

**Deep neural networks**

June 1<sup>st</sup>, 2017

Yong Jae Lee  
UC Davis

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

---

---

---

---

---

---

---

**Announcements**

- Post questions on Piazza for review-session (6/8 lecture)

2

---

---

---

---

---

---

---

**Outline**

- Deep Neural Networks
- Convolutional Neural Networks (CNNs)

---

---

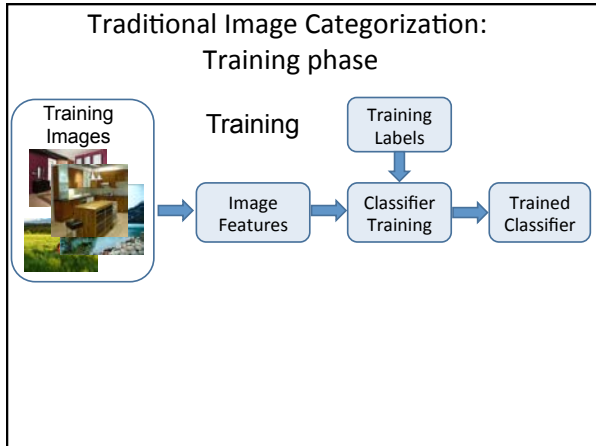
---

---

---

---

---



---

---

---

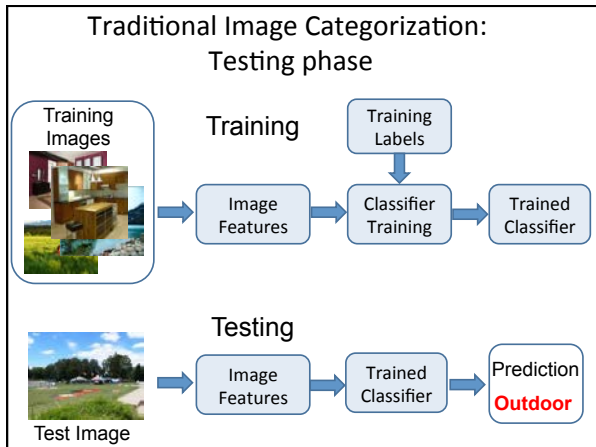
---

---

---

---

---



---

---

---

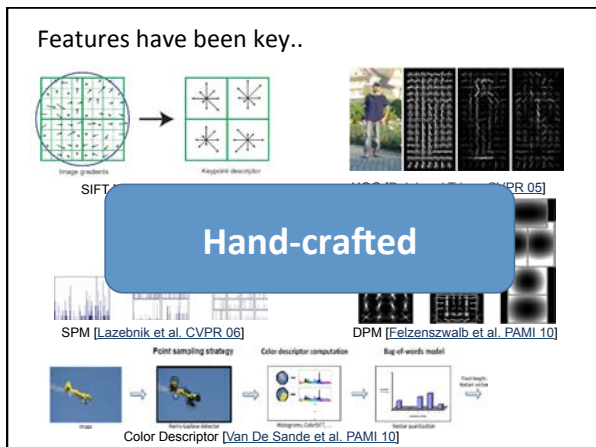
---

---

---

---

---



---

---

---

---

---

---

---

---

### 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




---

---

---

---

---

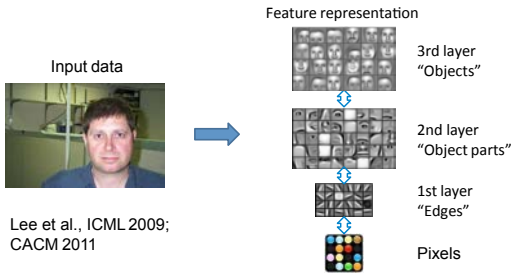
---

---

---

### Learning Feature Hierarchy

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



Lee et al., ICML 2009;  
CACM 2011

Slide: Rob Fergus

---

---

---

---

---

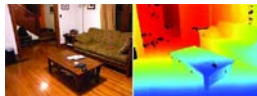
---

---

---

### 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

---

---

---

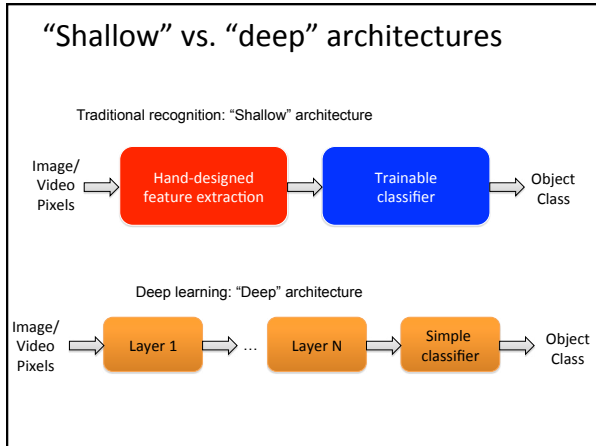
---

---

---

---

---




---

---

---

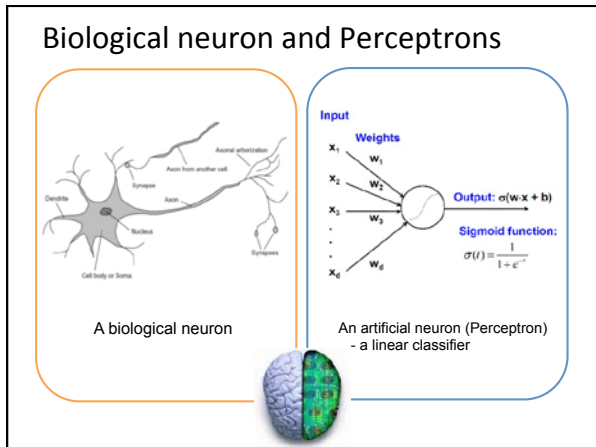
---

---

---

---

---




---

---

---

---

---

---

---

---

### Simple, Complex, and Hyper-complex cells

[video](#)

David H. Hubel and Torsten Wiesel

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](#)

