RWTH AACHEN UNIVERSITY

Advanced Machine Learning
Lecture 15

Convolutional Neural Networks III

12.01.2017

Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de/

leibe@vision.rwth-aachen.de

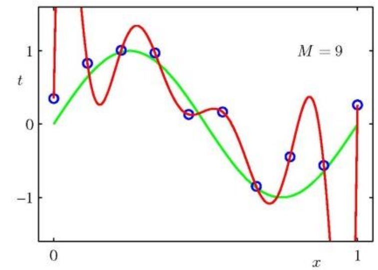
Announcement

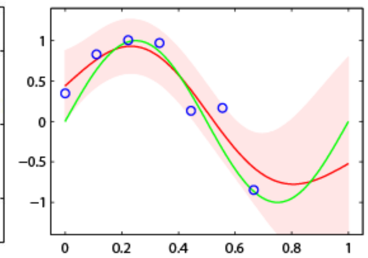
- **Lecture evaluation**
  - Please fill out the evaluation forms…

This Lecture: *Advanced Machine Learning*

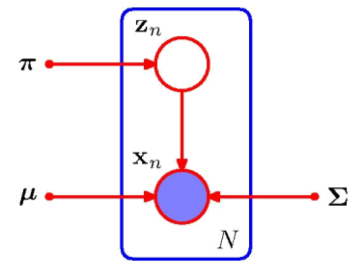
- Regression Approaches
  - Linear Regression
  - Regularization (Ridge, Lasso)
  - Kernels (Kernel Ridge Regression)
  - Gaussian Processes

$$f : \mathcal{X} \rightarrow \mathbb{R}$$

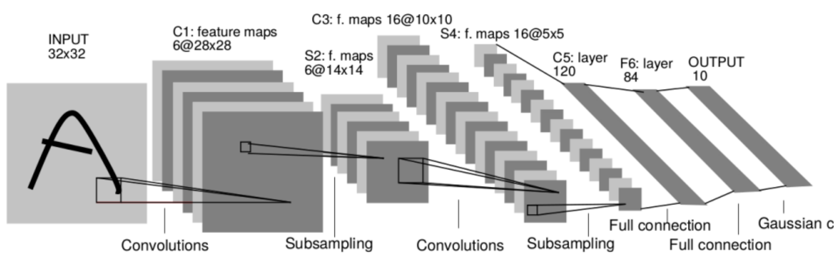




- Approximate Inference
  - Sampling Approaches
  - MCMC



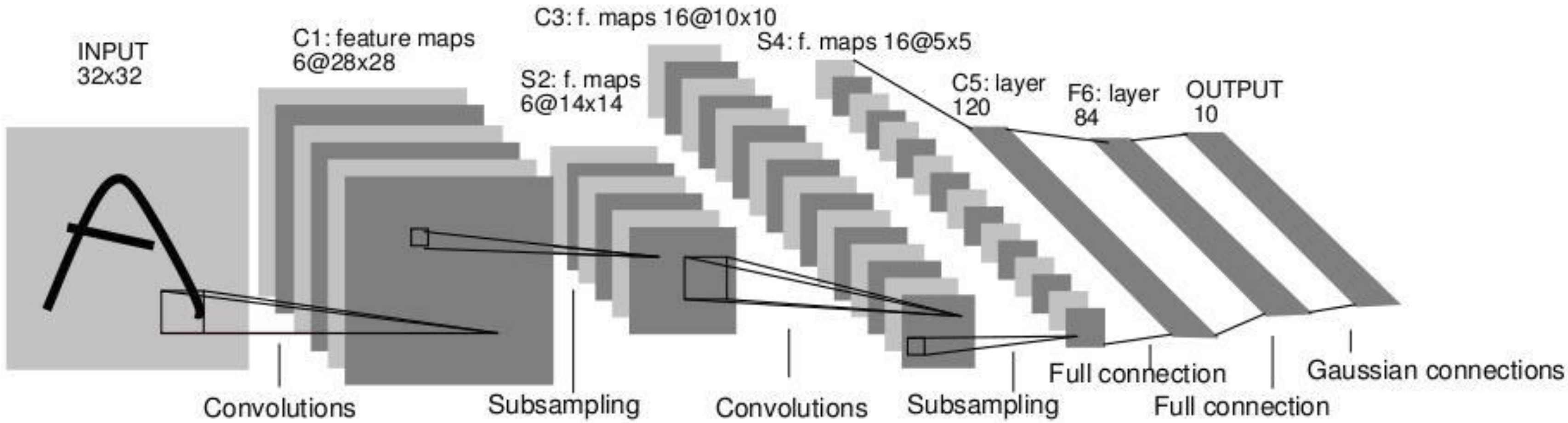
- Deep Learning
  - Linear Discriminants
  - Neural Networks
  - Backpropagation & Optimization
  - CNNs, RNNs, ResNets, etc.



Topics of This Lecture

- Recap: CNN Architectures
- Residual Networks
- Applications of CNNs
  - Object detection
  - Semantic segmentation
  - Face identification

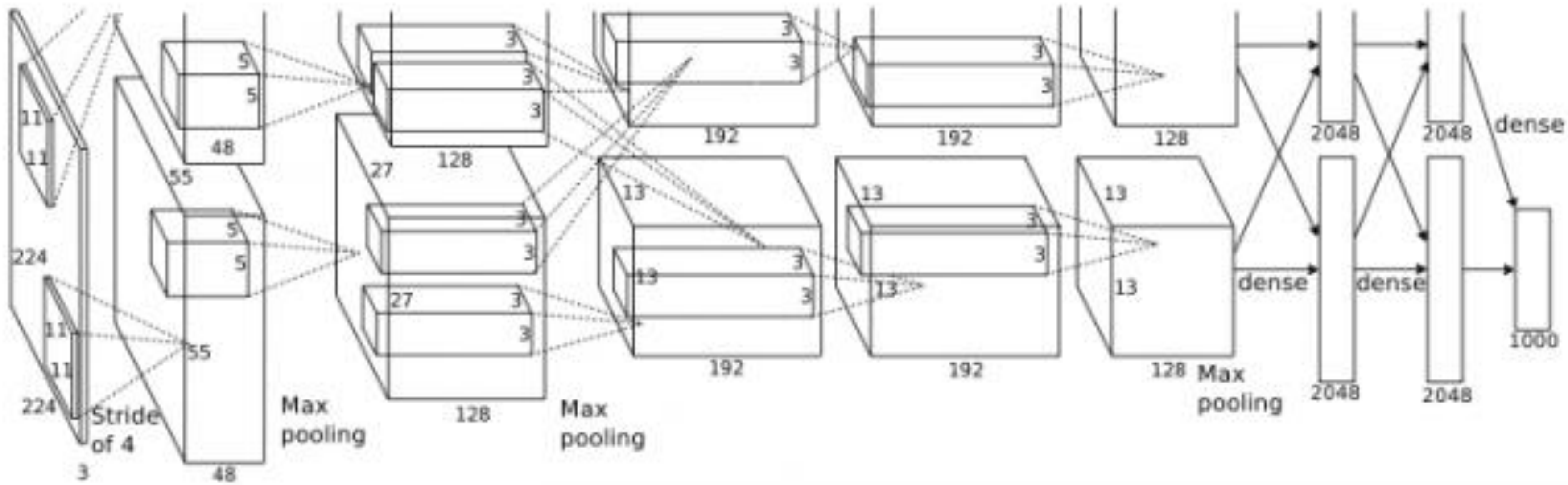
Recap: Convolutional Neural Networks



- Neural network with specialized connectivity structure
  - Stack multiple stages of feature extractors
  - Higher stages compute more global, more invariant features
  - Classification layer at the end

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, [Gradient-based learning applied to document recognition](), Proceedings of the IEEE 86(11): 2278–2324, 1998.

Recap: AlexNet (2012)



- Similar framework as LeNet, but
  - Bigger model (7 hidden layers, 650k units, 60M parameters)
  - More data (10^6 images instead of 10^3)
  - GPU implementation
  - Better regularization and up-to-date tricks for training (Dropout)

A. Krizhevsky, I. Sutskever, and G. Hinton, [ImageNet Classification with Deep Convolutional Neural Networks](), NIPS 2012.

Image source: A. Krizhevsky, I. Sutskever and G.E. Hinton, NIPS 2012

Recap: VGGNet (2014/15)

- **Main ideas**
  - Deeper network
  - Stacked convolutional layers with smaller filters (+ nonlinearity)
  - Detailed evaluation of all components
- **Results**
  - Improved ILSVRC top-5 error rate to 6.7%.

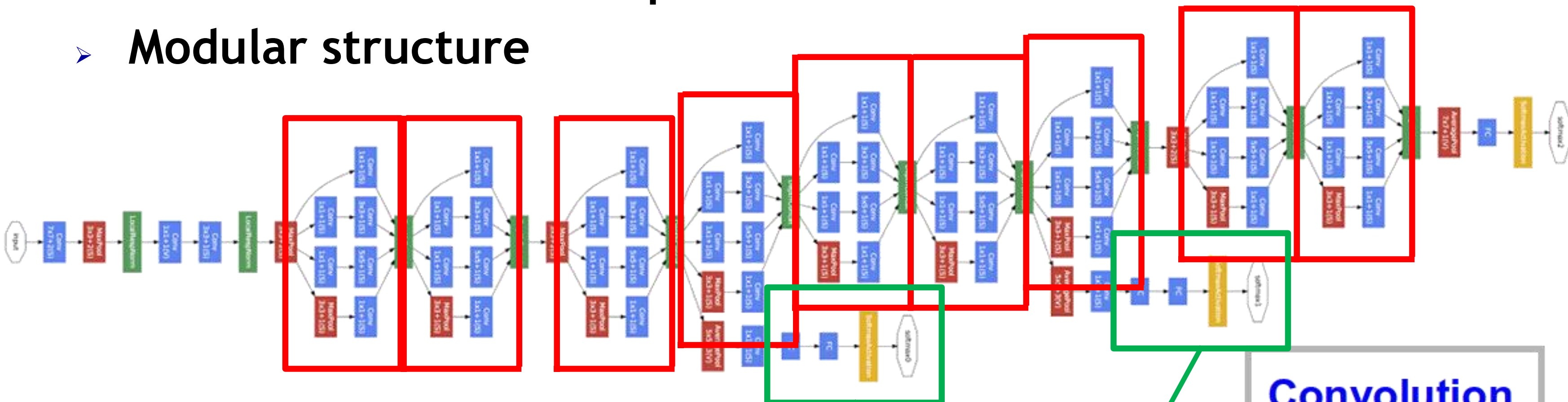
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	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					

Mainly used

Image source: Simonyan & Zisserman

Recap: GoogLeNet (2014)

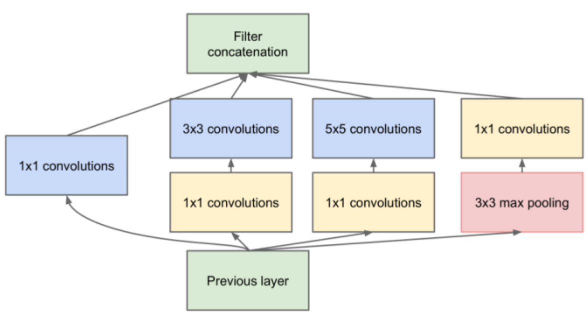
- Ideas:
  - Learn features at multiple scales
  - Modular structure



Inception module + copies

Auxiliary classification outputs for training the lower layers (deprecated)

