Convolutional Networks

Honglak Lee

CSE division, EECS department

University of Michigan, Ann Arbor

8/6/2015

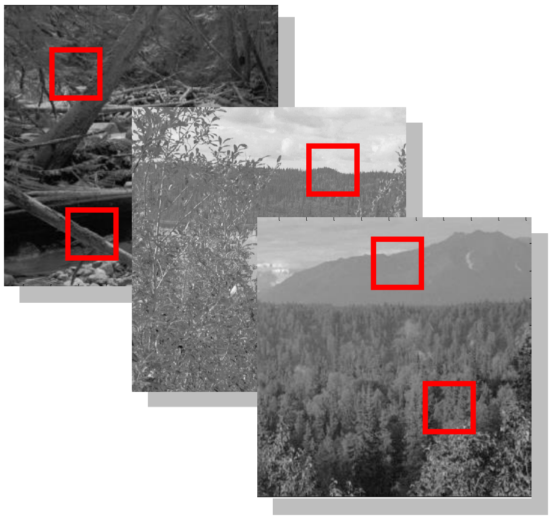
Deep Learning Summer School @ Montreal

Unsupervised Convolutional Networks

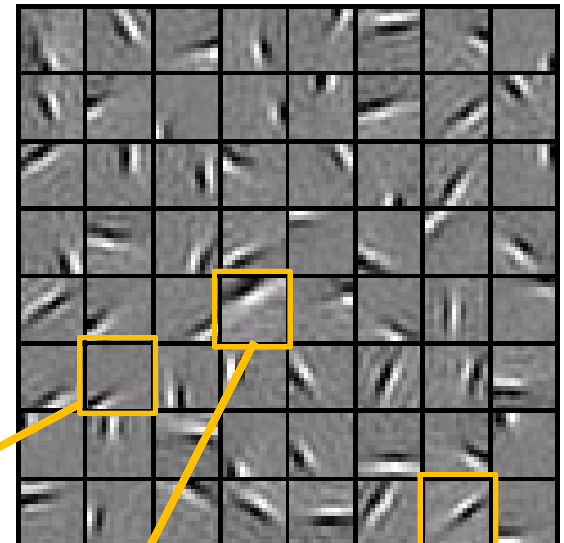
Learning Feature Hierarchy

[Lee et al., NIPS 2007; Ranzato et al., 2007]

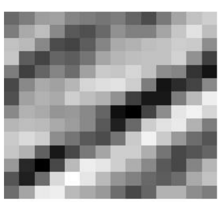
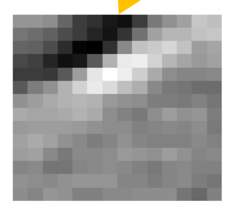
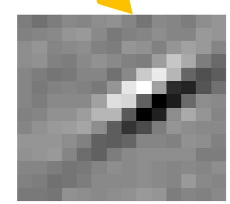
Natural Images



Learned bases: "Edges"



Test example

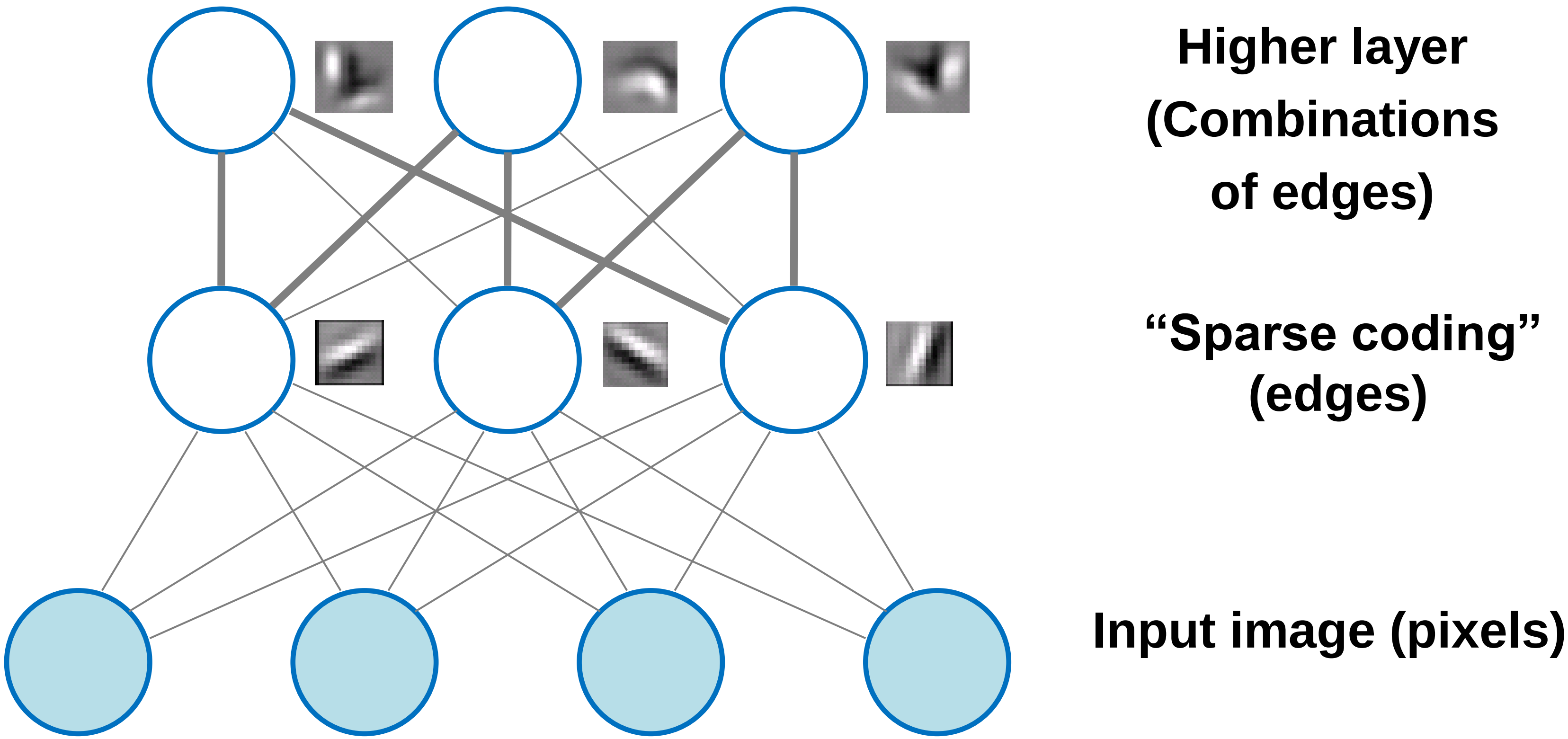
 ~  +  + 

$$x \sim 1 * b_{36} + 1 * b_{42} + 1 * b_{65}$$

[0, 0, …, 0, **1**, 0, …, 0, **1**, 0, …, 0, **1**, …]
= coefficients (feature representation)

Compact & easily interpretable

Learning Feature Hierarchy

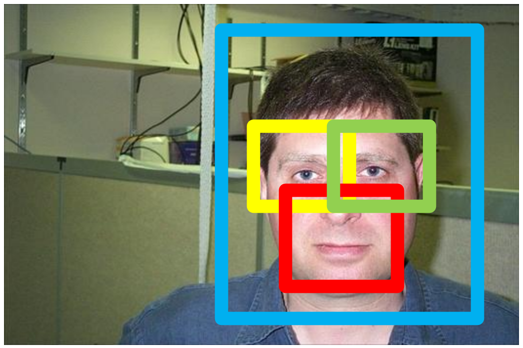


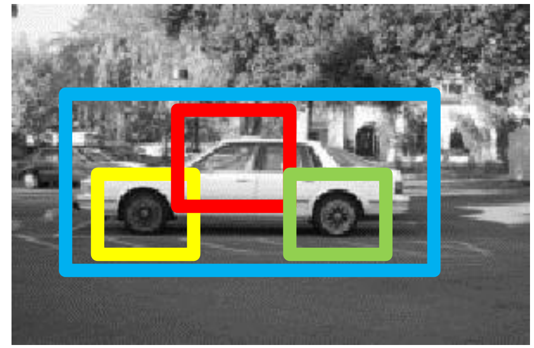
Lee et al., NIPS 2007: DBN (Hinton et al., 2006) with additional sparseness constraint.

[Related work: Bengio et al., 2006; Ranzato et al., 2007, and others.]

Learning object representations

- Learning objects and parts in images





- Large image patches contain interesting higher-level structures.
  - E.g., object parts and full objects

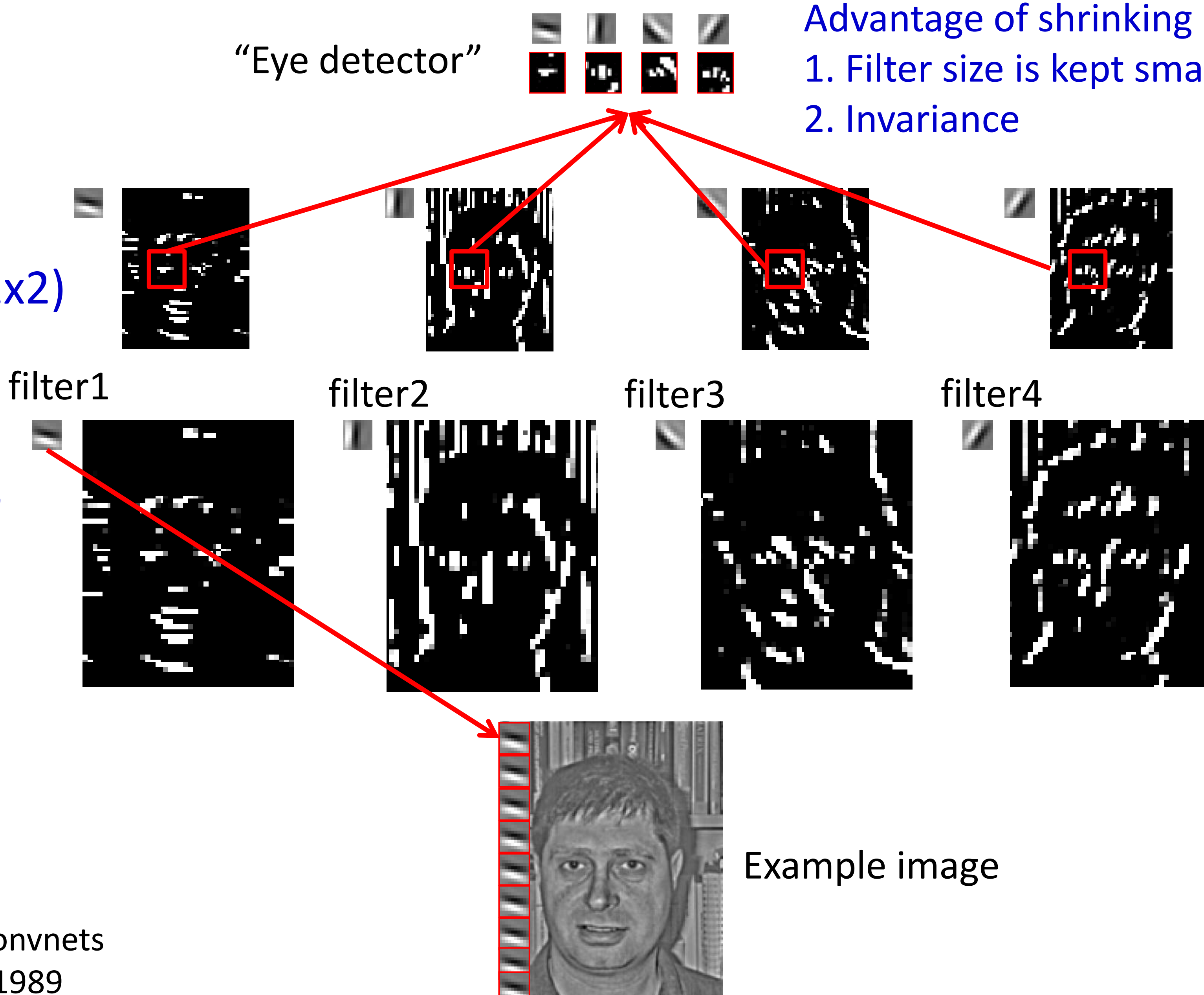
- Challenge: high-dimensionality and spatial correlations

Illustration: Learning an "eye" detector

Advantage of shrinking
1. Filter size is kept small
2. Invariance

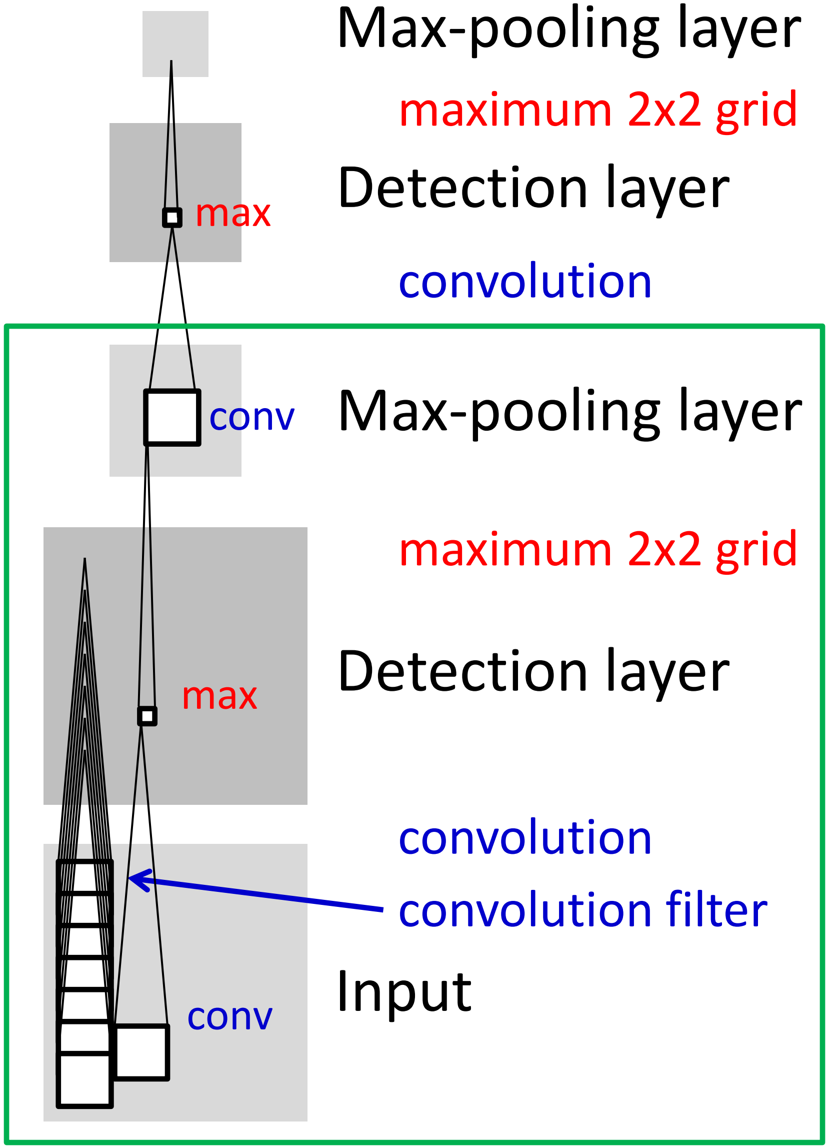
"Shrink"
(max over 2x2)

"Filtering"
output



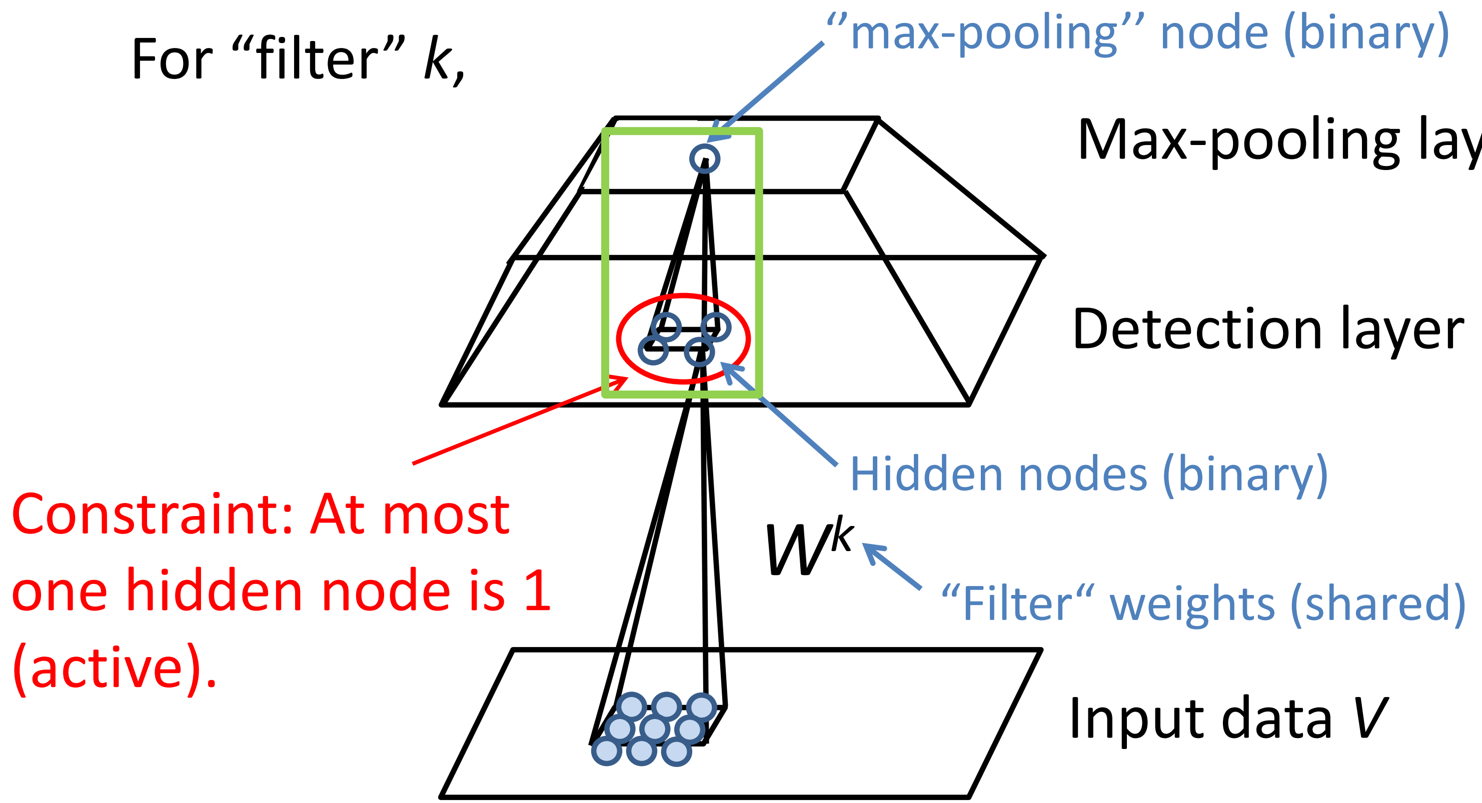
Related work: Convnets
by LeCun et al., 1989

Convolutional architectures



- Weight sharing by “filtering” (convolution) [Lecun et al., 1989]
- “Max-pooling”
  - Invariance
  - Computational efficiency
- Convolutional Restricted Boltzmann machine.
  - Unsupervised
  - Probabilistic max-pooling
  - Can be stacked to form convolutional DBN

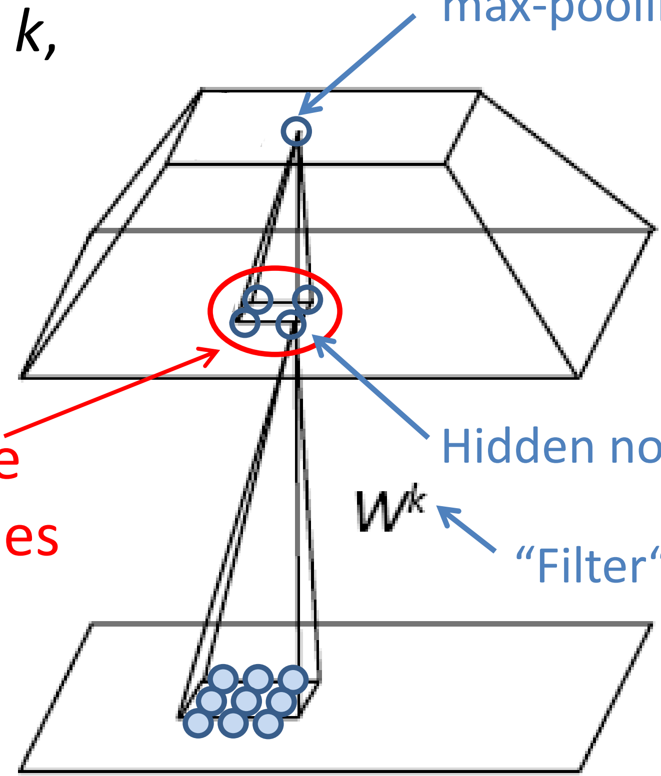
Convolutional RBM (CRBM) [Lee et al., ICML 2009]



- Key properties:
  - RBM (probabilistic model)
  - Convolutional structure (weight sharing)
  - Constraint for max-pooling ("mutual exclusion")

Convolutional RBM (CRBM) [Lee et al., ICML 2009]

For “filter” *k*,



$$P(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\exp(-E(\mathbf{v},\mathbf{h}))$$

