Scene Understanding With Deep Learning

Yann LeCun
Center for Data Science
Courant Institute of Mathematical Sciences
Center for Neural Science
New York University
http://yann.lecun.com

Deep Learning = Learning Representations/Features

- The traditional model of pattern recognition (since the late 50's)
  - Fixed/engineered features (or fixed kernel) + trainable classifier

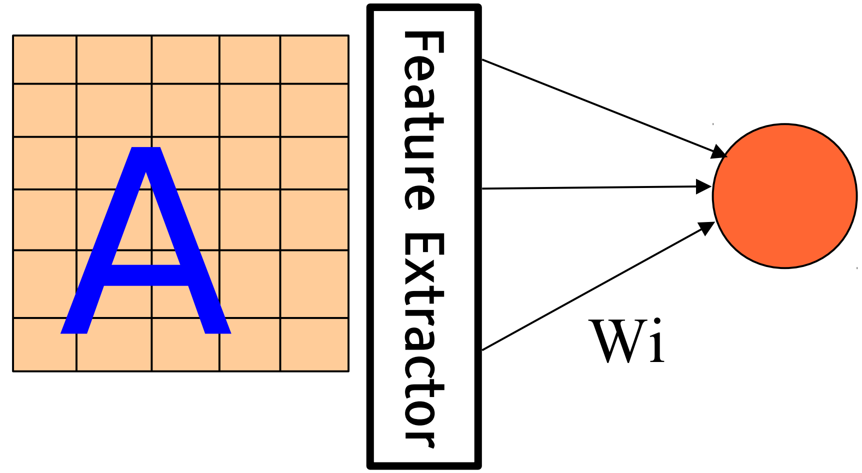
hand-crafted Feature Extractor → "Simple" Trainable Classifier →

- End-to-end learning / Feature learning / Deep learning
  - Trainable features (or kernel) + trainable classifier

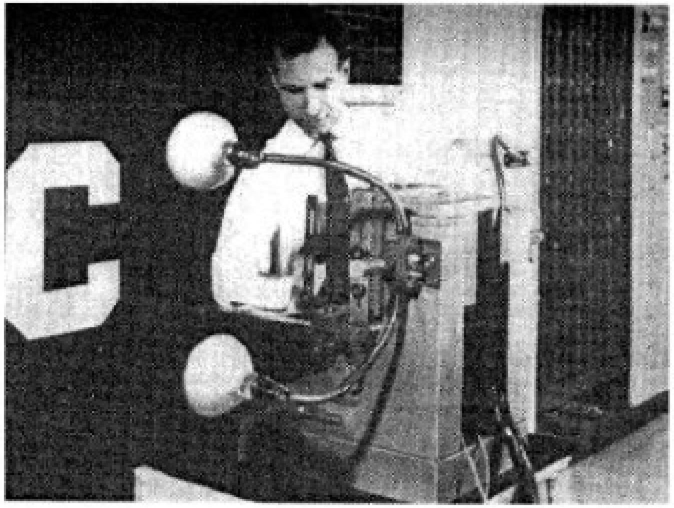
Trainable Feature Extractor → Trainable Classifier →

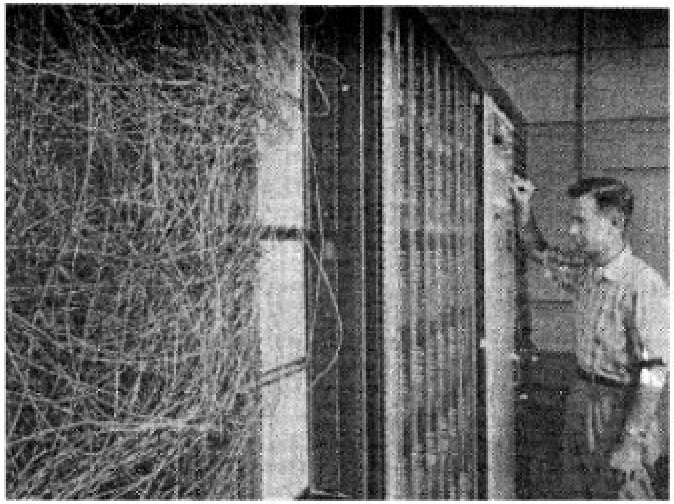
This Basic Model has not evolved much since the 50's

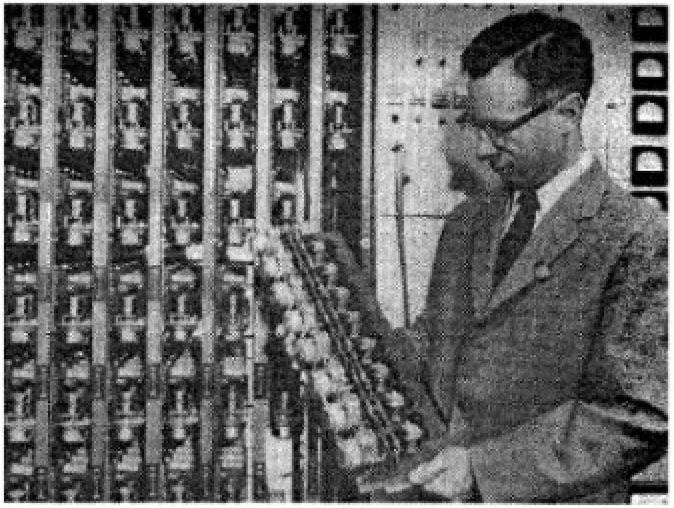
- The first learning machine: the Perceptron
  - Built at Cornell in 1960
- The Perceptron was a linear classifier on top of a simple feature extractor
- The vast majority of practical applications of ML today use glorified linear classifiers or glorified template matching.
- Designing a feature extractor requires considerable efforts by experts.



$$y = sign\left(\sum_{i=1}^{N} W_i F_i(X) + b\right)$$



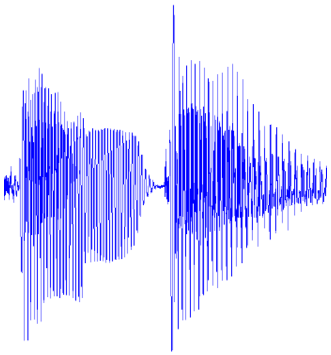




Architecture of "Mainstream" Pattern Recognition Systems

Modern architecture for pattern recognition

- Speech recognition: early 90's – 2011



MFCC → Mix of Gaussians → Classifier →

fixed | unsupervised | supervised

- Object Recognition: 2006 - 2012

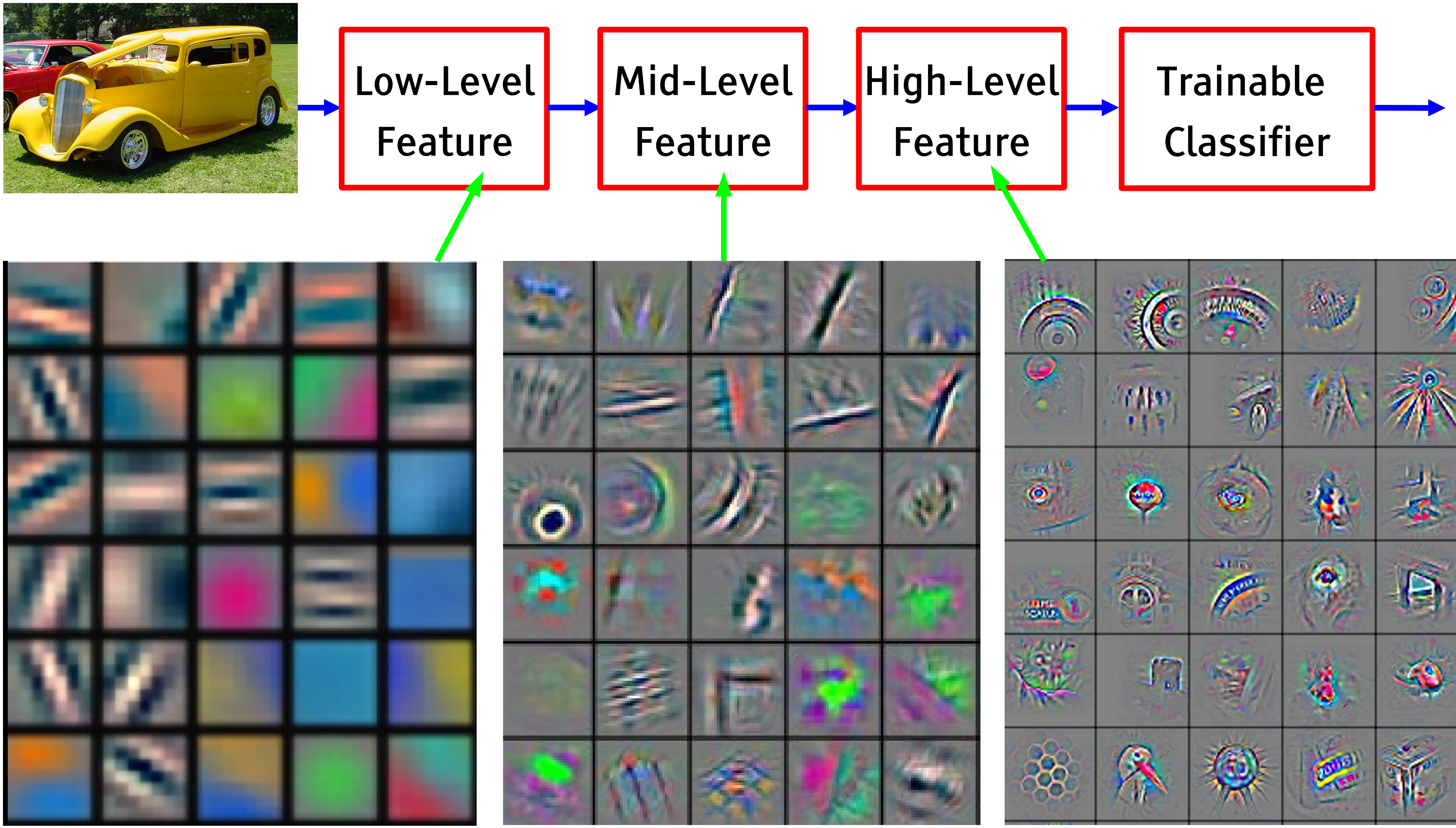
SIFT HoG → K-means Sparse Coding → Pooling → Classifier →

fixed | unsupervised | supervised

Low-level Features | Mid-level Features

Deep Learning = Learning Hierarchical Representations

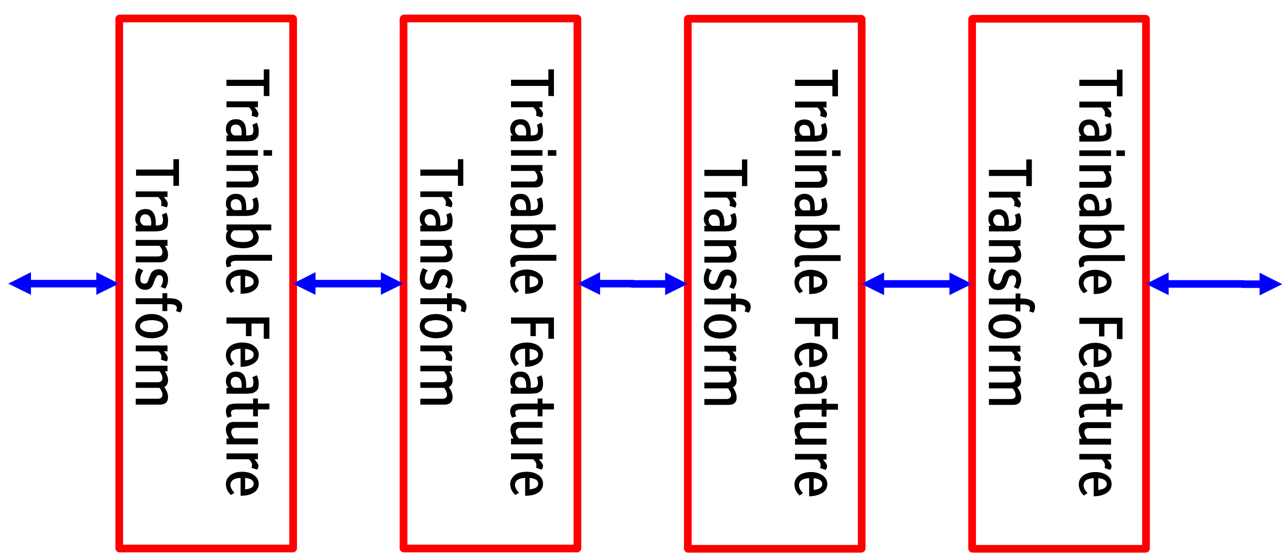
- It's deep if it has more than one stage of non-linear feature transformation



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Trainable Feature Hierarchy

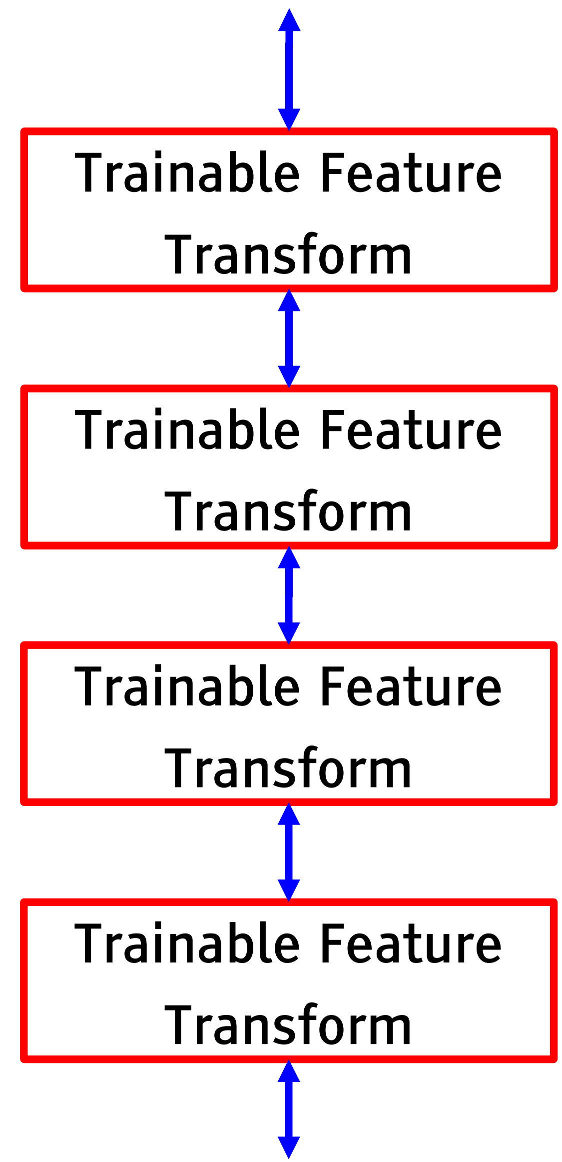
- Hierarchy of representations with increasing level of abstraction
- Each stage is a kind of trainable feature transform
- Image recognition
  - Pixel → edge → texton → motif → part → object
- Text
  - Character → word → word group → clause → sentence → story
- Speech
  - Sample → spectral band → sound → … → phone → phoneme → word →



Learning Representations: a challenge for ML, CV, AI, Neuroscience, Cognitive Science...

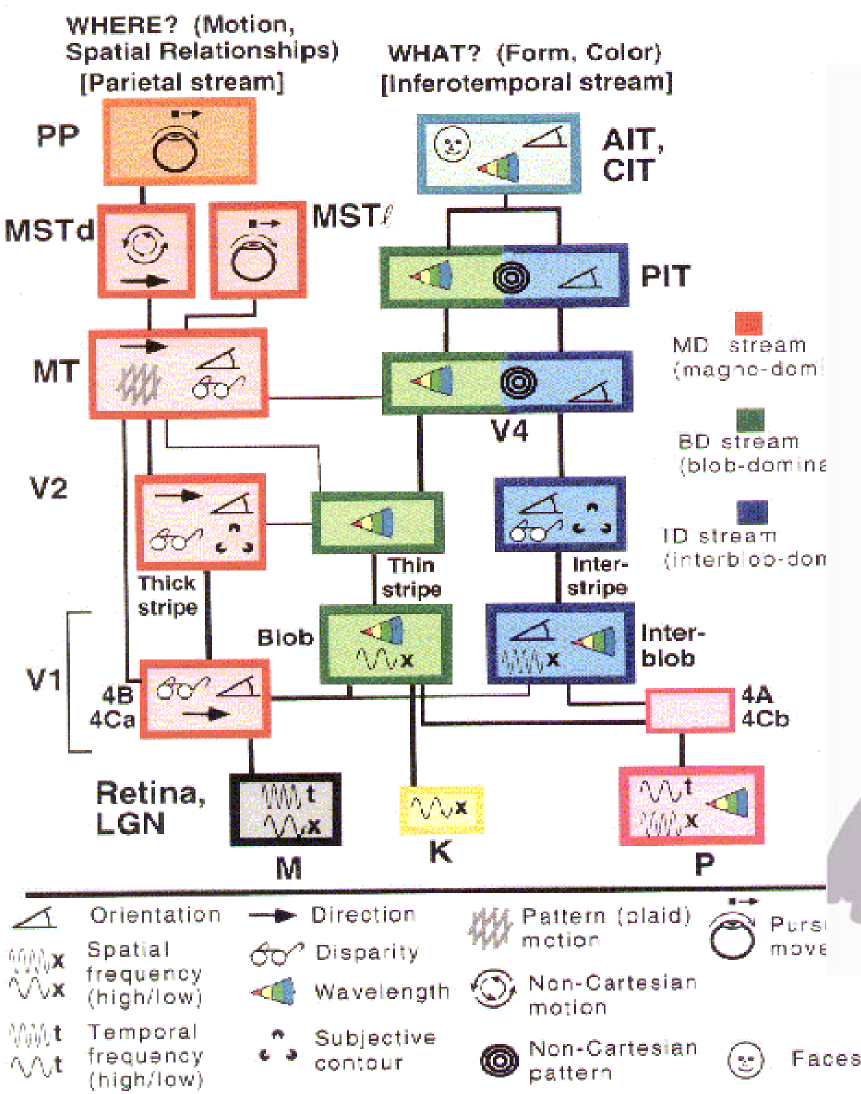
- How do we learn representations of the perceptual world?
  - How can a perceptual system build itself by looking at the world?
  - How much prior structure is necessary
- ML//CV/AI: learning features or feature hierarchies
  - What is the fundamental principle? What is the learning algorithm? What is the architecture?
- Neuroscience: how does the cortex learn perception?
  - Does the cortex "run" a single, general learning algorithm? (or a small number of them)
- Cognitive Science: how does the mind learn abstract concepts on top of less abstract ones?

- Deep Learning addresses the problem of learning hierarchical representations with a single algorithm
  - or perhaps with a few algorithms

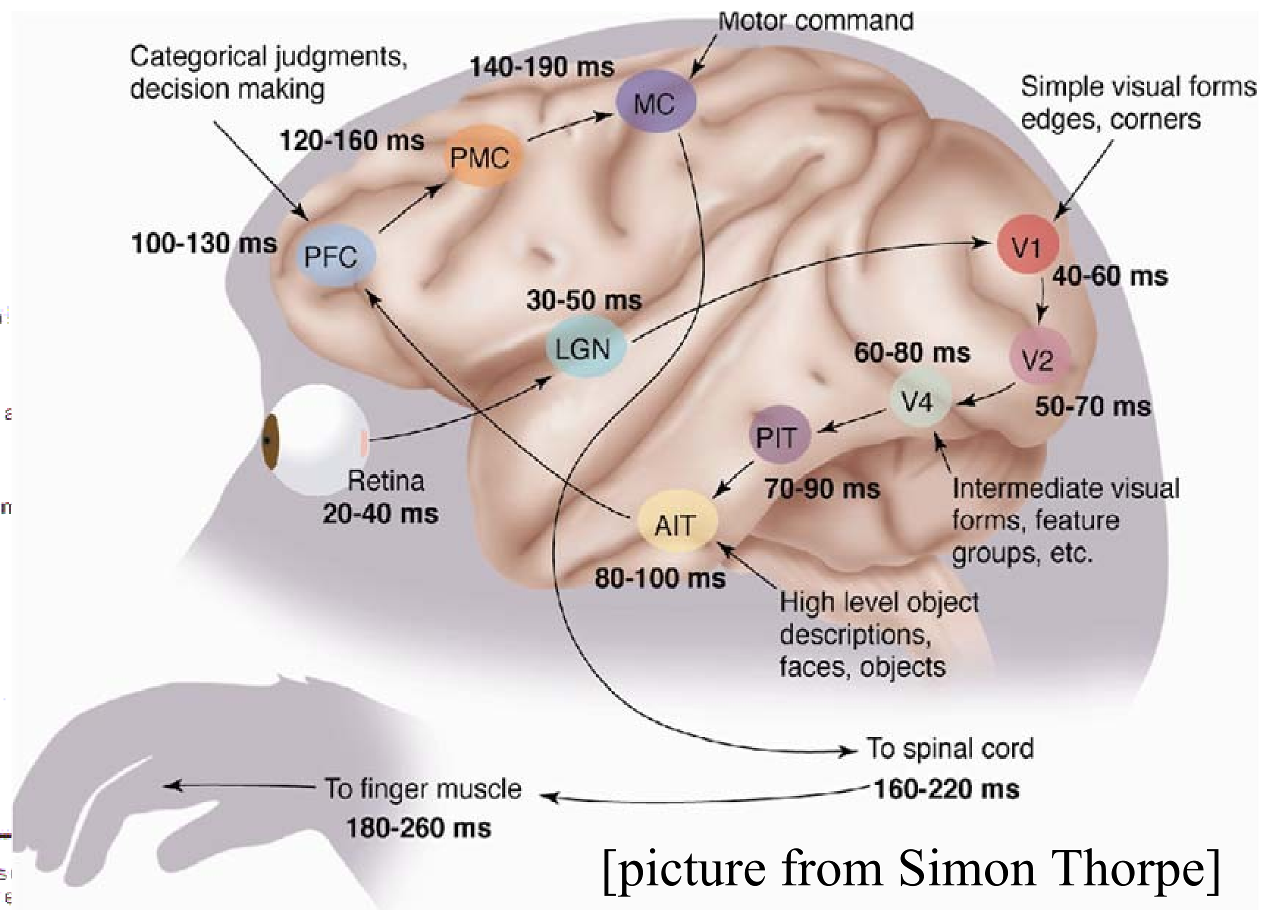


The Mammalian Visual Cortex is Hierarchical

- The ventral (recognition) pathway in the visual cortex has multiple stages
- Retina - LGN - V1 - V2 - V4 - PIT - AIT ....
- Lots of intermediate representations



[Gallant & Van Essen]



[picture from Simon Thorpe]

Trainable Feature Hierarchies: End-to-end learning

- A hierarchy of trainable feature transforms
  - Each module transforms its input representation into a higher-level one.
  - High-level features are more global and more invariant
  - Low-level features are shared among categories

Trainable Feature Transform

Trainable Feature Transform

Trainable Classifier/ Predictor

Learned Internal Representations

- How can we make all the modules trainable and get them to learn appropriate representations?

