



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 ...

2
Slide adapted from Antonio Torralba

Discriminative classifiers

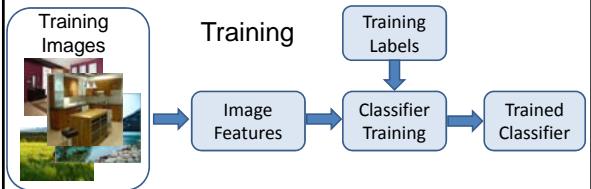
Nearest neighbor 10 ⁶ examples Shakhnarovich, Viola, Darrell 2003 Berg, Berg, Malik 2005...	Neural networks LeCun, Bottou, Bengio, Haffner 1998 Rowley, Baluja, Kanade 1998 ...	
Support Vector Machines Guyon, Vapnik Heisele, Serre, Poggio, 2001,....	Boosting Viola, Jones 2001, Torralba et al. 2004, Opelt et al. 2006,....	Conditional Random Fields McCallum, Freitag, Pereira 2000; Kumar, Hebert 2003 ...

3
Slide adapted from Antonio Torralba

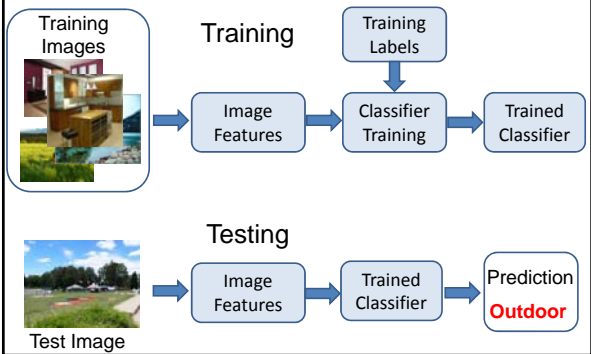
Outline

- Deep Neural Networks
- Convolutional Neural Networks (CNNs)

Traditional Image Categorization: Training phase



Traditional Image Categorization: Testing phase



Features have been key..

Hand-crafted

SIFT [Lazebnik et al. CVPR 06]

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”)

Image/Video Pixels → Layer 1 → Layer 2 → Layer 3 → Simple Classifier

Learning Feature Hierarchy

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

Input data

Feature representation

3rd layer "Objects"

2nd layer "Object parts"

1st layer "Edges"

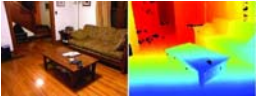
Pixels

Lee et al., ICML 2009; CACM 2011

Slide: Rob Fergus

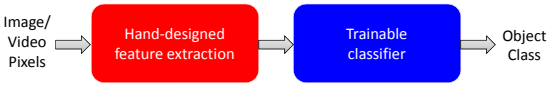
Learning Feature Hierarchy

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




“Shallow” vs. “deep” architectures

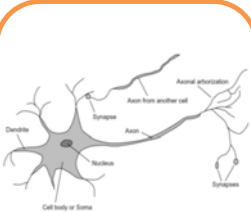
Traditional recognition: “Shallow” architecture



Deep learning: “Deep” architecture



Biological neuron and Perceptrons



A biological neuron

Input

Weights

x_1 w_1

x_2 w_2

x_3 w_3

\vdots


x_n w_n

Output: $\sigma(wx + b)$

Sigmoid function:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

An artificial neuron (Perceptron)
- a linear classifier



Simple, Complex, and Hyper-complex cells

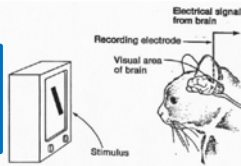


David H. Hubel and Torsten Wiesel

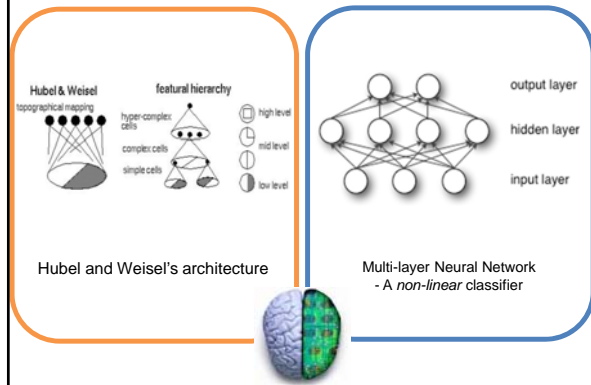
[video](#)