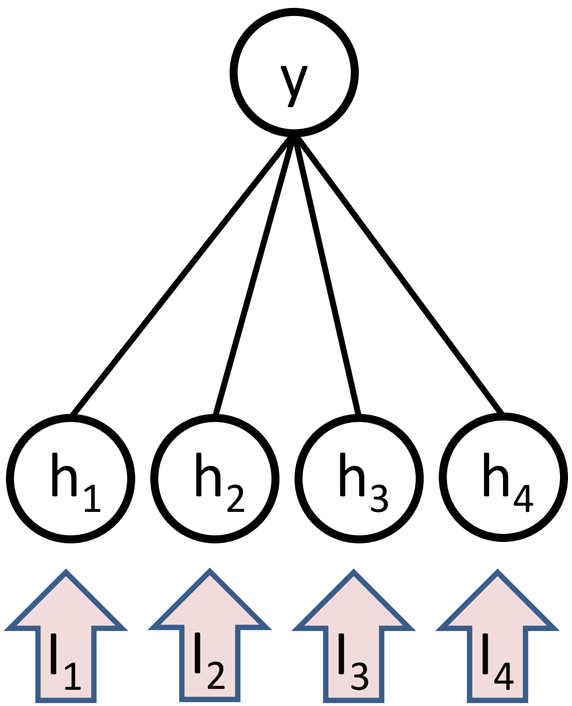
$$E(\mathbf{v},\mathbf{h}) = -\sum_{i,j}\left(\sum_k h^k_{i,j}(\tilde{W}^k * v)_{i,j} + b^k h^k_{i,j} + c\, v_{i,j}\right)$$

$$\text{subject to} \quad \sum_{(i,j)\in\text{“}cell(y)\text{”}} h^k_{i,j} \le 1, \forall k, y.$$

Inference: probabilistic max-pooling

$$P(h_j = 1) \;=\; \frac{\exp(I_j)}{1 + \sum_{\ell} \exp(I_{\ell})}$$

Pooling node

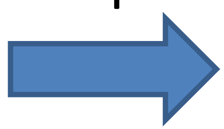


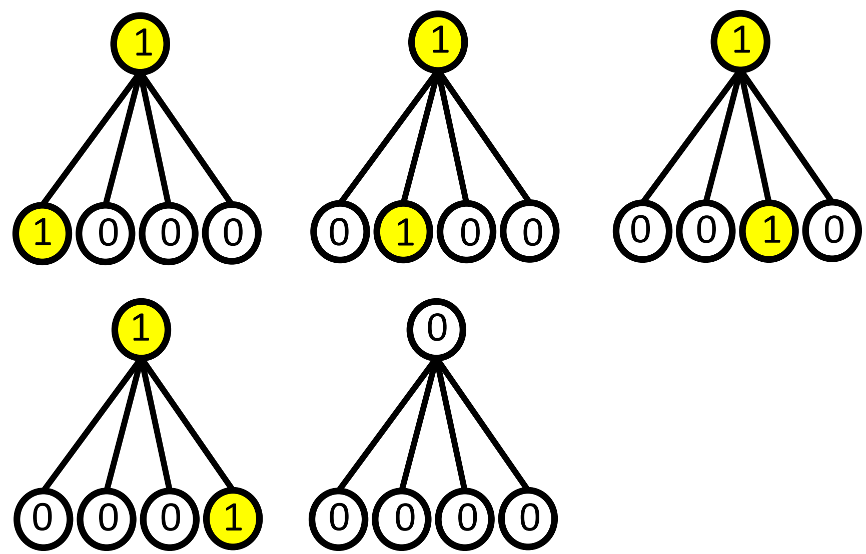
Detection nodes

Output of convolution
W*V from below

Softmax function

Sample





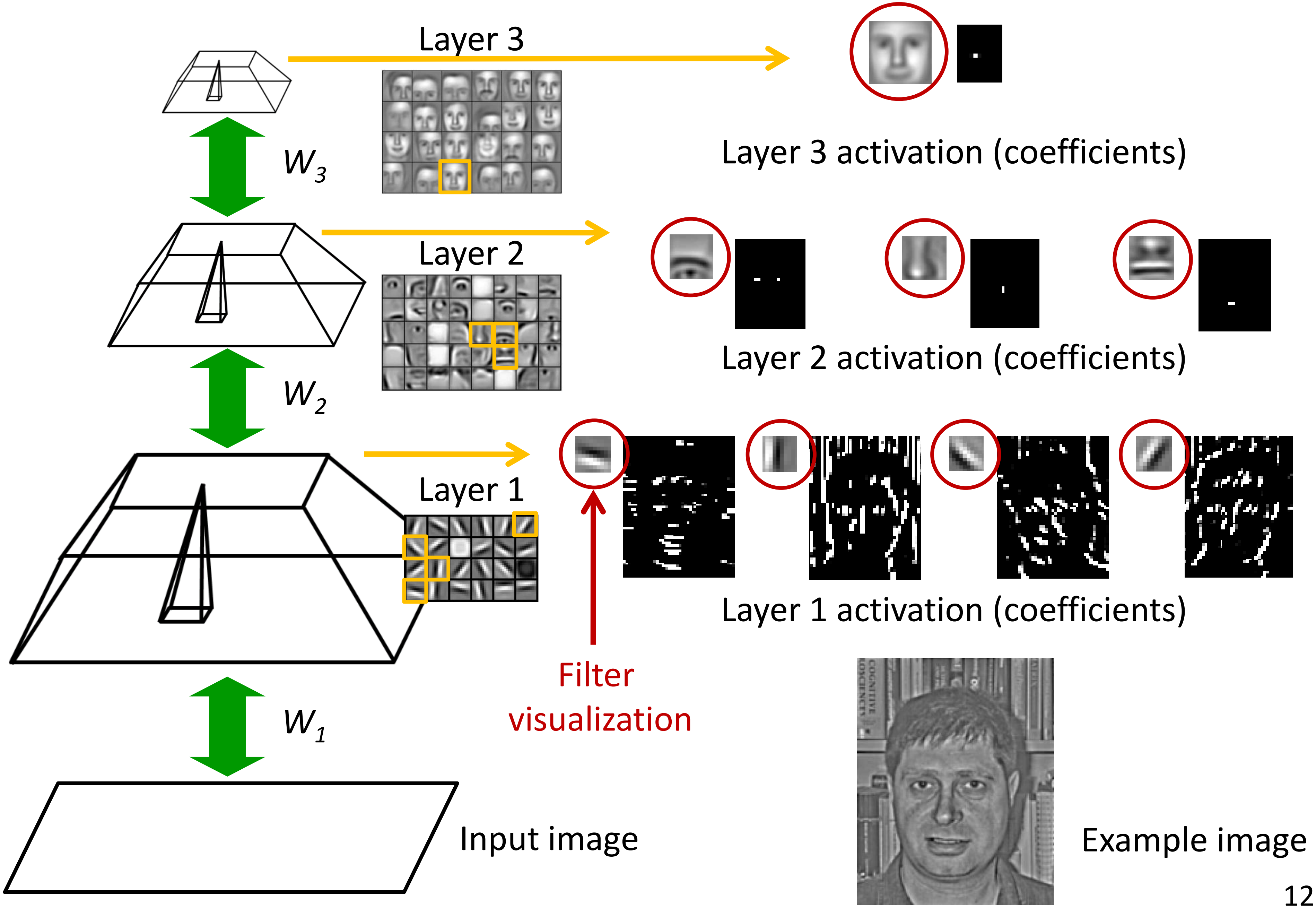
Collapse 2^n configurations into n+1 configurations.

Convolutional Deep Belief Networks (CDBN)

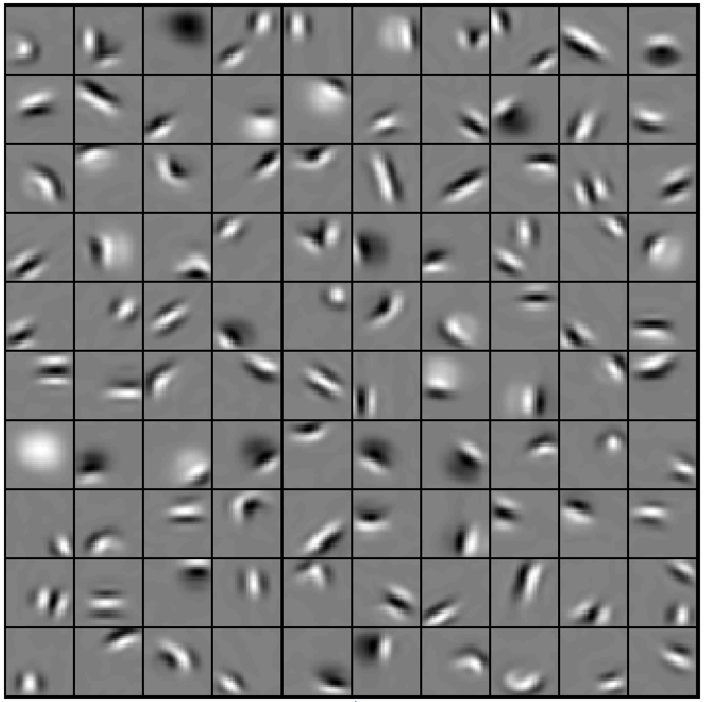
- Bottom-up (greedy), layer-wise training
  - Train one layer (convolutional RBM) at a time.

- Feedforward Inference (approximate)

Convolutional Deep Belief Networks (CDBN)

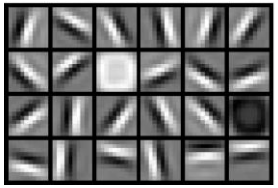


Unsupervised learning from natural images



Second layer bases

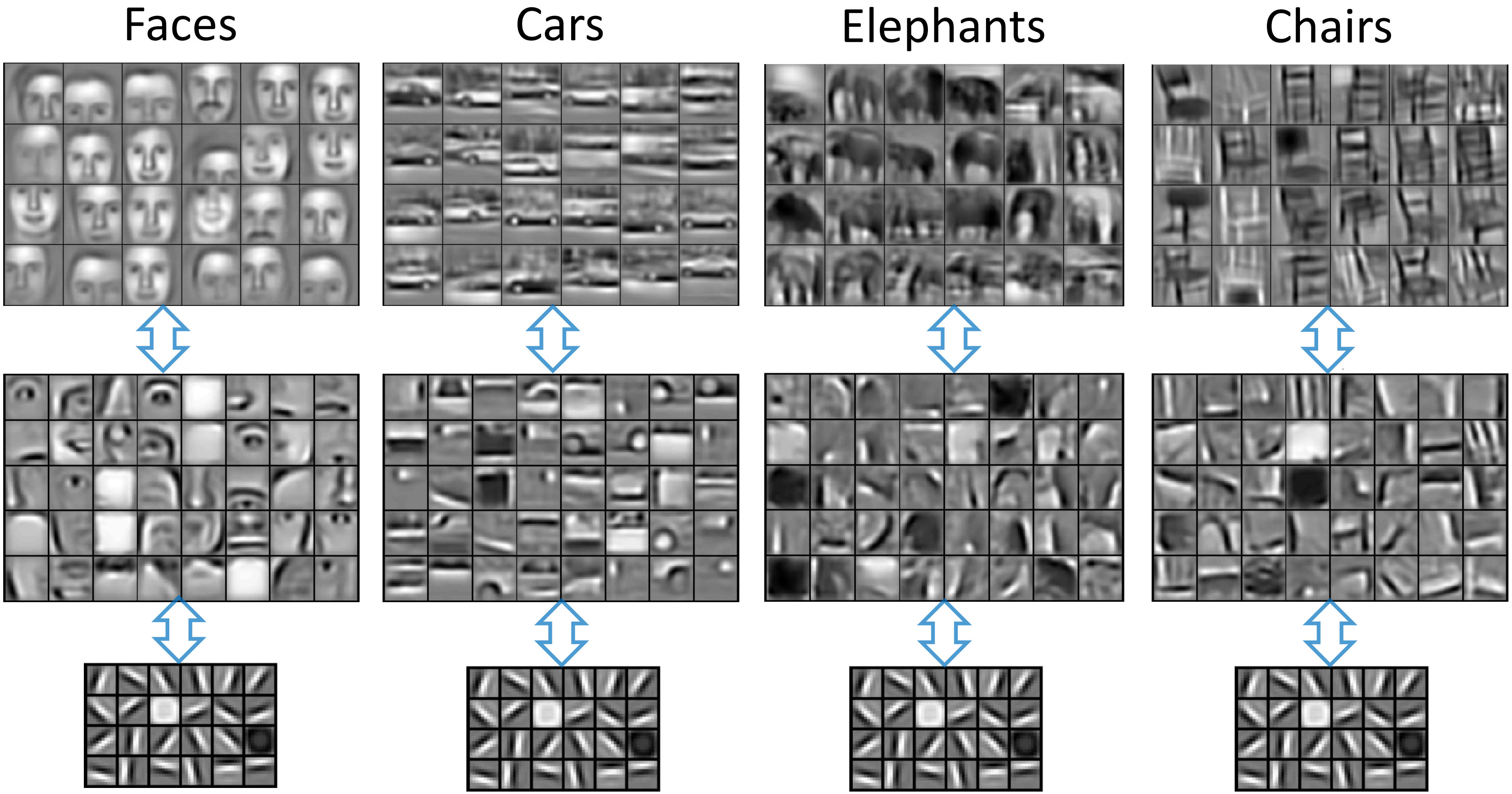
contours, corners, arcs, surface boundaries



First layer bases

localized, oriented edges

Unsupervised learning of object-parts



Applications:

- Classification (ICML 2009, NIPS 2009, ICCV 2011, ICML 2013)
- Verification (CVPR 2012)
- Image alignment (NIPS 2012)

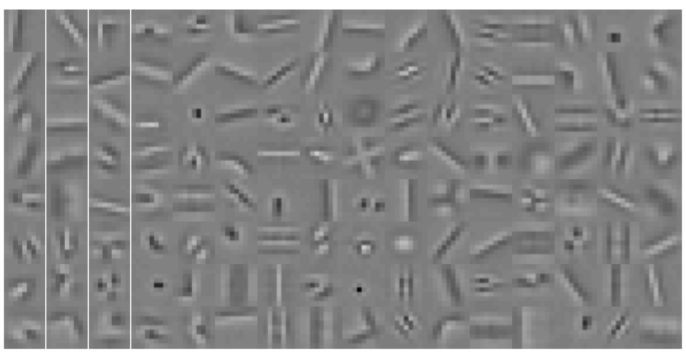
Convolutional Sparse Coding

- Learning objective

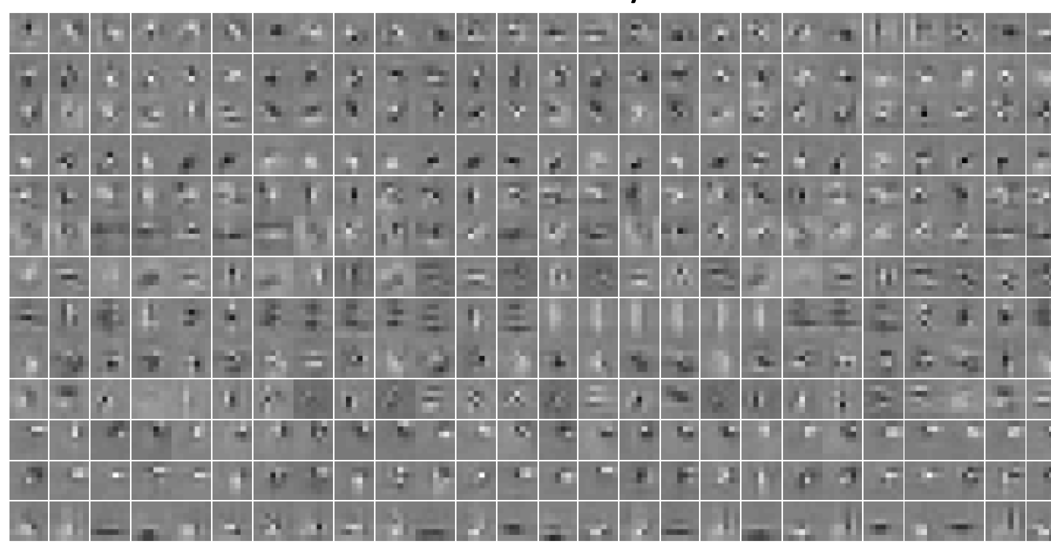
$$\mathcal{L}(x, z, \mathcal{D}, W) = \frac{1}{2}||x - \sum_{k=1}^{K} \mathcal{D}_k * z_k||_2^2 + \sum_{k=1}^{K} ||z_k - f(W^k * x)||_2^2 + |z|_1$$

- Learned filters

First layer



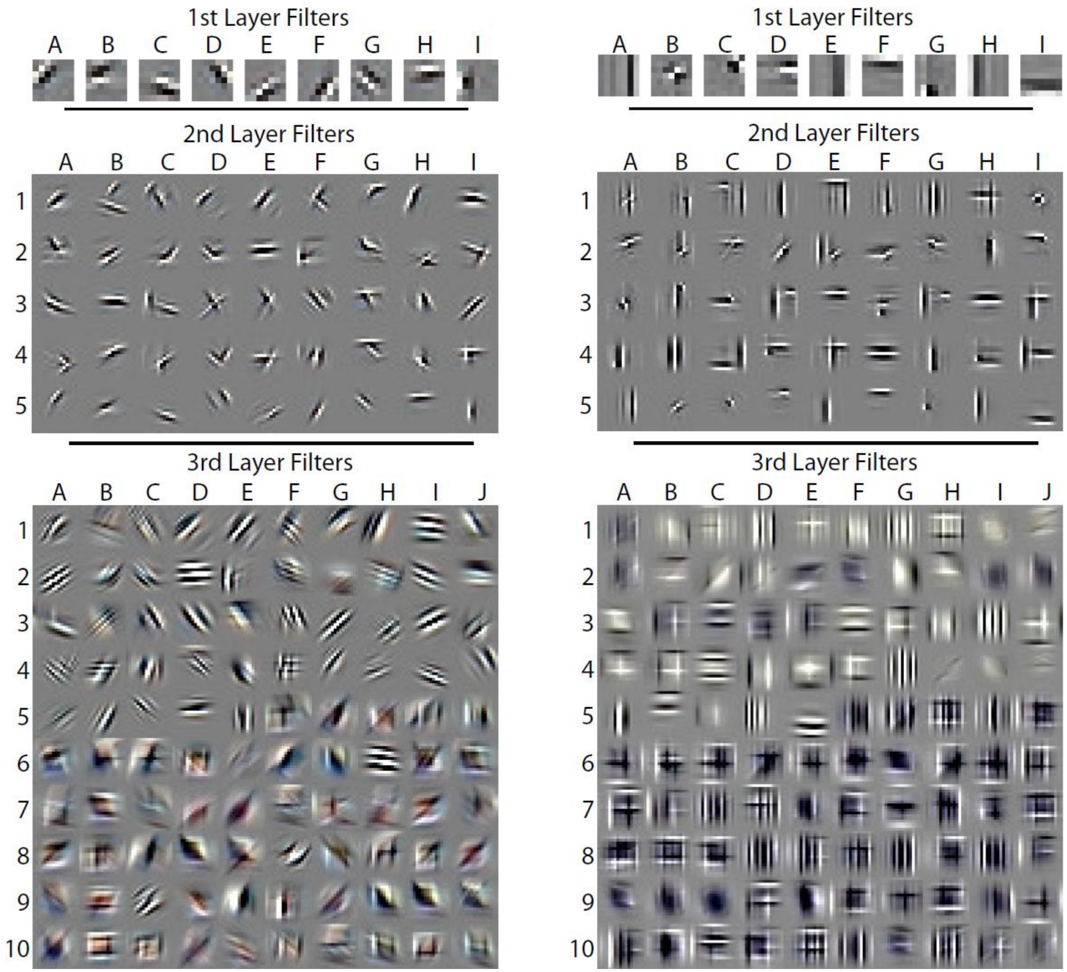
Second layer



Kavukcuoglu et al. Learning convolutional feature hierarchies for visual recognition. NIPS 2010.