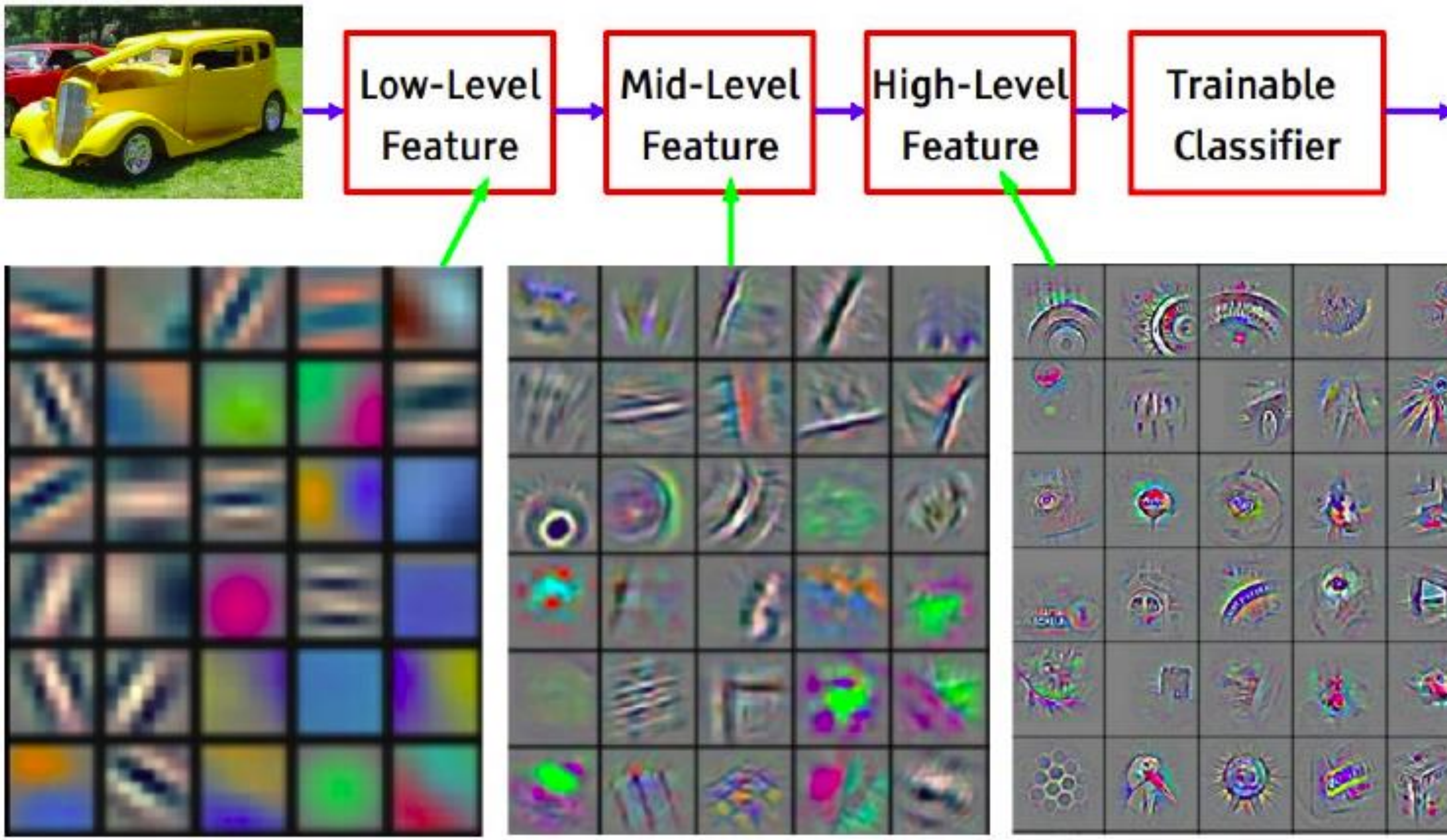
Convolution
Pooling
Softmax
Other



(b) Inception module with dimension reductions

Image source: Szegedy et al.

Recap: Visualizing CNNs



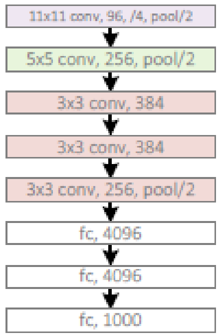
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Topics of This Lecture

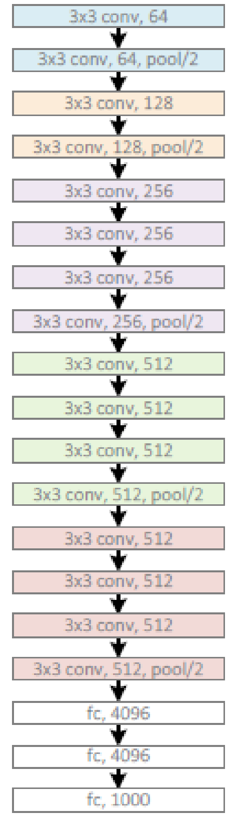
- Recap: CNN Architectures
- Residual Networks
- Applications of CNNs
  - Object detection
  - Semantic segmentation
  - Face identification

Newest Development: Residual Networks

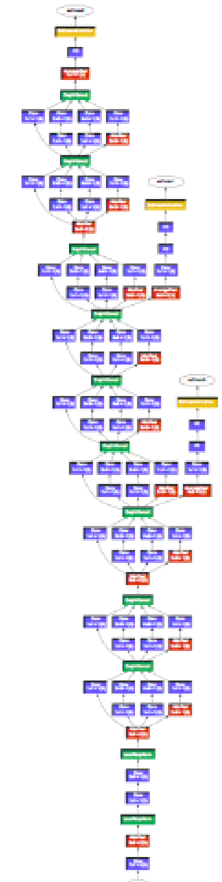
AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



GoogleNet, 22 layers
(ILSVRC 2014)



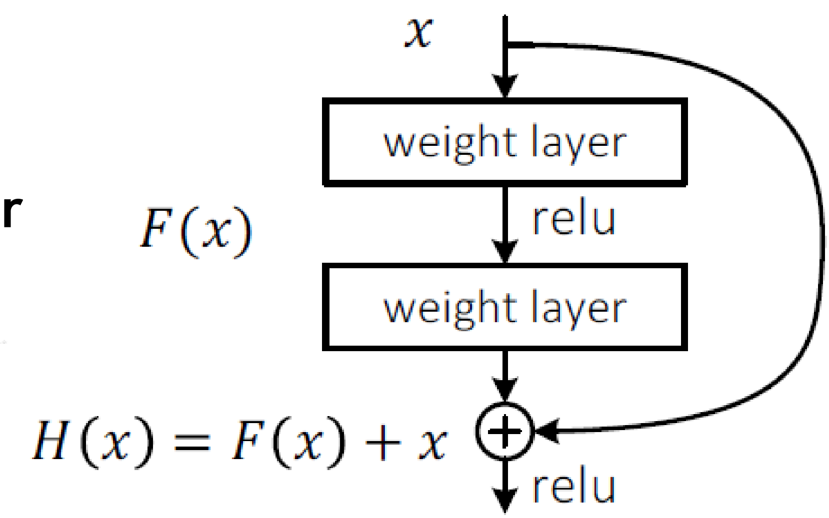
Newest Development: Residual Networks

AlexNet, 8 layers
(ILSVRC 2012)

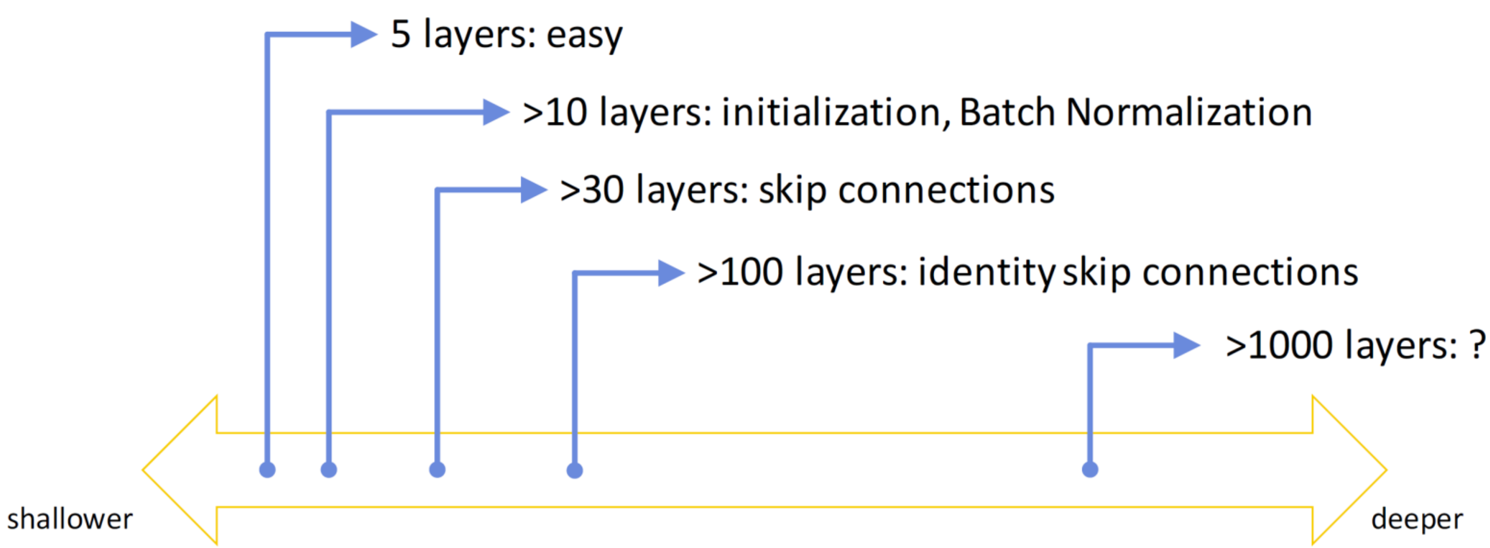
VGG, 19 layers
(ILSVRC 2014)

ResNet, 152 layers
(ILSVRC 2015)

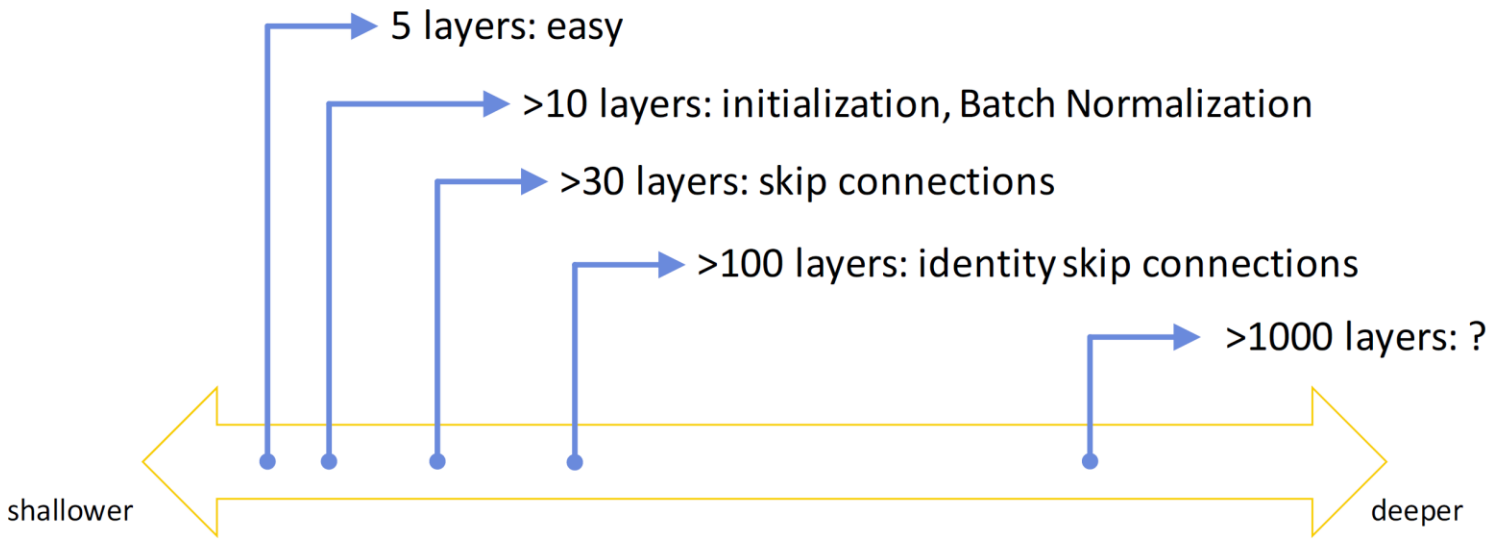
- Core component
  - Skip connections bypassing each layer
  - Better propagation of gradients to the deeper layers