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*

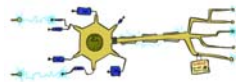


Hubel/Wiesel Architecture and Multi-layer Neural Network



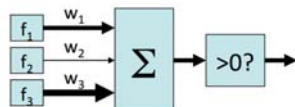
Neuron: Linear Perceptron

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

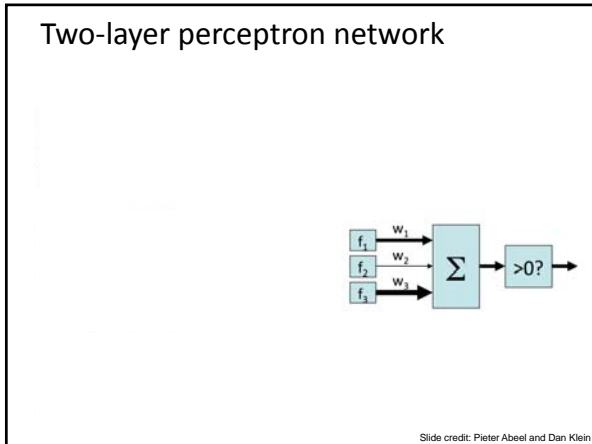


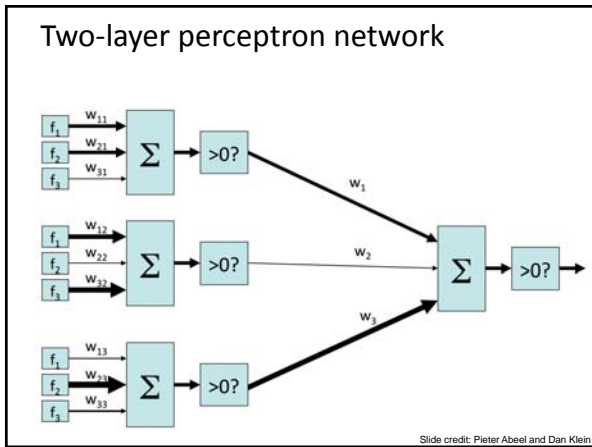
$$\text{activation}_w(x) = \sum_i w_i \cdot f_i(x) = w \cdot f(x)$$

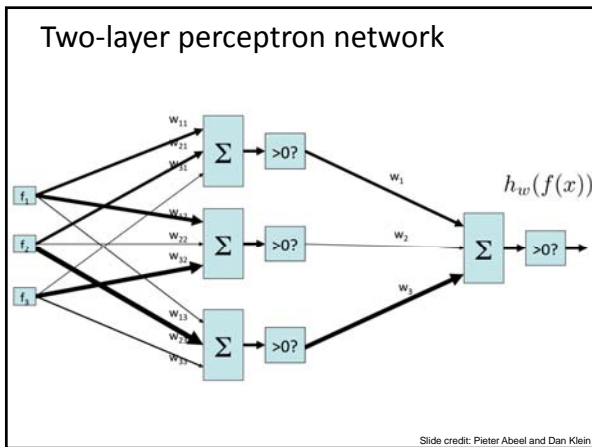
- If the activation is:
 - Positive, output +1
 - Negative, output -1



Slide credit: Pieter Abeel and Dan Klein







Learning w

- Training examples

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

- Objective: a misclassification loss

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

- 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

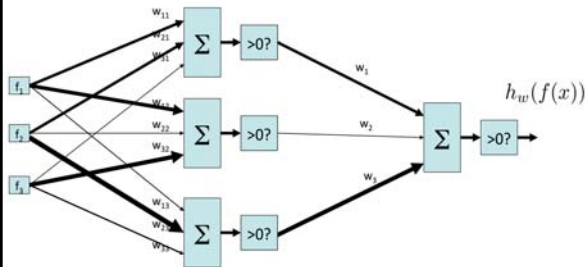
- What's bad?

- Optimal?