Deconvolutional Networks

- Learning objective: $$\frac{\lambda}{2}\sum_{c=1}^{K_0}\|\sum_{k=1}^{K_1} z_k^i \oplus f_{k,c} - y_c^i\|_2^2 + \sum_{k=1}^{K_1}|z_k^i|^p$$
- Learned filters:

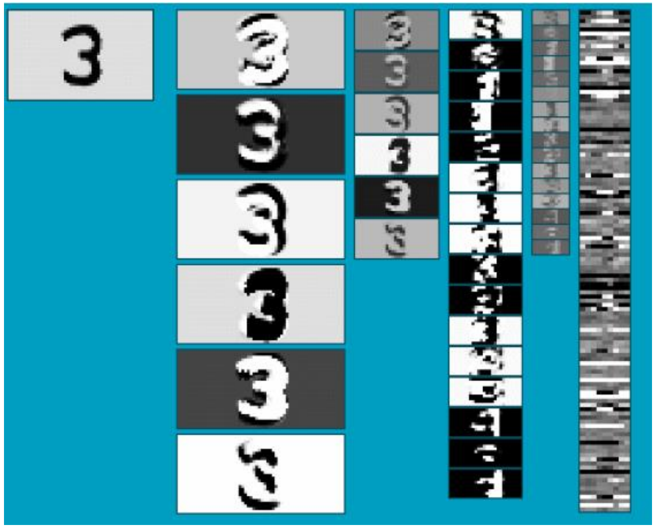


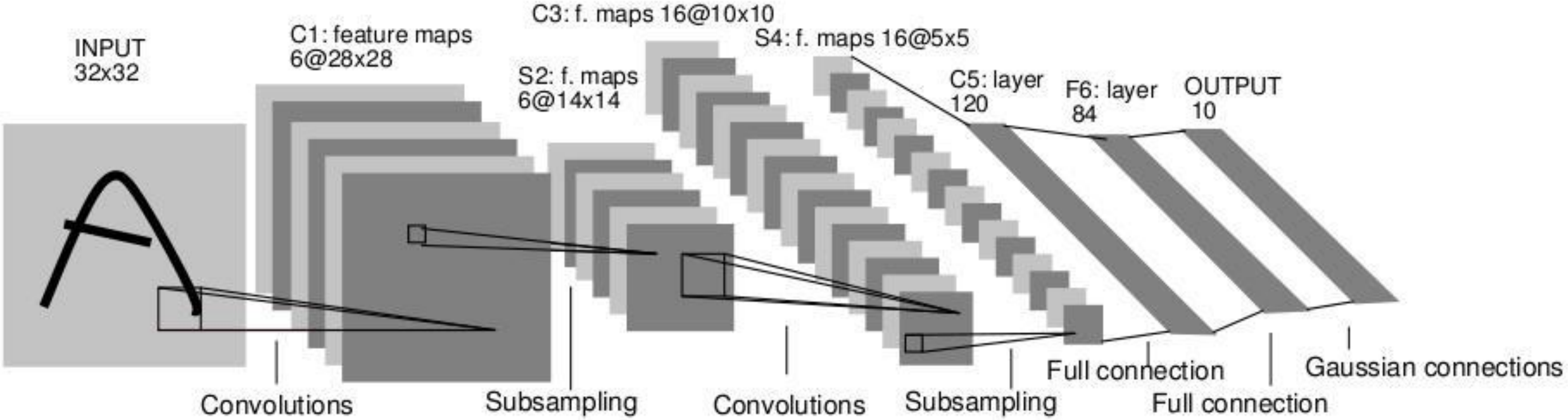
Zeiler et al. "Deconvolutional networks." *CVPR 2010*

Supervised Convolutional Networks

Example: Convolutional Neural Networks

- LeCun et al. 1989
- Neural network with specialized connectivity structure

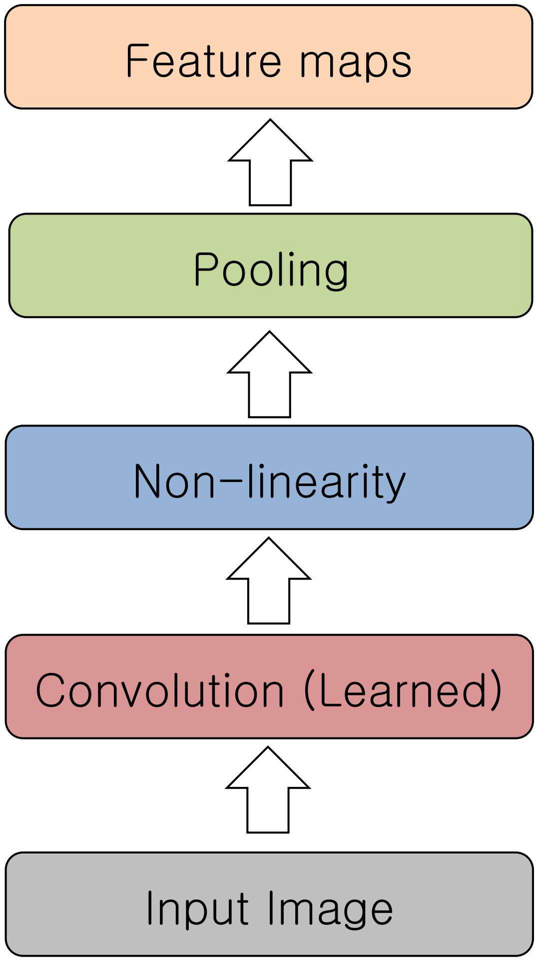




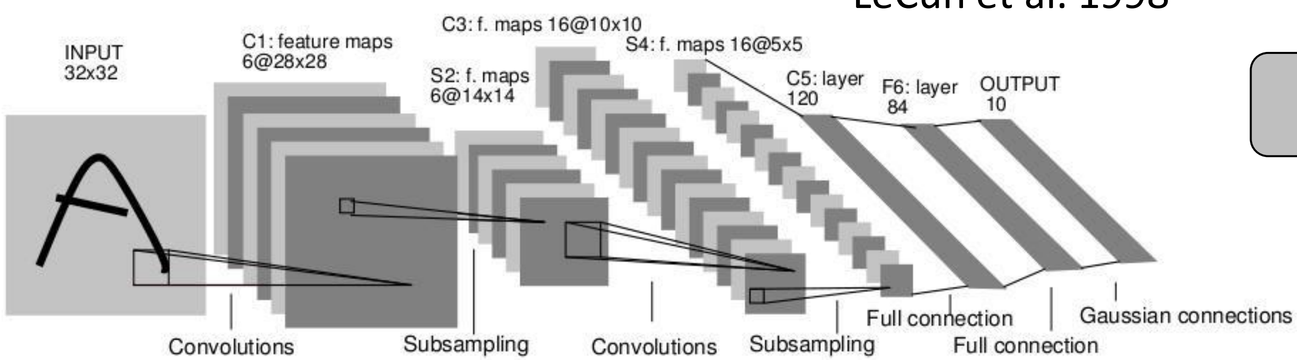
Slide: R. Fergus

Convolutional Neural Networks

- Feed-forward:
  - Convolve input
  - Non-linearity (rectified linear)
  - Pooling (local max)
- Supervised
- Train convolutional filters by back-propagating classification error



LeCun et al. 1998



Slide: R. Fergus

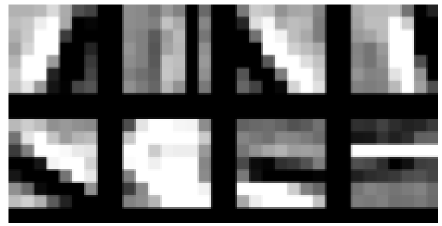
Components of Each Layer

Pixels /

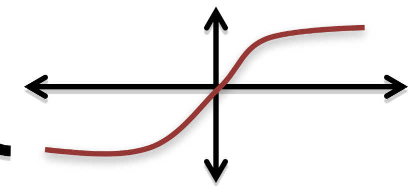
Features

Filter with Dictionary

(convolutional or tiled)

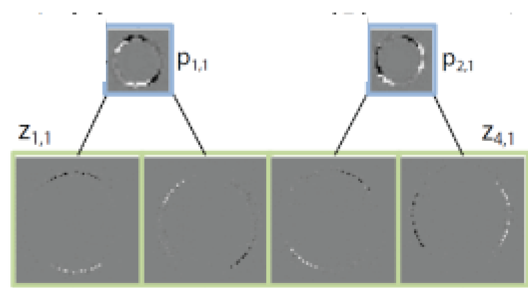


+ Non-linearity



Spatial/Feature

(Sum or Max)



[Optional]

Normalization between feature responses

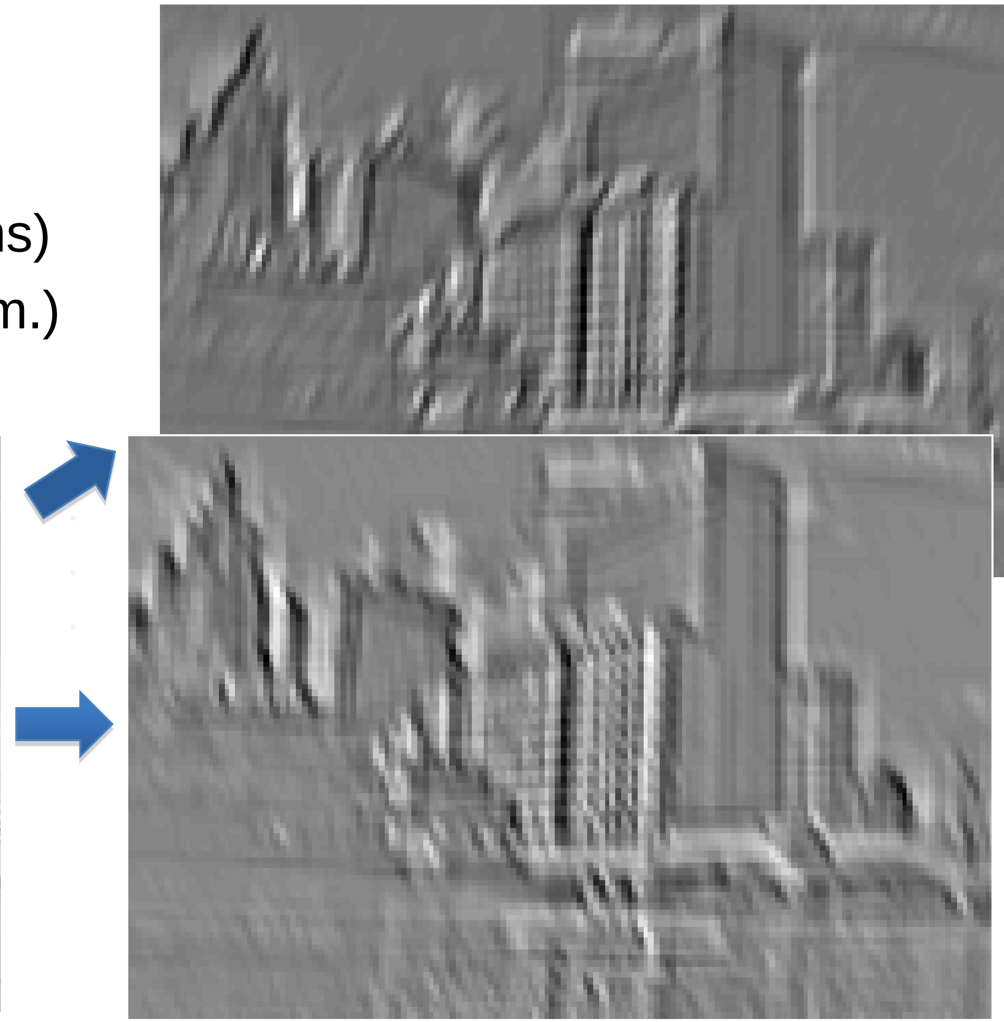
Output Features

Slide: R. Fergus

Filtering

- Convolutional
  - Dependencies are local
  - Translation equivariance
  - Tied filter weights (few params)
  - Stride 1,2,… (faster, less mem.)

Input



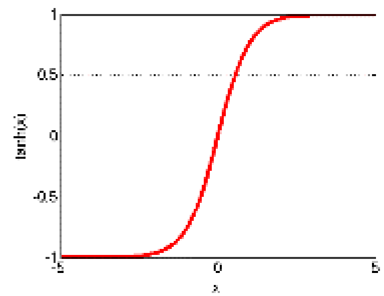
Feature Map

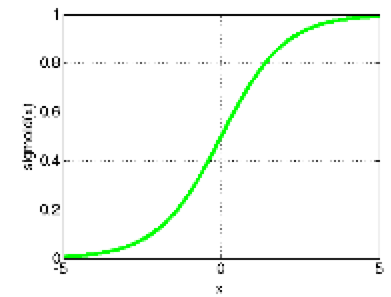
Slide: R. Fergus

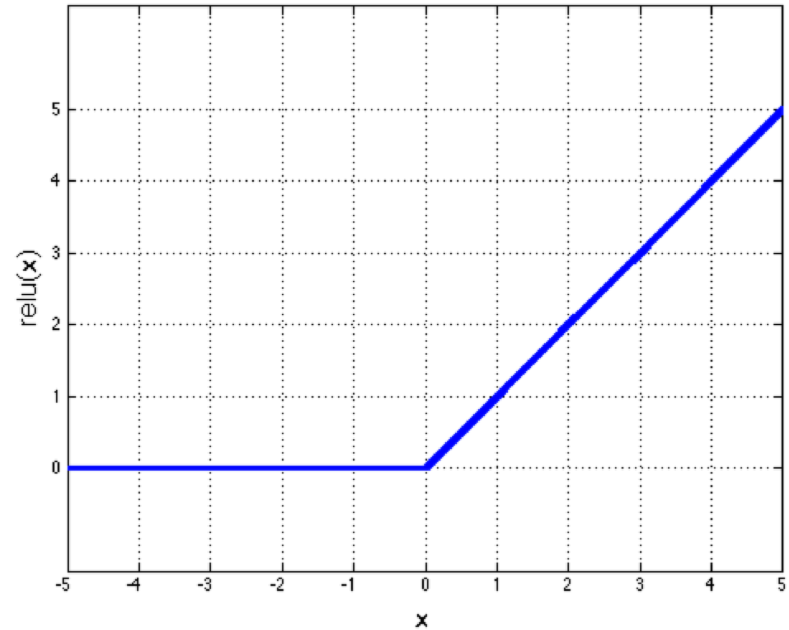
Non-Linearity

- Non-linearity
  - Per-element (independent)
  - Tanh
  - Sigmoid: 1/(1+exp(-x))
  - Rectified linear
    - Simplifies backprop
    - Makes learning faster
    - Avoids saturation issues

  → Preferred option

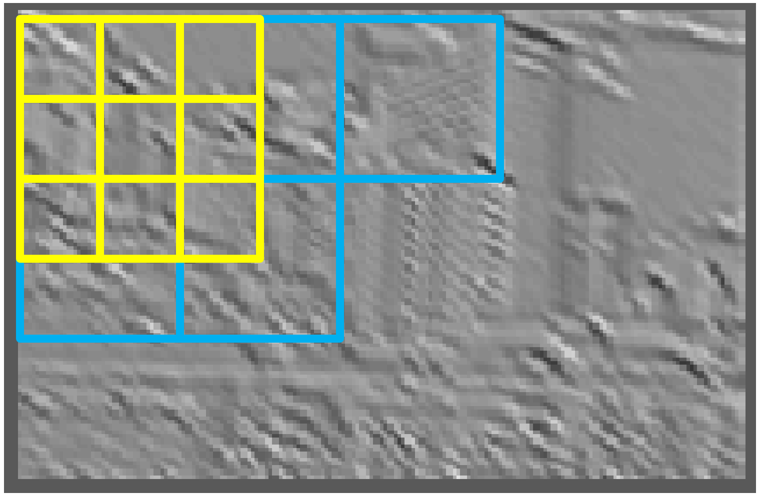


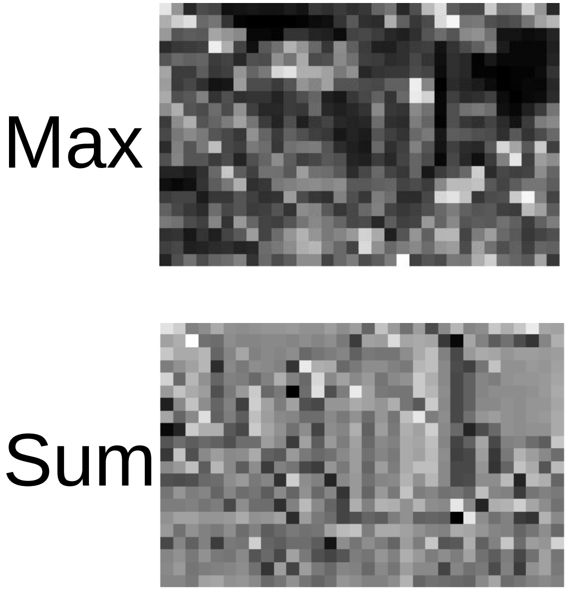




Pooling

- Spatial Pooling
  - Non-overlapping / overlapping regions
  - Sum or max
  - Boureau et al. ICML'10 for theoretical analysis



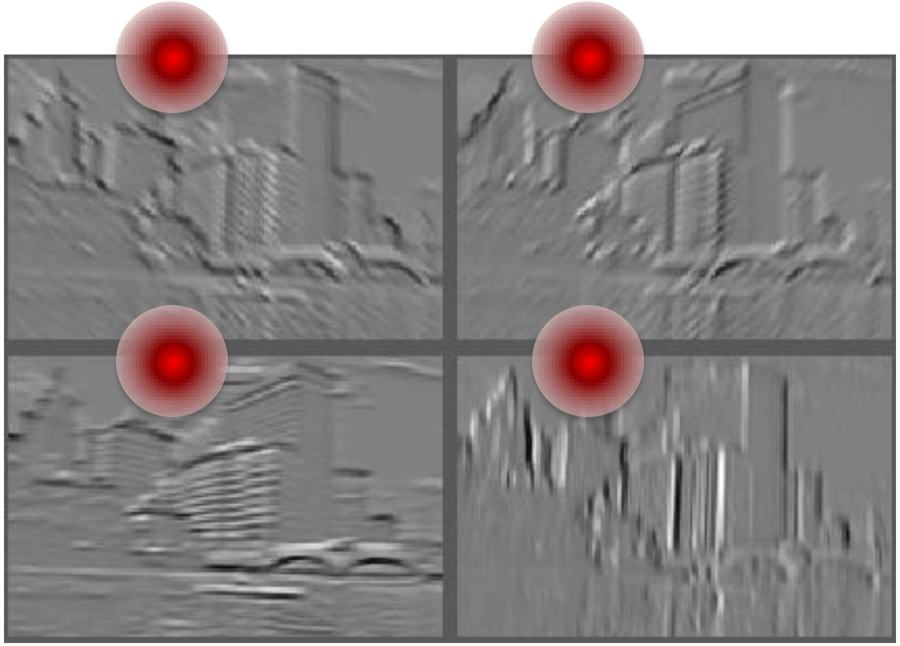


Slide: R. Fergus

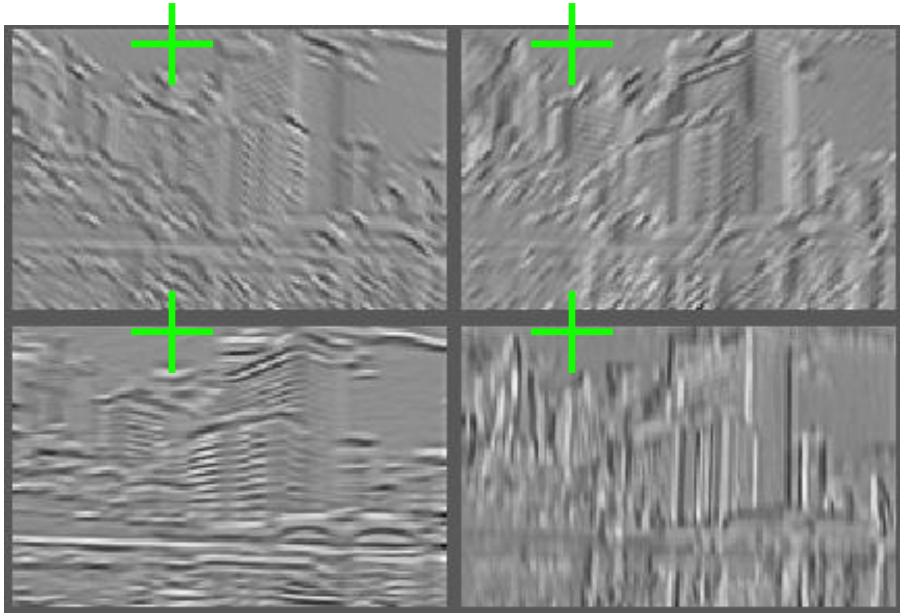
Normalization

- Contrast normalization (across feature maps)
  - Local mean = 0, local std. = 1, "Local" → 7x7 Gaussian
  - Equalizes the features maps

Feature Maps



Feature Maps
After Contrast Normalization



Slide: R. Fergus

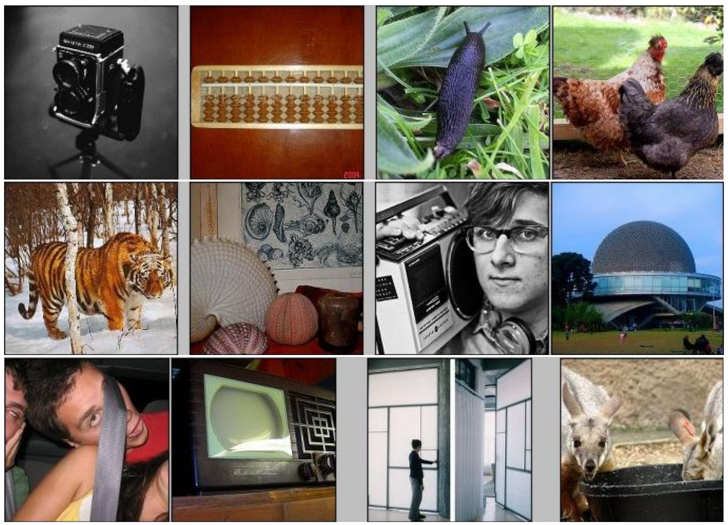
Applications

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

Slide: R. Fergus

Application: ImageNet

IMAGENET

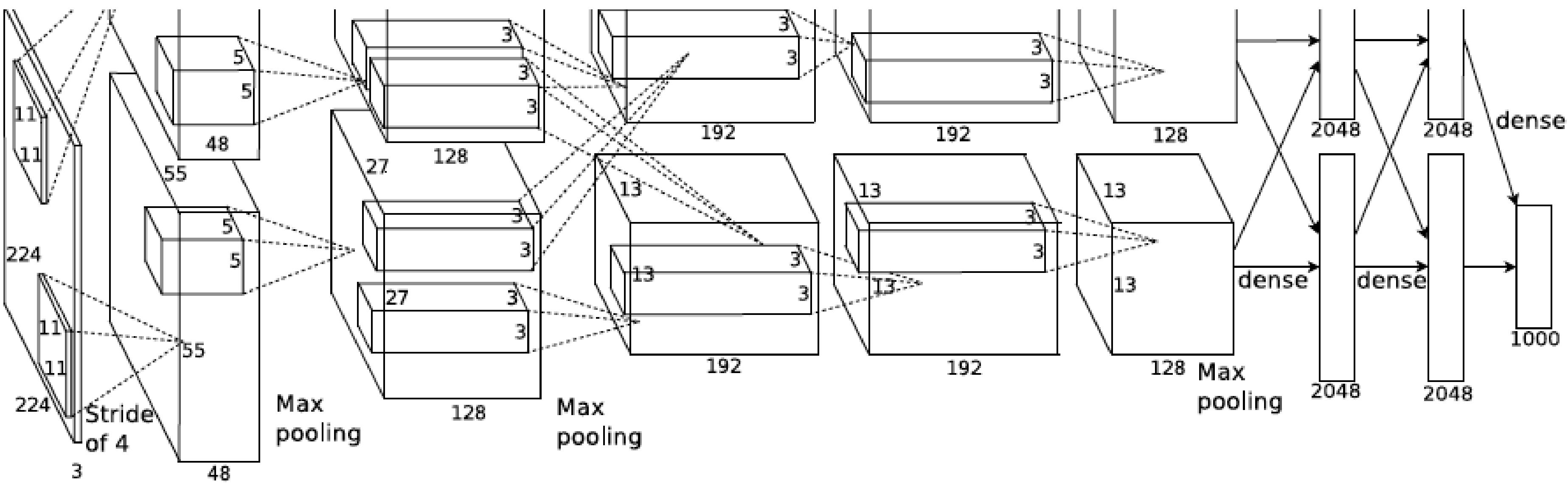


[Deng et al. CVPR 2009]

