Deep Neural Networks

Marcello Pelillo
University of Venice, Italy

Artificial Intelligence

a.y. 2018/19

Ca' Foscari University of Venice

European Centre for Living Technology

The Age of “Deep Learning”

News & Analysis

Microsoft, Google Beat Humans at Image Recognition

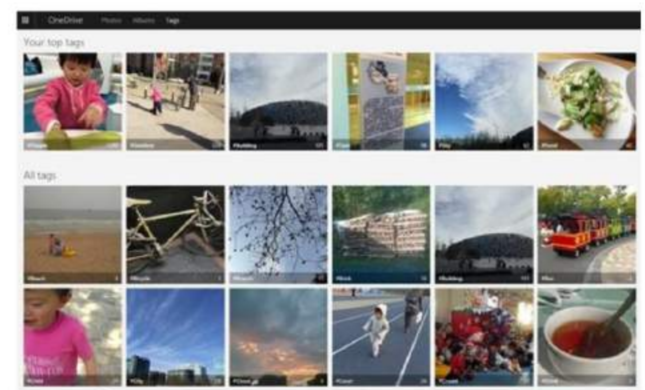
Deep learning algorithms compete at ImageNet challenge

R. Colin Johnson
2/18/2015 08:15 AM EST
14 comments

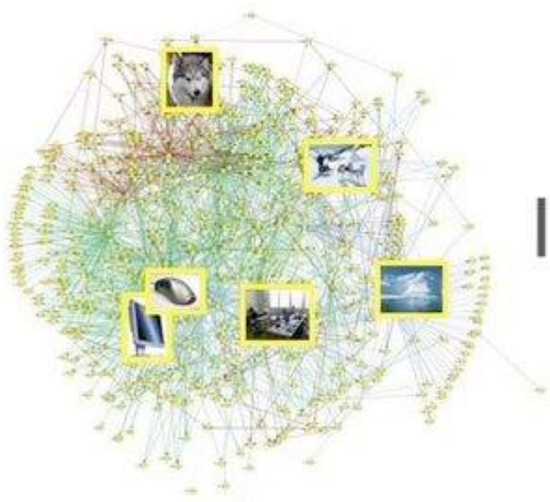
NO RATINGS
1 saves
LOGIN TO RATE

PORTLAND, Ore. -- First computers beat the best of us at chess, then poker, and finally Jeopardy. The next hurdle is image recognition — surely a computer can't do that as well as a human. Check that one off the list, too. Now Microsoft has programmed the first computer to beat the humans at image recognition.

The competition is fierce, with the ImageNet Large Scale Visual Recognition Challenge doing the judging for the 2015 championship on December 17. Between now and then expect to see a stream of papers claiming they have one-upped humans too. For instance, only 5 days after Microsoft announced it had beat the human benchmark of 5.1% errors with a 4.94% error grabbing neural network, Google announced it had one-upped Microsoft by 0.04%.



The top row is a representative of the categories that Microsoft's algorithm found in the database and the image columns below are examples that fit. (Source: Microsoft)



IMAGENET

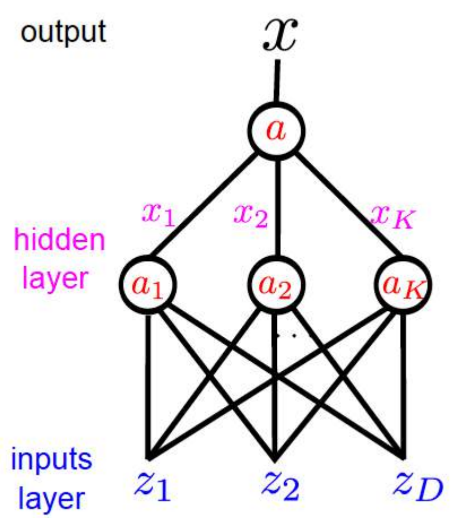
The Deep Learning “Philosophy”

- Learn a feature hierarchy all the way from pixels to classifier
- Each layer extracts features from the output of previous layer
- Train all layers jointly

Shallow *vs* Deep Networks

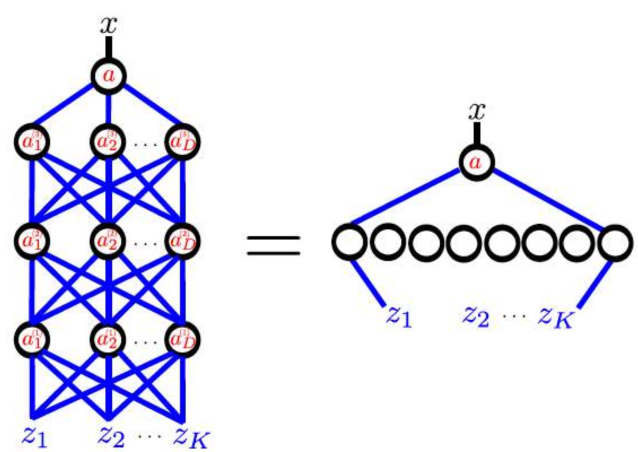
Shallow architectures are inefficient at representing deep functions

single layer neural network
implements: $x = f_\theta(\mathbf{z})$



networks we met last lecture
with large enough single hidden layer
can implement **any** function
'universal approximator'

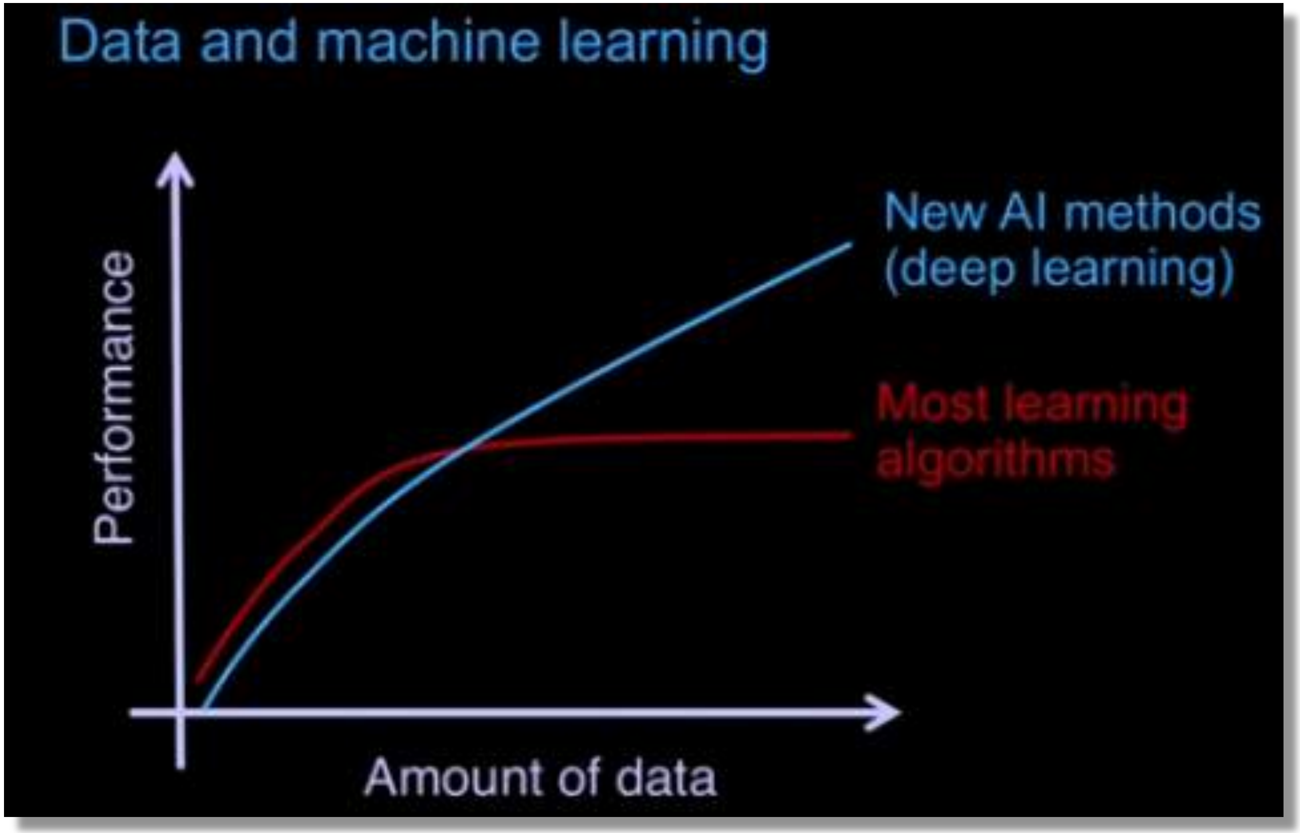
shallow networks can be
computationally inefficient



however, if the function is 'deep'
a very large hidden layer may
be required

From. R. E. Turner

Performance Improves with More Data



Old Idea… Why Now?

1. We have more data - from Lena to ImageNet.

2. We have more computing power, GPUs are really good at this.

3. Last but not least, we have new ideas

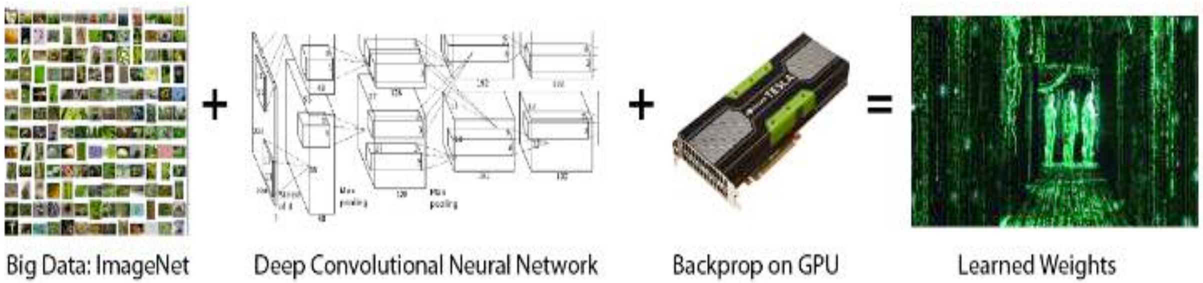
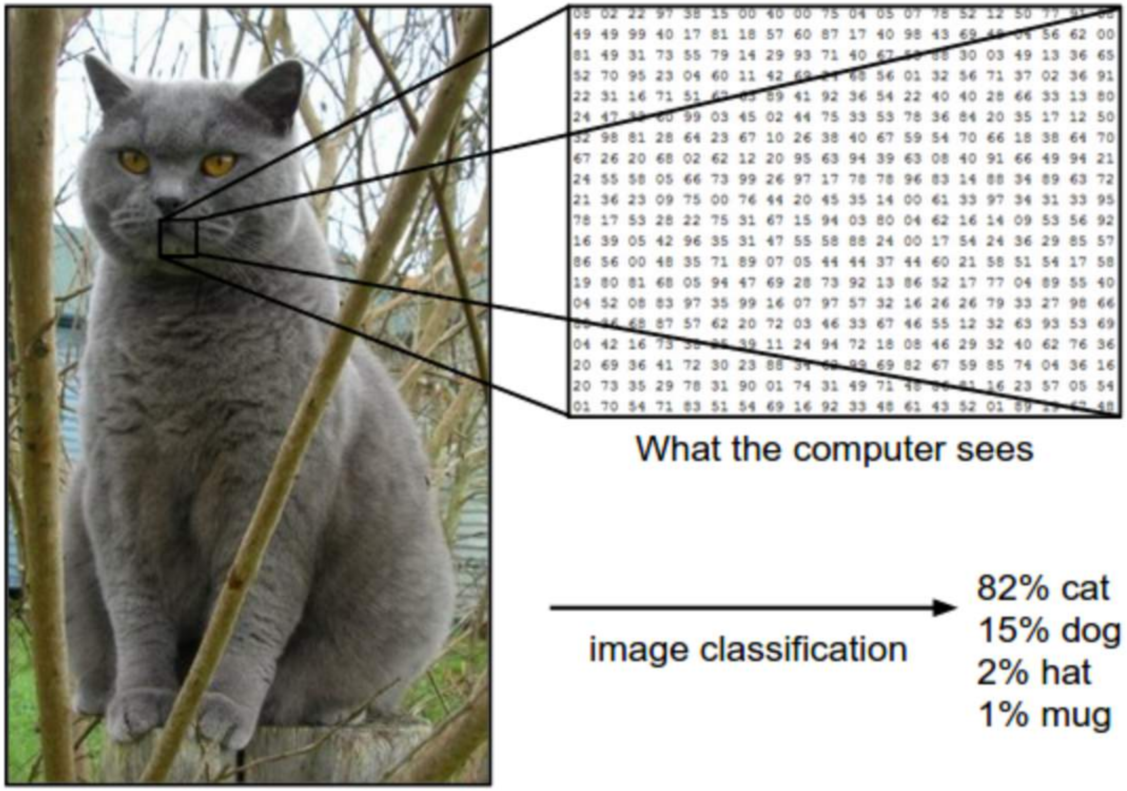


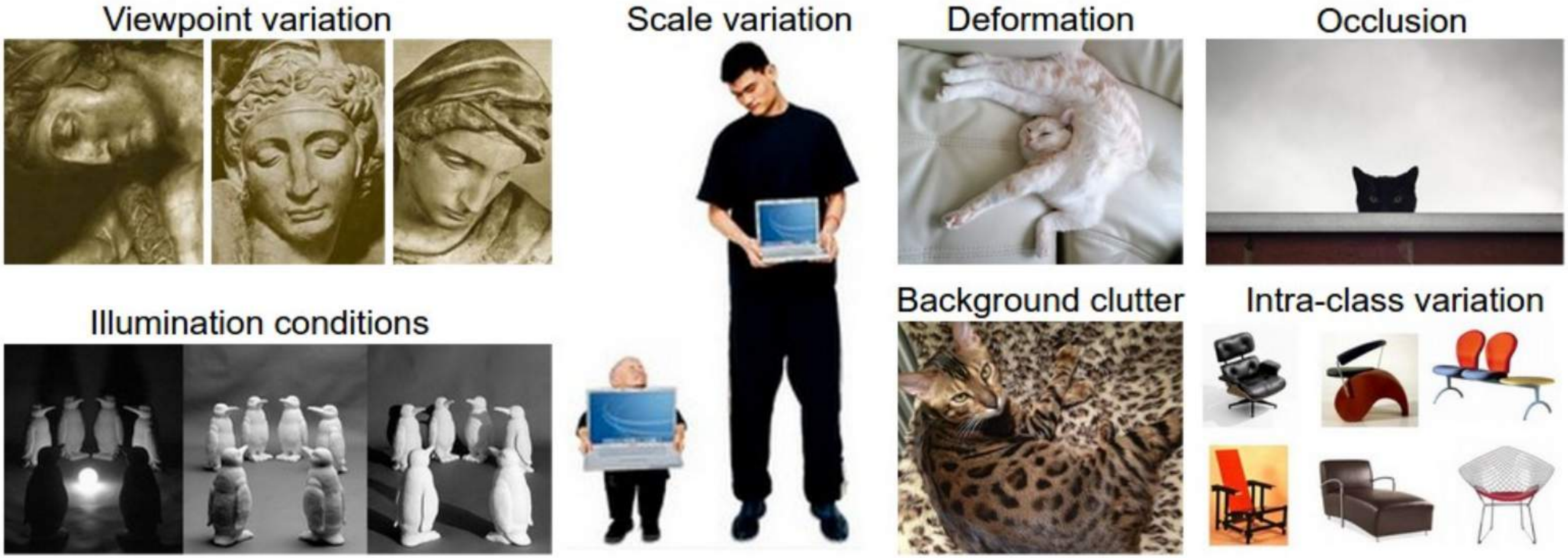
Image Classification



Predict a single label (or a distribution over labels as shown here to indicate our confidence) for a given image. Images are 3-dimensional arrays of integers from 0 to 255, of size Width x Height x 3. The 3 represents the three color channels Red, Green, Blue.

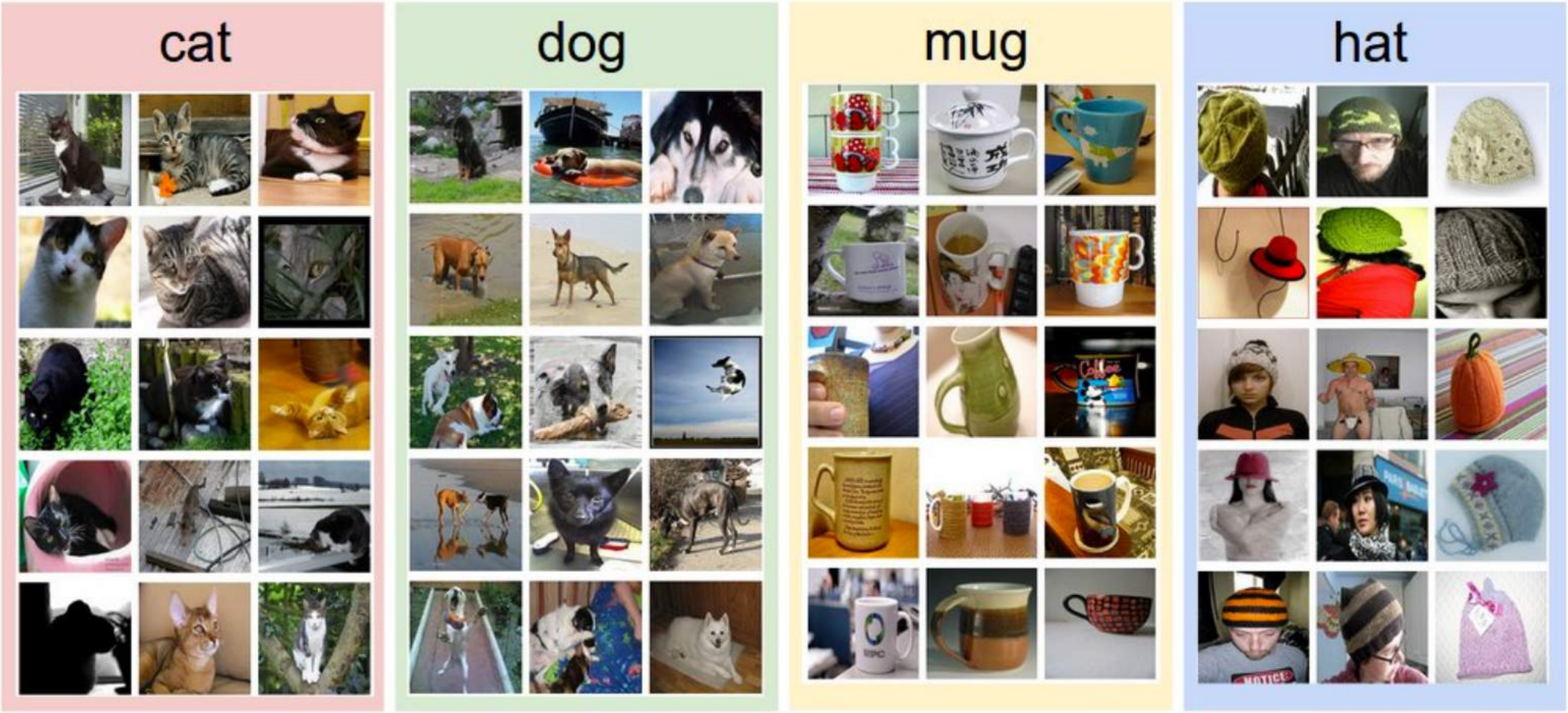
From: A. Karpathy

Challenges



From: A. Karpathy

The Data-Driven Approach



An example training set for four visual categories.

In practice we may have thousands of categories and hundreds of thousands of images for each category.

From: A. Karpathy

Inspiration from Biology

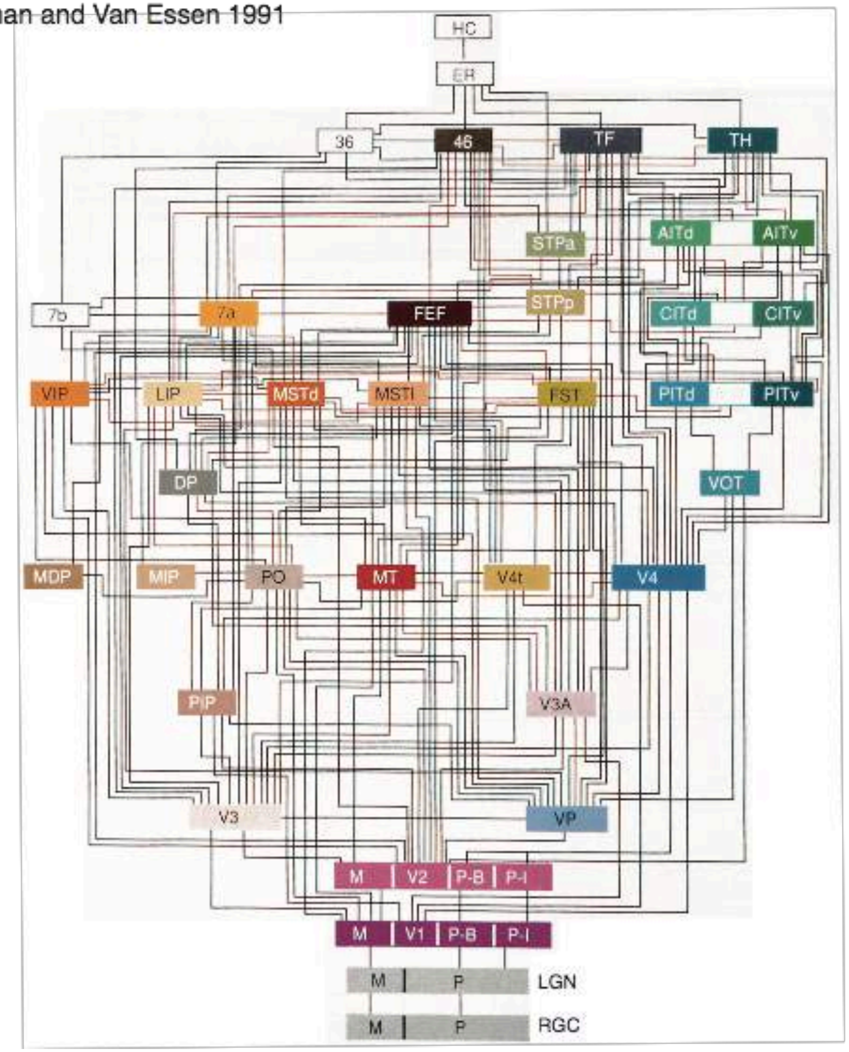
Biological vision is hierachically organized

object	trees	Inferotemporal cortex
↑	↑	
object parts	bark, leaves, etc.	V4: different textures
↑	↑	
primitive features	oriented edges	V1: simple and complex cells
↑	↑	
input image	forest image	photo-receptors retina

From. R. E. Turner

Hierarchy of Visual Areas

Hierarchy of Cortical Visual Areas
Felleman and Van Essen 1991



From. D. Zoccolan

The Retina

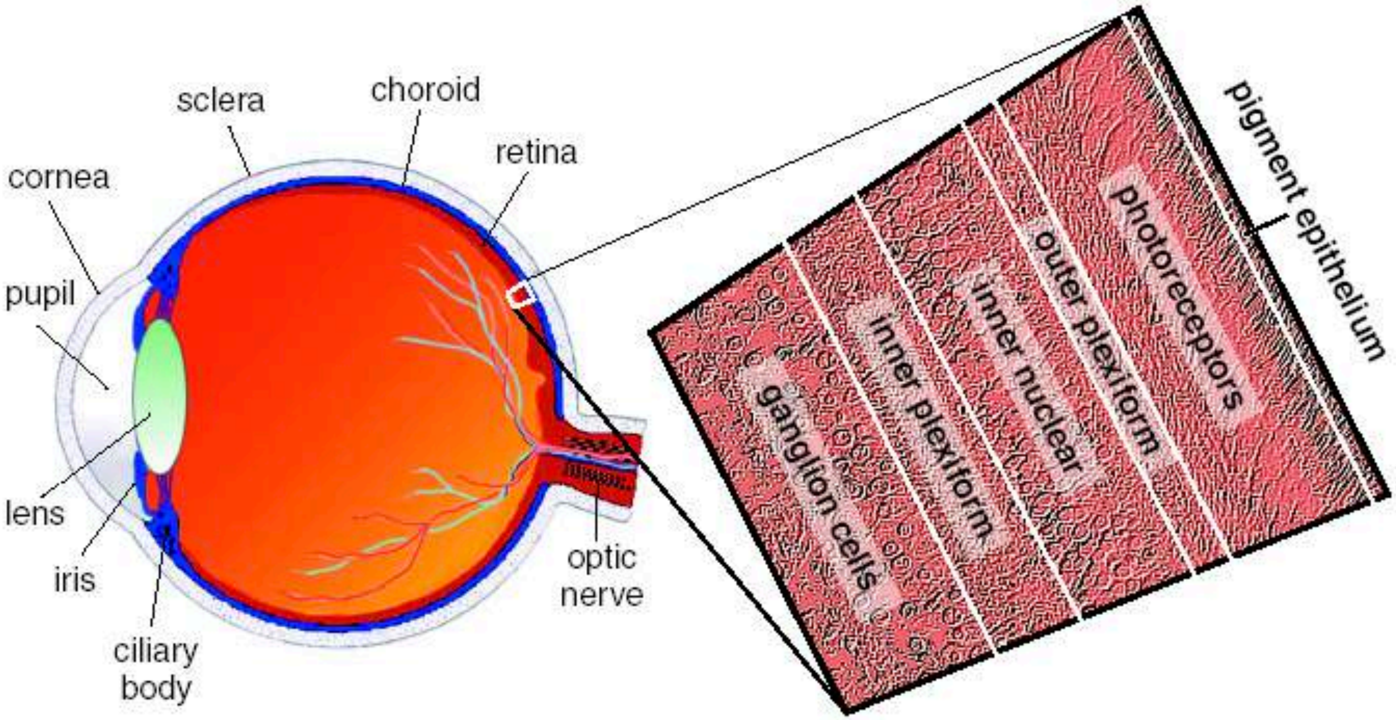


Figure 2. Diagram of a human eye shows its various structures *(left)*. A thin piece of retina is enlarged in a photomicrograph *(right)*, revealing its layers. The photoreceptors lie against a dark row of cells called the pigment epithelium. (Drawing by the author. Except where noted, photographs by Nicolas Cuenca and the author.)

The Retina

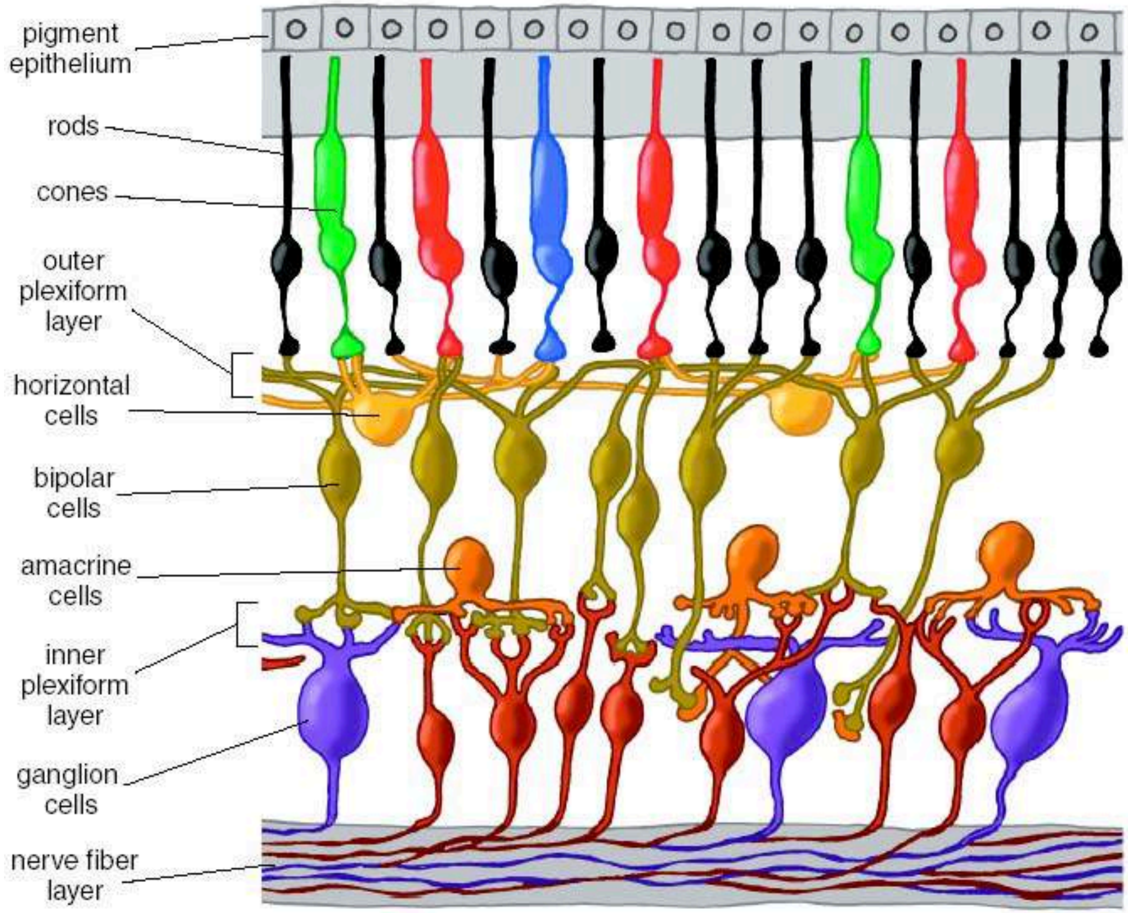
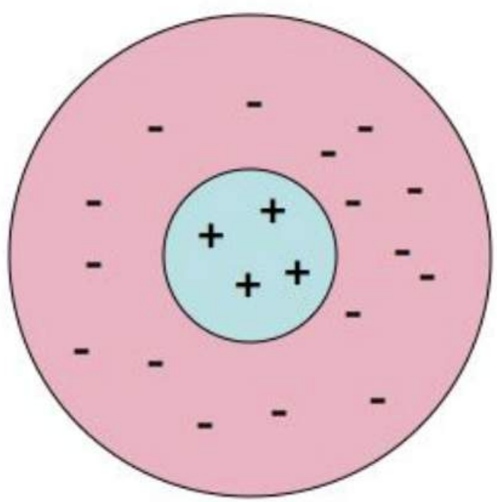


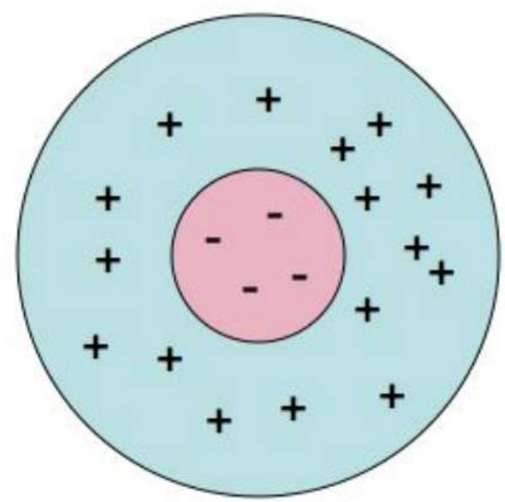
Figure 3. Cells in the retina are arrayed in discrete layers. The photoreceptors are at the top of this rendering, close to the pigment epithelium. The bodies of horizontal cells and bipolar cells compose the inner nuclear layer. Amacrine cells lie close to ganglion cells near the surface of the retina. Axon-to-dendrite neural connections make up the plexiform layers separating rows of cell bodies.

Receptive Fields

"The region of the visual field in which light stimuli evoke responses of a given neuron."



On-center, Off-surround



Off-center, On-surround

Cellular Recordings

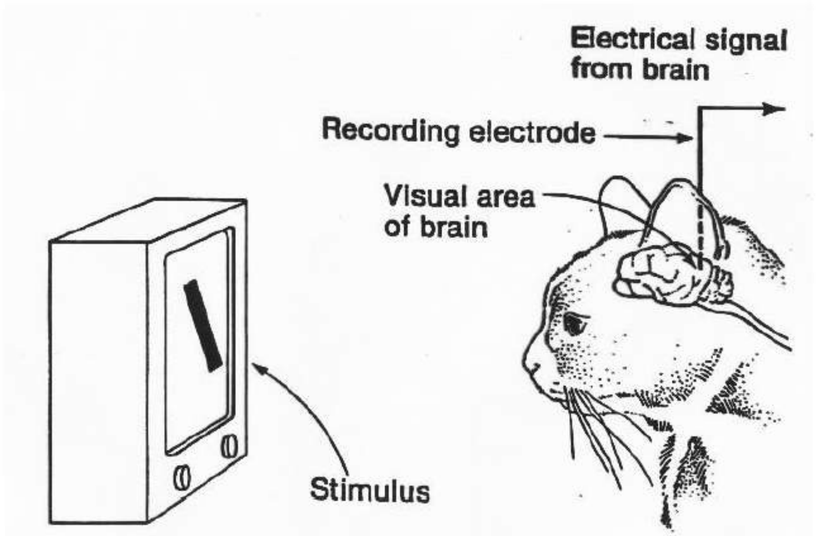
Kuffler, Hubel, Wiesel, …

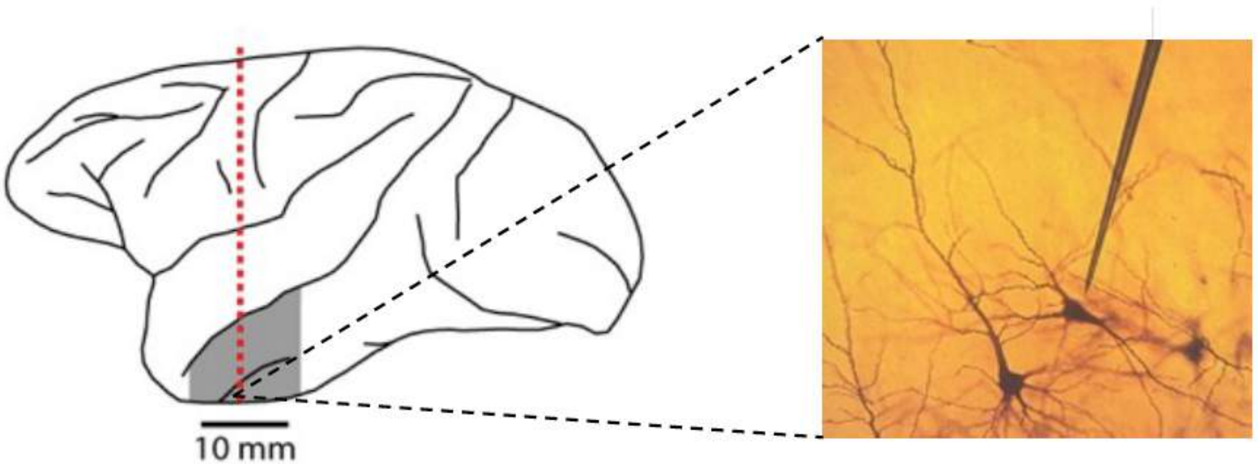
1953: *Discharge patterns and functional organization of mammalian retina*

1959: *Receptive fields of single neurones in the cat's striate cortex*

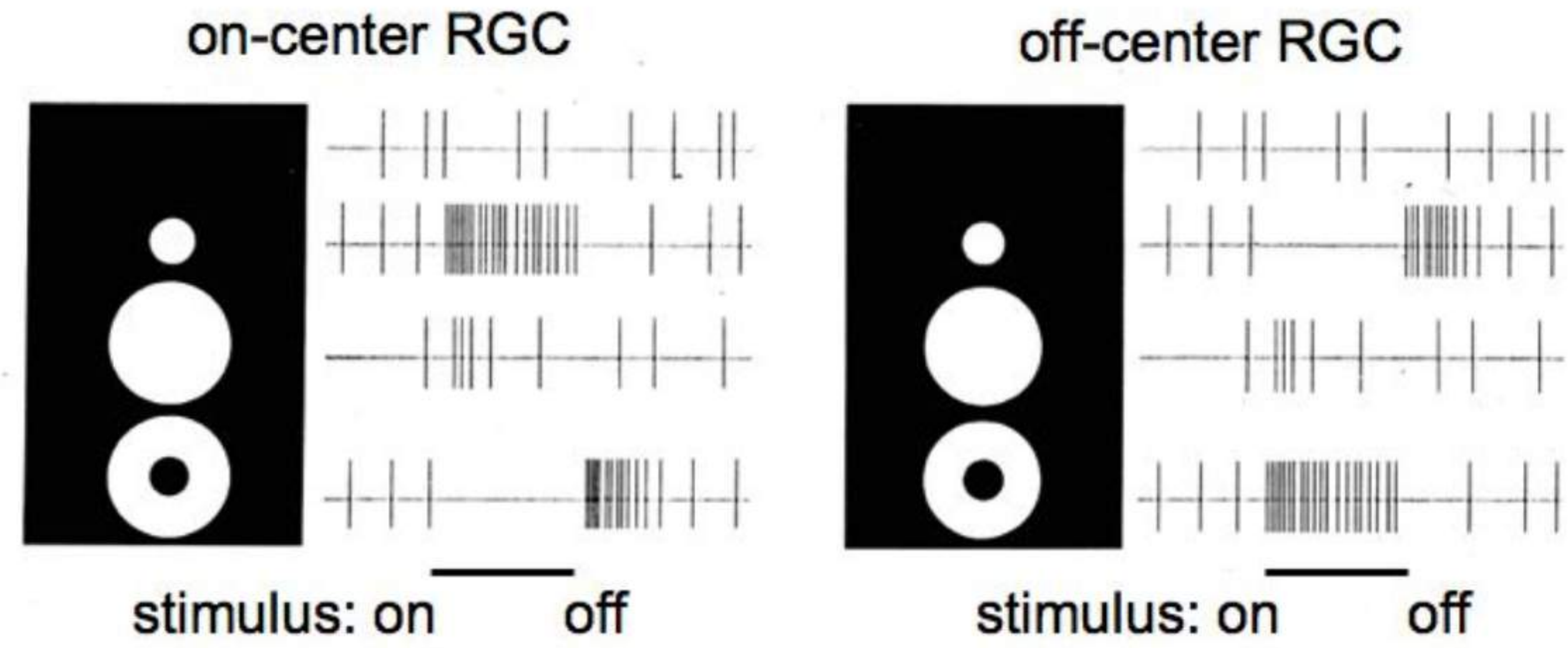
1962: *Receptive fields, binocular interaction and functional architecture in the cat's visual cortex*

1968 ..