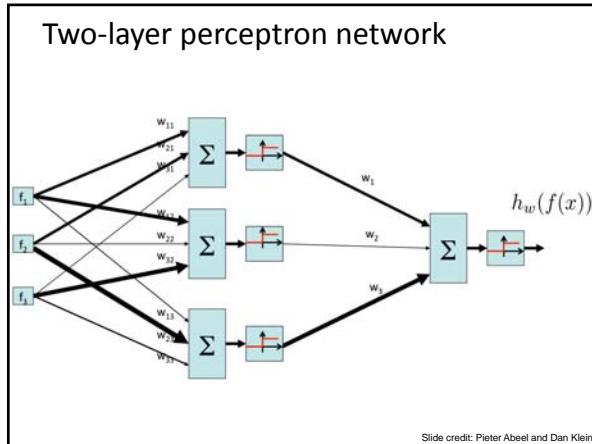


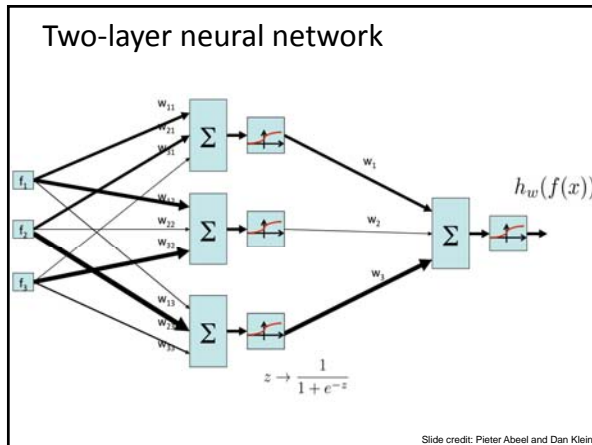
Slide credit: Pieter Abeel and Dan Klein

Two-layer perceptron network



Slide credit: Pieter Abeel and Dan Klein





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 \mathbf{w} to minimize the error between true training labels and estimated labels

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

- 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

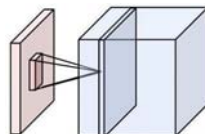
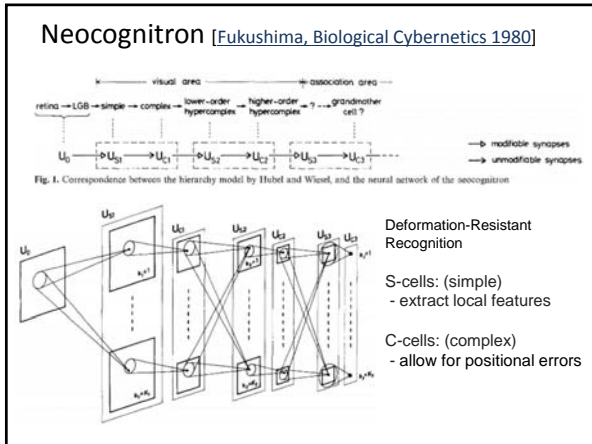
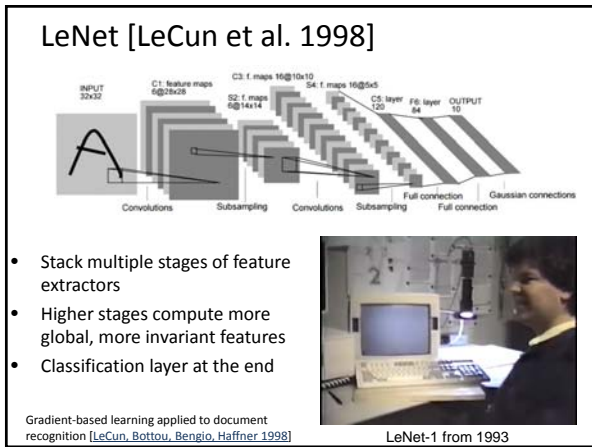
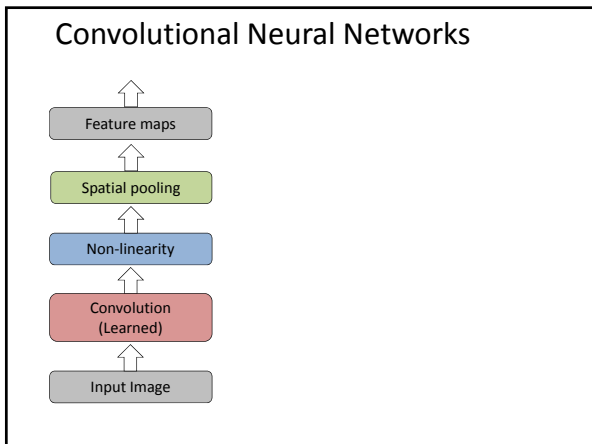


