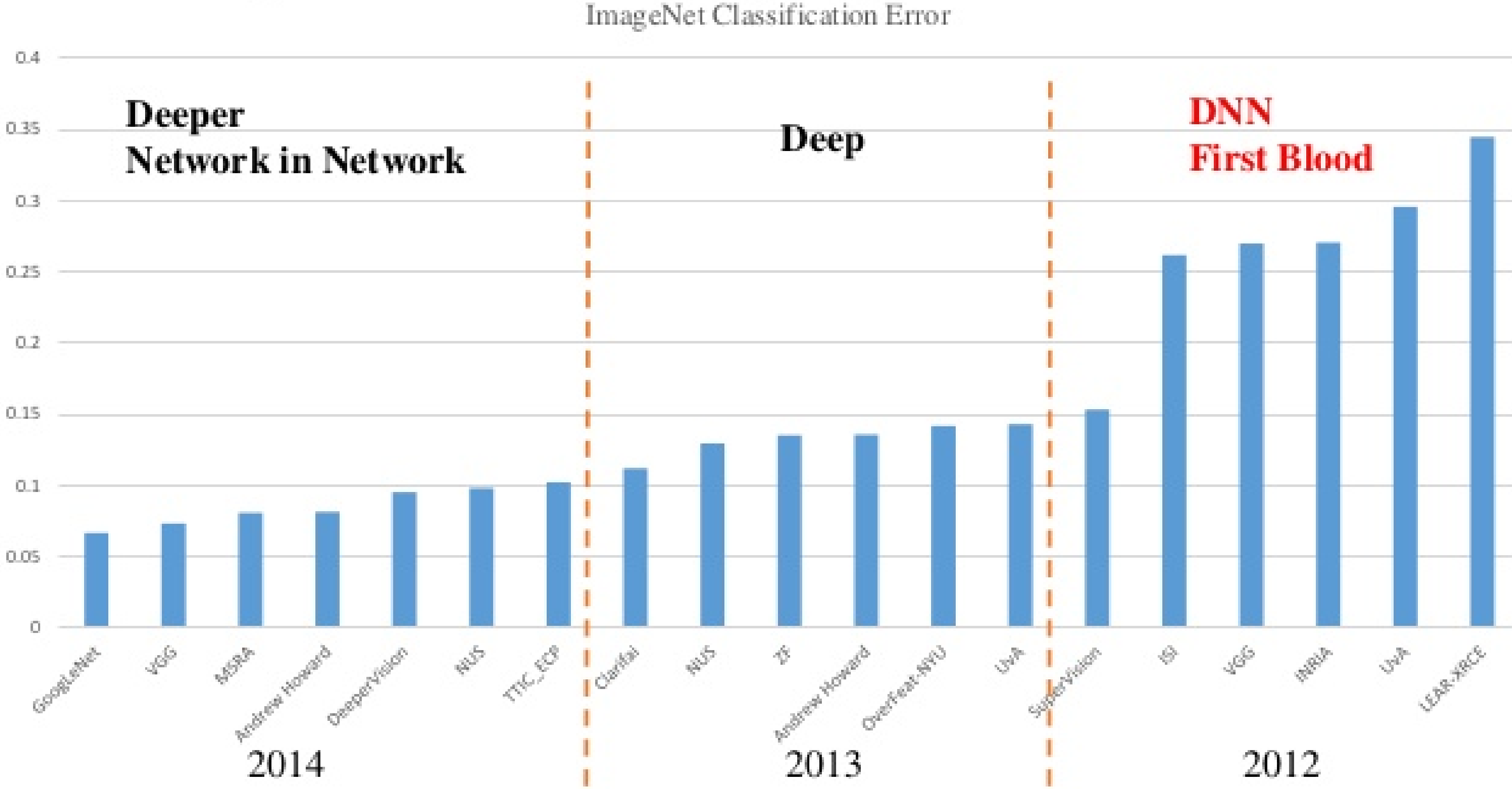
Neural Networks

Jan Chorowski

ImageNet Classification

- **1000** categories and **1.2** million training images



Li Fei-Fei: ImageNet Large Scale Visual Recognition Challenge, 2014 http://image-net.org/

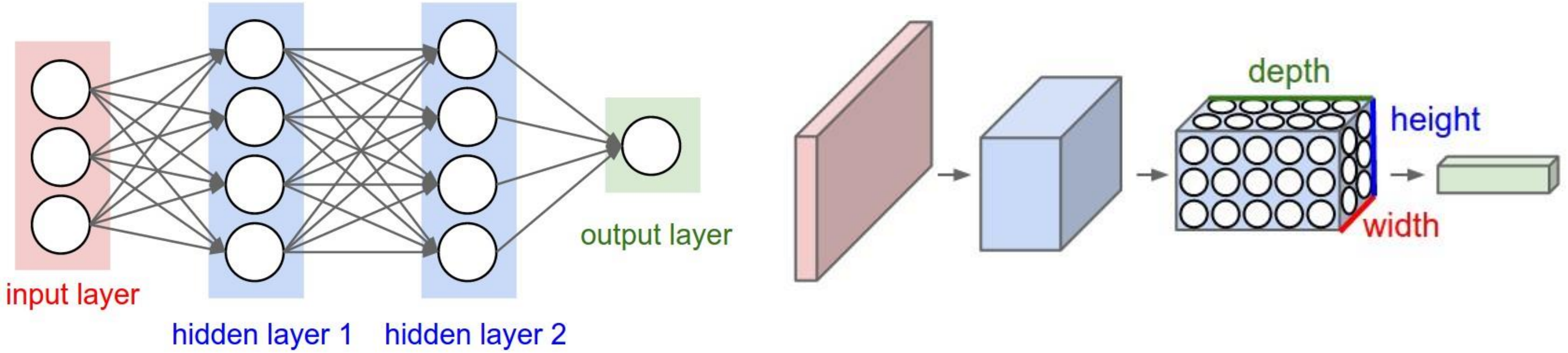

CONVNETS, CONVNETS
EVERYWHERE
memegen.com

Sharing neurons - convolutions

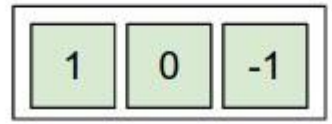
Note: material from http://cs231n.github.io/convolutional-networks/

In a conv net we use a different connection pattern between layers:

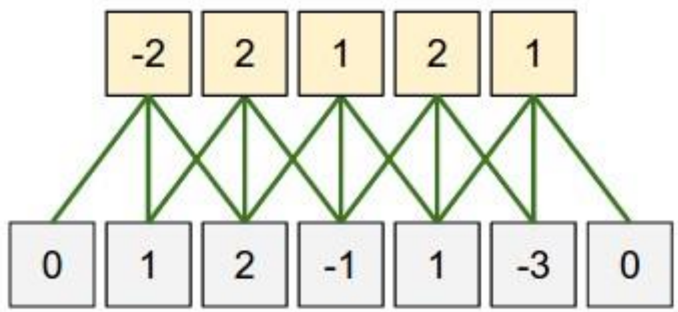
- Typically we use an all-to-all scheme
- In a conv-net we use local connectivity!



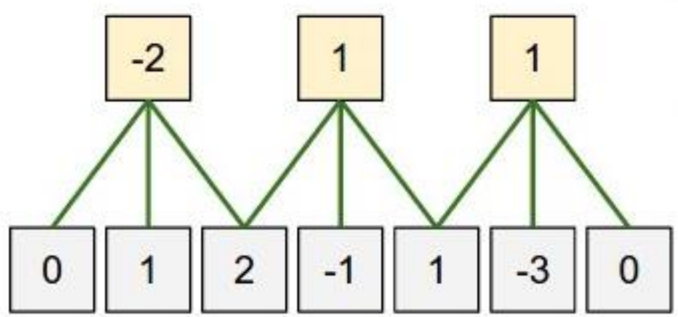
1D Convolution layer

- Small filter (neuron): 

- Swipe the filter over the sequence

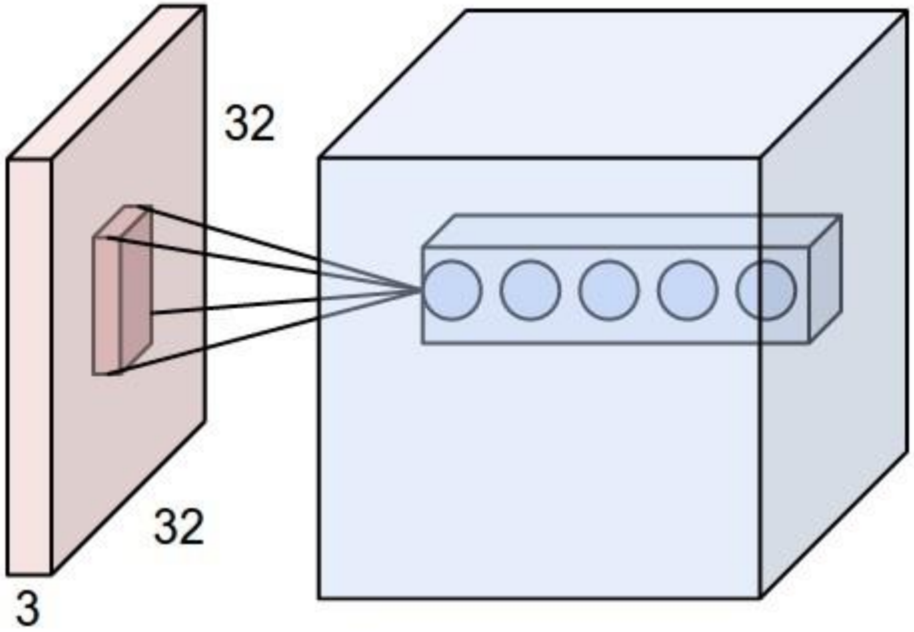


- Pool or stride (select only a few outputs)

