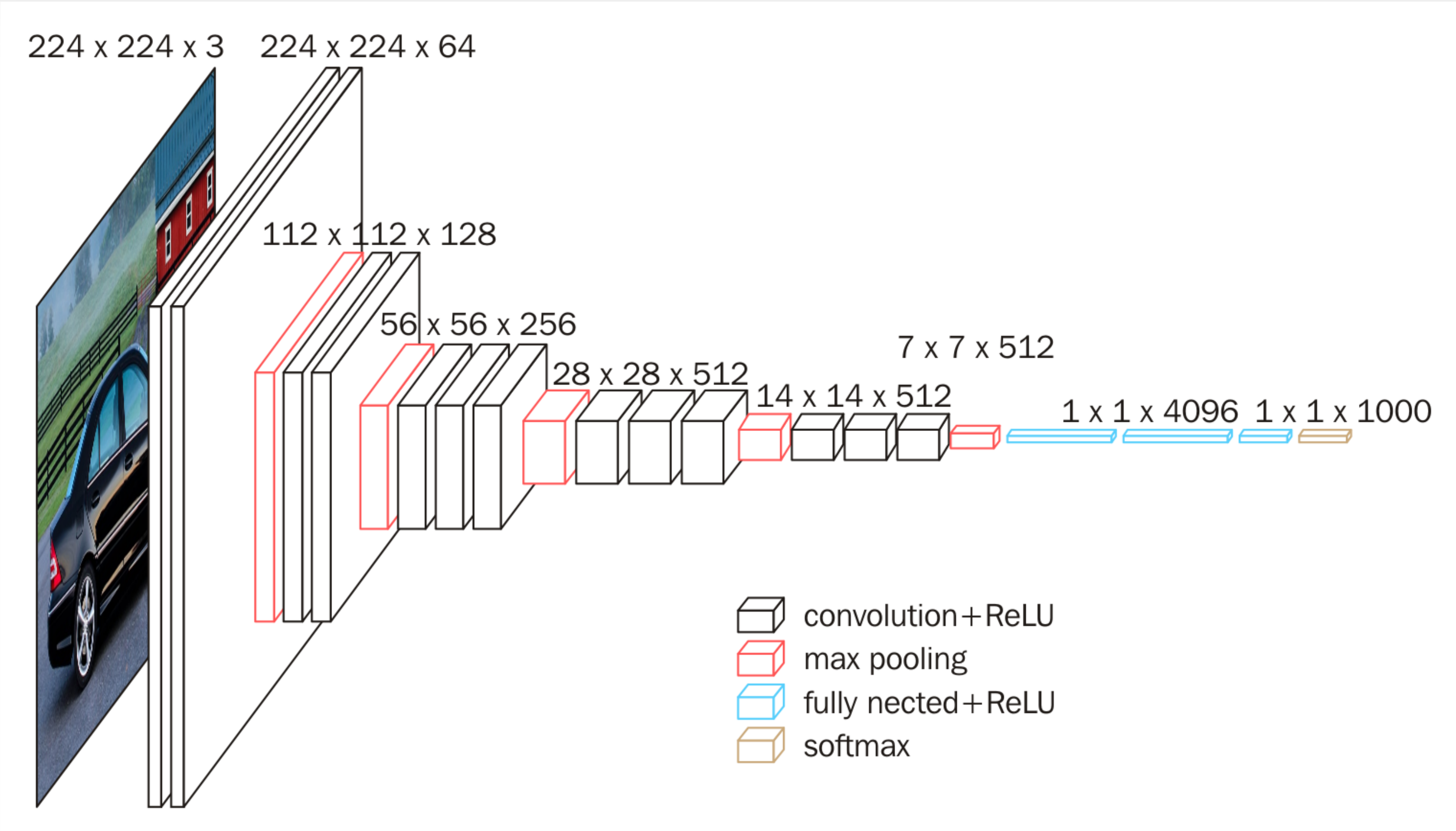


Pooling

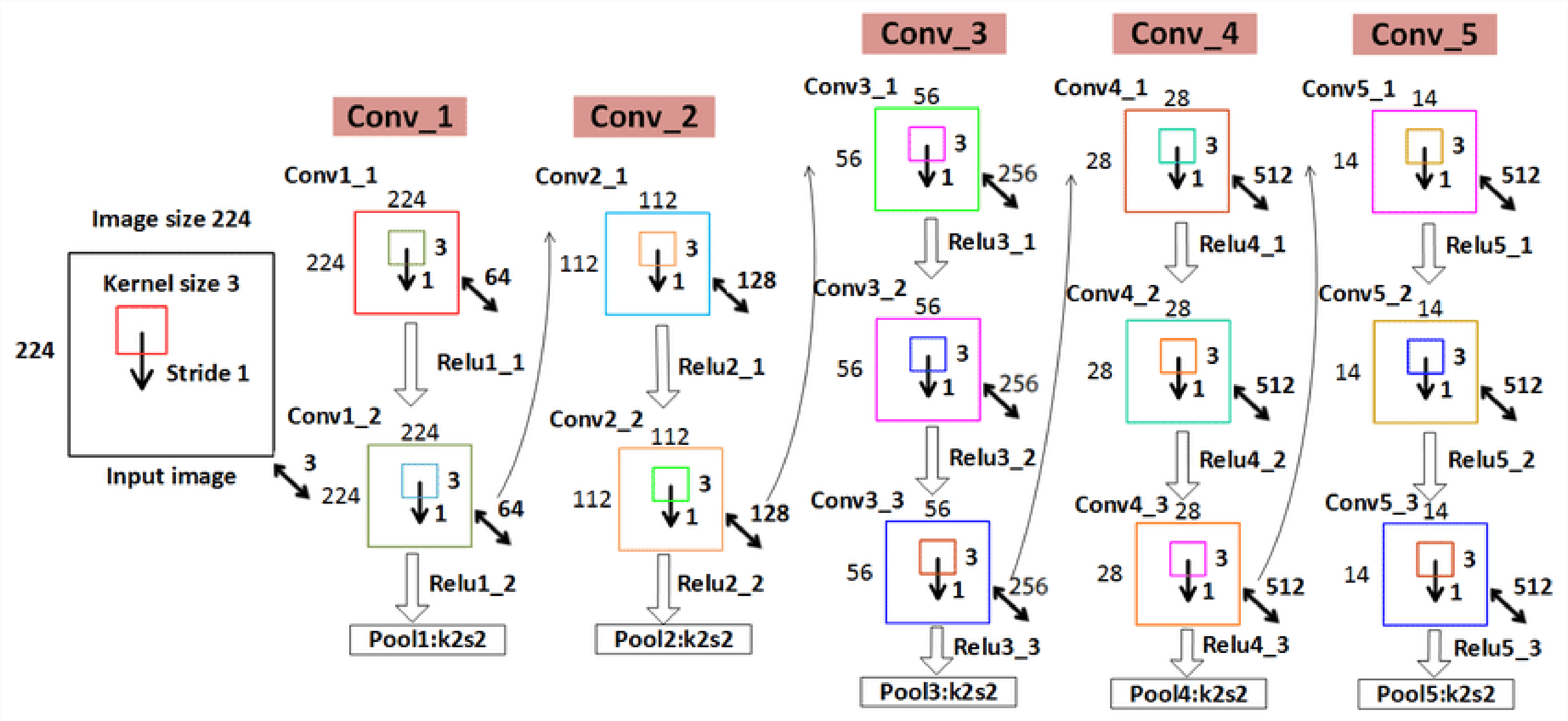
- Pooling is used to reduce dimensionality



