Deep neural networks

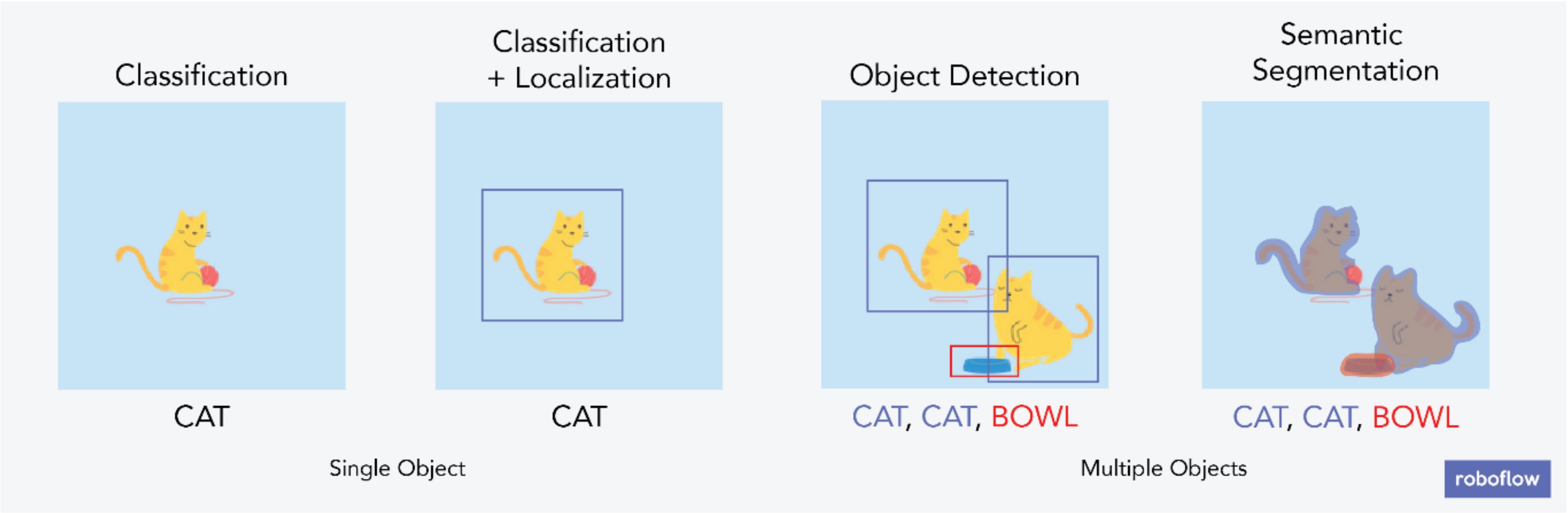
May 23rd, 2023

Mahdi Javanmardi

Amirkabir University of Technology

Many slides from Rob Fergus, Svetlana Lazebnik, Jia-Bin Huang, Derek Hoiem, Yong Jae Lee

Image classification



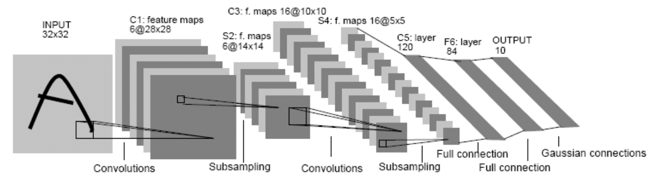
Discriminative classifiers

Nearest neighbor

10^6 examples

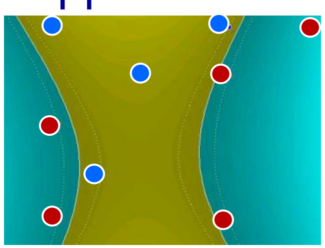
Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005...

Neural networks



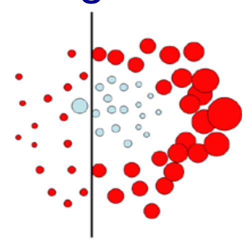
LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998
…

Support Vector Machines



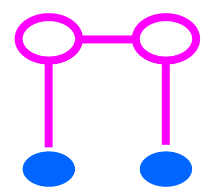
Guyon, Vapnik
Heisele, Serre, Poggio, 2001,…

Boosting



Viola, Jones 2001,
Torralba et al. 2004,
Opelt et al. 2006,…

Conditional Random Fields



McCallum, Freitag, Pereira 2000; Kumar, Hebert 2003
…

Slide adapted from Antonio Torralba

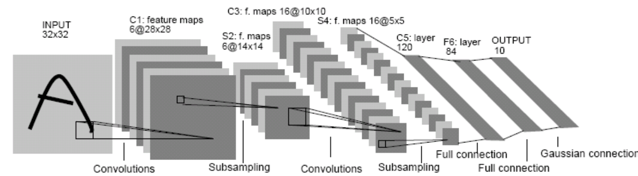
Discriminative classifiers

Nearest neighbor

10^6 examples

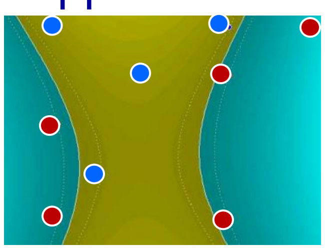
Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005...

Neural networks



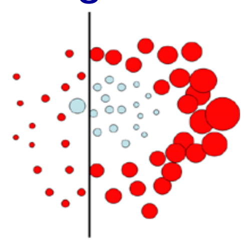
LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998
…

Support Vector Machines



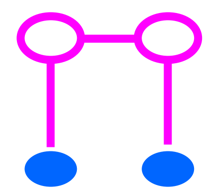
Guyon, Vapnik
Heisele, Serre, Poggio, 2001,…

Boosting



Viola, Jones 2001,
Torralba et al. 2004,
Opelt et al. 2006,…

Conditional Random Fields



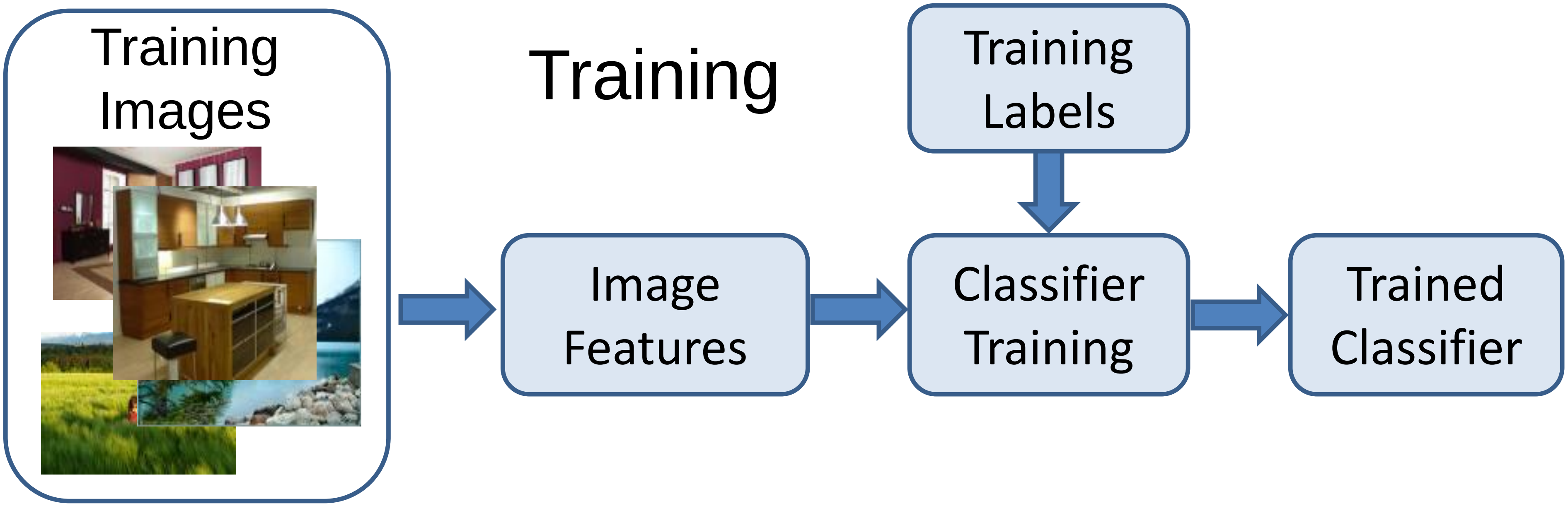
McCallum, Freitag, Pereira 2000; Kumar, Hebert 2003
…

Slide adapted from Antonio Torralba

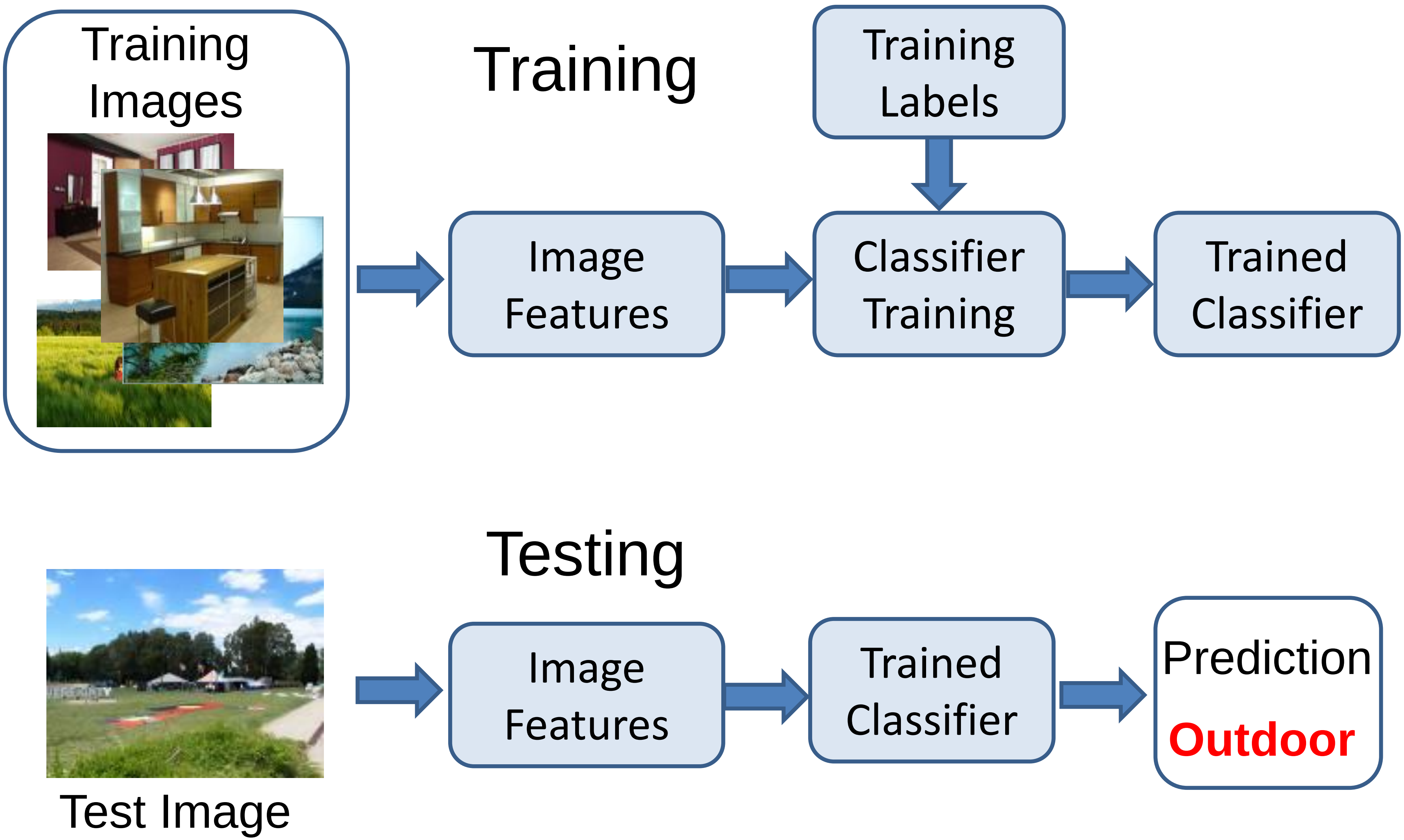
Outline

- Deep Neural Networks
- Convolutional Neural Networks (CNNs)

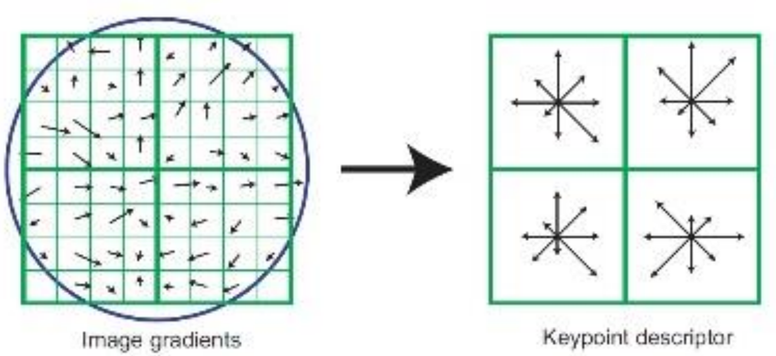
Traditional Image Categorization: Training phase



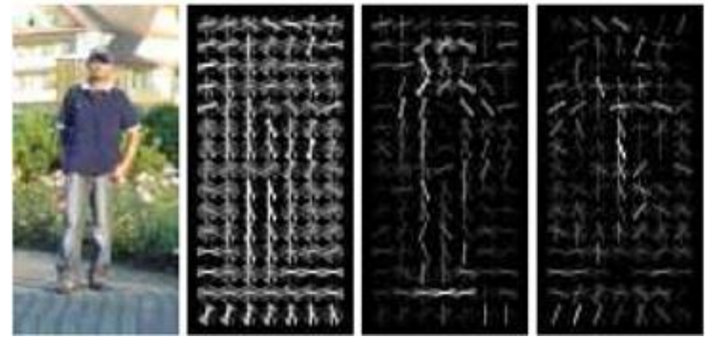
Traditional Image Categorization: Testing phase



Features have been key..



SIFT [

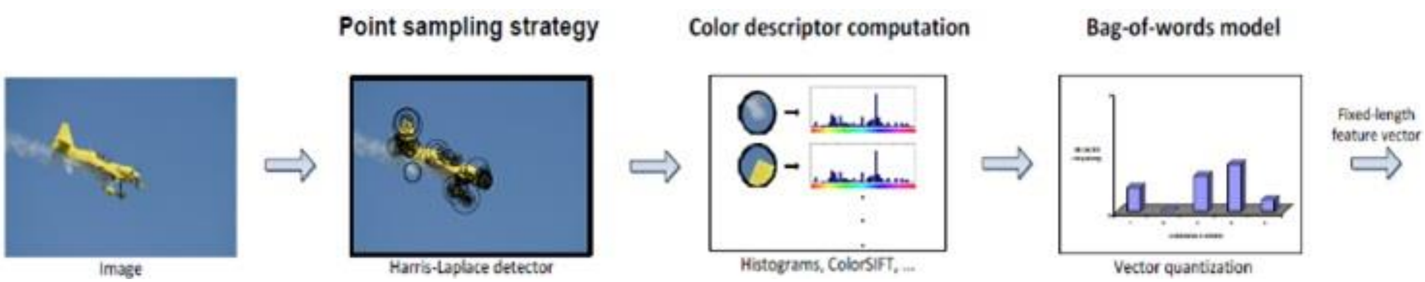


PR 05]

Hand-crafted

SPM [Lazebnik et al. CVPR 06]

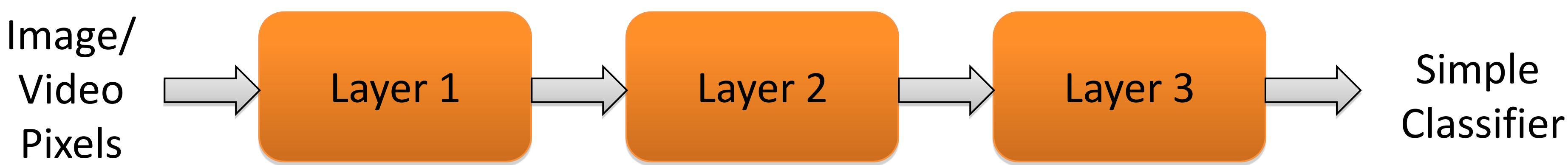
DPM [Felzenszwalb et al. PAMI 10]



Color Descriptor [Van De Sande et al. PAMI 10]

What about learning the features?