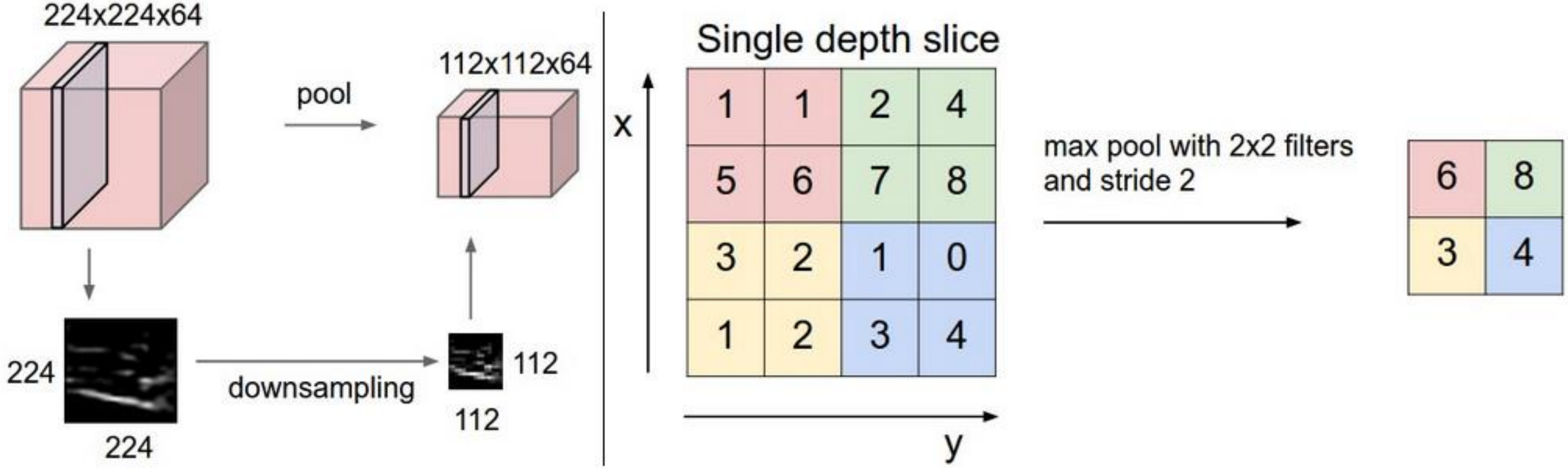


2D conv layer

- http://cs231n.github.io/convolutional-networks/



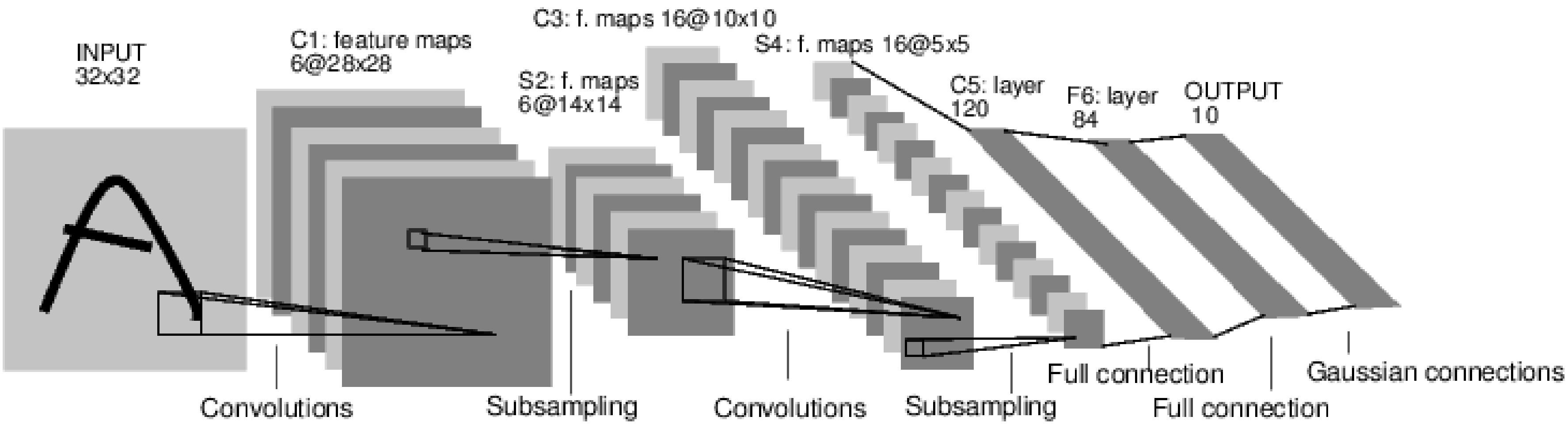
Pooling



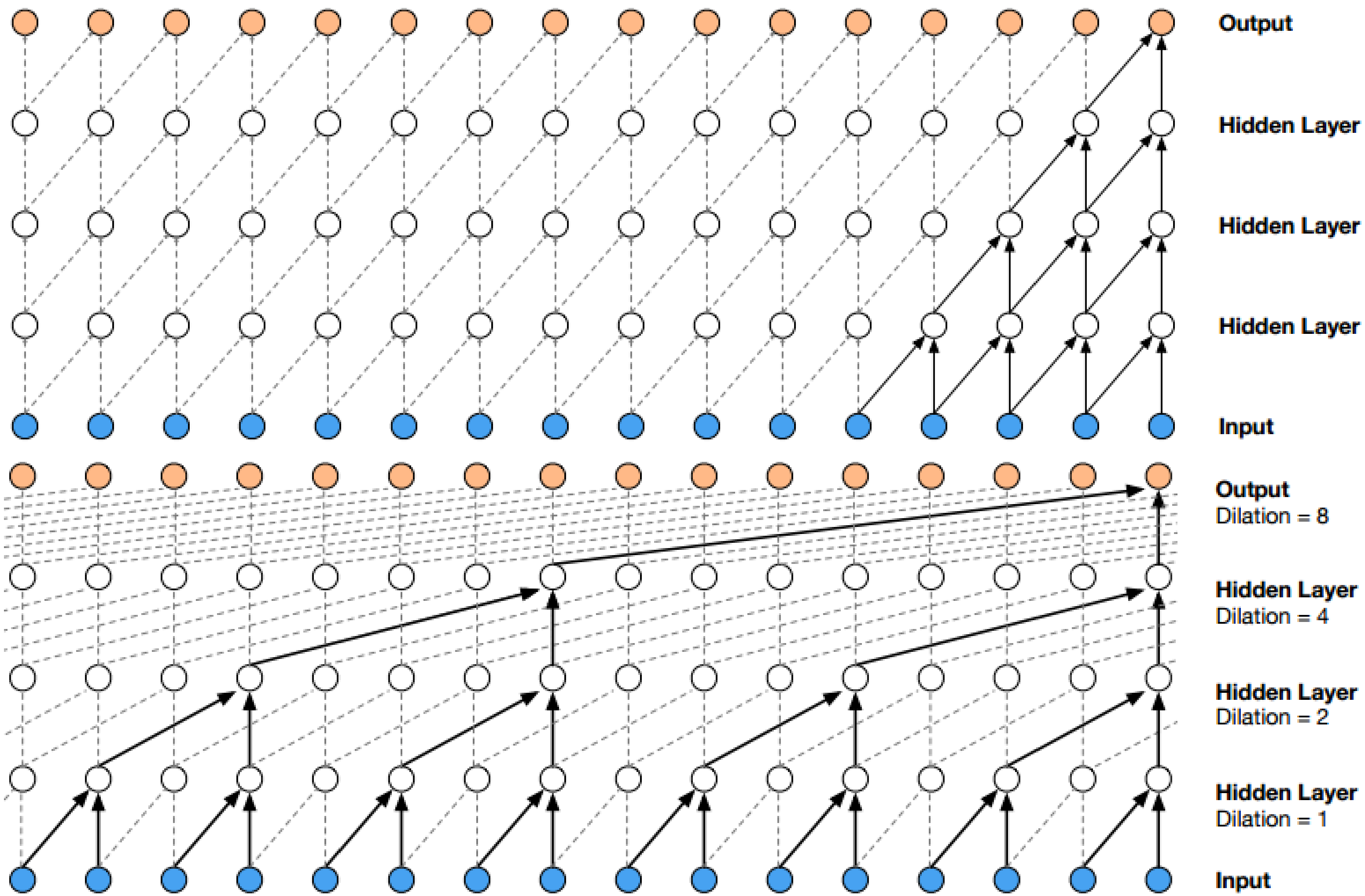
Pooling layer downsamples the volume spatially, independently in each depth slice of the input volume. **Left:** In this example, the input volume of size [224x224x64] is pooled with filter size 2, stride 2 into output volume of size [112x112x64]. Notice that the volume depth is preserved. **Right:** The most common downsampling operation is max, giving rise to **max pooling**, here shown with a stride of 2. That is, each max is taken over 4 numbers (little 2x2 square).