Spectrum of Depth



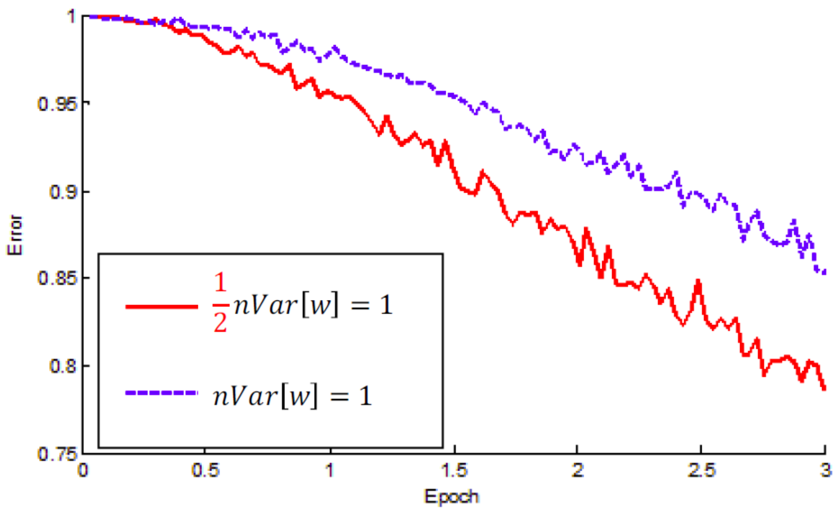
Spectrum of Depth



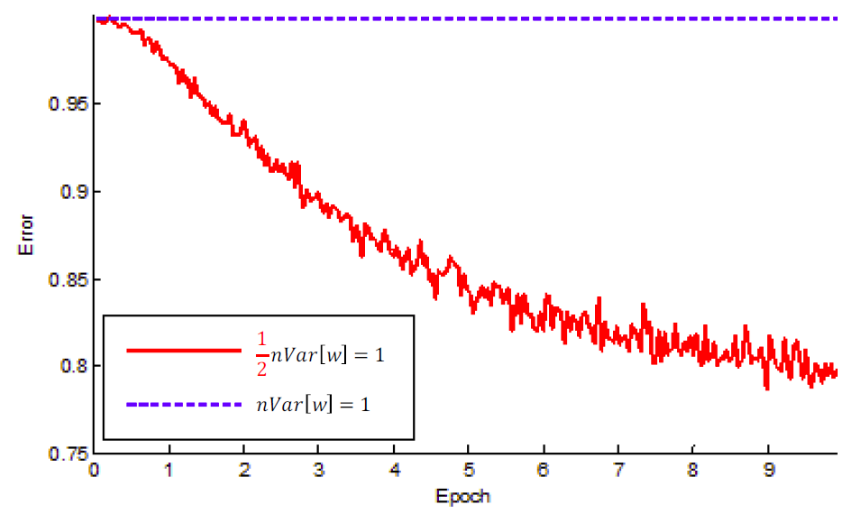
- **Deeper models are more powerful**
  - **But training them is harder.**
  - **Main problem: getting the gradients back to the early layers**
  - **The deeper the network, the more effort is required for this.**

Initialization

22-layer ReLU net:
good init converges faster



30-layer ReLU net:
good init is able to converge

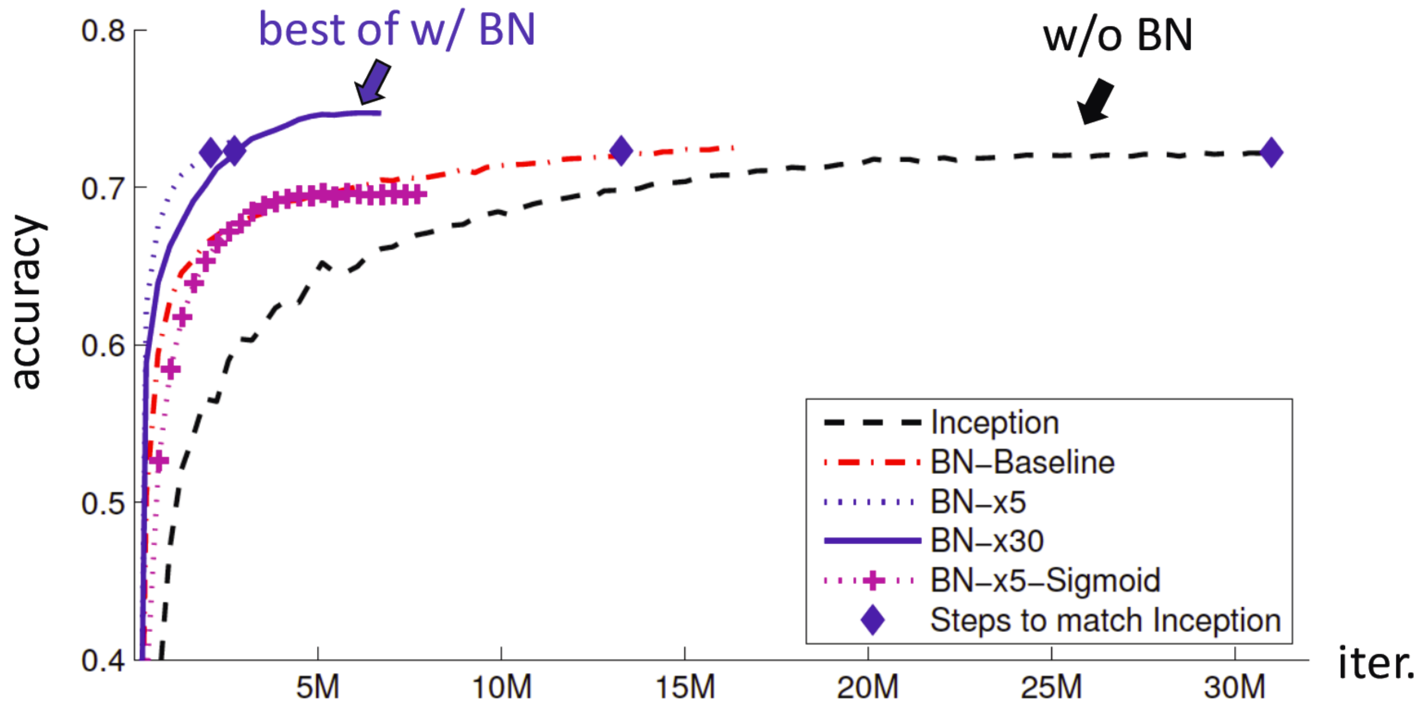


- Importance of proper initialization (Recall Lecture 11)
  - Glorot initialization for tanh nonlinearities
  - He initialization for ReLU nonlinearities

$\Rightarrow$ For deep networks, this really makes a difference!

Slide credit: Kaiming He

Batch Normalization



- **Effect of batch normalization**
  - **Greatly improved speed of convergence**

Image source: Ioffe & Szegedy

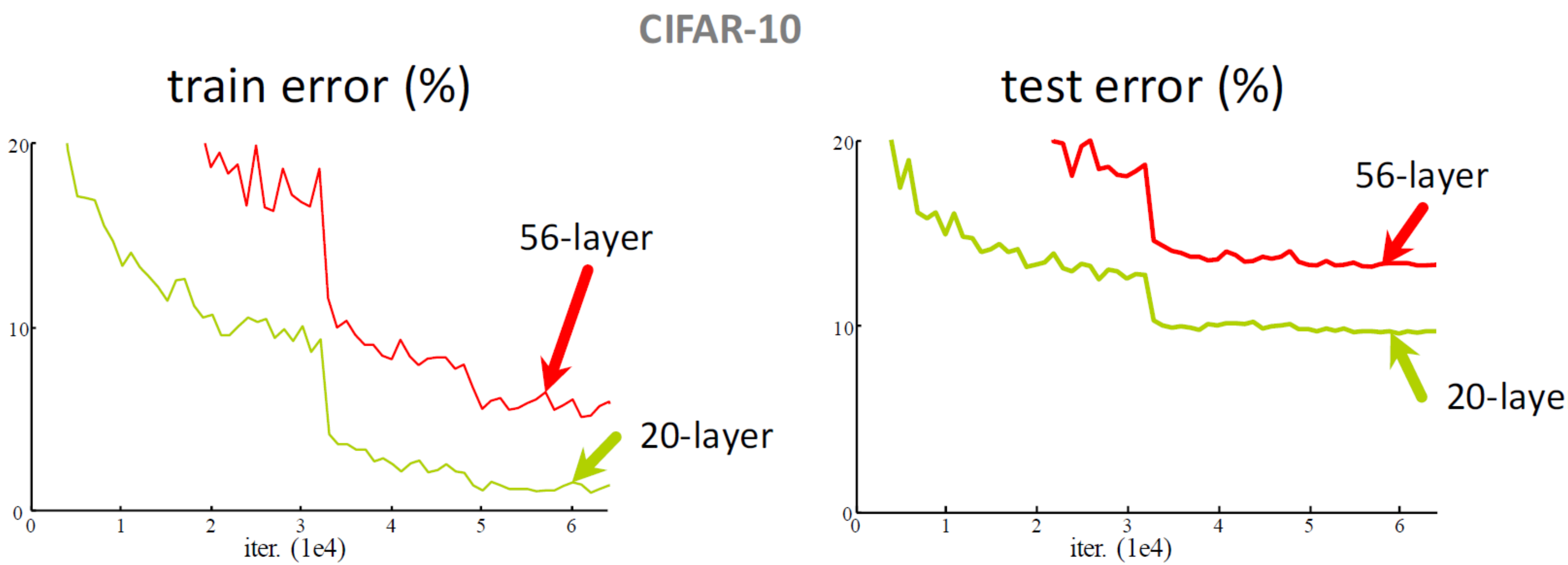
Going Deeper

- Checklist
  - Initialization ok
  - Batch normalization ok

  - Are we now set?
    - Is learning better networks now as simple as stacking more layers?

Slide credit: Kaiming He

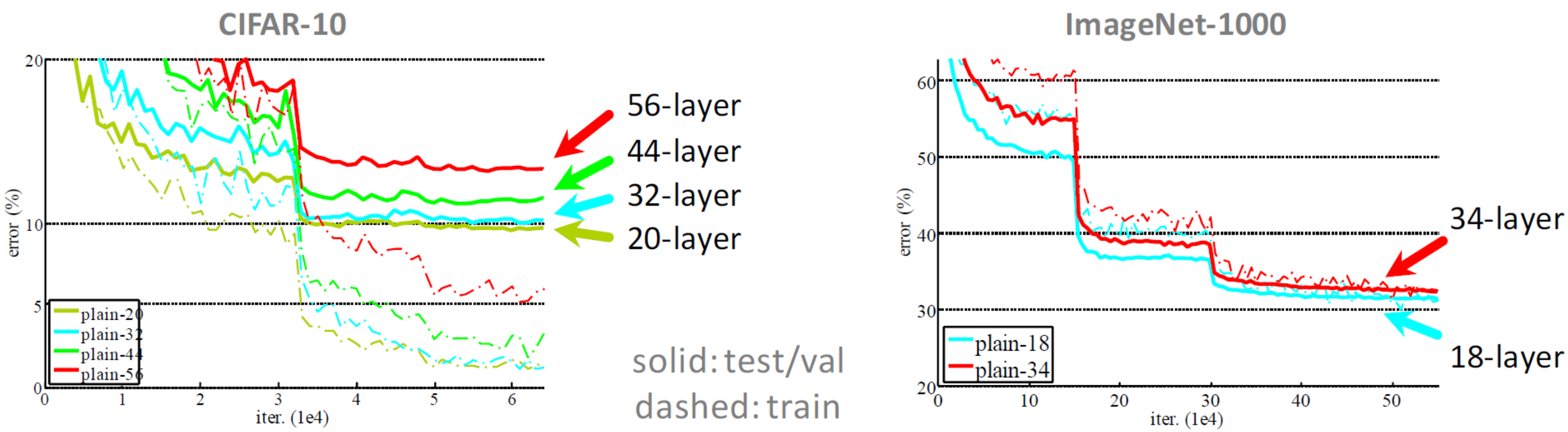
Simply Stacking Layers?