- Learn a *feature hierarchy* all the way from pixels to classifier
- Each layer extracts features from the output of previous layer
- Layers have (nearly) the same structure
- Train all layers jointly ("end-to-end")

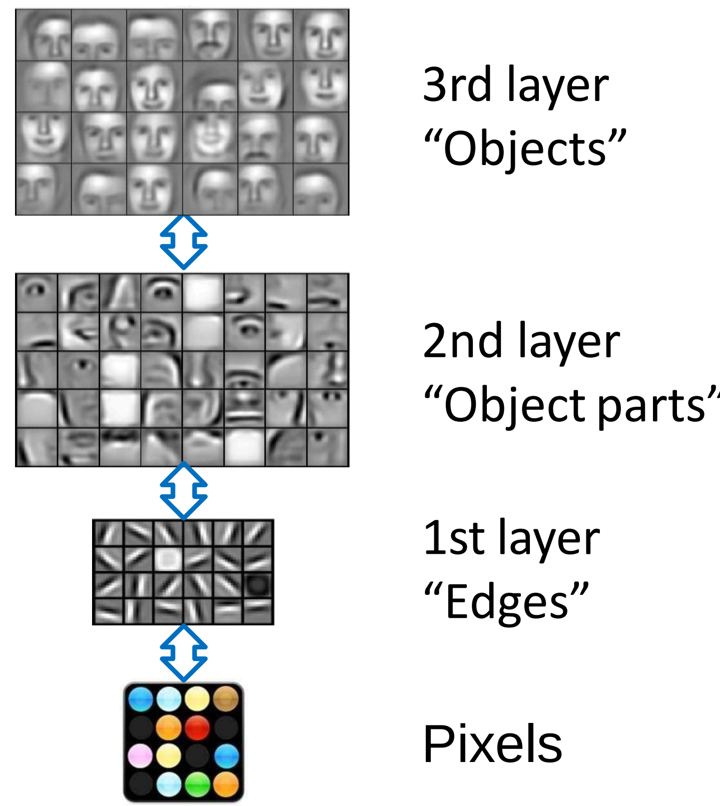


Learning Feature Hierarchy

Goal: Learn **useful higher-level features** from images

Input data

Feature representation

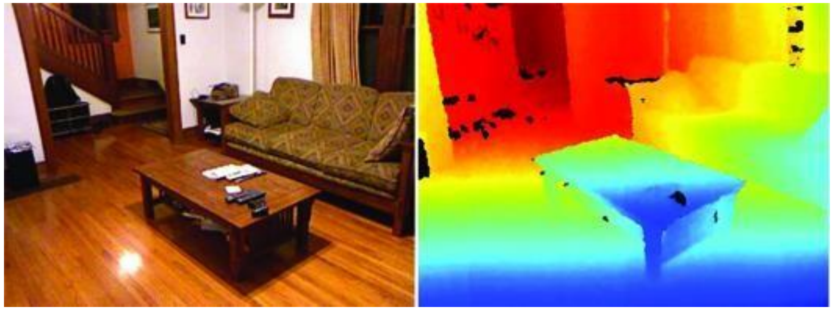


Lee et al., ICML 2009;
CACM 2011

Slide: Rob Fergus

Learning Feature Hierarchy

- Better performance
- Other domains (unclear how to hand engineer):
  - Kinect
  - Video
  - Multi spectral

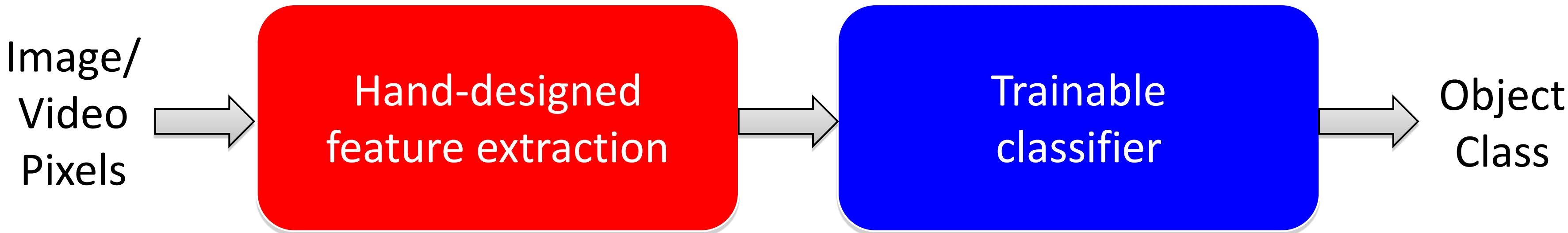


- Feature computation time
  - Dozens of features now regularly used
  - Getting prohibitive for large datasets (10’s sec /image)

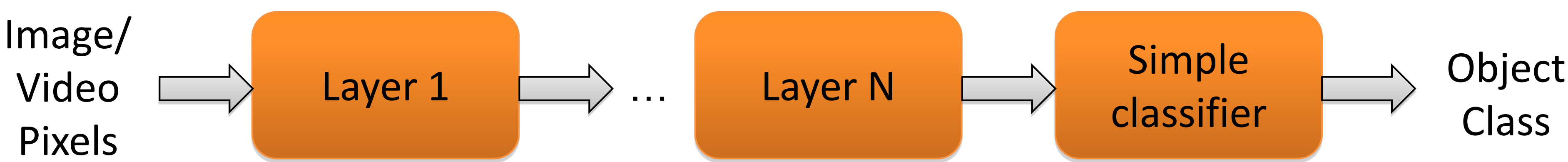
Slide: R. Fergus

“Shallow” vs. “deep” architectures

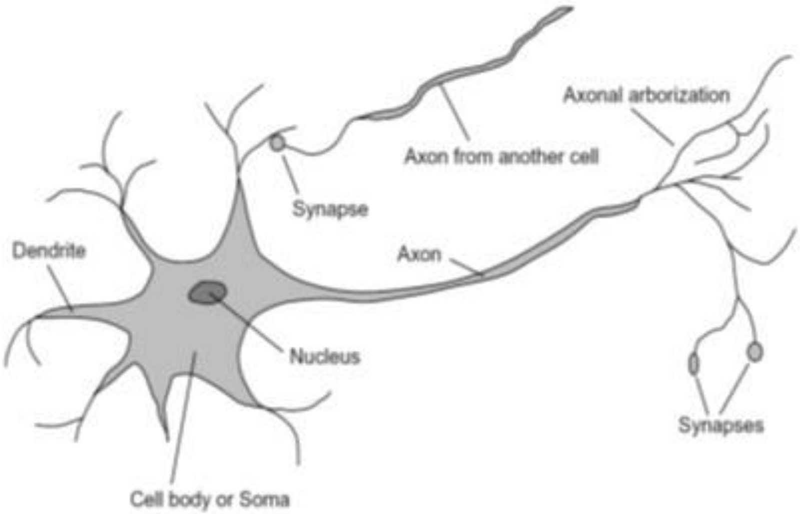
Traditional recognition: “Shallow” architecture



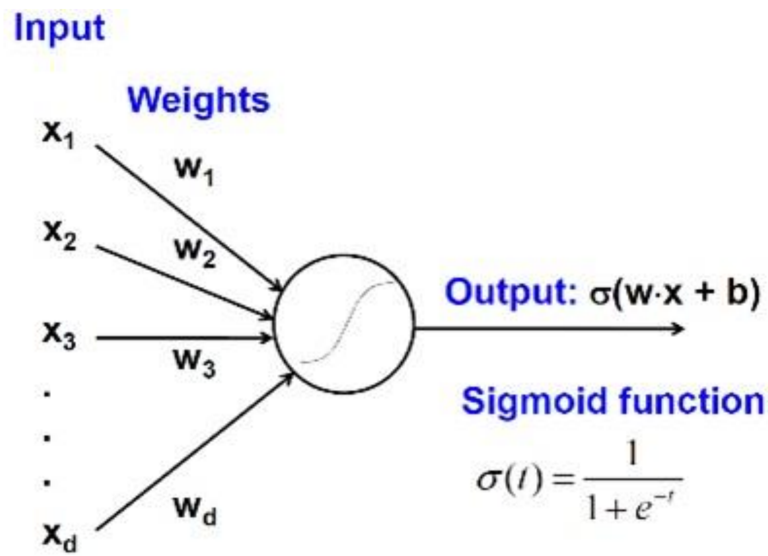
Deep learning: “Deep” architecture



Biological neuron and Perceptrons



A biological neuron



An artificial neuron (Perceptron)
- a linear classifier

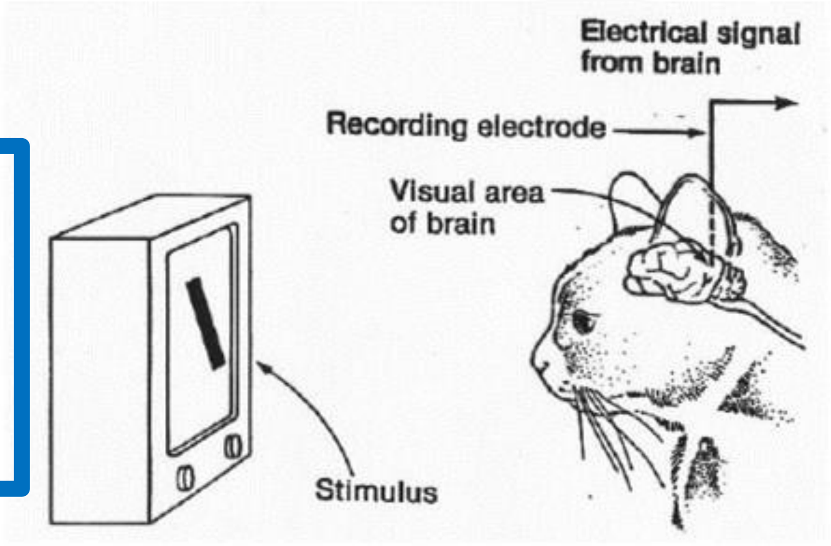
Simple, Complex, and Hyper-complex cells

David H. Hubel and Torsten Wiesel

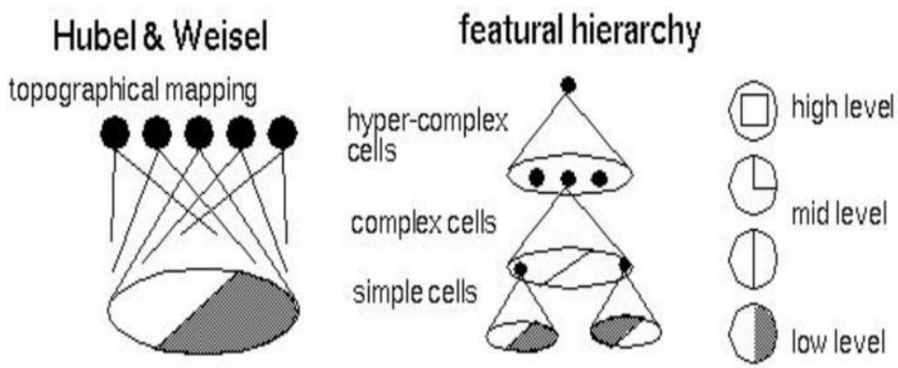
video

Suggested a **hierarchy** of **feature detectors** in the visual cortex, with higher level features responding to patterns of activation in lower level cells, and propagating activation upwards to still higher level cells.

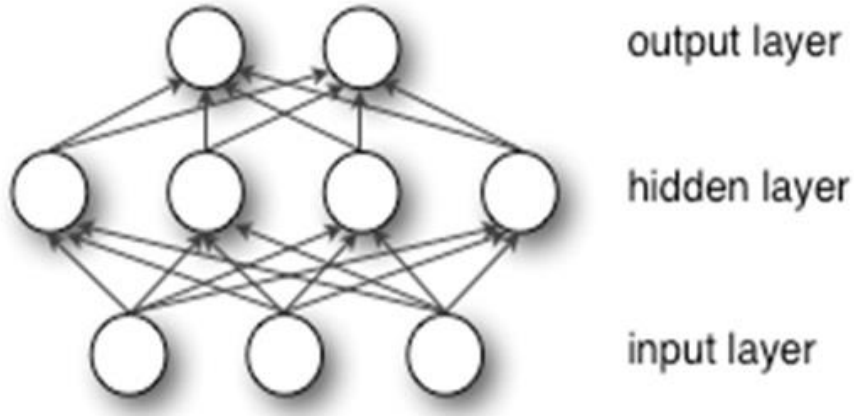
David Hubel's Eye, Brain, and Vision



Hubel/Wiesel Architecture and Multi-layer Neural Network



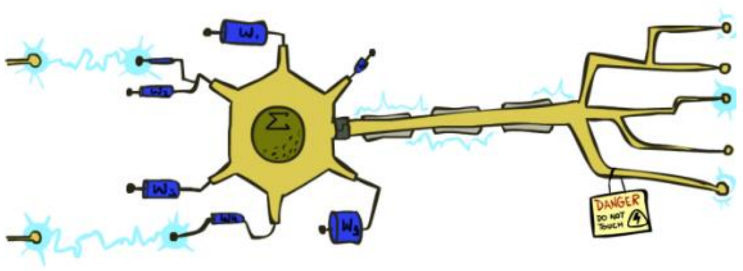
Hubel and Weisel's architecture



Multi-layer Neural Network
- A *non-linear* classifier

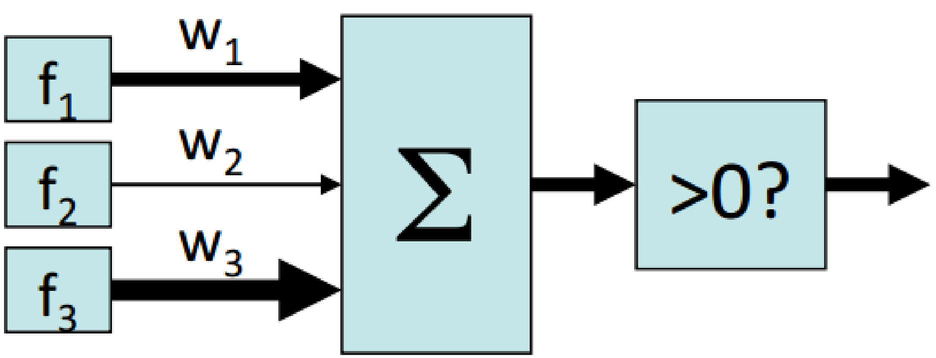
Neuron: Linear Perceptron

- Inputs are feature values
- Each feature has a weight
- Sum is the activation



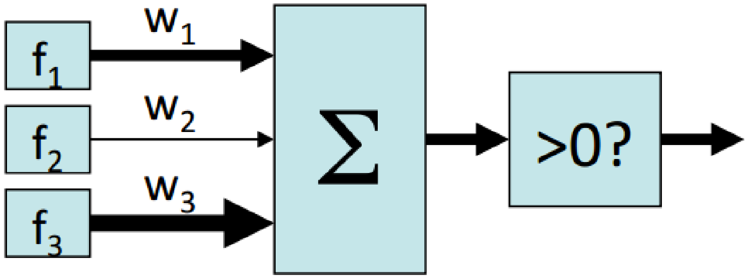
$$\text{activation}_w(x) = \sum_i w_i \cdot f_i(x) = w \cdot f(x)$$

- If the activation is:
  - Positive, output +1
  - Negative, output -1



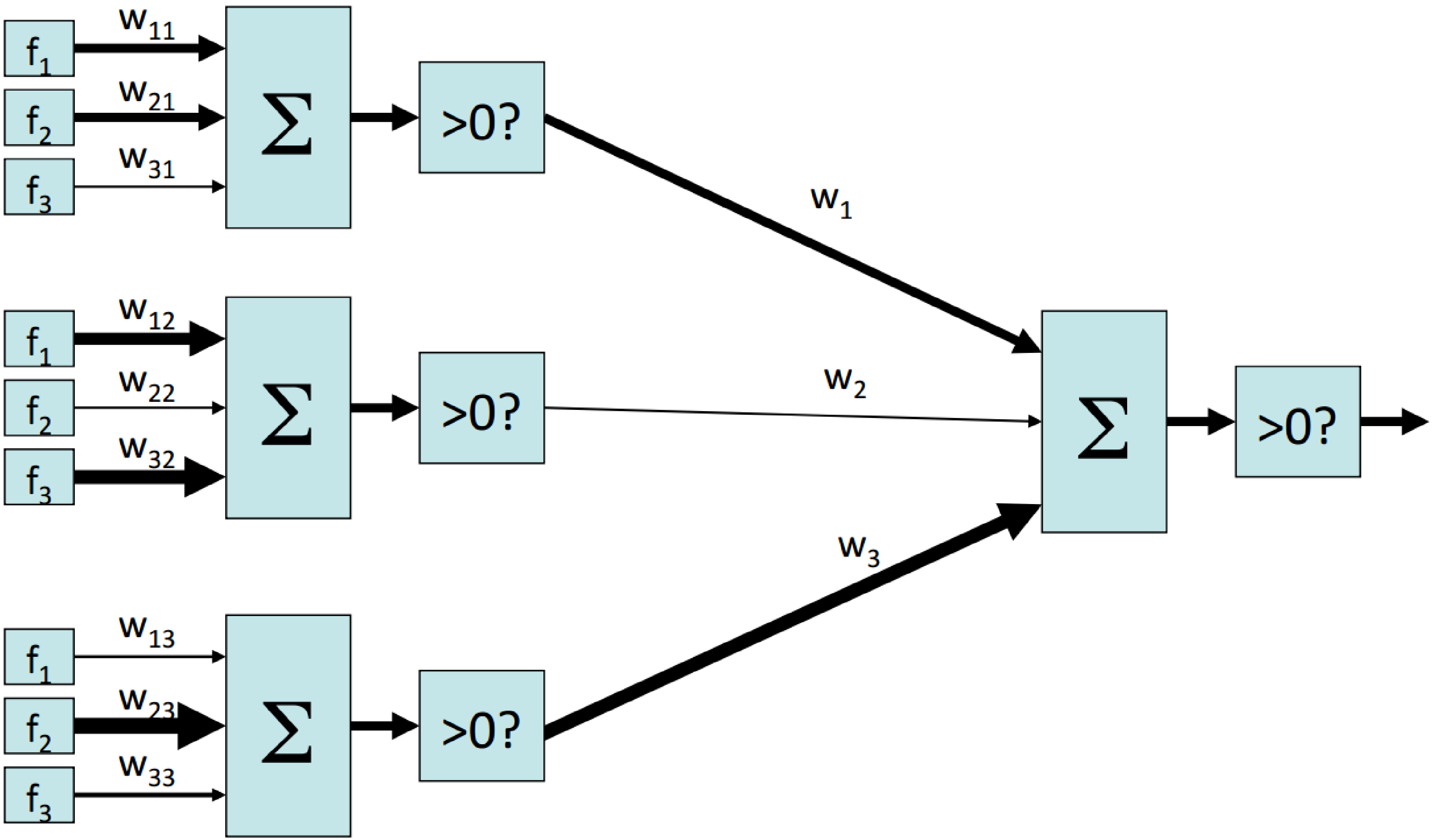
Slide credit: Pieter Abeel and Dan Klein

