

# Machine Learning – Lecture 16

## Convolutional Neural Networks II

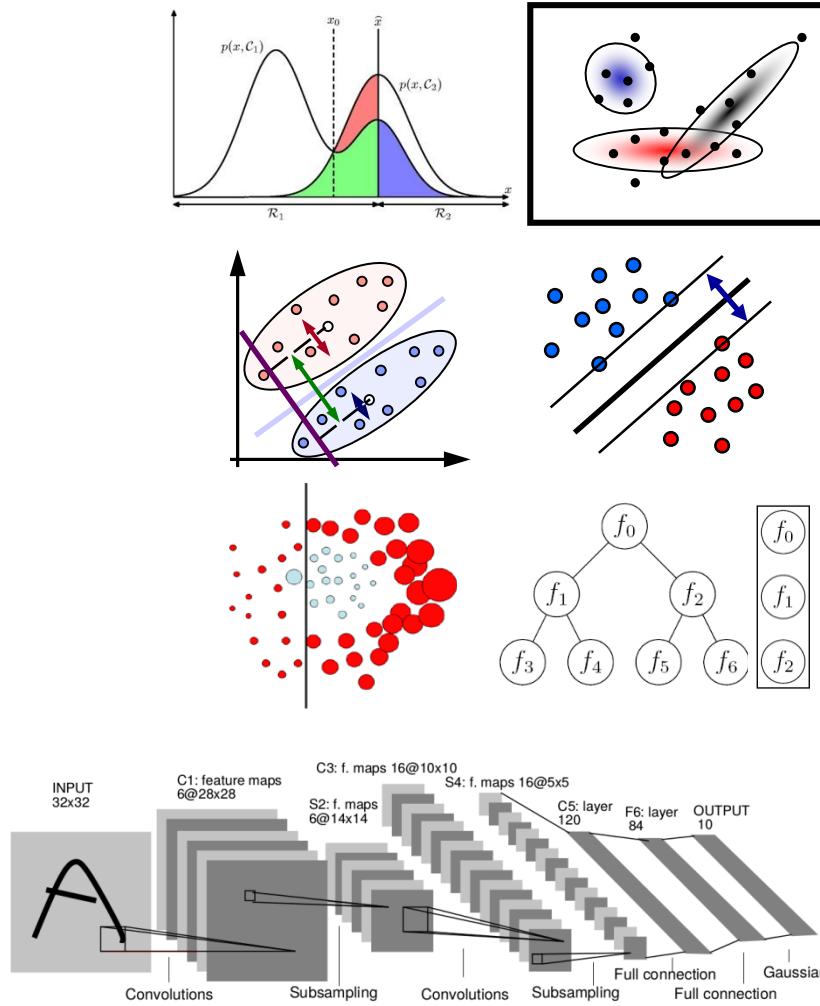
18.12.2017

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

leibe@vision.rwth-aachen.de

# Course Outline

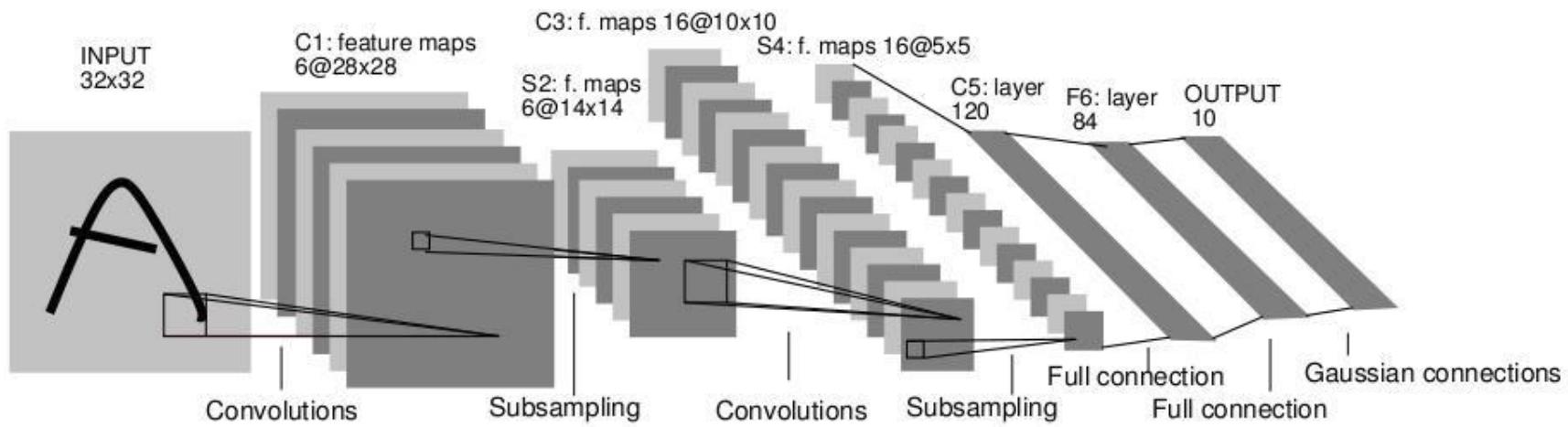
- Fundamentals
  - Bayes Decision Theory
  - Probability Density Estimation
- Classification Approaches
  - Linear Discriminants
  - Support Vector Machines
  - Ensemble Methods & Boosting
  - Random Forests
- Deep Learning
  - Foundations
  - Convolutional Neural Networks
  - Recurrent Neural Networks



# Topics of This Lecture

- Recap: CNNs
- CNN Architectures
  - LeNet
  - AlexNet
  - VGGNet
  - GoogLeNet
  - ResNets
- Visualizing CNNs
  - Visualizing CNN features
  - Visualizing responses
  - Visualizing learned structures
- Applications

# Recap: Convolutional Neural Networks

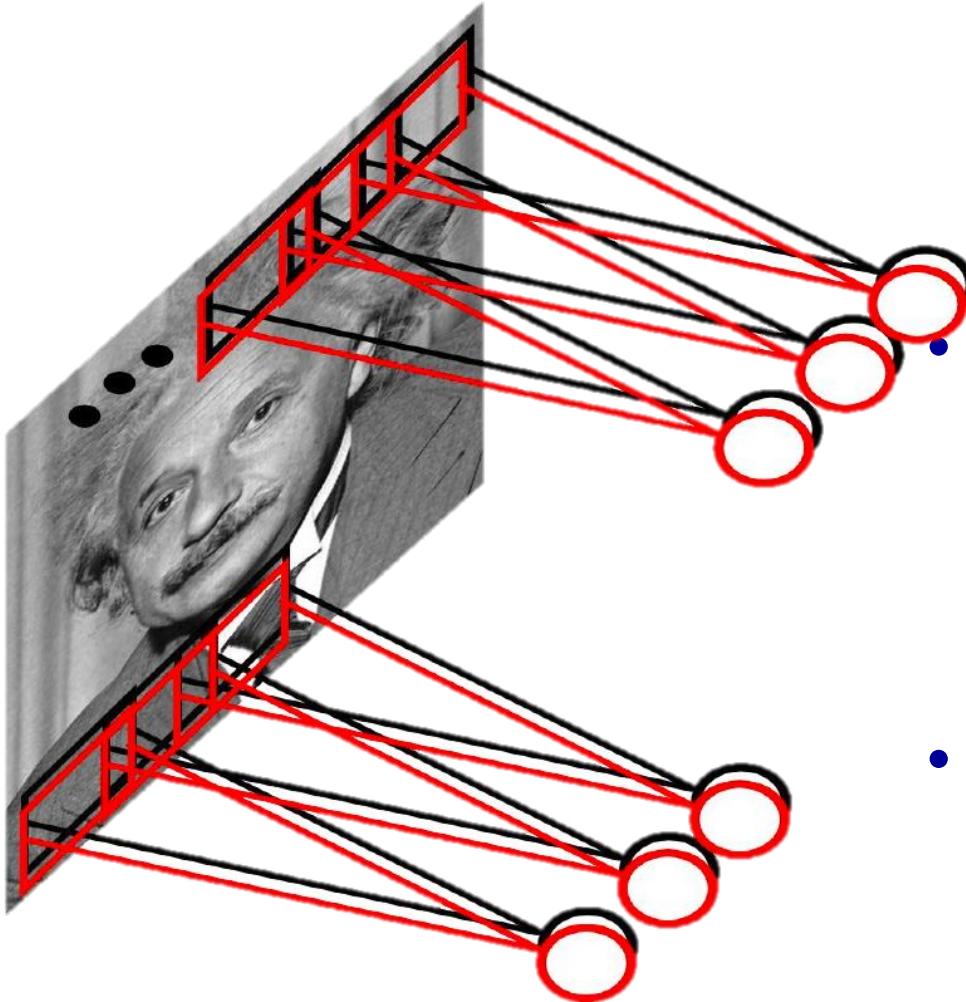


- Neural network with specialized connectivity structure
  - Stack multiple stages of feature extractors
  - Higher stages compute more global, more invariant features
  - Classification layer at the end

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: Intuition of CNNs

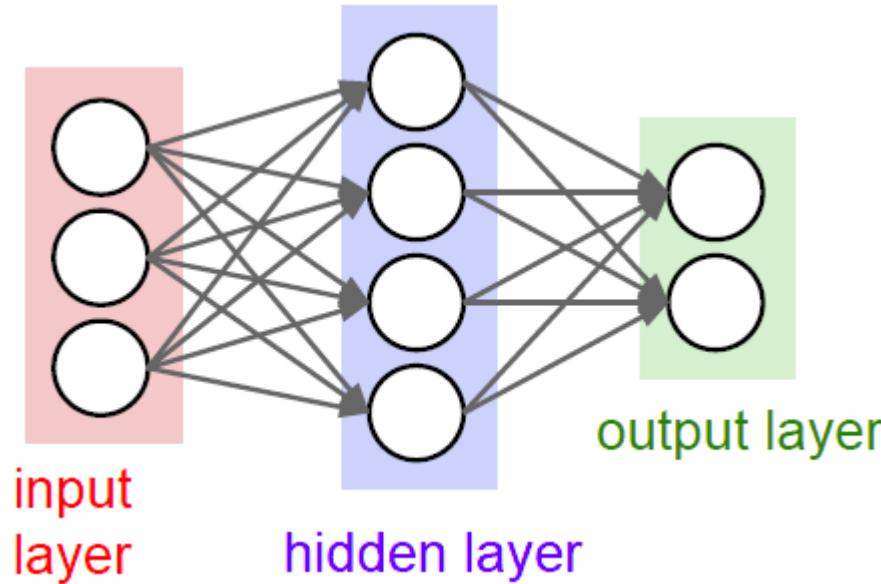
- **Convolutional net**
  - Share the same parameters across different locations
  - Convolutions with learned kernels
- Learn *multiple* filters
  - E.g.  $1000 \times 1000$  image
  - 100 filters
  - $10 \times 10$  filter size
  - $\Rightarrow$  only 10k parameters
- **Result: Response map**
  - size:  $1000 \times 1000 \times 100$
  - Only memory, not params!



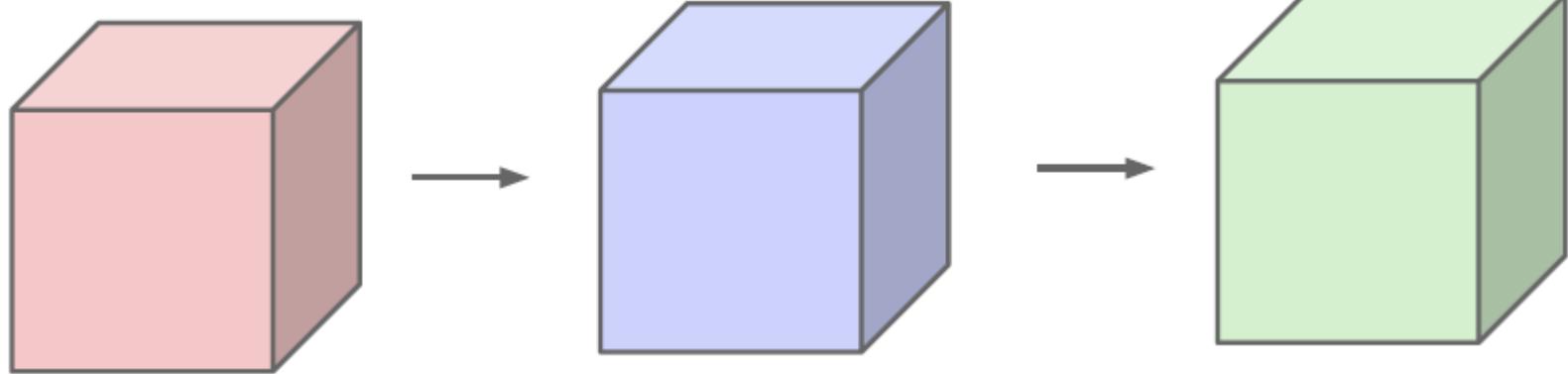
B. Leibe

# Important Conceptual Shift

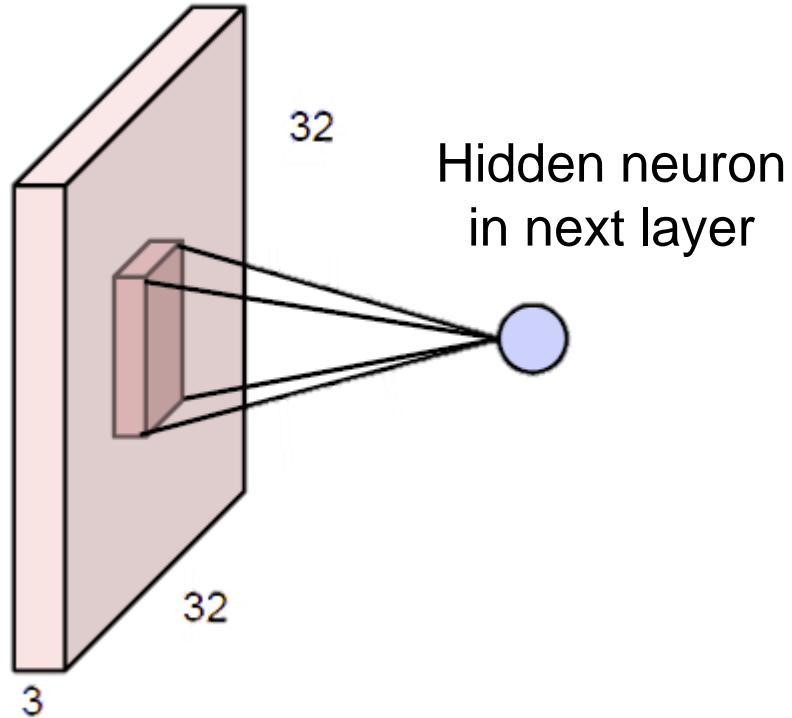
- Before



- Now:



# Convolution Layers



Example

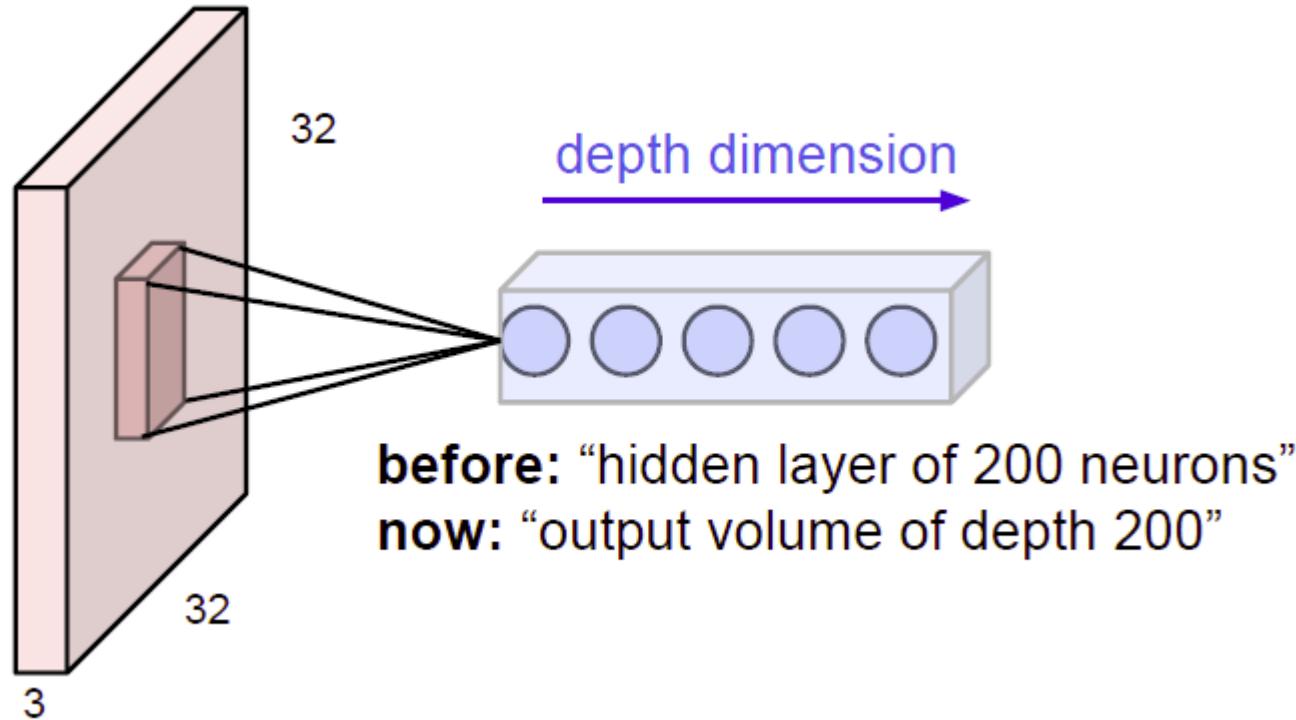
image:  $32 \times 32 \times 3$  volume

**Before:** Full connectivity  
 $32 \times 32 \times 3$  weights

**Now:** Local connectivity  
One neuron connects to, e.g.,  
 $5 \times 5 \times 3$  region.  
⇒ Only  $5 \times 5 \times 3$  **shared weights**.