Full network - LeNet

Y. LeCun et al. „Gradient-Based Learning Applied to Document Recognition”
http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf



Dilated convolutions:
wide receptive field at high resolutions



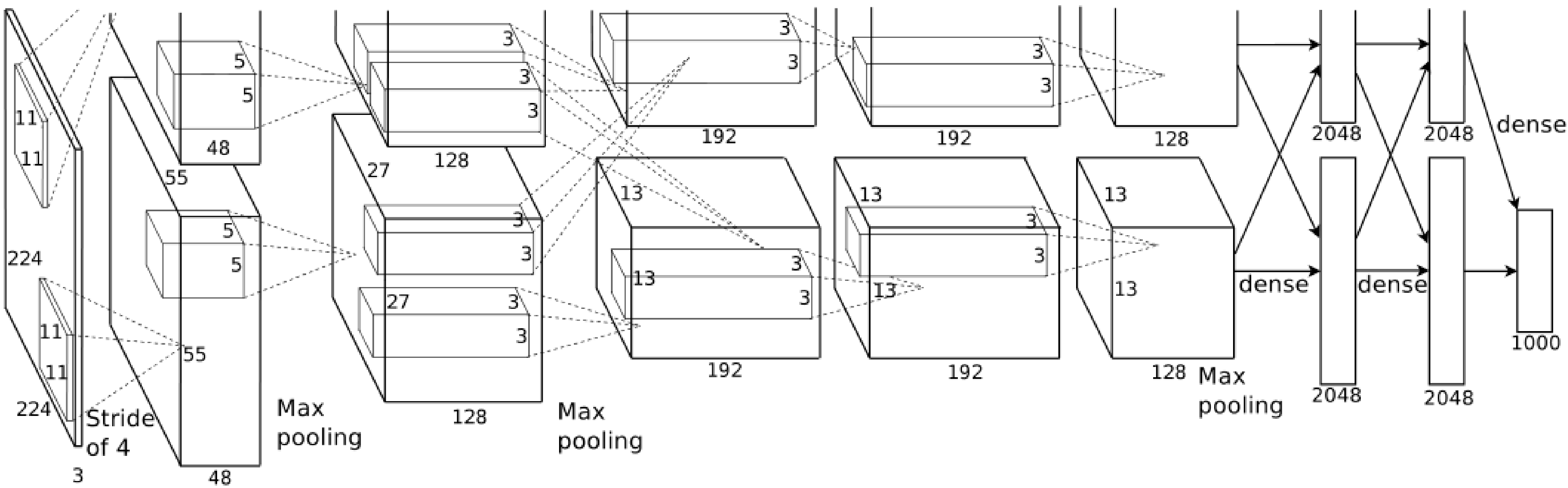
Wavenet: https://arxiv.org/pdf/1609.03499.pdf

https://deepmind.com/blog/wavenet-generative-model-raw-audio/

Technicality: Edge padding

Convs can mimick FC layers

AlexNet (2012)



60M parameters

The last dense layers takes: 4096*4096 + 4096*1000=20M params!

VGGNet: Network Design

Key design choices:

- 3x3 conv. kernels – very small
- conv. stride 1 – no loss of information

Other details:

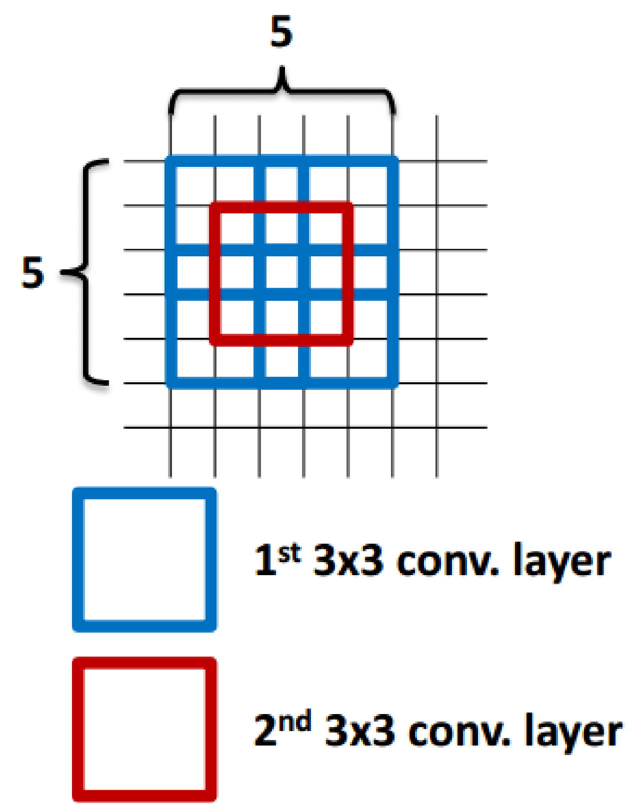
- Rectification (ReLU) non-linearity
- 5 max-pool layers (x2 reduction)
- no normalisation
- 3 fully-connected (FC) layers

Slide from http://www.robots.ox.ac.uk/~karen/pdf/ILSVRC_2014.pdf

VGGNet: Discussion

Why 3x3 layers?

- Stacked conv. layers have a large receptive field
  - two 3x3 layers – 5x5 receptive field
  - three 3x3 layers – 7x7 receptive field
- More non-linearity
- Less parameters to learn
  - ~140M per net

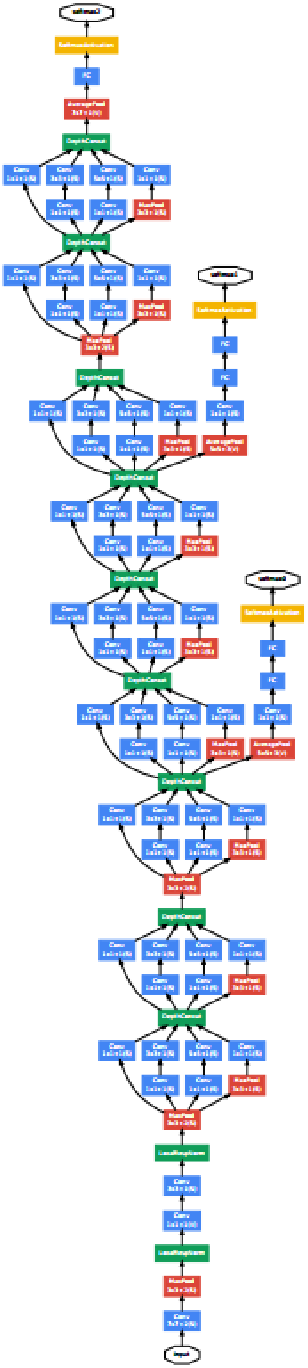


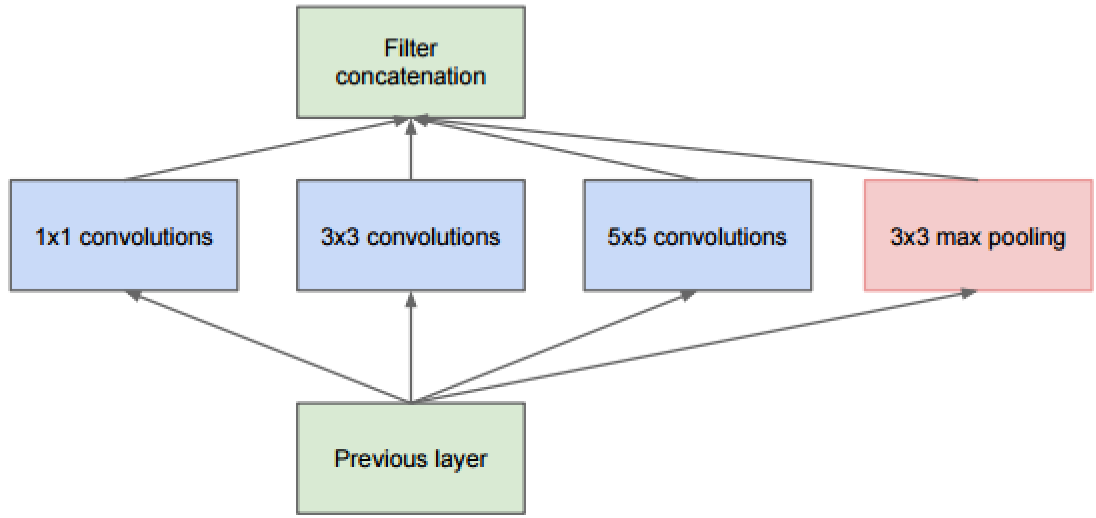
Slide from http://www.robots.ox.ac.uk/~karen/pdf/ILSVRC_2014.pdf