Two-layer perceptron network



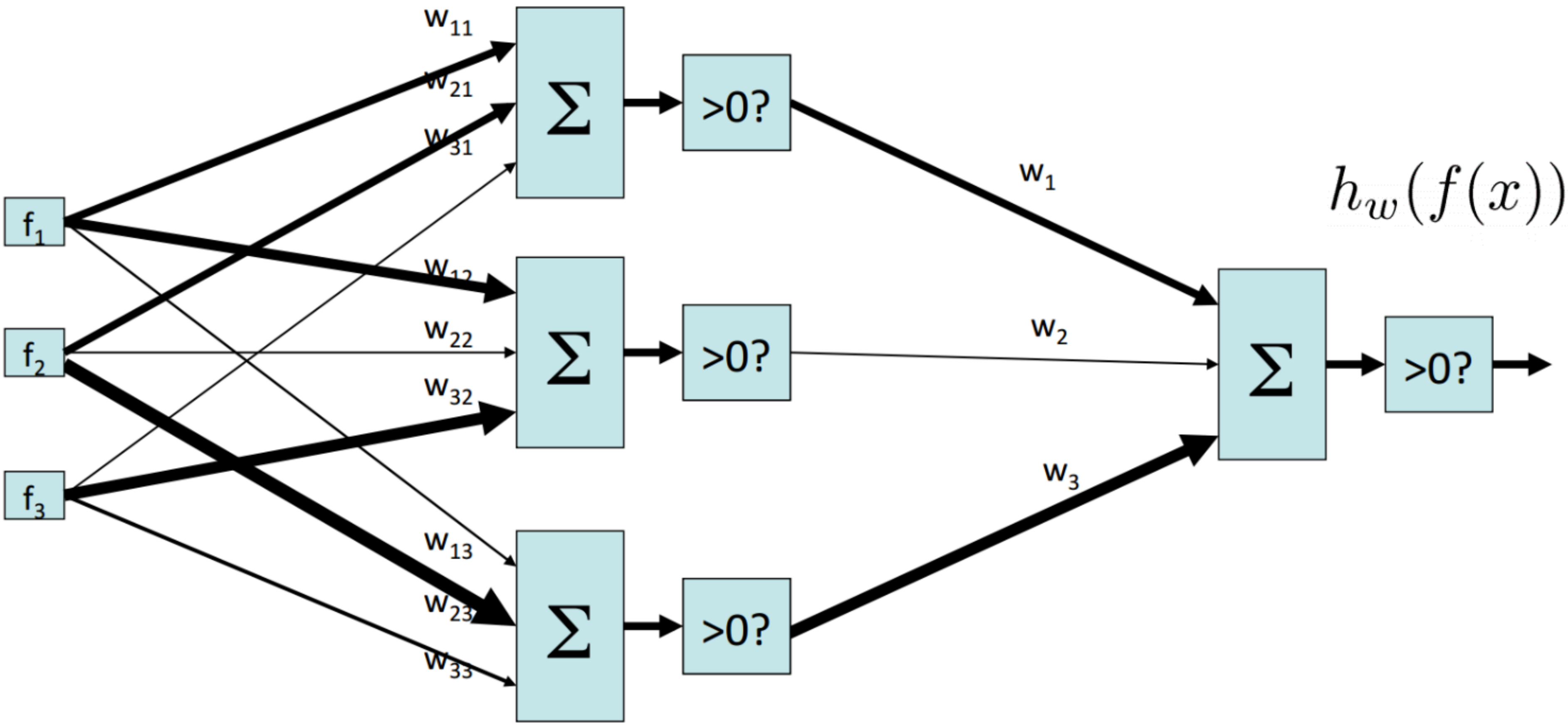
Slide credit: Pieter Abeel and Dan Klein

Two-layer perceptron network



Slide credit: Pieter Abeel and Dan Klein

Two-layer perceptron network



Slide credit: Pieter Abeel and Dan Klein

Learning w

- Training examples

$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})$$

- Objective: a misclassification loss

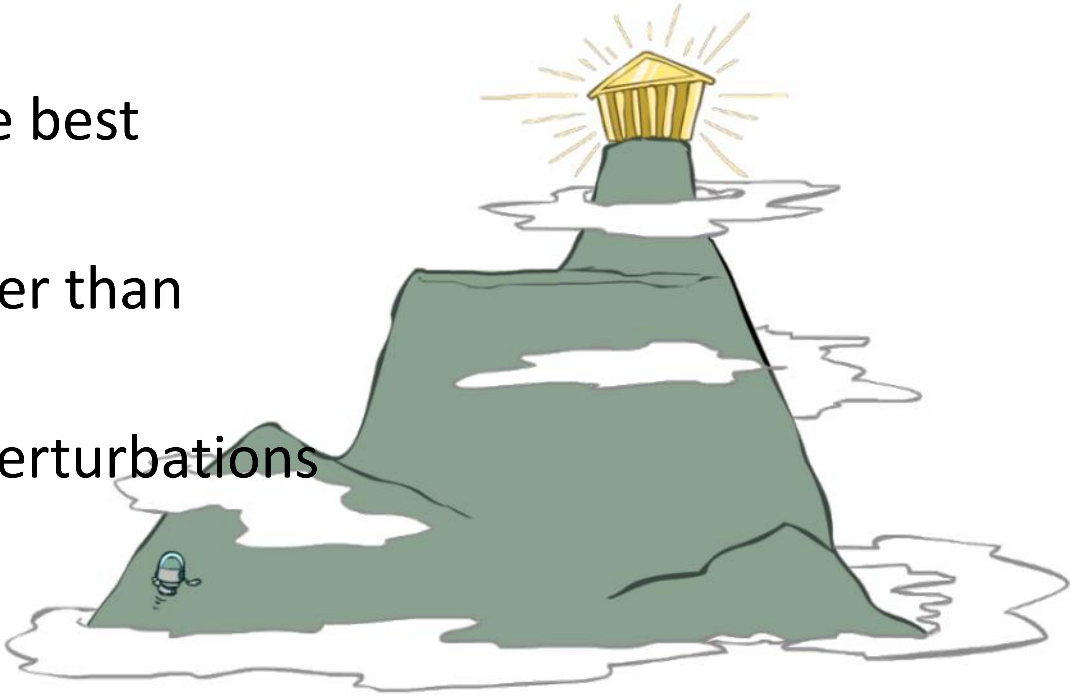
$$\min_w \sum_{i=1}^{m} \left(y^{(i)} - h_w(f(x^{(i)})) \right)^2$$

- Procedure:
  - Gradient descent / hill climbing

Slide credit: Pieter Abeel and Dan Klein

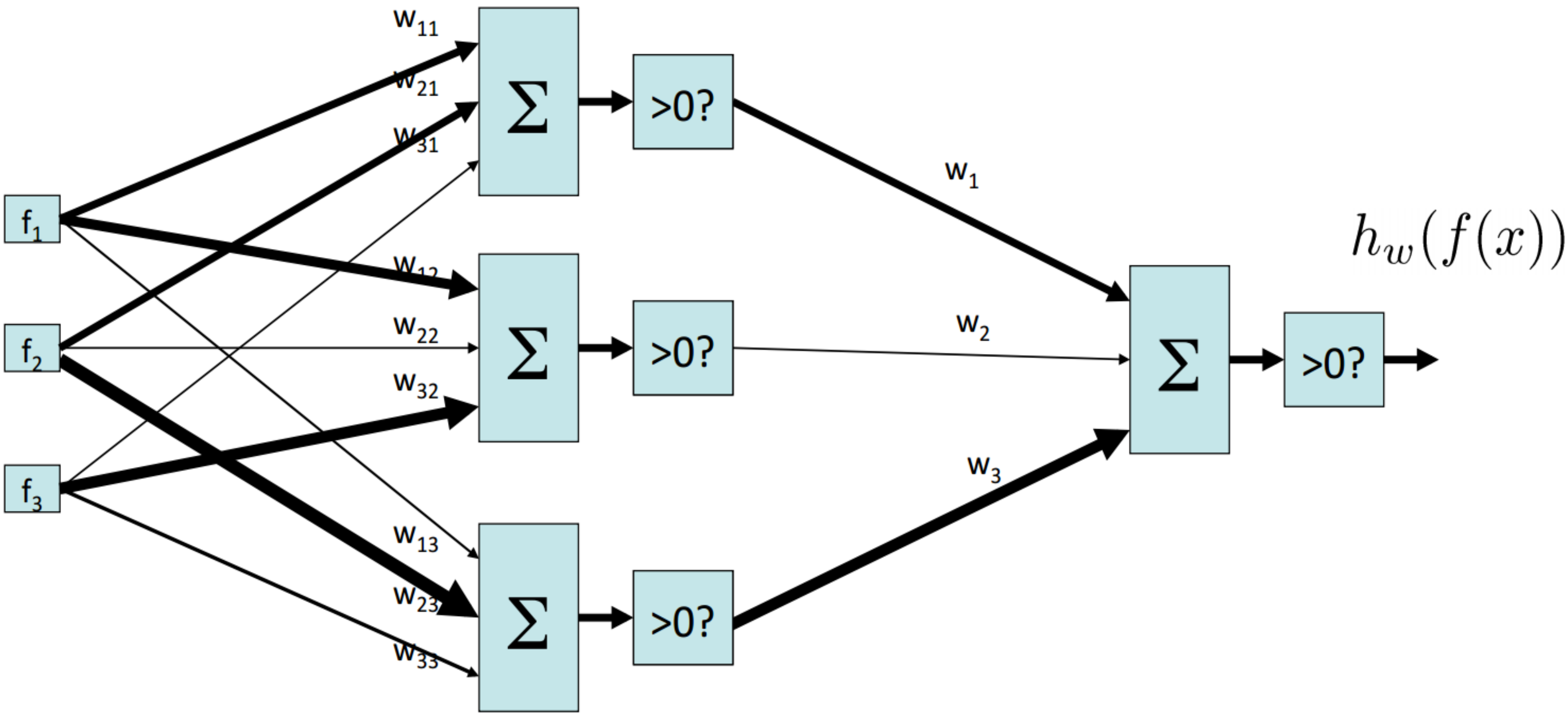
Hill climbing

- Simple, general idea:
  - Start wherever
  - Repeat: move to the best neighboring state
  - If no neighbors better than current, quit
  - Neighbors = small perturbations of w
- What's bad?
  - Optimal?



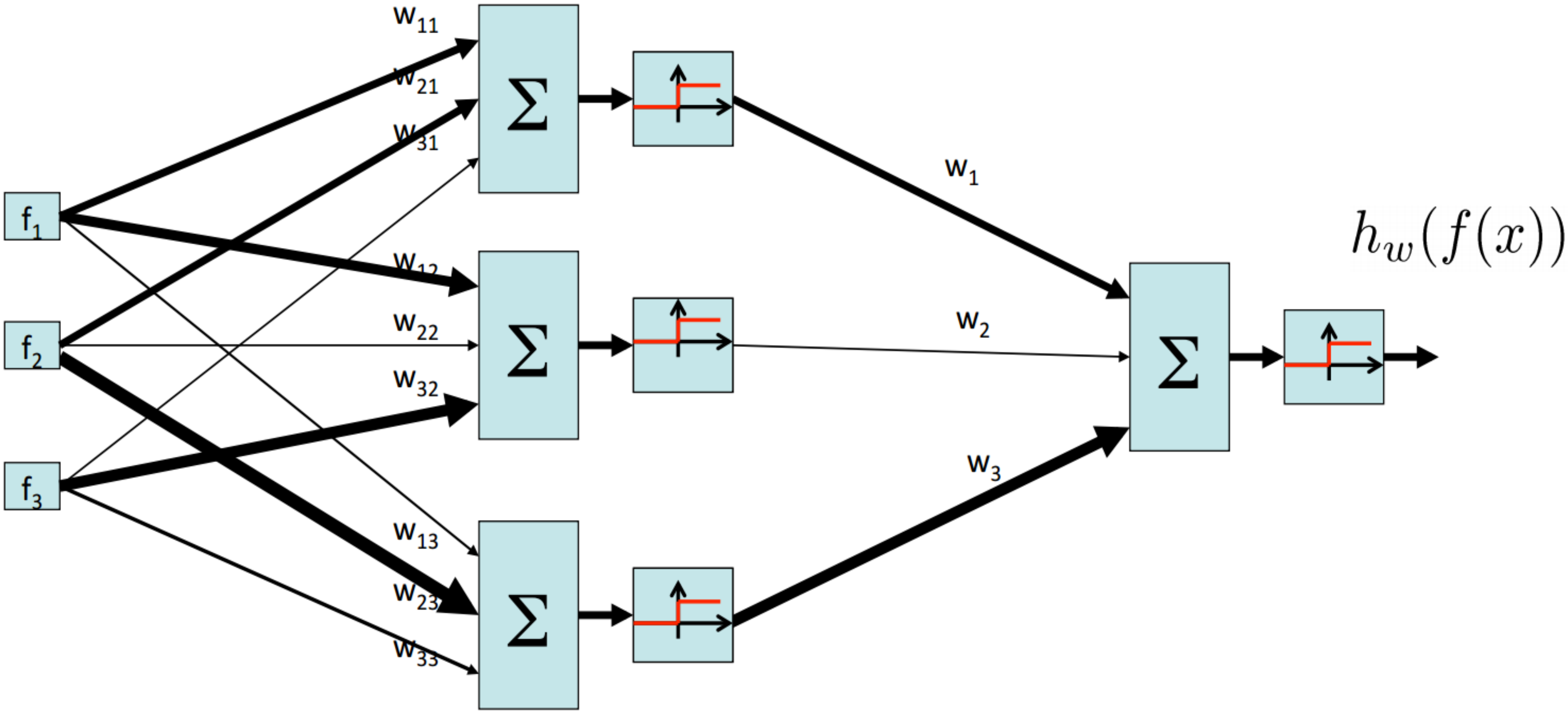
Slide credit: Pieter Abeel and Dan Klein

Two-layer perceptron network



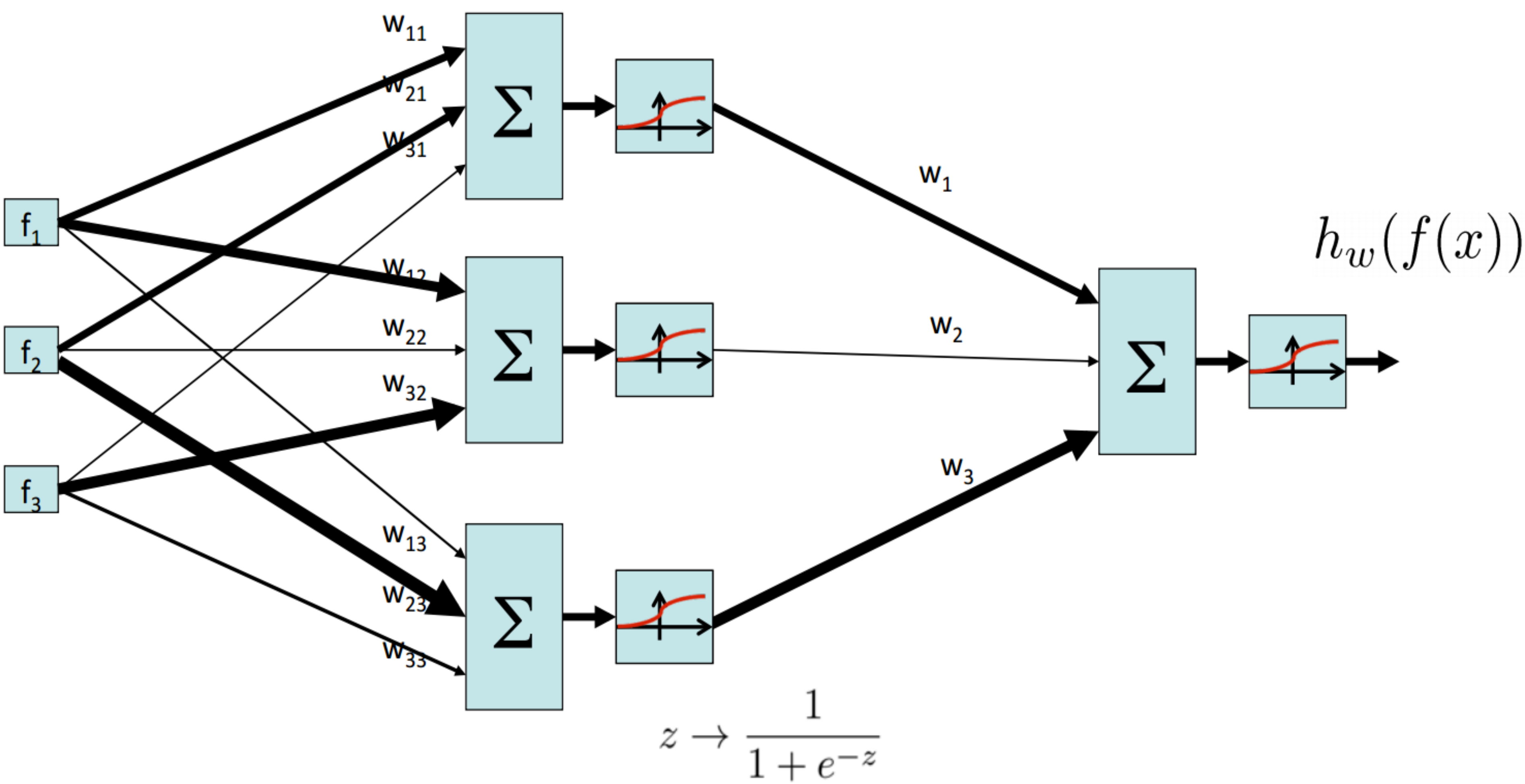
Slide credit: Pieter Abeel and Dan Klein

Two-layer perceptron network



Slide credit: Pieter Abeel and Dan Klein

Two-layer neural network



Slide credit: Pieter Abeel and Dan Klein

Neural network properties

- **Theorem (Universal function approximators):** A two-layer network with a sufficient number of neurons can approximate any continuous function to any desired accuracy

- **Practical considerations:**
  - Can be seen as learning the features
  - Large number of neurons
    - Danger for overfitting
  - Hill-climbing procedure can get stuck in bad local optima

Approximation by Superpositions of Sigmoidal Function,1989

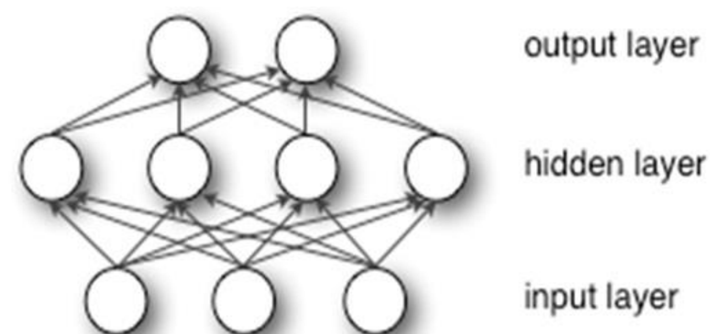
Slide credit: Pieter Abeel and Dan Klein

