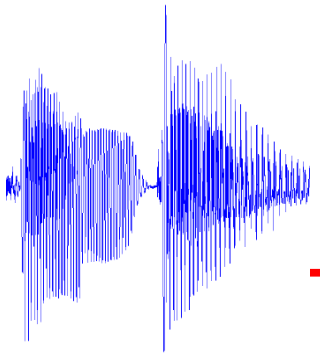
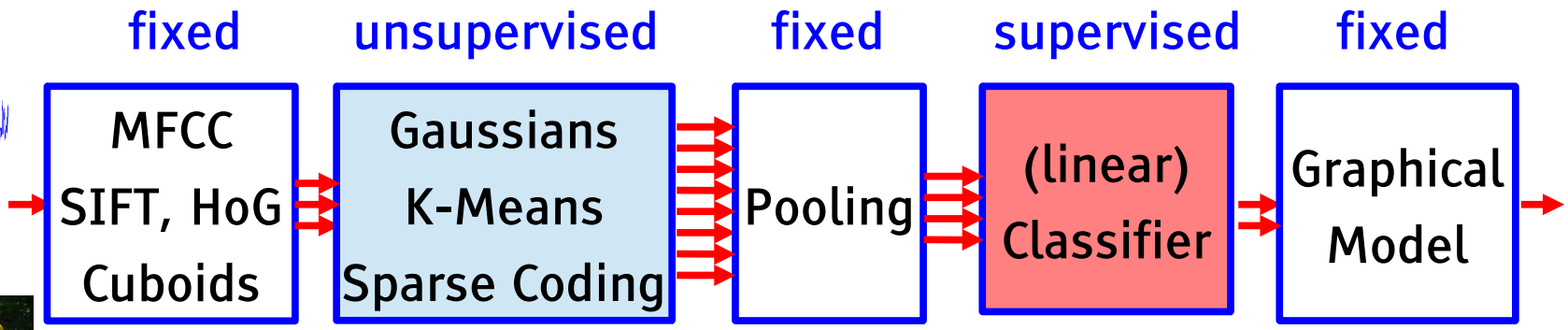
Deep Learning & Structured Prediction

Yann LeCun
Facebook AI Research &
Center for Data Science, NYU
yann@cs.nyu.edu
http://yann.lecun.com

Architecture of "Classical" Recognition Systems

- "Classic" architecture for pattern recognition
  - Speech, and Object recognition (until recently)
  - Handwriting recognition (long ago)
  - Graphical model has latent variables (locations of parts)

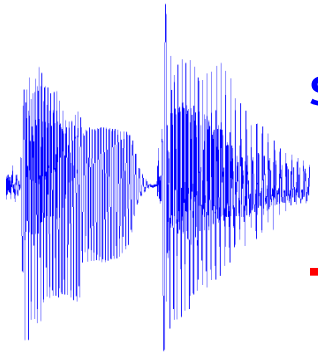


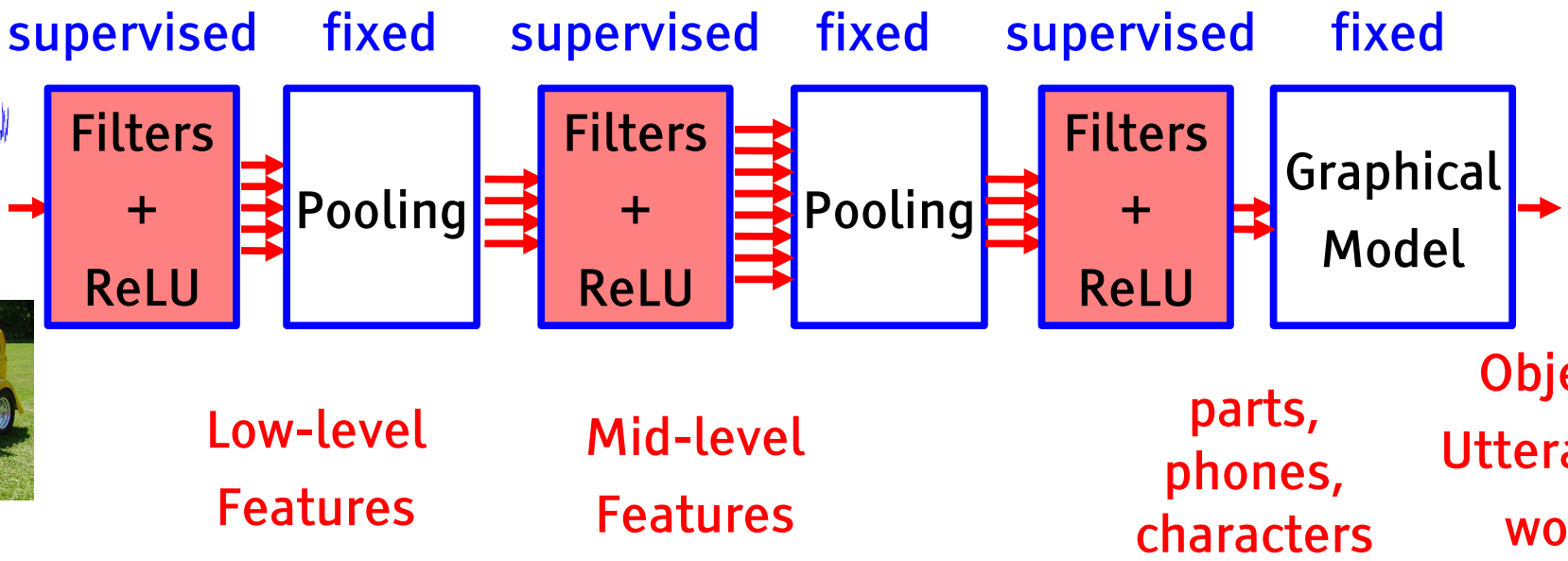


Architecture of Deep Learning-Based Recognition Systems

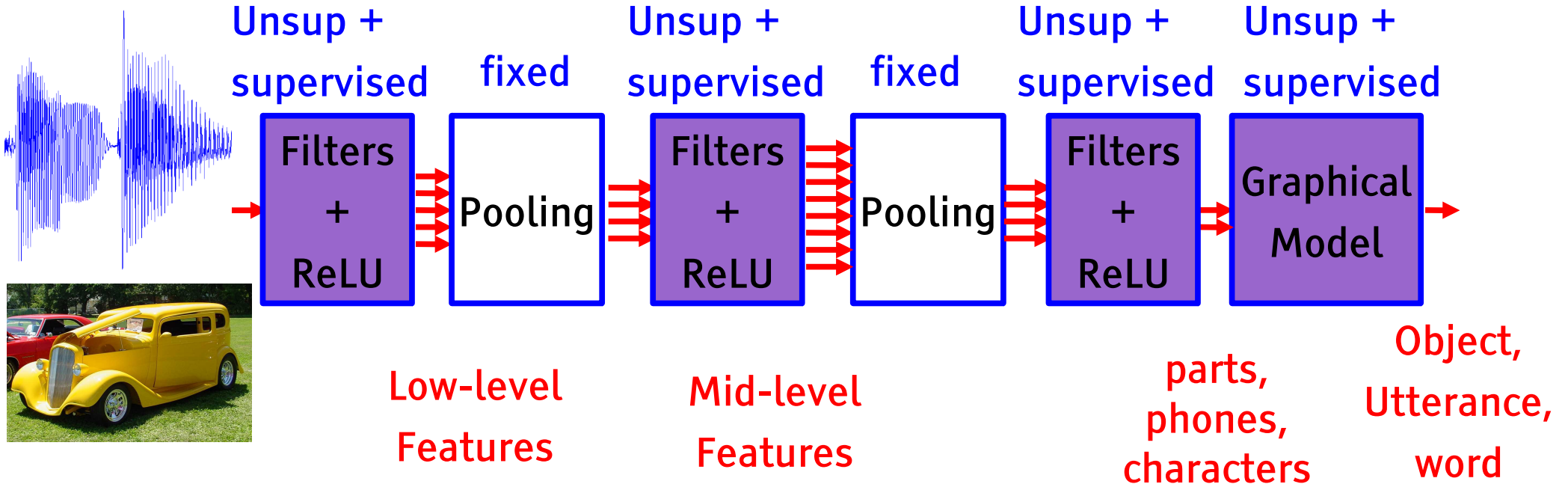
Y LeCun

- "Deep" architecture for pattern recognition
  - Speech, and Object recognition (recently)
  - Handwriting recognition (since the 1990s)
  - Convolutional Net with optional Graphical Model on top
  - Trained purely supervised
  - Graphical model has latent variables (locations of parts)



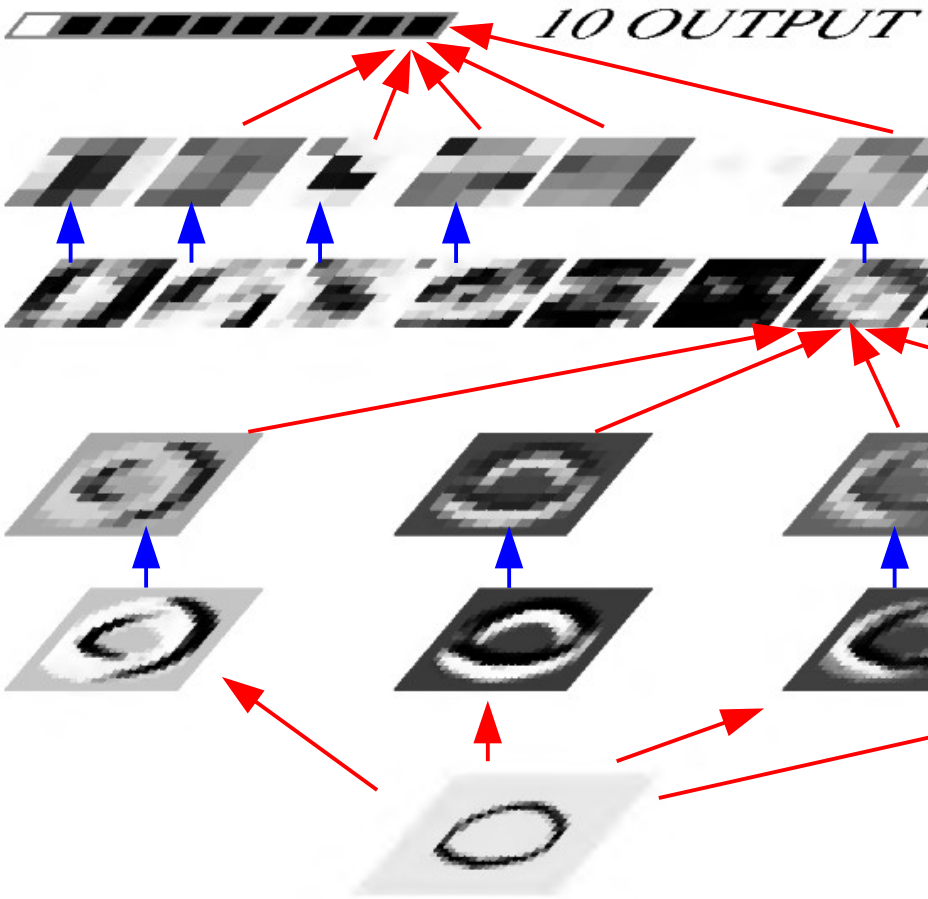


- Globally-trained deep architecture
  - Handwriting recognition (since the mid 1990s)
  - All the modules are trained with a combination of unsupervised and supervised learning
  - **End-to-end training == deep structured prediction**




Trained Features &
Fixed Post-processing
ConvNets

Convolutional Network



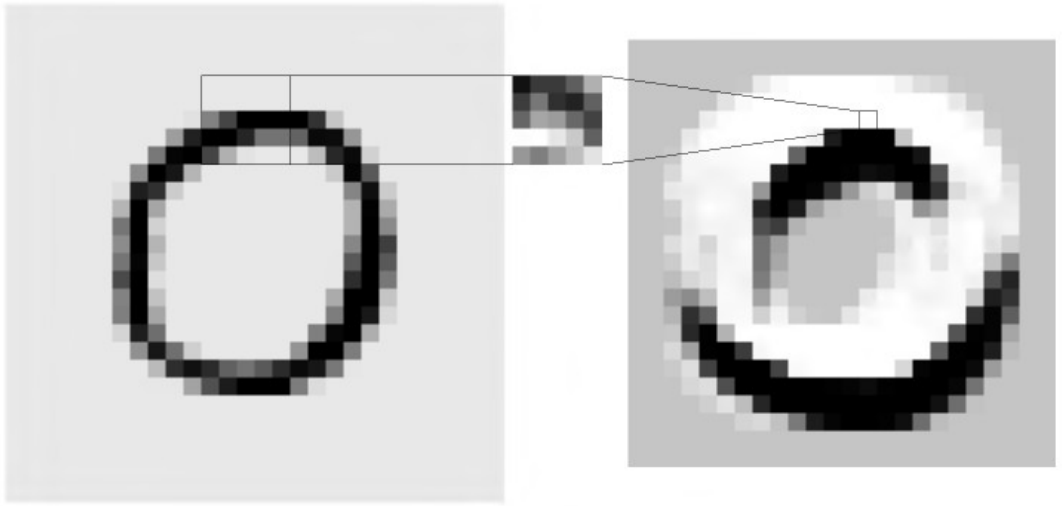
Filter Bank +non-linearity

Pooling

Filter Bank +non-linearity

Pooling

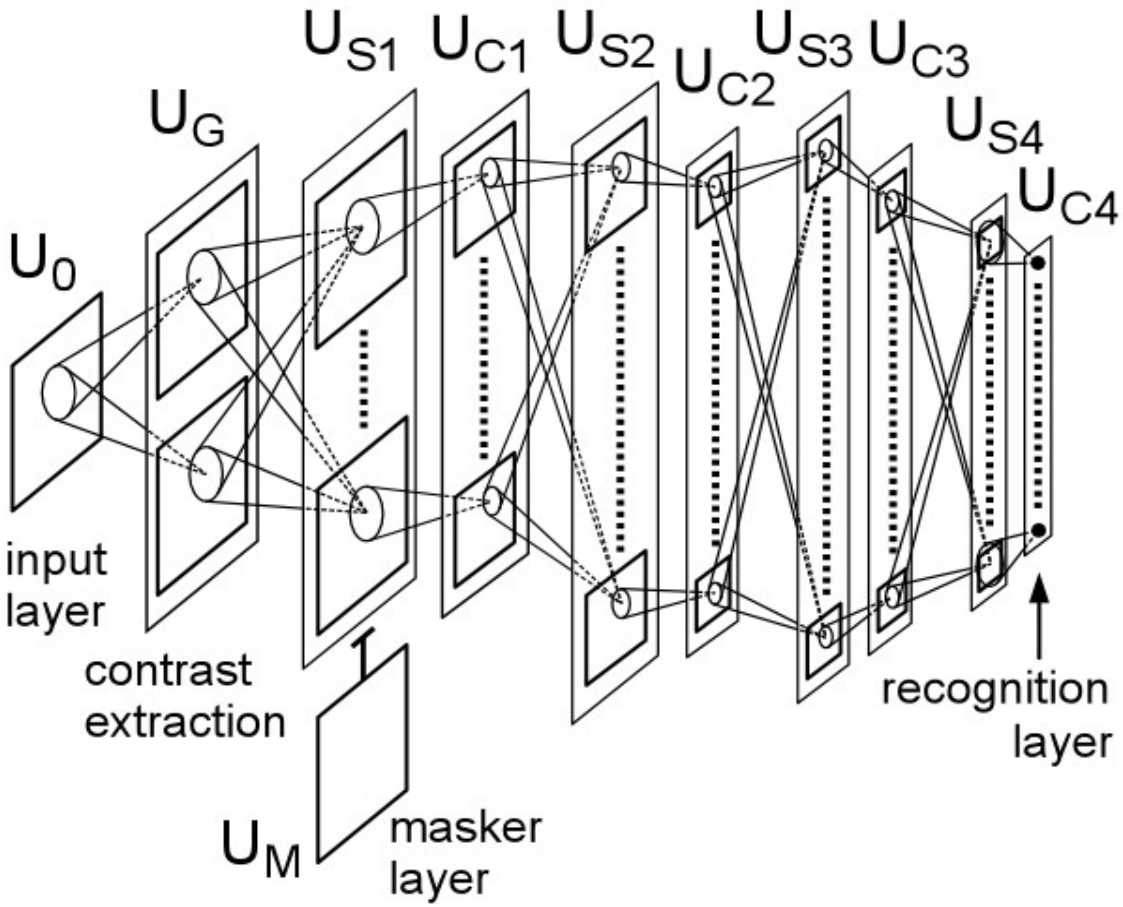
Filter Bank +non-linearity

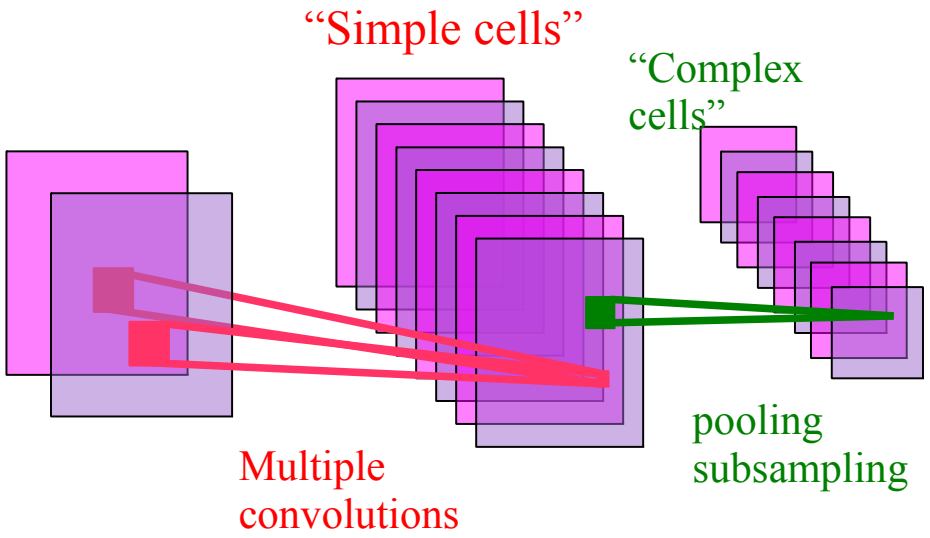


[LeCun et al. NIPS 1989]

Early Hierarchical Feature Models for Vision

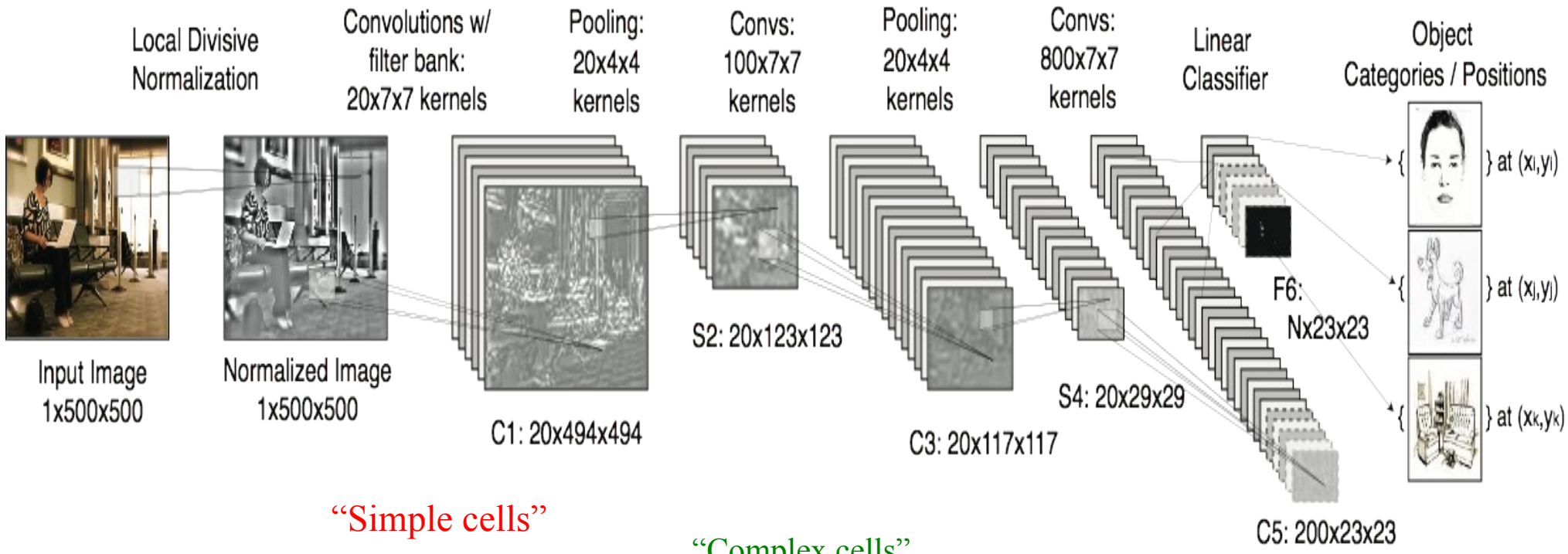
- [Hubel & Wiesel 1962]:
  - simple cells detect local features
  - complex cells "pool" the outputs of simple cells within a retinotopic neighborhood.

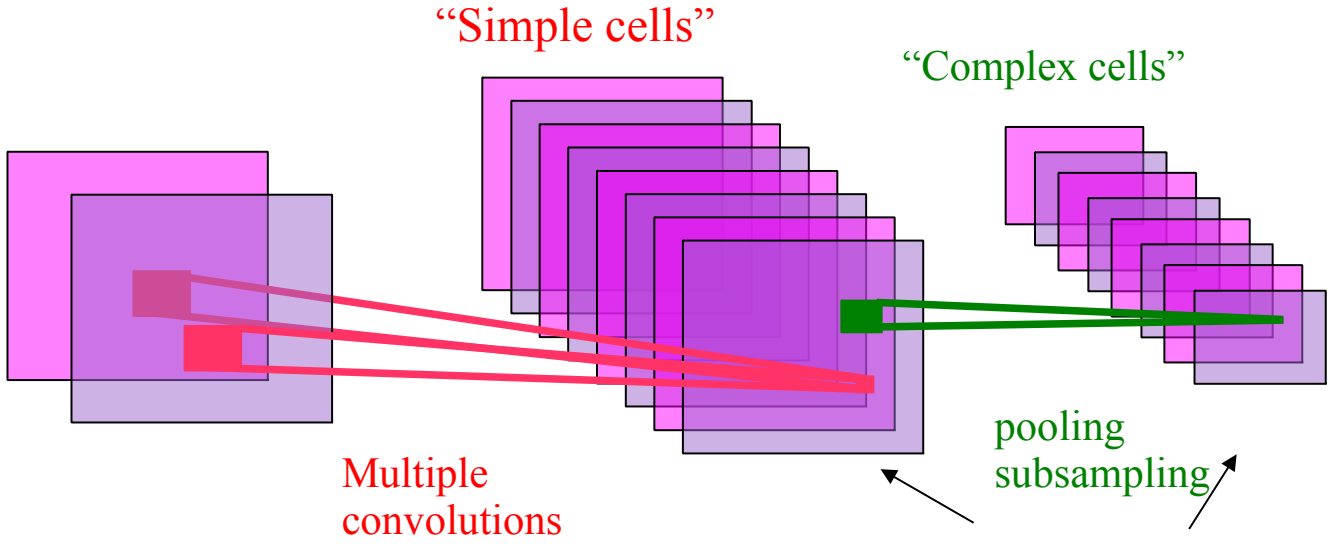




Cognitron & Neocognitron [Fukushima 1974-1982]

The Convolutional Net Model
(Multistage Hubel-Wiesel system)



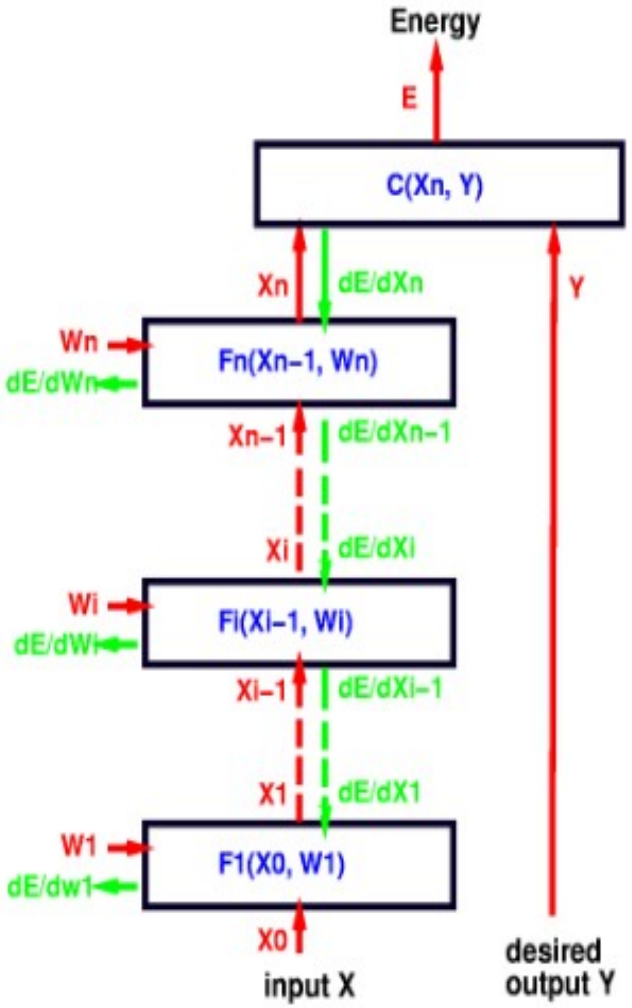


- Training is supervised
- With stochastic gradient descent

[LeCun et al. 89]
[LeCun et al. 98]

Supervised Training: Stochastic (Sub)Gradient Optimization

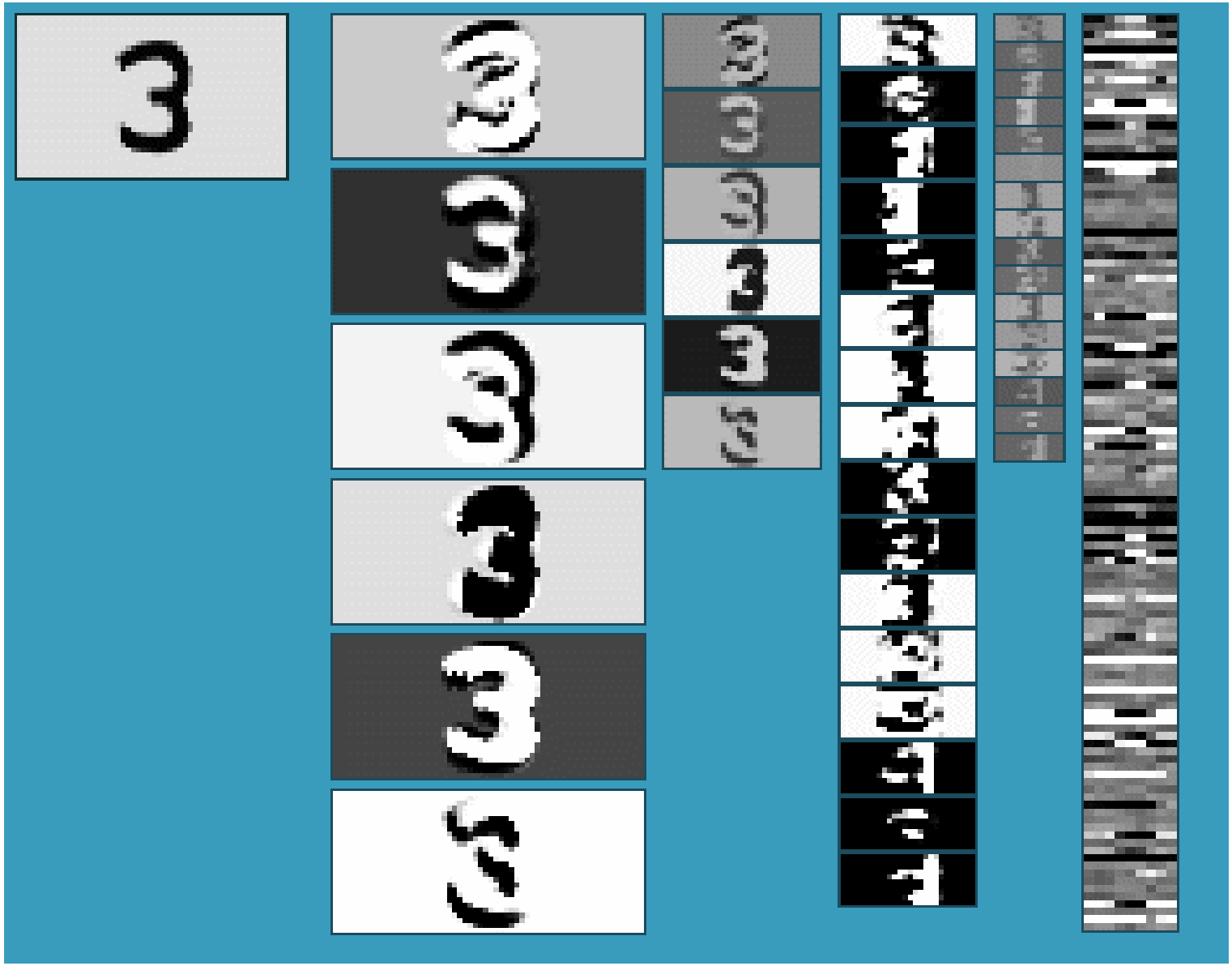
To compute all the derivatives, we use a backward sweep called the **back-propagation algorithm** that uses the recurrence equation for $\frac{\partial E}{\partial X_i}$



- $\frac{\partial E}{\partial X_n} = \frac{\partial C(X_n, Y)}{\partial X_n}$
- $\frac{\partial E}{\partial X_{n-1}} = \frac{\partial E}{\partial X_n} \frac{\partial F_n(X_{n-1}, W_n)}{\partial X_{n-1}}$
- $\frac{\partial E}{\partial W_n} = \frac{\partial E}{\partial X_n} \frac{\partial F_n(X_{n-1}, W_n)}{\partial W_n}$
- $\frac{\partial E}{\partial X_{n-2}} = \frac{\partial E}{\partial X_{n-1}} \frac{\partial F_{n-1}(X_{n-2}, W_{n-1})}{\partial X_{n-2}}$
- $\frac{\partial E}{\partial W_{n-1}} = \frac{\partial E}{\partial X_{n-1}} \frac{\partial F_{n-1}(X_{n-2}, W_{n-1})}{\partial W_{n-1}}$
- ....etc, until we reach the first module.
- we now have all the $\frac{\partial E}{\partial W_i}$ for $i \in [1, n]$.

Convolutional Network (vintage 1990)

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



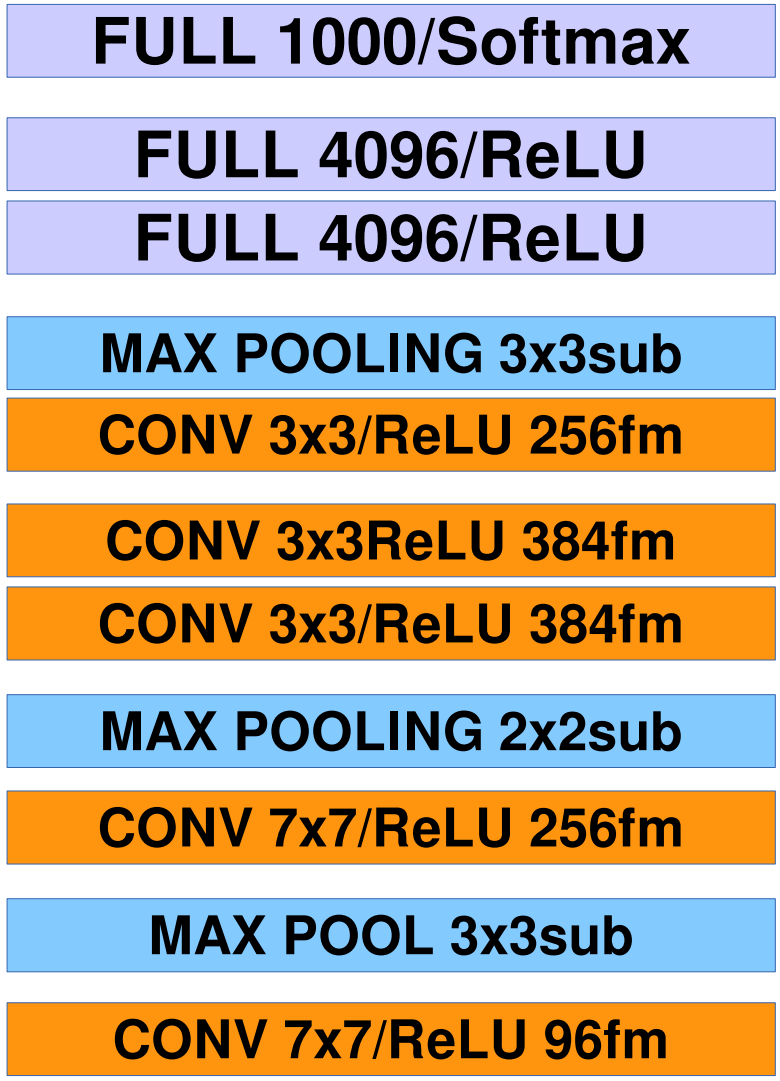
Curved manifold

Flatter manifold

NYU ConvNet Trained on ImageNet: OverFeat

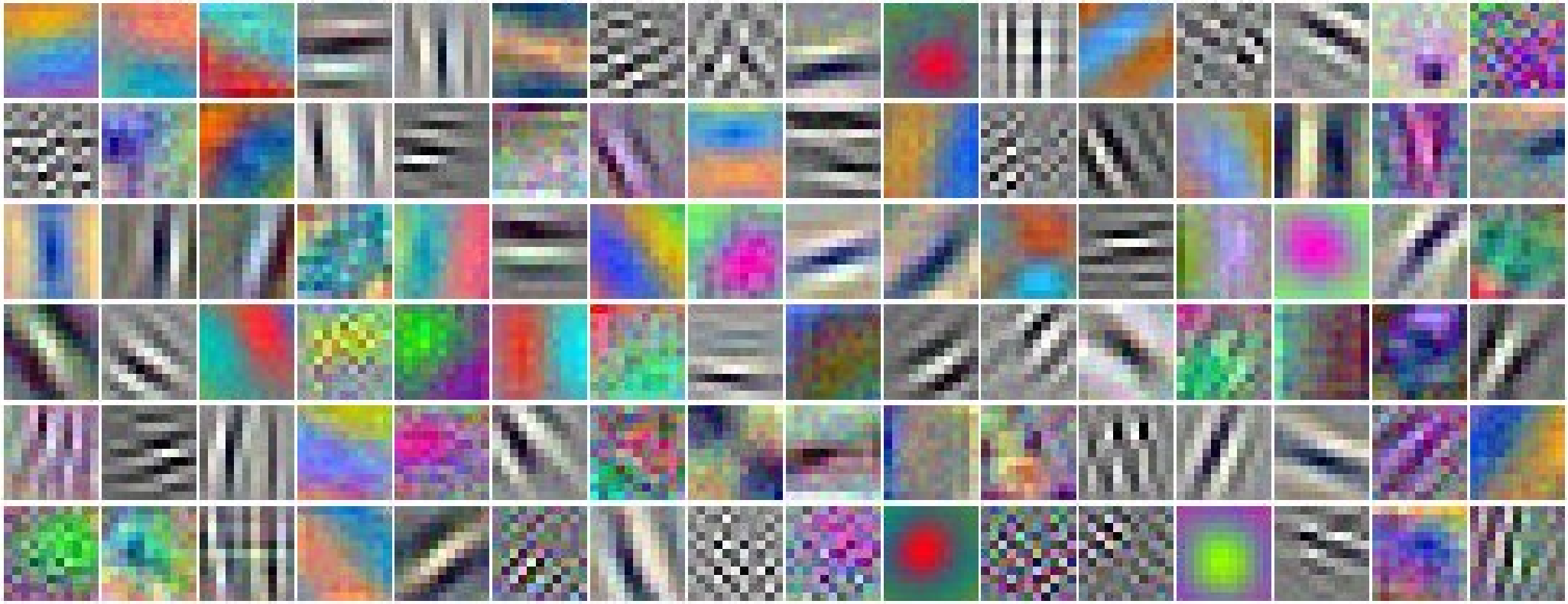
Y LeCun

- [Sermanet et al. arXiv:1312.6229]
- Trained on GPU using Torch7
- Uses a number of new tricks
- Classification 1000 categories:
  - 13.8% error (top 5) with an ensemble of 7 networks (Krizhevsky: 15%)
  - 15.4% error (top 5) with a single network (Krizhevksy: 18.2%)
- Classification+Localization
  - 30% error (Krizhevsky: 34%)
- Detection (200 categories)
  - 19% correct
- Dowloadable code (running, no training)
  - **Search for "overfeat NYU" on Google**
  - **http://cilvr.nyu.edu → software**



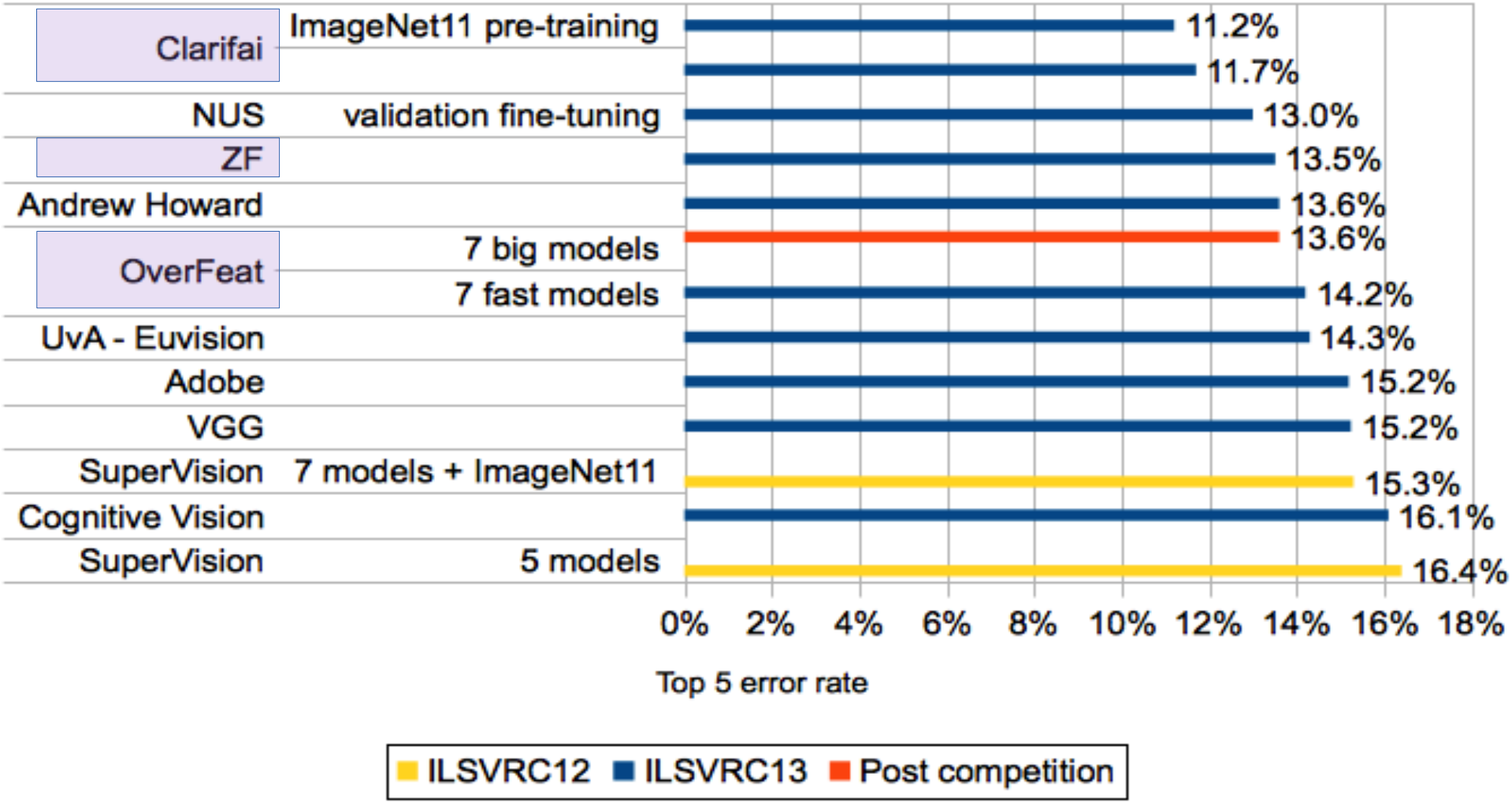
Kernels: Layer 1 (11x11)