- ~14 million labeled images, 20k classes
- Images gathered from Internet
- Human labels via Amazon Turk

Krizhevsky et al. [NIPS 2012]

- Same model as LeCun'98 but:
  - Bigger model (8 layers)
  - More data (10^6 vs 10^3 images)
  - GPU implementation (50x speedup over CPU)
  - Better regularization (DropOut)

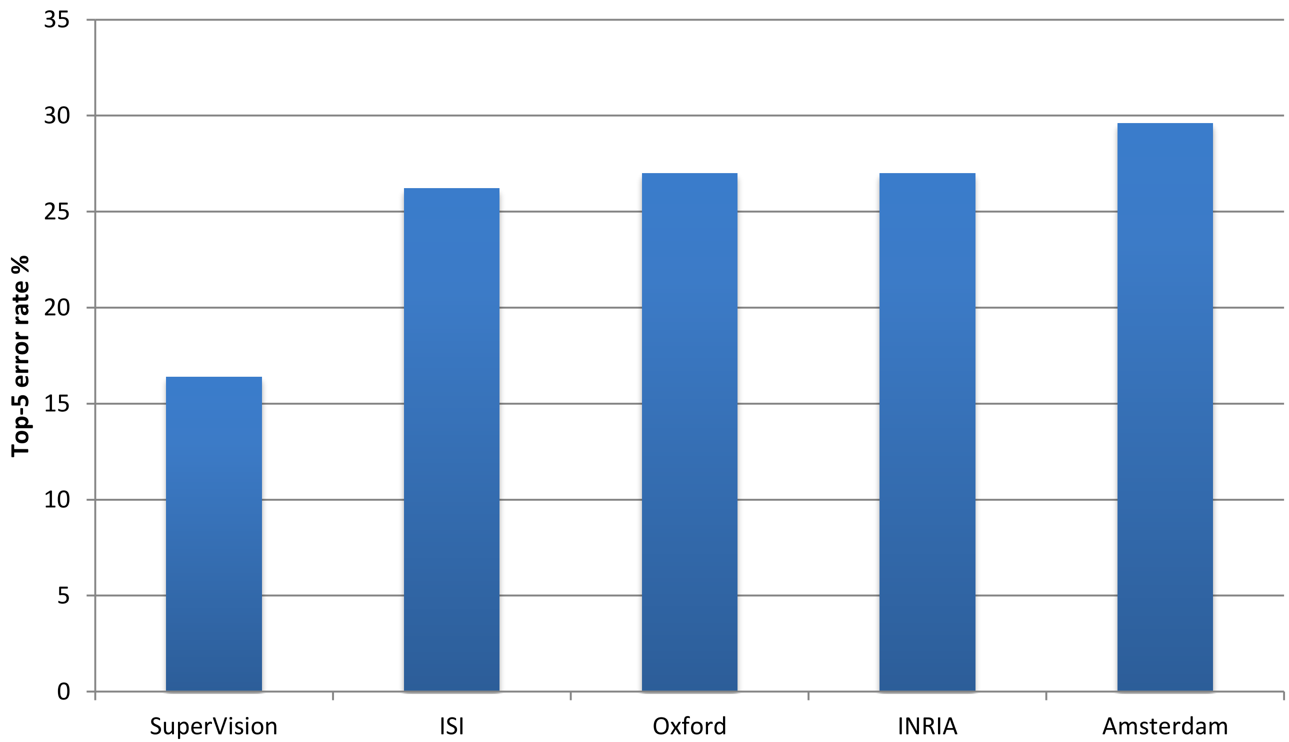


- 7 hidden layers, 650,000 neurons, 60,000,000 parameters
- Trained on 2 GPUs for a week

Slide: R. Fergus

ImageNet Classification 2012

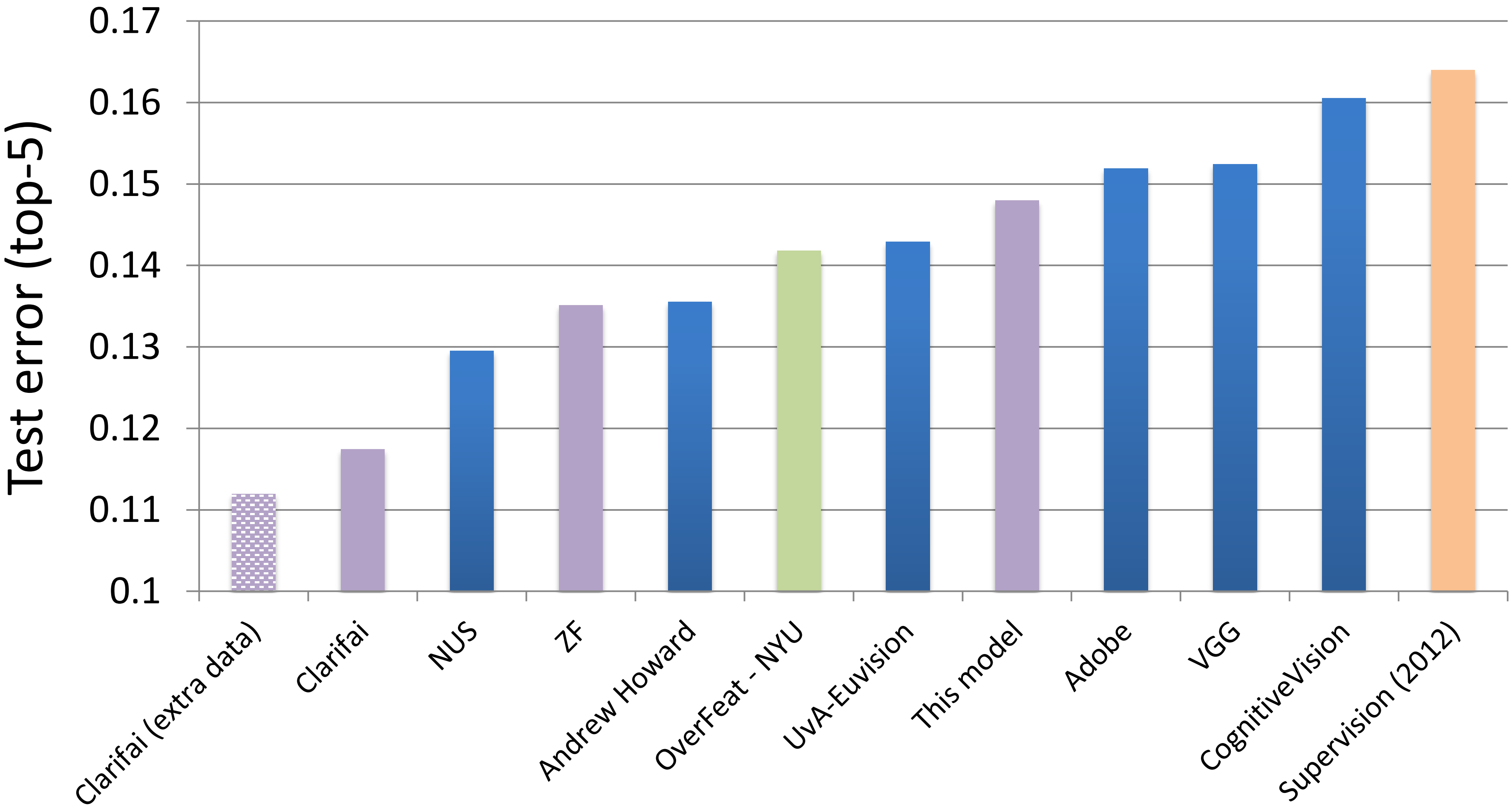
- Krizhevsky et al. -- 16.4% error (top-5)
- Next best (non-convnet) – 26.2% error



Slide: R. Fergus

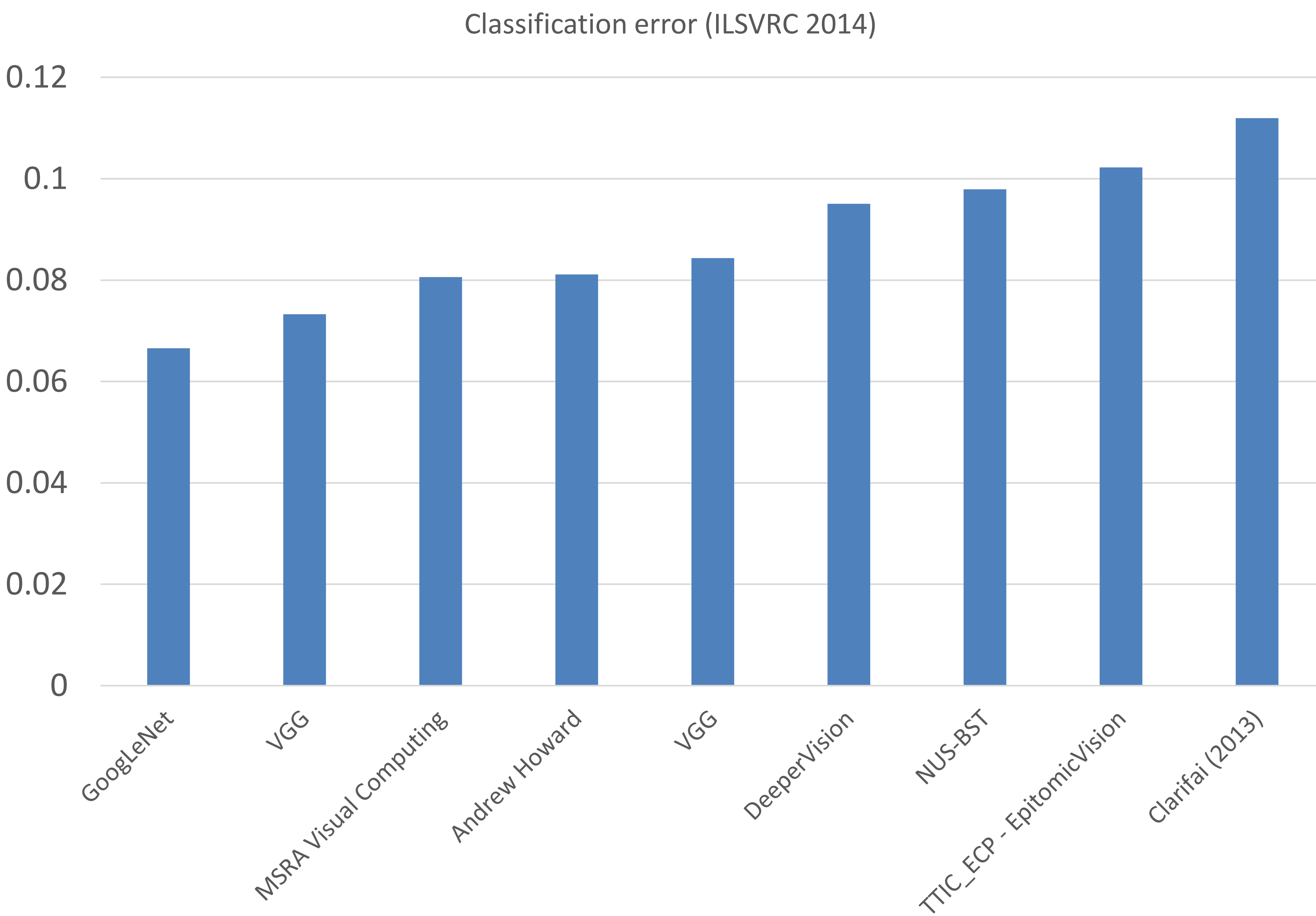
ImageNet Classification 2013 Results

- http://www.image-net.org/challenges/LSVRC/2013/results.php



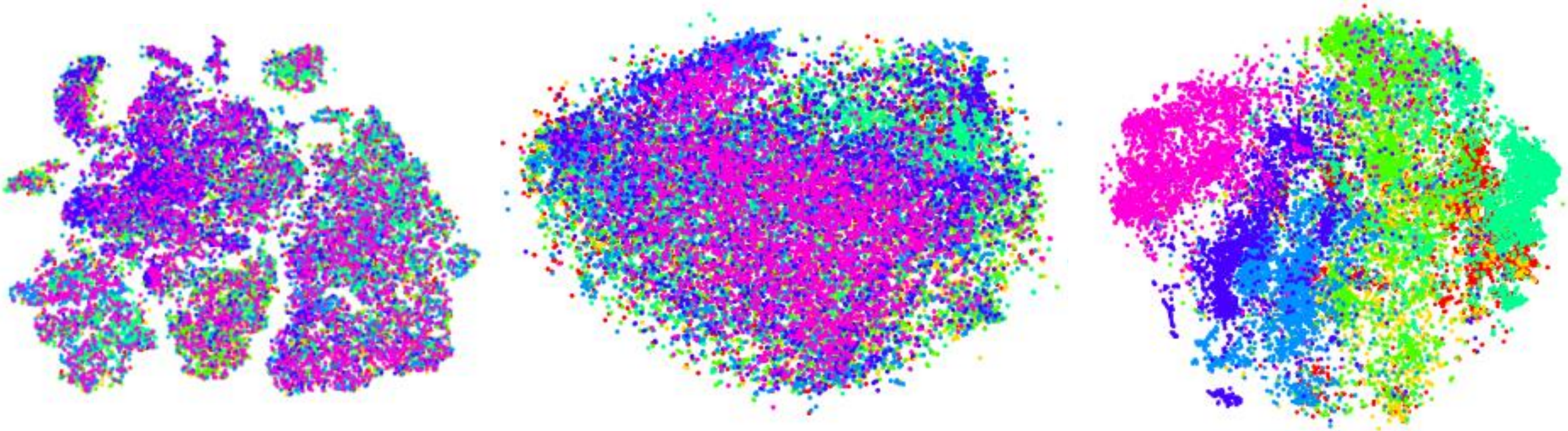
Slide: R. Fergus

ImageNet Classification 2014 Results



Feature Generalization

- Visualization of features (via t-SNE embedding)



Gist DeCAF1 DeCAF6

ILSVRC-2012 validation set

J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition, ICML 2014

Feature Generalization

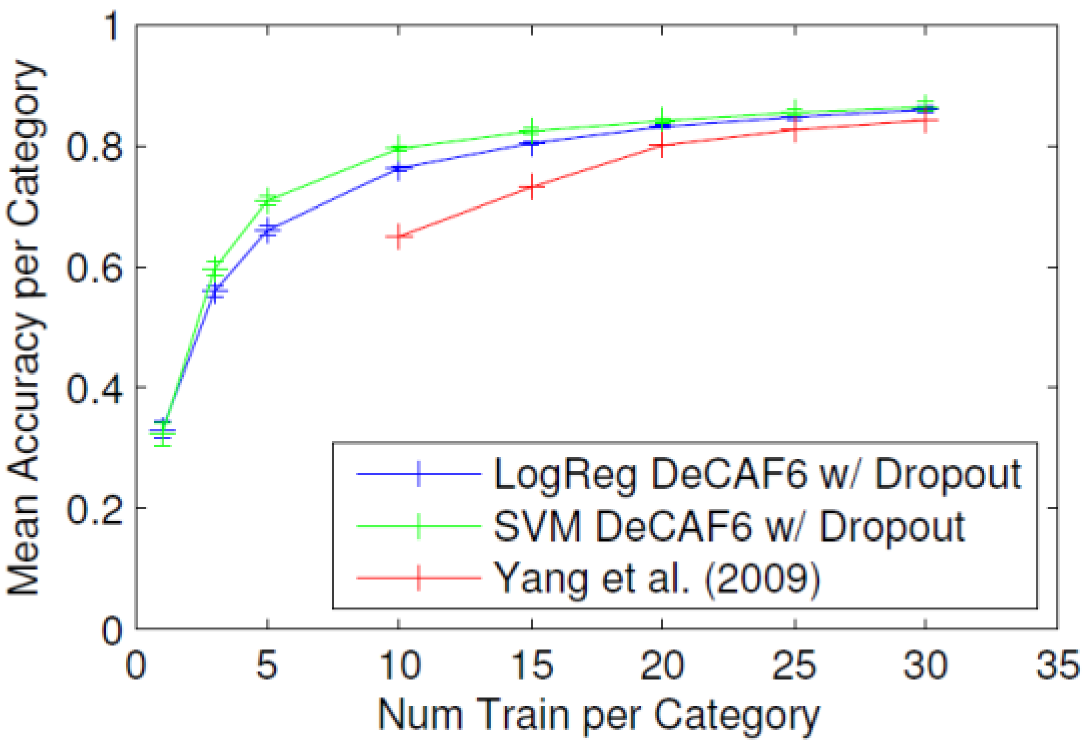
- Domain adaptation task

	Amazon → Webcam		
	SURF	DeCAF_6	DeCAF_7
Logistic Reg. (S)	9.63 ± 1.4	48.58 ± 1.3	53.56 ± 1.5
SVM (S)	11.05 ± 2.3	52.22 ± 1.7	53.90 ± 2.2
Logistic Reg. (T)	24.33 ± 2.1	72.56 ± 2.1	74.19 ± 2.8
SVM (T)	51.05 ± 2.0	78.26 ± 2.6	78.72 ± 2.3
Logistic Reg. (ST)	19.89 ± 1.7	75.30 ± 2.0	76.32 ± 2.0
SVM (ST)	23.19 ± 3.5	80.66 ± 2.3	79.12 ± 2.1
Daume III (2007)	40.26 ± 1.1	$\mathbf{82.14 \pm 1.9}$	81.65 ± 2.4
Hoffman et al. (2013)	37.66 ± 2.2	80.06 ± 2.7	80.37 ± 2.0
Gong et al. (2012)	39.80 ± 2.3	75.21 ± 1.2	77.55 ± 1.9
Chopra et al. (2013)		58.85	

J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition, ICML 2014

Feature Generalization

- Caltech 101 classification



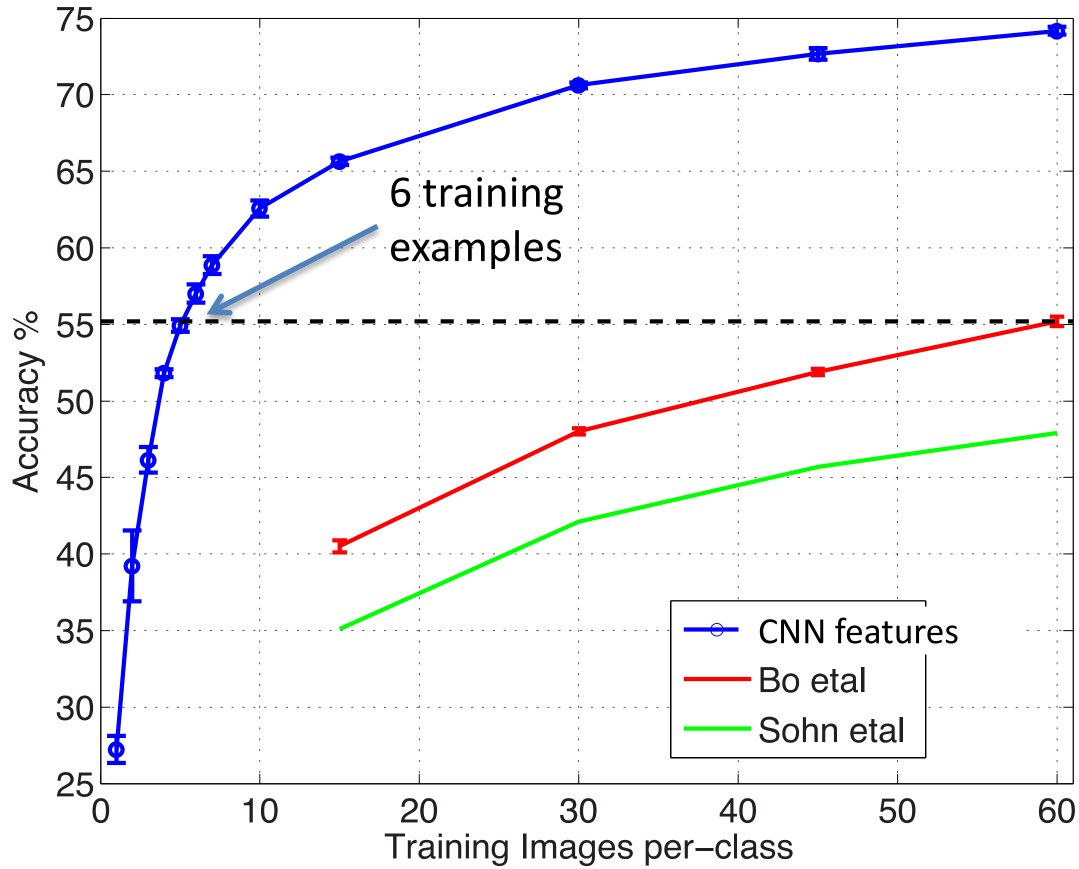
J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition, ICML 2014

