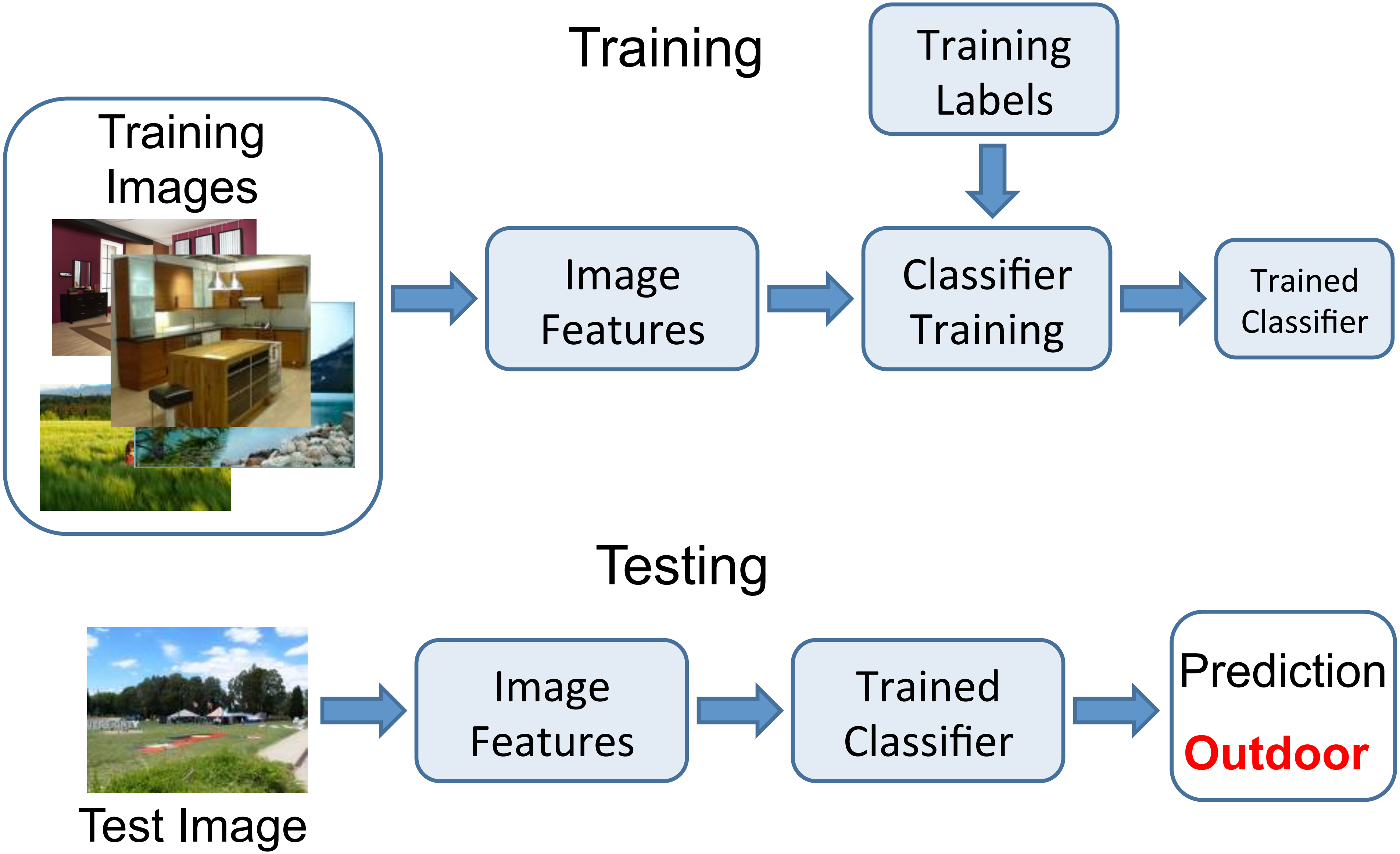
Object Detection II

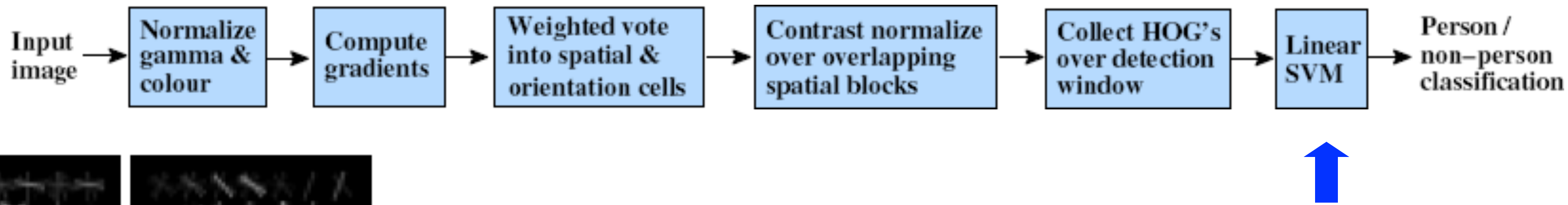
Ali Farhadi
CSE 576

Some slides from Derek Hoiem, Larry Zitnick, Ross Girchik

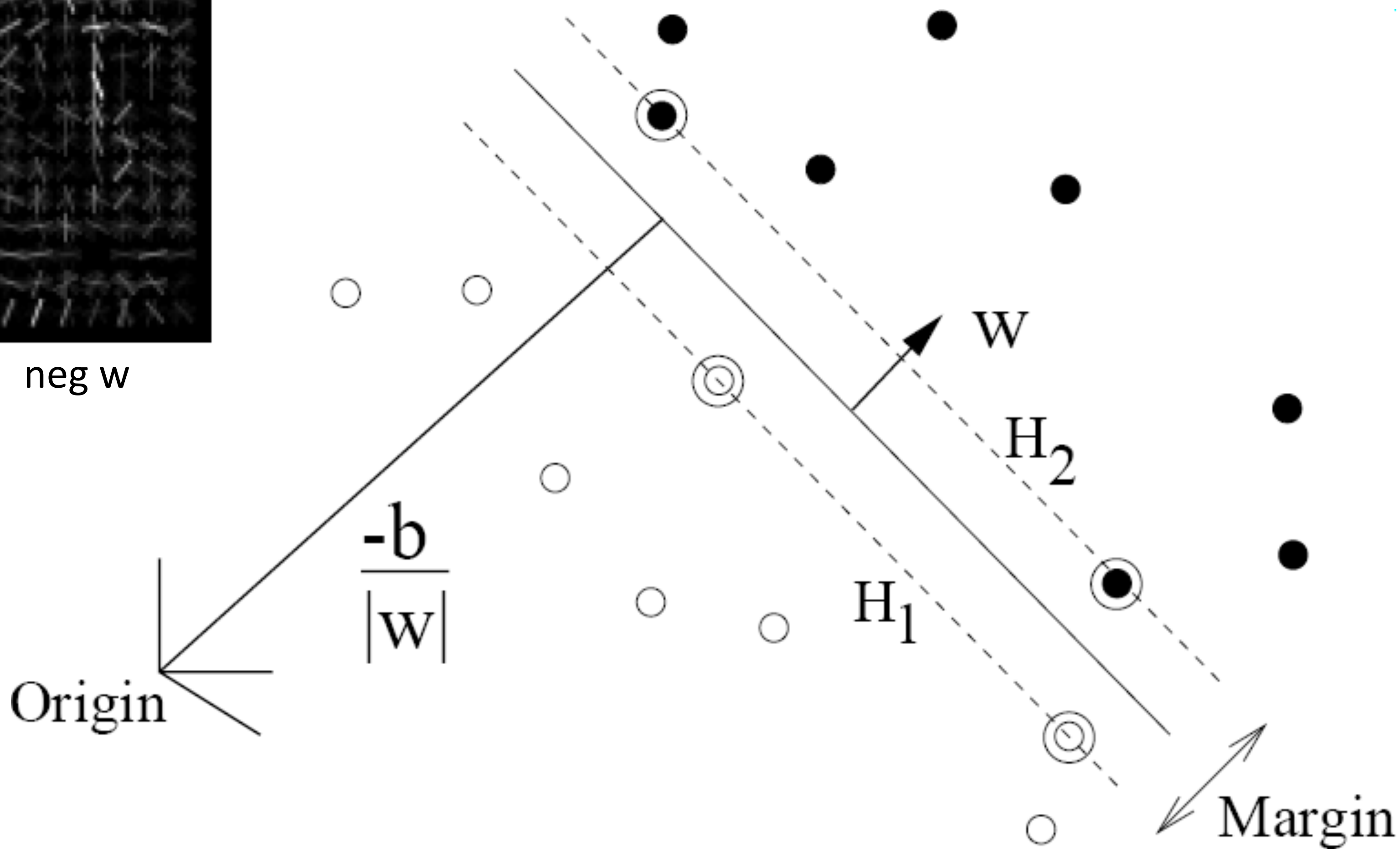
Image Categorization



Training set

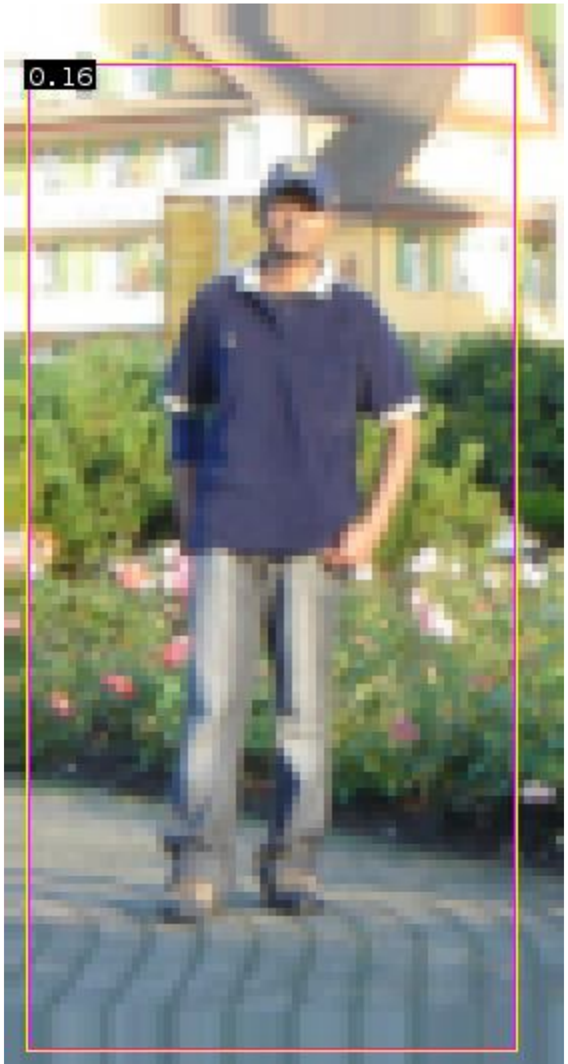


pos w neg w



Slides by Pete Barnum

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05



$$0.16 = w^T x - b$$

$$sign(0.16) = 1$$

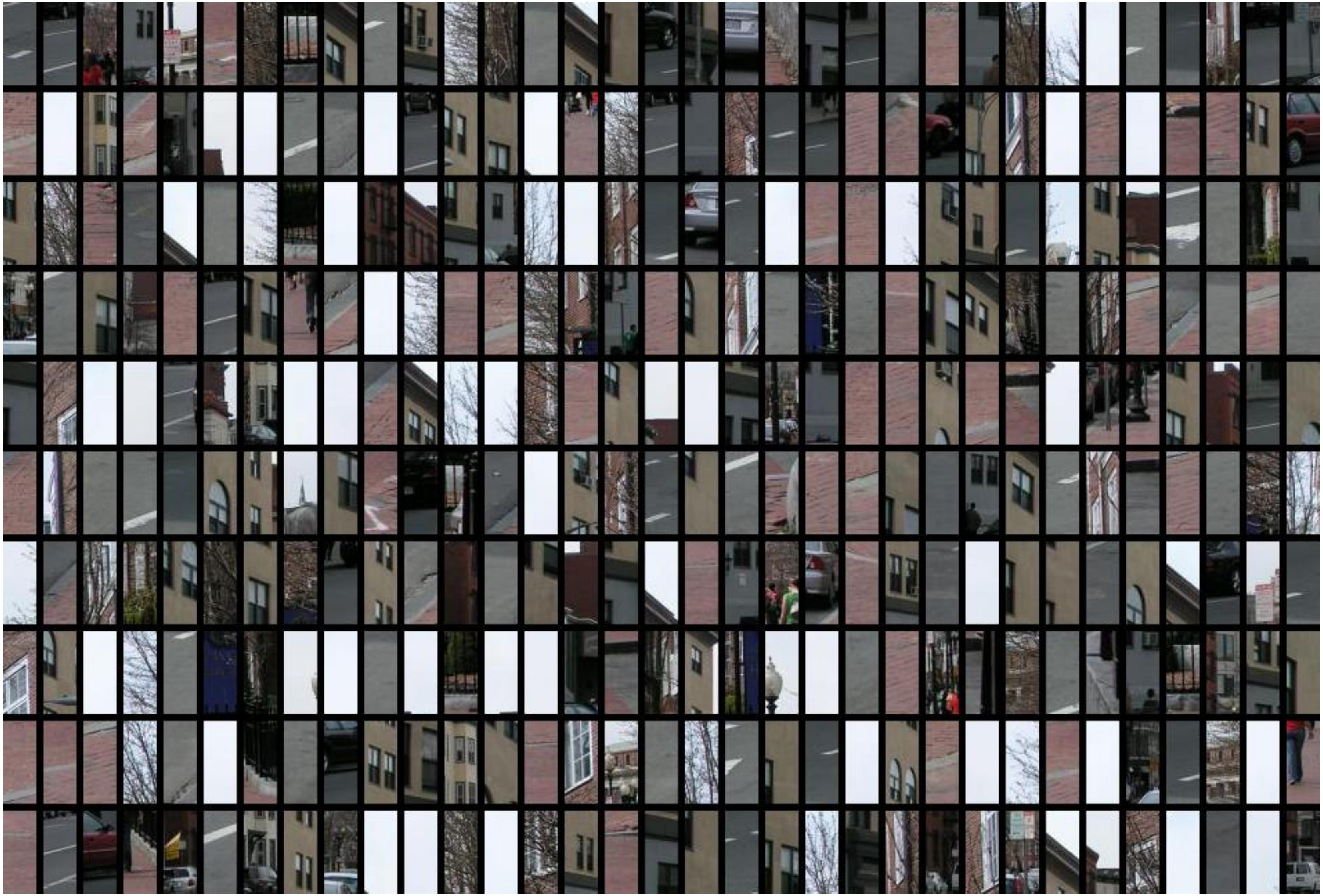
$=>$ pedestrian

Slides by Pete Barnum

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

Detection examples

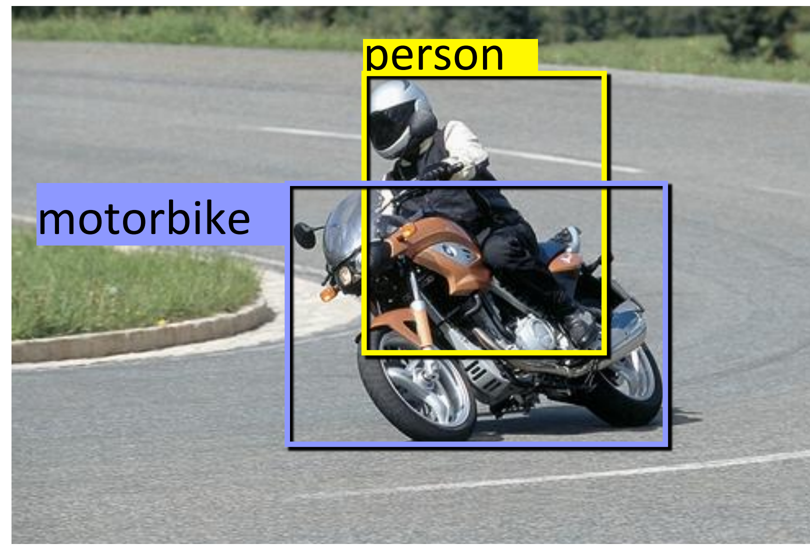
Each window is separately classified



Problem formulation