---

---

---

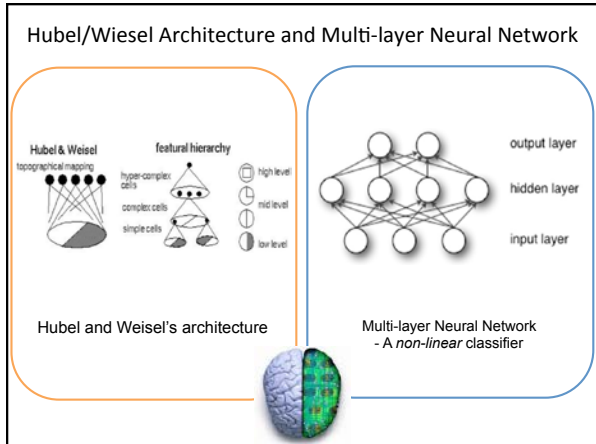
---

---

---

---

---




---

---

---

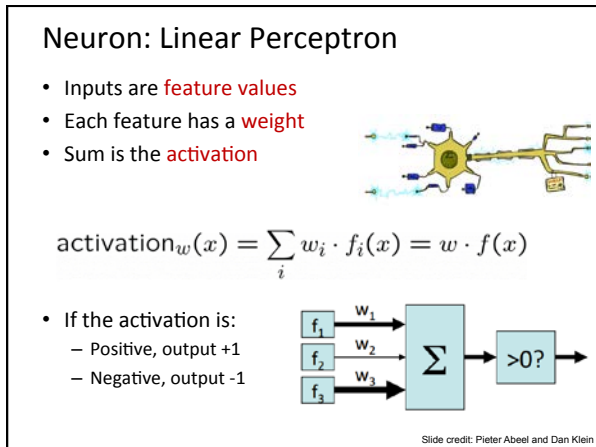
---

---

---

---

---




---

---

---

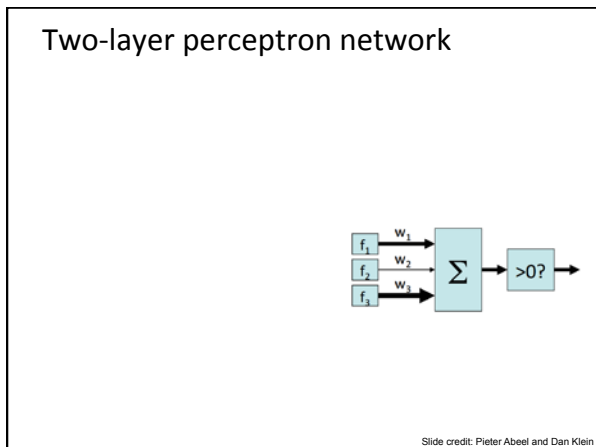
---

---

---

---

---




---

---

---

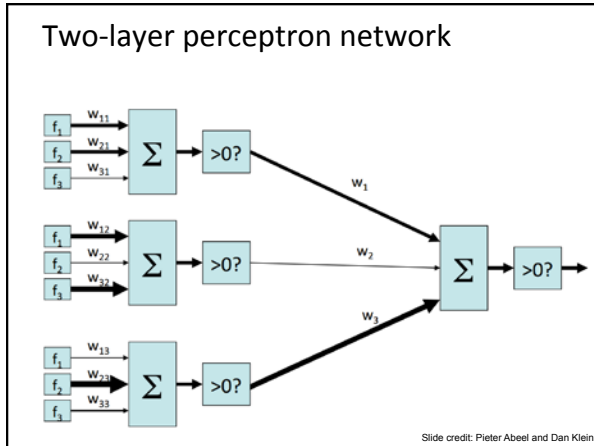
---

---

---

---

---




---

---

---

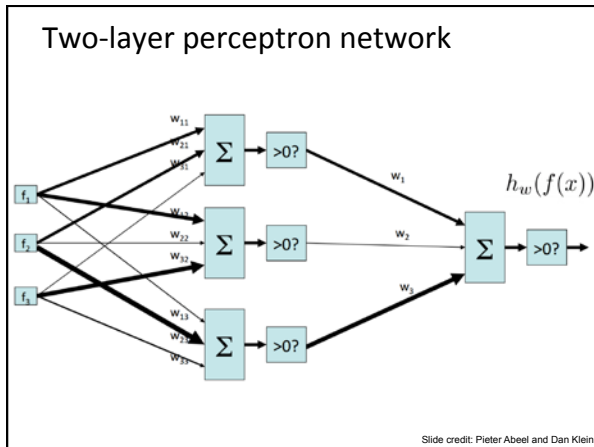
---

---

---

---

---




---

---

---

---

---

---

---

---

### 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

---

---

---

---

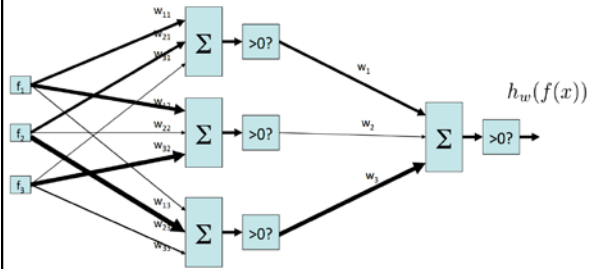
---

---

---

---

### Two-layer perceptron network



Slide credit: Pieter Abeel and Dan Klein

---

---

---

---

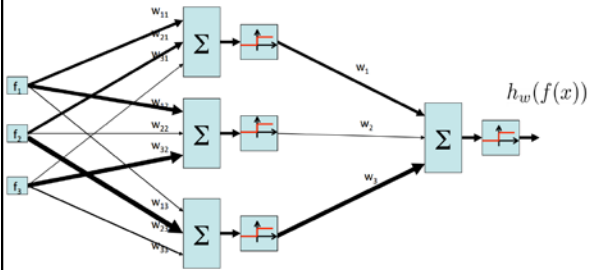
---

---

---

---

### Two-layer perceptron network



Slide credit: Pieter Abeel and Dan Klein