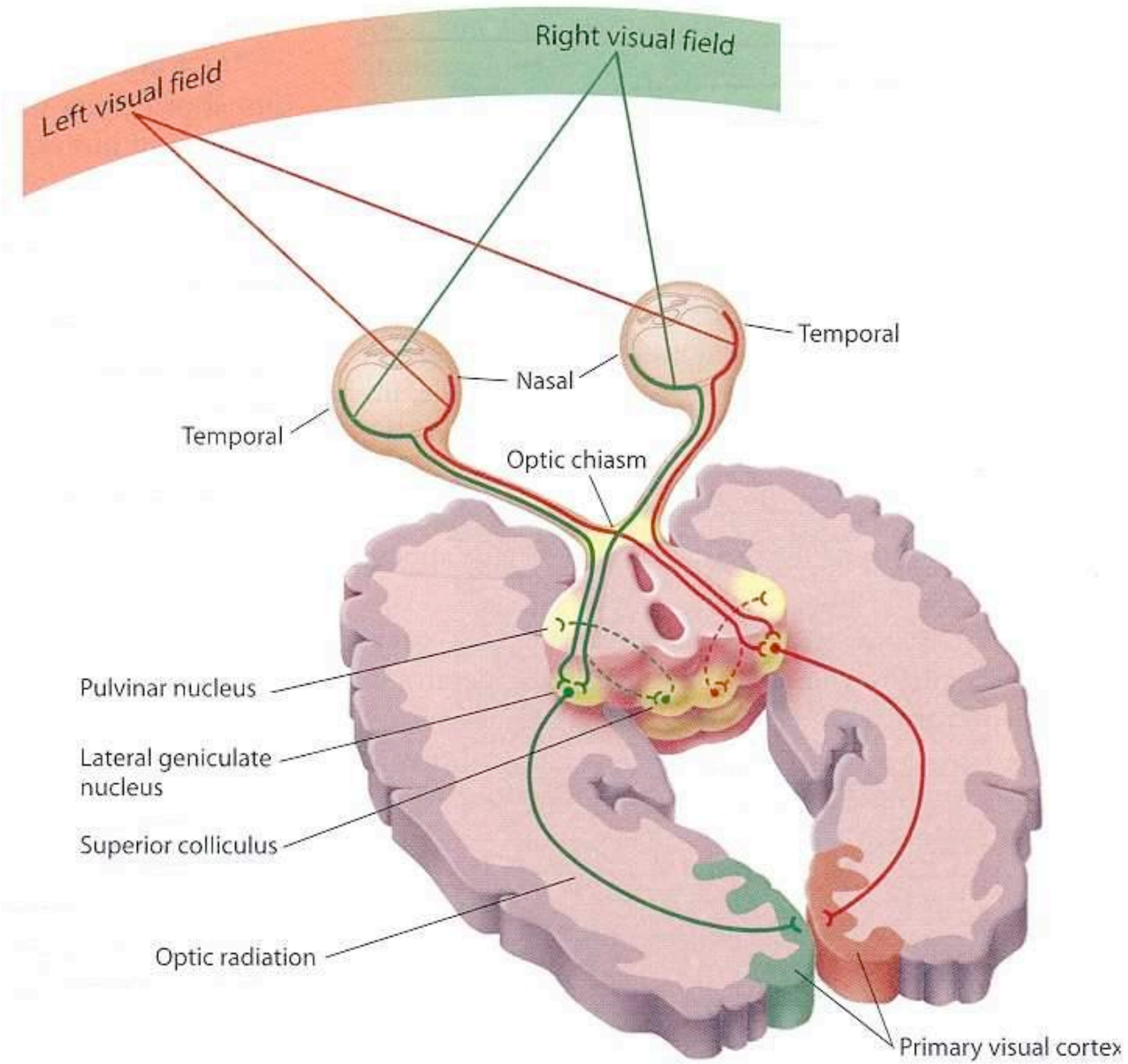




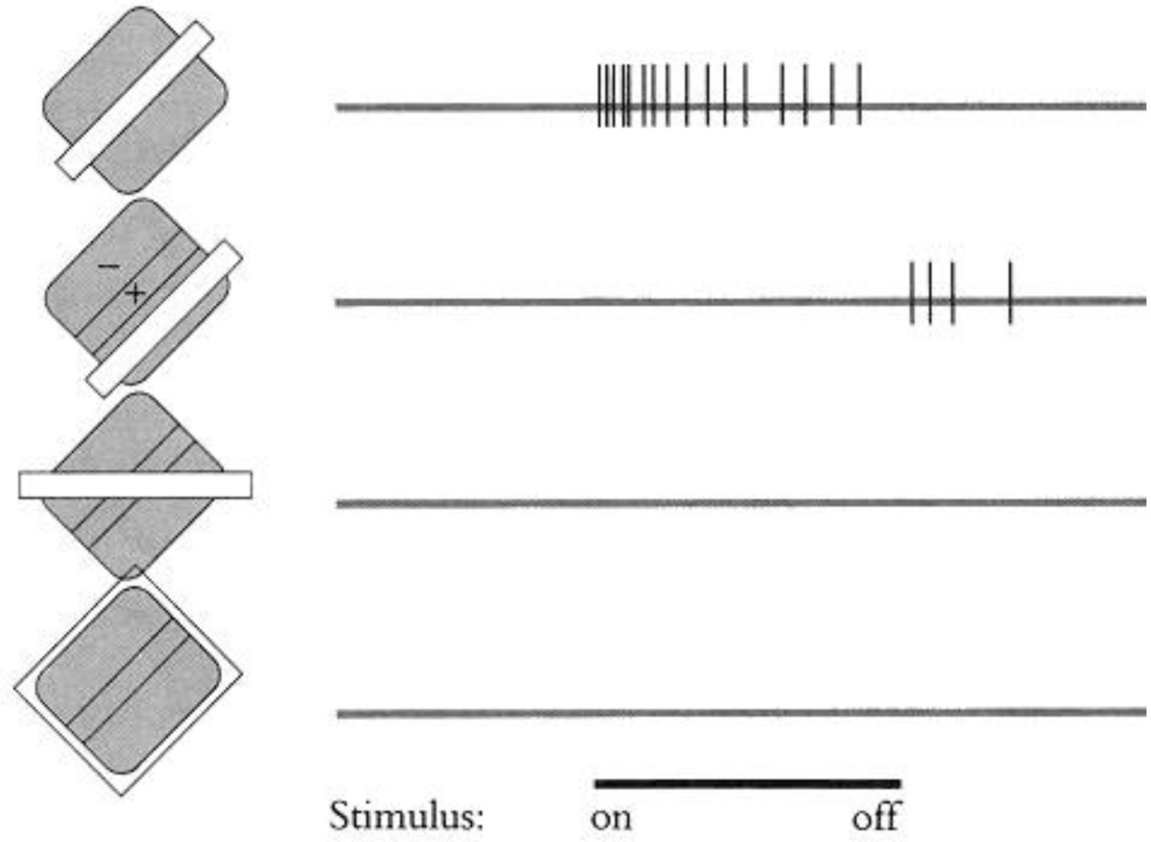
Retinal Ganglion Cell Response



Beyond the Retina

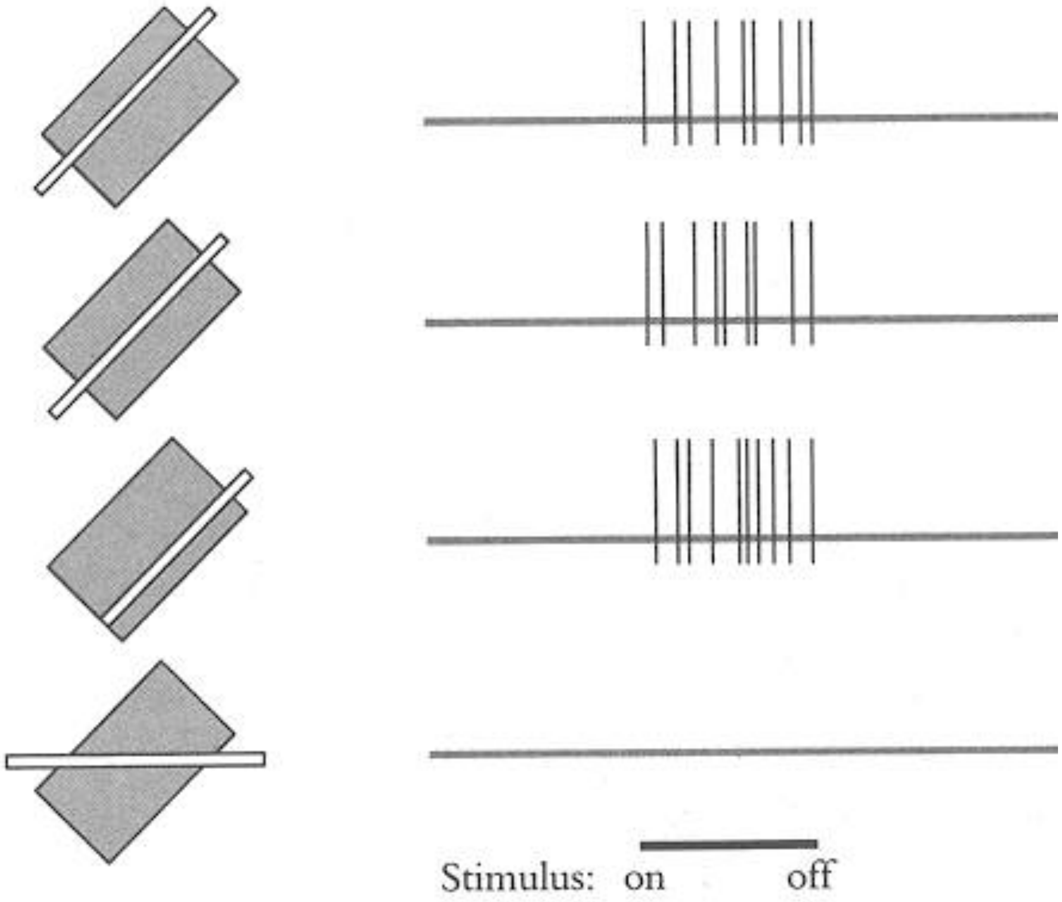


Simple Cells

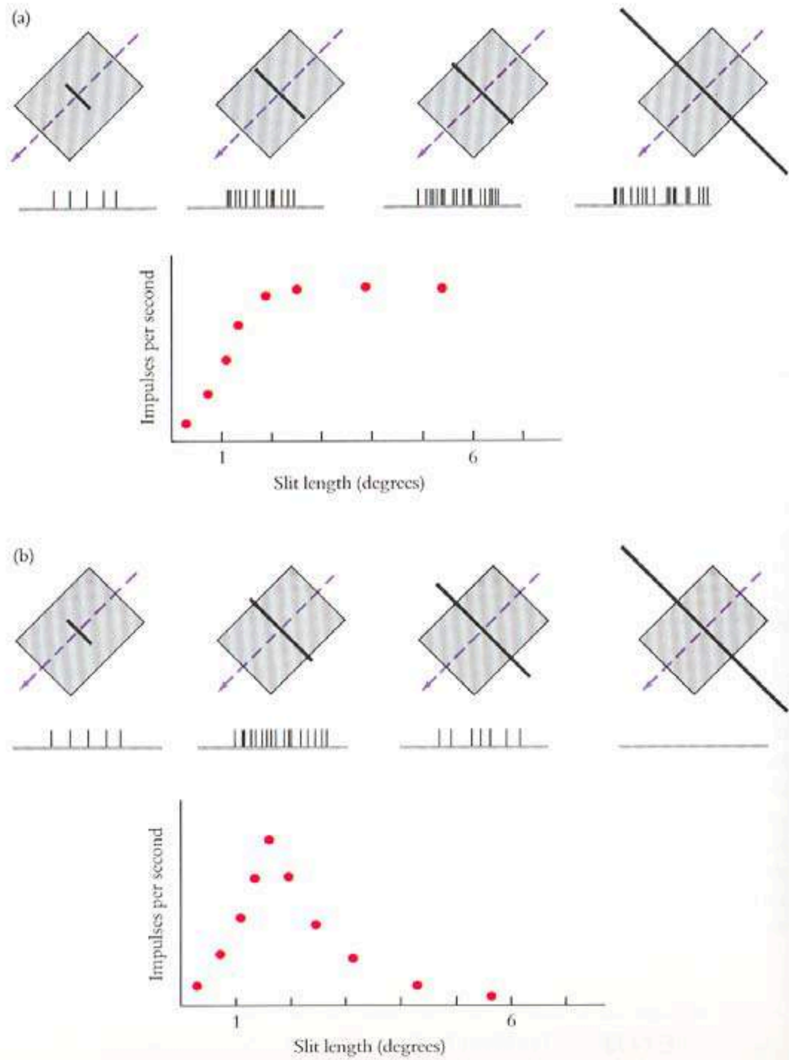


Orientation selectivity: Most V1 neurons are orientation selective meaning that they respond strongly to lines, bars, or edges of a particular orientation (e.g., vertical) but not to the orthogonal orientation (e.g., horizontal).

Complex Cells

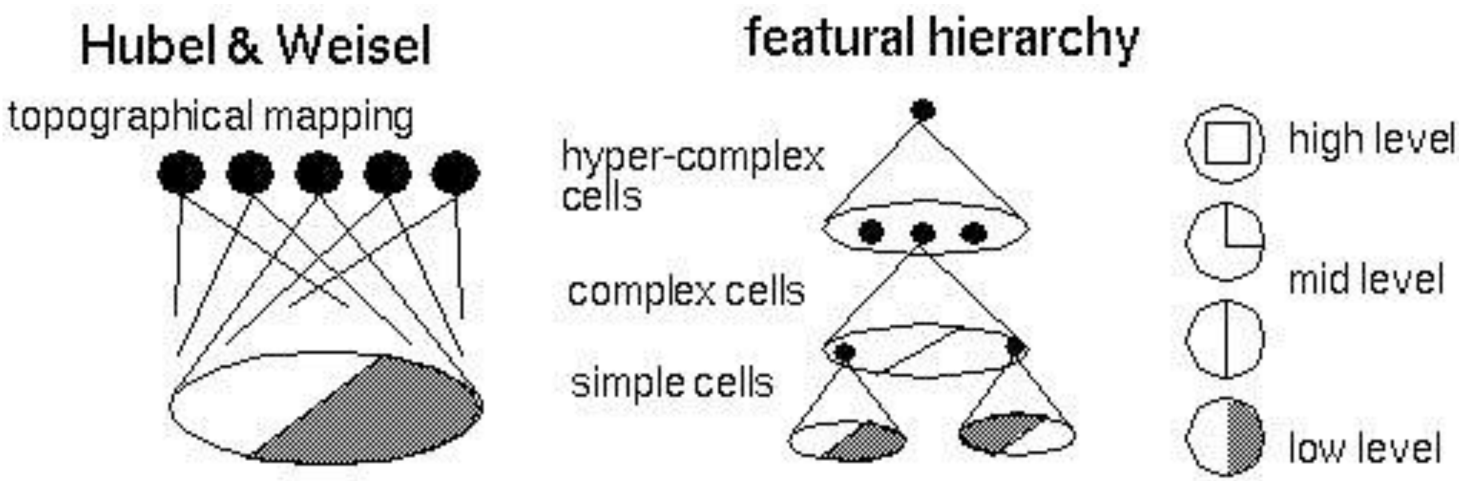


Hypercomplex Cells (end-stopping)

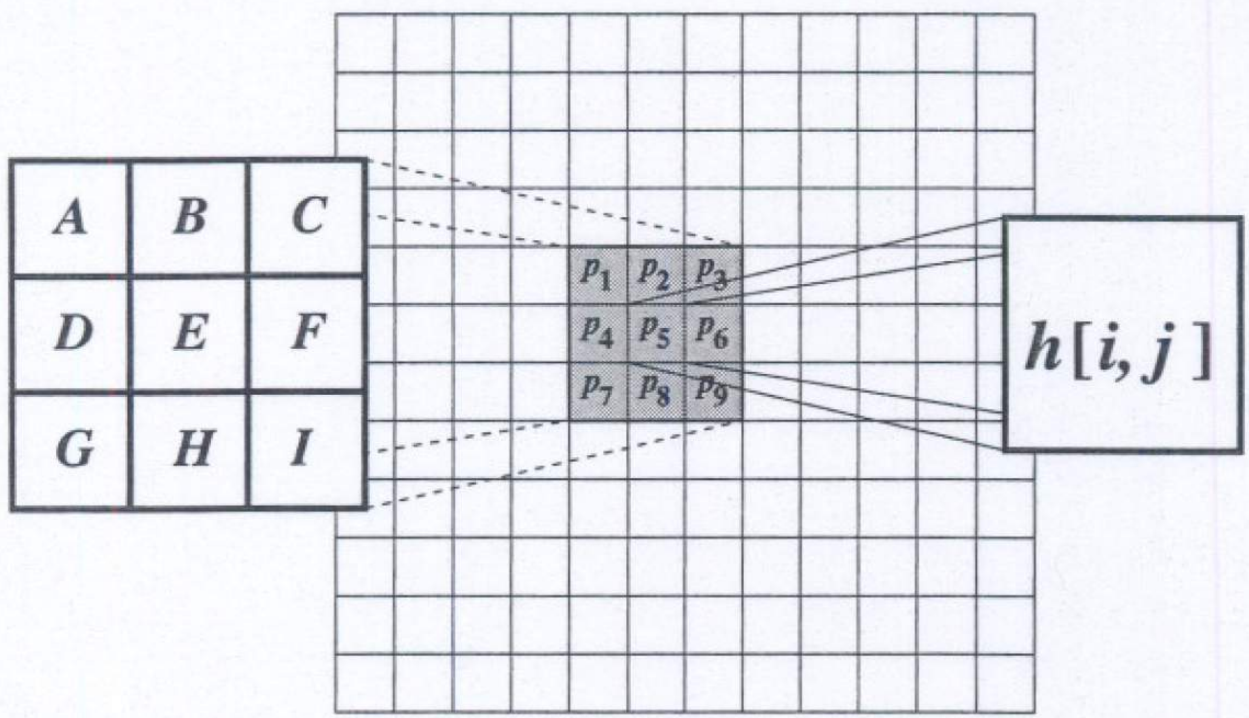


Top: An ordinary complex cell responds to various lengths of a slit of light. The duration of each record is 2 seconds. As indicated by the graph of response versus slit length, for this cell the response increases with length up to about 2 degrees, after which there is no change. *Bottom:* For this end-stopped cell, responses improve up to 2 degrees but then decline, so that a line 6 degrees or longer gives no response.

Take-Home Message: Visual System as a Hierarchy of Feature Detectors

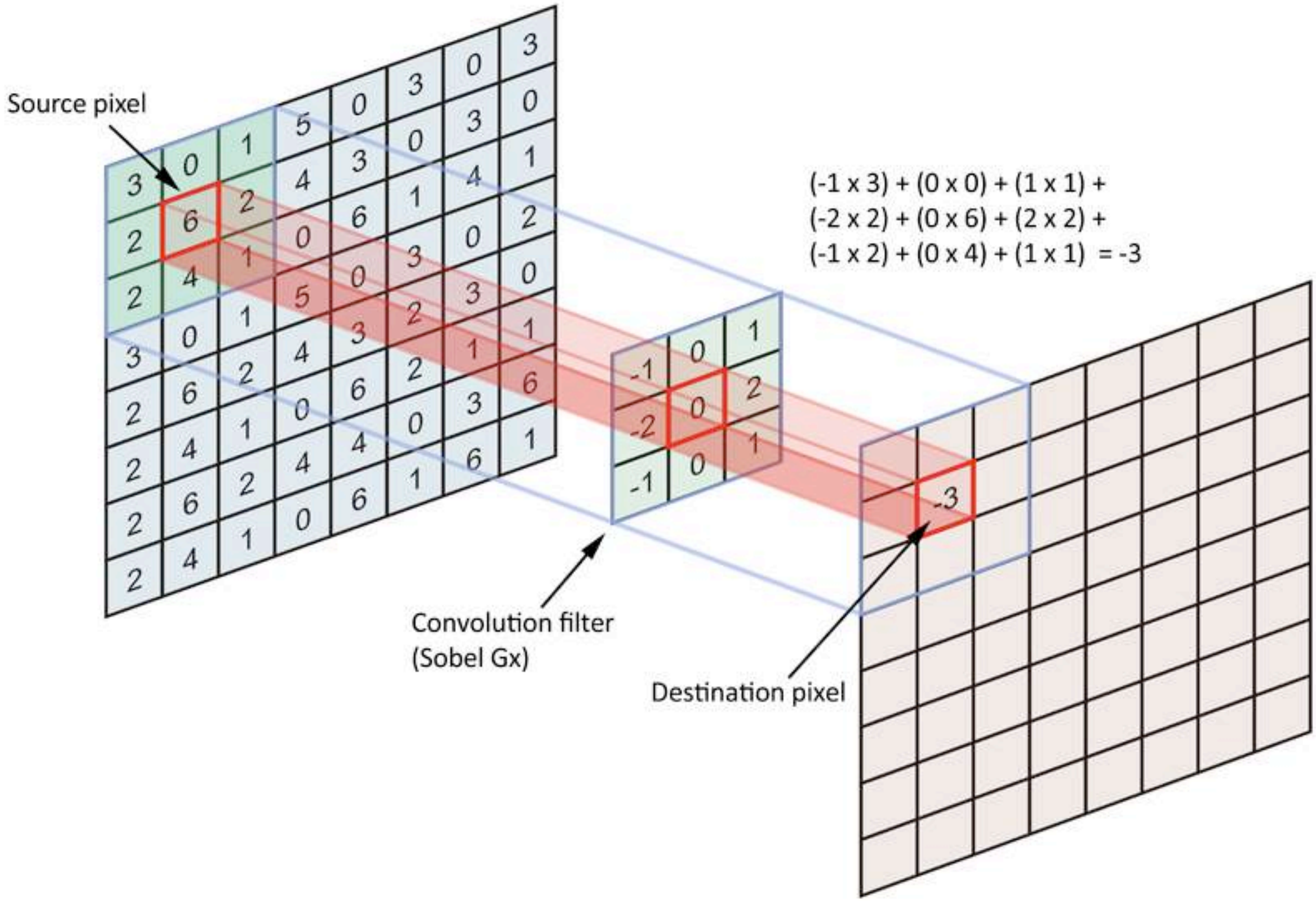


Convolution

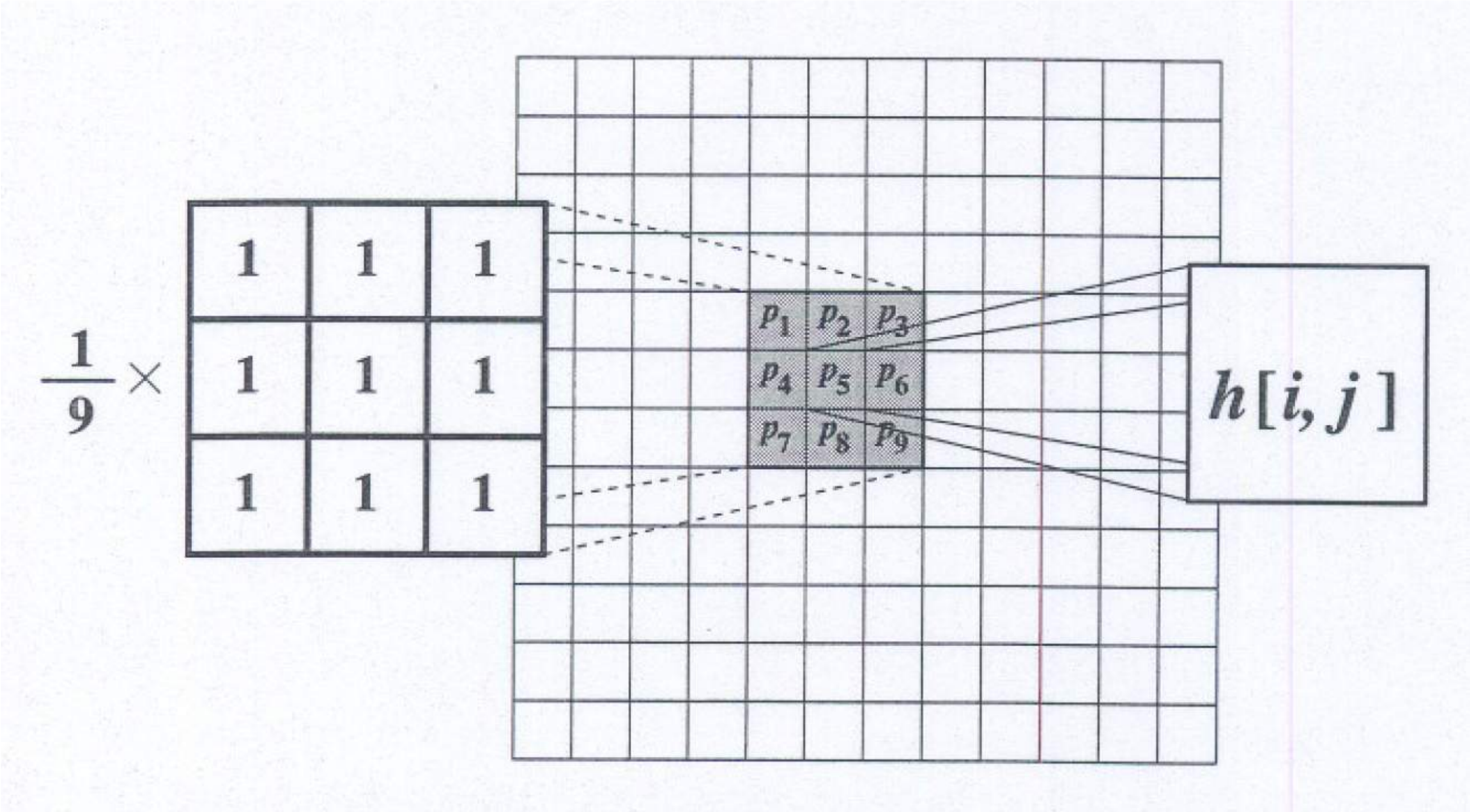


$$h[i,j] = A\,p_1 + B\,p_2 + C\,p_3 + D\,p_4 + E\,p_5 + F\,p_6 + G\,p_7 + H\,p_8 + I\,p_9$$

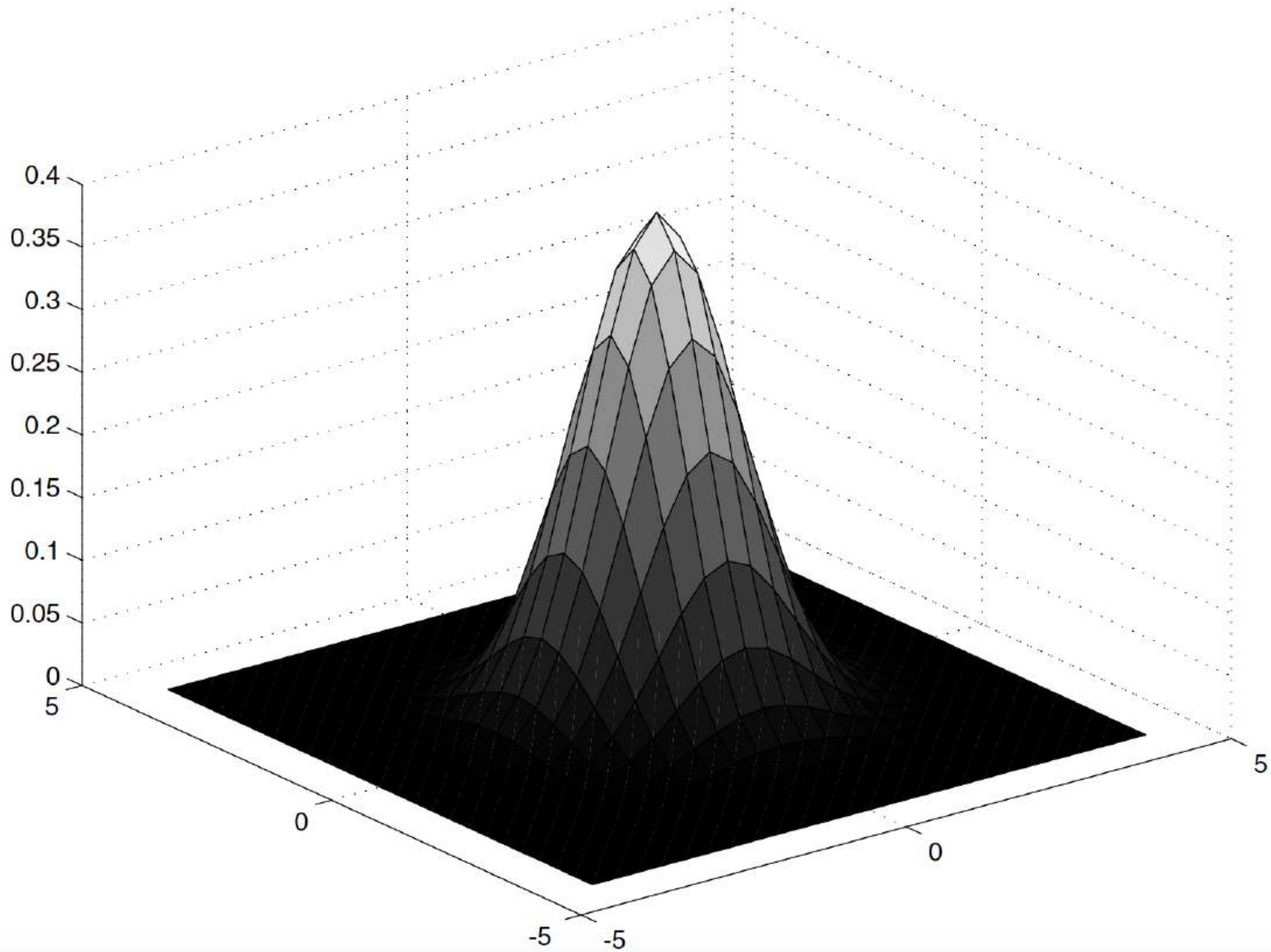
Convolution



Mean Filters



Gaussian Filters



Gaussian Filters

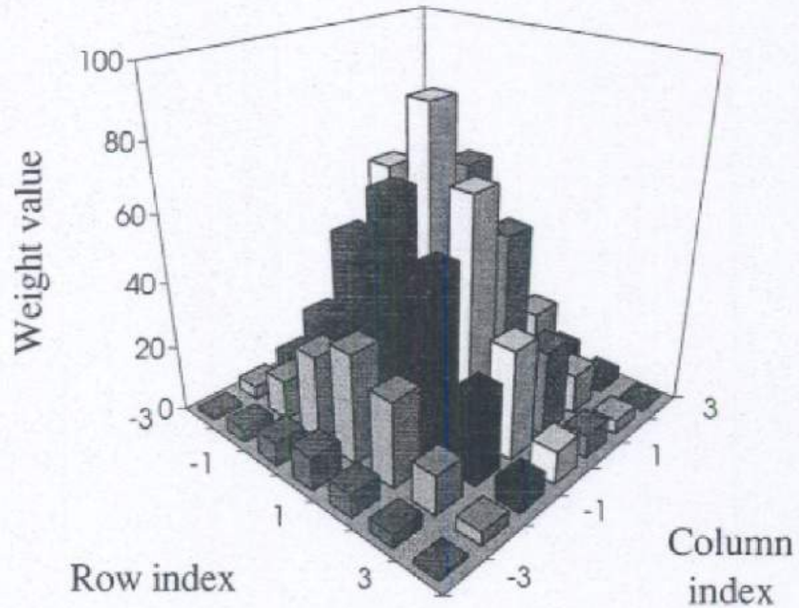
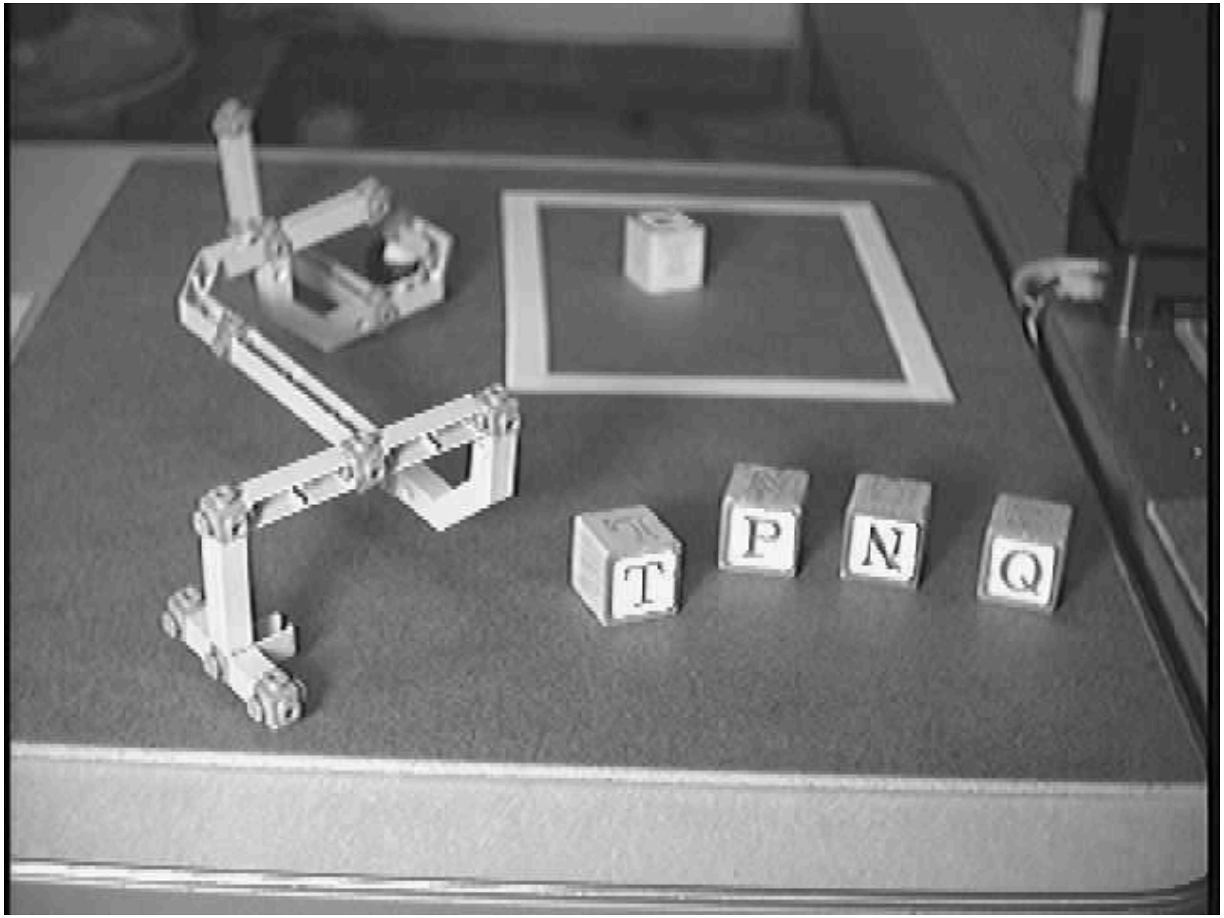


Figure 4.15: A 3-D plot of the 7 × 7 Gaussian mask.