- **Experiment going deeper**
  - Plain nets: stacking 3×3 convolution layers

  ⇒ 56-layer net has higher training error than 20-layer net

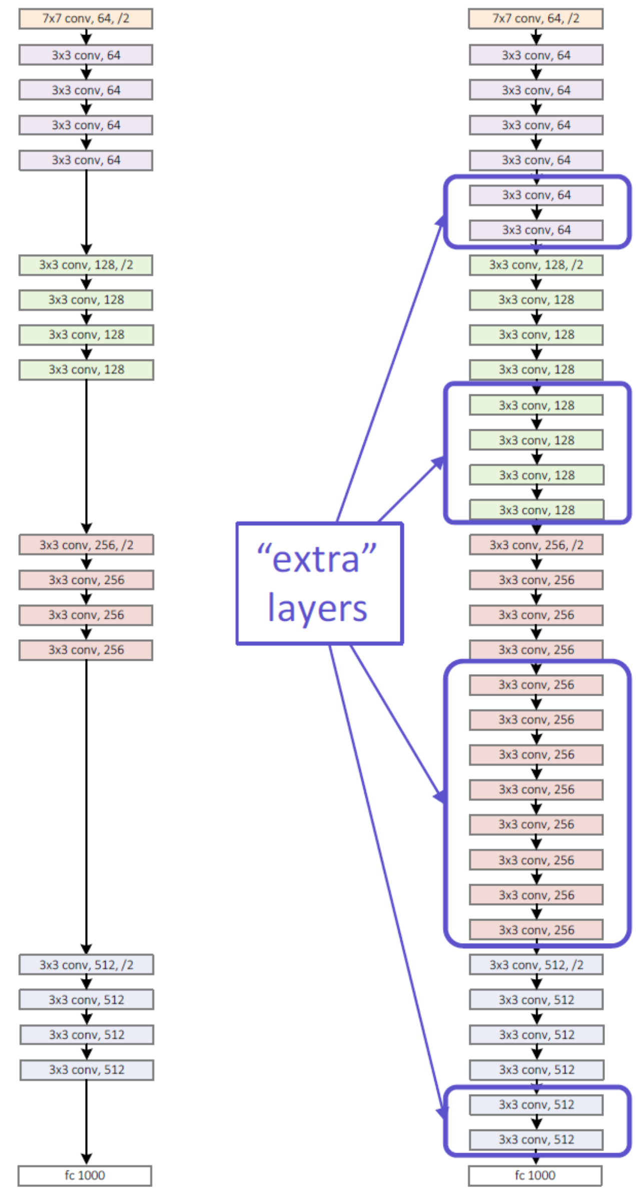
Simply Stacking Layers?



- **General observation**
  - **Overly deep networks have higher training error**
  - **A general phenomenon, observed in many training sets**

Why Is That???

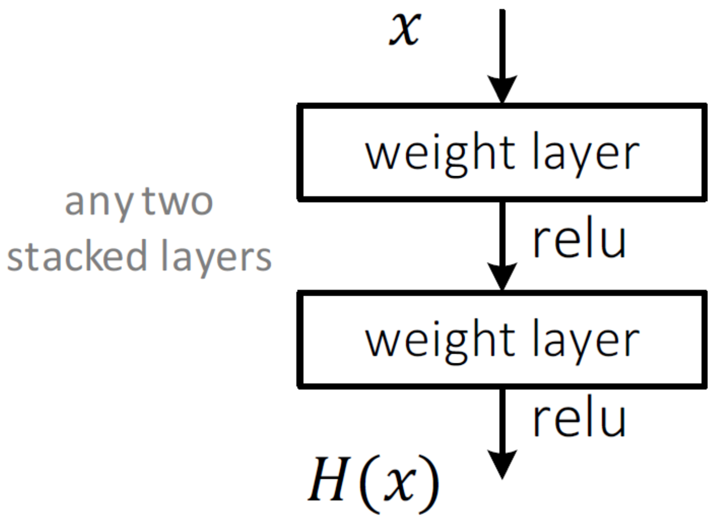
- A deeper model should not have higher training error!
  - Richer solution space should allow it to find better solutions
- Solution by construction
  - Copy the original layers from a learned shallower model
  - Set the extra layers as identity
  - Such a network should achieve at least the same low training error.
- Reason: Optimization difficulties
  - Solvers cannot find the solution when going deeper…



Slide credit: Kaiming He

Deep Residual Learning

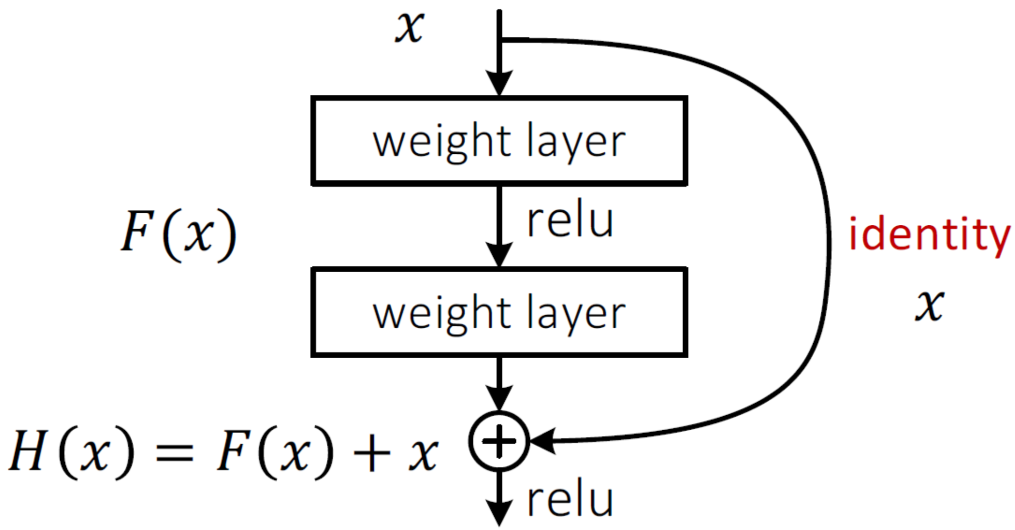
- **Plain net**



  - **$H(x)$ is any desired mapping**
  - **Hope the 2 weight layers fit $H(x)$**

Deep Residual Learning

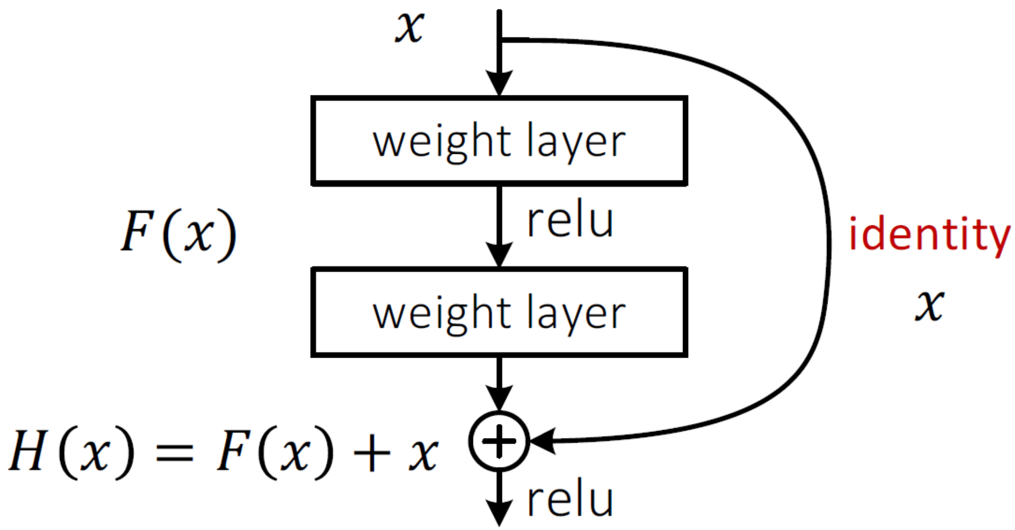
- **Residual net**



- $H(x)$ **is any desired mapping**
- ~~**Hope the 2 weight layers fit** $H(x)$~~
- **Hope the 2 weight layers fit** $F(x)$
  Let $H(x) = F(x) + x$

Deep Residual Learning

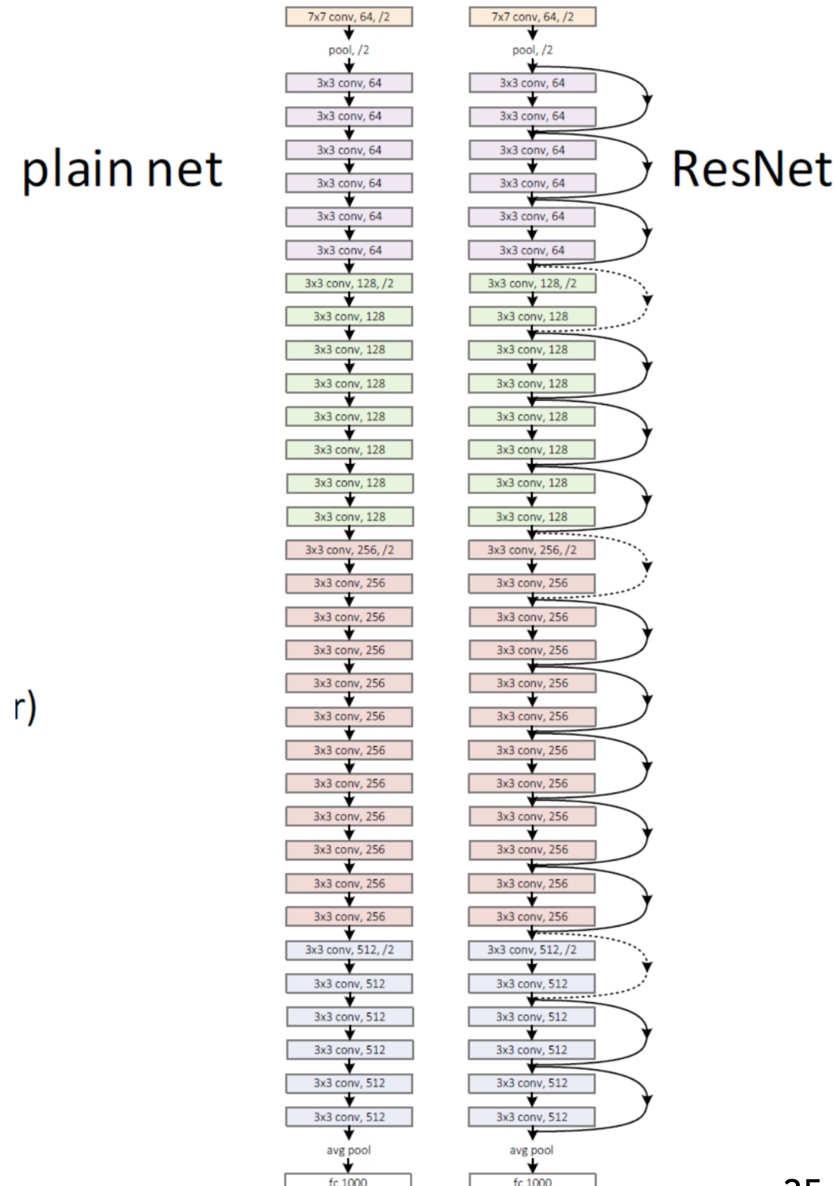
- $F(x)$ is a **residual** mapping w.r.t. **identity**



