

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/298263303>

Introduction to Deep Convolutional Neural Networks

Presentation · March 2016

CITATIONS

3

READS

2,523

1 author:



Asifullah Khan

Pakistan Institute of Engineering and Applied Sciences

234 PUBLICATIONS 4,580 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Project Membrane Protein Prediction [View project](#)



Project Malware Classification [View project](#)

Introduction to Deep Convolutional Neural Networks

Dr. Asifullah Khan,
Professor,
DCIS, PIEAS, Islamabad,
Pakistan

Outline

- Artificial Neurons and Perceptron
- Building blocks of CNN
 - Convolution
 - Pooling
- Motivations for CNN
- Working of a CNN
- Why use Hierarchical Learning
- Visualizing CNN Activations (features)
- Deep CNN architectures: case studies
- Transfer Learning

Biological neuron

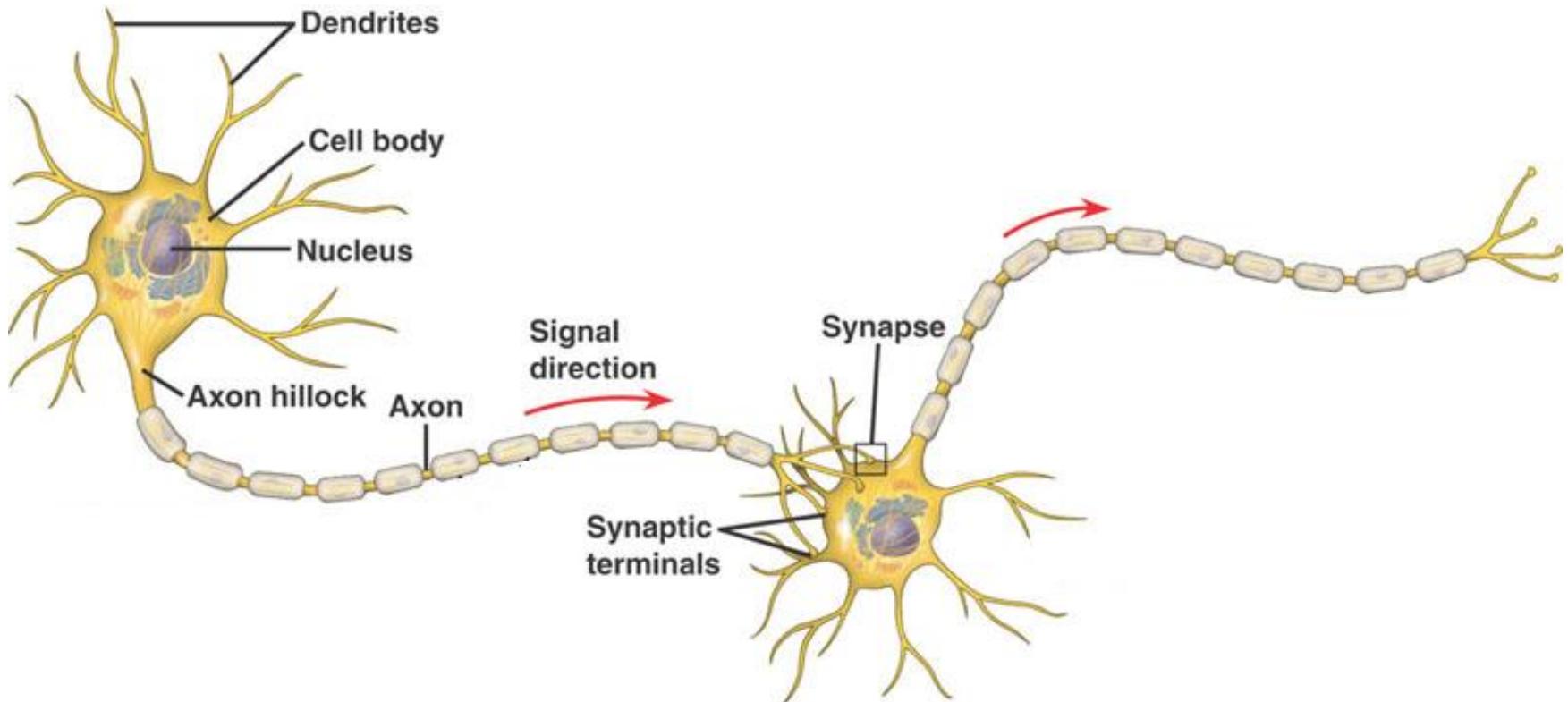
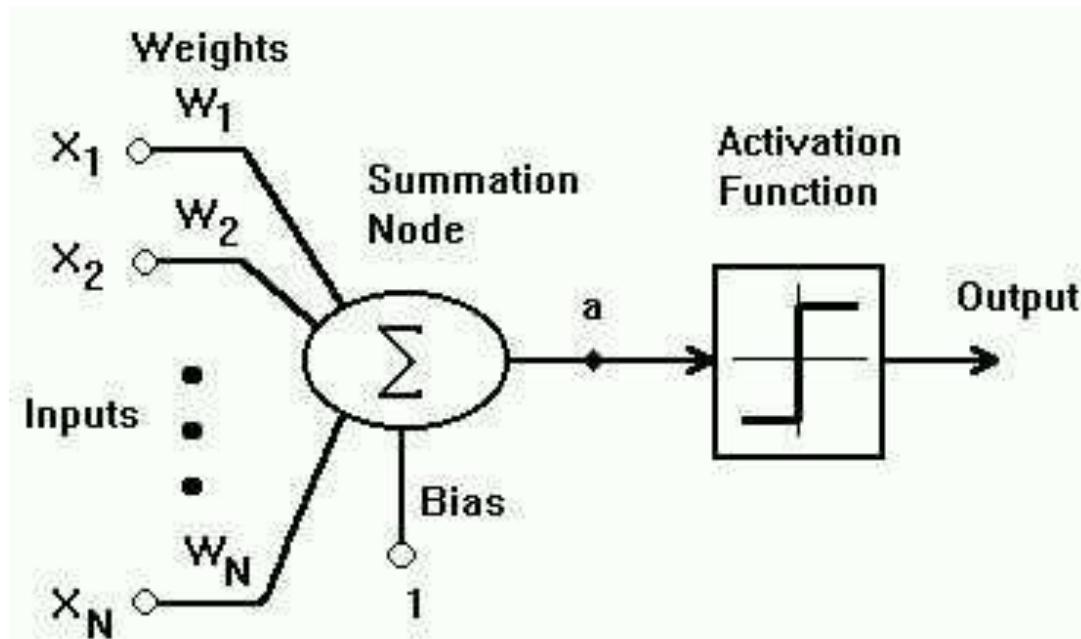


Image from: <http://hplusmagazine.com/2012/10/17/four-statements-about-the-future>

Single layer perceptron



$$a = w_1x_1 + w_2x_2 + \dots + w_Nx_N + \text{Bias}$$

output = Threshold(a)

where $\text{Threshold}(a) = \begin{cases} -1, & \text{for all } a \leq 0 \\ 1, & \text{for all } a > 0 \end{cases}$

Image from: <https://battleprogrammer.wordpress.com/2011/03/23/jaringan-syaraf-tiruan-apa-apa-apa/>

Multi-layered Neural Net

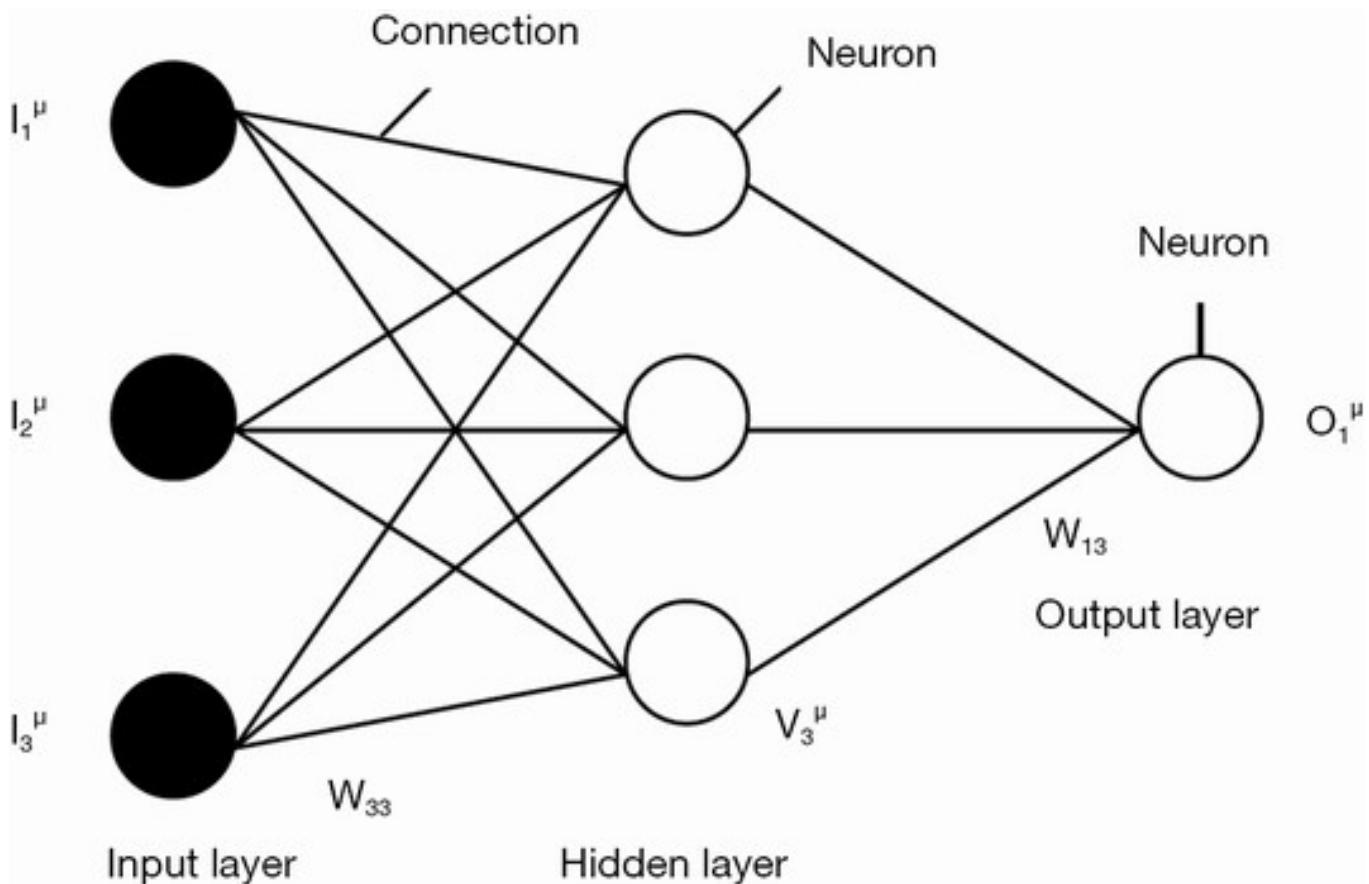


Image from: Pasini, Antonello. "Artificial neural networks for small dataset analysis." *Journal of thoracic disease* 7.5 (2015): 953.