VGG-16 conceptual scheme



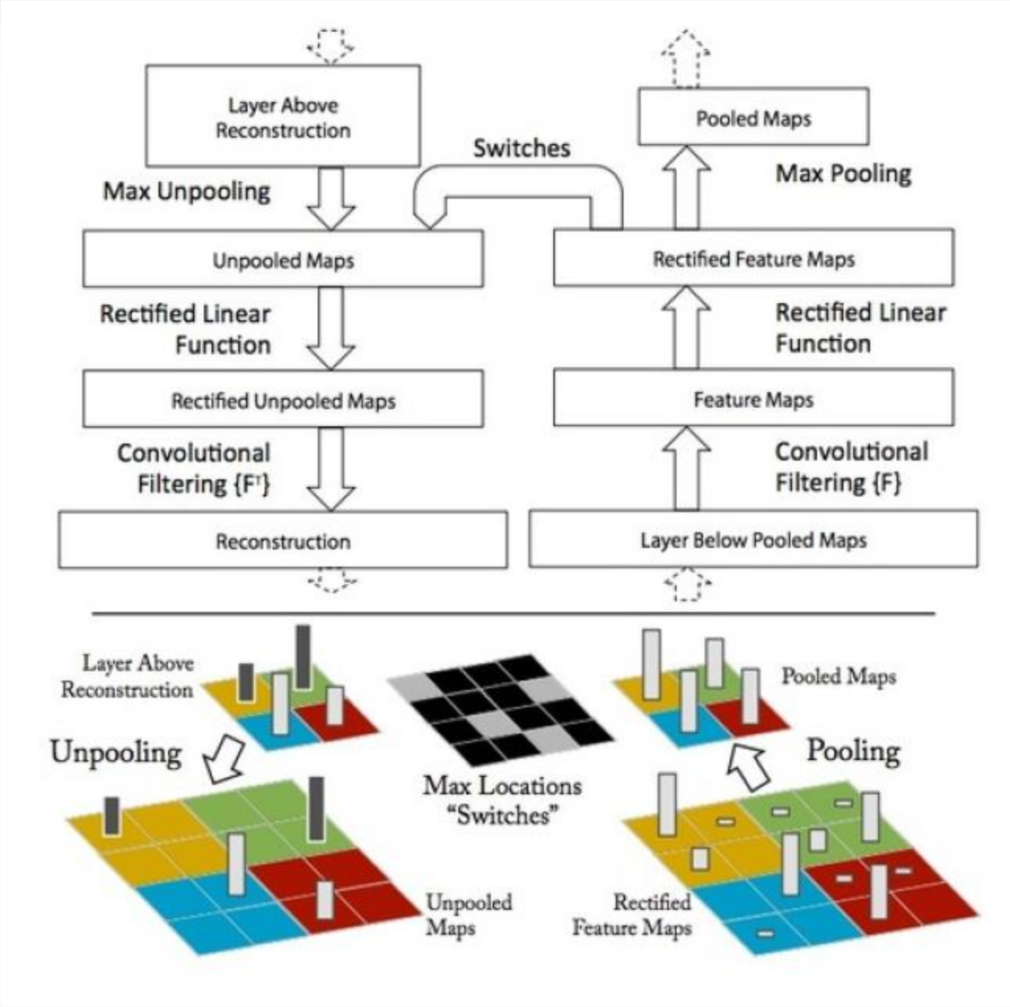
VGG-16 technical scheme



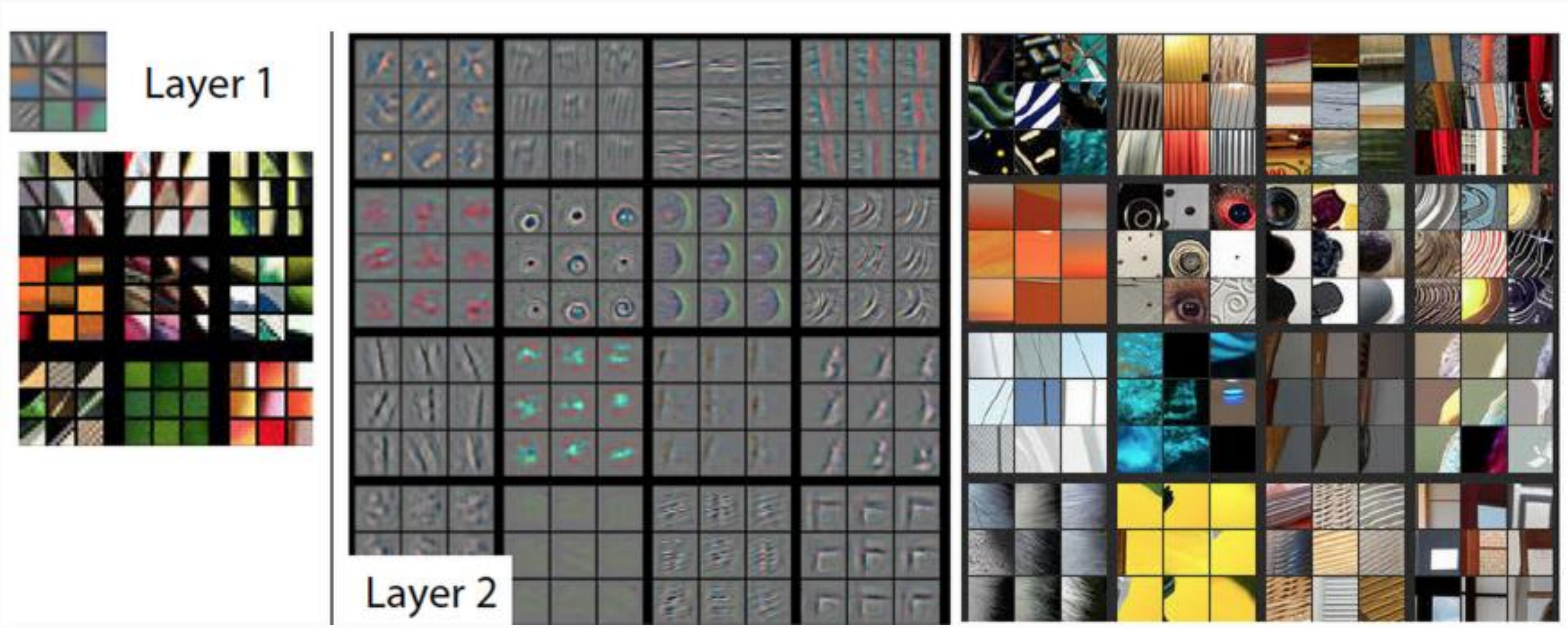
Lecture plan

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

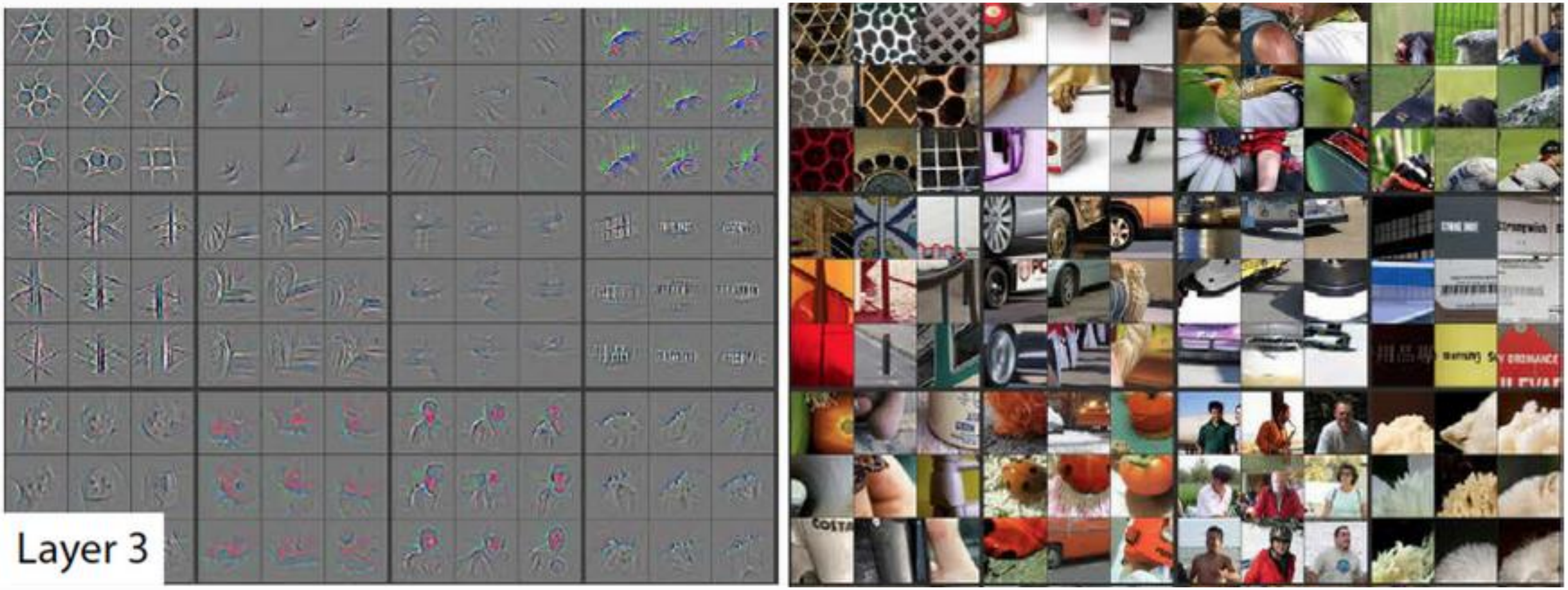