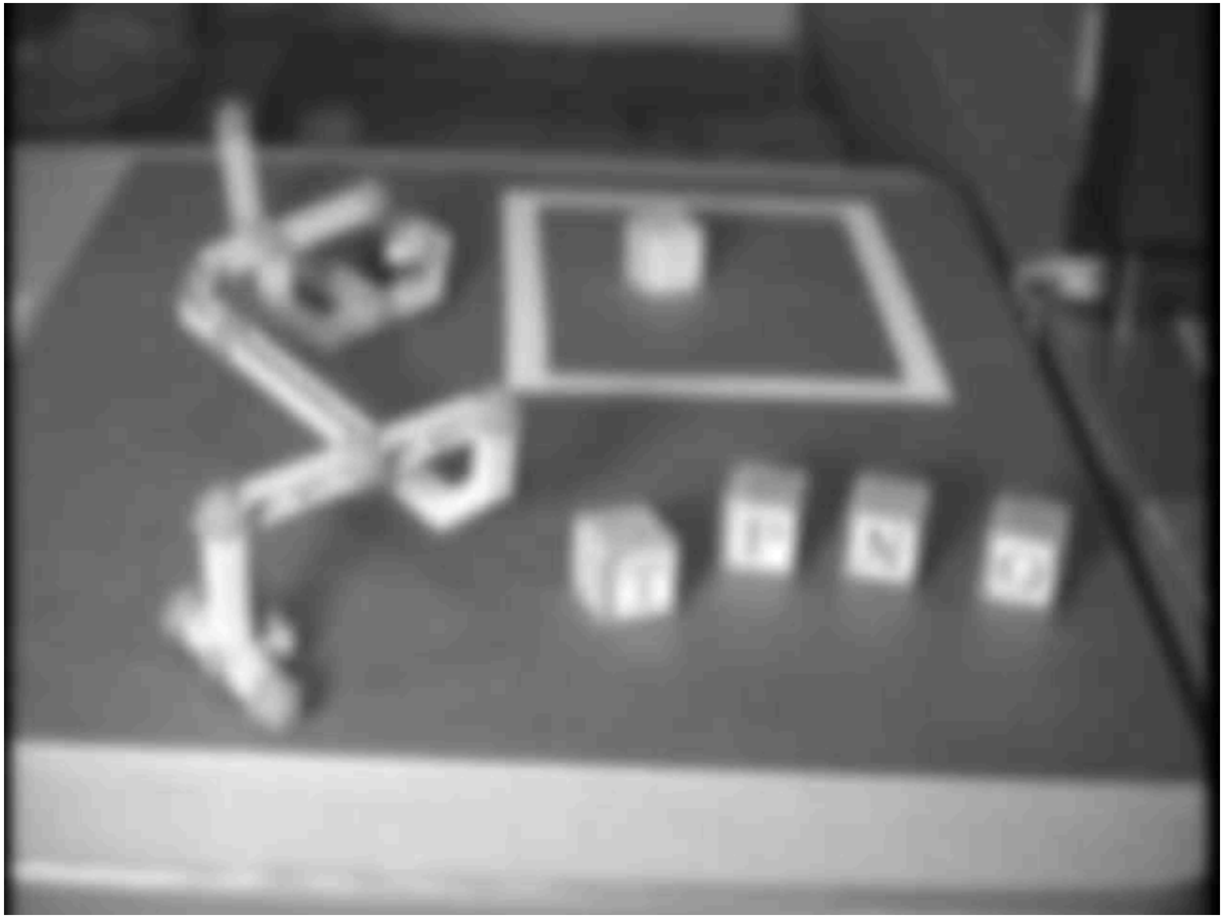
7 × 7 Gaussian mask

1	1	2	2	2	1	1
1	2	2	4	2	2	1
2	2	4	8	4	2	2
2	4	8	16	8	4	2
2	2	4	8	4	2	2
1	2	2	4	2	2	1
1	1	2	2	2	1	1

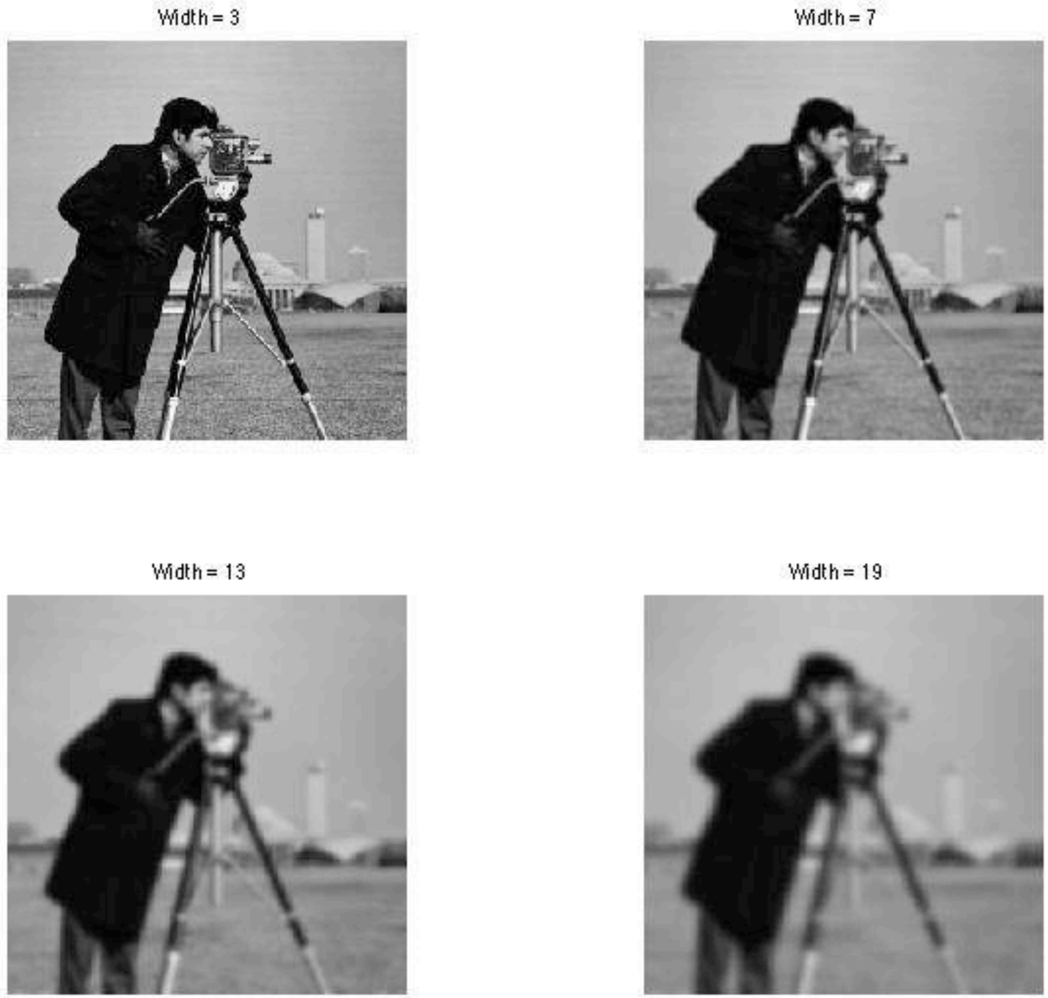
The Effect of Gaussian Filters



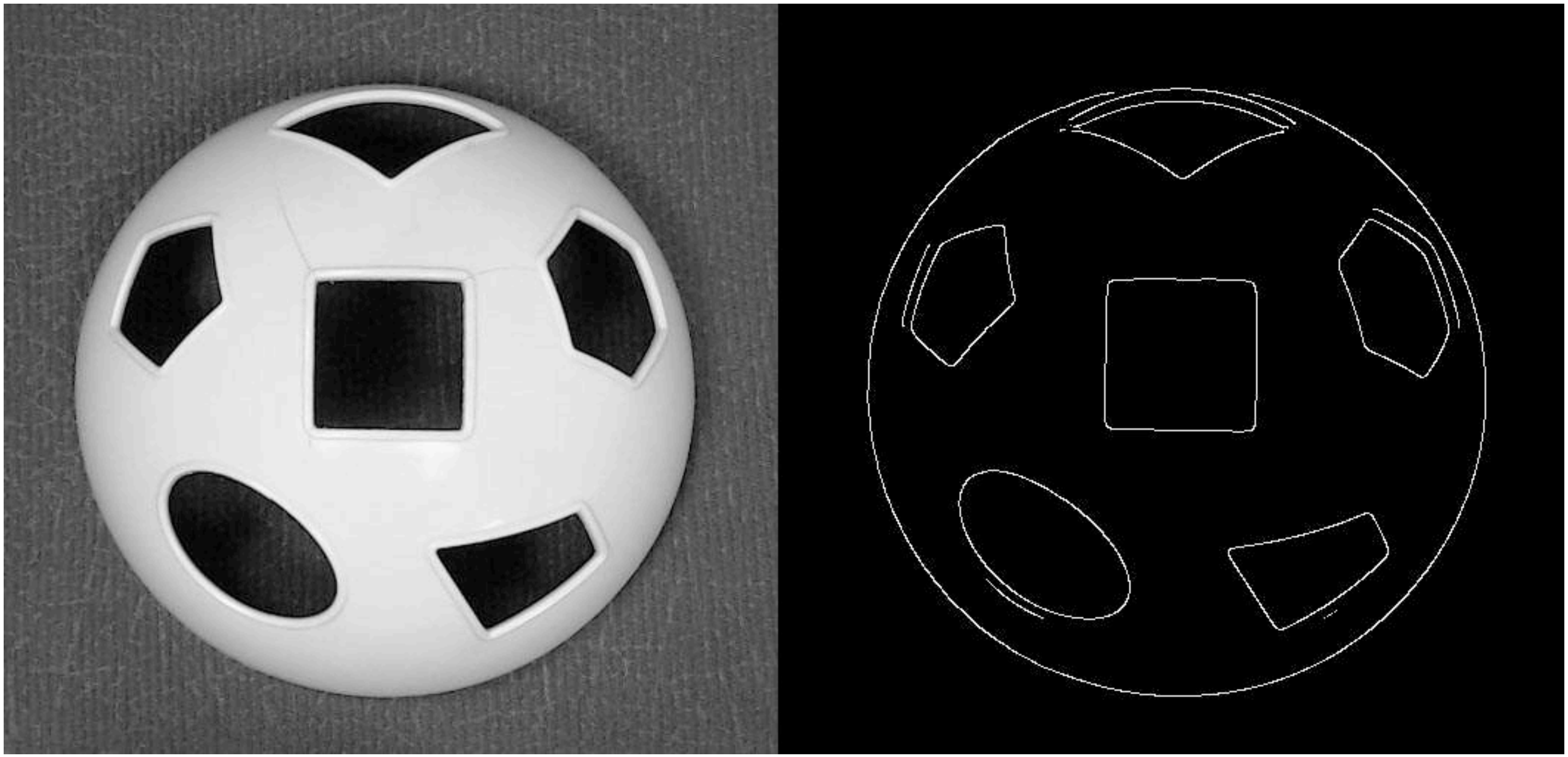
The Effect of Gaussian Filters



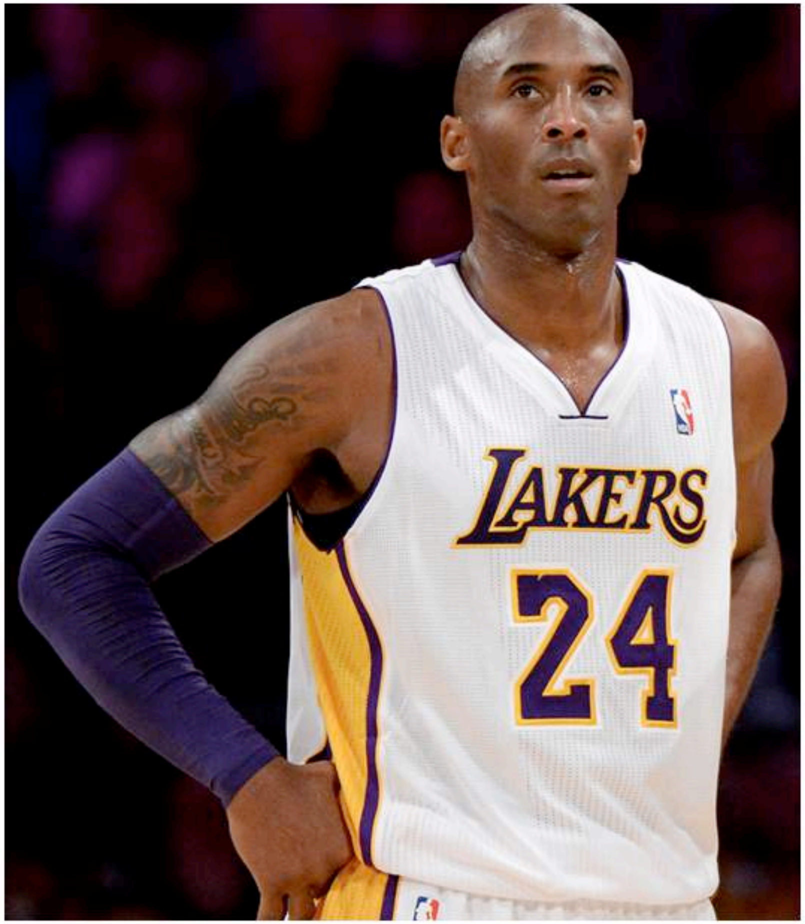
Kernel Width Affects Scale

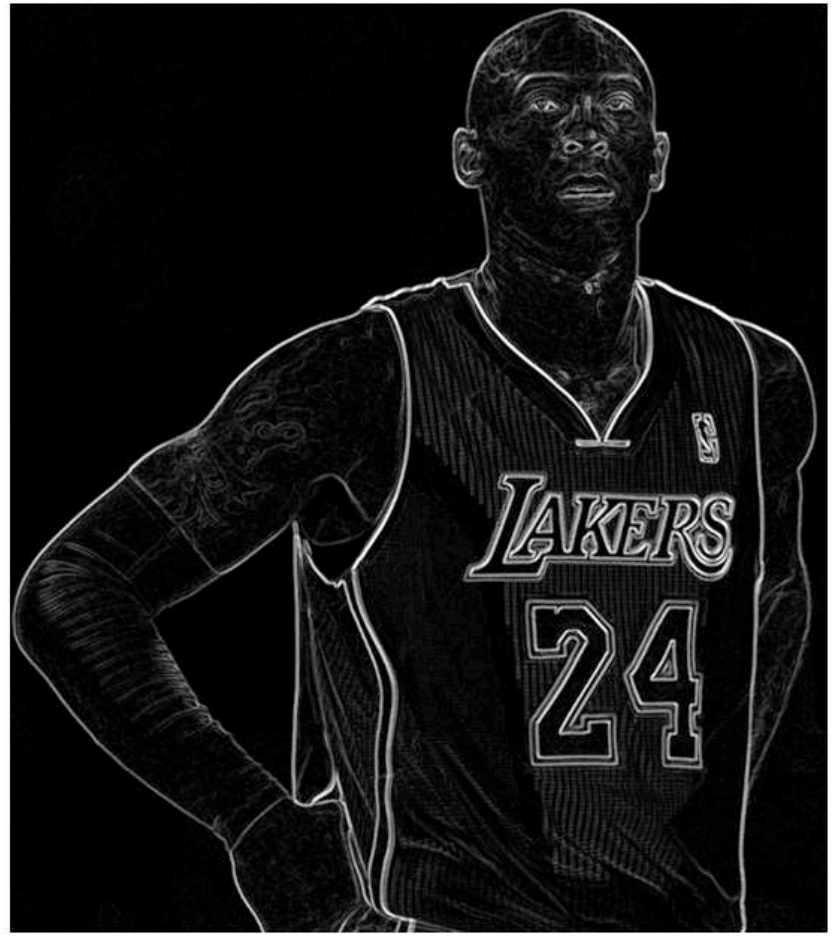


Edge detection



Edge detection





Using Convolution for Edge Detection

Roberts Operator

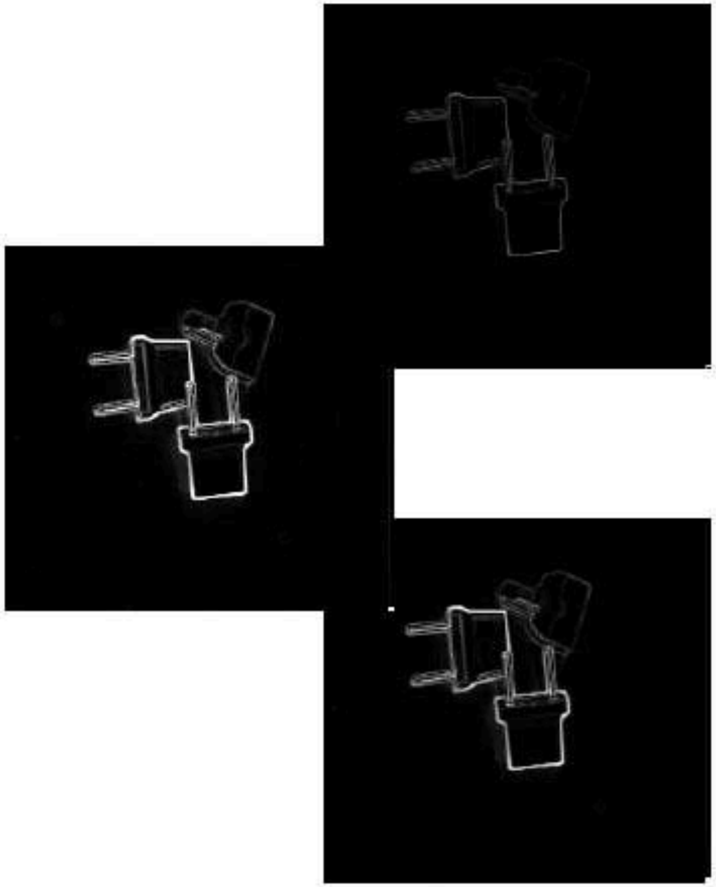
$$G_x \approx \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad G_y \approx \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Sobel Operator

$$G_x \approx \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad G_y \approx \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

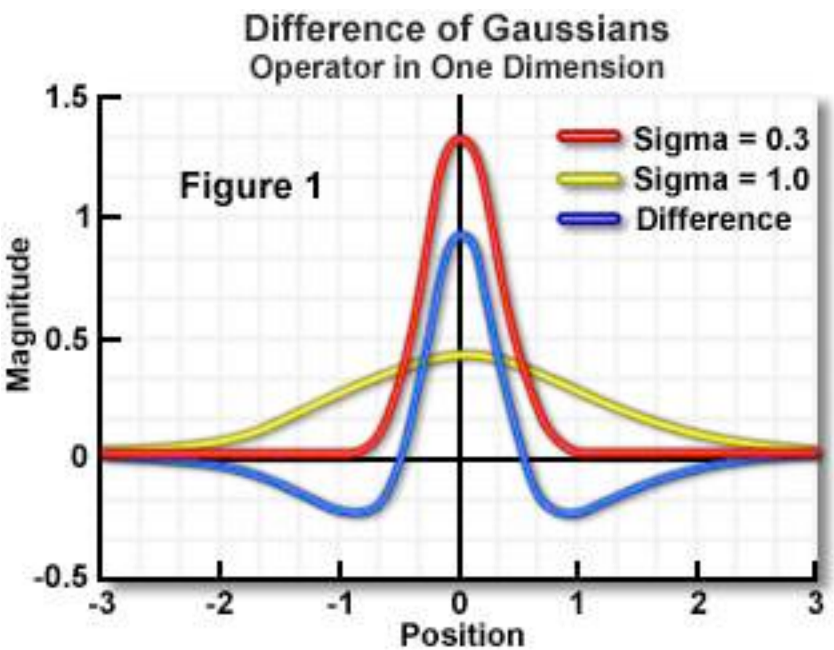
Prewitt Operator

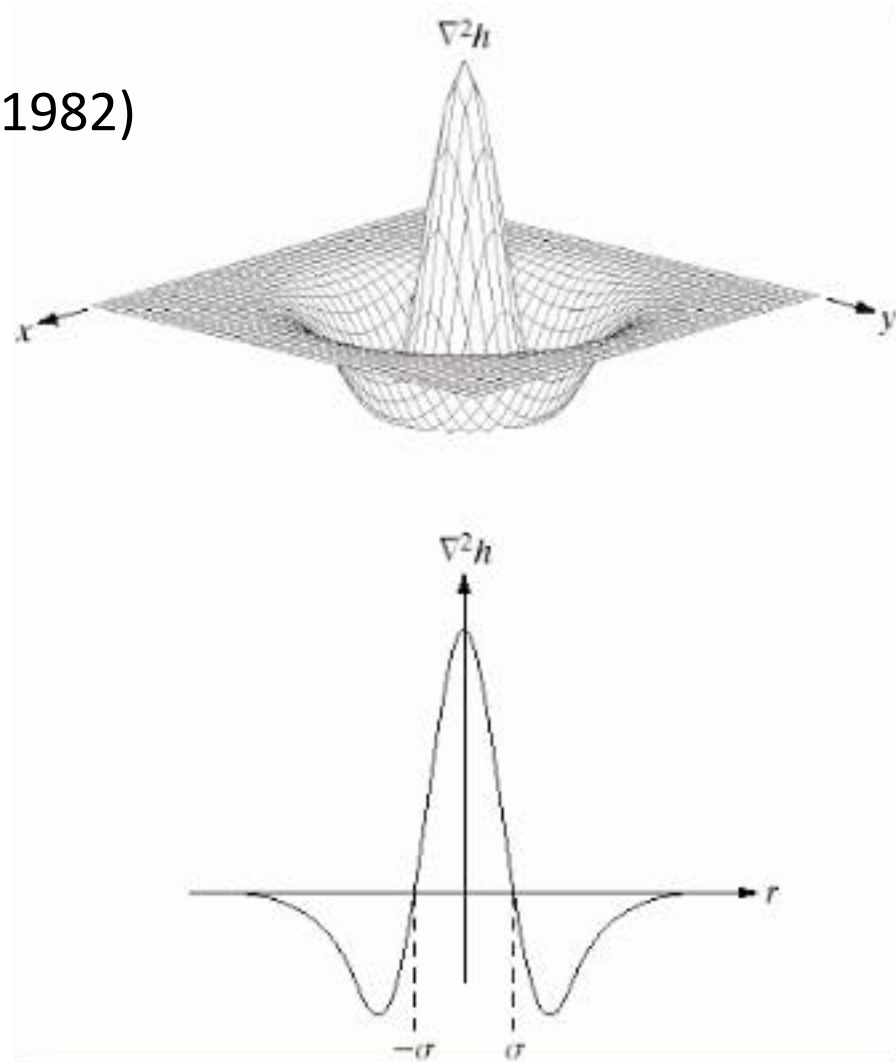
$$G_x \approx \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad G_y \approx \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$



A Variety of Image Filters

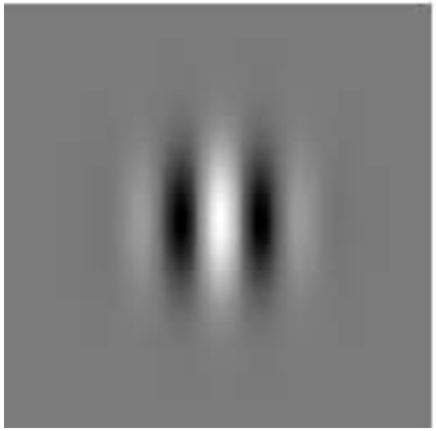
Laplacian of Gaussians (LoG) (Marr 1982)

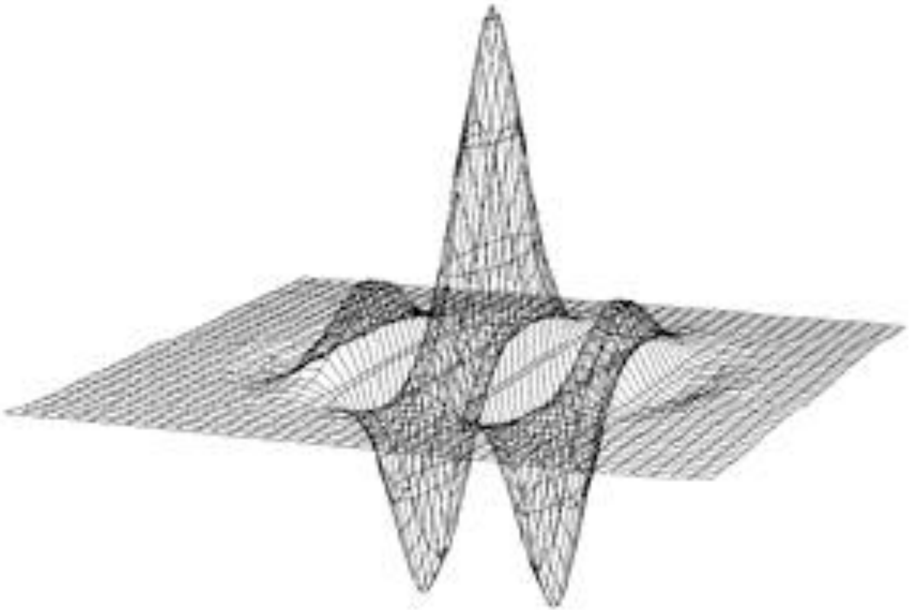




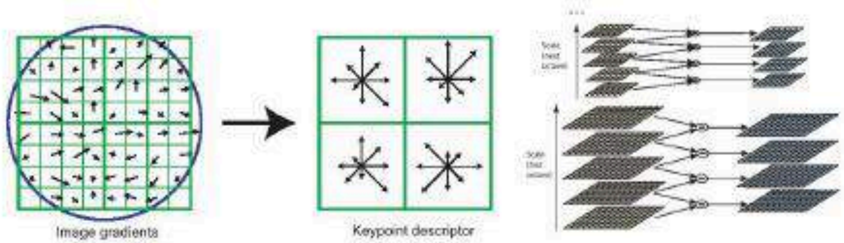
A Variety of Image Filters

Gabor filters (directional) (Daugman 1985)

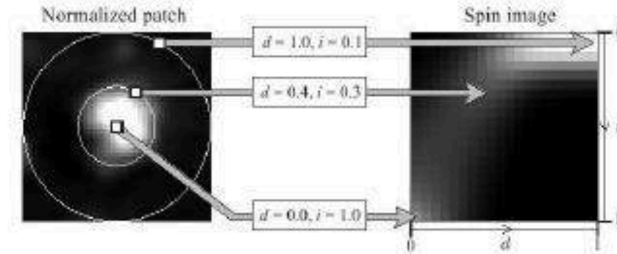




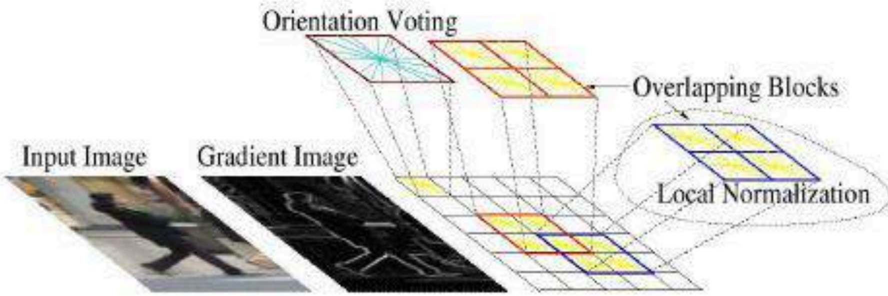
A Variety of Image Filters



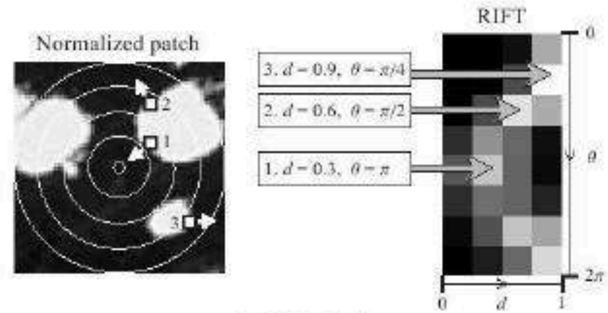
SIFT



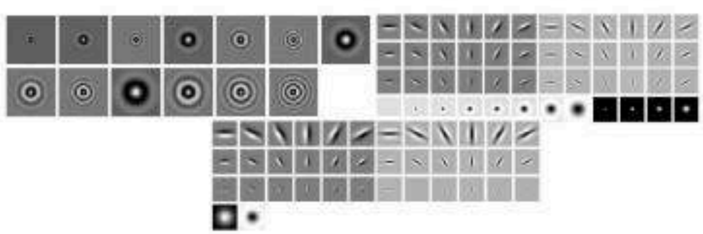
Spin image



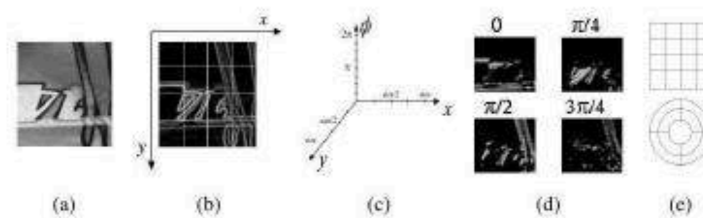
HoG



RIFT



Textons



GLOH

From: M. Sebag

Traditional *vs* Deep Learning Approach

Traditional approach

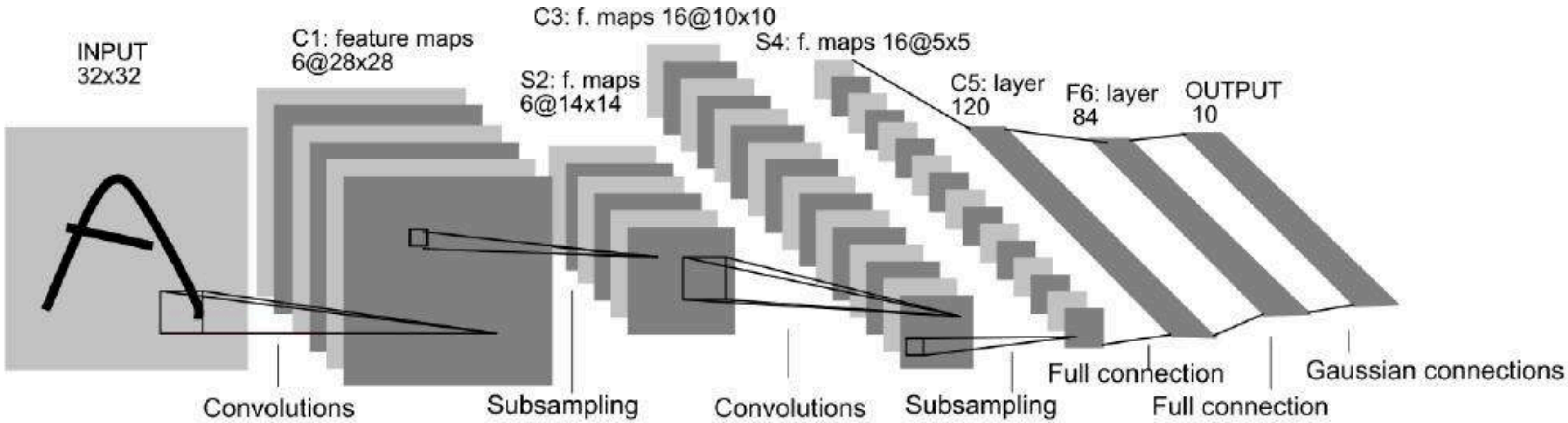
→ Manually crafted features → Trainable classifier

Deep learning

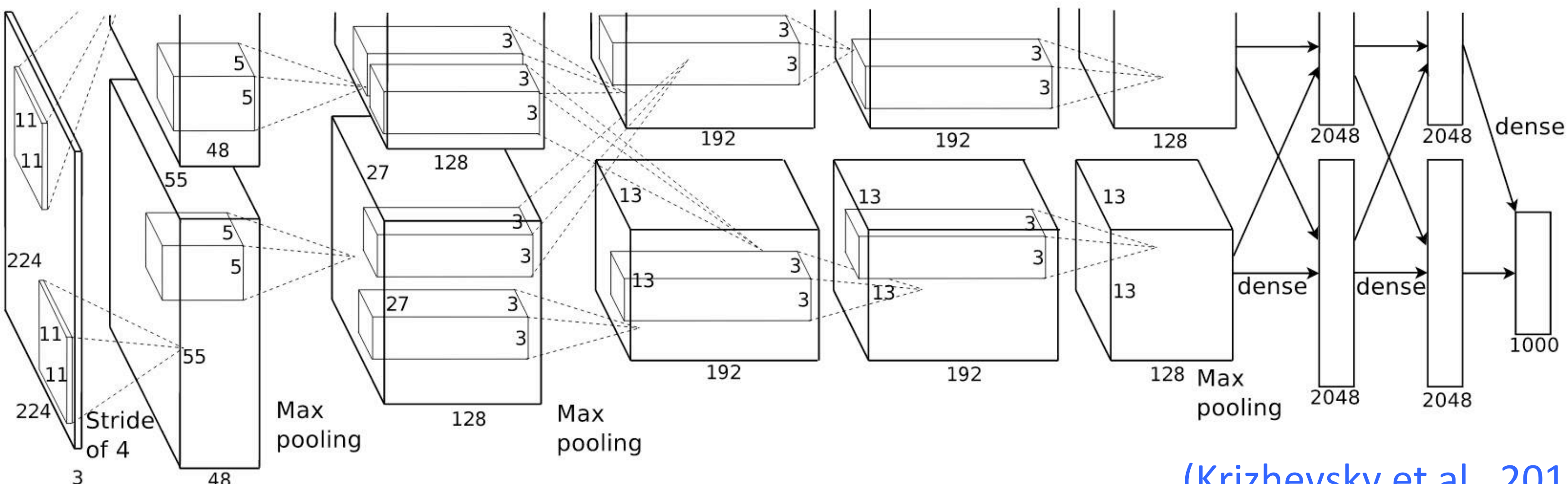
→ Trainable feature extractor → Trainable classifier