Convolution

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Image

1	0	1
0	1	0
1	0	1

Filter/kernel

Convolution

1 <small>x1</small>	1 <small>x0</small>	1 <small>x1</small>	0	0
0 <small>x0</small>	1 <small>x1</small>	1 <small>x0</small>	1	0
0 <small>x1</small>	0 <small>x0</small>	1 <small>x1</small>	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved
Feature

Convolution

1	1 <small>×1</small>	1 <small>×0</small>	0 <small>×1</small>	0
0	1 <small>×0</small>	1 <small>×1</small>	1 <small>×0</small>	0
0	0 <small>×1</small>	1 <small>×0</small>	1 <small>×1</small>	1
0	0	1	1	0
0	1	1	0	0

Image

4	3	

Convolved
Feature

Convolution

1	1	1 _{x1}	0 _{x0}	0 _{x1}
0	1	1 _{x0}	1 _{x1}	0 _{x0}
0	0	1 _{x1}	1 _{x0}	1 _{x1}
0	0	1	1	0
0	1	1	0	0

Image

4	3	4

Convolved
Feature

Convolution

1	1	1	0	0
0 <small>$\times 1$</small>	1 <small>$\times 0$</small>	1 <small>$\times 1$</small>	1	0
0 <small>$\times 0$</small>	0 <small>$\times 1$</small>	1 <small>$\times 0$</small>	1	1
0 <small>$\times 1$</small>	0 <small>$\times 0$</small>	1 <small>$\times 1$</small>	1	0
0	1	1	0	0

Image

4	3	4
2		

Convolved
Feature

Convolution

1	1	1	0	0
0	1	1	1	0
0	0	1 _{x1}	1 _{x0}	1 _{x1}
0	0	1 _{x0}	1 _{x1}	0 _{x0}
0	1	1 _{x1}	0 _{x0}	0 _{x1}

Image

$$W_{\text{conv}} = W_{\text{img}} - W_{\text{filt}} + 1$$
$$H_{\text{conv}} = H_{\text{img}} - H_{\text{filt}} + 1$$

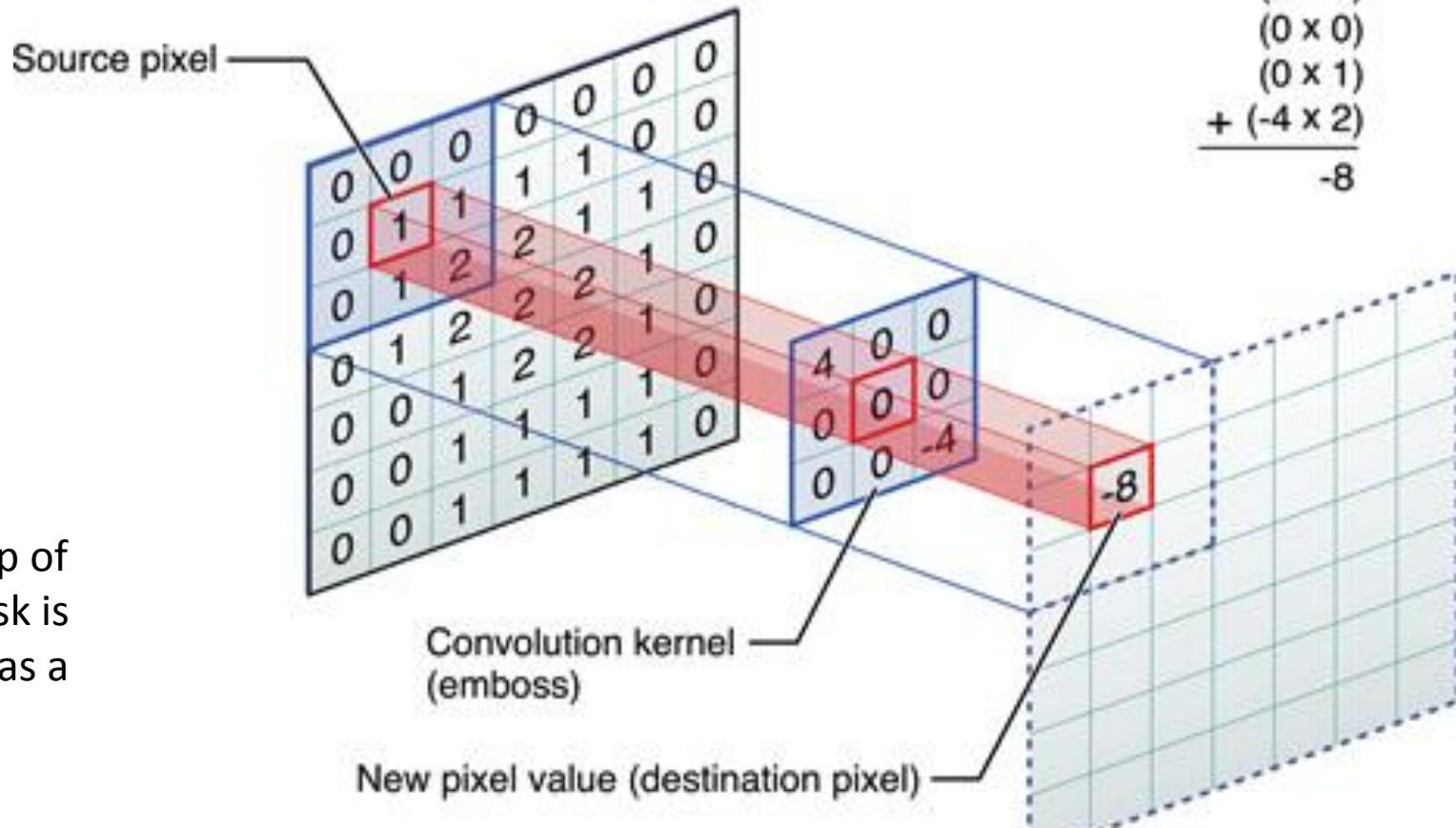
4	3	4
2	4	3
2	3	4

Convolved Feature

Convolution and Correlation becomes the same if the Kernel is symmetric; no effect from flipping.
In Machine Learning community, mostly correlation Kernels are used.

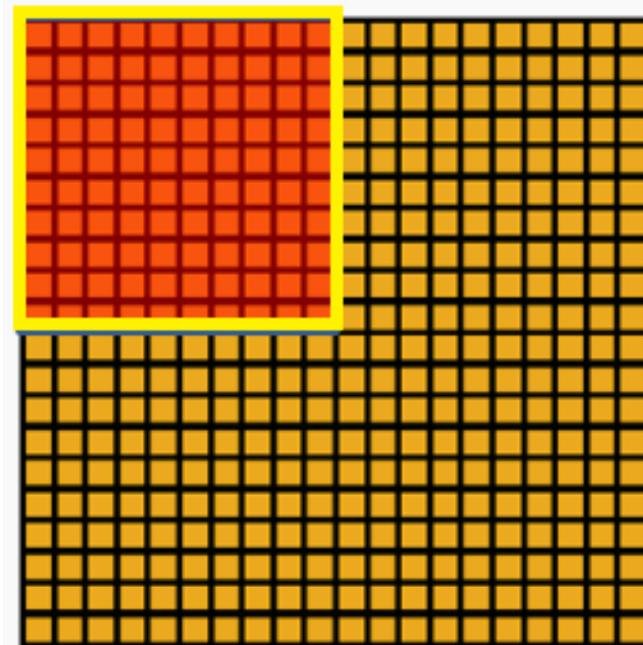
Convolution

Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.