{ airplane, bird, motorbike, person, sofa }

Input



Desired output

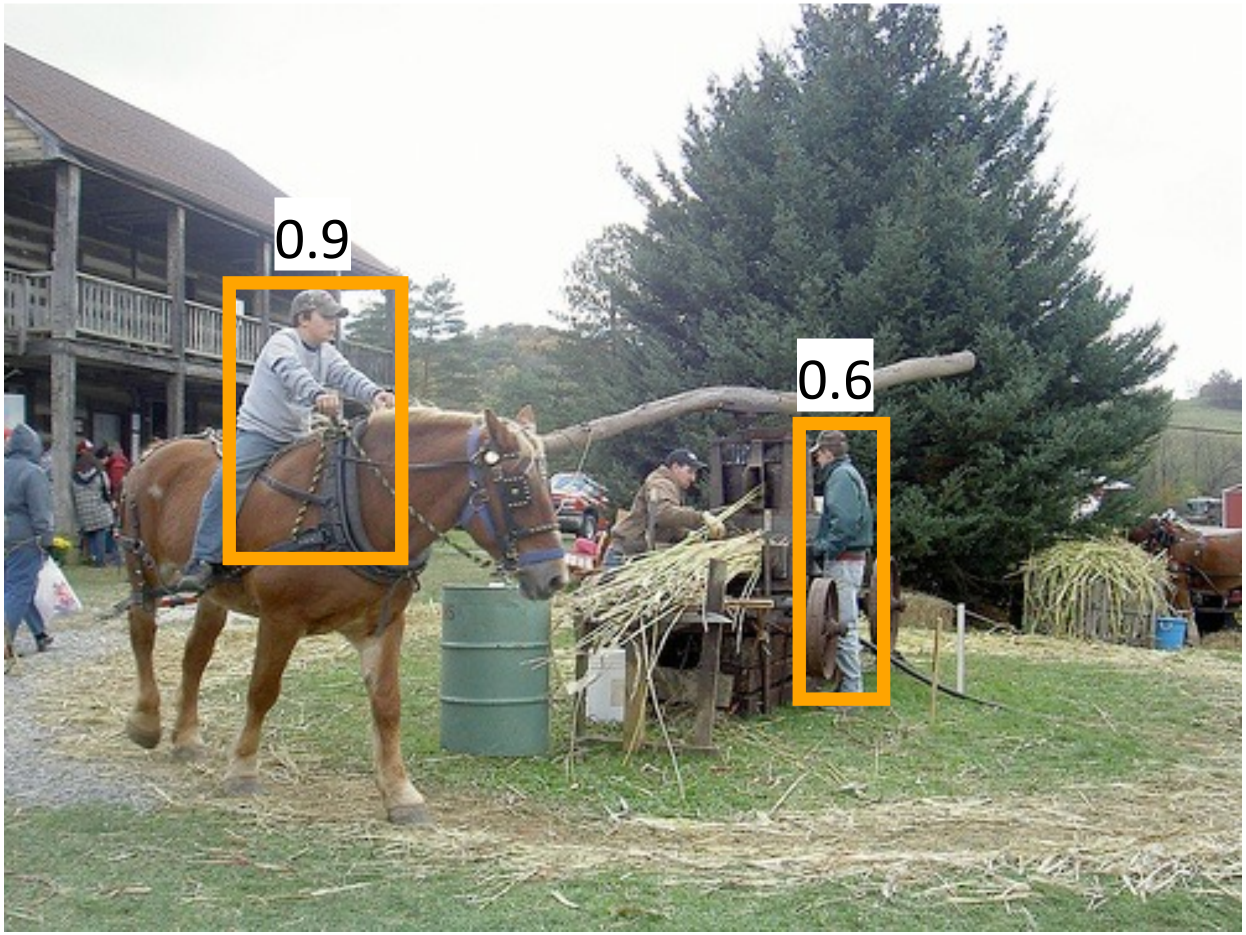
Evaluating a detector

Test image (previously unseen)

First detection ...

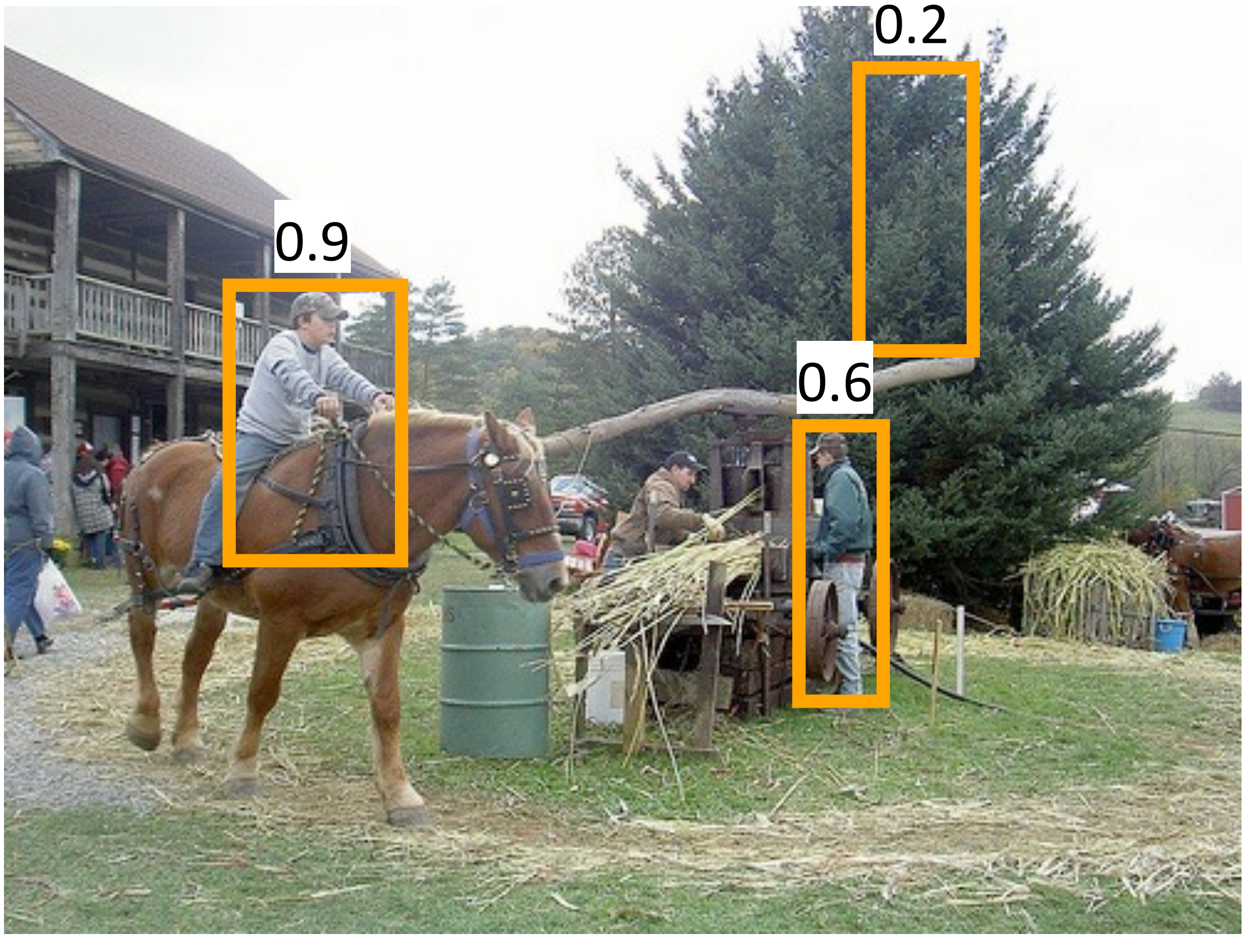
'person' detector predictions

Second detection ...



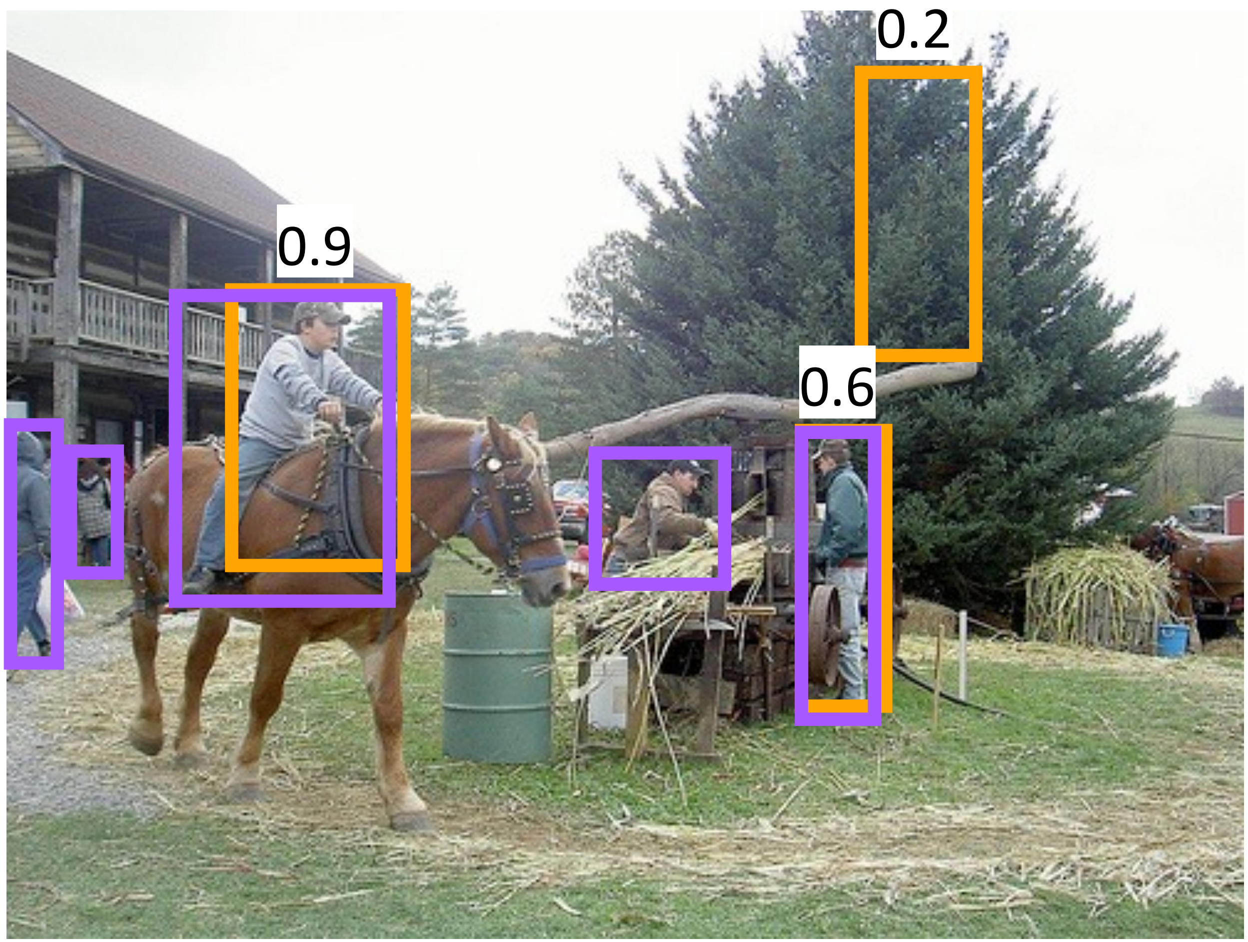
‘person’ detector predictions

Third detection ...



'person' detector predictions

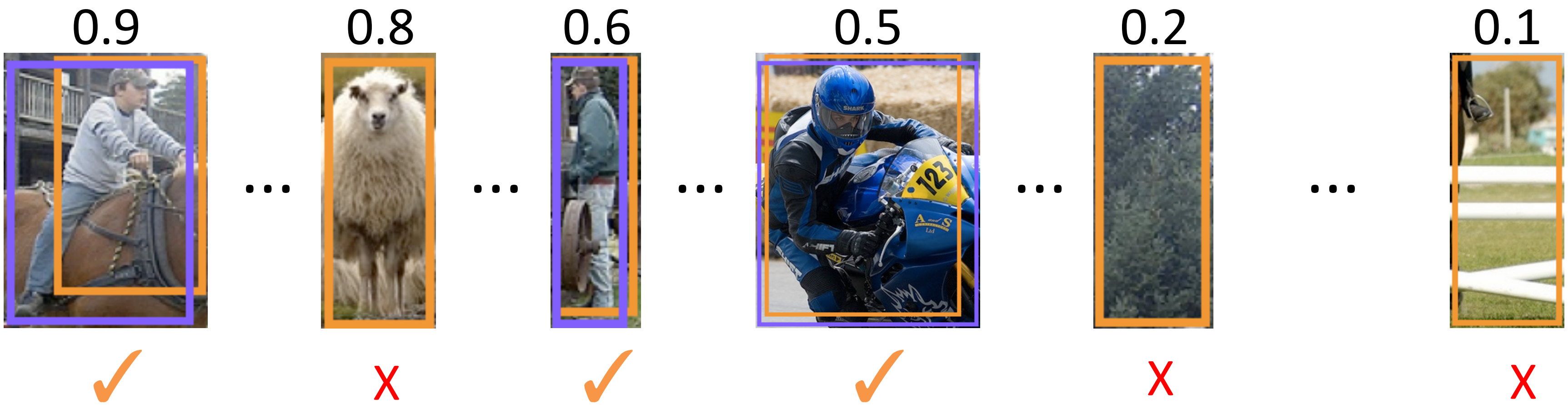
Compare to ground truth



'person' detector predictions

ground truth 'person' boxes

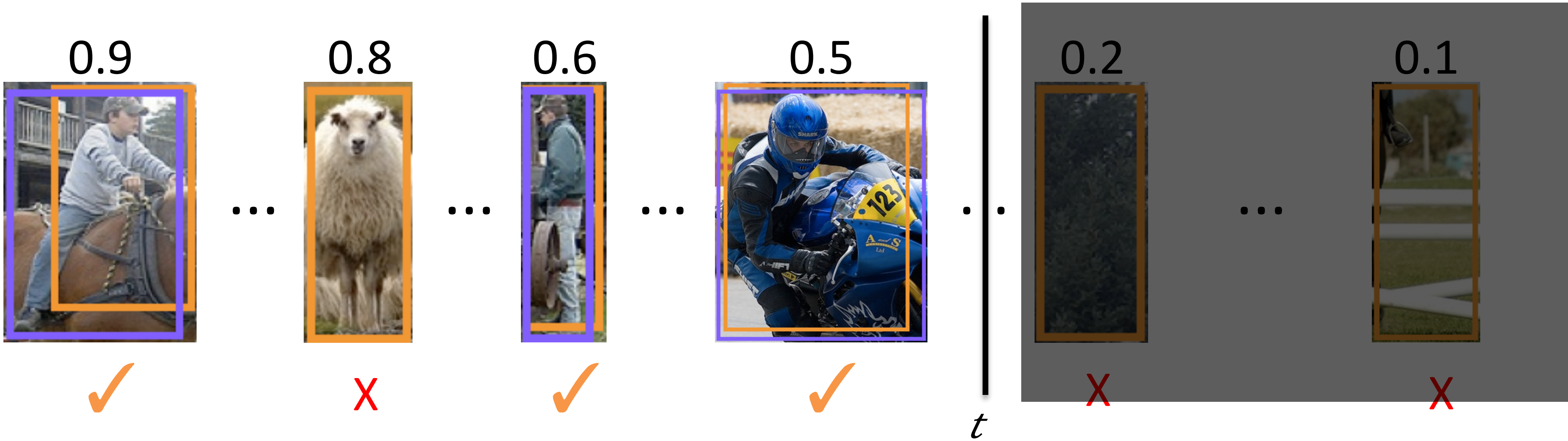
Sort by confidence



true positive (high overlap)

false positive (no overlap, low overlap, or duplicate)