Three Types of Training Protocols

- Purely Supervised
  - Initialize parameters randomly
  - Train in supervised mode
    - typically with SGD, using backprop to compute gradients
  - Used in most practical systems for speech and image recognition
- Unsupervised, layerwise + supervised classifier on top
  - Train each layer unsupervised, one after the other
  - Train a supervised classifier on top, keeping the other layers fixed
  - Good when very few labeled samples are available
- Unsupervised, layerwise + global supervised fine-tuning
  - Train each layer unsupervised, one after the other
  - Add a classifier layer, and retrain the whole thing supervised
  - Good when label set is poor (e.g. pedestrian detection)
- Unsupervised pre-training often uses regularized auto-encoders

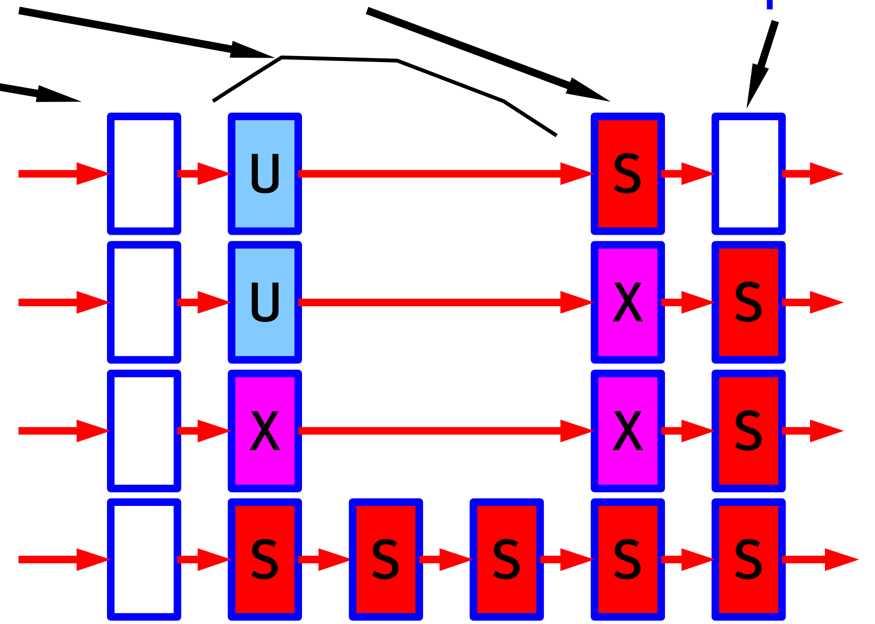
Deep Learning and Feature Learning Today

- Deep Learning has been the hottest topic in speech recognition in the last 2 years
  - A few long-standing performance records were broken with deep learning methods
  - Microsoft and Google have both deployed DL-based speech recognition system in their products
  - Microsoft, Google, IBM, Nuance, AT&T, and all the major academic and industrial players in speech recognition have projects on deep learning
- Deep Learning is the hottest topic in Computer Vision
  - Feature engineering is the bread-and-butter of a large portion of the CV community, which creates some resistance to feature learning
  - But the record holders on ImageNet and Semantic Segmentation are convolutional nets
- Deep Learning is becoming hot in Natural Language Processing
- Deep Learning/Feature Learning in Applied Mathematics
  - The connection with Applied Math is through sparse coding, non-convex optimization, stochastic gradient algorithms, etc...

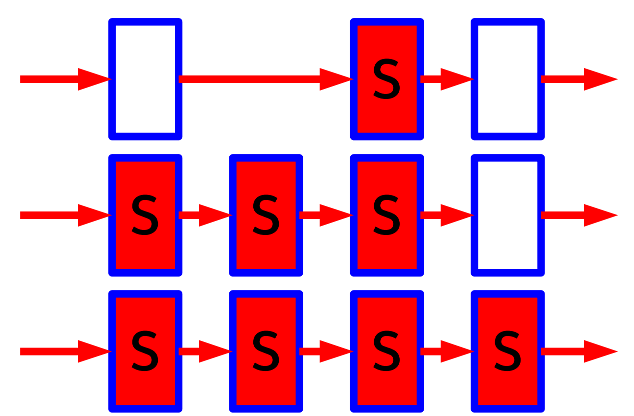
In Several Fields, Feature Learning Has Caused Revolutions: Speech Recognition, Handwriting Recogntiion

- U= unsupervised, S=supervised, X=unsupervised+supervised
- Low-level feat. → mid-level feat. → classifier → contextual post-proc

- Speech Recognition
  - Early 1980s: DTW
  - Late 1980s: GMM
  - 1990s: discriminative GMM
  - 2010: deep neural nets

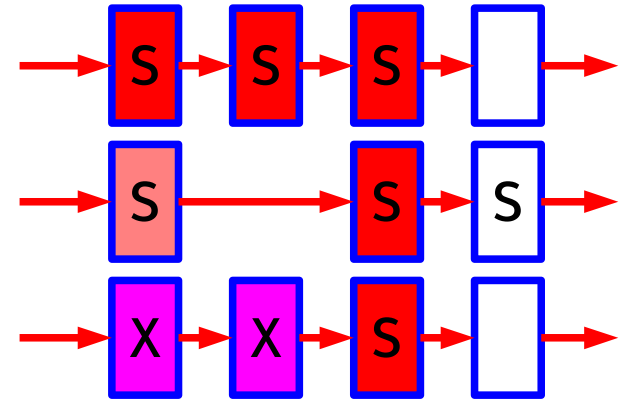


- Handwriting Recognition and OCR
  - Early 80's: features+classifier
  - Late 80's: supervised convnet
  - Mid 90's: convnet+CRF

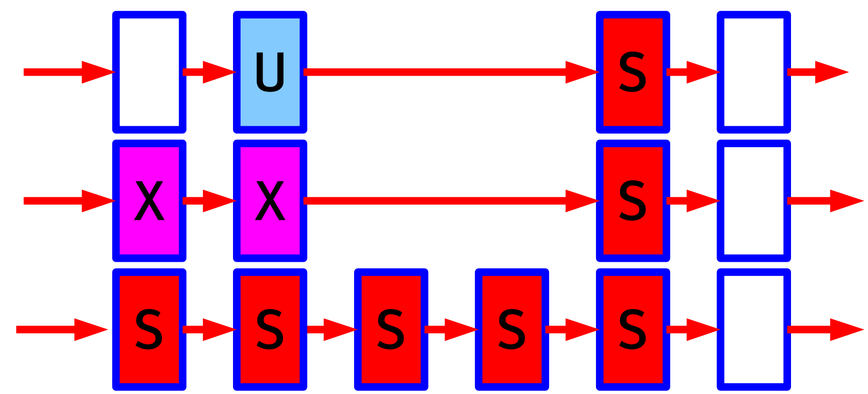


In Several Fields, Feature Learning Has Caused Revolutions: Object Detection, Object Recognition, Scene Labeling

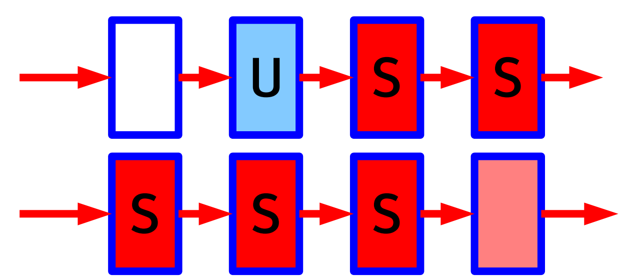
- **Face & People Detection (1993-now)**
  - Supervised ConvNet on pixels (93, 94, 05, 07)
  - Selected Haar features + Adaboost (2001)
  - Unsup+Sup ConvNet on raw pixels (2011)



- **Object Recognition**
  - SIFT/HoG+sparse+SVM (2005, 2006)
  - unsup+sup convnet (2009, 2010)
  - supervised convnet (2012)

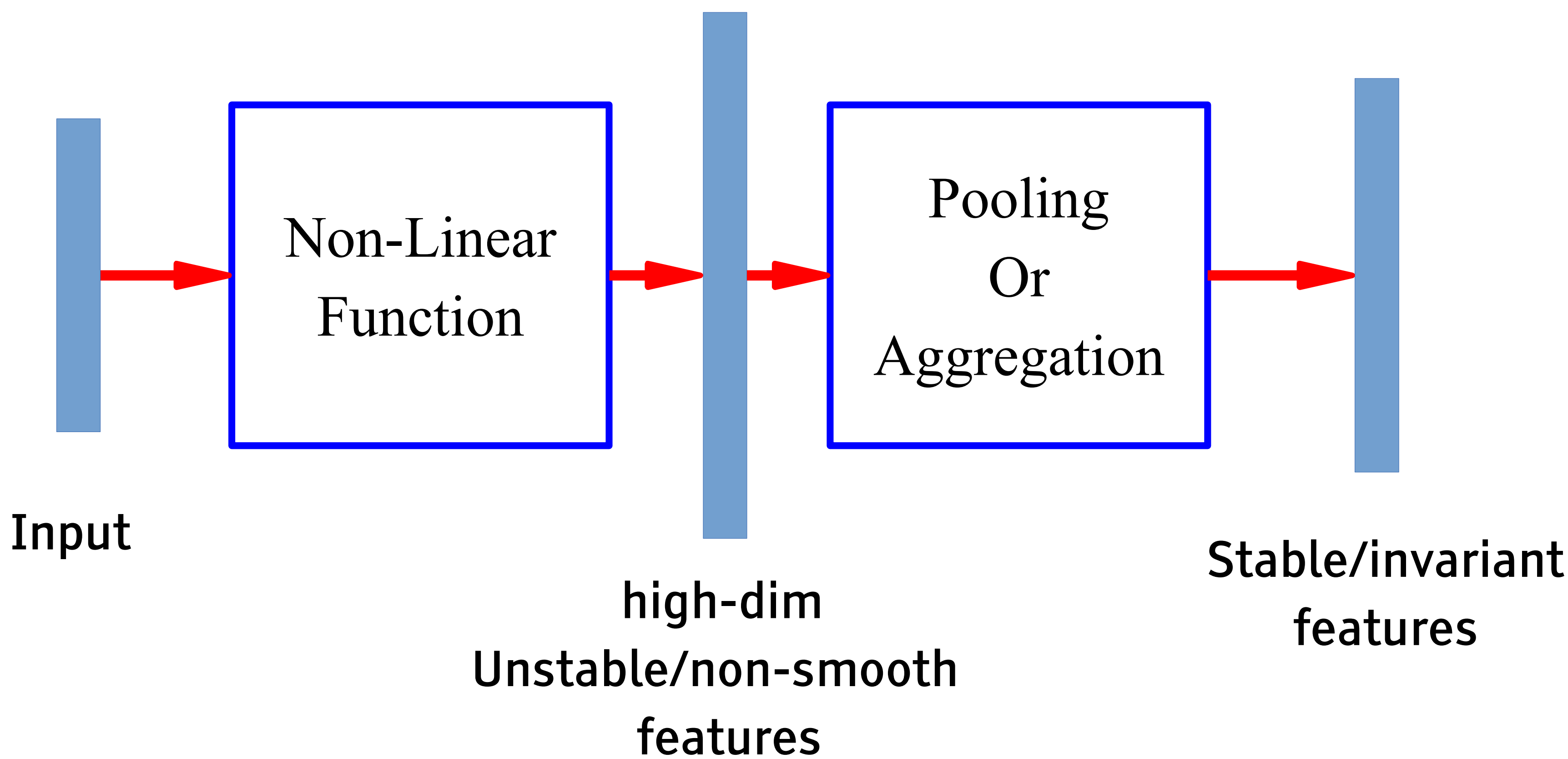


- **Semantic Segmentation / scene labeling**
  - unsup mid-lvl, CRF (2009, 10, 11, 12)
  - supervised convnet (2008, 12, 13)



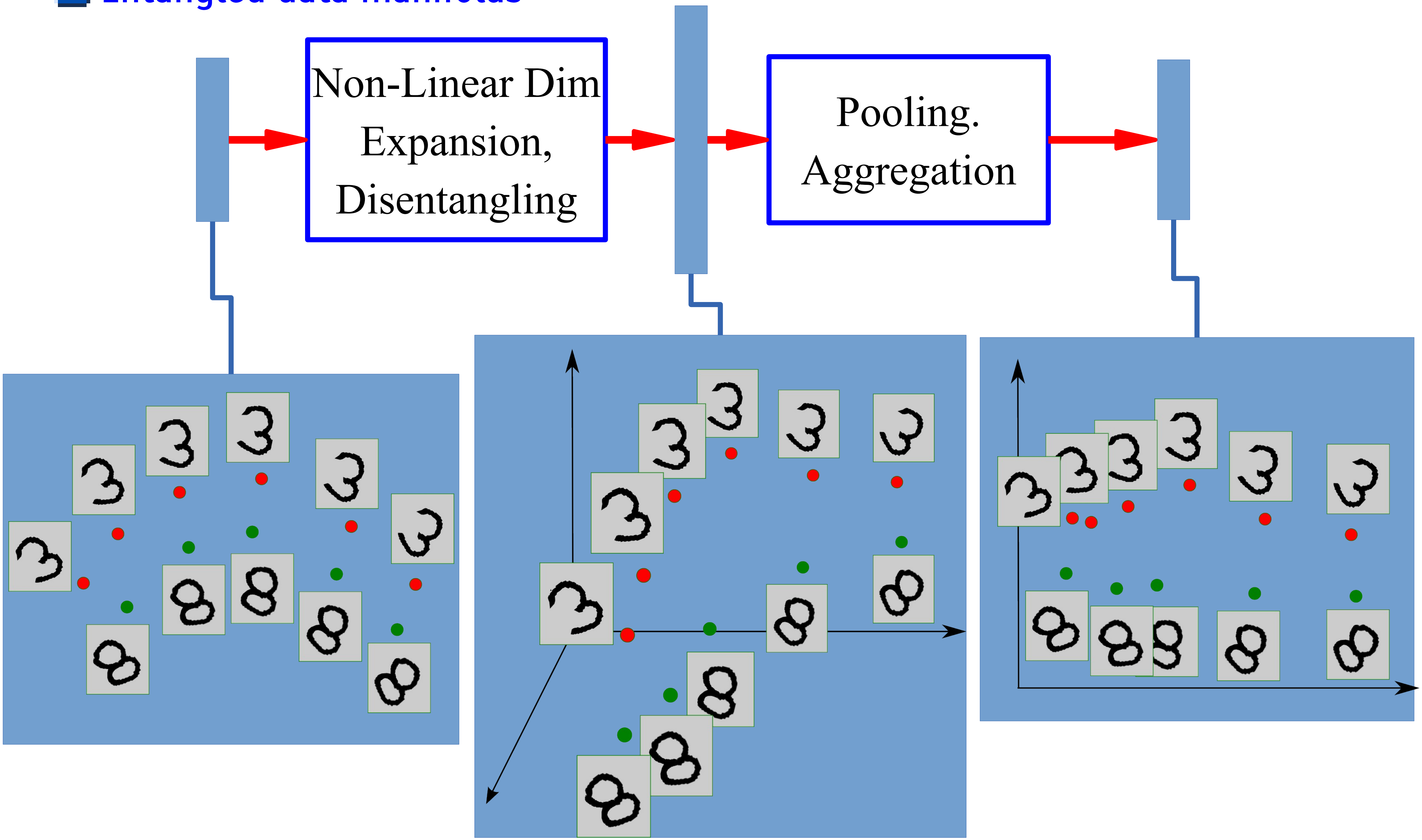
Basic Modules for Feature Learning

- Embed the input non-linearly into a high(er) dimensional space
  - In the new space, things that were non separable may become separable
- Pool regions of the new space together
  - Bringing together things that are semantically similar. Like pooling.



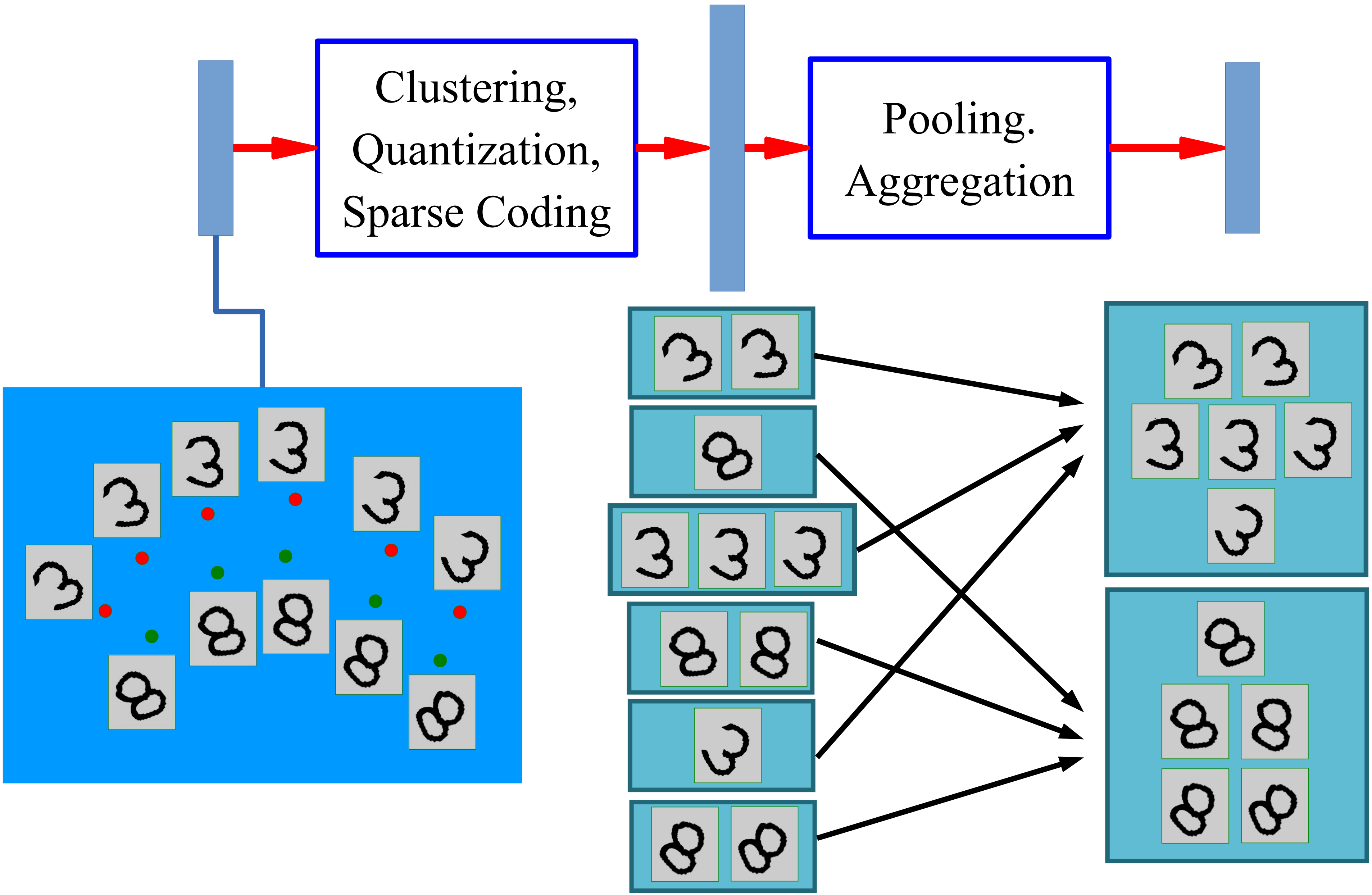
Non-Linear Expansion → Pooling

- Entangled data manifolds



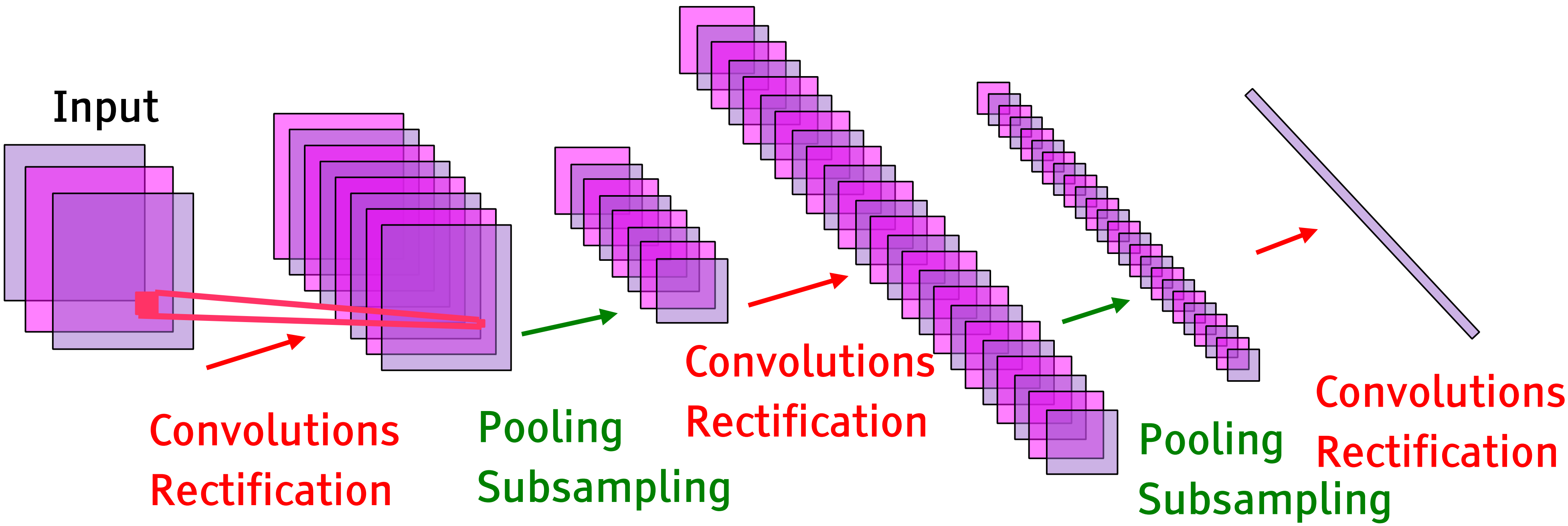
Sparse Non-Linear Expansion → Pooling

- Use clustering to break things apart, pool together similar things



Convolutional Networks

Convolutional Network (ConvNet)



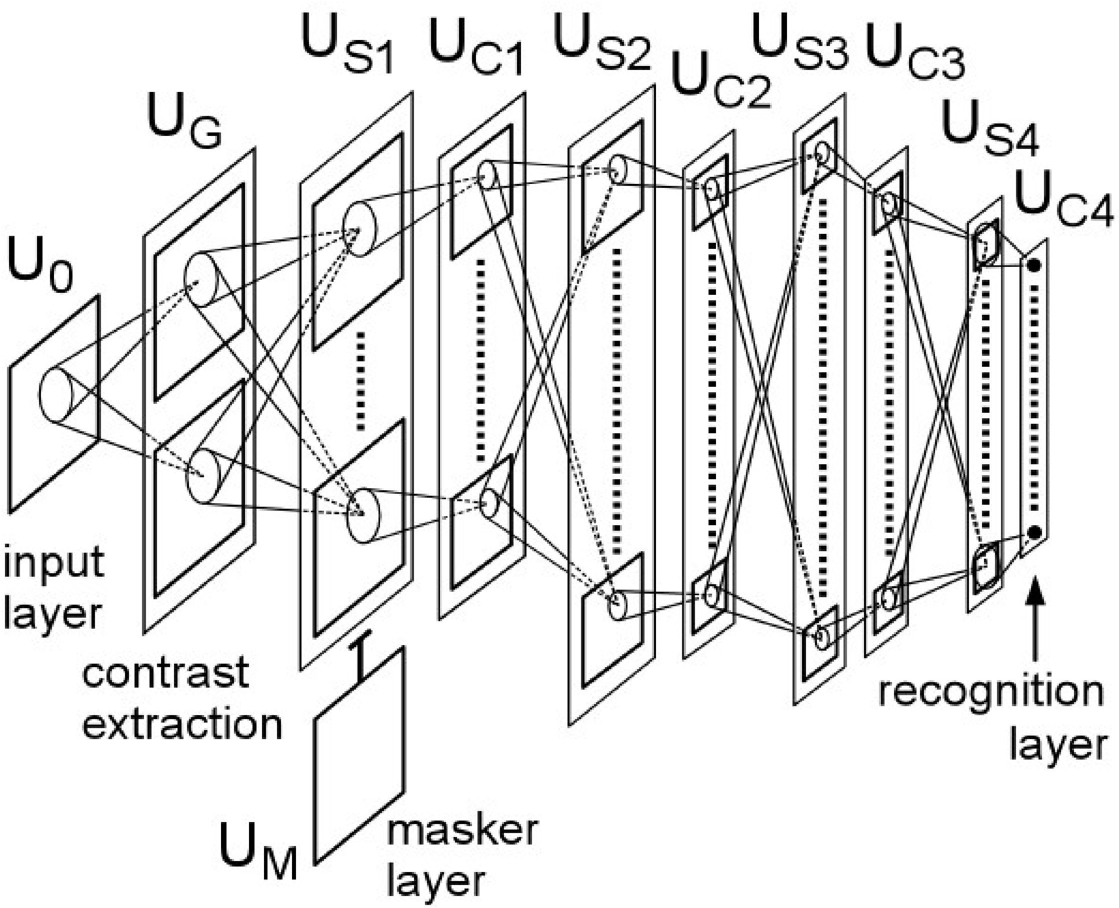
- Non-Linearity: half-wave rectification: out = max(0, in)
- Pooling: max, L2 norm, Lp norm....
- Training:
  - Supervised (1988-2006),
  - Unsupervised+Supervised (2006-now)