From: M. Sebag

Convolutional Neural Networks (CNNs)



(LeCun 1998)



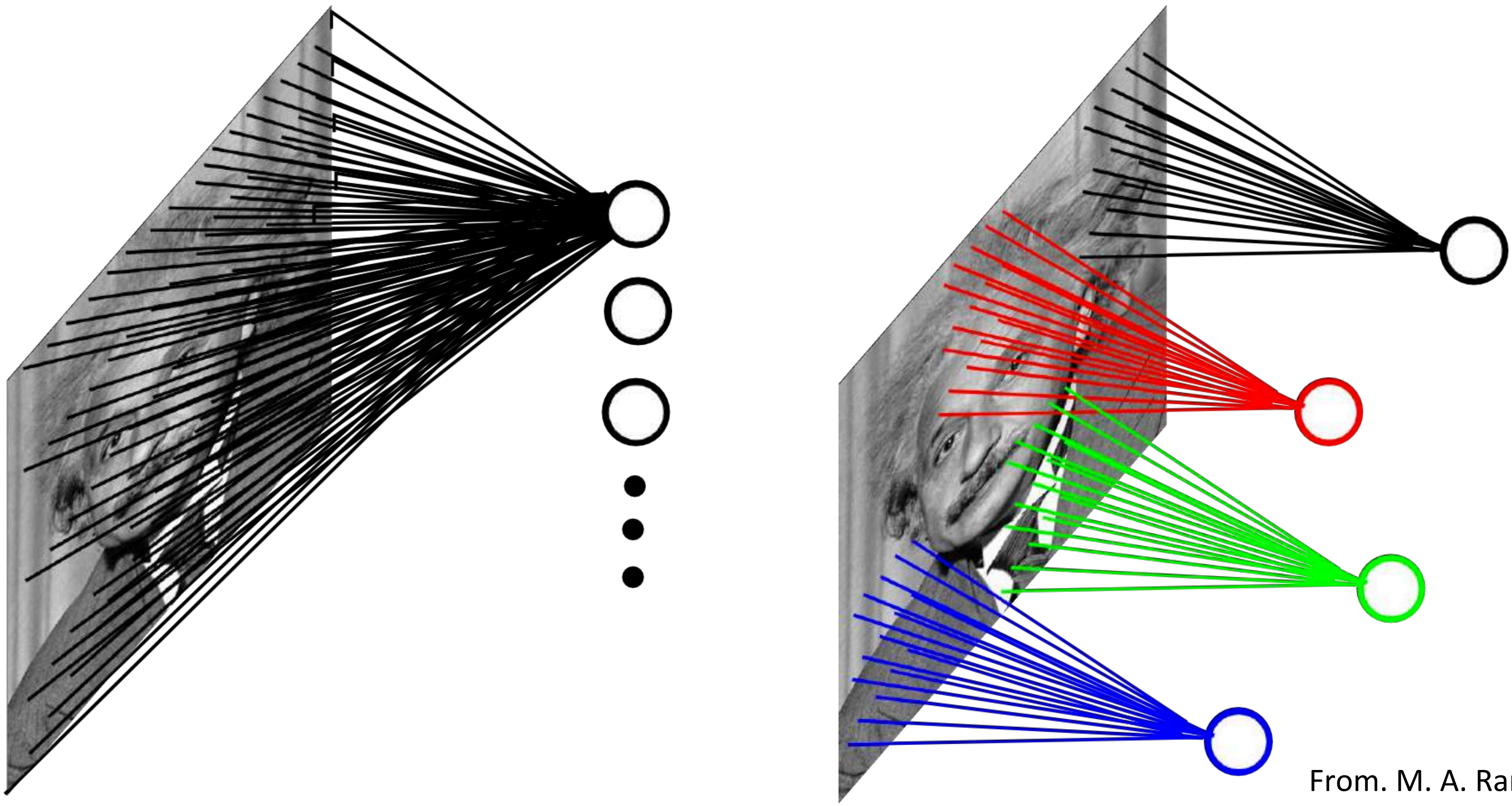
(Krizhevsky et al. 2012)

Fully- *vs* Locally-Connected Networks

Fully-connected: 400,000 hidden units = 16 billion parameters

Locally-connected: 400,000 hidden units 10 x 10 fields = 40 million parameters

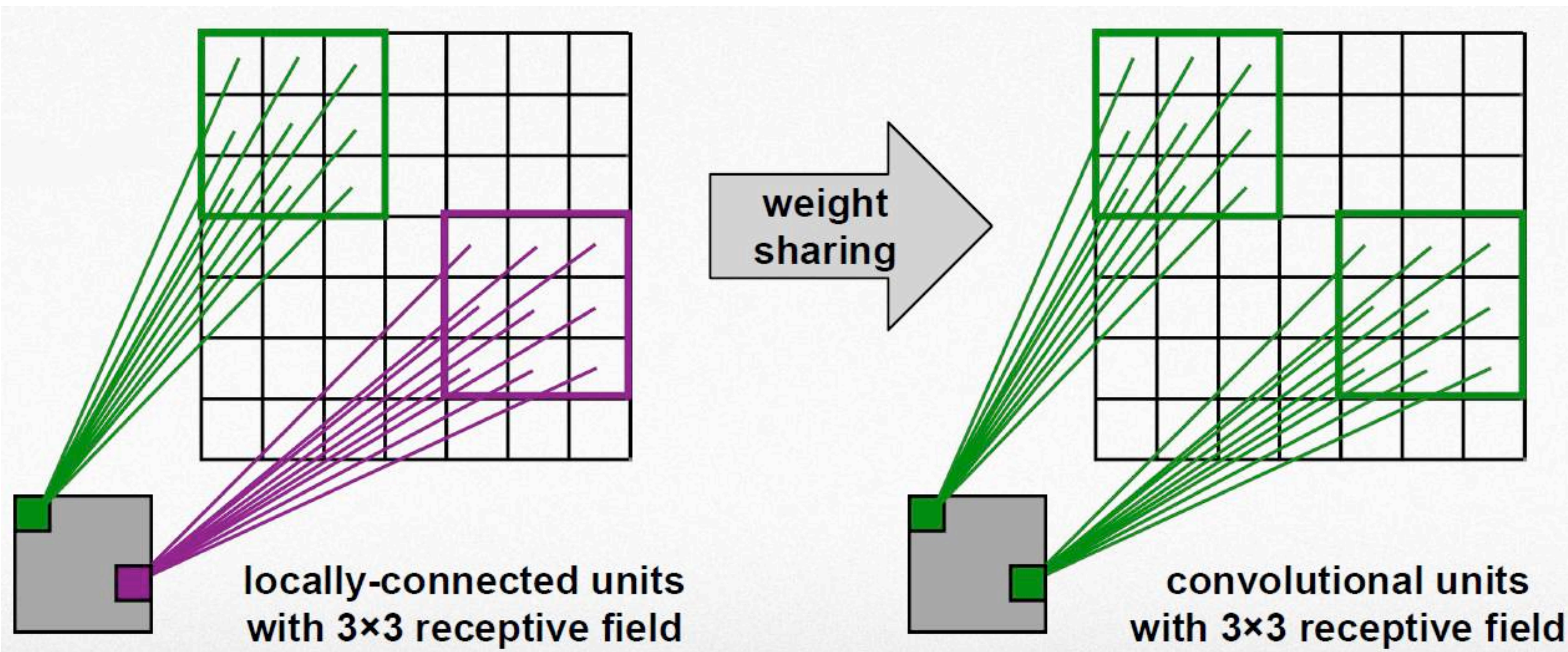
Local connections capture local dependencies



From. M. A. Ranzato

Weight Sharing

We can dramatically reduce the number of parameters by making one reasonable assumption: That if one feature is useful to compute at some spatial position (x1,y1), then it should also be useful to compute at a different position (x2,y2).



Convolutional Neural Networks (CNN, ConvNet, DCN)

- CNN = a multi-layer neural network with
  - **Local** connectivity
  - **Share** weight parameters across spatial positions

- One activation map (a depth slice), computed with one set of weights

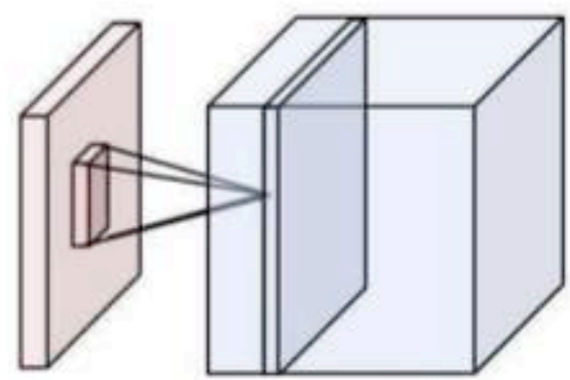
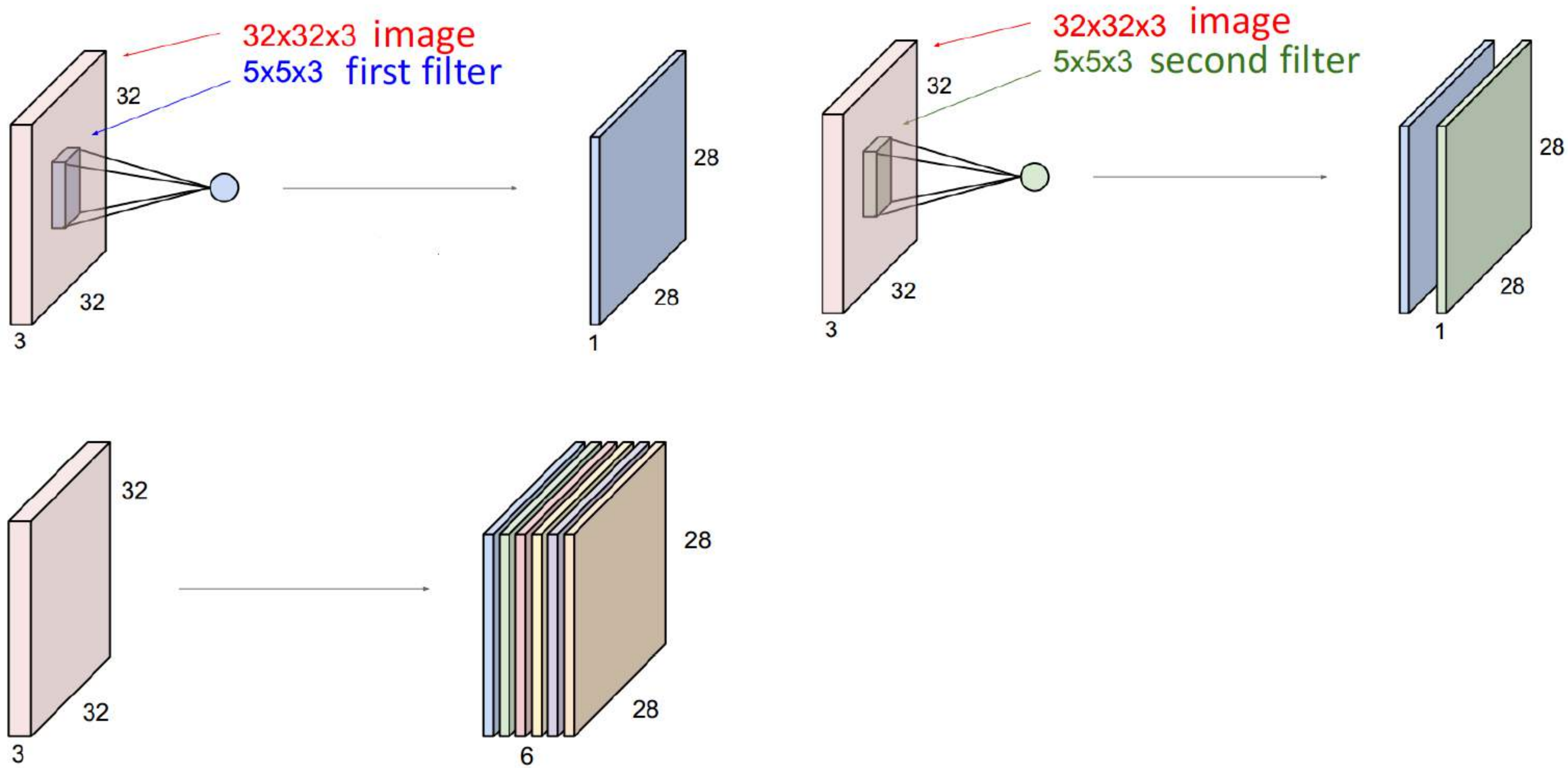


Image credit: A. Karpathy

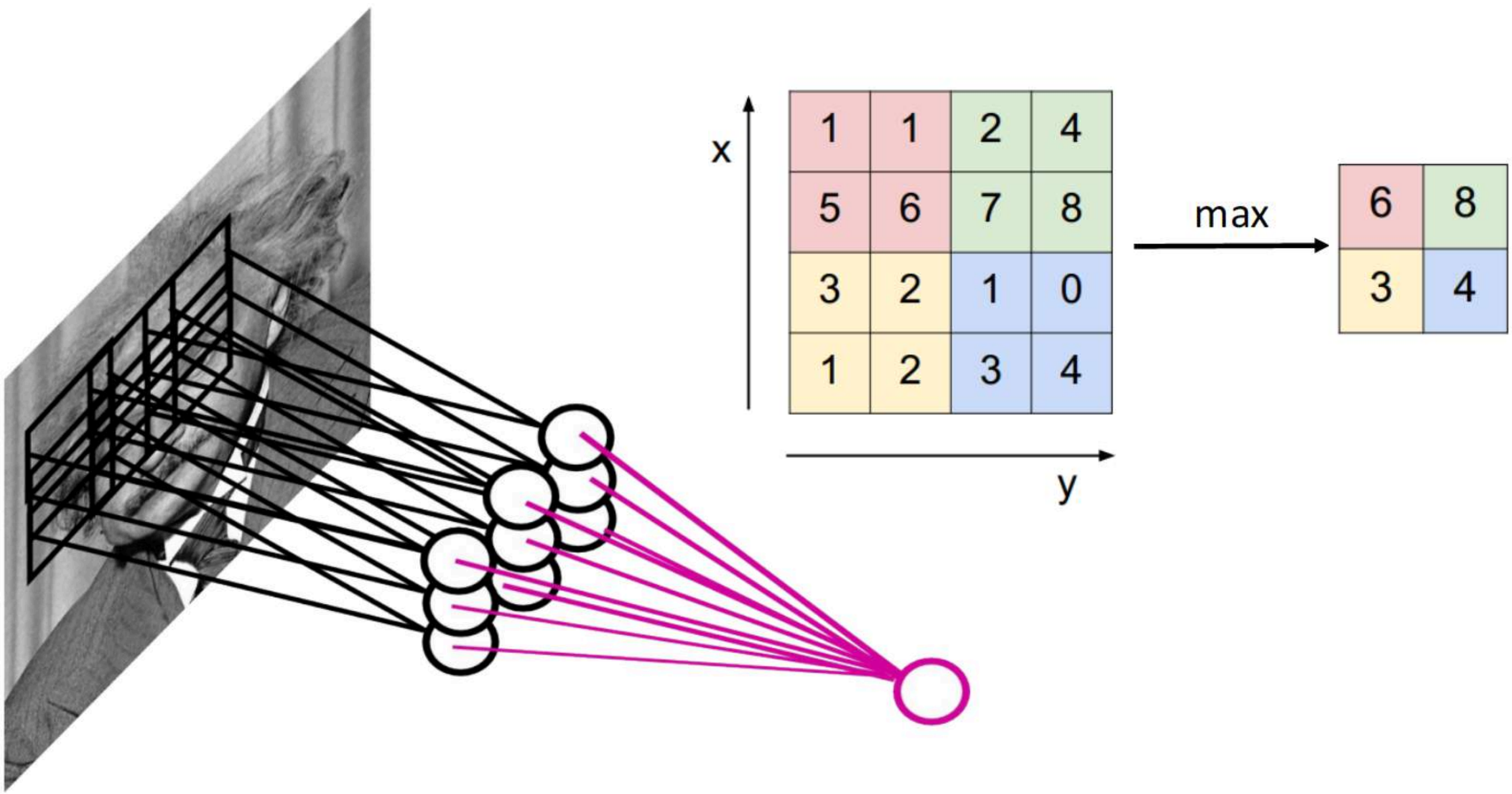
Using Several Trainable Filters

Normally, several filters are packed together and learnt automatically during training

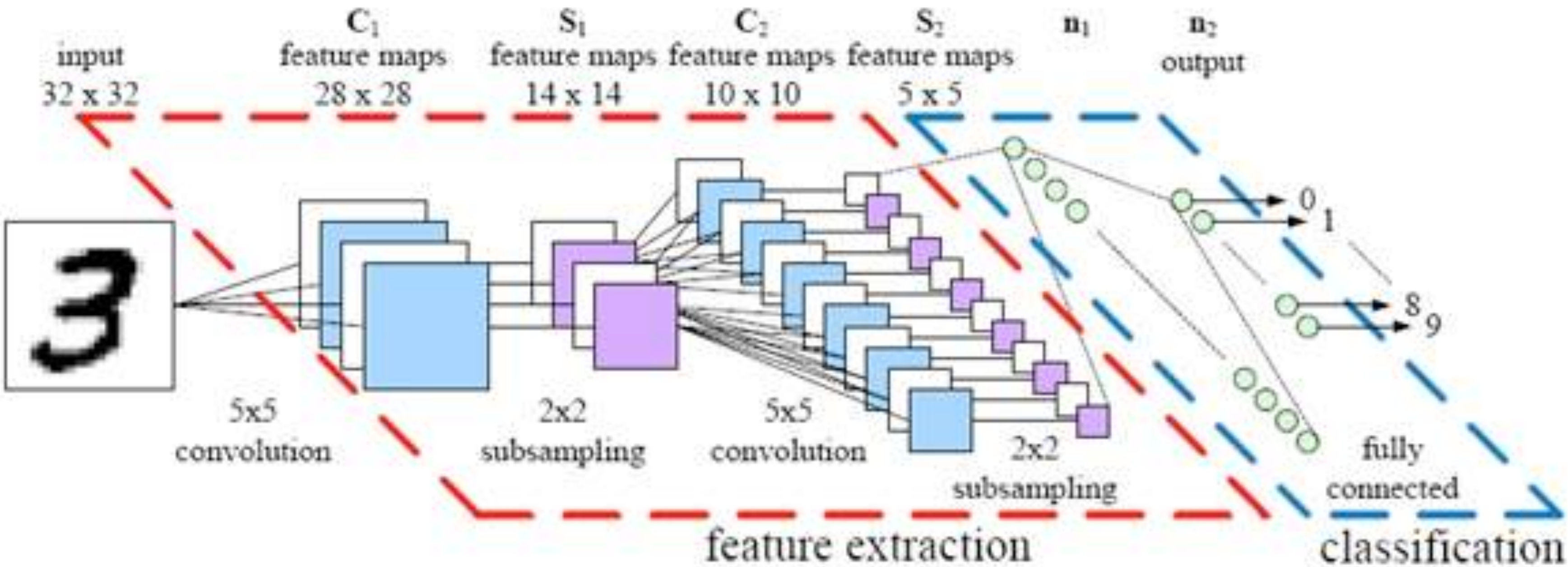


Pooling

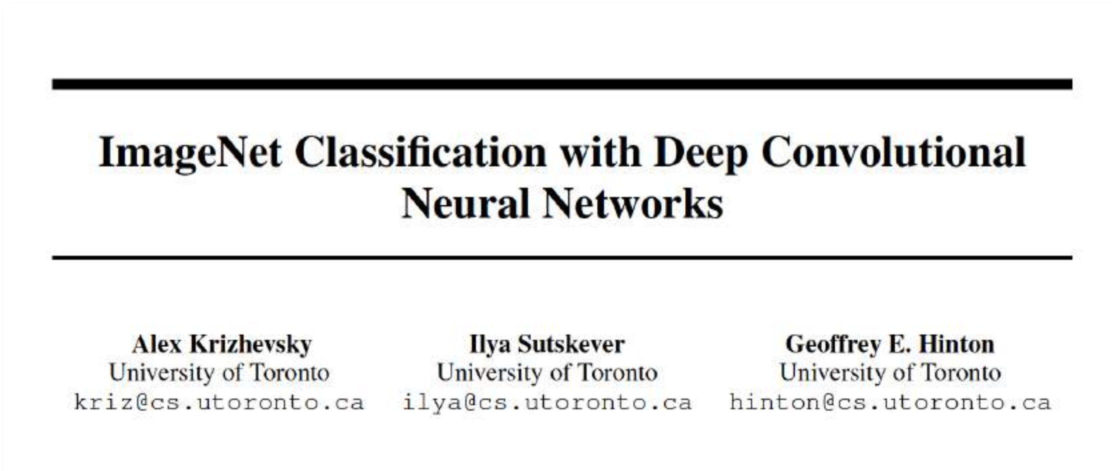
Max pooling is a way to simplify the network architecture, by downsampling the number of neurons resulting from filtering operations.



Combining Feature Extraction and Classification



AlexNet (2012)



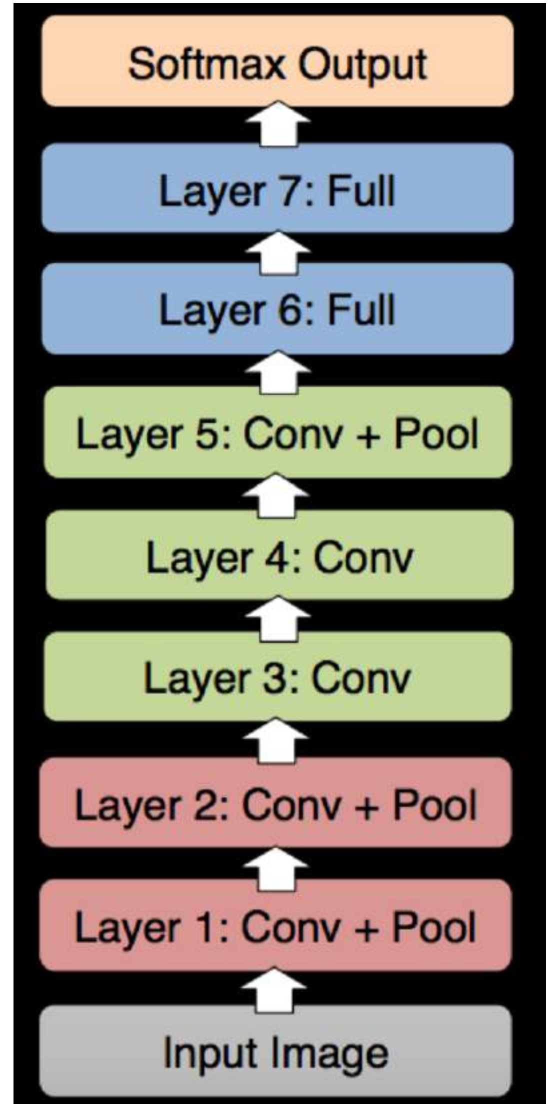
ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

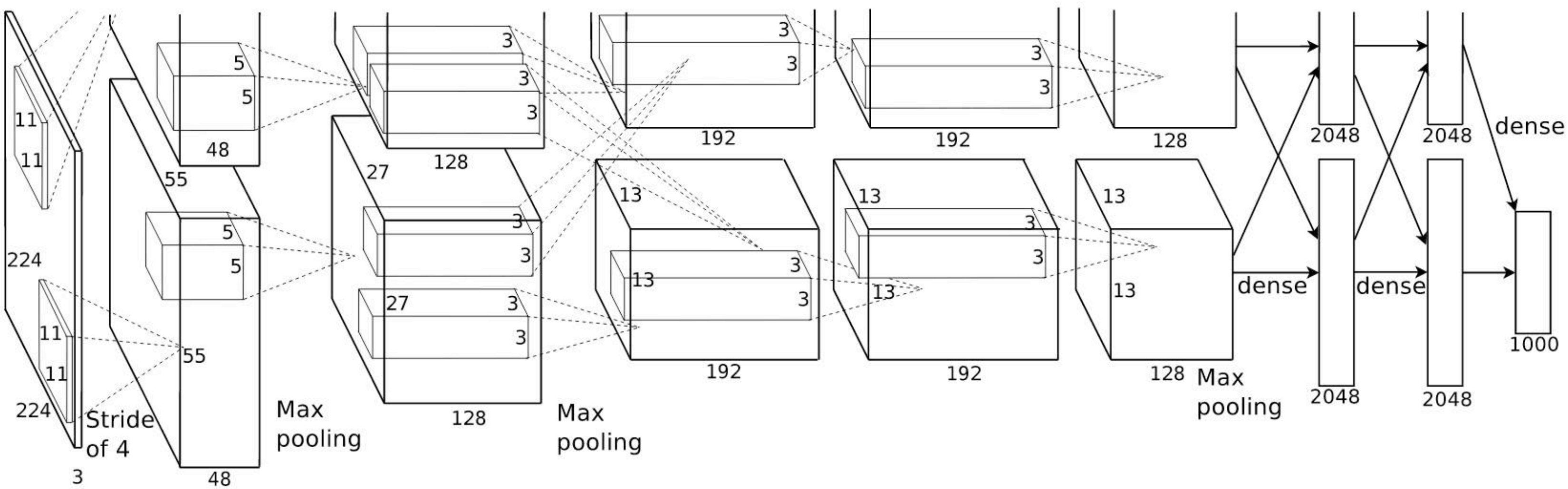
- 8 layers total
- Trained on Imagenet Dataset (1000 categories, 1.2M training images, 150k test images)



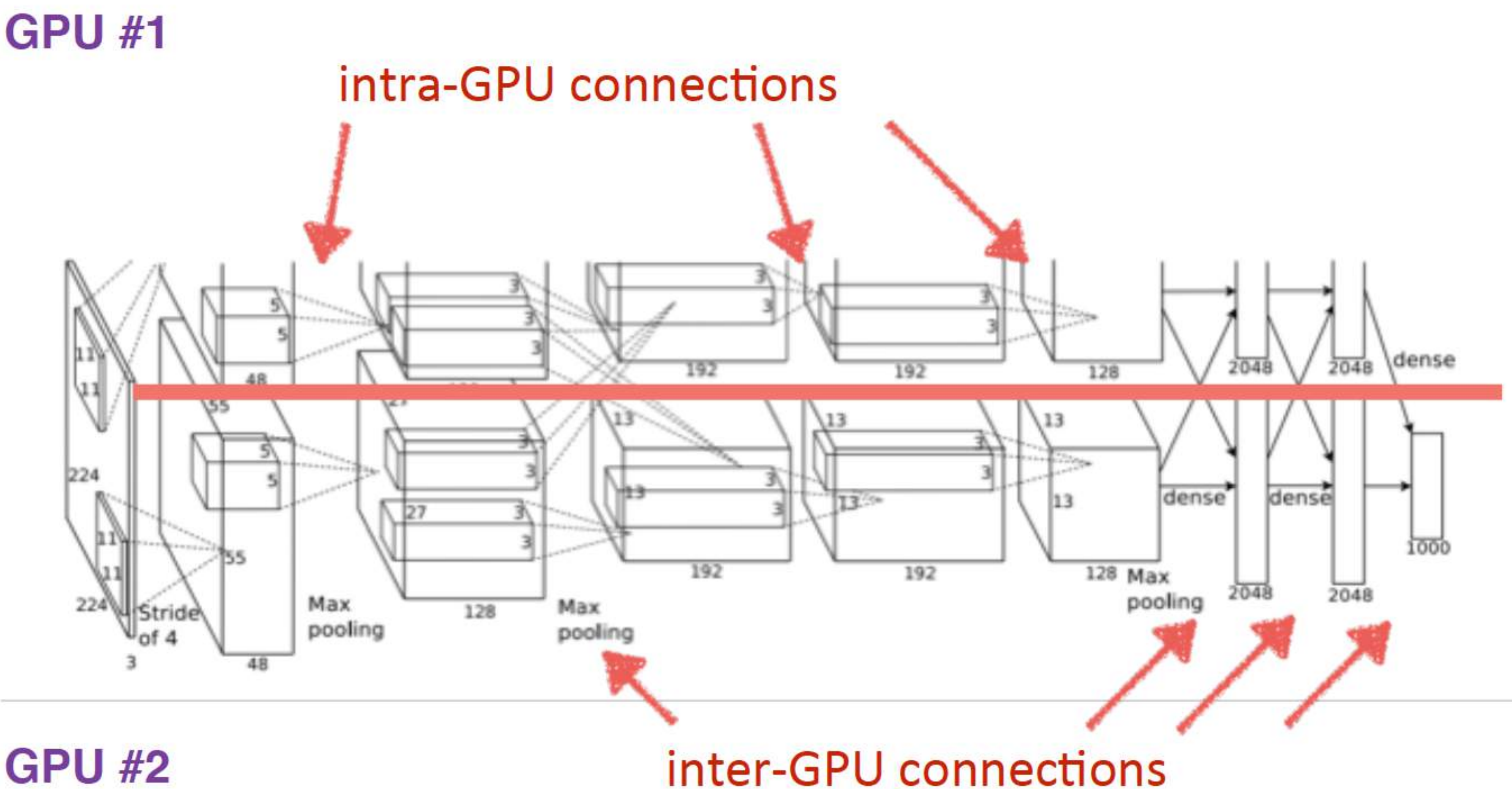
AlexNet Architecture

- 1st layer: 96 kernels (11 x 11 x 3)
- Normalized, pooled
- 2nd layer: 256 kernels (5 x 5 x 48)
- Normalized, pooled
- 3rd layer: 384 kernels (3 x 3 x 256)
- 4th layer: 384 kernels (3 x 3 x 192)
- 5th layer: 256 kernels (3 x 3 x 192)
- Followed by 2 fully connected layers, 4096 neurons each
- Followed by a 1000-way SoftMax layer

650,000 neurons
60 million parameters

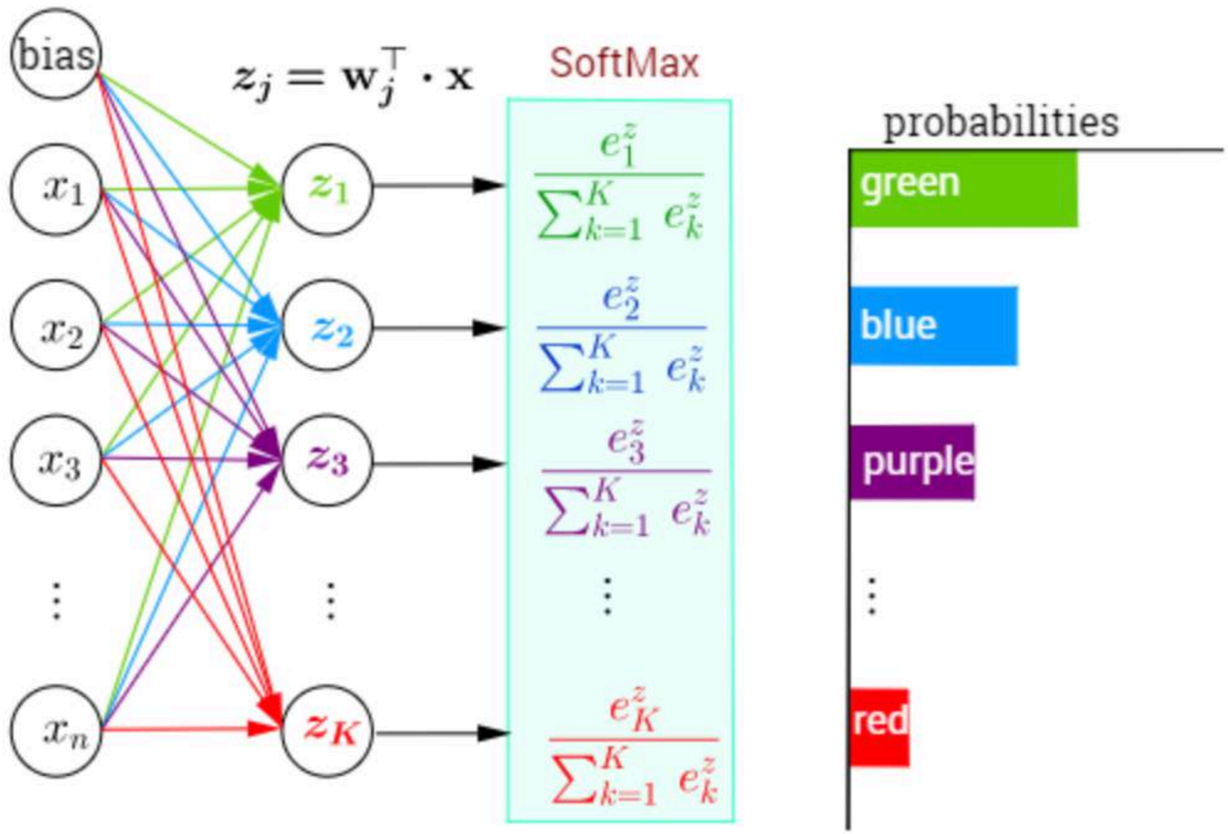


Training on Multiple GPU's



Output Layer: Softmax

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_K \end{bmatrix} = \begin{bmatrix} \mathbf{w}_1^\top \\ \mathbf{w}_2^\top \\ \mathbf{w}_3^\top \\ \vdots \\ \mathbf{w}_K^\top \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

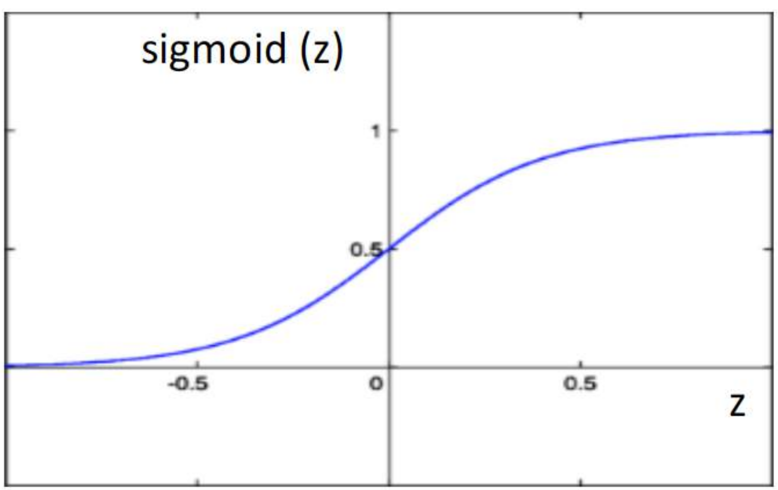


Rectified Linear Units (ReLU's)

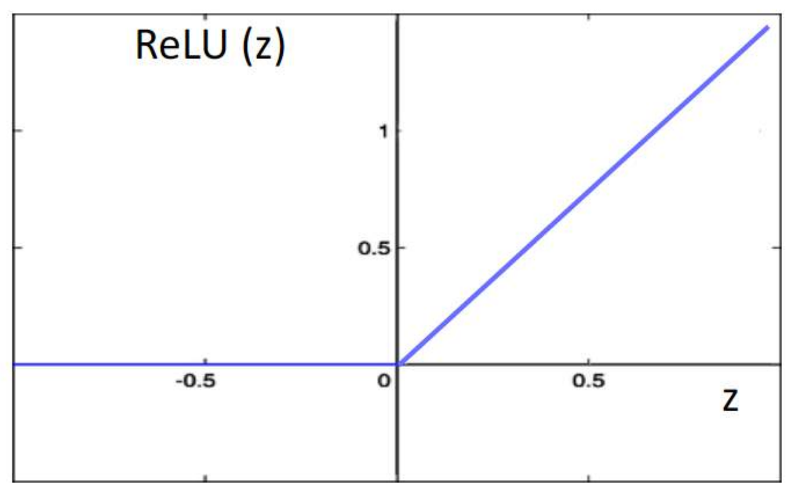
Problem: Sigmoid activation takes on values in (0,1). Propagating the gradient back to the initial layers, it tends to become 0 (vanishing gradient problem).