- Layer 1: 3x96 kernels, RGB->96 feature maps, 11x11 Kernels, stride 4

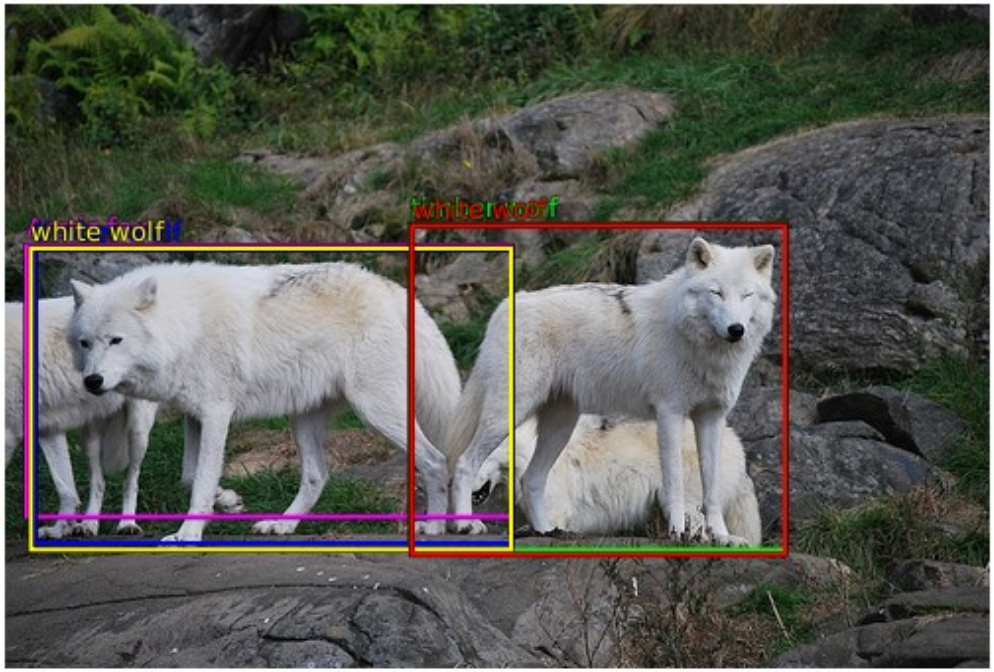


ImageNet 2013: Classification

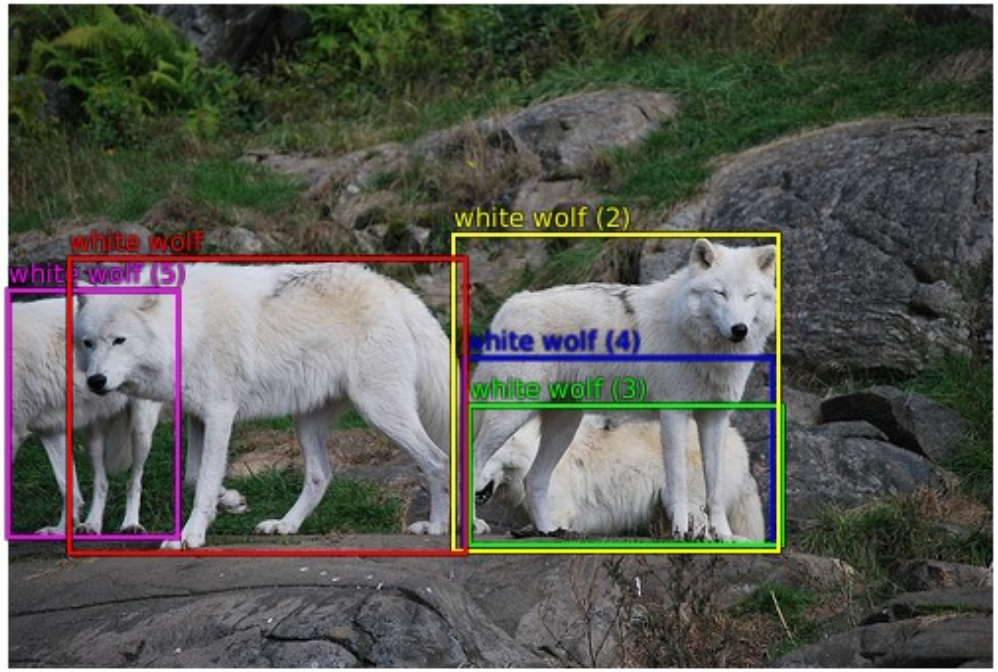
- Give the name of the dominant object in the image
- Top-5 error rates: if correct class is not in top 5, count as error
- (NYU Teams in Purple)



Classification+Localization. Results



Top 5:
white wolf
white wolf
timber wolf
timber wolf
Arctic fox

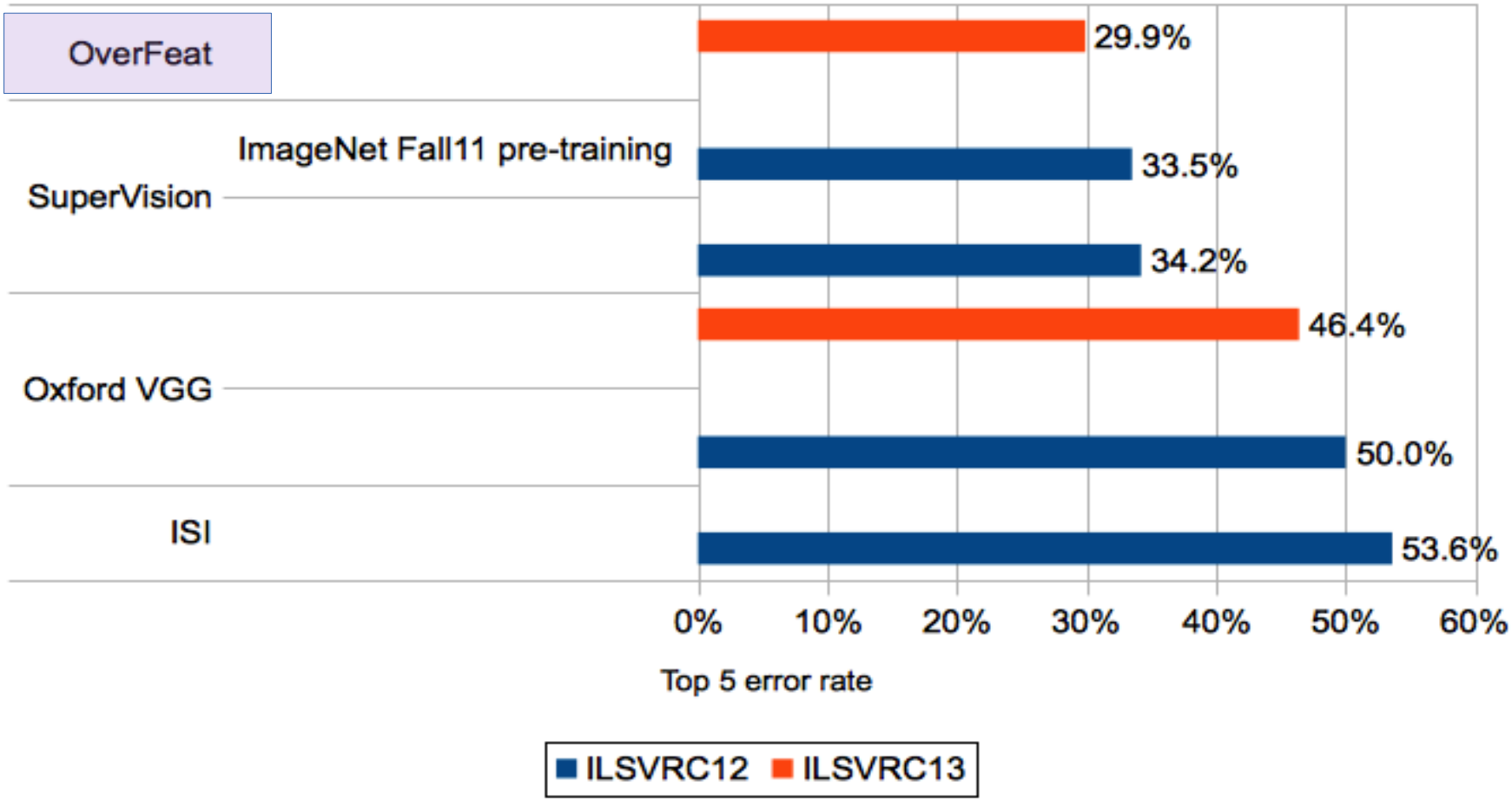


Groundtruth:
white wolf
white wolf (2)
white wolf (3)
white wolf (4)
white wolf (5)

ILSVRC2012_val_00000027.JPEG

Classification+Localization. Error Rates

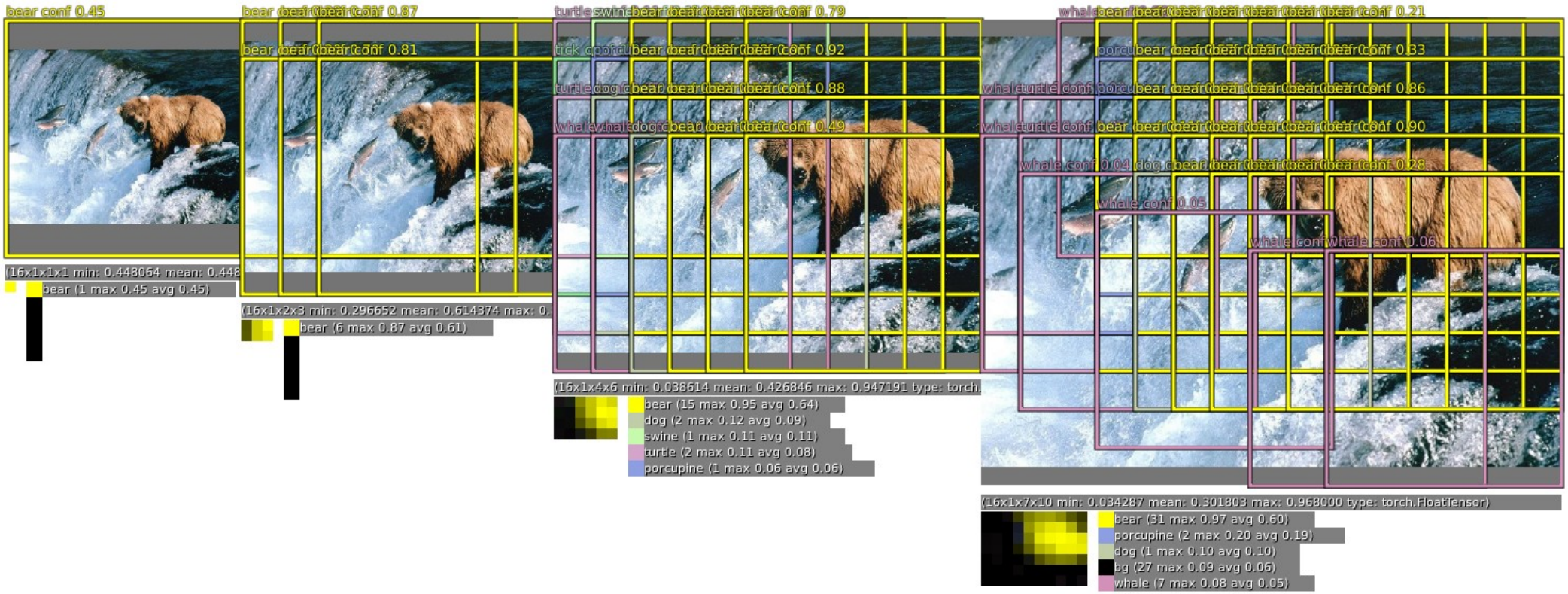
- It's best to propose several categories for the same window
  - One of them might be right



Classification + Localization: multiscale sliding window

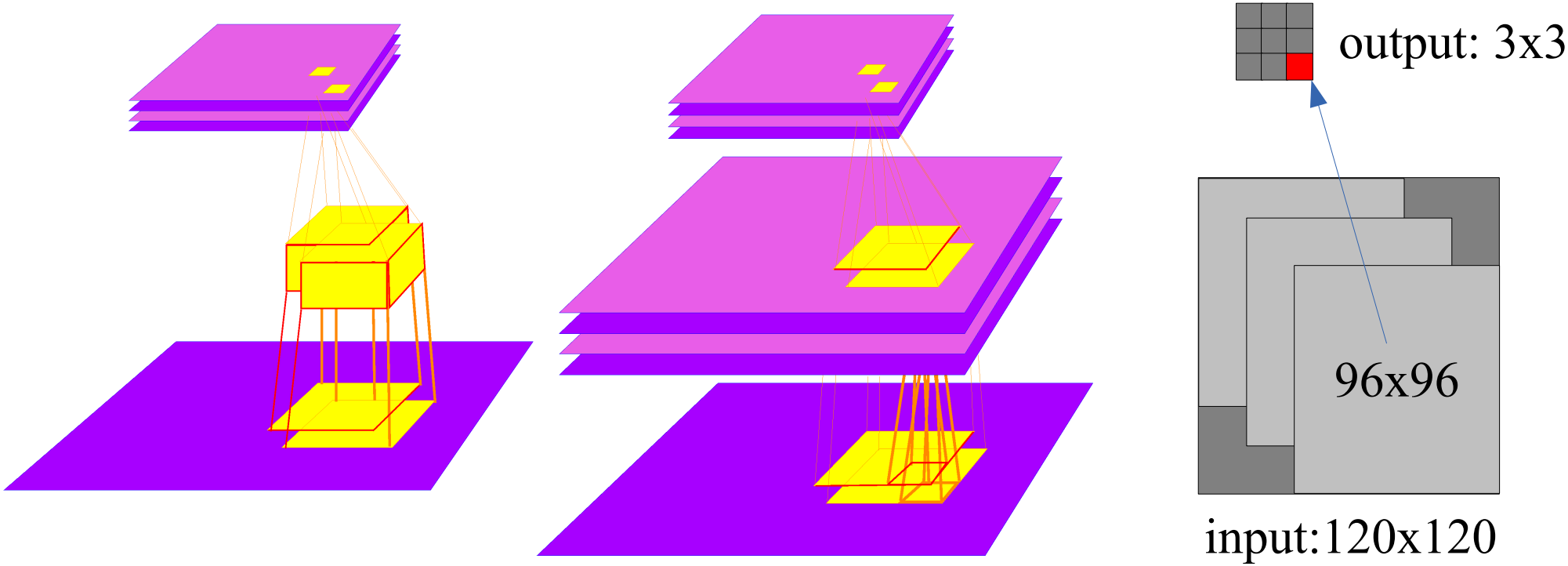
Y LeCun

- Apply convnet with a sliding window over the image at multiple scales
- Important note: it's very cheap to slide a convnet over an image
  - Just compute the convolutions over the whole image and replicate the fully-connected layers



Applying a ConvNet on Sliding Windows is Very Cheap!

Y LeCun

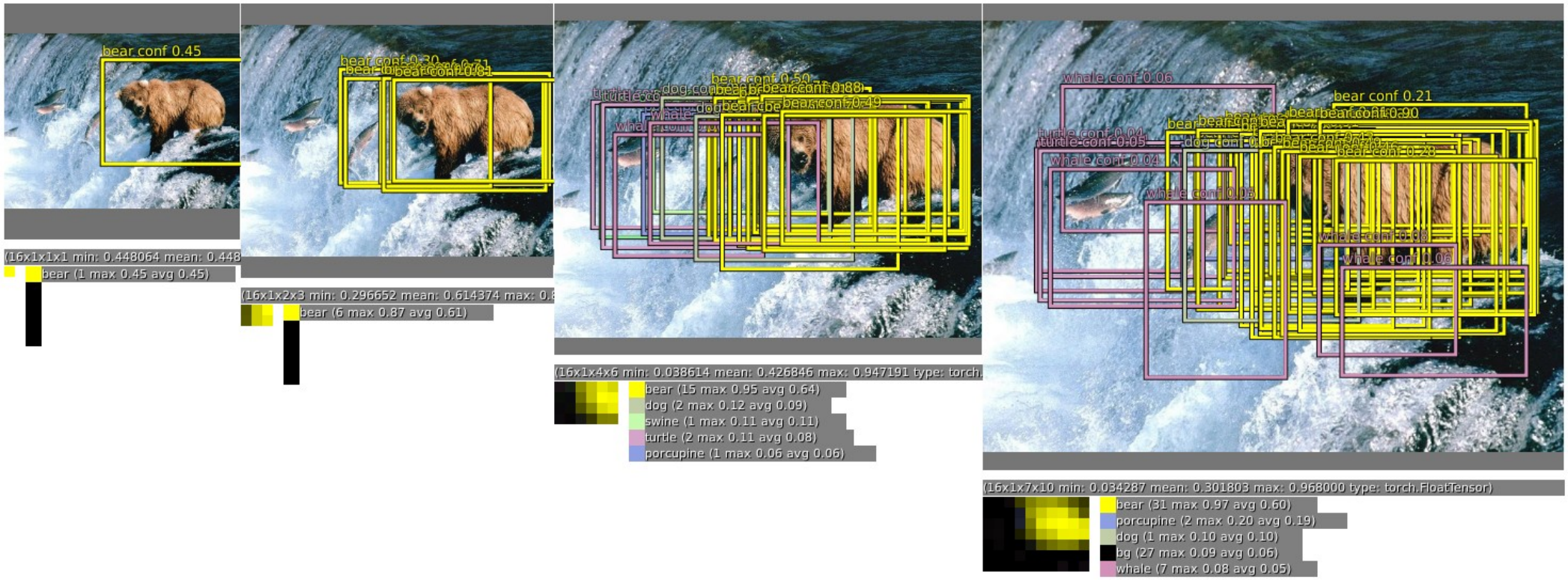


- Traditional Detectors/Classifiers must be applied to every location on a large input image, at multiple scales.
- Convolutional nets can replicated over large images very cheaply.
- Simply apply the convolutions to the entire image and spatially replicate the fully-connected layers

Classification + Localization: sliding window + bounding box regression

Y LeCun

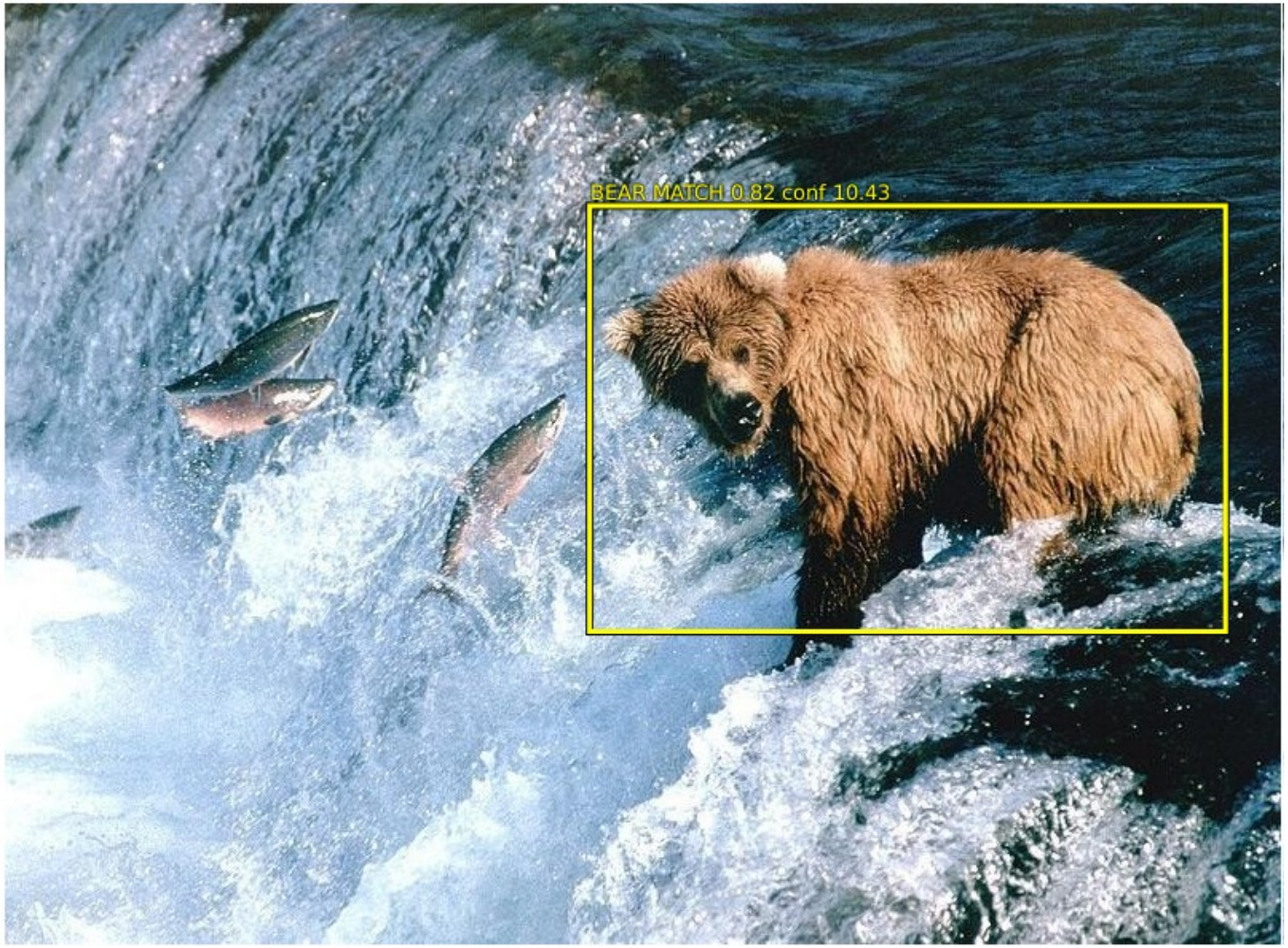
- Apply convnet with a sliding window over the image at multiple scales
- For each window, predict a class and bounding box parameters
  - Evenif the object is not completely contained in the viewing window, the convnet can predict where it thinks the object is.



Classification + Localization: sliding window + bounding box regression + bbox voting

Y LeCun

- Apply convnet with a sliding window over the image at multiple scales
- For each window, predict a class and bounding box parameters
- Compute an “average” bounding box, weighted by scores

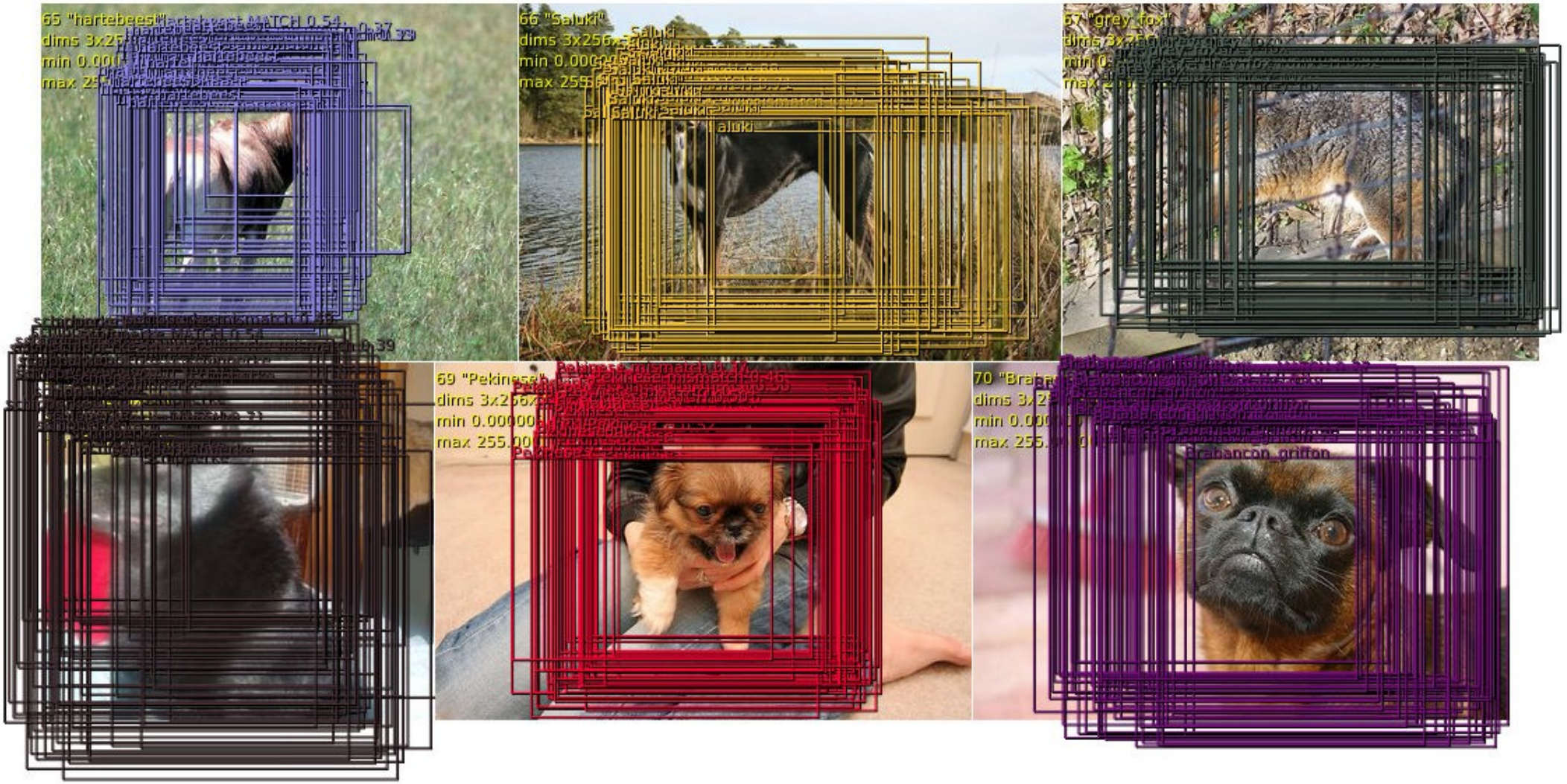


Localization: Sliding Window + bbox vote + multiscale

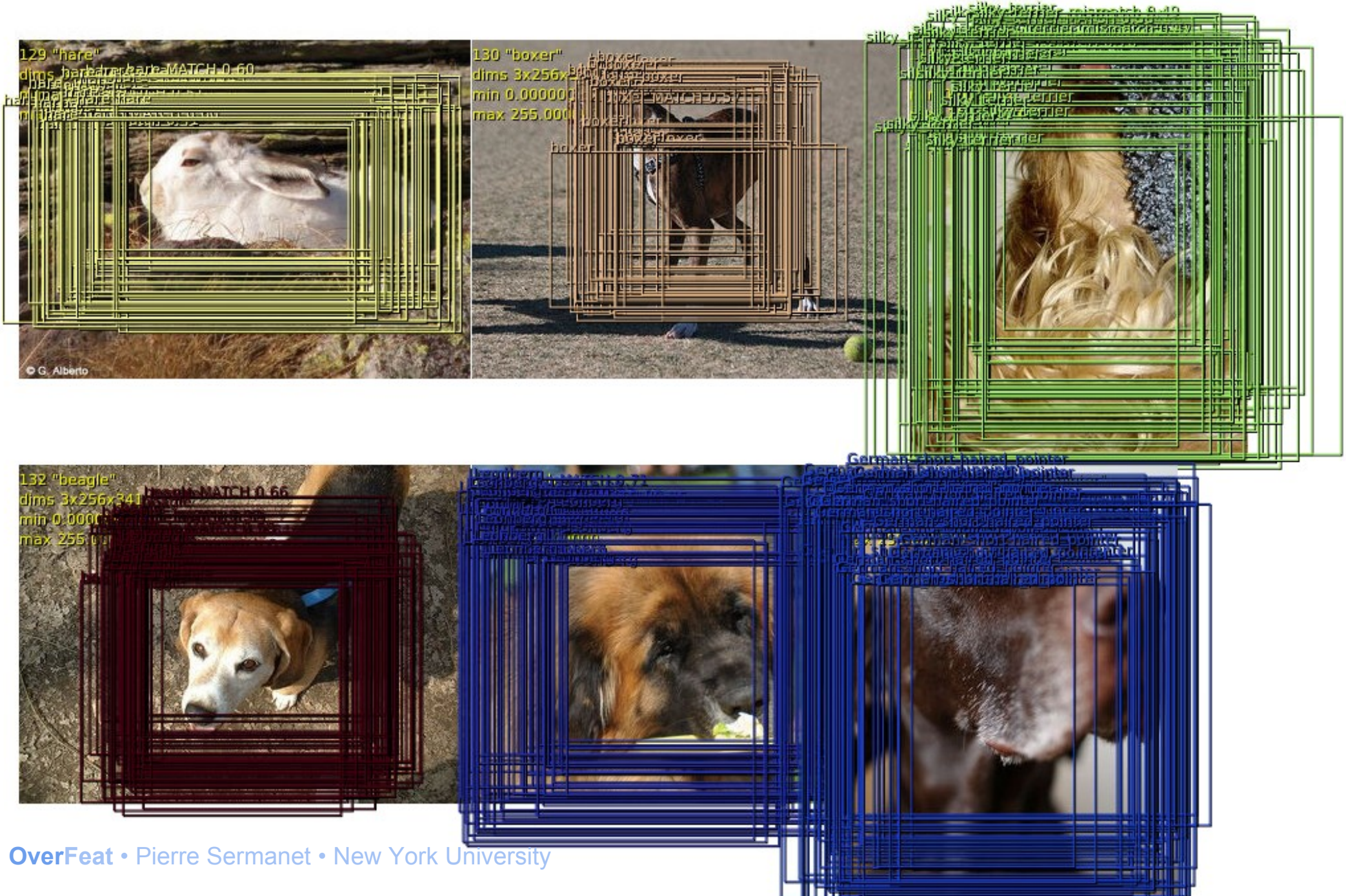
Y LeCun

Detection / Localization

Detection / Localization

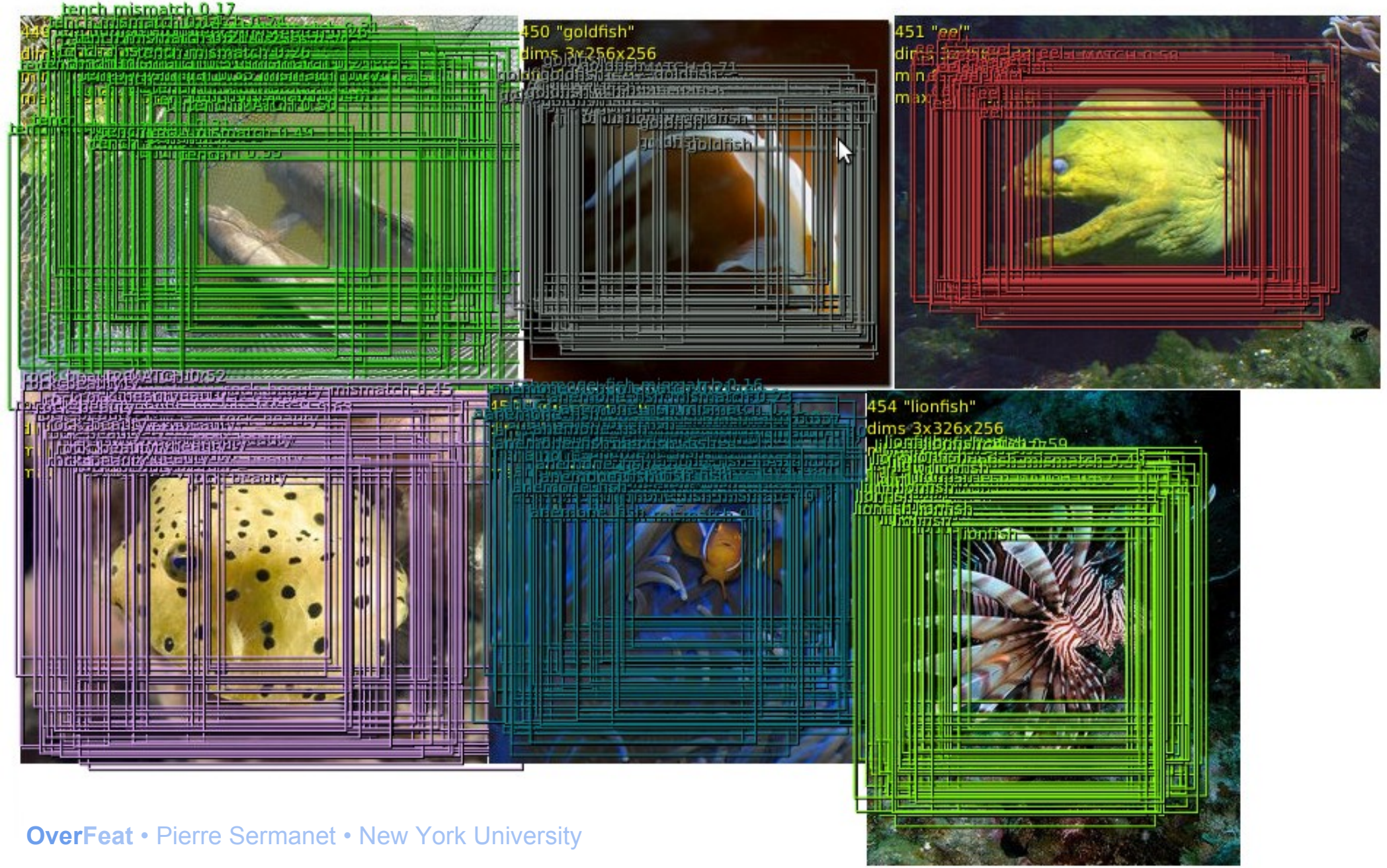


Detection / Localization



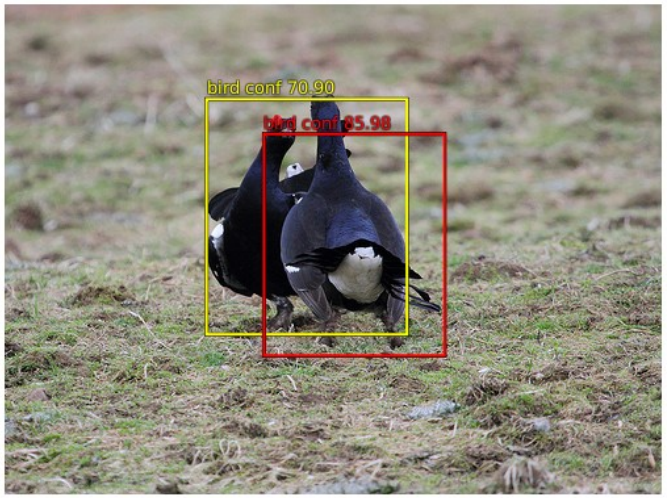
Detection / Localization

Detection / Localization

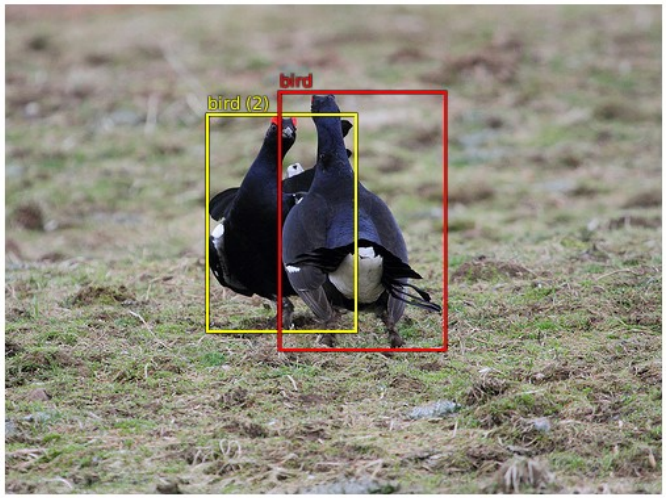


Detection: Examples

- 200 broad categories
- There is a penalty for false positives
- Some examples are easy some are impossible/ambiguous
- Some classes are well detected
  - Burritos?