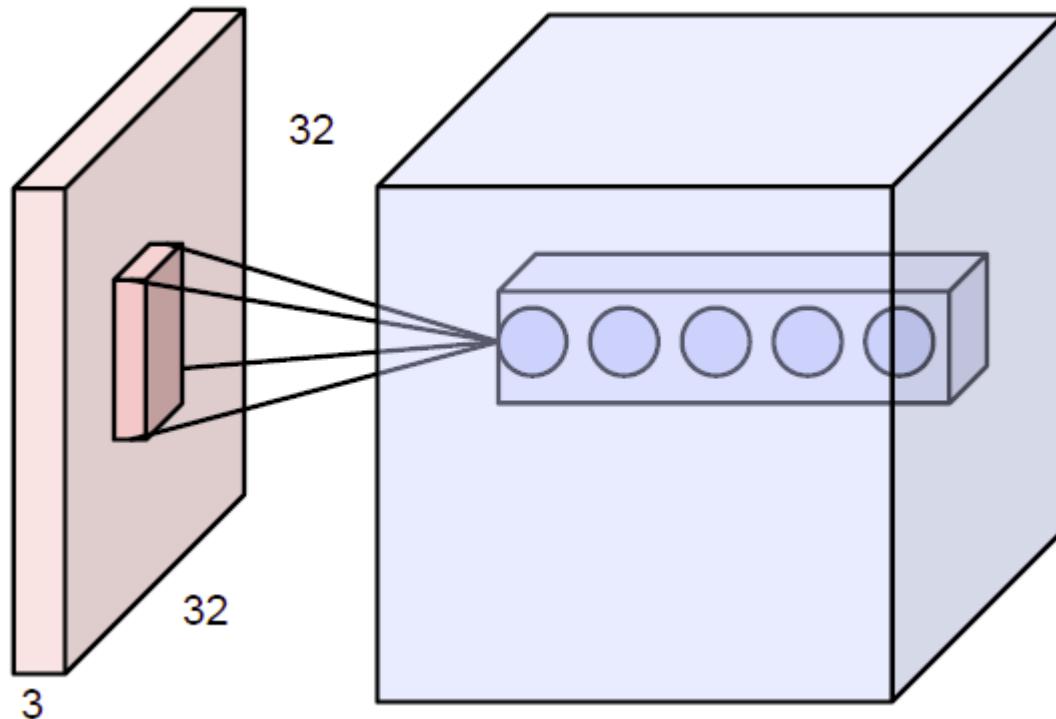
- Note: Connectivity is
  - Local in space ( $5 \times 5$  inside  $32 \times 32$ )
  - But full in depth (all 3 depth channels)

# Convolution Layers

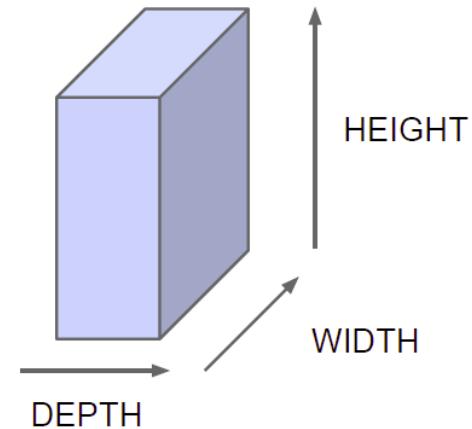


- All Neural Net activations arranged in 3 dimensions
  - Multiple neurons all looking at the same input region, stacked in depth

# Convolution Layers

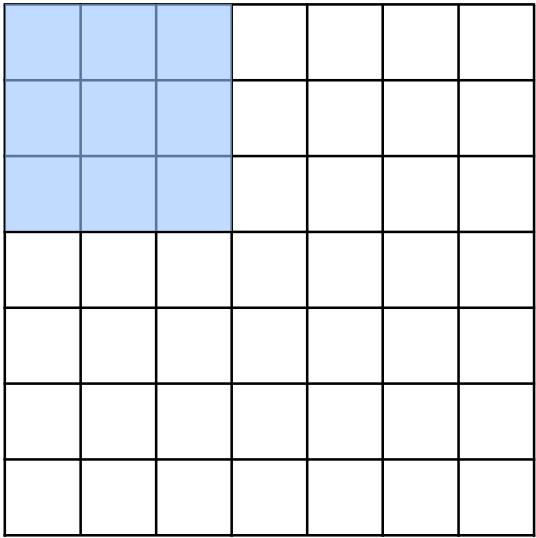


Naming convention:



- All Neural Net activations arranged in 3 dimensions
  - Multiple neurons all looking at the same input region, stacked in depth
  - Form a single  $[1 \times 1 \times \text{depth}]$  depth column in output volume.

# Convolution Layers



Example:

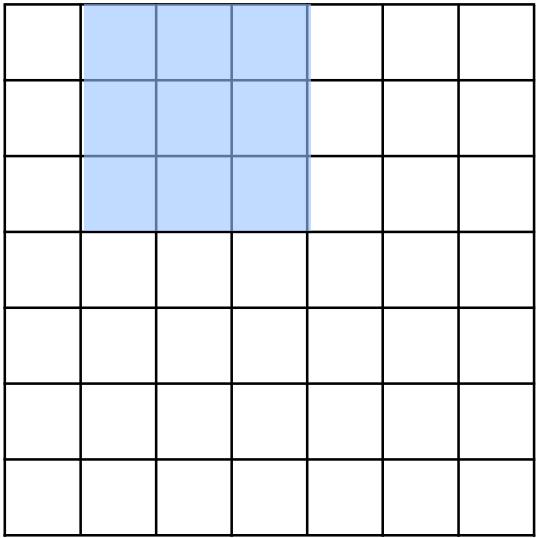
$7 \times 7$  input

assume  $3 \times 3$  connectivity

stride 1

- Replicate this column of hidden neurons across space, with some **stride**.

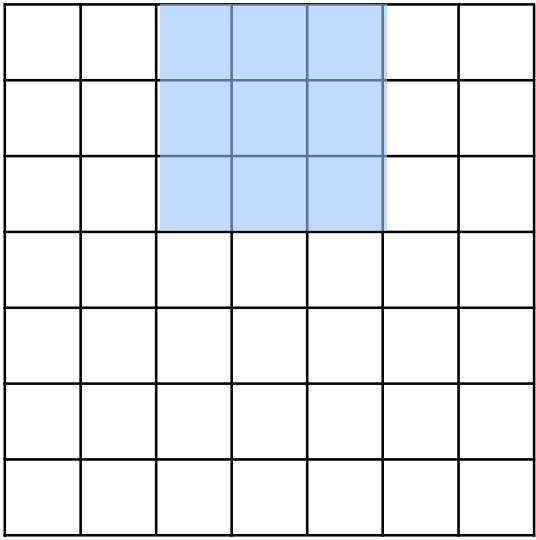
# Convolution Layers



Example:  
 $7 \times 7$  input  
assume  $3 \times 3$  connectivity  
stride 1

- Replicate this column of hidden neurons across space, with some **stride**.

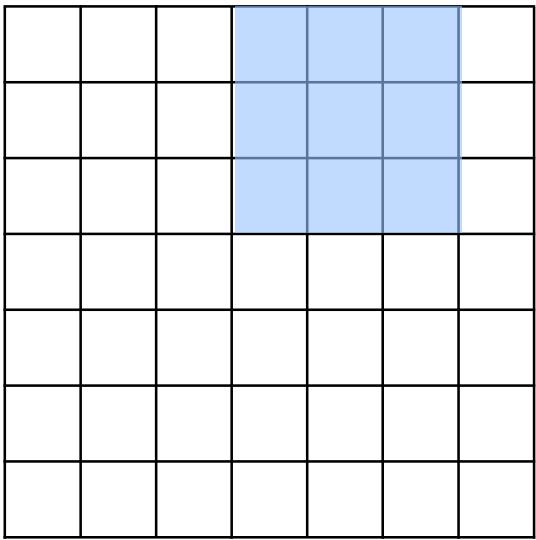
# Convolution Layers



Example:  
 $7 \times 7$  input  
assume  $3 \times 3$  connectivity  
stride 1

- Replicate this column of hidden neurons across space, with some **stride**.

# Convolution Layers



Example:

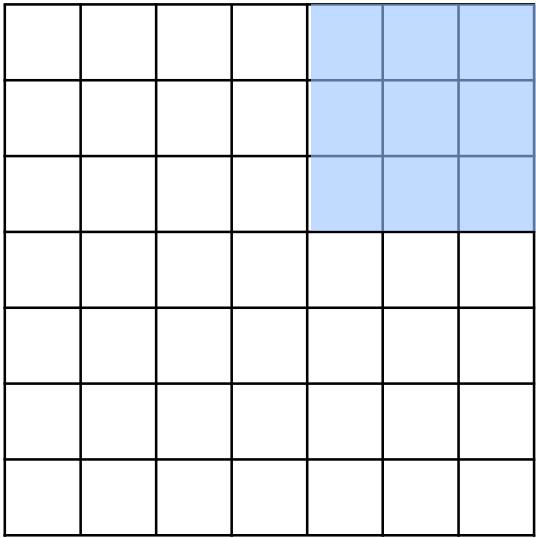
$7 \times 7$  input

assume  $3 \times 3$  connectivity

stride 1

- Replicate this column of hidden neurons across space, with some **stride**.

# Convolution Layers



Example:

$7 \times 7$  input

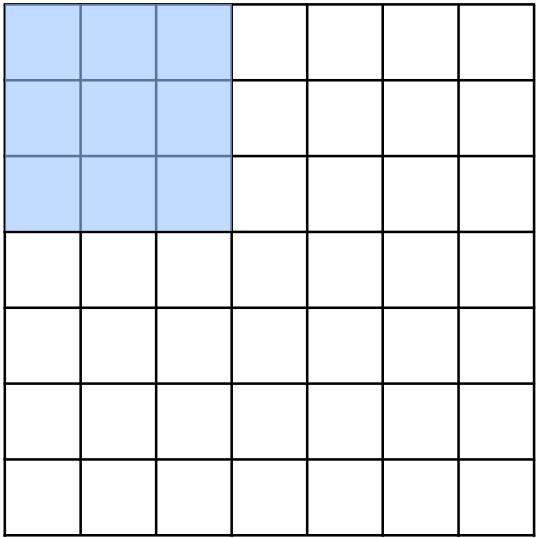
assume  $3 \times 3$  connectivity

stride 1

$\Rightarrow 5 \times 5$  output

- Replicate this column of hidden neurons across space, with some **stride**.

# Convolution Layers



Example:

$7 \times 7$  input

assume  $3 \times 3$  connectivity

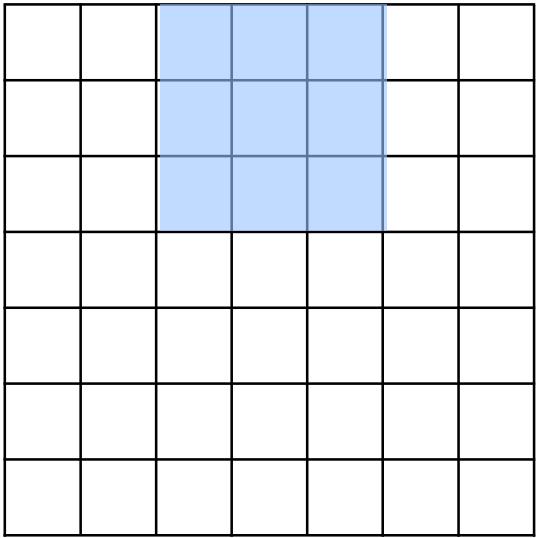
stride 1

$\Rightarrow 5 \times 5$  output

What about stride 2?

- Replicate this column of hidden neurons across space, with some **stride**.

# Convolution Layers



Example:

$7 \times 7$  input

assume  $3 \times 3$  connectivity

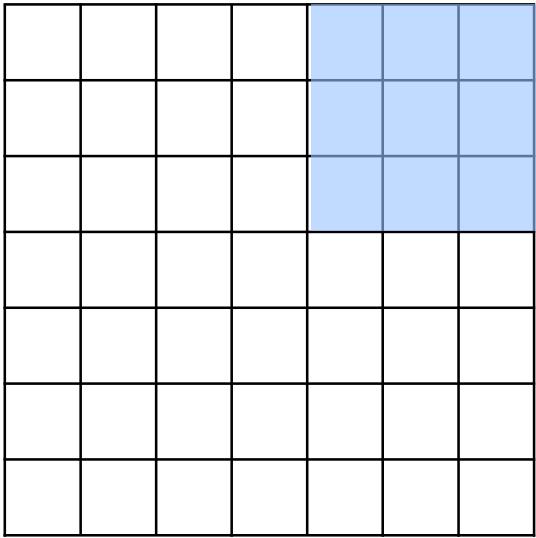
stride 1

$\Rightarrow 5 \times 5$  output

What about stride 2?

- Replicate this column of hidden neurons across space, with some **stride**.

# Convolution Layers



Example:

$7 \times 7$  input

assume  $3 \times 3$  connectivity

stride 1

$\Rightarrow 5 \times 5$  output

What about stride 2?

$\Rightarrow 3 \times 3$  output

- Replicate this column of hidden neurons across space, with some **stride**.

# Convolution Layers

0	0	0	0	0				
0								
0								
0								
0								

Example:

$7 \times 7$  input

assume  $3 \times 3$  connectivity

stride 1

$\Rightarrow 5 \times 5$  output

What about stride 2?

$\Rightarrow 3 \times 3$  output

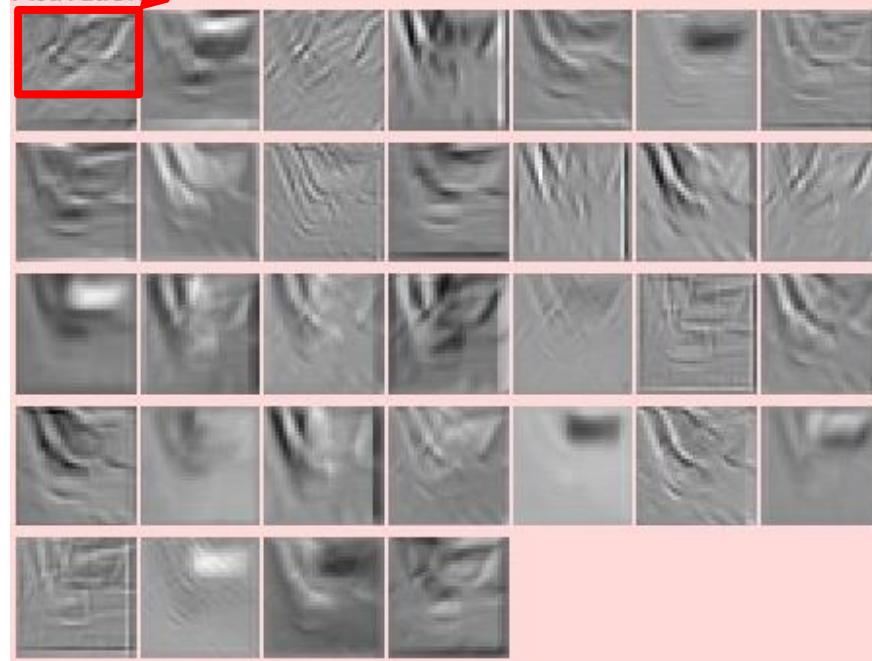
- Replicate this column of hidden neurons across space, with some **stride**.
- In practice, common to zero-pad the border.
  - Preserves the size of the input spatially.

# Activation Maps of Convolutional Filters

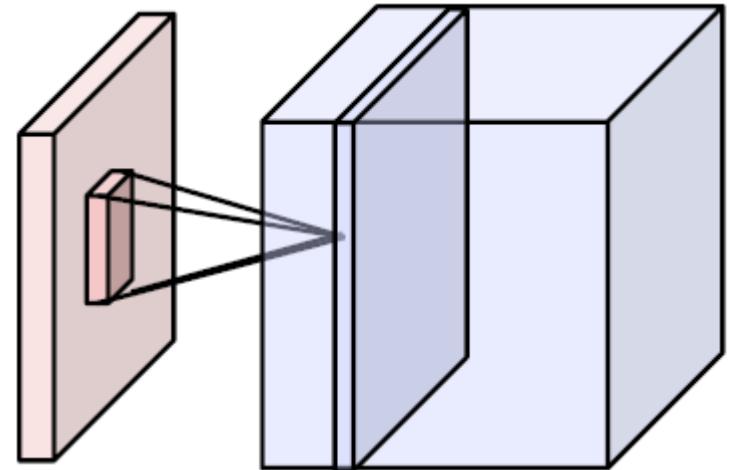
Activations:



Activation

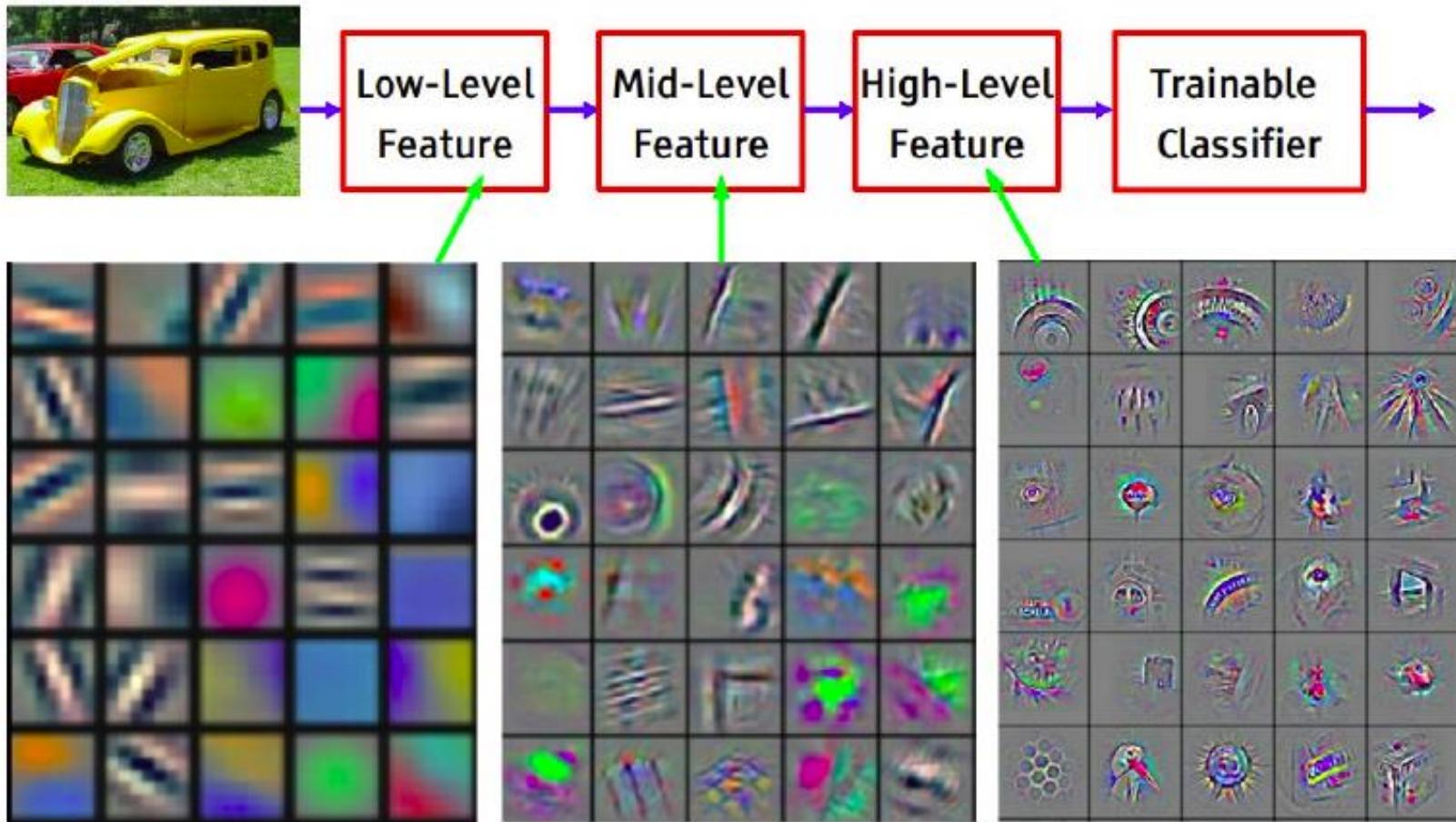


Activation maps



Each activation map is a depth slice through the output volume.