From a practical perspective, this slows down the training procedure of the initial layers of the network.

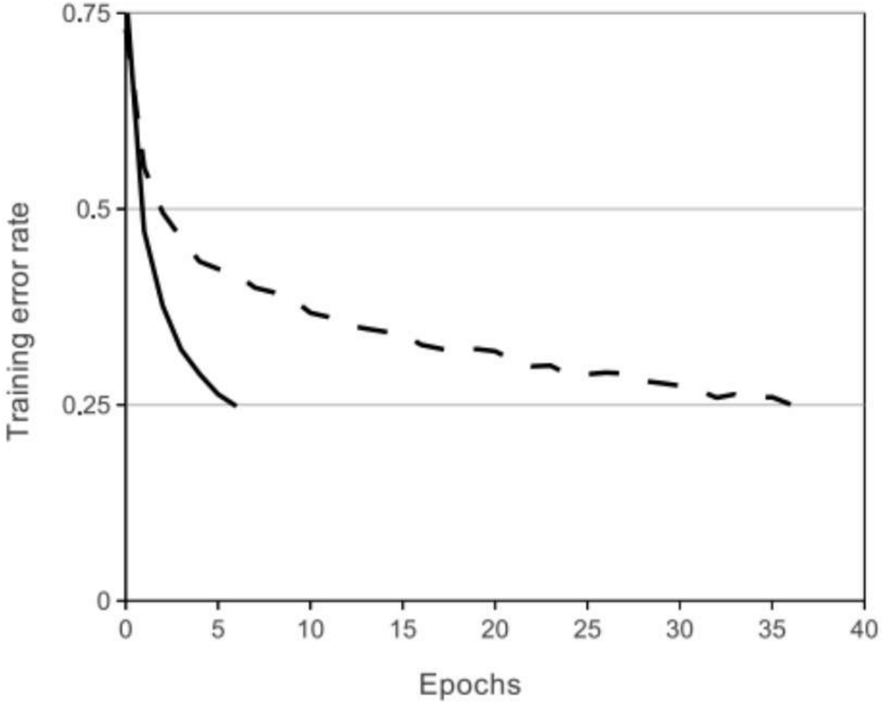
sigmoid(z) = $\frac{1}{1+e^{-z}}$



ReLU(z) = $\max(0, z)$



Rectified Linear Units (ReLU's)



A 4 layer CNN with ReLUs (solid line) converges six times faster than an equivalent network with tanh neurons (dashed line) on CIFAR-10 dataset

Mini-batch Stochastic Gradient Descent

Loop:

1. **Sample** a batch of data
2. **Forward** prop it through the graph, get loss
3. **Backprop** to calculate the gradients
4. **Update** the parameters using the gradient

Data Augmentation

The easiest and most common method to reduce overfitting on image data is to artificially enlarge the dataset using label-preserving transformations

AlexNet uses two forms of this **data augmentation**.

- The first form consists of generating image translations and horizontal reflections.
- The second form consists of altering the intensities of the RGB channels in training images.

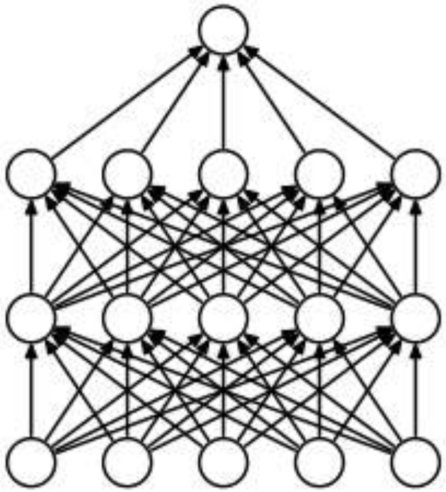
Dropout

Set to zero the output of each hidden neuron with probability 0.5.

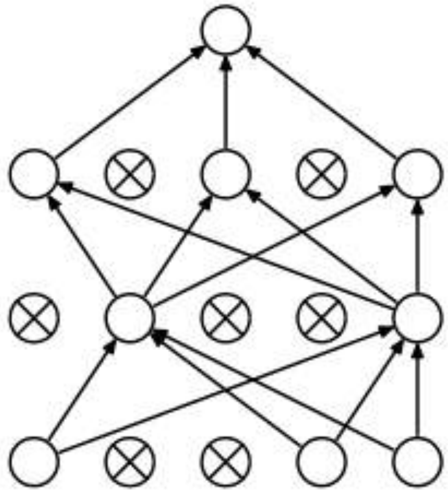
The neurons which are "dropped out" in this way do not contribute to the forward pass and do not participate in backpropagation.

So every time an input is presented, the neural network samples a different architecture, but all these architectures share weights.

Reduces complex co-adaptations of neurons, since a neuron cannot rely on the presence of particular other neurons.



Standard Neural Net



After applying dropout.

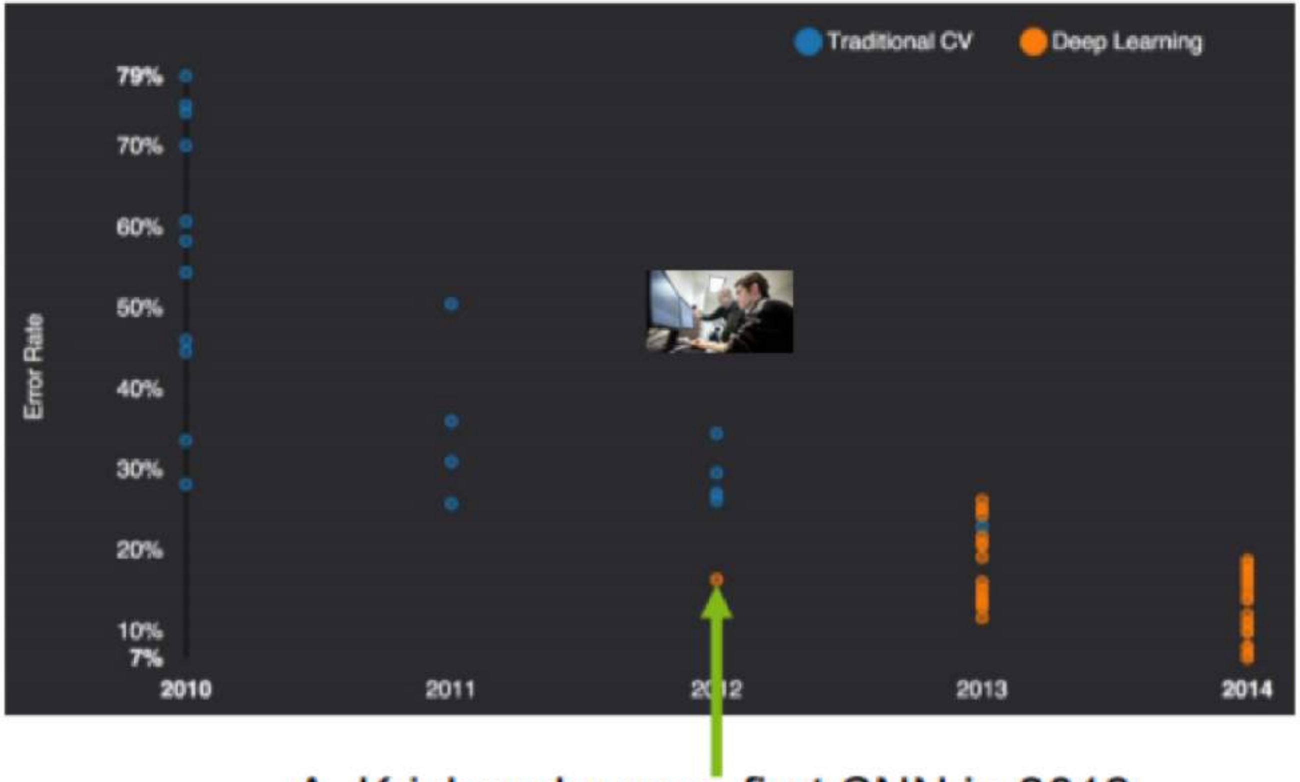
ImageNet

IMAGENET

[Deng et al. CVPR 2009]

- ~14 million labeled images, 20k classes
- Images gathered from Internet
- Human labels via Amazon Turk
- Challenge: 1.2 million training images, 1000 classes

ImageNet Challenges

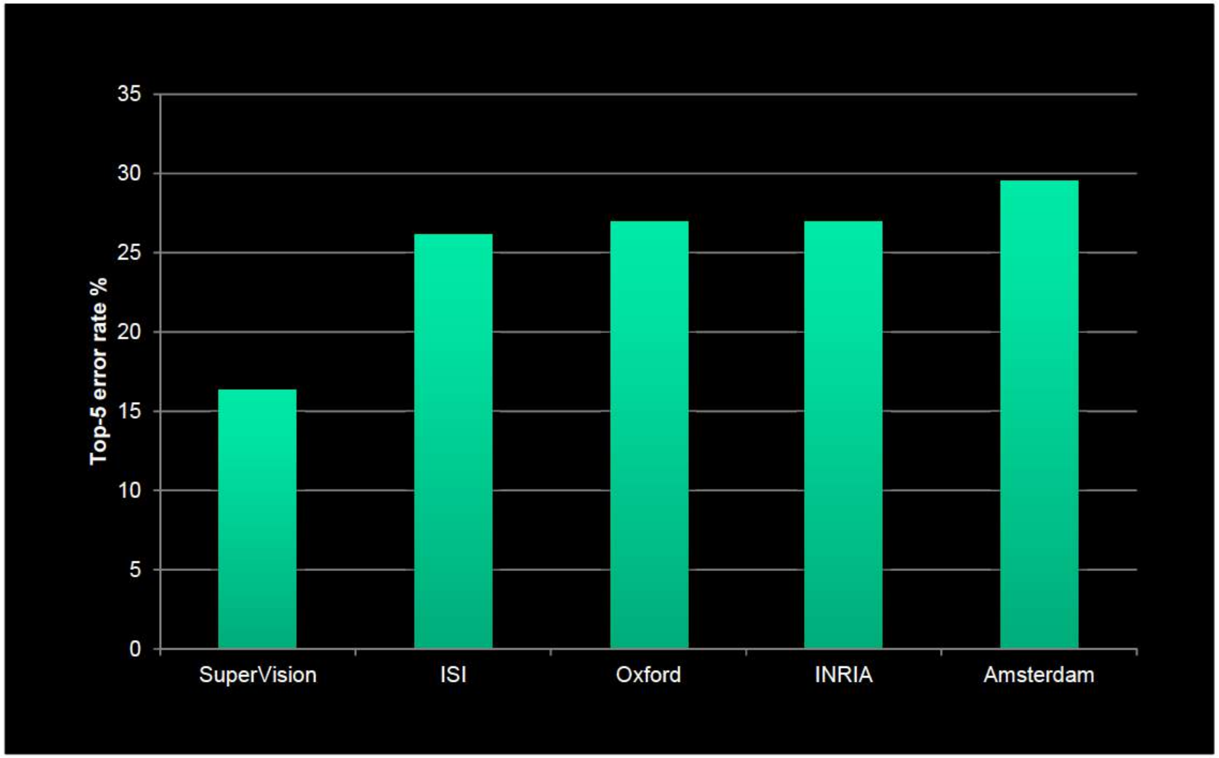


A. Krizhevsky uses first CNN in 2012.
Trained on Gaming Graphic Cards

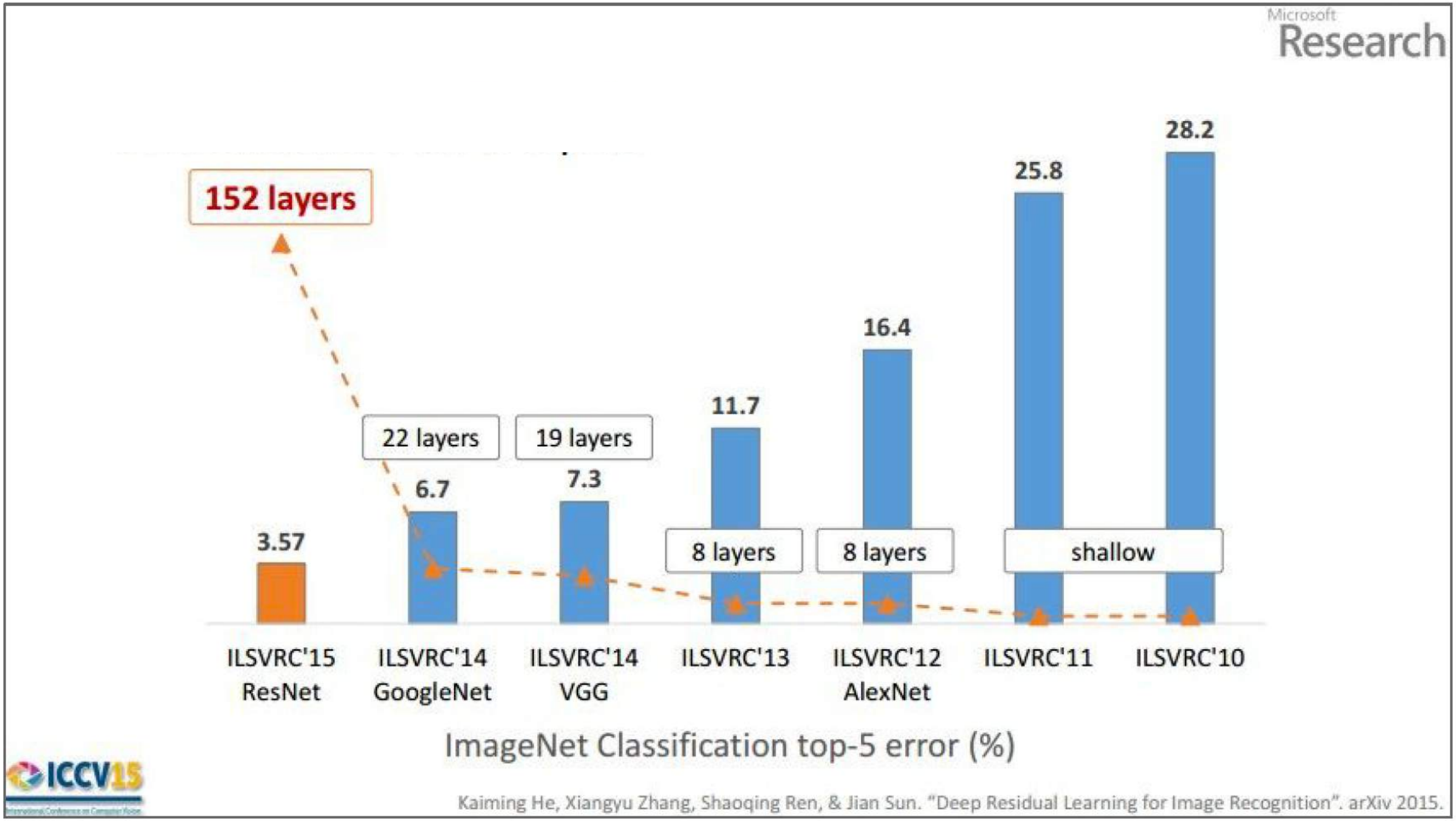
ImageNet Challenge 2012

Krizhevsky et al. -- **16.4% error** (top-5)

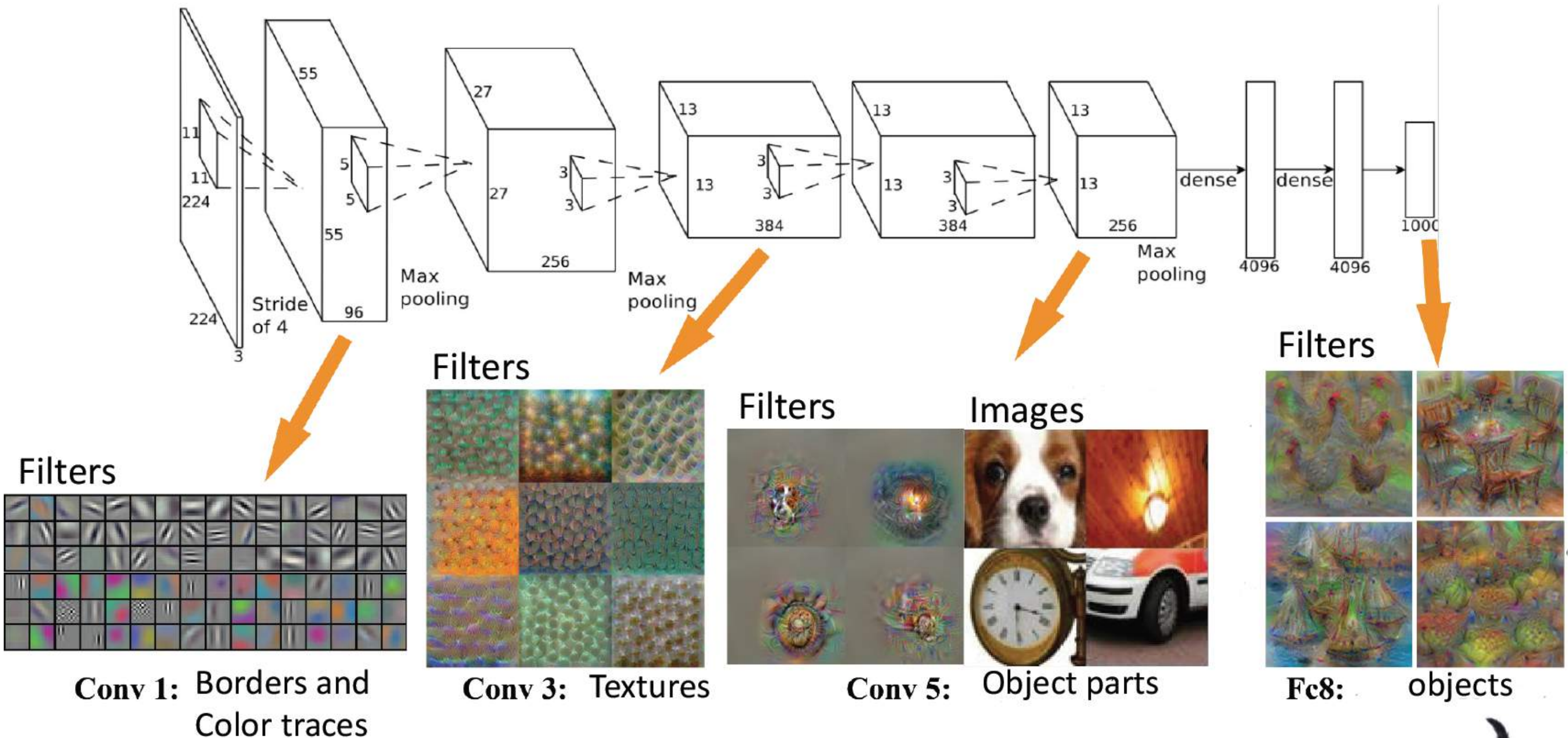
Next best (non-convnet) – **26.2% error**



Revolution of Depth



A Hierarchy of Features

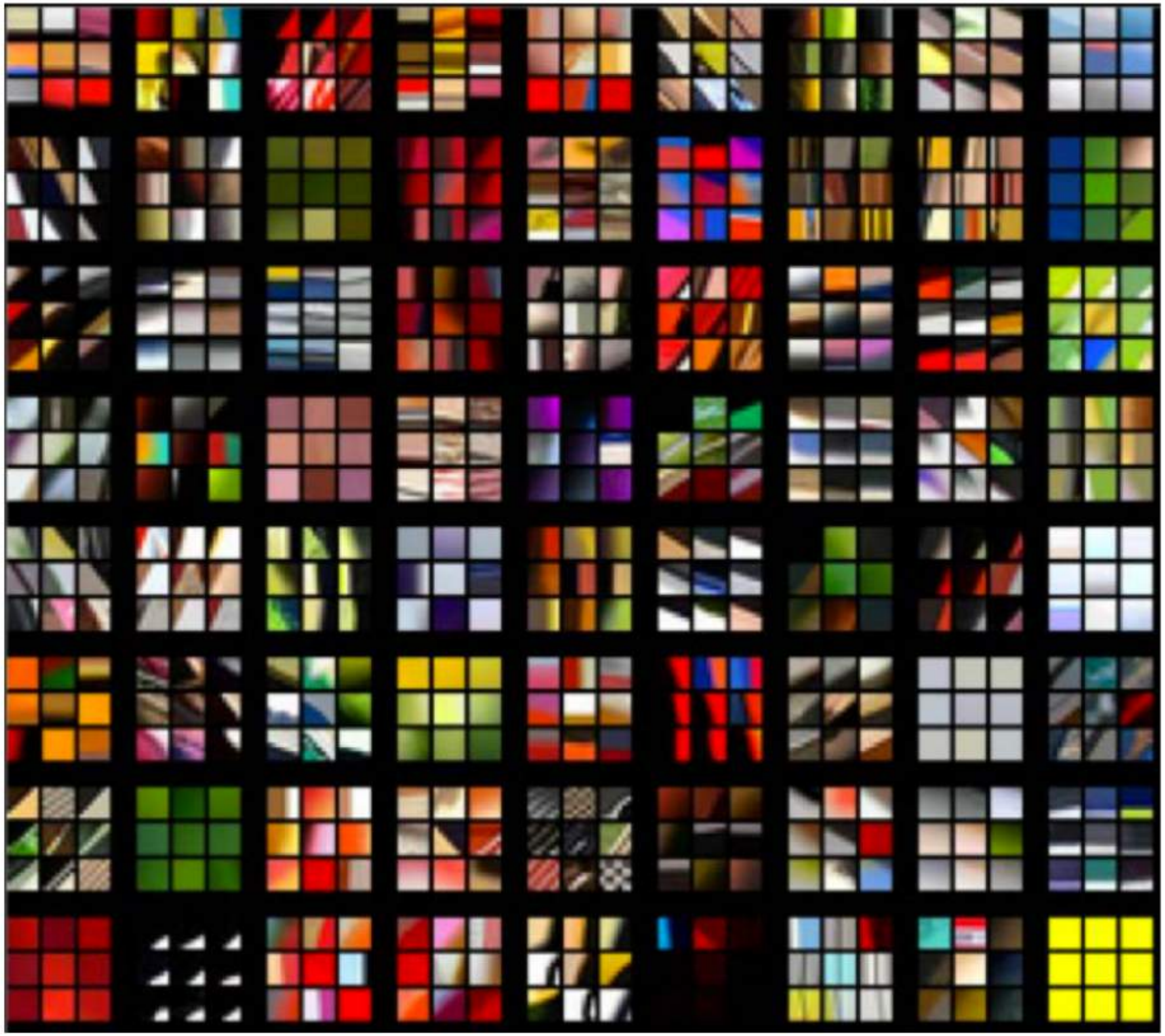


The deep network gradually learns more complex and abstract notions

From: B. Biggio

Layer 1

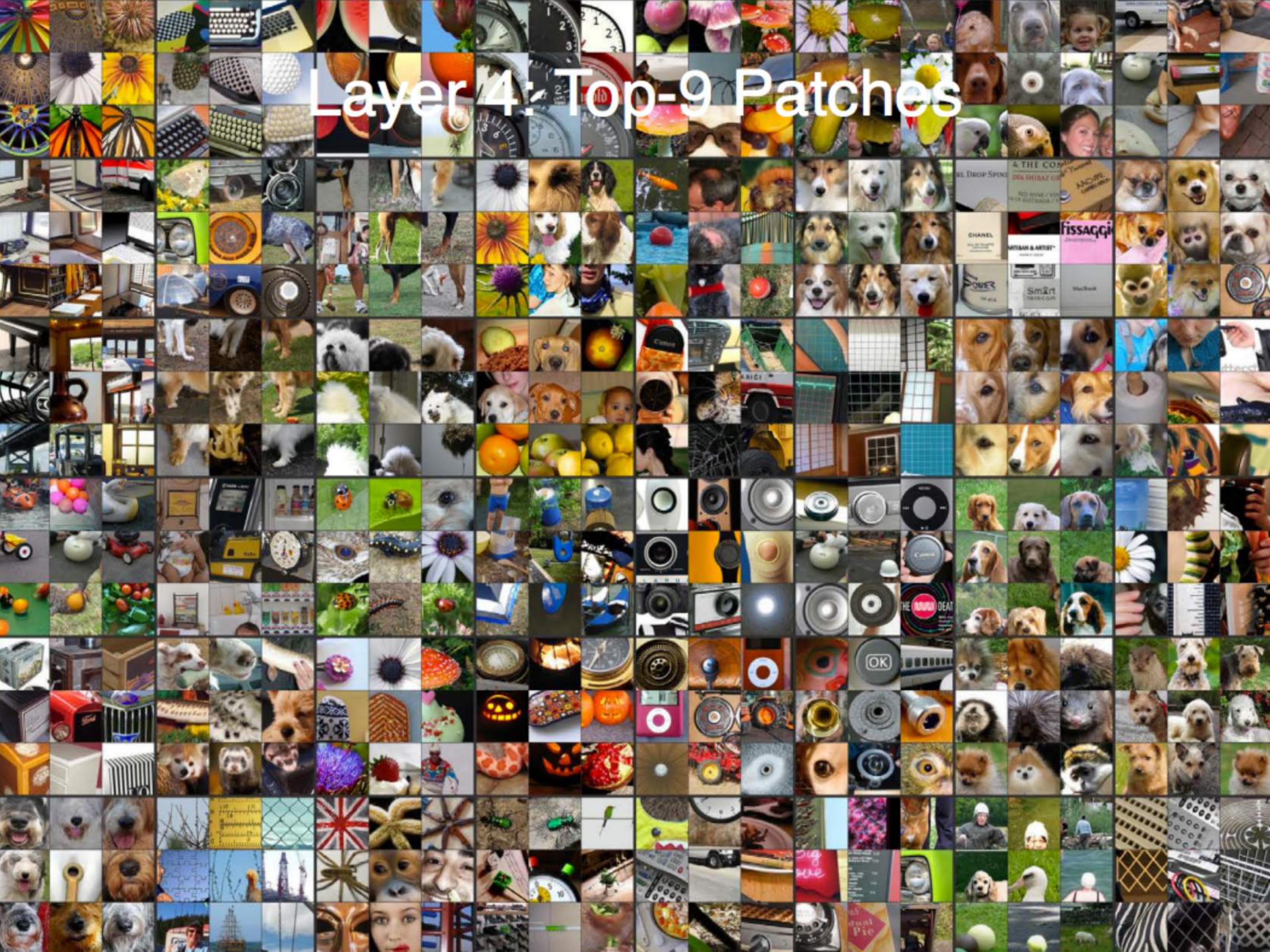
Each 3x3 block shows the top 9 patches for one filter



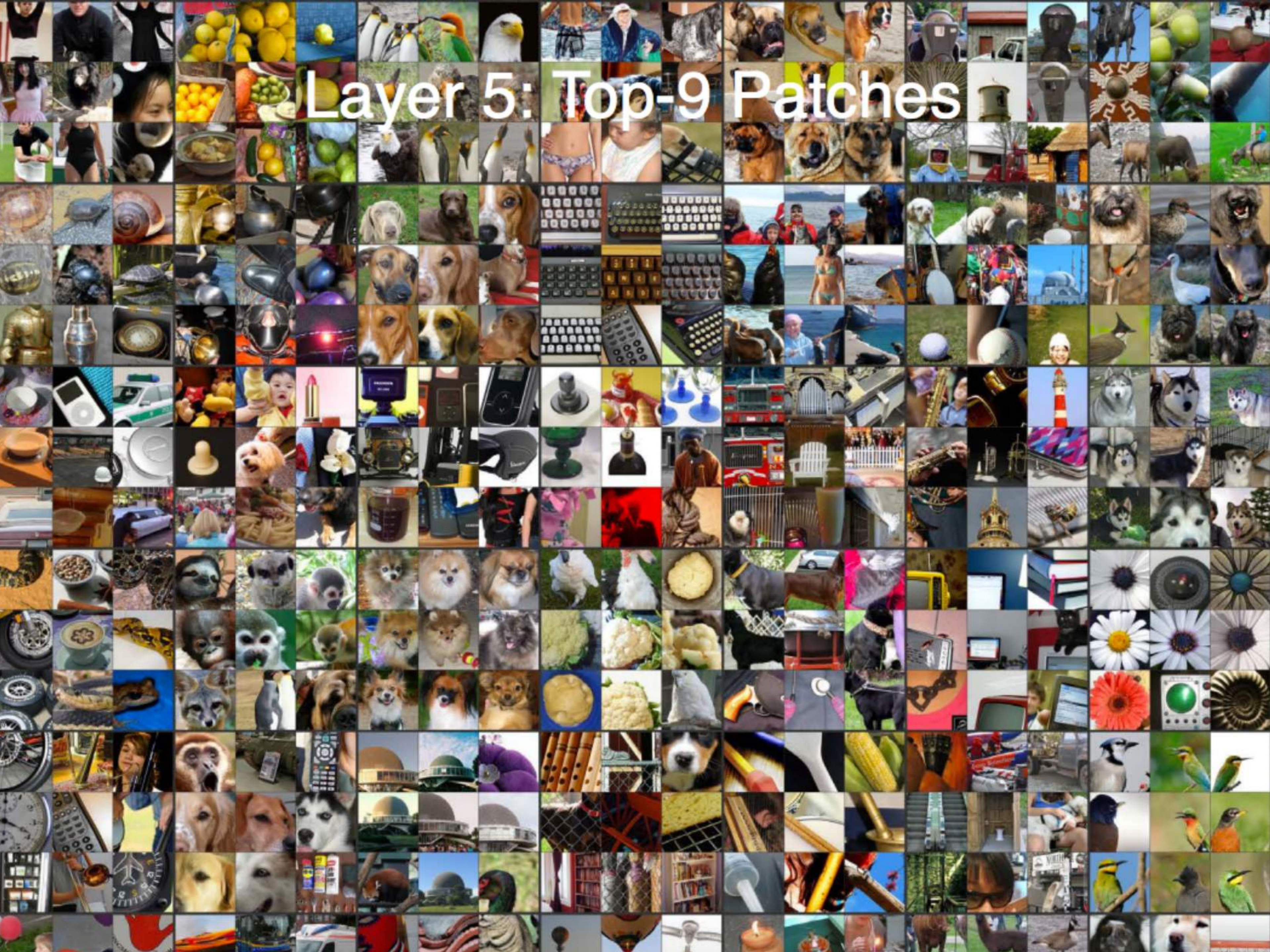
Layer 2


Layer 3: Top-9 Patches

Layer 4: Top-9 Patches



Layer 5: Top-9 Patches



Feature Analysis

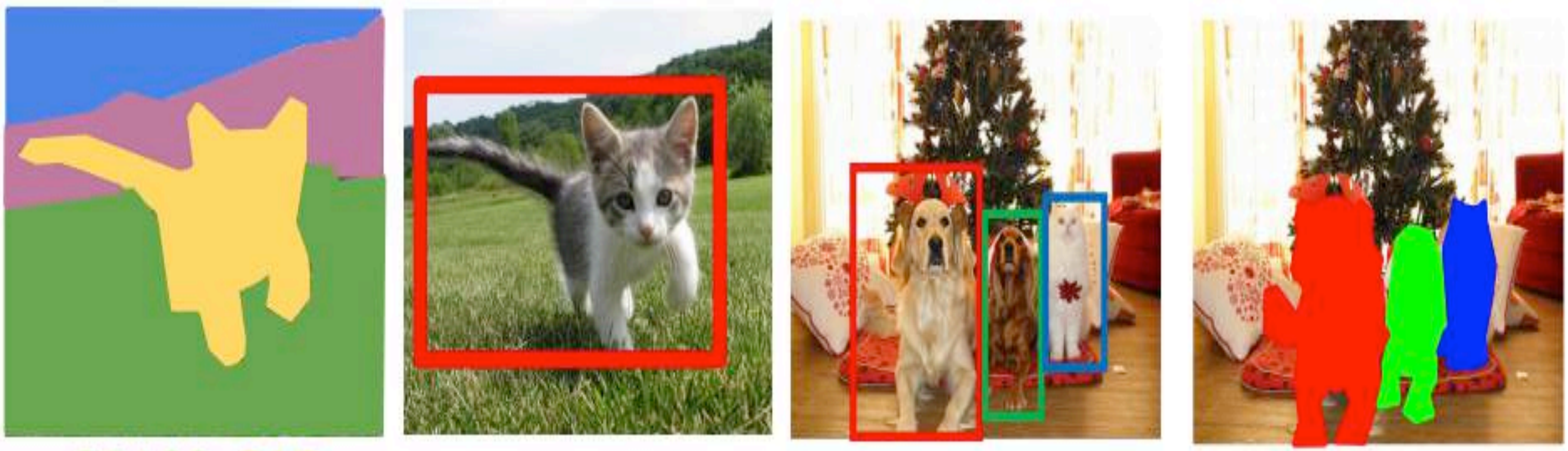
- A well-trained ConvNet is an excellent **feature extractor**.
- Chop the network at desired layer and use the output as a feature representation to train an SVM on some other dataset (Zeiler-Fergus 2013):

	Cal-101 (30/class)	Cal-256 (60/class)
SVM (1)	44.8 ± 0.7	24.6 ± 0.4
SVM (2)	66.2 ± 0.5	39.6 ± 0.3
SVM (3)	72.3 ± 0.4	46.0 ± 0.3
SVM (4)	76.6 ± 0.4	51.3 ± 0.1
SVM (5)	$\mathbf{86.2 \pm 0.8}$	65.6 ± 0.3
SVM (7)	$\mathbf{85.5 \pm 0.4}$	$\mathbf{71.7 \pm 0.2}$
Softmax (5)	82.9 ± 0.4	65.7 ± 0.5
Softmax (7)	$\mathbf{85.4 \pm 0.4}$	$\mathbf{72.6 \pm 0.1}$

- Improve further by taking a pre-trained ConvNet and re-training it on a different dataset (Fine tuning).