Multi-layer Neural Network

- A non-linear classifier
- **Training:** find network weights **w** to minimize the error between true training labels and estimated labels

$$E(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - f_{\mathbf{w}}(\mathbf{x}_i)\right)^2$$

- Minimization can be done by gradient descent provided *f* is differentiable
- This training method is called **back-propagation**



Outline

- Deep Neural Networks
- **Convolutional Neural Networks (CNNs)**

Convolutional Neural Networks (CNN, ConvNet, DCN)

- CNN = a multi-layer neural network with
  - **Local** connectivity:
    - Neurons in a layer are only connected to a small region of the layer before it
  - **Share** weight parameters across spatial positions:
    - Learning shift-invariant filter kernels

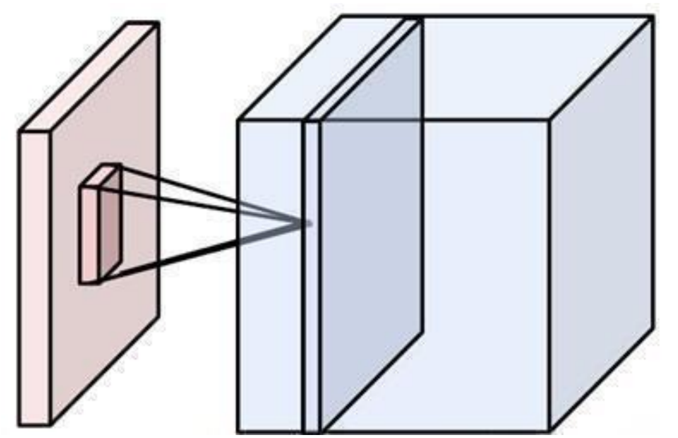


Image credit: A. Karpathy

Neocognitron [Fukushima, Biological Cybernetics 1980]

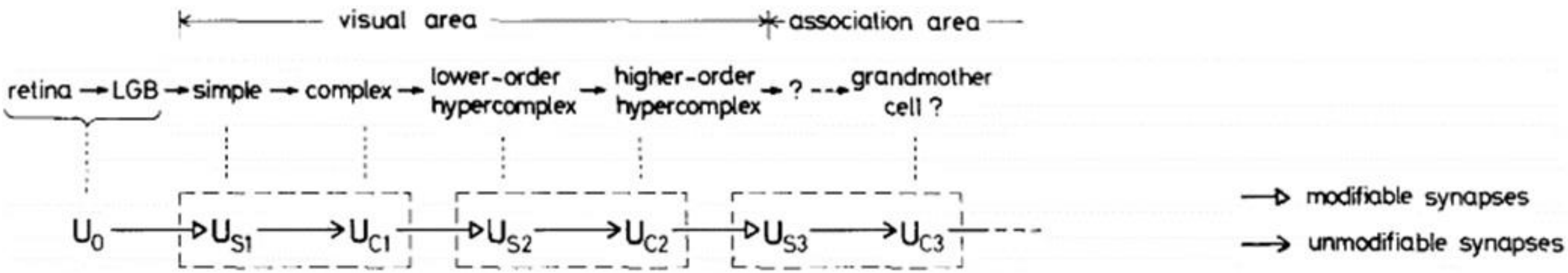
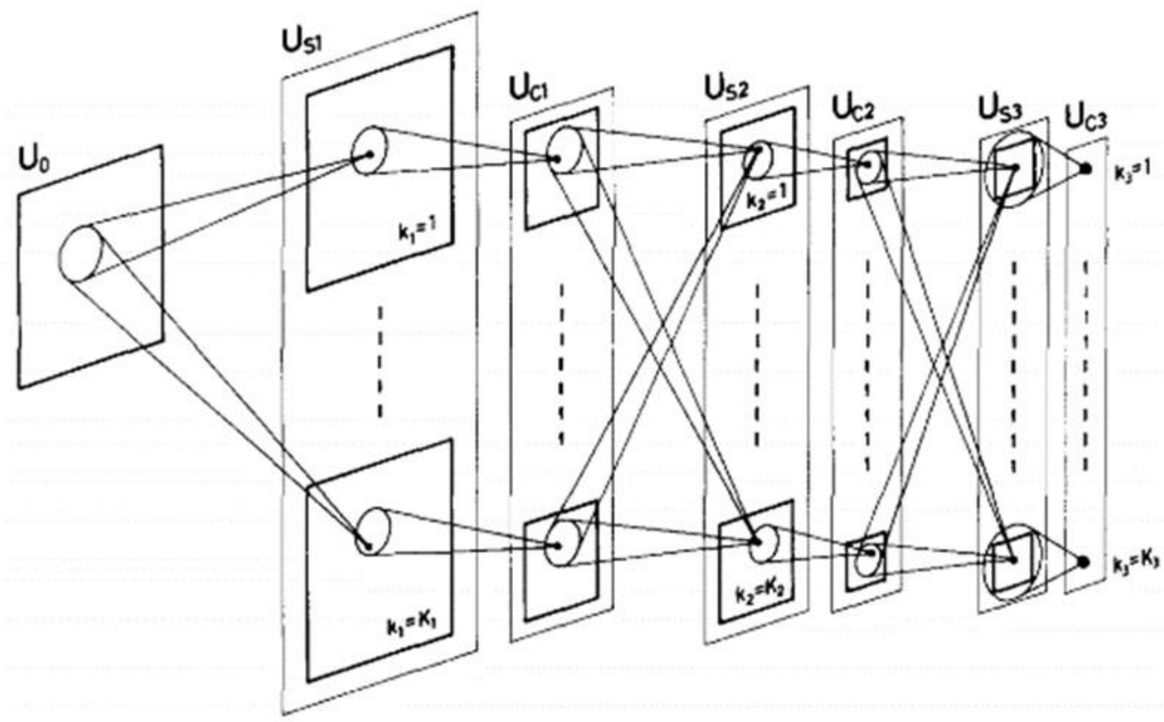


Fig. 1. Correspondence between the hierarchy model by Hubel and Wiesel, and the neural network of the neocognitron

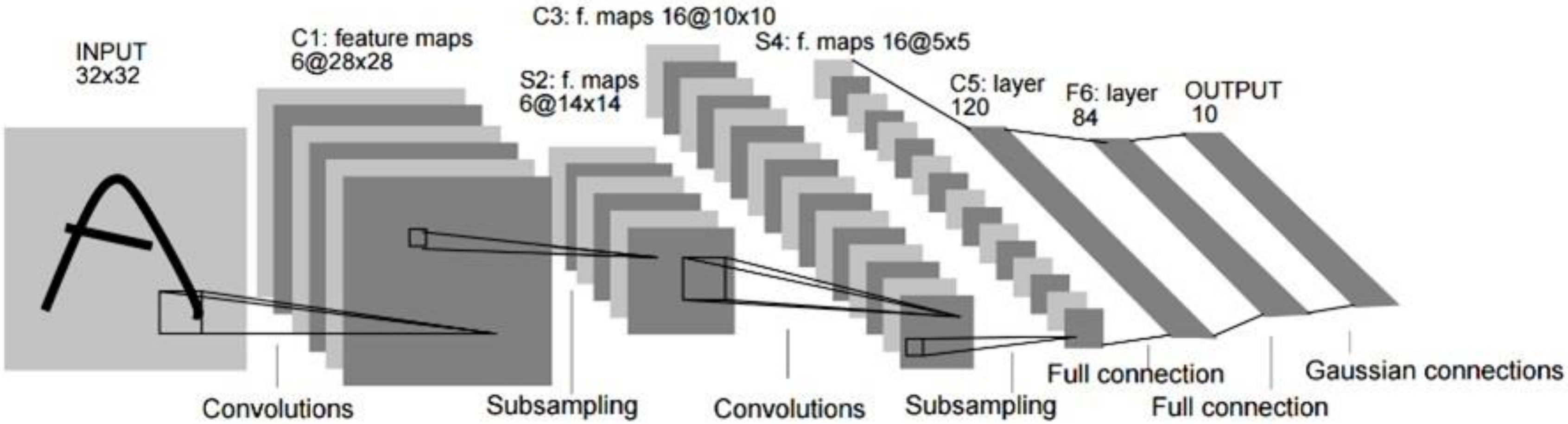


Deformation-Resistant Recognition

S-cells: (simple)
- extract local features

C-cells: (complex)
- allow for positional errors

LeNet [LeCun et al. 1998]

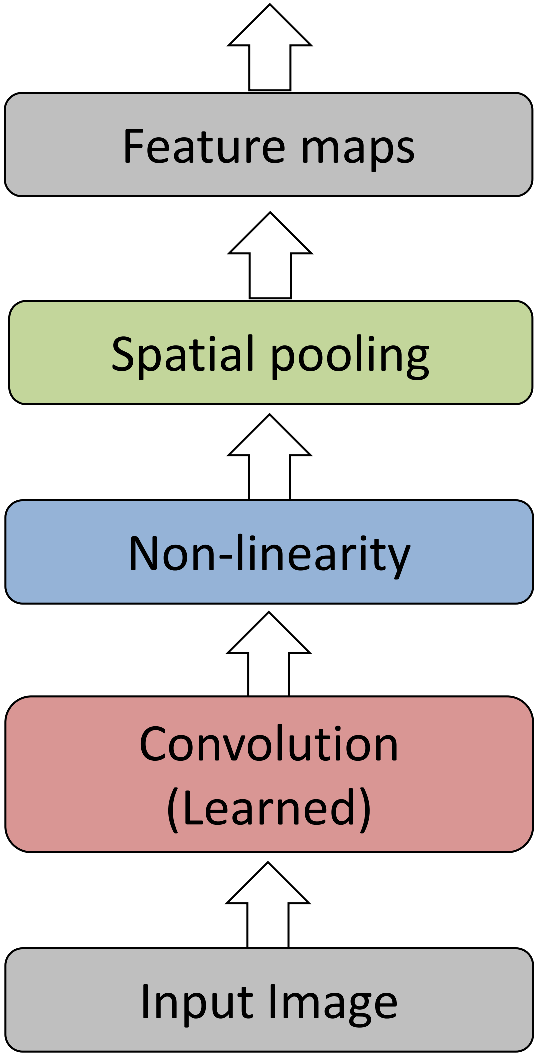


- Stack multiple stages of feature extractors
- Higher stages compute more global, more invariant features
- Classification layer at the end

Gradient-based learning applied to document recognition [LeCun, Bottou, Bengio, Haffner 1998]

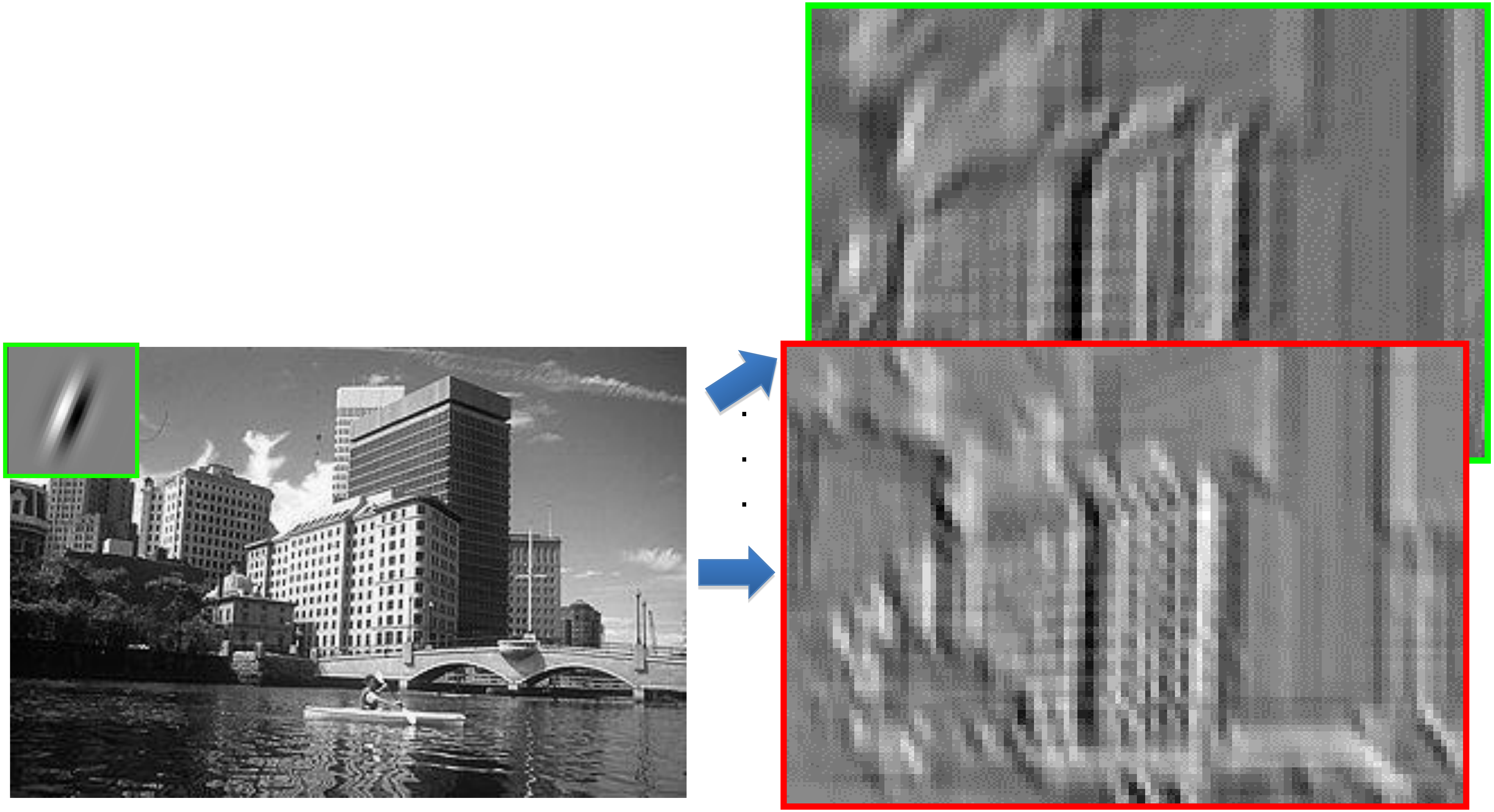
LeNet-1 from 1993

Convolutional Neural Networks



What is a Convolution?

- Weighted moving sum

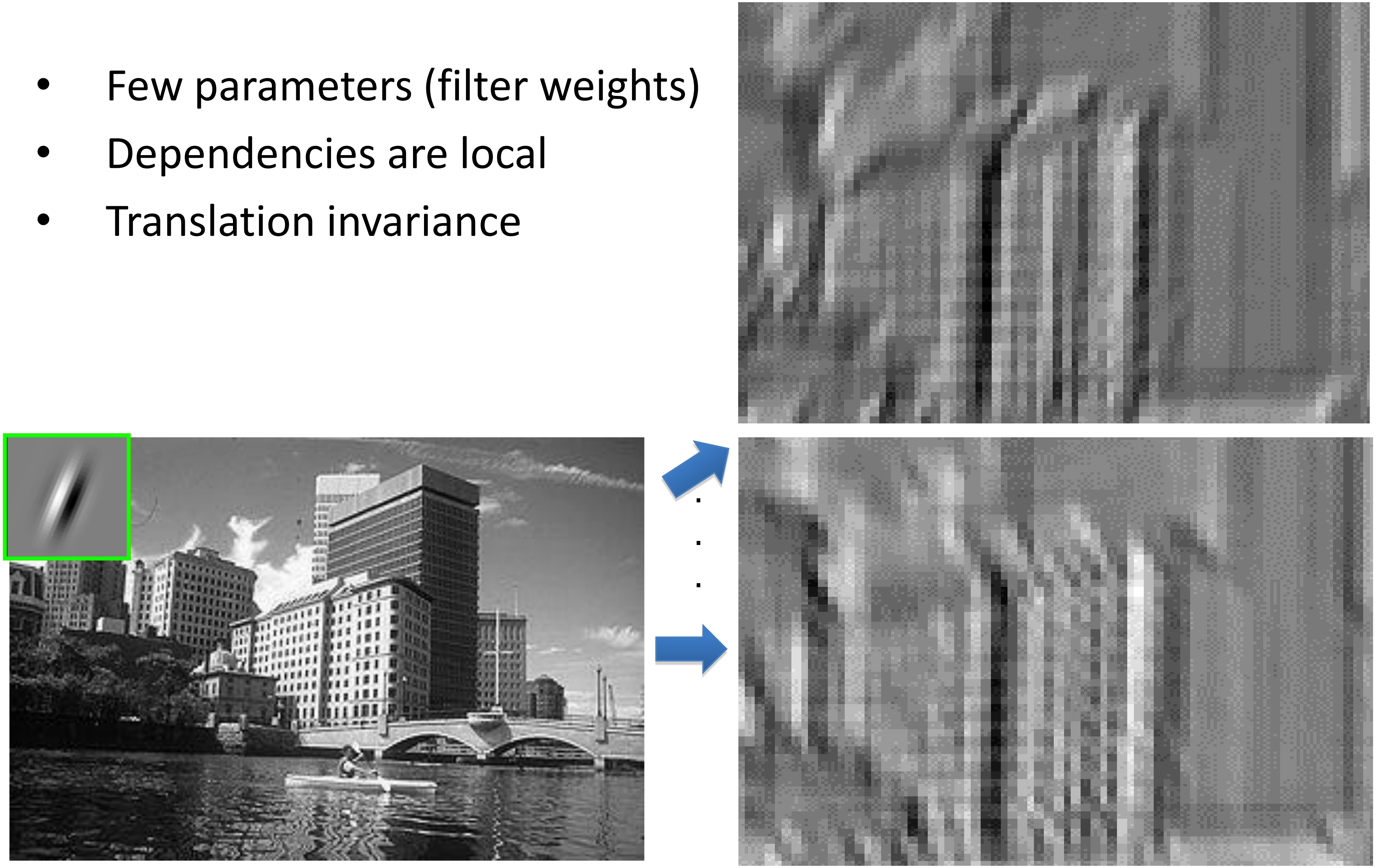


Input

Feature Activation Map

Why Convolution?