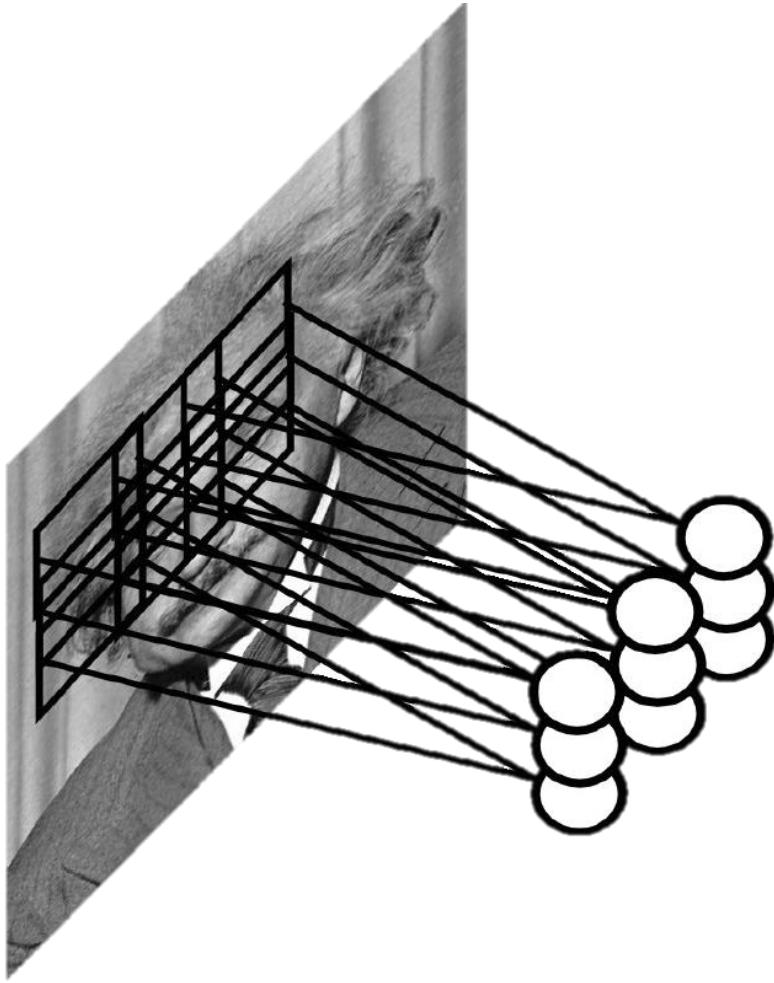
# Effect of Multiple Convolution Layers



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

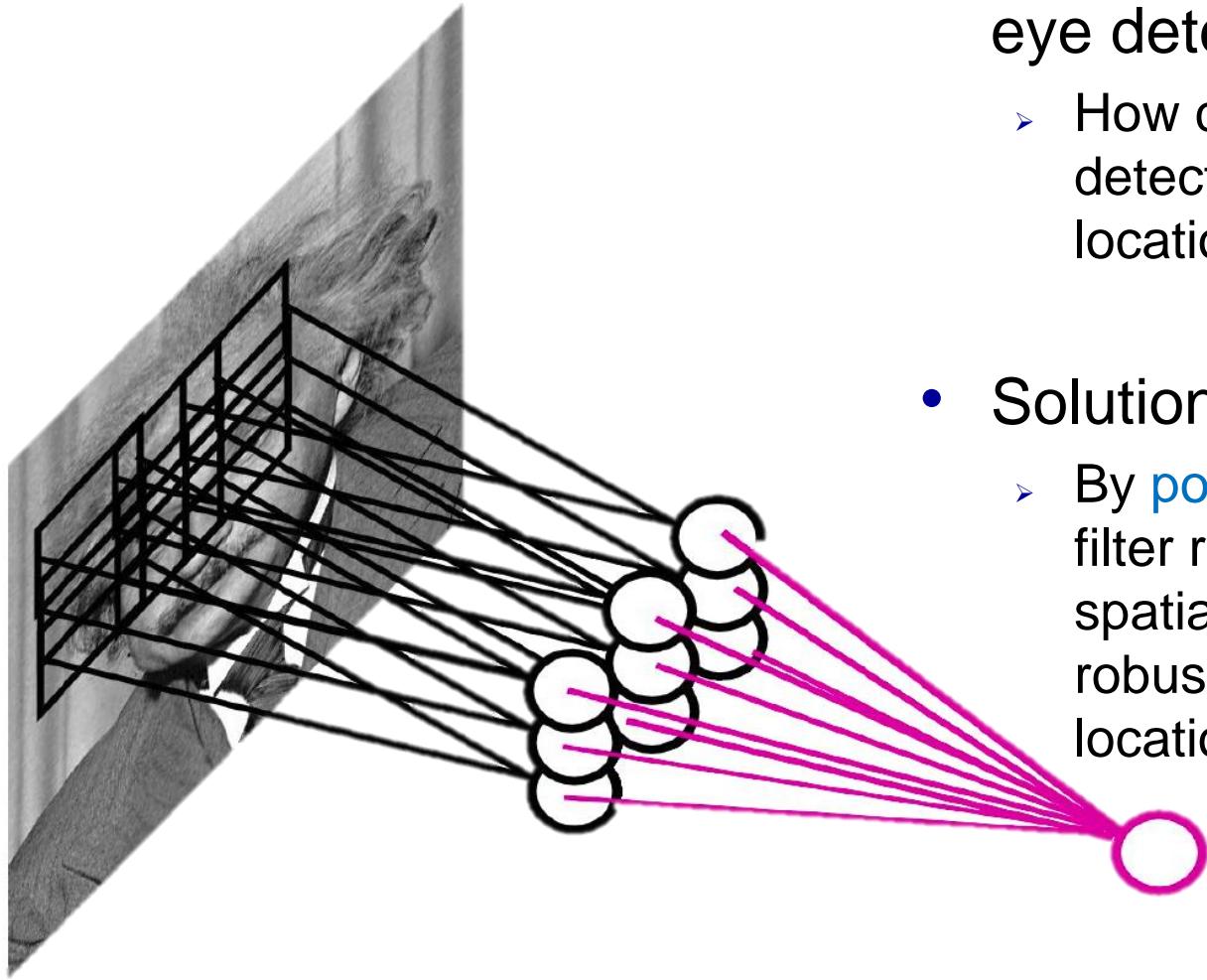
# Convolutional Networks: Intuition

- Let's assume the filter is an eye detector
  - How can we make the detection robust to the exact location of the eye?

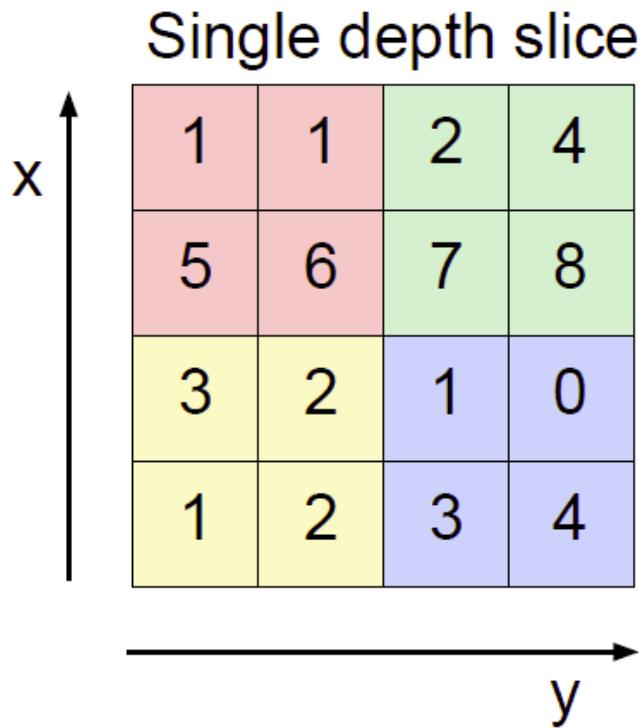


# Convolutional Networks: Intuition

- Let's assume the filter is an eye detector
  - How can we make the detection robust to the exact location of the eye?
- Solution:
  - By **pooling** (e.g., max or avg) filter responses at different spatial locations, we gain robustness to the exact spatial location of features.



# Max Pooling

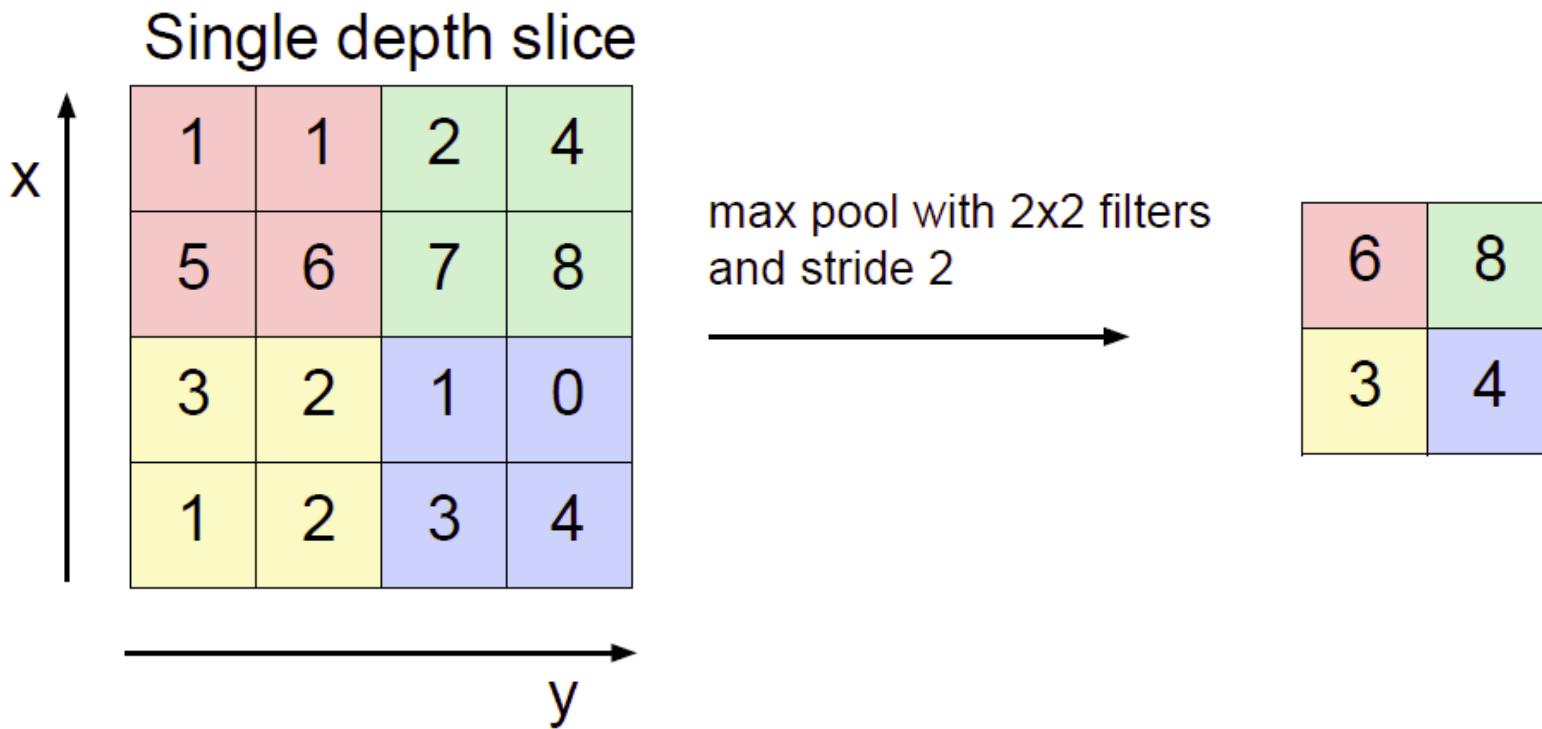


max pool with 2x2 filters  
and stride 2

6	8
3	4

- Effect:
  - Make the representation smaller without losing too much information
  - Achieve robustness to translations

# Max Pooling



- Note
  - Pooling happens independently across each slice, preserving the number of slices.

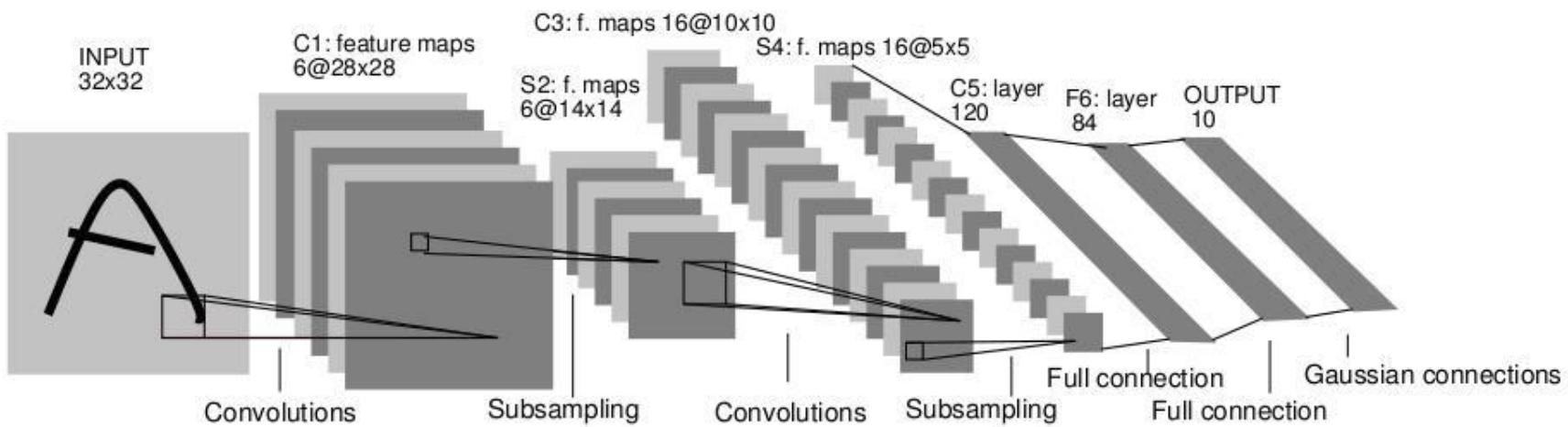
# CNNs: Implication for Back-Propagation

- Convolutional layers
  - Filter weights are shared between locations  
⇒ Gradients are added for each filter location.

# Topics of This Lecture

- Recap: CNNs
- CNN Architectures
  - LeNet
  - AlexNet
  - VGGNet
  - GoogLeNet
- Visualizing CNNs
  - Visualizing CNN features
  - Visualizing responses
  - Visualizing learned structures
- Applications

# CNN Architectures: LeNet (1998)



- Early convolutional architecture
  - 2 Convolutional layers, 2 pooling layers
  - Fully-connected NN layers for classification
  - Successfully used for handwritten digit recognition (MNIST)

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.

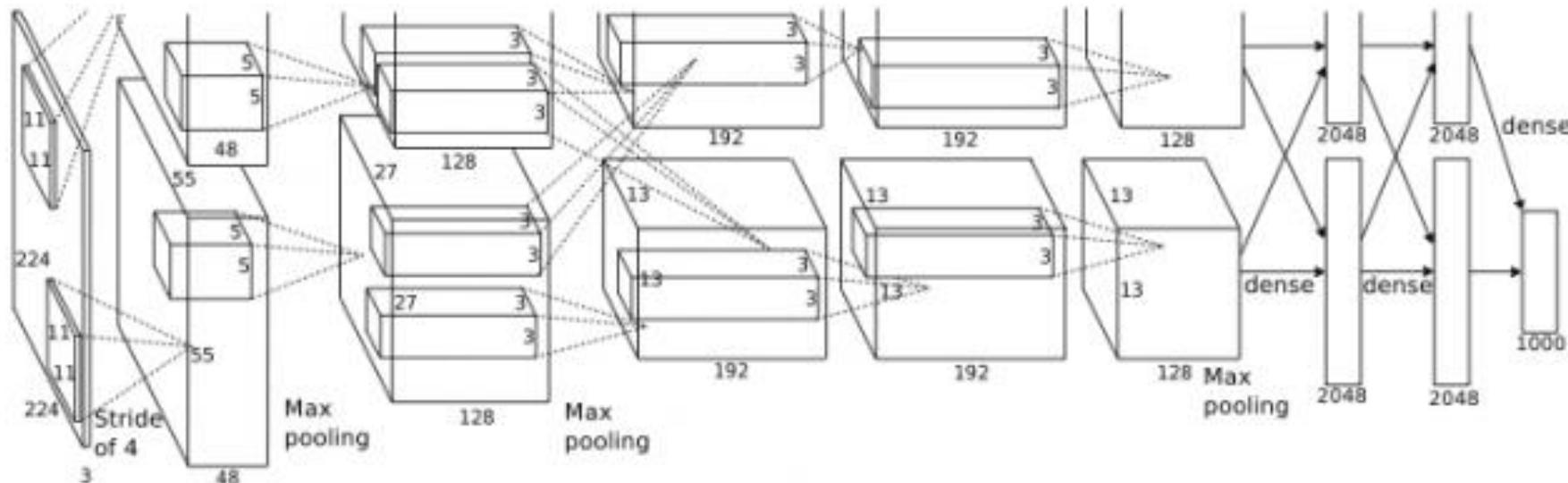
# ImageNet Challenge 2012

- ImageNet
  - ~14M labeled internet images
  - 20k classes
  - Human labels via Amazon Mechanical Turk
- Challenge (ILSVRC)
  - 1.2 million training images
  - 1000 classes
  - Goal: Predict ground-truth class within top-5 responses
  - Currently one of the top benchmarks in Computer Vision



[Deng et al., CVPR'09]