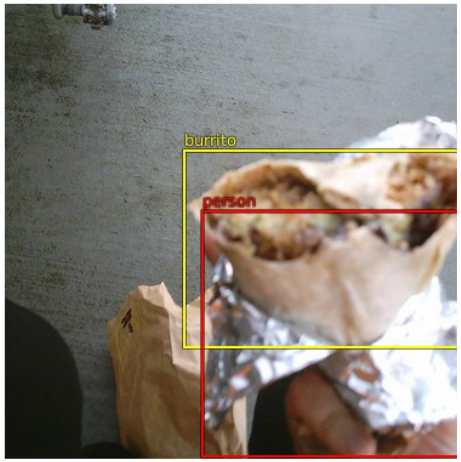


Top predictions:
bird (confidence 86.(
bird (confidence 70.9
ILSVRC2012_val_00001136.JPEG



Groundtruth:

Top predictions:
burrito (confidence 28.9)
ILSVRC2012_val_00000572.JPEG



Groundtruth:
person
burrito

Top predictions:
burrito (confidence 17.4)
ILSVRC2012_val_00000606.JPEG

Groundtruth:
burrito
burrito (2)

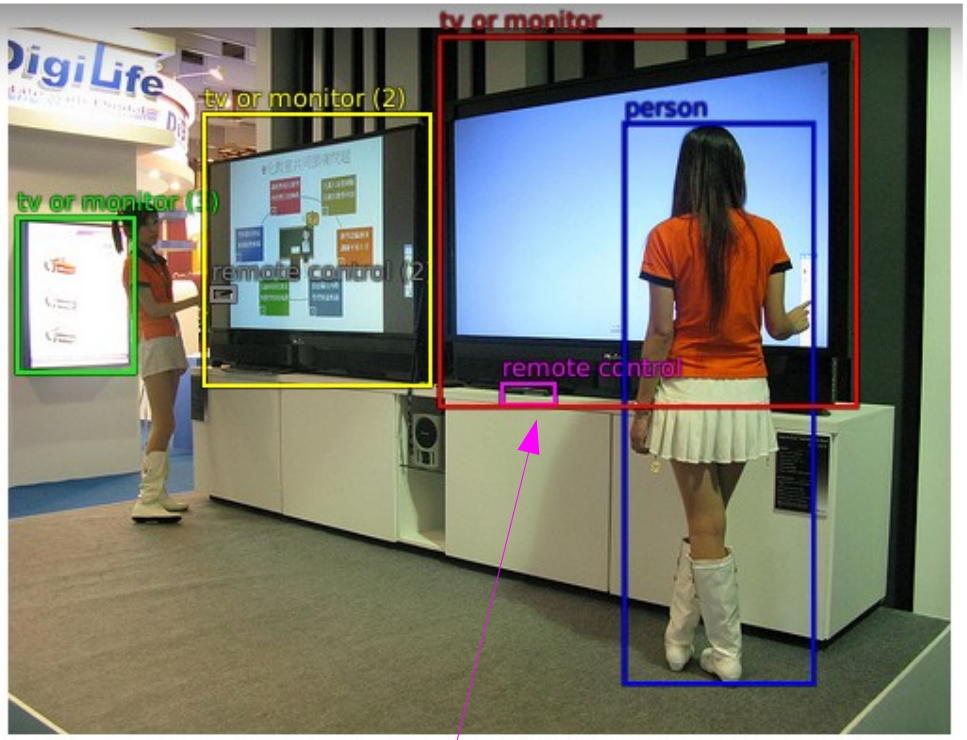
Detection: Examples

- Groundtruth is sometimes ambiguous or incomplete
- Large overlap between objects stops non-max suppression from working

Top predictions:

tv or monitor (confidence 11.5)
person (confidence 4.5)
miniskirt (confidence 3.1)

ILSVRC2012_val_00000119.JPEG

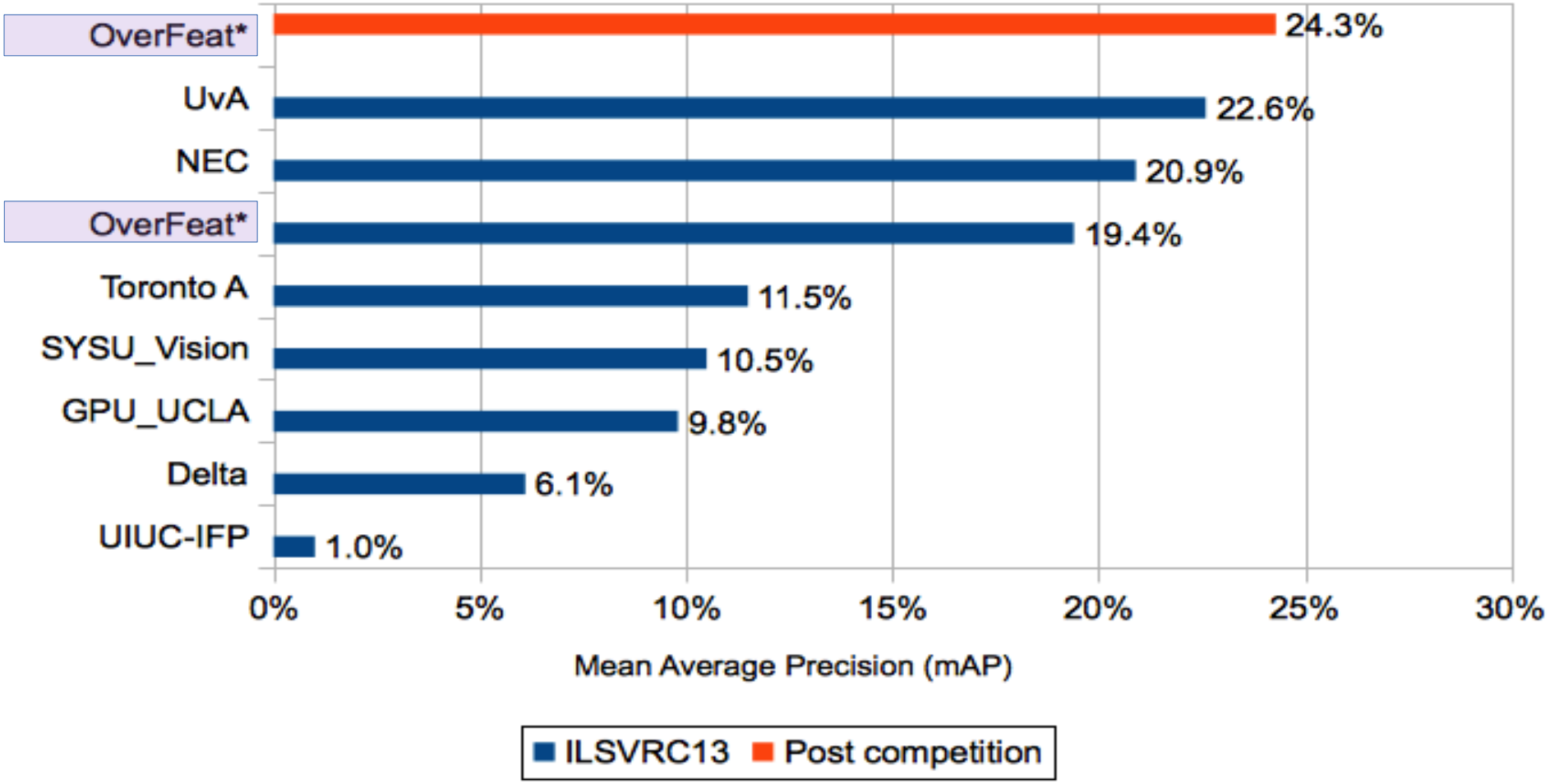


Groundtruth:

tv or monitor
tv or monitor (2)
tv or monitor (3)
person
remote control
remote control (2)

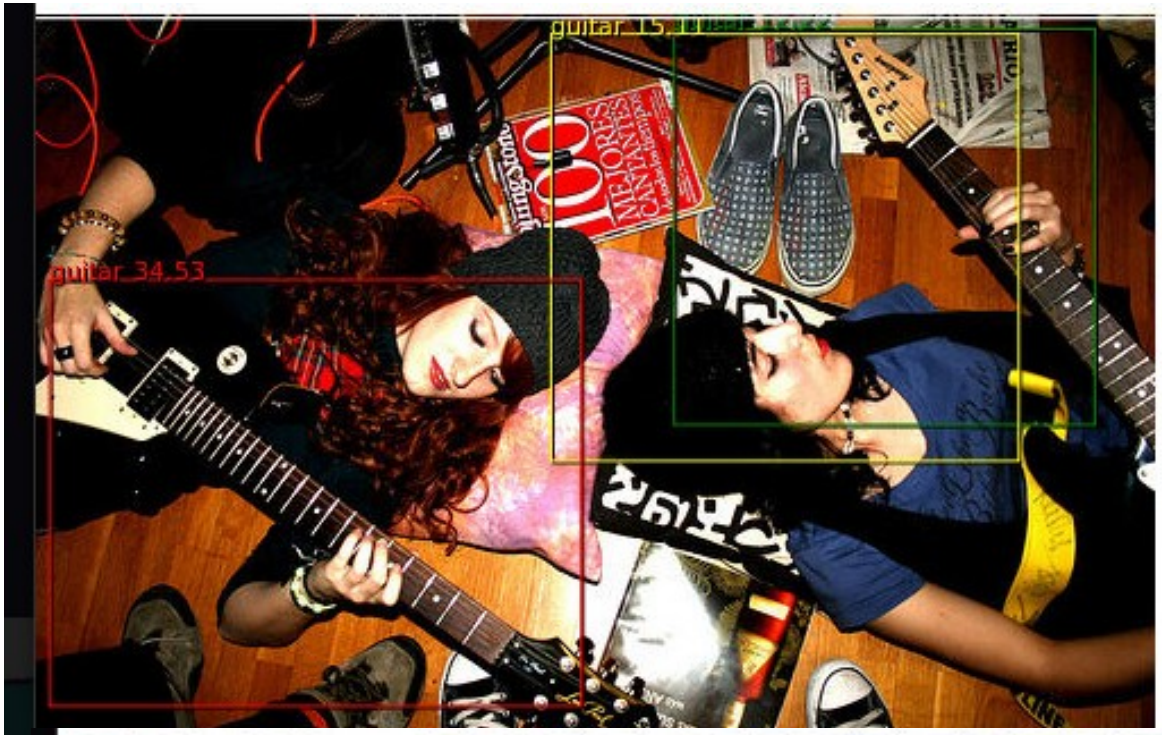
ImageNet 2013: Detection

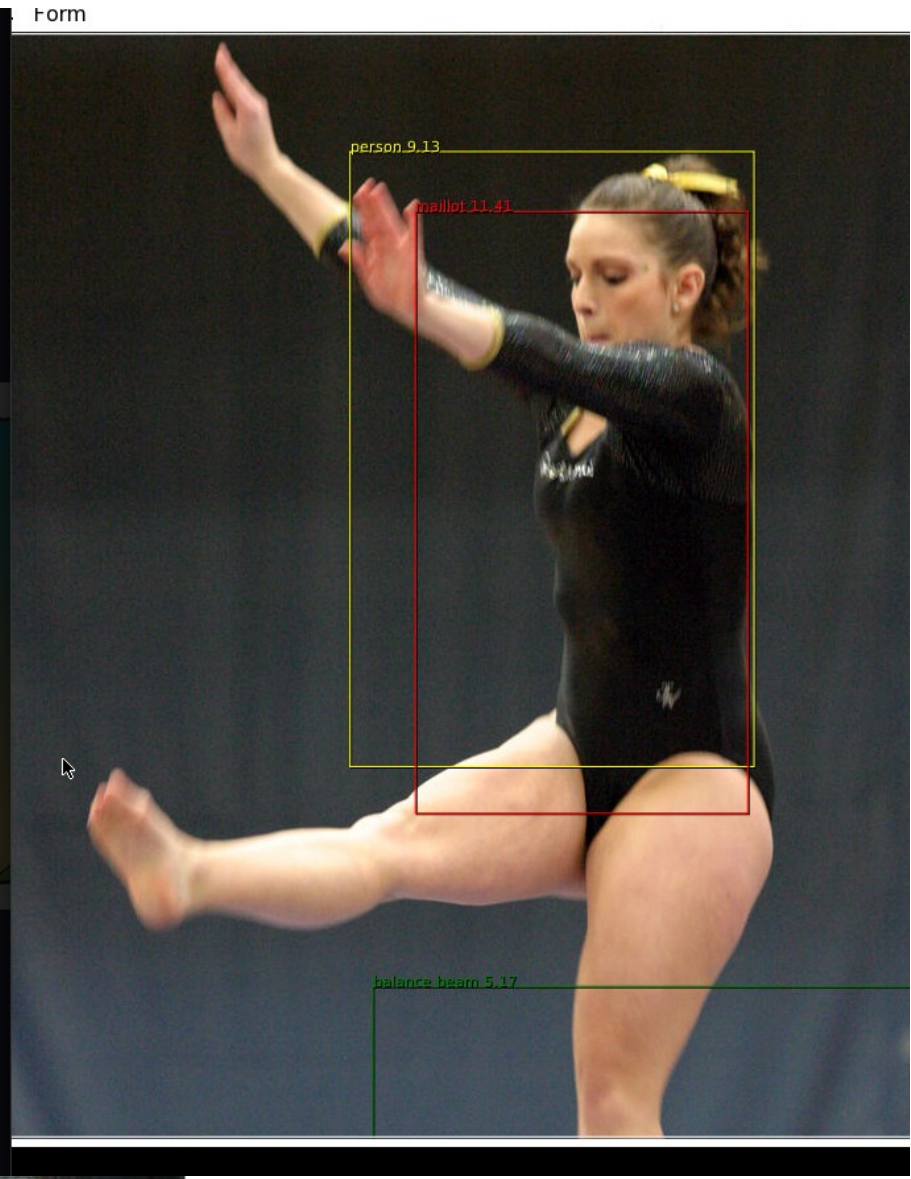
- 200 categories. System must give 5 bounding boxes with categories
- OverFeat: pre-trained on ImageNet 1K, fine-tuned on ImageNet Detection.
- Post-deadline result: 0.243 mean average precision



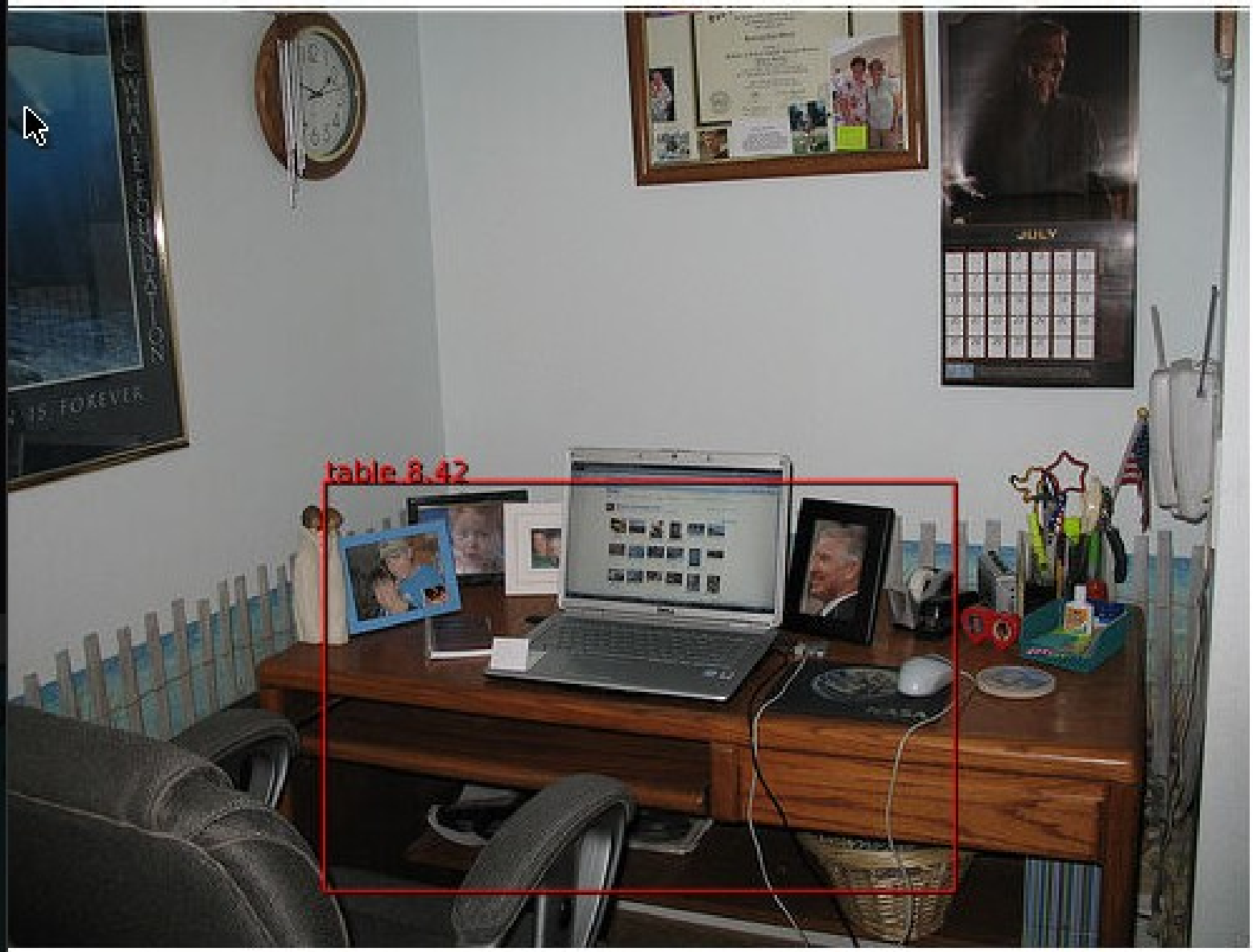
Results: pre-trained on ImageNet1K, fine-tuned on ImageNet Detection

Y LeCun



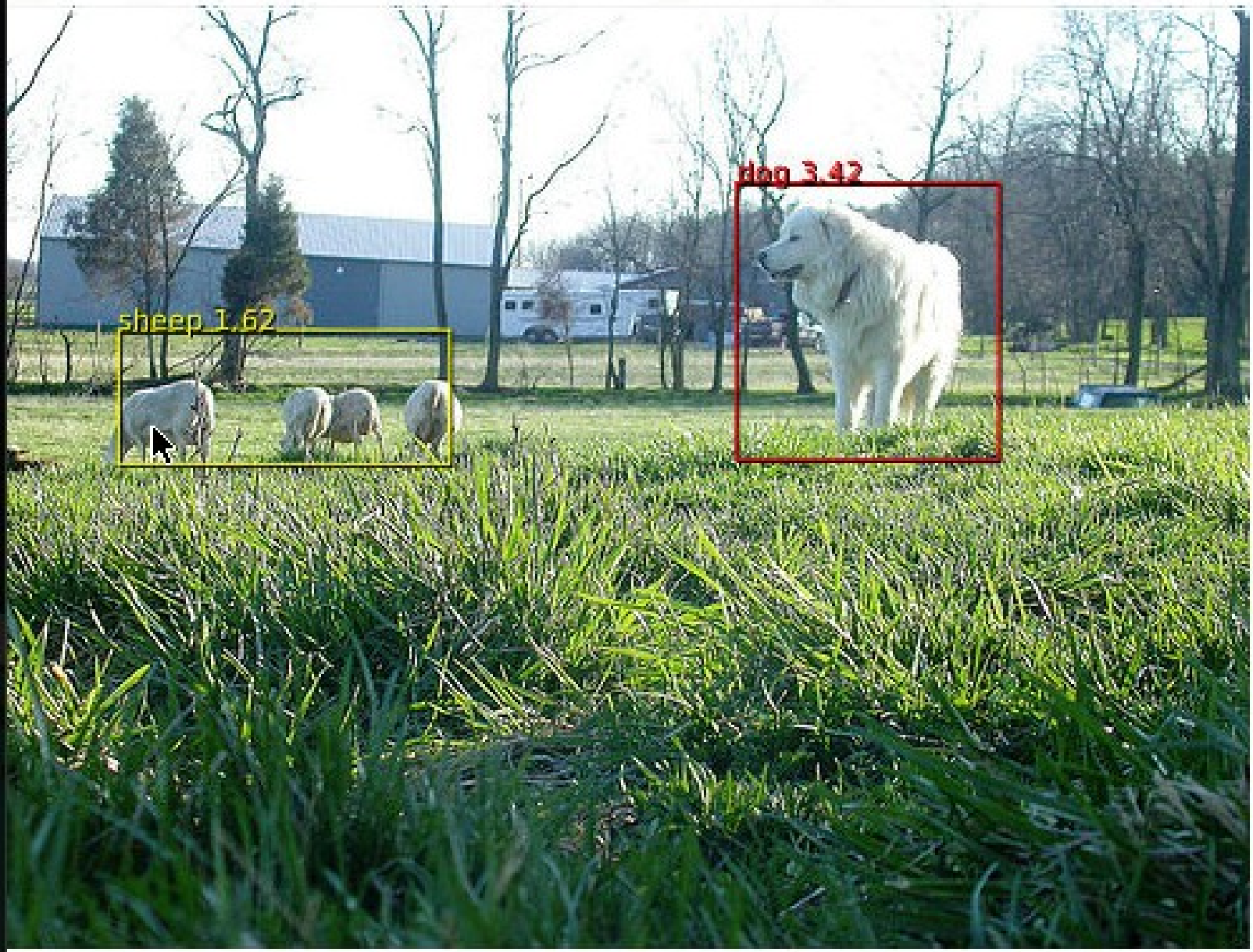


Detection Examples



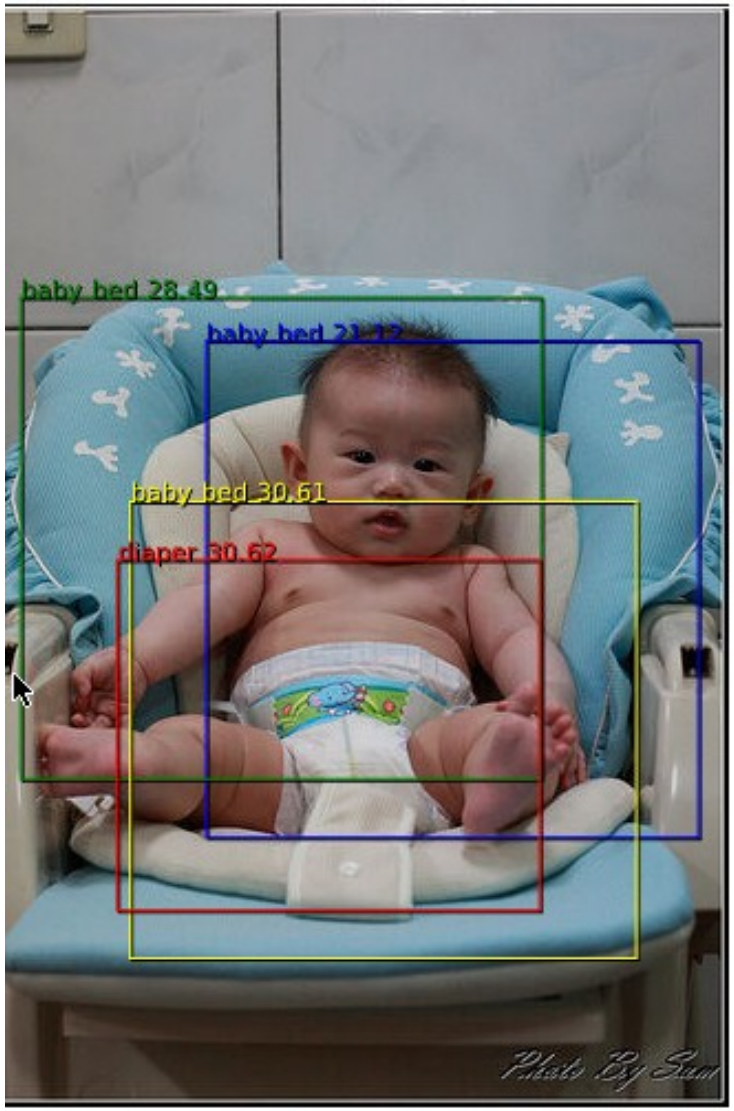
Detection Examples

Detection Examples



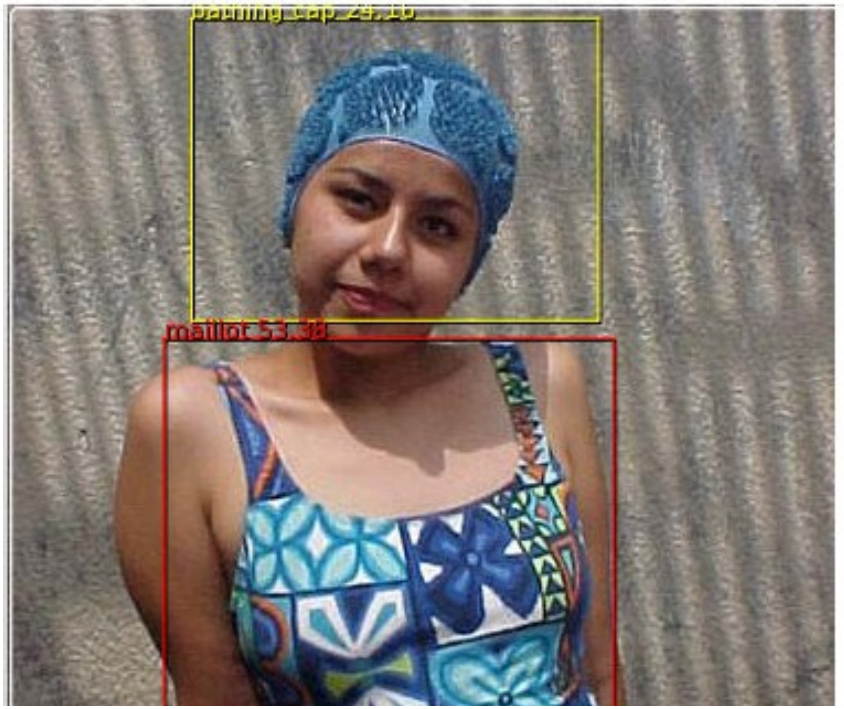
Detection Examples

Detection Examples

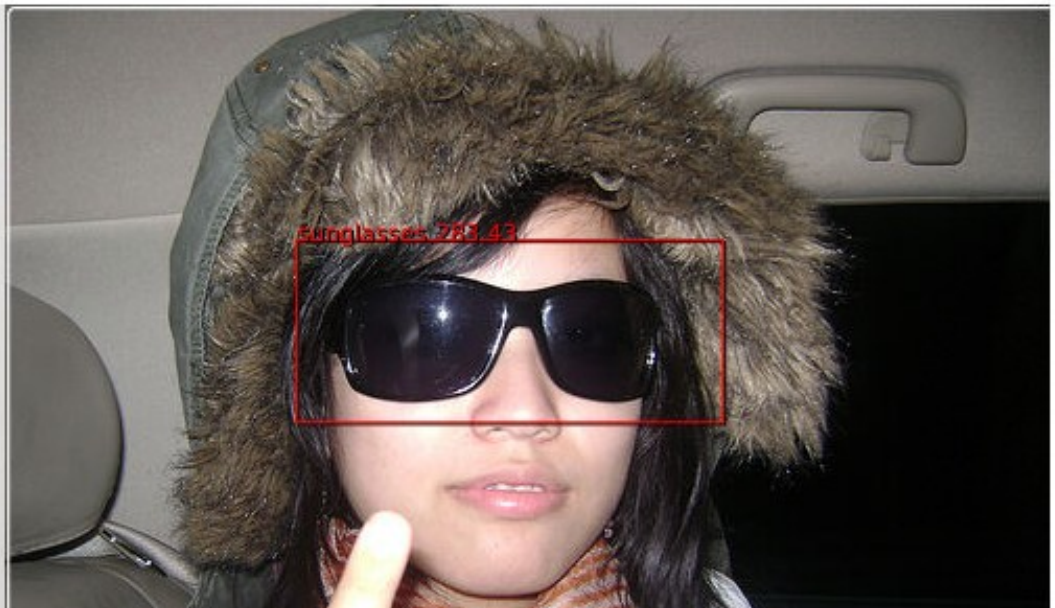


/home/snwiz/data/imagenet12/original/det/ILSVRC2013_DET_test/I
diaper conf 30.616426
baby bed conf 30.607744
baby bed conf 28.493934
baby bed conf 21.117424

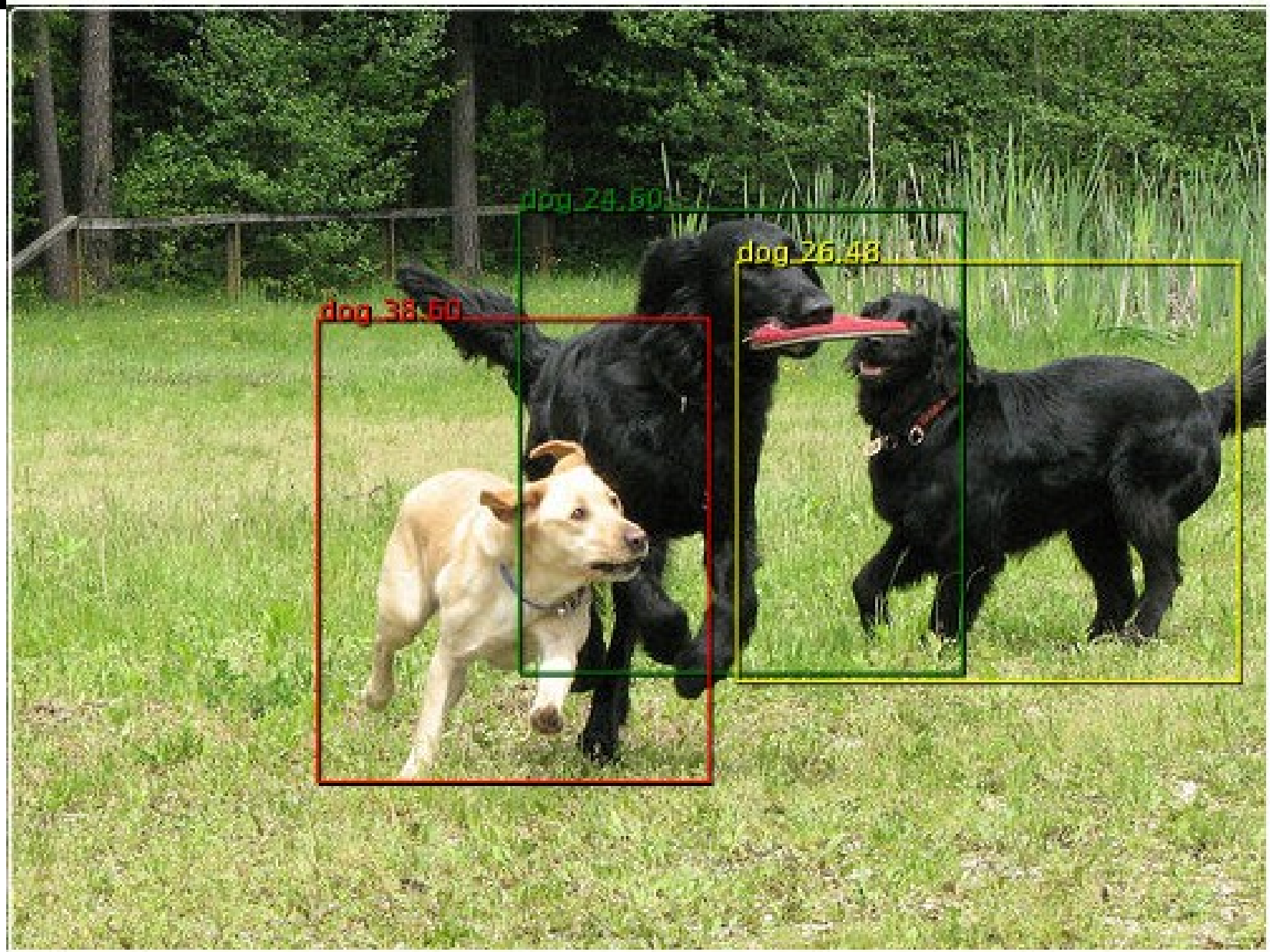
Form



Form



Detection Examples



/home/snwiz/data/imagenet12/original/det/ILSVRC2013_DET_test/ILSVRC2012_test_00000172.JPEG
dog conf 38.603936


WE PROVIDI
THIS SERVICE
AT YOUR
person 9.06
french horn 21.89

Detection: Difficult Examples

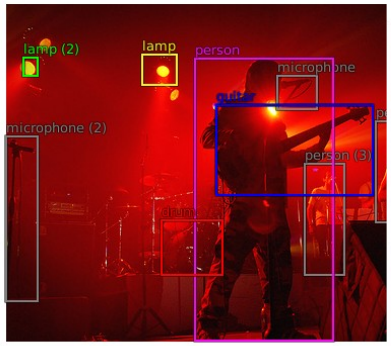
- Groundtruth is sometimes ambiguous or incomplete

Top predictions:
microwave (confidence 5.6)
refrigerator (confidence 2.5)

Groundtruth:
bowl
microwave

Top predictions:
person (confidence 6.0)

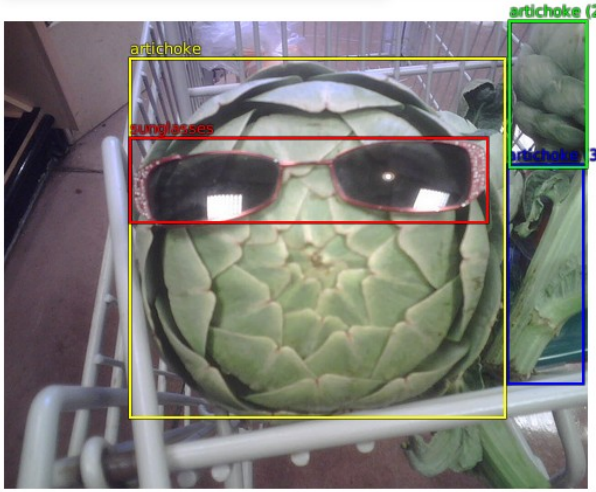
ILSVRC2012_val_00001273.JPEG



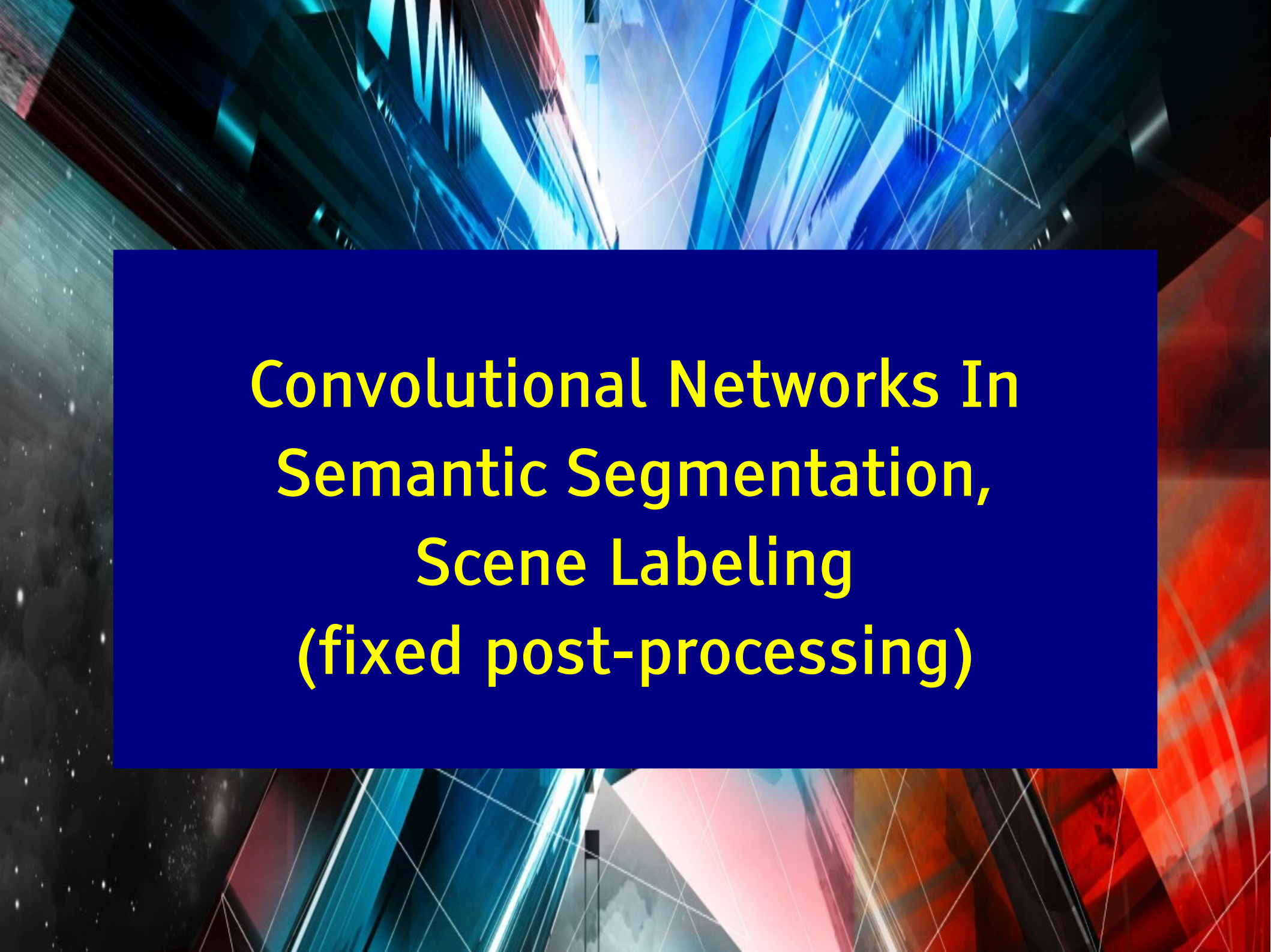
Groundtruth:
drum
lamp
lamp (2)
guitar
person
person (2)
person (3)
microphone
microphone (2)
microphone (3)

Top predictions:
artichoke (confidence 162.8)

ILSVRC2012_val_00001549.JPEG

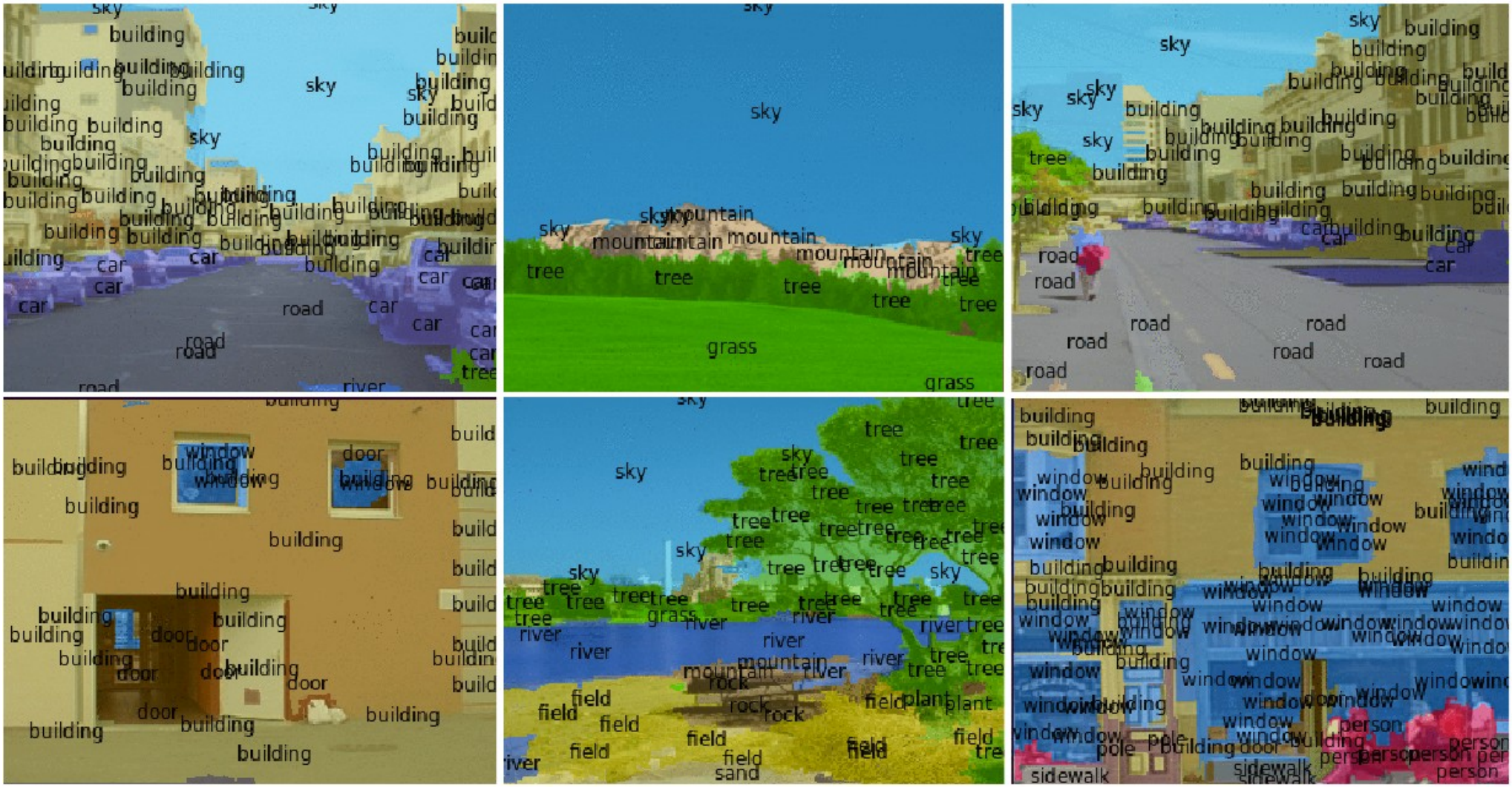


Groundtruth:
sunglasses
artichoke
artichoke (2)
artichoke (3)