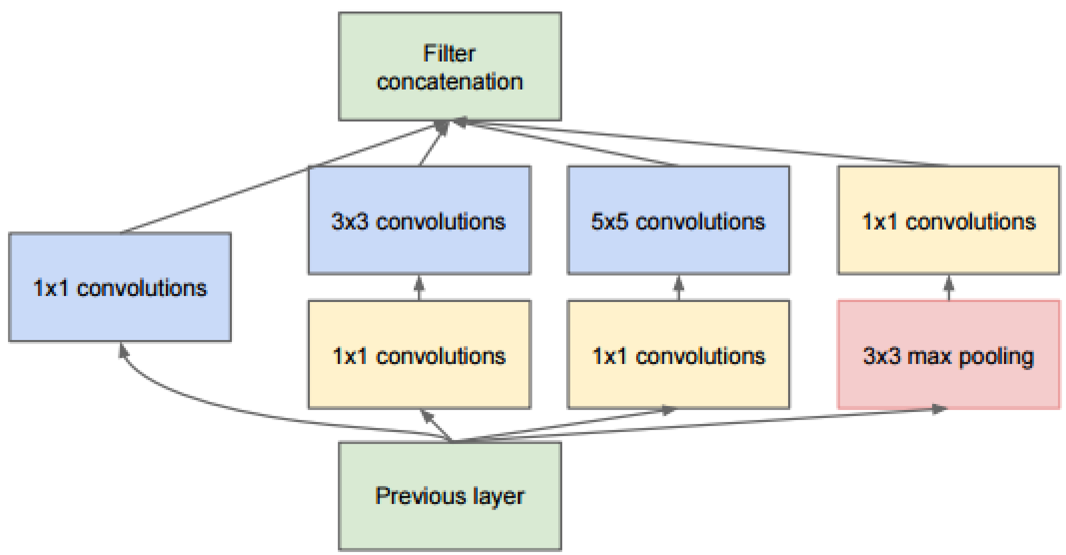
VGGNet (2014)

INPUT: [224x224x3] memory: 224*224*3=150K weights: 0
CONV3-64: [224x224x64] memory: 224*224*64=3.2M weights: (3*3*3)*64 = 1,728
CONV3-64: [224x224x64] memory: 224*224*64=3.2M weights: (3*3*64)*64 = 36,864
POOL2: [112x112x64] memory: 112*112*64=800K weights: 0
CONV3-128: [112x112x128] memory: 112*112*128=1.6M weights: (3*3*64)*128 = 73,728
CONV3-128: [112x112x128] memory: 112*112*128=1.6M weights: (3*3*128)*128 = 147,456
POOL2: [56x56x128] memory: 56*56*128=400K weights: 0
CONV3-256: [56x56x256] memory: 56*56*256=800K weights: (3*3*128)*256 = 294,912
CONV3-256: [56x56x256] memory: 56*56*256=800K weights: (3*3*256)*256 = 589,824
CONV3-256: [56x56x256] memory: 56*56*256=800K weights: (3*3*256)*256 = 589,824
POOL2: [28x28x256] memory: 28*28*256=200K weights: 0
CONV3-512: [28x28x512] memory: 28*28*512=400K weights: (3*3*256)*512 = 1,179,648
CONV3-512: [28x28x512] memory: 28*28*512=400K weights: (3*3*512)*512 = 2,359,296
CONV3-512: [28x28x512] memory: 28*28*512=400K weights: (3*3*512)*512 = 2,359,296
POOL2: [14x14x512] memory: 14*14*512=100K weights: 0
CONV3-512: [14x14x512] memory: 14*14*512=100K weights: (3*3*512)*512 = 2,359,296
CONV3-512: [14x14x512] memory: 14*14*512=100K weights: (3*3*512)*512 = 2,359,296
CONV3-512: [14x14x512] memory: 14*14*512=100K weights: (3*3*512)*512 = 2,359,296
POOL2: [7x7x512] memory: 7*7*512=25K weights: 0
FC: [1x1x4096] memory: 4096 weights: 7*7*512*4096 = 102,760,448
FC: [1x1x4096] memory: 4096 weights: 4096*4096 = 16,777,216
FC: [1x1x1000] memory: 1000 weights: 4096*1000 = 4,096,000
TOTAL memory: 24M * 4 bytes ~= 93MB / image (only forward! ~*2 for bwd) TOTAL params: 138M parameters

GoogLeNet (Inception) 2014







Separable Filters

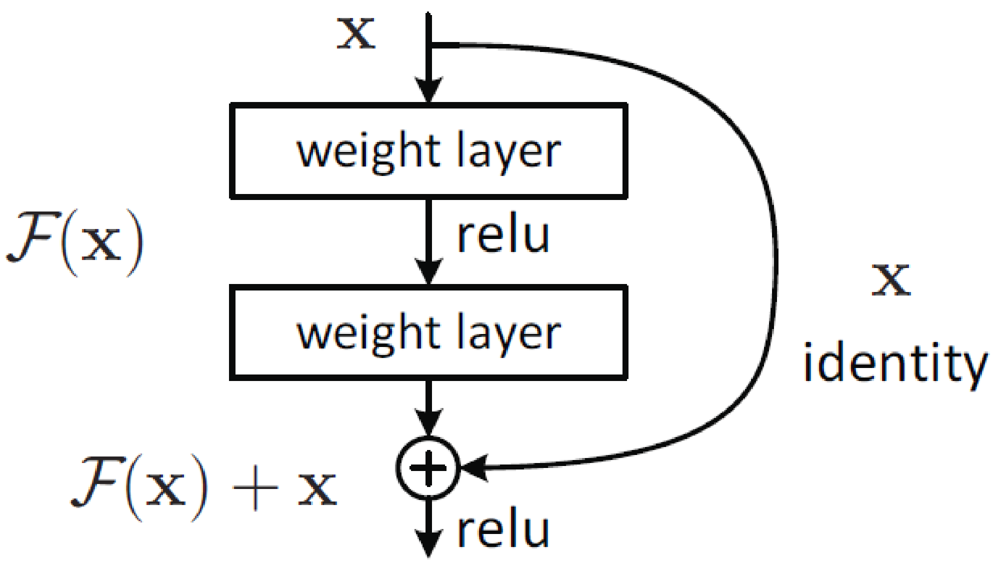
- Someties a 2D conv == 2* 1D conv:

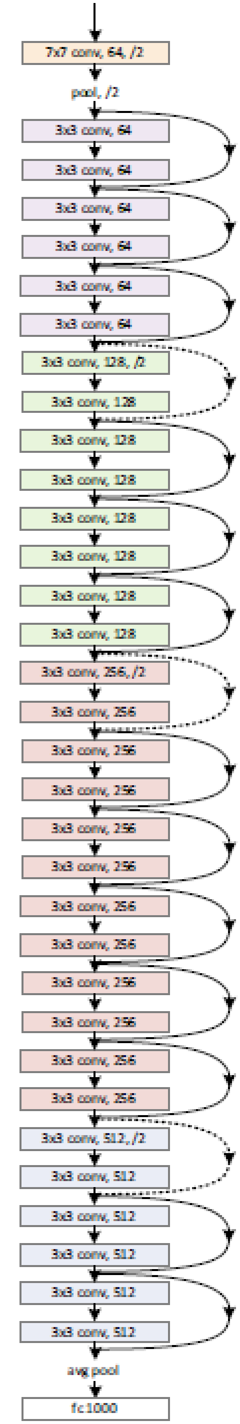
$$\frac{1}{3}\begin{bmatrix}1\\1\\1\end{bmatrix} * \frac{1}{3}\begin{bmatrix}1 & 1 & 1\end{bmatrix} = \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix} \qquad \mathbf{G_x} = \begin{bmatrix}1 & 0 & -1\\2 & 0 & -2\\1 & 0 & -1\end{bmatrix} * A = \begin{bmatrix}1\\2\\1\end{bmatrix} * \begin{bmatrix}+1 & 0 & -1\end{bmatrix} * A$$

- Inception and VGGnet use approximate sepration, with more nonlinearity

ResNest (Residual Nets, 2015)

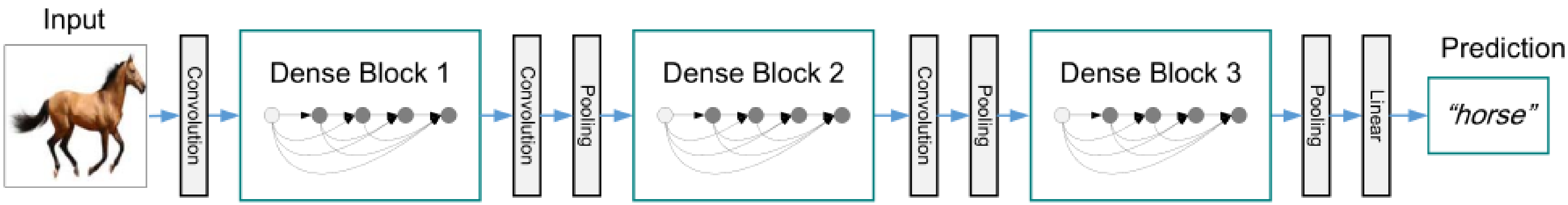
- Add bypases to ease gradient flow
- Networks with more than 150 layers!
- 3x3 convs with number of feature maps doubling after every pooling