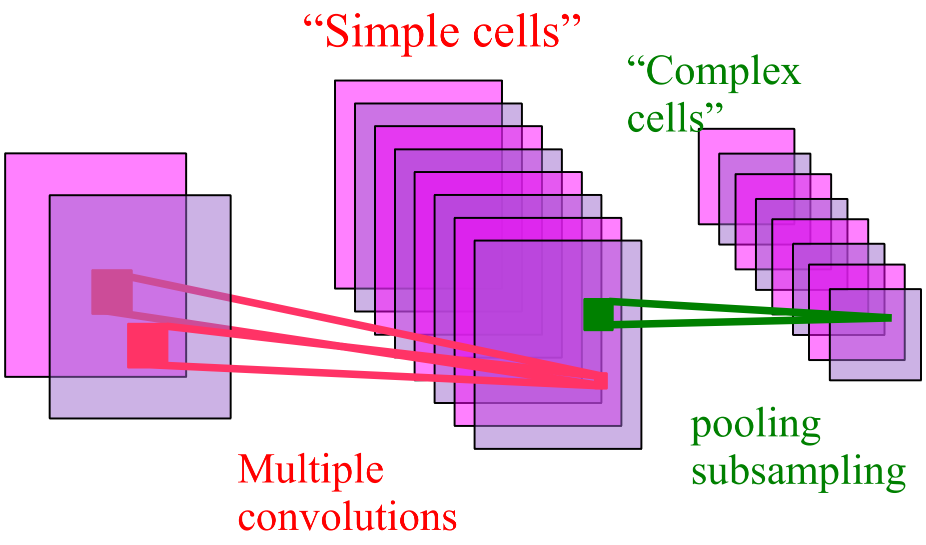
Convolutional Nets

- Are deployed in many commercial applications
  - Check reading: AT&T 1996
  - Handwriting recognition: Microsoft early 2000
  - Face and person detection: NEC 2005
  - Gender and age recognition: NEC 2010
  - Photo tagging: Google and Baidu 2013
- Have won several competitions
  - ImageNet LSVRC, Kaggle Facial Expression, Kaggle Multimodal Learning, German Traffic Signs, Connectomics, Handwriting....
- Are applicable to array data where nearby values are correlated
  - Images, sound, time-frequency representations, video,
  - volumetric images, RGB-Depth images,.....
- One of the few deep models that can be trained purely supervised

Early Hierarchical Feature Models for Vision

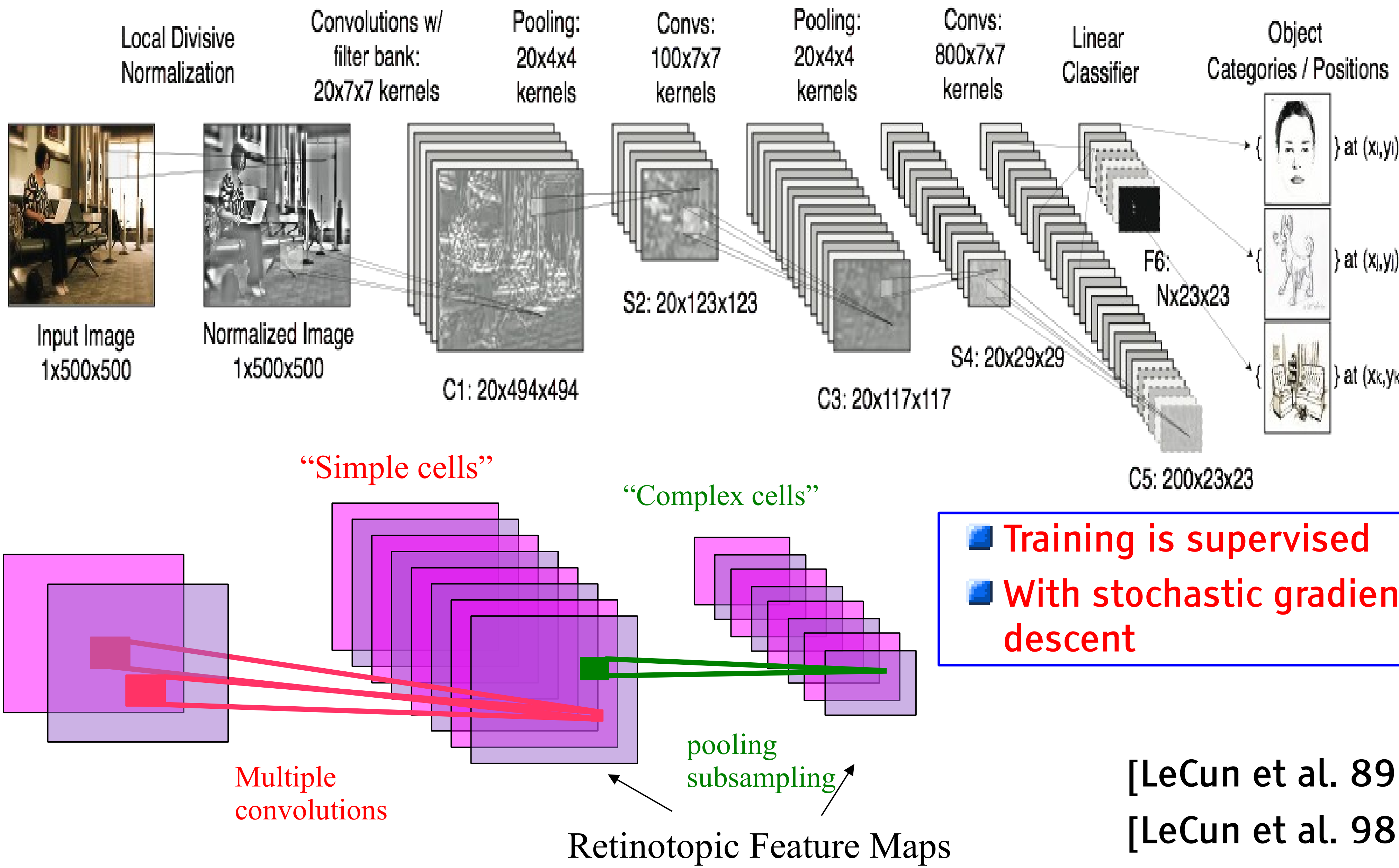
- [Hubel & Wiesel 1962]:
  - simple cells detect local features
  - complex cells "pool" the outputs of simple cells within a retinotopic neighborhood.





Cognitron & Neocognitron [Fukushima 1974-1982]

The Convolutional Net Model
(Multistage Hubel-Wiesel system)

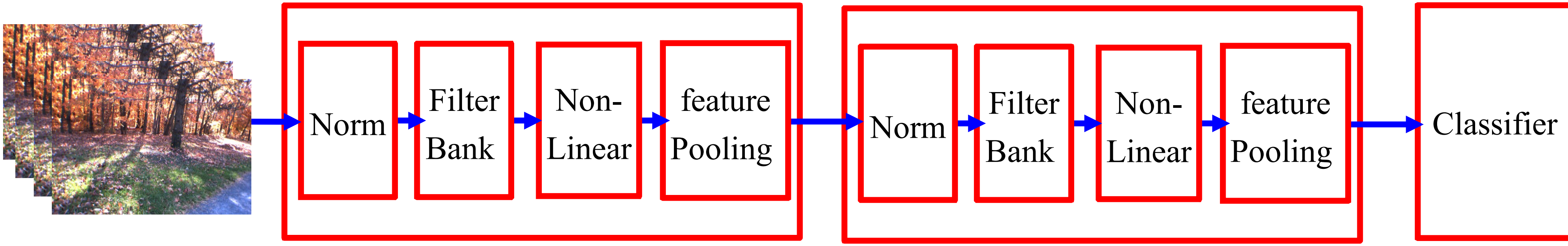


- Training is supervised
- With stochastic gradient descent

[LeCun et al. 89]

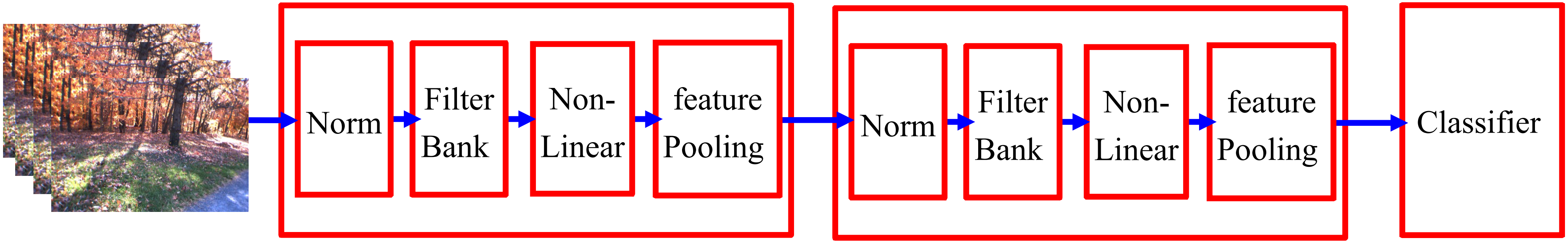
[LeCun et al. 98]

Feature Transform: Normalization → Filter Bank → Non-Linearity → Pooling



- Stacking multiple stages of
  - [Normalization → Filter Bank → Non-Linearity → Pooling].
- Normalization: variations on whitening
  - Subtractive: average removal, high pass filtering
  - Divisive: local contrast normalization, variance normalization
- Filter Bank: dimension expansion, projection on overcomplete basis
- Non-Linearity: sparsification, saturation, lateral inhibition....
  - Rectification, Component-wise shrinkage, tanh, winner-takes-all
- Pooling: aggregation over space or feature type, subsampling
  - $X_i; \quad L_p: \sqrt[p]{X_i^p}; \quad PROB: \frac{1}{b}\log\left(\sum_i e^{bX_i}\right)$

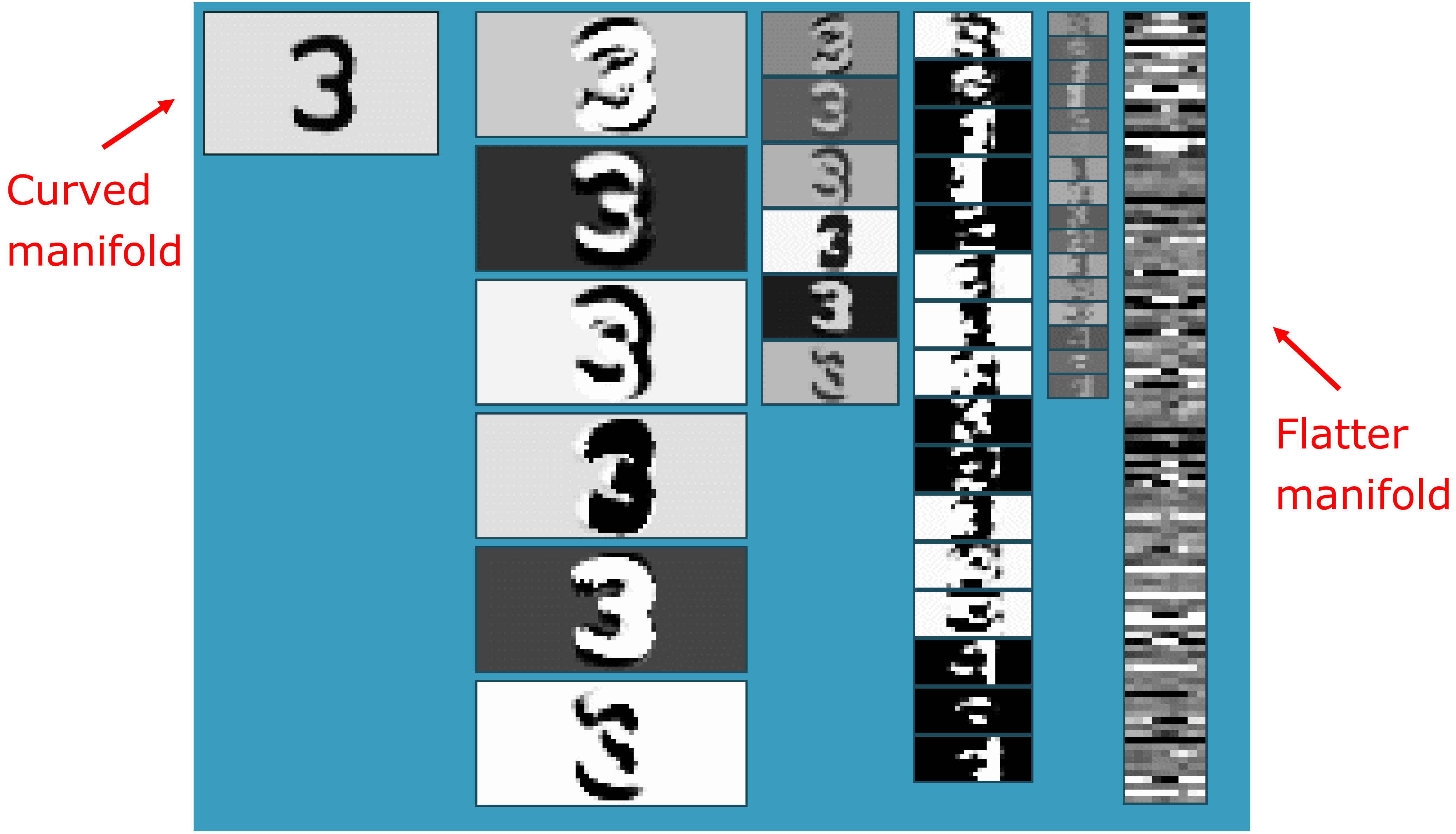
Feature Transform: Normalization → Filter Bank → Non-Linearity → Pooling



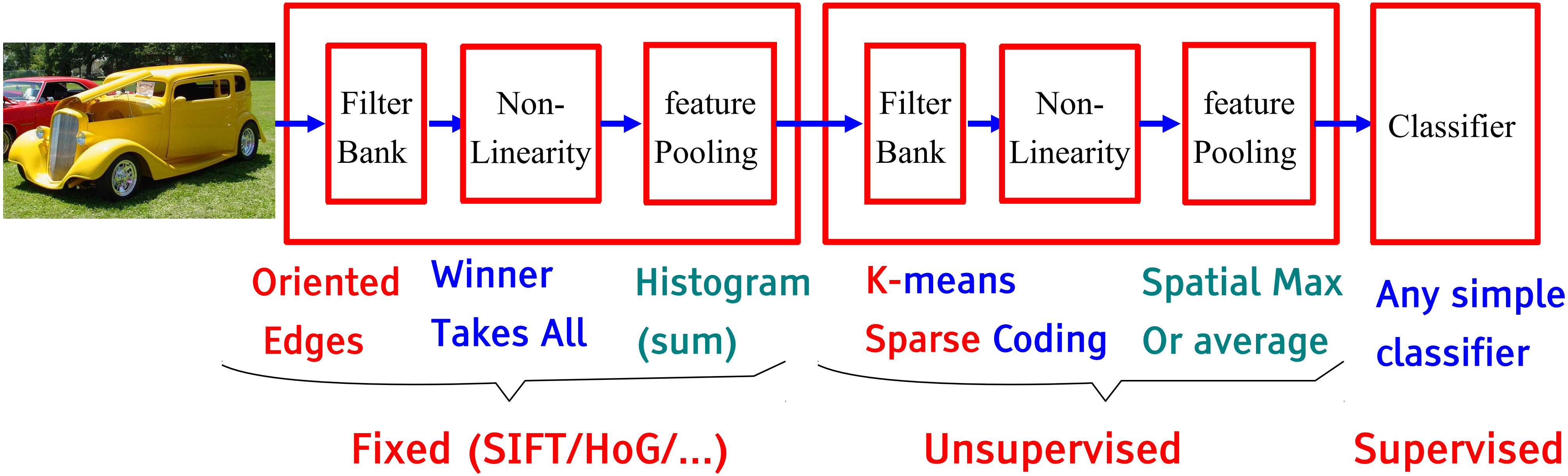
- Filter Bank → Non-Linearity = Non-linear embedding in high dimension
- Feature Pooling = contraction, dimensionality reduction, smoothing
- Learning the filter banks at every stage
- Creating a hierarchy of features
- Basic elements are inspired by models of the visual (and auditory) cortex
  - Simple Cell + Complex Cell model of [Hubel and Wiesel 1962]
  - Many “traditional” feature extraction methods are based on this
  - SIFT, GIST, HoG, SURF...
- [Fukushima 1974-1982], [LeCun 1988-now],
  - since the mid 2000: Hinton, Seung, Poggio, Ng,....

Convolutional Network (vintage 1990)

- filters → tanh → average-tanh → filters → tanh → average-tanh → filters → tanh



"Mainstream" object recognition pipeline 2006-2012 is not very different from ConvNets



- Fixed Features + unsupervised mid-level features + simple classifier
  - SIFT + Vector Quantization + Pyramid pooling + SVM
    - [Lazebnik et al. CVPR 2006]
  - SIFT + Local Sparse Coding Macrofeatures + Pyramid pooling + SVM
    - [Boureau et al. ICCV 2011]
  - SIFT + Fisher Vectors + Deformable Parts Pooling + SVM
    - [Perronin et al. 2012]

Tasks for Which Deep Convolutional Nets are the Best

- Handwriting recognition MNIST (many), Arabic HWX (IDSIA)
- OCR in the Wild [2011]: StreetView House Numbers (NYU and others)
- Traffic sign recognition [2011] GTSRB competition (IDSIA, NYU)
- Pedestrian Detection [2013]: INRIA datasets and others (NYU)
- Volumetric brain image segmentation [2009] connectomics (IDSIA, MIT)
- Human Action Recognition [2011] Hollywood II dataset (Stanford)
- Object Recognition [2012] ImageNet competition
- Scene Parsing [2012] Stanford bgd, SiftFlow, Barcelona (NYU)
- Scene parsing from depth images [2013] NYU RGB-D dataset (NYU)
- Speech Recognition [2012] Acoustic modeling (IBM and Google)
- Breast cancer cell mitosis detection [2011] MITOS (IDSIA)

- The list of perceptual tasks for which ConvNets hold the record is growing.
- Most of these tasks (but not all) use purely supervised convnets.

Simple ConvNet Applications with State-of-the-Art Performance

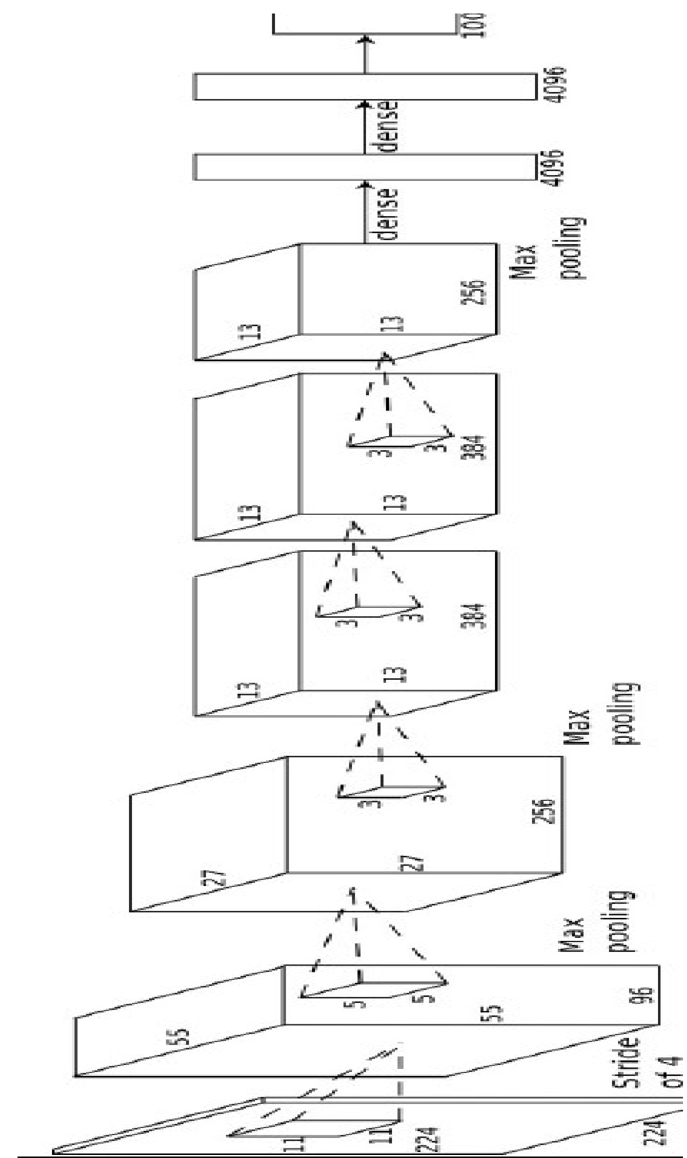
- **Traffic Sign Recognition (GTSRB)**
  - German Traffic Sign Reco Bench
  - 99.2% accuracy

- **House Number Recognition (Google)**
  - Street View House Numbers
  - 94.3 % accuracy

Object Recognition [Krizhevsky, Sutskever, Hinton 2012]

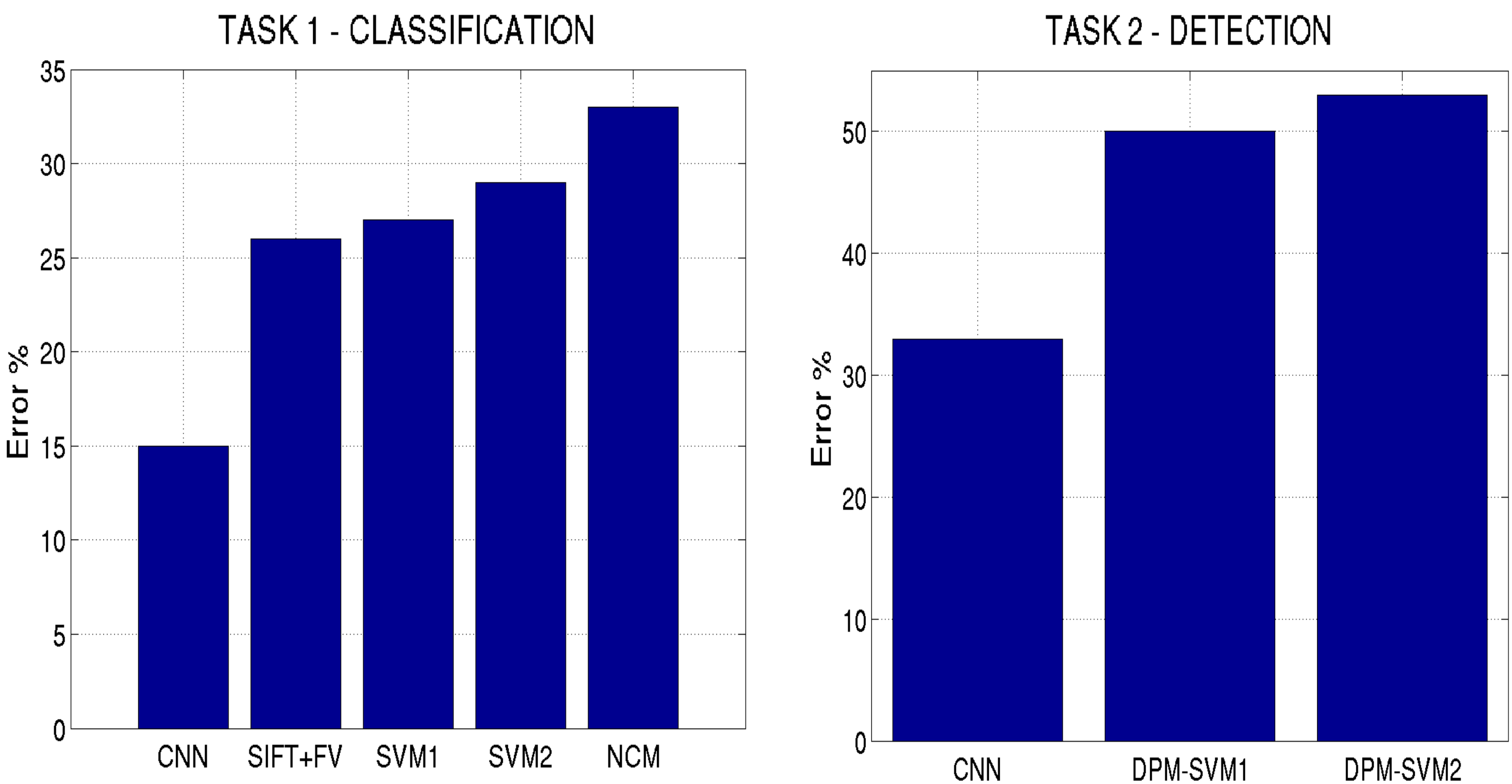
- Won the 2012 ImageNet LSVRC. 60 Million parameters, 832M MAC ops