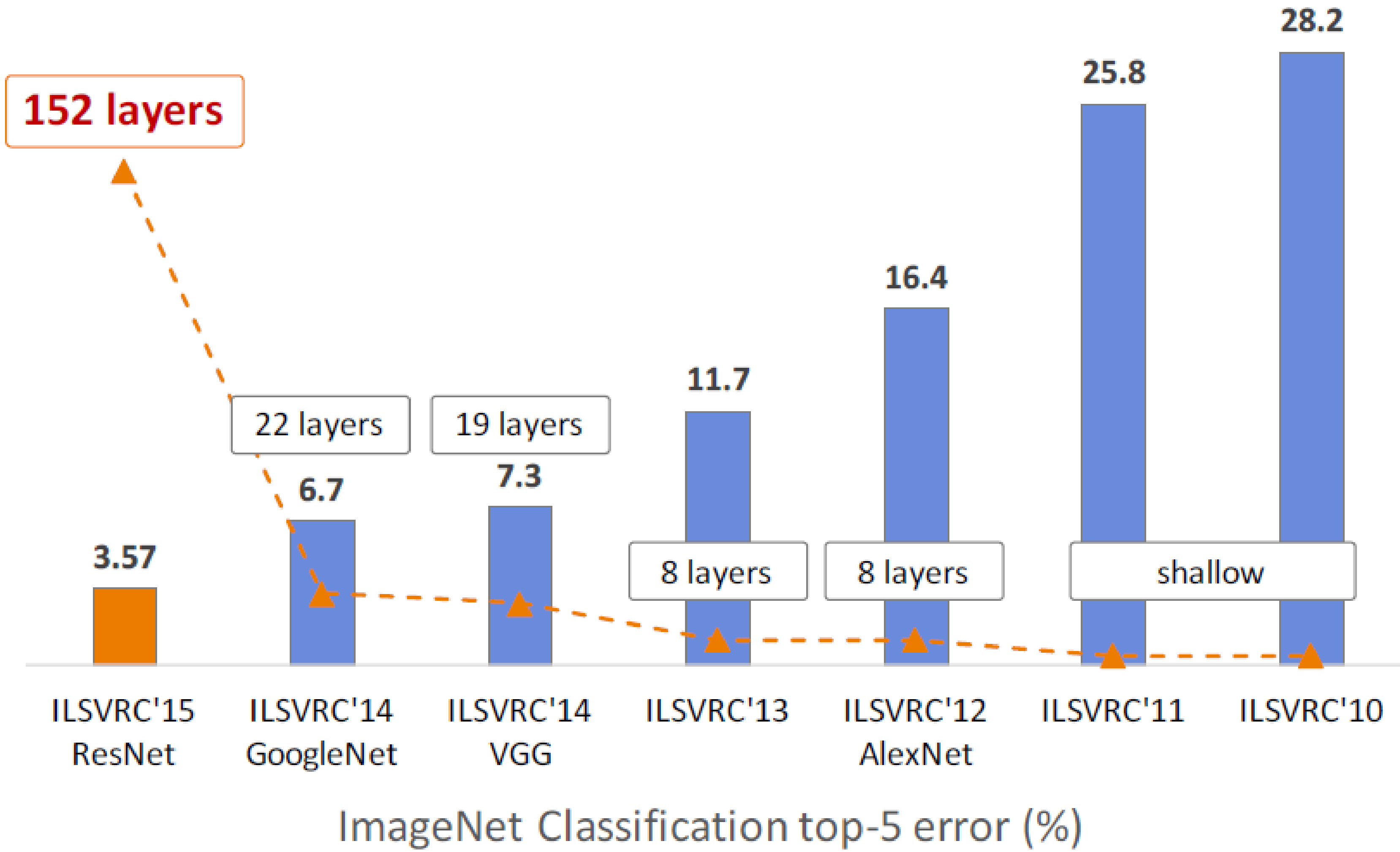
  - If identity were optimal, it is easy to set weights as 0
  - If optimal mapping is closer to identity, it is easier to find small fluctuations
  - Further advantage: direct path for the gradient to flow to the previous stages

Network Design

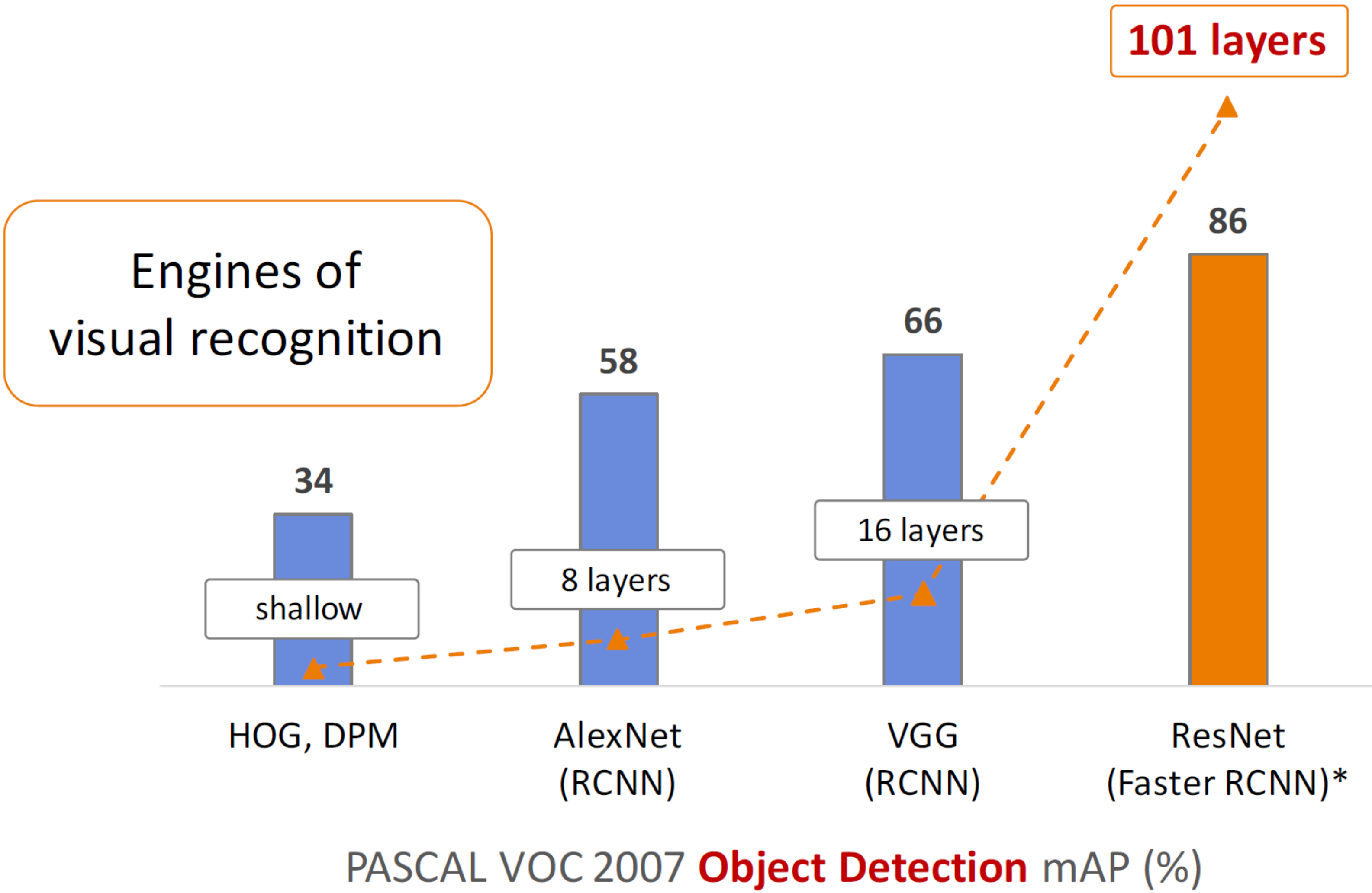
- Simple, VGG-style design
  - (Almost) all 3×3 convolutions
  - Spatial size /2 ⇒ #filters · 2 (same complexity per layer)
  - Batch normalization
  - ⇒ Simple design, just deep.



ImageNet Performance



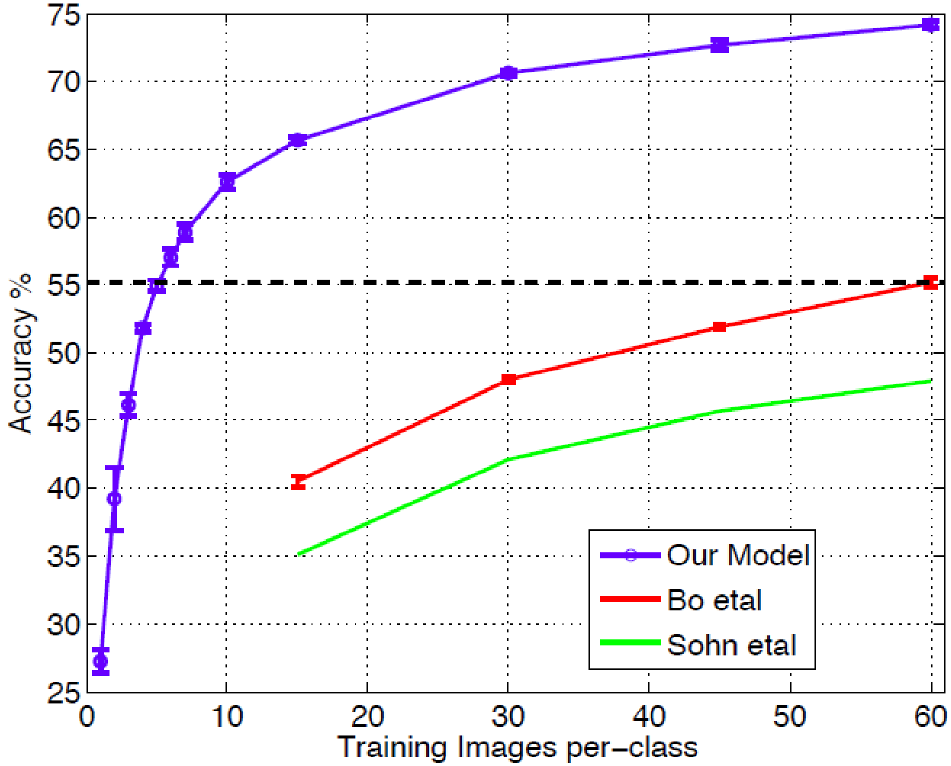
PASCAL VOC Object Detection Performance



Topics of This Lecture

- Recap: CNN Architectures
- Residual Networks
- Applications of CNNs
  - Object detection
  - Semantic segmentation
  - Face identification

The Learned Features are Generic



state of the art level (pre-CNN)

- **Experiment: feature transfer**
  - Train AlexNet-like network on ImageNet
  - Chop off last layer and train classification layer on CalTech256

  ⇒ State of the art accuracy already with only 6 training images!