---

---

---

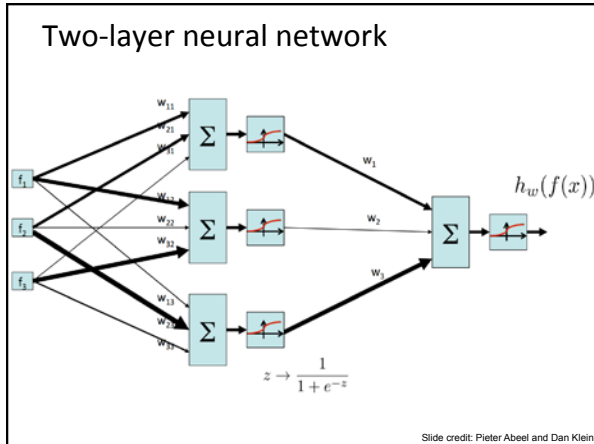
---

---

---

---

---




---

---

---

---

---

---

---

---

### 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**

output layer  
hidden layer  
input layer

---

---

---

---

---

---

---

---



### 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

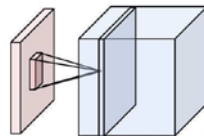


Image credit: A. Karpathy

---

---

---

---

---

---

---

---

### Neocognitron [Fukushima, Biological Cybernetics 1980]

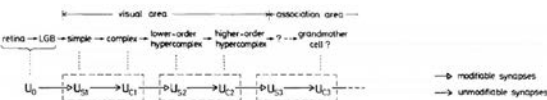
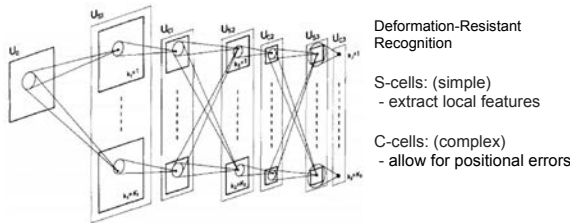


Fig. 1. Correspondence between the hierarchy model by Hubel and Wiesel, and the neural network of the neocognitron.




---

---

---

---

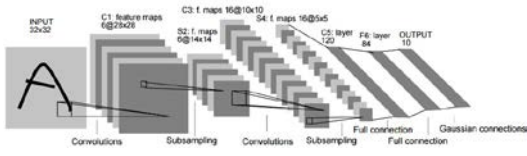
---

---