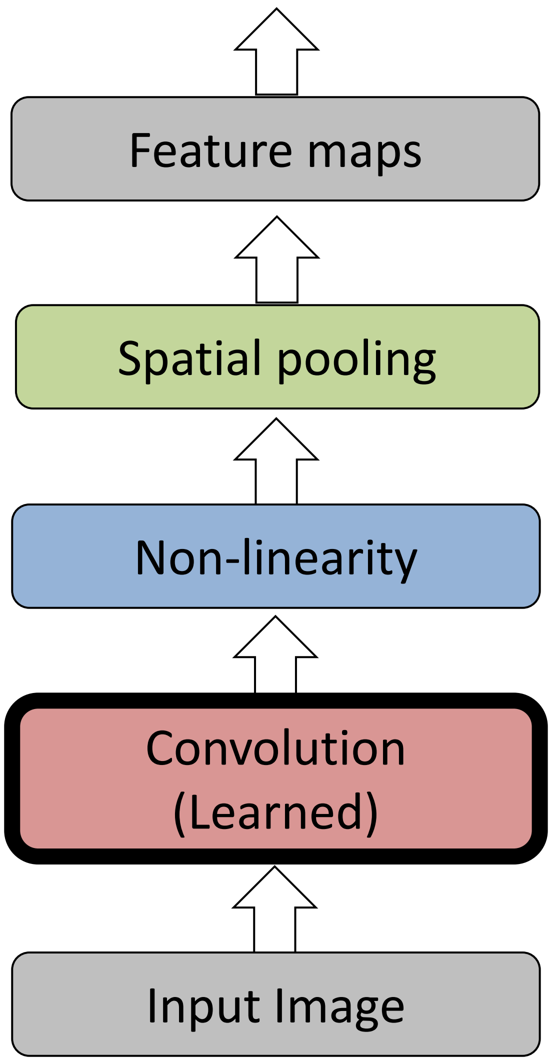
- Few parameters (filter weights)
- Dependencies are local
- Translation invariance

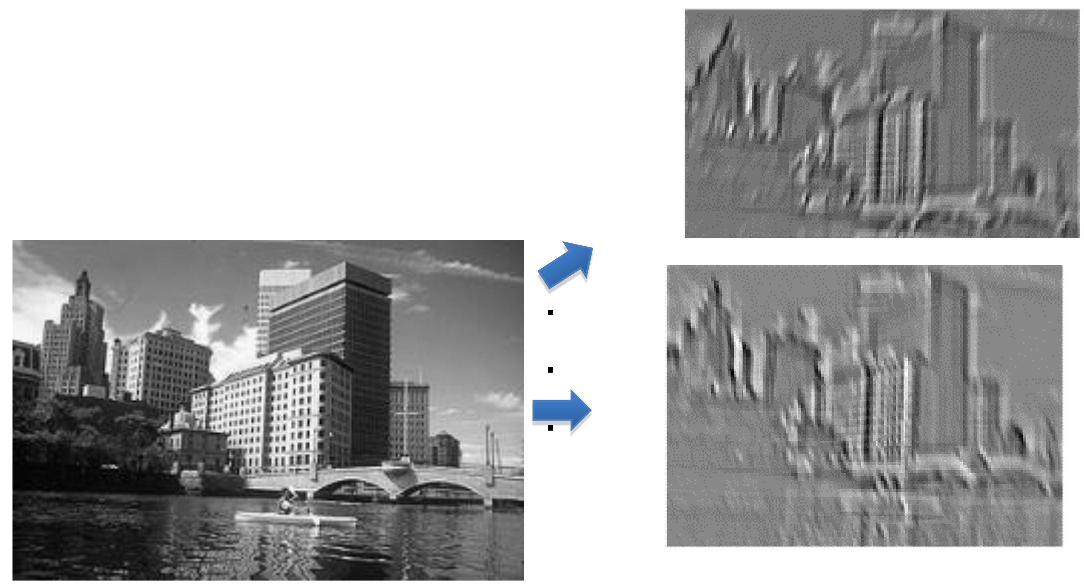


Input

Feature Map

Convolutional Neural Networks

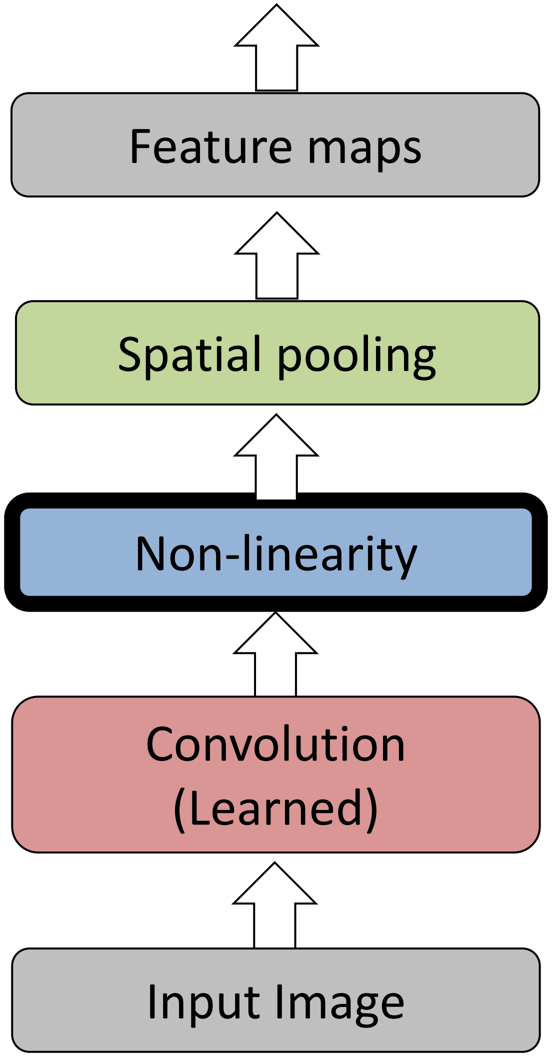




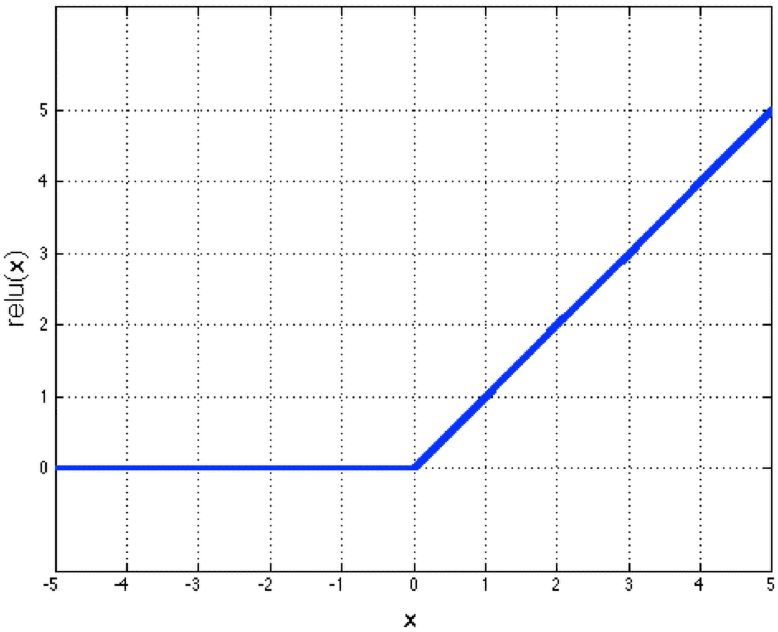
Input

Feature Map

Convolutional Neural Networks



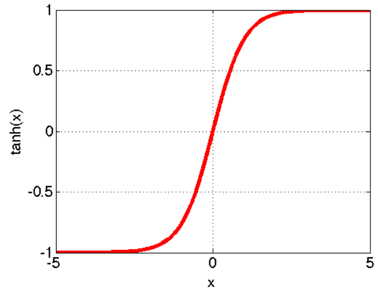
Rectified Linear Unit (ReLU)

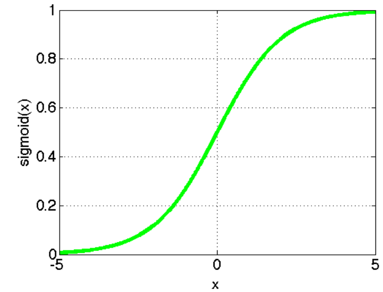


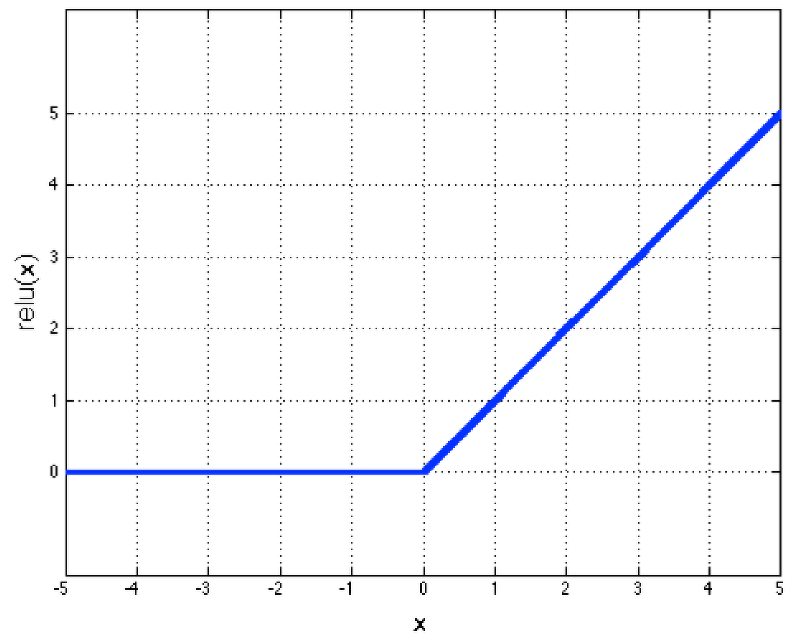
slide credit: S. Lazebnik

Non-Linearity

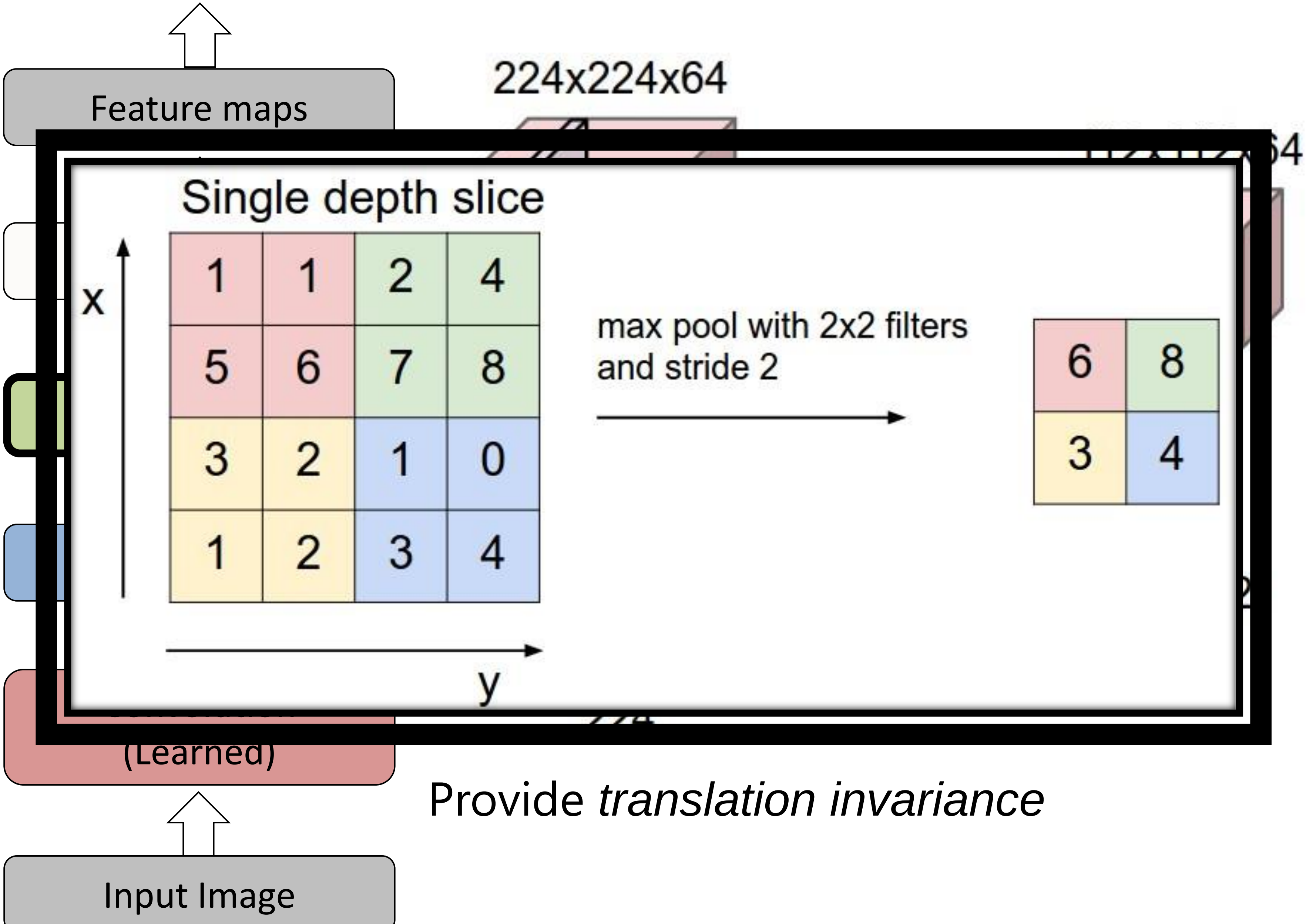
- Per-element (independent)
- Options:
  - Tanh
  - Sigmoid: 1/(1+exp(-x))
  - Rectified linear unit (ReLU)
    - Makes learning faster
    - Simplifies backpropagation
    - Avoids saturation issues
      → Preferred option







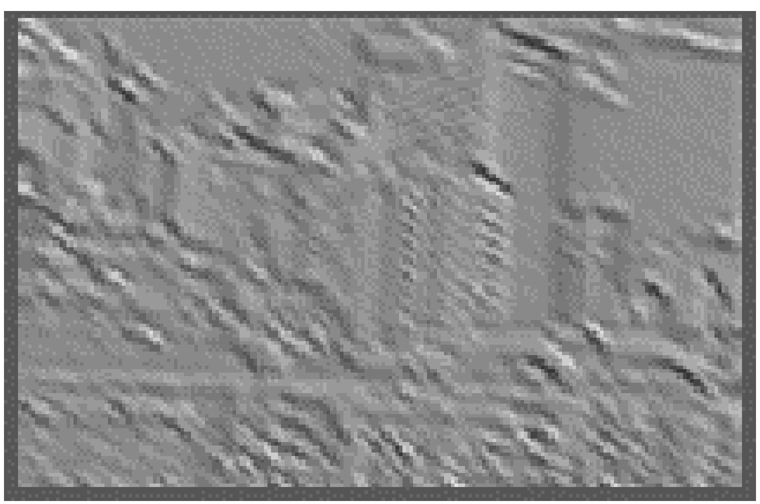
Convolutional Neural Networks



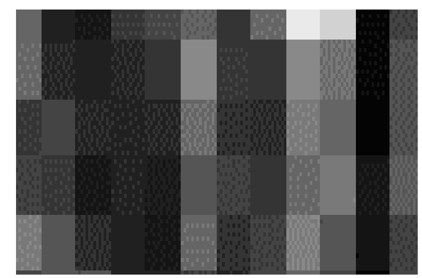
Provide *translation invariance*

Spatial Pooling

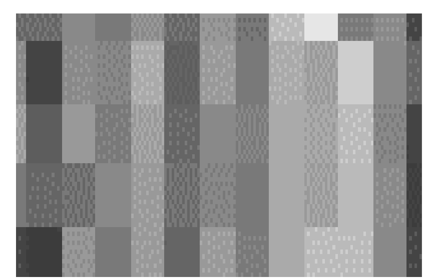
- Average or max
- Non-overlapping / overlapping regions
- Role of pooling:
  - Invariance to small transformations
  - Larger receptive fields (see more of input)



Max

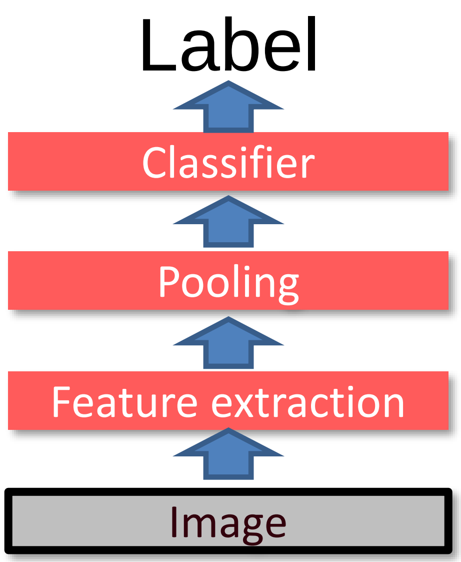


Average



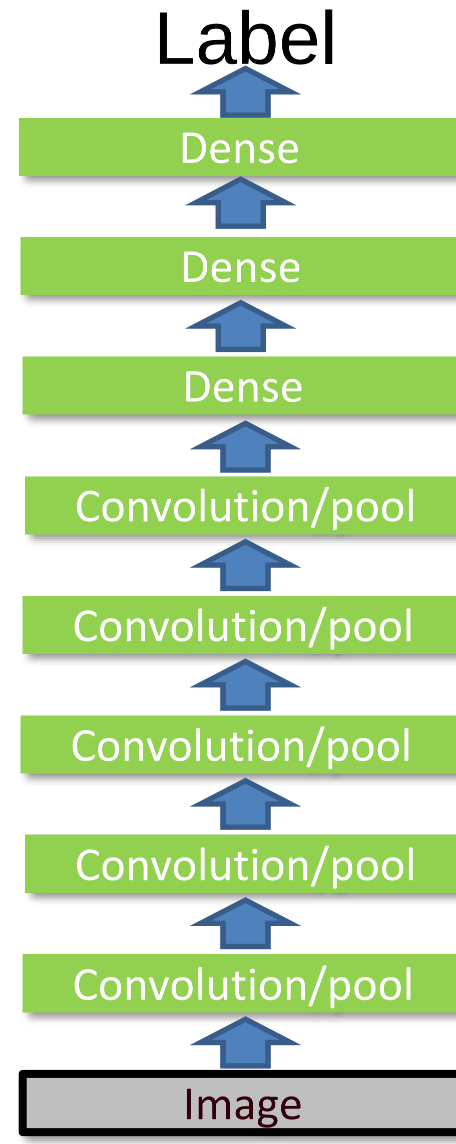
Engineered vs. learned features

Convolutional filters are trained in a supervised manner by back-propagating classification error