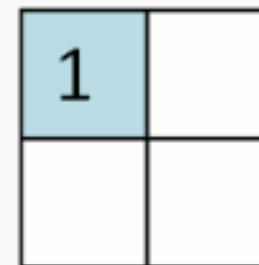


The step of
the mask is
known as a
stride

Pooling

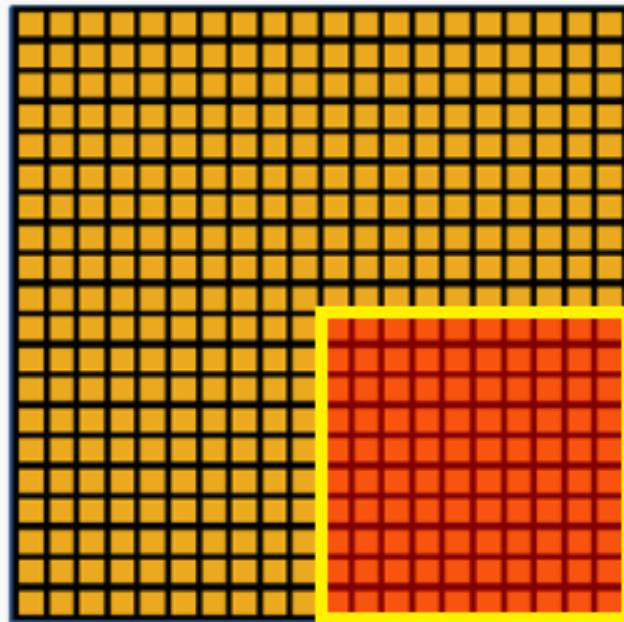


Convolved
feature



Pooled
feature

Pooling

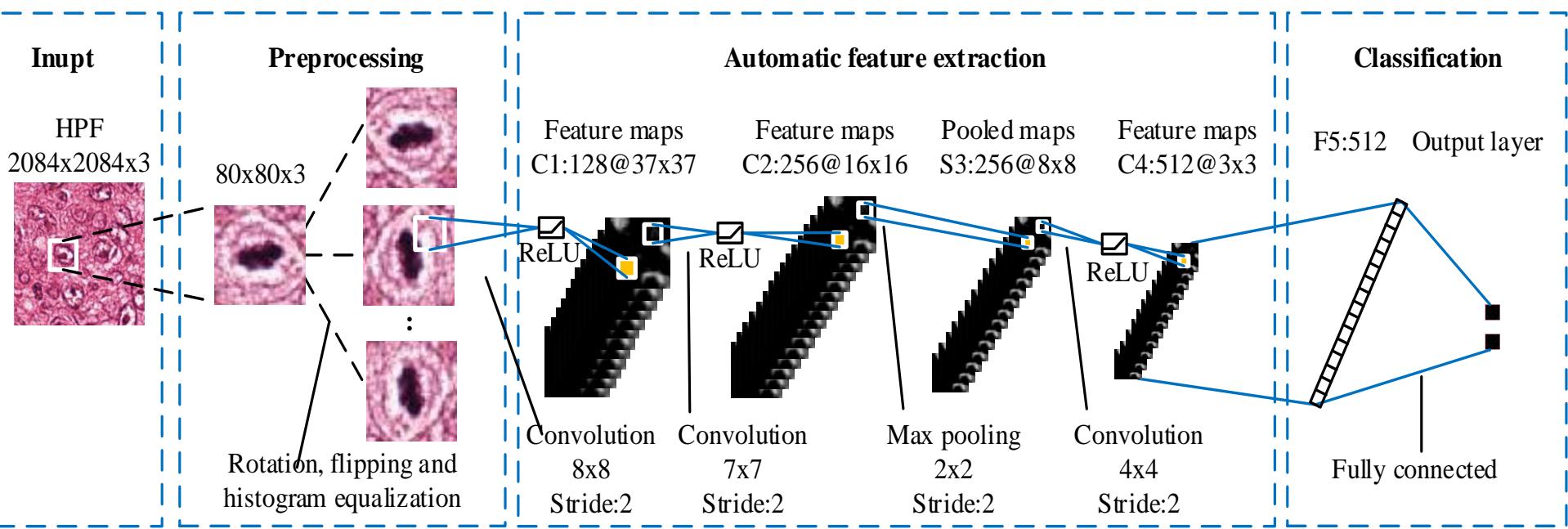


Convolved
feature

1	7
5	9

Pooled
feature

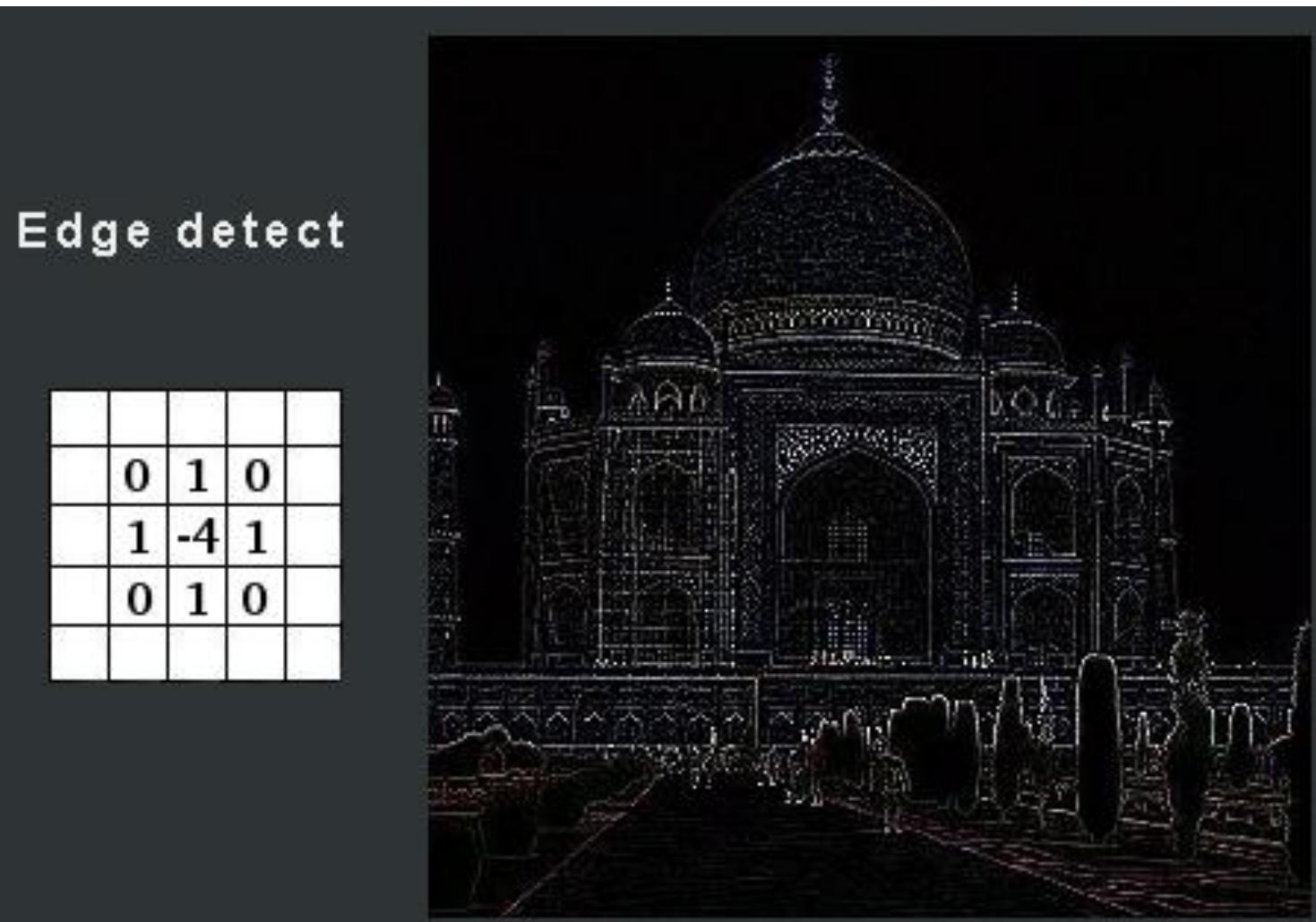
One of our Proposed CNN Architecture;



$$\text{featureMapSize} = [(\text{inputsize} - \text{filtersize} + 2 \times \text{zeropadding}) / \text{stride}] + 1$$

$$\text{pooledMapSize} = [(\text{inputsize} - \text{filtersize}) / \text{stride}] + 1$$

Filters/Kernels/Mask/Feature-extractor



Static/hand-made Filters

Motivation for CNN

- **Invariant Features:** Detection and classification independent of pose, scale, illumination, occlusion and clutter
- **Dynamic Feature Extraction:** how could an artificial vision system learn appropriate internal representations automatically, the way humans seem to by simply looking at the world?
- **Reduced Learnable Parameters** compared to a BackPropagator.
- **Hierarchical Learning**

Motivation for CNN; Invariance

Goal

- Detection and classification independent of pose, scale, illumination, occlusion and clutter

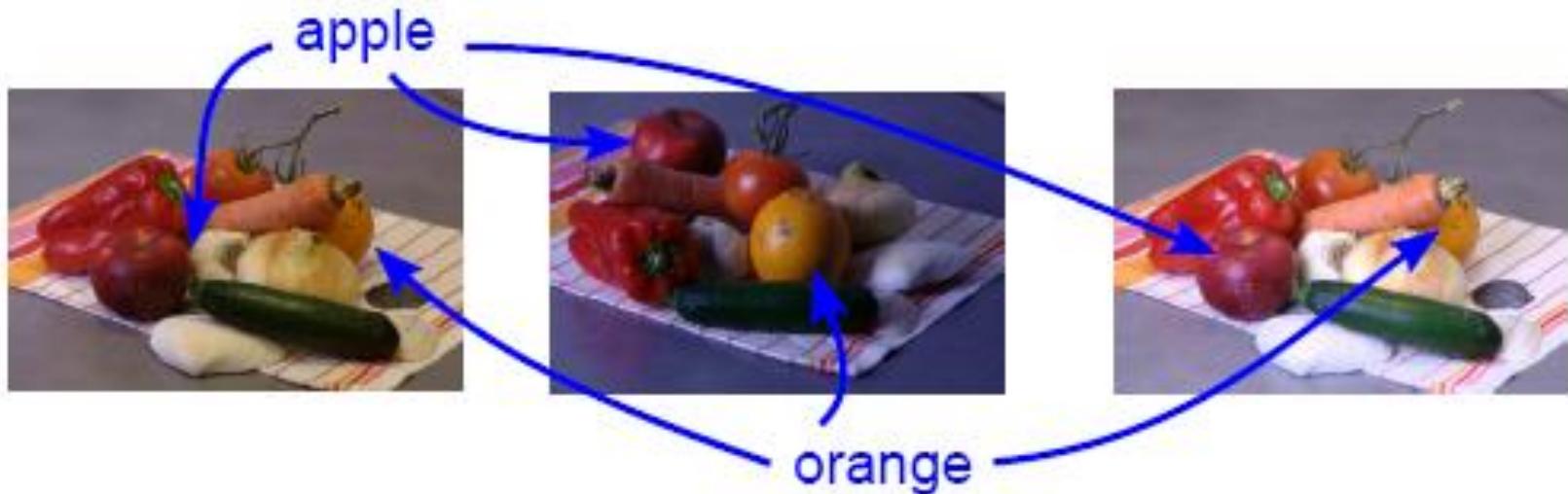


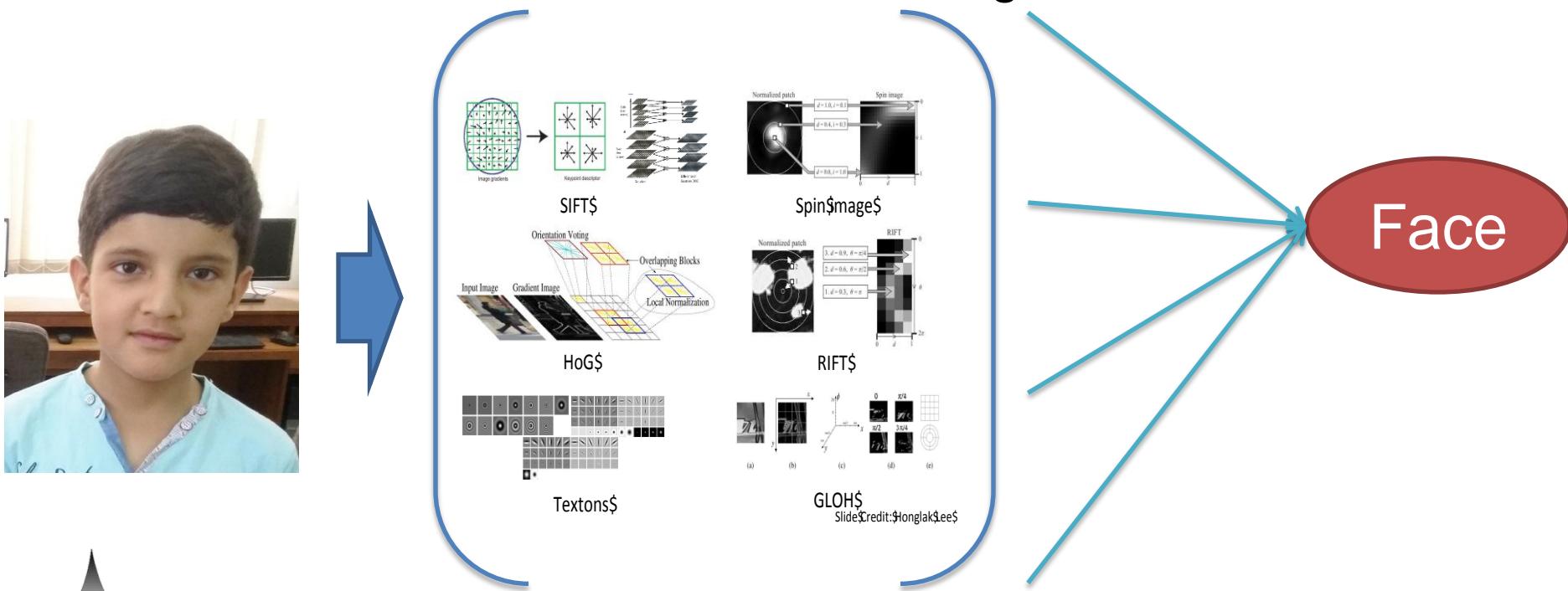
Image from: Dr. Richard E. Turner presentation (2014)

Static Feature Extraction: Standard image classification approach

Input



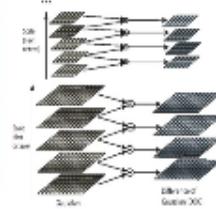
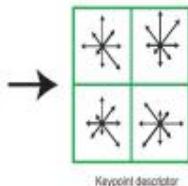
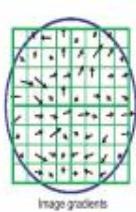
Extract features



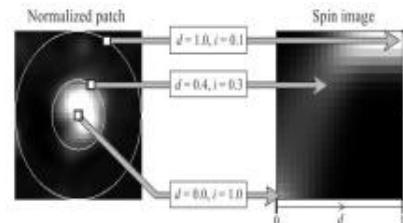
Use simple classifier
e.g., logistic regression, SVMs

Face

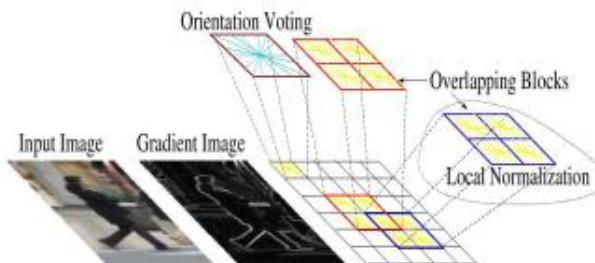
Static Feature Extraction: many hand crafted Features exist...