Other Computer Vision Tasks

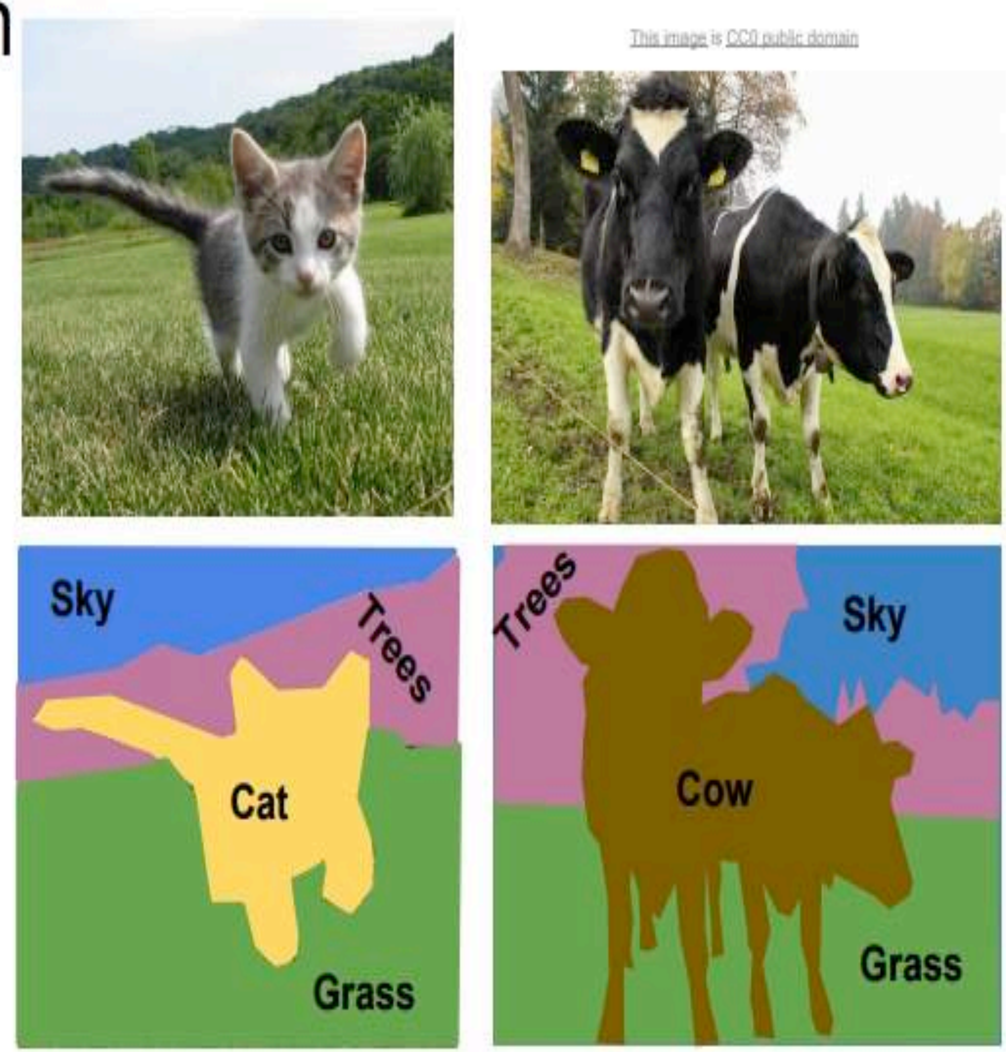
Semantic Segmentation | **Classification + Localization** | **Object Detection** | **Instance Segmentation**



GRASS, CAT, TREE, SKY | CAT | DOG, DOG, CAT | DOG, DOG, CAT

No objects, just pixels | Single Object | Multiple Object

Semantic Segmentation

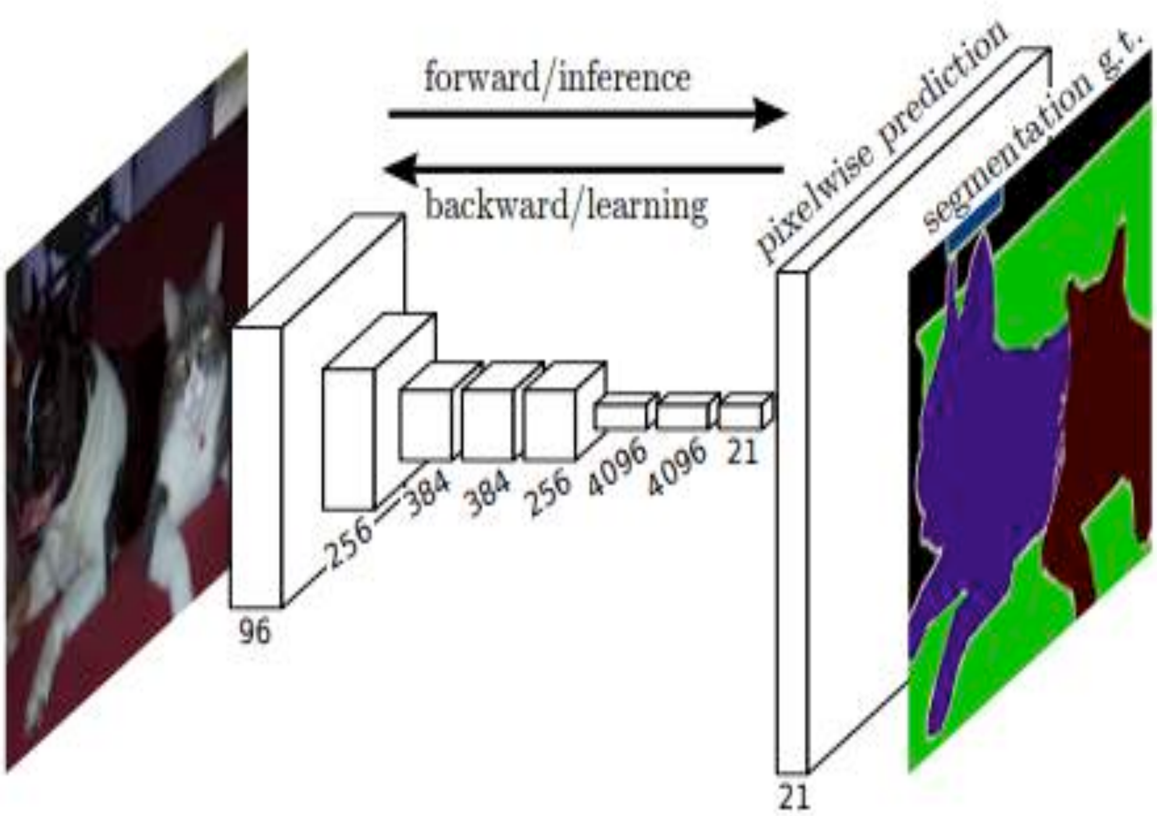


Label each pixel in the image with a category label

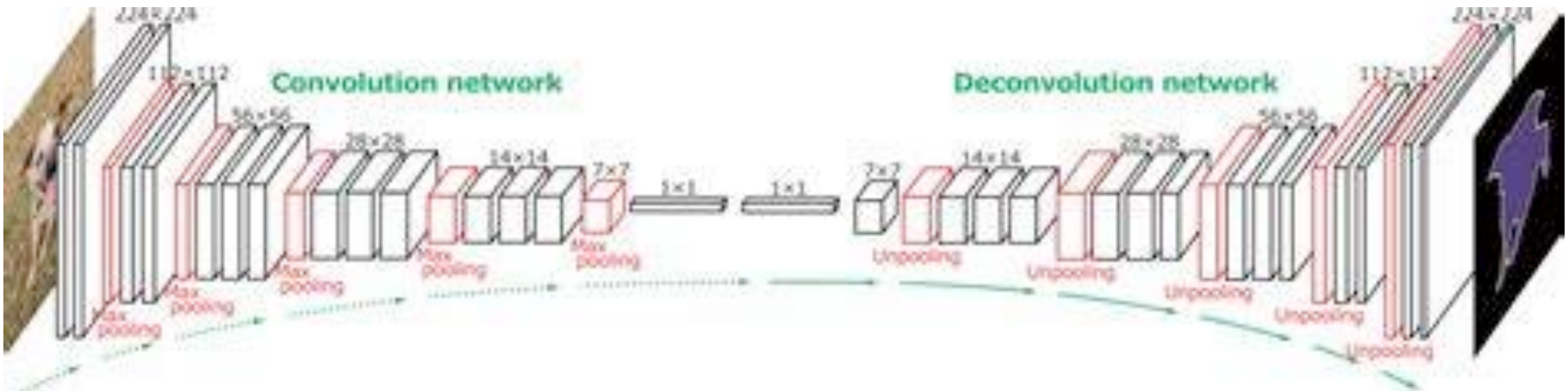
Don't differentiate instances, only care about pixels

FCN for Semantic Segmentation

- Fully connected layers at the end are replaced by convolutional layers with very large receptive fields.
- They capture the global context of the scene.
- End-to-end training



Long, Shelhamer, Darrell - Fully Convolutional Networks for Semantic Segmentation, CVPR 2015, PAMI 2016



224×224
112×112
Convolution network
56×56
28×28
14×14
7×7
1×1
1×1
7×7
14×14
28×28
56×56
112×112
224×224
Deconvolution network
Max pooling
Max pooling
Max pooling
Max pooling
Max pooling
Unpooling
Unpooling
Unpooling
Unpooling
Unpooling

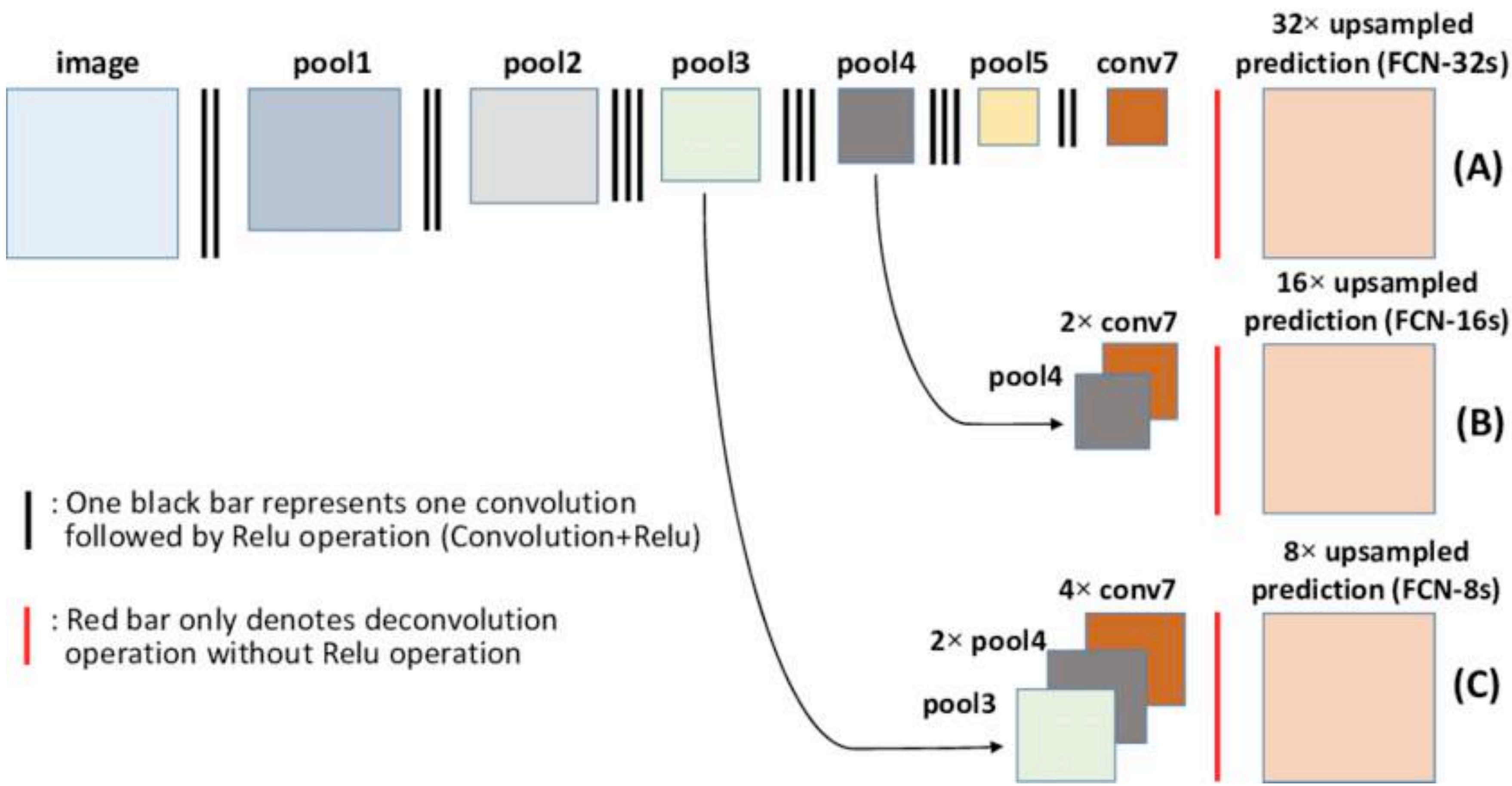
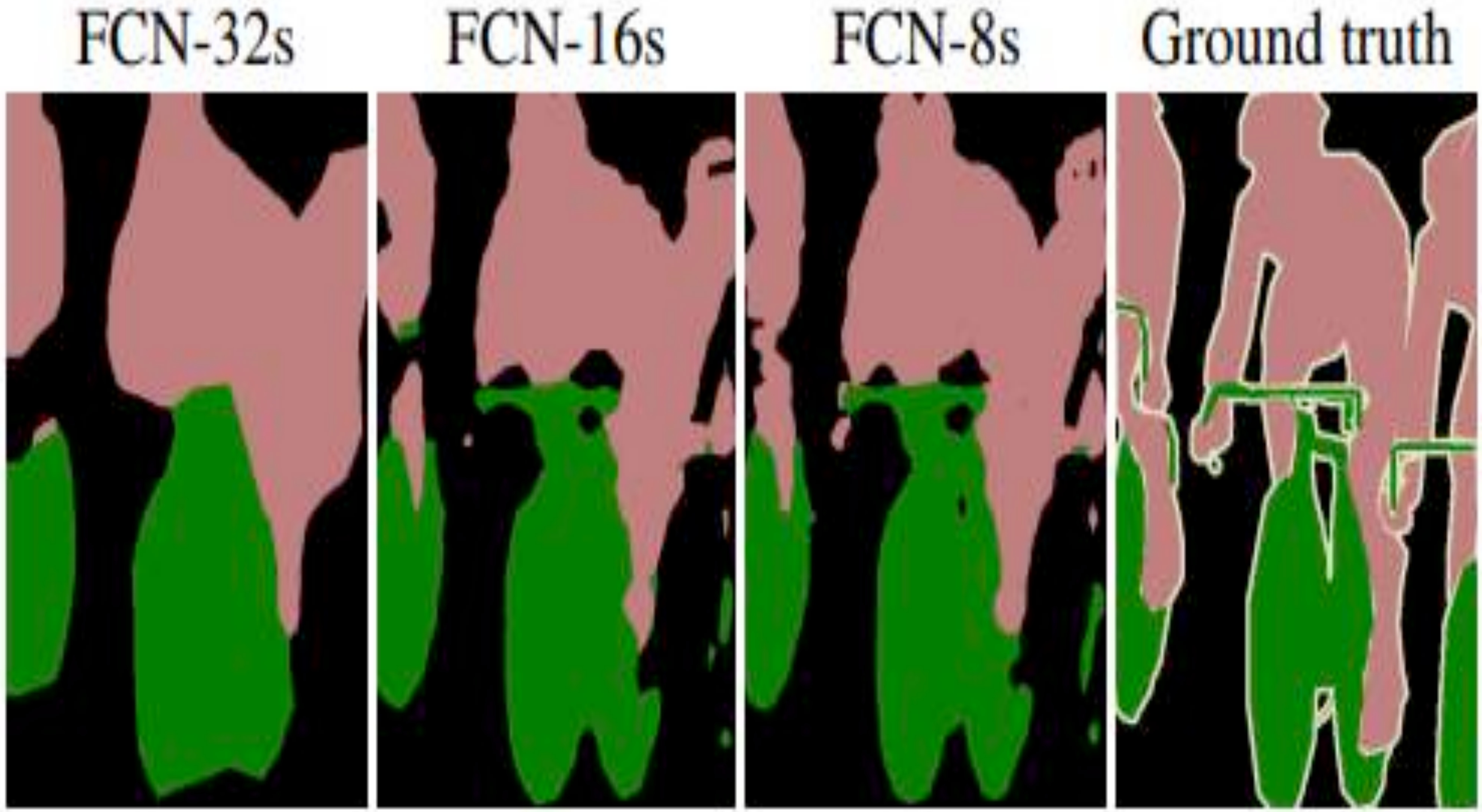
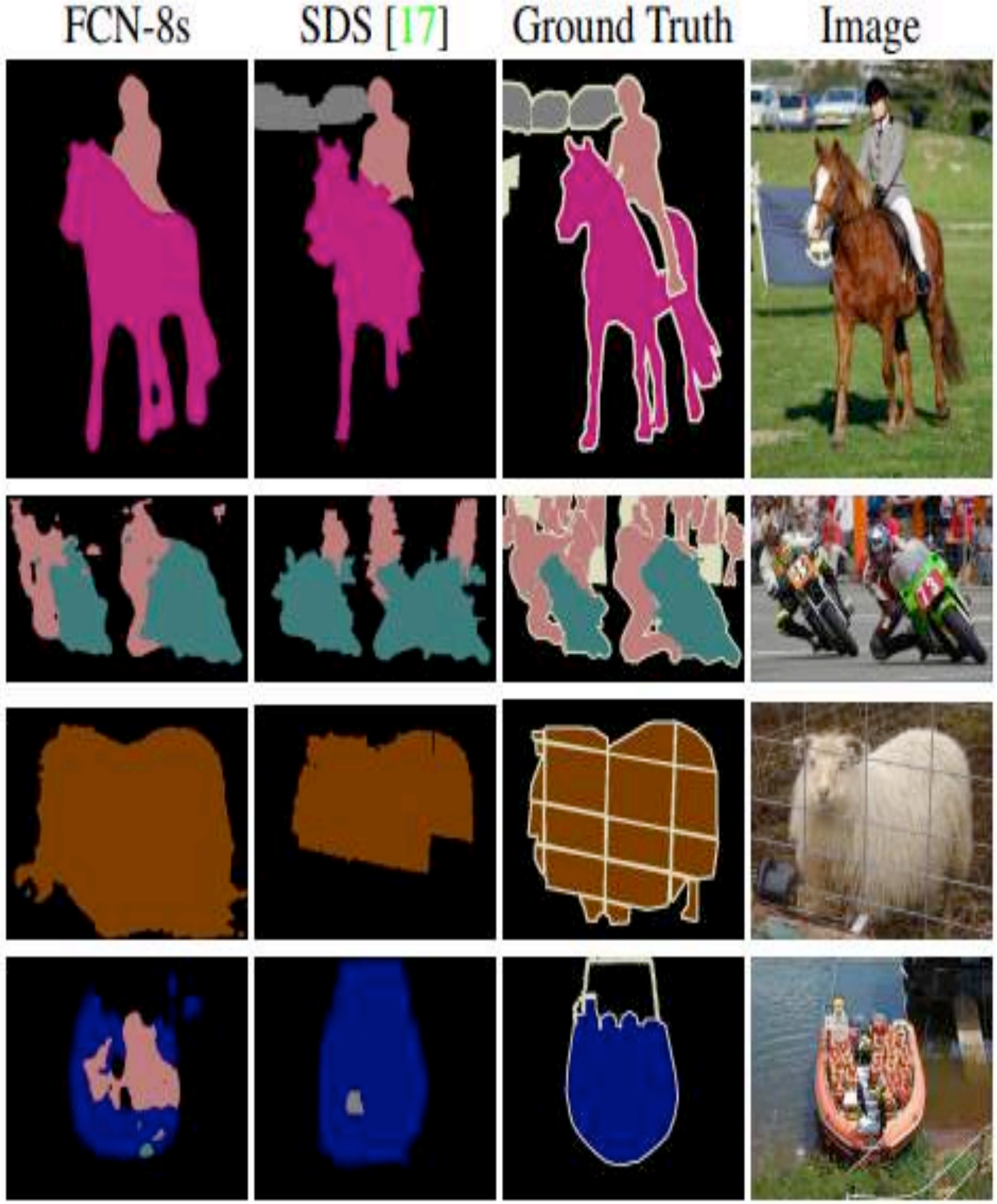


image
pool1
pool2
pool3
pool4
pool5
conv7
32× upsampled
prediction (FCN-32s)
(A)
2× conv7
pool4
16× upsampled
prediction (FCN-16s)
(B)
: One black bar represents one convolution
followed by Relu operation (Convolution+Relu)
: Red bar only denotes deconvolution
operation without Relu operation
4× conv7
2× pool4
pool3
8× upsampled
prediction (FCN-8s)
(C)




FCN-8s
SDS [17]
Ground Truth
Image

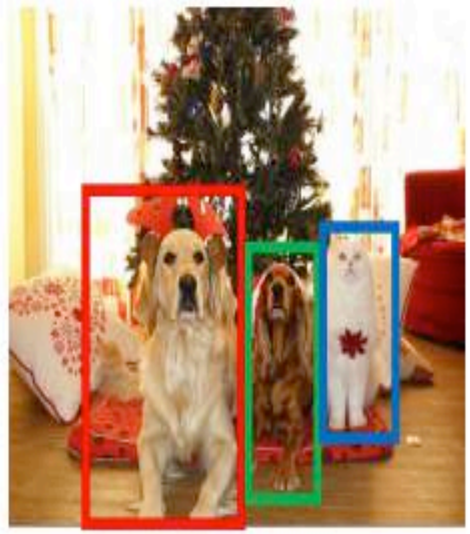
2D Object Detection

Semantic Segmentation

GRASS, CAT, TREE, SKY

No objects, just pixels

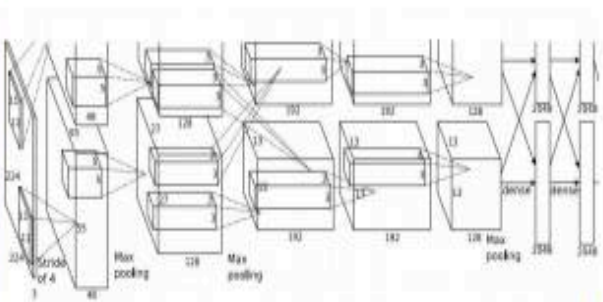
2D Object Detection



DOG, DOG, CAT

Object categories + 2D bounding boxes

Classification + Localization



Vector:
4096

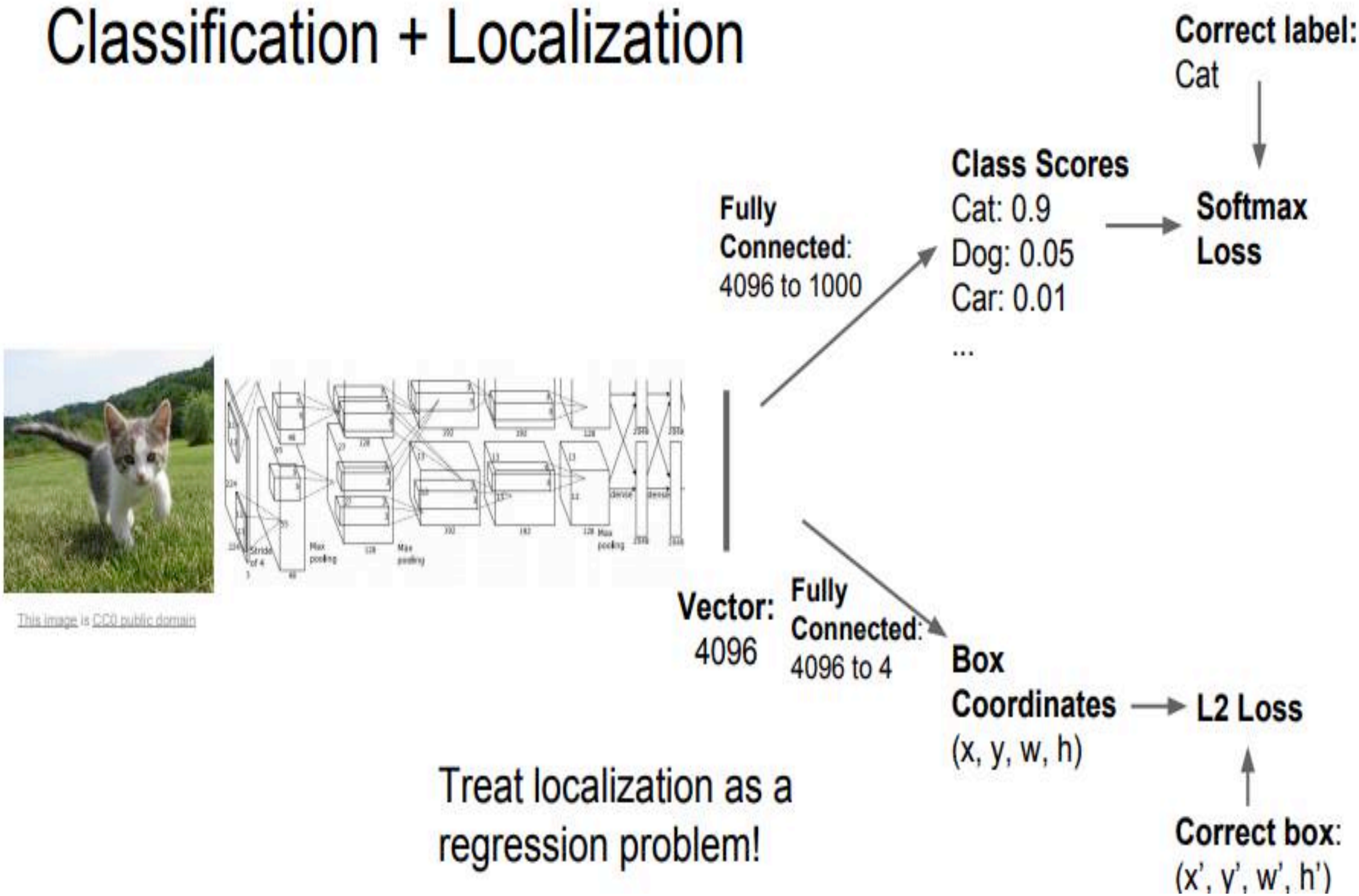
Fully Connected:
4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

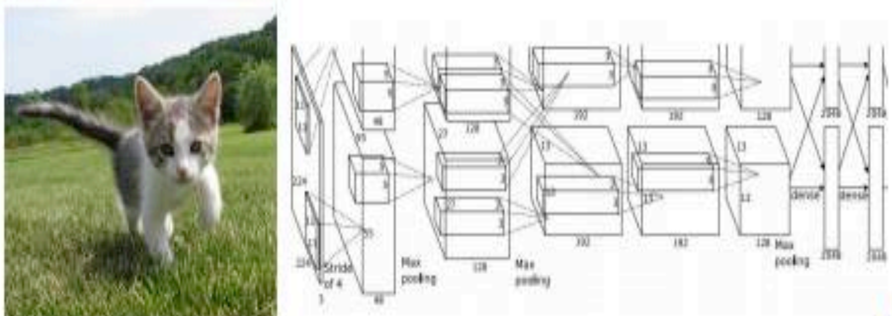
Fully Connected:
4096 to 4

Box Coordinates
(x, y, w, h)

Treat localization as a
regression problem!


Classification + Localization
Correct label:
Cat
Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...
Softmax
Loss
Fully
Connected:
4096 to 1000
This image is CC0 public domain
Vector:
4096
Fully
Connected:
4096 to 4
Box
Coordinates
(x, y, w, h)
L2 Loss
Treat localization as a
regression problem!
Correct box:
(x', y', w', h')

Classification + Localization



Vector: 4096

Fully Connected: 4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Correct label:
Cat

Softmax Loss

Fully Connected: 4096 to 4

Box Coordinates
(x, y, w, h)

L2 Loss

Correct box:
(x', y', w', h')

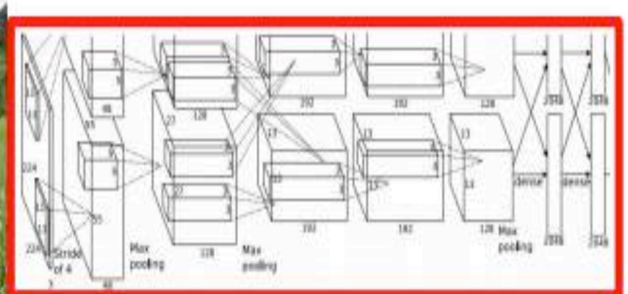
Multitask Loss

+ → **Loss**

Treat localization as a regression problem!

Classification + Localization

This image is CC0 public domain



Often pretrained on ImageNet
(Transfer learning)

Vector: 4096

Fully Connected: 4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Correct label: Cat

Softmax Loss

Fully Connected: 4096 to 4

Box Coordinates (x, y, w, h)

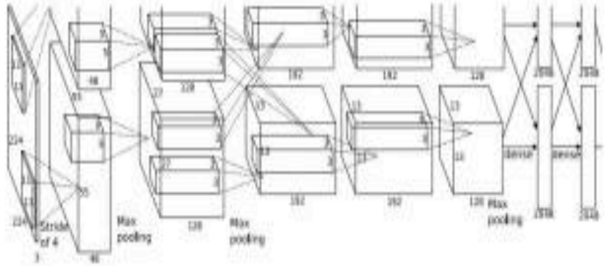
L2 Loss

Correct box: (x', y', w', h')

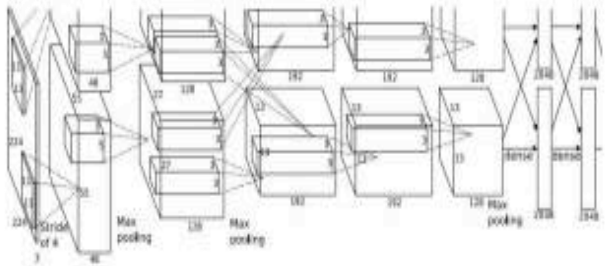
+ → **Loss**

Treat localization as a regression problem!

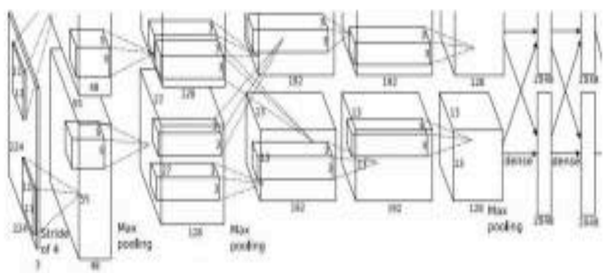
Object Detection as Regression?



CAT: (x, y, w, h)



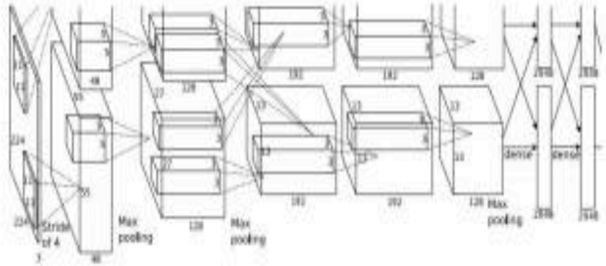
DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)



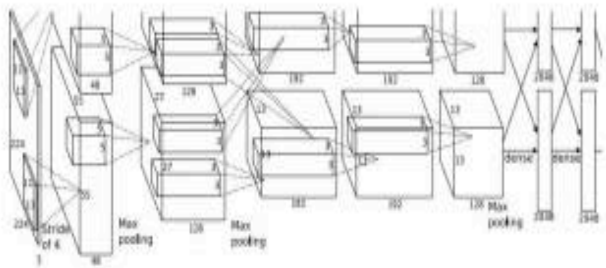
DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
....

Object Detection as Regression?

Each image needs a different number of outputs!

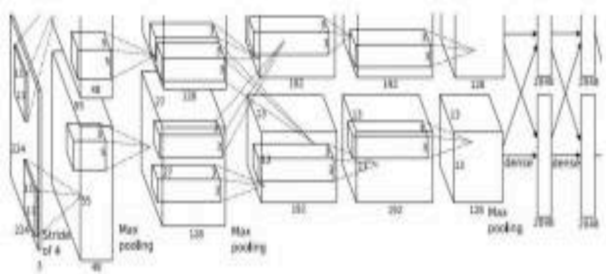



CAT: (x, y, w, h) 4 numbers

DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)

16 numbers

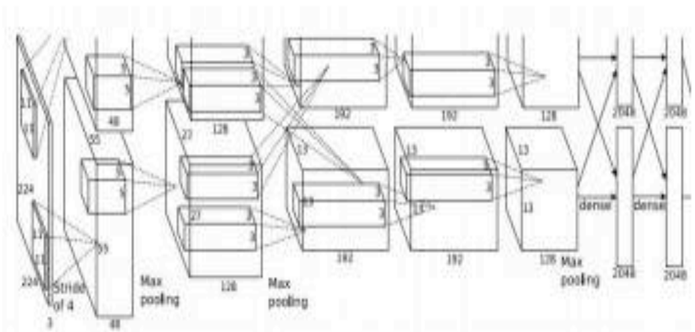



DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
....

Many numbers!