Image credit: A. Karpathy







What is a Convolution?

- Weighted moving sum

Input Feature Activation Map

Why Convolution?

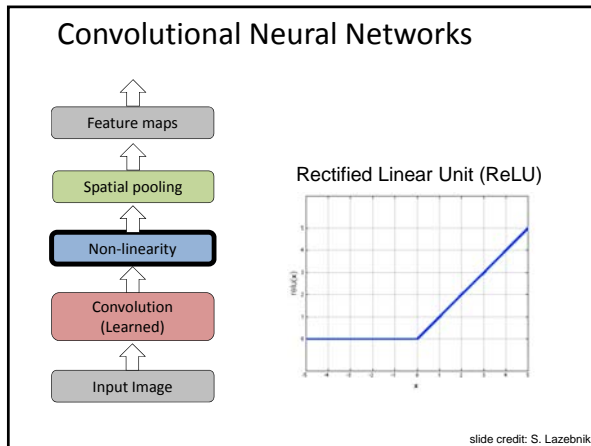
- Few parameters (filter weights)
- Dependencies are local
- Translation invariance

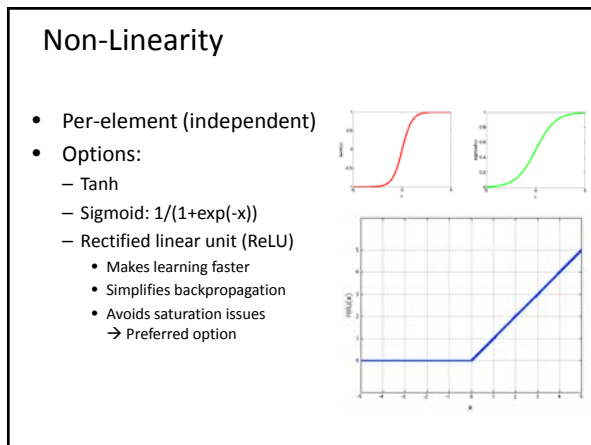
Input Feature Map

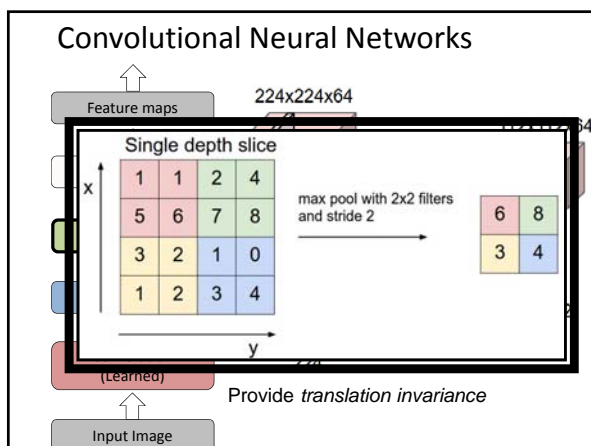
Convolutional Neural Networks

Input Image Convolution (Learned) Non-linearity Spatial pooling Feature maps

Input Feature Map

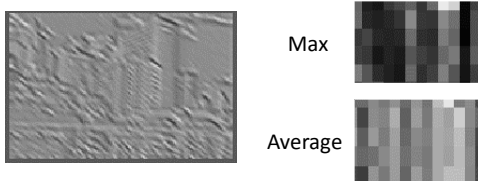






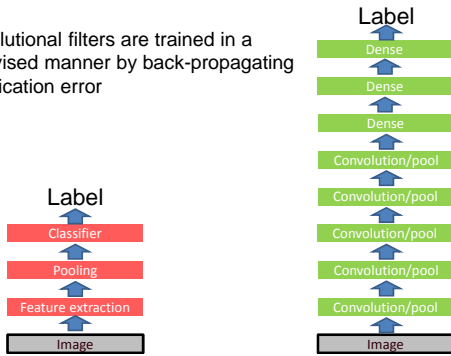
Spatial Pooling

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

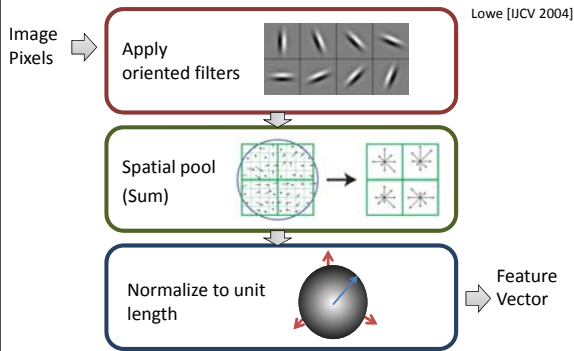


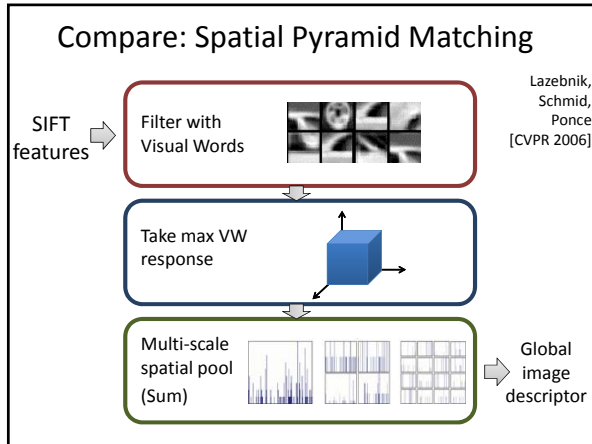
Engineered vs. learned features

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



Compare: SIFT Descriptor





Earlier convnet successes

- Handwritten text/digits