Deconvolution neural network



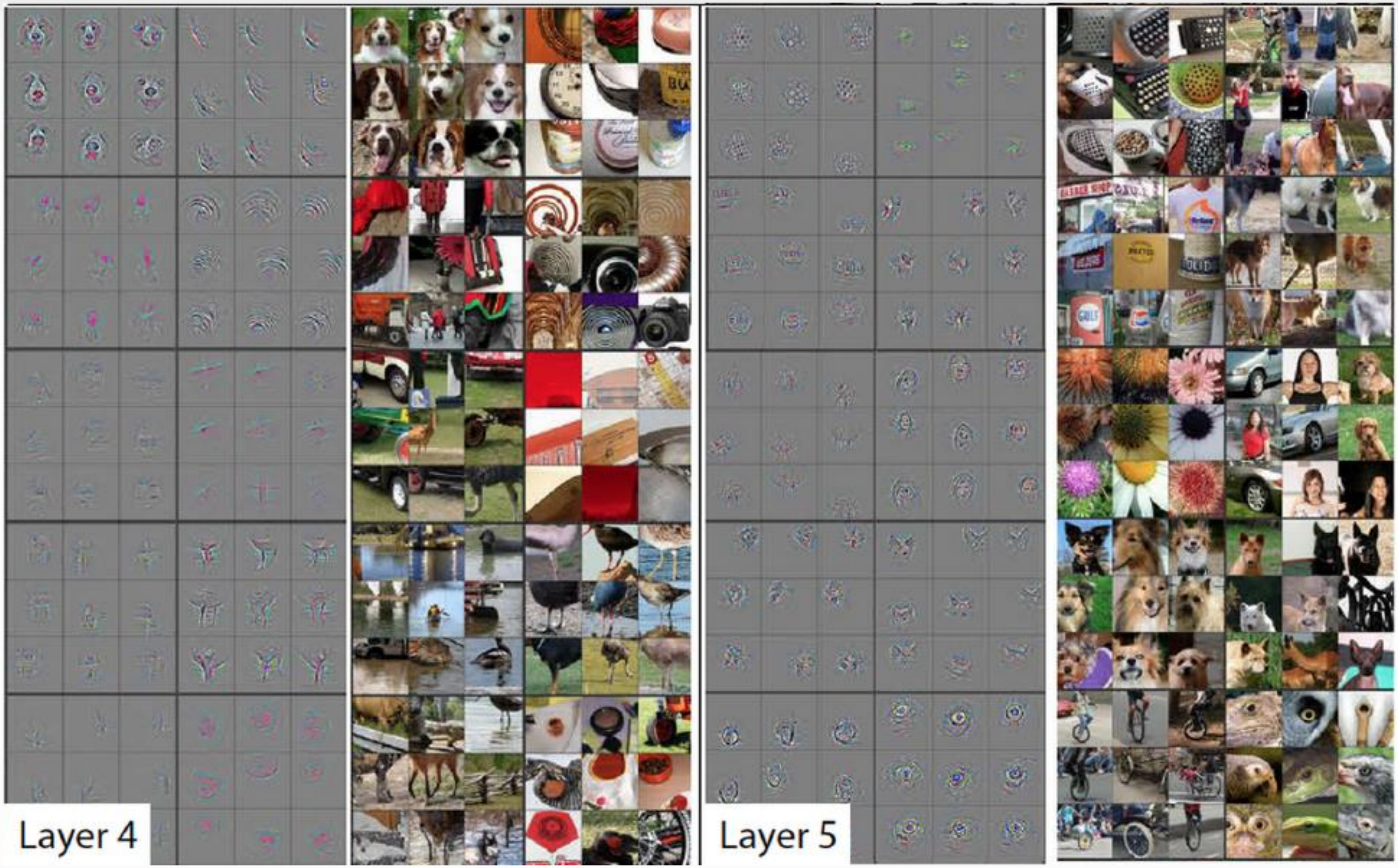
Visualization of neuron activation



Visualization of neuron activation



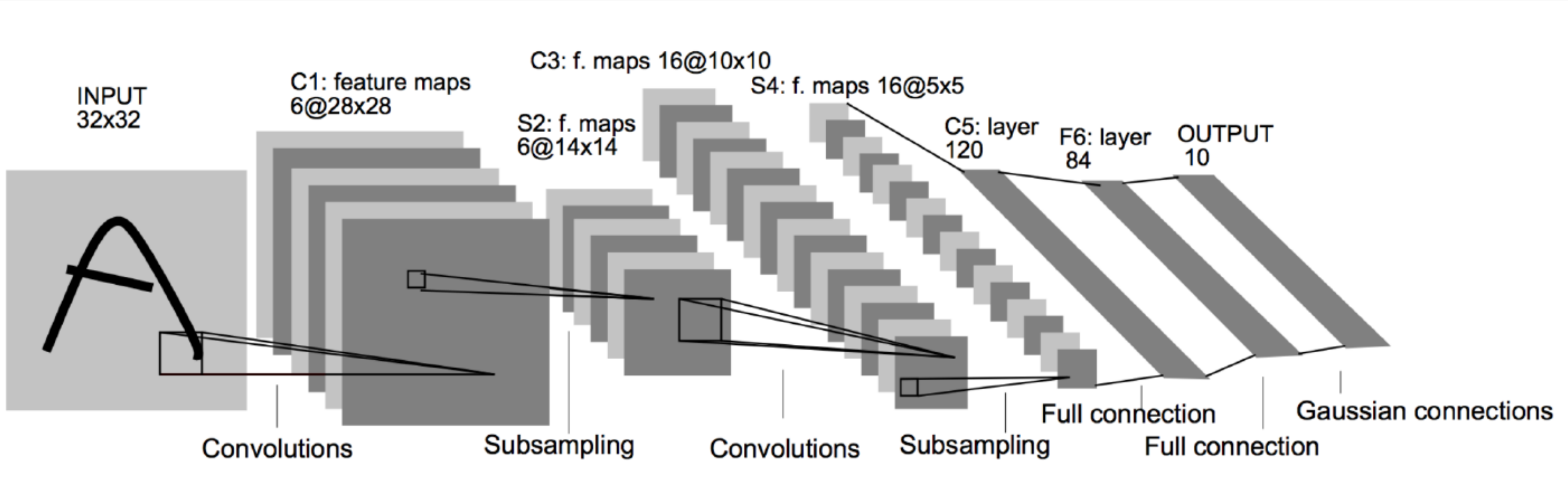
Visualization of neuron activation



Lecture plan