Convolutional Networks In
Semantic Segmentation,
Scene Labeling
(fixed post-processing)

Semantic Labeling:
Labeling every pixel with the object it belongs to

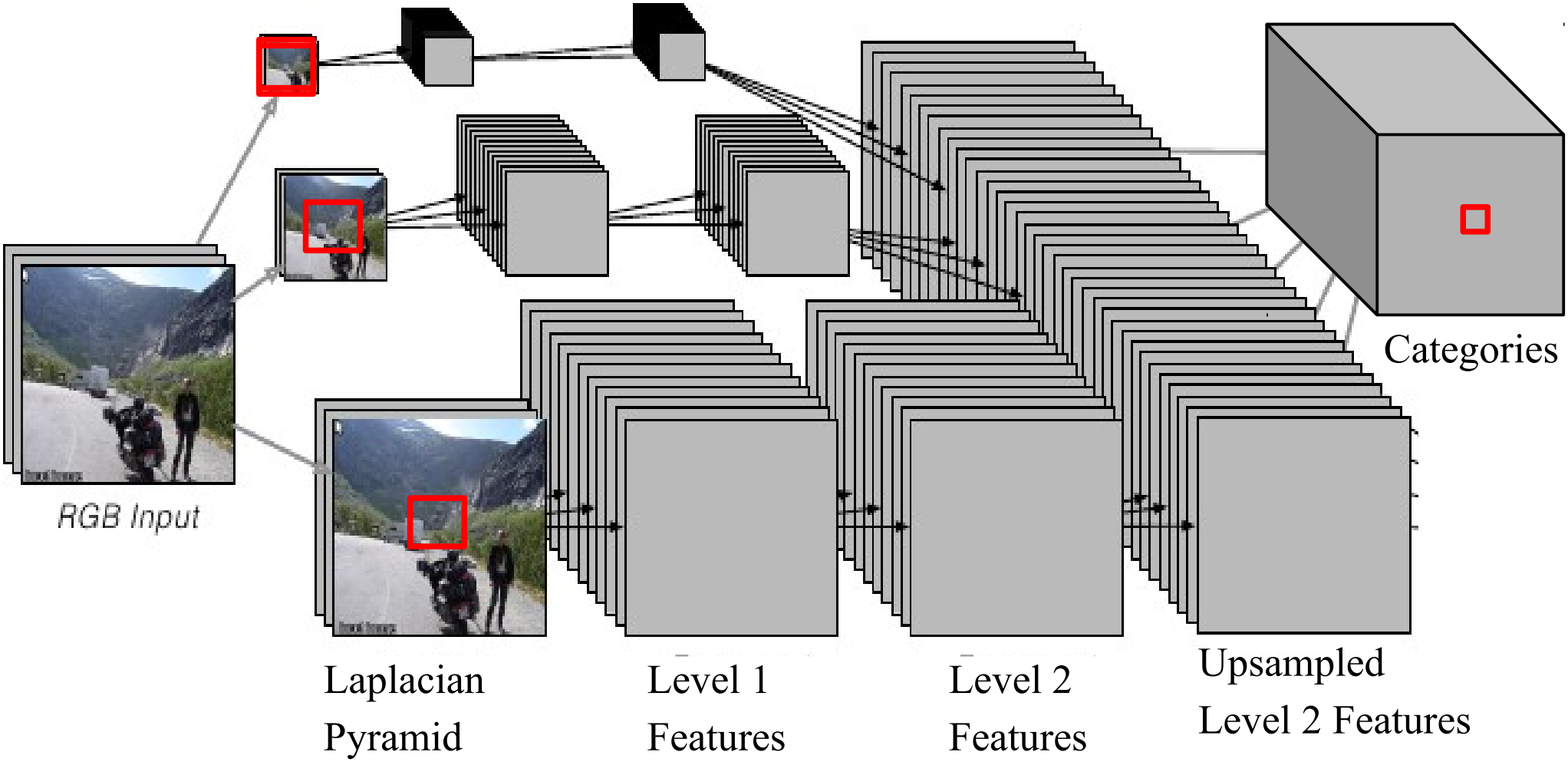
- Would help identify obstacles, targets, landing sites, dangerous areas
- Would help line up depth map with edge maps



[Farabet et al. ICML 2012, PAMI 2013]

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



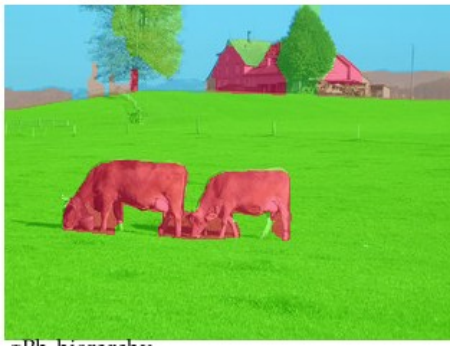
Method 1: majority over super-pixel regions

Input image

Super-pixel boundary hypetheses

Superpixel boundaries

Majority Vote Over Superpixels



Categories aligned With region boundaries

Multi-scale ConvNet

Features from Convolutional net (d=768 per pixel)

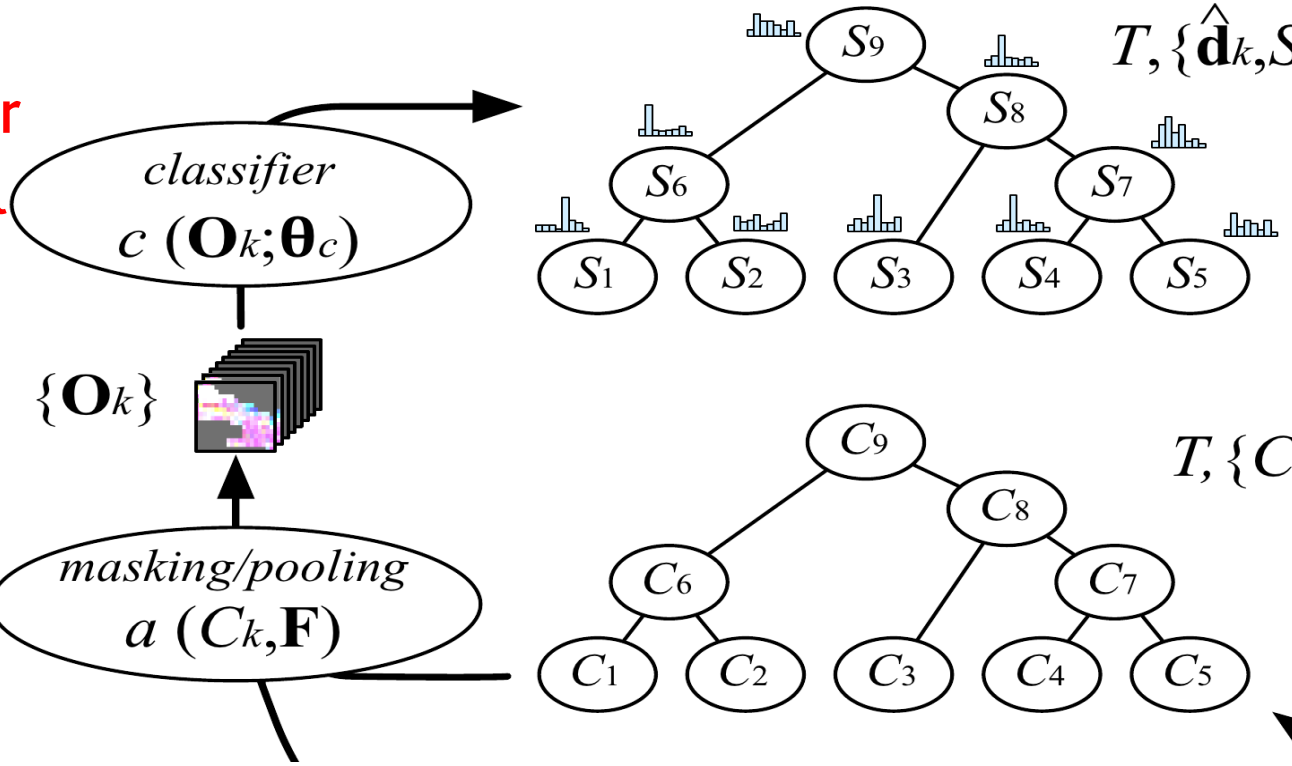
Convolutional classifier

"soft" categories scores

[Farabet et al. IEEE T. PAMI 2013]

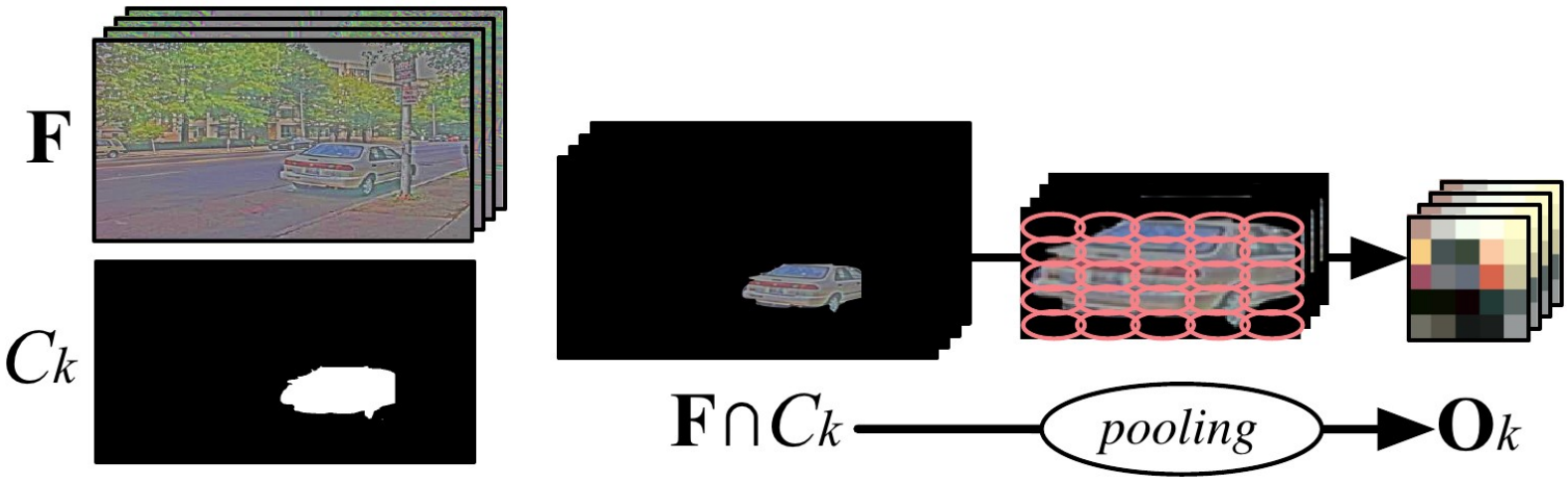
Method 2: optimal cover of purity tree

2-layer Neural net



Distribution of Categories within Each Segment

Spanning Tree From pixel Similarity graph



[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. 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

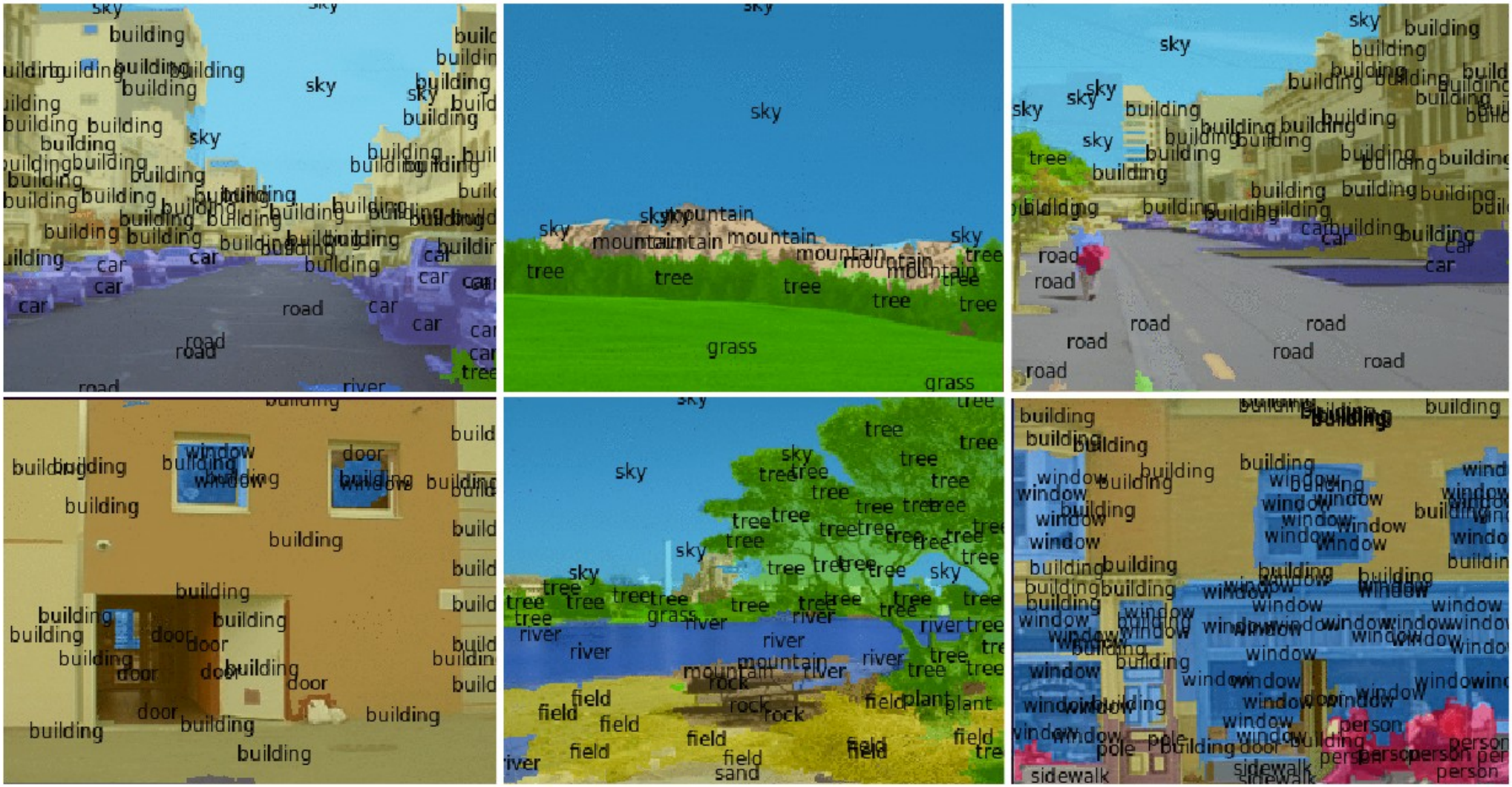
- Barcelona dataset
- [Tighe 2010]:
- 170 categories.

	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%**

[Farabet et al. IEEE T. PAMI 2012]

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

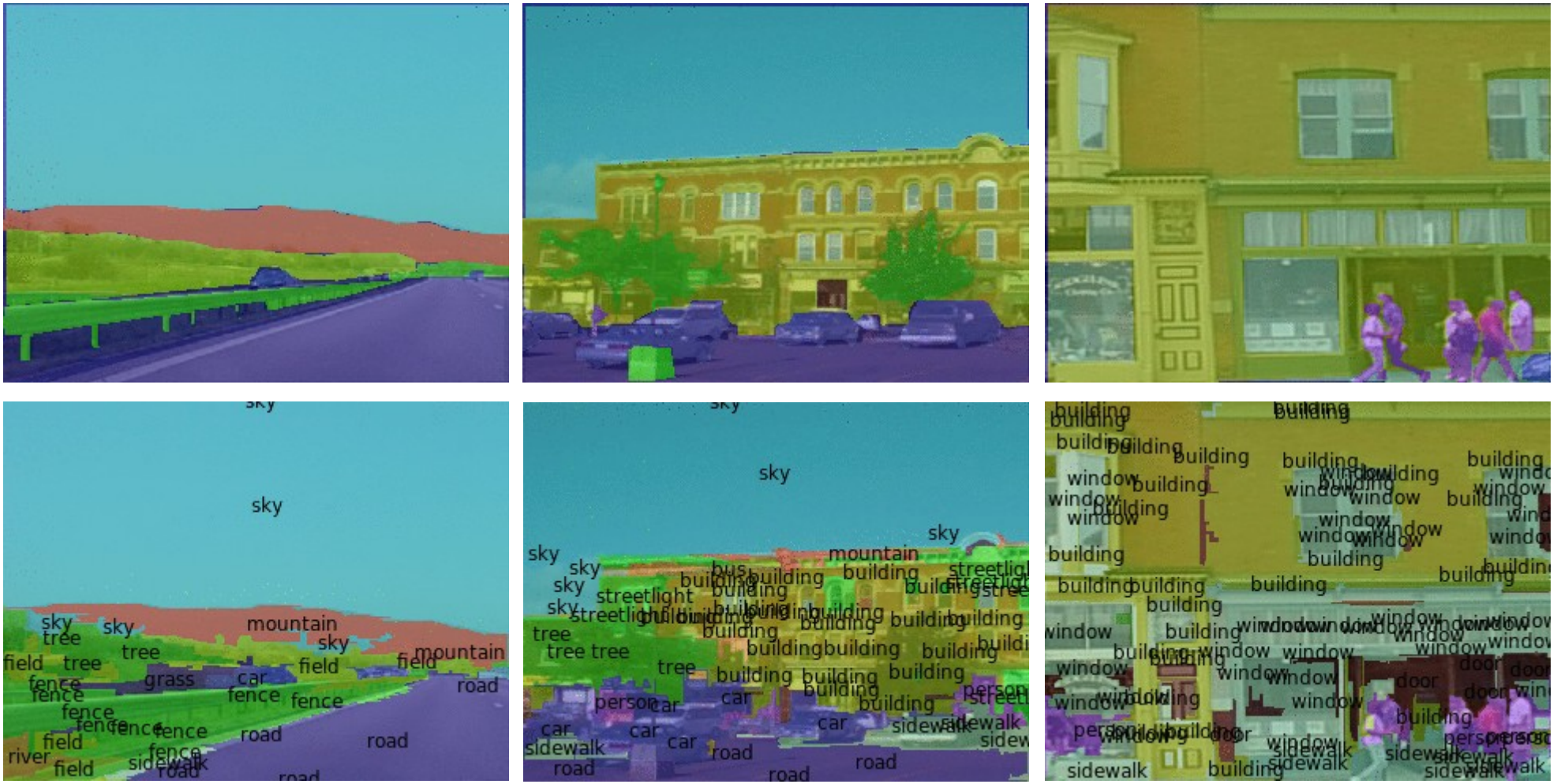
- Samples from the SIFT-Flow dataset (Liu)



[Farabet et al. ICML 2012, PAMI 2013]

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

Y LeCun

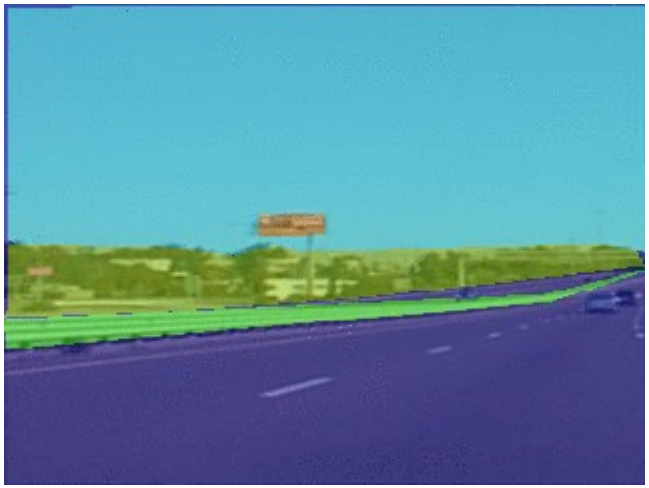


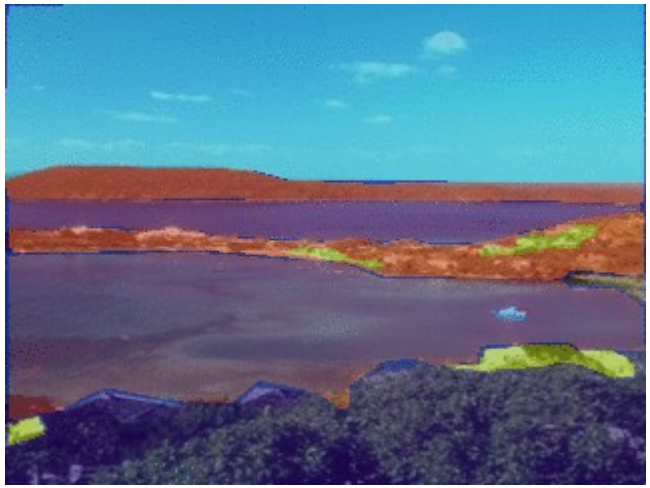
[Farabet et al. ICML 2012, PAMI 2013]

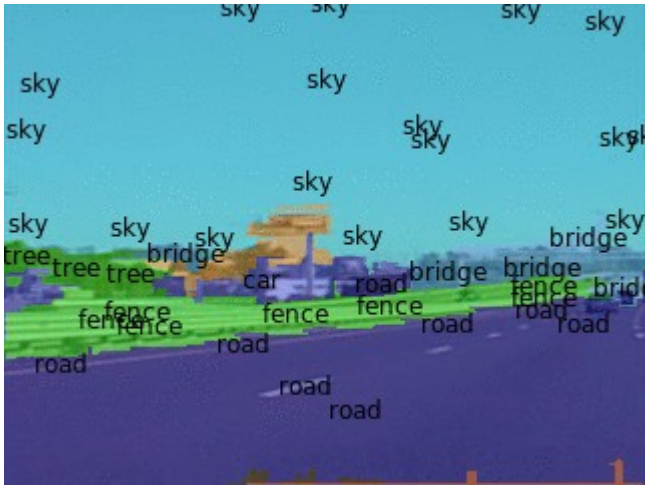
Scene Parsing/Labeling

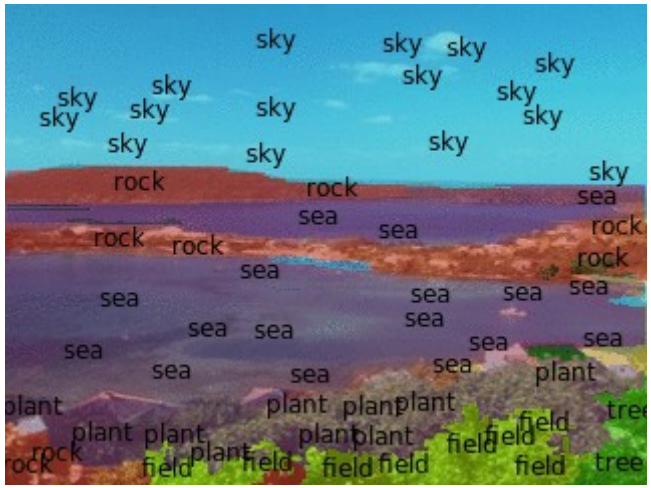
[Farabet et al. ICML 2012, PAMI 2013]

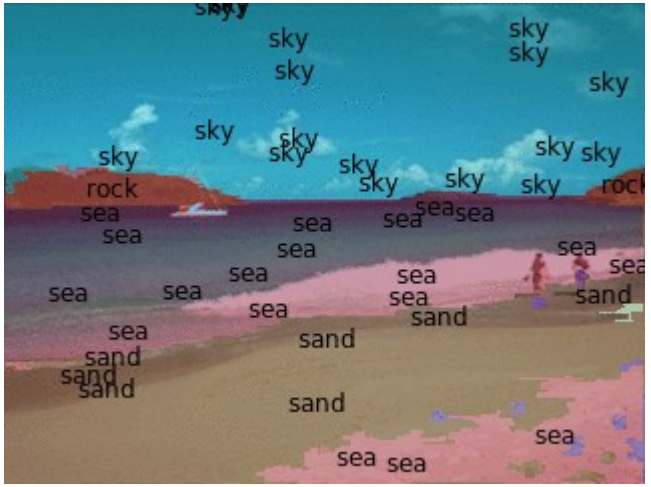
Scene Parsing/Labeling





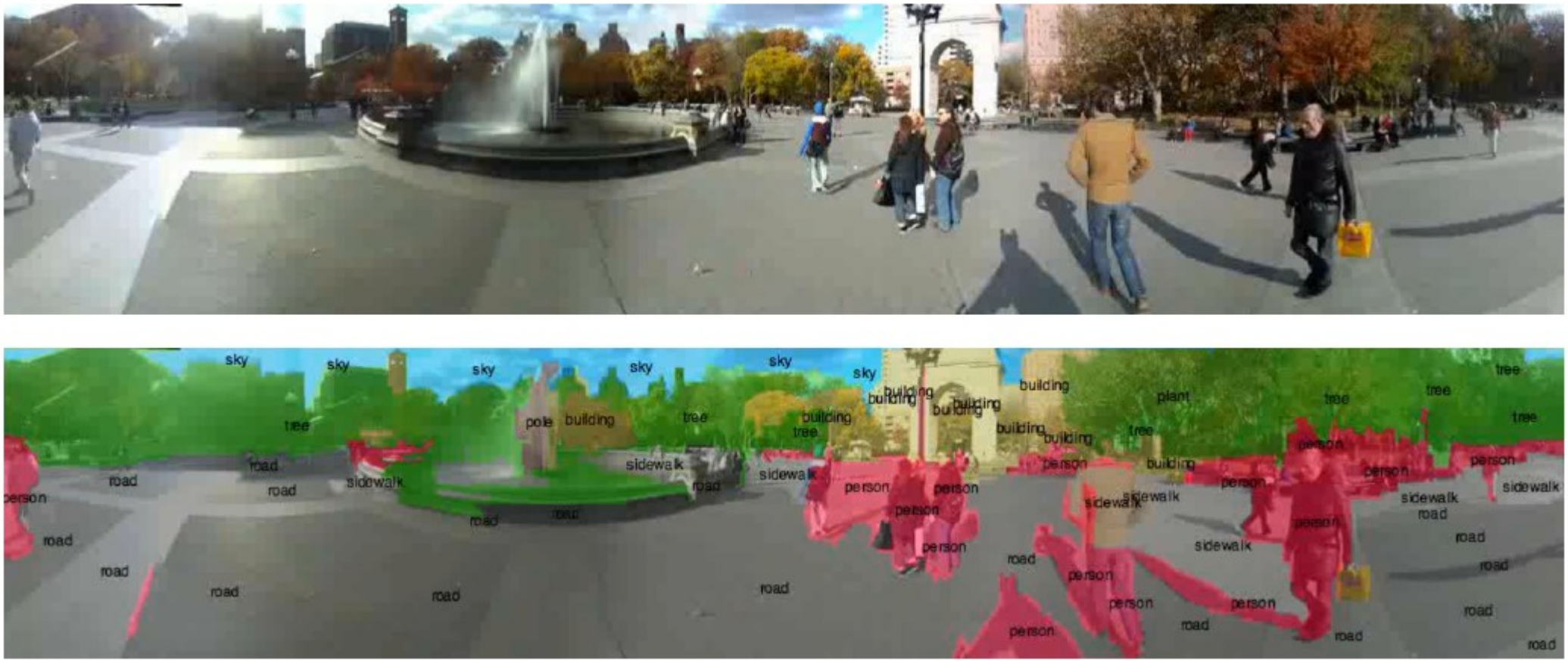






[Farabet et al. ICML 2012, PAMI 2013]

Scene Parsing/Labeling



[Farabet et al. ICML 2012, PAMI 2013]

Scene Parsing/Labeling

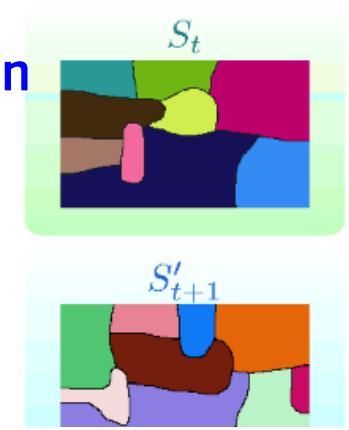
[Farabet et al. ICML 2012, PAMI 2013]

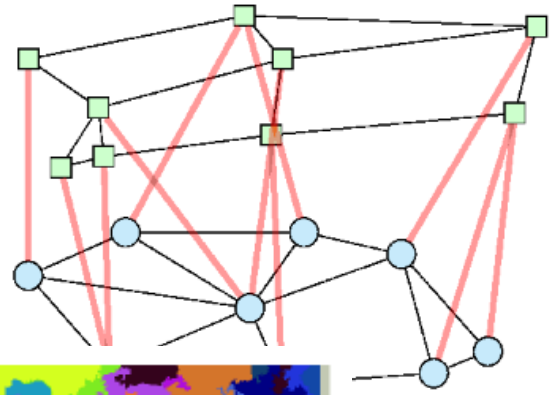
Scene Parsing/Labeling

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

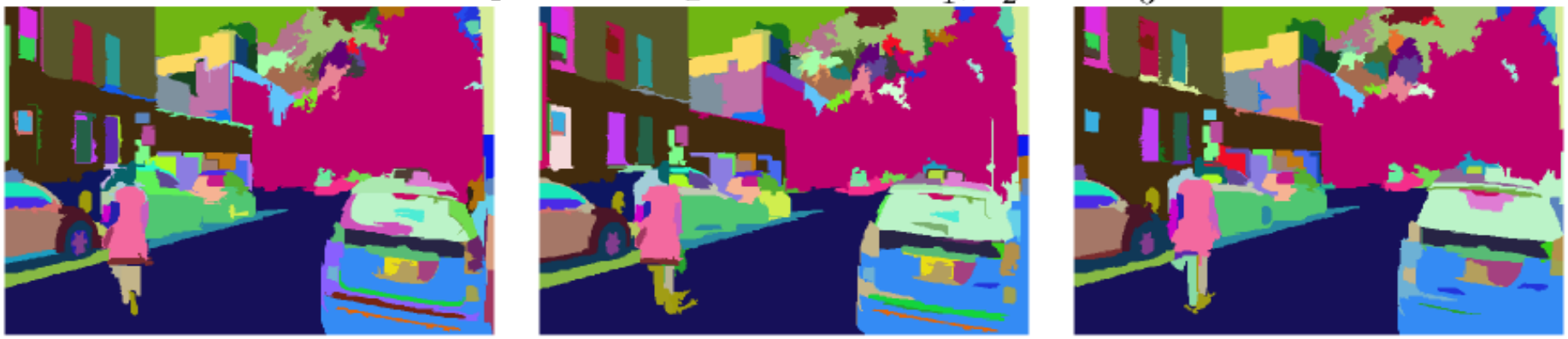
Temporal Consistency

- 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

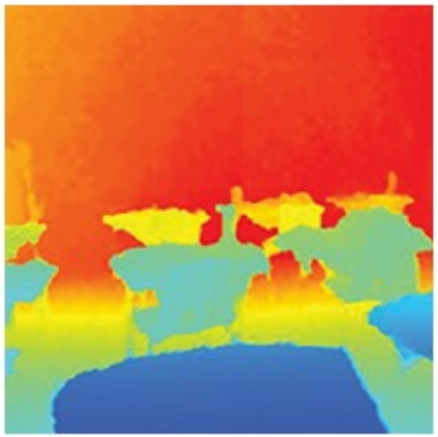
Scene Parsing/Labeling: Temporal Consistency

- Causal method for temporal consistency

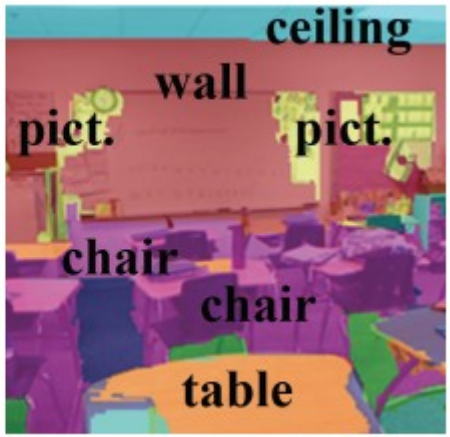
[Couprie, Farabet, Najman, LeCun ICLR 2013, ICIP 2013]

NYU RGB-D Dataset

Captured with a Kinect on a steadycam



?



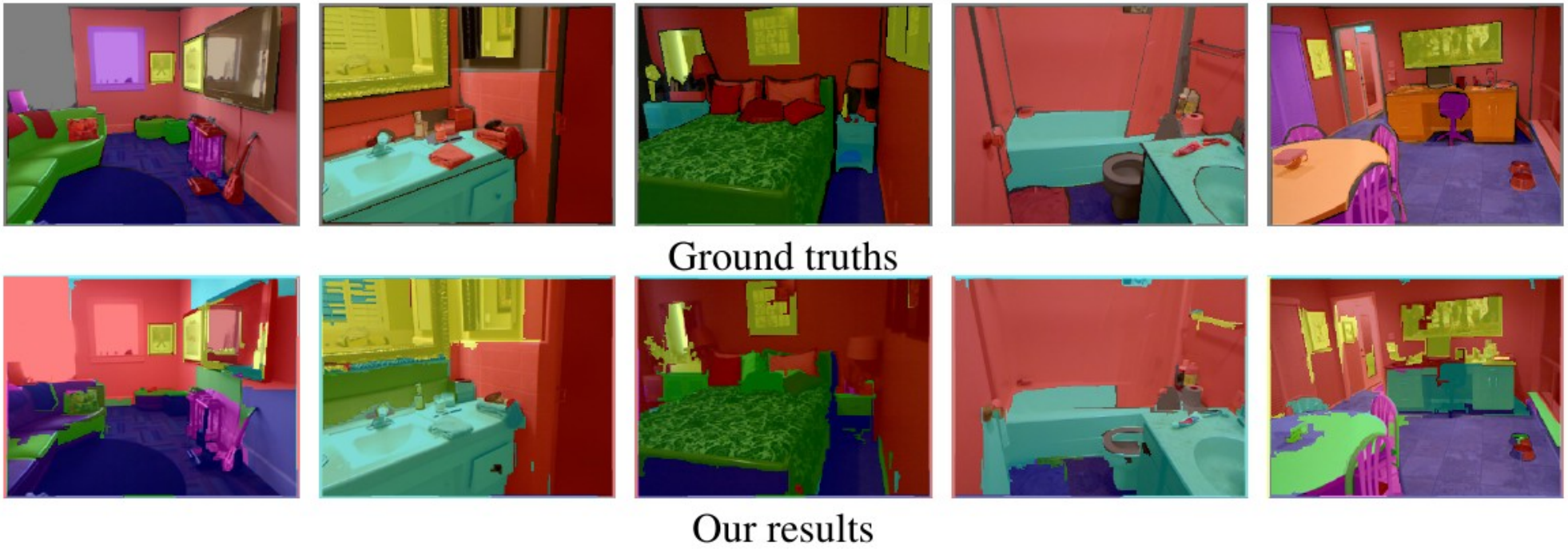
Results

- **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**

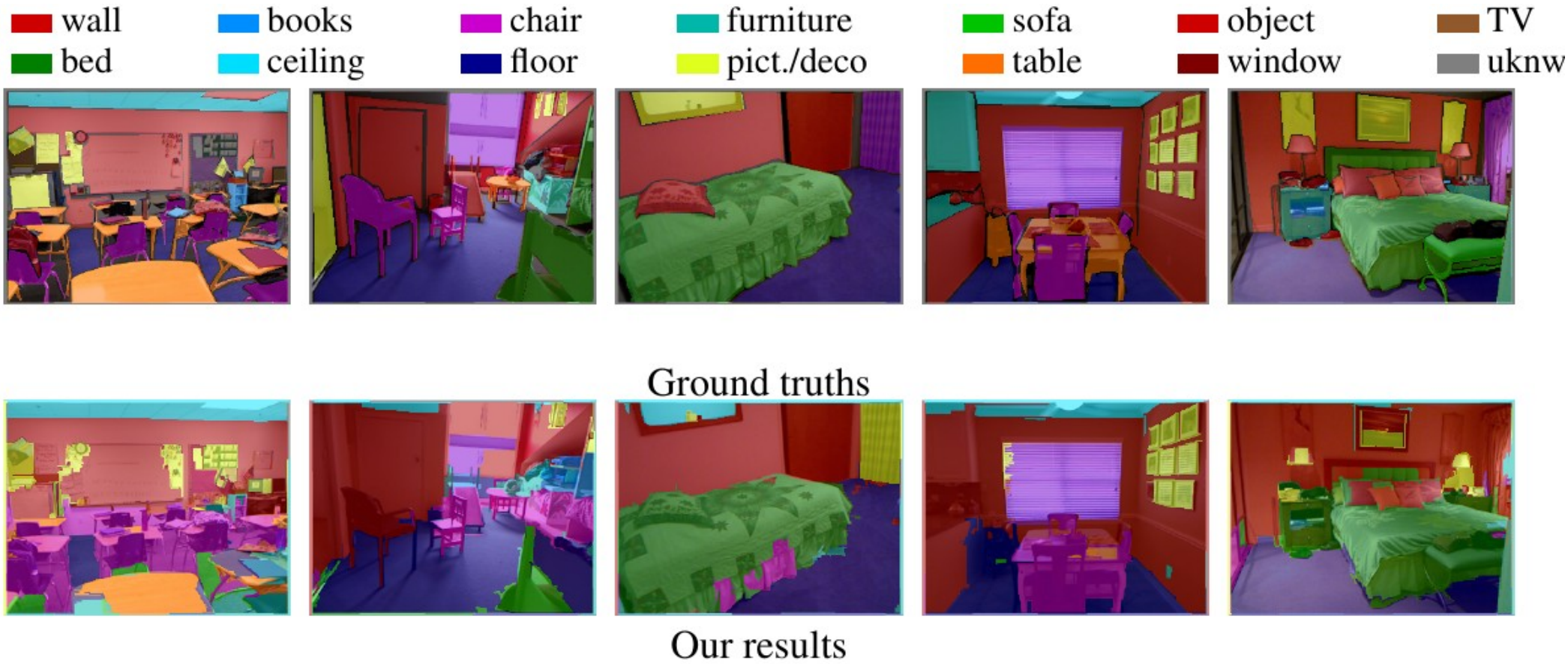
[C. Cadena, J. Kosecka "Semantic Parsing for Priming Object Detection in RGB-D Scenes" Semantic Perception Mapping and Exploration (SPME), Karlsruhe 2013]

Scene Parsing/Labeling on RGB+Depth Images



[Couprie, Farabet, Najman, LeCun ICLR 2013, ICIP 2013]

Scene Parsing/Labeling on RGB+Depth Images



[Couprie, Farabet, Najman, LeCun ICLR 2013, ICIP 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)

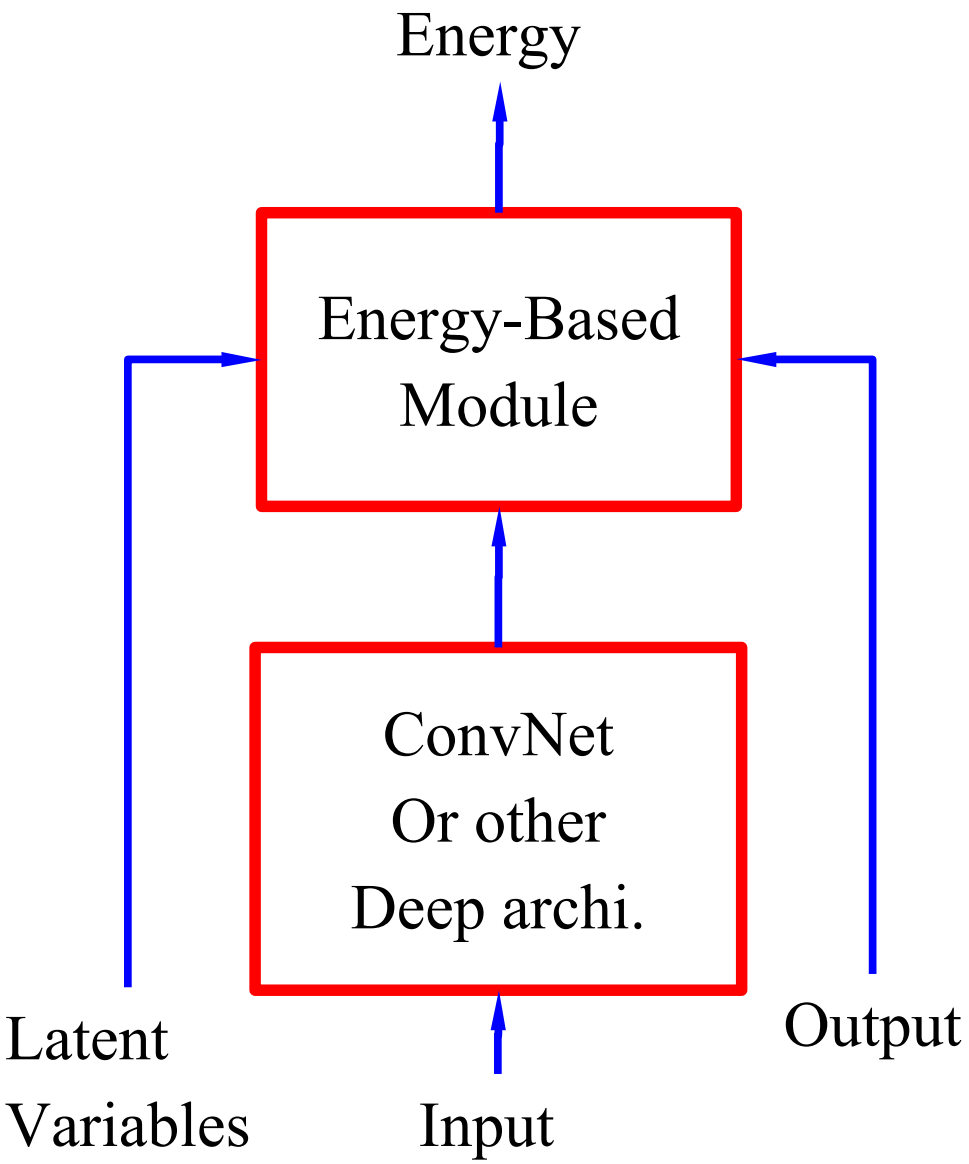
[Couprie, Farabet, Najman, LeCun ICLR 2013]
[Couprie, Farabet, Najman, LeCun ICIP 2013]
[Couprie, Farabet, Najman, LeCun submitted to JMLR]

Semantic Segmentation on RGB+D Images and Videos

[Couprie, Farabet, Najman, LeCun ICLR 2013, ICIP 2013]


Global (end-to-end) Training.
Training through
the post-processor.
Energy-Based Models

Global (End-to-End) Learning: Energy-Based Models.