Feature Generalization

- Zeiler & Fergus, arXiv 1311.2901, 2013 (Caltech-101,256)
- Girshick et al. CVPR'14 (Caltech-101, SunS)
- Oquab et al. CVPR'14 (VOC 2012)
- Razavian et al. arXiv 1403.6382, 2014 (lots of datasets)

- Pre-train on Imagnet

  Retrain classifier on Caltech256

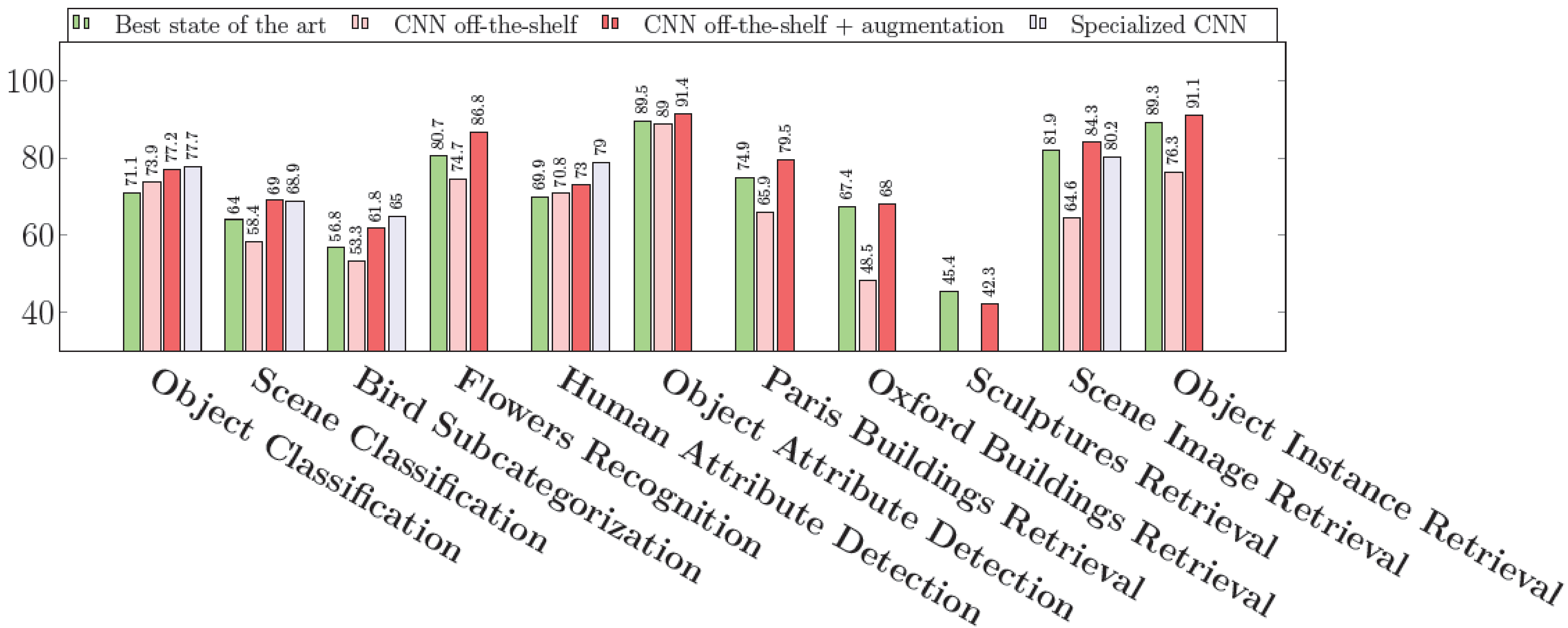


From Zeiler & Fergus, *Visualizing and Understanding Convolutional Networks*, arXiv 1311.2901, 2013

Slide credit: R. Fergus

Feature generalization over multiple tasks

- Generalization over multiple tasks



Ali Razavian, H. Azizpour, J. Sullivan, S. Carlsson, CNN Features off-the-shelf: an Astounding Baseline for Recognition, Arxiv 2014

P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. ICLR 2014

Using very deep layers: VGG Network

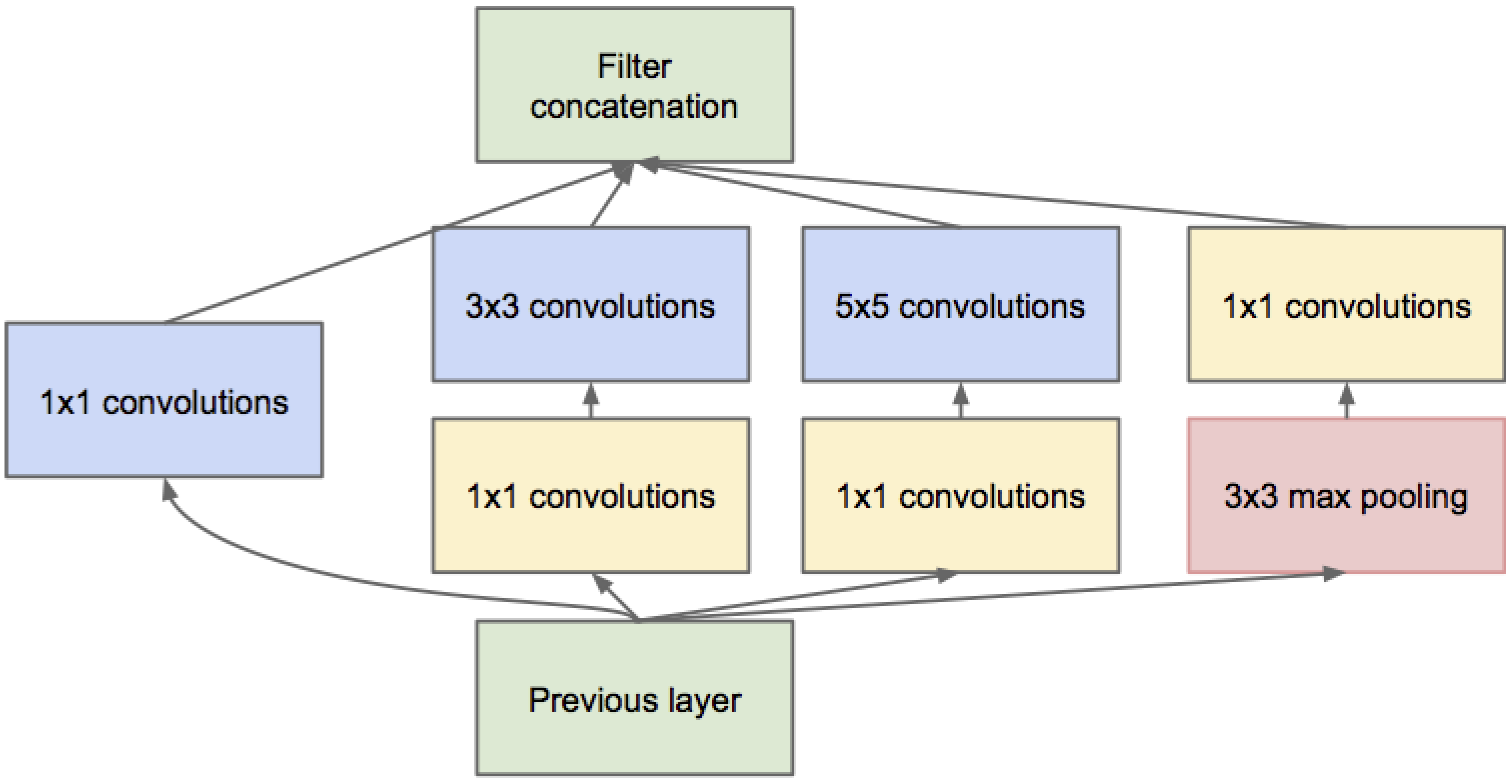
- Main idea: use many small convolutions with deep layers

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Simouyan et al., Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015

Going deeper: GoogLeNet

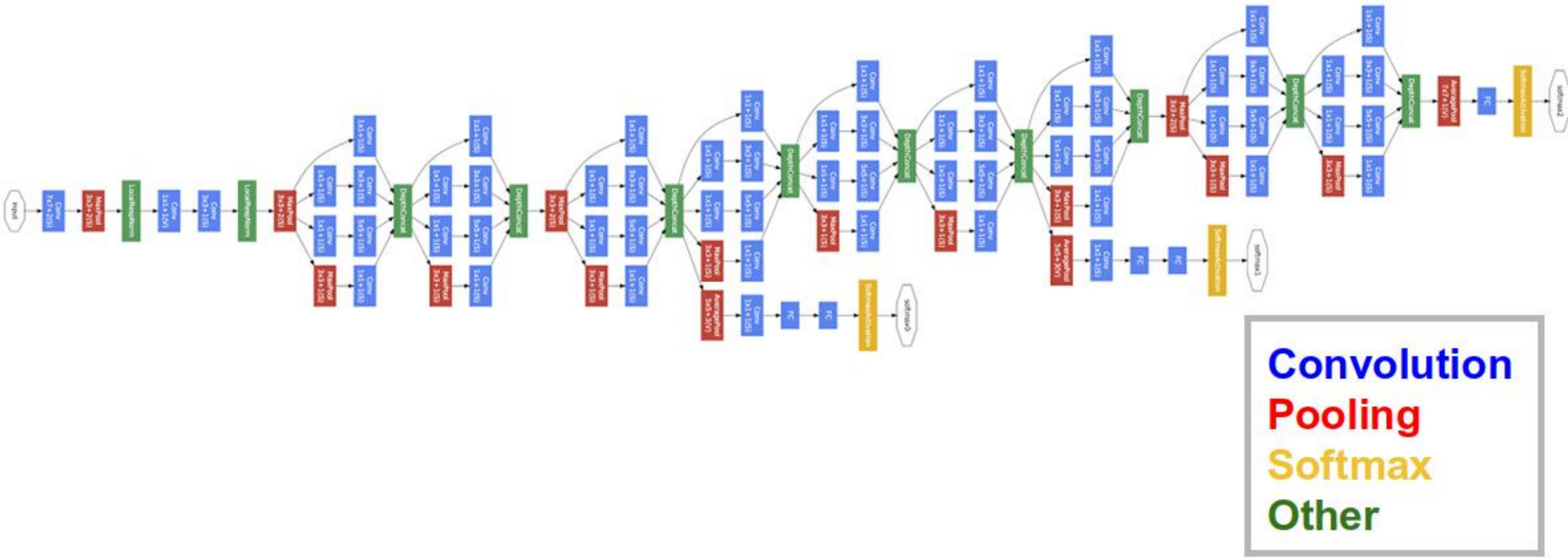
- Main idea: use multiple receptive fields + go deep



Szegedy et al. "Going deeper with convolutions." *CVPR 2015*

Going deeper: GoogLeNet

- Main idea: use multiple receptive fields + go deep



Szegedy et al. "Going deeper with convolutions." *CVPR 2015*

Experimental results on ILSVRC

Table 7: **Comparison with the state of the art in ILSVRC classification**. Our method is denoted as "VGG". Only the results obtained without outside training data are reported.

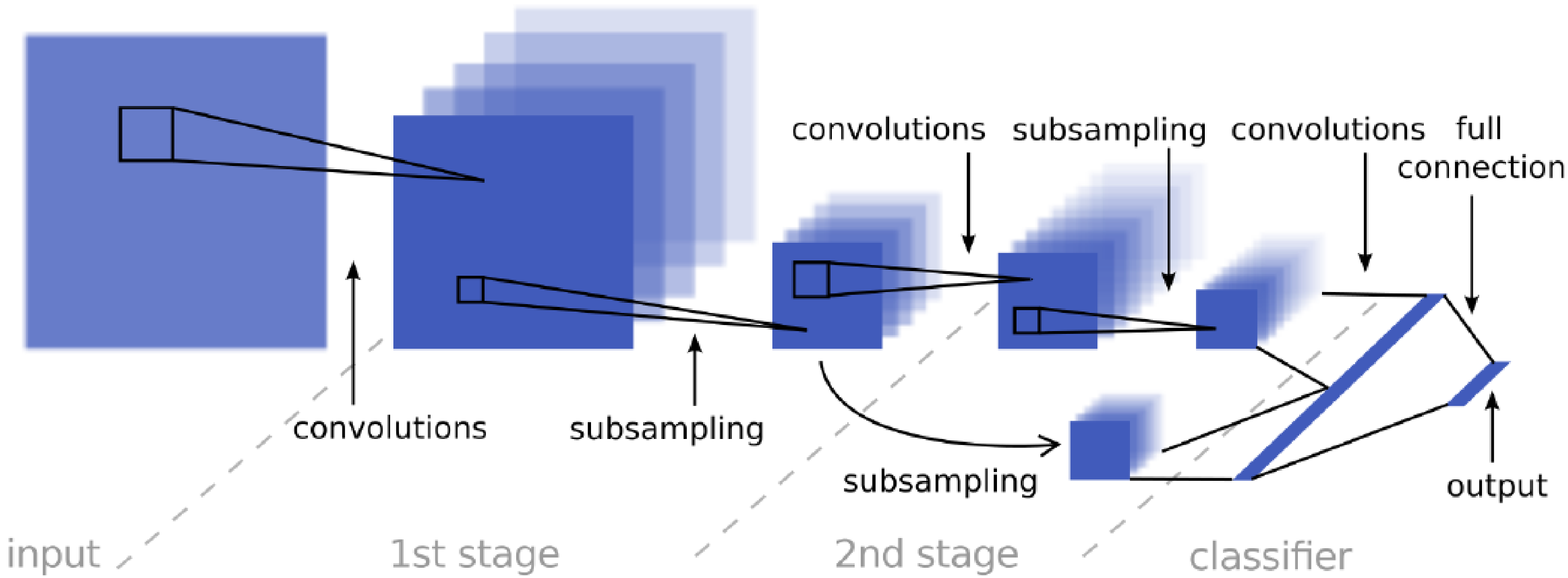
Method	top-1 val. error (%)	top-5 val. error (%)	top-5 test error (%)
VGG (2 nets, multi-crop & dense eval.)	**23.7**	**6.8**	**6.8**
VGG (1 net, multi-crop & dense eval.)	24.4	7.1	7.0
VGG (ILSVRC submission, 7 nets, dense eval.)	24.7	7.5	7.3
GoogLeNet (Szegedy et al., 2014) (1 net)	-	7.9	
GoogLeNet (Szegedy et al., 2014) (7 nets)	-	**6.7**	
MSRA (He et al., 2014) (11 nets)	-	-	8.1
MSRA (He et al., 2014) (1 net)	27.9	9.1	9.1
Clarifai (Russakovsky et al., 2014) (multiple nets)	-	-	11.7
Clarifai (Russakovsky et al., 2014) (1 net)	-	-	12.5
Zeiler & Fergus (Zeiler & Fergus, 2013) (6 nets)	36.0	14.7	14.8
Zeiler & Fergus (Zeiler & Fergus, 2013) (1 net)	37.5	16.0	16.1
OverFeat (Sermanet et al., 2014) (7 nets)	34.0	13.2	13.6
OverFeat (Sermanet et al., 2014) (1 net)	35.7	14.2	-
Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)	38.1	16.4	16.4
Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)	40.7	18.2	-

Simouyan et al., Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015

Other vision applications

Object detection using multi-scale CNN

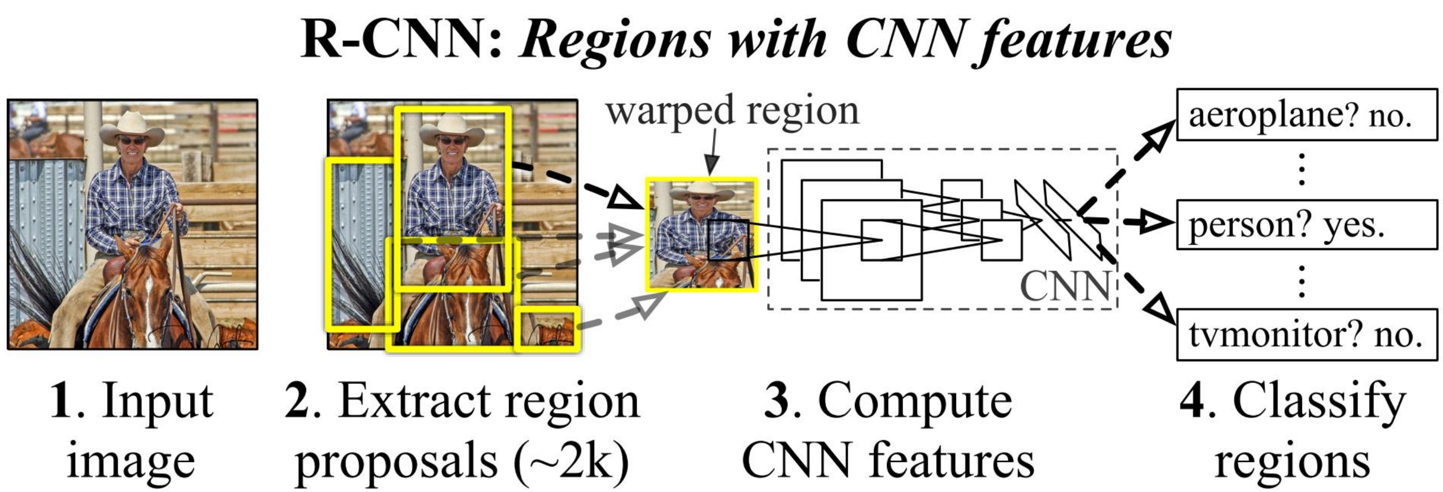
- Initialization for convolutional network



Kavukcuoglu et al. Learning convolutional feature hierarchies for visual recognition. NIPS 2010.

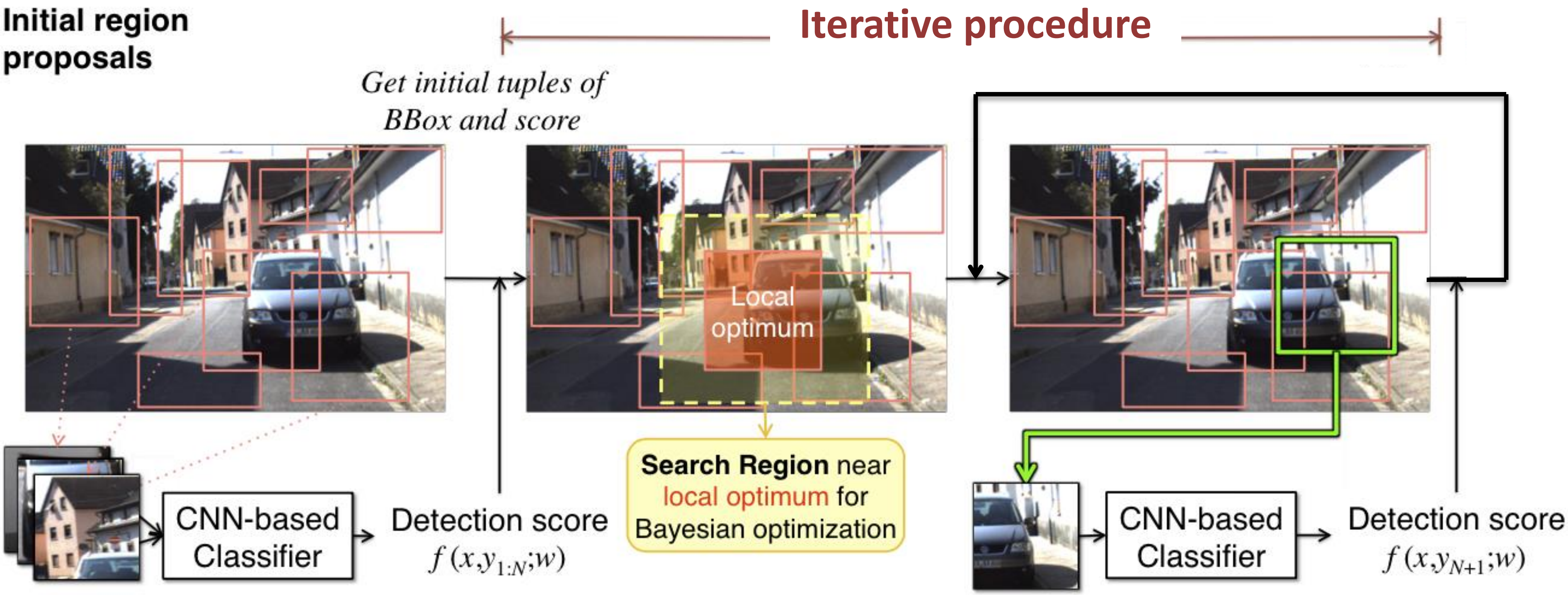
Object detection using Convolutional Neural Networks

- Object detection systems based on the deep convolutional neural network (CNN) have recently made ground-breaking advances.
- The state-of-the-art: “Regions with CNN” (R-CNN)



Girshick et al, “Region-based Convolutional Networks for Accurate Object Detection and Semantic Segmentation”, PAMI, 2015.