Evaluation metric

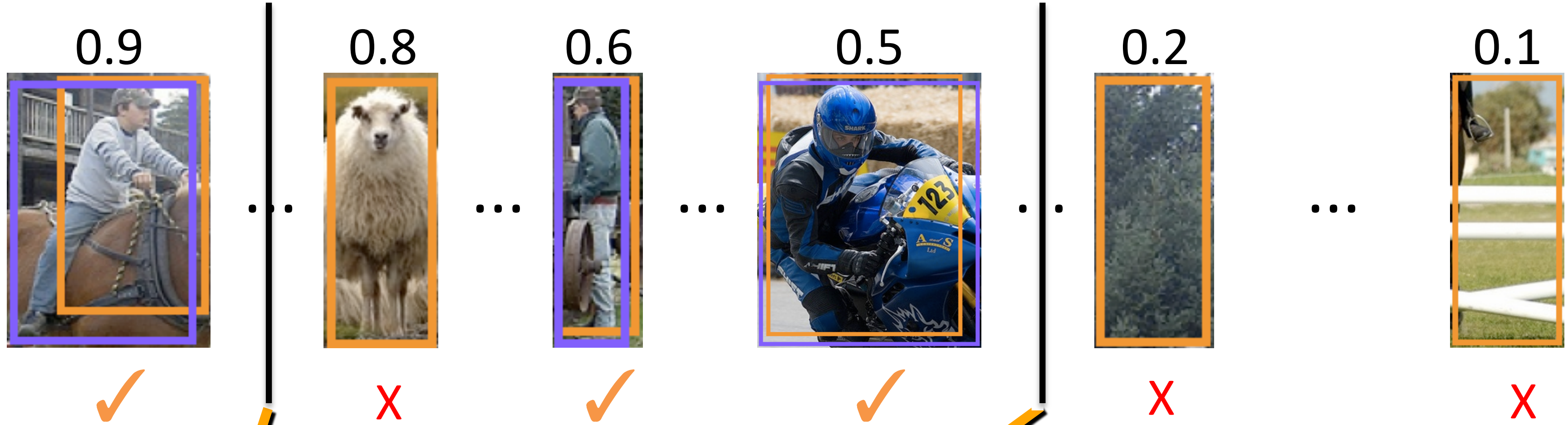


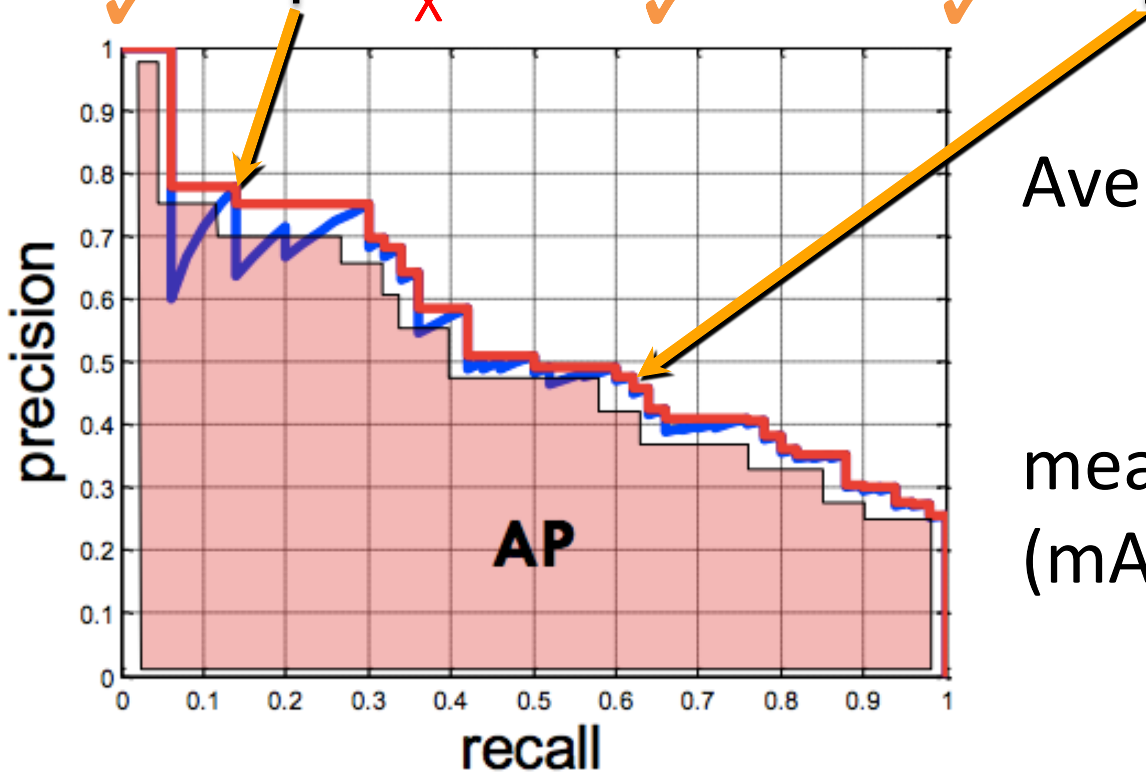
$$precision@t = \#true\ positives@t / \#true\ positives@t + \#false\ positives@t$$

$$\frac{✓}{✓ + X}$$

$$recall@t = \#true\ positives@t / \#ground\ truth\ objects$$

Evaluation metric





Average Precision (AP)

mean AP over classes (mAP)

What about this one?

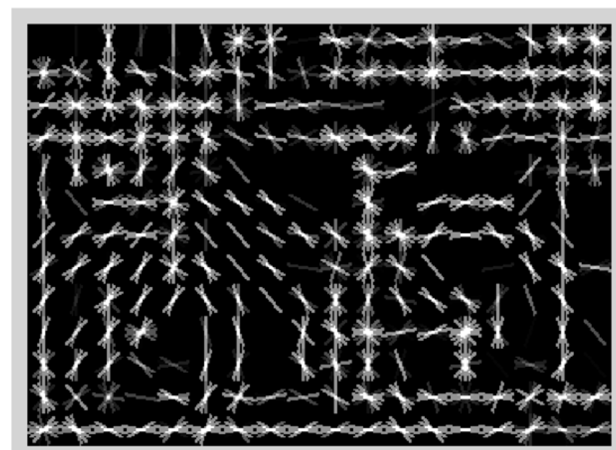
Can the model we trained for pedestrians detect the person in this image?

Specifying an object model

Statistical Template in Bounding Box

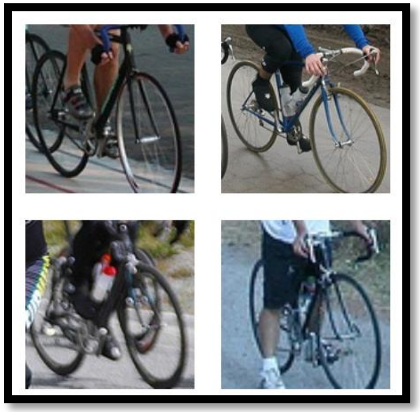
- Object is some (x,y,w,h) in image
- Features defined wrt bounding box coordinates

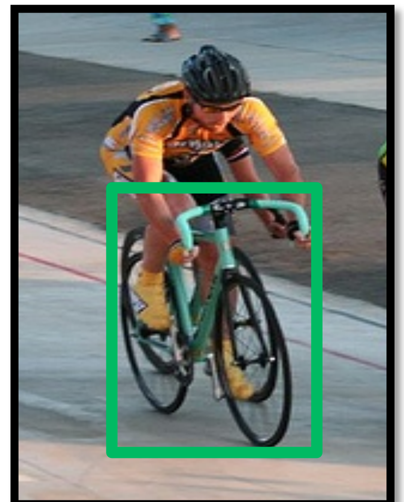
Image

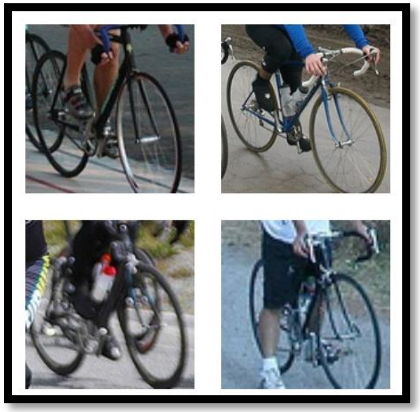
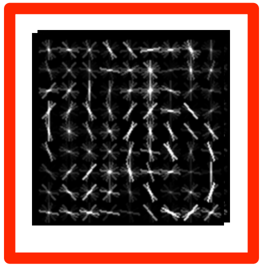
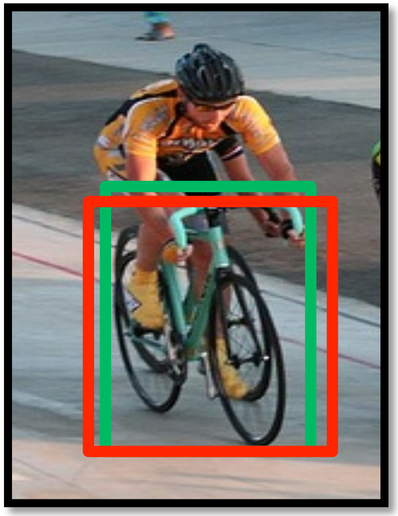


Template Visualization

Images from Felzenszwalb

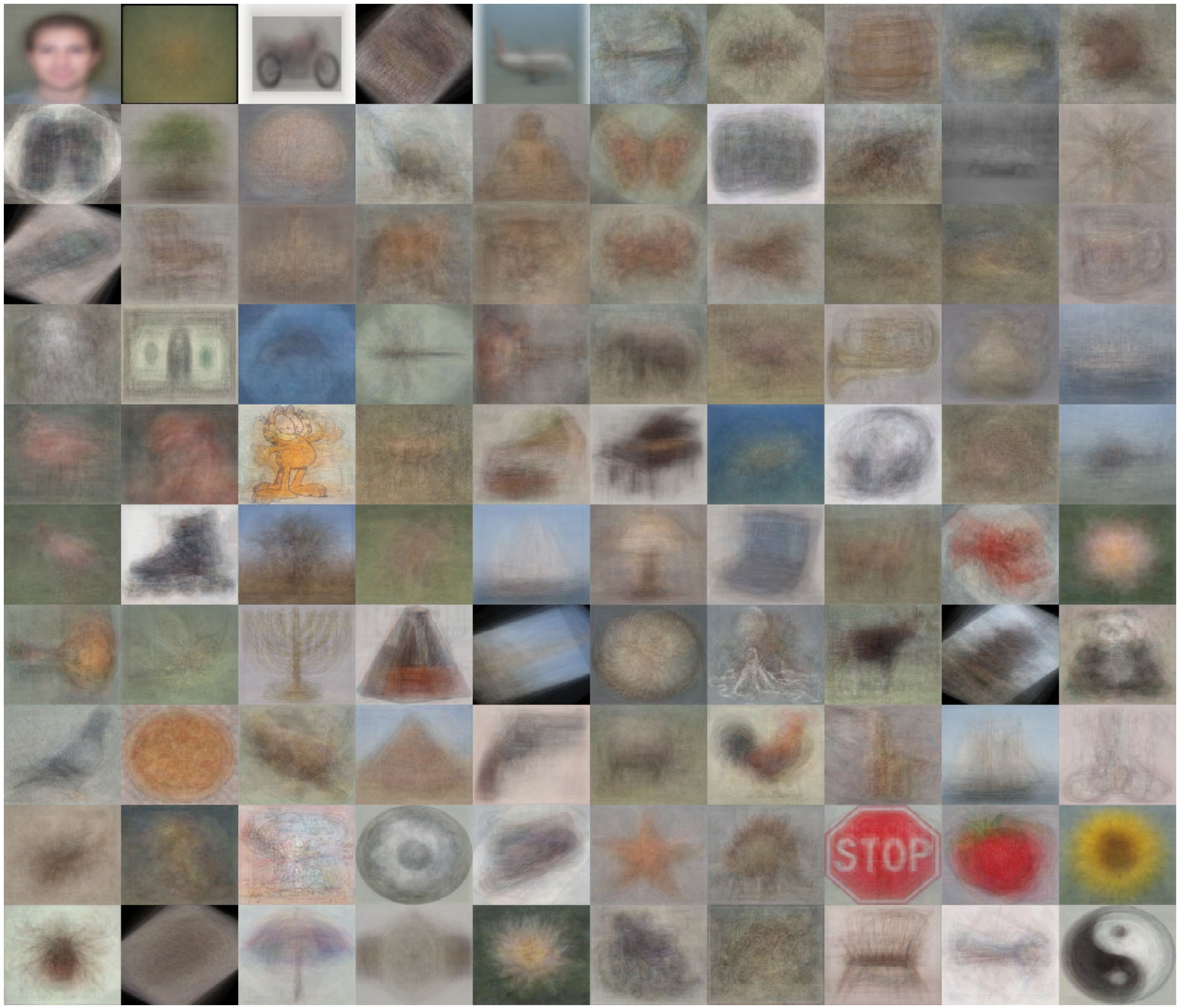




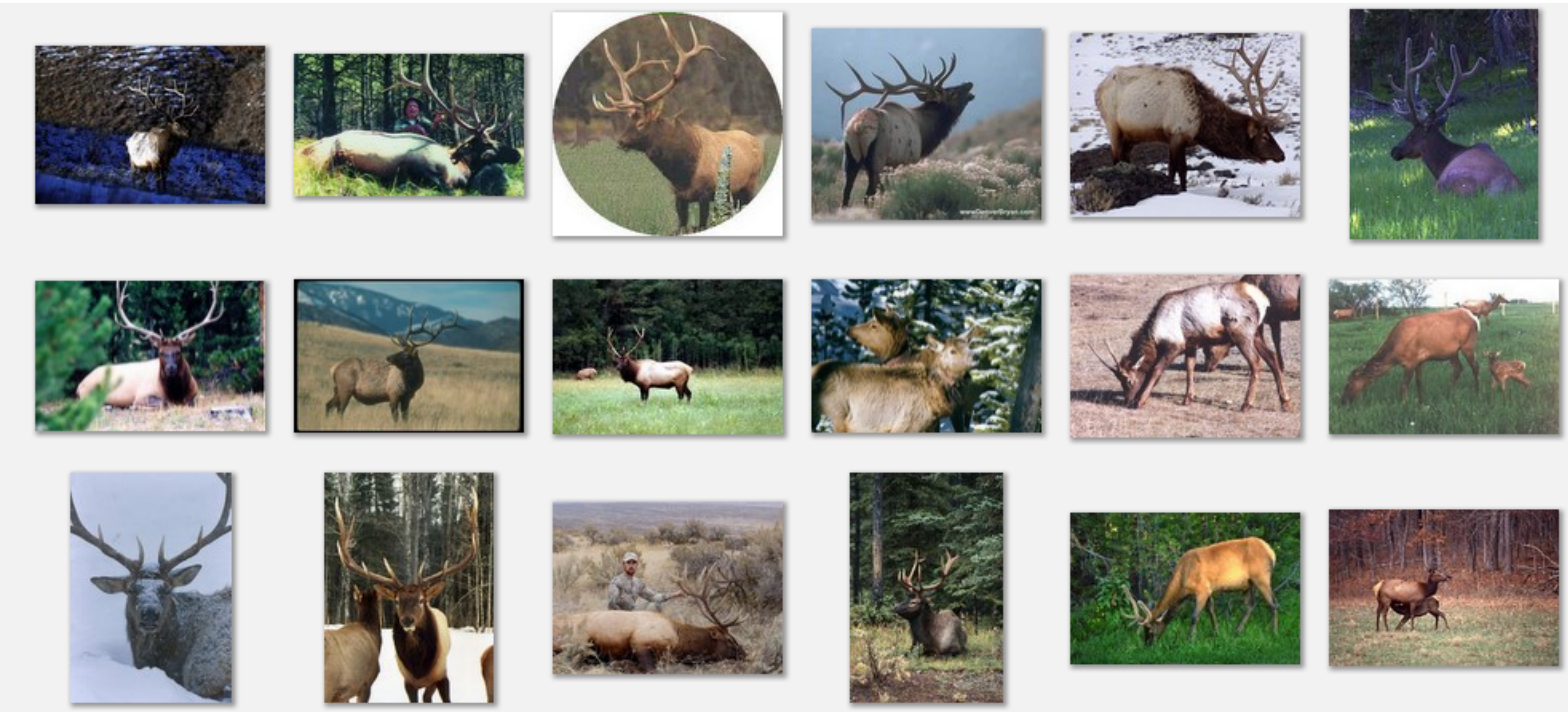

Octane

When do statistical templates make sense?



Caltech 101 Average Object Images

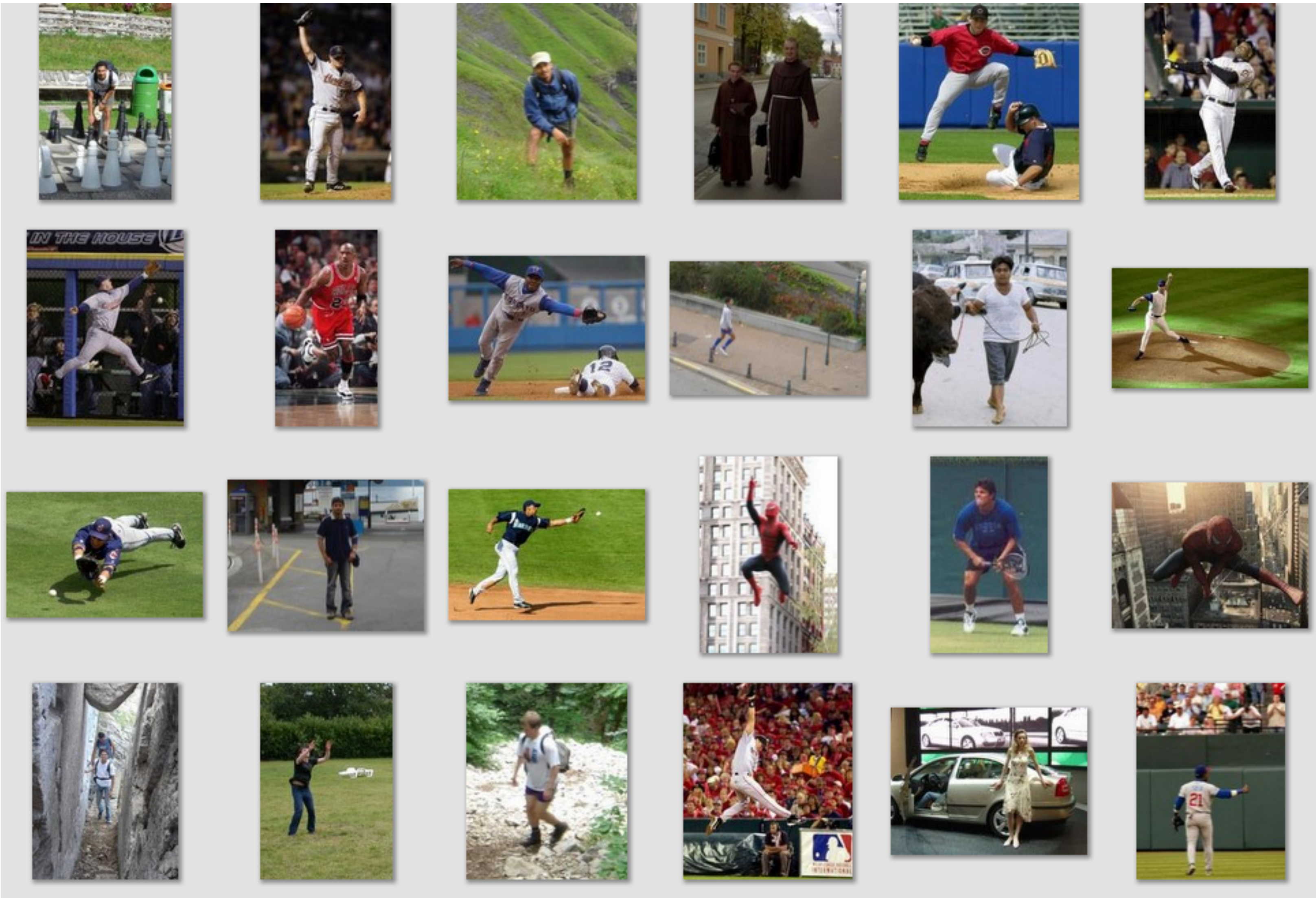
Deformable objects



Images from Caltech-256

Slide Credit: Duan Tran

Deformable objects



Images from D. Ramanan's dataset

Slide Credit: Duan Tran

Parts-based Models

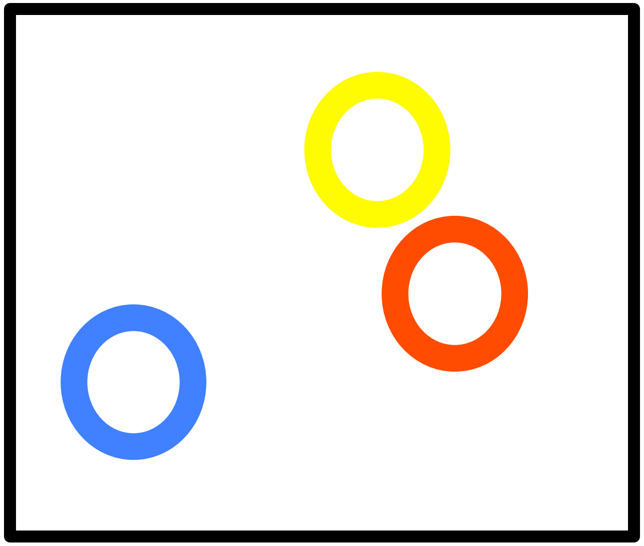
Define objects by collection of parts modeled by