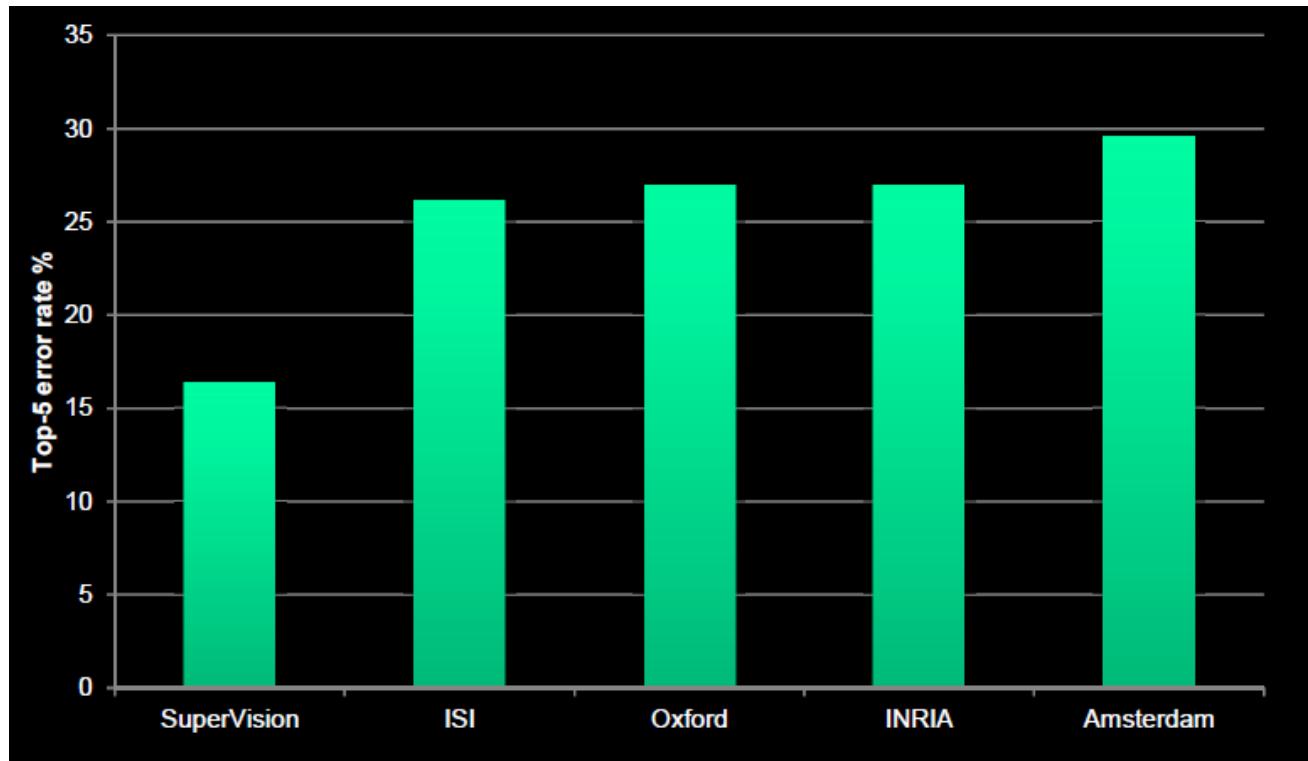
# CNN Architectures: AlexNet (2012)



- 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)

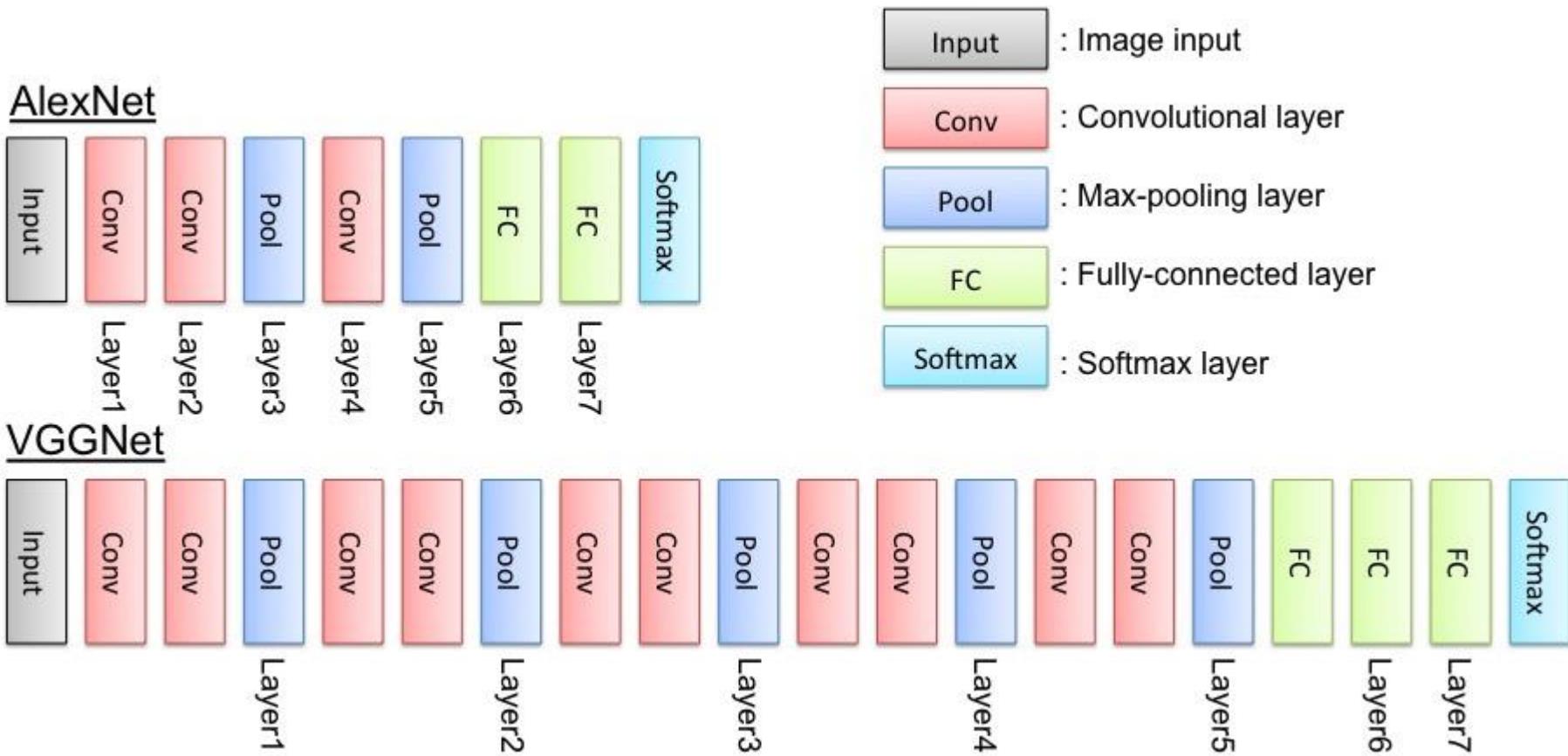
A. Krizhevsky, I. Sutskever, and G. Hinton, [ImageNet Classification with Deep Convolutional Neural Networks](#), NIPS 2012.

# ILSVRC 2012 Results



- AlexNet almost halved the error rate
  - 16.4% error (top-5) vs. 26.2% for the next best approach
  - ⇒ A revolution in Computer Vision
  - Acquired by Google in Jan '13, deployed in Google+ in May '13

# CNN Architectures: VGGNet (2014/15)



K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015

# CNN Architectures: 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%.

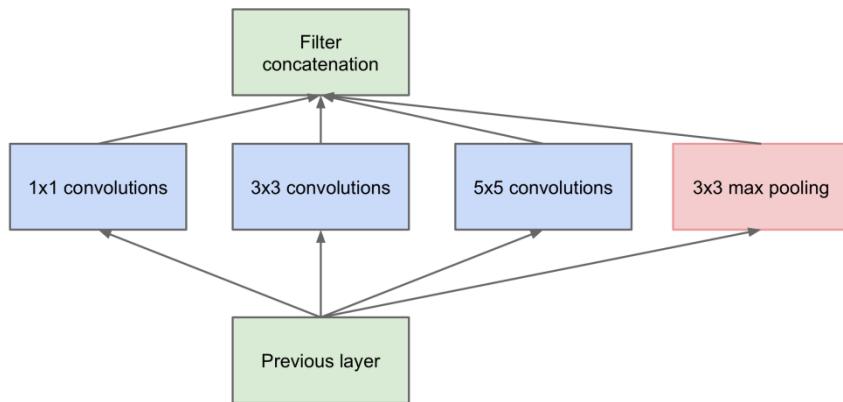
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input ( $224 \times 224$ RGB image)					
conv3-64	conv3-64 LRN	conv3-64 <b>conv3-64</b>	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	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 <b>conv1-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 conv3-256 <b>conv3-256</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Mainly used

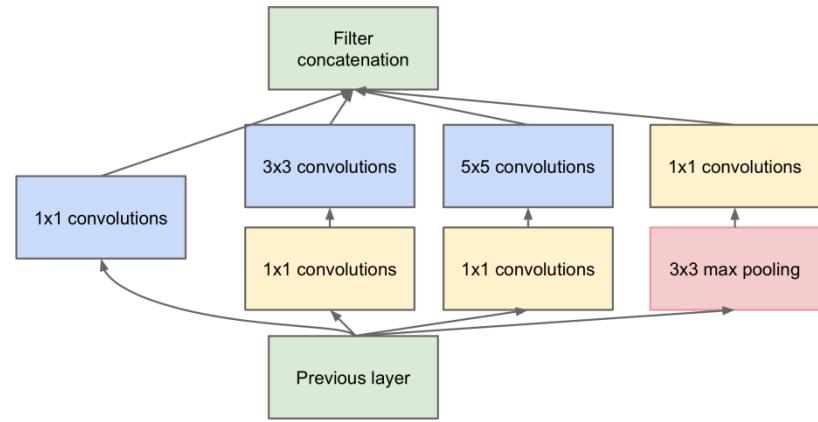
# Comparison: AlexNet vs. VGGNet

- Receptive fields in the first layer
  - AlexNet:  $11 \times 11$ , stride 4
  - Zeiler & Fergus:  $7 \times 7$ , stride 2
  - VGGNet:  $3 \times 3$ , stride 1
- Why that?
  - If you stack a  $3 \times 3$  on top of another  $3 \times 3$  layer, you effectively get a  $5 \times 5$  receptive field.
  - With three  $3 \times 3$  layers, the receptive field is already  $7 \times 7$ .
  - But much fewer parameters:  $3 \cdot 3^2 = 27$  instead of  $7^2 = 49$ .
  - In addition, non-linearities in-between  $3 \times 3$  layers for additional discriminativity.

# CNN Architectures: GoogLeNet (2014/2015)



(a) Inception module, naïve version

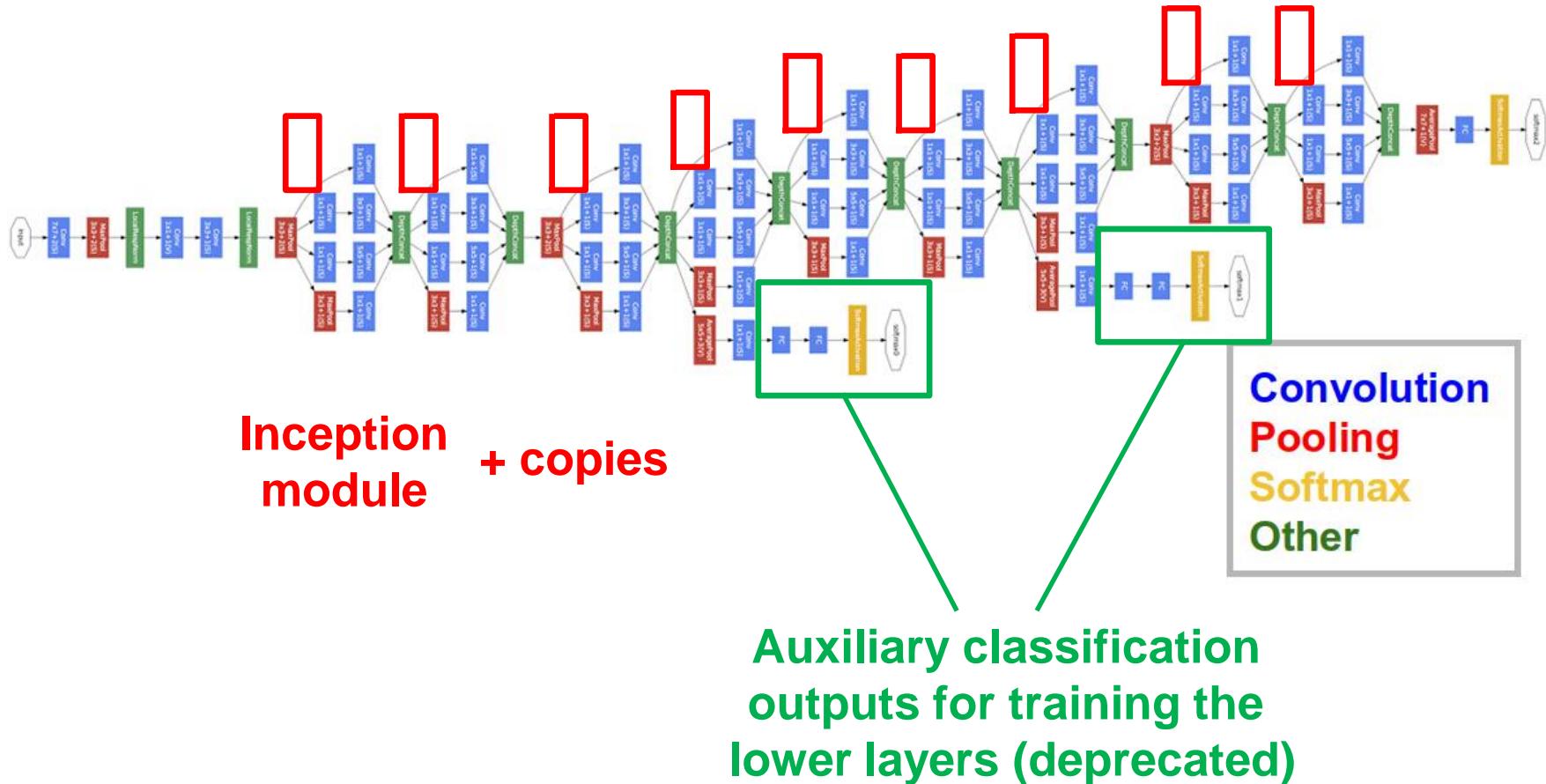


(b) Inception module with dimension reductions

- Main ideas
  - “Inception” module as modular component
  - Learns filters at several scales within each module

C. Szegedy, W. Liu, Y. Jia, et al, [Going Deeper with Convolutions](#),  
arXiv:1409.4842, 2014, CVPR’15, 2015.

# GoogLeNet Visualization



# Results on ILSVRC

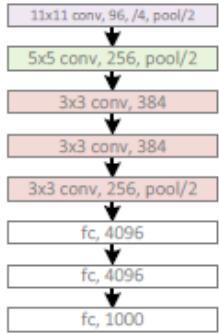
Method	top-1 val. error (%)	top-5 val. error (%)	top-5 test error (%)
VGG (2 nets, multi-crop & dense eval.)	<b>23.7</b>	<b>6.8</b>	<b>6.8</b>
VGG (1 net, multi-crop & dense eval.)	24.4	7.1	7.0
VGG (ILSVRC submission, 7 nets, dense eval.)	24.7	7.5	7.3
GoogLeNet (Szegedy et al., 2014) (1 net)	-		7.9
GoogLeNet (Szegedy et al., 2014) (7 nets)	-		<b>6.7</b>
MSRA (He et al., 2014) (11 nets)	-	-	8.1
MSRA (He et al., 2014) (1 net)	27.9	9.1	9.1
Clarifai (Russakovsky et al., 2014) (multiple nets)	-	-	11.7
Clarifai (Russakovsky et al., 2014) (1 net)	-	-	12.5
Zeiler & Fergus (Zeiler & Fergus, 2013) (6 nets)	36.0	14.7	14.8
Zeiler & Fergus (Zeiler & Fergus, 2013) (1 net)	37.5	16.0	16.1
OverFeat (Sermanet et al., 2014) (7 nets)	34.0	13.2	13.6
OverFeat (Sermanet et al., 2014) (1 net)	35.7	14.2	-
Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)	38.1	16.4	16.4
Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)	40.7	18.2	-

- VGGNet and GoogLeNet perform at similar level
  - Comparison: human performance ~5% [Karpathy]

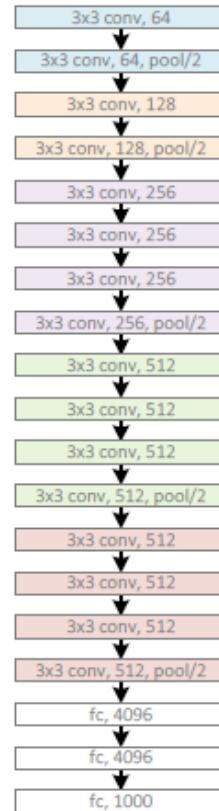
<http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/>

# Newer Developments: Residual Networks

AlexNet, 8 layers  
(ILSVRC 2012)



VGG, 19 layers  
(ILSVRC 2014)



GoogleNet, 22 layers  
(ILSVRC 2014)



# Newer Developments: Residual Networks

AlexNet, 8 layers  
(ILSVRC 2012)

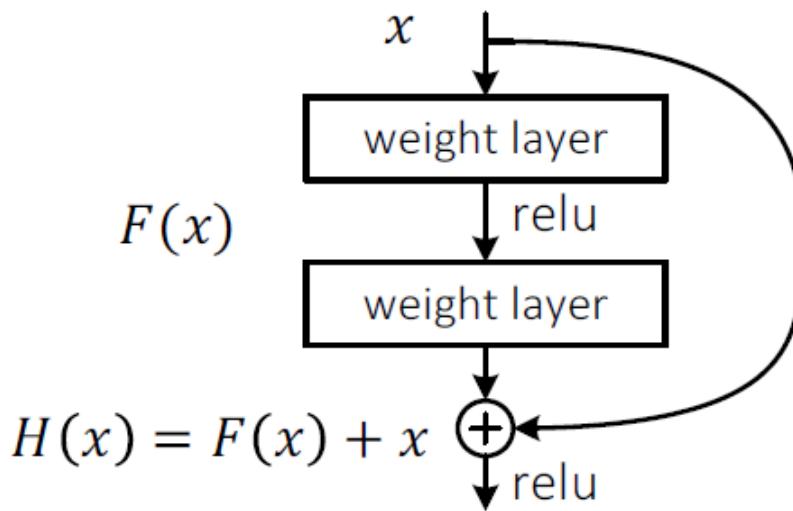


VGG, 19 layers  
(ILSVRC 2014)