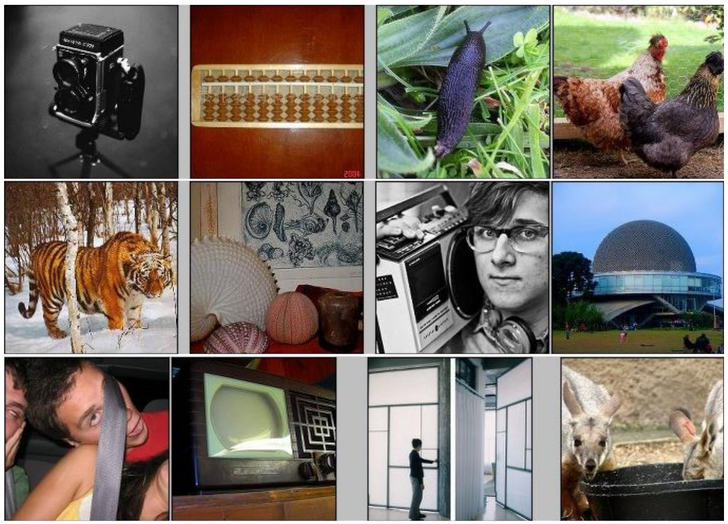
Softmax activation

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$



ImageNet Challenge 2012

IMAGENET



[Deng et al. CVPR 2009]

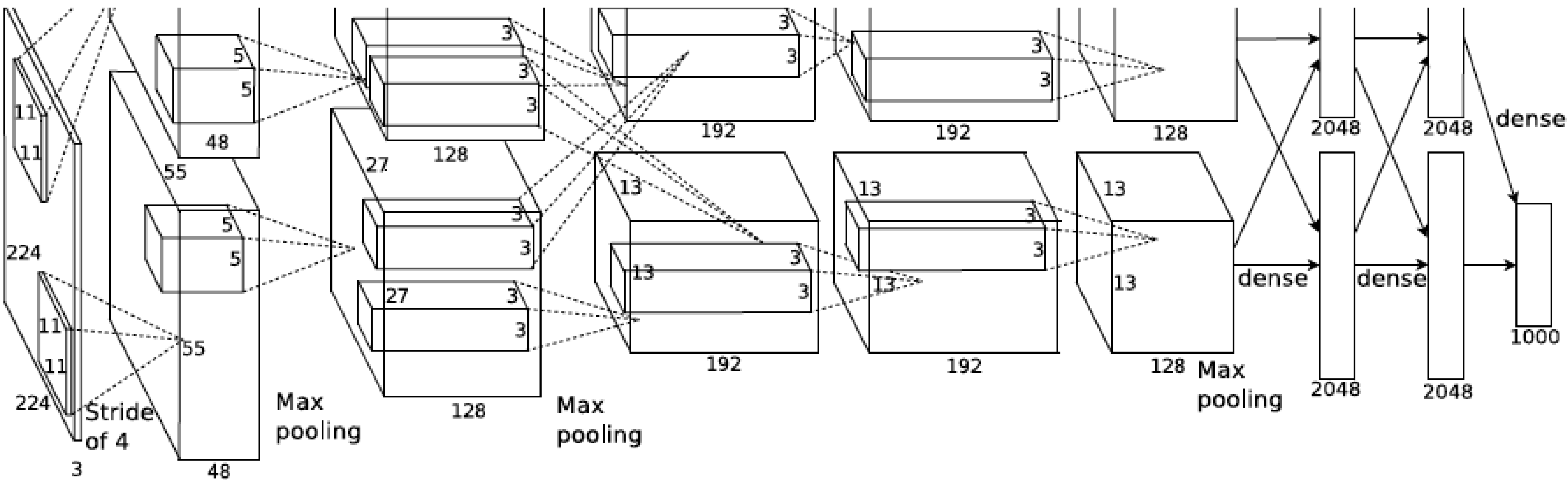
- ~14 million labeled images, 20k classes
- Images gathered from Internet
- Human labels via Amazon Turk
- **ImageNet Challenge**: *1.2 million training images, 1000 classes*

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

AlexNet

Similar framework to LeCun'98 but:

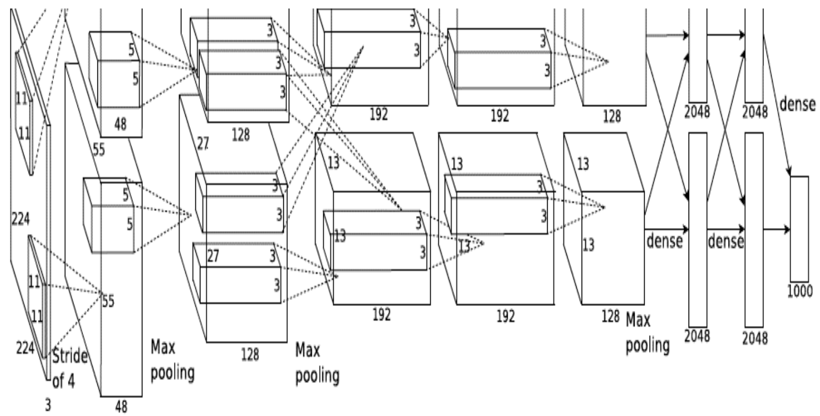
- Bigger model (7 hidden layers, 650,000 units, 60,000,000 params)
- More data (10^6 vs. 10^3 images)
- GPU implementation (50x speedup over CPU)
  - Trained on two GPUs for a week



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

AlexNet for image classification

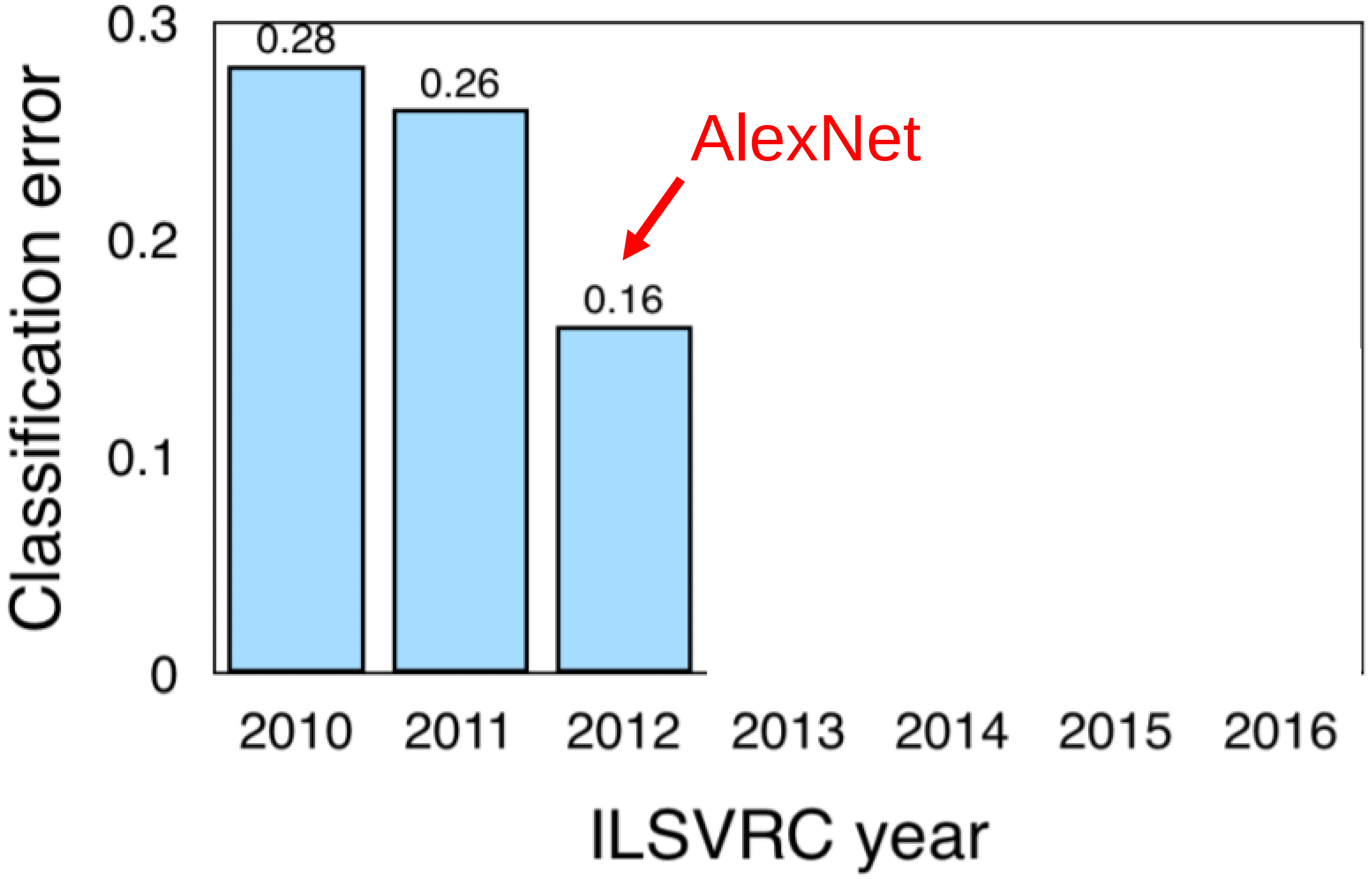
AlexNet



"car"

Fixed input size: 224x224x3

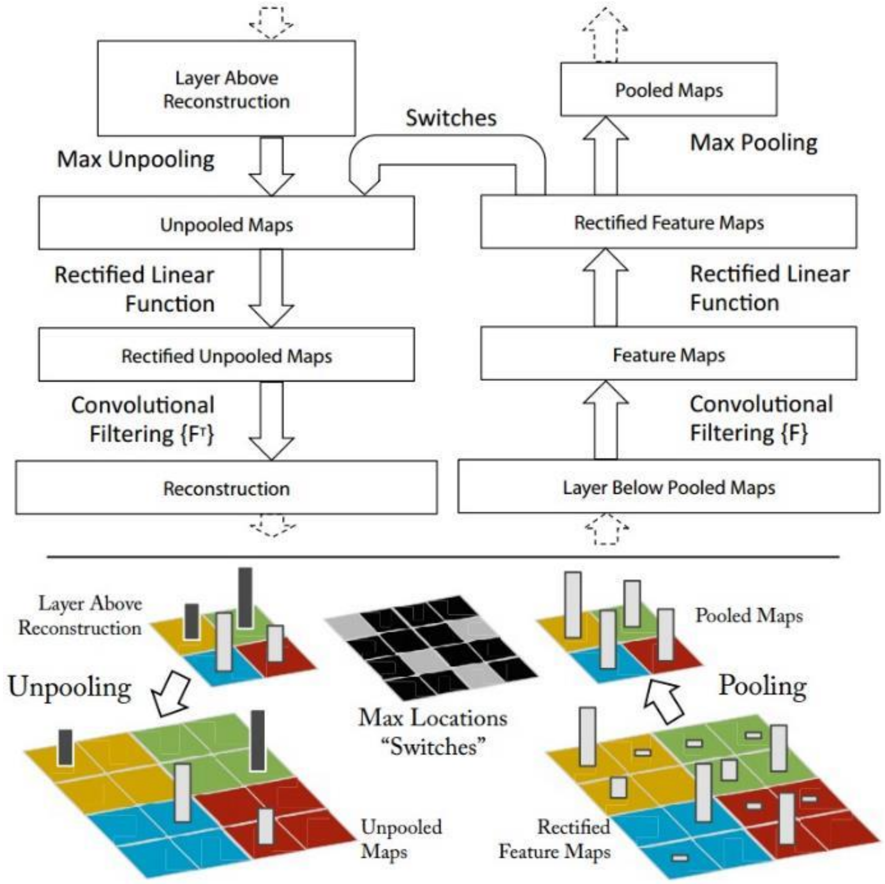
ImageNet Classification Challenge



http://image-net.org/challenges/talks/2016/ILSVRC2016_10_09_clsloc.pdf

Visualizing CNNs

- What input pattern originally caused a given activation in the feature maps?

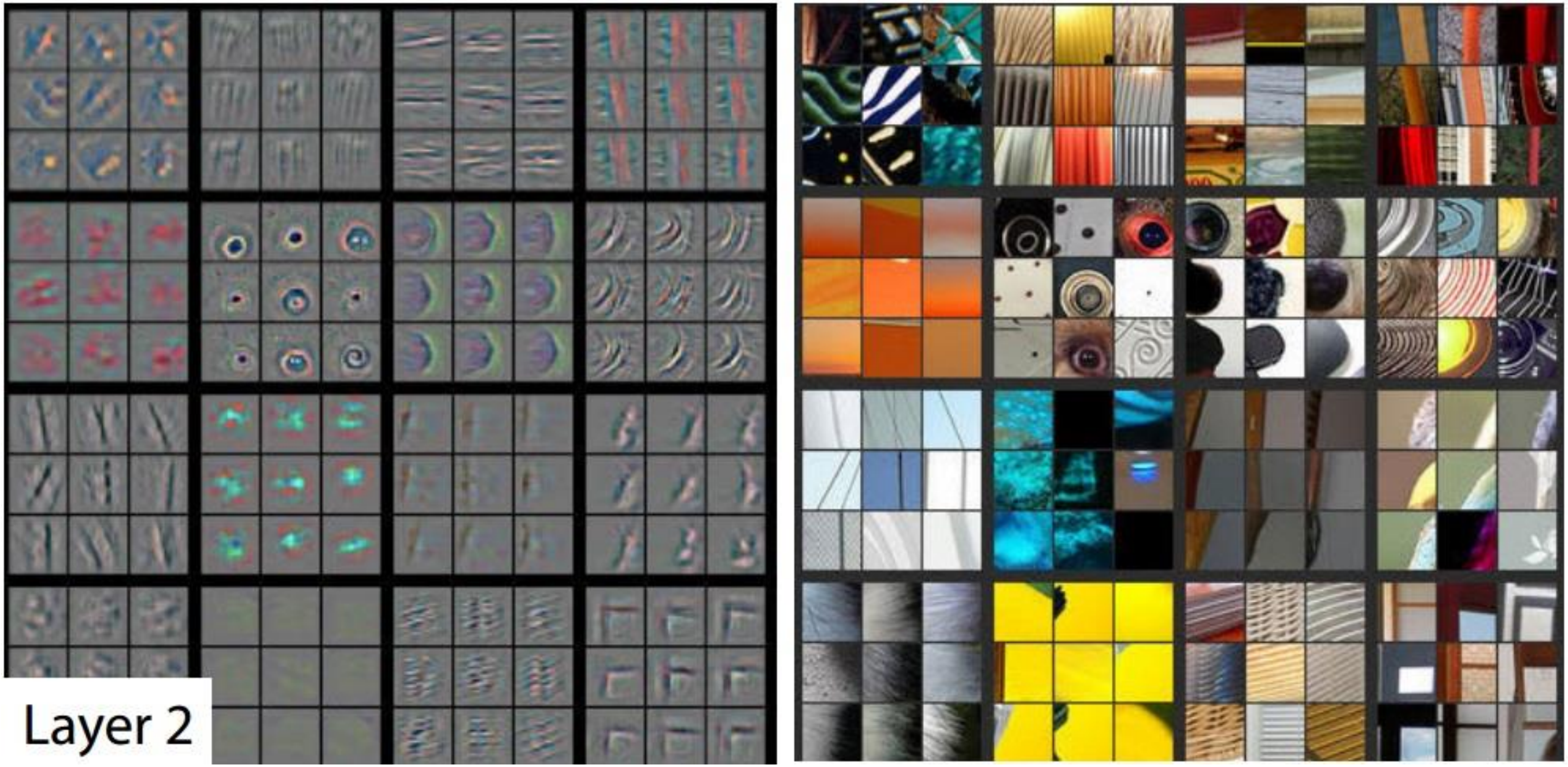


Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]