CNN Object detection with Bayesian optimization



	IoU>0.5	IoU>0.7
Mean Average Precision	**Standard localization**	**More accurate localization**
R-CNN (VGGNet)	65.4	35.2
Zhang et al., 2015	**68.5**	**43.0**

Improving Object Detection with Deep Convolutional Networks via Bayesian Optimization and Structured Prediction. Zhang, Sohn, Villegas, Pan, and Lee, CVPR 2015

CNN object detection with structured loss

- Linear classifier

$$g(x; w) = \mathrm{argmax}_{y \in \mathcal{Y}} f(x, y; w)$$

$$f(x, y; w) = w^\top \tilde{\phi}(x, y)$$

CNN features

$$\tilde{\phi}(x, y) = \begin{cases} \phi(x, y), & l = +1 \\ \mathbf{0}, & l = -1 \end{cases}$$

- Minimizing the structured loss (Blaschko and Lampert, 2008)*

$$\hat{w} = \mathrm{argmax}_w \sum_{i=1}^{M} \Delta(g(x_i; w), y_i)$$

$$\Delta(y, y_i) = \begin{cases} 1 - \mathrm{IoU}(y, y_i), & \text{if } l = l_i = 1 \\ 0, & \text{if } l = l_i = -1 \\ 1, & \text{if } l \neq l_i \end{cases}$$

* Blaschko and Lampert, "Learning to localize objects with structured output regression", ECCV, 2008.

Other related work: LeCun et al. 1989; Taskar et al. 2005; Joachims et al. 2005; Veldaldi et al. 2014; Thomson et al. 2014; and many others

CNN object detection with structured loss

- The objective is hard to solve. Replace it with an upper-bound surrogate using structured SVM framework

$$\min_{w} \quad \frac{1}{2} \| w \|^2 + \frac{C}{M} \sum_{i=1}^{M} \xi_i \qquad \text{, subject to}$$

$$w^\top \tilde{\phi}(x_i, y_i) \geq w^\top \tilde{\phi}(x_i, y) + \Delta(y, y_i) - \xi_i, \forall y \in \mathcal{Y}, \forall i$$

$$\xi_i \geq \; 0, \forall i$$

- The constraints can be re-written as:

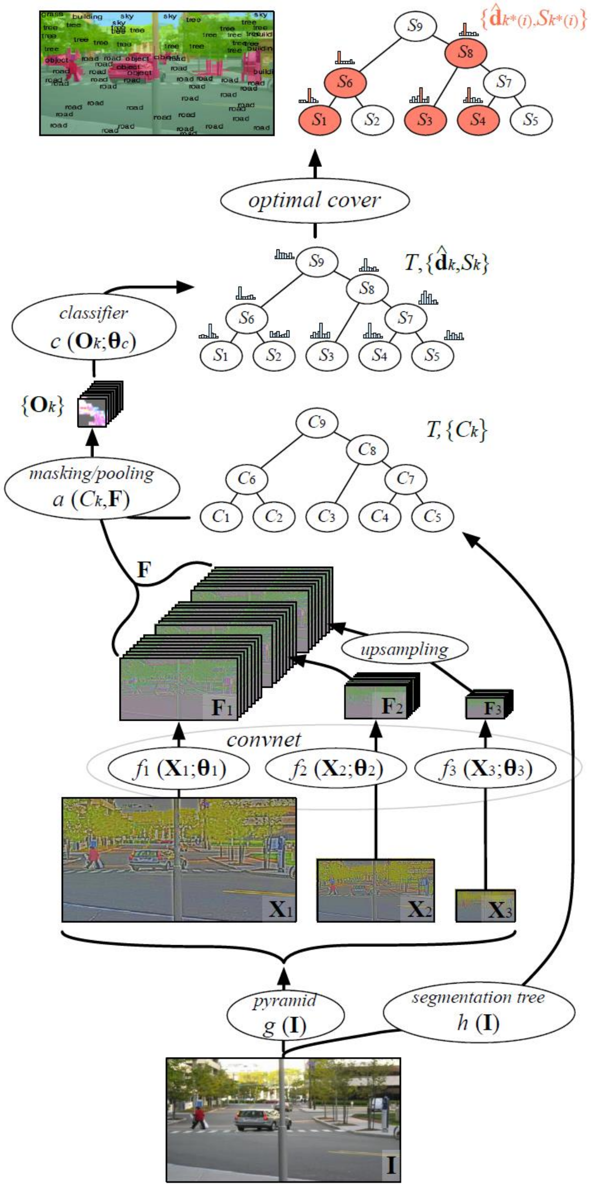
$$\left.\begin{aligned} w^\top \phi(x_i, y_i) &\geq \; 1 - \xi_i, & \forall i \in I_{\text{pos}}, \\ w^\top \phi(x_i, y) &\leq \; -1 + \xi_i, & \forall y \in \mathcal{Y}, \forall i \in I_{neg}, \end{aligned}\right\} \text{Recognition}$$

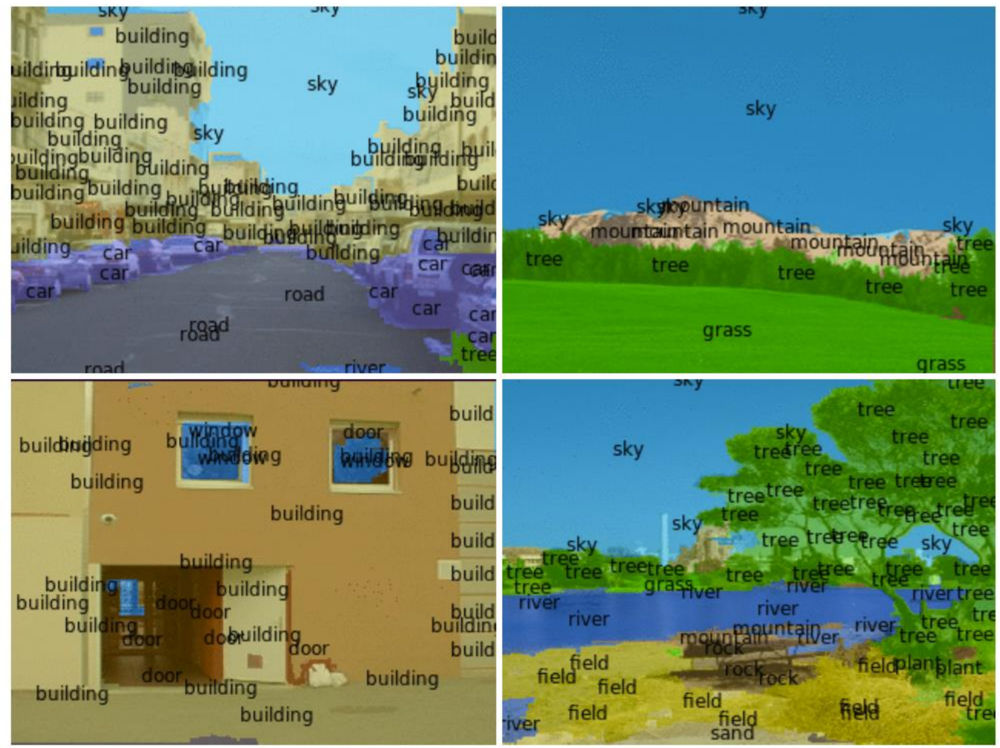
$$\left.\begin{aligned} w^\top \phi(x_i, y_i) \geq \; & w^T \phi(x_i, y) + \Delta^{\text{loc}}(y, y_i) - \xi_i, \\ & \forall y \in \mathcal{Y}, \forall i \in I_{\text{pos}}, \end{aligned}\right\} \text{Localization}$$

where $\Delta^{\text{loc}}(y, y_i) = 1 - \text{IoU}(y, y_i)$.

Improving Object Detection with Deep Convolutional Networks via Bayesian Optimization and Structured Prediction. Zhang, Sohn, Villegas, Pan, and Lee, CVPR 2015

Image segmentation and parsing





Farabet et al., Scene Parsing with Multiscale Feature Learning, Purity Trees, and Optimal Covers, ICML 2012

Other Applications

- Tracking (Bazzani et. al. 2010, and many others)

- Pose estimation (Toshev et al. 2013, Jain et al., 2013, …)

- Caption generation (Vinyals et al. 2015, Xu et al. 2015, …)

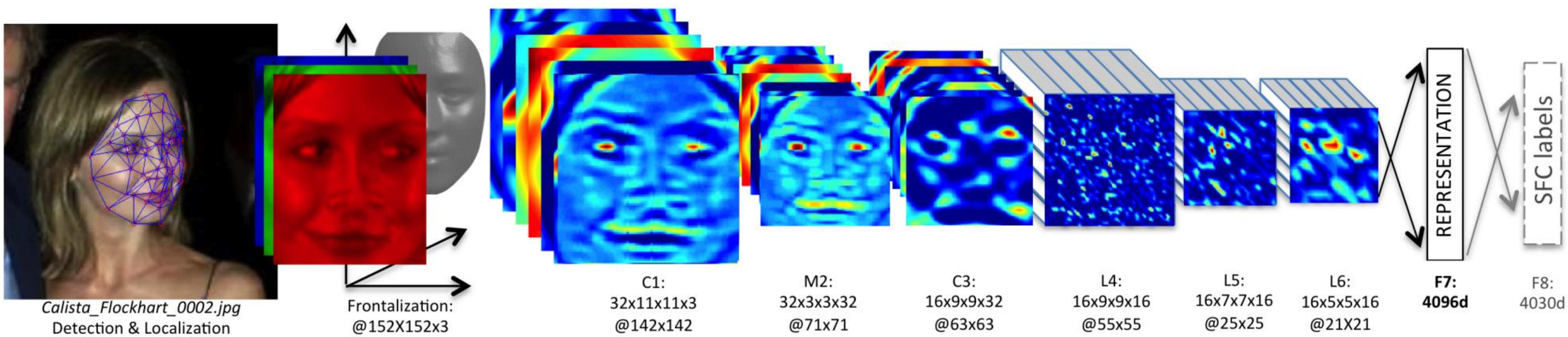
A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

Industry Deployment

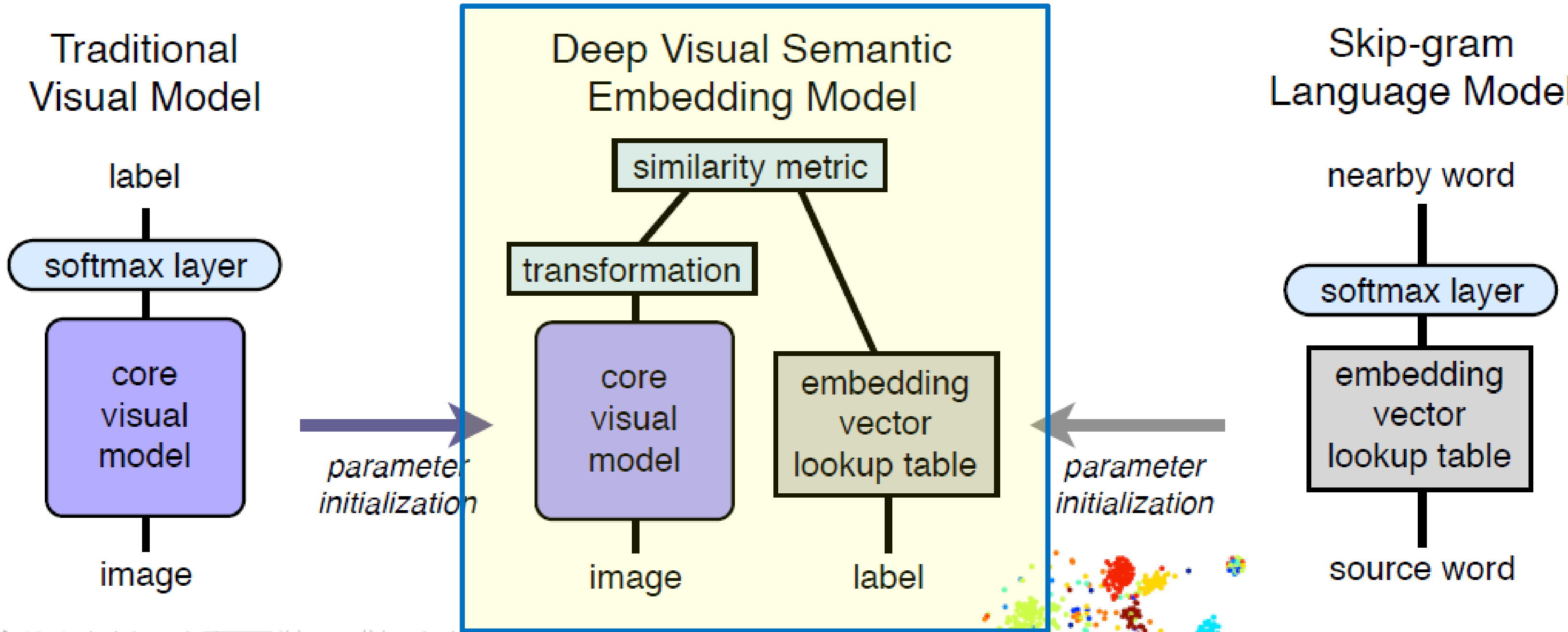
- Used in Facebook, Google, Microsoft
- Image Recognition, Speech Recognition, ….
- Fast at test time



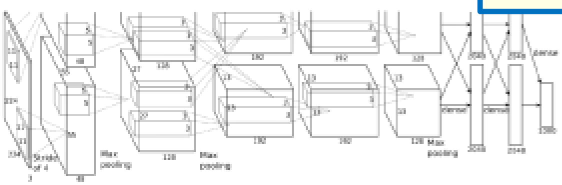
Taigman et al. DeepFace: Closing the Gap to Human-Level Performance in Face Verification, CVPR'14

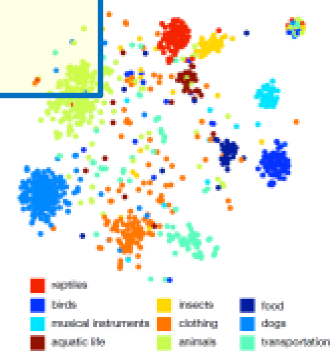
Slide: R. Fergus

Deep Visual-Semantic Embedding



CNN





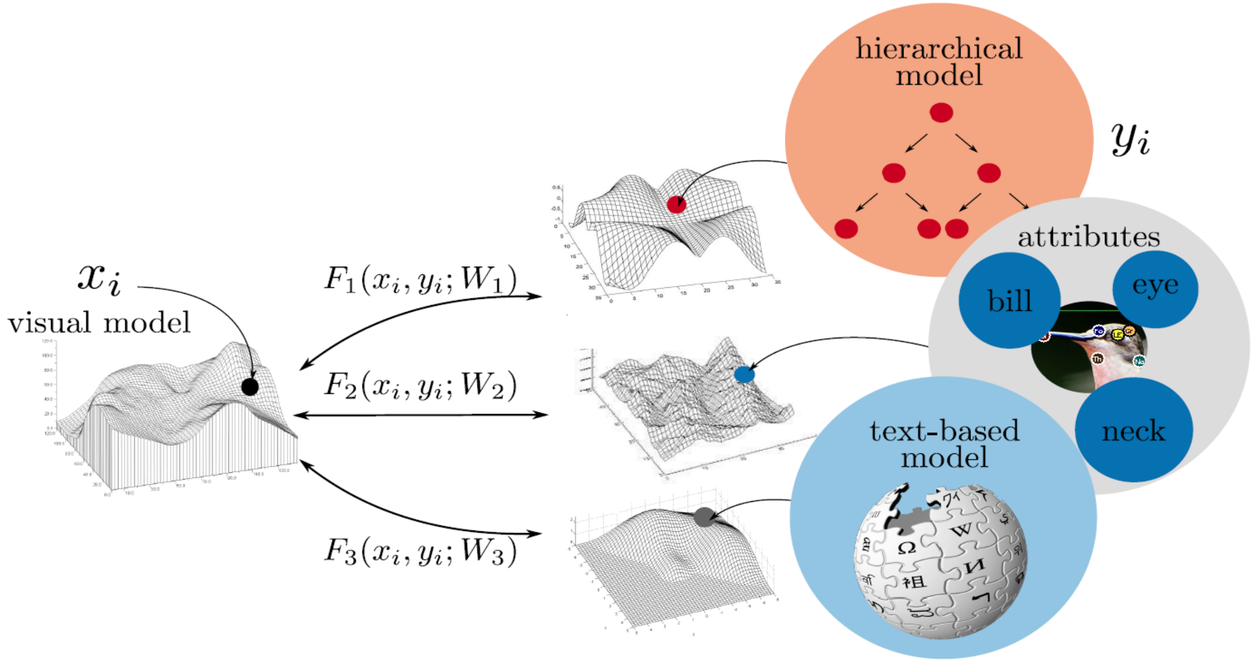
Skipgram for
word embedding

Visualization of
label embedding

$$loss(image, label) = \sum_{j \neq label} \max[0, margin - \vec{t}_{label} M \vec{v}(image) + \vec{t}_j M \vec{v}(image)]$$

Frome et al., NIPS 2013

Multiple output embeddings for zero-shot learning



Classification using compatibility function: $f(x;W) = \arg\max_{y \in \mathcal{Y}} F(x,y;W)$

$$= \arg\max_{y \in \mathcal{Y}} \theta(x)^\top W \varphi(y)$$

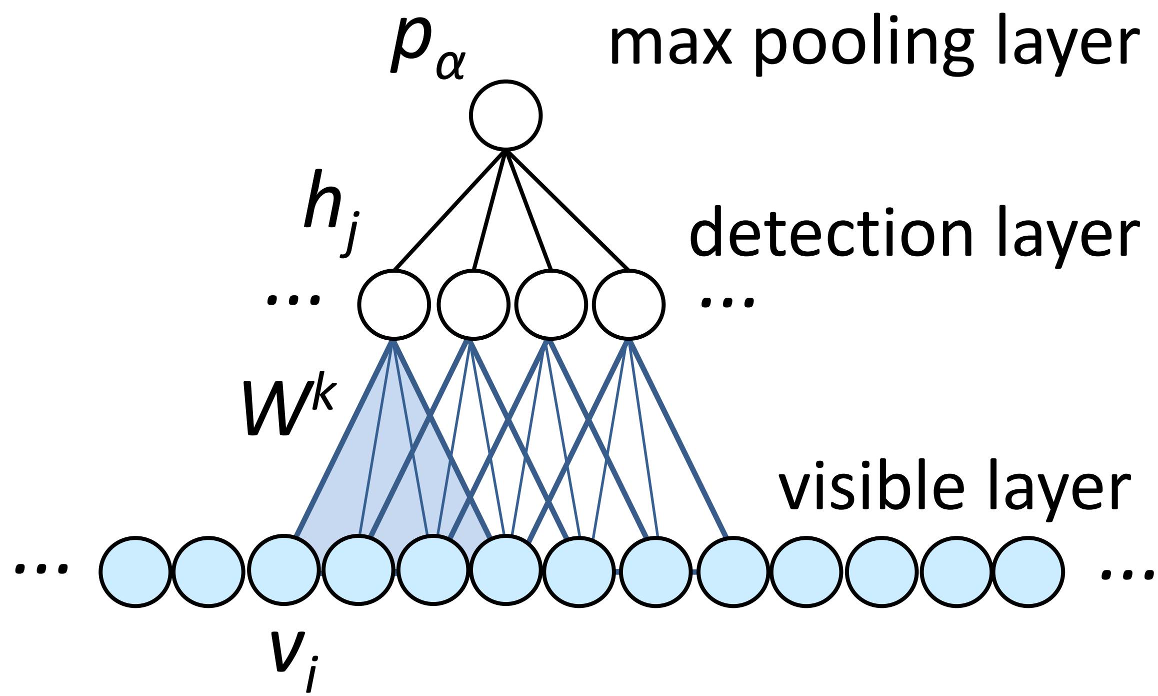
Combination of multiple output embeddings:

$$F(x,y;\{W\}_{1..K}) = \sum_k \alpha_k \theta(x)^\top W_k \varphi_k(y) \text{ s.t. } \sum_k \alpha_k = 1$$

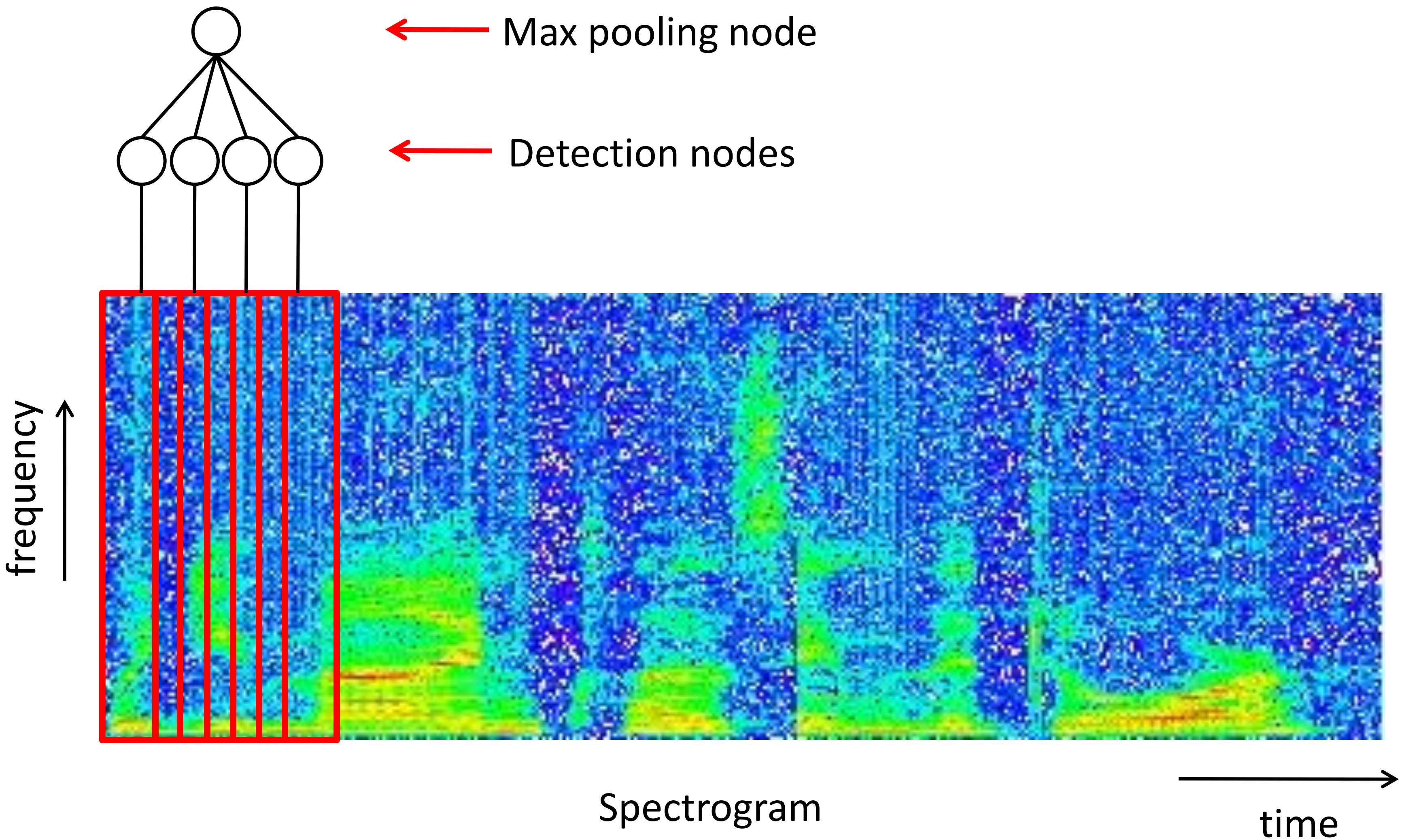
Akata, Reed, Walter, Lee, & Schiele. Evaluation of Output Embeddings for Fine-Grained Image Classification. CVPR 2015.