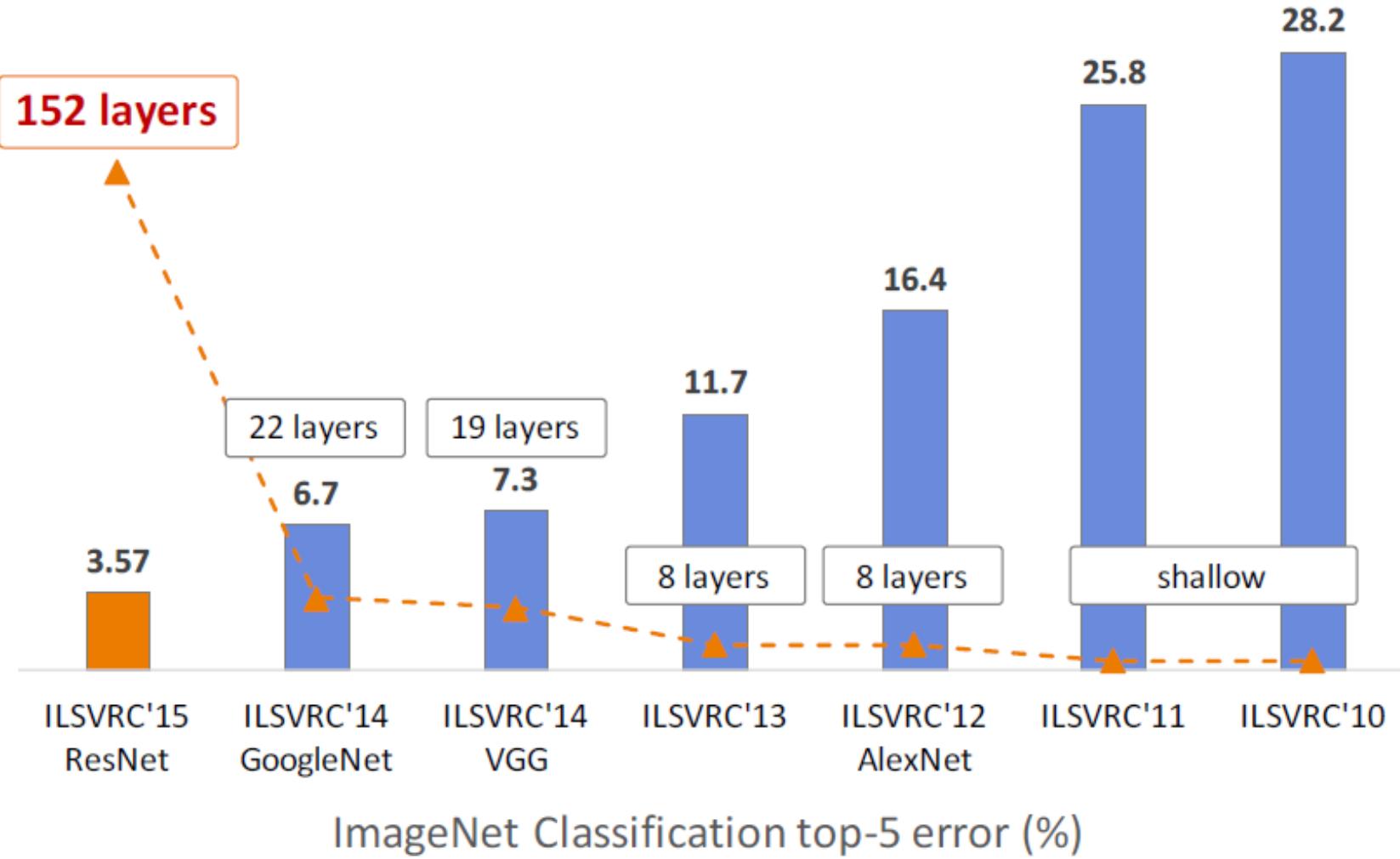


ResNet, 152 layers  
(ILSVRC 2015)

- Core component
  - Skip connections bypassing each layer
  - Better propagation of gradients to the deeper layers
  - We'll analyze this mechanism in more detail later...



# ImageNet Performance



# Understanding the ILSVRC Challenge

- Imagine the scope of the problem!
  - 1000 categories
  - 1.2M training images
  - 50k validation images
- This means...
  - Speaking out the list of category names at 1 word/s...  
...takes 15mins.
  - Watching a slideshow of the validation images at 2s/image...  
...takes a full day (24h+).
  - Watching a slideshow of the training images at 2s/image...  
...takes a full month.



rier, Airedale, airliner, airship, albatross, alligator lizard, alp, altar, ambulance, American alligator, American black bear, American chameleon, American coot, American egret, American lobster, American Staffordshire terrier, amphibian, analog clock, anemone fish, Angora, ant, apiary, Appenzeller, apron, Arabian camel, Arctic fox, armadillo, artichoke, ashcan, assault rifle, Australian terrier, axolotl, baboon, backpack, badger, bagel, bakery, balance beam, bald eagle, balloon, ballplayer, ballpoint, banana, Band Aid, banded gecko, banjo, bannister, barbell, barber chair, barbershop, barn, barn spider, barometer, barracouta, barrel, barrow, baseball, basenji, basketball, basset, bassinet, bassoon, bath towel, bathing cap, bathtub, beach wagon, beacon, beagle, beaker, bearskin, beaver, Bedlington terrier, bee, bee eater, beer bottle, beer glass, bell cote, bell pepper, Bernese mountain dog, bib, bicycle-built-for-two, bighorn, bikini, binder, binoculars, birdhouse, bison, bittern, black and gold garden spider, black grouse, black stork, black swan, black widow, black-and-tan coonhound, black-footed ferret, Blenheim spaniel, bloodhound, bluetick, boa constrictor, boathouse, bobsled, bolete, bolo tie, bonnet, book jacket, bookcase, bookshop, Border collie, Border terrier, borzoi, Boston bull, bottlecap, Bouvier des Flandres, bow, bow tie, box turtle, boxer, Brabancon griffon, brain coral, brambling, brass, brassiere, breakwater, breastplate, briard, Brittany spaniel, broccoli, broom, brown bear, bubble, bucket, buckeye, buckle, bulbul, bull mastiff, bullet train, bulletproof vest, bullfrog, burrito, bustard, butcher shop, butternut squash, cab, cabbage butterfly, cairn, caldron, can opener, candle, cannon, canoe, capuchin, car mirror, car wheel, carbonara, Cardigan, cardigan, cardoon, carousel, carpenter's kit, carton, cash machine, cassette, cassette player, castle, catamaran, cauliflower, CD player, cello, cellular telephone, centipede, chain, chain mail, chain saw, chain-link fence, chambered nautilus, cheeseburger, cheetah, Chesapeake Bay retriever, chest, chickadee, chiffonier, Chihuahua, chime, chimpanzee, china cabinet, chiton, chocolate sauce, chow, Christmas stocking, church, cicada, cinema, cleaver, cliff, cliff dwelling, cloak, clog, clumber, cock, cocker spaniel, cockroach, cocktail shaker, coffee mug, coffeepot, coho, coil, collie, colobus, combination lock, comic book, common iguana, common newt, computer keyboard, conch, confectionery, consomme, container ship, convertible, coral fungus, coral reef, corkscrew, corn, cornet, coucal, cougar, cowboy boot, cowboy hat, coyote, cradle, crane, crane, crash helmet, crate, crayfish, crib, cricket, Crock Pot, croquet ball, crossword puzzle, crutch, cucumber, cuirass, cup, curly-coated retriever, custard apple, daisy, dalmatian, dam, damselfly, Dandie Dinmont, desk, desktop computer, dhole, dial telephone, diamondback, diaper, digital clock, digital watch, dingo, dining table, dishrag, dishwasher, disk brake, Doberman, dock, dogsled, dome, doormat, dough, dowitcher, dragonfly, drake, drilling platform, drum, drumstick, dugong, dumbbell, dung beetle, Dungeness crab, Dutch oven, ear, earthstar, echidna, eel, eft, eggnog, Egyptian cat, electric fan, electric guitar, electric locomotive, electric ray, English foxhound, English setter, English springer, entertainment center, EntleBucher, envelope, Eskimo dog, espresso, espresso maker, European fire salamander, European gallinule, face powder, feather boa, fiddler crab, fig, file, fire engine, fire screen, fireboat, flagpole, flamingo, flat-coated retriever, flatworm, flute, fly, folding chair, football helmet, forklift, fountain, fountain pen, four-poster, fox squirrel, freight car, French bulldog, French horn, French loaf, frilled lizard, frying pan, fur coat, gar, garbage truck, garden spider, garter snake, gas pump, gasmask, gazelle, German shepherd, German short-haired pointer, geyser, giant panda, giant schnauzer, gibbon, Gila monster, go-kart, goblet, golden retriever, goldfinch, goldfish, golf ball, golfcart, gondola, gong, goose, Gordon setter, gorilla, gown, grand piano, Granny Smith, grasshopper, Great Dane, great grey owl, Great Pyrenees, great white shark,

# More Finegraded Classes

birds



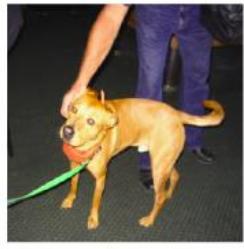
bird

cats



cat

dogs



dog

PASCAL



flamingo



cock



ruffed grouse



quail



partridge

...

ILSVRC



Egyptian cat



Persian cat



Siamese cat

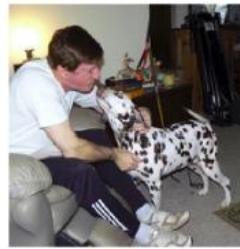


tabby



lynx

...



dalmatian



keeshond



miniature schnauzer



standard schnauzer



giant schnauzer

...

# Quirks and Limitations of the Data Set



- Generated from WordNet ontology
    - Some animal categories are overrepresented
    - E.g., 120 subcategories of dog breeds
- ⇒ 6.7% top-5 error looks all the more impressive

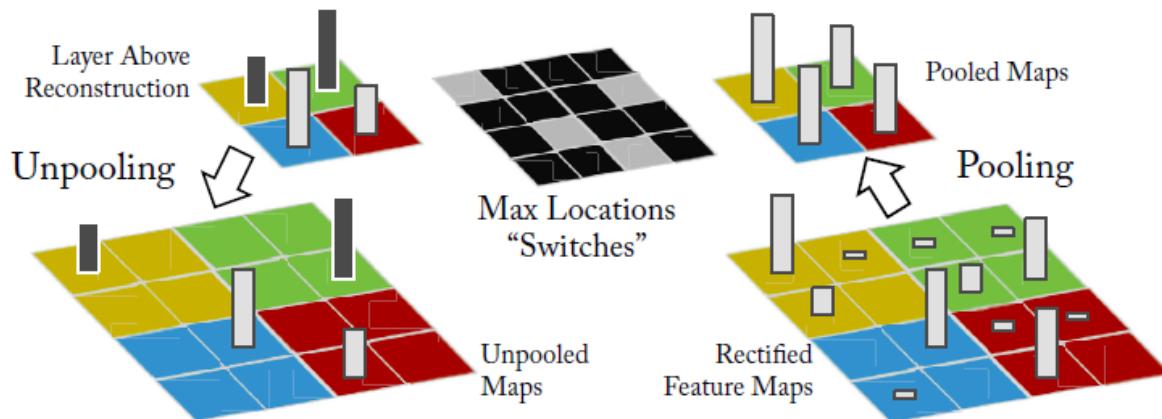
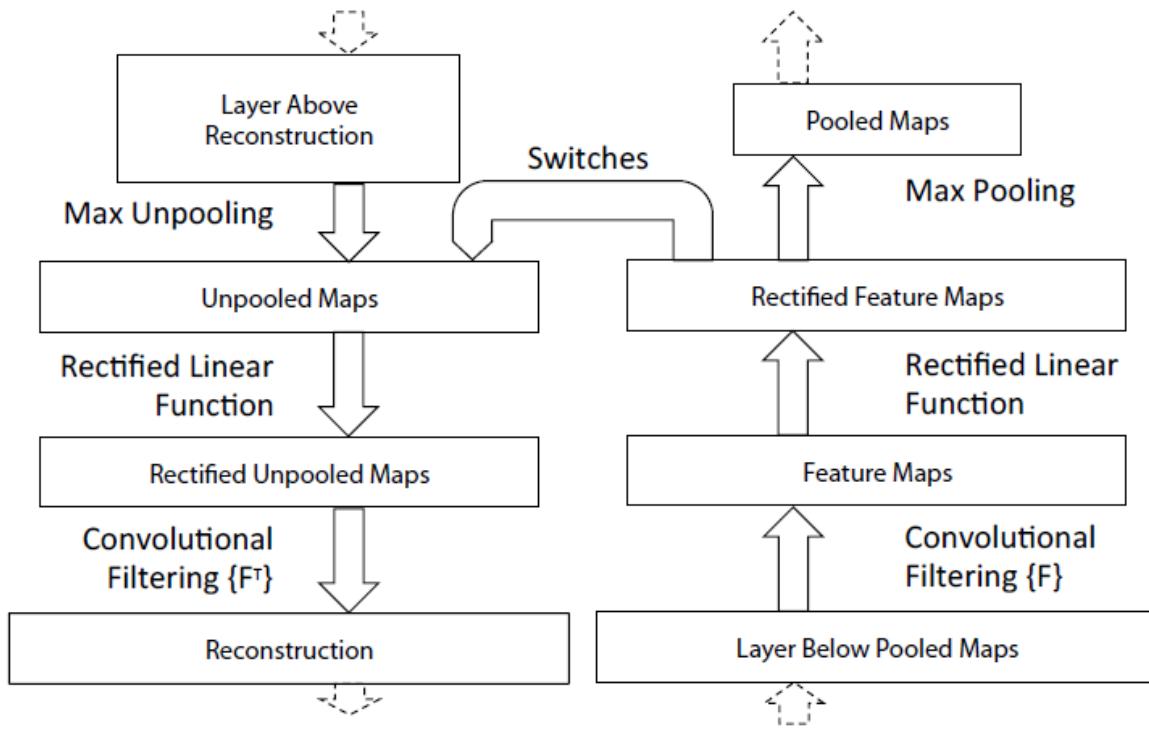
# Topics of This Lecture

- Recap: CNNs
- CNN Architectures
  - LeNet
  - AlexNet
  - VGGNet
  - GoogLeNet
- Visualizing CNNs
  - Visualizing CNN features
  - Visualizing responses
  - Visualizing learned structures
- Applications