---

---

### 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

---

---

---

---

---

---

---

---

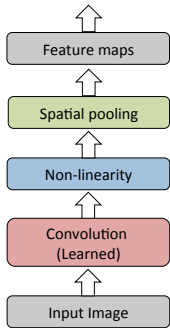
---

---

---

---

### Convolutional Neural Networks




---

---

---

---

---

---

---

---

---

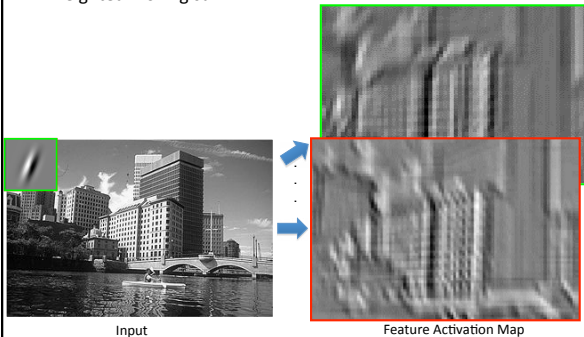
---

---

---

### What is a Convolution?

- Weighted moving sum




---

---

---

---

---

---

---

---

---

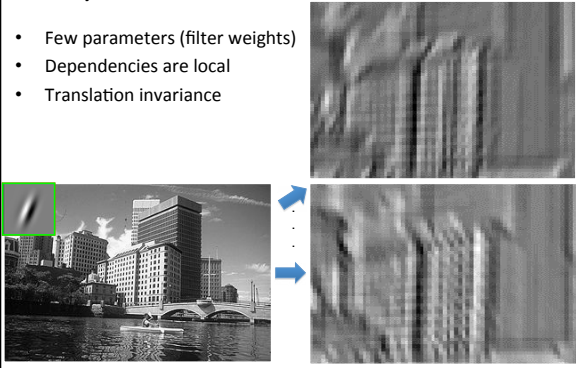
---

---

---

### Why Convolution?

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



Input Feature Map

---

---

---

---


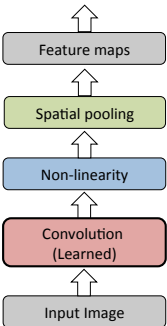
---

---

---

---

### Convolutional Neural Networks



Input Feature Map

---

---

---

---

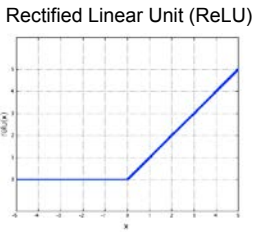
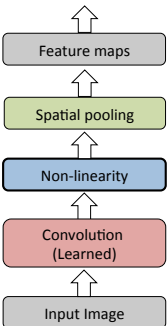
---