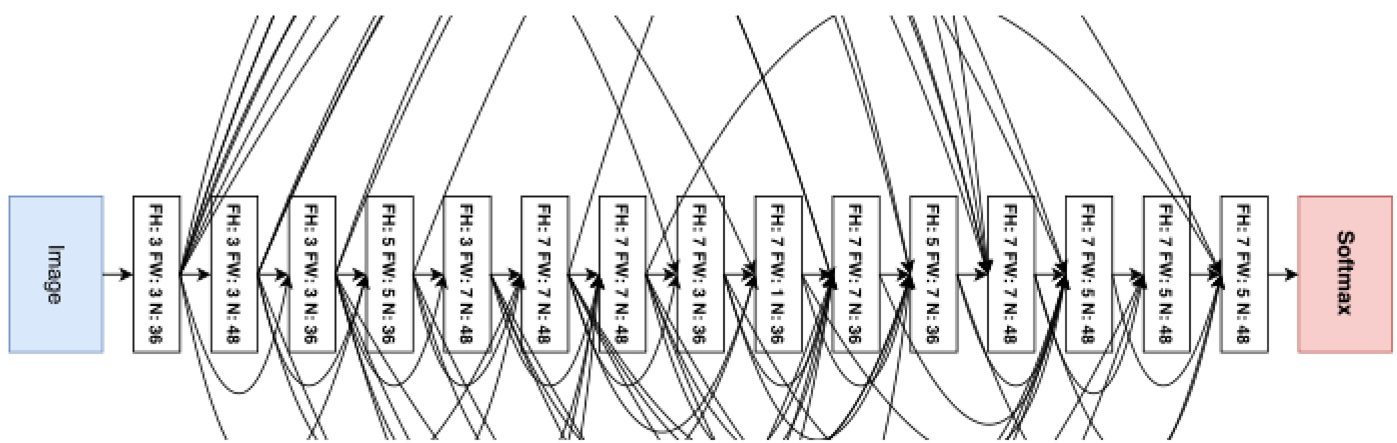


Quest for models

- DenseNets (https://arxiv.org/pdf/1608.06993)



- Automatic architecture search (https://arxiv.org/pdf/1611.01578)

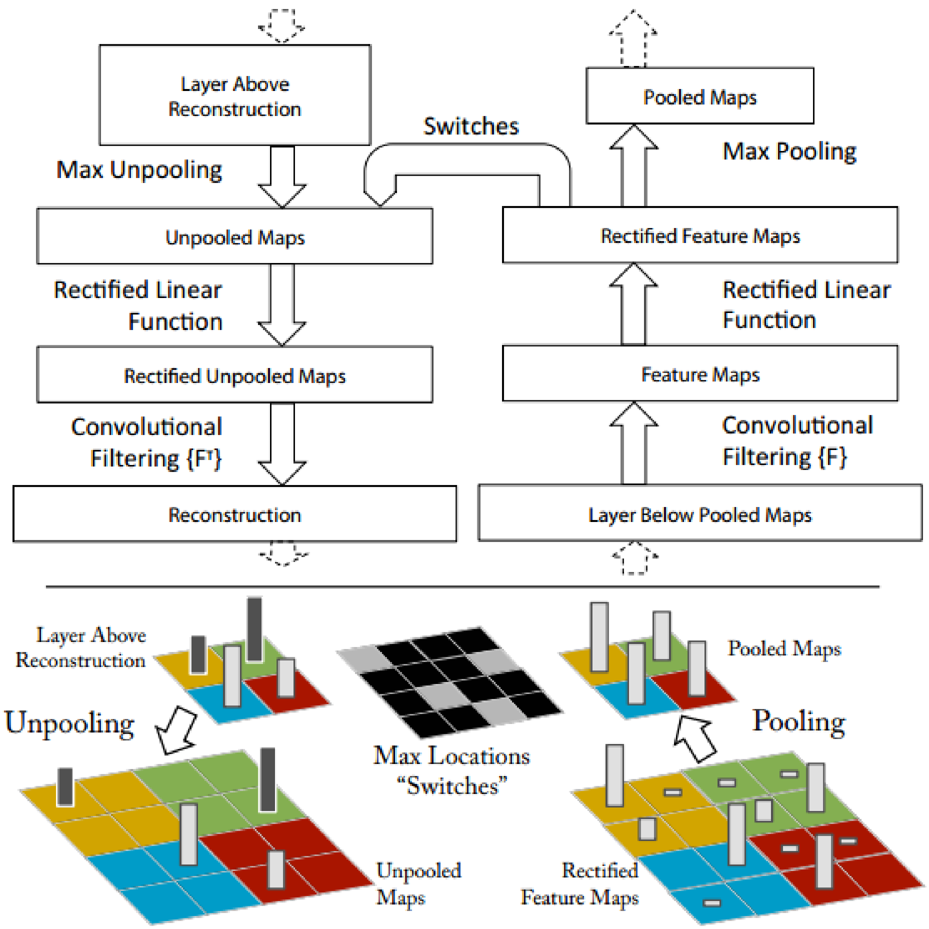


Which ConvNet to use?

- Take current best on Imagenet
- Many networks are available for download (google tensorflow model and caffe model zoo)

- For your use – take a pretrained net and retrain only the last layers!
- On ImageNet (1M images) data augmentation and regularization is as important as the network!

Visualizing CNN features

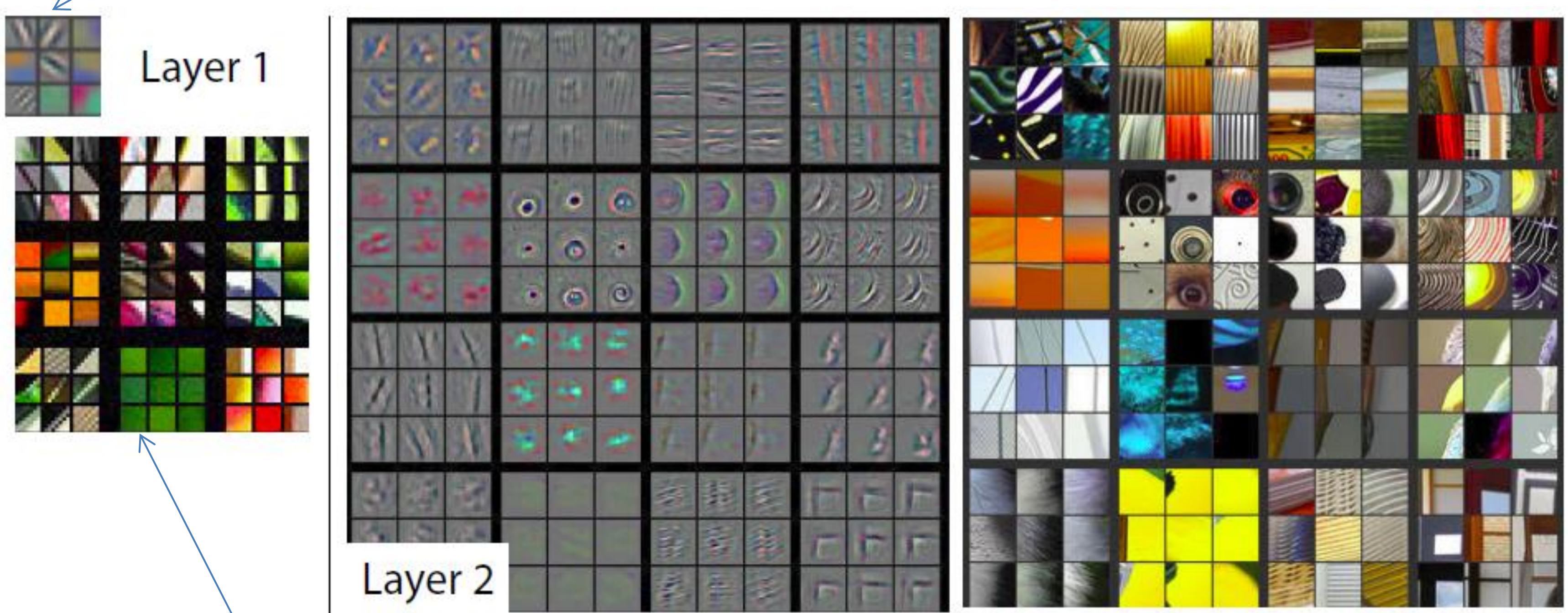


Main idea:
Do a forward propagation,
Save state of nonlinearities,
Do a full transposed convolution (deconvolution) – kind of like inverse convolution