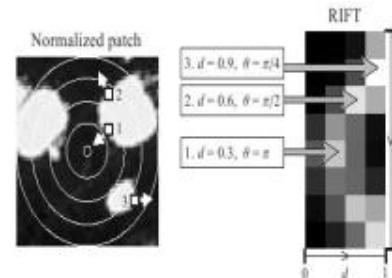
SIFT\$



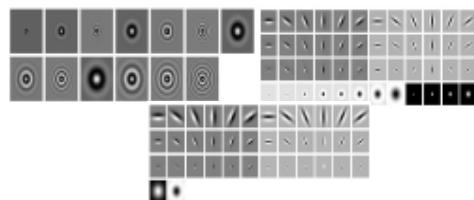
Spin\$image\$



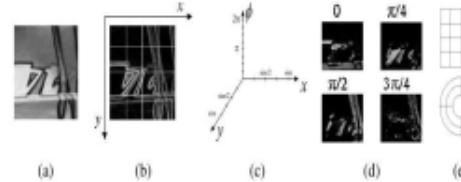
HoG\$



RIFT\$



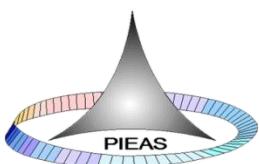
Textons\$



GLOH\$

Slide Credit: Honglak Lee\$

... but very painful to design



Dynamic Feature Extraction: Change Image Classification Approach?

Input

Extract features dynamically

Use simple classifier
e.g., SVM, etc.

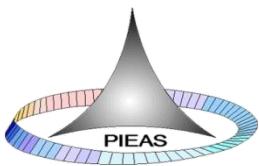


Can we automatically learn
features
from data?

Face

Textons\$

GLOH\$
Slide Credit: Honglak Lee\$



Motivation for CNN; Less Parameters

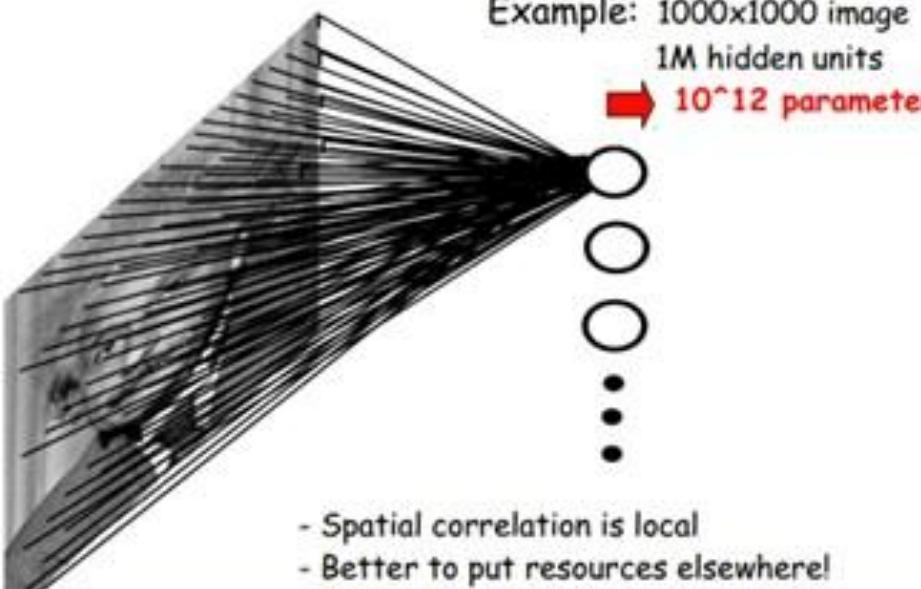
Cons of multilayer perceptron

- The number of **trainable parameters** becomes extremely large. For example, a 24×24 input layer would already have 576 connections per single neuron in the hidden layer
- Secondly, it offers little or **no invariance to shifting, scaling, and other forms of distortion**
- Hand-crafted feature extraction requires a great deal of time

Motivation for CNN; Less Parameters

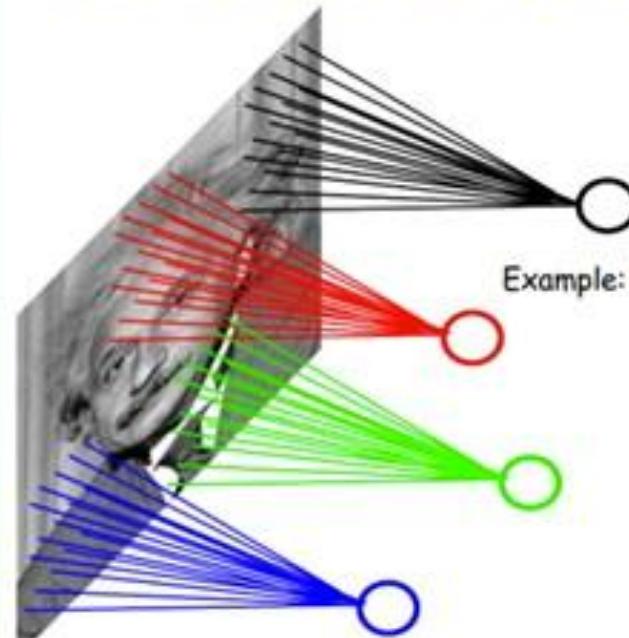
FULLY CONNECTED NEURAL NET

Example: 1000x1000 image
1M hidden units
→ 10^{12} parameters!!!



LOCALLY CONNECTED NEURAL NET

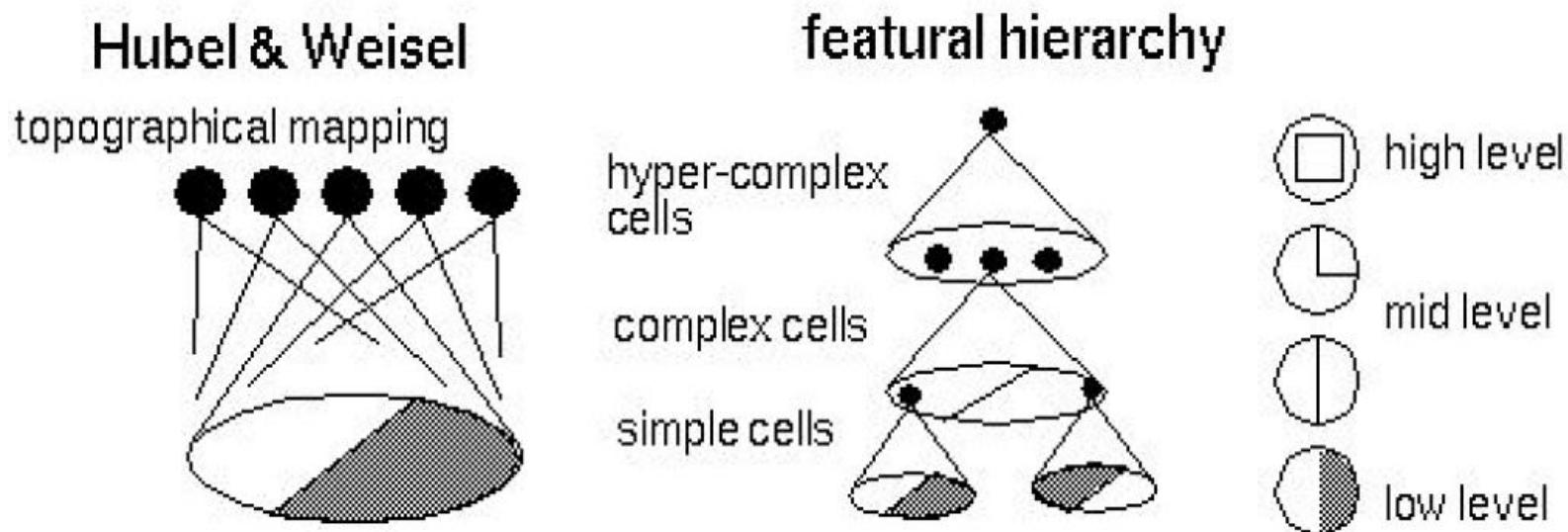
Example: 1000x1000 image
1M hidden units
Filter size: 10x10
100M parameters



Ranzo

Motivation for CNN; Hierarchical Learning

- Hubel/Wiesel Architecture
- D. Hubel and T. Wiesel (Nobel Prize 1981)
- Visual cortex consists of a hierarchy of *simple*, *complex*, and *hyper-complex* cells