Convolutional networks for other domains: speech

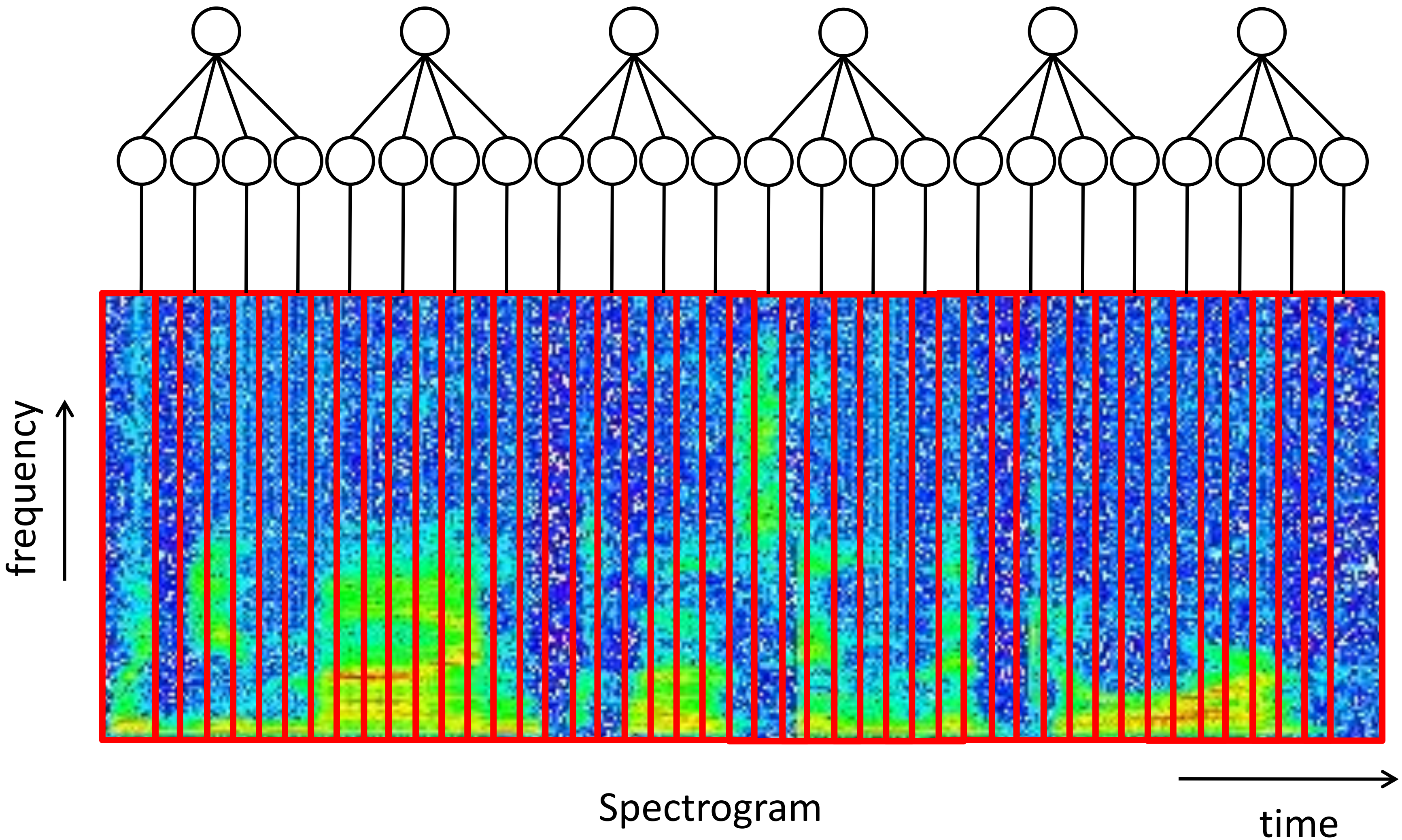
Convolutional RBM for time-series data



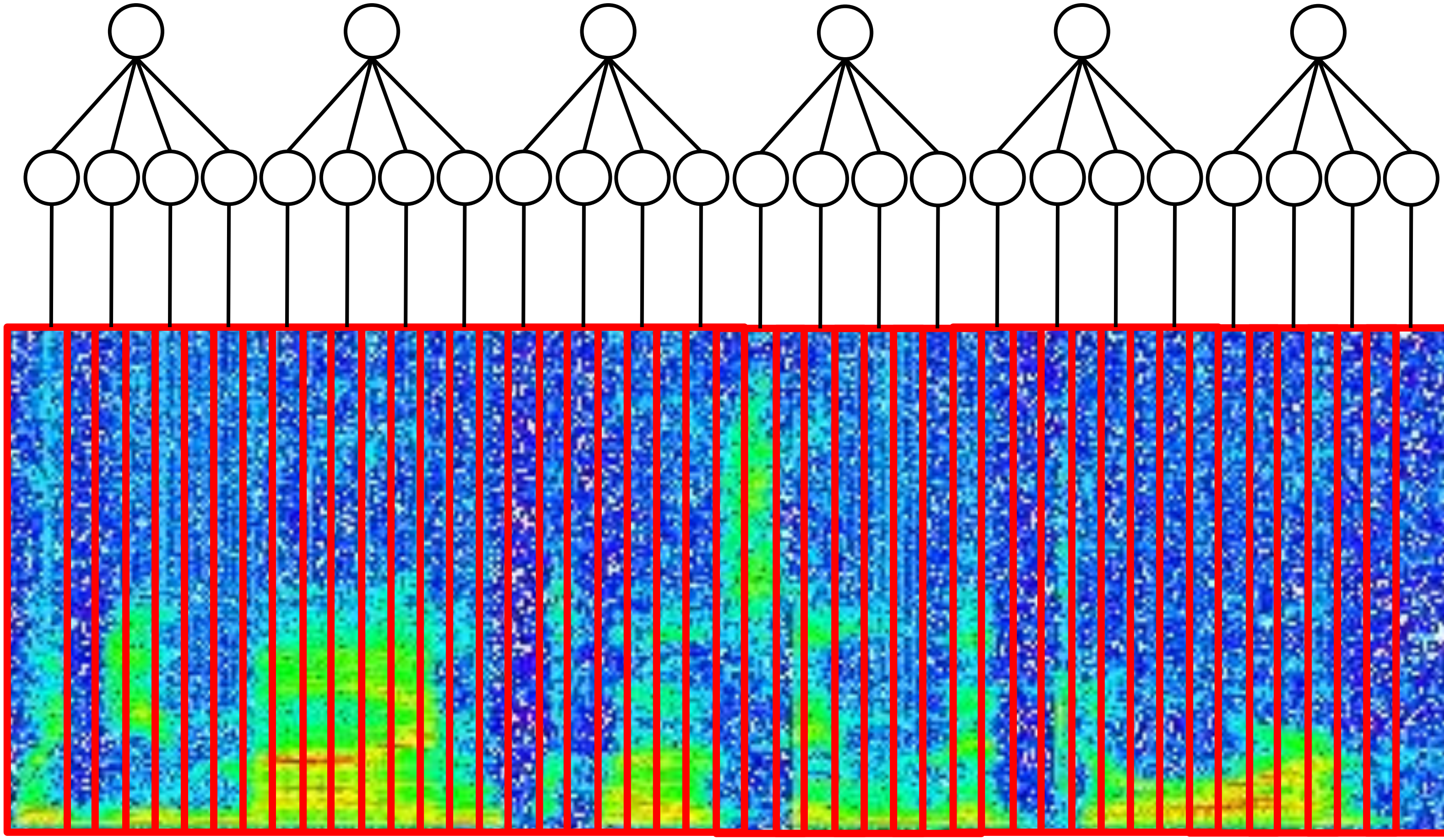
Convolutional DBN for audio [NIPS 2009]



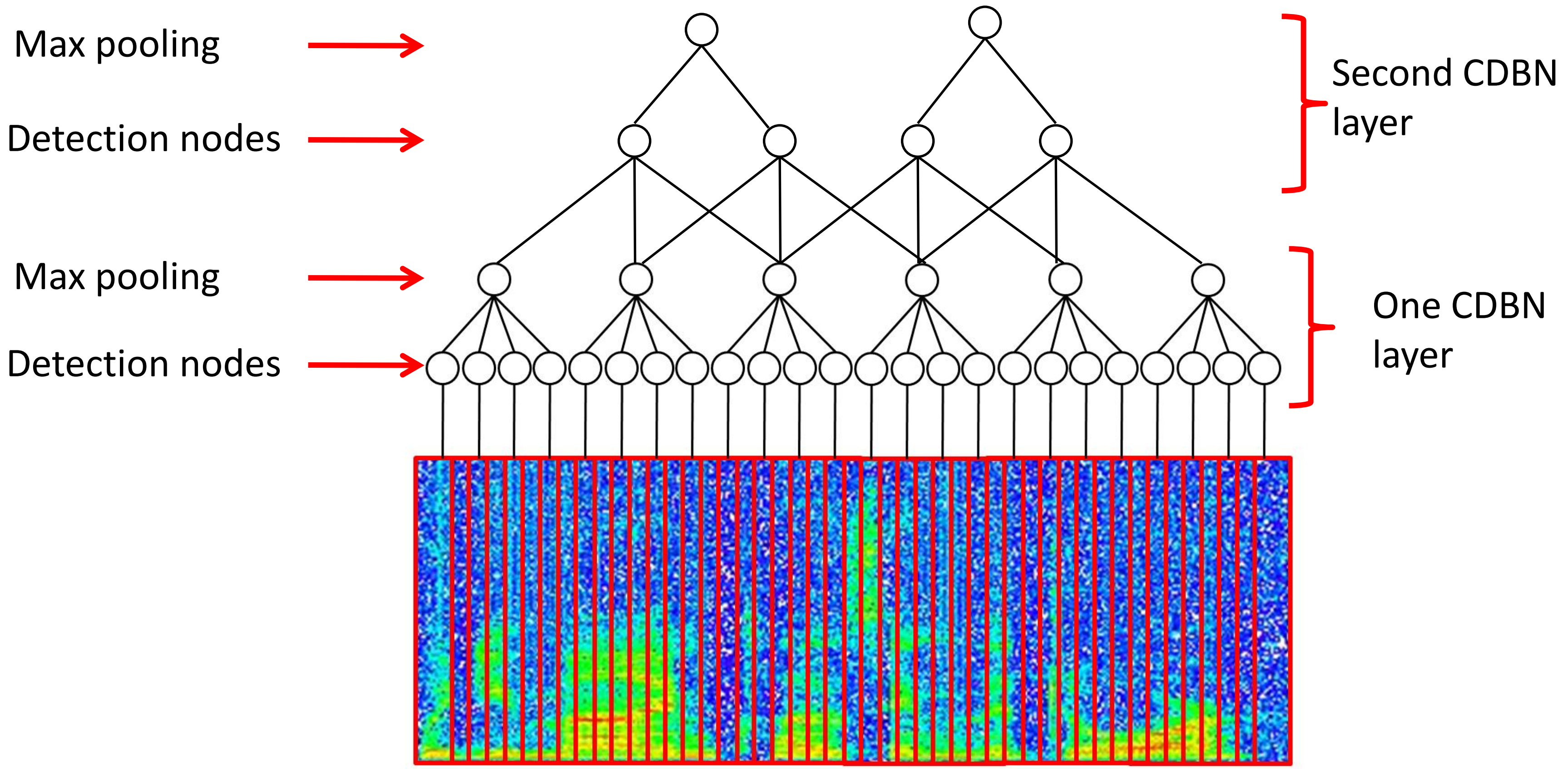
Convolutional DBN for audio [NIPS 2009]



Convolutional DBN for audio [NIPS 2009]

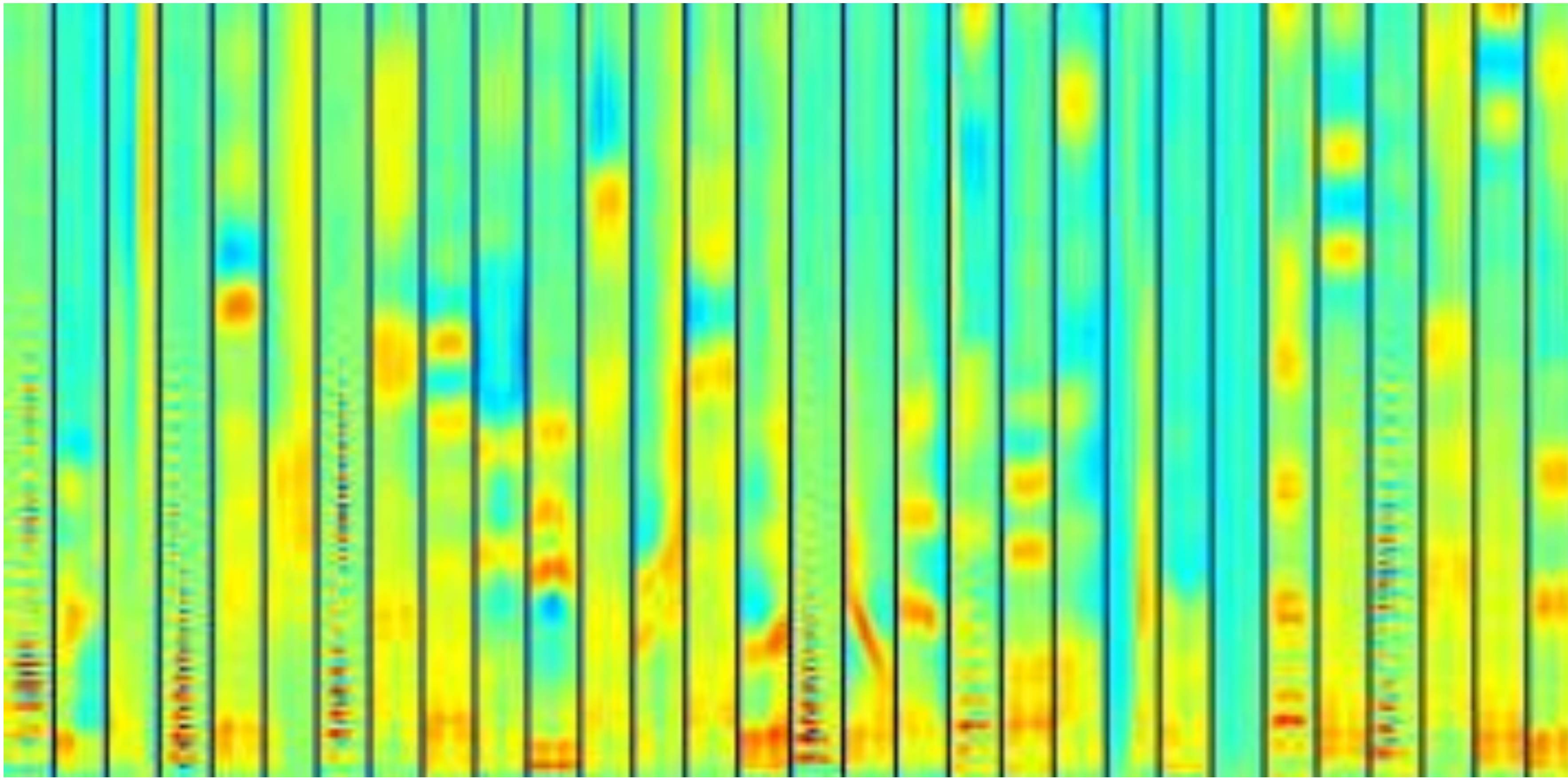


Convolutional DBN for audio [NIPS 2009]



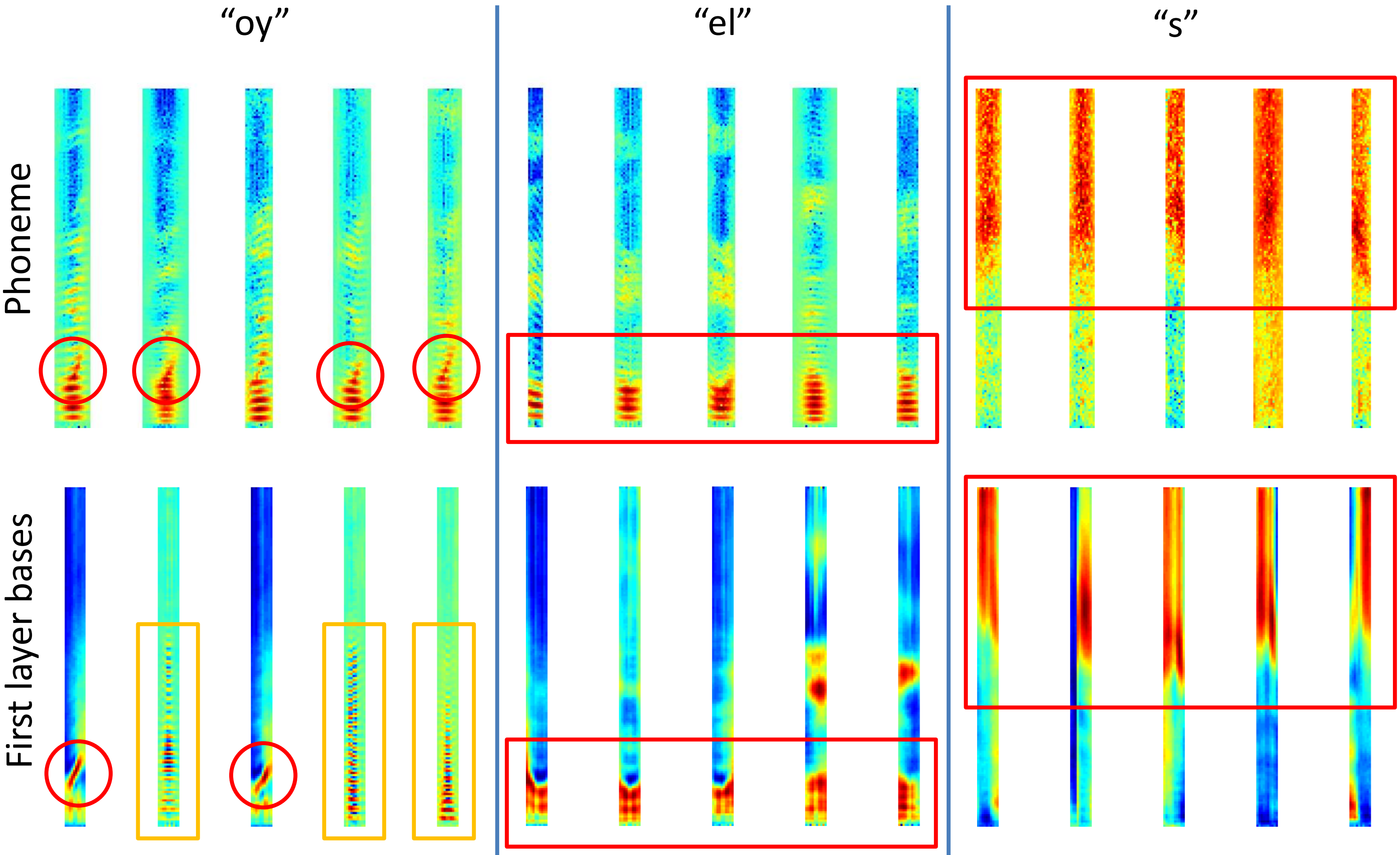
CDBNs for speech

Trained on unlabeled TIMIT corpus

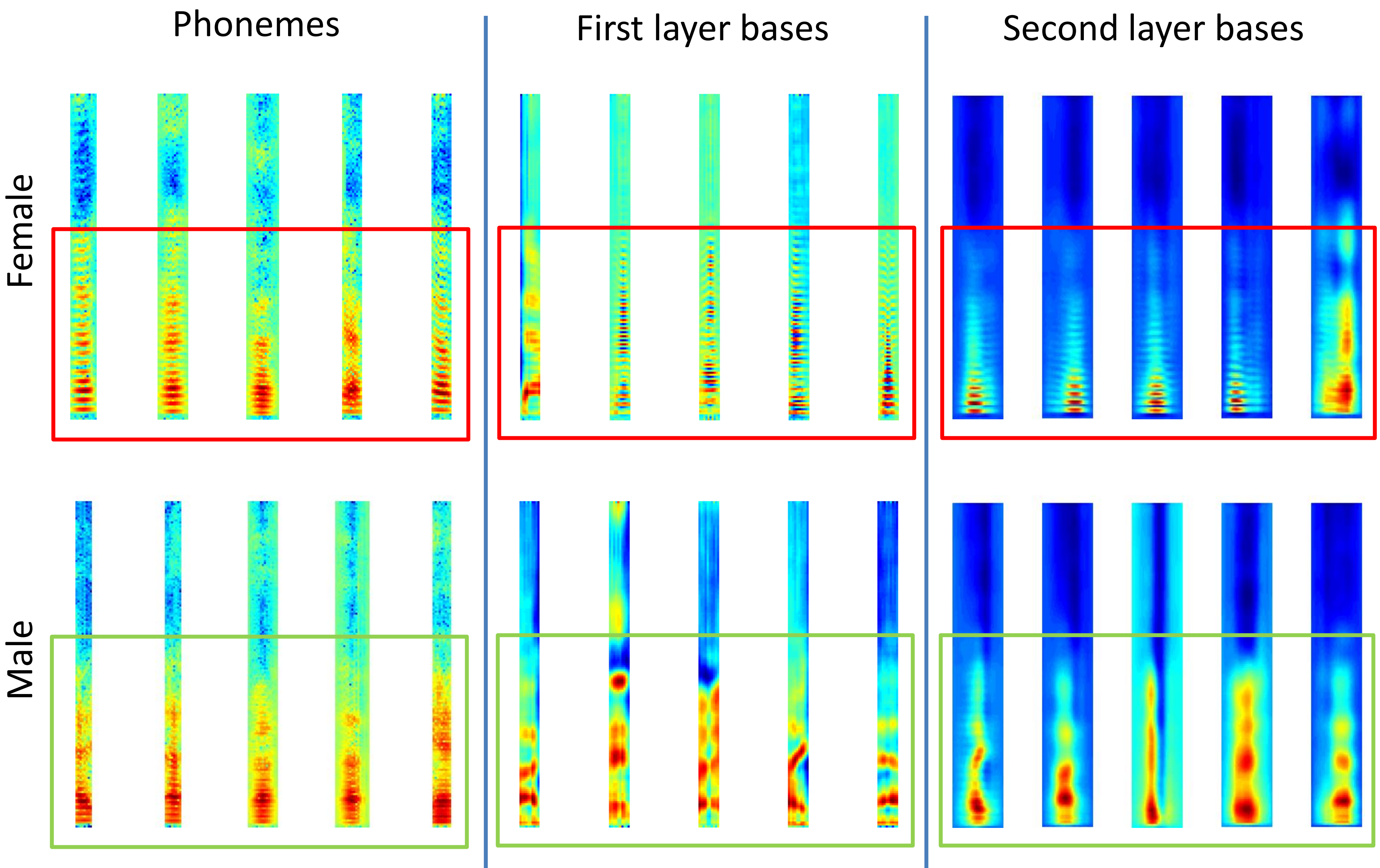


Learned first-layer bases

Comparison of bases to phonemes



Comparison of bases to gender ("ae" phoneme)