4M	FULL CONNECT	4Mflop
16M	FULL 4096/ReLU	16M
37M	FULL 4096/ReLU	37M
	MAX POOLING	
442K	CONV 3x3/ReLU 256fm	74M
1.3M	CONV 3x3ReLU 384fm	224M
884K	CONV 3x3/ReLU 384fm	149M
	MAX POOLING 2x2sub	
	LOCAL CONTRAST NORM	
307K	CONV 11x11/ReLU 256fm	223M
	MAX POOL 2x2sub	
	LOCAL CONTRAST NORM	
35K	CONV 11x11/ReLU 96fm	105M



Object Recognition: ILSVRC 2012 results

- ImageNet Large Scale Visual Recognition Challenge
- 1000 categories, 1.5 Million labeled training samples



Object Recognition [Krizhevsky, Sutskever, Hinton 2012]

- Method: large convolutional net
  - 650K neurons, 832M synapses, 60M parameters
  - Trained with backprop on GPU
  - Trained “with all the tricks Yann came up with in the last 20 years, plus dropout” (Hinton, NIPS 2012)
  - Rectification, contrast normalization,...
- Error rate: 15% (whenever correct class isn't in top 5)
- Previous state of the art: 25% error

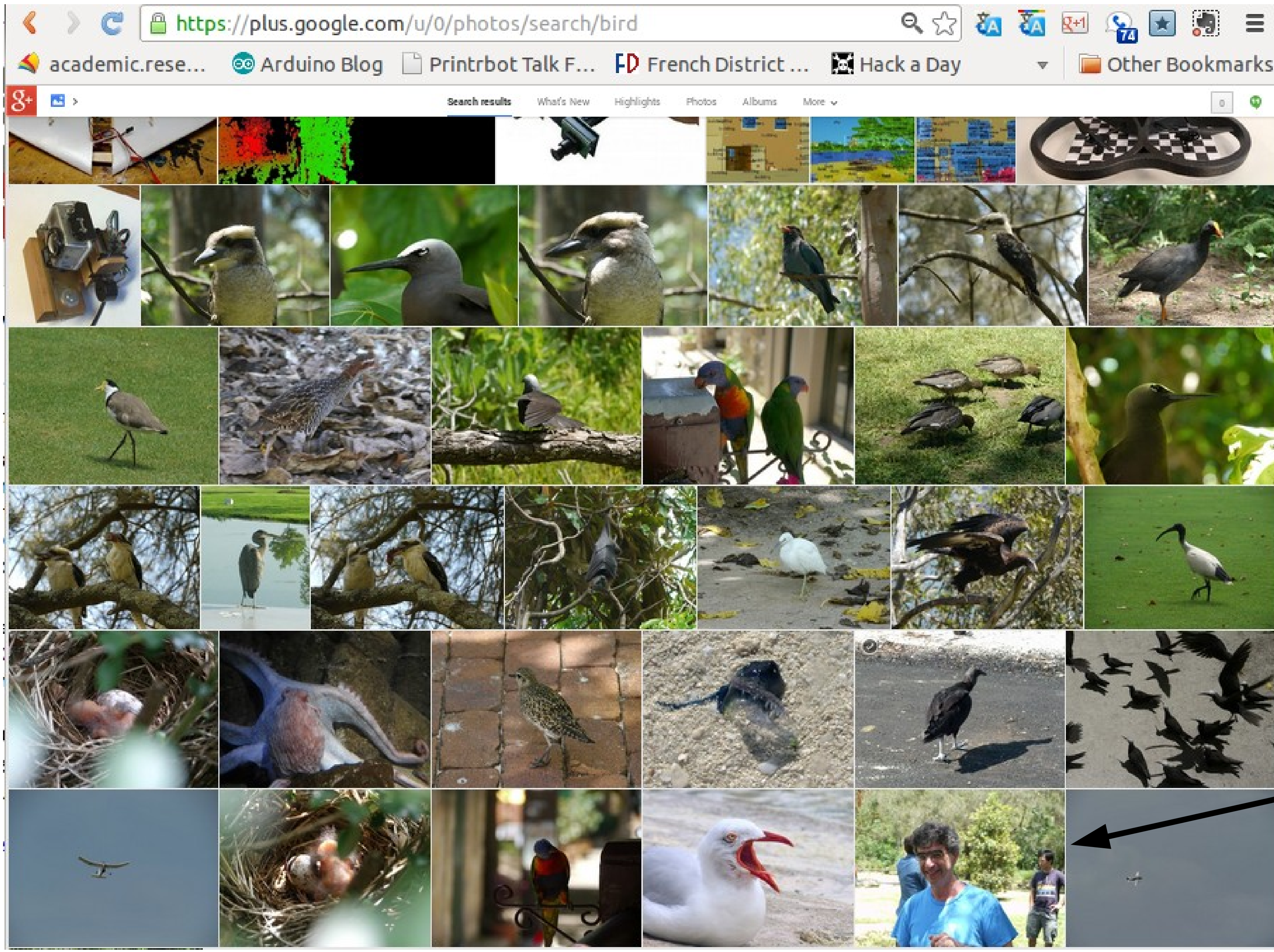
- Has changed many people's opinion of ConvNets in the vision community.

- Acquired by Google in Jan 2013
- Deployed in Google+ Photo Tagging in May 2013

Object Recognition [Krizhevski, Sutskever, Hinton 2012]

ConvNet-Based Google+ Photo Tagger

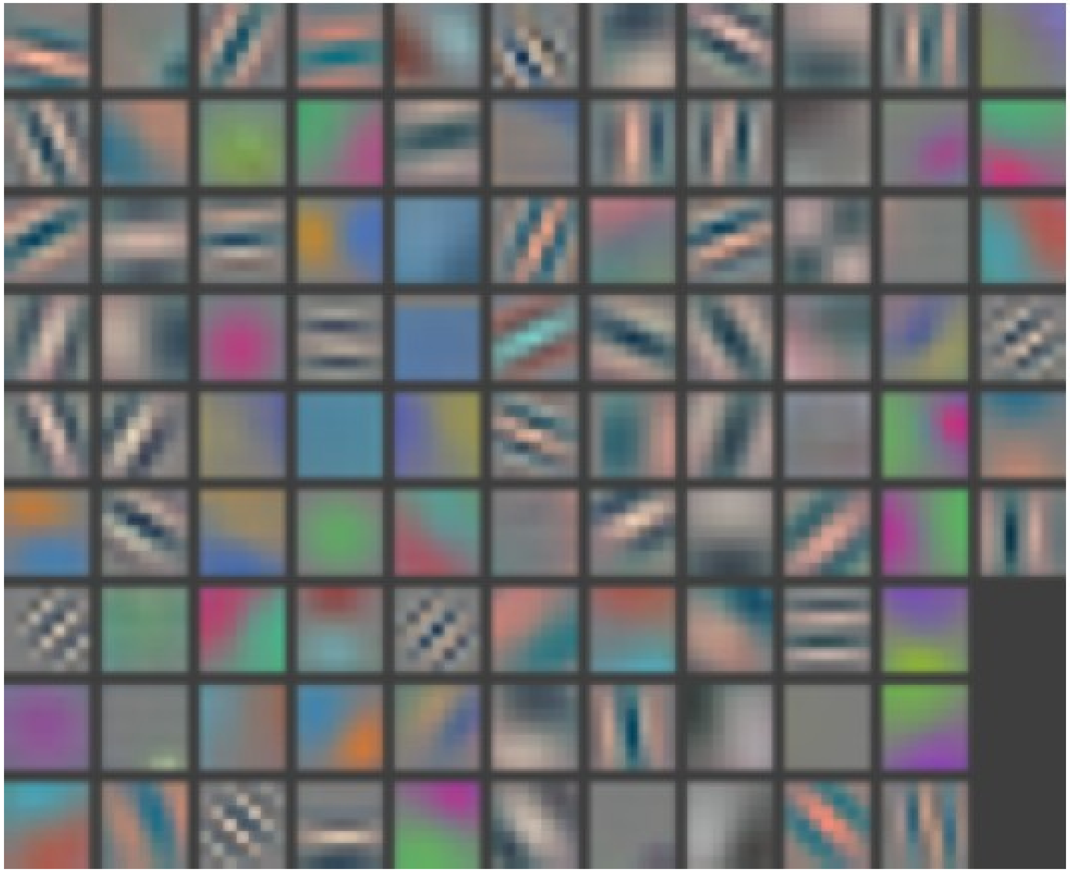
- Searched my personal collection for "bird"



Samy
Bengio
???

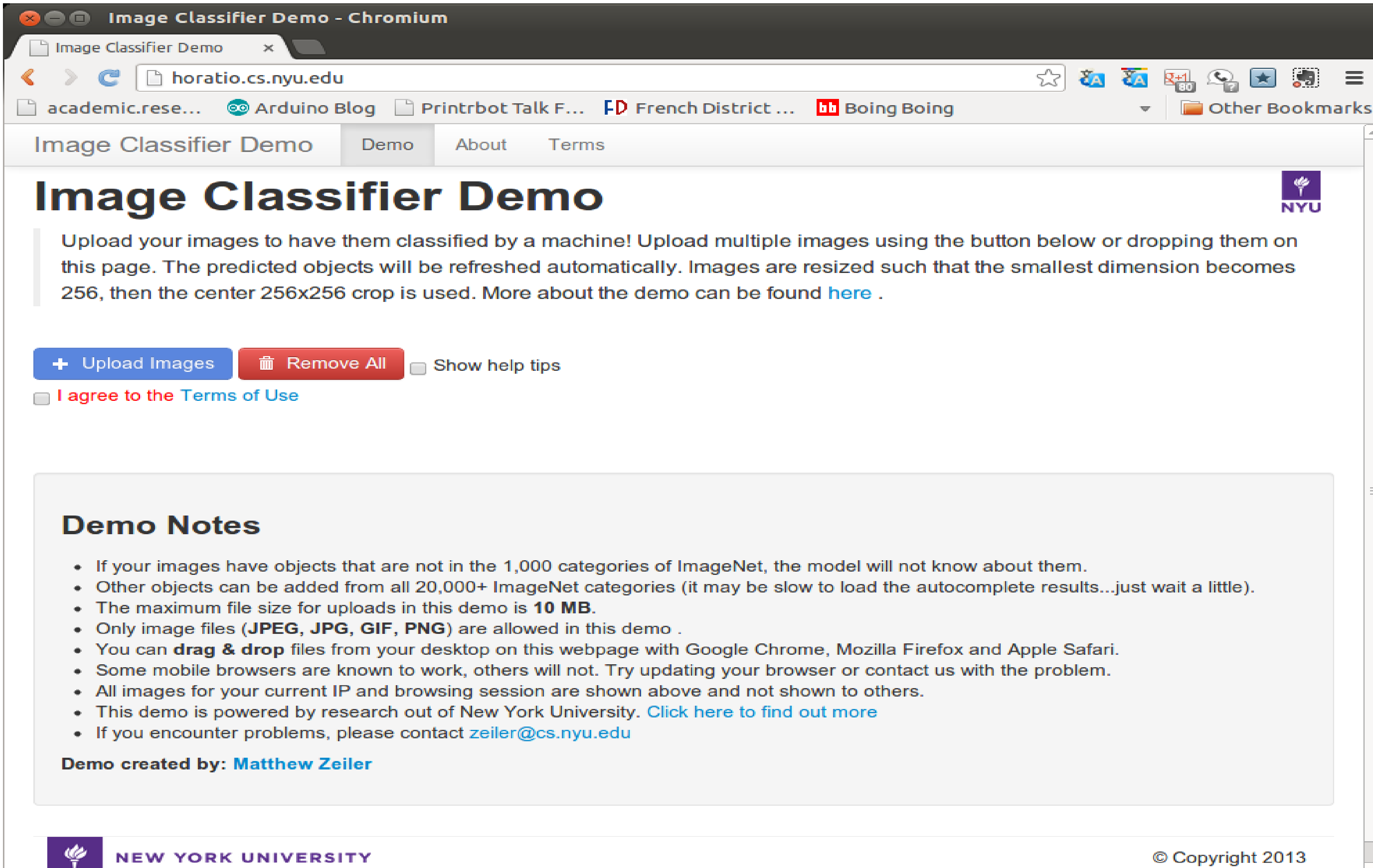
Another ImageNet-trained ConvNet [Zeiler & Fergus 2013]

- Convolutional Net with 8 layers, input is 224x224 pixels
  - conv-pool-conv-pool-conv-conv-conv-full-full-full
  - Rectified-Linear Units (ReLU): y = max(0,x)
  - Divisive contrast normalization across features [Jarrett et al. ICCV 2009]
- Trained on ImageNet 2012 training set
  - 1.3M images, 1000 classes
  - 10 different crops/flips per image
- Regularization: Dropout
  - [Hinton 2012]
  - zeroing random subsets of units
- Stochastic gradient descent
  - for 70 epochs (7-10 days)
  - With learning rate annealing



Object Recognition on-line demo [Zeiler & Fergus 2013]

http://horatio.cs.nyu.edu

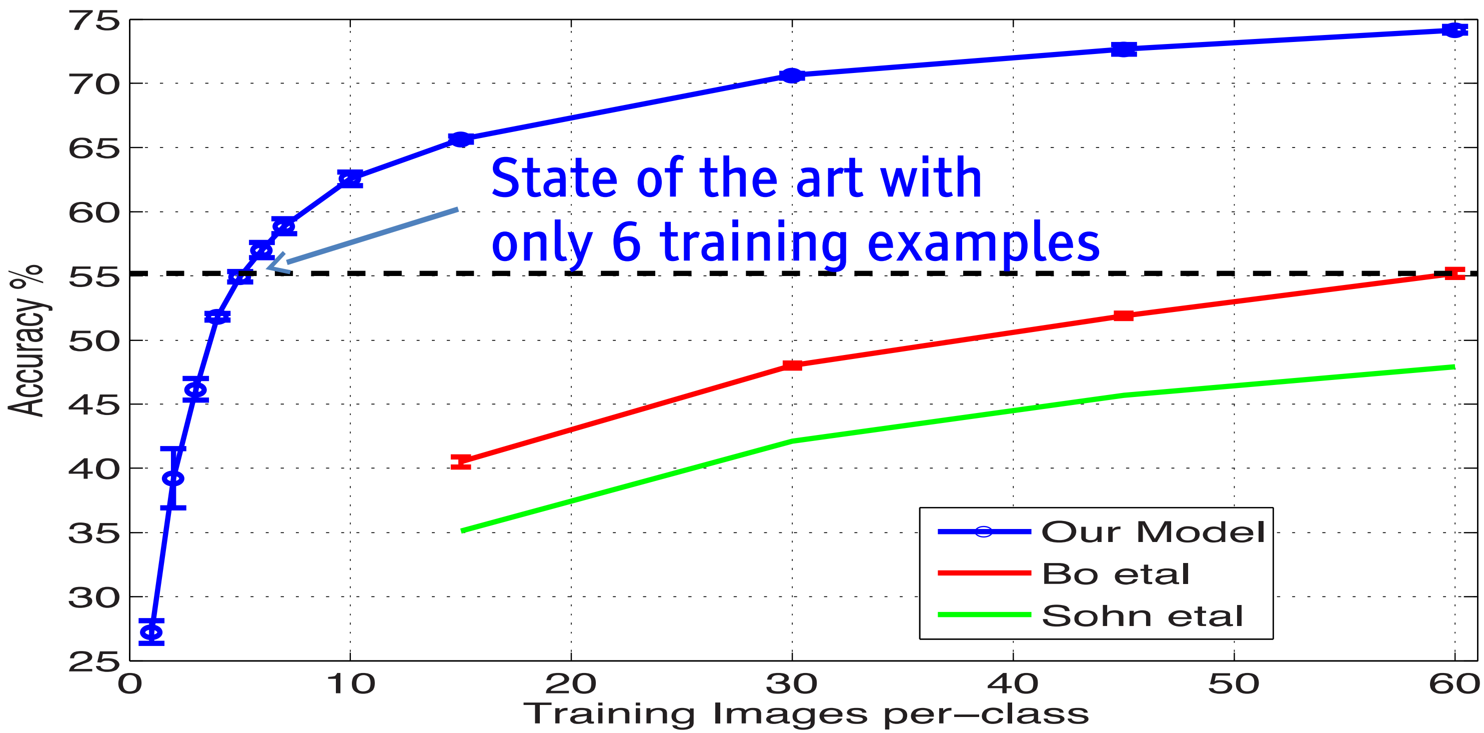


ConvNet trained ImageNet [Zeiler & Fergus 2013]

Error %	Val Top-1	Val Top-5	Test Top-5
Deng *et al.* SIFT + FV [7]	— —	— —	26.2
Krizhevsky *et al.* [12], 1 convnet	40.7	18.2	— —
Krizhevsky *et al.* [12], 5 convnets	38.1	16.4	16.4
*Krizhevsky *et al.* [12], 1 convnets	39.0	16.6	— —
*Krizhevsky *et al.* [12], 7 convnets	36.7	15.4	15.3
Our replication of [12], 1 convnet	41.7	19.0	— —
1 convnet - our model	38.4 ± 0.05	16.5 ± 0.05	— —
5 convnets - our model (a)	36.7	15.3	15.3
1 convnet - tweaked model (b)	37.5	16.0	16.1
6 convnets, (a) & (b) combined	**36.0**	**14.7**	**14.8**

Features are generic: Caltech 256

- Network first trained on ImageNet.
- Last layer chopped off
- Last layer trained on Caltech 256,
- first layers N-1 kept fixed.



# Train	Acc % 15/class	Acc % 30/class	Acc % 45/class	Acc % 60/class
Sohn *et al.* [16]	35.1	42.1	45.7	47.9
Bo *et al.* [3]	40.5 ± 0.4	48.0 ± 0.2	51.9 ± 0.2	55.2 ± 0.3
Non-pretr.	9.0 ± 1.4	22.5 ± 0.7	31.2 ± 0.5	38.8 ± 1.4
ImageNet-pretr.	**65.7 ± 0.2**	**70.6 ± 0.2**	**72.7 ± 0.4**	**74.2 ± 0.3**

3: [Bo, Ren, Fox. CVPR, 2013] 16: [Sohn, Jung, Lee, Hero ICCV 2011]

Features are generic: PASCAL VOC 2012

- Network first trained on ImageNet.
- Last layer trained on Pascal VOC, keeping N-1 first layers fixed.

Acc %	[15]	[19]	Ours	Acc %	[15]	[19]	Ours
Airplane	92.0	**97.3**	96.0	Dining table	63.2	**77.8**	67.7
Bicycle	74.2	**84.2**	77.1	Dog	68.9	83.0	**87.8**
Bird	73.0	80.8	**88.4**	Horse	78.2	**87.5**	86.0
Boat	77.5	85.3	**85.5**	Motorbike	81.0	**90.1**	85.1
Bottle	54.3	**60.8**	55.8	Person	91.6	**95.0**	90.9
Bus	85.2	**89.9**	85.8	Potted plant	55.9	**57.8**	52.2
Car	81.9	**86.8**	78.6	Sheep	69.4	79.2	**83.6**
Cat	76.4	89.3	**91.2**	Sofa	65.4	**73.4**	61.1
Chair	65.2	**75.4**	65.0	Train	86.7	**94.5**	91.8
Cow	63.2	**77.8**	74.4	Tv/monitor	77.4	**80.7**	76.1
Mean	74.3	**82.2**	79.0	# won	0	**15**	5

[15] K. Sande, J. Uijlings, C. Snoek, and A. Smeulders. Hybrid coding for selective search. In PASCAL VOC Classification Challenge 2012,

[19] S. Yan, J. Dong, Q. Chen, Z. Song, Y. Pan, W. Xia, Z. Huang, Y. Hua, and S. Shen. Generalized hierarchical matching for sub-category aware object classification. In PASCAL VOC Classification Challenge 2012

Building a ConvNet Model: Example in Torch7

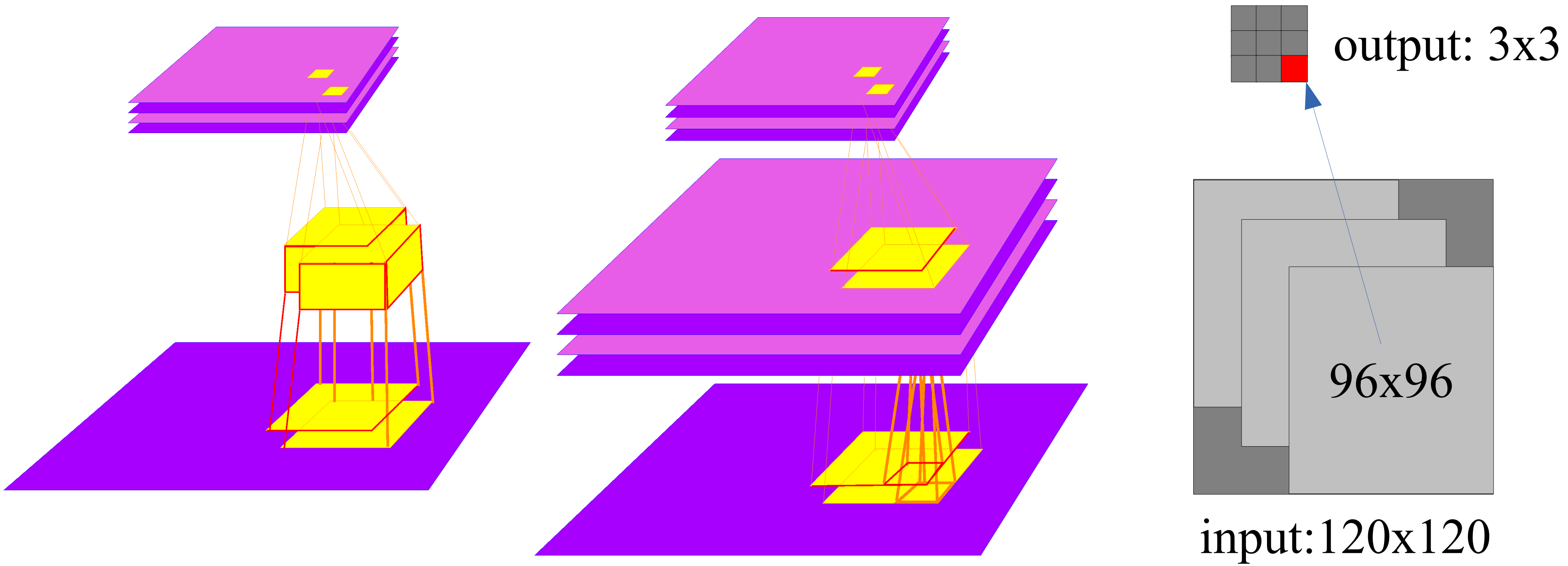
```
model = nn.Sequential()
-- stage 1 : filter bank -> squashing -> L2 pooling -> normalization
model:add(nn.SpatialConvolutionMM(nfeats, nstates[1], filtsiz, filtsiz))
model:add(nn.Tanh())
model:add(nn.SpatialLPPooling(nstates[1],2,poolsiz,poolsiz,poolsiz,poolsiz))
model:add(nn.SpatialSubtractiveNormalization(nstates[1], normkernel))
-- stage 2 : filter bank -> squashing -> L2 pooling -> normalization
model:add(nn.SpatialConvolutionMM(nstates[1],nstates[2],filtsiz,filtsiz))
model:add(nn.Tanh())
model:add(nn.SpatialLPPooling(nstates[2],2,poolsiz,poolsiz,poolsiz,poolsiz))
model:add(nn.SpatialSubtractiveNormalization(nstates[2], normkernel))
-- stage 3 : 2 fully-connected layers
model:add(nn.Reshape(nstates[2]*filtsize*filtsize))
model:add(nn.Linear(nstates[2]*filtsize*filtsize, nstates[3]))
model:add(nn.Tanh())
model:add(nn.Linear(nstates[3], noutputs))
```

– http://www.torch.ch (Torch7: Lua-based dev environment for ML, CV....)
– http://code.cogbits.com/wiki/doku.php (Torch7 tutorials/demos by C. Farabet)
- http://eblearn.sf.net (C++ Library with convnet support by P. Sermanet)

Convolutional Networks For Semantic Segmentation, Scene Labeling/parsing

Y LeCun

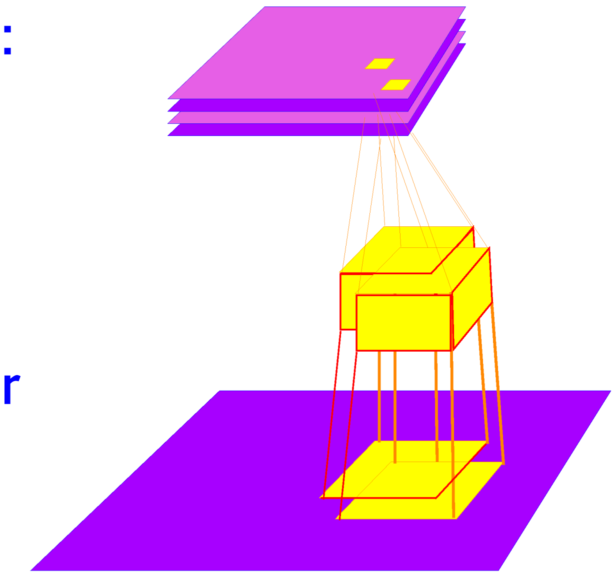
Applying a ConvNet on Sliding Windows is Very Cheap!