1. Appearance
2. Spatial configuration

Slide credit: Rob Fergus

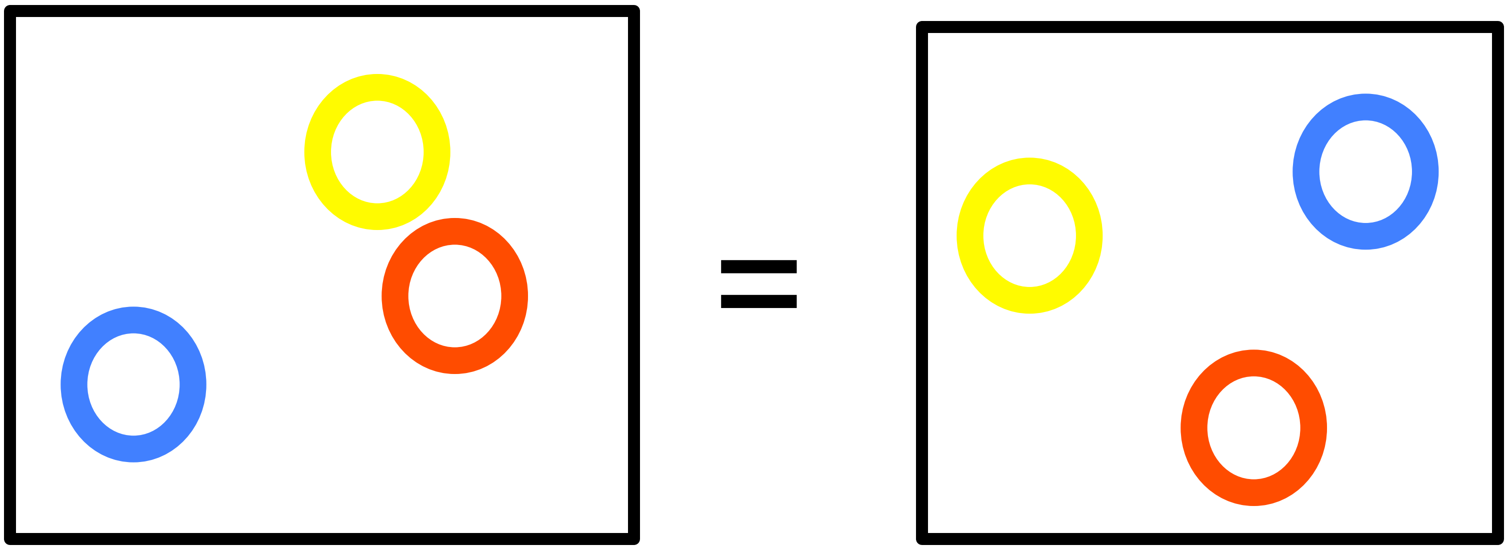
How to model spatial relations?

- One extreme: fixed template



How to model spatial relations?

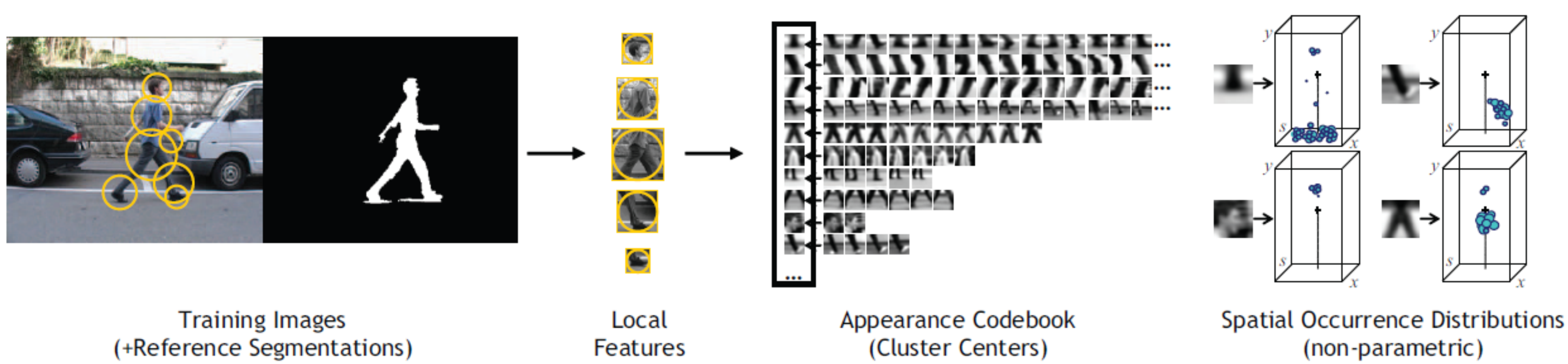
- Another extreme: bag of words



ISM: Implicit Shape Model

Training overview

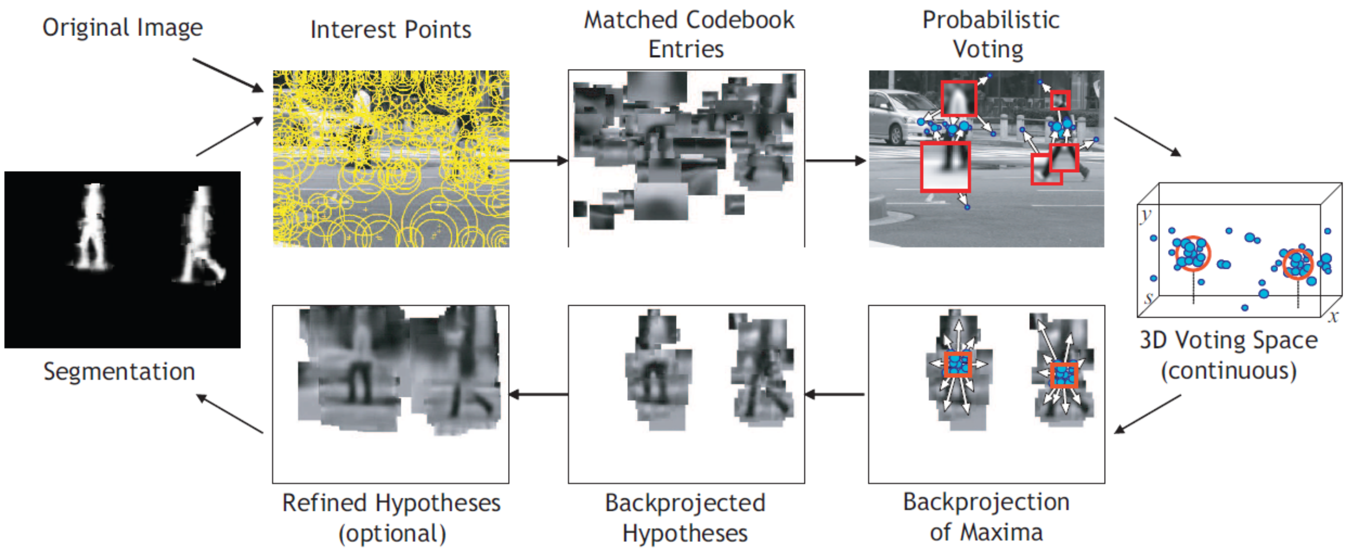
- Start with bounding boxes and (ideally) segmentations of objects
- Extract local features (e.g., patches or SIFT) at interest points on objects
- Cluster features to create codebook
- Record relative bounding box and segmentation for each codeword



ISM: Implicit Shape Model

Testing overview

- Extract interest points in test image
- Softly match to codebook entries
- Each matched codeword votes for object bounding box
- Compute modes of votes using mean-shift
- Check which codewords voted for modes
- Refine

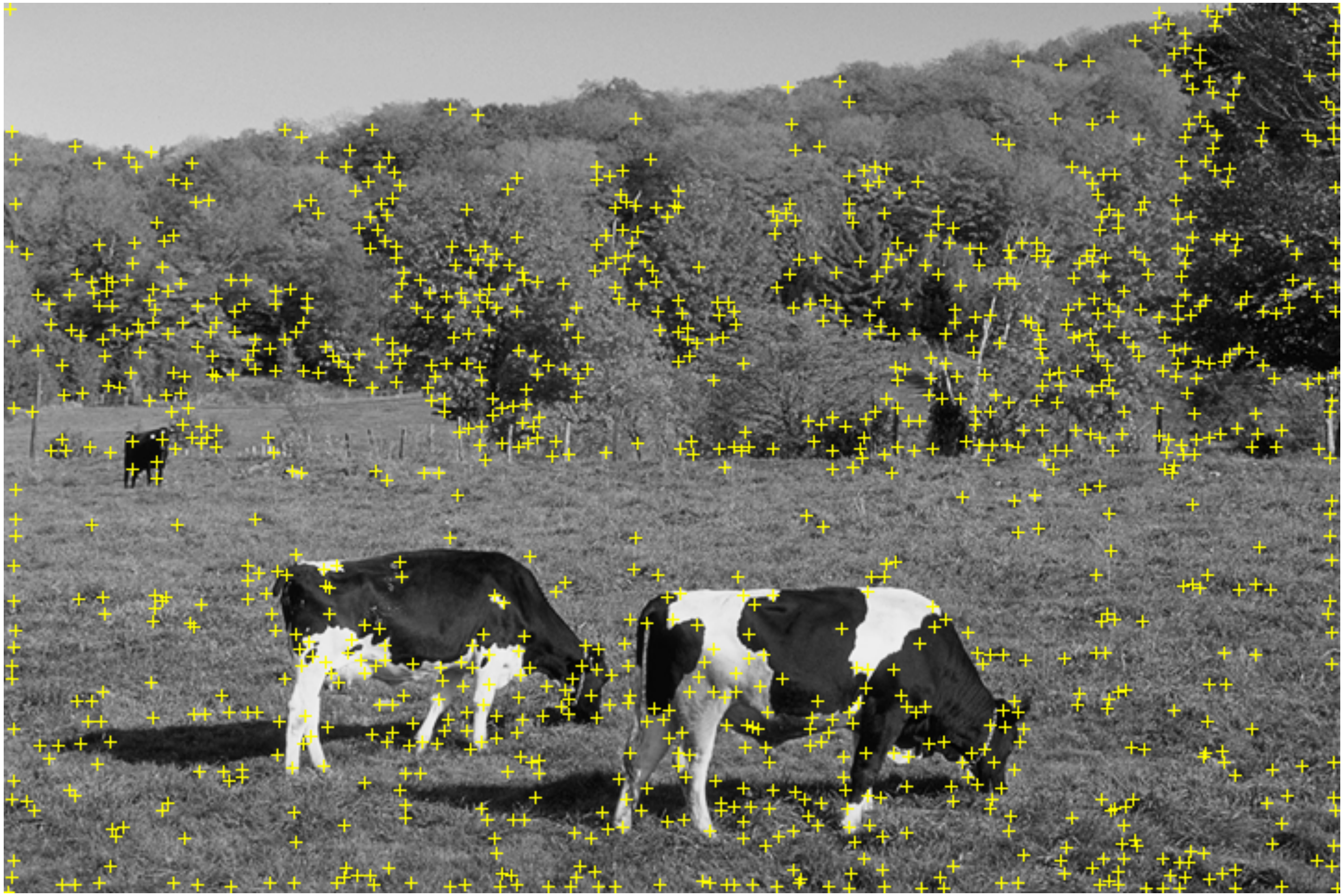


Example: Results on Cows

[redacted]

K. Grauman, B. Leibe

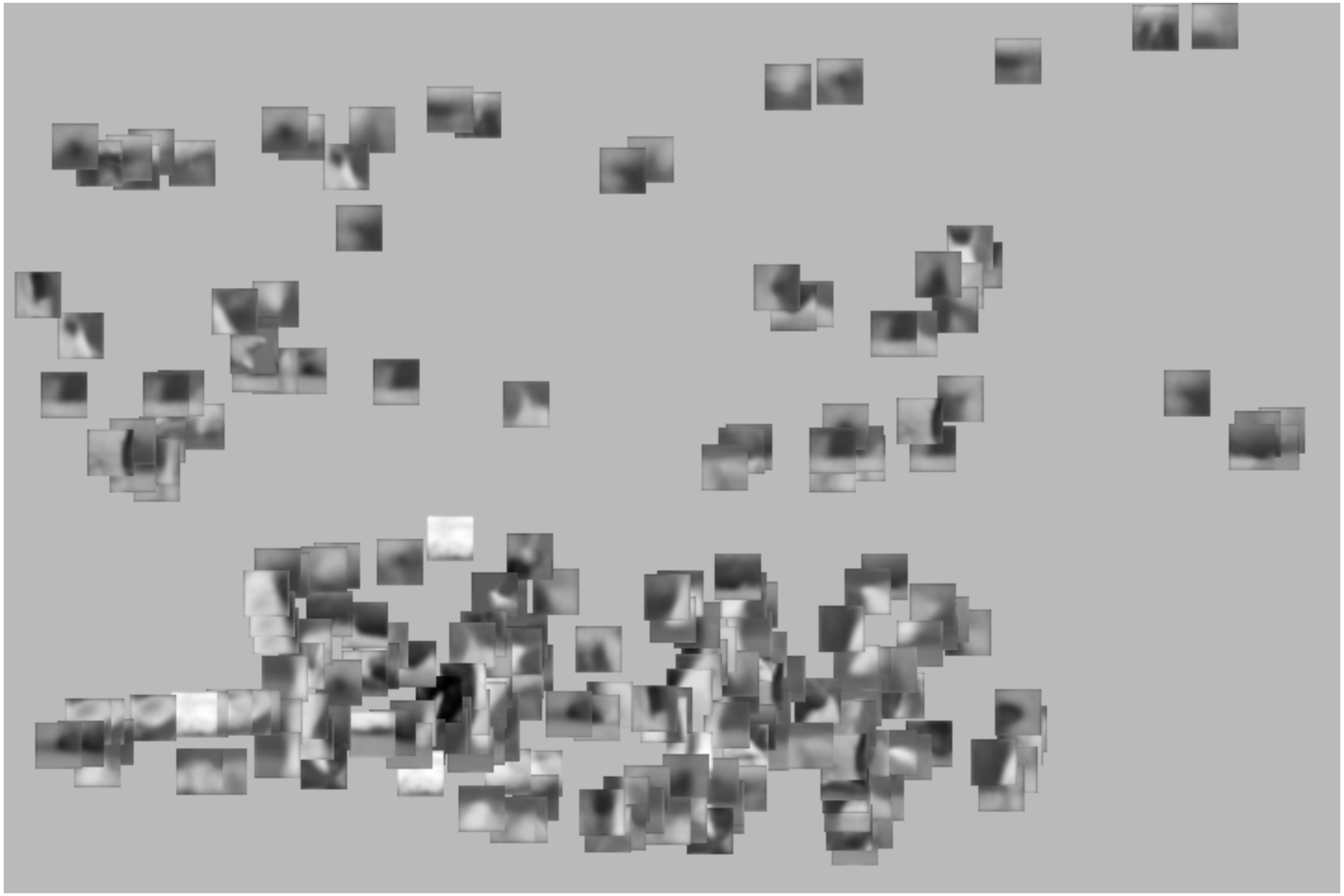
Example: Results on Cows



[redacted]