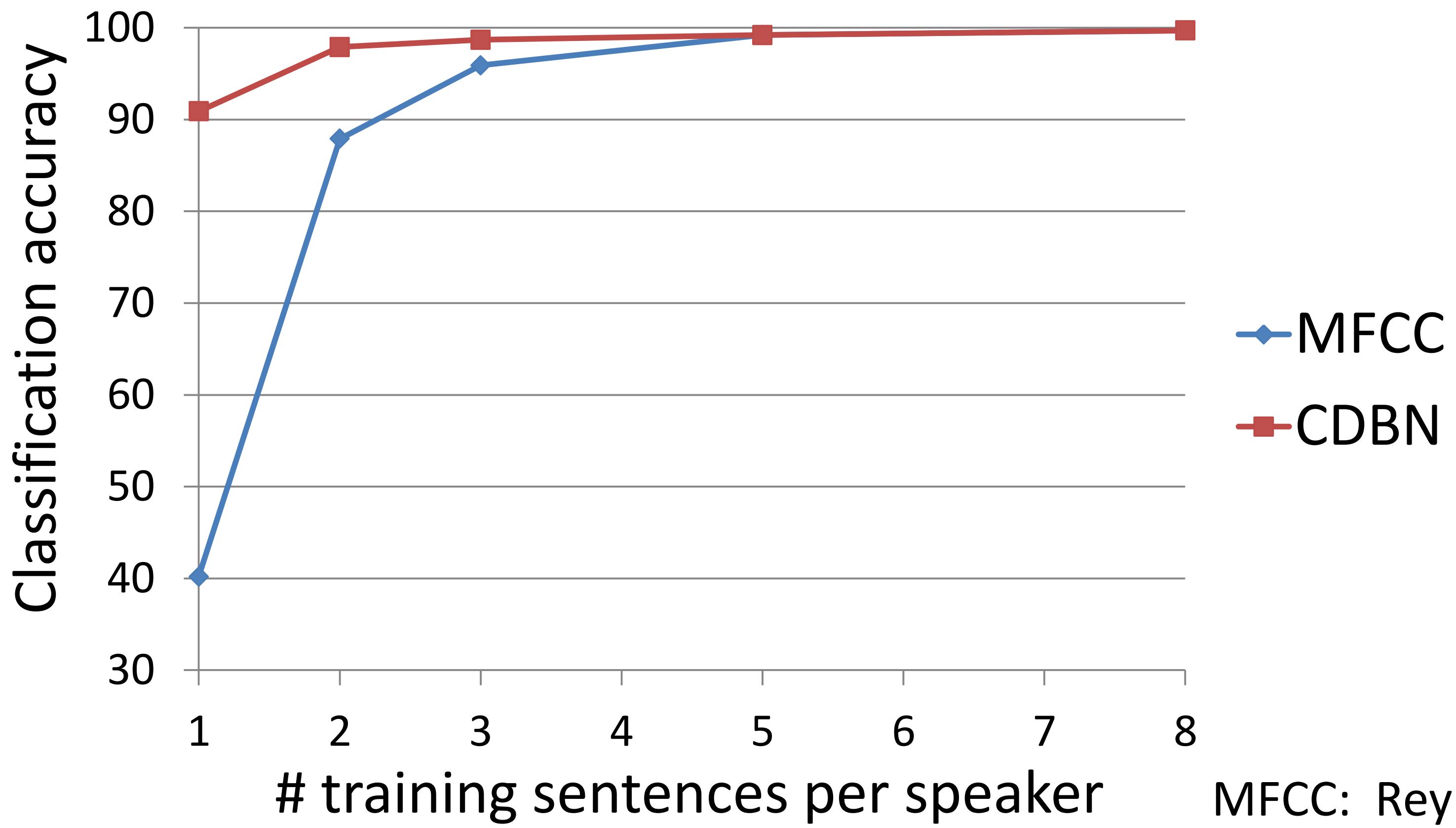
Application to speech recognition tasks

- Speaker identification
- Phoneme classification
- Gender classification

Use same set of learned features (computed from the same CDBN) for all three tasks.

Speaker Identification [NIPS 2009]

* 168 speakers, 10 sentences/speaker.

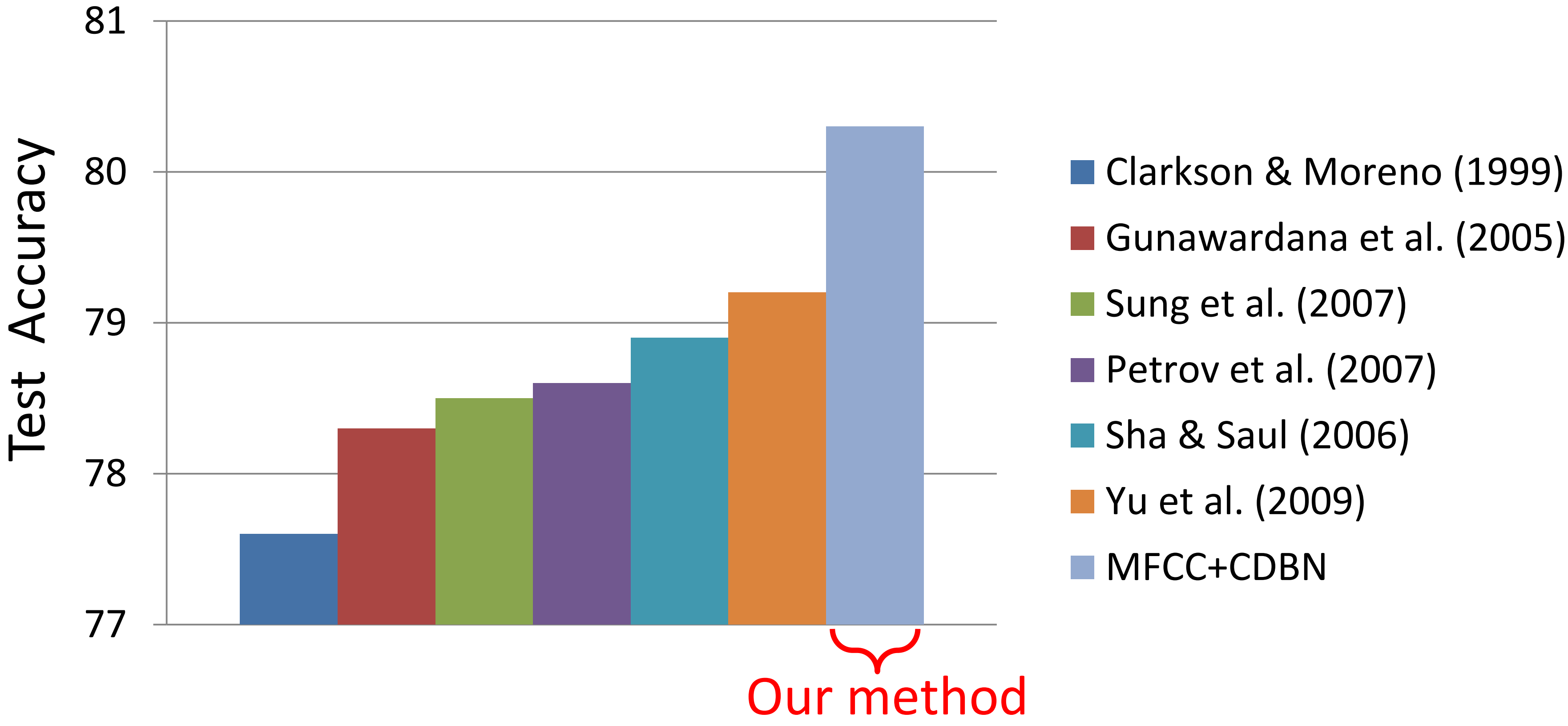


MFCC: Reynolds (1995)

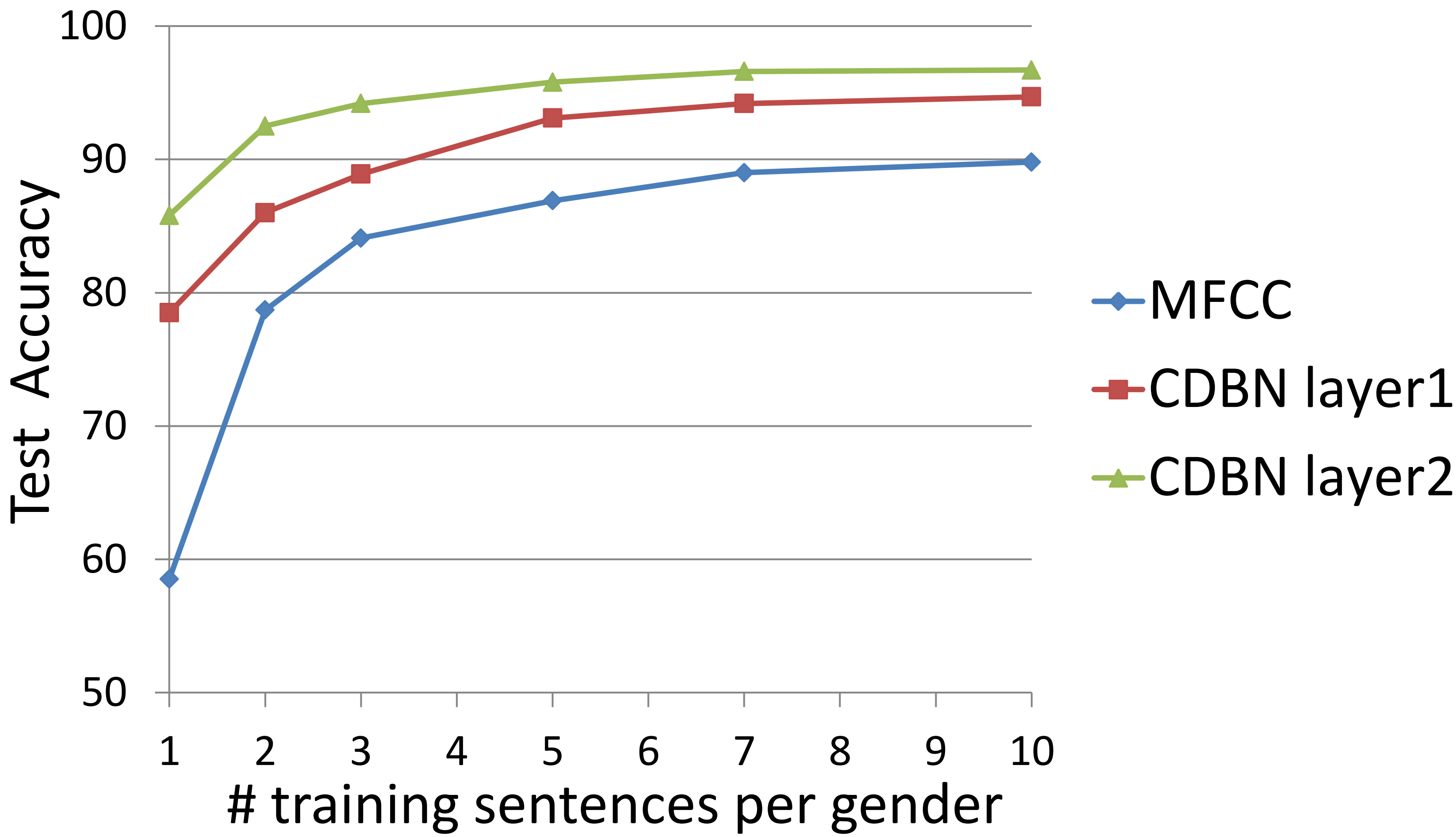
The CDBN features outperform the MFCC features especially when the number of training examples is small.

Phoneme Classification [NIPS 2009]

* Tested on the (standard) TIMIT core test set.

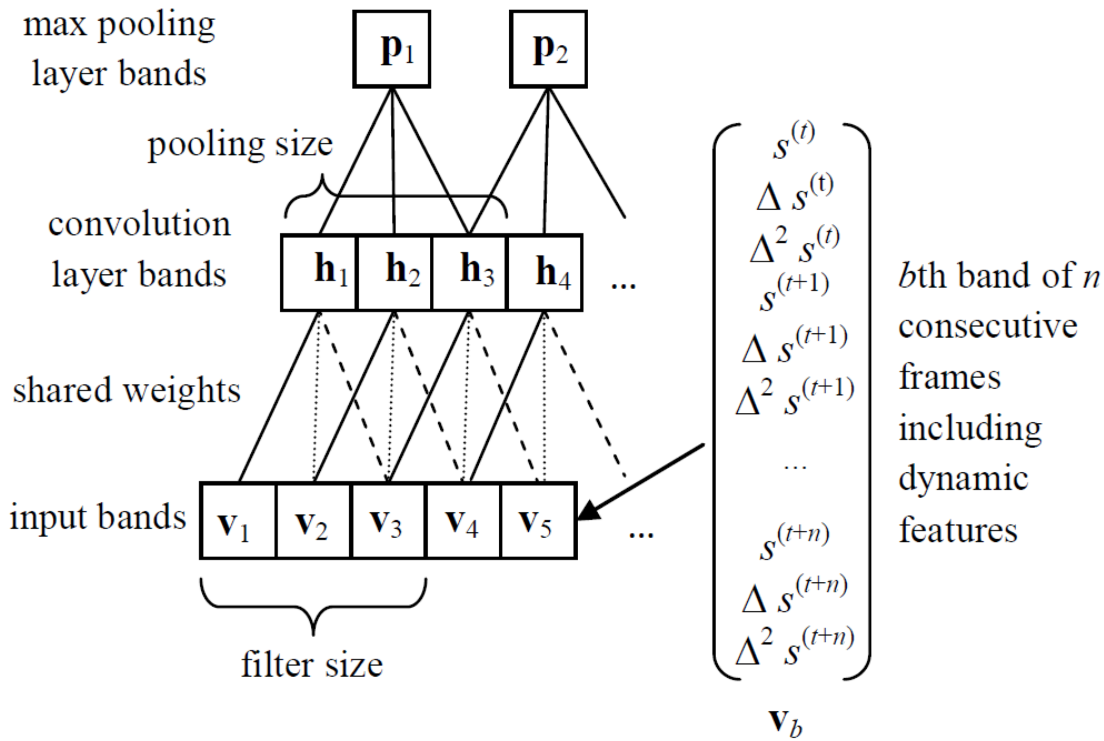


Gender Classification [NIPS 2009]



- The CDBN features outperform the MFCC features.
- The second layer CDBN features give better performance than the first layer CDBN features.

Convolutional neural networks for speech recognition



Abdel-Hamid, O., Mohamed, A. R., Jiang, H., & Penn, G. Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In *ICASSP 2012.*

Convoultional networks for music recommendation

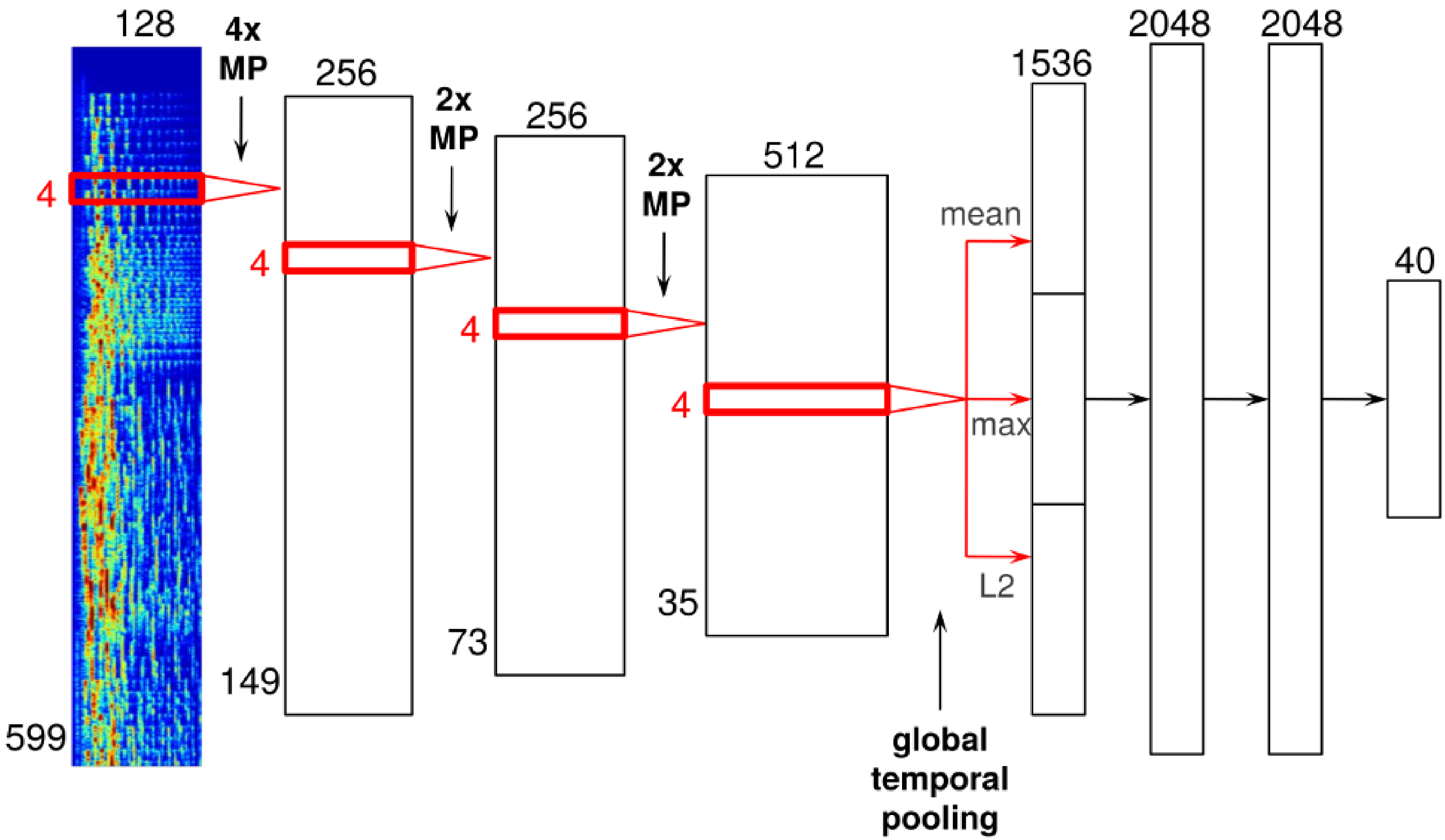


Image from: http://benanne.github.io/2014/08/05/spotify-cnns.html
Related work: Van den Oord, Dieleman & Schrauwen. Deep content-based music recommendation. In NIPS 2013