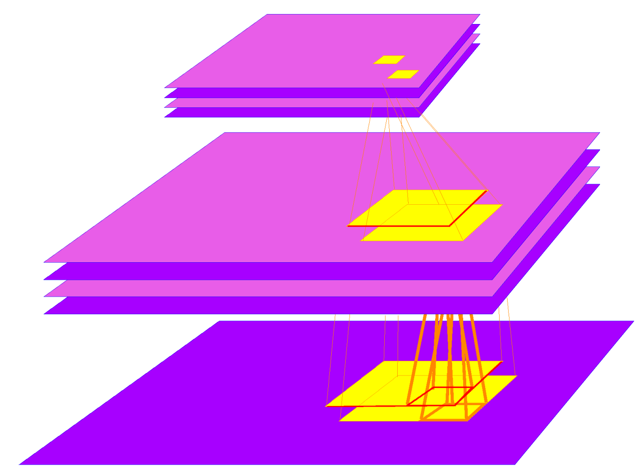


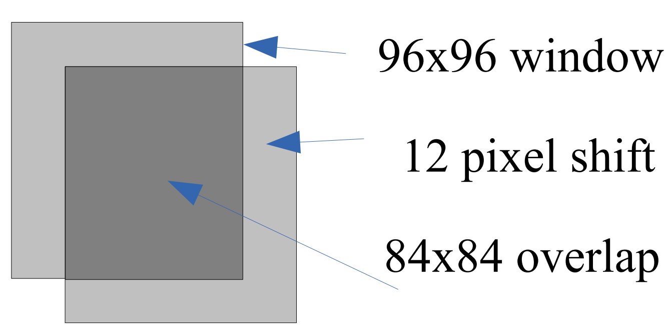
- Traditional Detectors/Classifiers must be applied to every location on a large input image, at multiple scales.
- Convolutional nets can be applied to large images very cheaply.
- The network can be applied to multiple scales every half octave

Building a Detector/Recognizer: Replicated Convolutional Nets

- Computational cost for replicated convolutional net:
  - 96x96 -> 4.6 million multiply-accumulate operations
  - 120x120 -> 8.3 million multiply-accumulate ops
  - 240x240 -> 47.5 million multiply-accumulate ops
  - 480x480 -> 232 million multiply-accumulate ops
- Computational cost for a non-convolutional detector of the same size, applied every 12 pixels:
  - 96x96 -> 4.6 million multiply-accumulate operations
  - 120x120 -> 42.0 million multiply-accumulate operations
  - 240x240 -> 788.0 million multiply-accumulate ops
  - 480x480 -> 5,083 million multiply-accumulate ops







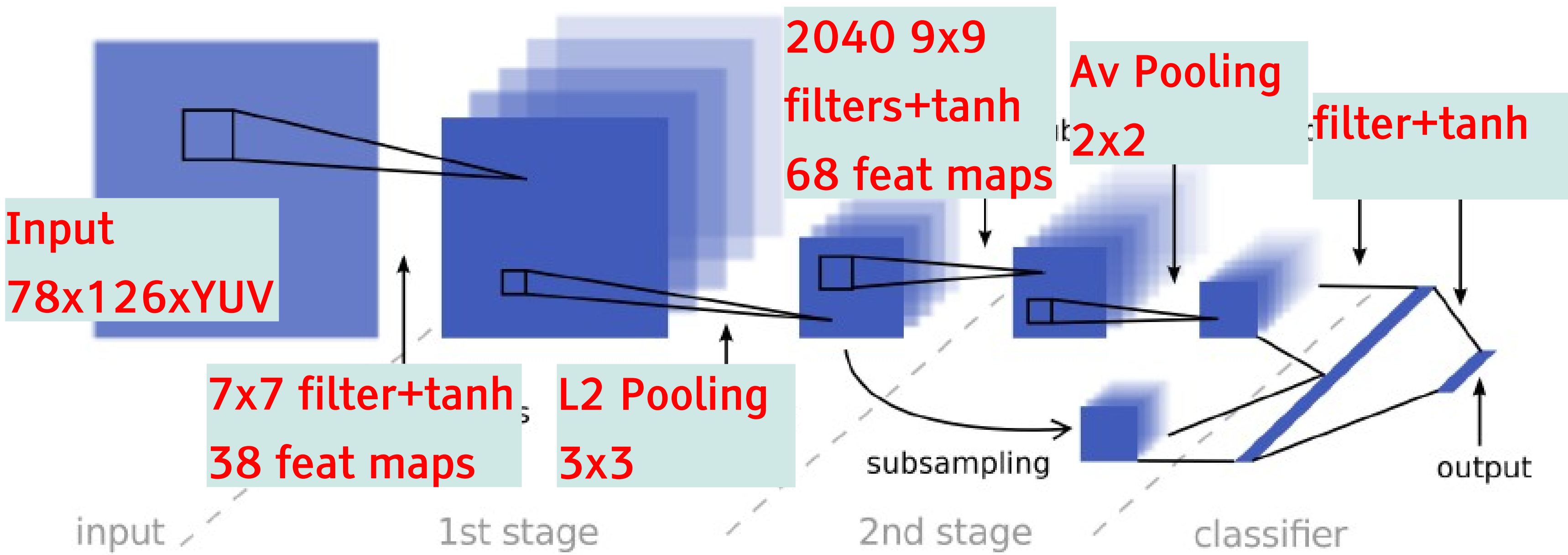
ConvNets for Pedestrian Detection, Face Detection

Face [Vaillant et al IEE 1994] [Garcia et al PAMI 2005] [Osadchy et al JMLR 2007]

Pedestrian: [Kavukcuoglu et al. NIPS 2010] [Sermanet et al. CVPR 2013]

ConvNet Architecture with Multi-Stage Features for Object Detection

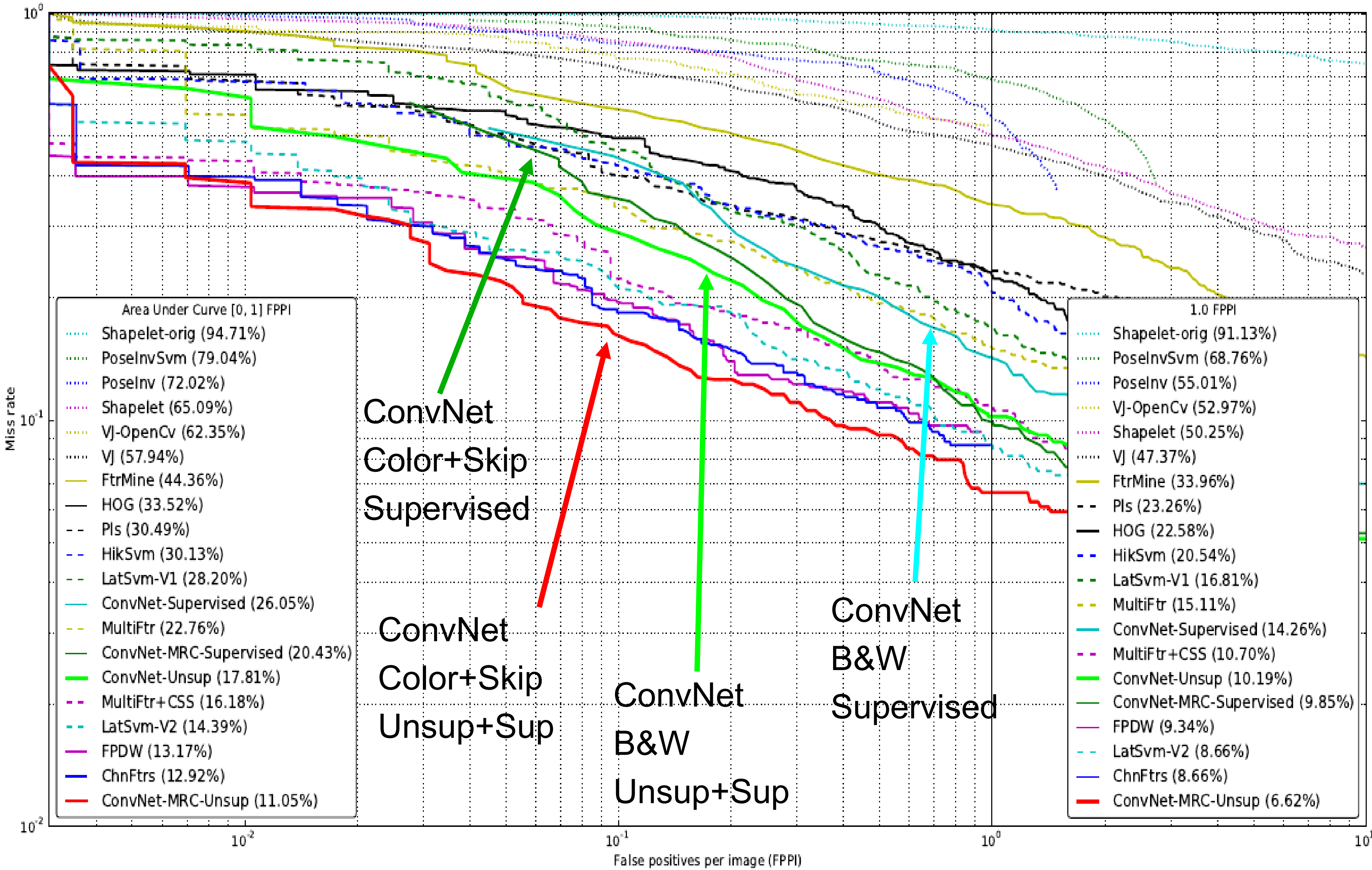
- Feature maps from all stages are pooled/subsampled and sent to the final classification layers
  - Pooled low-level features: good for textures and local motifs
  - High-level features: good for "gestalt" and global shape



Task	Single-Stage features	Multi-Stage features	Improvement %
Pedestrians detection (INRIA)	14.26%	9.85%	31%
Traffic Signs classification (GTSRB) [33]	1.80%	0.83%	54%
House Numbers classification (SVHN) [32]	5.54%	5.36%	3.2%

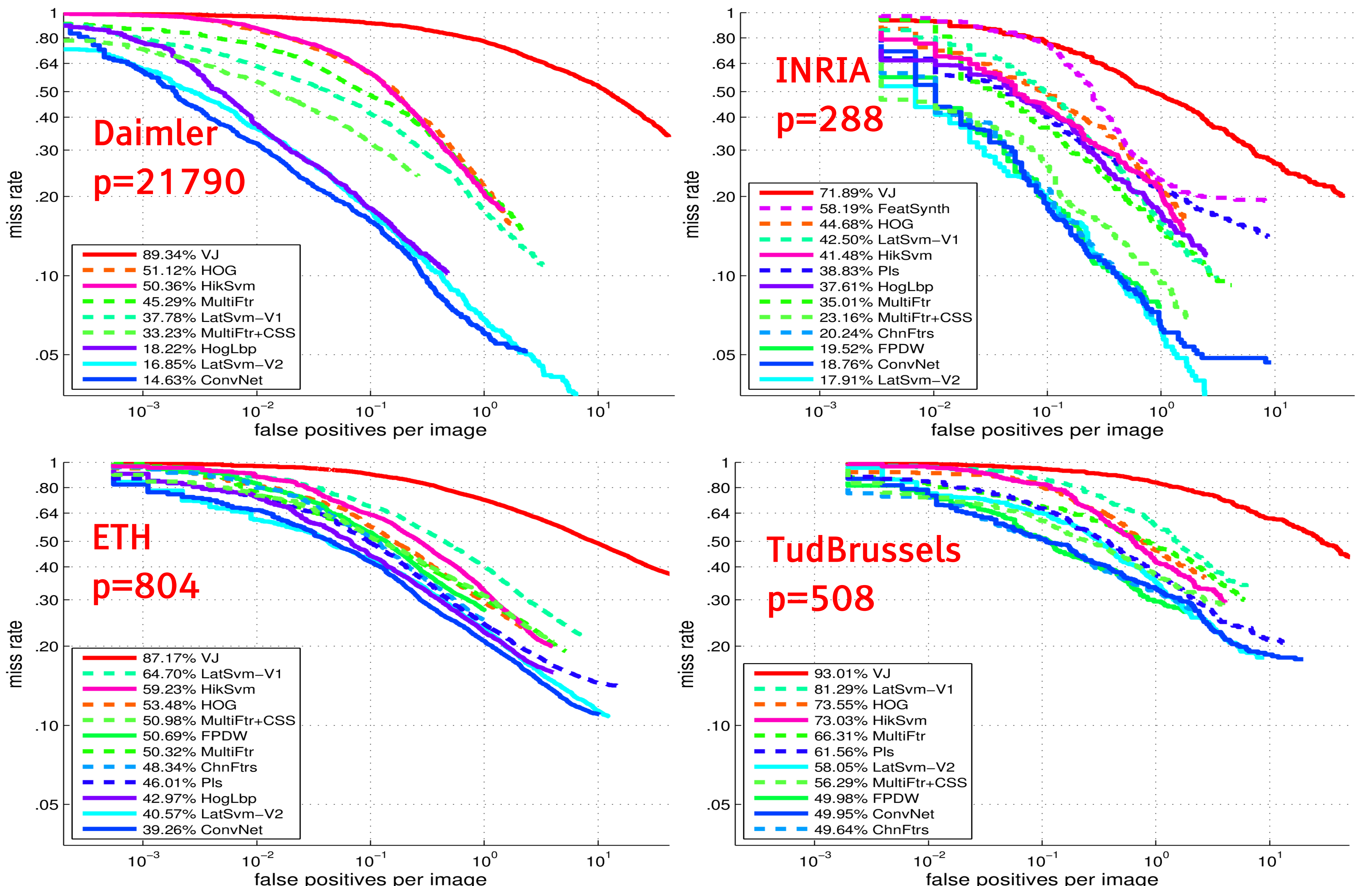
[Sermanet, Chintala, LeCun CVPR 2013]

Pedestrian Detection: INRIA Dataset. Miss rate vs false positives

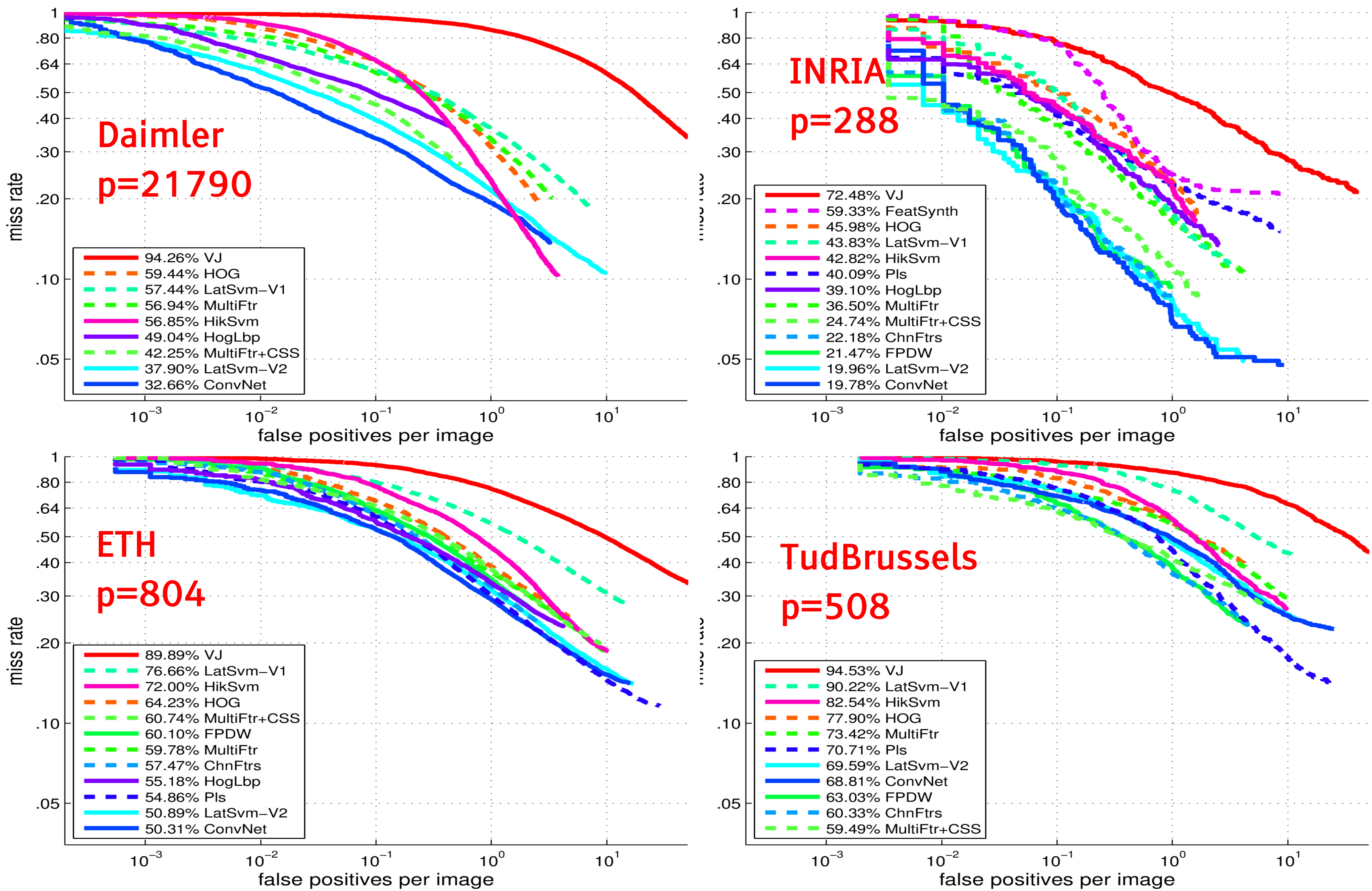


[Kavukcuoglu et al. NIPS 2010] [Sermanet et al. ArXiv 2012]

Results on “Near Scale” Images (>80 pixels tall, no occlusions)

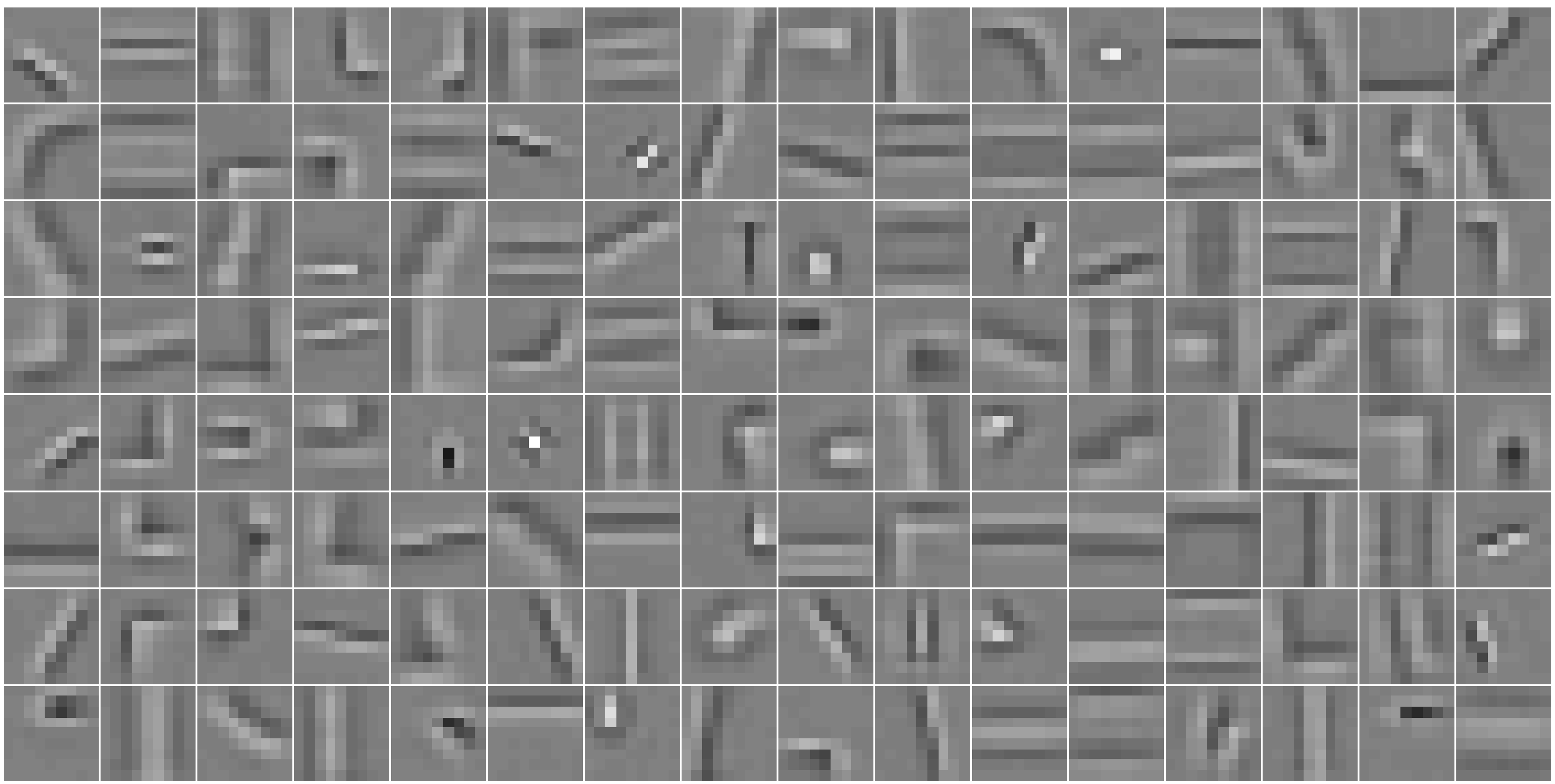


Results on “Reasonable” Images (>50 pixels tall, few occlusions)



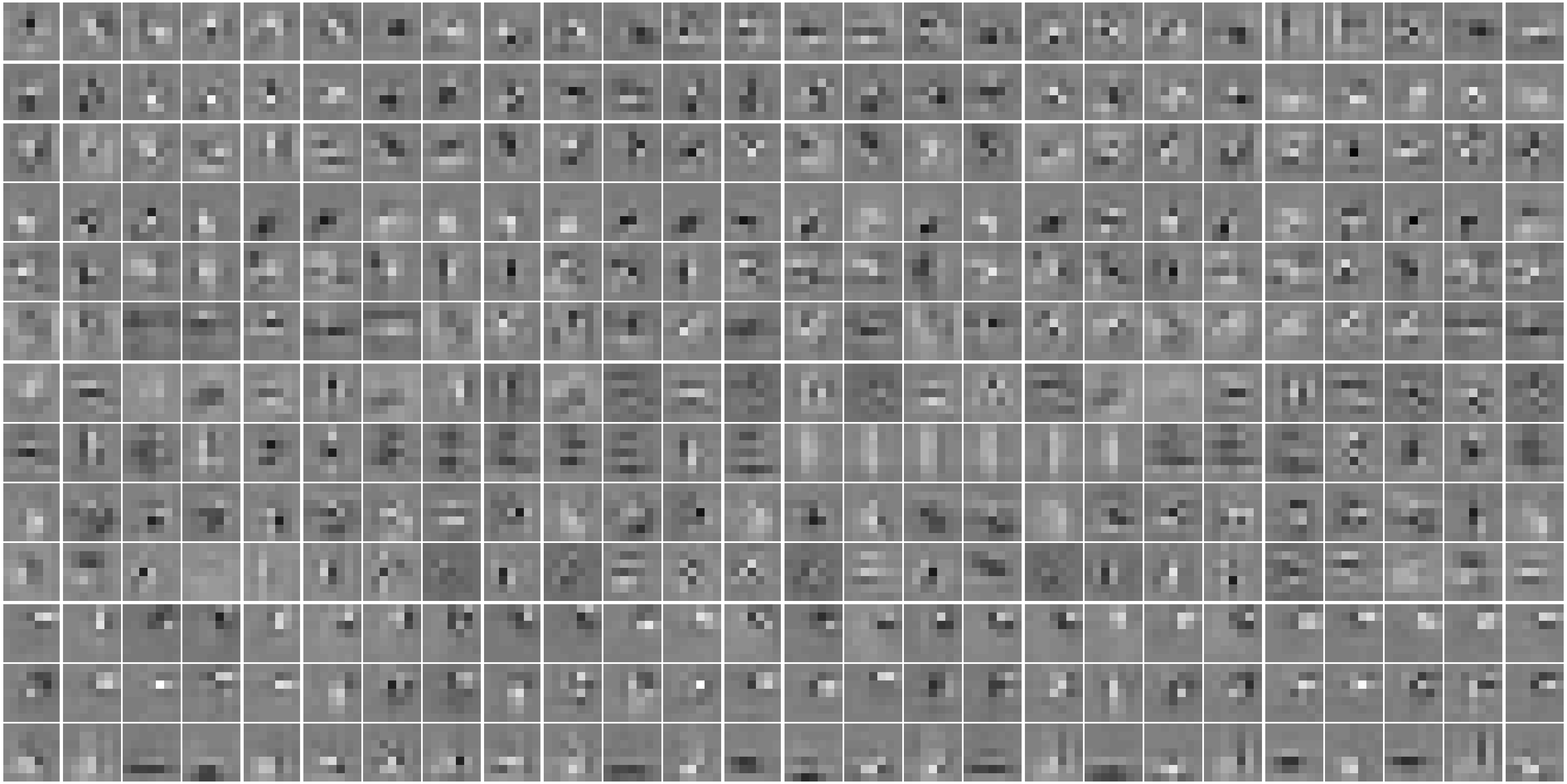
Unsupervised pre-training with convolutional PSD