---

---

---

### Convolutional Neural Networks



Rectified Linear Unit (ReLU)

slide credit: S. Lazebnik

---

---

---

---

---

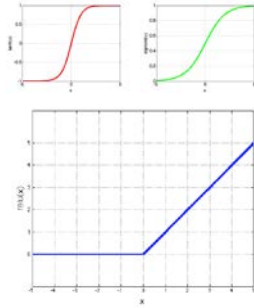
---

---

---

### 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




---

---

---

---

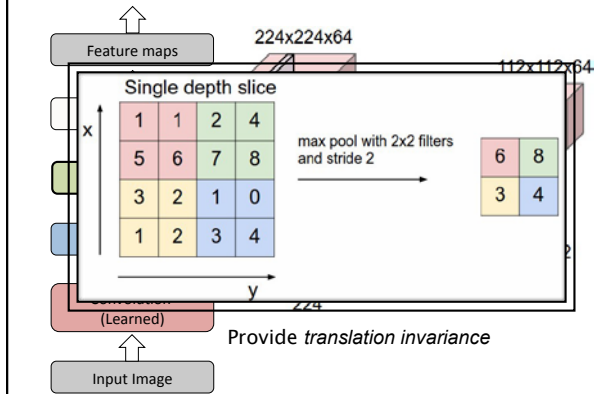
---

---

---

---

### Convolutional Neural Networks




---

---

---

---

---

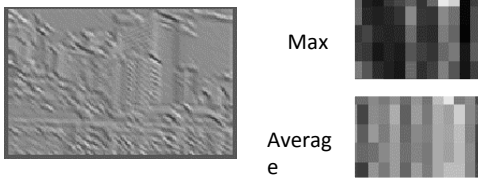
---

---

---

### Spatial Pooling

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




---

---

---

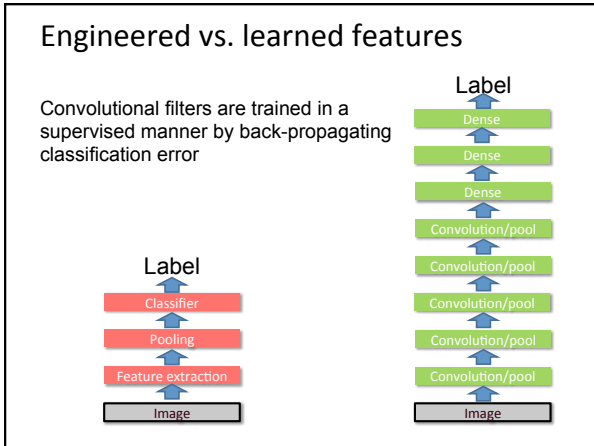
---

---

---

---

---



---

---

---

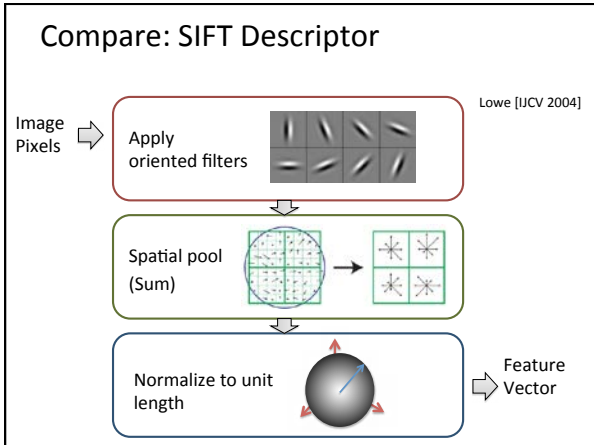
---

---

---

---

---



---

---

---

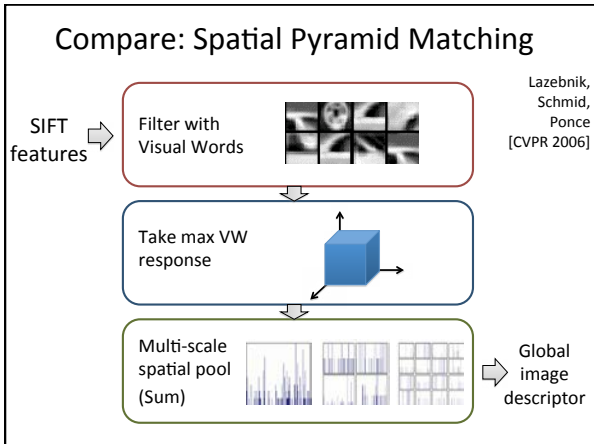
---

---

---

---

---



---

---

---

---

---

---

---

---

### Previous 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]




---

---

---

---

---

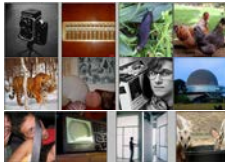
---

---

---

### ImageNet Challenge 2012

#### IMAGENET



- ~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

---

---

---

---

---

---

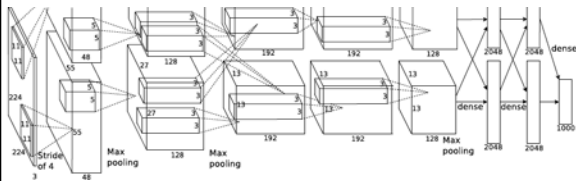
---

---

### AlexNet

Similar framework to LeCun'98 but:

- Bigger model (7 hidden layers, 650,000 units, 60,000,000 params)
- More data ( $10^6$  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