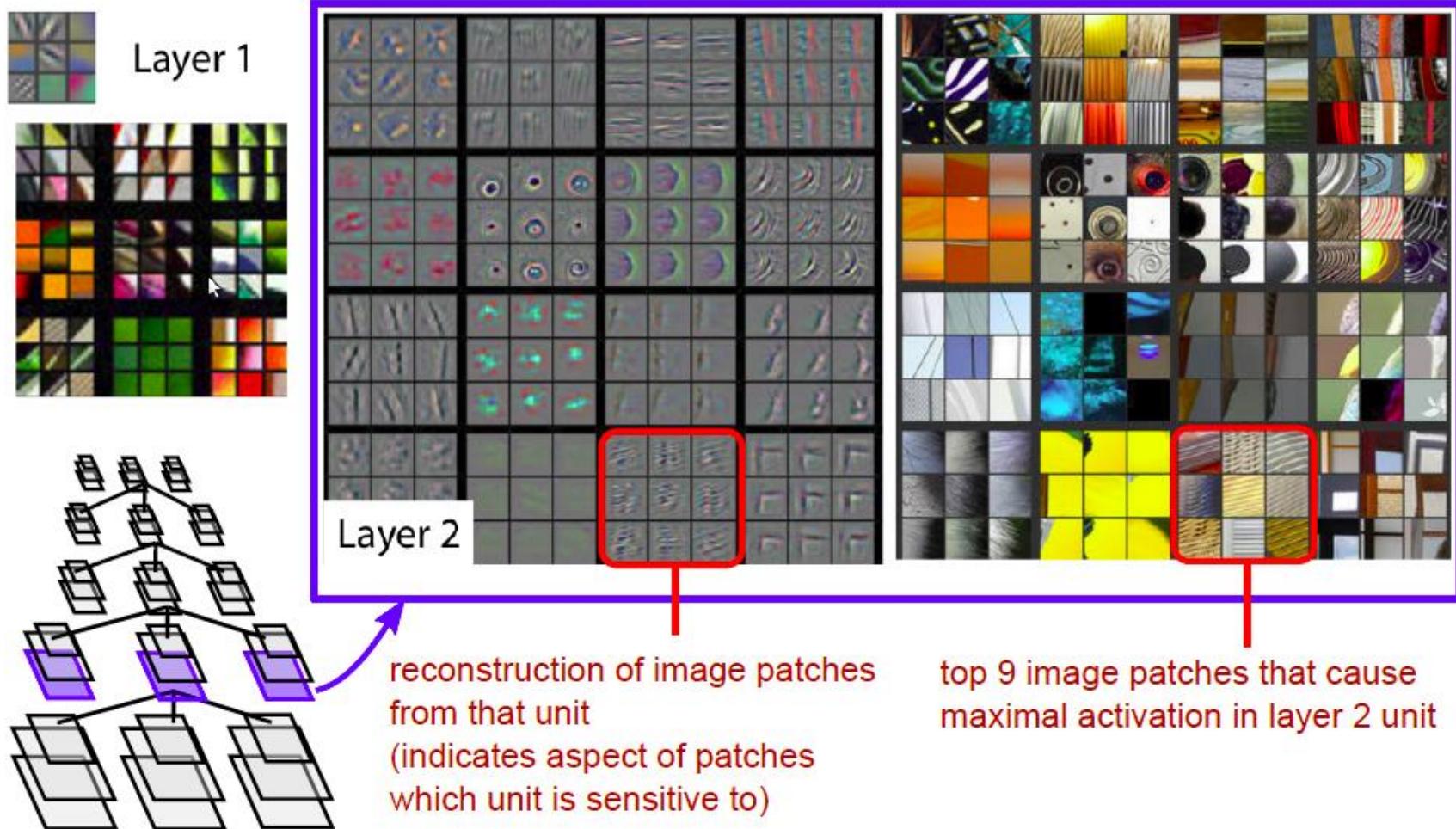
# Visualizing CNNs

DeconvNet

ConvNet

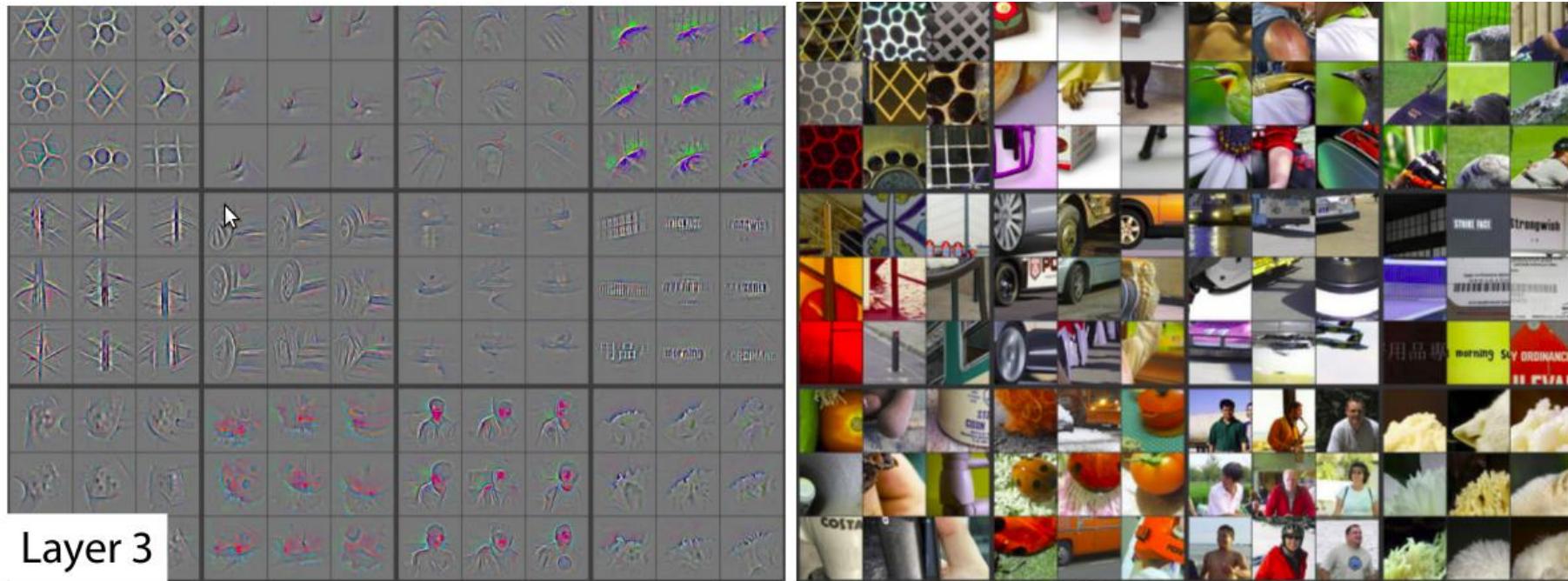


# Visualizing CNNs

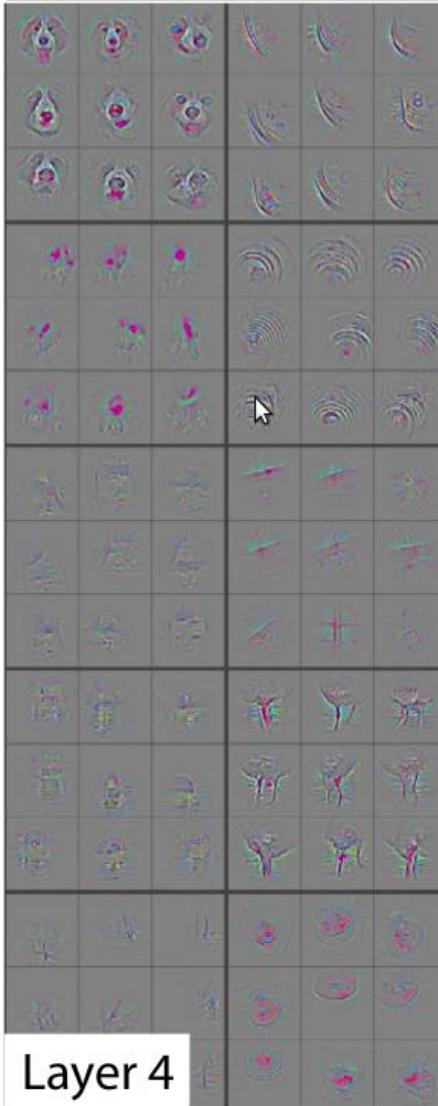


M. Zeiler, R. Fergus, [Visualizing and Understanding Convolutional Neural Networks](#),  
ECCV 2014.

# Visualizing CNNs



# Visualizing CNNs



Layer 4



Layer 5

# What Does the Network React To?

- Occlusion Experiment
  - Mask part of the image with an occluding square.
  - Monitor the output

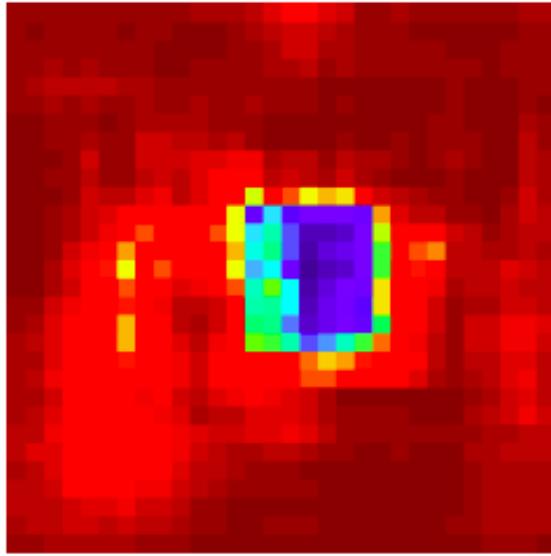


# What Does the Network React To?

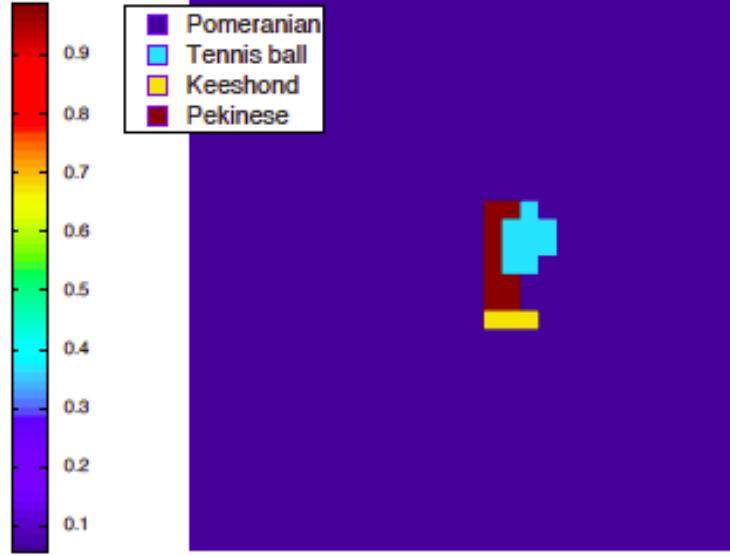
Input image



$p(\text{True class})$



Most probable class

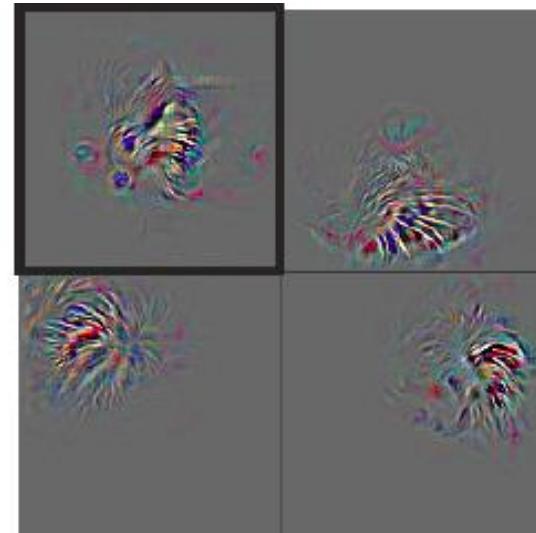
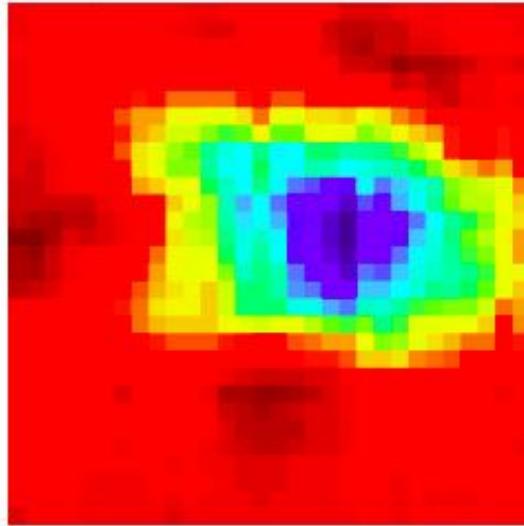


# What Does the Network React To?

Input image



Total activation in most active 5<sup>th</sup> layer feature map



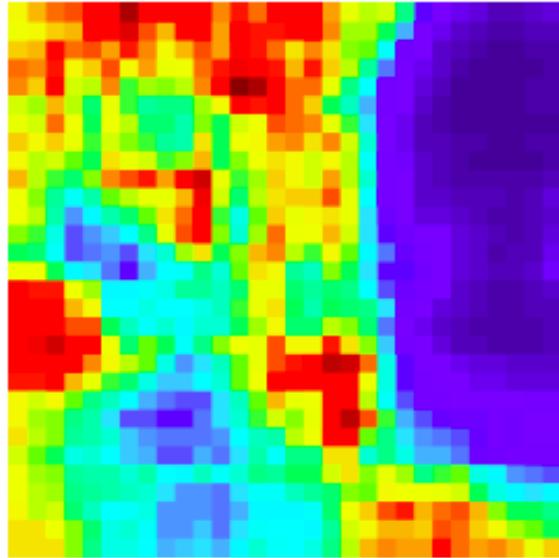
Other activations from the same feature map.

# What Does the Network React To?

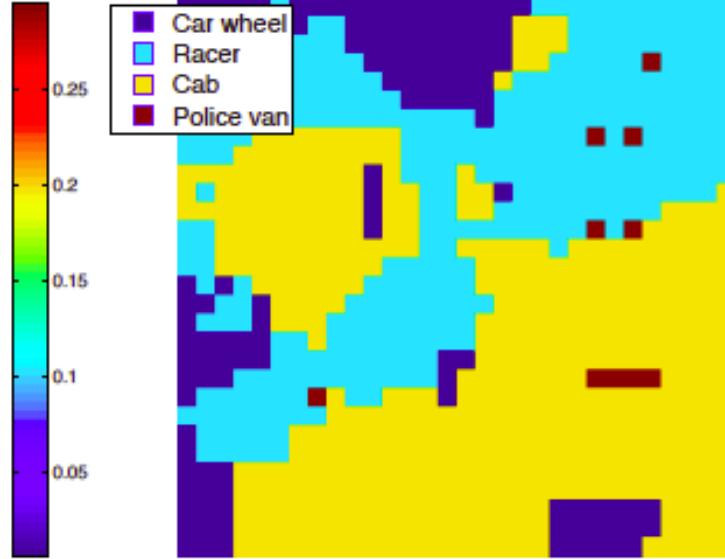
Input image



$p(\text{True class})$



Most probable class

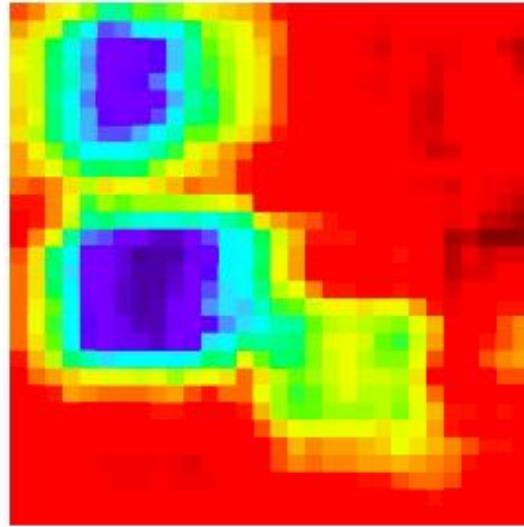


# What Does the Network React To?

Input image



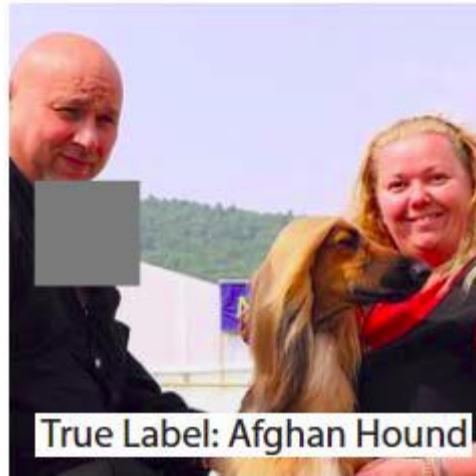
Total activation in most active 5<sup>th</sup> layer feature map



Other activations from the same feature map.

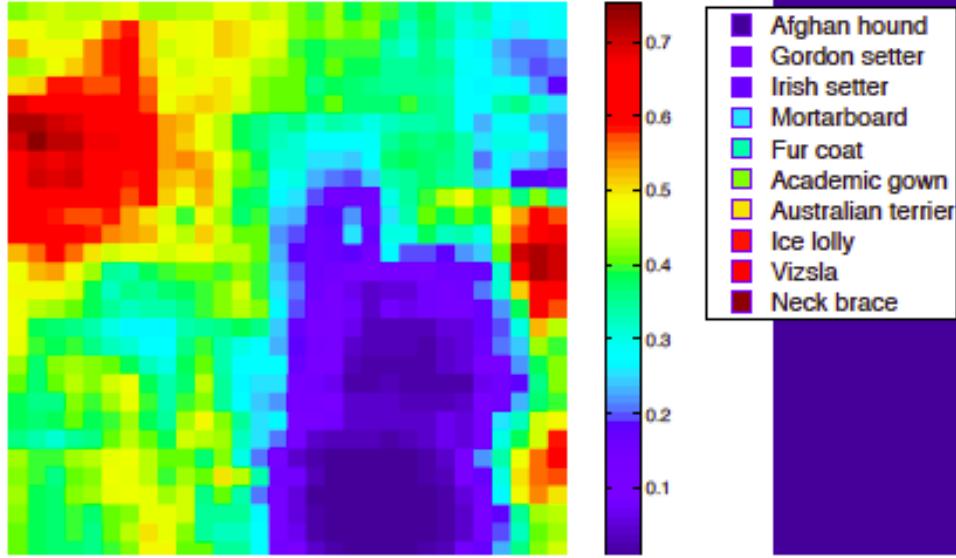
# What Does the Network React To?

Input image

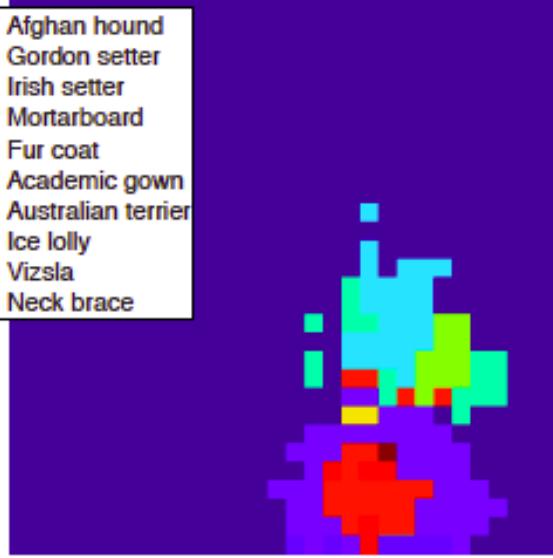


True Label: Afghan Hound

$p(\text{True class})$

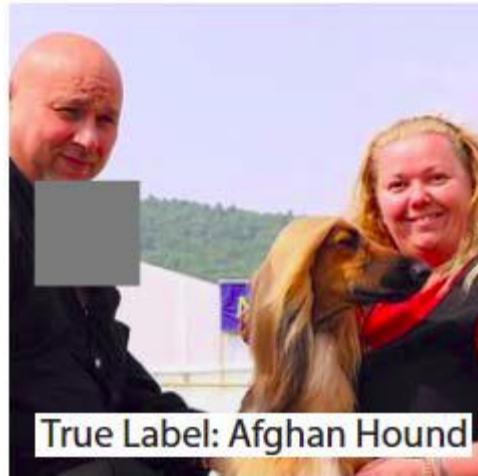


Most probable class

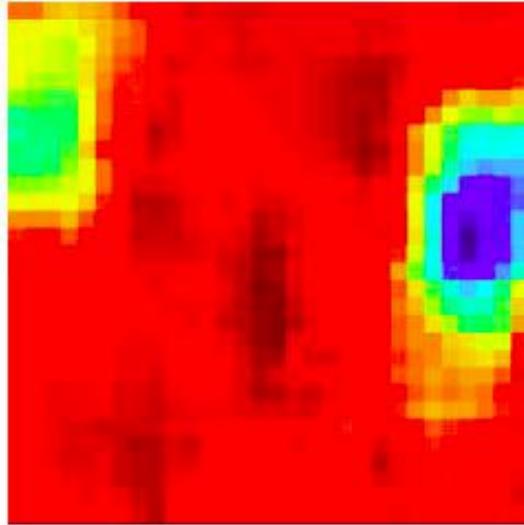


# What Does the Network React To?

Input image

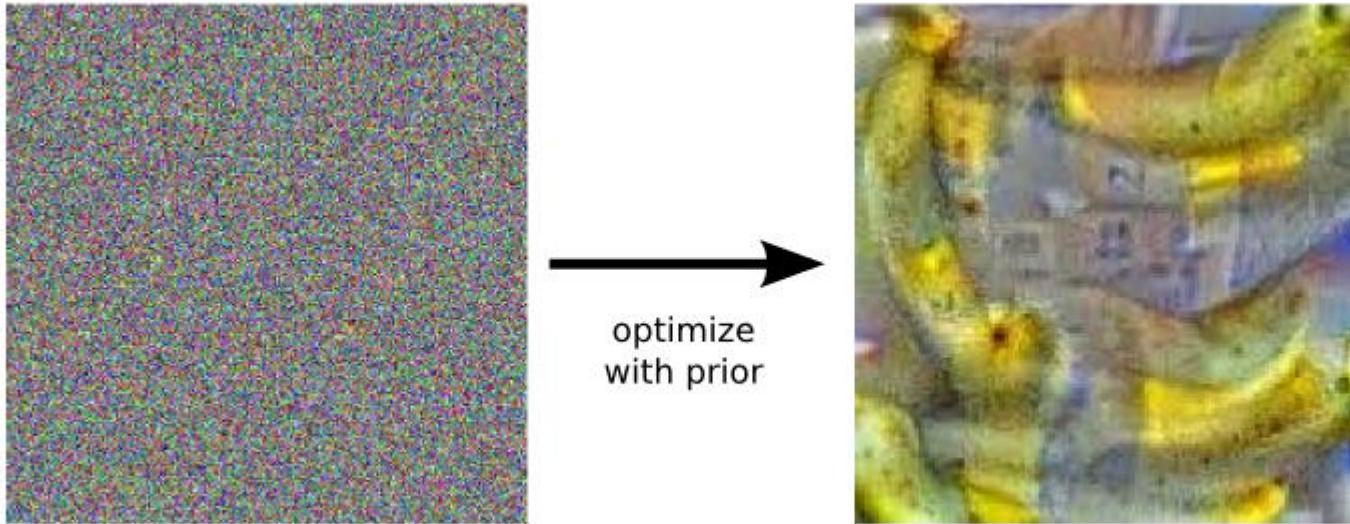


Total activation in most active 5<sup>th</sup> layer feature map



Other activations from the same feature map.