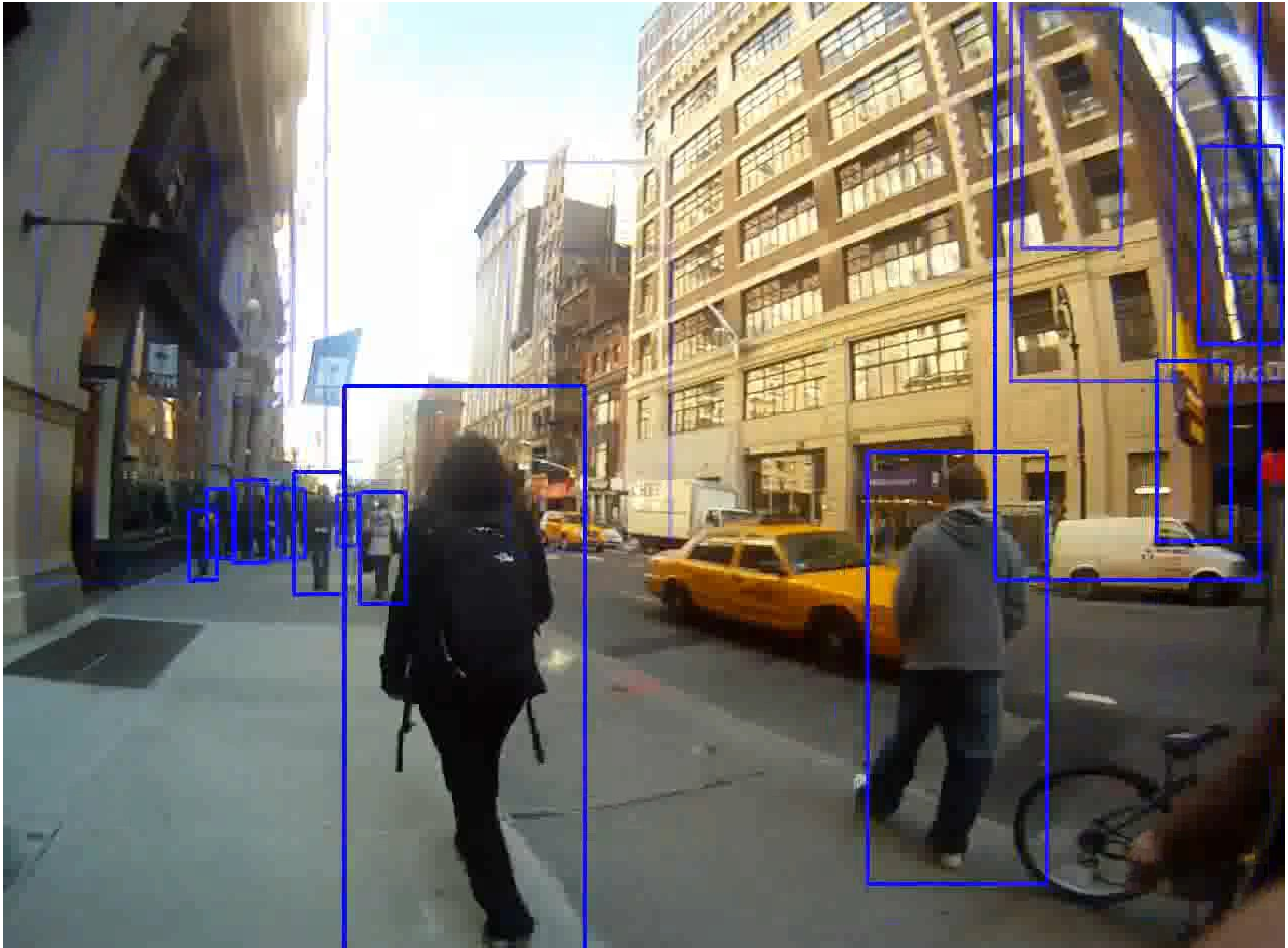
- 128 stage-1 filters on Y channel.
- Unsupervised training with convolutional predictive sparse decomposition



Unsupervised pre-training with convolutional PSD

- Stage 2 filters.
- Unsupervised training with convolutional predictive sparse decomposition





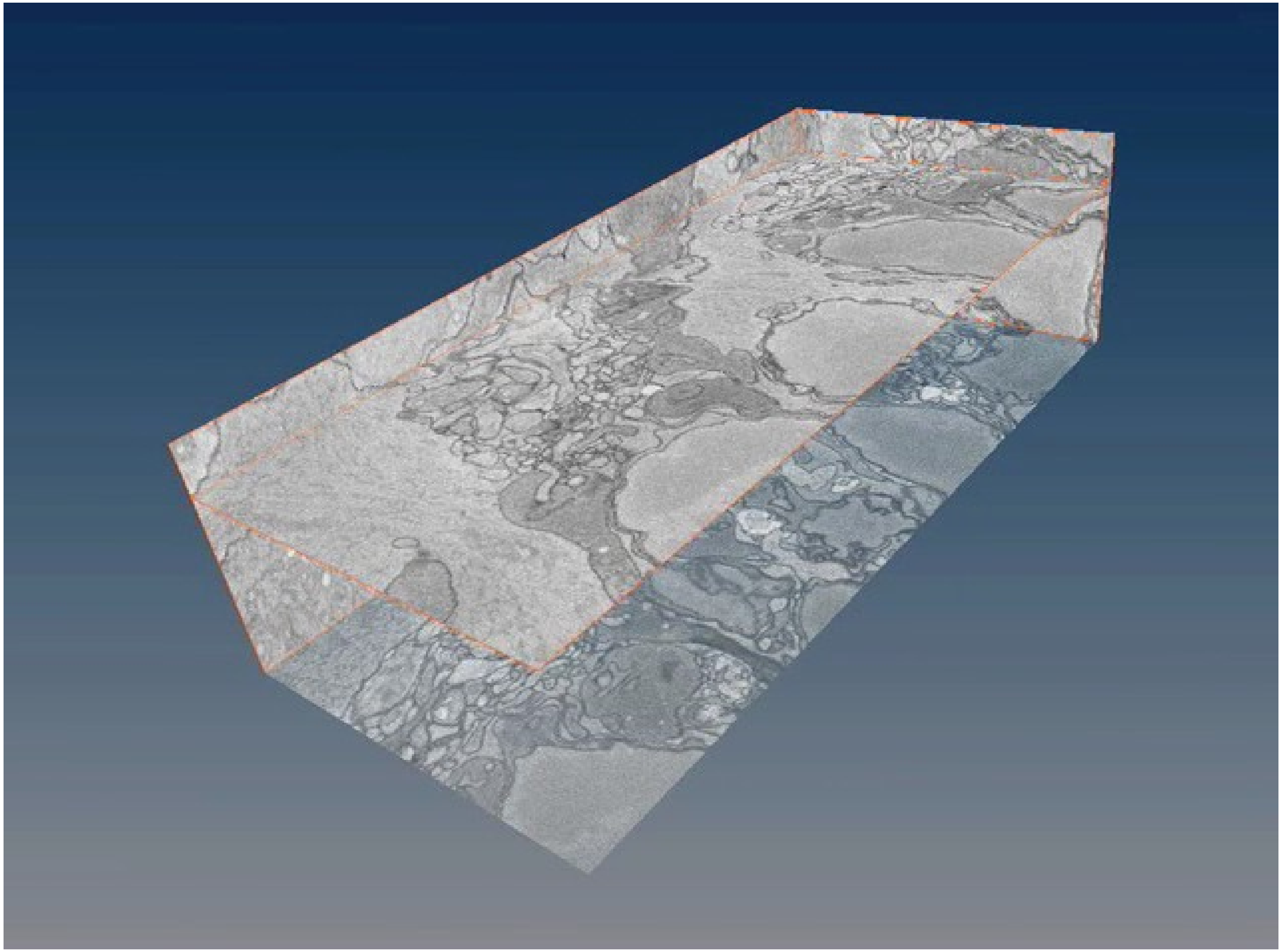
ConvNets for Image Segmentation

- Biological Image Segmentation [Ning et al. IEEE-TIP 2005]

- Image Labeling for Off-Road Robots [Hadsell JFR 2008]

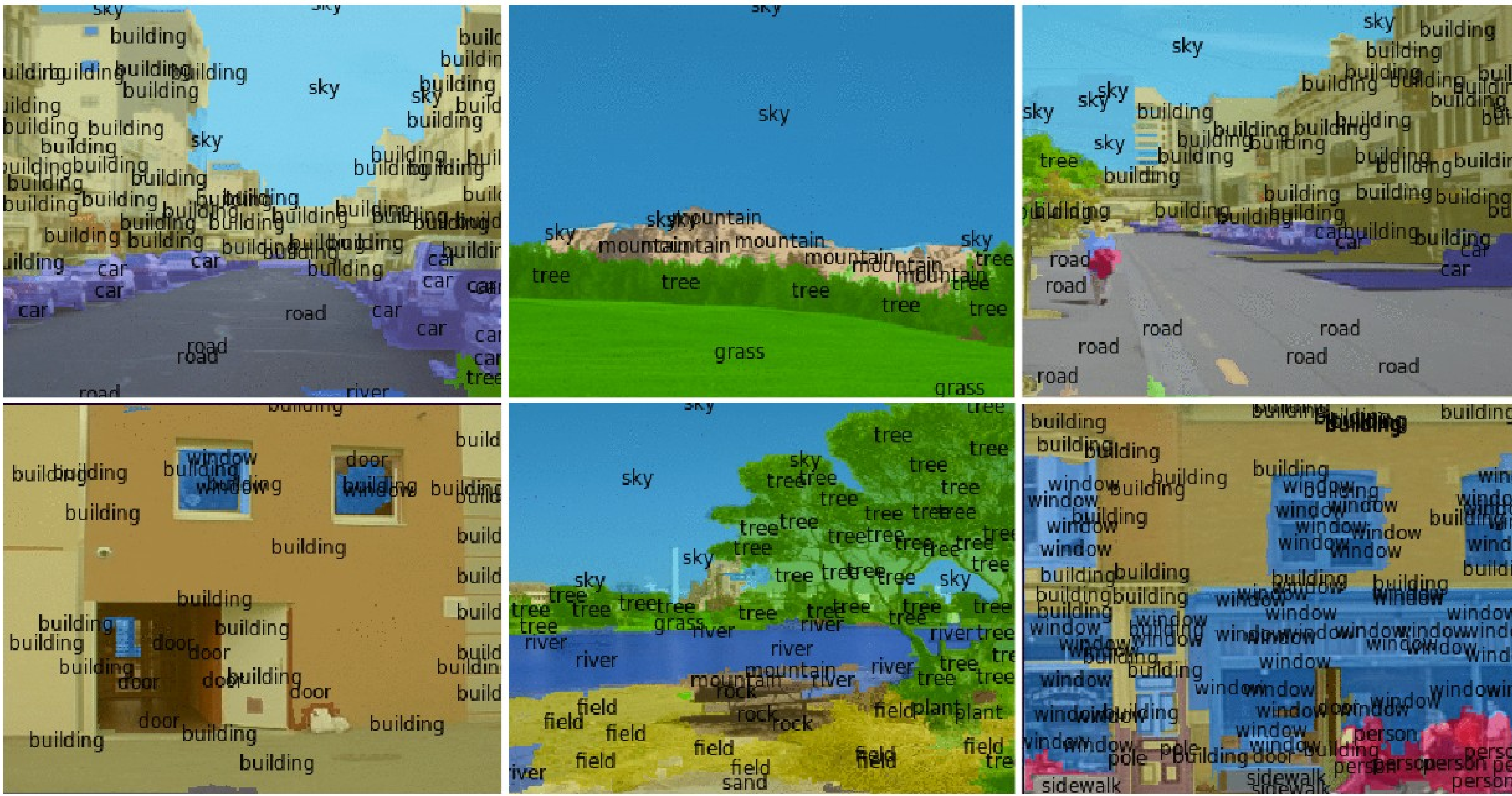
ConvNet in Volumetric Image Segmentation

- 3D convnet to segment volumetric images [Jain, Turaga, Seung 2007]



Semantic Segmentation

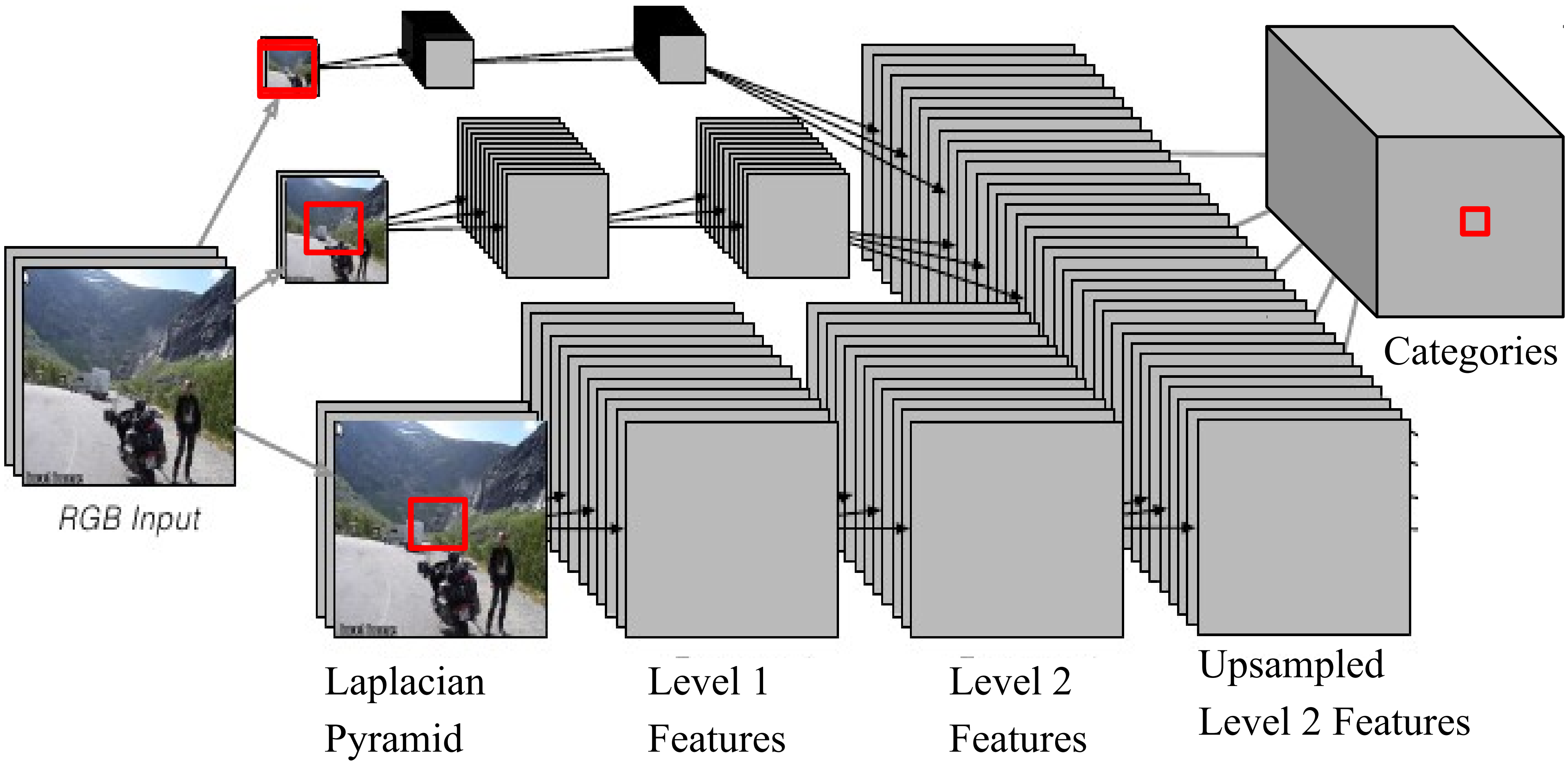
- Labeling each pixel with the category of the object it belongs to



[Farabet et al. ICML 2012]

Scene Parsing/Labeling: ConvNet Architecture

- Each output sees a large input context:
  - 46x46 window at full rez; 92x92 at ½ rez; 184x184 at ¼ rez
  - [7x7conv]->[2x2pool]->[7x7conv]->[2x2pool]->[7x7conv]->
  - Trained supervised on fully-labeled images



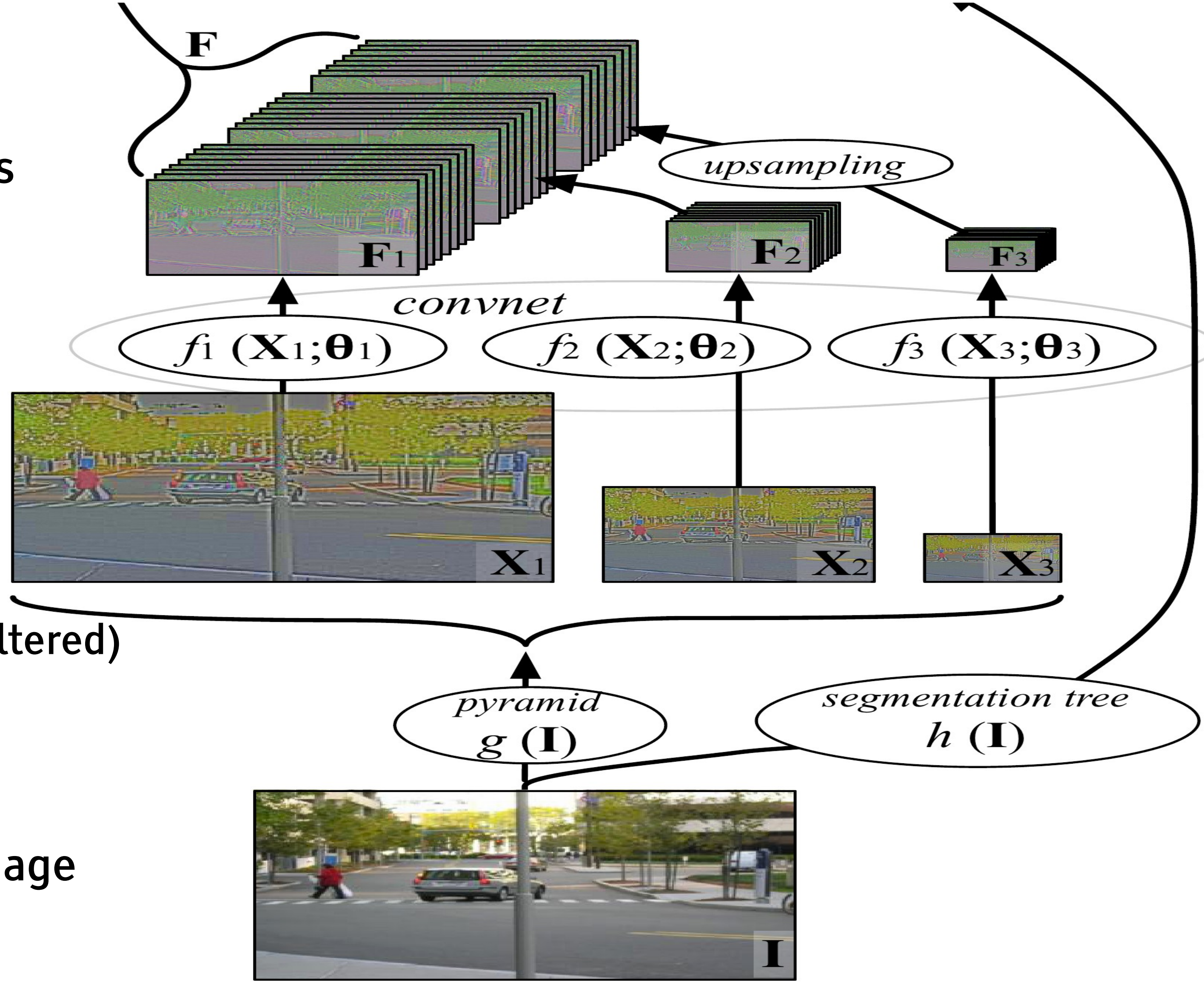
Scene Parsing/Labeling: System Architecture

Dense
Feature Maps

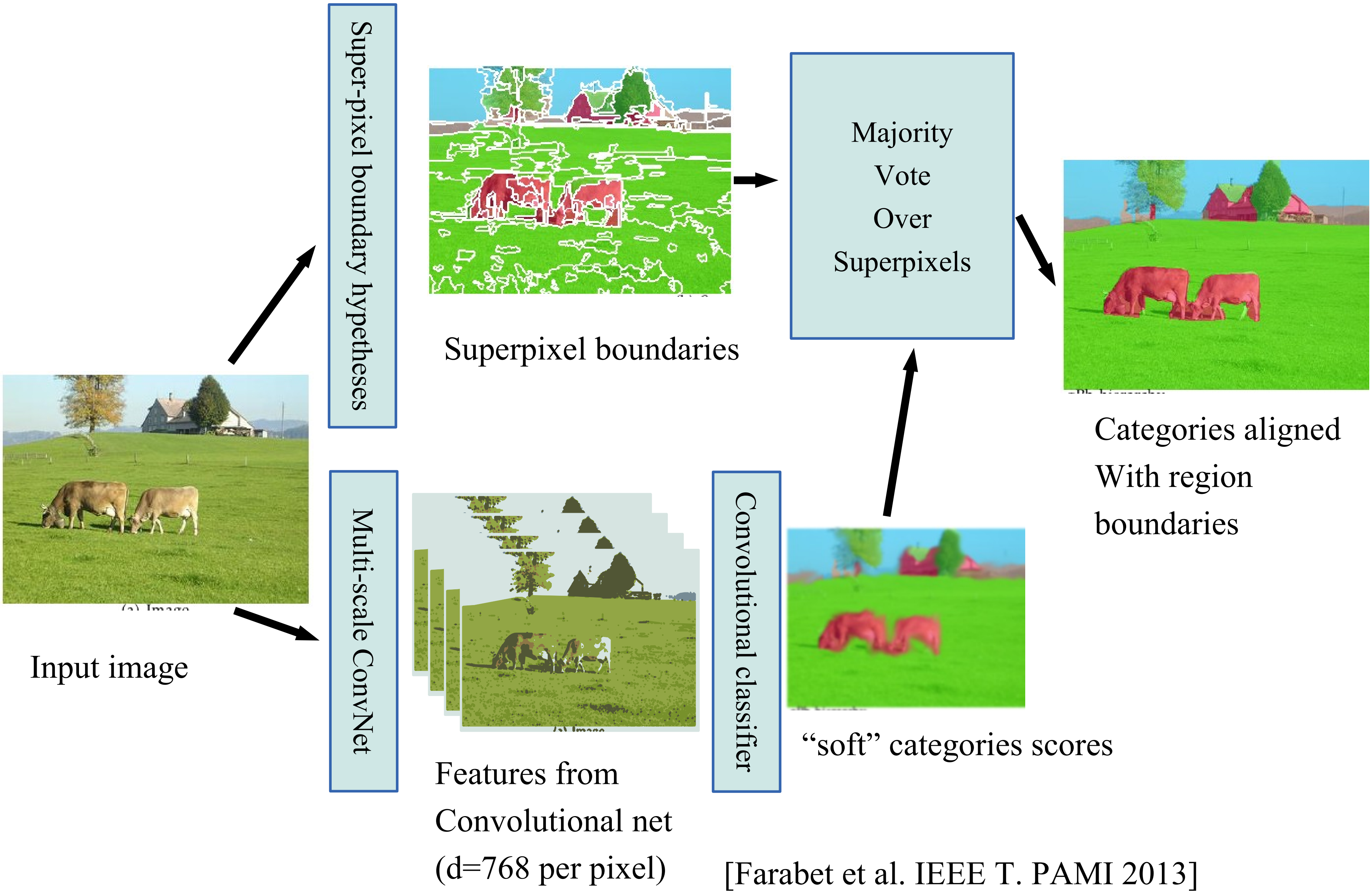
ConvNet

Multi-Scale
Pyramid
(Band-pass Filtered)