Fixed input size: 224x224x3

---

---

---

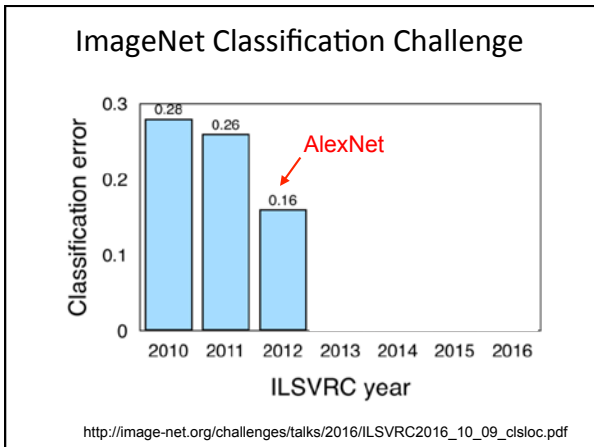
---

---

---

---

---




---

---

---

---

---

---

---

---

### Industry Deployment

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

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

---

---

---

---

---

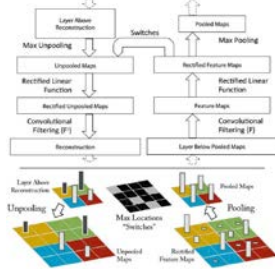
---

---

---

### Visualizing CNNs

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



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

---

---

---

---

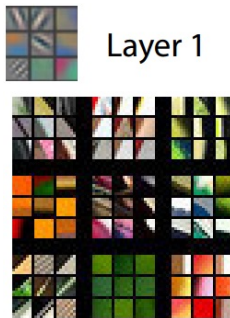
---

---

---

---

### Layer 1



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

---

---

---

---

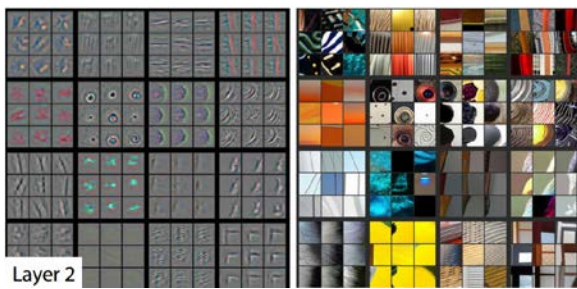
---

---

---

---

### Layer 2



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

---

---

---

---

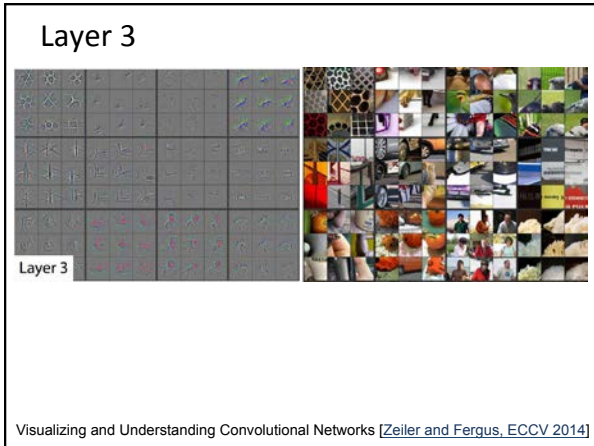
---

---

---

---





---

---

---

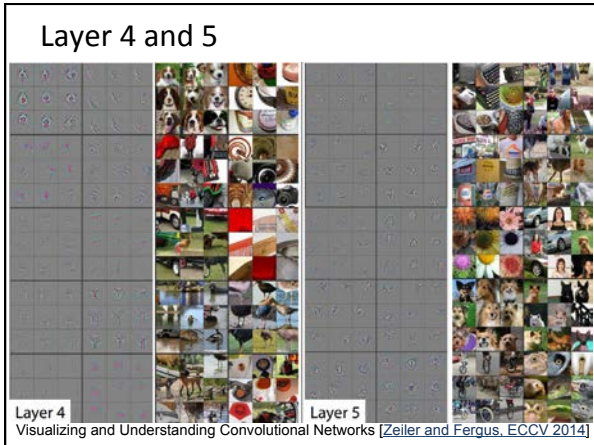
---

---

---

---

---



---

---

---

---

---

---

---

---

### Beyond classification

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

and many more...

---

---

---

---

---

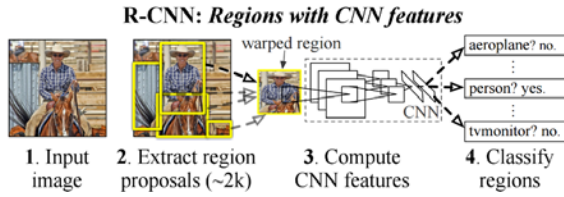
---

---

---

### R-CNN: Regions with CNN features

- Trained on ImageNet classification
- Finetune CNN on PASCAL



RCNN [Girshick et al. CVPR 2014]

---

---

---

---

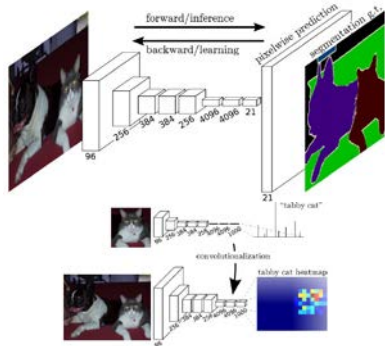
---

---

---

---