Making every single module in the system trainable.

Every module is trained simultaneously so as to optimize a global loss function.

Includes the feature extractor, the recognizer, and the contextual post-processor (graphical model)

Problem: back-propagating gradients through the graphical model.

Energy-Based Models

Highly popular methods in the Machine Learning and Natural Language Processing Communities have their roots in Speech and Handwriting Recognition

- Structured Perceptron, Conditional Random Fields, and related learning models for "structured prediction" are descendants of discriminative learning methods for speech recognition and word-level handwriting recognition methods from the early 90's

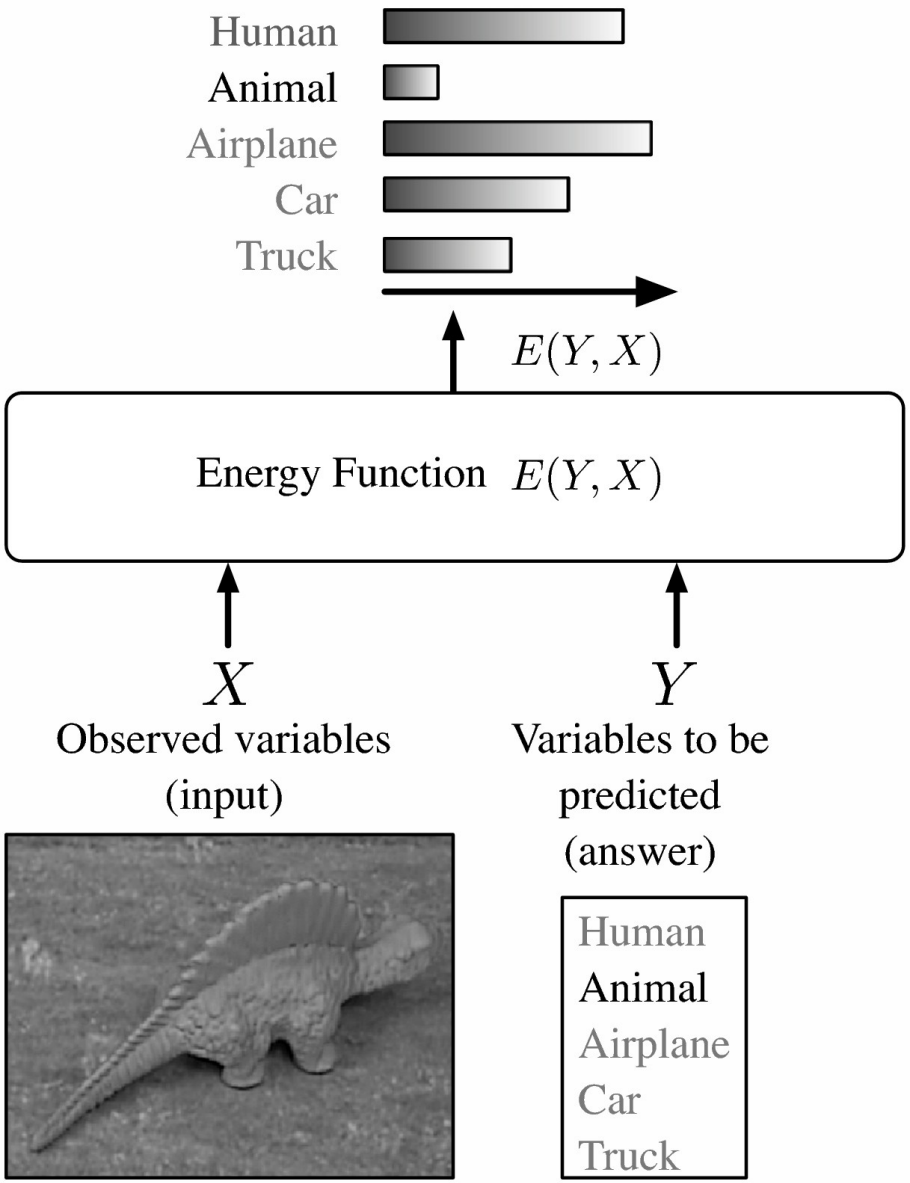
A Tutorial and Energy-Based Learning:

- [LeCun & al., 2006]

Discriminative Training for "Structured Output" models

- The whole literature on discriminative speech recognition [1987-]
- The whole literature on neural-net/HMM hybrids for speech [Bottou 1991, Bengio 1993, Haffner 1993, Bourlard 1994]
- Graph Transformer Networks [LeCun & al. Proc IEEE 1998]
- Structured Perceptron [Collins 2001]
- Conditional Random Fields [Lafferty & al 2001]
- Max Margin Markov Nets [Altun & al 2003, Taskar & al 2003]

Energy-Based Models for Decision Making

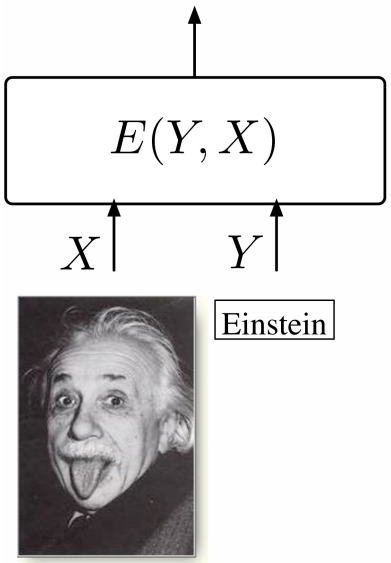


- **Model:** Measures the compatibility between an observed variable X and a variable to be predicted Y through an energy function E(Y,X).

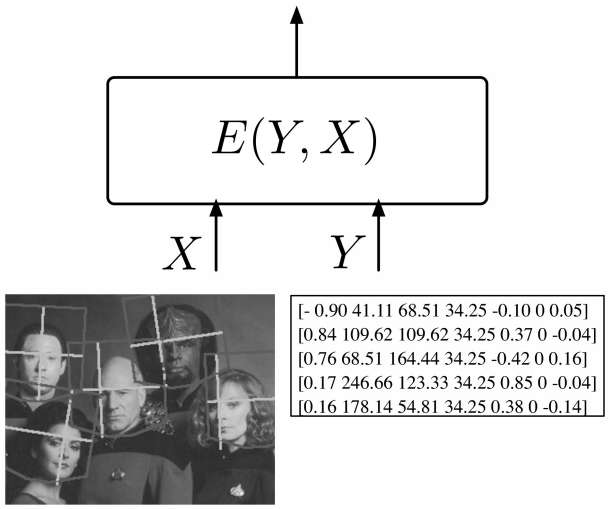
$$Y^* = \text{argmin}_{Y \in \mathcal{Y}} E(Y, X).$$

- **Inference:** Search for the Y that minimizes the energy within a se $\mathcal{Y}$
- If the set has low cardinality, we can use exhaustive search.

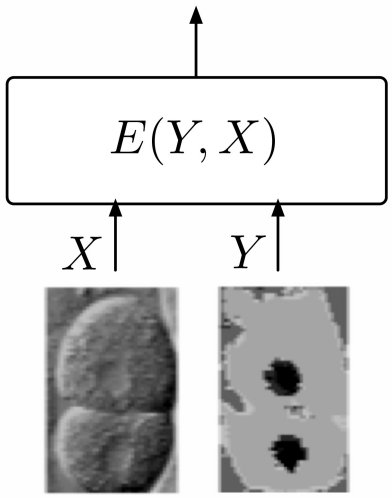
But Inference Might Be Complicated



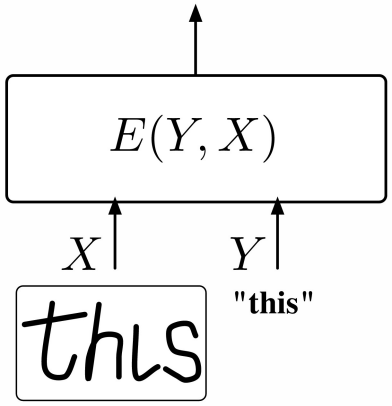
(a)



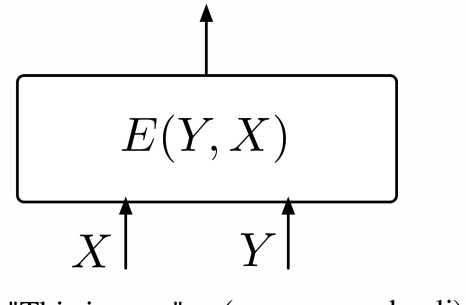
(b)



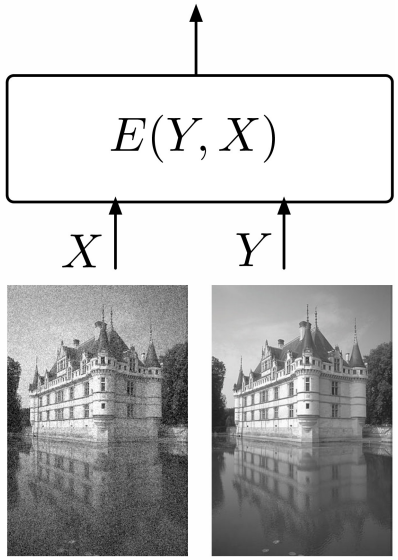
(c)



(d)



(e)



(f)

- When the cardinality or dimension of Y is large, exhaustive search is impractical.
- We need to use "smart" inference procedures: min-sum, Viterbi, min cut, belief propagation, gradient decent.....

Converting Energies to Probabilities

Energies are uncalibrated

- The energies of two separately-trained systems cannot be combined
- The energies are uncalibrated (measured in arbitrary untis)

How do we calibrate energies?

- We turn them into probabilities (positive numbers that sum to 1).
- Simplest way: Gibbs distribution
- Other ways can be reduced to Gibbs by a suitable redefinition of the energy.

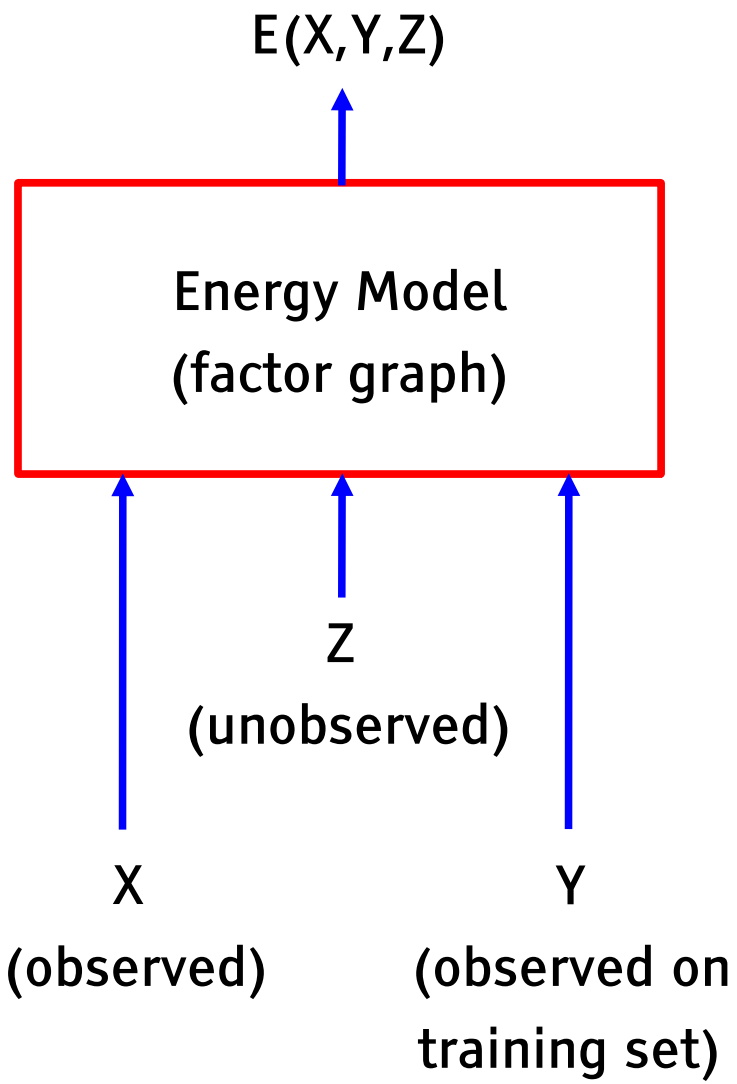
$$P(Y|X) = \frac{e^{-\beta E(Y,X)}}{\int_{y \in \mathcal{Y}} e^{-\beta E(y,X)}},$$

Partition function

Inverse temperature

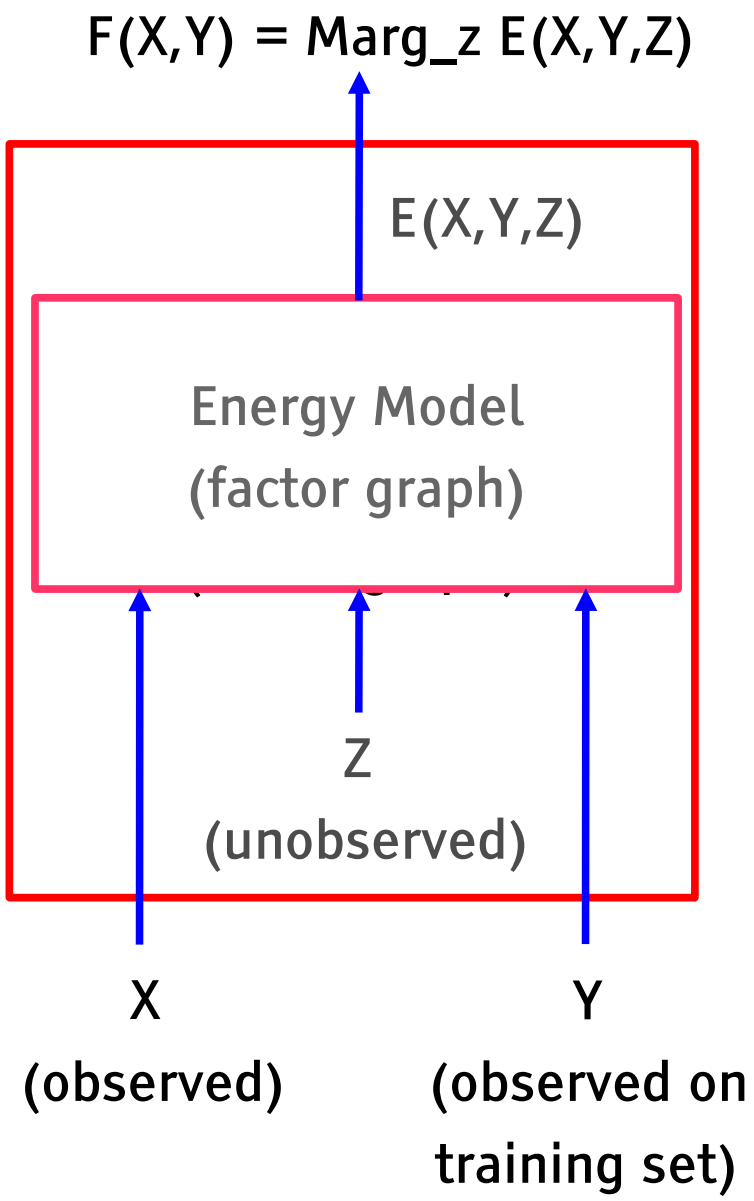
Integrating Deep Learning and Structured Prediction

- Deep Learning systems can be assembled into factor graphs
  - Energy function is a sum of factors
  - Factors can embed whole deep learning systems
  - X: observed variables (inputs)
  - Z: never observed (latent variables)
  - Y: observed on training set (output variables)
- Inference is energy minimization (MAP) or free energy minimization (marginalization) over Z and Y given an X



Integrating Deep Learning and Structured Prediction

- **Deep Learning systems can be assembled into factor graphs**
  - Energy function is a sum of factors
  - Factors can embed whole deep learning systems
  - X: observed variables (inputs)
  - Z: never observed (latent variables)
  - Y: observed on training set (output variables)
- **Inference is energy minimization (MAP) or free energy minimization (marginalization) over Z and Y given an X**
  - F(X,Y) = MIN_z E(X,Y,Z)
  - F(X,Y) = -log SUM_z exp[-E(X,Y,Z)]



Handwriting recognition Sequence labeling

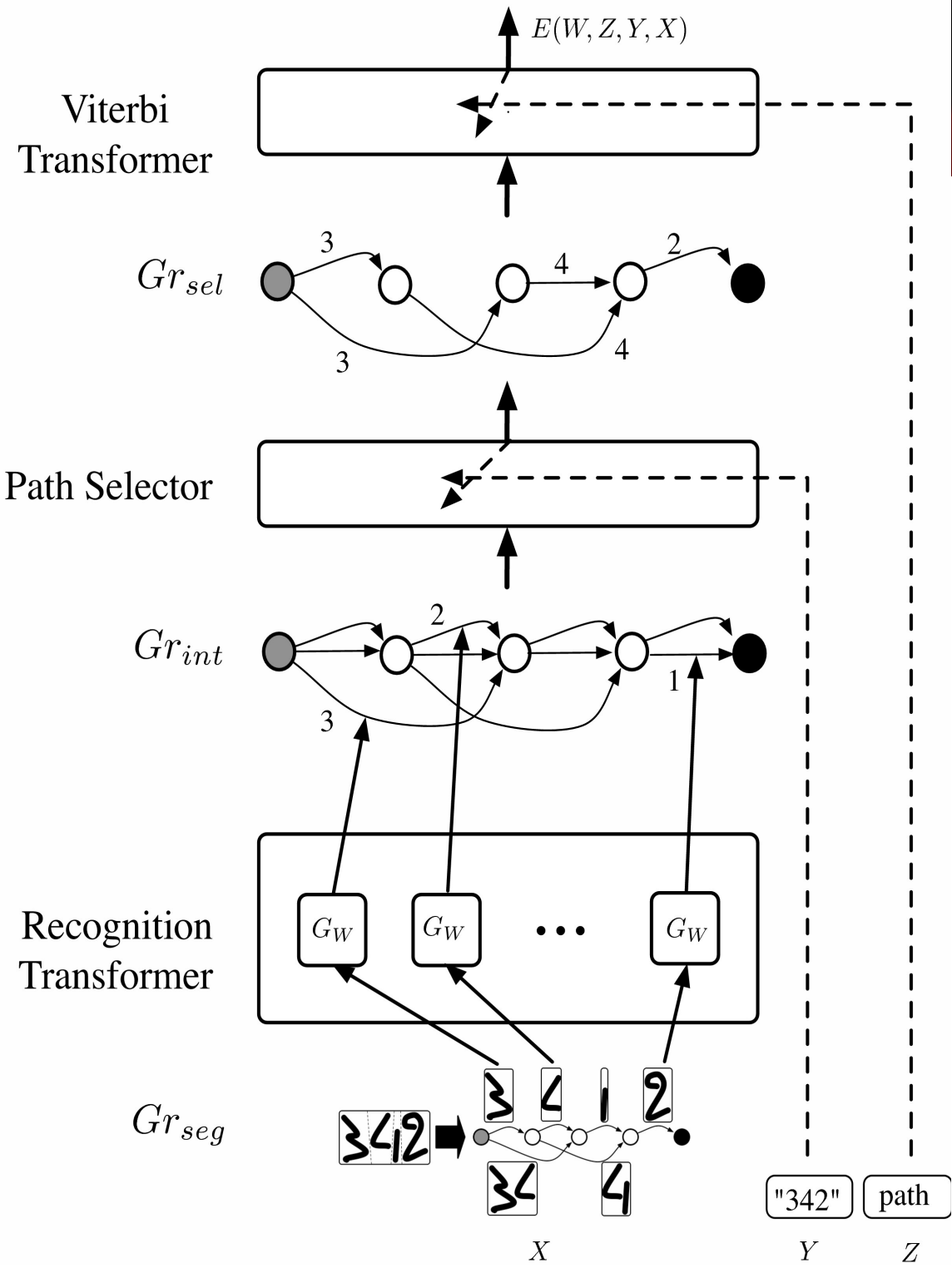
integrated segmentation and recognition of sequences.

Each segmentation and recognition hypothesis is a path in a graph

inference = finding the shortest path in the interpretation graph.

Un-normalized hierarchical HMMs a.k.a. Graph Transformer Networks

- [LeCun, Bottou, Bengio, Haffner, Proc IEEE 1998]

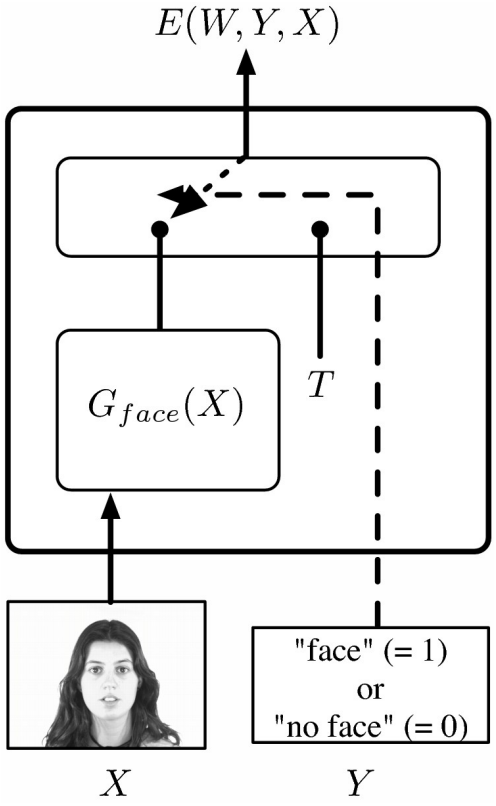


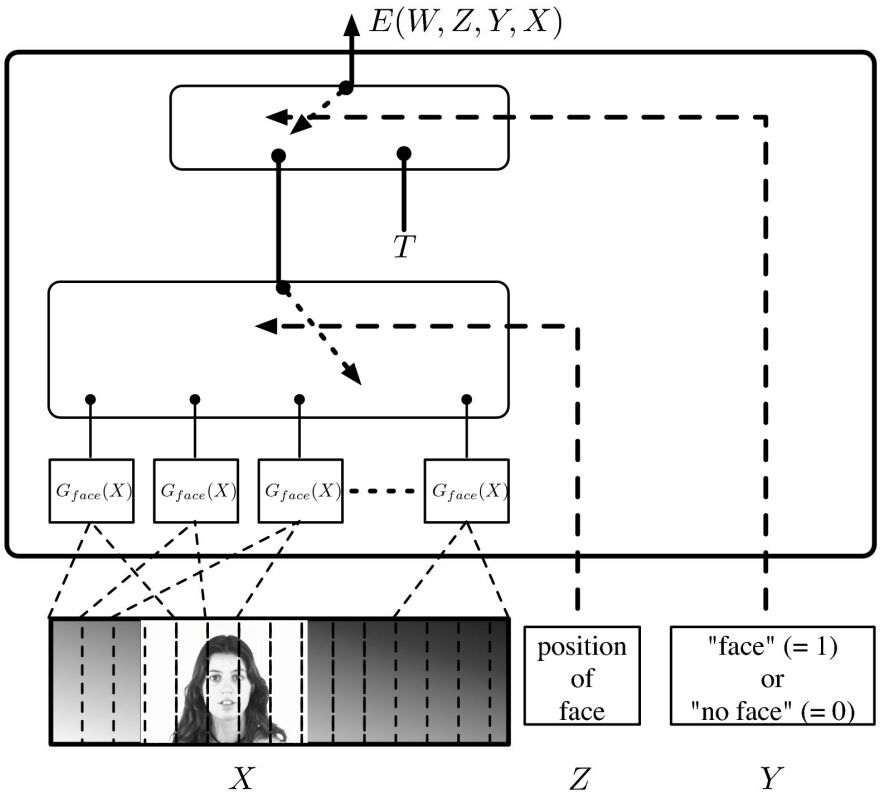
Example of Latent Variable Models: object detection

The energy includes "hidden" variables Z whose value is never given to us

$$E(Y, X) = \min_{Z \in \mathcal{Z}} E(Z, Y, X).$$

$$Y^* = \text{argmin}_{Y \in \mathcal{Y}, Z \in \mathcal{Z}} E(Z, Y, X).$$





Latent Variable Models

The energy includes "hidden" variables Z whose value is never given to us

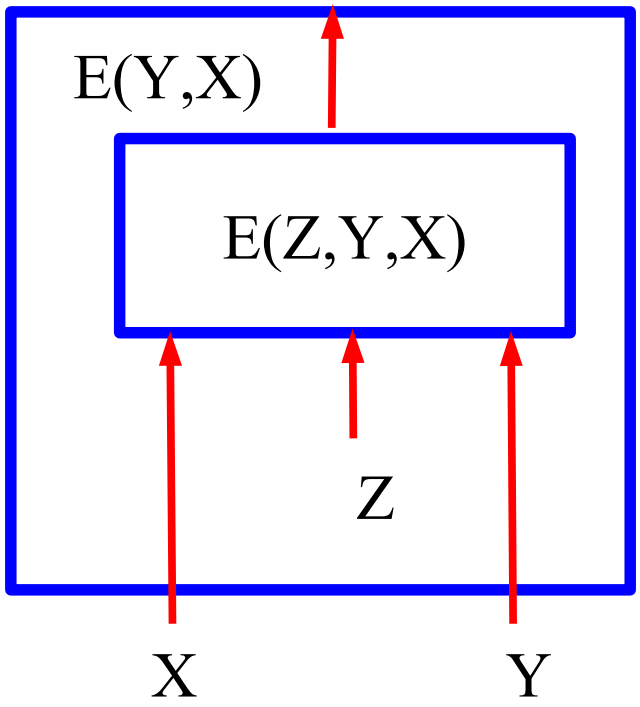
- We can minimize the energy over those latent variables
- We can also "marginalize" the energy over the latent variables

Minimization over latent variables:

$$E(Y,X) = \min_{Z \in \mathcal{Z}} E(Z,Y,X).$$

Marginalization over latent variables:

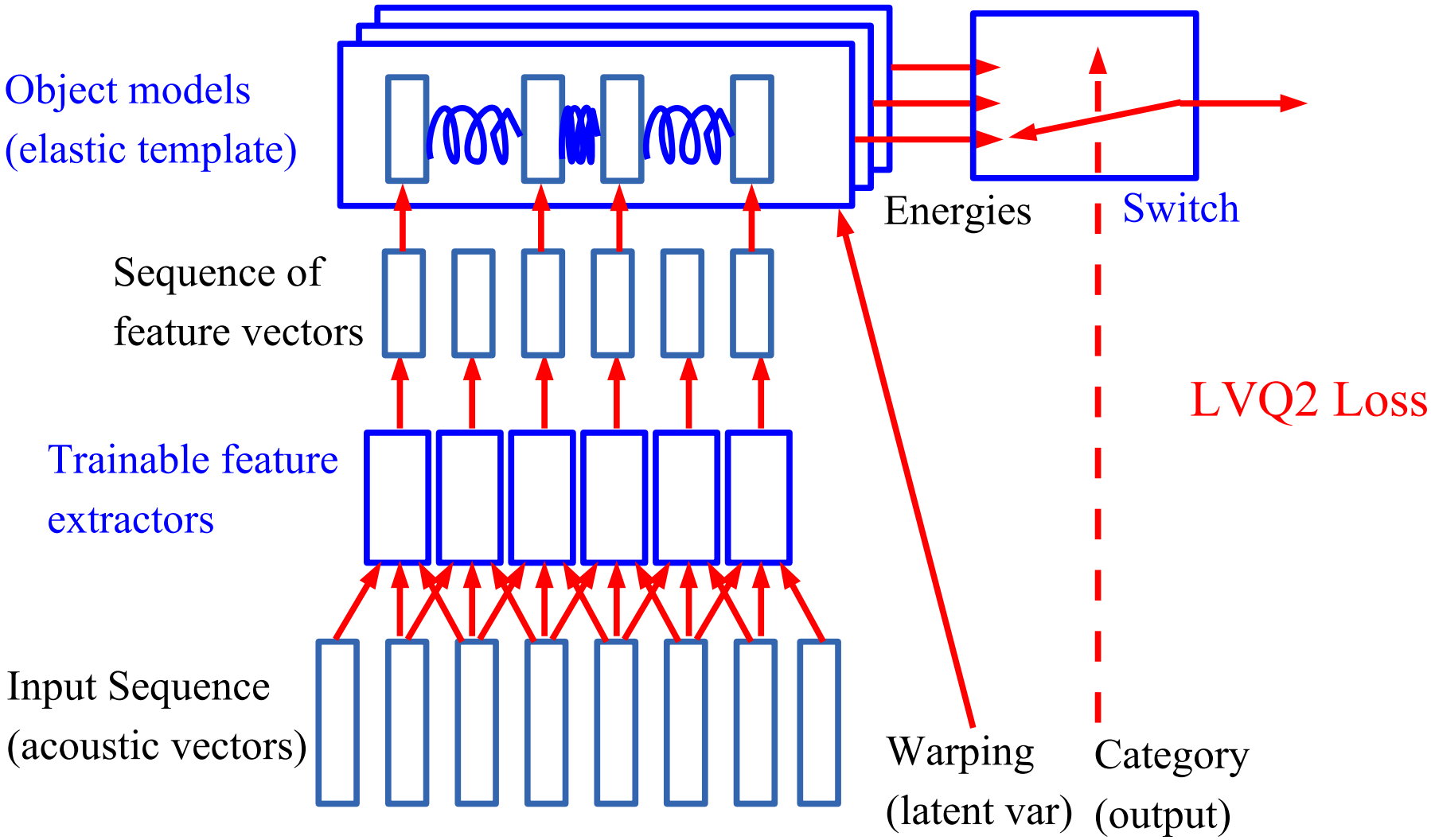
$$E(X,Y) = -\frac{1}{\beta} \log \int_{z \in \mathcal{Z}} e^{-\beta E(z,Y,X)}$$



Estimation this integral may require some approximations (sampling, variational methods,....)

Example 1: Integrated Training with Sequence Alignment

Spoken word recognition with trainable elastic templates and trainable feature extraction [Driancourt&Bottou 1991, Bottou 1991, Driancourt 1994]

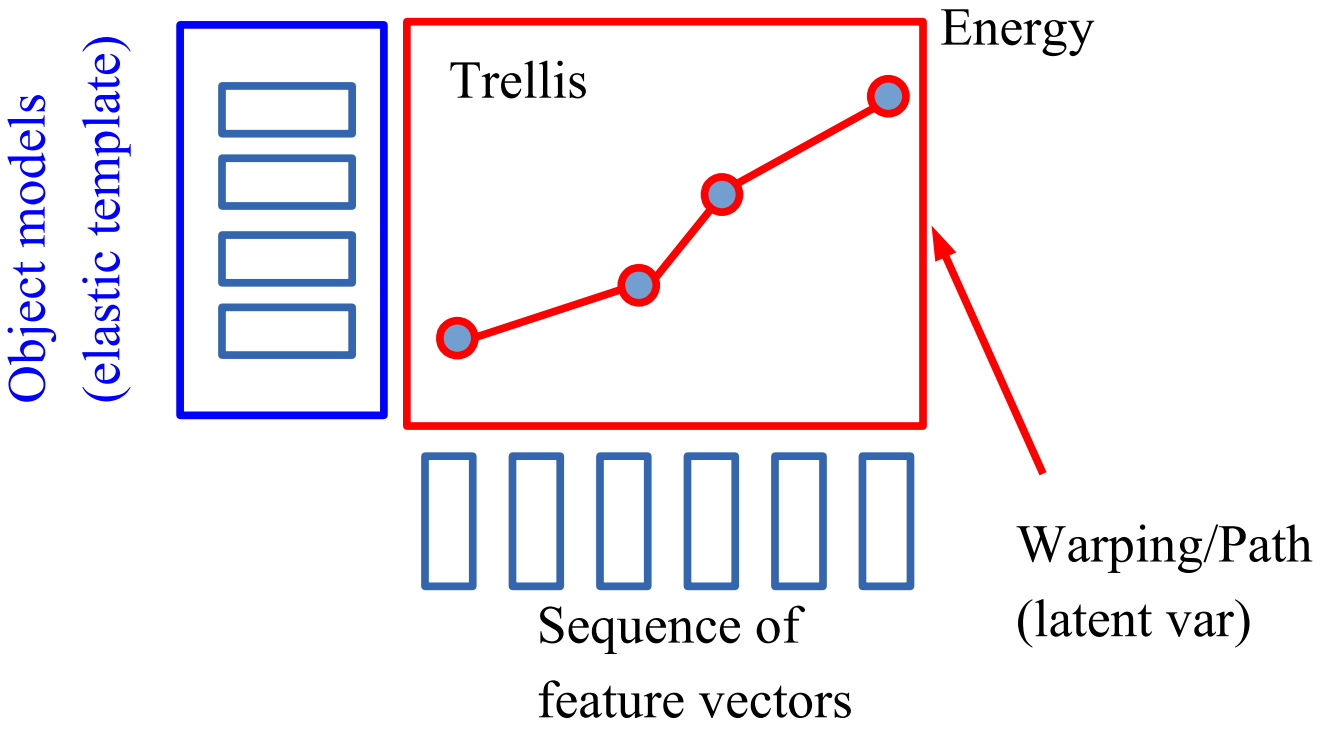


Example: 1-D Constellation Model (a.k.a. Dynamic Time Warping)

Spoken word recognition with trainable elastic templates and trainable feature extraction [Driancourt&Bottou 1991, Bottou 1991, Driancourt 1994]

Elastic matching using dynamic time warping (Viterbi algorithm on a trellis).

The corresponding EBFG is implicit (it changes for every new sample).