Image source: M. Zeiler, R. Fergus

Transfer Learning with CNNs

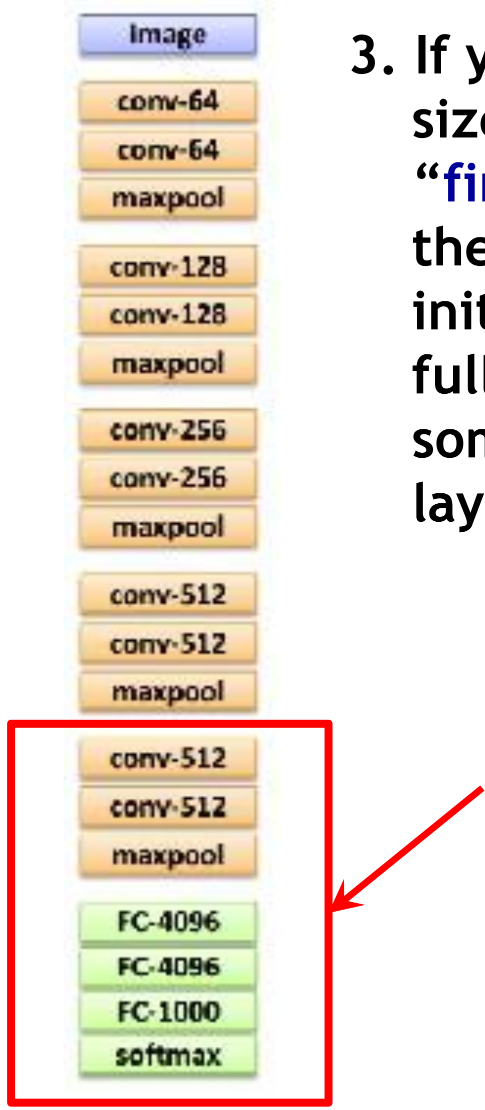
1. Train on ImageNet

2. If small dataset: fix all weights (treat CNN as fixed feature extractor), retrain only the classifier

I.e., swap the Softmax layer at the end

Transfer Learning with CNNs

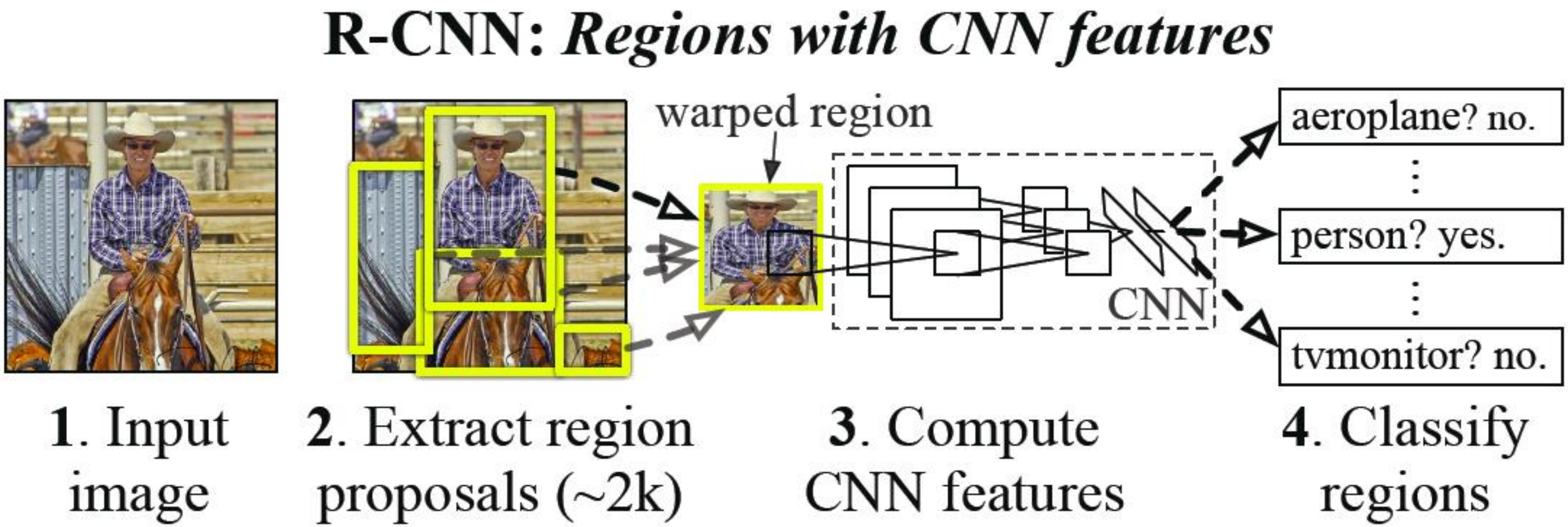
1. Train on ImageNet



3. If you have medium sized dataset, "finetune" instead: use the old weights as initialization, train the full network or only some of the higher layers.

Retrain bigger portion of the network

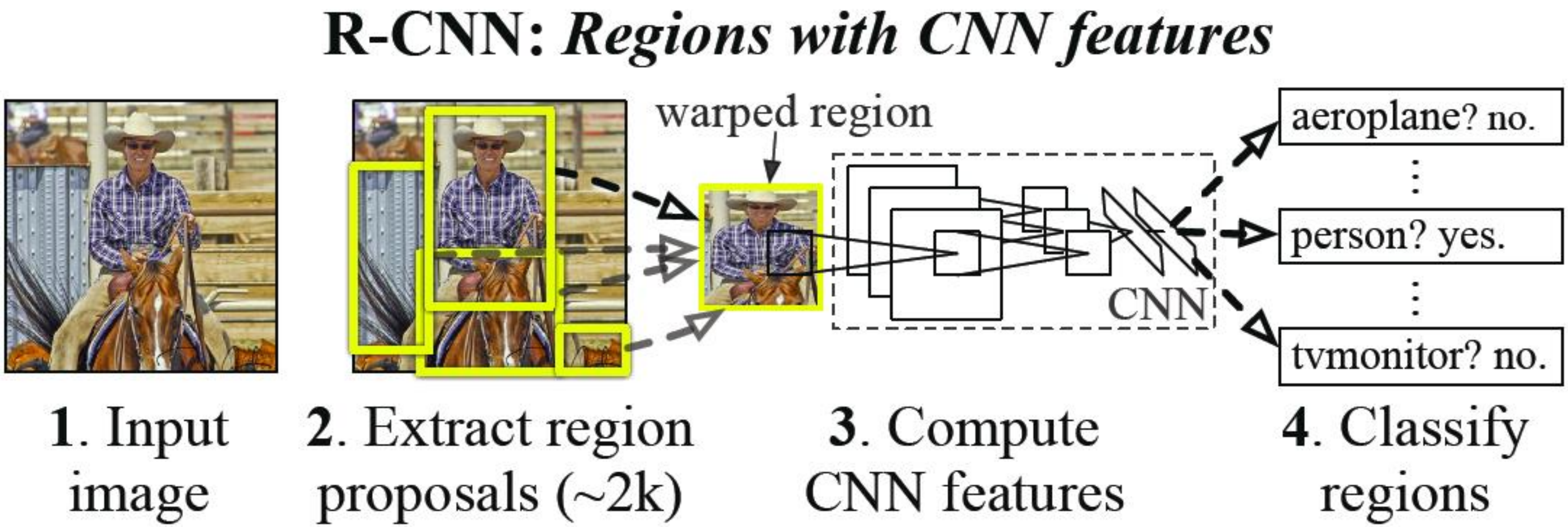
Other Tasks: Object Detection



- **Key ideas**
  - **Extract region proposals (Selective Search)**
  - **Use a pre-trained/fine-tuned classification network as feature extractor (initially AlexNet, later VGGNet) on those regions**

R. Girshick, J. Donahue, T. Darrell, and J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, CVPR 2014

Object Detection: R-CNN



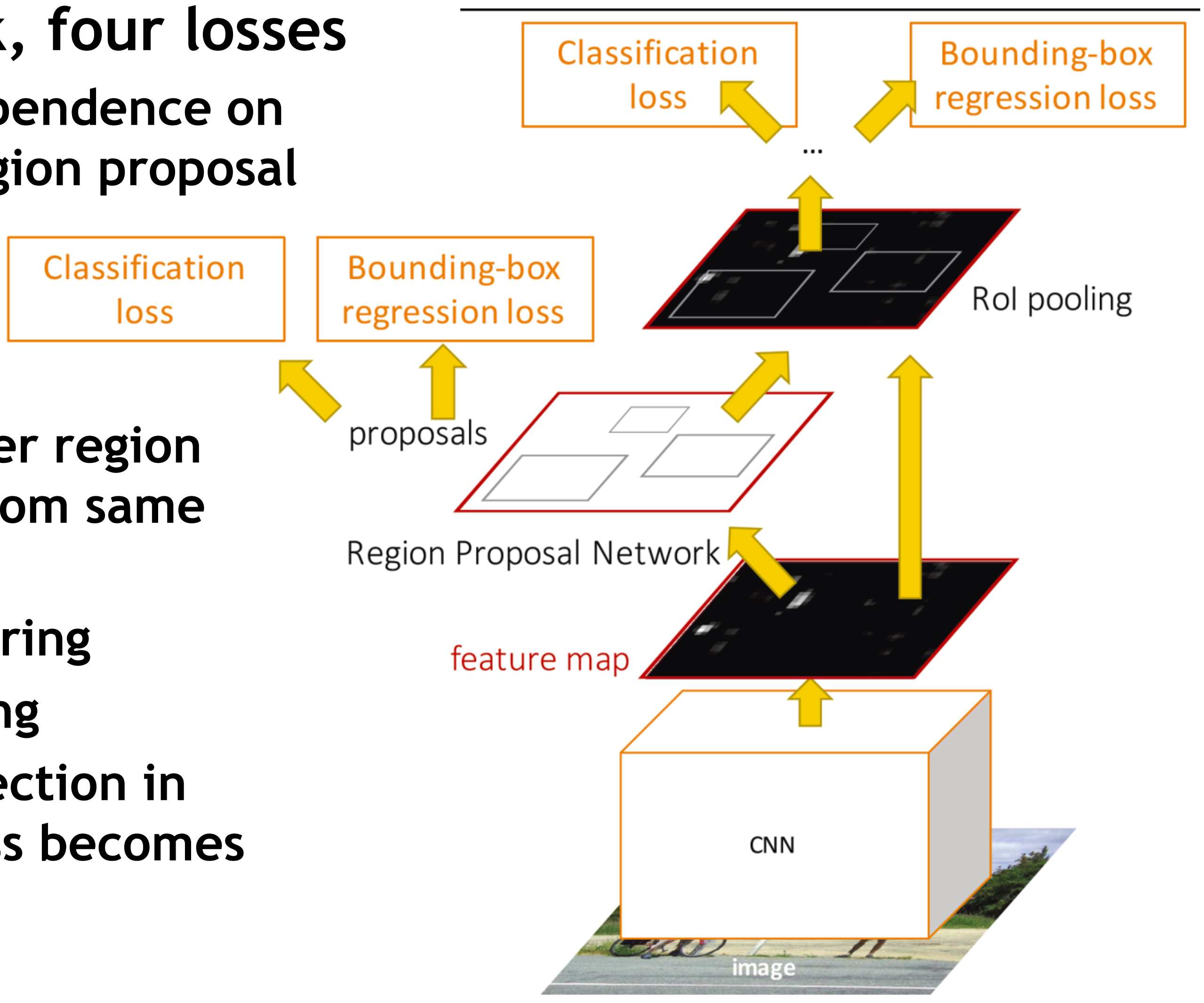
- **Results on PASCAL VOC Detection benchmark**
  - **Pre-CNN state of the art: 35.1% mAP [Uijlings et al., 2013]**
    33.4% mAP DPM
  - **R-CNN: 53.7% mAP**

R. Girshick, J. Donahue, T. Darrell, and J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, CVPR 2014

Most Recent Version: Faster R-CNN

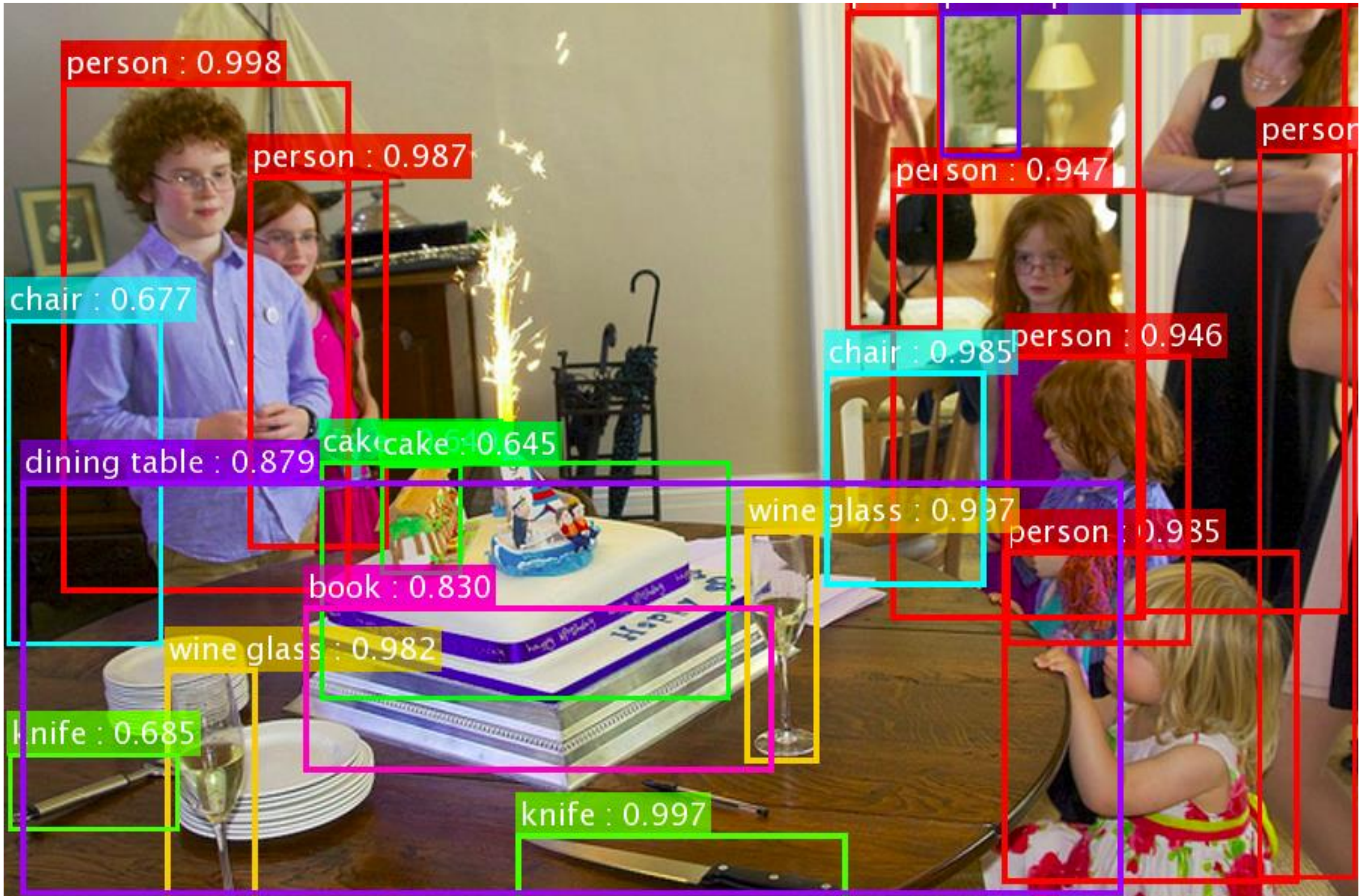
- One network, four losses
  - Remove dependence on external region proposal algorithm.
  - Instead, infer region proposals from same CNN.
  - Feature sharing
  - Joint training

  ⇒ Object detection in a single pass becomes possible.