### Labeling Pixels: Semantic Labels



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

---

---

---

---

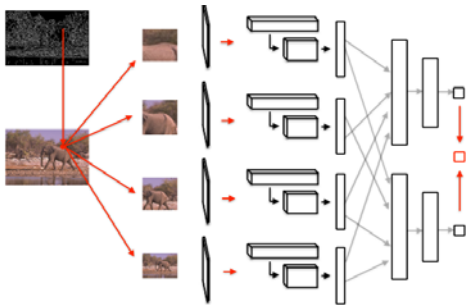
---

---

---

---

### 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]

---

---

---

---

---

---

---

---

### CNN for Image Generation

Learning to Generate Chairs with Convolutional Neural Networks [Dosovitskiy et al. CVPR 2015]

---

---

---

---

---

---

---

---

### Chair Morphing



Learning to Generate Chairs with Convolutional Neural Networks [Dosovitskiy et al., CVPR 2015]

---

---

---

---

---

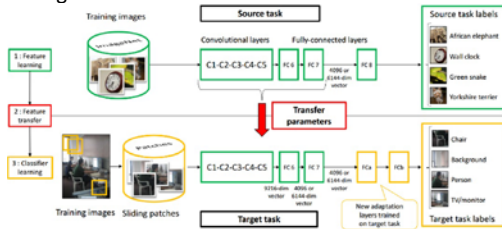
---

---

---

### 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

correct +distort ostrich    correct +distort ostrich    correct +distort ostrich

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]

---

---

---

---

---

---

---

---

---

---

---

---

### What is going on?

“panda” 57.7% confidence    “nematode” 8.2% confidence    “gibbon” 99.3% confidence

$x + .007 \times \frac{\partial E}{\partial x} = \text{gibbon}$

$x \leftarrow x + \alpha \frac{\partial E}{\partial x}$

Explaining and Harnessing Adversarial Examples [Goodfellow ICLR 2015]  
<http://karpathy.github.io/2015/03/30/breaking-convnets/>

---

---

---

---

---

---

---

---

---

---

---

---

### Questions?

See you Tuesday!

63

---

---

---

---

---

---

---

---

---

---

---

---