Motivation for CNN: Hierarchical Learning

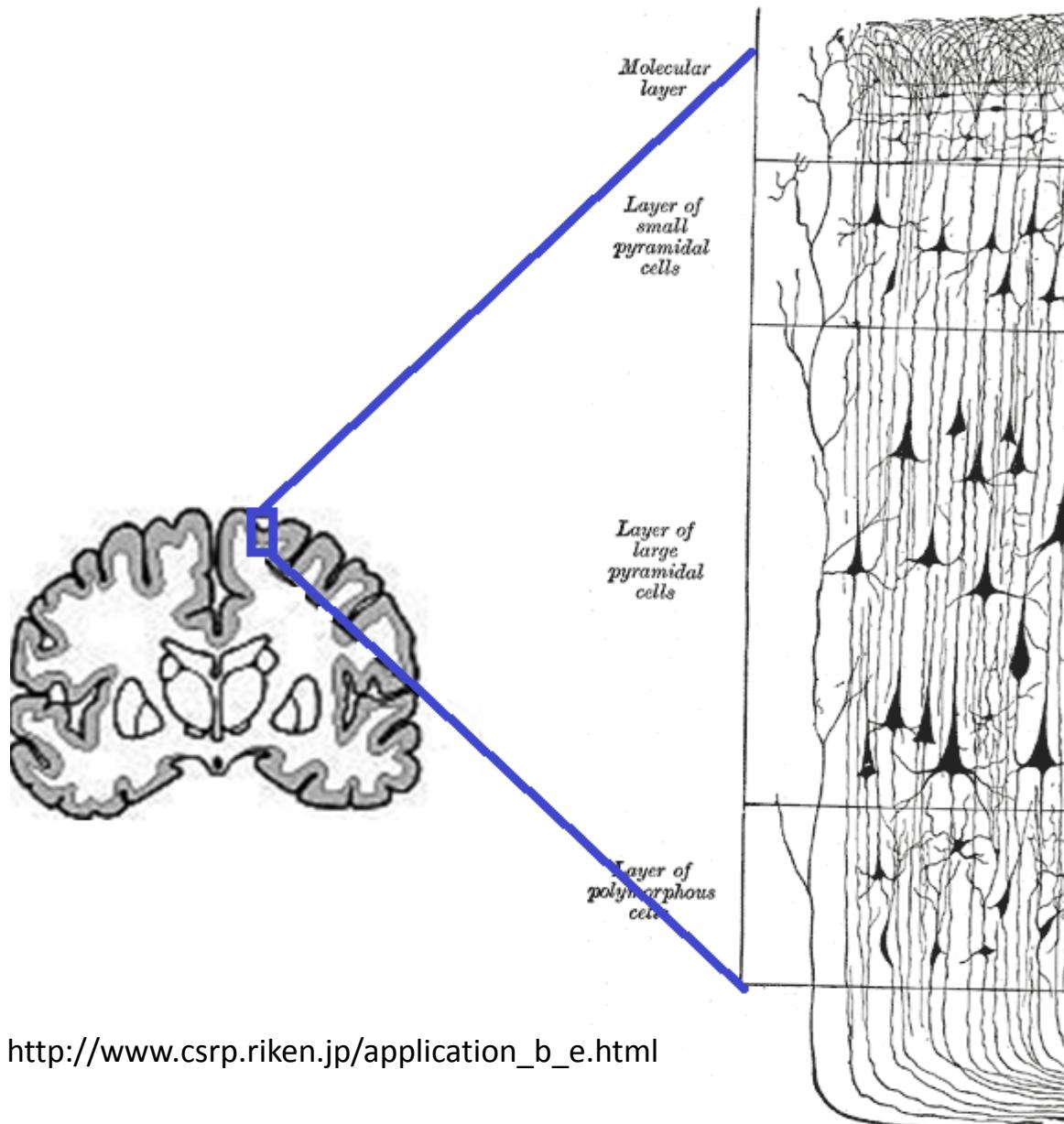


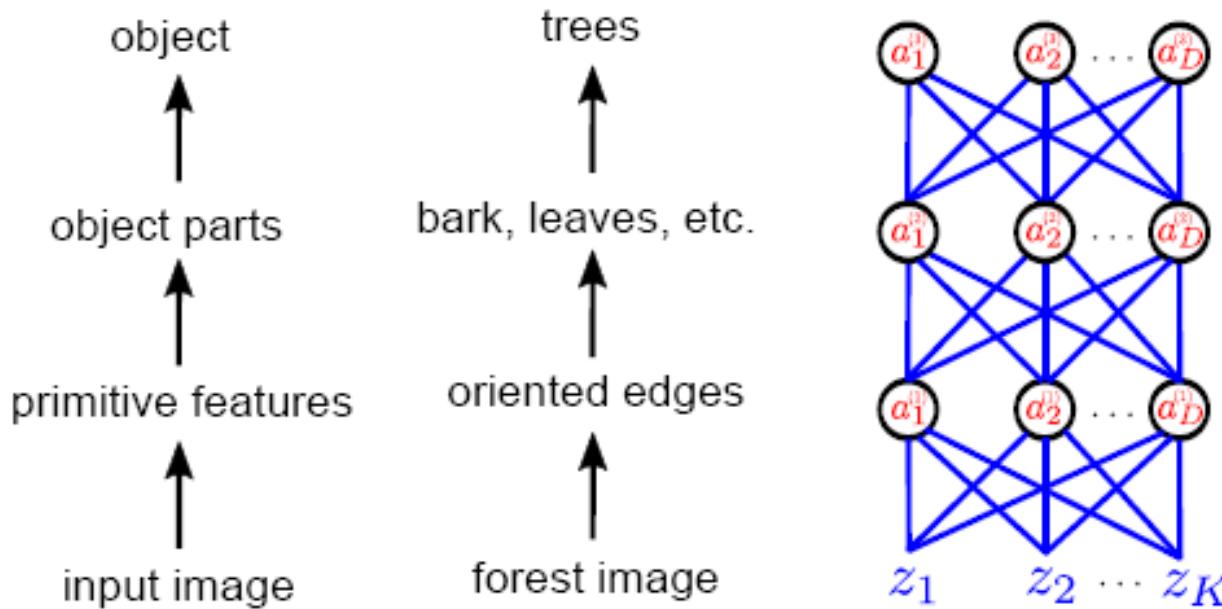
Image from: http://www.csrp.riken.jp/application_b_e.html

Motivation for CNN; Hierarchical Learning

- In 1995, Yann LeCun and Yoshua Bengio introduced the concept of CNN
- Neurobiologically motivated by the findings of **locally sensitive** and orientation-selective nerve cells in the **visual cortex** of the cat

Motivation for CNN; Why use hierarchical multi-layered models?

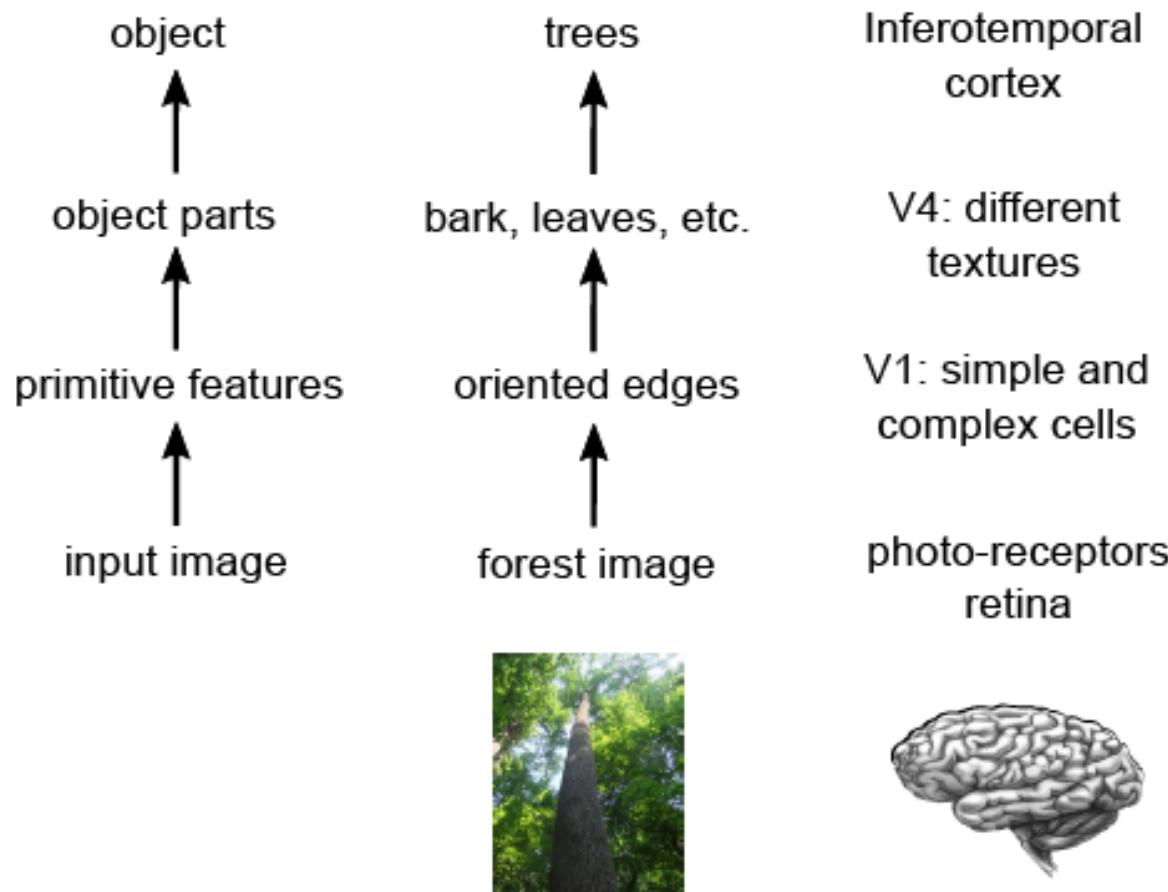
- visual scenes are hierarchically organized



Slide credit: Dr. Richard E. Turner presentation (2014)

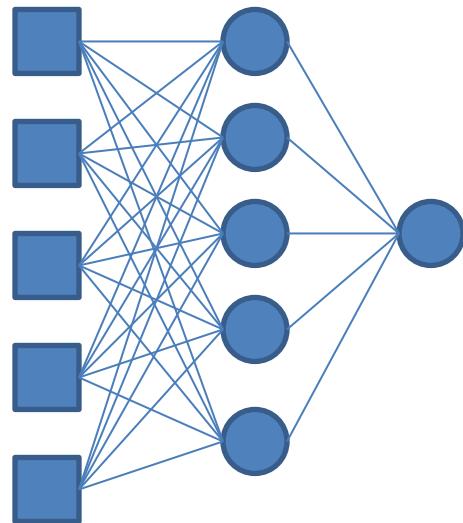
Motivation for CNN; Why use hierarchical multi-layered models?

- biological vision is hierarchically organized

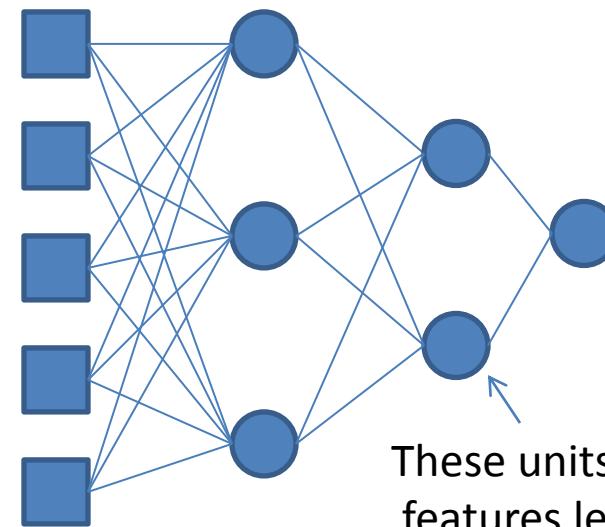


Motivation for CNN; Why use hierarchical multi-layered models?

- Shallow architectures are inefficient at representing deep functions
- Deep net, deep (enriched) features



Params: $5 \times 5 + 5 = 30$



Params: $5 \times 3 + 6 + 2 = 23$

These units fine-tune the features learned by those in the previous layer

Motivation for CNN; Hierarchical Representation Example

very high level representation:

MAN SITTING ...



slightly higher level representation

raw input vector representation:

$$\mathcal{X} = [x_1 \ x_2 \ x_3 \ \dots \ x_n]$$

$x_1 = 23$ $x_2 = 19$ $x_3 = 20$... $x_n = 18$



Yoshua Bengio (2009)

Working of Convolutional Neural Networks

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

Working of Convolutional Neural Network

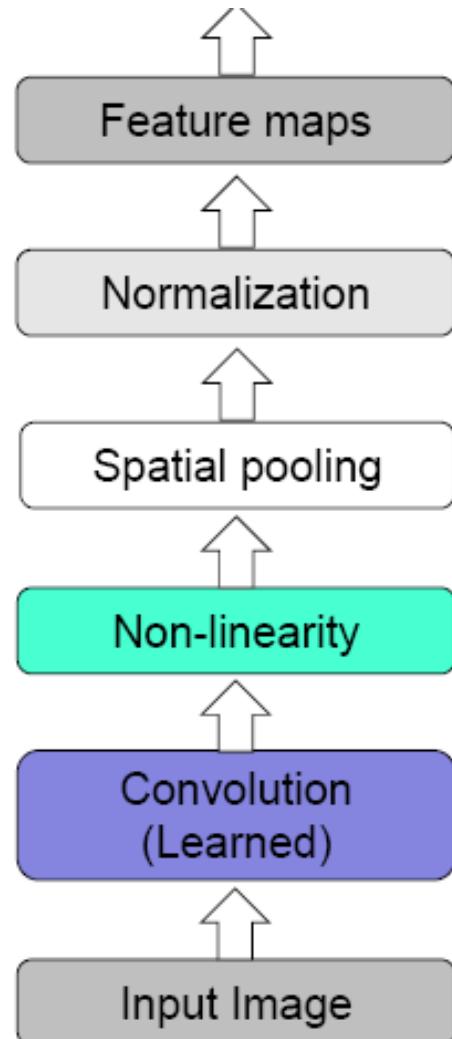
- Feed-forward feature extraction:
- 1. Convolve input with filters
- 2. Non-linearity (ReLU)
- 3. Spatial pooling
- 4. Normalization

Supervised training of convolutional filters by back-propagating classification error.