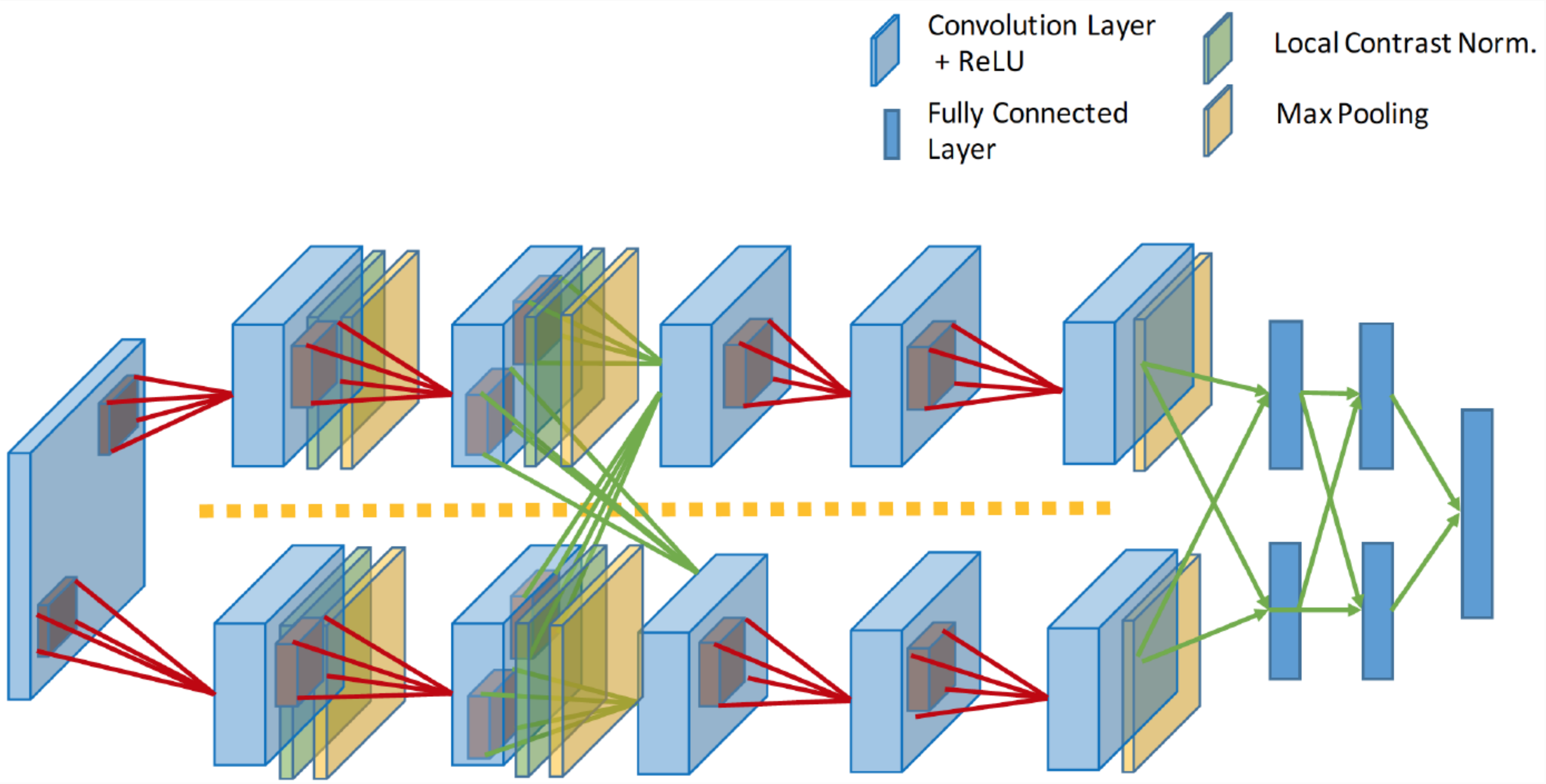
- Brief overview of ImageNet
- Earlier approaches in computer vision
- Convolutional neural networks
- Deconvolution and visualization of neurons
- Architecture overview
- Computer vision problems