 - MNIST (0.17% error [Ciresan et al. 2011])
 - Arabic & Chinese [Ciresan et al. 2012]
- Simpler recognition benchmarks
 - CIFAR-10 (9.3% error [Wan et al. 2013])
 - Traffic sign recognition
 - 0.56% error vs 1.16% for humans [Ciresan et al. 2011]

ImageNet Challenge 2012

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

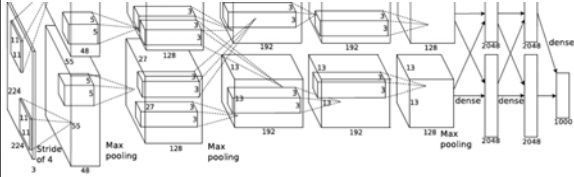
[Deng et al. CVPR 2009]

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

AlexNet

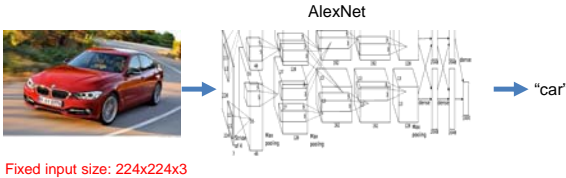
Similar framework to LeCun'98 but:

- Bigger model (7 hidden layers, 650,000 units, 60,000,000 params)
- More data (10^9 vs. 10^3 images)
- GPU implementation (50x speedup over CPU)
 - Trained on two GPUs for a week

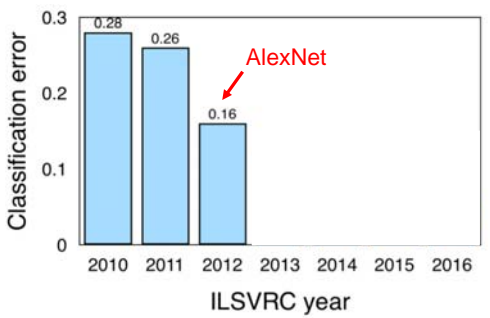


A. Krizhevsky, I. Sutskever, and G. Hinton, [ImageNet Classification with Deep Convolutional Neural Networks](#), NIPS 2012