A CNN can be both Generative and Discriminative Learning Mechanism

Working of Convolutional Neural Network

- Feed-forward feature extraction:



Working of Convolutional Neural Network

- CNN is a type of feed-forward artificial neural network.
- Individual neurons are **tiled** in such a way that they respond to **overlapping regions** in the **visual field**
- The **lower layers** obtain **low-level features** (like pixels, edges, lines and corners) while the **higher layers** obtain **high-level features** (like shapes)
- The more layers the network has, the higher-level features it will get

Working of Convolutional Neural Network; **Convolutional layer**

- Unlike a hand-coded **convolution kernel** (Sobel, Prewitt, Roberts), in a CNN, the parameters of each convolution kernel are trained by the backpropagation algorithm
- Convolution operators extract **different features of the input**
- There are many convolution kernels in each layer, and **each kernel is replicated over the entire image with the same parameters (weights and bias)**

Working of Convolutional Neural Network: Convolutional layer

- Dependencies are local
- Translation invariance
- Few parameters (filter weights)
- Stride can be greater than 1 (faster, less memory)



Input



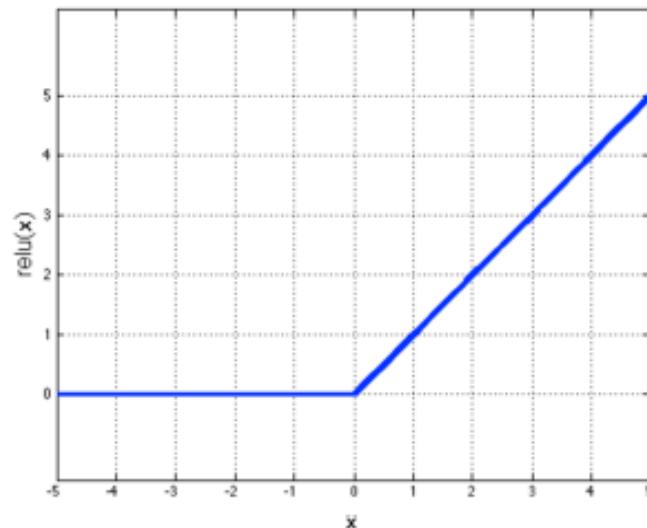
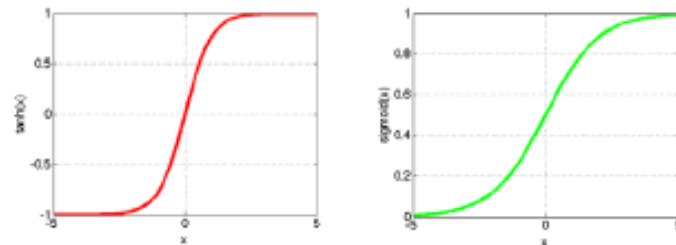
Feature Map

Slide credit: Rob Fergus (NIPS 2013 tutorial)

http://cs.nyu.edu/~fergus/presentations/nips2013_final.pdf

Working of Convolutional Neural Network: Non-linearity

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

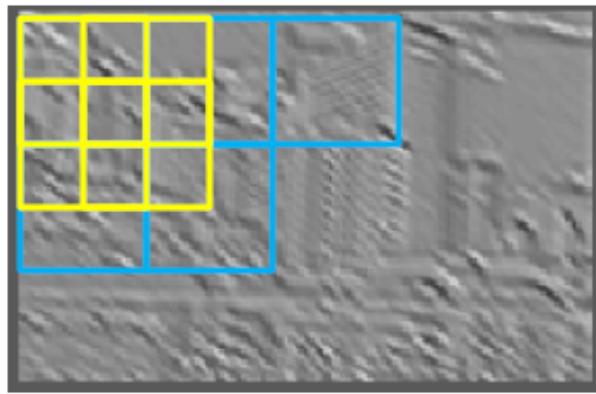


Working of Convolutional Neural Network: Pooling Layer

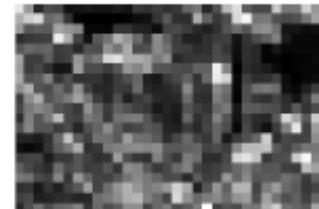
- In order to reduce variance, pooling layers compute the max or average value of a particular feature over a region of the image
- This will ensure that the same result will be obtained, even when image features have **small translations**
- This is an important operation for object classification and detection

Working of Convolutional Neural Network: Pooling Layer

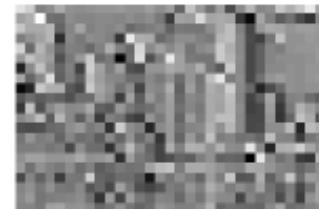
- Sum or max
- Non-overlapping / overlapping regions
- Role of pooling:
 - Invariance to small transformations
 - Larger receptive fields (see more of input)



Max

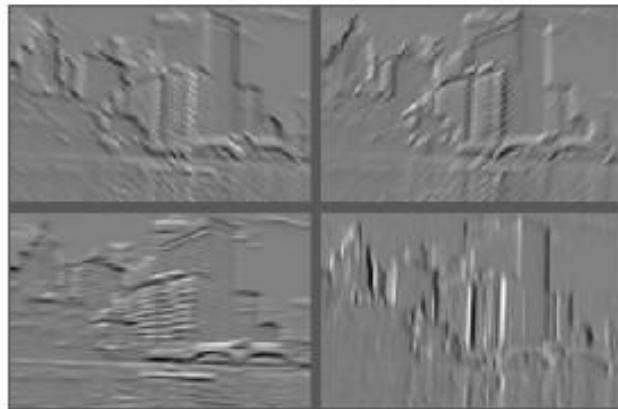


Sum

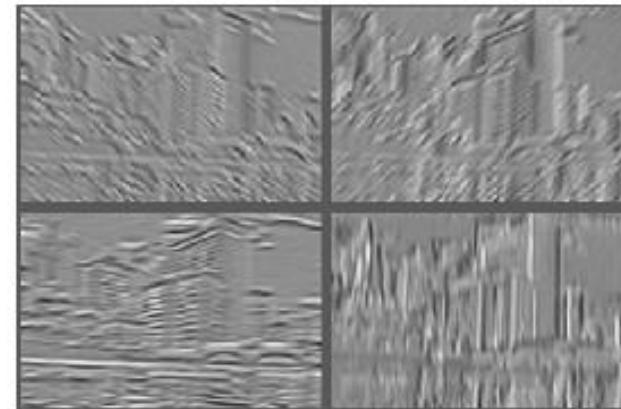


Working of Convolutional Neural Network: Normalization

- Within or across feature maps
- Before or after spatial pooling



Feature Maps



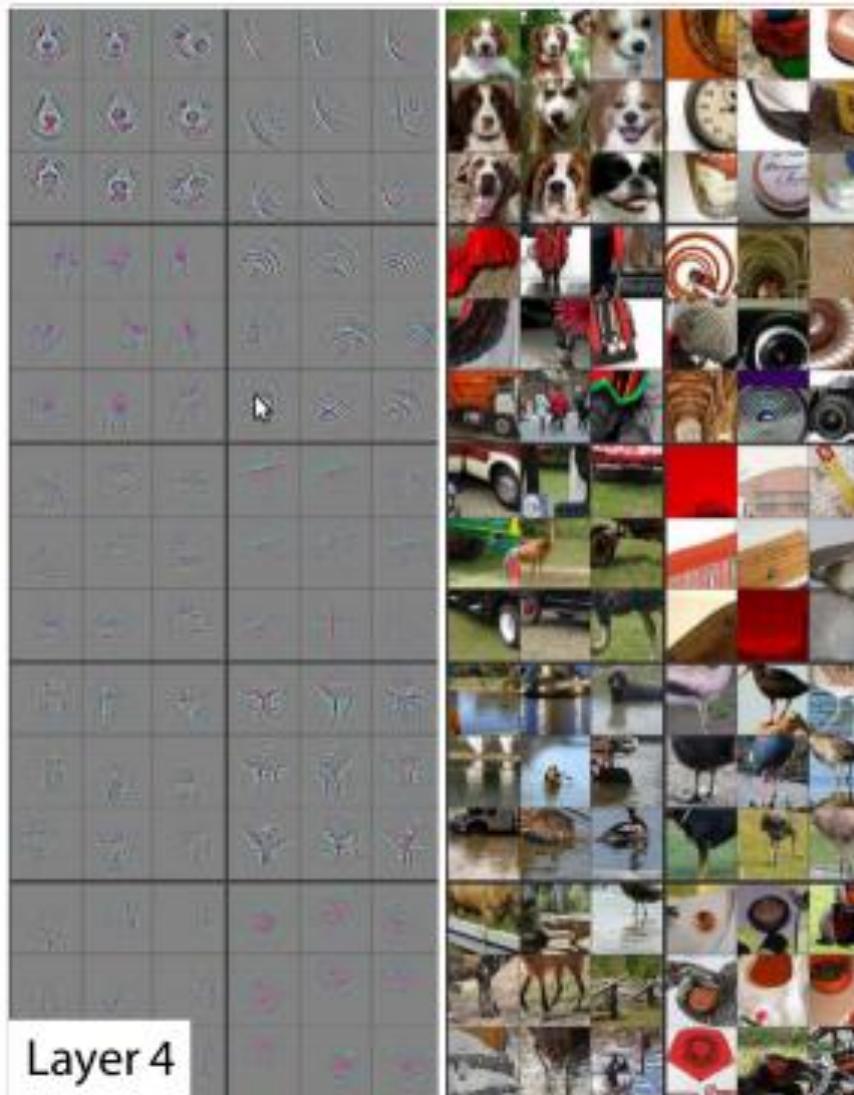
**Feature Maps
After Contrast Normalization**