The Oldest Example of Structured Prediction

Trainable Automatic Speech Recognition system with a convolutional net (TDNN) and dynamic time warping (DTW)

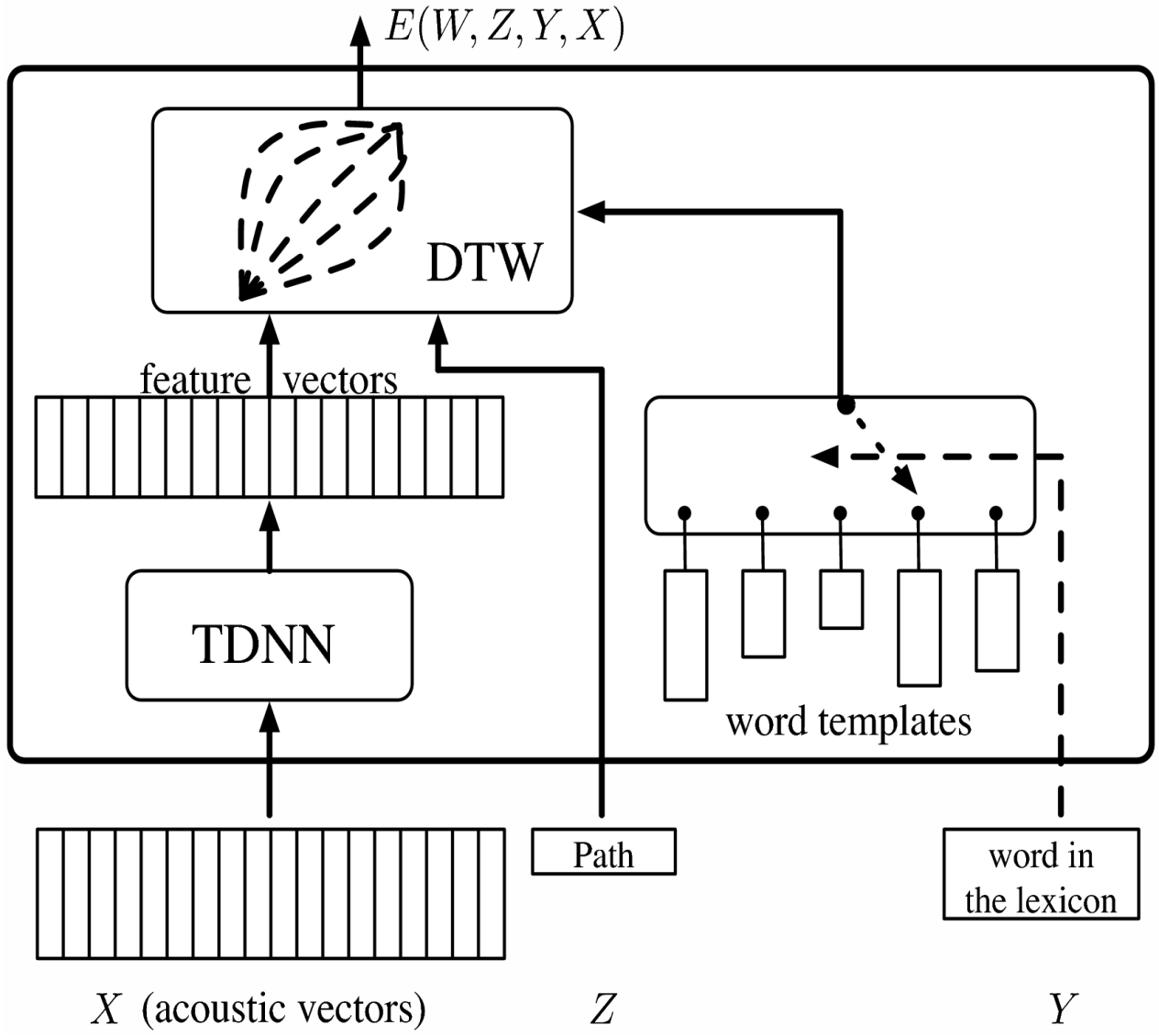
The feature extractor and the structured classifier are trained simultanesously in an integrated fashion.

with the LVQ2 Loss :

- Driancourt and Bottou's speech recognizer (1991)

with NLL:

- Bengio's speech recognizer (1992)
- Haffner's speech recognizer (1993)



What can the latent variables represent?

Variables that would make the task easier if they were known:

- **Face recognition**: the gender of the person, the orientation of the face.
- **Object recognition**: the pose parameters of the object (location, orientation, scale), the lighting conditions.
- **Parts of Speech Tagging**: the segmentation of the sentence into syntactic units, the parse tree.
- **Speech Recognition**: the segmentation of the sentence into phonemes or phones.
- **Handwriting Recognition**: the segmentation of the line into characters.
- **Object Recognition/Scene Parsing:** the segmentation of the image into components (objects, parts,...)

In general, we will search for the value of the latent variable that allows us to get an answer (Y) of smallest energy.

Probabilistic Latent Variable Models

Marginalizing over latent variables instead of minimizing.

$$P(Z,Y|X) = \frac{e^{-\beta E(Z,Y,X)}}{\int_{y\in\mathcal{Y},\ z\in\mathcal{Z}} e^{-\beta E(y,z,X)}}.$$

$$P(Y|X) = \frac{\int_{z\in\mathcal{Z}} e^{-\beta E(Z,Y,X)}}{\int_{y\in\mathcal{Y},\ z\in\mathcal{Z}} e^{-\beta E(y,z,X)}}.$$

Equivalent to traditional energy-based inference with a redefined energy function:

$$Y^* = \text{argmin}_{Y\in\mathcal{Y}} -\frac{1}{\beta}\log\int_{z\in\mathcal{Z}} e^{-\beta E(z,Y,X)}.$$

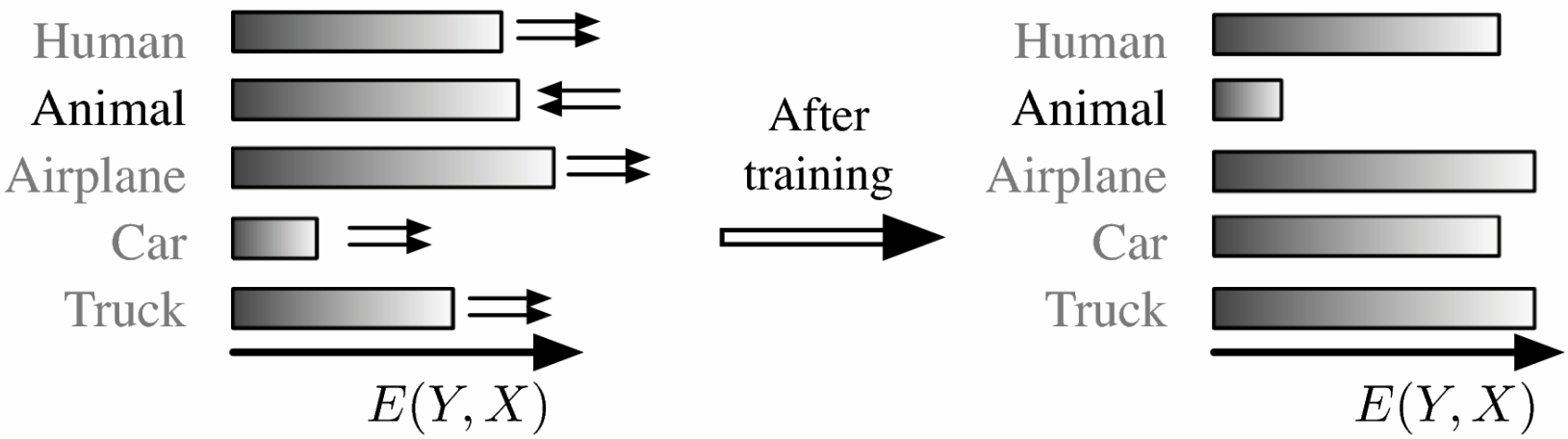
Reduces to traditional minimization when Beta->infinity

Training an EBM

Training an EBM consists in shaping the energy function so that the energies of the correct answer is lower than the energies of all other answers.

- Training sample: X = image of an animal, Y = "animal"

$$E(\text{animal}, X) < E(y, X) \; \forall \, y \neq \text{animal}$$



Architecture and Loss Function

Family of energy functions $\mathcal{E} = \{E(W, Y, X) : \quad W \in \mathcal{W}\}.$

Training set $\mathcal{S} = \{(X^i, Y^i) : i = 1 \ldots P\}$

Loss functional / Loss function $\mathcal{L}(E, \mathcal{S}) \qquad \mathcal{L}(W, \mathcal{S})$

- Measures the quality of an energy function on training set

Training $W^* = \min_{W \in \mathcal{W}} \mathcal{L}(W, \mathcal{S}).$

Form of the loss functional

- invariant under permutations and repetitions of the samples

$$\mathcal{L}(E, \mathcal{S}) = \frac{1}{P} \sum_{i=1}^{P} L(Y^i, E(W, \mathcal{Y}, X^i)) + R(W).$$

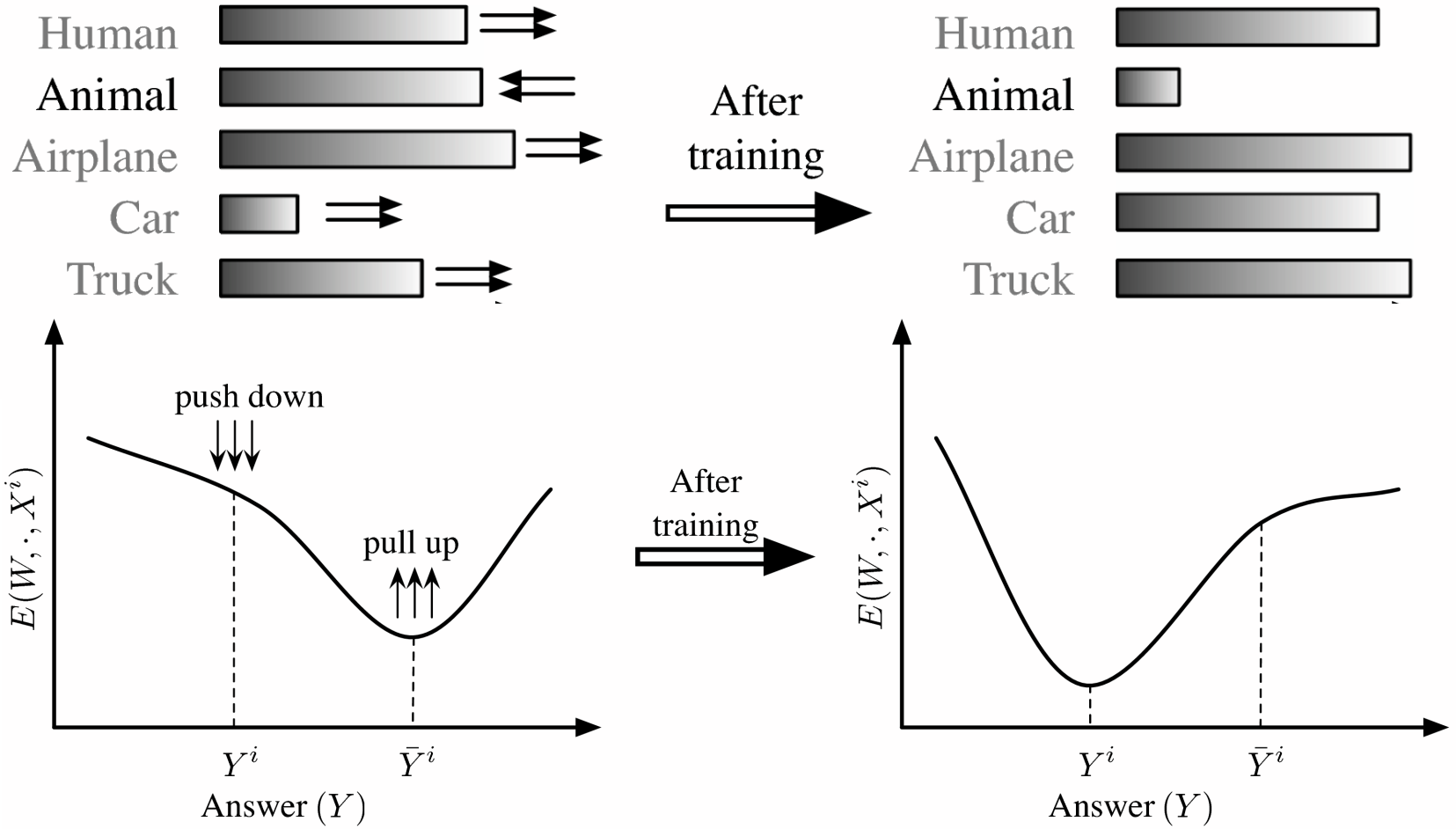
Per-sample loss

Desired answer

Energy surface for a given Xi as Y varies

Regularizer

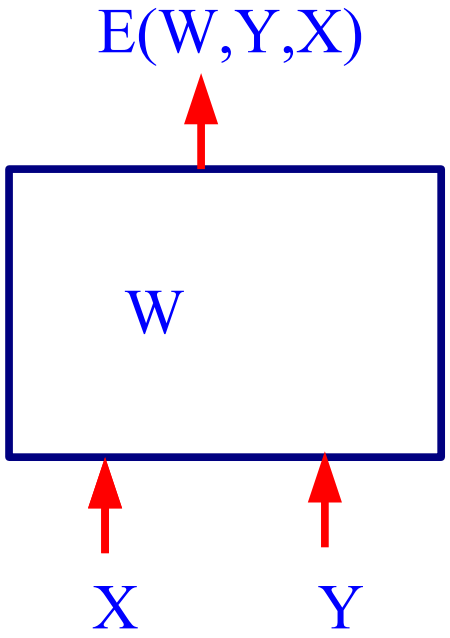
Designing a Loss Functional



Push down on the energy of the correct answer

Pull up on the energies of the incorrect answers, particularly if they are smaller than the correct one

Architecture + Inference Algo + Loss Functional = EBM



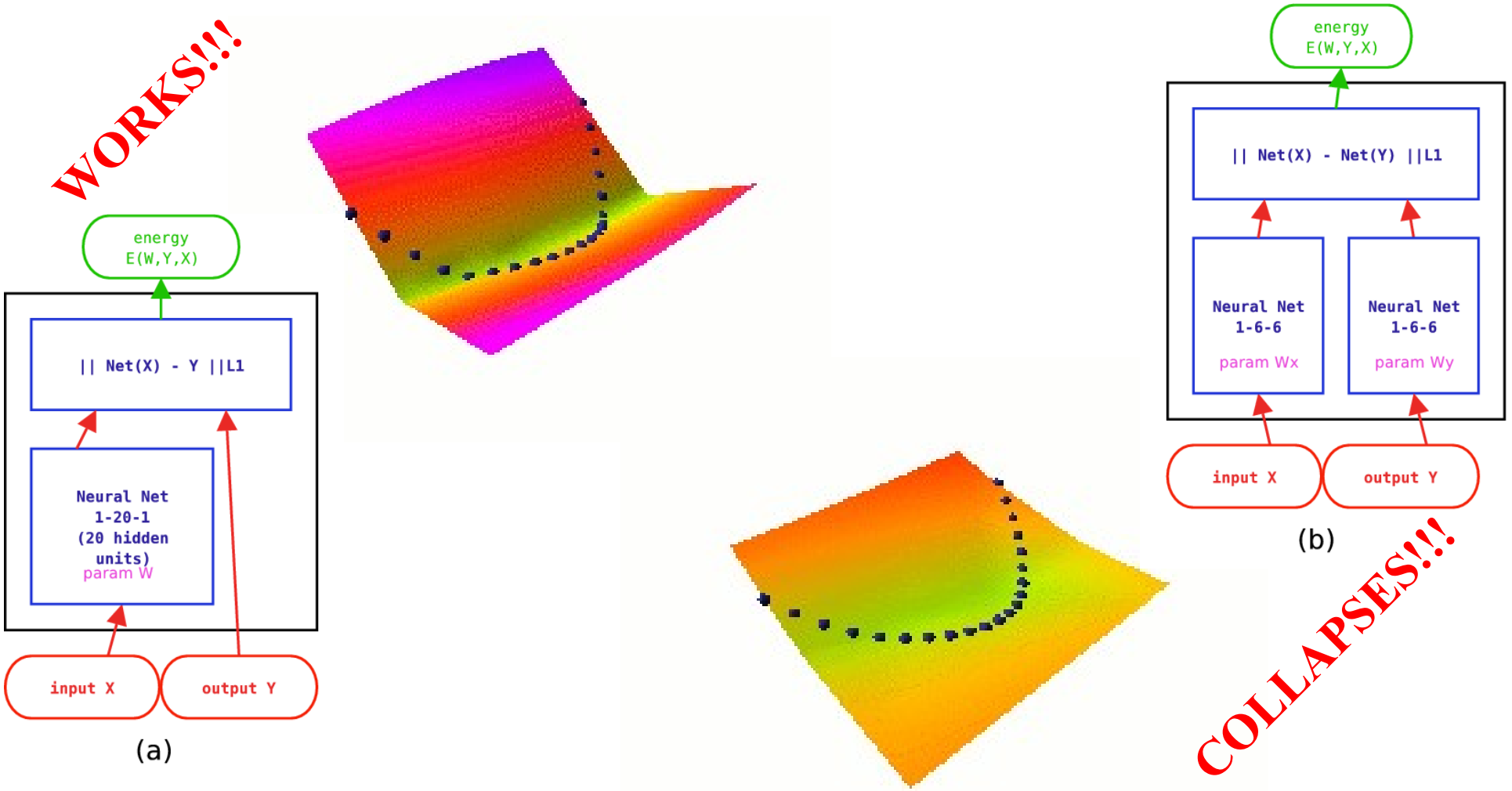
- **1. Design an architecture:** a particular form for E(W,Y,X).
- **2. Pick an inference algorithm for Y:** MAP or conditional distribution, belief prop, min cut, variational methods, gradient descent, MCMC, HMC.....
- **3. Pick a loss function:** in such a way that minimizing it with respect to W over a training set will make the inference algorithm find the correct Y for a given X.
- **4. Pick an optimization method.**

- **PROBLEM: What loss functions will make the machine approach the desired behavior?**

Examples of Loss Functions: Energy Loss

Energy Loss $$L_{energy}(Y^i, E(W, \mathcal{Y}, X^i)) = E(W, Y^i, X^i).$$

- Simply pushes down on the energy of the correct answer



Negative Log-Likelihood Loss

Conditional probability of the samples (assuming independence)

$$P(Y^1, \ldots, Y^P | X^1, \ldots, X^P, W) = \prod_{i=1}^{P} P(Y^i | X^i, W).$$

$$-\log \prod_{i=1}^{P} P(Y^i | X^i, W) = \sum_{i=1}^{P} -\log P(Y^i | X^i, W).$$

Gibbs distribution: $$P(Y | X^i, W) = \frac{e^{-\beta E(W, Y, X^i)}}{\int_{y \in \mathcal{Y}} e^{-\beta E(W, y, X^i)}}.$$

$$-\log \prod_{i=1}^{P} P(Y^i | X^i, W) = \sum_{i=1}^{P} \beta E(W, Y^i, X^i) + \log \int_{y \in \mathcal{Y}} e^{-\beta E(W, y, X^i)}.$$

We get the NLL loss by dividing by P and Beta:

$$\mathcal{L}_{\text{nll}}(W, \mathcal{S}) = \frac{1}{P} \sum_{i=1}^{P} \left(E(W, Y^i, X^i) + \frac{1}{\beta} \log \int_{y \in \mathcal{Y}} e^{-\beta E(W, y, X^i)} \right).$$

Reduces to the perceptron loss when Beta->infinity

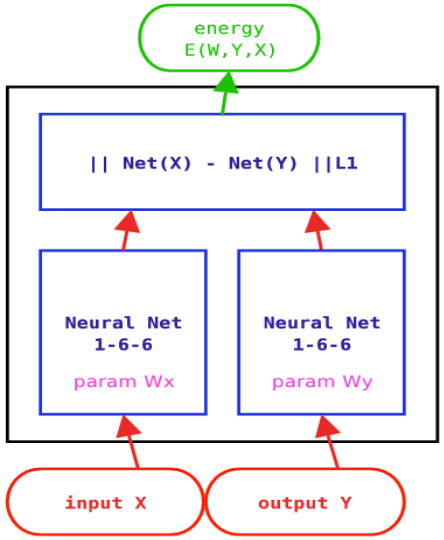
Negative Log-Likelihood Loss

Pushes down on the energy of the correct answer

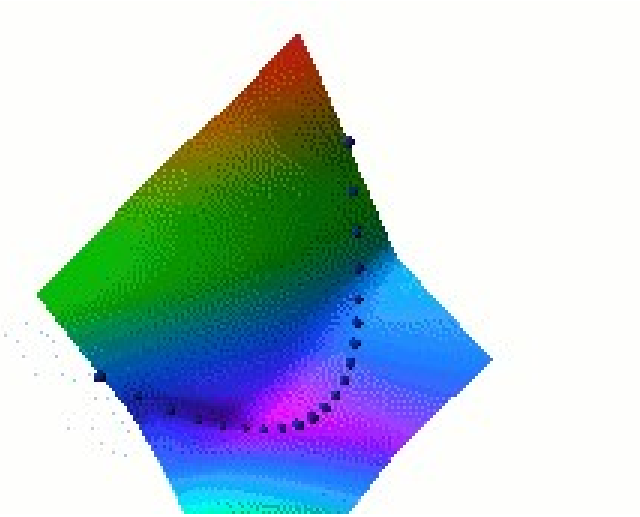
Pulls up on the energies of all answers in proportion to their probability

$$\mathcal{L}_{\text{nll}}(W, \mathcal{S}) = \frac{1}{P} \sum_{i=1}^{P} \left(E(W, Y^i, X^i) + \frac{1}{\beta} \log \int_{y \in \mathcal{Y}} e^{-\beta E(W, y, X^i)} \right).$$

$$\frac{\partial L_{\text{nll}}(W, Y^i, X^i)}{\partial W} = \frac{\partial E(W, Y^i, X^i)}{\partial W} - \int_{Y \in \mathcal{Y}} \frac{\partial E(W, Y, X^i)}{\partial W} P(Y | X^i, W),$$



(b)



Negative Log-Likelihood Loss

A probabilistic model is an EBM in which:

- The energy can be integrated over Y (the variable to be predicted)
- The loss function is the negative log-likelihood

Negative Log Likelihood Loss has been used for a long time in many communities for discriminative learning with structured outputs

- Speech recognition: many papers going back to the early 90's [Bengio 92], [Bourlard 94]. They call "Maximum Mutual Information"
- Handwriting recognition [Bengio LeCun 94], [LeCun et al. 98]
- Bio-informatics [Haussler]
- Conditional Random Fields [Lafferty et al. 2001]
- Lots more.....
- In all the above cases, it was used with non-linearly parameterized energies.

A Simpler Loss Functions:Perceptron Loss

$$L_{perceptron}(Y^i, E(W, \mathcal{Y}, X^i)) = E(W, Y^i, X^i) - \min_{Y \in \mathcal{Y}} E(W, Y, X^i).$$

Perceptron Loss [LeCun et al. 1998], [Collins 2002]

- Pushes down on the energy of the correct answer
- Pulls up on the energy of the machine's answer
- Always positive. Zero when answer is correct
- No "margin": technically does not prevent the energy surface from being almost flat.
- Works pretty well in practice, particularly if the energy parameterization does not allow flat surfaces.
- This is often called **"discriminative Viterbi training"** in the speech and handwriting literature

Perceptron Loss for Binary Classification

$$L_{perceptron}(Y^i, E(W, \mathcal{Y}, X^i)) = E(W, Y^i, X^i) - \min_{Y \in \mathcal{Y}} E(W, Y, X^i).$$

Energy: $$E(W, Y, X) = -Y G_W(X),$$

Inference: $$Y^* = \text{argmin}_{Y \in \{-1,1\}} - Y G_W(X) = \text{sign}(G_W(X)).$$

Loss: $$\mathcal{L}_{\text{perceptron}}(W, \mathcal{S}) = \frac{1}{P} \sum_{i=1}^{P} \left(\text{sign}(G_W(X^i)) - Y^i\right) G_W(X^i).$$

Learning Rule: $$W \leftarrow W + \eta \left(Y^i - \text{sign}(G_W(X^i)\right) \frac{\partial G_W(X^i)}{\partial W},$$

If Gw(X) is linear in W: $$E(W, Y, X) = -Y W^T \Phi(X)$$

$$W \leftarrow W + \eta \left(Y^i - \text{sign}(W^T \Phi(X^i))\right) \Phi(X^i)$$

A Better Loss Function: Generalized Margin Losses

First, we need to define the Most Offending Incorrect Answer

Most Offending Incorrect Answer: discrete case

Definition 1 *Let Y be a discrete variable. Then for a training sample (X^i, Y^i), the* ***most offending incorrect answer*** *$\bar{Y}^i$ is the answer that has the lowest energy among all answers that are incorrect:*

$$\bar{Y}^i = \text{argmin}_{Y \in \mathcal{Y} and Y \neq Y^i} E(W, Y, X^i). \quad (8)$$

Most Offending Incorrect Answer: continuous case

Definition 2 *Let Y be a continuous variable. Then for a training sample (X^i, Y^i), the* ***most offending incorrect answer*** *$\bar{Y}^i$ is the answer that has the lowest energy among all answers that are at least ϵ away from the correct answer:*

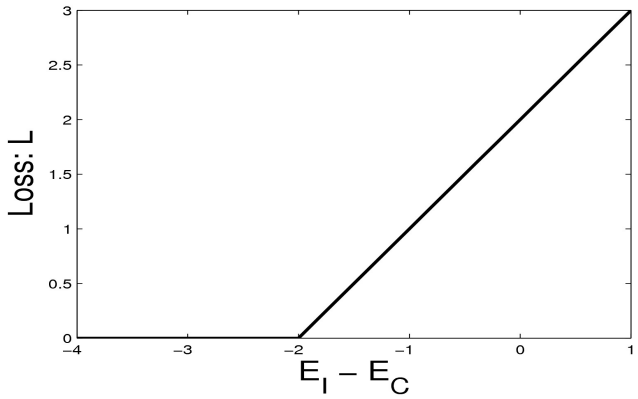
$$\bar{Y}^i = \text{argmin}_{Y \in \mathcal{Y}, \|Y - Y^i\| > \epsilon} E(W, Y, X^i). \quad (9)$$

Examples of Generalized Margin Losses

$$L_{\text{hinge}}(W, Y^i, X^i) = \max\left(0, m + E(W, Y^i, X^i) - E(W, \bar{Y}^i, X^i)\right),$$

Hinge Loss

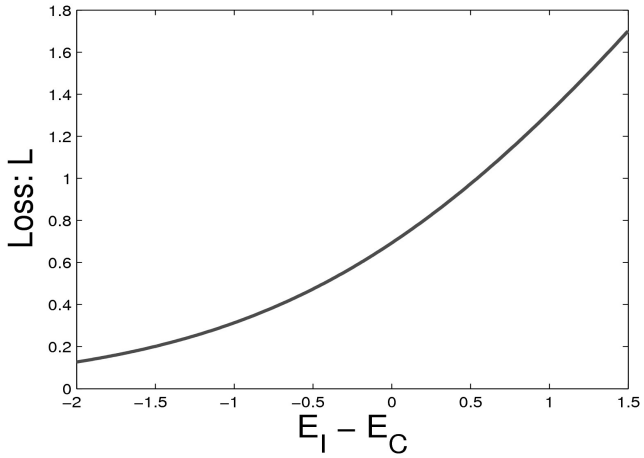
- [Altun et al. 2003], [Taskar et al. 2003]
- With the linearly-parameterized binary classifier architecture, we get linear SVMs



$$L_{\text{log}}(W, Y^i, X^i) = \log\left(1 + e^{E(W,Y^i,X^i) - E(W,\bar{Y}^i,X^i)}\right).$$

Log Loss

- "soft hinge" loss
- With the linearly-parameterized binary classifier architecture, we get linear Logistic Regression

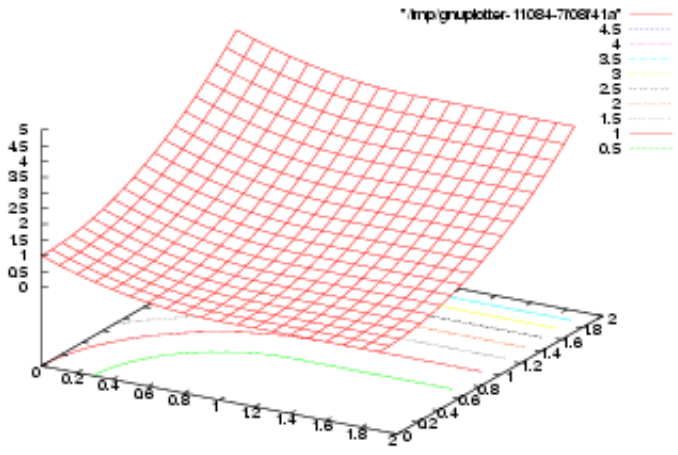


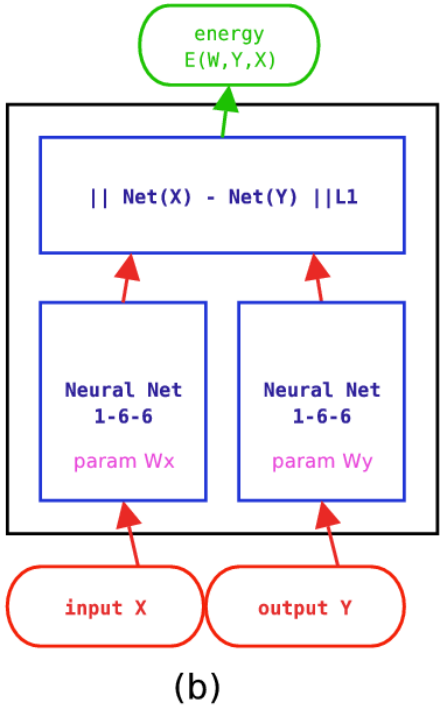
Examples of Margin Losses: Square-Square Loss

$$L_{\text{sq-sq}}(W, Y^i, X^i) = E(W, Y^i, X^i)^2 + \left(\max(0, m - E(W, \bar{Y}^i, X^i))\right)^2 .$$

Square-Square Loss

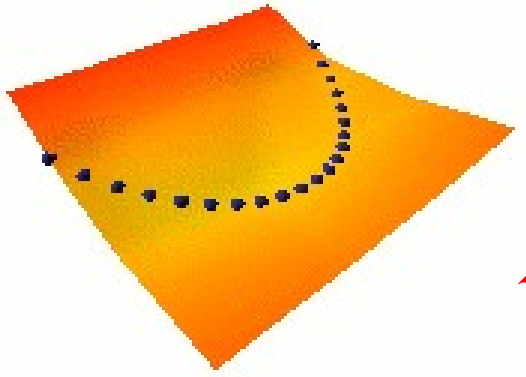
- [LeCun-Huang 2005]
- Appropriate for positive energy functions





(b)

Learning Y = X^2



NO COLLAPSE!!!

Other Margin-Like Losses

LVQ2 Loss [Kohonen, Oja], Driancourt-Bottou 1991]

$$L_{\text{lvq2}}(W, Y^i, X^i) = \min\left(1, \max\left(0, \frac{E(W, Y^i, X^i) - E(W, \bar{Y}^i, X^i)}{\delta E(W, \bar{Y}^i, X^i)}\right)\right),$$

Minimum Classification Error Loss [Juang, Chou, Lee 1997]

$$L_{\text{mce}}(W, Y^i, X^i) = \sigma\left(E(W, Y^i, X^i) - E(W, \bar{Y}^i, X^i)\right),$$

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

Square-Exponential Loss [Osadchy, Miller, LeCun 2004]

$$L_{\text{sq-exp}}(W, Y^i, X^i) = E(W, Y^i, X^i)^2 + \gamma e^{-E(W, \bar{Y}^i, X^i)},$$

What Make a "Good" Loss Function

Good and bad loss functions

Loss (equation #)	Formula	Margin
energy loss	$E(W, Y^i, X^i)$	none
perceptron	$E(W, Y^i, X^i) - \min_{Y \in \mathcal{Y}} E(W, Y, X^i)$	0
hinge	$\max\left(0, m + E(W, Y^i, X^i) - E(W, \bar{Y}^i, X^i)\right)$	m
log	$\log\left(1 + e^{E(W,Y^i,X^i) - E(W,\bar{Y}^i,X^i)}\right)$	> 0
LVQ2	$\min\left(M, \max(0, E(W, Y^i, X^i) - E(W, \bar{Y}^i, X^i)\right)$	0
MCE	$\left(1 + e^{-\left(E(W,Y^i,X^i) - E(W,\bar{Y}^i,X^i)\right)}\right)^{-1}$	> 0
square-square	$E(W, Y^i, X^i)^2 - \left(\max(0, m - E(W, \bar{Y}^i, X^i))\right)^2$	m
square-exp	$E(W, Y^i, X^i)^2 + \beta e^{-E(W,\bar{Y}^i,X^i)}$	> 0
NLL/MMI	$E(W, Y^i, X^i) + \frac{1}{\beta} \log \int_{y \in \mathcal{Y}} e^{-\beta E(W,y,X^i)}$	> 0
MEE	$1 - e^{-\beta E(W,Y^i,X^i)} / \int_{y \in \mathcal{Y}} e^{-\beta E(W,y,X^i)}$	> 0

Slightly more general form:

$$L(W, X^i, Y^i) = \sum_y H\left(E(W, Y^i, X^i) - E(W, y, X^i) + C(Y^i, y)\right)$$

Advantages/Disadvantages of various losses

Loss functions differ in how they pick the point(s) whose energy is pulled up, and how much they pull them up

Losses with a log partition function in the contrastive term pull up all the bad answers simultaneously.

- This may be good if the gradient of the contrastive term can be computed efficiently
- This may be bad if it cannot, in which case we might as well use a loss with a single point in the contrastive term

Variational methods pull up many points, but not as many as with the full log partition function.

Efficiency of a loss/architecture: how many energies are pulled up for a given amount of computation?