LeNet



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

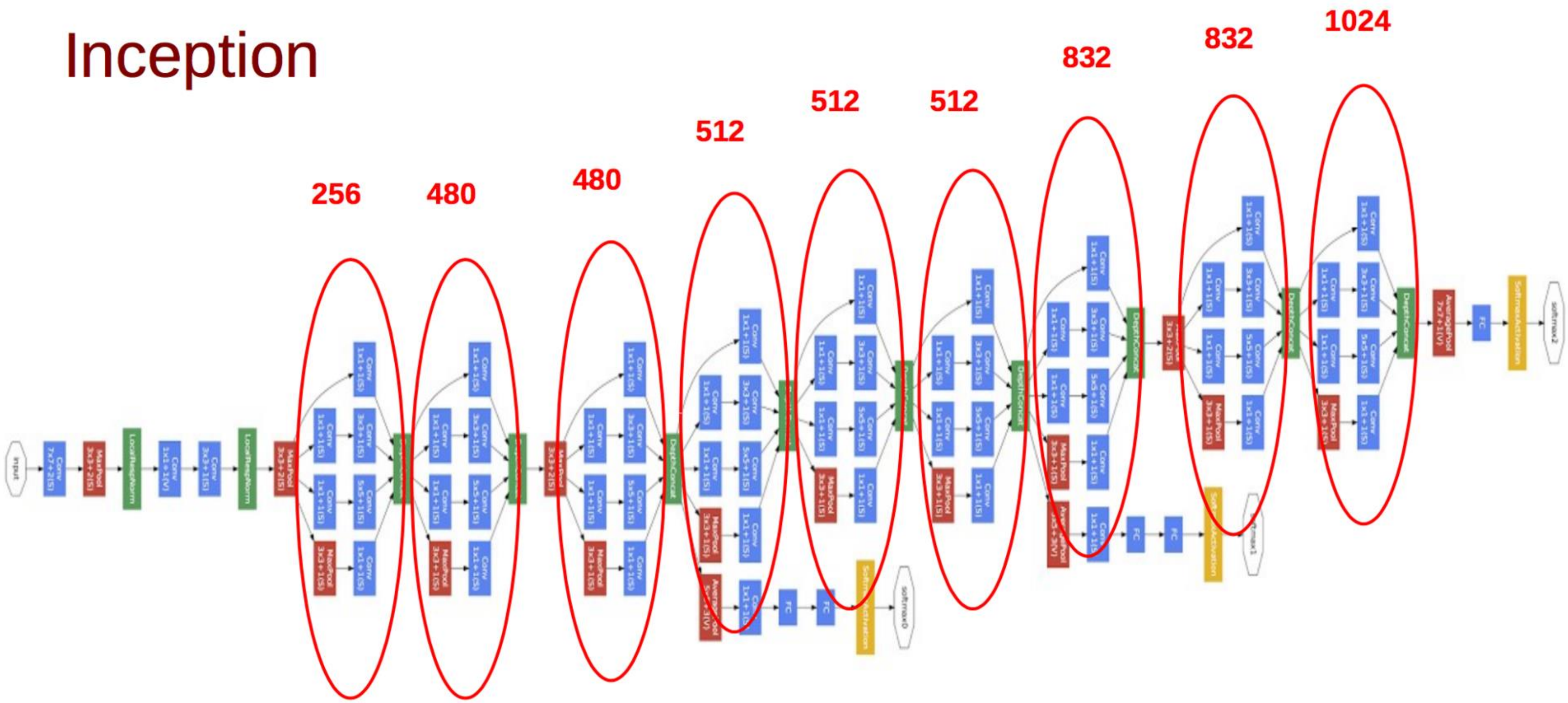

Convolution Layer + ReLU
Local Contrast Norm.
Fully Connected Layer
Max Pooling

VGG-16

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					

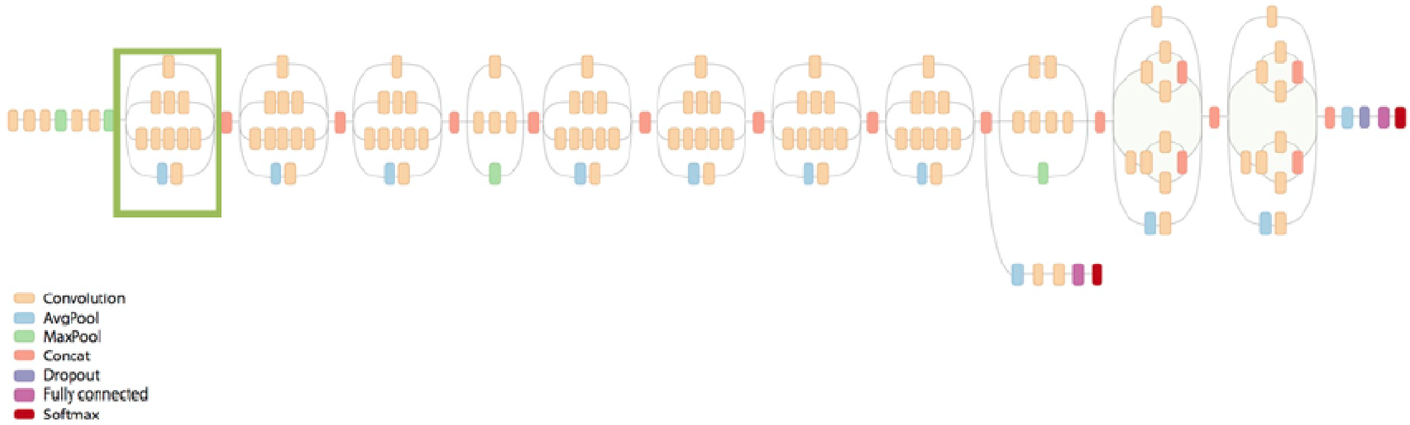
Inception

Inception

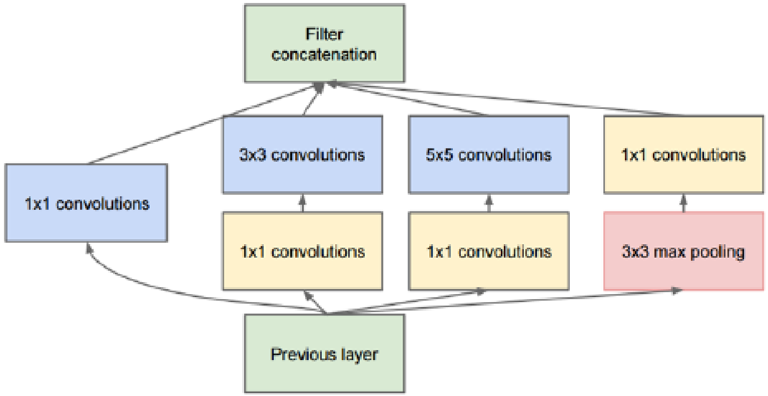


Width of **inception modules** ranges from 256 filters (in early modules) to 1024 in top inception modules.

Inception aka GoogleNet



Green box shows parallel region of GoogLeNet



ResNet (1/2)

- Additional layers not always help

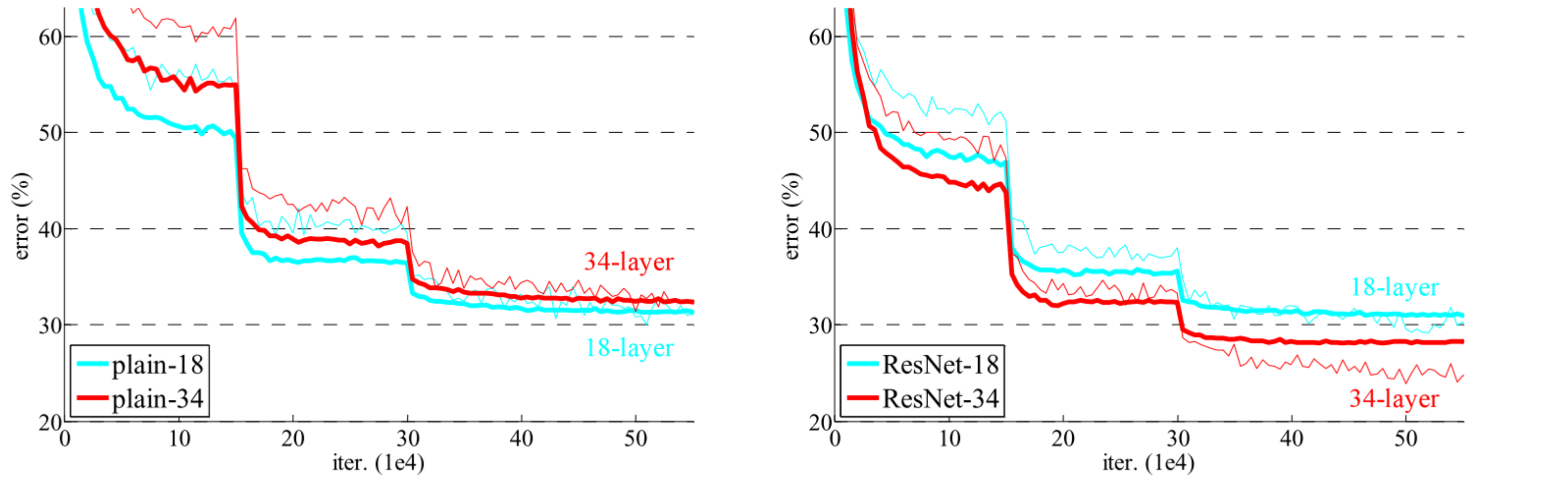


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

ResNet (2/2)