M. Zieler, "Visualizing and Understanding Convolutional Networks"

Low-level features

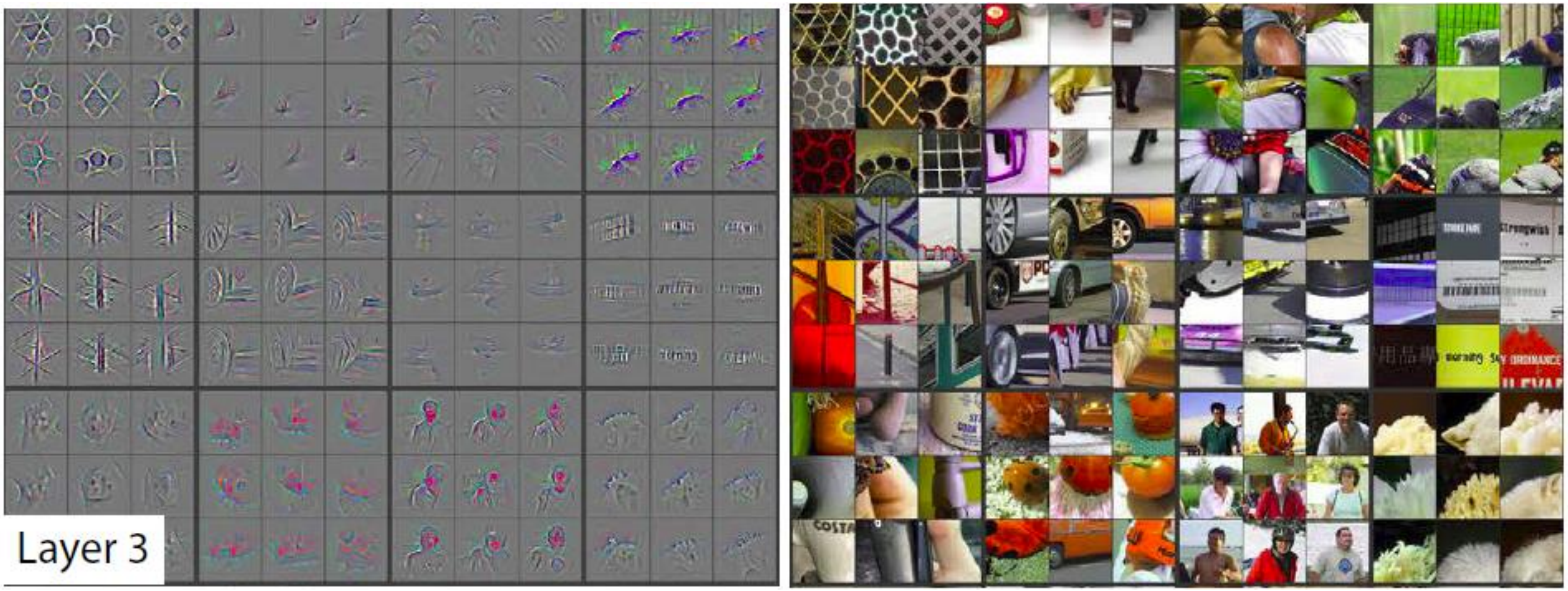
What the neuron (feature-detector looks for)



What images are selected by the neuron

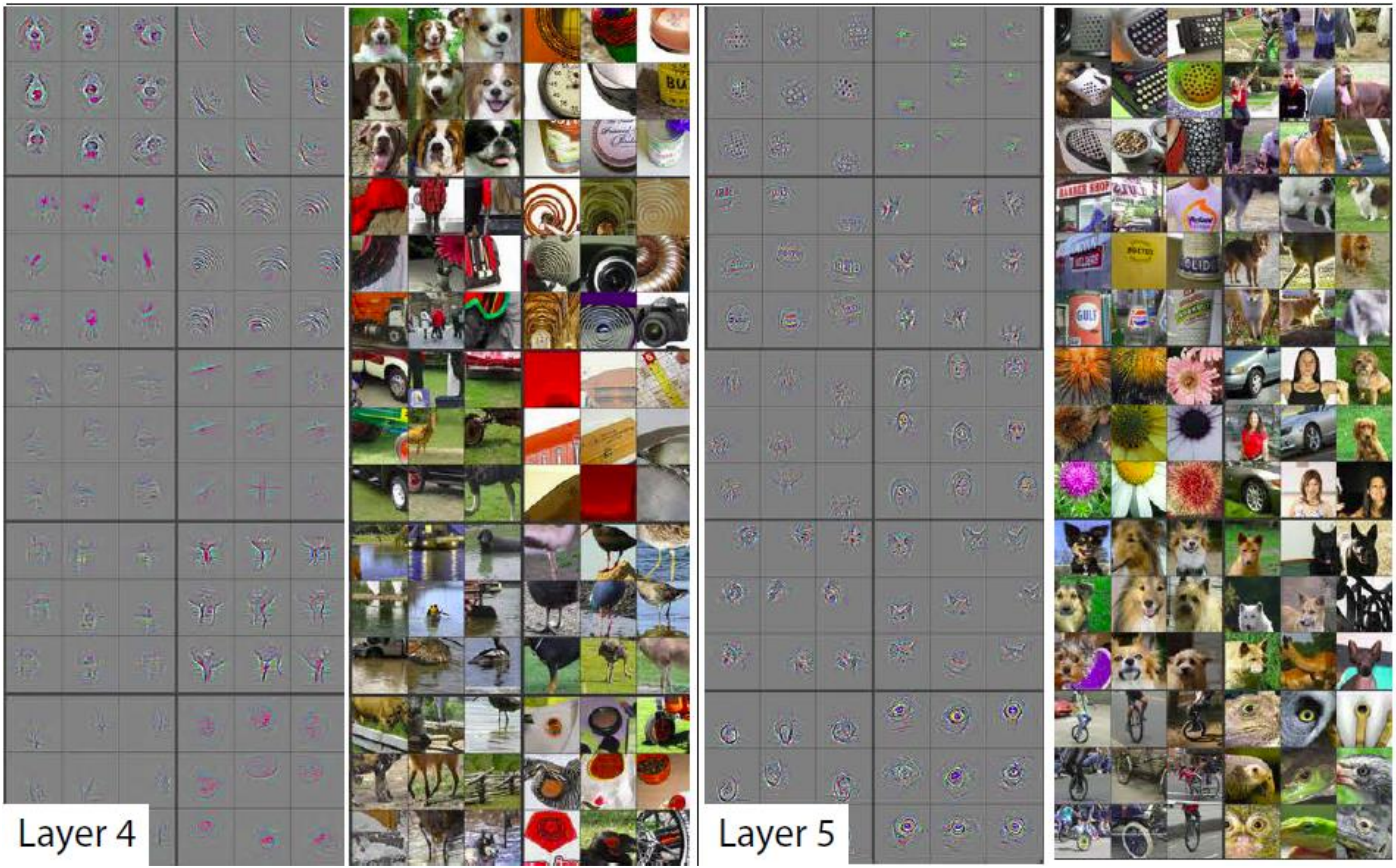
M. Zieler, “Visualizing and Understanding Convolutional Networks”

Mid-level features



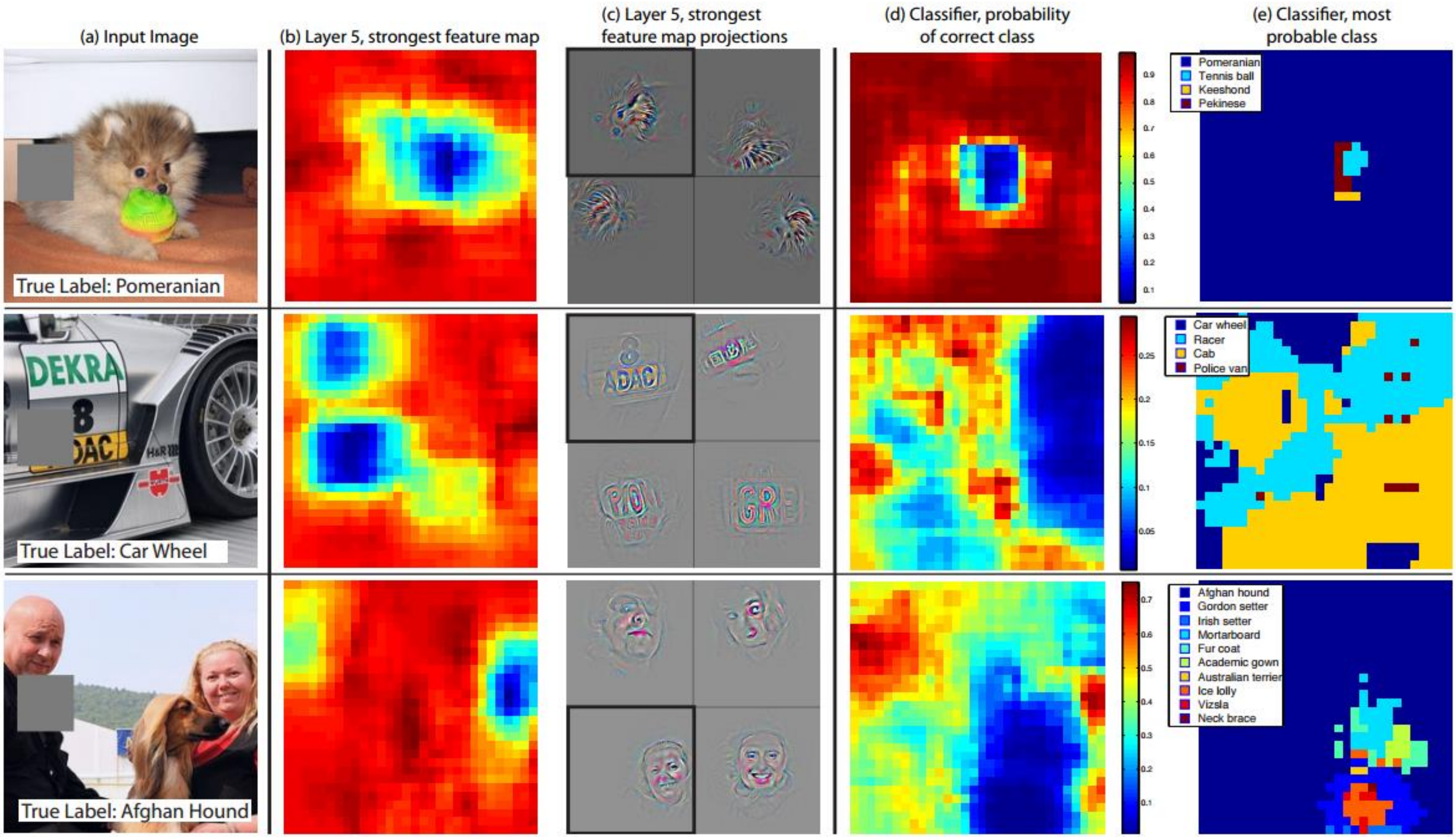
M. Zieler, “Visualizing and Understanding Convolutional Networks”