Slide credit: Rob Fergus (NIPS 2013 tutorial)

http://cs.nyu.edu/~fergus/presentations/nips2013_final.pdf

Visualizing activations in CNN

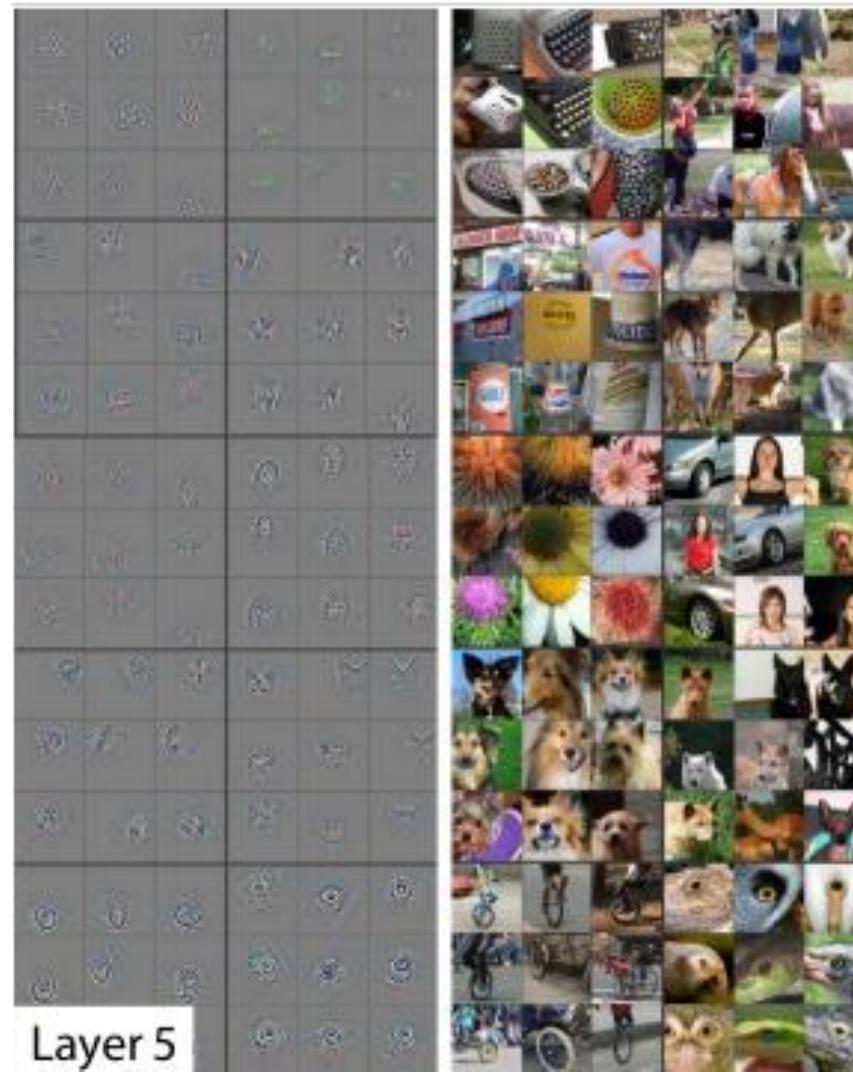
Visualizing activations



Zeiler, Matthew (2014)

Visualizing activations in CN cont..

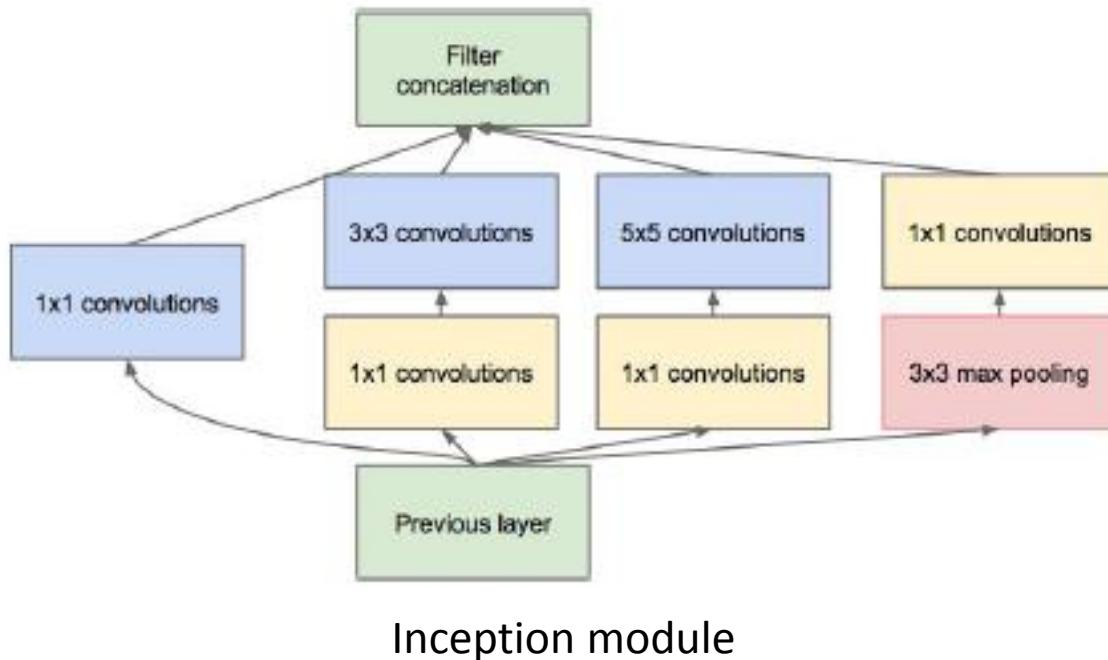
Visualizing activations



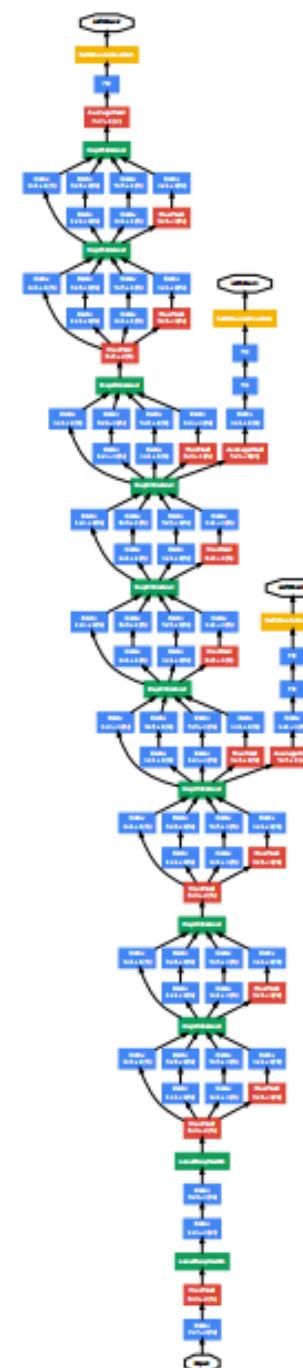
Zeiler, Matthew (2014)

Deep CNN recent examples:

GoogLeNet (ILSVRC 2014 winner)

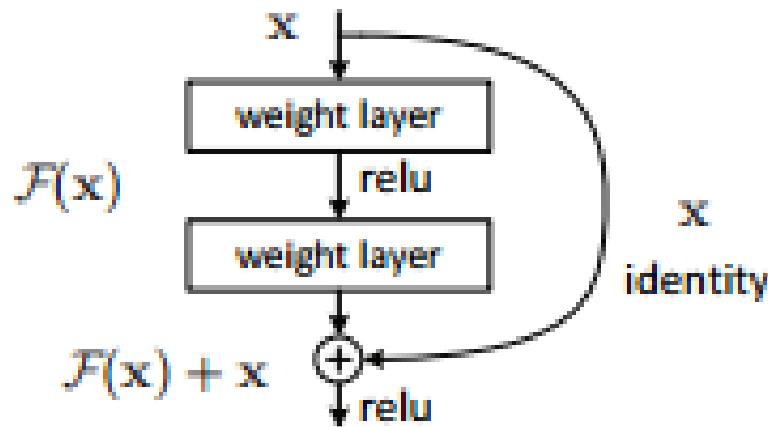


Szegedy, et al (2015)



Deep CNN recent examples:

ResNet (ILSVRC 2015 winner)



Residual learning: a building block

152 layers; 3.57 top 5% error

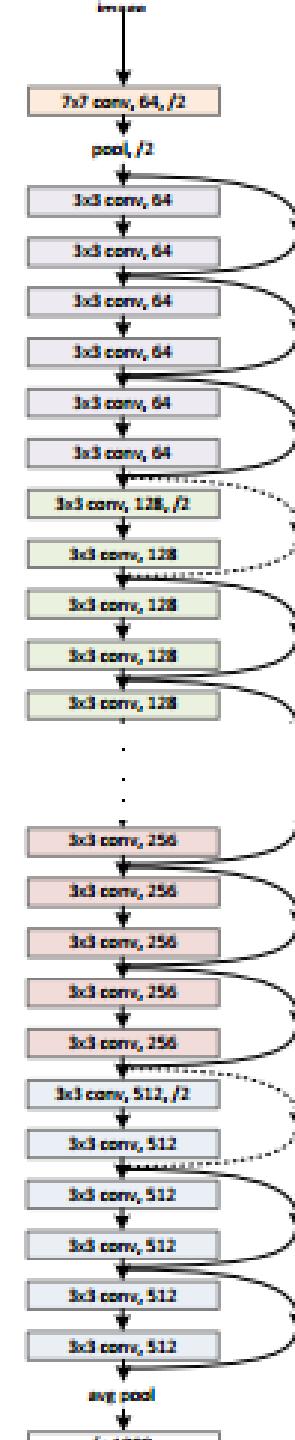
ImageNet 2012 dataset;

1000 classes, 1.28 million
training Images,

50K validation images, 100 K
test images;

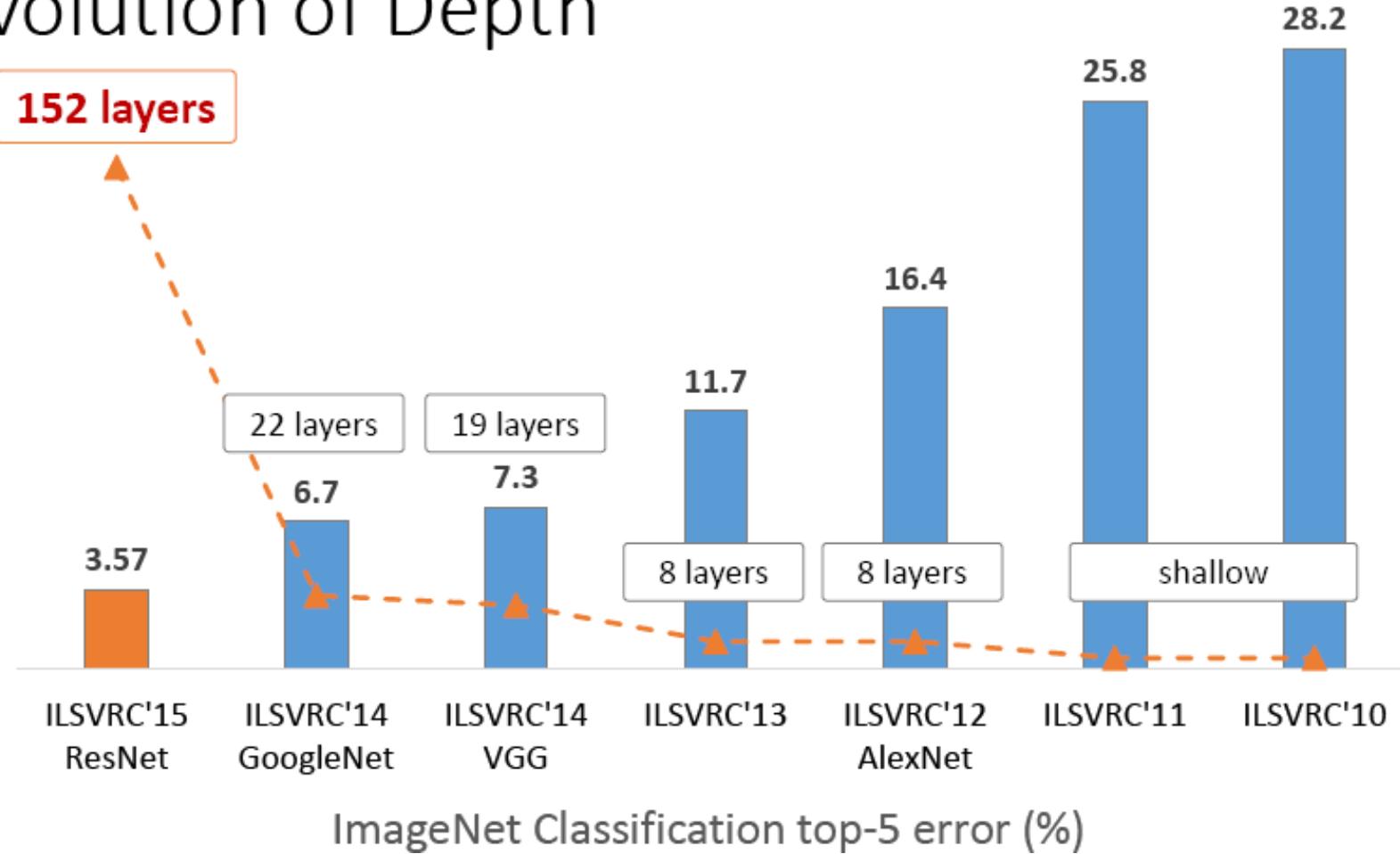
millions of parameters;