K. Grauman, B. Leibe

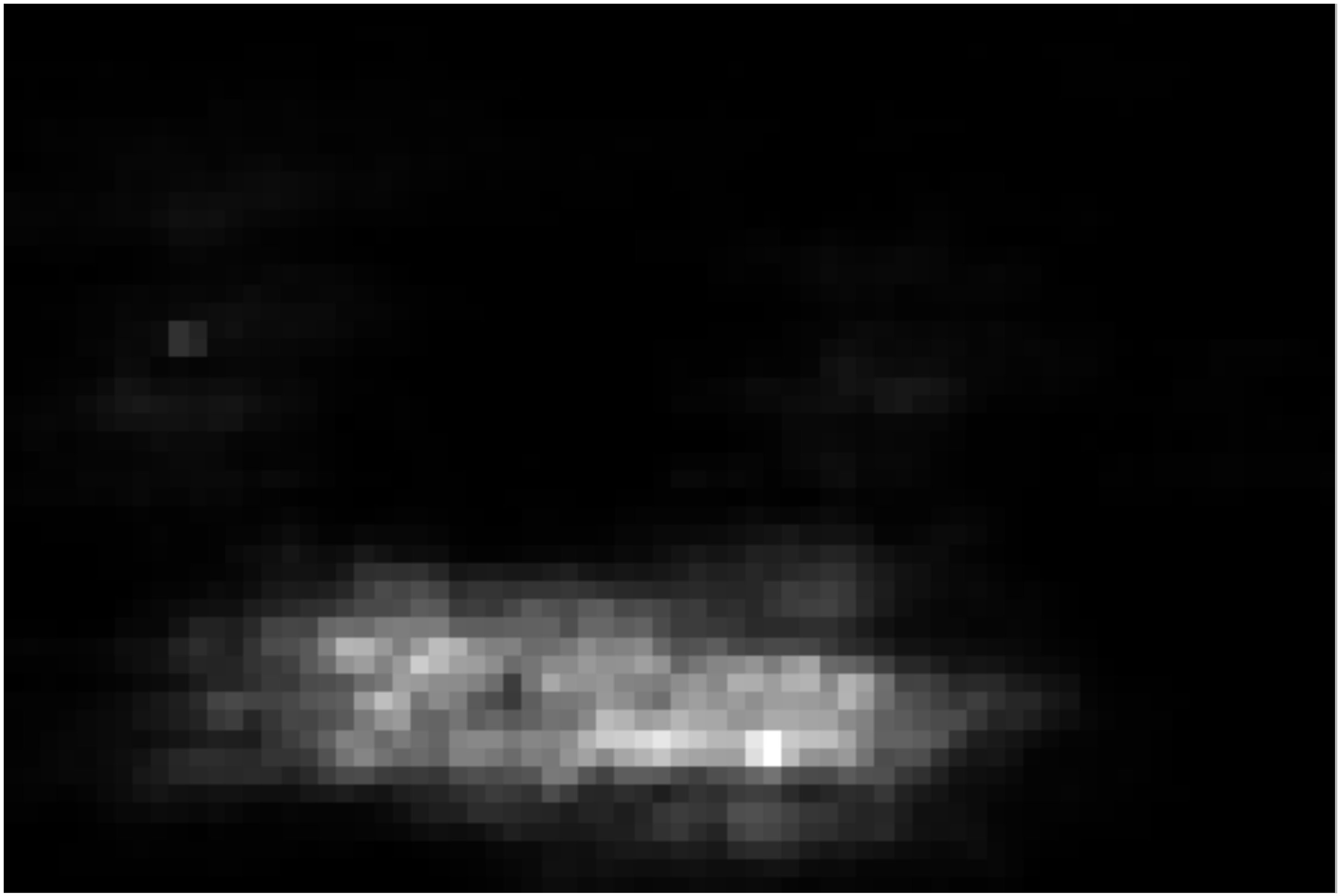
Example: Results on Cows



[redacted]

K. Grauman, B. Leibe

Example: Results on Cows



[redacted]

K. Grauman, B. Leibe

Example: Results on Cows

[redacted]

K. Grauman, B. Leibe

Example: Results on Cows

[redacted]

K. Grauman, B. Leibe

Example: Results on Cows

[redacted]

K. Grauman, B. Leibe

ISM: Detection Results

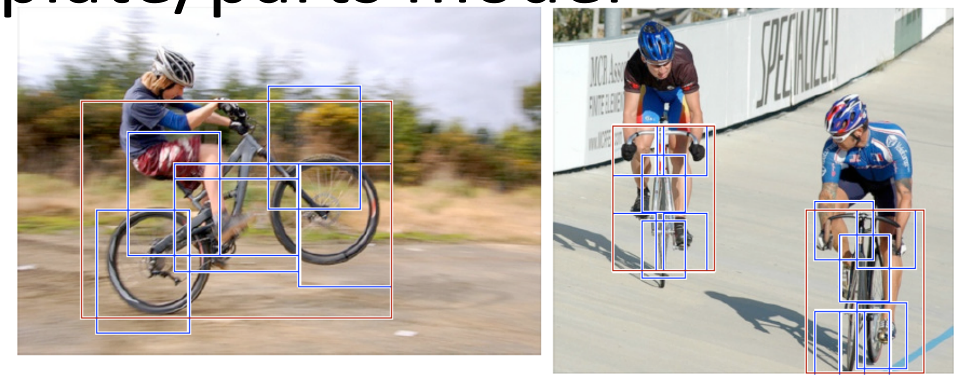
- Qualitative Performance
  – Robust to clutter, occlusion, noise, low contrast

K. Grauman, B. Leibe

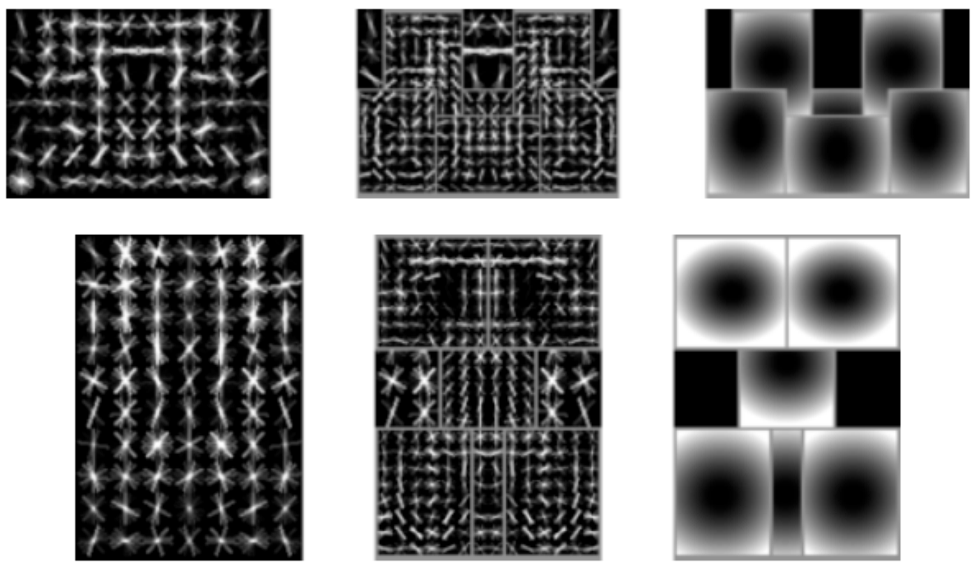
Explicit Models

Hybrid template/parts model

Detections



Template Visualization



root filters
coarse resolution

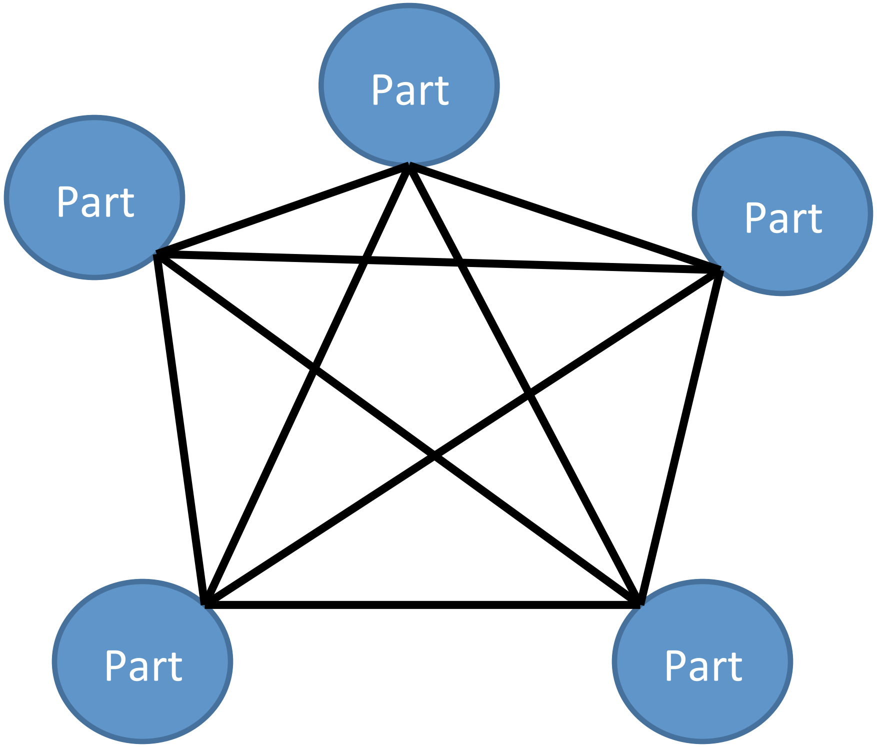
part filters
finer resolution

deformation
models

Felzenszwalb et al. 2008

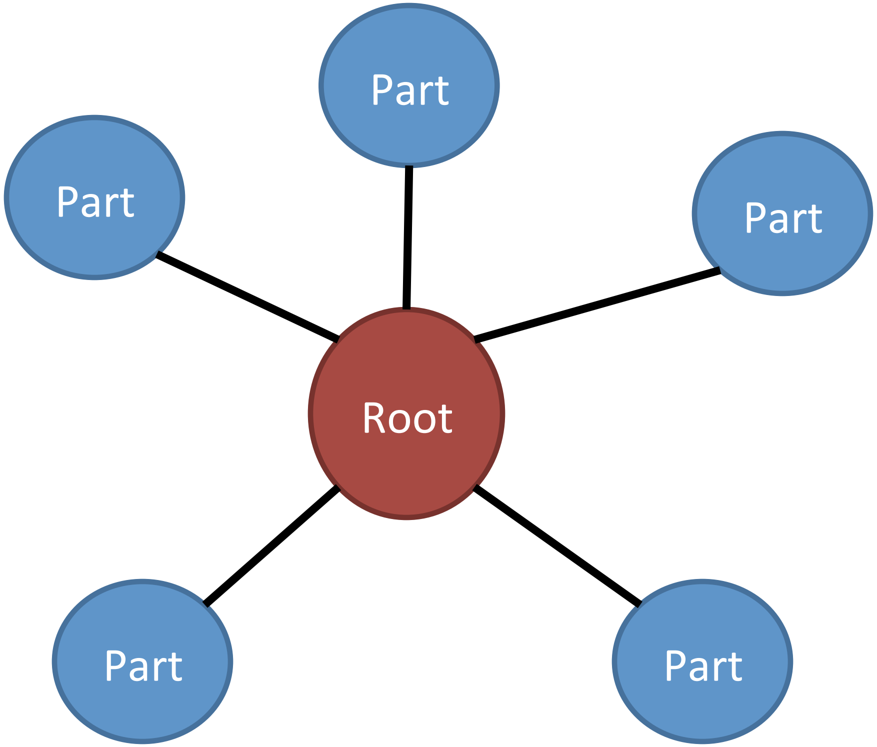
How to model spatial relations?

- Explicit Models
- Too expensive



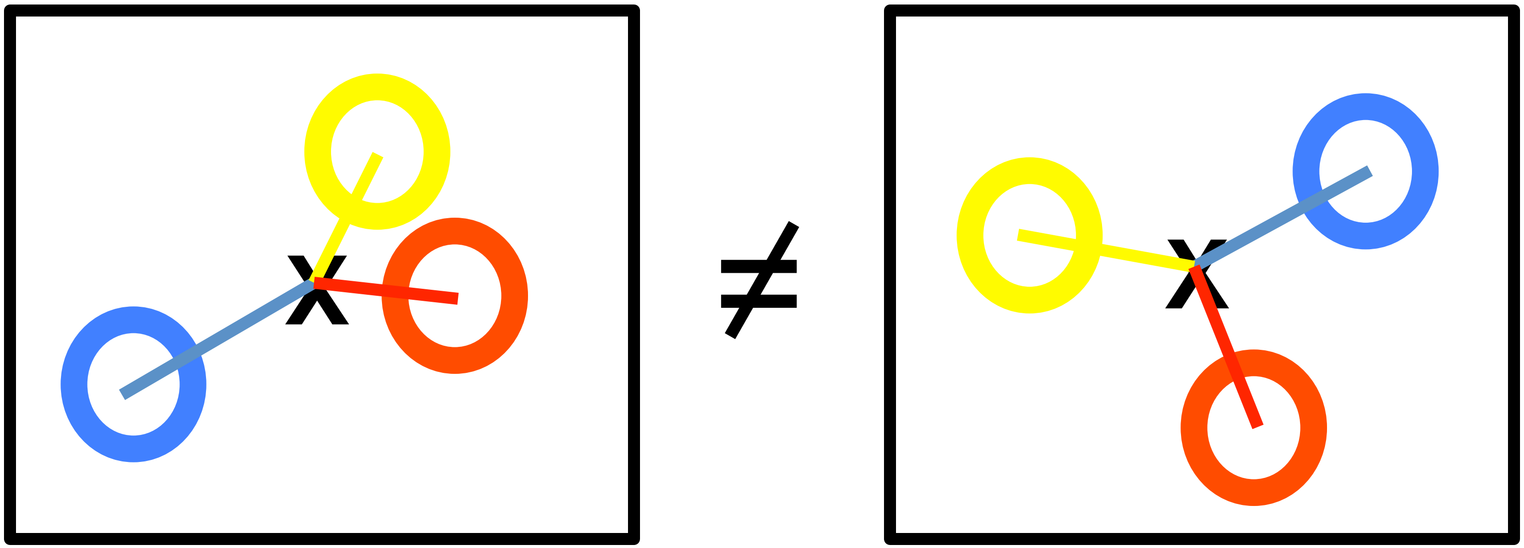
How to model spatial relations?

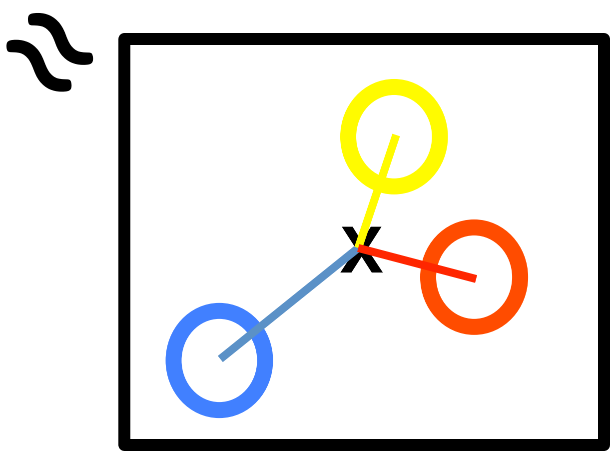
- Star-shaped model

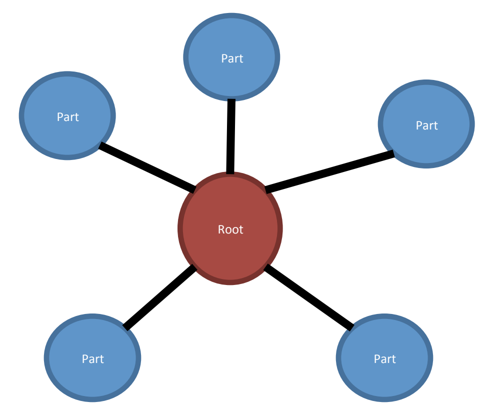


How to model spatial relations?