AlexNet for image classification



ImageNet Classification Challenge



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

Industry Deployment

- Used in Facebook, Google, Microsoft
- Startups
- Image Recognition, Speech Recognition, ...
- Fast at test time

Taiyan et al. DeepFace: Closing the Gap to Human-Level Performance in Face Verification, CVPR'14

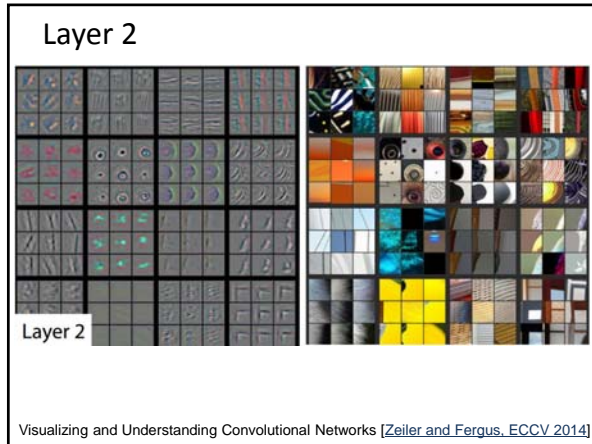
Visualizing CNNs

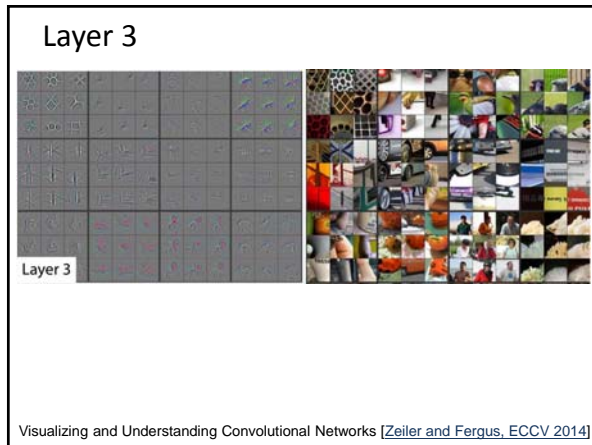
- What input pattern originally caused a given activation in the feature maps?

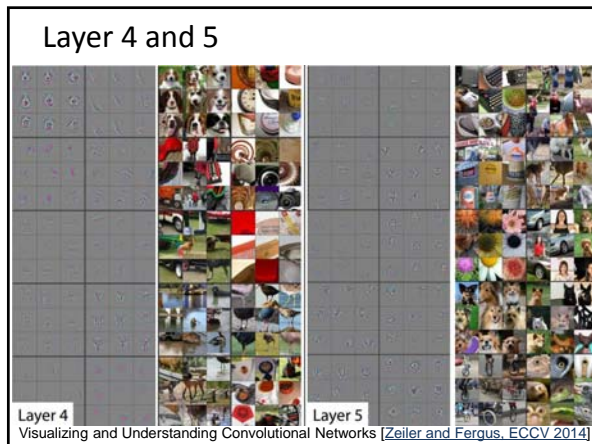
Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]

Layer 1

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







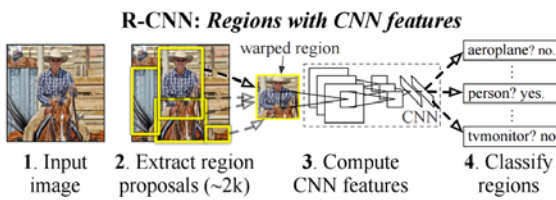
Beyond classification

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

and many more...