Kaiming He, et al. (2015)

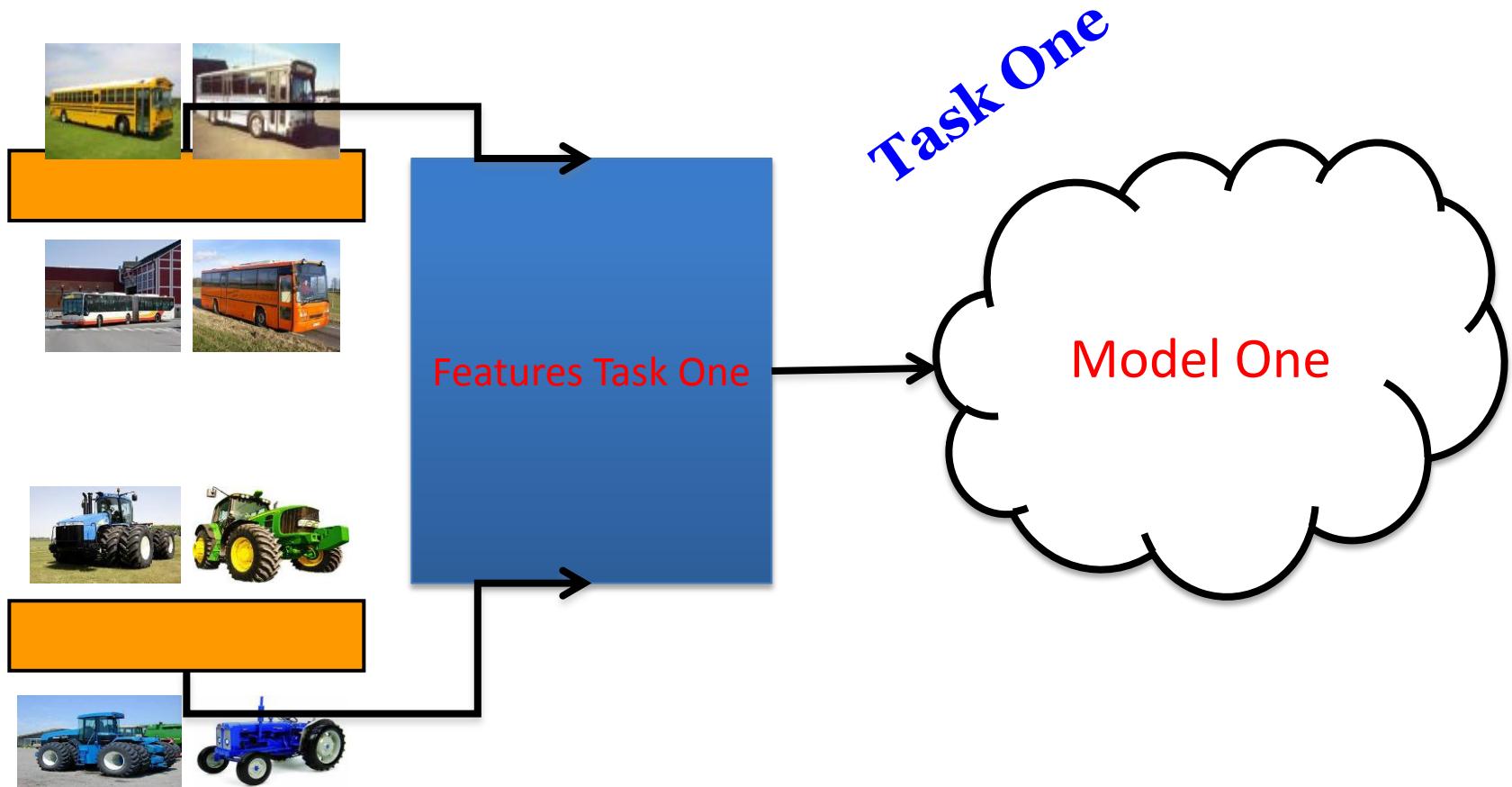


CNN; Competition in Depth

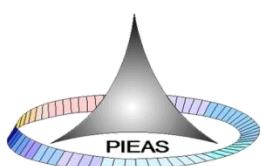
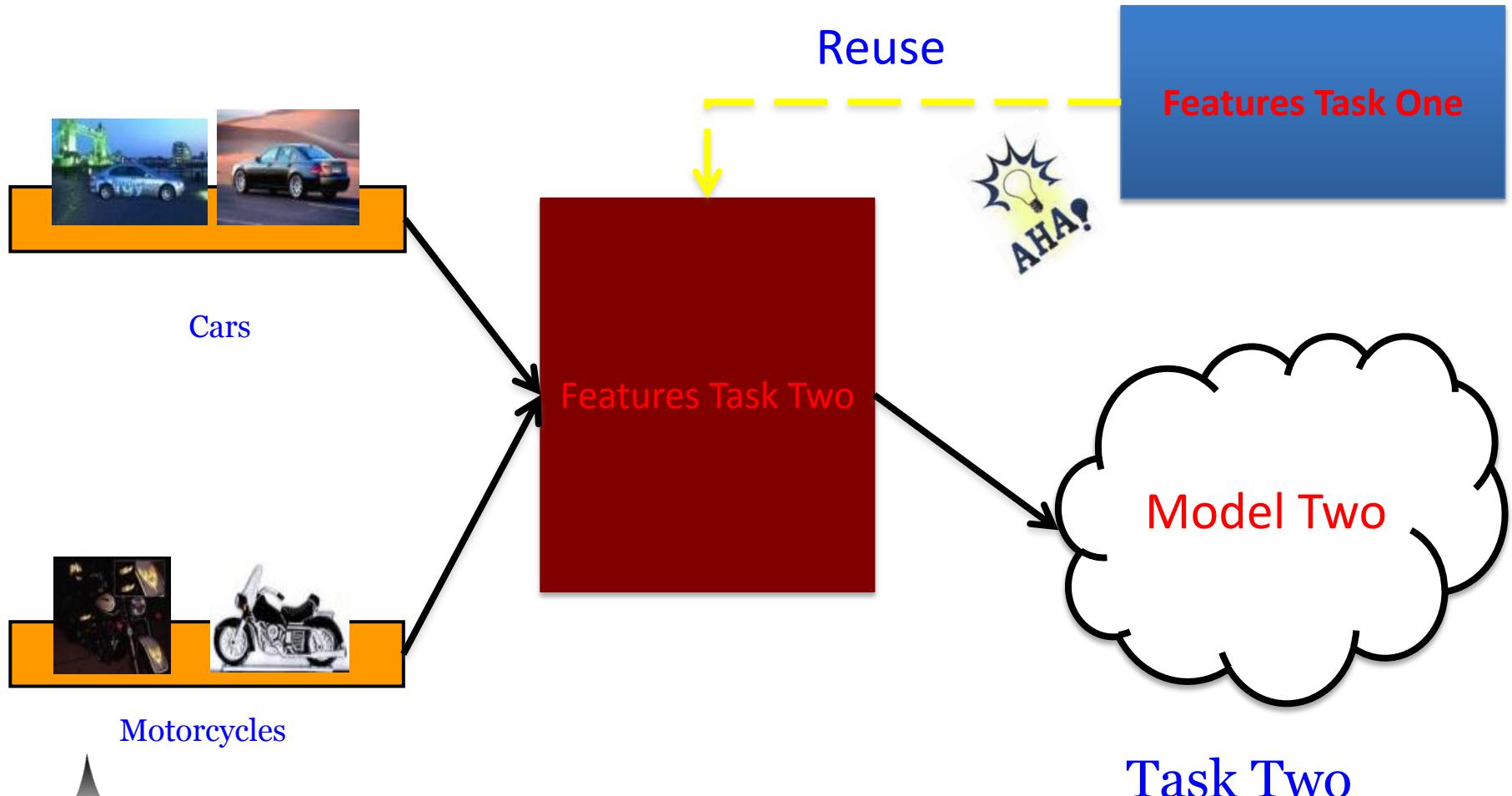
Revolution of Depth



Transfer Learning; Image Classification example



Transfer Learning; Image Classification example



Convolutional Neural Networks

To ponder on:

- Computational cost vs. performance/results
- What should be the filter sizes at different layers?
- How much pooling?
- How many neurons to keep in different layers?
- How many layers to employ?
- How to increase generalization of a CNN?
 - Use a good cross-validation startegy
 - Use pruning (Dropout, Swapout, etc.)
 - Use unsupervised pre-training

References

- <http://ufldl.stanford.edu/tutorial/supervised/FeatureExtractionUsingConvolution/>
- Tutorial on Deep Learning and Applications Honglak Lee (University of Michigan)
- <http://docs.gimp.org/en/plug-in-convmatrix.html>
- <https://developer.apple.com/library/ios/documentation/Performance/Conceptual/vImage/ConvolutionOperations/ConvolutionOperations.html>
- <http://www.slideshare.net/AhmedMahany/convolution-neural-networks>
- <http://www.robots.ox.ac.uk/~vgg/research/text/index.html>

References cont...

- <http://learning.eng.cam.ac.uk/pub/Public/Turner/Teaching/ml-lecture-3-slides.pdf>
- LeCun, Yann, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86.11 (1998): 2278-2324.
- Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *Computer vision–ECCV 2014*. Springer International Publishing, 2014. 818-833.
- Bengio, Yoshua. "Learning deep architectures for AI." *Foundations and trends® in Machine Learning* 2.1 (2009): 1-127.
- Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.
- He, Kaiming, et al. "Deep Residual Learning for Image Recognition." *arXiv preprint arXiv:1512.03385* (2015).