Slide credit: Ross Girshick

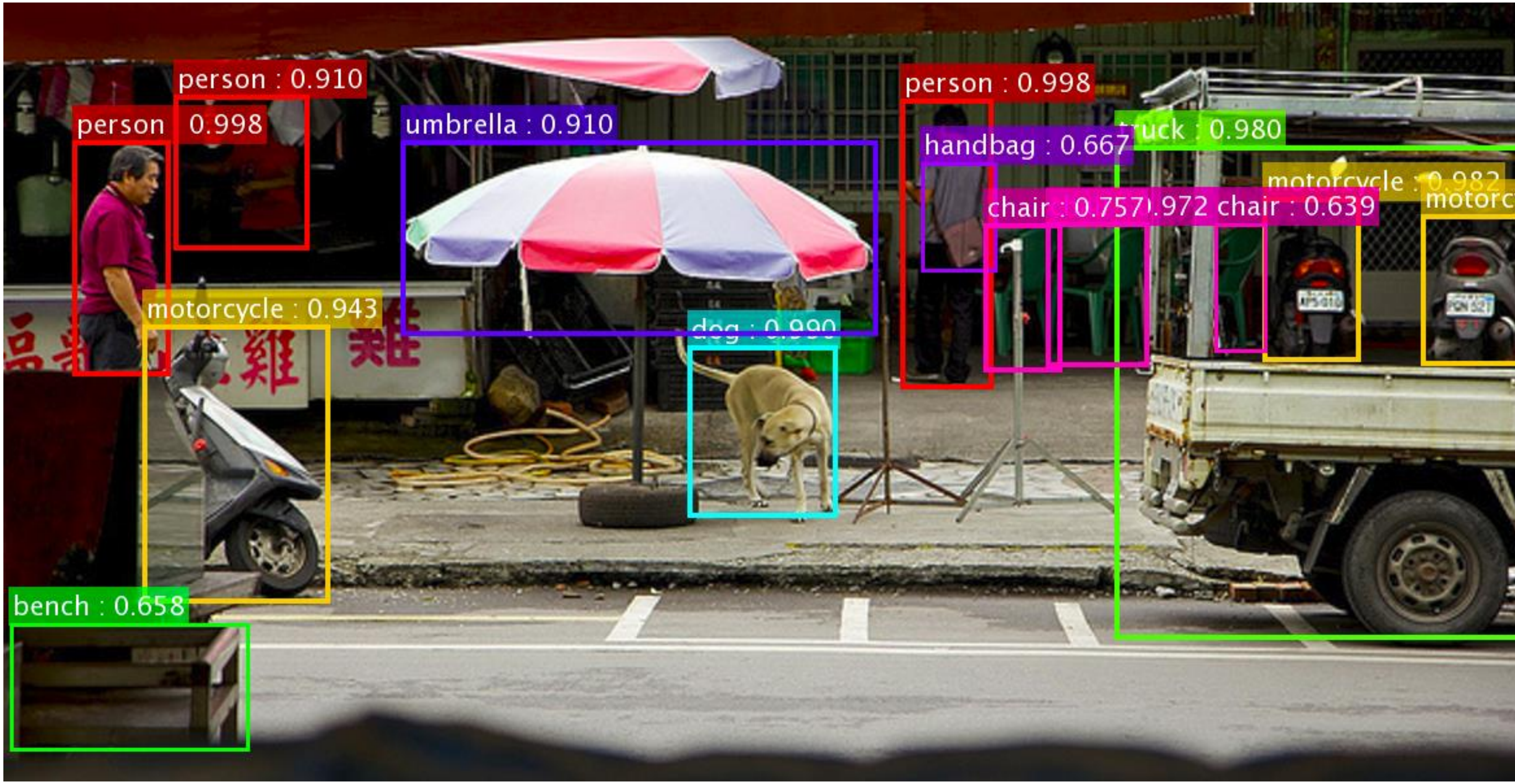
Faster R-CNN (based on ResNets)



K. He, X. Zhang, S. Ren, J. Sun, [Deep Residual Learning for Image Recognition](), CVPR 2016.

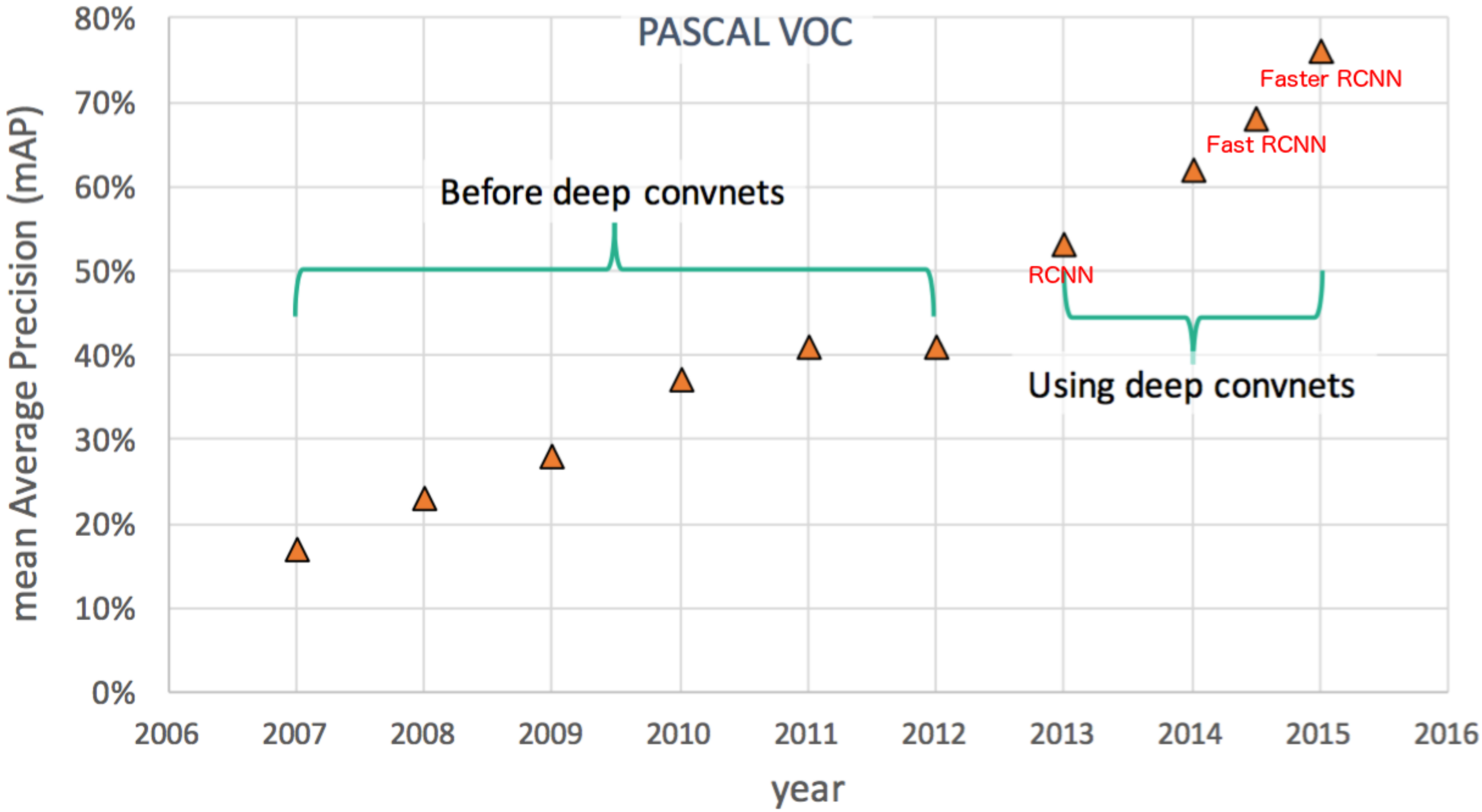
RWTH AACHEN UNIVERSITY

Faster R-CNN (based on ResNets)



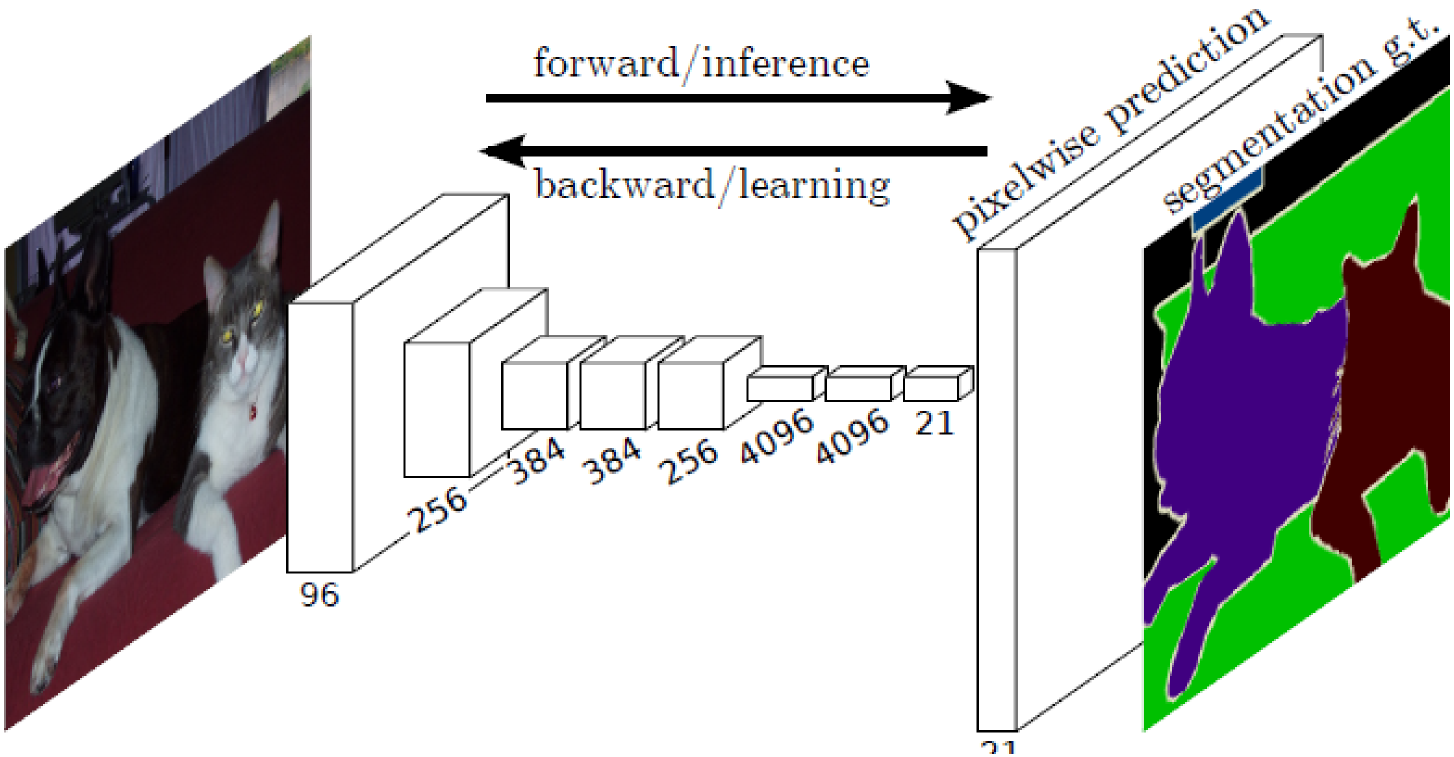
K. He, X. Zhang, S. Ren, J. Sun, [Deep Residual Learning for Image Recognition](), CVPR 2016.