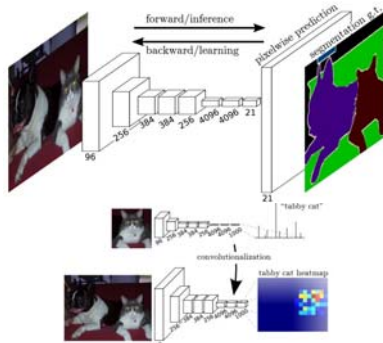
R-CNN: Regions with CNN features

- Trained on ImageNet classification
- Finetune CNN on PASCAL



RCNN [Girshick et al. CVPR 2014]

Labeling Pixels: Semantic Labels



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

Labeling Pixels: Edge Detection

DeepEdge: A Multi-Scale Bifurcated Deep Network for Top-Down Contour Detection
 [Bertasius et al. CVPR 2015]

CNN for Regression

DeepPose [Toshev and Szegedy CVPR 2014]

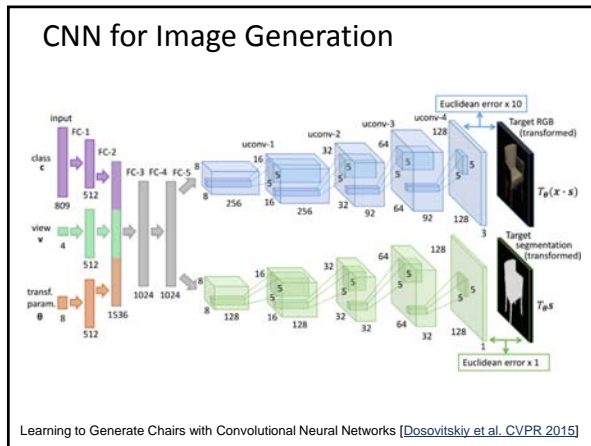
CNN as a Similarity Measure for Matching

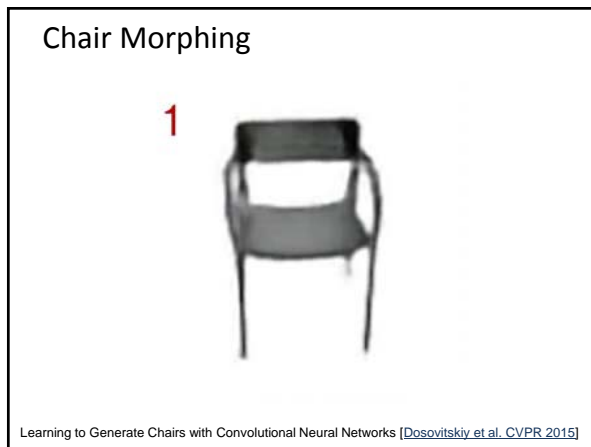
Stereo matching [Zbontar and LeCun CVPR 2015]
 Compare patch [Zagoruyko and Komodakis 2015]

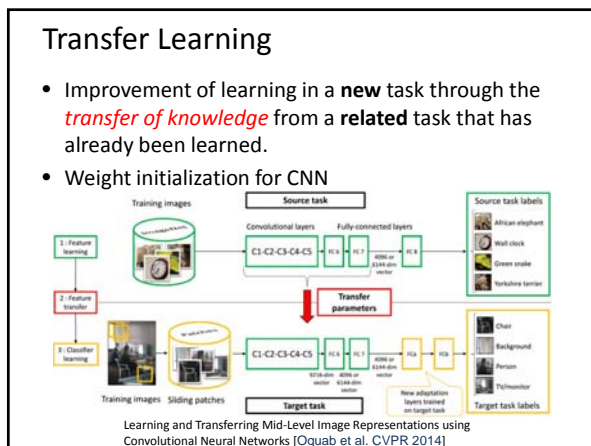
FaceNet [Schroff et al. 2015]

FlowNet [Fischer et al 2015]

Match ground and aerial images
 [Lin et al. CVPR 2015]



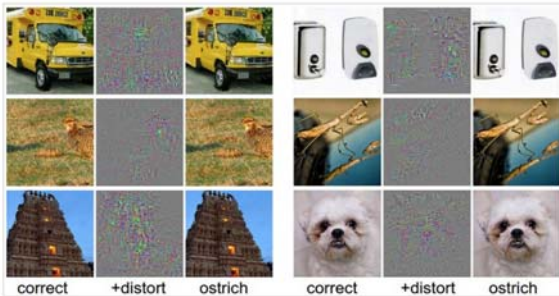




Deep learning libraries

- [Tensorflow](#)
- [PyTorch](#)
- [Caffe](#)
- [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]

Questions?

See you Thursday!