Original Image

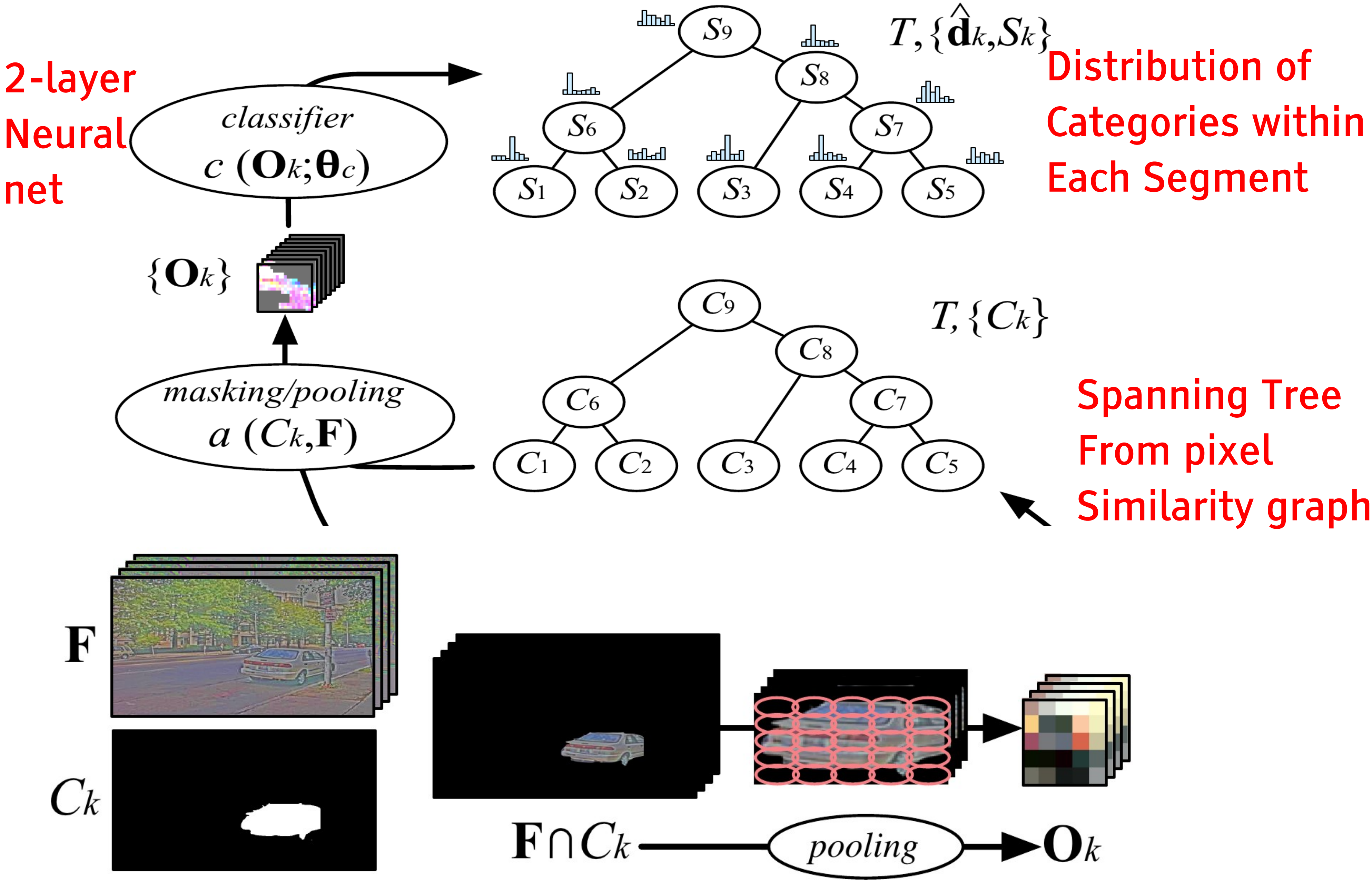


Method 1: majority over super-pixel regions



[Farabet et al. IEEE T. PAMI 2013]

Method 2: optimal cover of purity tree



[Farabet et al. ICML 2012]

Scene Parsing/Labeling: Performance

- Stanford Background Dataset [Gould 1009]: 8 categories

	Pixel Acc.	Class Acc.	CT (sec.)
Gould *et al.* 2009 [14]	76.4%	-	10 to 600s
Munoz *et al.* 2010 [32]	76.9%	66.2%	12s
Tighe *et al.* 2010 [46]	77.5%	-	10 to 300s
Socher *et al.* 2011 [45]	78.1%	-	?
Kumar *et al.* 2010 [22]	79.4%	-	$< 600s$
Lempitzky *et al.* 2011 [28]	**81.9%**	72.4%	$> 60s$
singlescale convnet	66.0 %	56.5 %	0.35s
multiscale convnet	78.8 %	72.4%	0.6s
multiscale net + superpixels	80.4%	74.56%	**0.7s**
multiscale net + gPb + cover	80.4%	75.24%	61s
multiscale net + CRF on gPb	81.4%	**76.0%**	60.5s

[Farabet et al., rejected from CVPR 2012]

[Farabet et al. ICML 2012] [Farabet et al. IEEE T. PAMI 2013]

Scene Parsing/Labeling: Performance

	Pixel Acc.	Class Acc.
Liu *et al.* 2009 [31]	74.75%	-
Tighe *et al.* 2010 [44]	76.9%	29.4%
raw multiscale net[1]	67.9%	45.9%
multiscale net + superpixels[1]	71.9%	**50.8%**
multiscale net + cover[1]	72.3%	**50.8%**
multiscale net + cover[2]	**78.5%**	29.6%

- SIFT Flow Dataset
- [Liu 2009]:
- 33 categories

	Pixel Acc.	Class Acc.
Tighe *et al.* 2010 [44]	66.9%	7.6%
raw multiscale net[1]	37.8%	**12.1%**
multiscale net + superpixels[1]	44.1%	**12.4%**
multiscale net + cover[1]	46.4%	**12.5%**
multiscale net + cover[2]	**67.8%**	9.5%

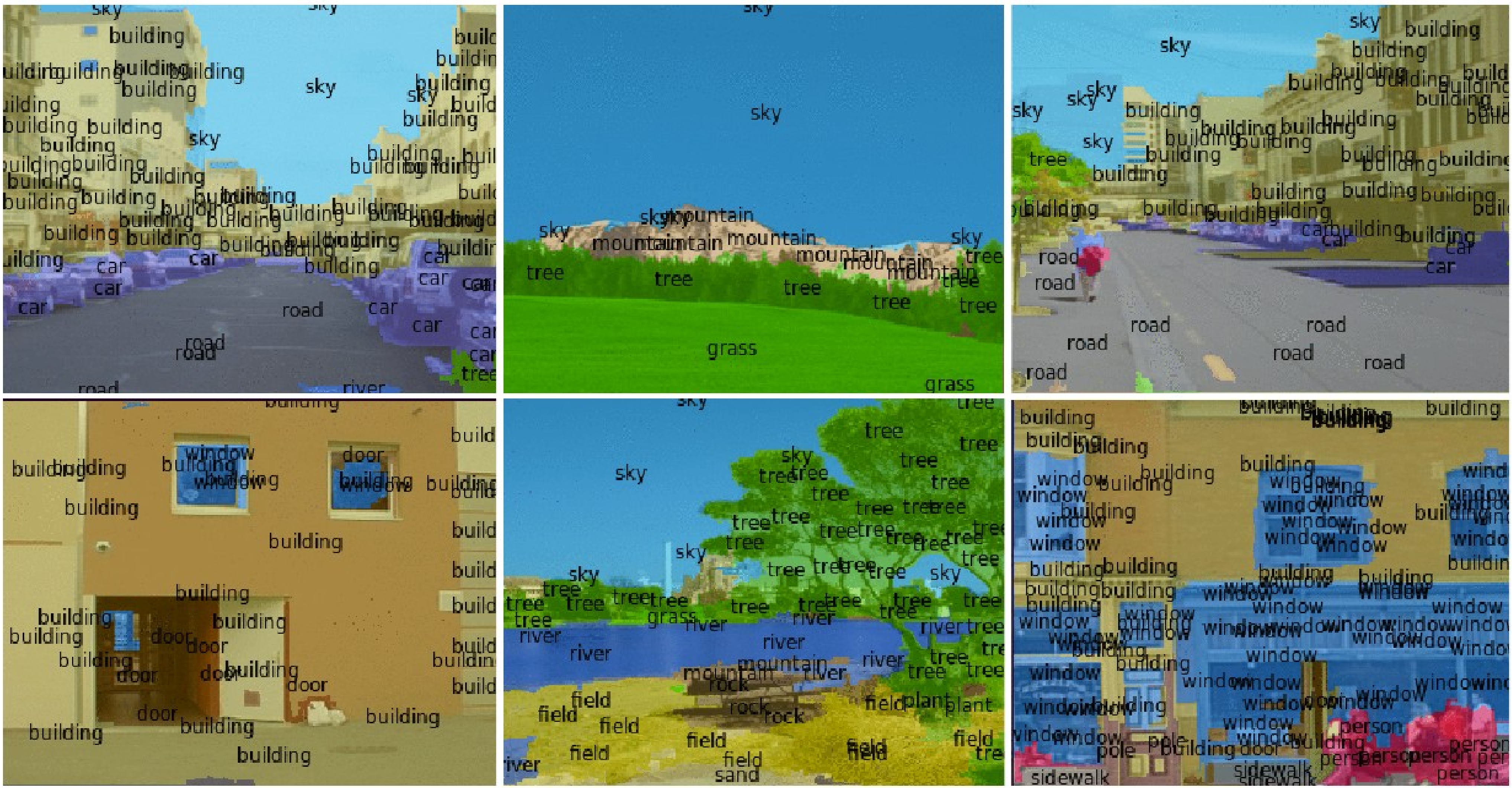
- Barcelona dataset
- [Tighe 2010]:
- 170 categories.

[Farabet et al., rejected from CVPR 2012]

[Farabet et al. ICML 2012] [Farabet et al. IEEE T. PAMI 2013]

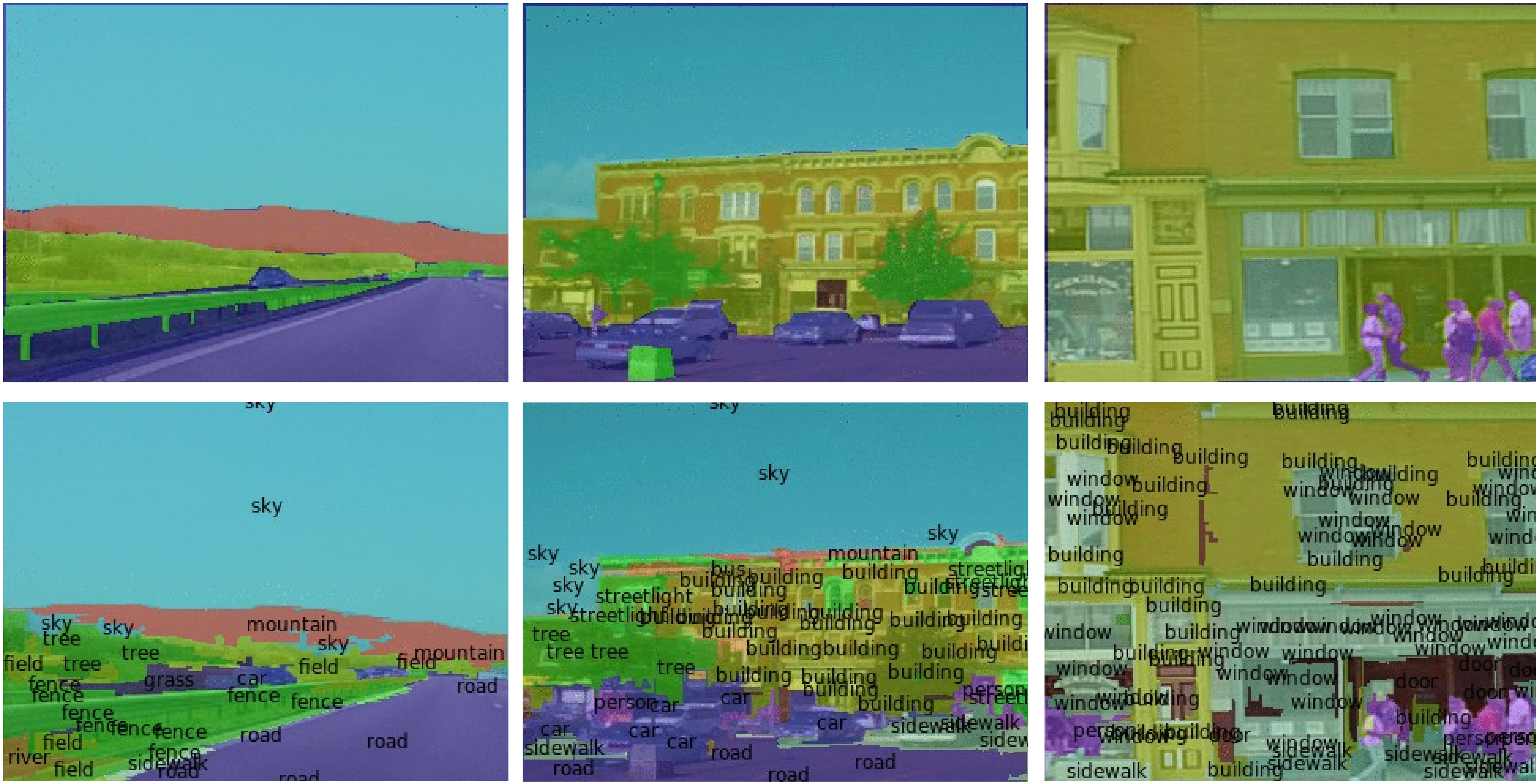
Scene Parsing/Labeling: SIFT Flow dataset (33 categories)

- Samples from the SIFT-Flow dataset (Liu)



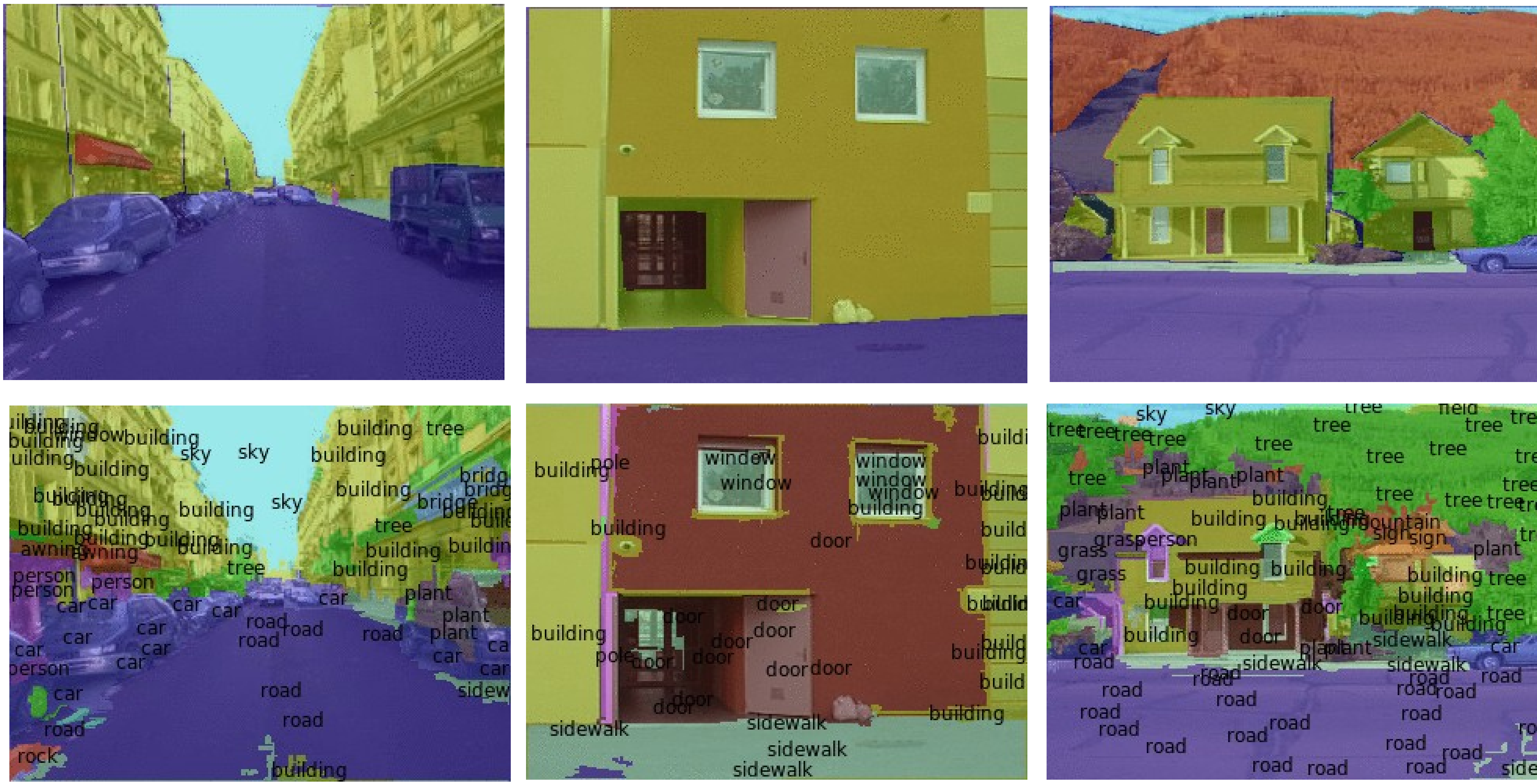
[Farabet et al. ICML 2012]

Scene Parsing/Labeling: SIFT Flow dataset (33 categories)



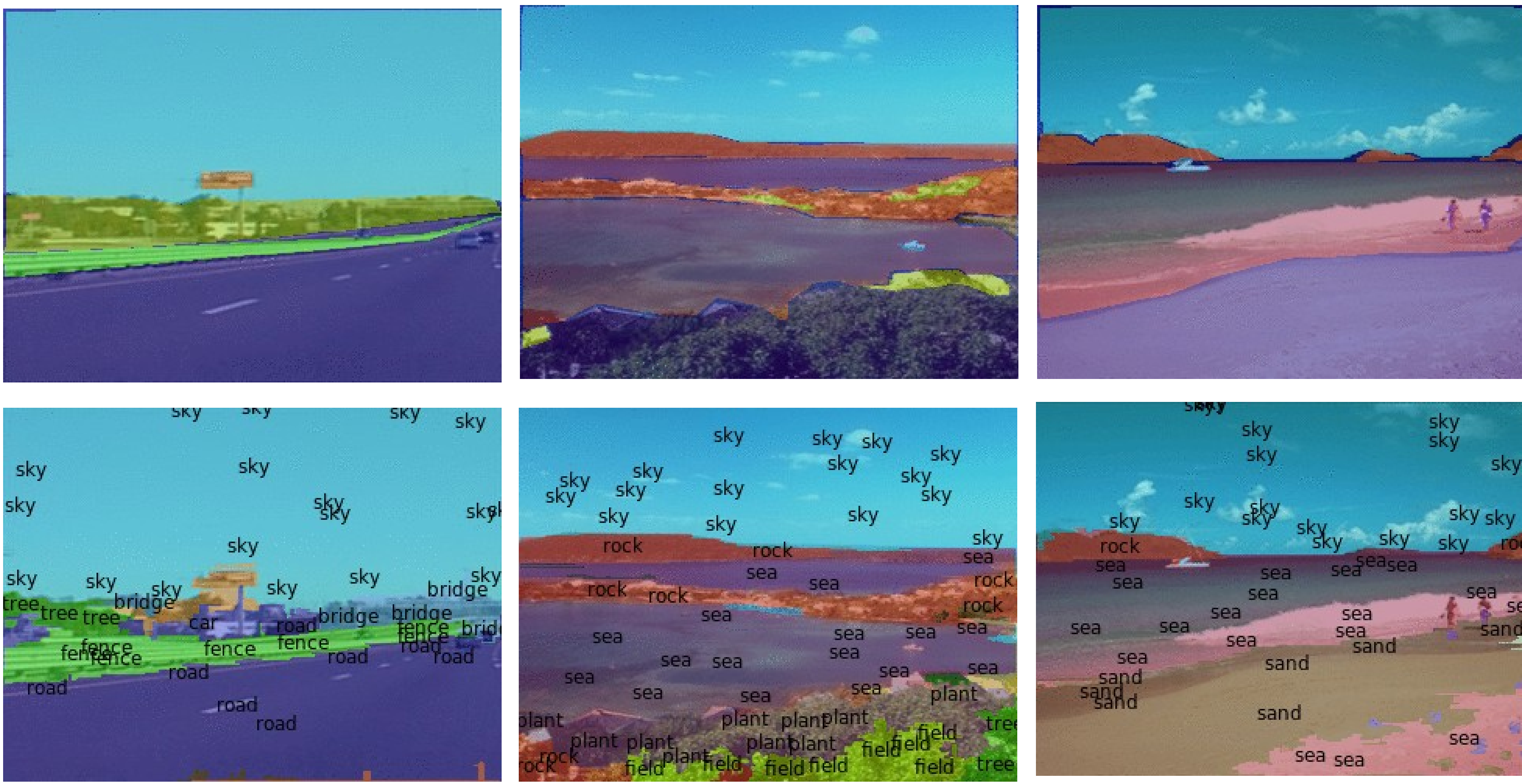
[Farabet et al. ICML 2012]

Scene Parsing/Labeling



[Farabet et al. ICML 2012]

Scene Parsing/Labeling



[Farabet et al. ICML 2012]

Scene Parsing/Labeling

[Farabet et al. 2012]

Scene Parsing/Labeling

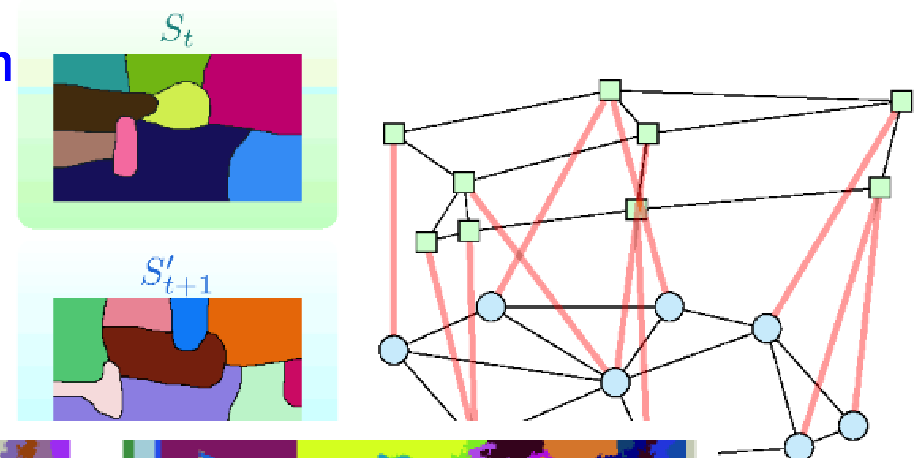
[Farabet et al. 2012]

Scene Parsing/Labeling

- No post-processing
- Frame-by-frame
- ConvNet runs at 50ms/frame on Virtex-6 FPGA hardware
  - But communicating the features over ethernet limits system perf.