- Adding skip layers may help

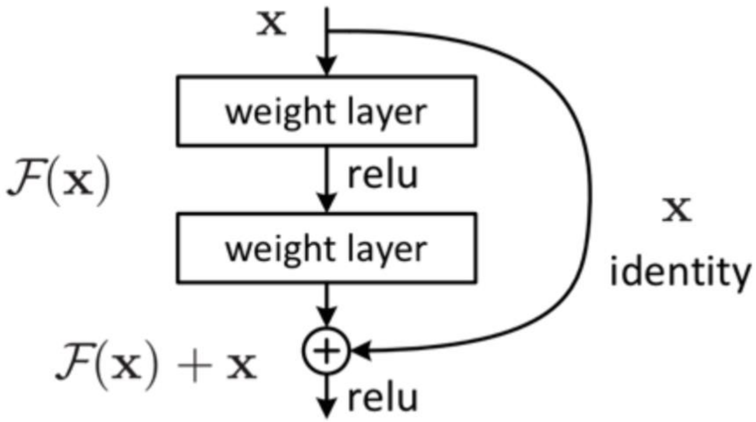


Figure 2. Residual learning: a building block.

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

Let's pretend we want to learn

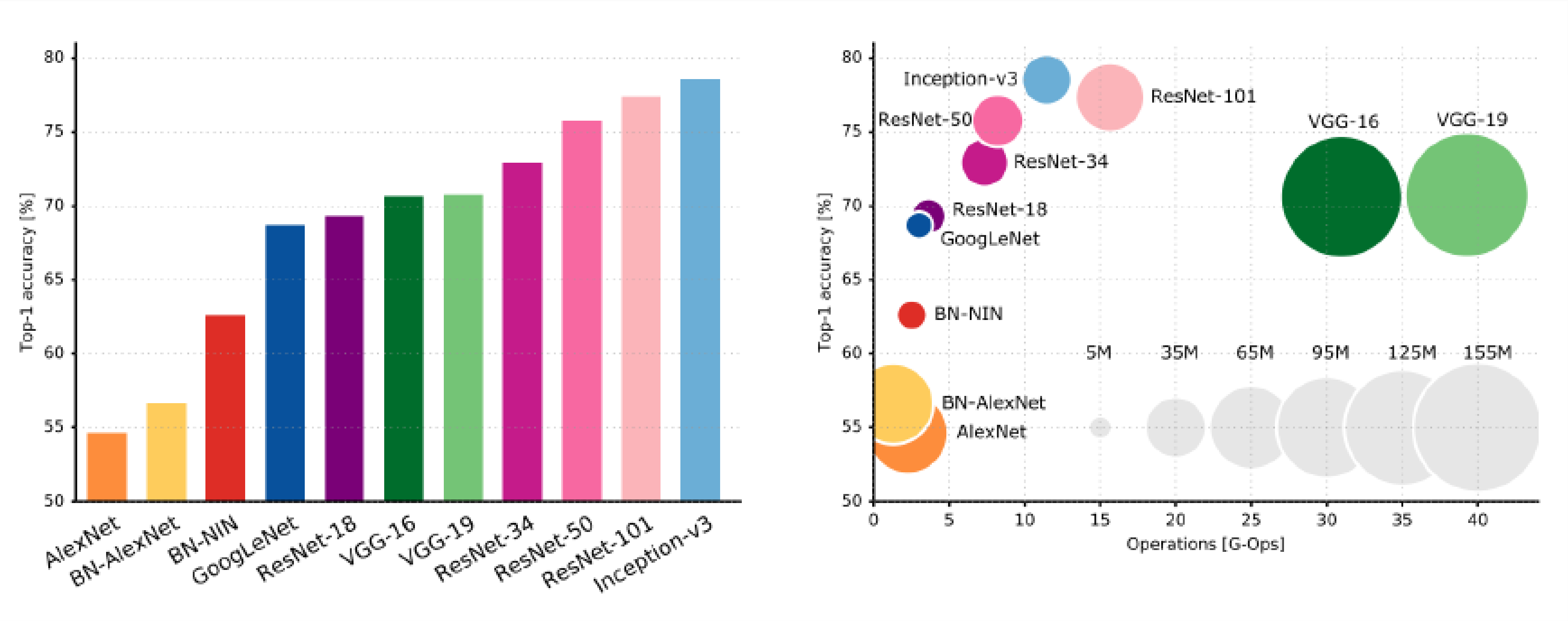
$$\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$$

instead.