High-level features

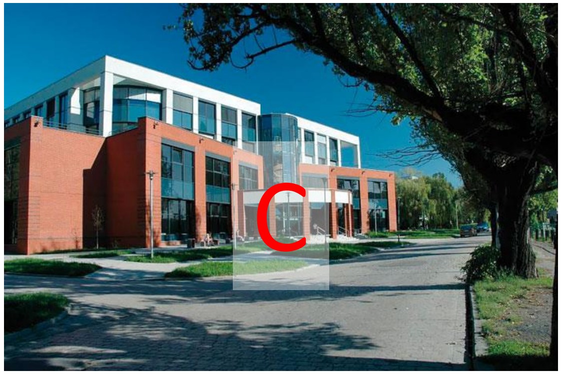


M. Zieler, “Visualizing and Understanding Convolutional Networks”

Where Convnets look?



Style Transfer



+

=

Find image (backpropagation toward pixels) minimizing:

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_S =$$
$$= (F(C) - F(I)\,)^2 + \big(G(C) - G(I)\big)^2$$

Where:

$F(x)$ features of a CNN layer on image x

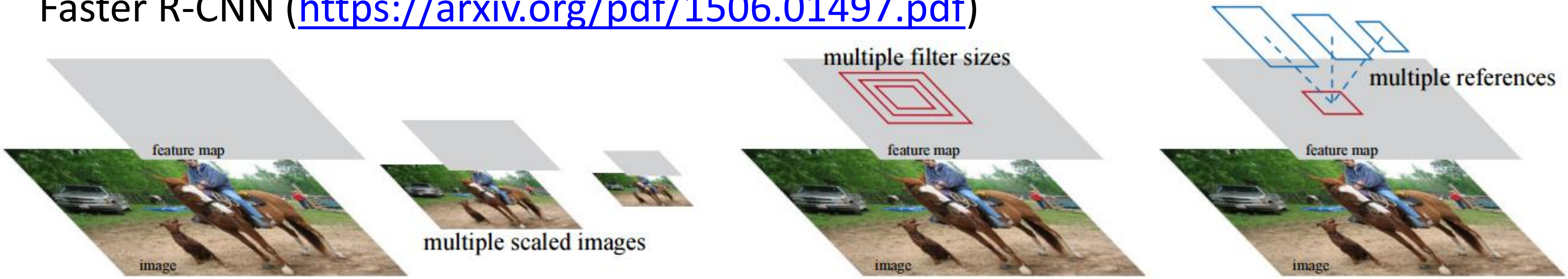
$G(x)$ matrix of correlations between a layer’s features over pixels of image x

Gatys et al., „A Neural Algorithm of Artistic Style”, 2015

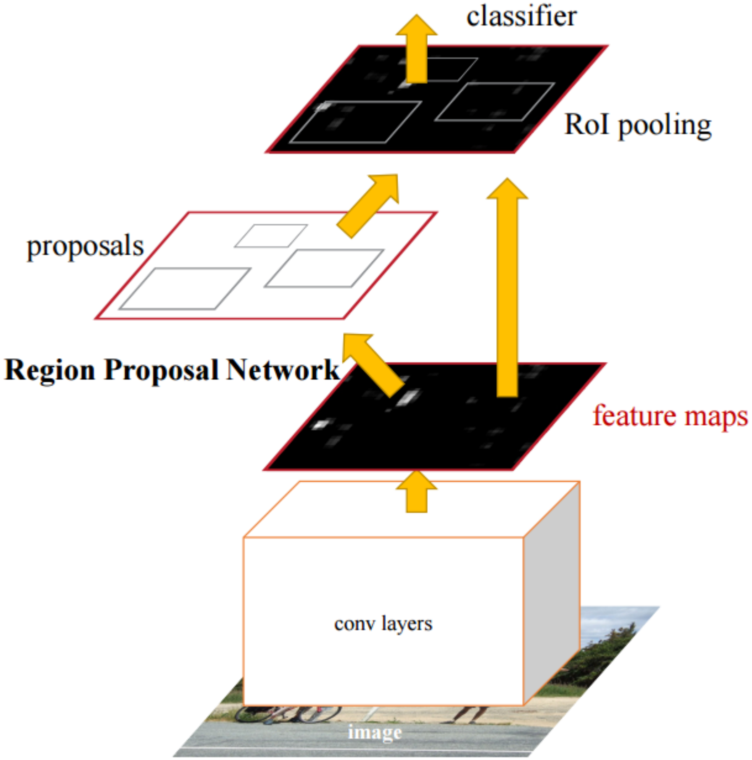
Object detection and segmentation

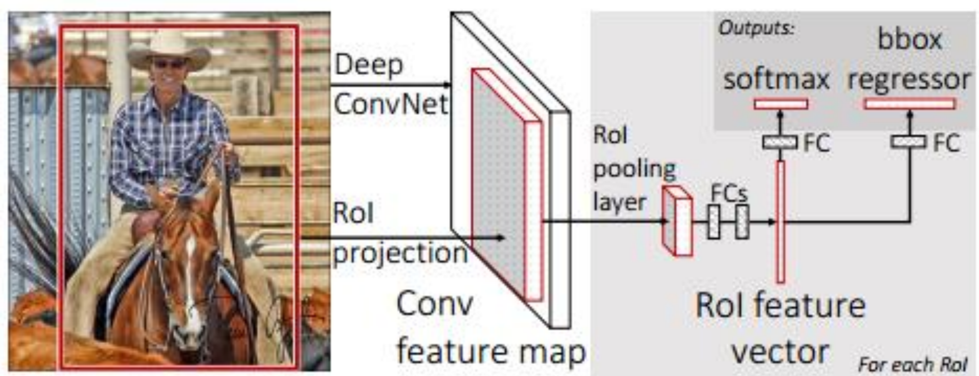
Fast R-CNN (https://arxiv.org/pdf/1504.08083.pdf)

Faster R-CNN (https://arxiv.org/pdf/1506.01497.pdf)



(b) (c)