Temporal Consistency

- Spatio-Temporal Super-Pixel segmentation
  - [Couprie et al ICIP 2013]
  - [Couprie et al JMLR under review]
  - Majority vote over super-pixels



Independent segmentations S'_1, S'_2 and S'_3

Temporally consistent segmentations $S_1 (= S'_1)$, S_2, and S_3

Scene Parsing/Labeling: Temporal Consistency

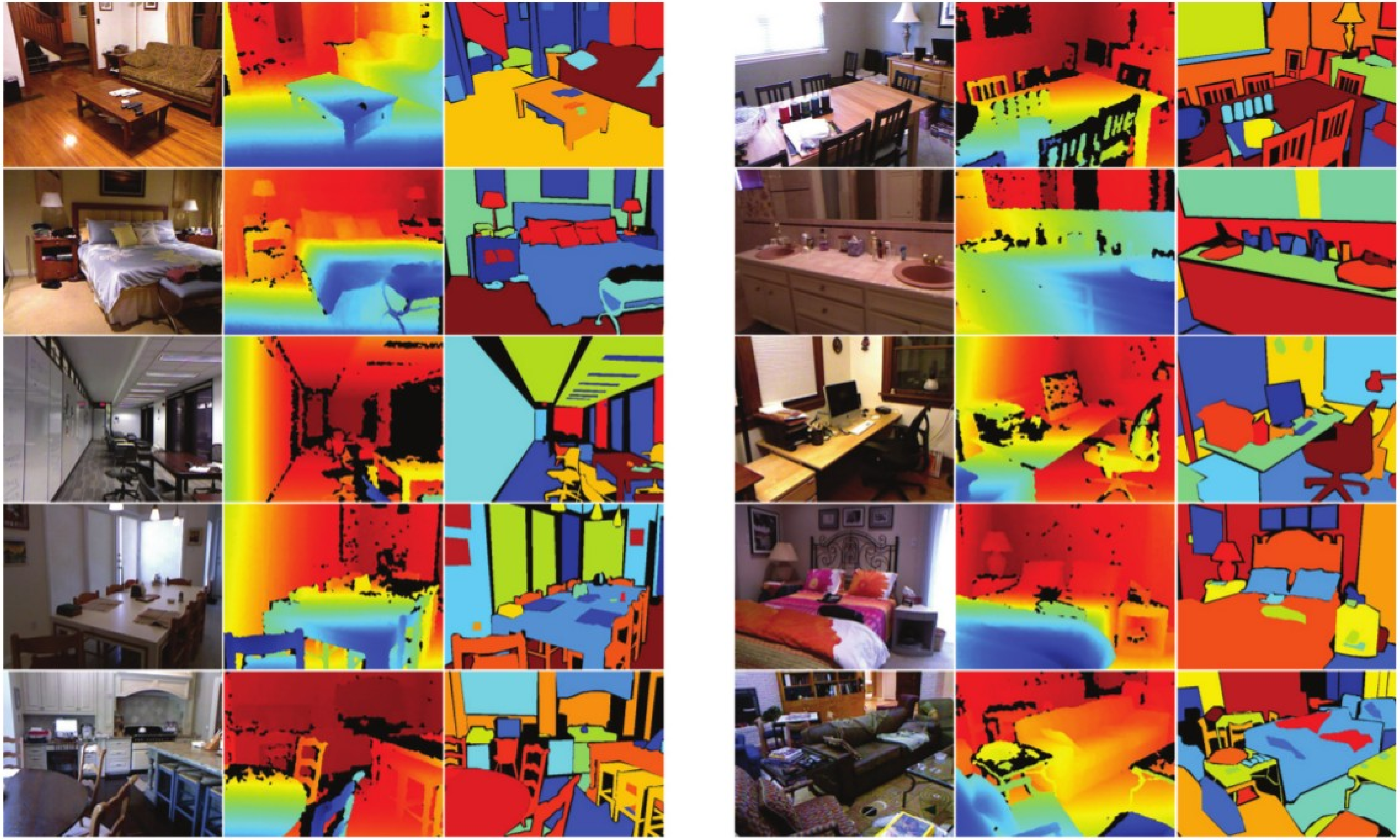
- Causal method for temporal consistency

[Couprie, Farabet, Najman, LeCun ICIP 2013]

NYU RGB-Depth v2: Indoor Scenes Dataset

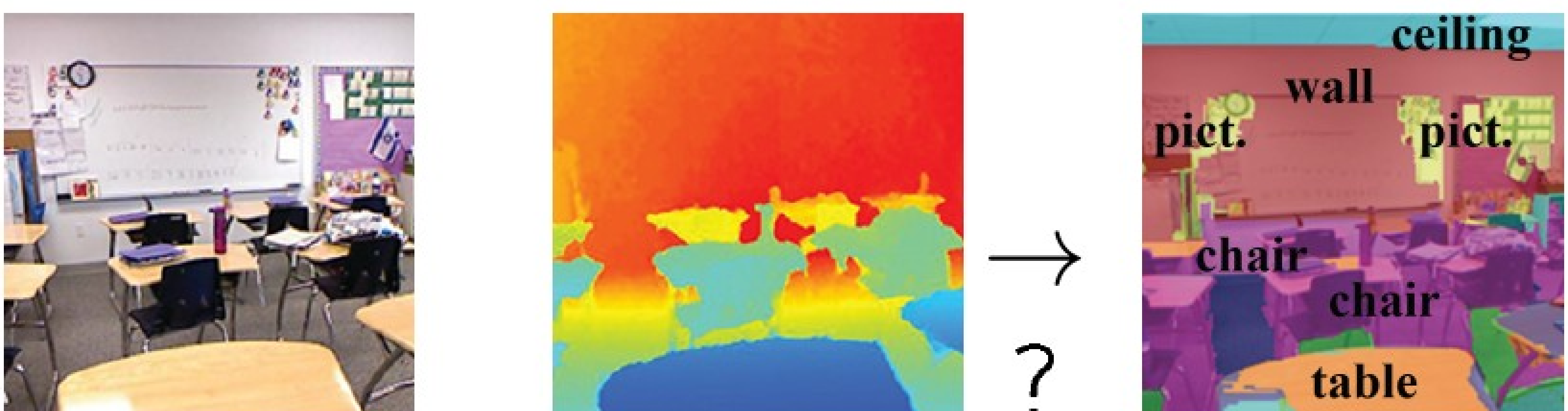
- 407024 RGB-D images of apartments
- 1449 labeled frames, 894 object categories

[Silberman et al. 2012]



NYU RGB-D Dataset

- Captured with a Kinect on a steadycam



Results

	Class Occurrences	Multiscale Convnet Acc. Farabet et al. (2013)	MultiScl. Cnet +depth Acc.
bed	4.4%	30.3	**38.1**
objects	7.1 %	**10.9**	8.7
chair	3.4%	**44.4**	34.1
furnit.	12.3%	28.5	**42.4**
ceiling	1.4%	33.2	**62.6**
floor	9.9%	68.0	**87.3**
deco.	3.4%	38.5	**40.4**
sofa	3.2%	**25.8**	24.6
table	3.7%	**18.0**	10.2
wall	24.5%	**89.4**	86.1
window	5.1%	**37.8**	15.9
books	2.9%	**31.7**	13.7
TV	1.0%	**18.8**	6.0
unkn.	17.8%	-	-
Avg. Class Acc.	-	35.8	**36.2**
Pixel Accuracy (mean)	-	51.0	**52.4**
Pixel Accuracy (median)	-	51.7	**52.9**
Pixel Accuracy (std. dev.)	-	15.2	15.2

Results

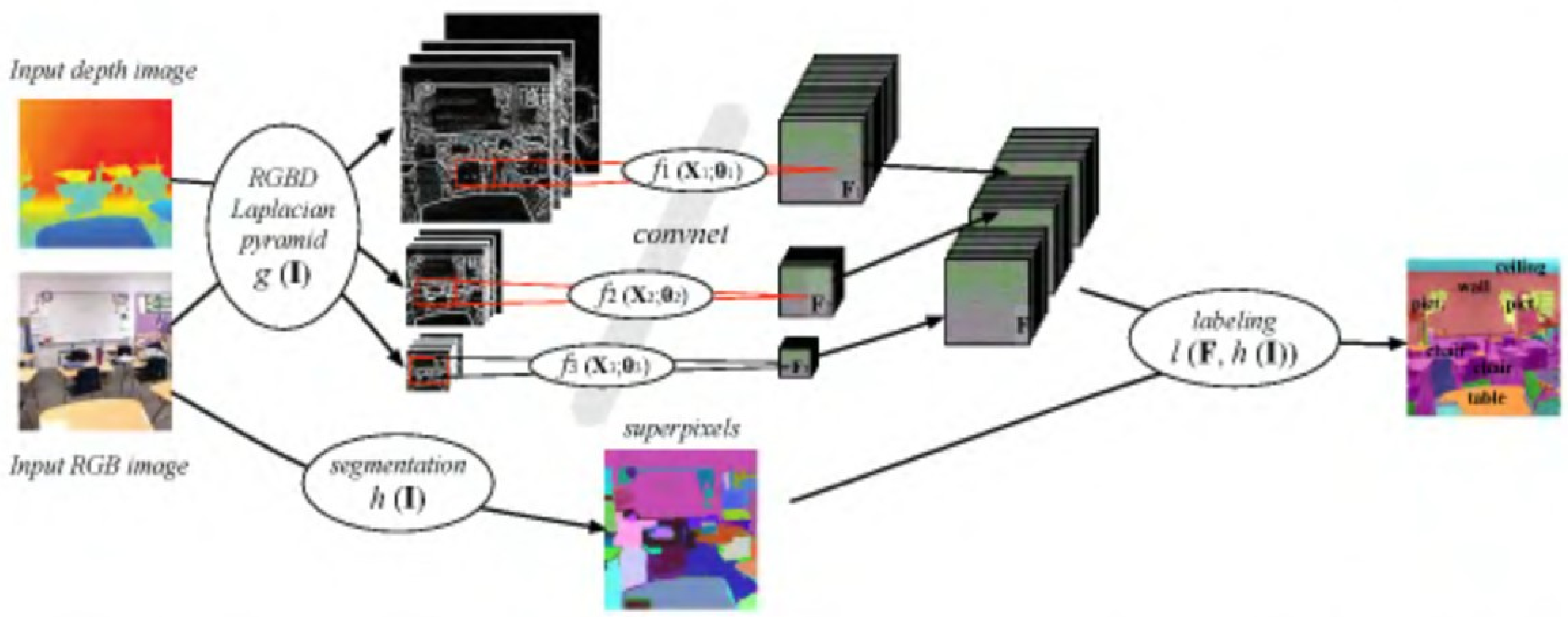
- **Depth helps a bit**
  - Helps a lot for floor and props
  - Helps surprisingly little for structures, and hurts for furniture

	Ground	Furniture	Props	Structure	Class Acc.	Pixel Acc.	Comput. time (s)
Silberman et al. (2012)	68	**70**	**42**	59	59.6	58.6	>3
Cadena and Kosecka (2013)	**87.9**	**64.1**	31.0	77.8	**65.2**	**66.9**	1.7
Multiscale convnet	68.1	51.1	29.9	**87.8**	59.2	63.0	**0.7**
Multiscale+depth convnet	**87.3**	45.3	**35.5**	**86.1**	**63.5**	**64.5**	**0.7**

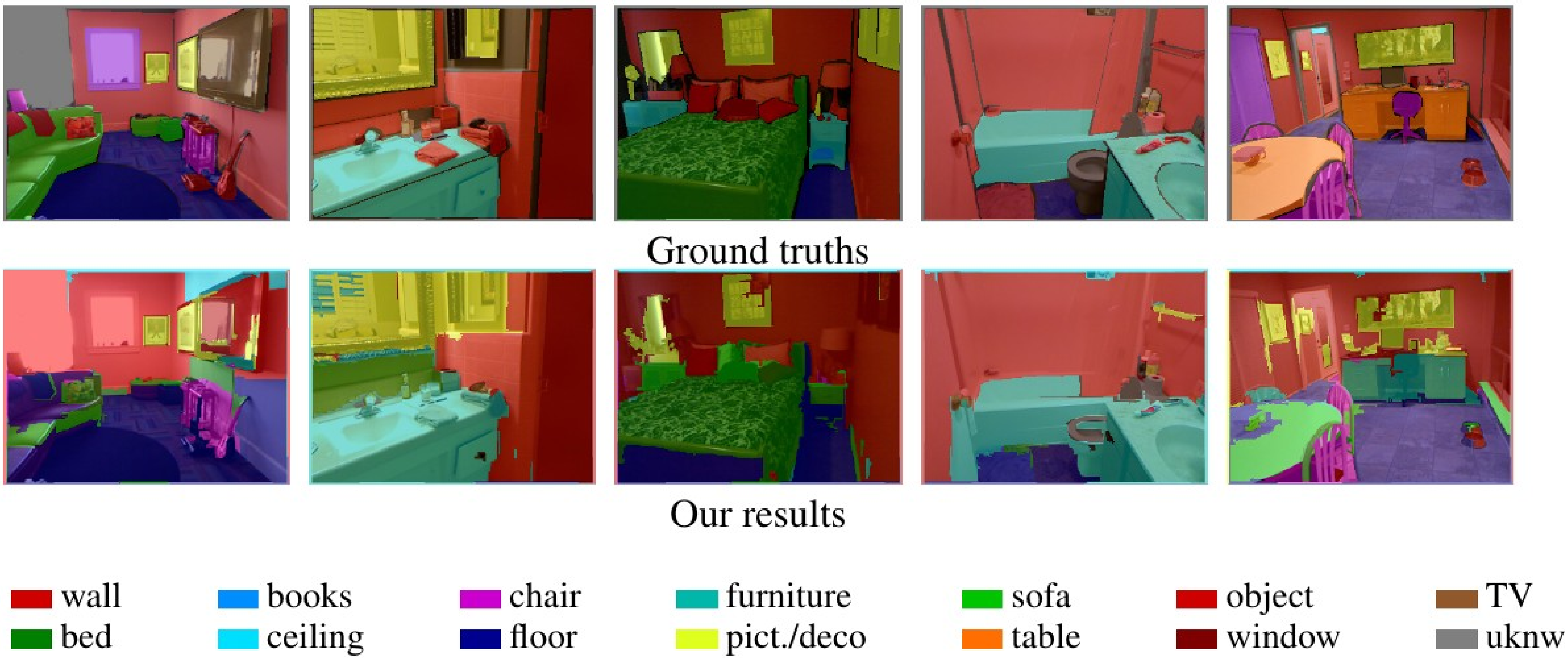
[C. Cadena, J. Kosecka "Semantic Parsing for Priming Object Detection in RGB-D Scenes" Semantic Perception Mapping and Exploration (SPME), Karlsruhe 2013]

Architecture for indoor RGB-D Semantic Segmentation

- Similar to outdoors semantic segmentation method
  - Convnet with 4 input channels
  - Vote over superpixels

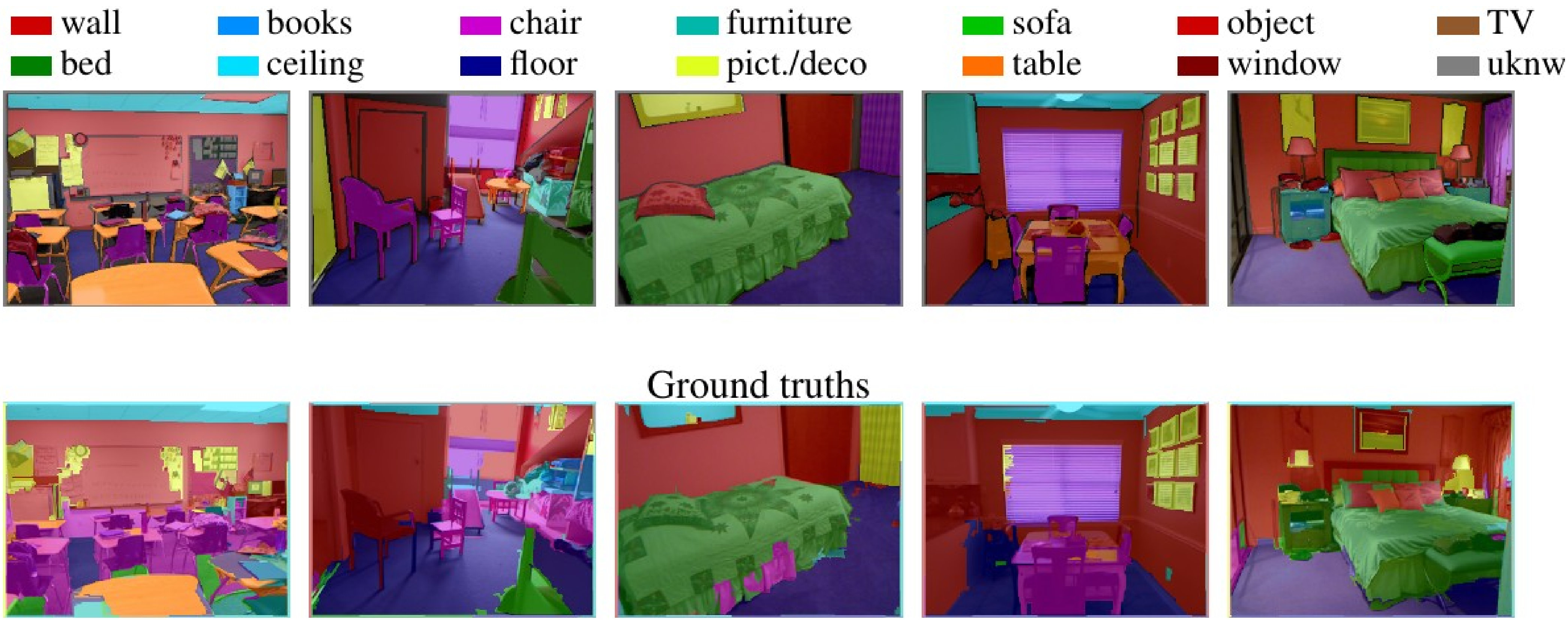


Scene Parsing/Labeling on RGB+Depth Images



[Couprie, Farabet, Najman, LeCun ICLR 2013]

Scene Parsing/Labeling on RGB+Depth Images



Ground truths

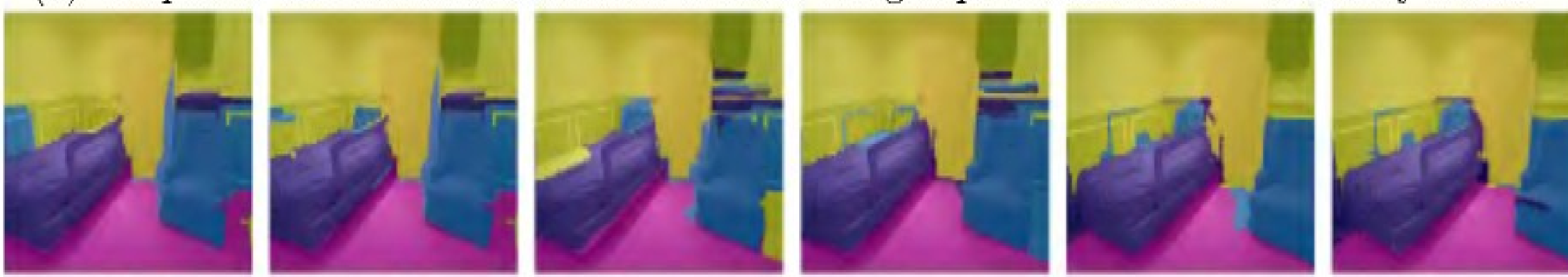
Our results

[Couprie, Farabet, Najman, LeCun ICLR 2013]

Labeling Videos

- Temporal consistency

(a) Output of the Multiscale convnet trained using depth information - frame by frame



(b) Results smoothed temporally using Couprie et al. (2013a)

[Couprie, Farabet, Najman, LeCun ICLR 2013]
[Couprie, Farabet, Najman, LeCun ICIP 2013]
[Couprie, Farabet, Najman, LeCun submitted to JMLR]

Semantic Segmentation on RGB+D Images and Videos

[Couprie, Farabet, Najman, LeCun ICIP 2013]

Backprop in Practice

- Use ReLU non-linearities (tanh and logistic are falling out of favor)
- Use cross-entropy loss for classification
- Use Stochastic Gradient Descent on minibatches
- Shuffle the training samples
- Normalize the input variables (zero mean, unit variance)
- Schedule to decrease the learning rate
- Use a bit of L1 or L2 regularization on the weights (or a combination)
  - But it's best to turn it on after a couple of epochs
- Use “dropout” for regularization
  - Hinton et al 2012 http://arxiv.org/abs/1207.0580
- Lots more in [LeCun et al. “Efficient Backprop” 1998]
- Lots, lots more in “Neural Networks, Tricks of the Trade” (2012 edition) edited by G. Montavon, G. B. Orr, and K-R Müller (Springer)

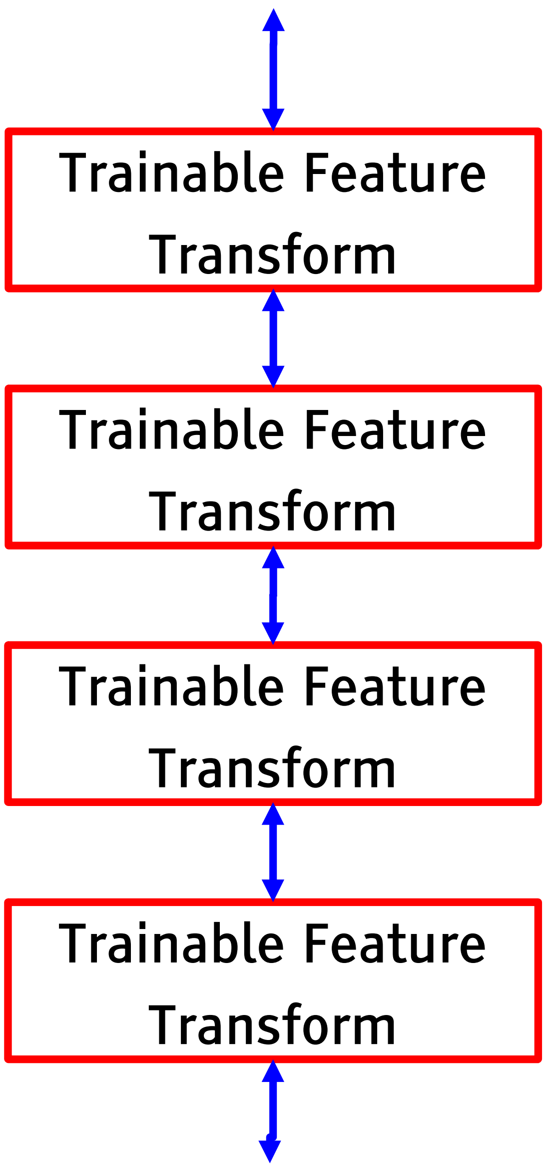
Challenges

Future Challenges

- Integrated feed-forward and feedback
  - Deep Boltzmann machine do this, but there are issues of scalability.
- Integrating supervised and unsupervised learning in a single algorithm
  - Again, deep Boltzmann machines do this, but....
- Integrating deep learning and structured prediction ("reasoning")
  - This has been around since the 1990's but needs to be revived
- Learning representations for complex reasoning
  - "recursive" networks that operate on vector space representations of knowledge [Pollack 90's] [Bottou 2010] [Socher, Manning, Ng 2011]
- Representation learning in natural language processing
  - [Y. Bengio 01],[Collobert Weston 10], [Mnih Hinton 11] [Socher 12]
- Better theoretical understanding of deep learning and convolutional nets
  - e.g. Stephane Mallat's "scattering transform", work on the sparse representations from the applied math community....

Integrating Feed-Forward and Feedback

- Marrying feed-forward convolutional nets with generative “deconvolutional nets”
  - Deconvolutional networks
    - [Zeiler-Graham-Fergus ICCV 2011]
- Feed-forward/Feedback networks allow reconstruction, multimodal prediction, restoration, etc...
  - Deep Boltzmann machines can do this, but there are scalability issues with training



Integrating Deep Learning and Structured Prediction

- Integrating deep learning and structured prediction is a very old idea
  - In fact, it predates structured prediction
- Globally-trained convolutional-net + graphical models
  - trained discriminatively at the word level
  - Loss identical to CRF and structured perceptron
  - Compositional movable parts model
- A system like this was reading 10 to 20% of all the checks in the US around 1998

[LeCun, Bottou, Bengio, Haffner "Gradient-Based Learning Applied to Document Recognition" Proceedings of the IEEE, 1998]