# Inceptionism: Dreaming ConvNets



- Idea
    - Start with a random noise image.
    - Enhance the input image such as to enforce a particular response (e.g., banana).
    - Combine with prior constraint that image should have similar statistics as natural images.
- ⇒ Network hallucinates characteristics of the learned class.

# Inceptionism: Dreaming ConvNets

- Results

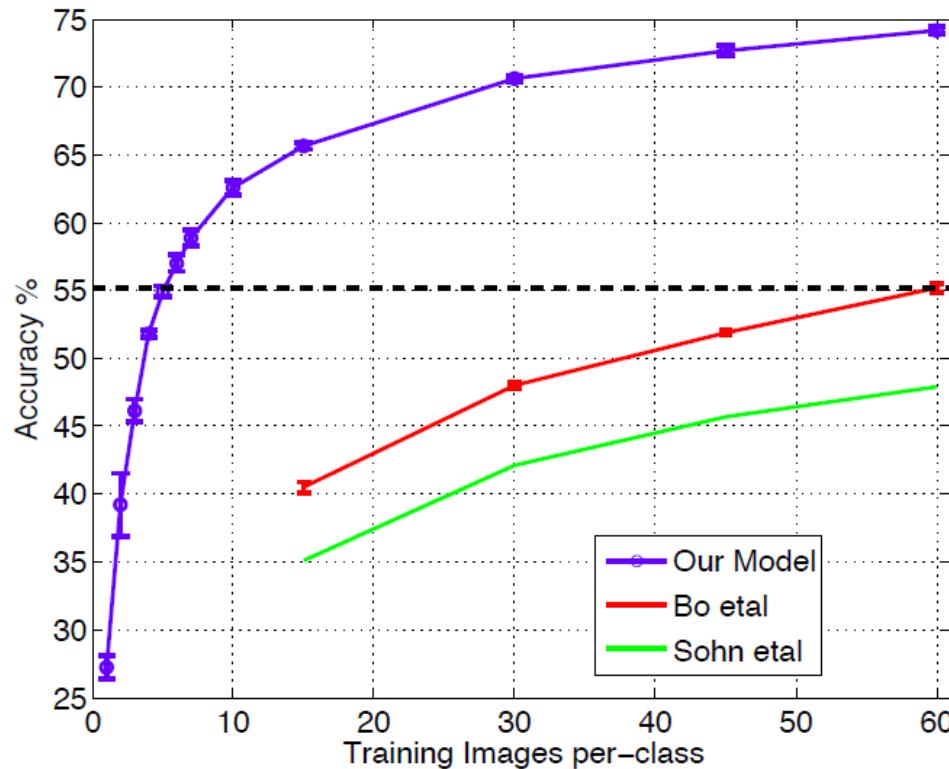


# Inceptionism: Dreaming ConvNets

# Topics of This Lecture

- Recap: CNNs
- CNN Architectures
  - LeNet
  - AlexNet
  - VGGNet
  - GoogLeNet
- Visualizing CNNs
  - Visualizing CNN features
  - Visualizing responses
  - Visualizing learned structures
- Applications

# The Learned Features are Generic

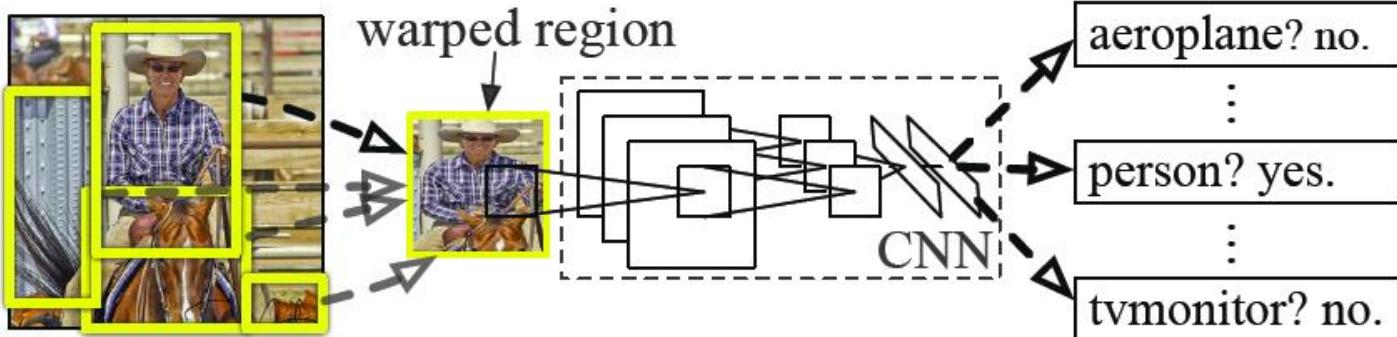


state of the art  
level (pre-CNN)

- Experiment: feature transfer
    - Train network on ImageNet
    - Chop off last layer and train classification layer on CalTech256
- ⇒ State of the art accuracy already with only 6 training images

# Other Tasks: Detection

## R-CNN: *Regions with CNN features*



1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

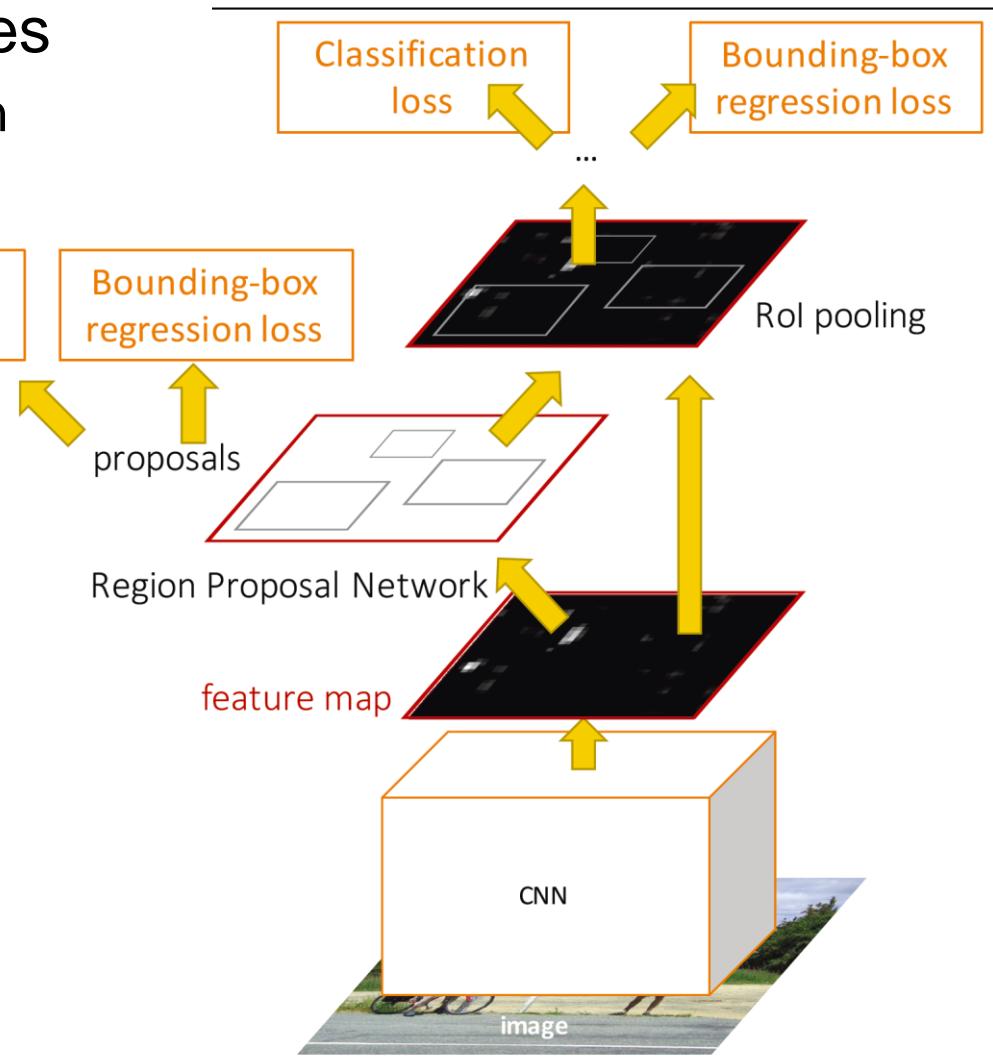
- Results on PASCAL VOC Detection benchmark

- Pre-CNN state of the art: 35.1% mAP [Uijlings et al., 2013]
  - 33.4% mAP DPM
  - R-CNN: 53.7% mAP

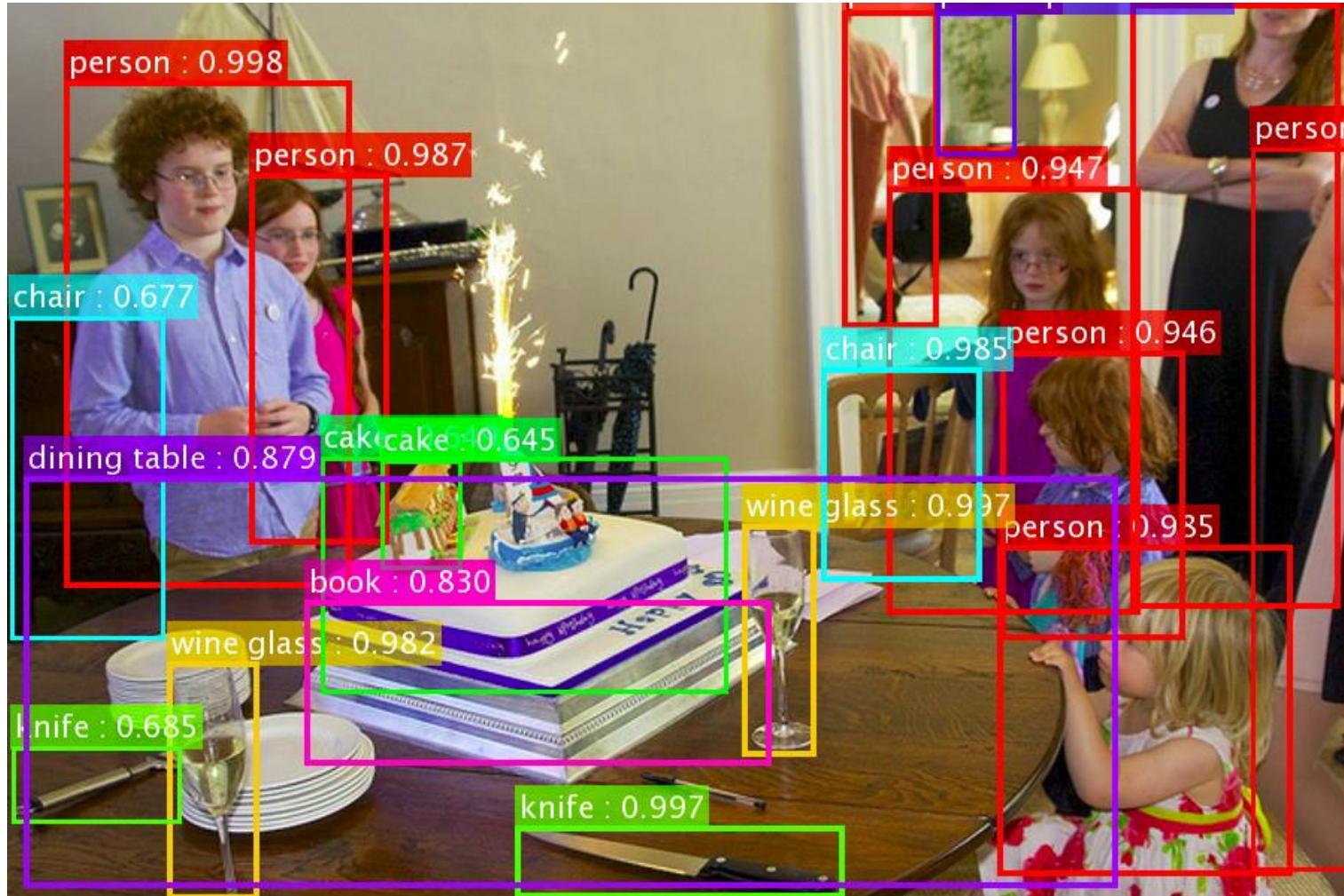
R. Girshick, J. Donahue, T. Darrell, and J. Malik, [Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation](#), CVPR 2014

# 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.
- ⇒ mAP improved to >70%

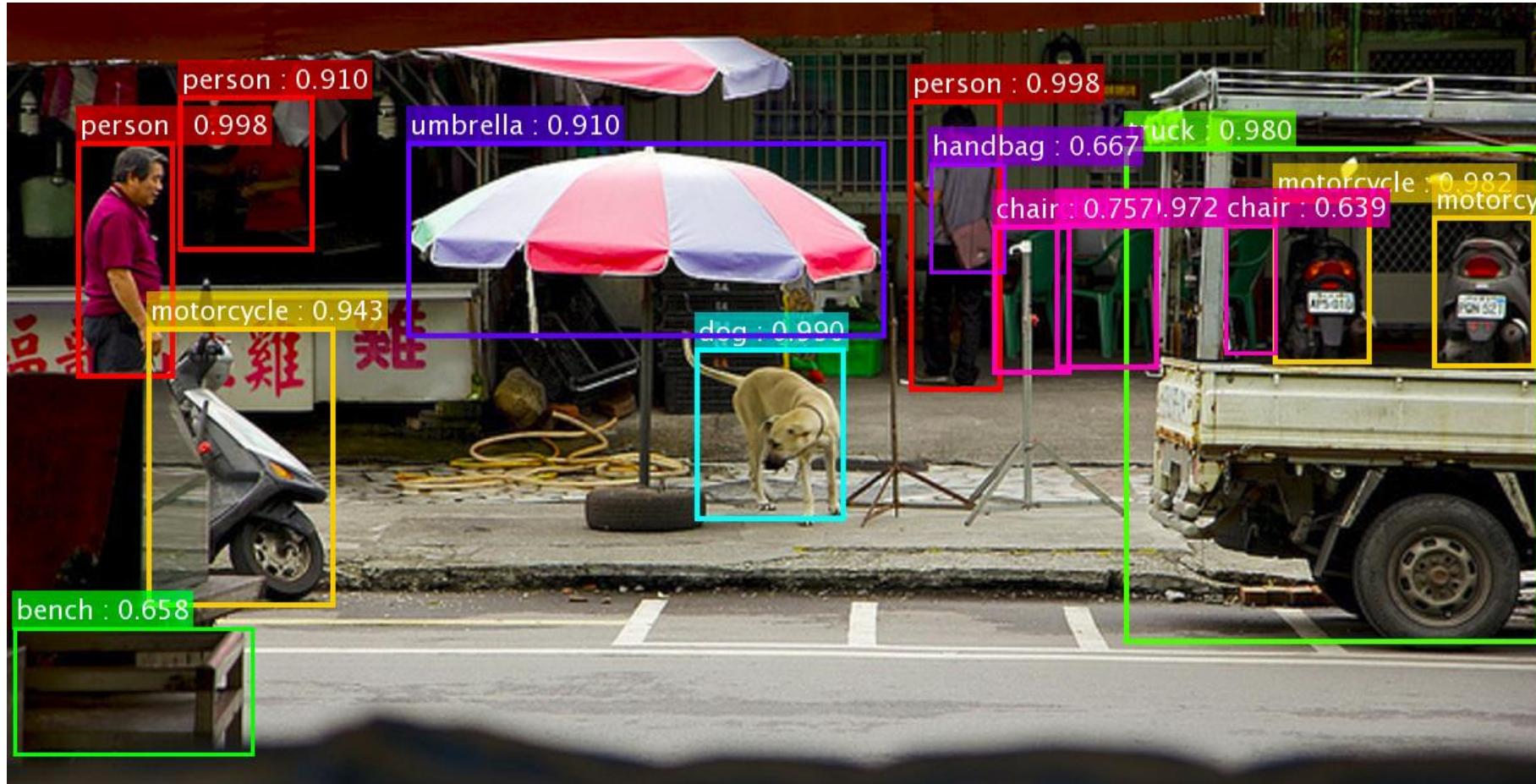


# Faster R-CNN (based on ResNets)



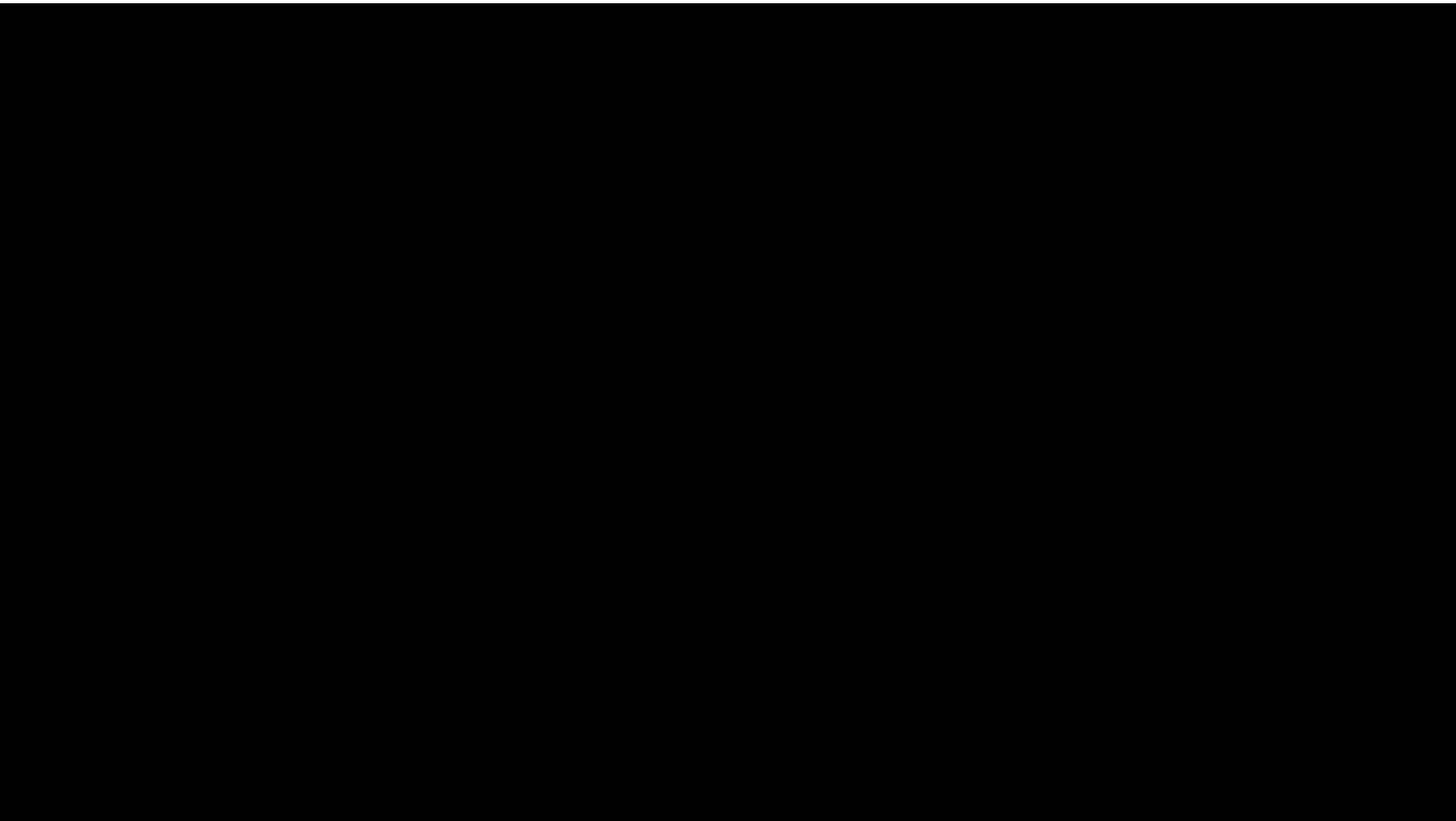
K. He, X. Zhang, S. Ren, J. Sun, [Deep Residual Learning for Image Recognition](#),  
CVPR 2016.