- Star-shaped model

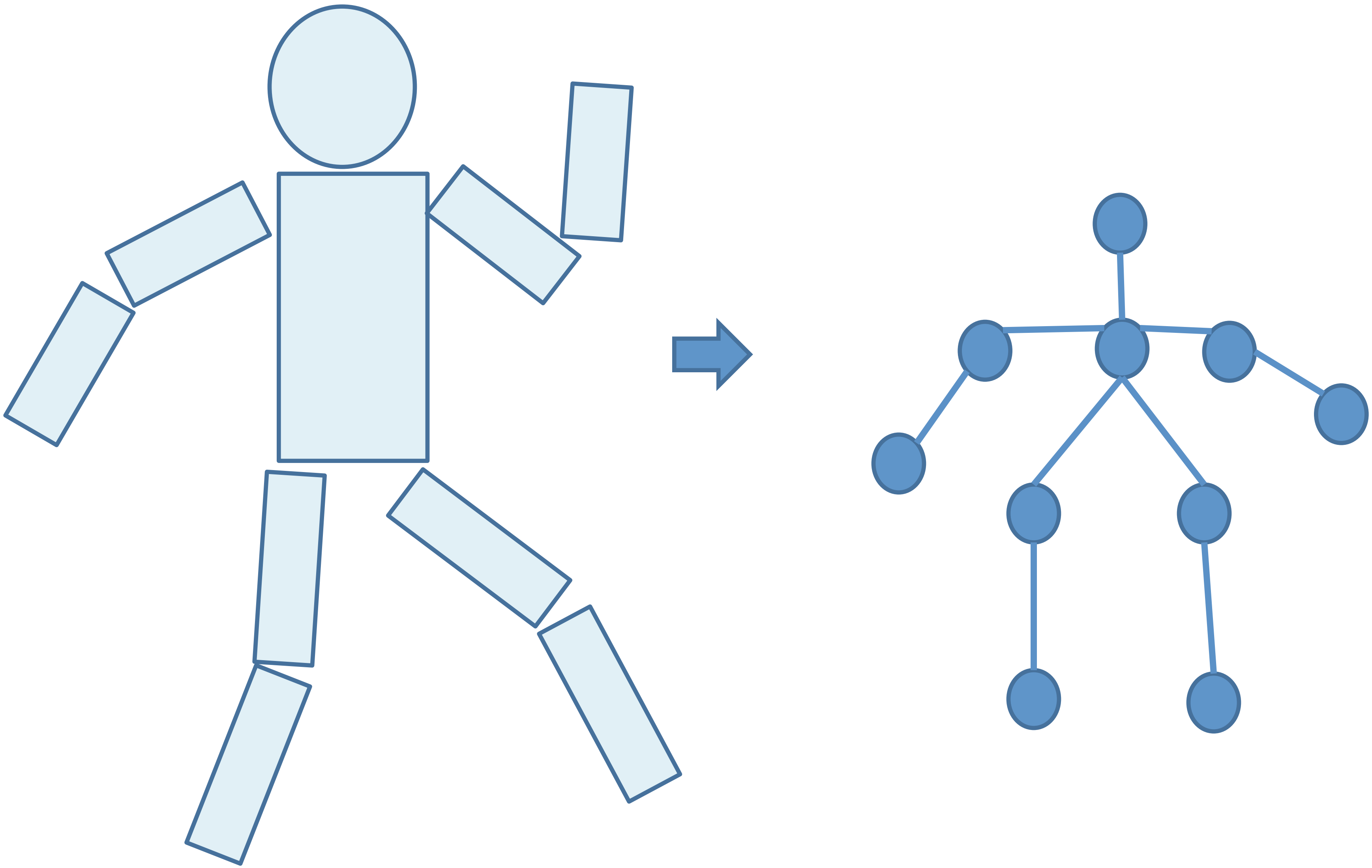






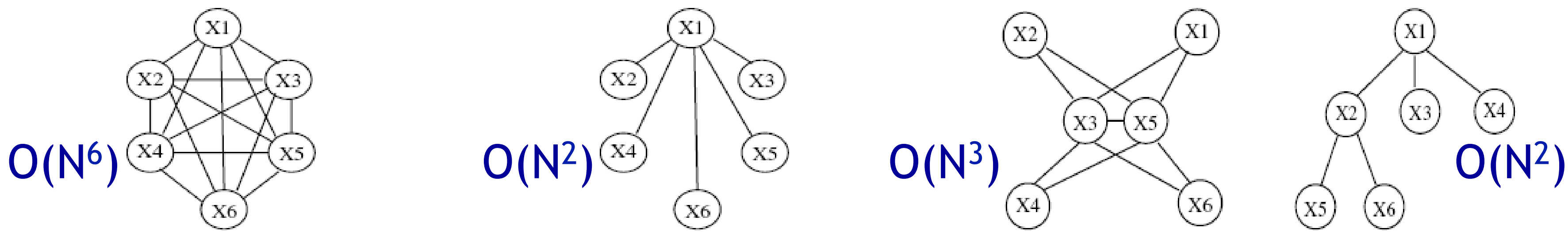
How to model spatial relations?

- Tree-shaped model



How to model spatial relations?

- Many others...



a) Constellation

b) Star shape

c) k-fan ($k = 2$)

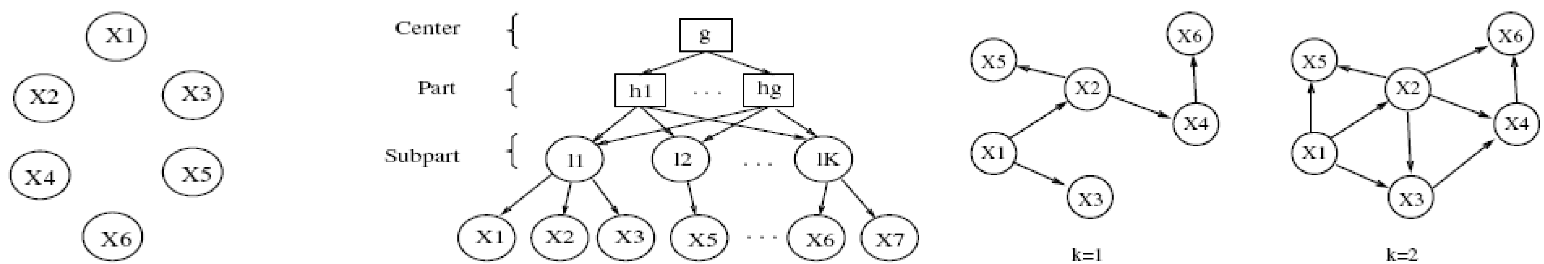
d) Tree

Fergus et al. '03
Fei-Fei et al. '03

Leibe et al. '04, '08
Crandall et al. '05
Fergus et al. '05

Crandall et al. '05

Felzenszwalb & Huttenlocher '05



e) Bag of features

f) Hierarchy

g) Sparse flexible model

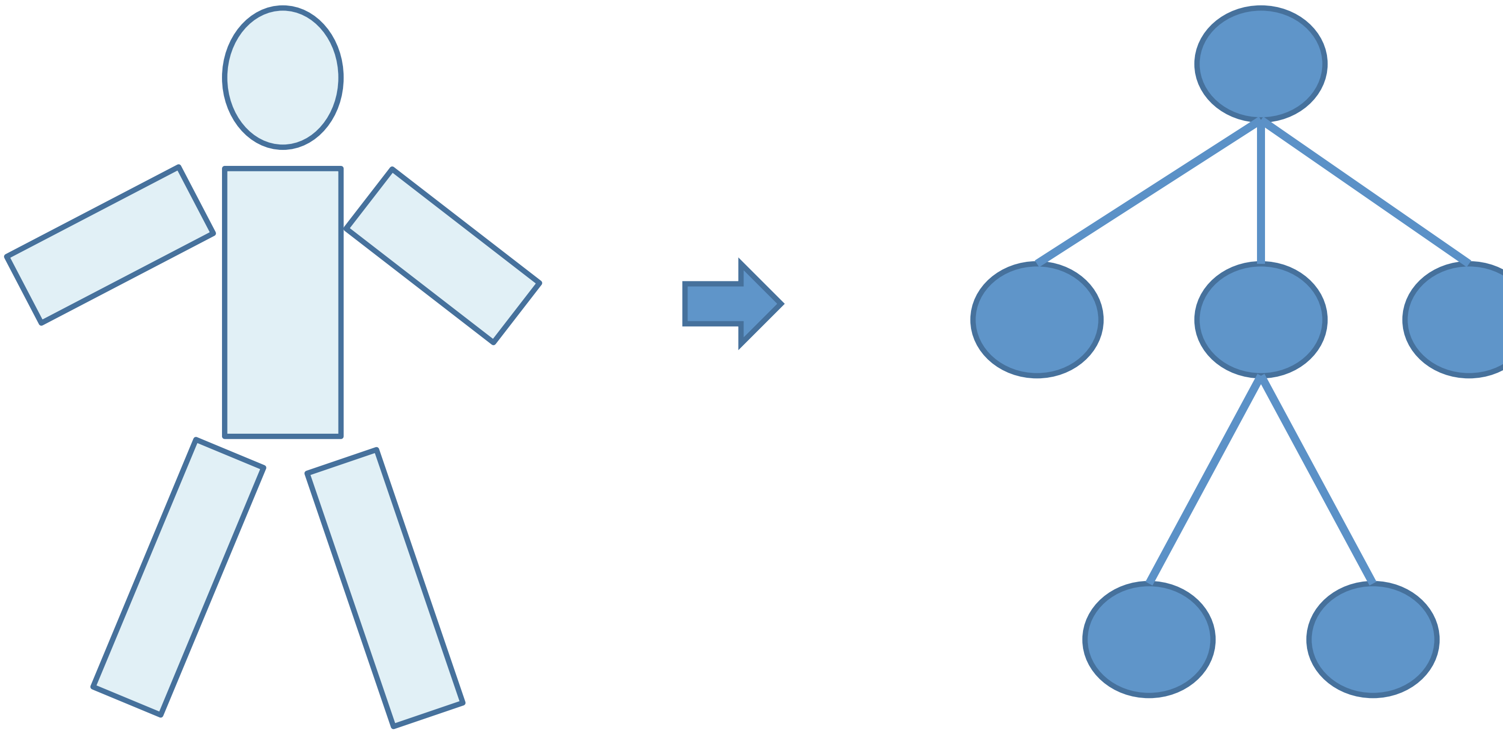
Csurka '04
Vasconcelos '00

Bouchard & Triggs '05

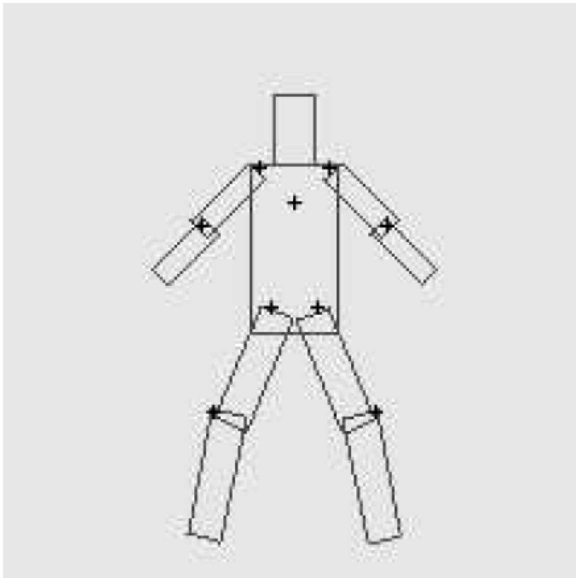
Carneiro & Lowe '06

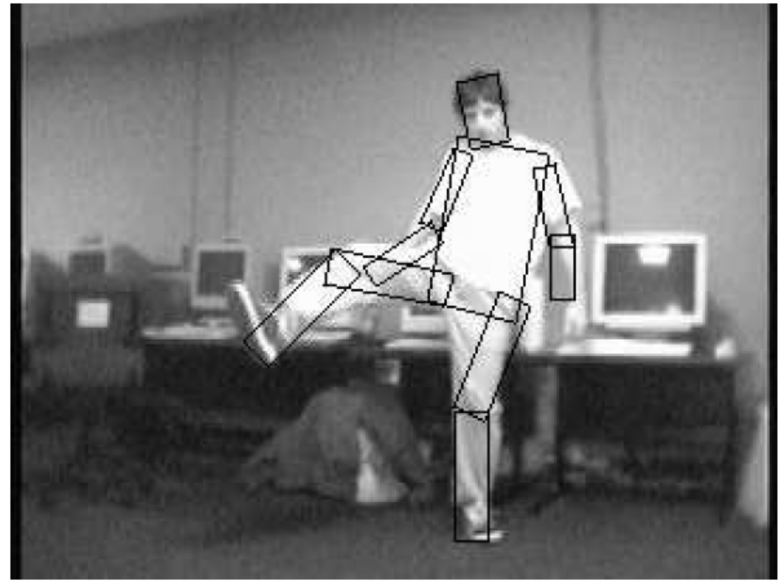
from [Carneiro & Lowe, ECCV'06]

Tree-shaped model



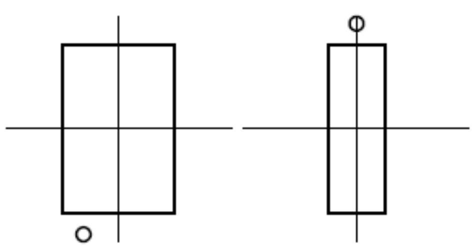
Pictorial Structures Model



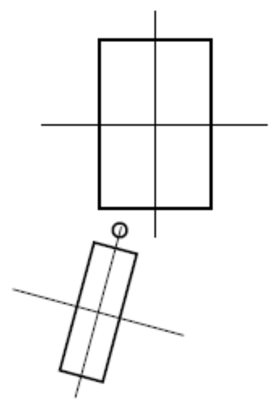


Part = oriented rectangle

Spatial model = relative size/orientation



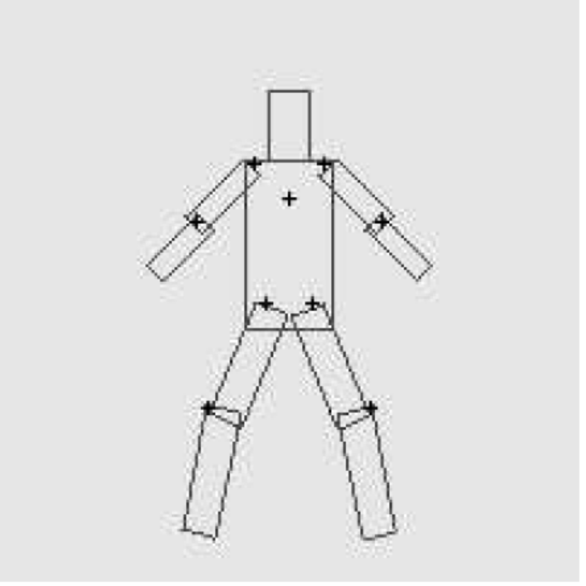
a



b

Felzenszwalb and Huttenlocher 2005

Pictorial Structures Model



$$P(L|I,\theta) \propto \left(\prod_{i=1}^{n} p(I|l_i, u_i) \prod_{(v_i,v_j)\in E} p(l_i, l_j|c_{ij}) \right)$$

Appearance likelihood

Geometry likelihood

Modeling the Appearance

- Any appearance model could be used
  - HOG Templates, etc.
  - Here: rectangles fit to background subtracted binary map

- Can train appearance models independently (easy, not as good) or jointly (more complicated but better)

$$P(L|I,\theta) \propto \left(\prod_{i=1}^{n} \underbrace{p(I|l_i,u_i)}_{\text{Appearance likelihood}} \prod_{(v_i,v_j)\in E} \underbrace{p(l_i,l_j|c_{ij})}_{\text{Geometry likelihood}} \right)$$

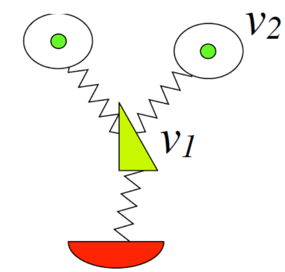
Part representation

- Background subtraction

Pictorial structures model

Optimization is tricky but can be efficient

$$L^* = \arg\min_L \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right)$$



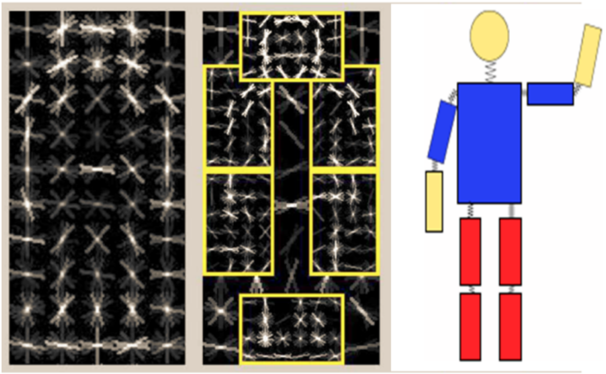
- For each l_1, find best l_2:

$$\text{Best}_2(l_1) = \min_{l_2} \left[m_2(l_2) + d_{12}(l_1, l_2) \right]$$

- Remove v_2, and repeat with smaller tree, until only a single part
- For k parts, n locations per part, this has complexity of $O(kn^2)$, but can be solved in ~$O(nk)$ using generalized distance transform

Pictorial Structures

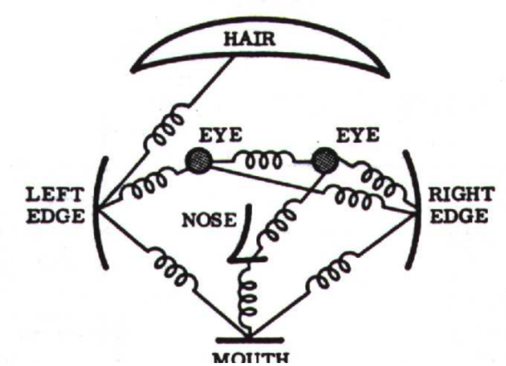
- Model is represented by a graph $G = (V, E)$.
  - $V = \{v_1, \ldots, v_n\}$ are the parts.
  - $(v_i, v_j) \in E$ indicates a connection between parts.
- $m_i(l_i)$ is the cost of placing part i at location l_i.
- $d_{ij}(l_i, l_j)$ is a deformation cost.
- Optimal location for object is given by $L^* = (l_1^*, \ldots, l_n^*)$,

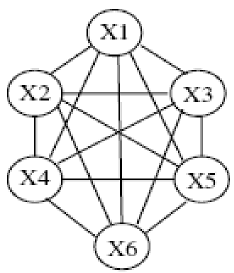


$$L^* = \underset{L}{\operatorname{argmin}} \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i, v_j) \in E} d_{ij}(l_i, l_j) \right)$$

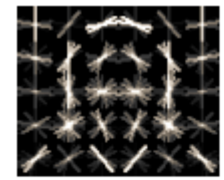
$$L^* = \underset{L}{\text{argmin}} \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right)$$

- n parts and h locations gives h^n configurations.