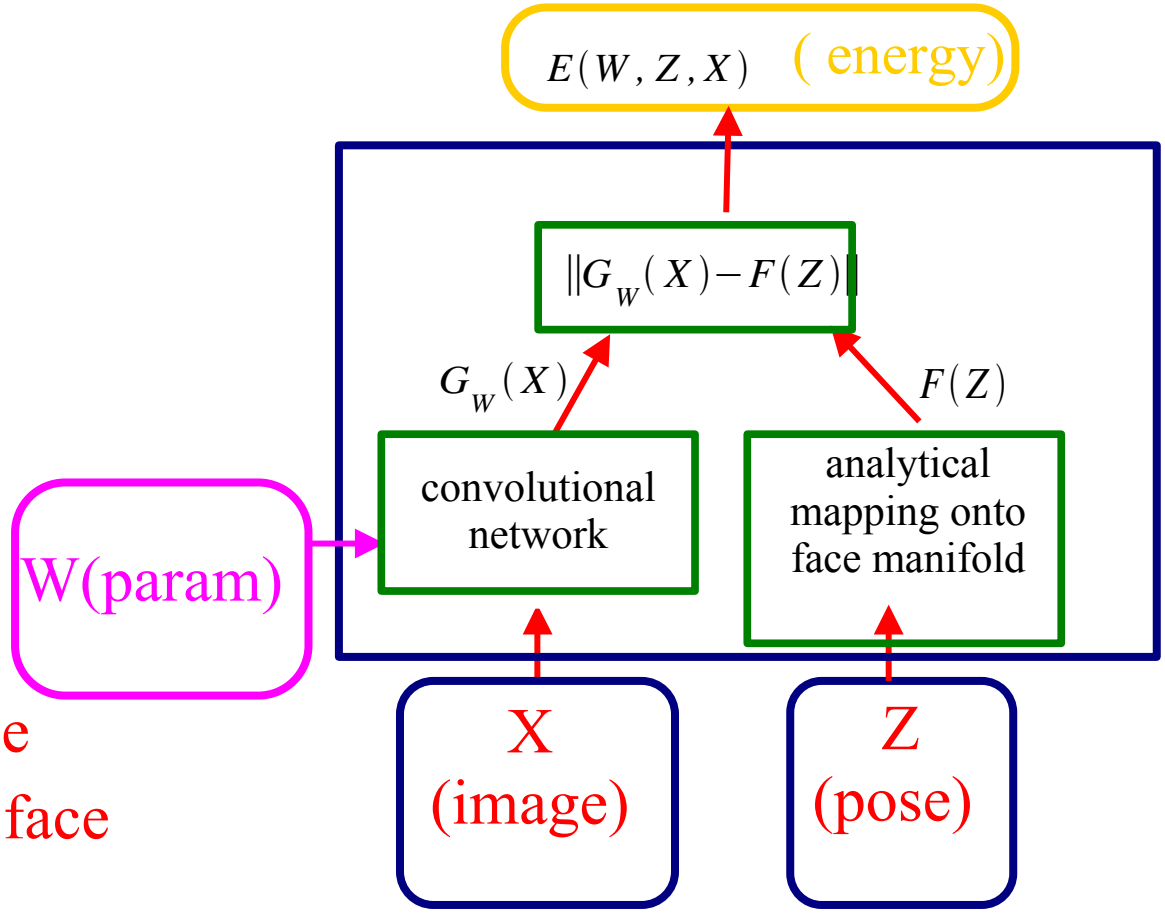
- The theory for this is to be developed

Face Detection and Pose Estimation with EBM

- **Training:** 52,850, 32x32 grey-level images of faces, 52,850 selected non-faces.
- Each training image was used 5 times with random variation in scale, in-plane rotation, brightness and contrast.
- **2nd phase:** half of the initial negative set was replaced by false positives of the initial version of the detector .

$$E^*(W,X) = \min_Z ||G_W(X) - F(Z)||$$

$$Z^* = \operatorname{argmin}_Z ||G_W(X) - F(Z)||$$

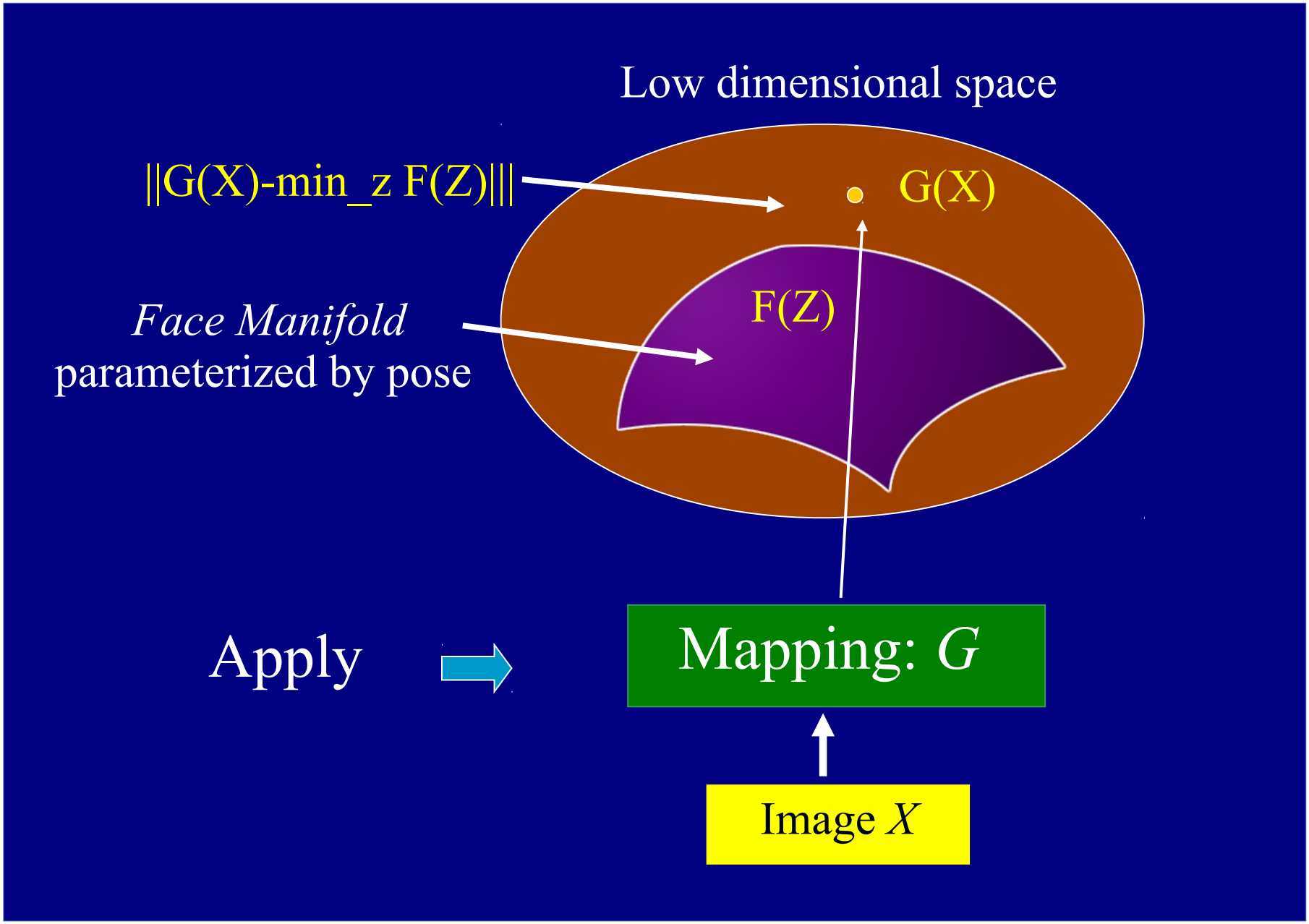


Small E*(W,X): face

Large E*(W,X): no face

[Osadchy, Miller, LeCun, NIPS 2004]

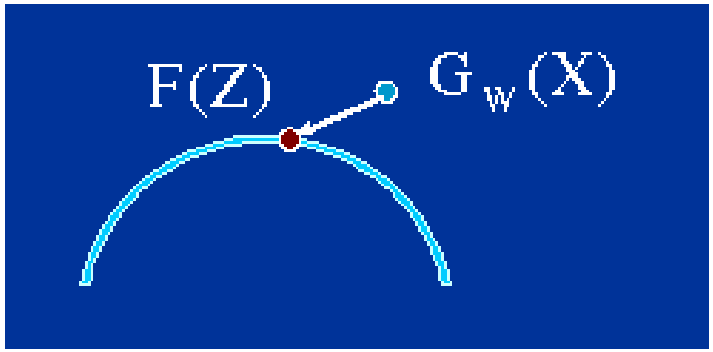
Face Manifold



Energy-Based Contrastive Loss Function

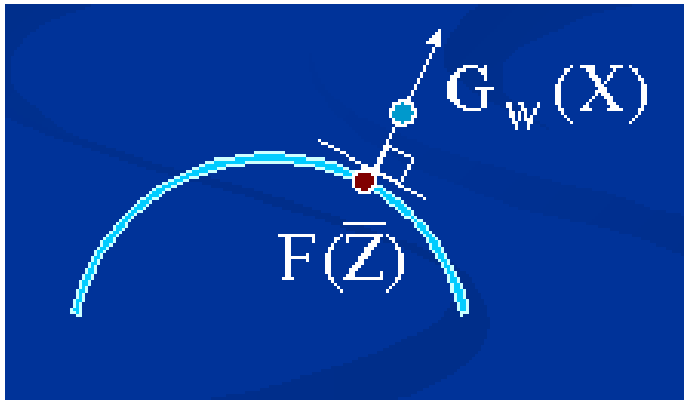
$$\mathcal{L}(W) = \frac{1}{|\mathrm{f}+\mathrm{p}|} \sum_{X,Z \in \mathrm{faces+pose}} \left[L^+\left(E(W,Z,X)\right)\right] + L^-\left(\min_{X,Z \in \mathrm{bckgnd,poses}} E(W,Z,X)\right)$$

$$L^+\left(E(W,Z,X)\right) = E(W,Z,X)^2 = ||G_W(X) - F(Z)||^2$$



Attract the network output Gw(X) to the location of the desired pose F(Z) on the manifold

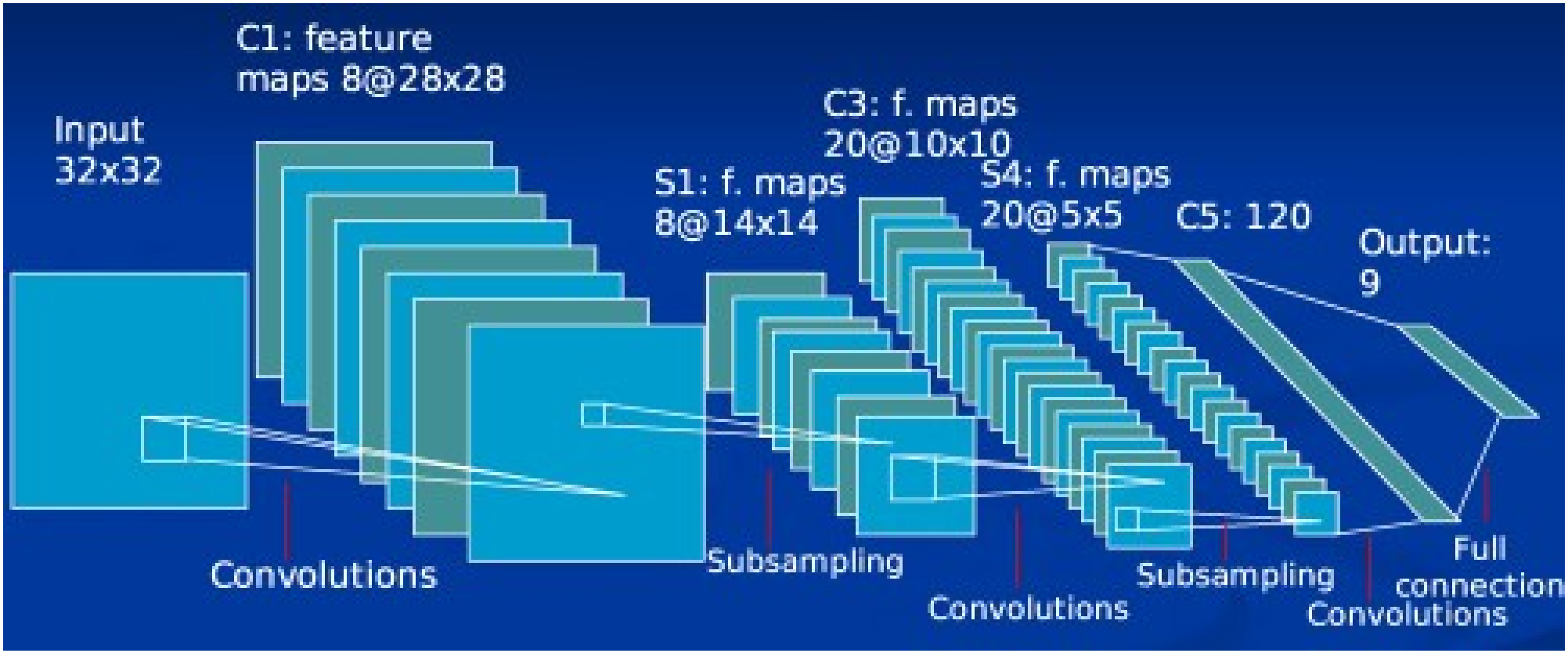
$$L^-\left(\min_{X,Z \in \mathrm{bckgnd,poses}} E(W,Z,X)\right) = K \exp\left(-\min_{X,Z \in \mathrm{bckgnd,poses}} ||G_W(X) - F(Z)||\right)$$



Repel the network output Gw(X) away from the face/pose manifold

Convolutional Net

[LeCun et al. 1988, 1989, 1998, 2005]



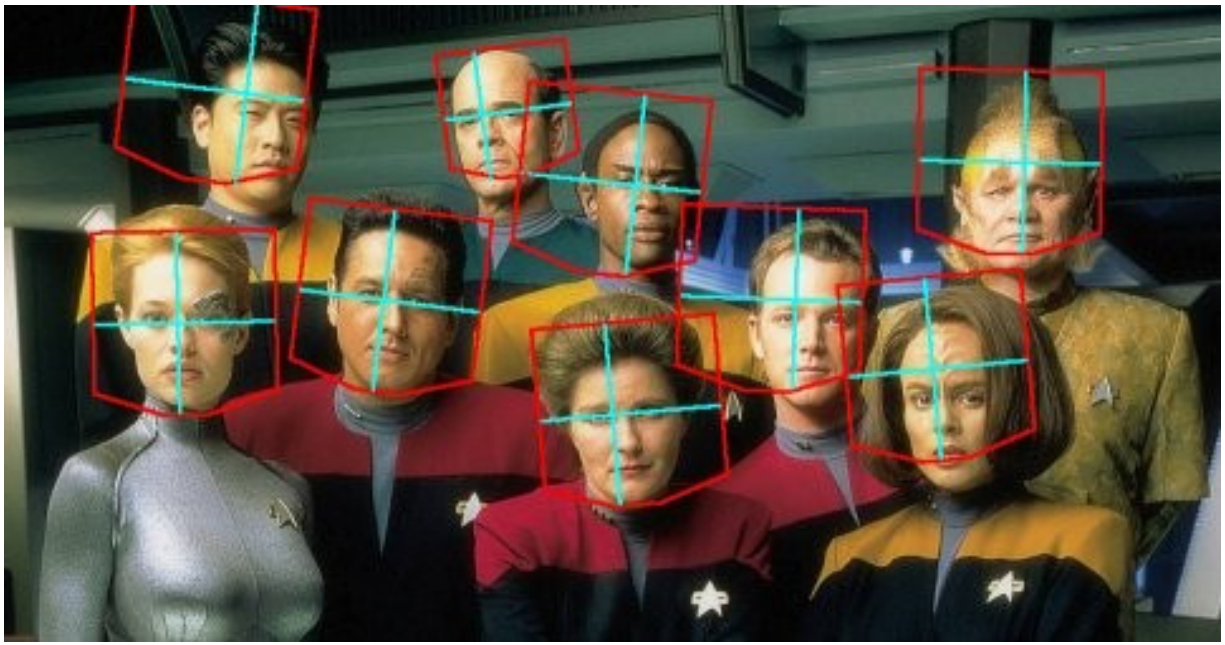
Hierarchy of local filters (convolution kernels),

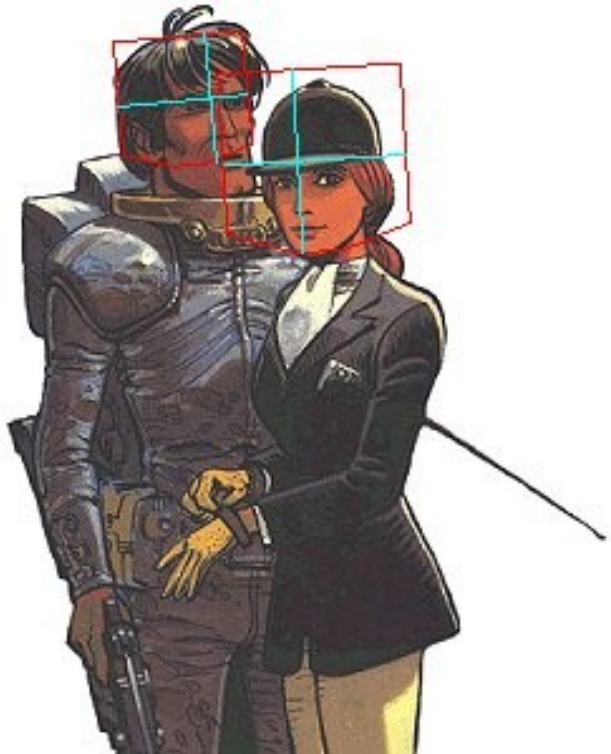
sigmoid pointwise non-linearities, and spatial subsampling

All the filter coefficients are learned with gradient descent (back-prop)

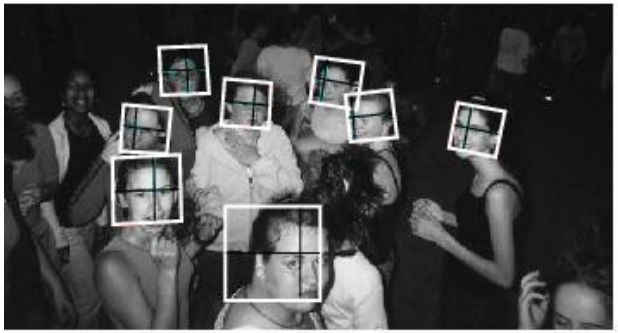
Face Detection Results

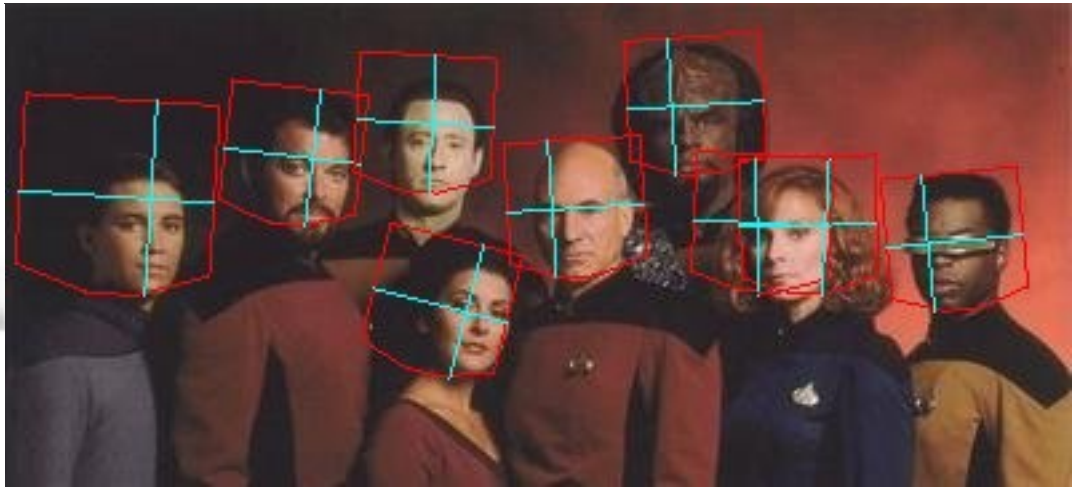
Data Set->	TILTED		PROFILE		MIT+CMU	
False positives per image->	4.42	26.9	0.47	3.36	0.5	1.28
Our Detector	90%	97%	67%	83%	83%	88%
Jones & Viola (tilted)	90%	95%	x		x	
Jones & Viola (profile)	x		70%	83%	x	

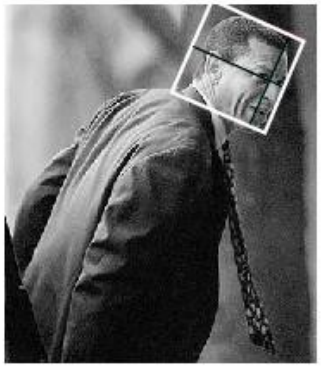


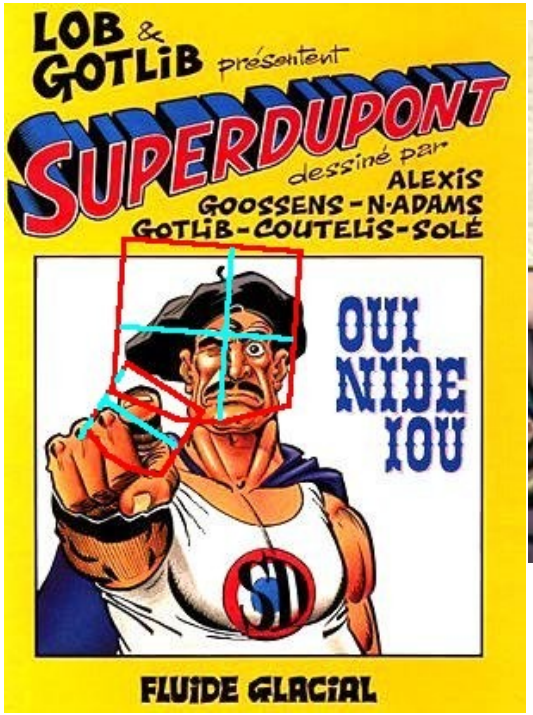


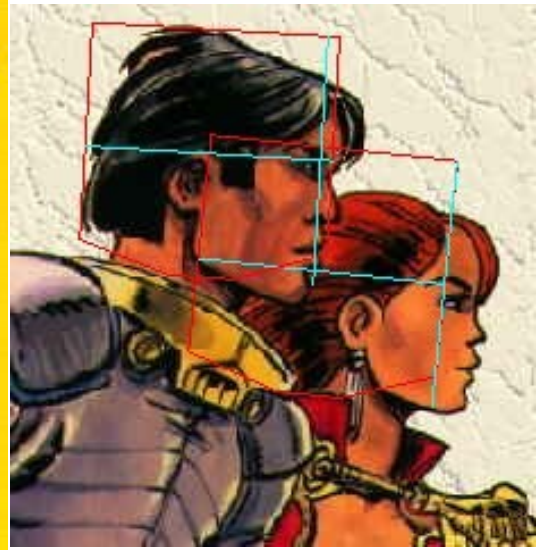
Face Detection and Pose Estimation Results









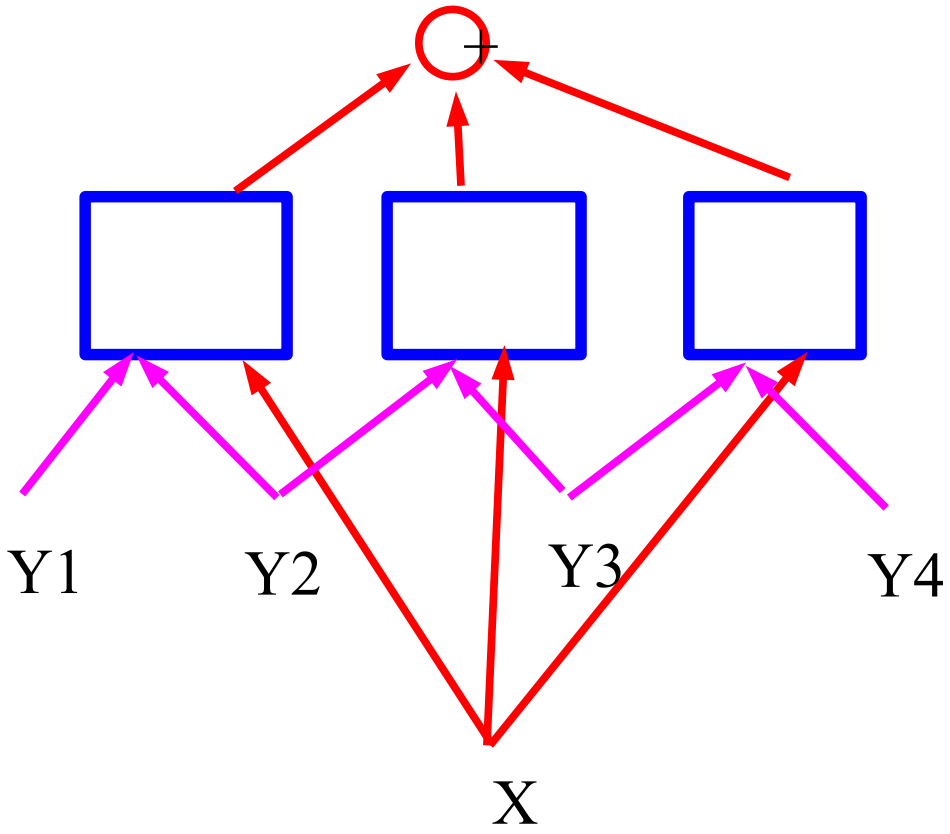


Energy-Based Factor Graphs: Energy = Sum of "factors"

Sequence Labeling

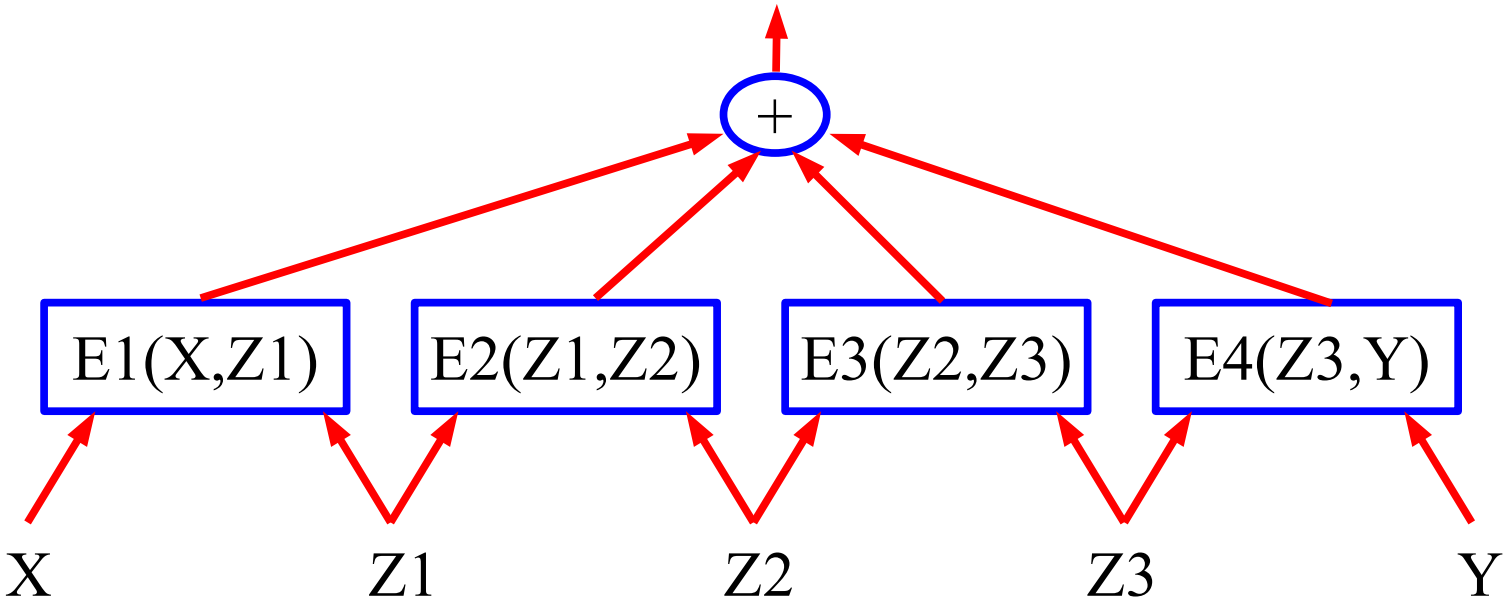
- Output is a sequence Y1,Y2,Y3,Y4.....
- NLP parsing, MT, speech/handwriting recognition, biological sequence analysis
- The factors ensure grammatical consistency
- They give low energy to consistent sub-sequences of output symbols
- The graph is generally simple (chain or tree)
- Inference is easy (dynamic programming, min-sum)

$$Y^* = \text{argmin}_{Y \in \mathcal{Y}, Z \in \mathcal{Z}} E(Z, Y, X).$$



When the energy is a sum of partial energy functions (or when the probability is a product of factors):

- Efficient inference algorithms can be used for inference (without the normalization step).



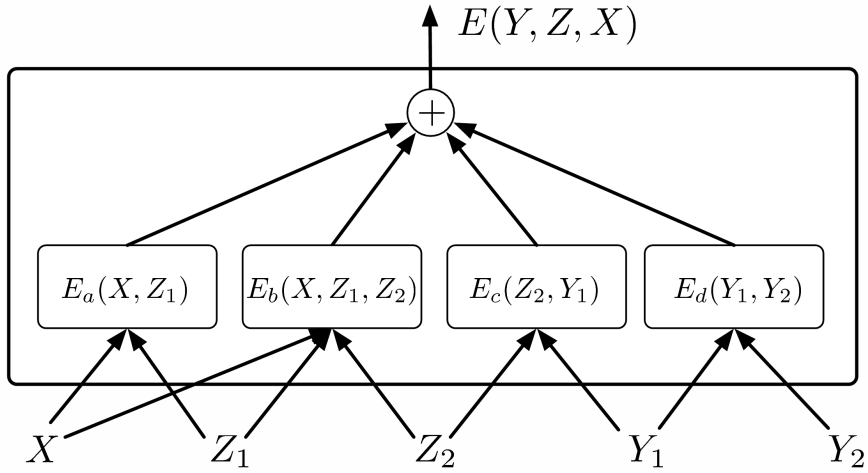
Efficient Inference: Energy-Based Factor Graphs

Example:

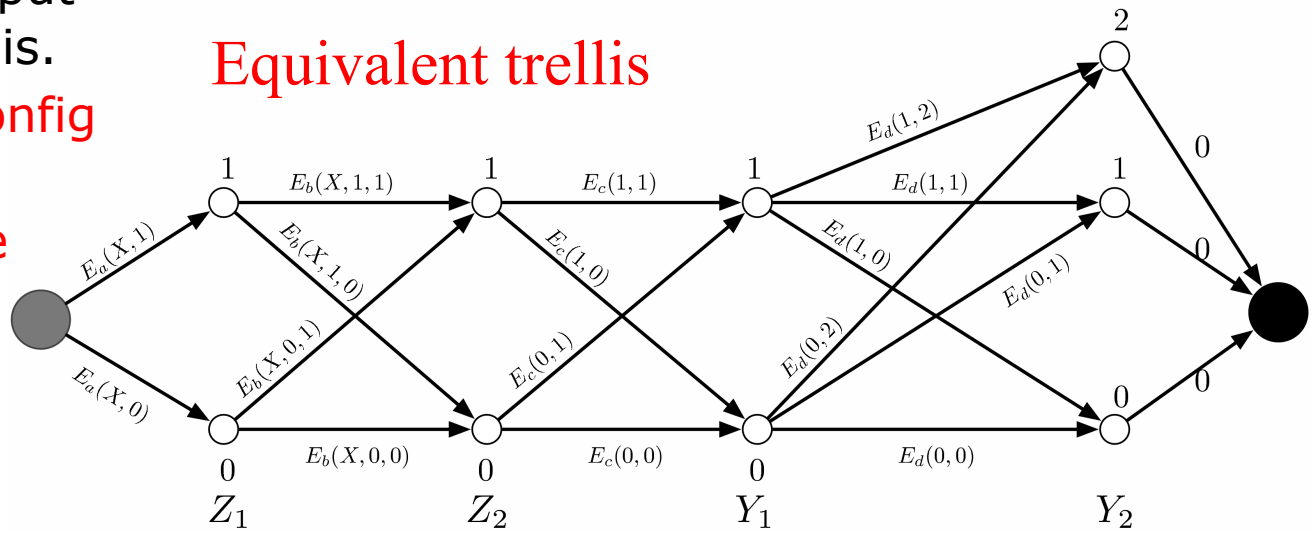
- Z1, Z2, Y1 are binary
- Z2 is ternary
- A naïve exhaustive inference would require 2x2x2x3=24 energy evaluations (= 96 factor evaluations)
- BUT: Ea only has 2 possible input configurations, Eb and Ec have 4, and Ed 6.
- Hence, we can precompute the 16 factor values, and put them on the arcs in a trellis.
- A path in the trellis is a config of variable
- The cost of the path is the energy of the config

The energy is a sum of "factor" functions

Factor graph



Equivalent trellis



Energy-Based Belief Prop

The previous picture shows a chain graph of factors with 2 inputs.

The extension of this procedure to trees, with factors that can have more than 2 inputs the "min-sum" algorithm (a non-probabilistic form of belief propagation)

Basically, it is the sum-product algorithm with a different semi-ring algebra (min instead of sum, sum instead of product), and no normalization step.

- [Kschischang, Frey, Loeliger, 2001][McKay's book]

Simple Energy-Based Factor Graphs with "Shallow" Factors

Linearly Parameterized Factors

with the NLL Loss :

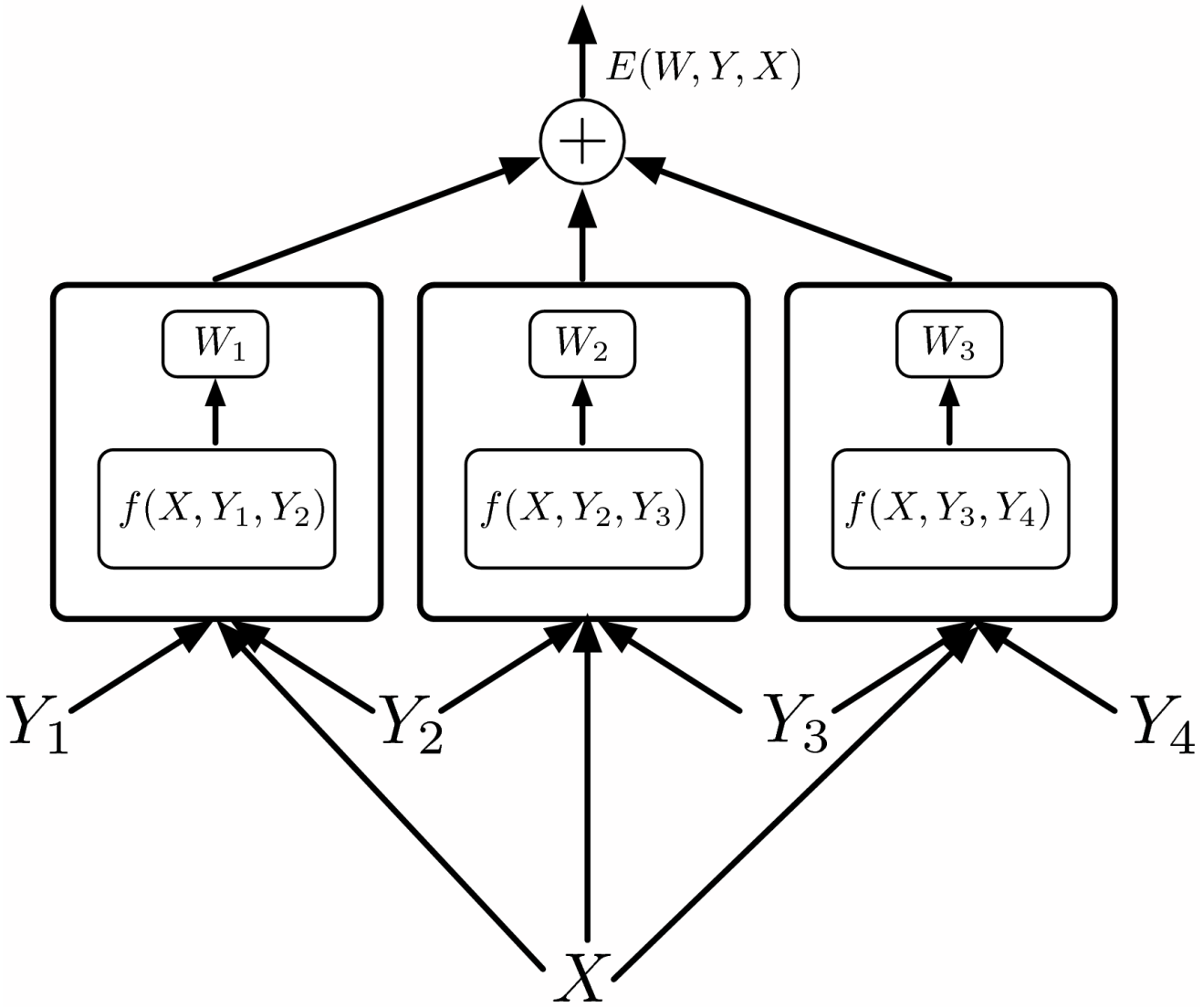
- Lafferty's **Conditional Random Field**

with Hinge Loss:

- Taskar and Altun/Hofmann's **Max Margin Markov Nets** and **Latent SVM**

with Perceptron Loss

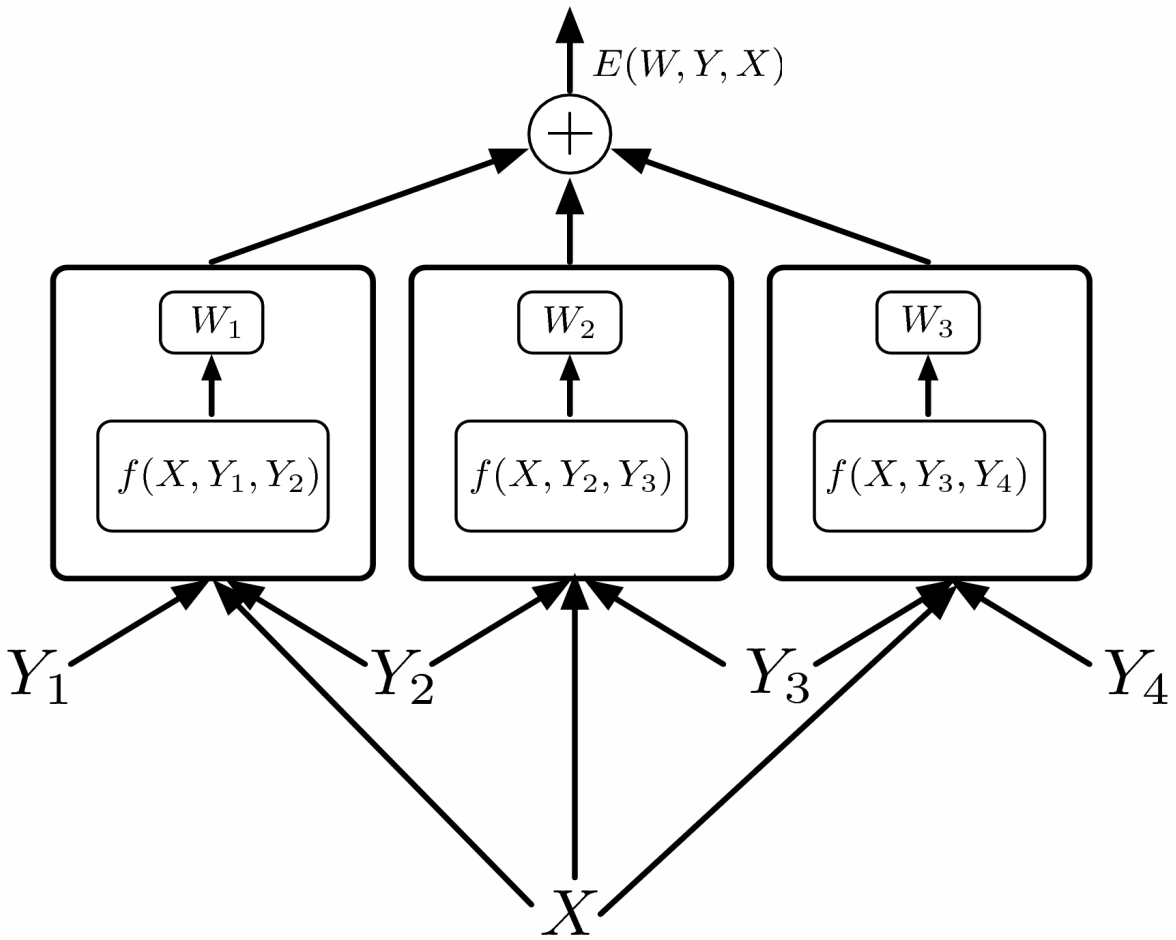
- Collins's **Structured Perceptron** model



Example : The Conditional Random Field Architecture

A CRF is an energy-based factor graph in which:

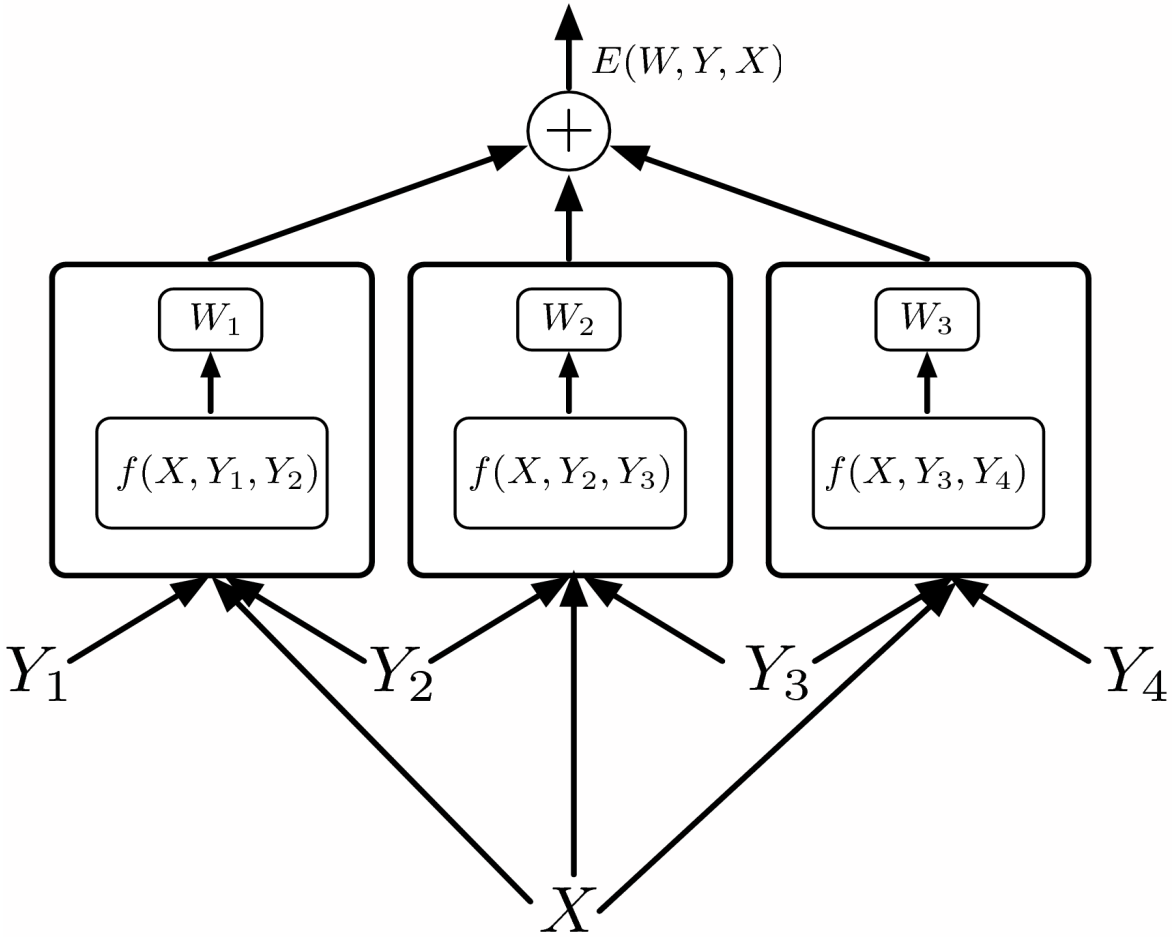
- the factors are **linear in the parameters** **(shallow factors)**
- The factors take neighboring output variables as inputs
- The factors are often all identical



Example : The Conditional Random Field Architecture

Applications:

- X is a sentence, Y is a sequence of Parts of Speech Tags (there is one Yi for each possible group of words).
- X is an image, Y is a set of labels for each window in the image (vegetation, building, sky....).