# Faster R-CNN (based on ResNets)

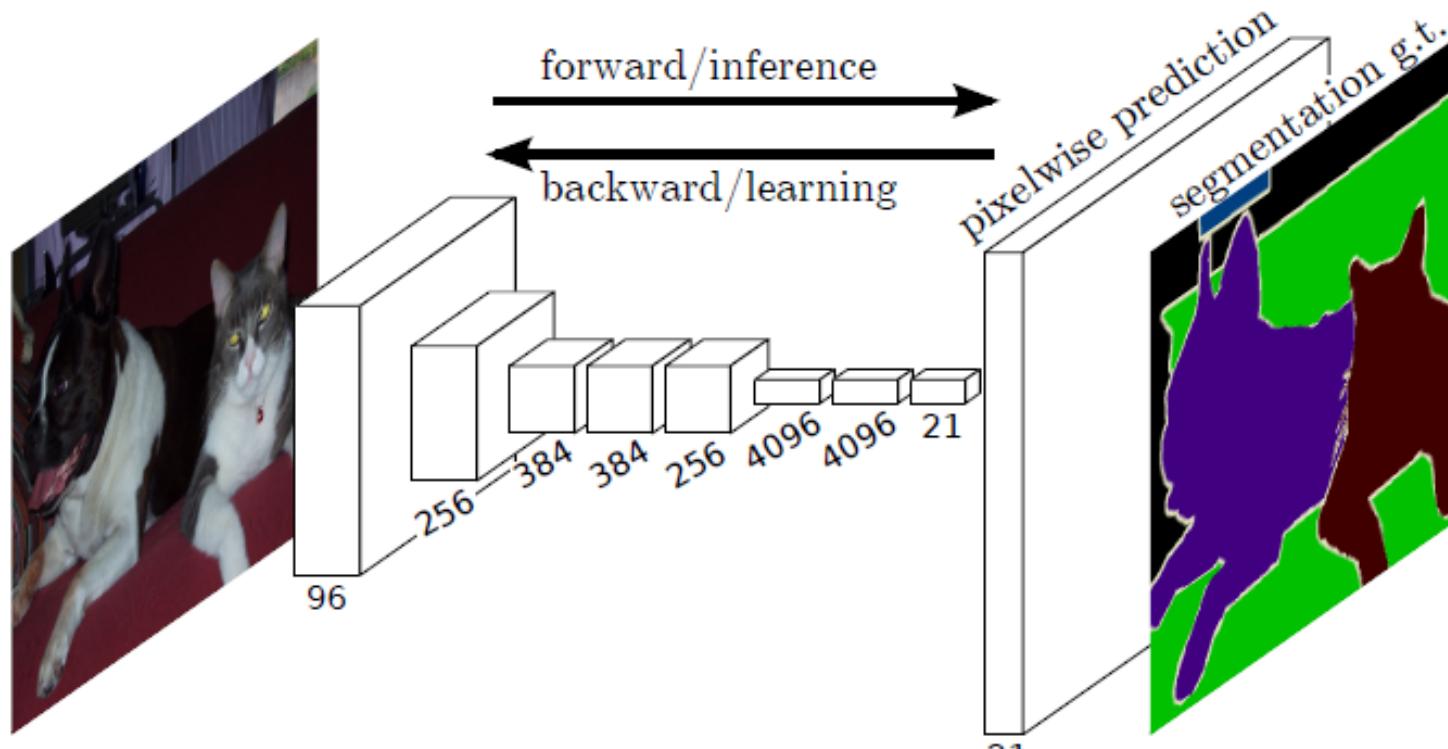


K. He, X. Zhang, S. Ren, J. Sun, [Deep Residual Learning for Image Recognition](#),  
CVPR 2016.

# YOLO

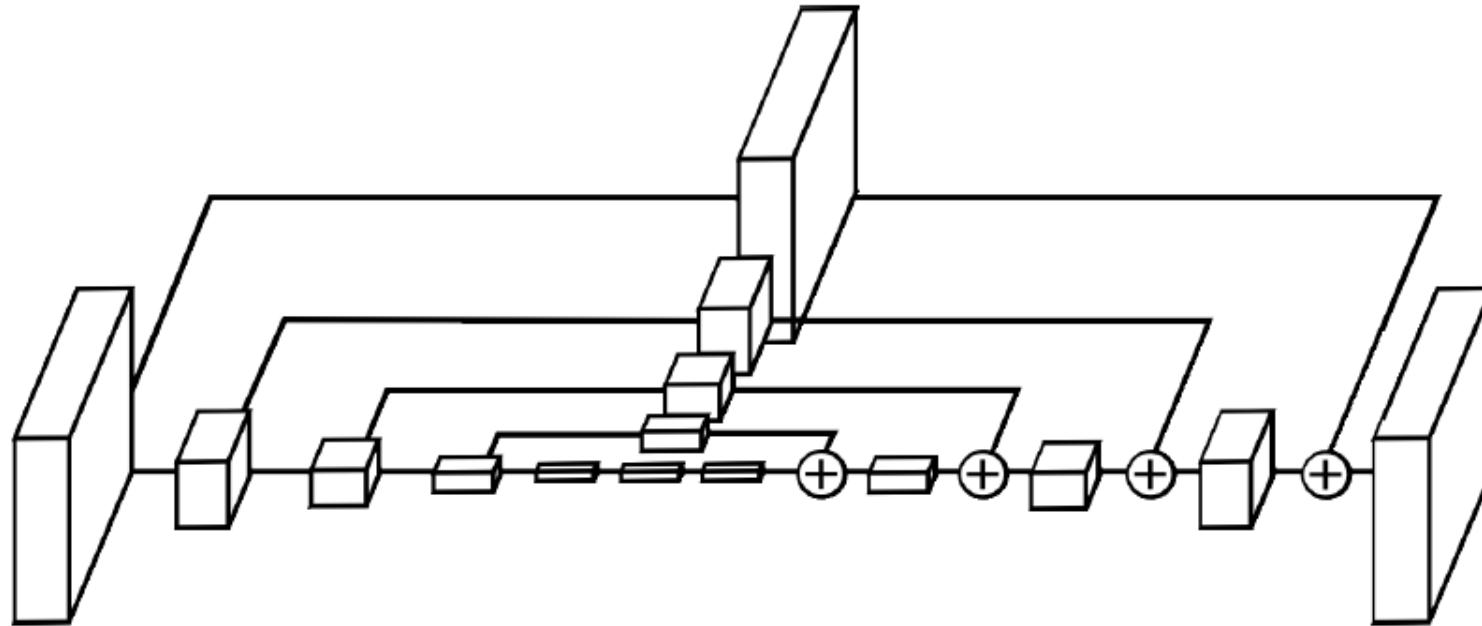


# Semantic Image Segmentation



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

# 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

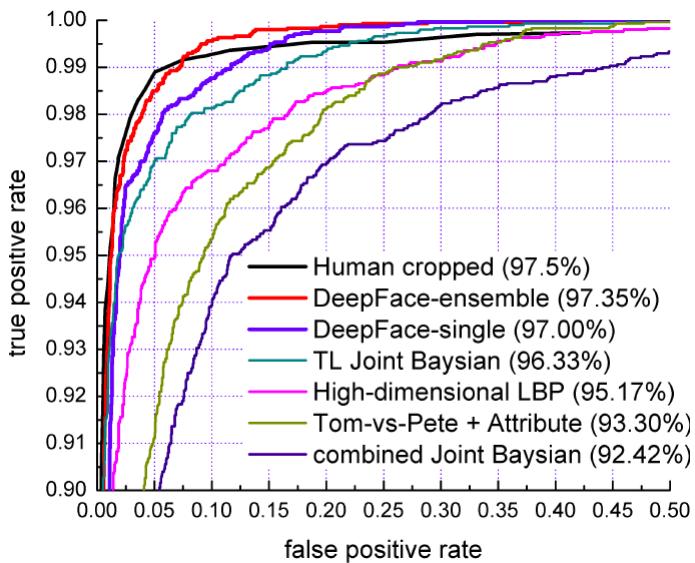
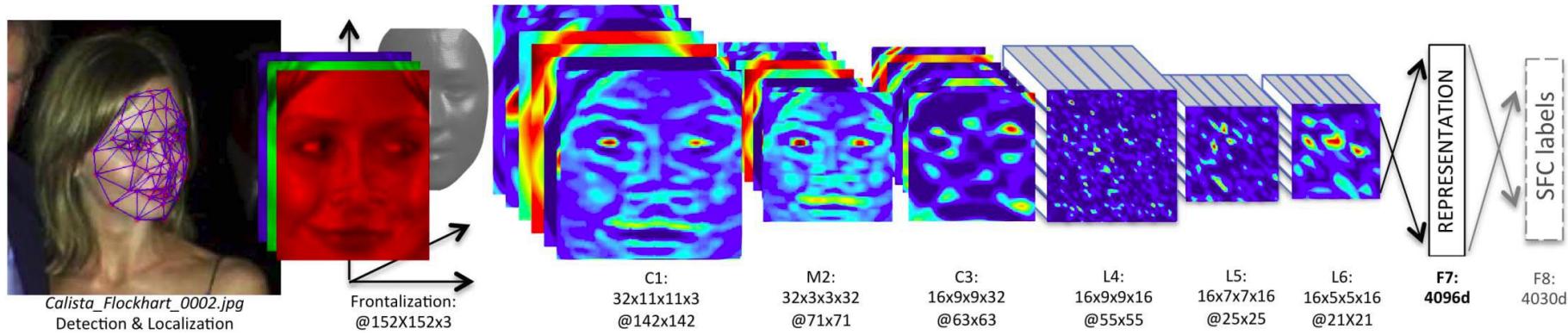
# Semantic Segmentation



[Pohlen, Hermans, Mathias, Leibe, CVPR 2017]

- More recent results
  - Based on an extension of ResNets

# Other Tasks: Face Verification



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

# Commercial Recognition Services

- E.g., **clarifai**



Try it out with your own media

Upload an image or video file under 100mb or give us a direct link to a file on the web.

Paste a url here...

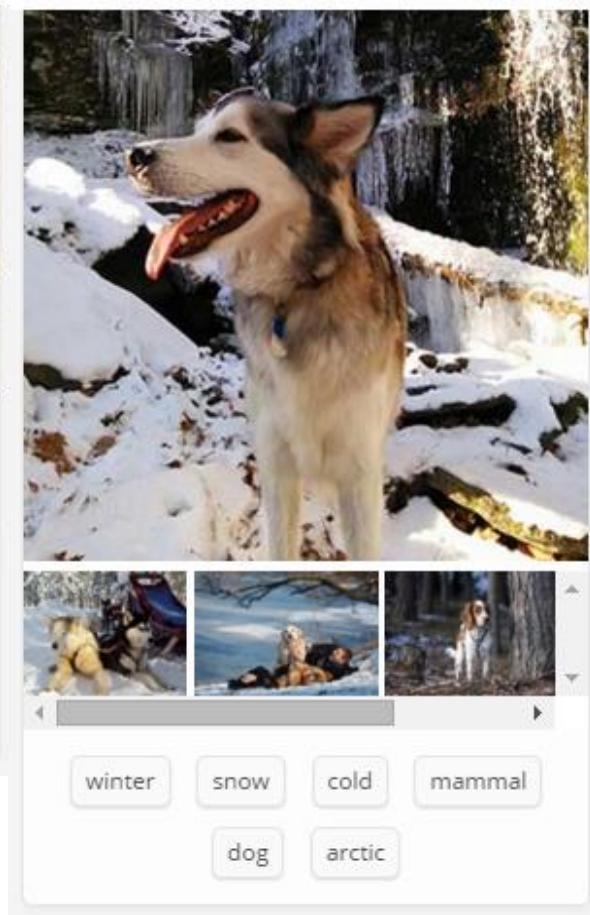
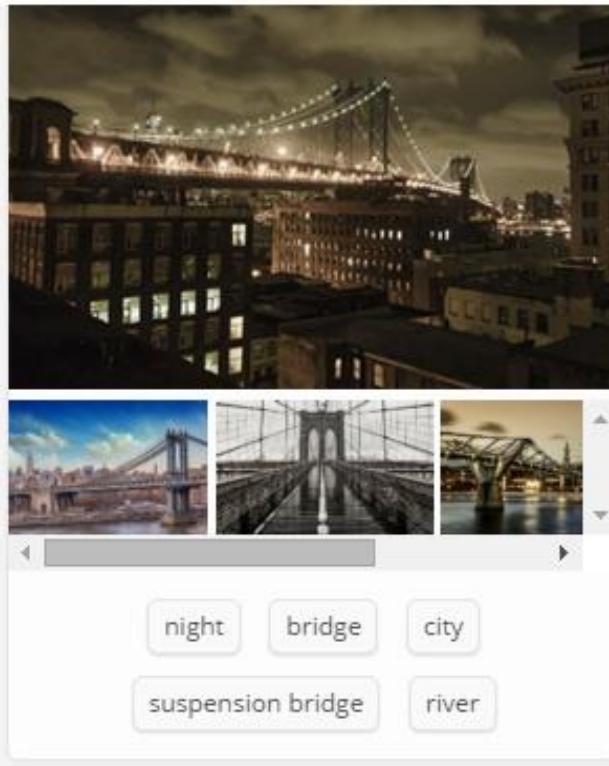
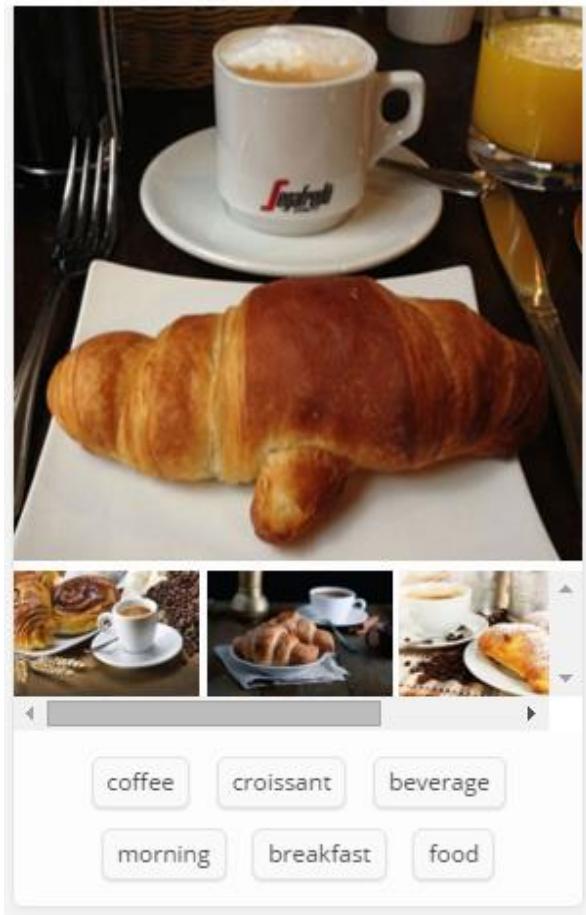
ENGLISH ▾

USE THE URL      CHOOSE A FILE INSTEAD

\*By using the demo you agree to our terms of service

- Be careful when taking test images from Google Search
  - Chances are they may have been seen in the training set...

# Commercial Recognition Services



clarifai

# References and Further Reading

- LeNet
  - 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.
- AlexNet
  - A. Krizhevsky, I. Sutskever, and G. Hinton, [ImageNet Classification with Deep Convolutional Neural Networks](#), NIPS 2012.
- VGGNet
  - K. Simonyan, A. Zisserman, [Very Deep Convolutional Networks for Large-Scale Image Recognition](#), ICLR 2015
- GoogLeNet
  - C. Szegedy, W. Liu, Y. Jia, et al, [Going Deeper with Convolutions](#), arXiv:1409.4842, 2014.

# References and Further Reading

- ResNet
  - K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, CVPR 2016.

# References

- ReLu
  - X. Glorot, A. Bordes, Y. Bengio, [Deep sparse rectifier neural networks](#), AISTATS 2011.
- Initialization
  - X. Glorot, Y. Bengio, [Understanding the difficulty of training deep feedforward neural networks](#), AISTATS 2010.
  - K. He, X.Y. Zhang, S.Q. Ren, J. Sun, [Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification](#), ArXiV 1502.01852v1, 2015.
  - A.M. Saxe, J.L. McClelland, S. Ganguli, [Exact solutions to the nonlinear dynamics of learning in deep linear neural networks](#), ArXiV 1312.6120v3, 2014.

# References and Further Reading

- Batch Normalization
  - S. Ioffe, C. Szegedy, [Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift](#), ArXiV 1502.03167, 2015.
- Dropout
  - N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, [Dropout: A Simple Way to Prevent Neural Networks from Overfitting](#), JMLR, Vol. 15:1929-1958, 2014.