Object Detection as Classification: Sliding Window

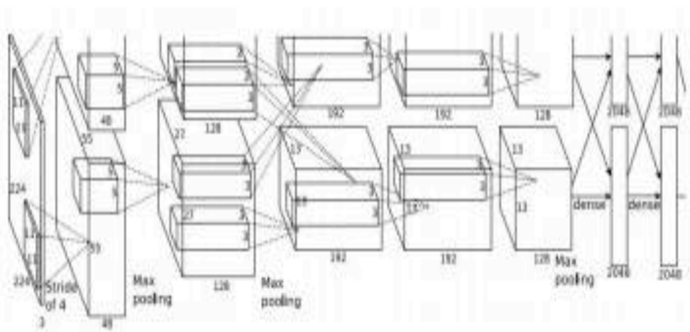
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? NO
Background? YES

Object Detection as Classification: Sliding Window

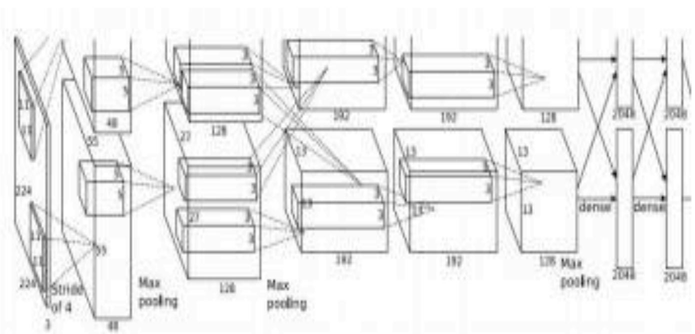
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES
Cat? NO
Background? NO

Object Detection as Classification: Sliding Window

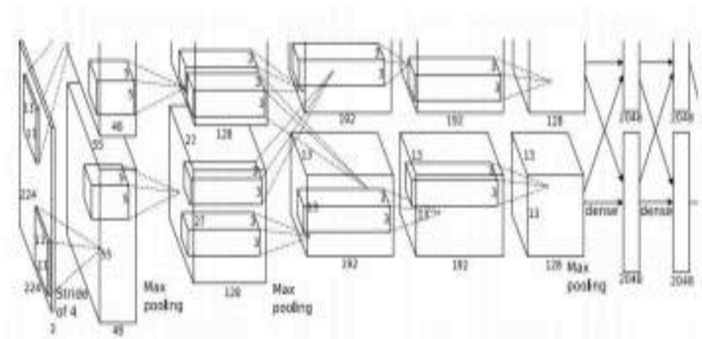
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES
Cat? NO
Background? NO

Object Detection as Classification: Sliding Window

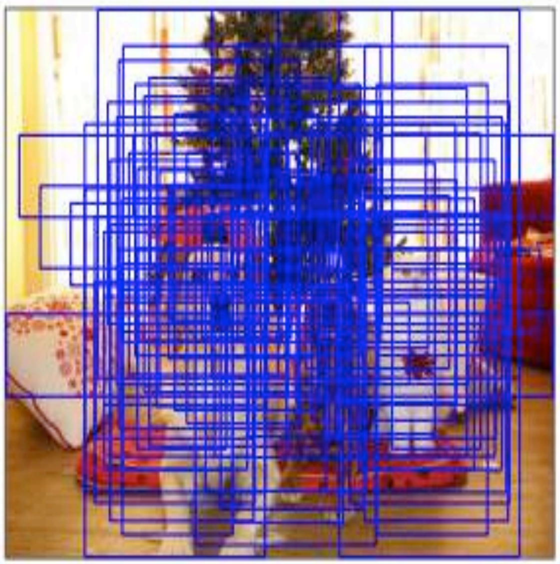
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

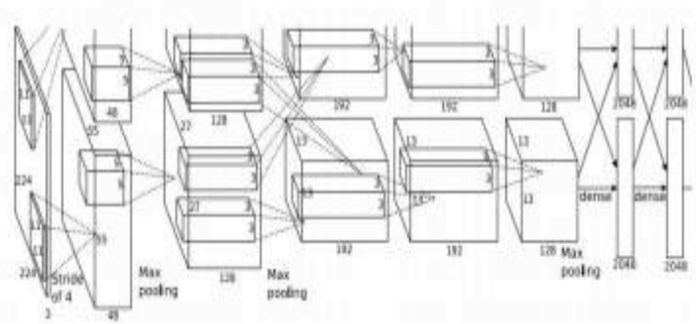


Dog? NO
Cat? YES
Background? NO

Object Detection as Classification: Sliding Window

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



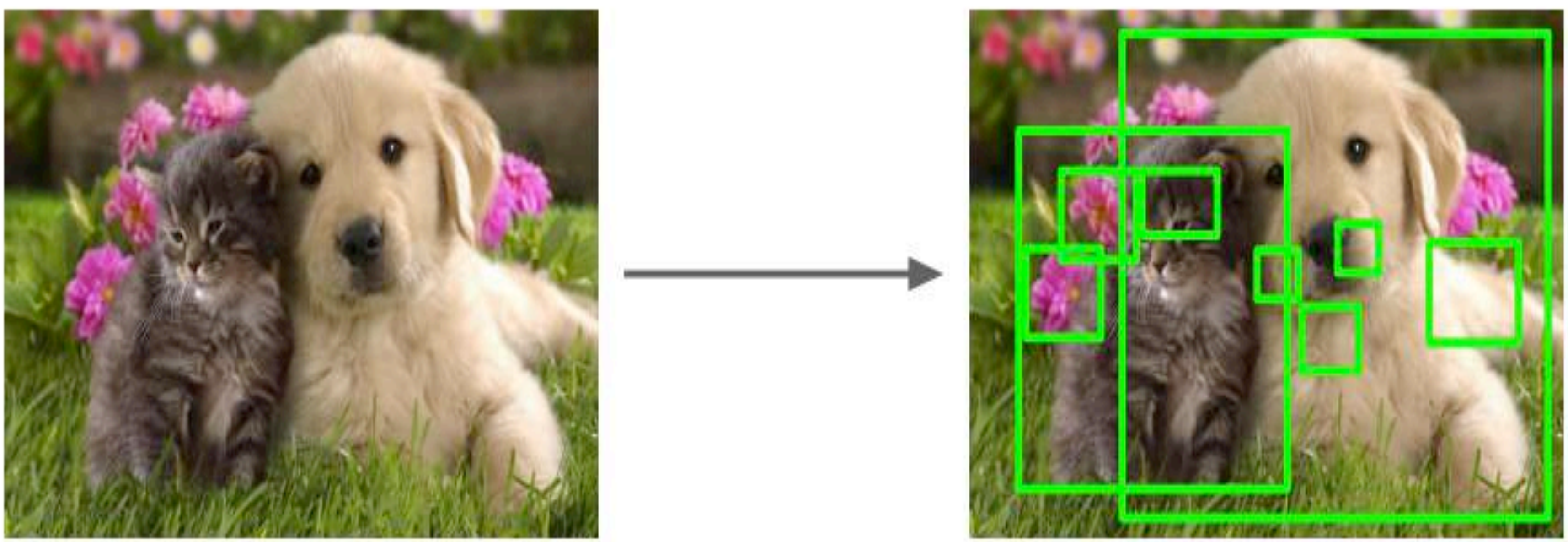


Dog? NO
Cat? YES
Background? NO

Problem: Need to apply CNN to huge number of locations, scales, and aspect ratios, very computationally expensive!

Region Proposals / Selective Search

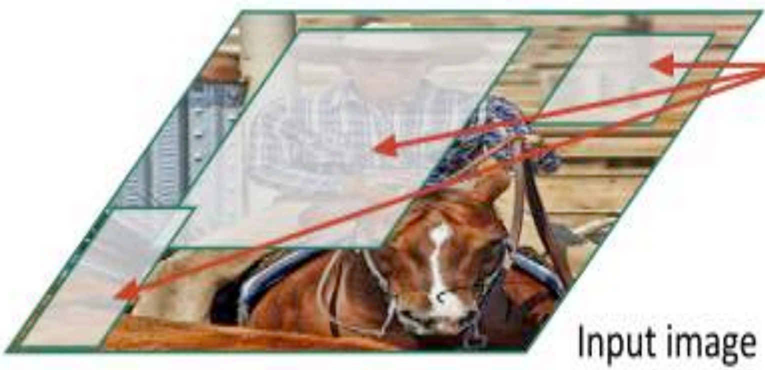
- Find “blobby” image regions that are likely to contain objects
- Relatively fast to run; e.g. Selective Search gives 2000 region proposals in a few seconds on CPU



R-CNN

[Girshick, Donahue, Darrell, Malik - Rich feature hierarchies for accurate object detection and semantic segmentation, CVPR 2014]

R-CNN

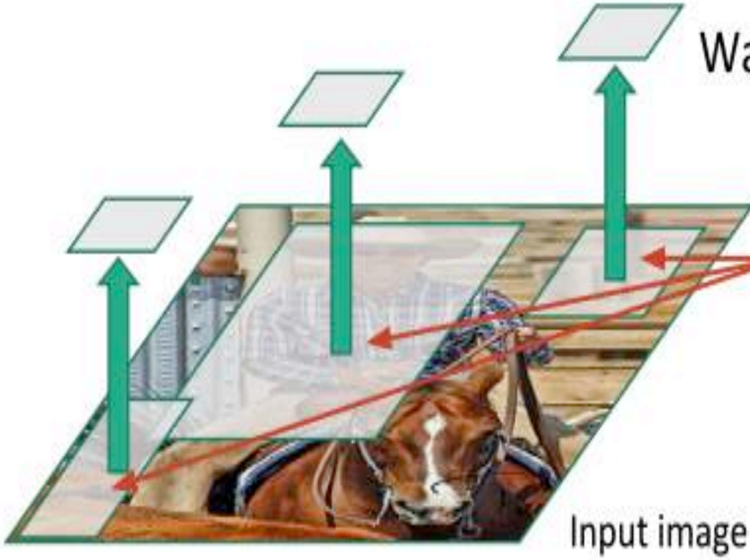


Regions of Interest (RoI) from a proposal method (~2k)

Input image

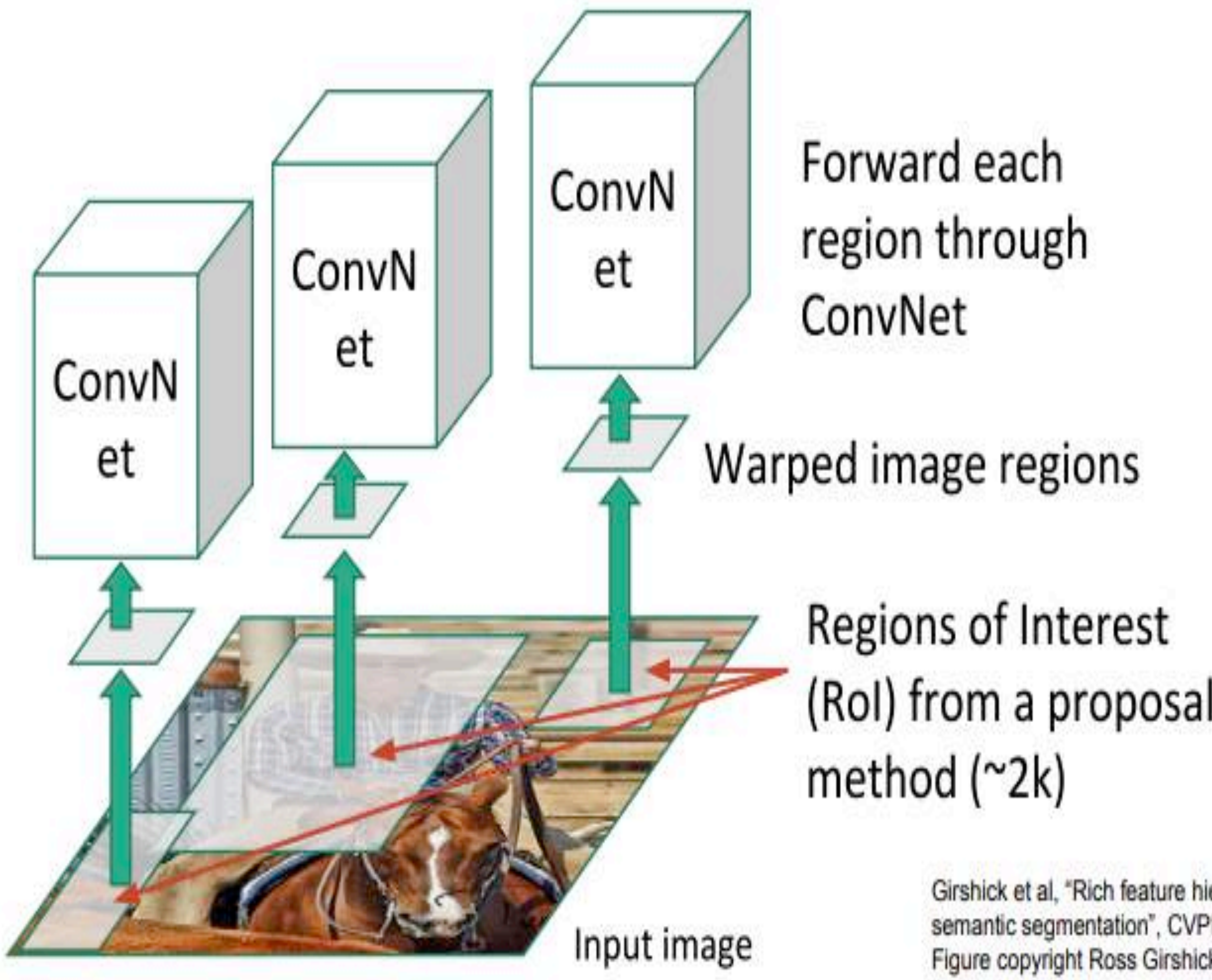
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

R-CNN



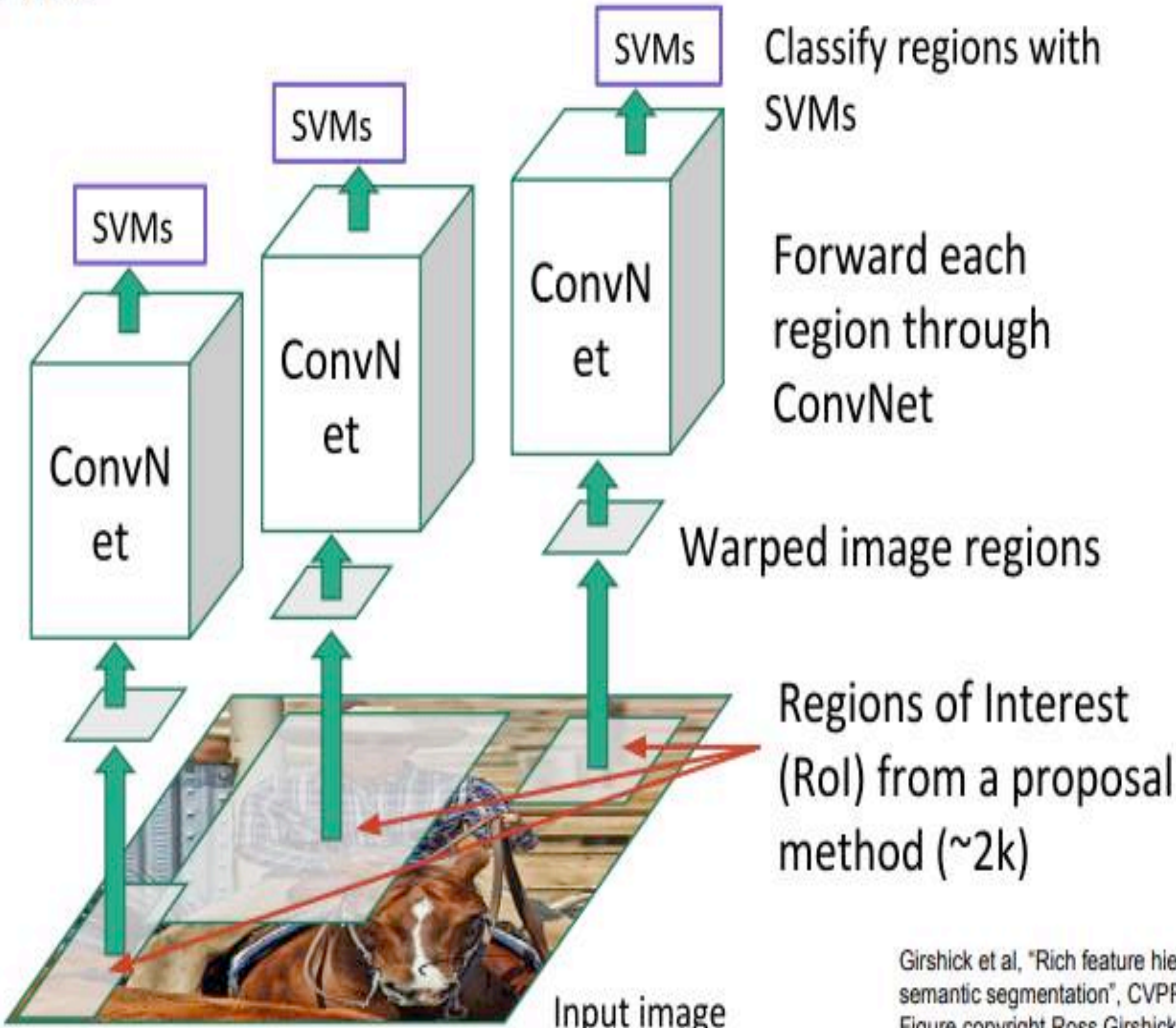
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

R-CNN



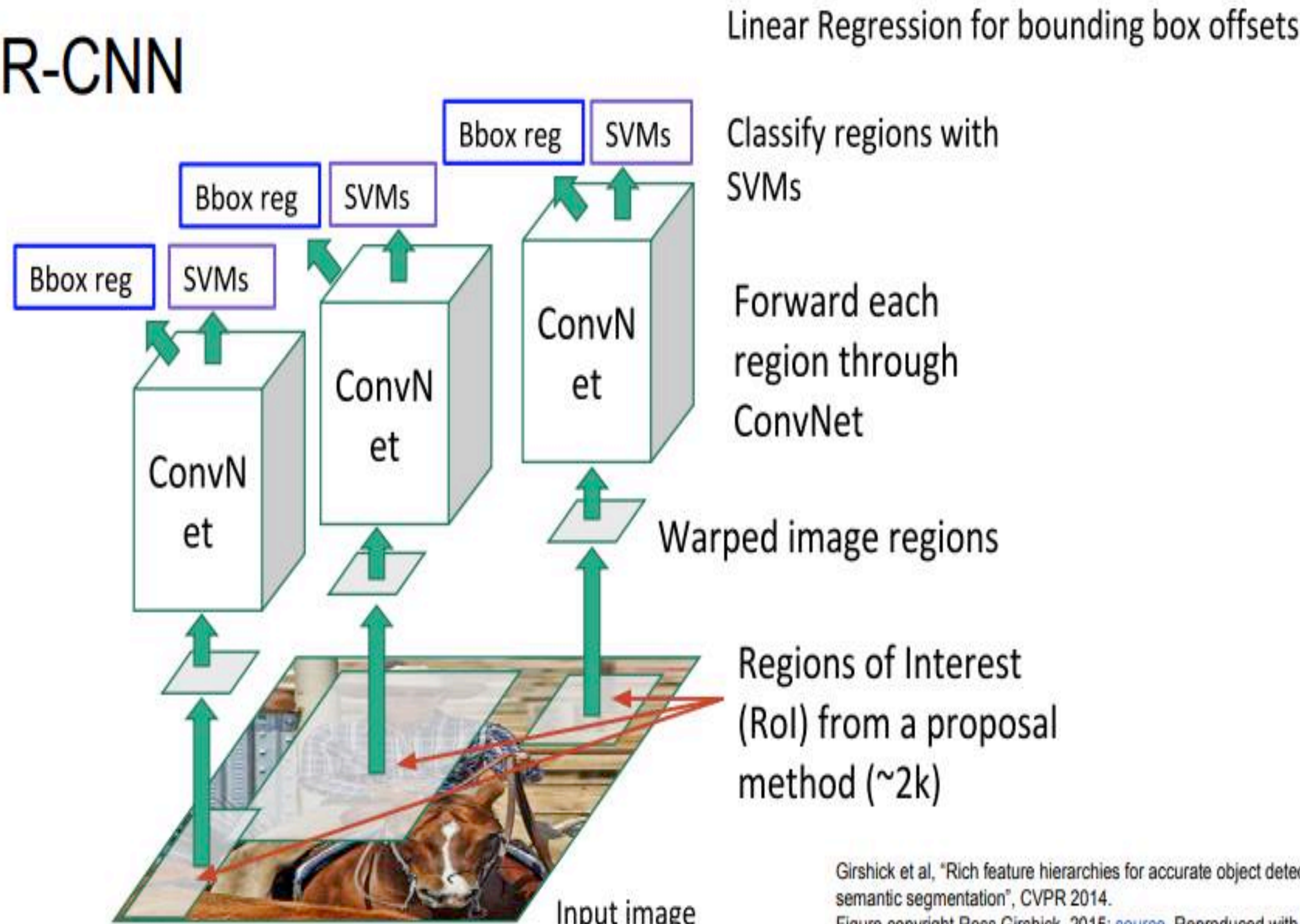
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

R-CNN



Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

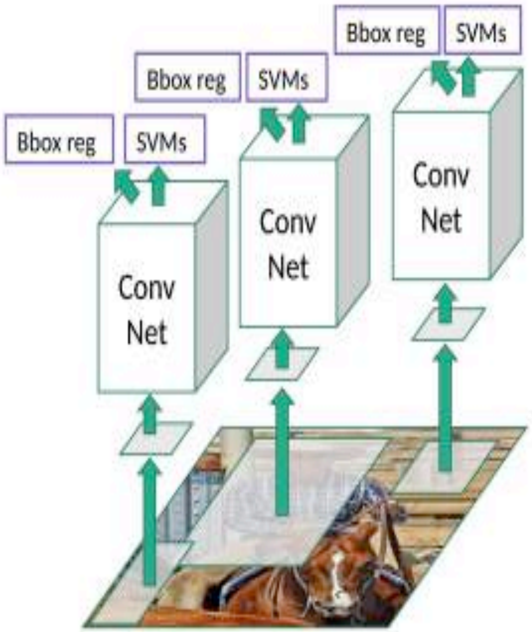
R-CNN



Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

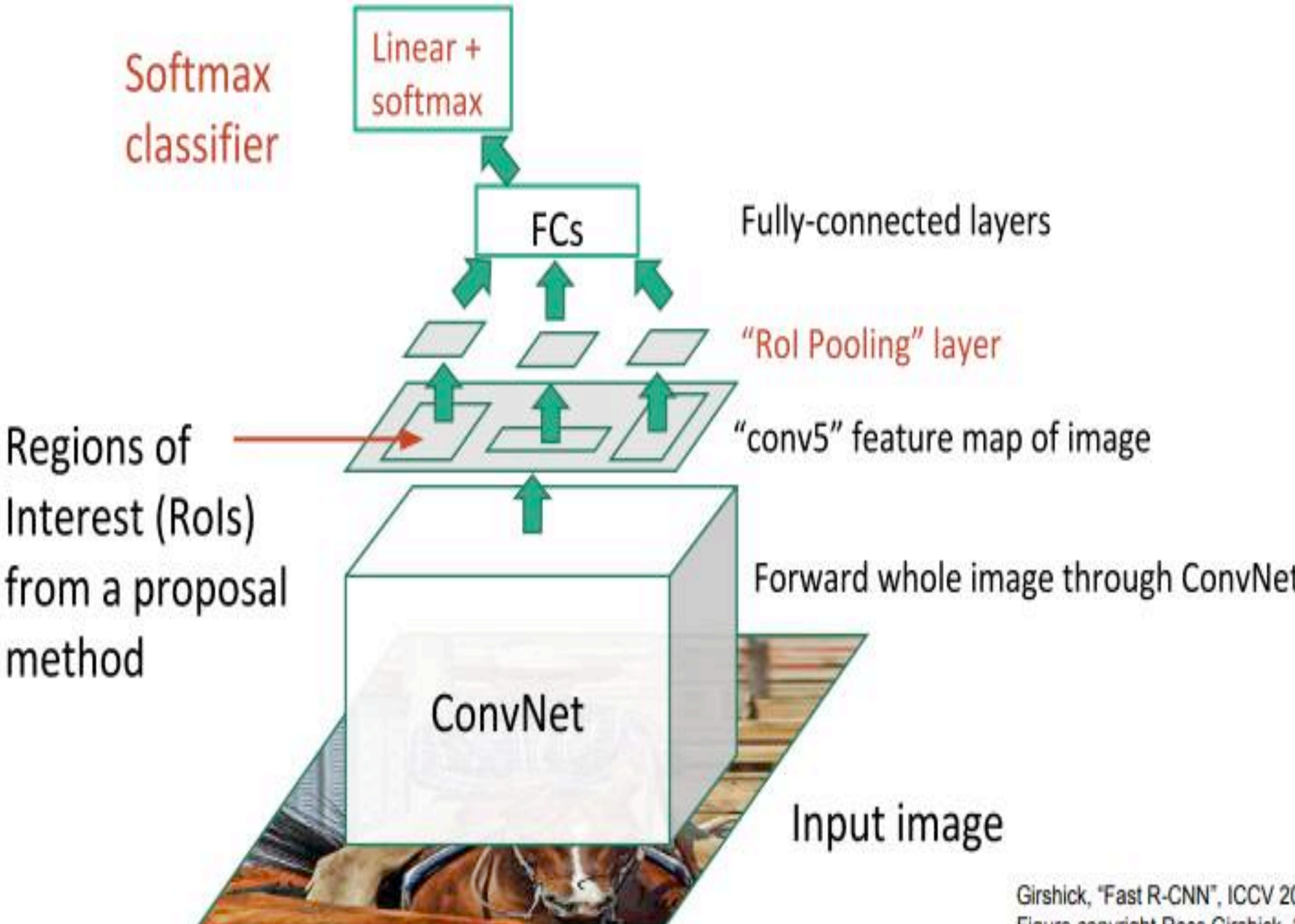
R-CNN: Problems

- Ad hoc training objectives
  - Fine-tune network with softmax classifier (log loss)
  - Train post-hoc linear SVMs (hinge loss)
  - Train post-hoc bounding-box regressions (least squares)
- Training is slow (84h), takes a lot of disk space
- Inference (detection) is slow
  - 47s / image with VGG16 [Simonyan & Zisserman. ICLR15]
  - Fixed by SPP-net [He et al. ECCV14]



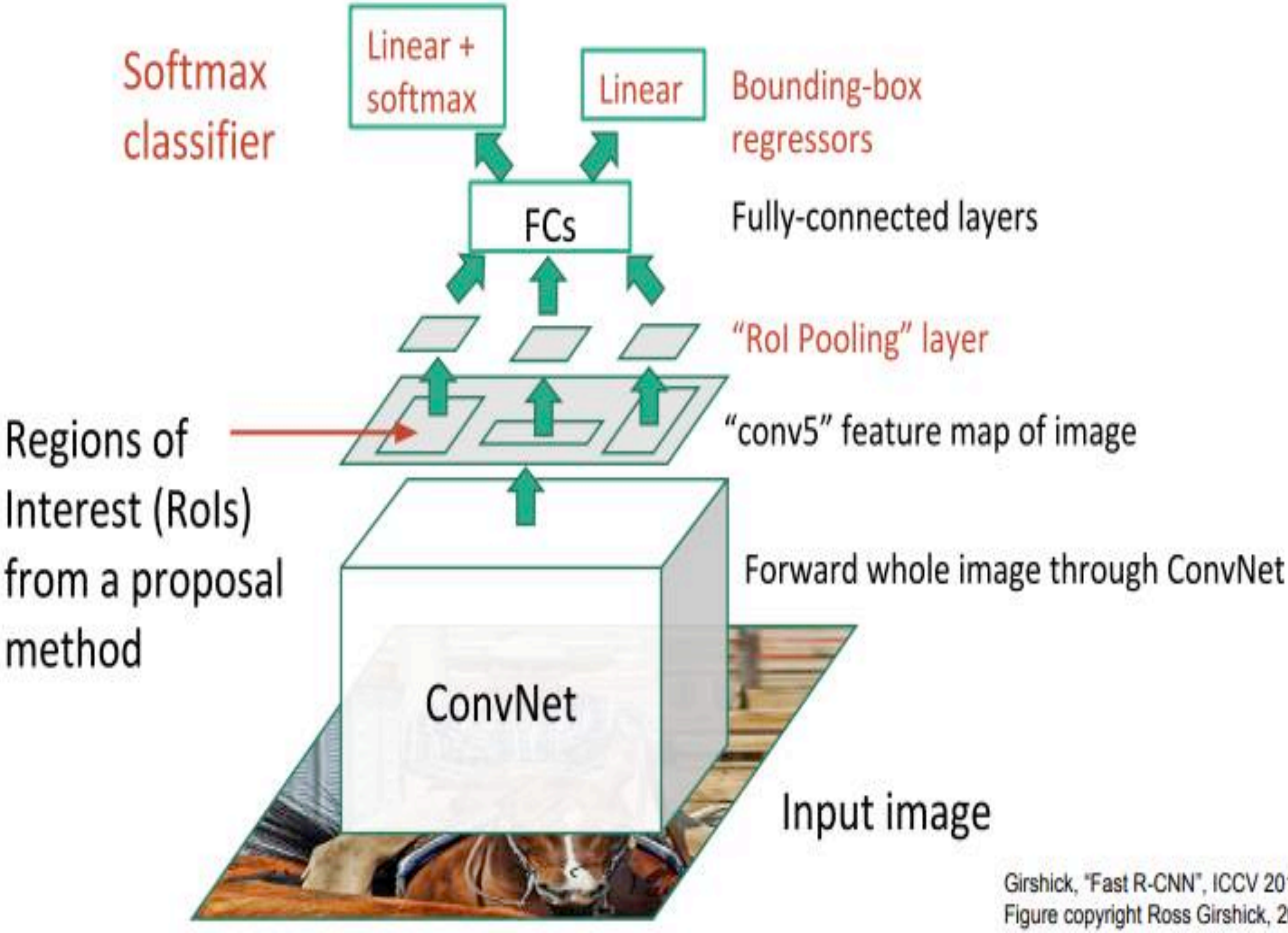
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

Fast R-CNN



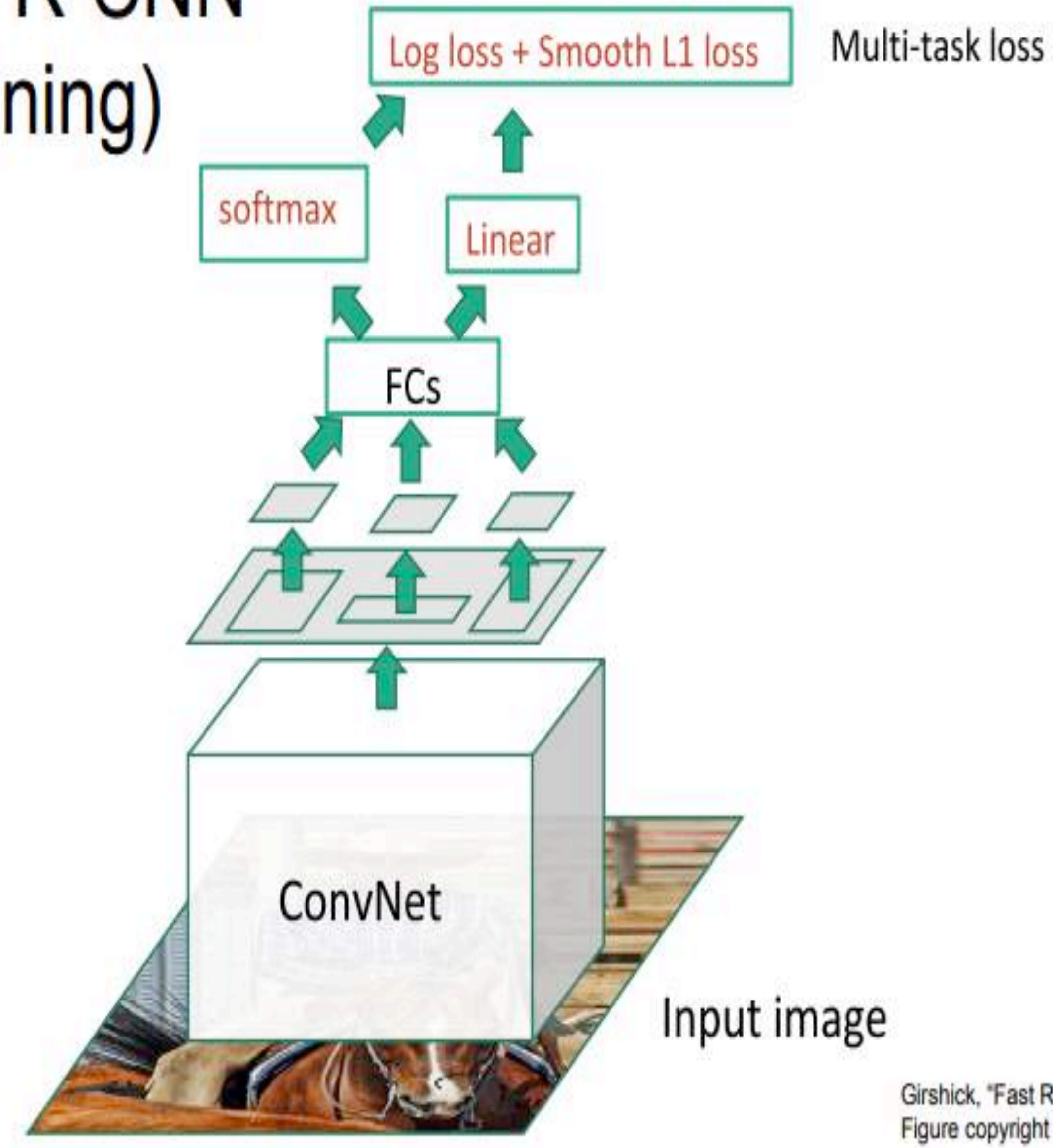
Girshick, "Fast R-CNN", ICCV 2015.

Fast R-CNN



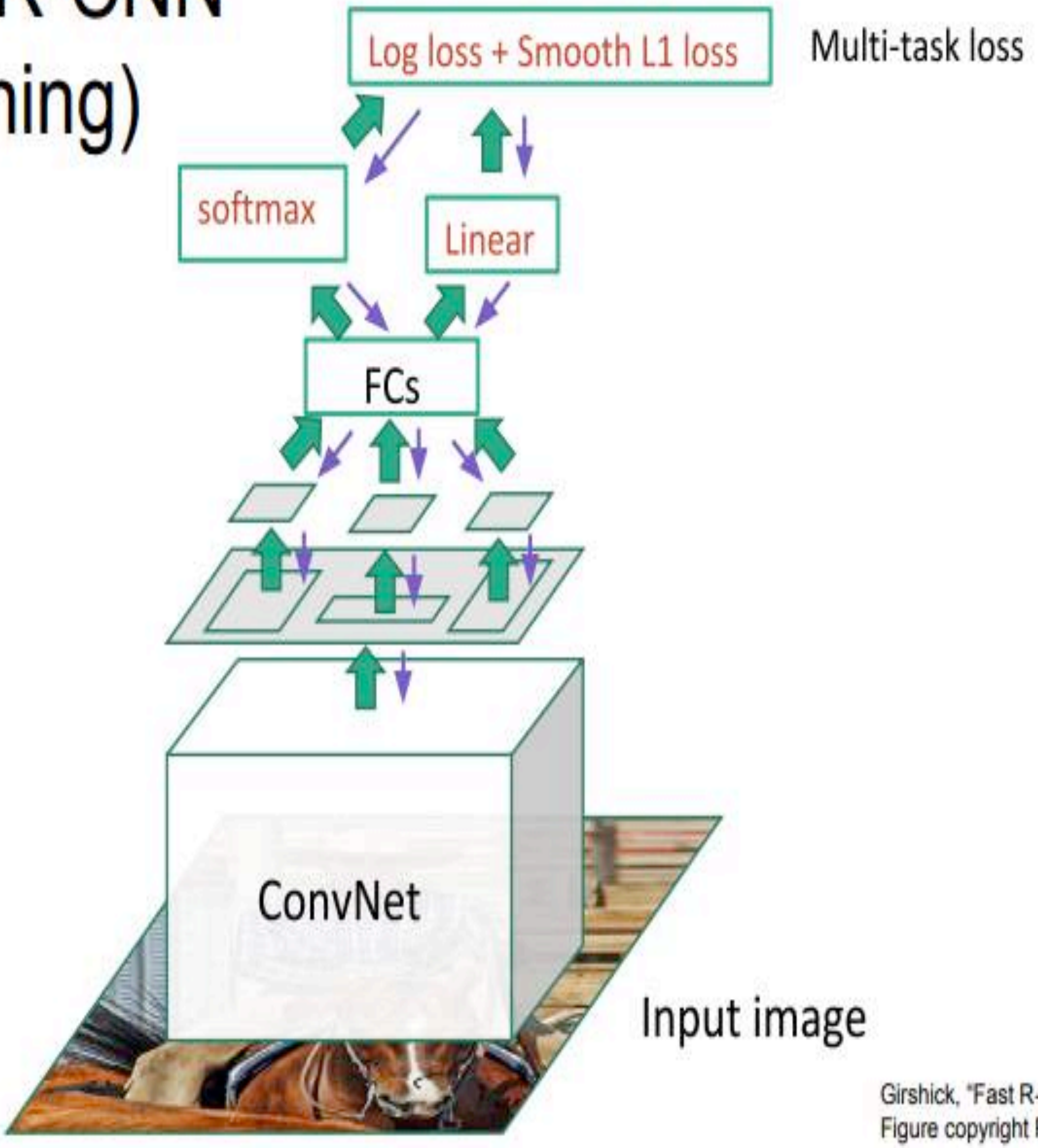
Girshick, "Fast R-CNN", ICCV 2015.