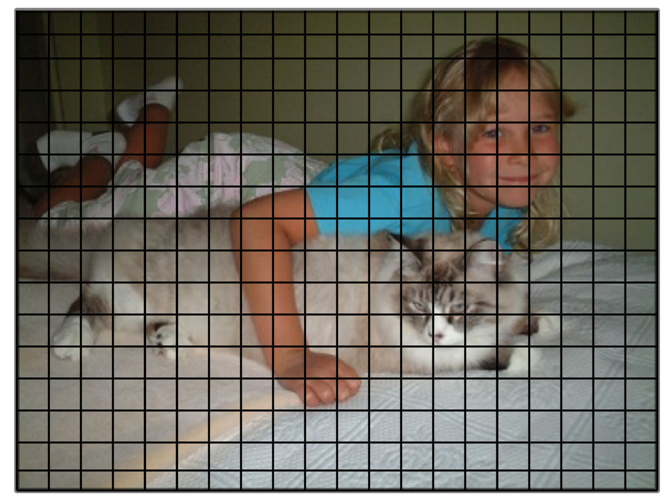




a) Constellation [13]



head filter



Complexity $O(h^n)$

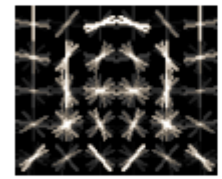
h: number of possible part placements

n: number of parts

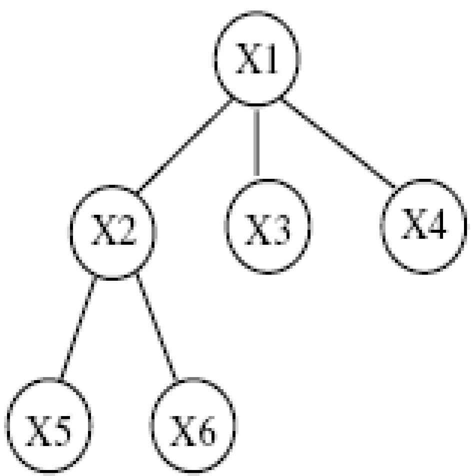
Efficient minimization

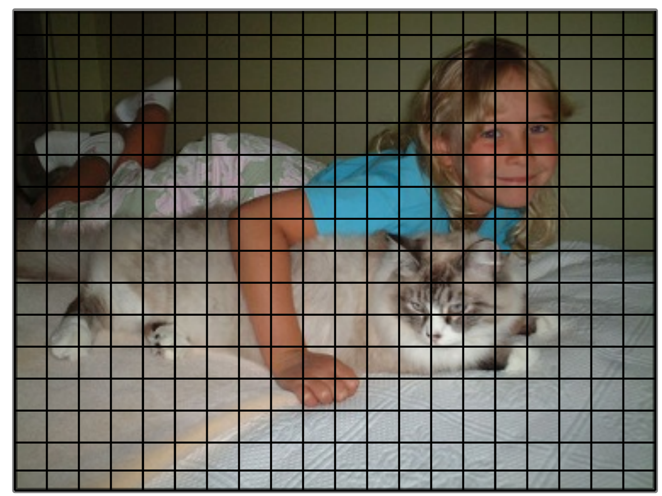
$$L^* = \underset{L}{\operatorname{argmin}} \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right)$$

- n parts and h locations gives h^n configurations.
- If graph is a tree we can use dynamic programming.
  - $O(nh^2)$, much better but still slow.



head filter





Complexity $O(nh^2)$

Efficient minimization

$$L^* = \underset{L}{\text{argmin}} \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right)$$

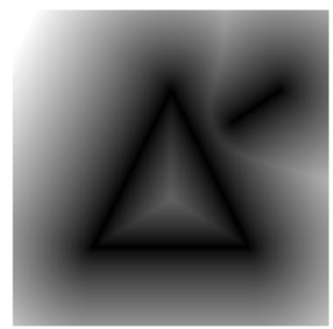
- n parts and h locations gives h^n configurations.
- If graph is a tree we can use dynamic programming.
  - $O(nh^2)$, much better but still slow.
- If $d_{ij}(l_i, l_j) = ||T_{ij}(l_i) - T_{ji}(l_j)||^2$ can use DT.
  - $O(nh)$, as good as matching each part separately!!

Distance transform

Given a set of points on a grid $P \subseteq \mathcal{G}$,
the quadratic distance transform of P is,

$$\mathcal{D}_P(q) = \min_{p \in P} ||q - p||^2$$

P



$\mathcal{D}_P$

Generalized distance transform

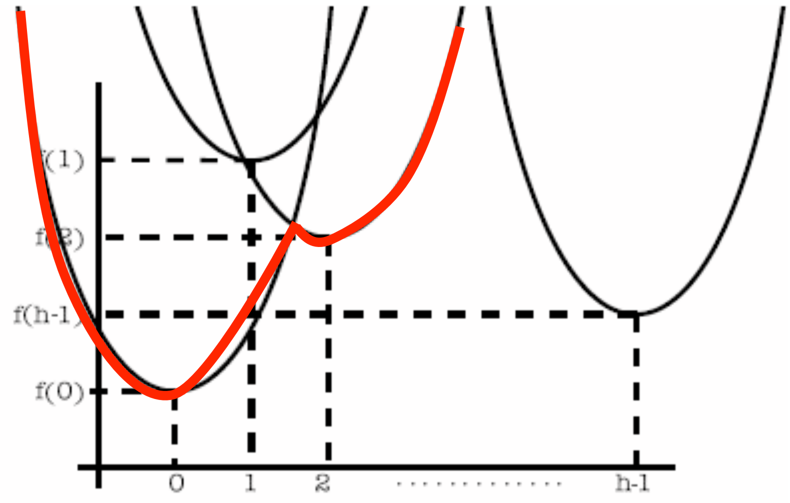
Given a function $f : \mathcal{G} \rightarrow \mathbb{R}$,

$$\mathcal{D}_f(q) = \min_{p \in \mathcal{G}} \left(||q - p||^2 + f(p) \right)$$

– for each location q, find nearby location p with $f(p)$ small.

1D case: $\mathcal{D}_f(q) = \text{min}_{p \in \mathcal{G}} \left((q-p)^2 + f(p) \right)$

For each p, $\mathcal{D}_f(q)$ is below the parabola rooted at $(p, f(p))$.



There is a simple geometric algorithm that computes $\mathcal{D}_f(p)$ in $O(h)$ time for the 1D case.

- similar to Graham's scan convex hull algorithm.
- about 20 lines of C code.

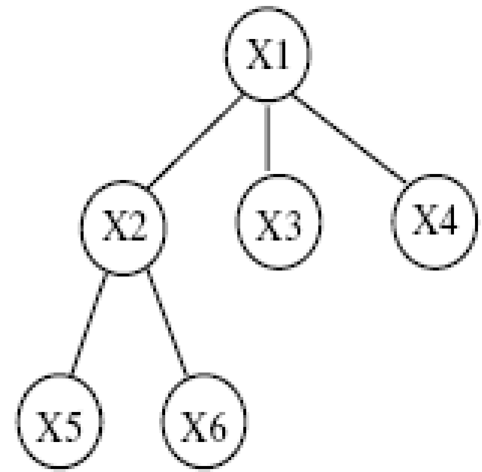
The 2D case is "separable", it can be solved by sequential 1D transformations along rows and columns of the grid.

See **Distance Transforms of Sampled Functions**, Felzenszwalb and Huttenlocher.

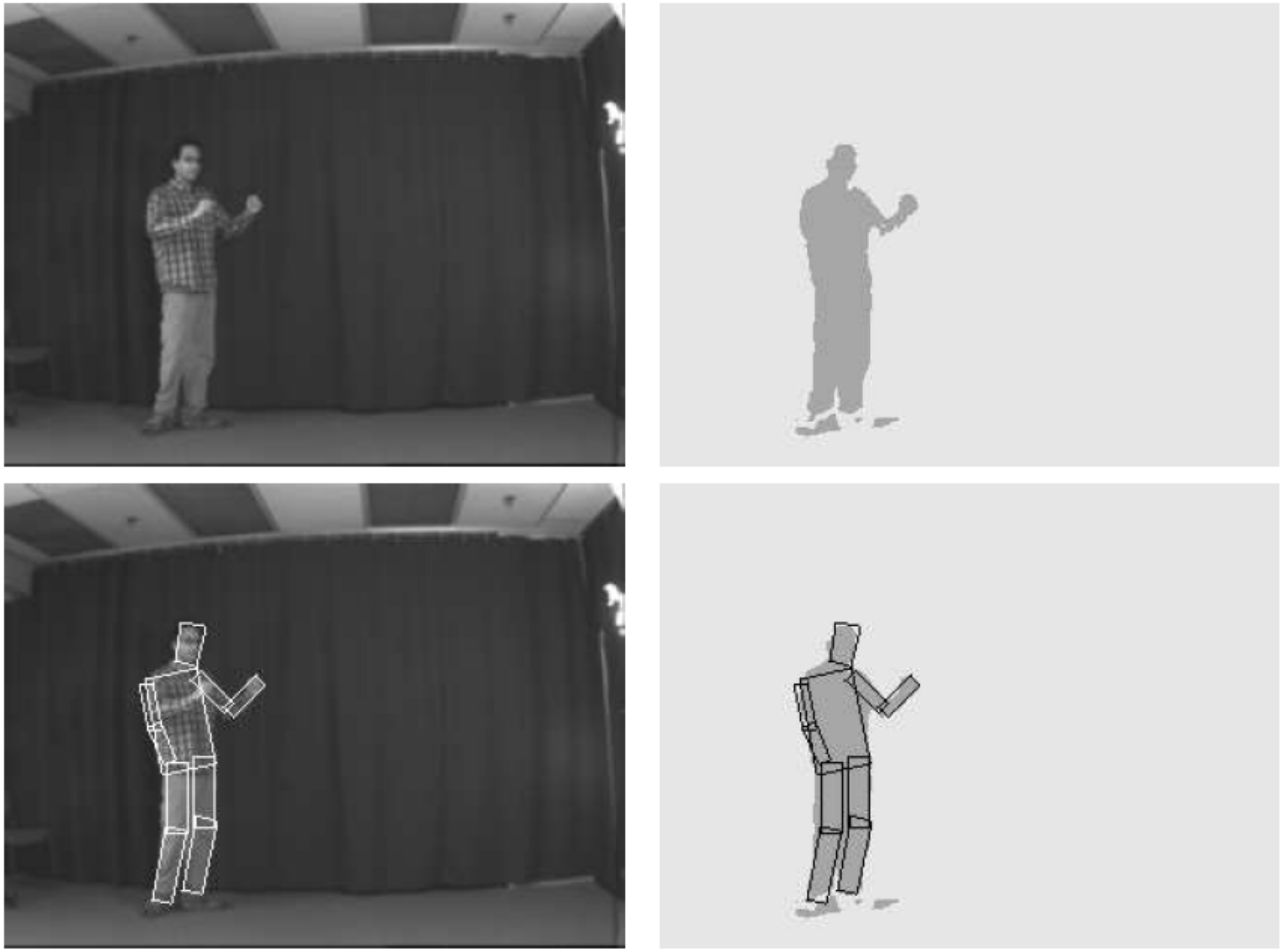
Pictorial Structures: Summary

$$L^* = \underset{L}{\operatorname{argmin}} \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right)$$

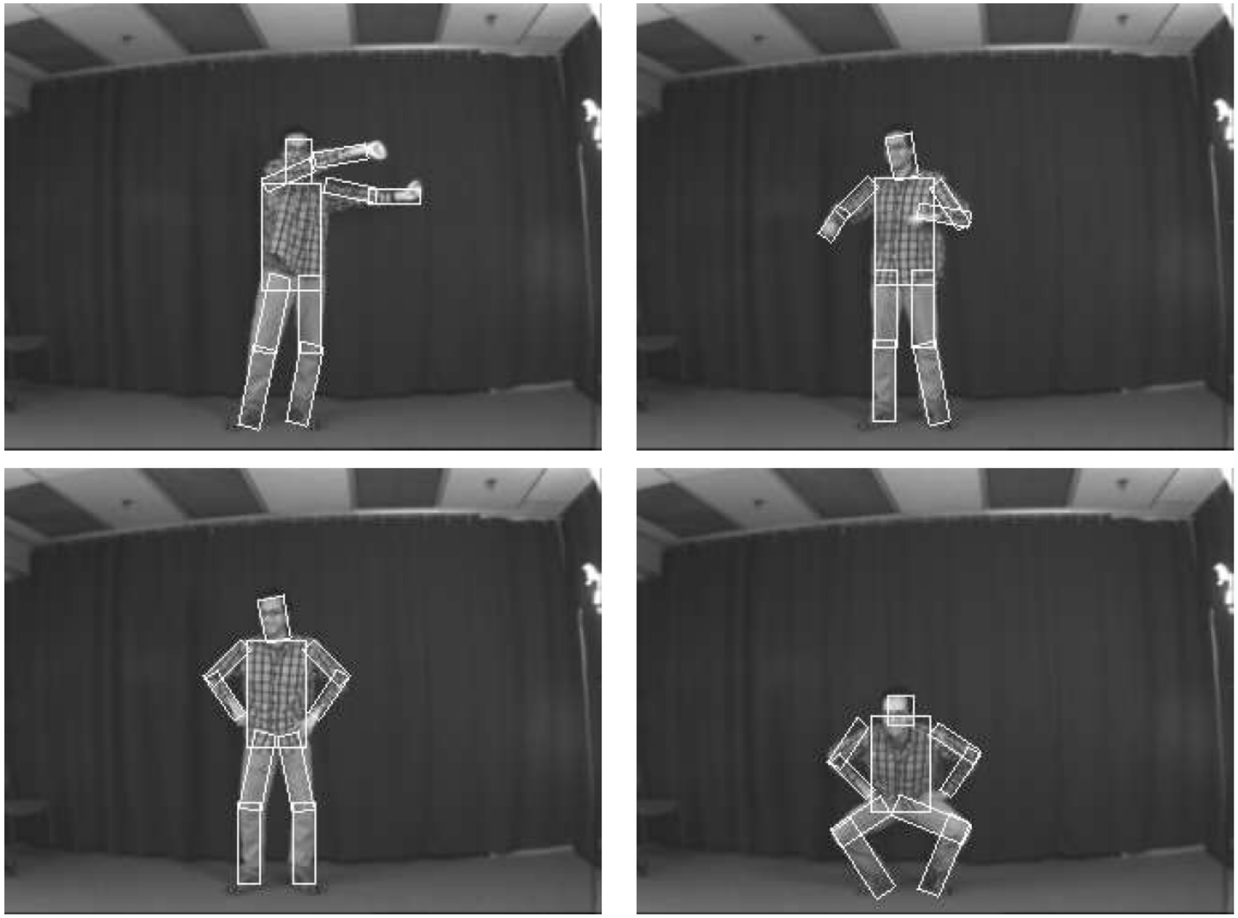
$$d_{ij}(l_i, l_j) = ||T_{ij}(l_i) - T_{ji}(l_j)||^2$$