Layer 1

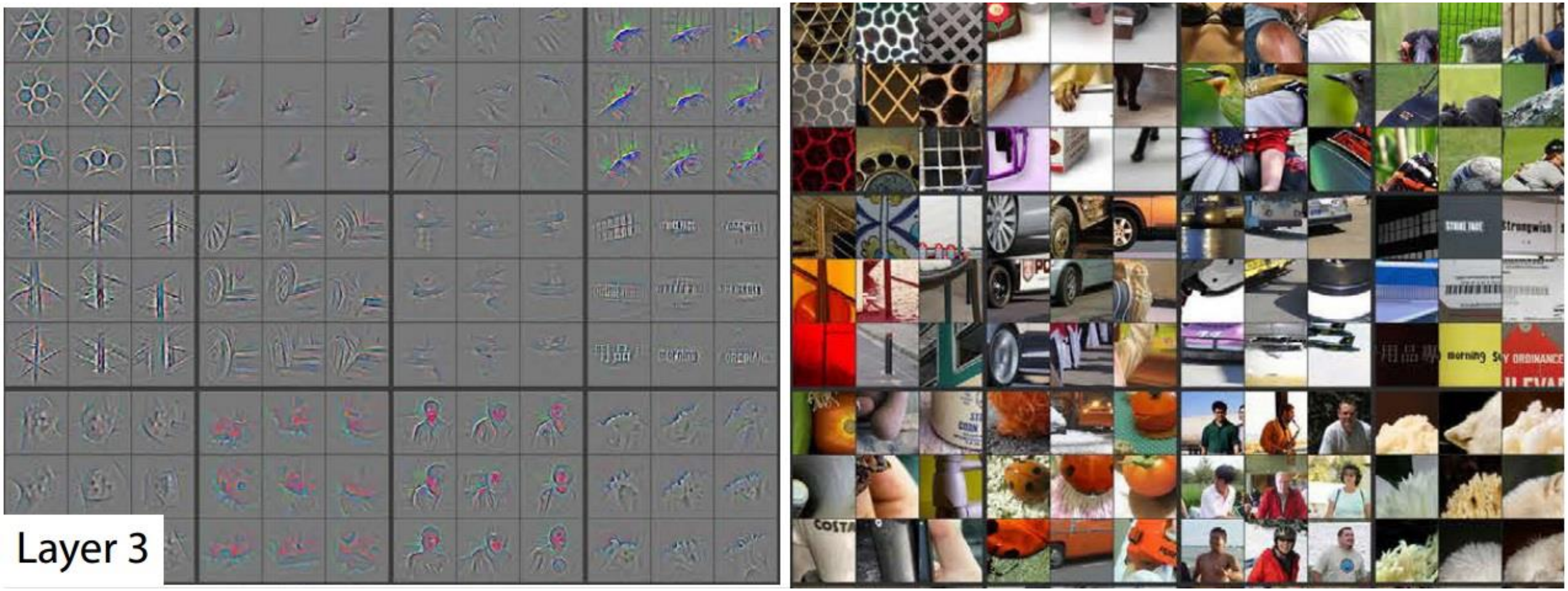
Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]

Layer 2



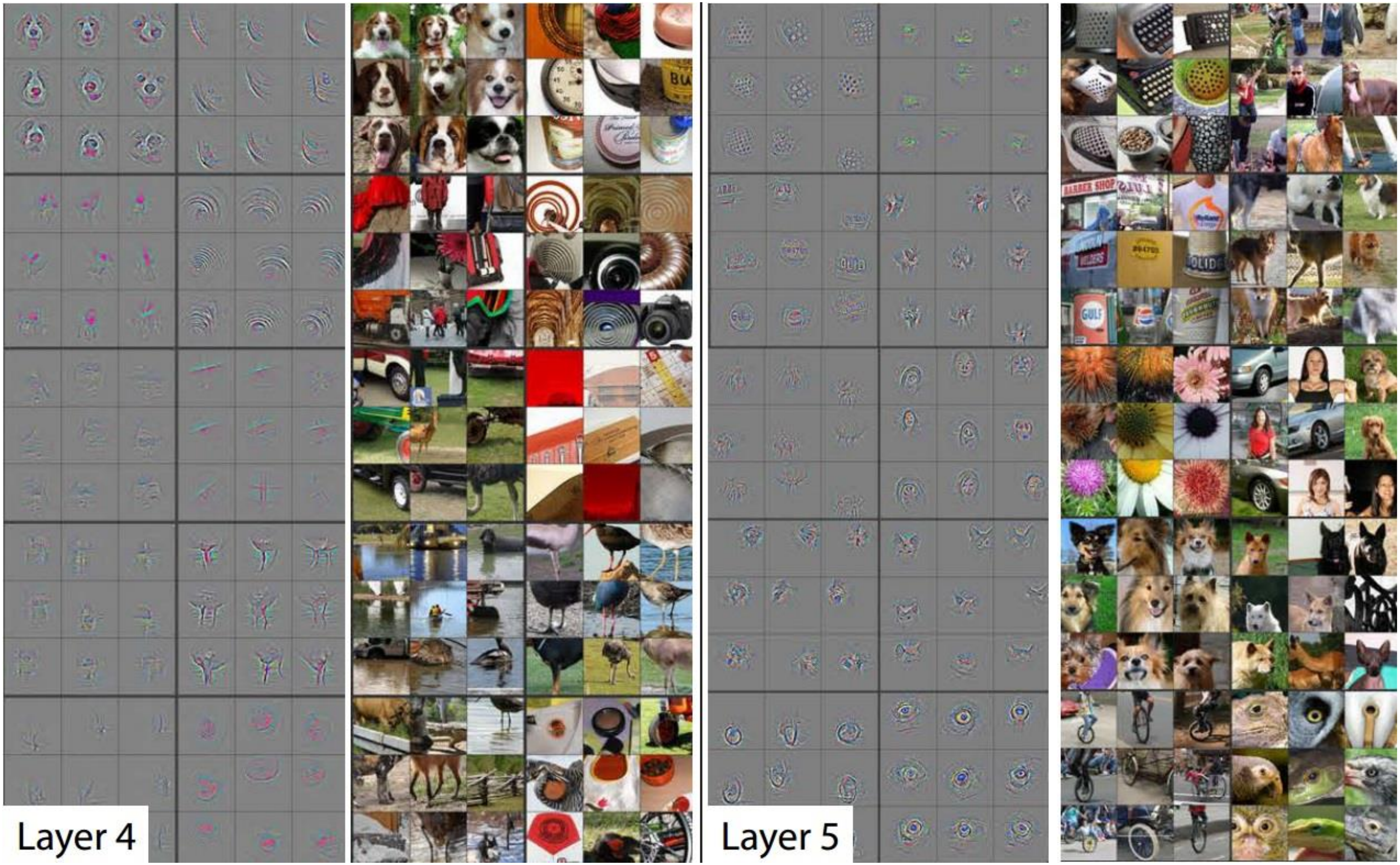
Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]

Layer 3



Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]

Layer 4 and 5



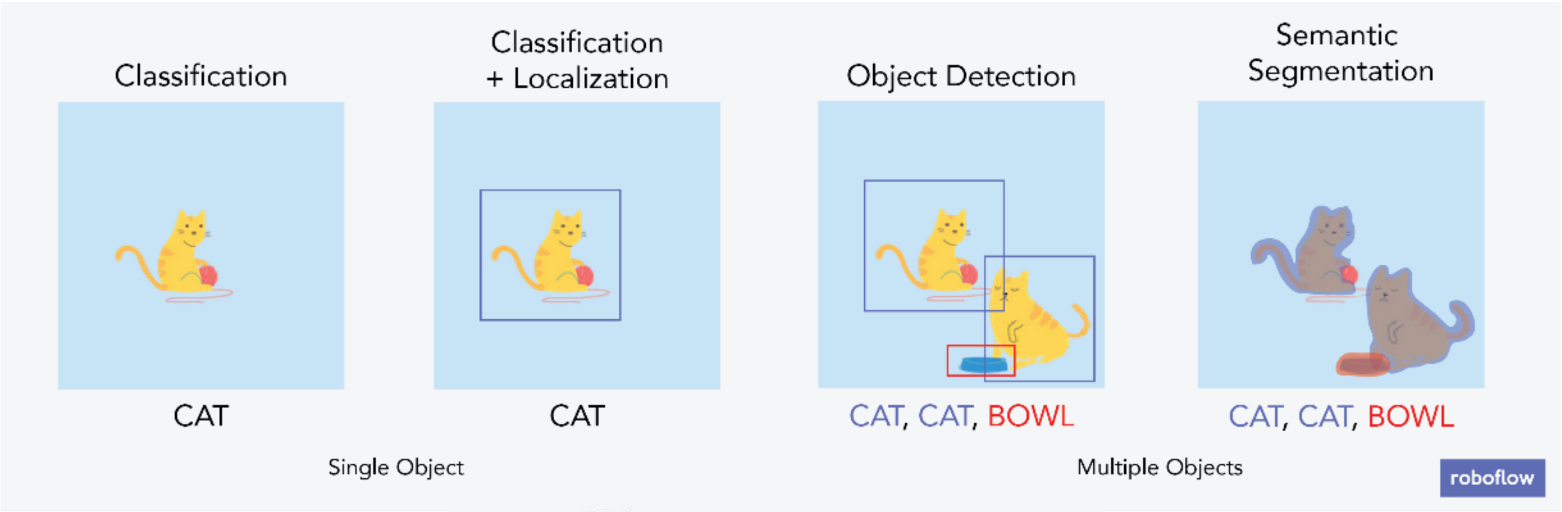
Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]

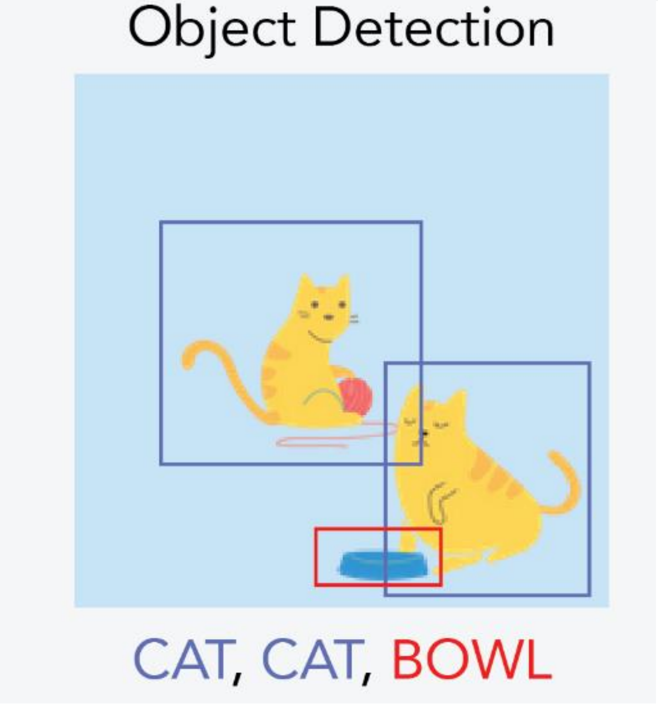
Beyond classification

- Detection
- Segmentation
- Regression
- Pose estimation
- Matching patches
- Synthesis

and many more…

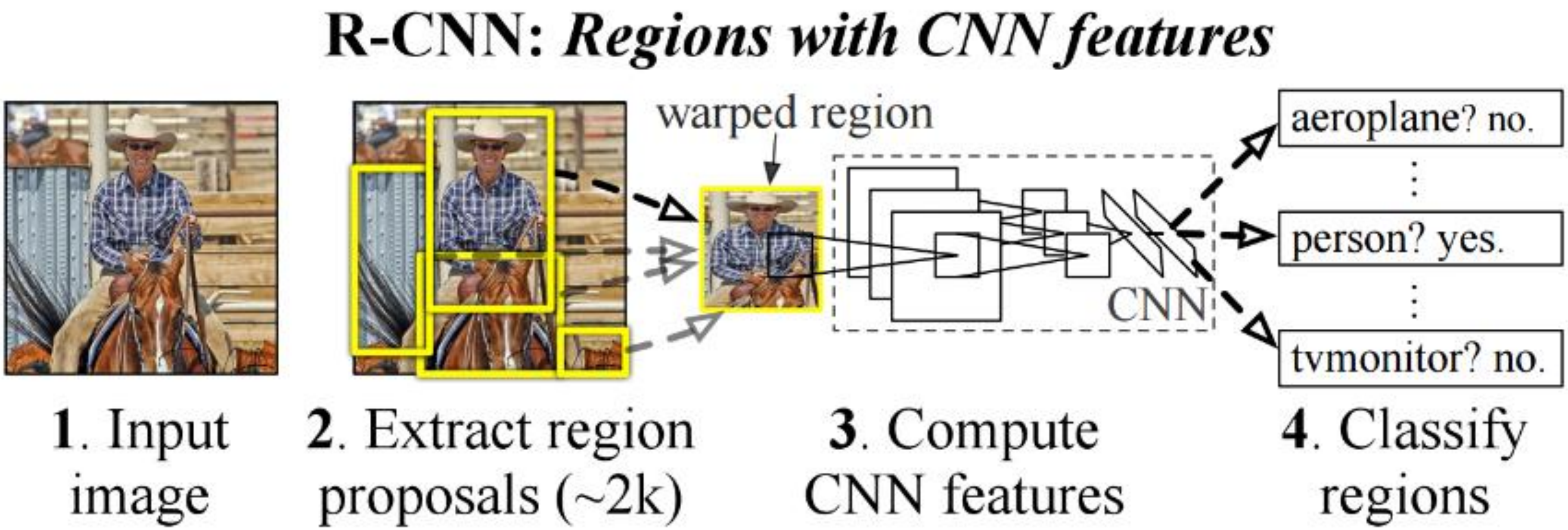
Object Detection using CNN





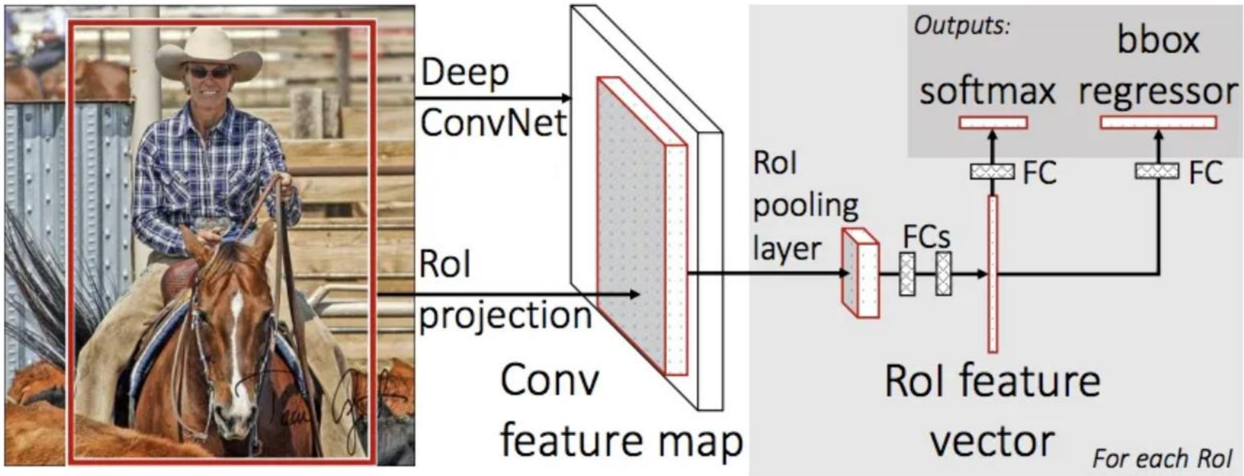
R-CNN: Regions with CNN features

- Trained on ImageNet classification
- Finetune CNN on PASCAL

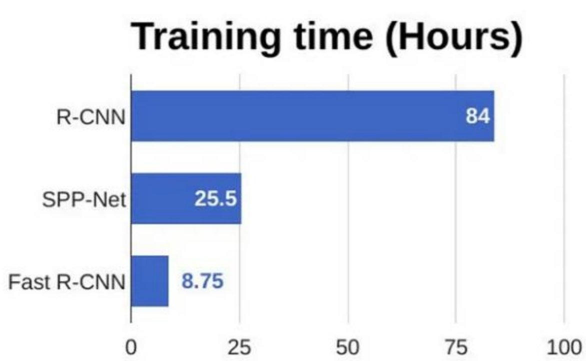


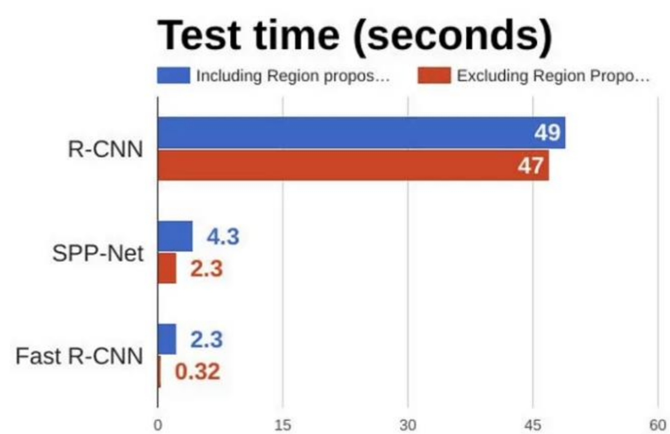
RCNN [Girshick et al. CVPR 2014]