Fast R-CNN (Training)



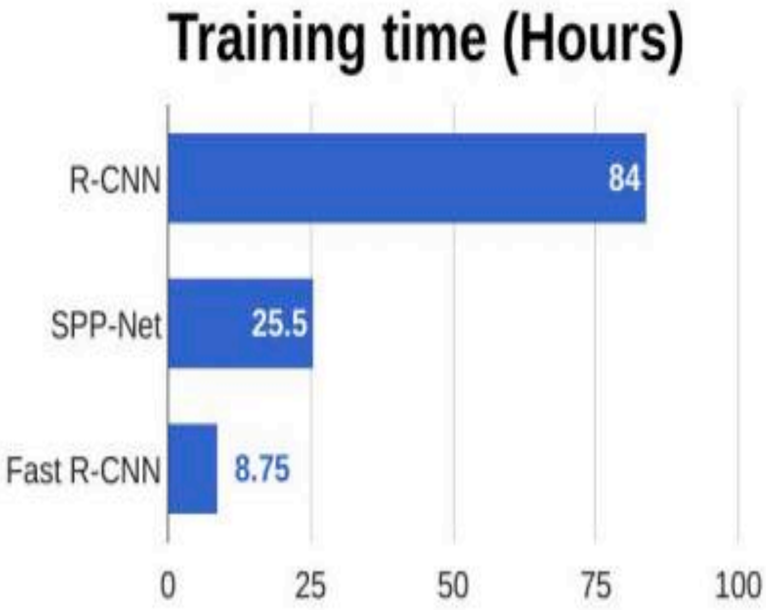
Girshick, "Fast R-CNN", ICCV 2015.

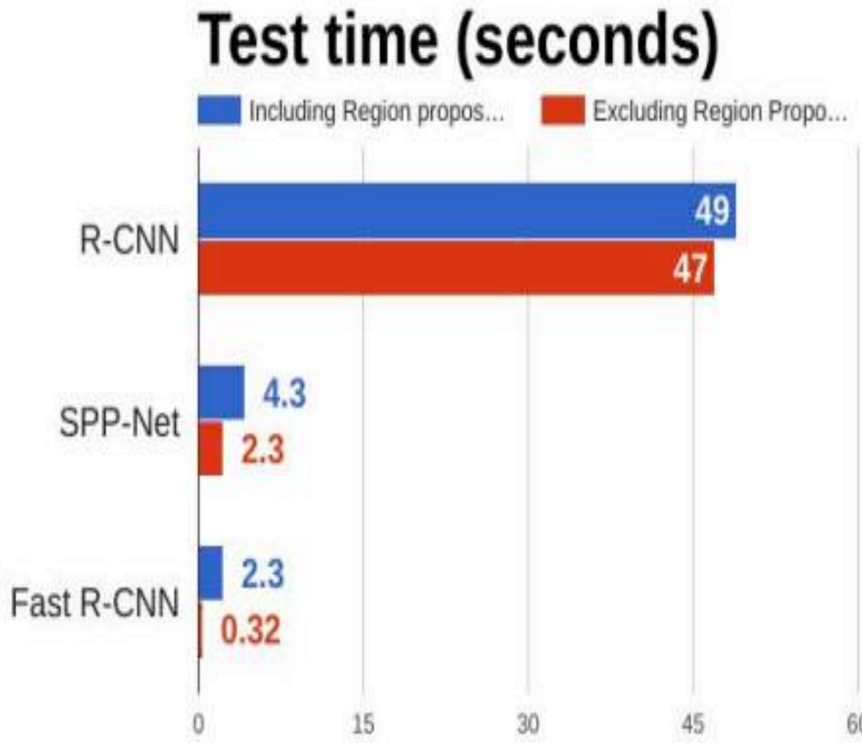
Fast R-CNN (Training)



Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

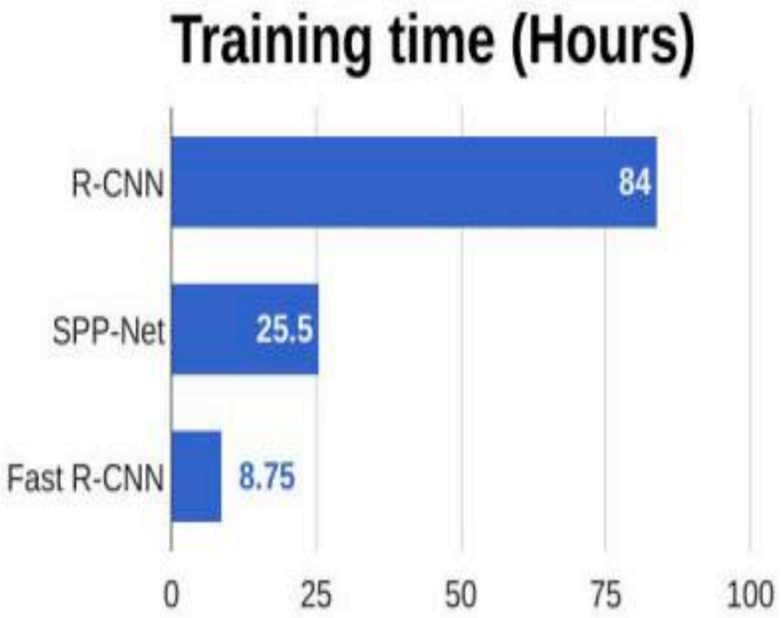
R-CNN vs SPP vs Fast R-CNN

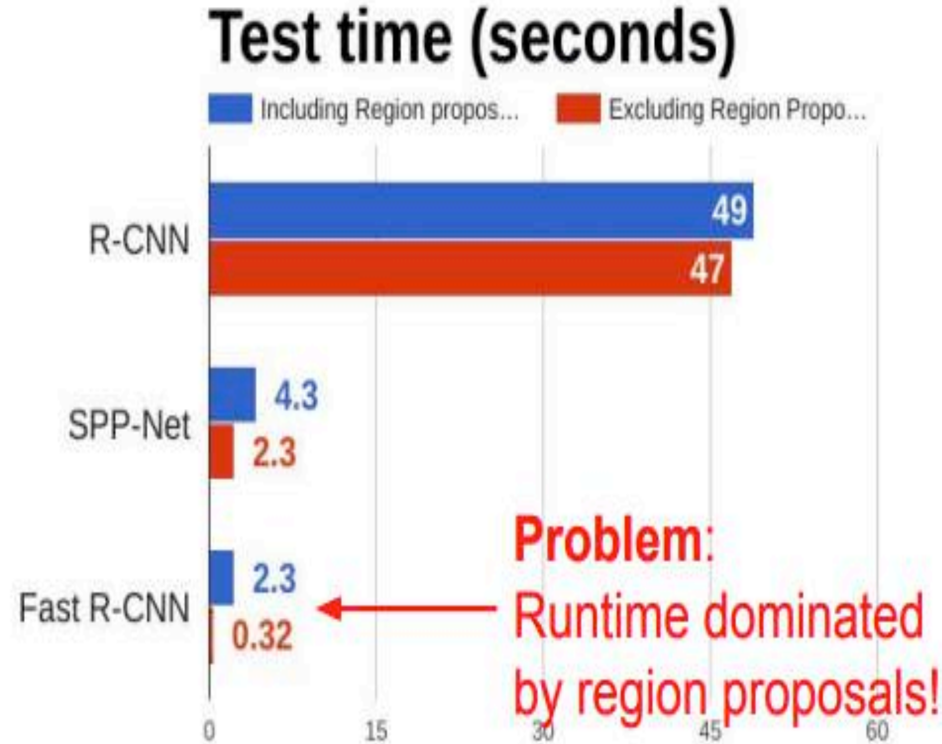




Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
He et al, "Spatial pyramid pooling in deep convolutional networks for visual recognition", ECCV 2014
Girshick, "Fast R-CNN", ICCV 2015

R-CNN vs SPP vs Fast R-CNN





Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
He et al, "Spatial pyramid pooling in deep convolutional networks for visual recognition", ECCV 2014
Girshick, "Fast R-CNN", ICCV 2015

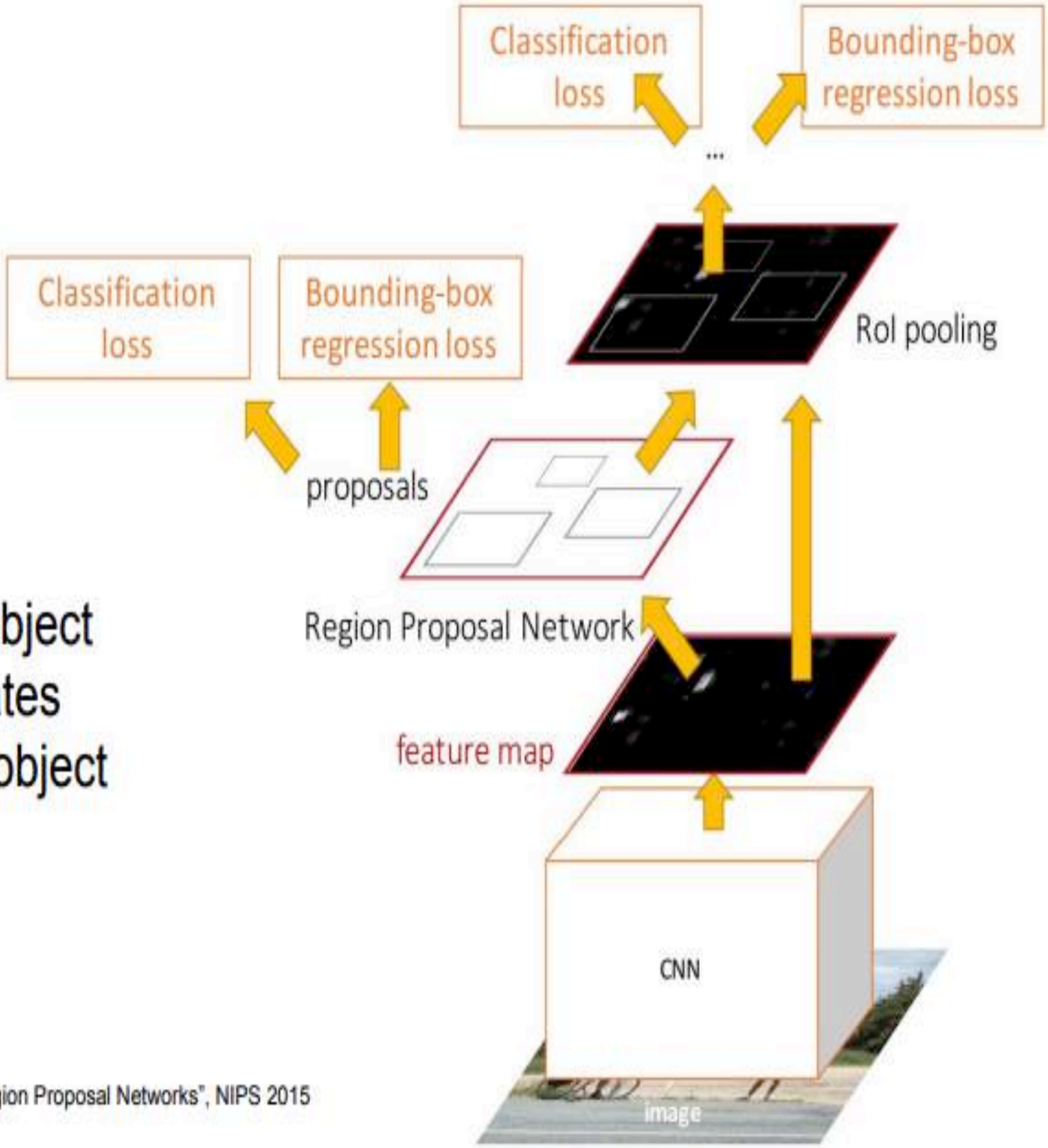
Fast<u>er</u> R-CNN:

Make CNN do proposals!

Insert **Region Proposal Network (RPN)** to predict proposals from features

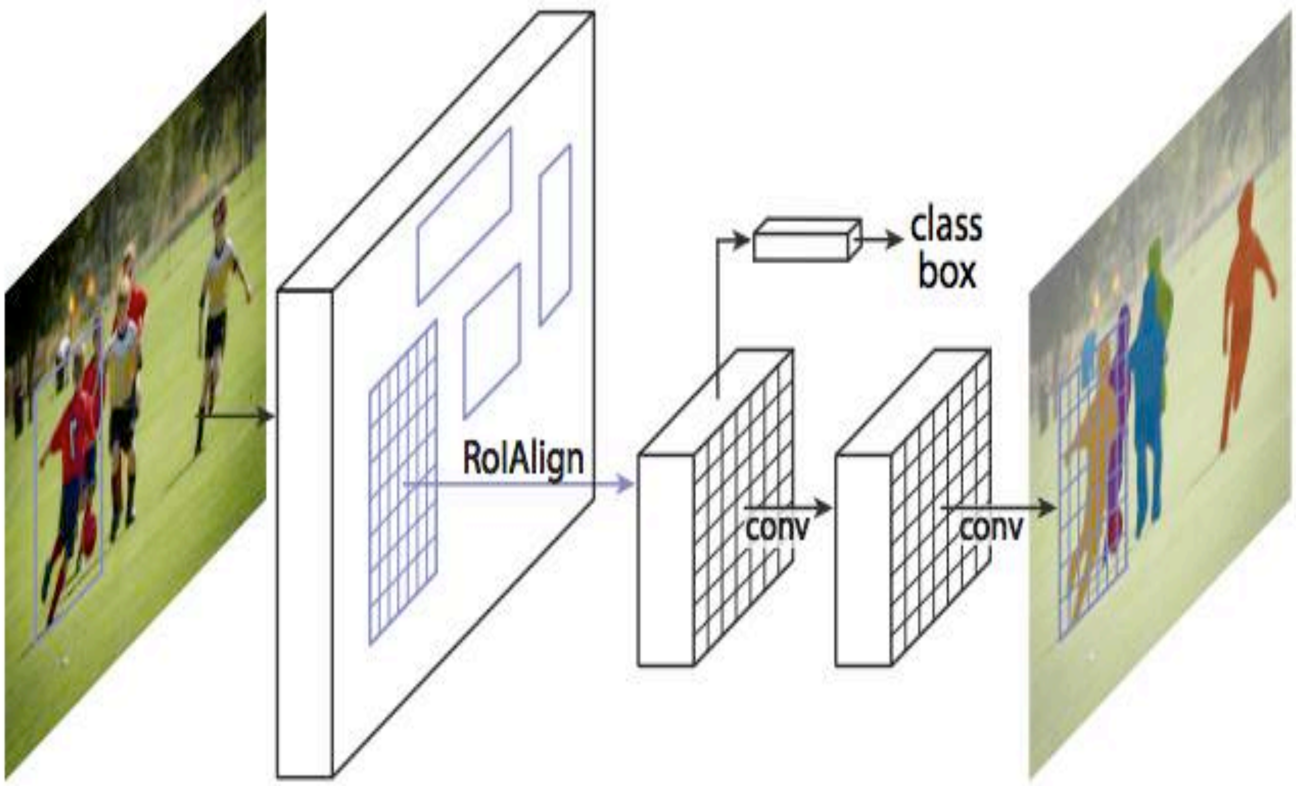
Jointly train with 4 losses:

1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates

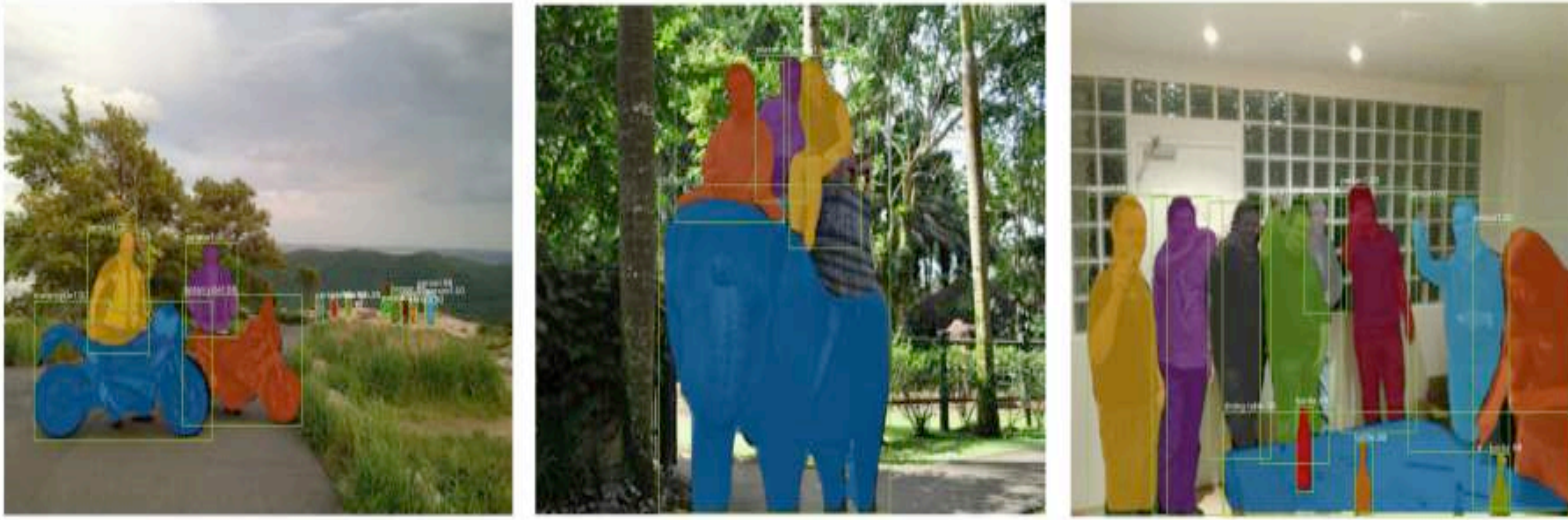


Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

Mask R-CNN



He, Gkioxari, Dollar and Girshick - Mask R-CNN, ICCV 2017 (Marr Prize).




FCIS
Mask R-CNN
umbrella
umbrella
person
person
person
person
skateboard
skateboard


brewers.com

YOLO

[Redmon, Divvala, Girshick, Farhadi - You Only Look Once, Real Time Object Detection, CVPR 2016.]

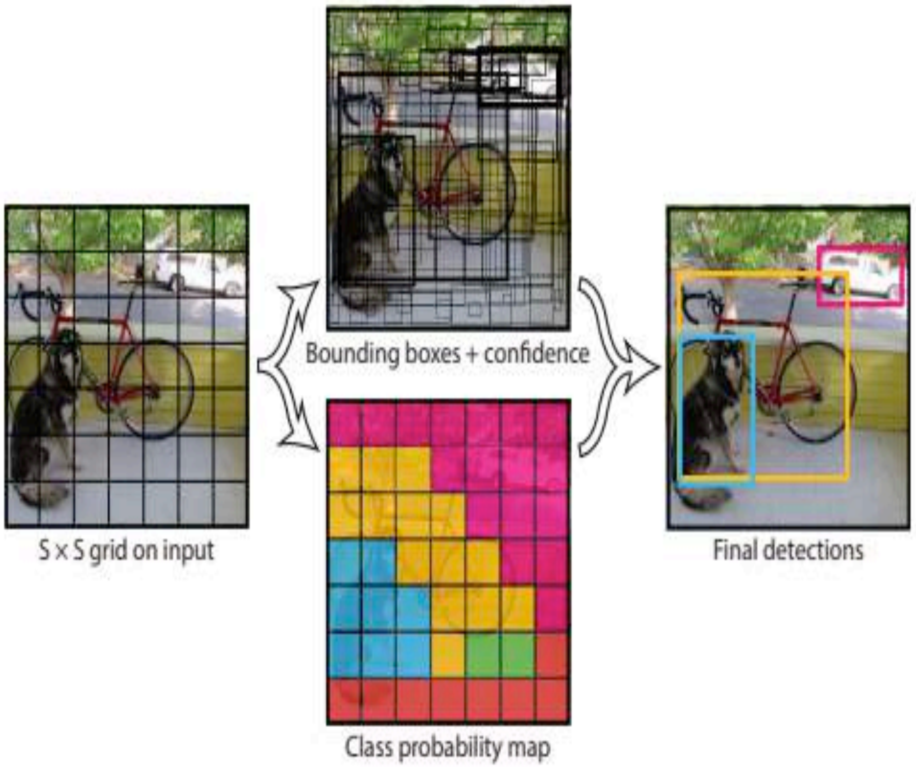
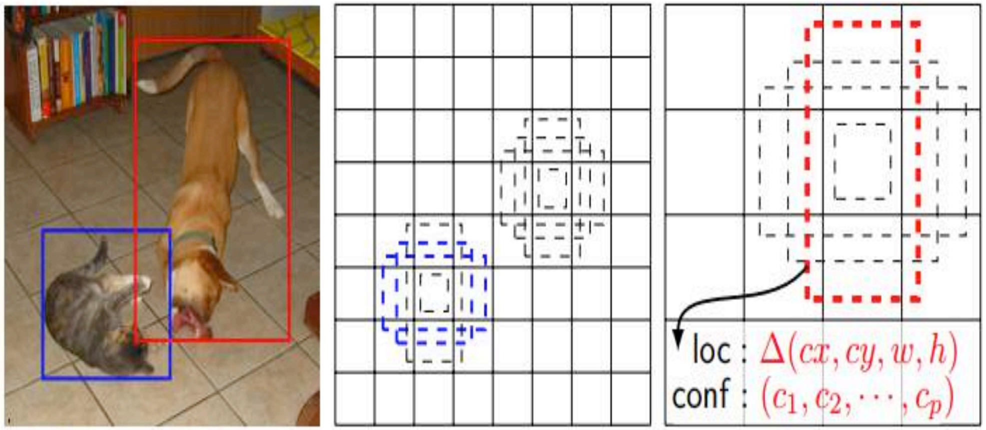


Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

SSD

[Liu, Anguelov, Erhan, Szegedy, Reed, Fu, Berg - SSD: Single Shot Box Detector, ECCV 2016.]



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

Fig. 1: **SSD framework.** (a) SSD only needs an input image and ground truth boxes for each object during training. In a convolutional fashion, we evaluate a small set (e.g. 4) of default boxes of different aspect ratios at each location in several feature maps with different scales (e.g. 8×8 and 4×4 in (b) and (c)). For each default box, we predict both the shape offsets and the confidences for all object categories ($(c_1, c_2, \cdots, c_p)$). At training time, we first match these default boxes to the ground truth boxes. For example, we have matched two default boxes with the cat and one with the dog, which are treated as positives and the rest as negatives. The model loss is a weighted sum between localization loss (e.g. Smooth L1 [6]) and confidence loss (e.g. Softmax).

Other Success Stories of Deep Learning

Today deep learning, in its several manifestations, is being applied in a variety of different domains besides computer vision, such as:

- Speech recognition
- Optical character recognition
- Natural language processing
- Autonomous driving
- Game playing (e.g., Google's AlphaGo)
- …

From Image to Text

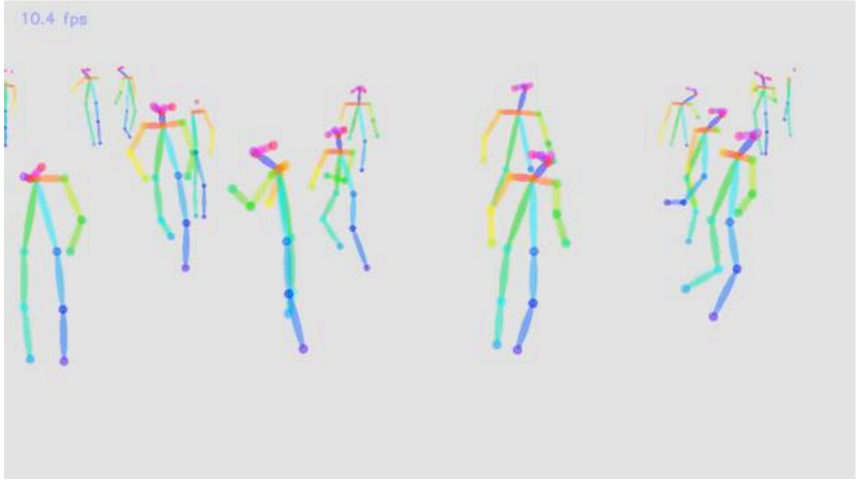
A person riding a motorcycle on a dirt road.

Two dogs play in the grass.

A group of young people playing a game of frisbee.

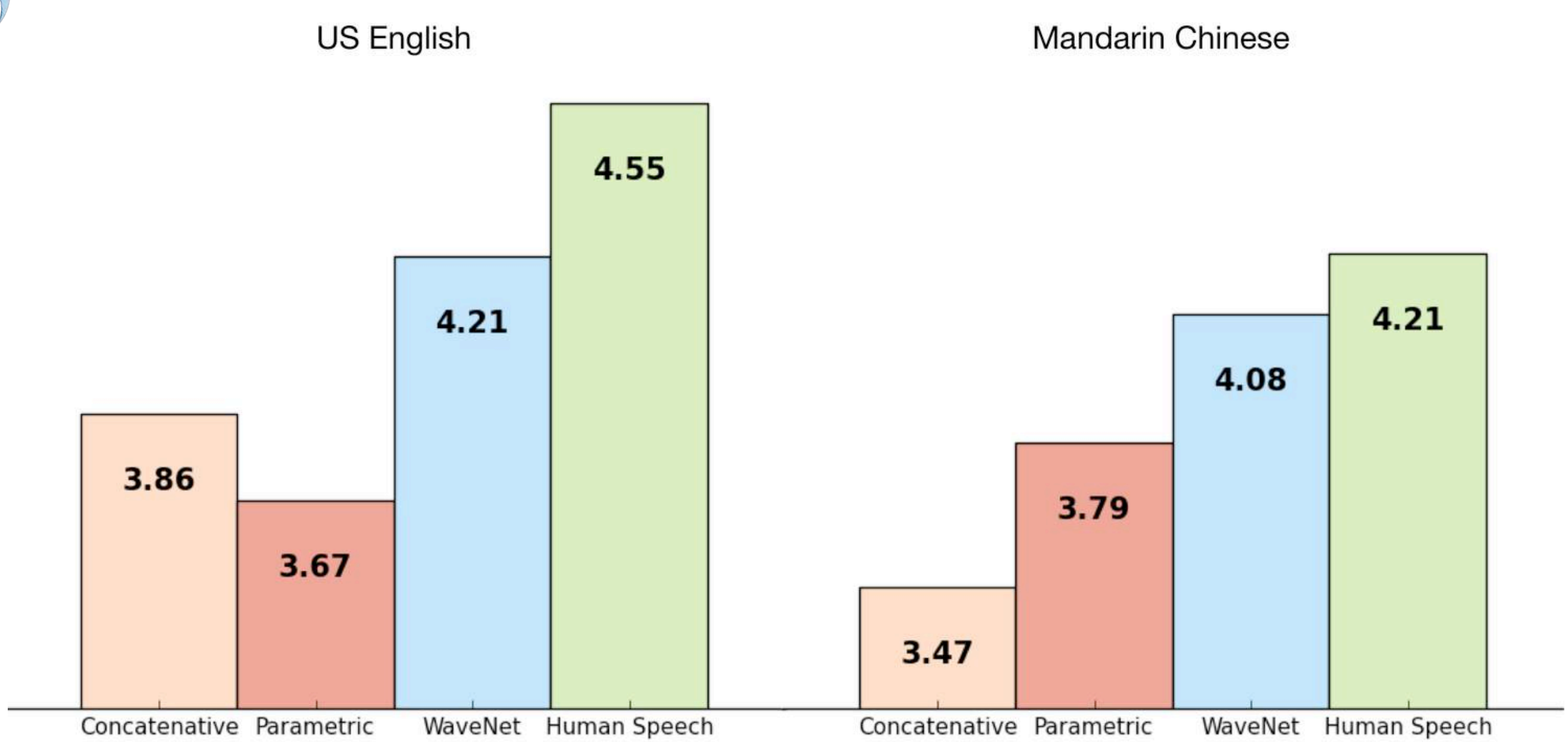
Two hockey players are fighting over the puck.

From Image to Video



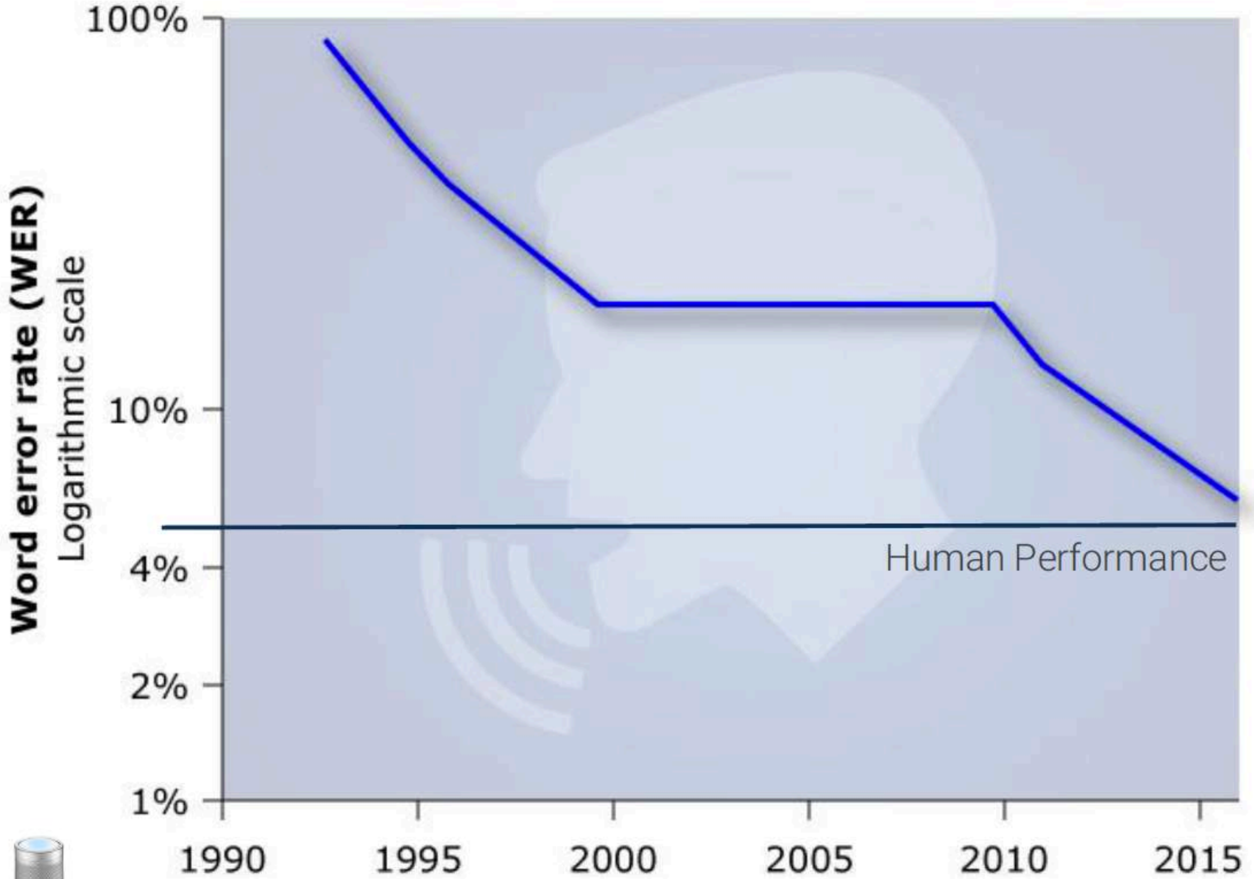
From Academy to the Market

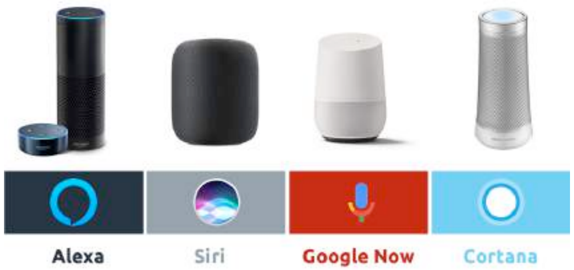
Machines that Talk



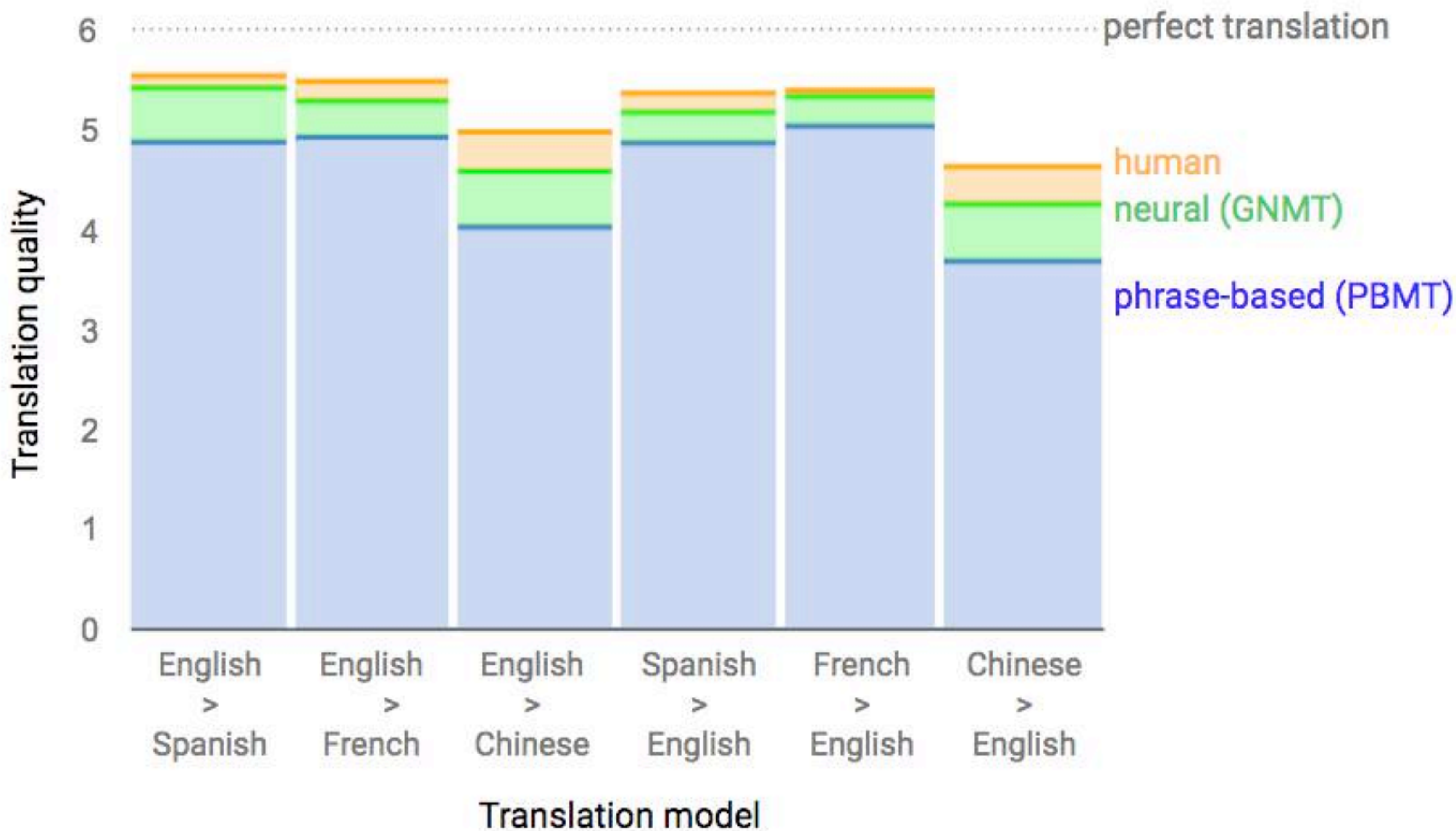
WaveNet can generate speech that reproduces any human voice and sounds more natural than the best text-to-speech systems available, reducing the gap with human performance by more than 50%.

Machines that Listen





Machines that Translate



References

- http://neuralnetworksanddeeplearning.com
- http://deeplearning.stanford.edu/tutorial/
- http://www.deeplearningbook.org/
- http://deeplearning.net/

Platforms:

- Theano
- Torch
- TensorFlow
- …