Object Detection Performance



Slide credit: Ross Girshick

Semantic Image Segmentation

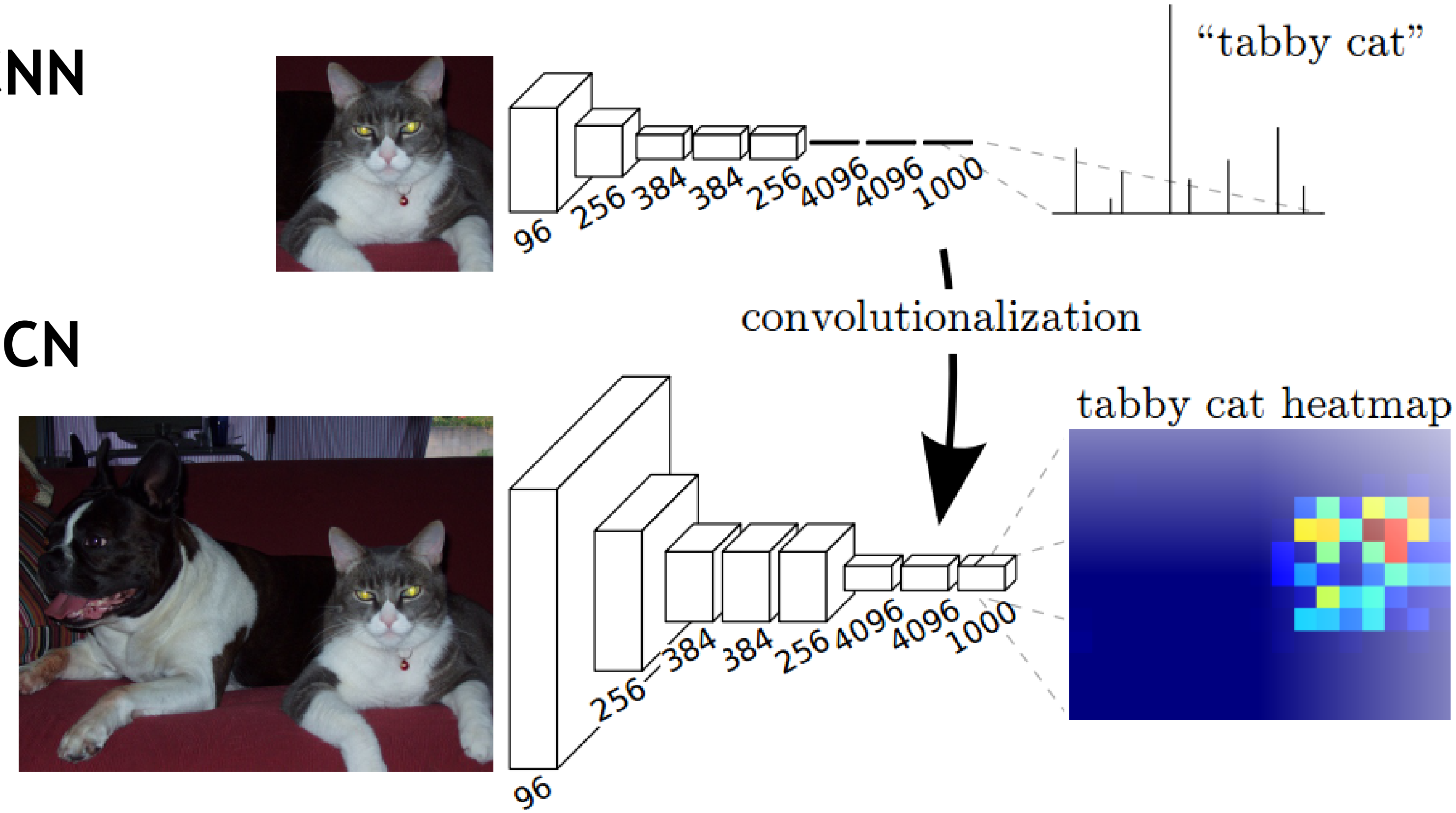


- **Perform pixel-wise prediction task**
  - Usually done using Fully Convolutional Networks (FCNs)
    - All operations formulated as convolutions
    - Advantage: can process arbitrarily sized images

Image source: Long, Shelhamer, Darrell

CNNs vs. FCNs

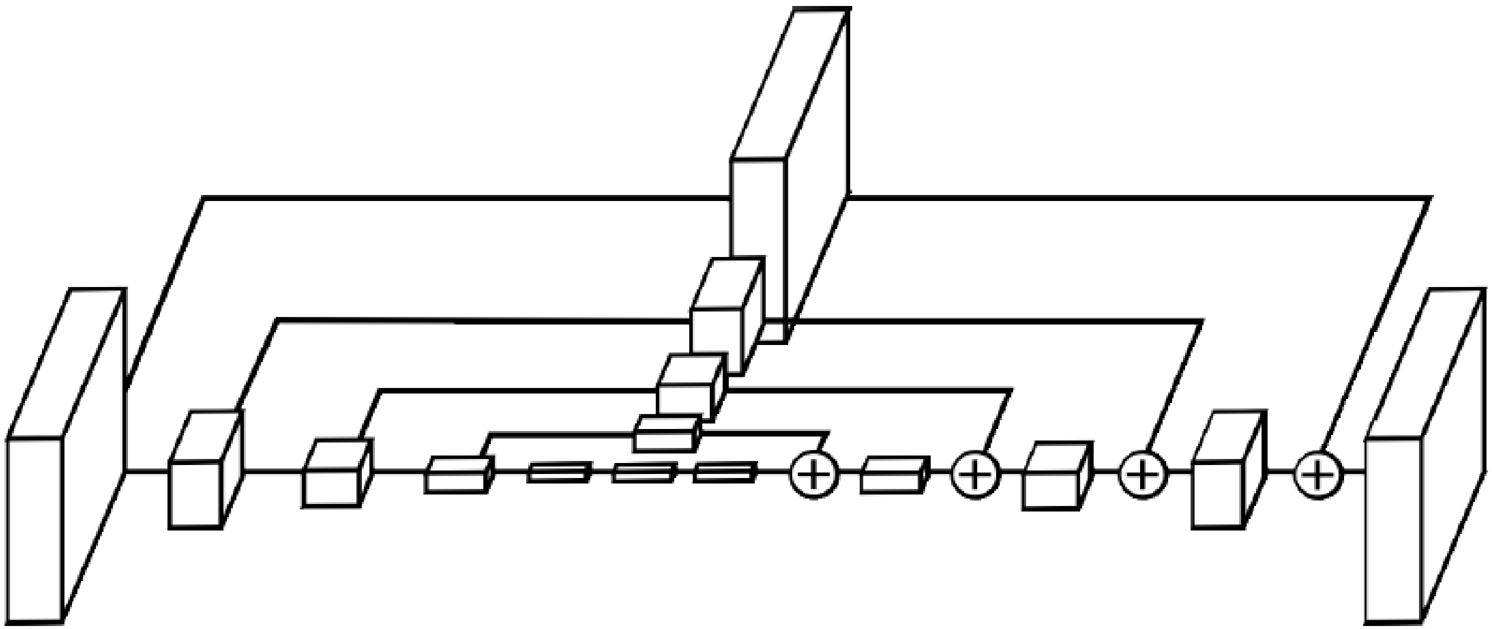
- **CNN**
- **FCN**



- **Intuition**
  - Think of FCNs as performing a sliding-window classification, producing a heatmap of output scores for each class

Image source: Long, Shelhamer, Darrell

Semantic Image Segmentation



- **Encoder-Decoder Architecture**
  - Problem: FCN output has low resolution
  - Solution: perform upsampling to get back to desired resolution
  - Use skip connections to preserve higher-resolution information

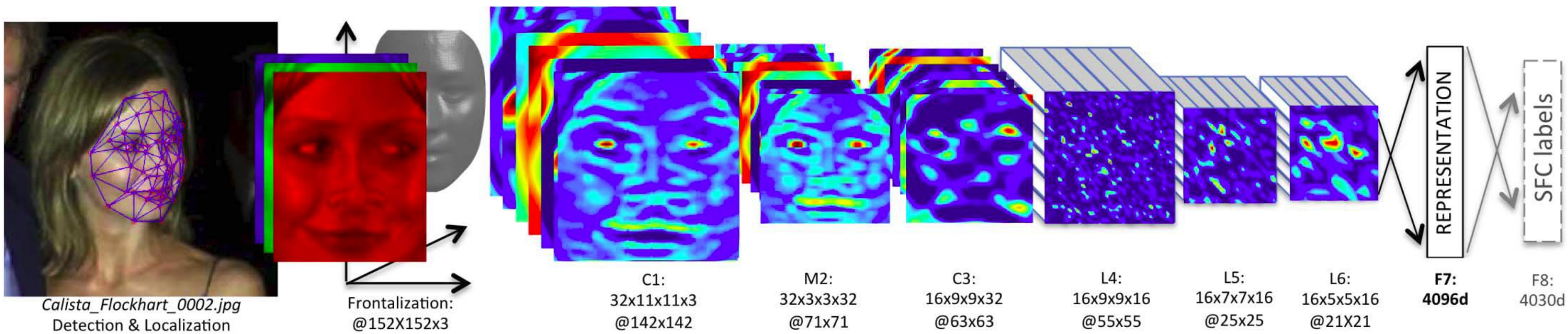
Image source: Newell et al.

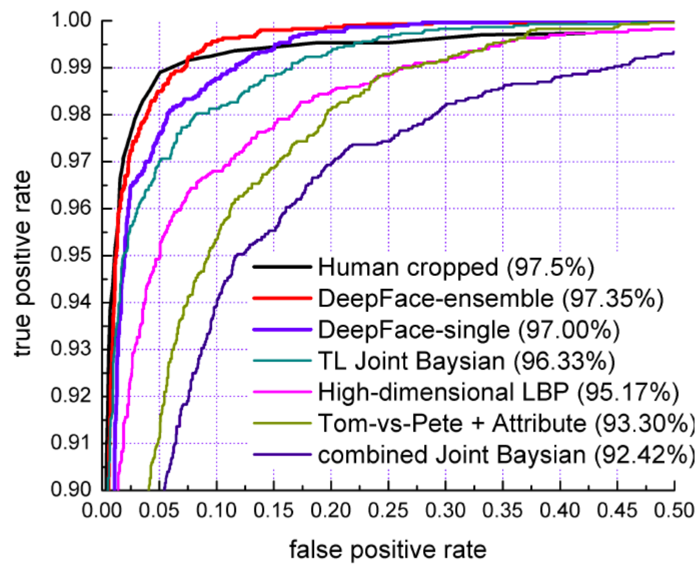
Semantic Segmentation

[Pohlen, Hermans, Mathias, Leibe, arXiv 2016]

- **More recent results**
  - **Based on an extension of ResNets**

Other Tasks: Face Identification





Y. Taigman, M. Yang, M. Ranzato, L. Wolf, DeepFace: Closing the Gap to Human-Level Performance in Face Verification, CVPR 2014

Slide credit: Svetlana Lazebnik

References: Computer Vision Tasks

- **Object Detection**
  - R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, CVPR 2014.
  - S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, NIPS 2015.
  - J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified Real-Time Object Detection, CVPR 2016.
  - W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C-Y. Fu, A.C. Berg, SSD: Single Shot Multi Box Detector, ECCV 2016.

References: Computer Vision Tasks

- **Semantic Segmentation**
  - J. Long, E. Shelhamer, T. Darrell, Fully Convolutional Networks for Semantic Segmentation, CVPR 2015.
  - H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid Scene Parsing Network, arXiv 1612.01105, 2016.