Fast RCNN



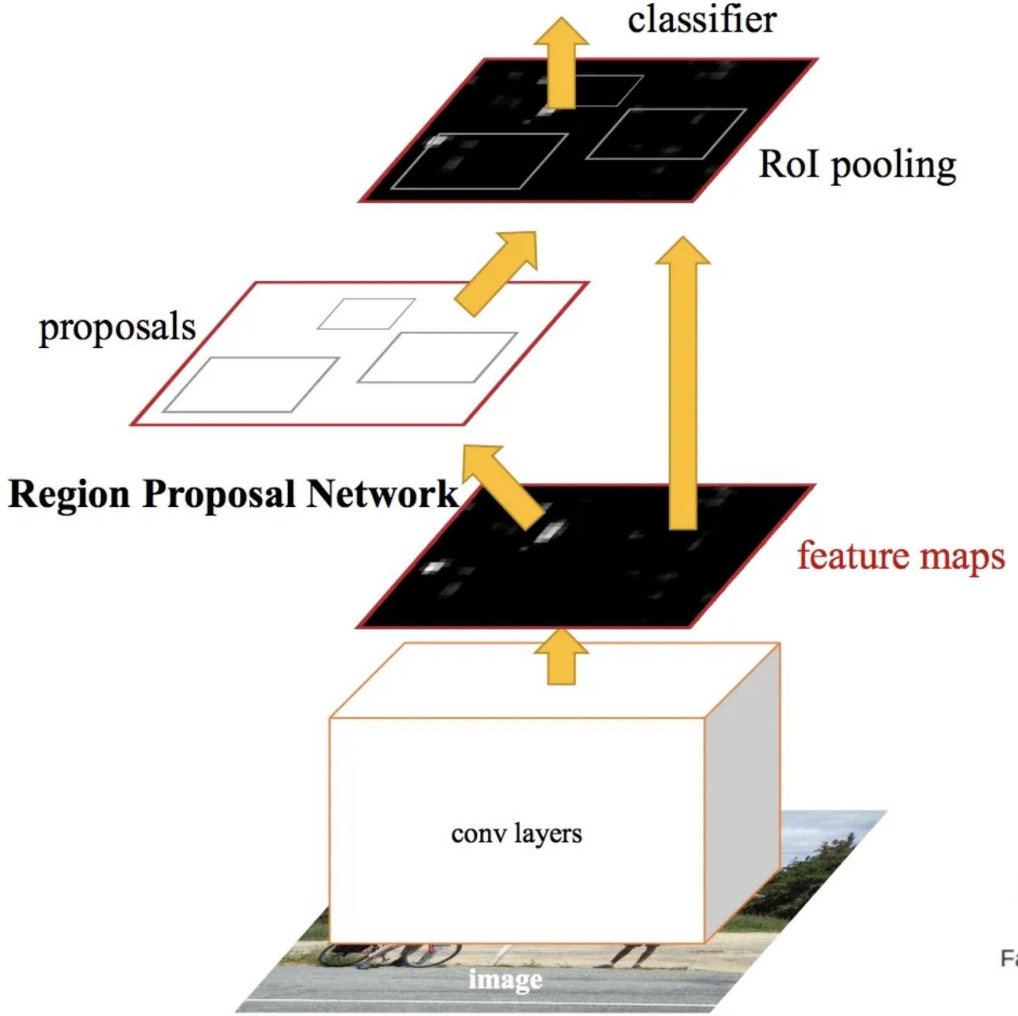
Fast R-CNN



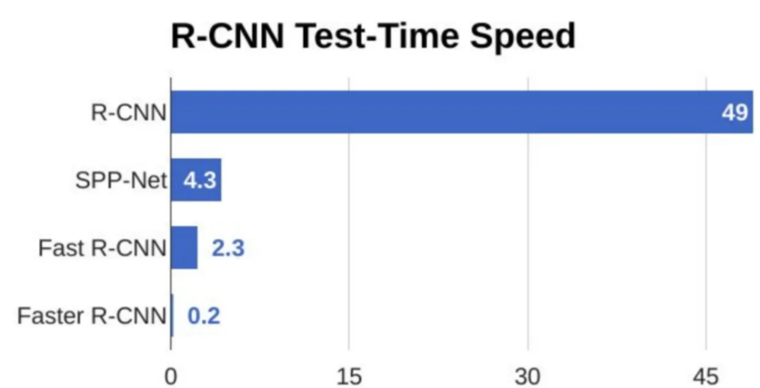


Comparison of object detection algorithms

Faster RCNN

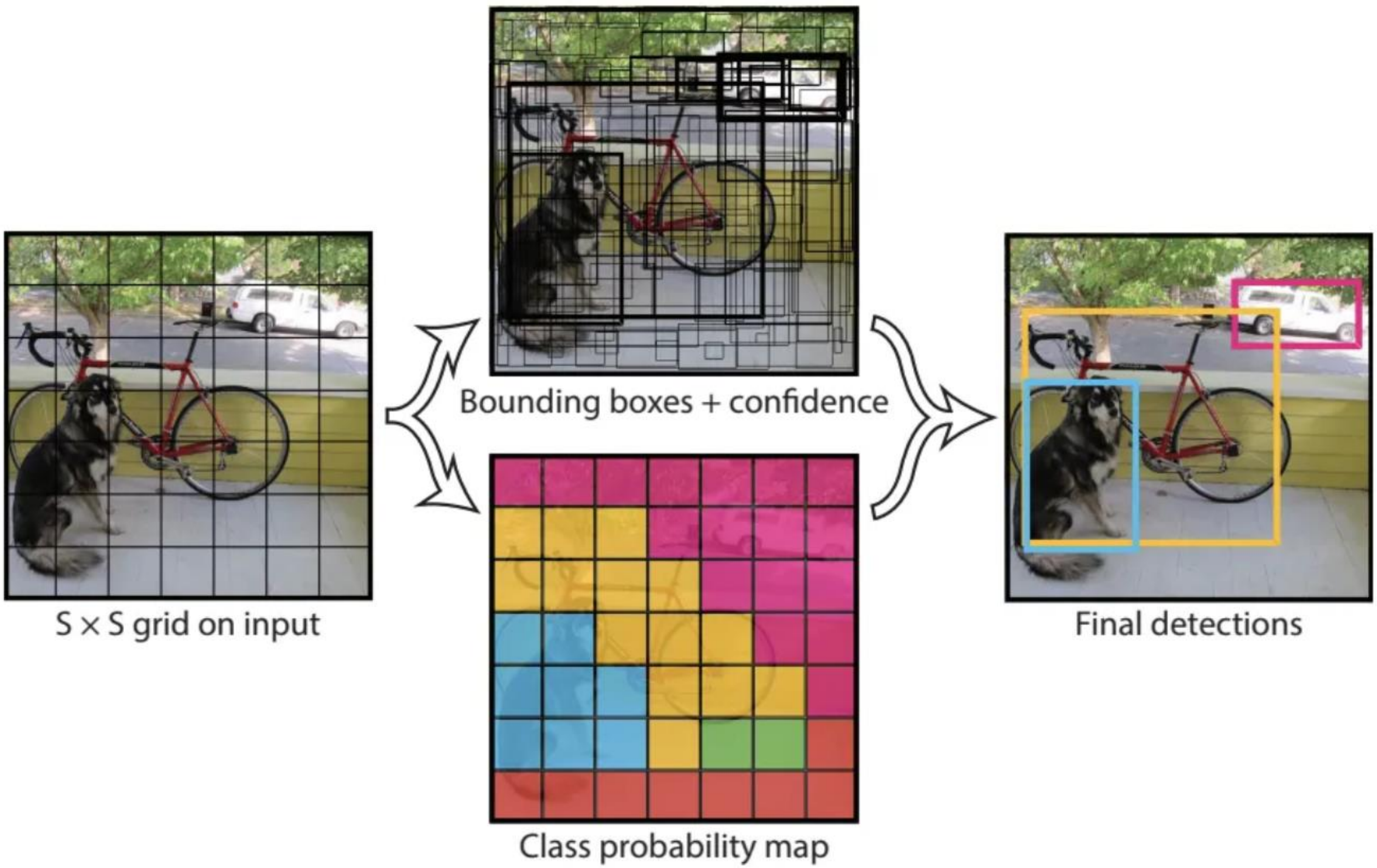


Faster R-CNN



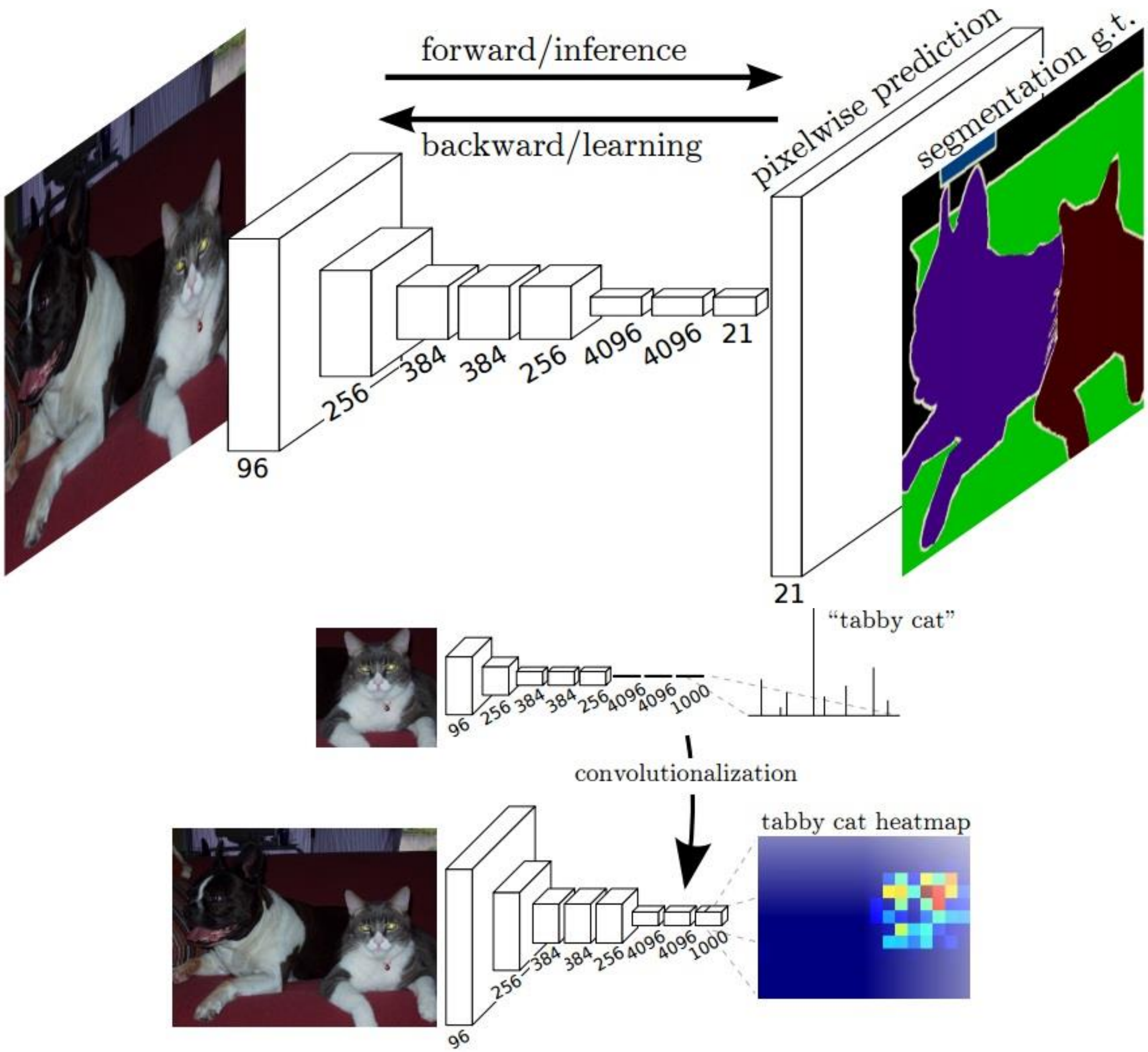
Comparison of test-time speed of object detection algorithms

Yolo You Look Only Once



YOLO is orders of magnitude faster(45 frames per second) than other

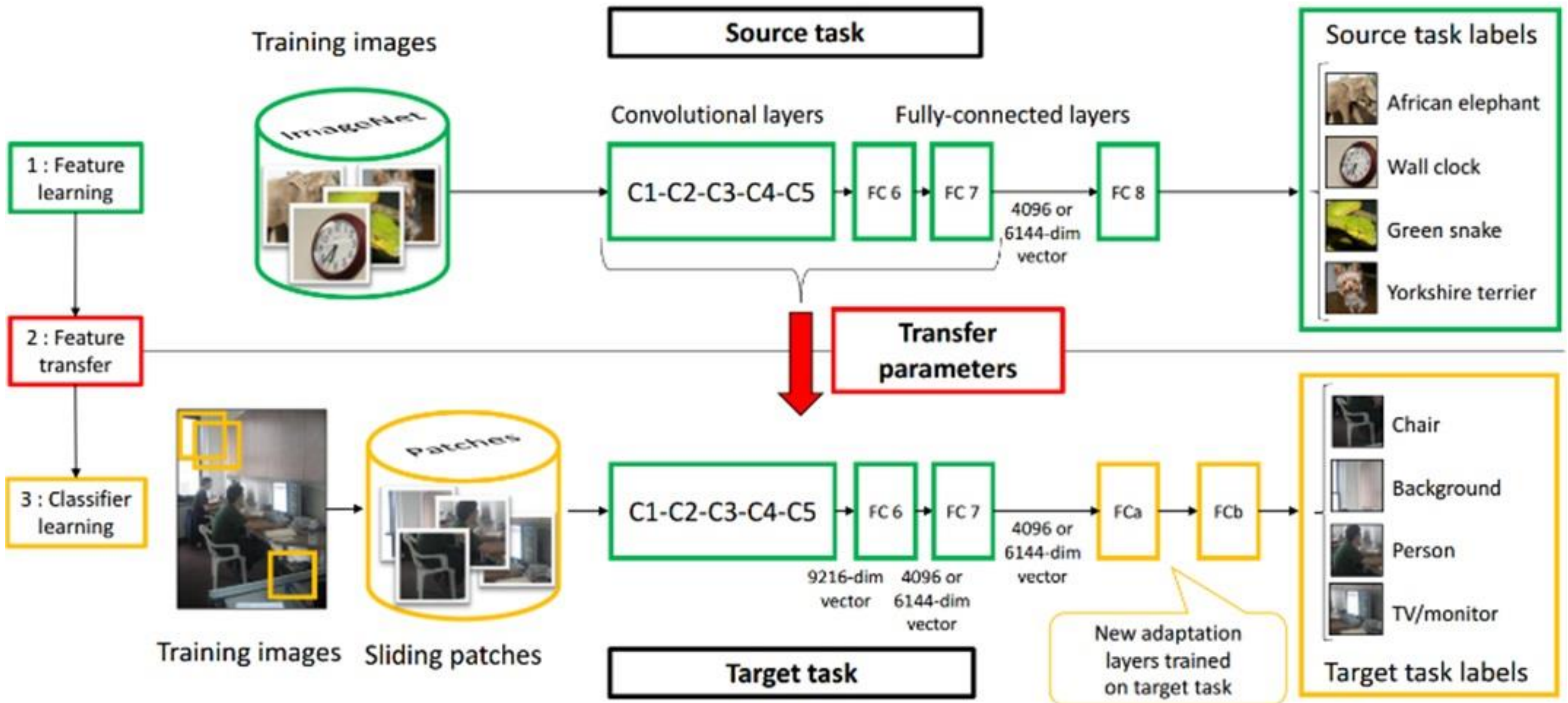
Labeling Pixels: Semantic Labels



Fully Convolutional Networks for Semantic Segmentation [Long et al. CVPR 2015]

Transfer Learning

- Improvement of learning in a **new** task through the *transfer of knowledge* from a **related** task that has already been learned.
- Weight initialization for CNN

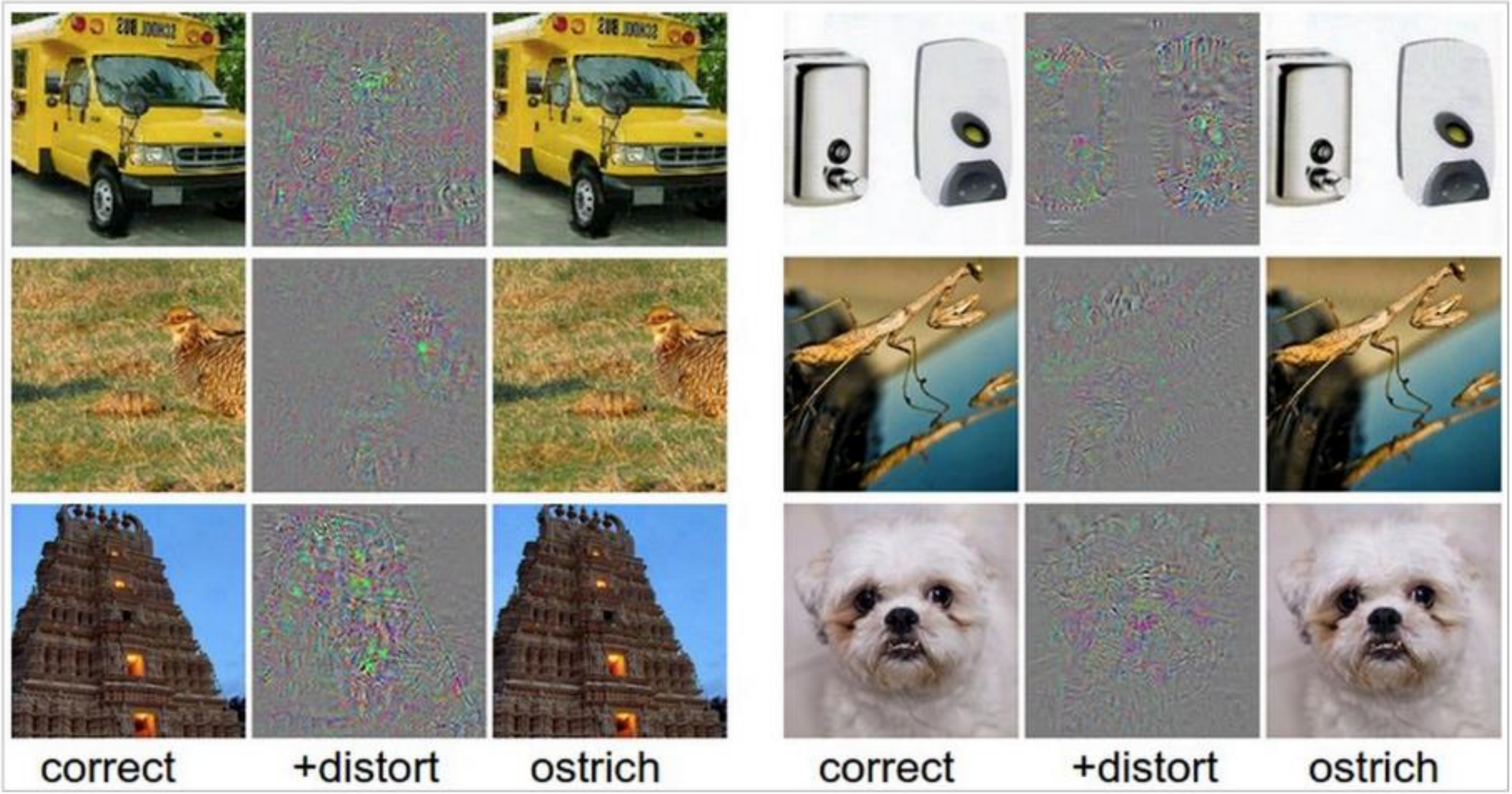


Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks [Oquab et al. CVPR 2014]

Deep learning libraries

- Tensorflow
- Caffe
- Torch
- MatConvNet

Fooling CNNs



Take a correctly classified image (left image in both columns), and add a tiny distortion (middle) to fool the ConvNet with the resulting image (right).

Intriguing properties of neural networks [Szegedy ICLR 2014]