The original function is then

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

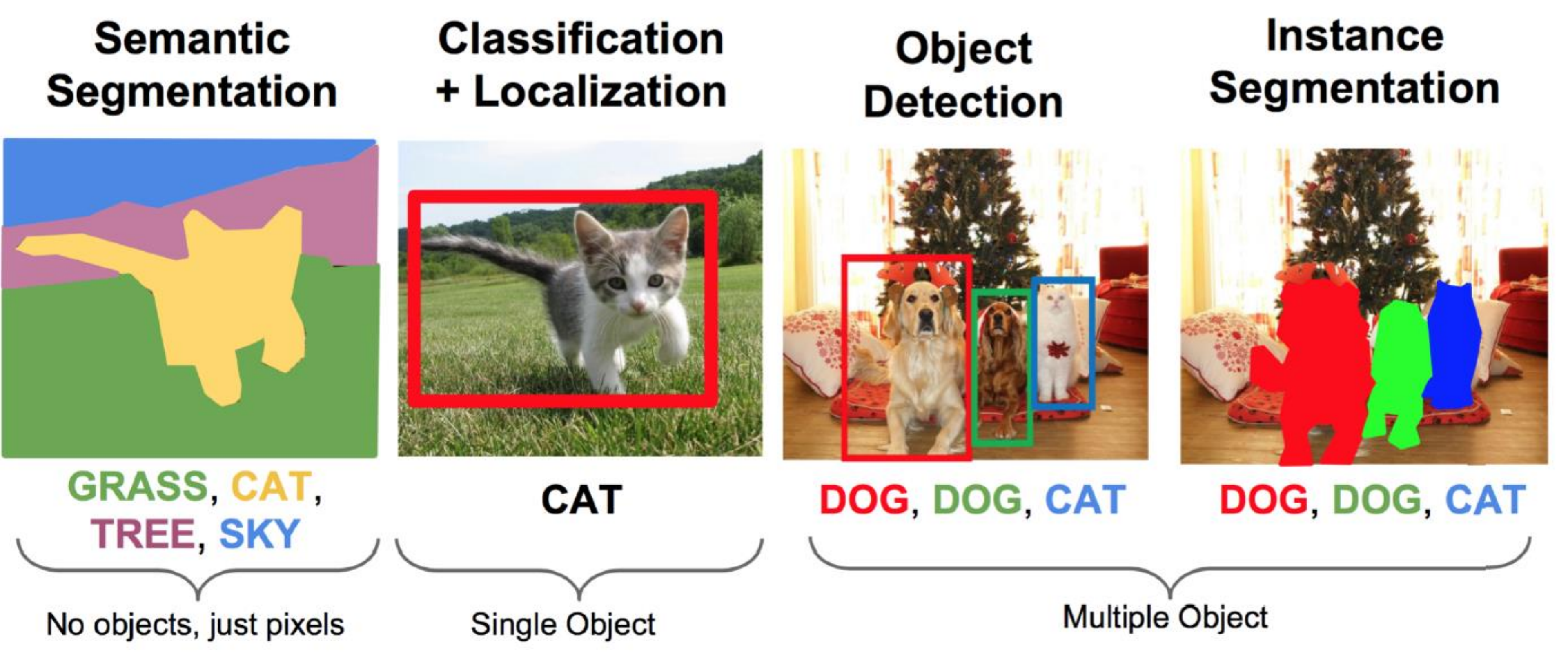
Network comparison



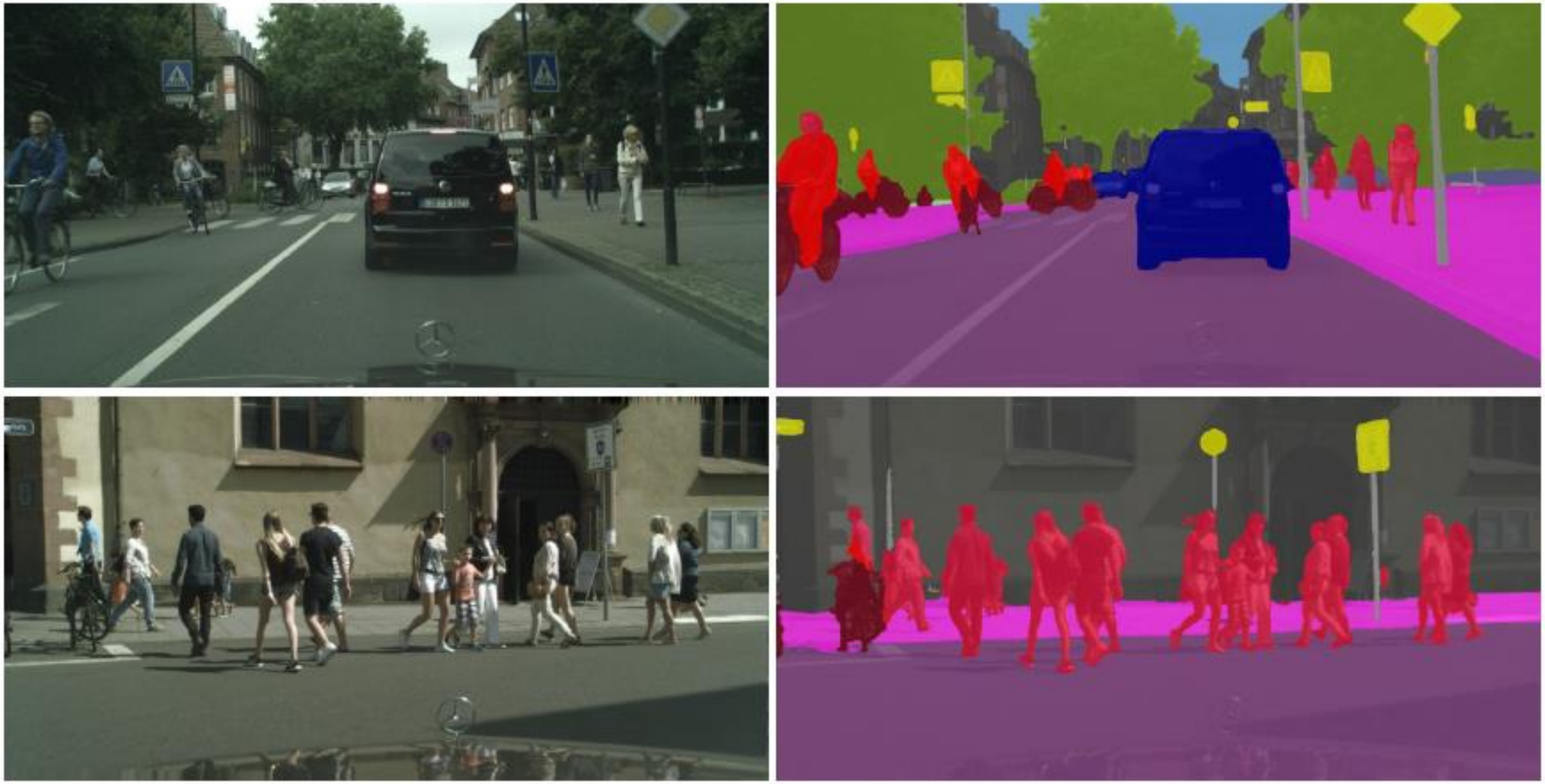
Lecture plan

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

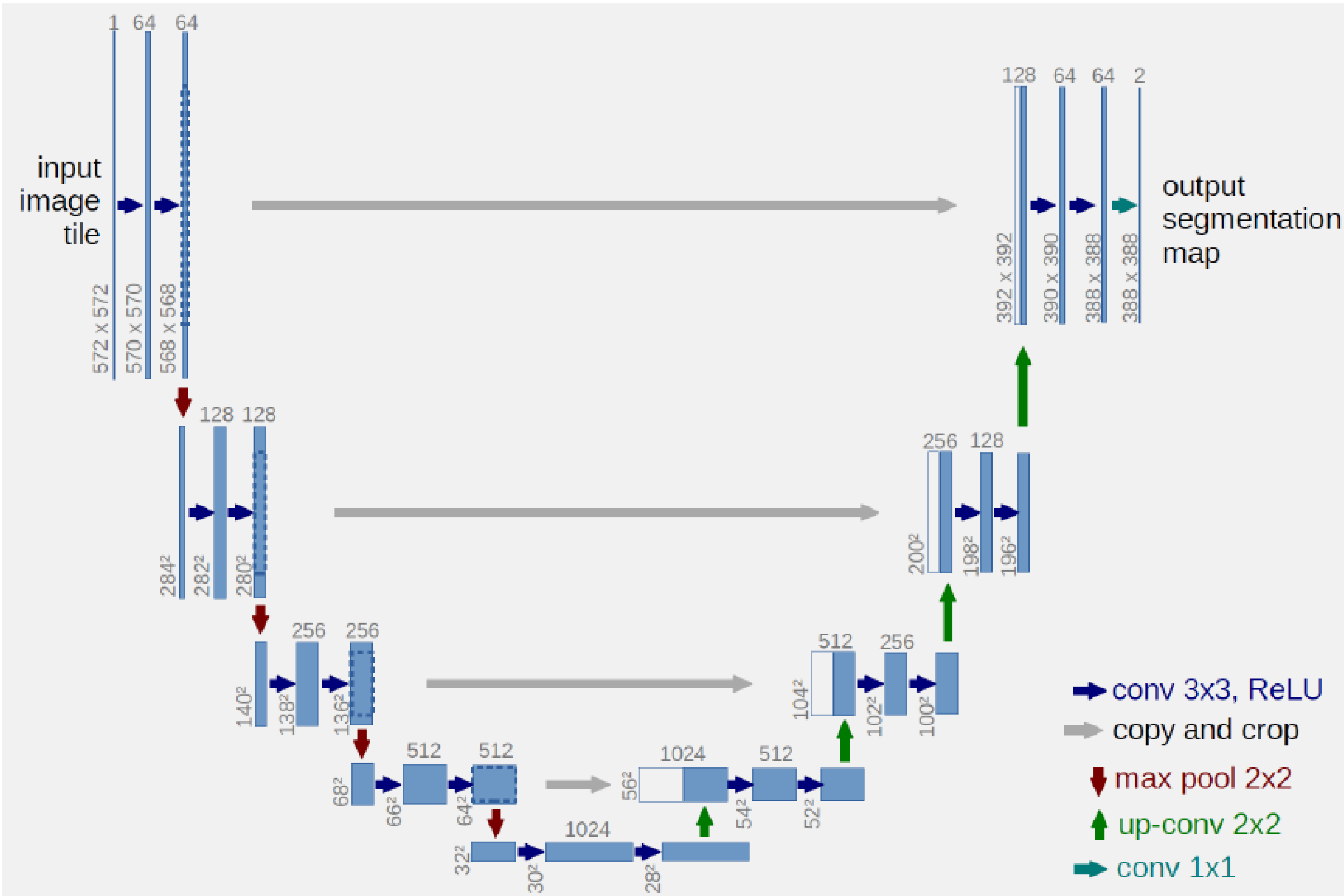
CV tasks


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

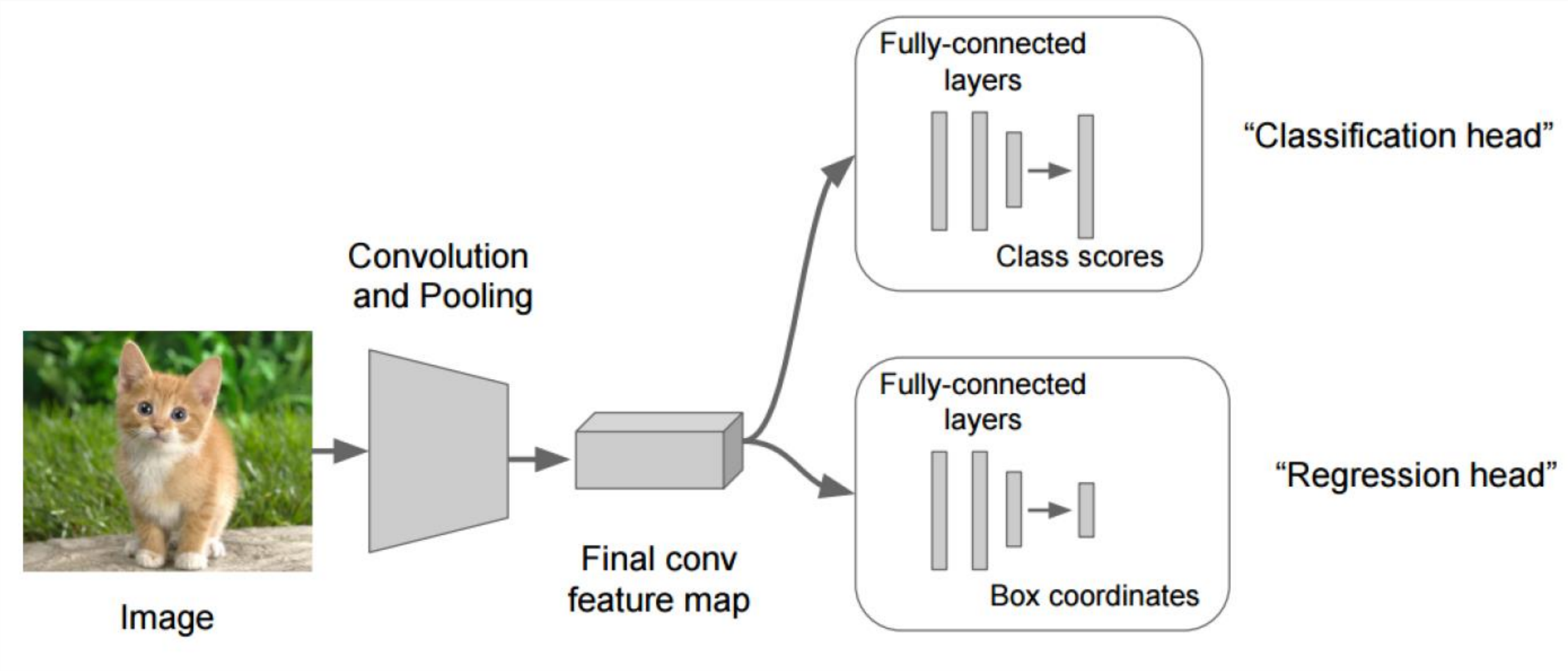


U-Net



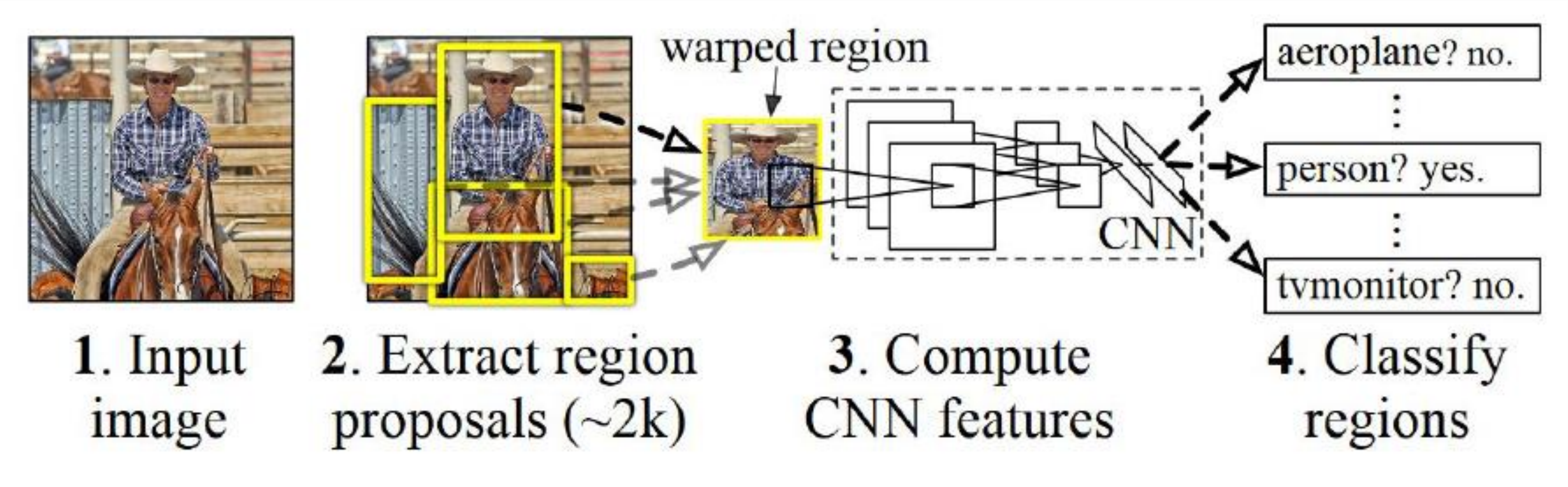
1
64
64
input
image
tile
572 x 572
570 x 570
568 x 568
128
128
284^2
282^2
280^2
256
256
140^2
138^2
136^2
512
512
68^2
66^2
64^2
1024
32^2
30^2
28^2
1024
512
56^2
54^2
52^2
512
256
104^2
102^2
100^2
256
128
200^2
198^2
196^2
128
64
64
2
392 x 392
390 x 390
388 x 388
388 x 388
output
segmentation
map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

Object localization



Object detection. Pascal VOC

Detection via R-CNN



Detection via YOLO