YOLO – fast object detection

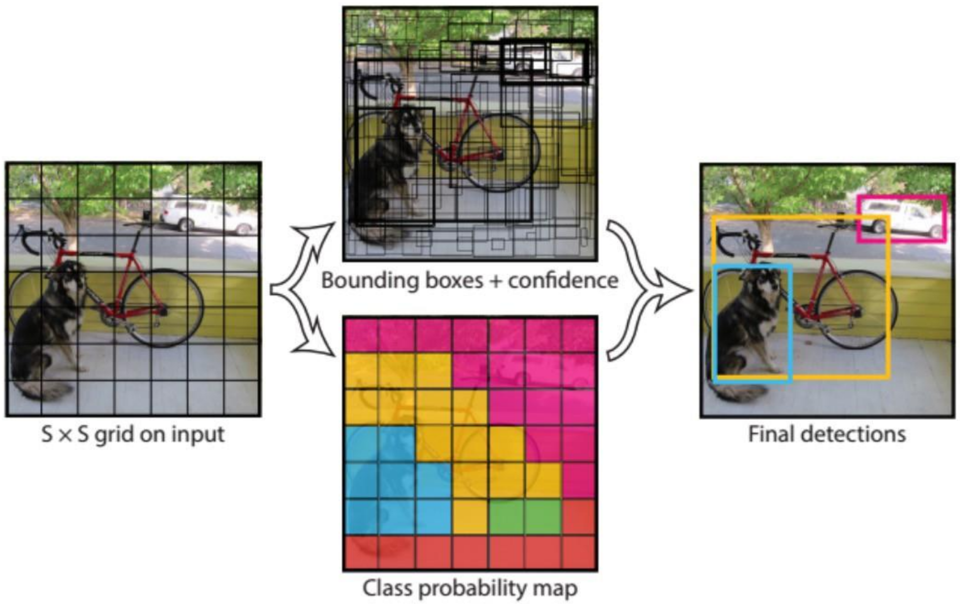


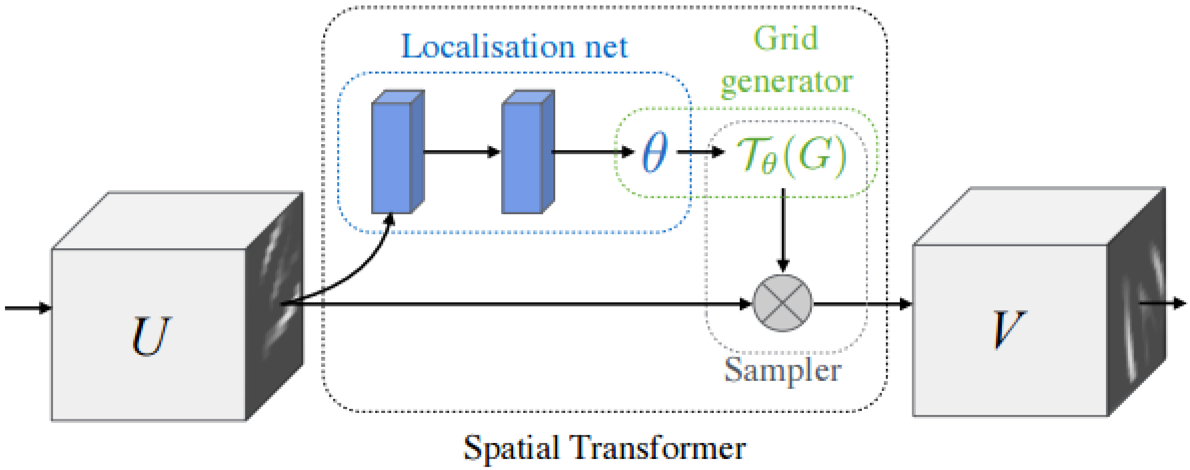
Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

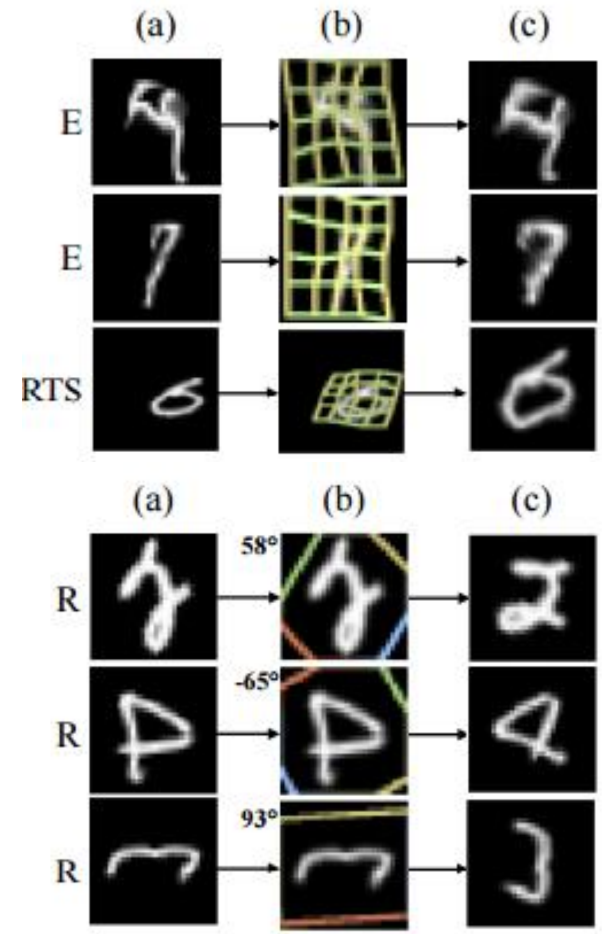
- No region proposal and looping over ROIs!
- In a fully convolutional net predict for each pixel of a feature map:
  - Objectness – is something there
  - Bounding Box – how large it is

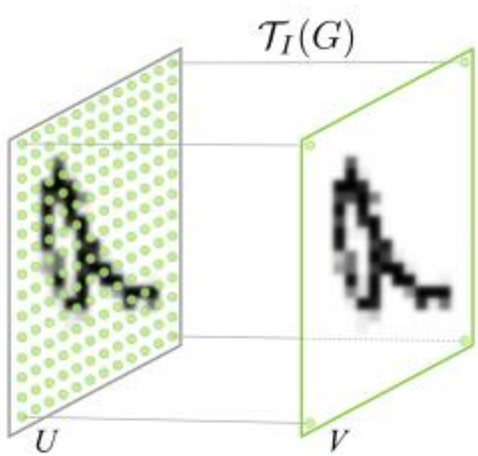
https://arxiv.org/pdf/1506.02640v5.pdf

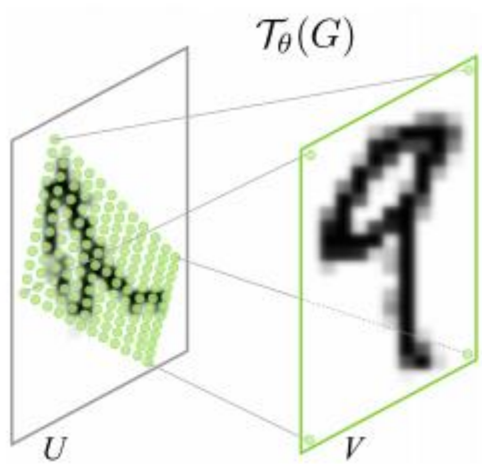
Spatial Transformer Networks

https://arxiv.org/pdf/1506.02025.pdf

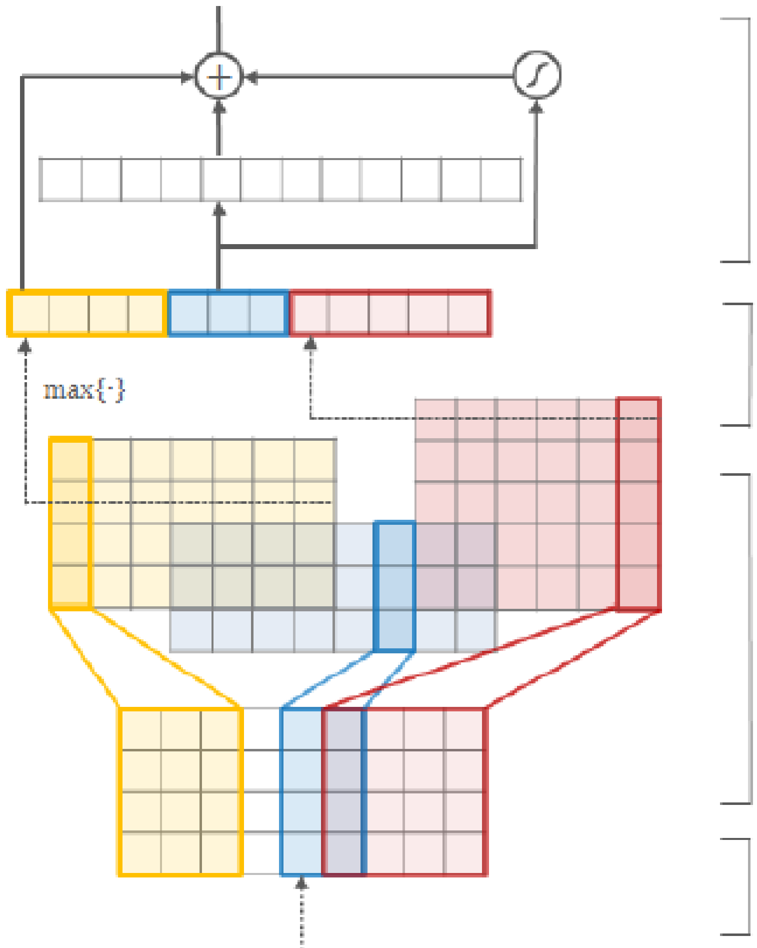








CNN beyond images
Or character-to-word embeding



„incomprehensible"

Highway layers – very nonlinear transformation of data

Glue best word representations together

Convolutional filters of varying lengths. Can react to pre-, in-, and post-fixes of words

Character embedings concatenated into a matrix

Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-Aware Neural Language Models," *arXiv:1508.06615 [cs, stat]*, Aug. 2015.

Additional references

- http://cs231n.github.io
- How transferable are features in deep neural networks? studies the transfer learning performance in detail, including some unintuitive findings about layer co-adaptations.
- CNN Features off-the-shelf: an Astounding Baseline for Recognition trains SVMs on features from ImageNet-pretrained ConvNet and reports several state of the art results.