Deep/non-linear Factors for Speech and Handwriting

Trainable Speech/Handwriting Recognition systems that integrate Neural Nets (or other "deep" classifiers) with dynamic time warping, Hidden Markov Models, or other graph-based hypothesis representations

Training the feature extractor as part of the whole process.

with the LVQ2 Loss :

- Driancourt and Bottou's speech recognizer (1991)

with NLL:

- Bengio's speech recognizer (1992)
- Haffner's speech recognizer (1993)

With Minimum Empirical Error loss

- Ljolje and Rabiner (1990)

with NLL:

- Bengio (1992), Haffner (1993), Bourlard (1994)

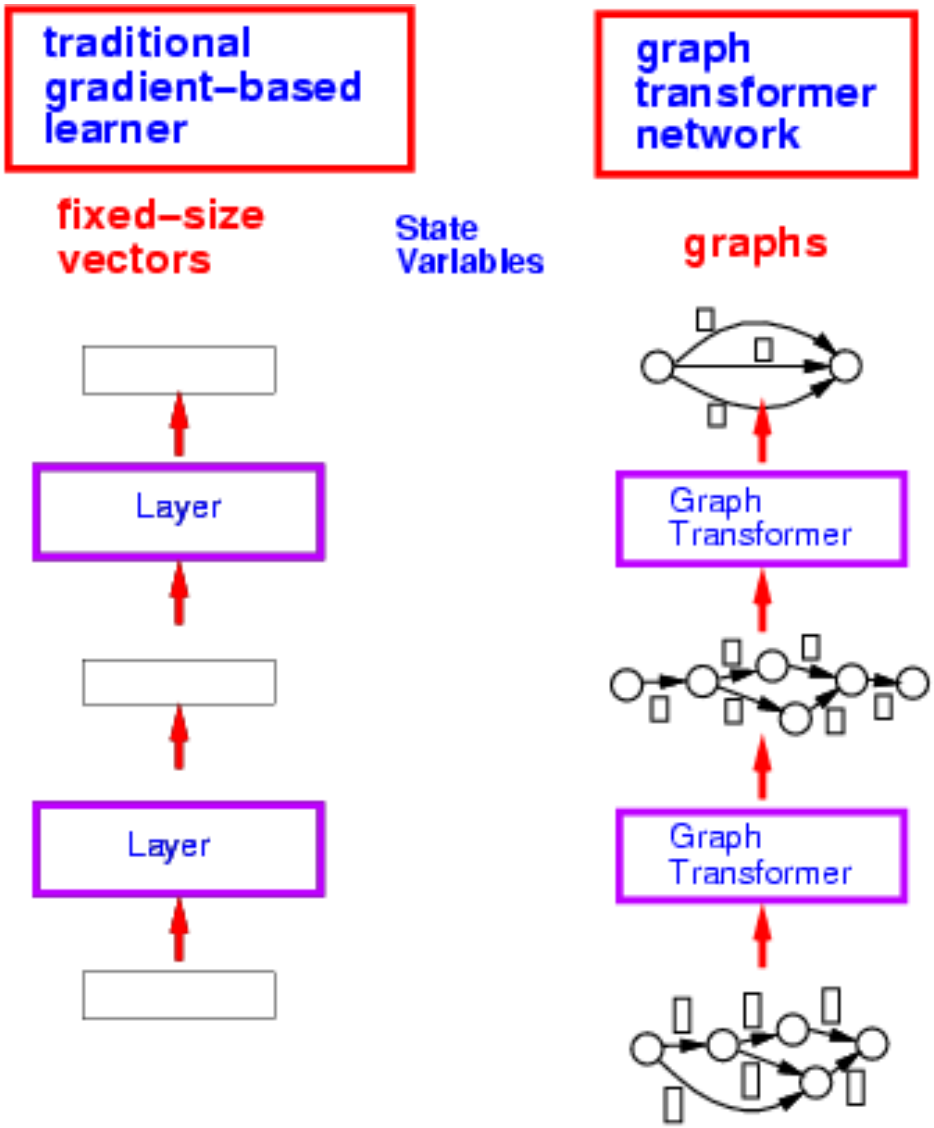
With MCE

- Juang et al. (1997)

Late normalization scheme (un-normalized HMM)

- Bottou pointed out the **label bias problem** (1991)
- Denker and Burges proposed a solution (1995)

Using Graphs instead of Vectors.



Whereas traditional learning machines manipulate fixed-size vectors, Graph Transformer Networks manipulate graphs.

Deep Factors & implicit graphs: GTN

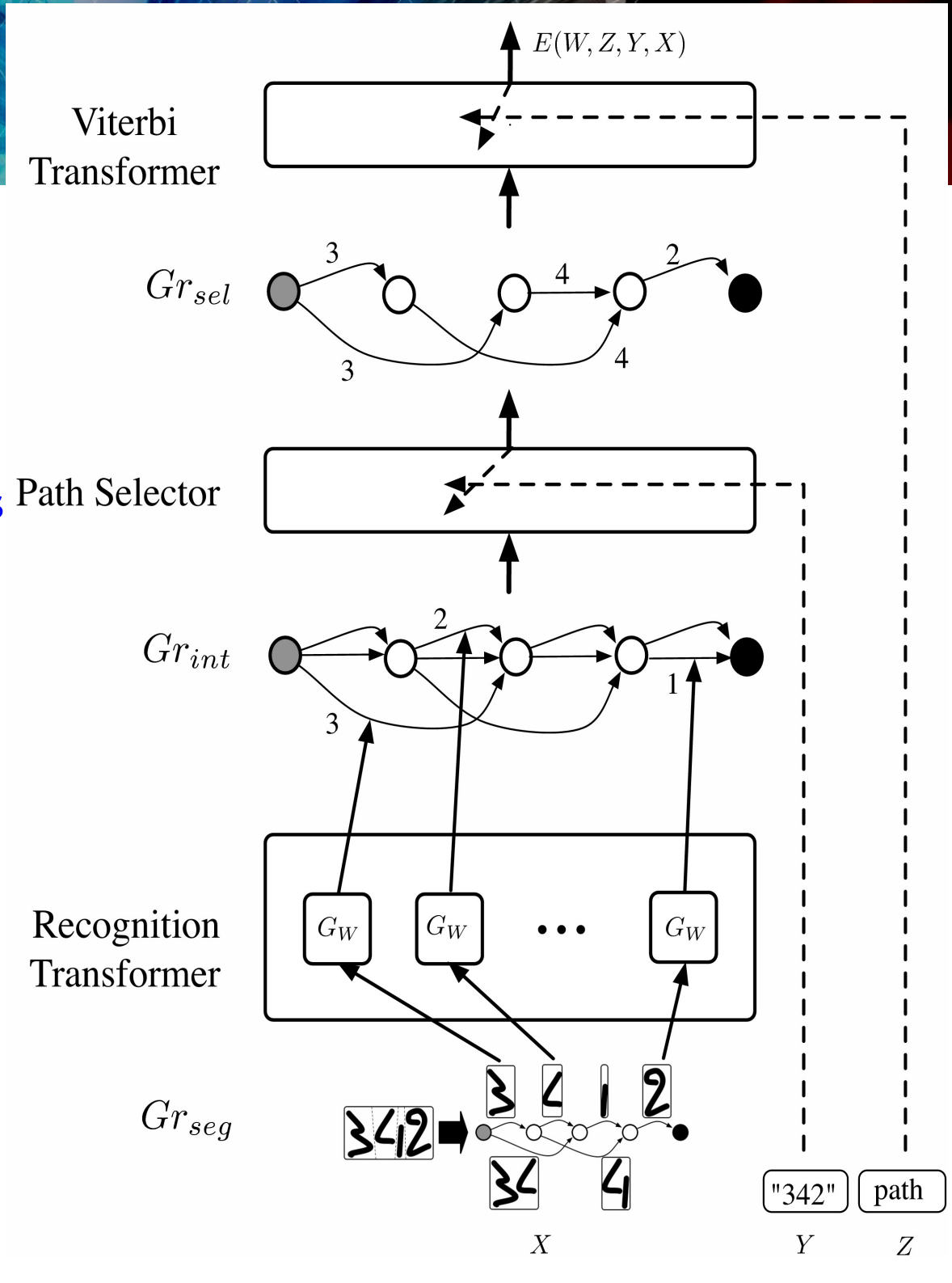
Handwriting Recognition with Graph Transformer Networks

Un-normalized hierarchical HMMs

- Trained with Perceptron loss [LeCun, Bottou, Bengio, Haffner 1998]
- Trained with NLL loss [Bengio, LeCun 1994], [LeCun, Bottou, Bengio, Haffner 1998]

Answer = sequence of symbols

Latent variable = segmentation



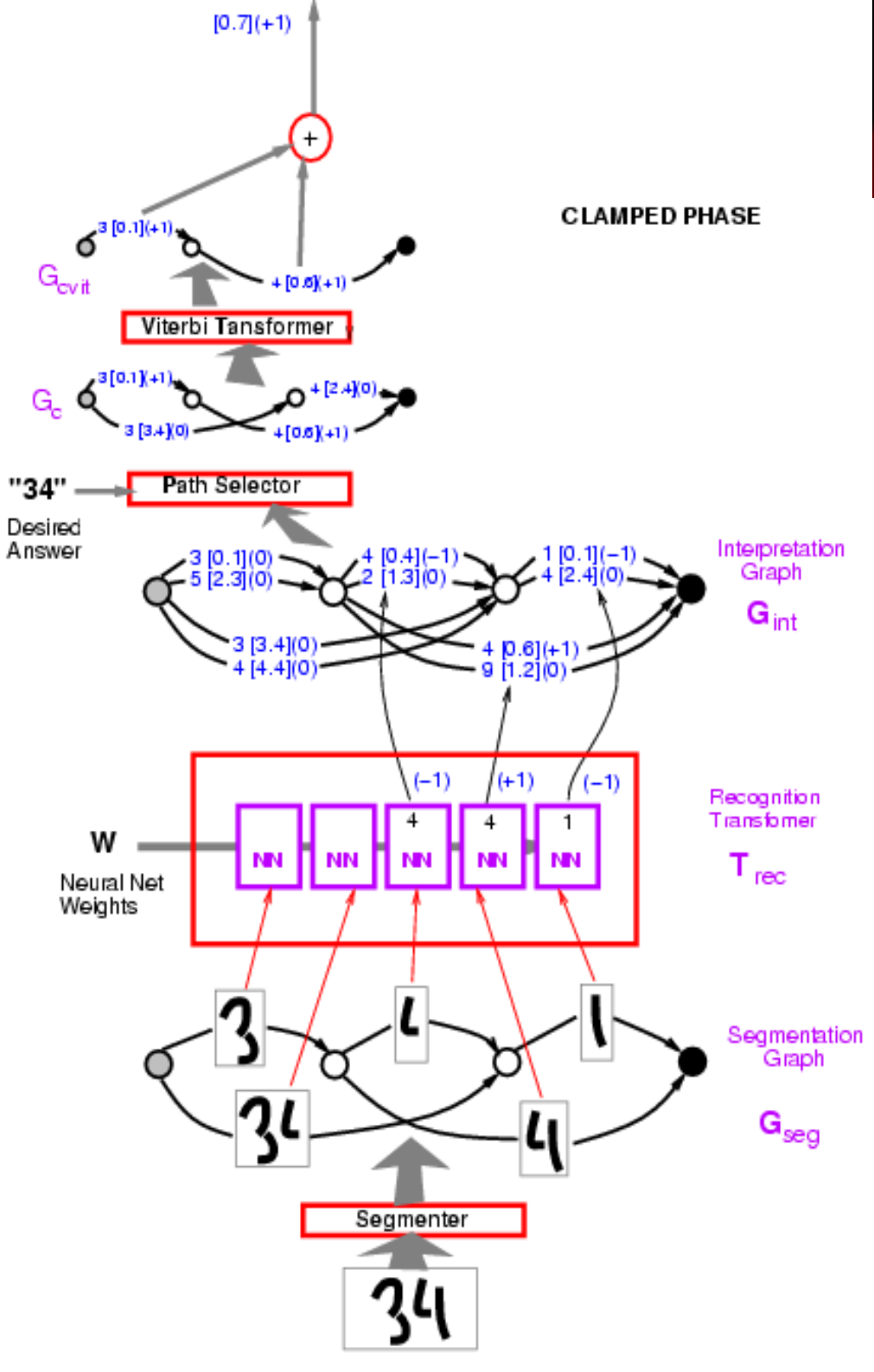
Graph Transformer Networks

Variables:

- X: input image
- Z: path in the interpretation graph/segmentation
- Y: sequence of labels on a path

Loss function: computing the energy of the desired answer:

$$E(W, Y, X)$$



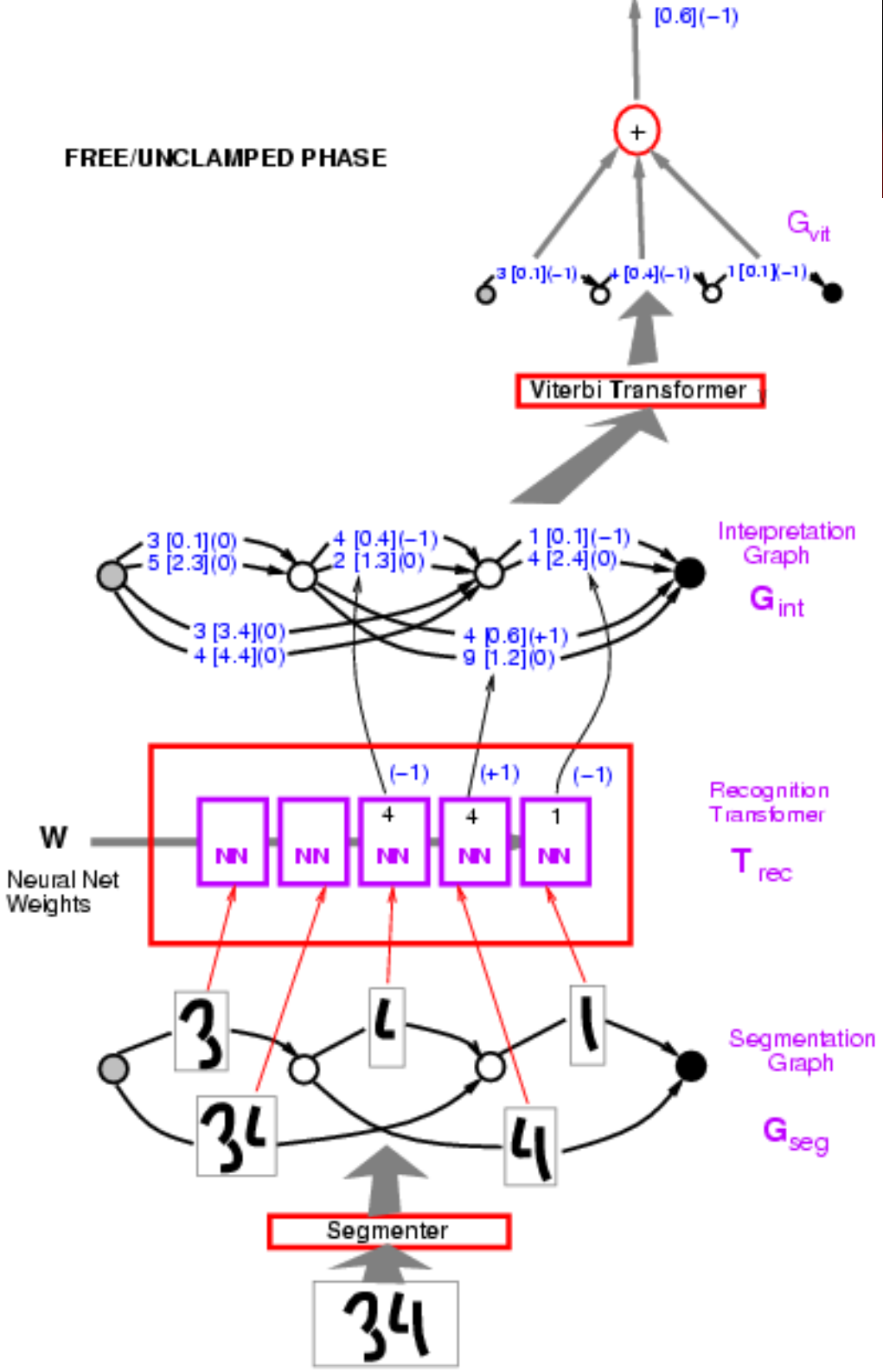
Graph Transformer Networks

Variables:

- X: input image
- Z: path in the interpretation graph/segmentation
- Y: sequence of labels on a path

Loss function: computing the constrastive term:

$$E(W, \check{Y}, X)$$

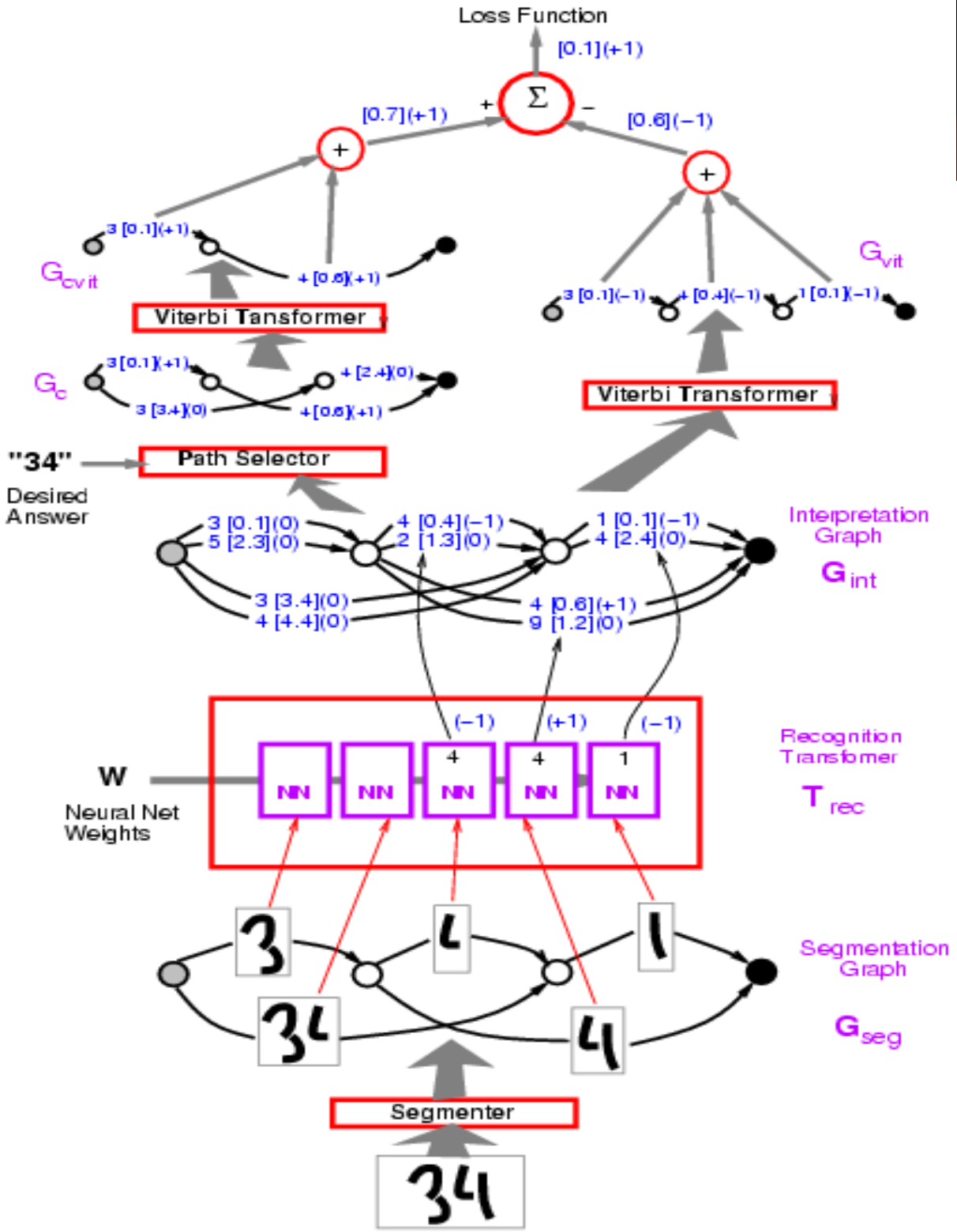


Graph Transformer Networks

Example: Perceptron loss

Loss = Energy of desired answer – Energy of best answer.

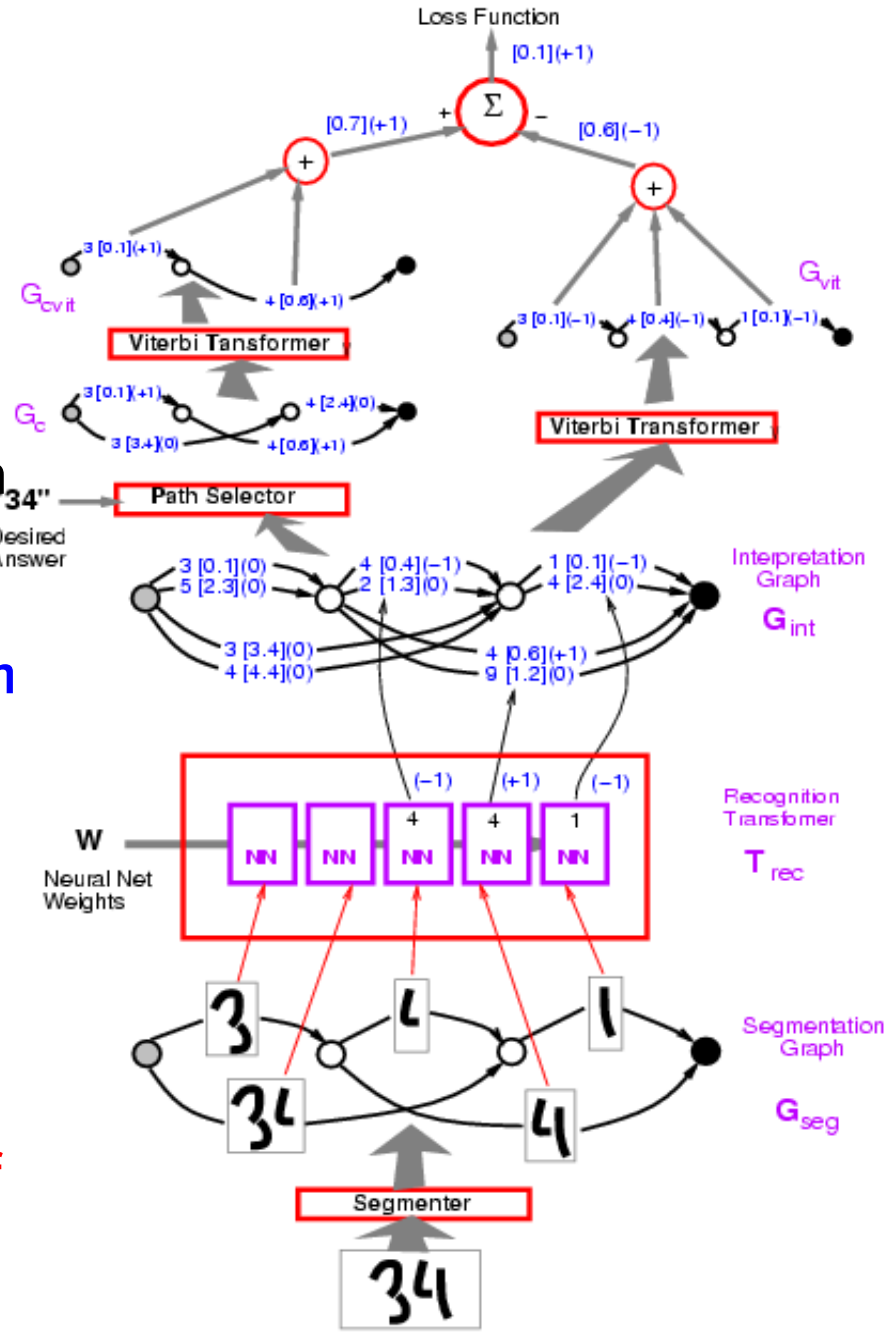
- (no margin)



Integrating Deep Learning and Structured Prediction

Y LeCun

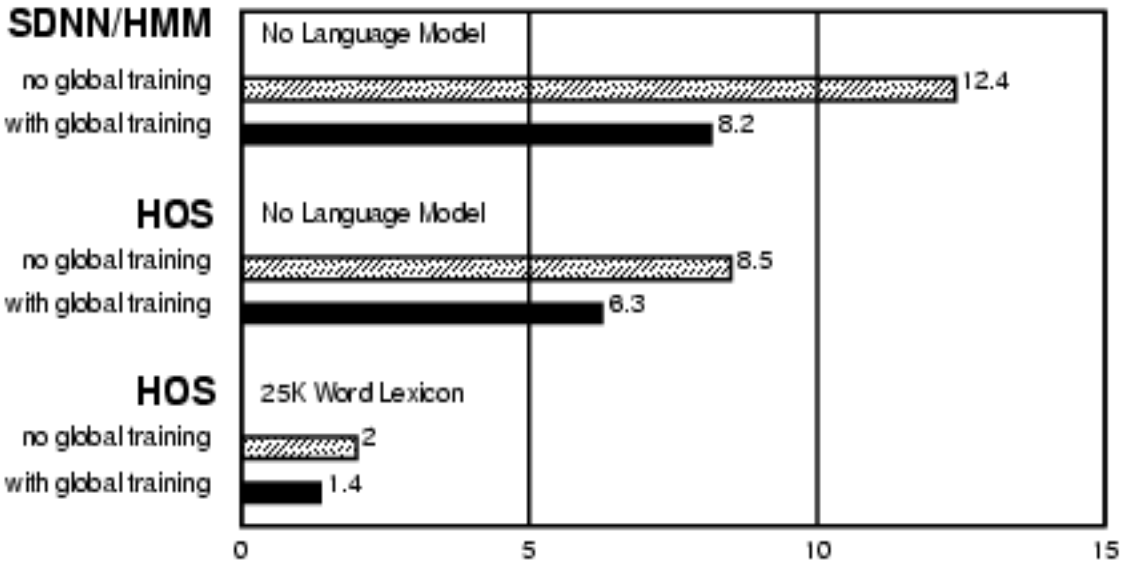
- Structured prediction: when the output is structured: string, graph.....
- Integrating deep learning and structured prediction is an old idea
  - In fact, it predates structured prediction [LeCun, Bottou, Bengio, Haffner 1998]
- Globally-trained convolutional-net + graphical models for handwriting recognition
  - 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

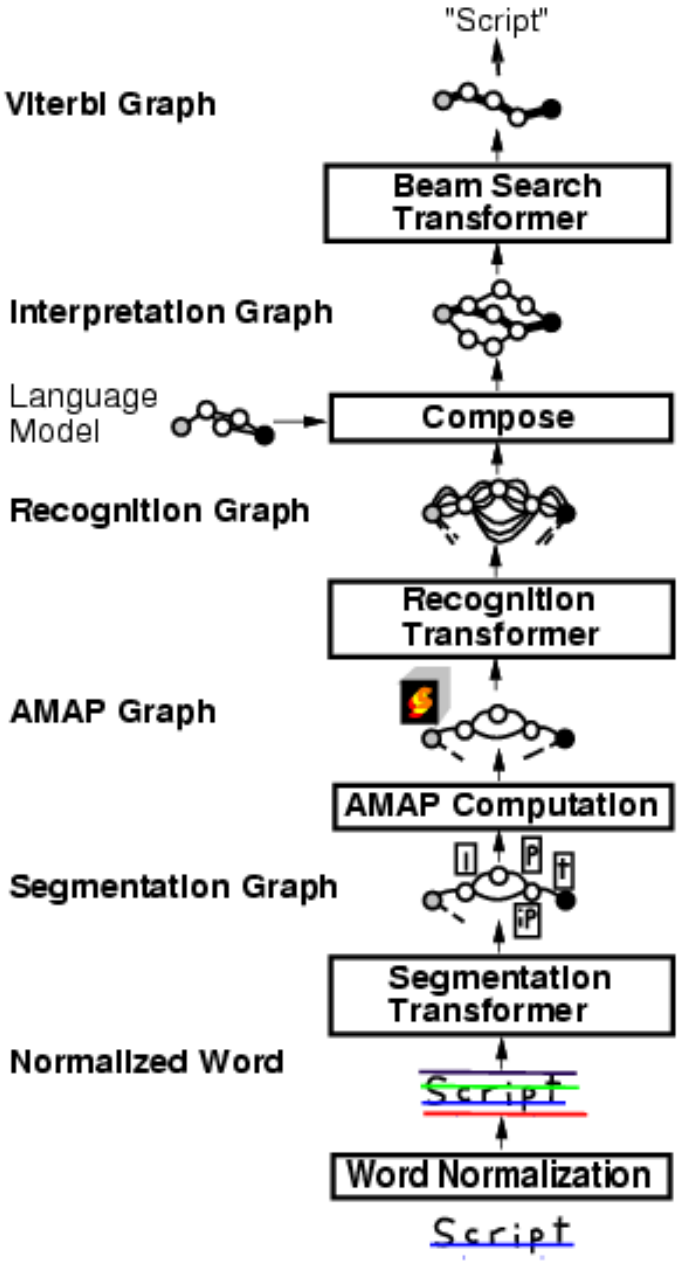


Global (word-level) Training Helps

Pen-based handwriting recognition (for tablet computer)

- [Bengio&LeCun 1995]

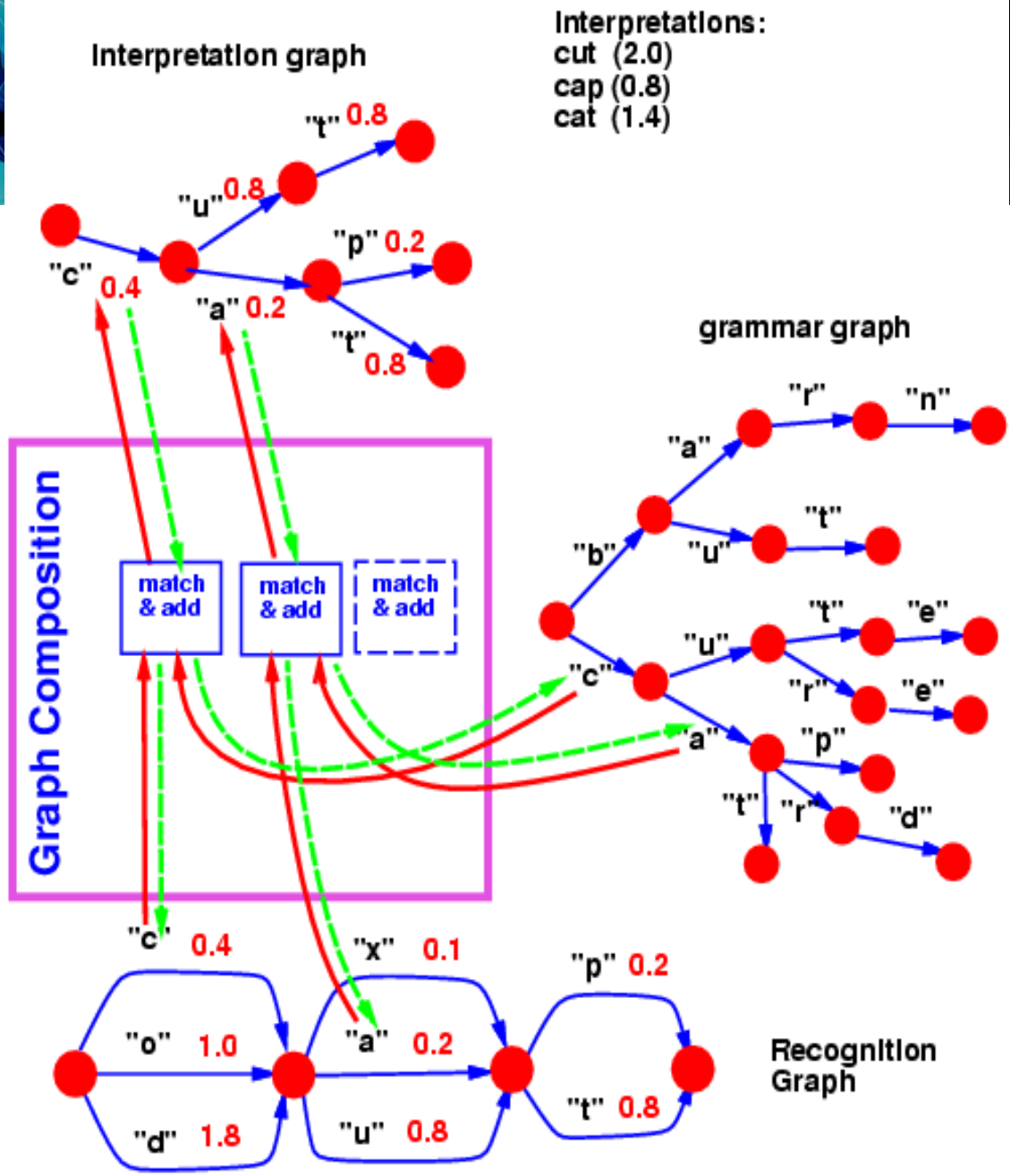




Graph Composition, Transducers.

The composition of two graphs can be computed, the same way the dot product between two vectors can be computed.

General theory: semi-ring algebra on weighted finite-state transducers and acceptors.



Check Reader

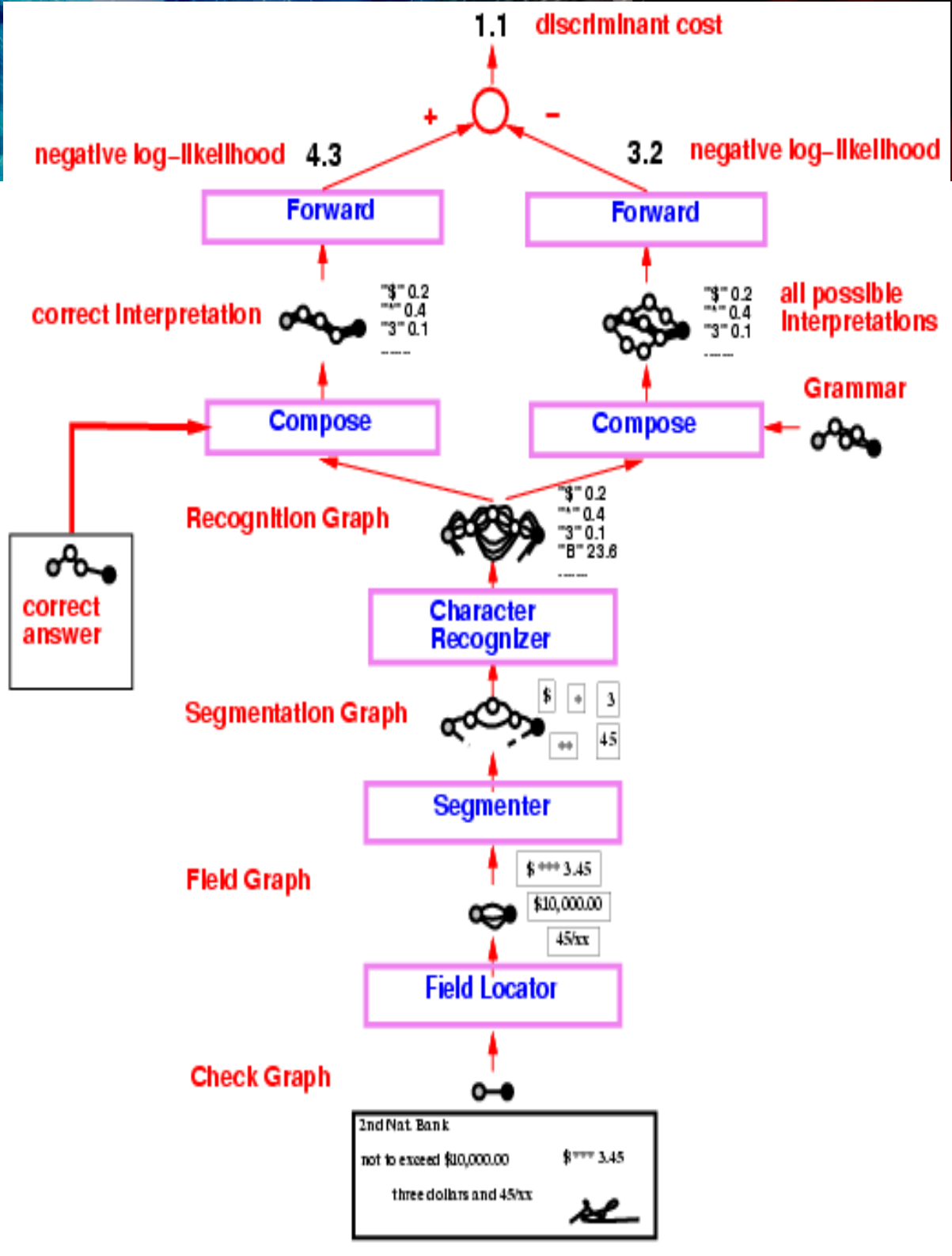
Graph transformer network trained to read check amounts.

Trained globally with Negative-Log-Likelihood loss.

50% percent corrent, 49% reject, 1% error (detectable later in the process.

Fielded in 1996, used in many banks in the US and Europe.

Processes an estimated 10% of all the checks written in the US.



Deep Factors / Deep Graph: ASR with TDNN/HMM

Discriminative Automatic Speech Recognition system with HMM and various acoustic models

- Training the acoustic model (feature extractor) and a (normalized) HMM in an integrated fashion.

With Minimum Empirical Error loss

- Ljolje and Rabiner (1990)

with NLL:

- Bengio (1992)
- Haffner (1993)
- Bourlard (1994)

With MCE

- Juang et al. (1997)

Late normalization scheme (un-normalized HMM)

- Bottou pointed out the **label bias problem** (1991)
- Denker and Burges proposed a solution (1995)


Future
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....

Towards Practical AI: Challenges

- Applying deep learning to NLP (requires "structured prediction")
- Video analysis/understanding (requires unsupervised learning)
- High-performance/low power embedded systems for ConvNets (FPGA/ASIC?)
- Very-large-scale deep learning (distributed optimization)
- Integrating reasoning with DL ("energy-based models", recursive neural nets)

- Then we can have
  - Automatically-created high-performance data analytics systems
  - Vector-space embedding of everything (language, users,...)
  - Multimedia content understanding, search and indexing
  - Multilingual speech dialog systems
  - Driver-less cars
  - Autonomous maintenance robots / personal care robots