Results for person matching



Results for person matching



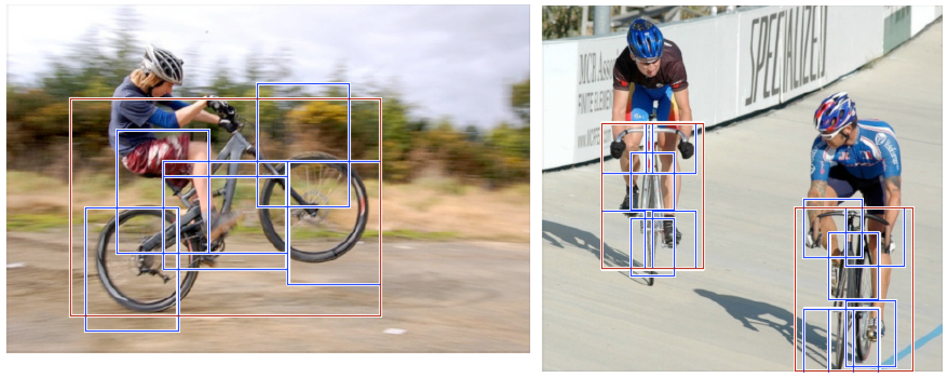
Enhanced pictorial structures

BMVC 2009

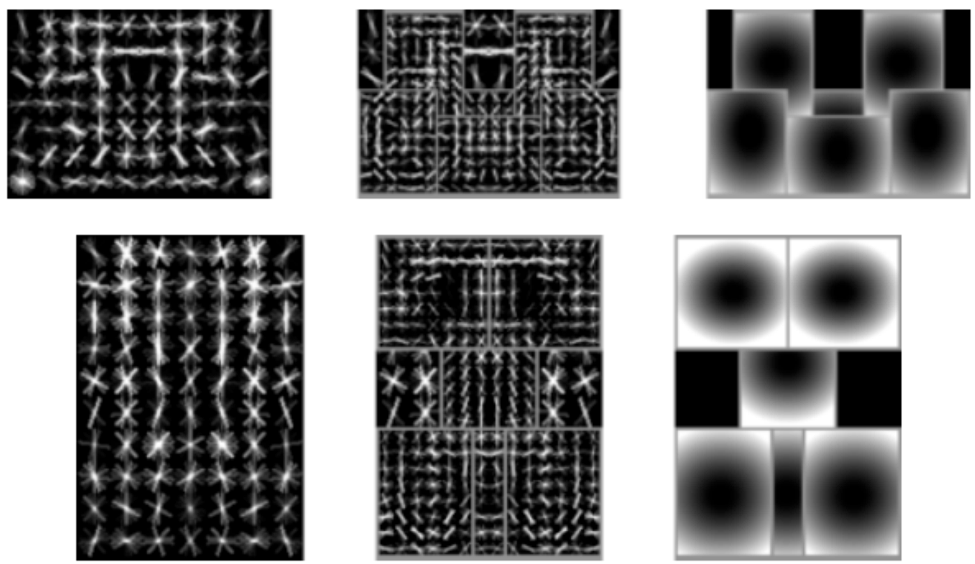
Deformable Latent Parts Model

Useful parts discovered during training

Detections



Template Visualization



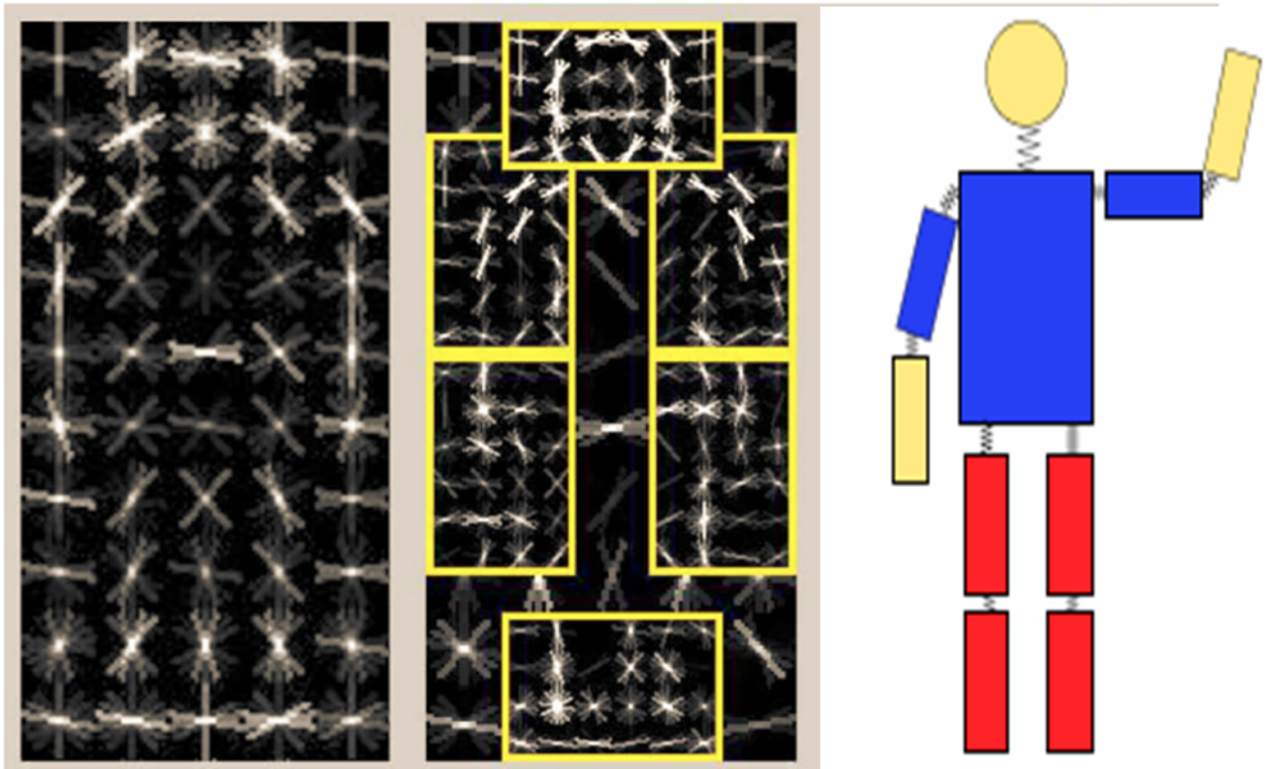
root filters
coarse resolution

part filters
finer resolution

deformation
models

Felzenszwalb et al. 2008

Deformable Part Models



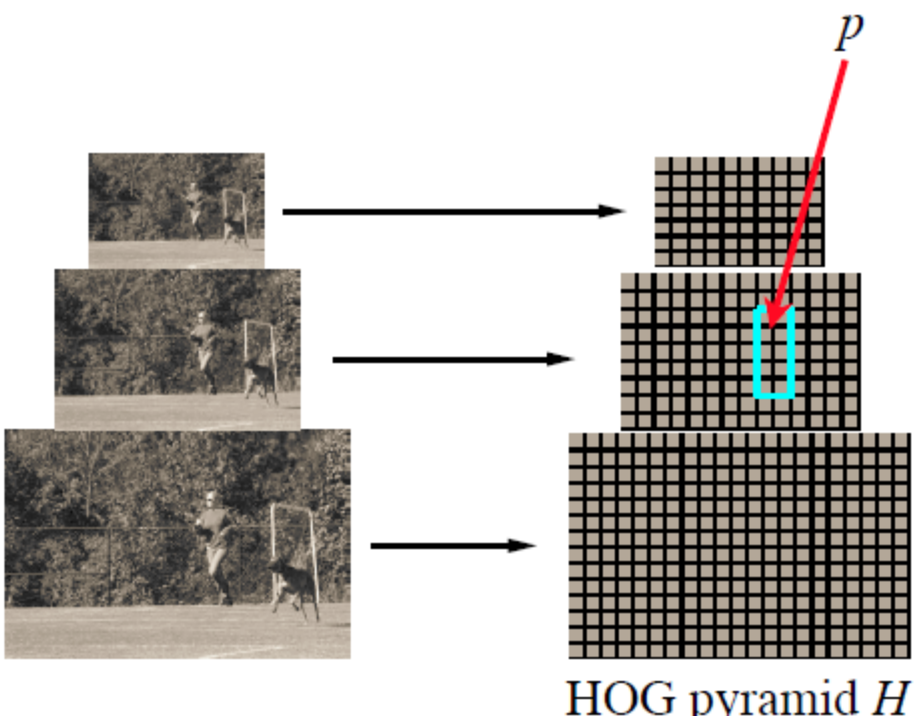
Root filter Part filter Quadratic

$$\textit{Score} = F_0.\Phi(p_0,H) + \Sigma F_i.\Phi(p_i,H) - \Sigma d_i.\Phi_d(x,y)$$

$$\left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i,l_j)\right)$$

HOG Filters

- Array of weights for features in subwindow of HOG pyramid
- Score is dot product of filter and feature vector

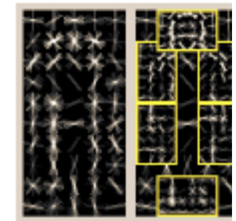


Filter F

Score of F at position p is
$F \cdot \phi(p, H)$

$\phi(p, H)$ = concatenation of
HOG features from
subwindow specified by p

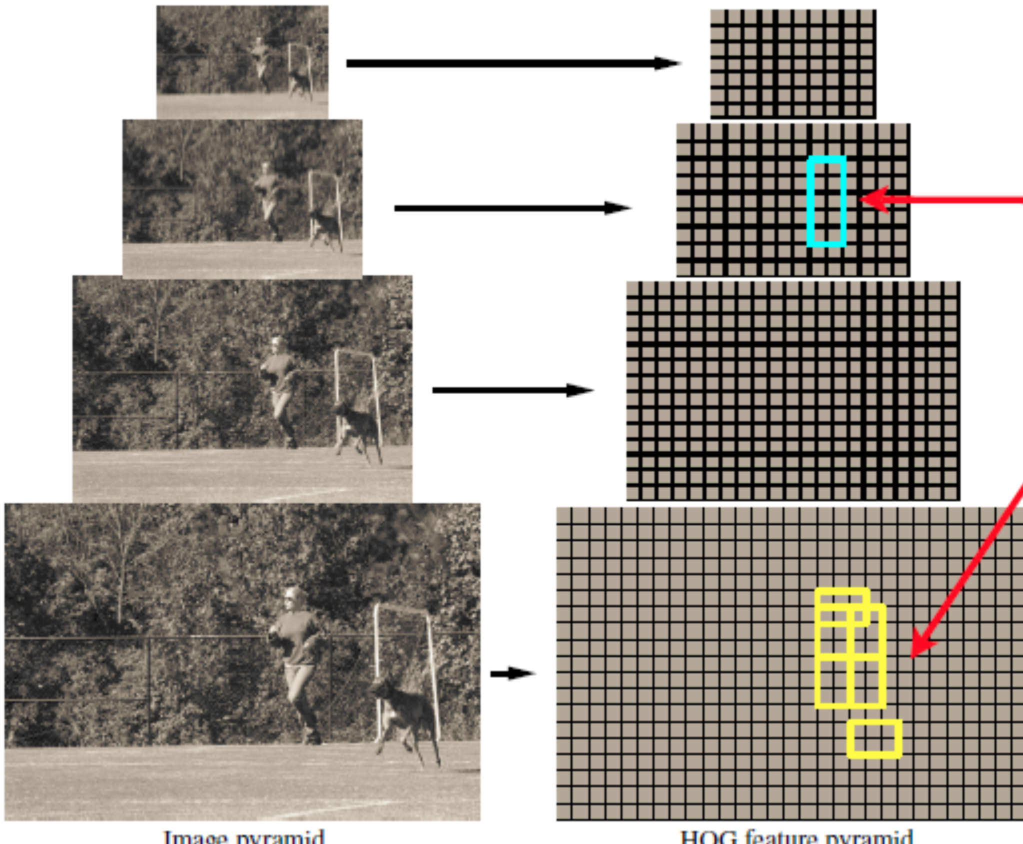
Object hypothesis



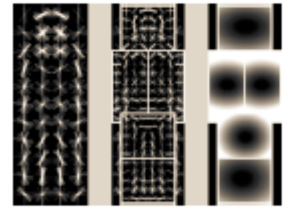
$z = (p_0, \ldots, p_n)$

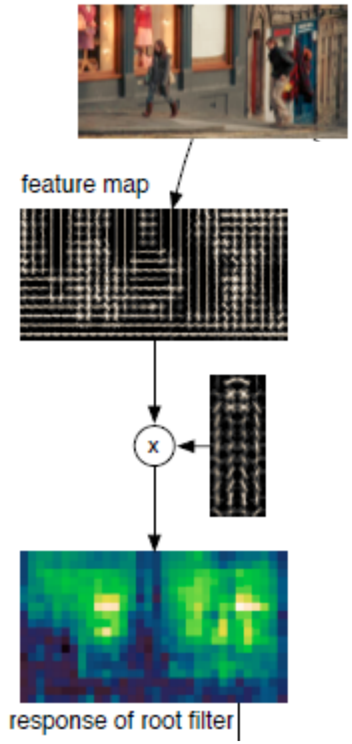
p_0 : location of root

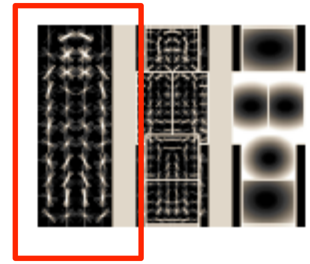
$p_1, \ldots, p_n$: location of parts

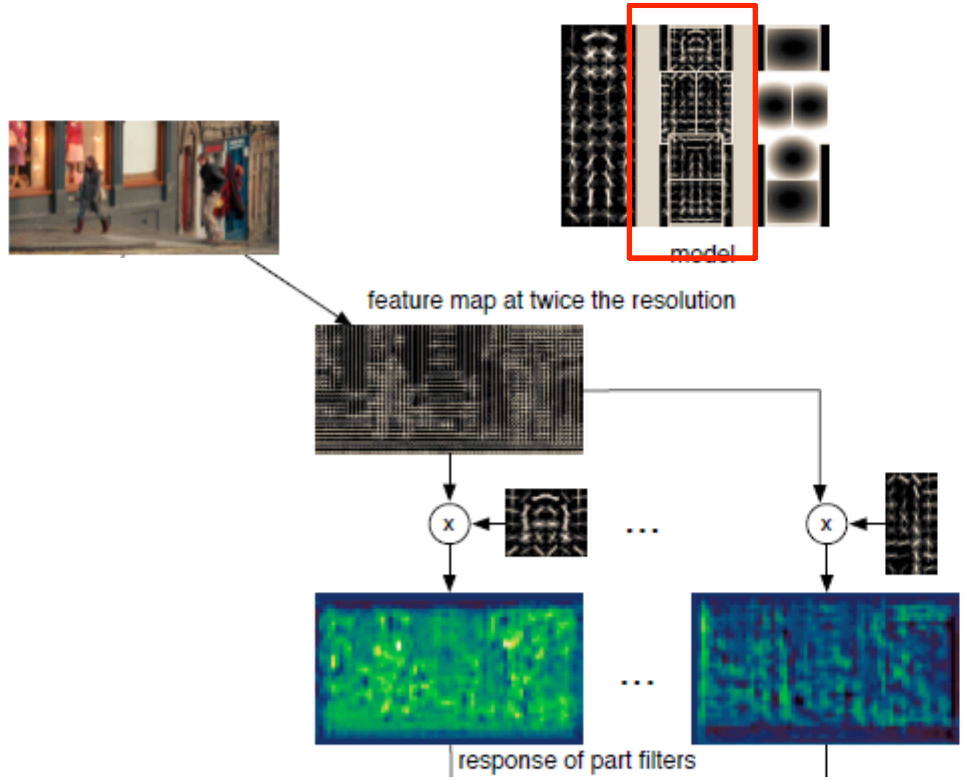


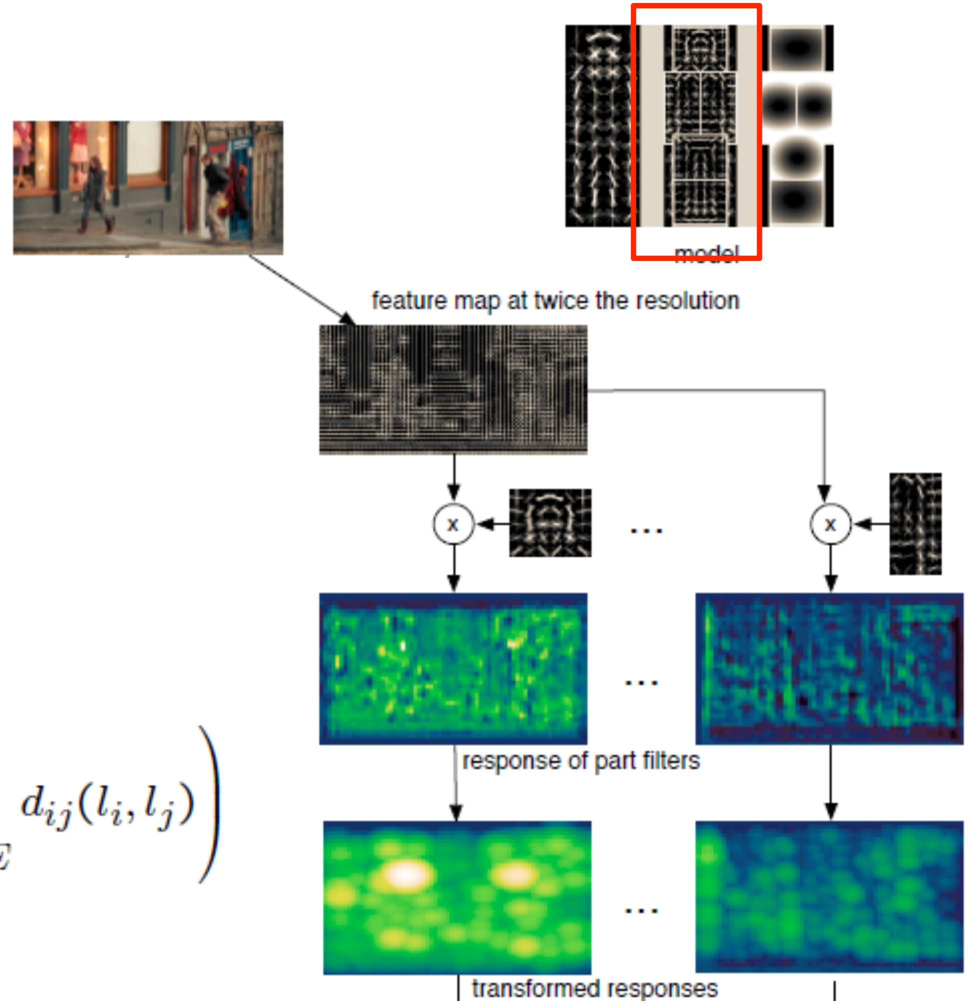
Multiscale model captures features at two-resolutions



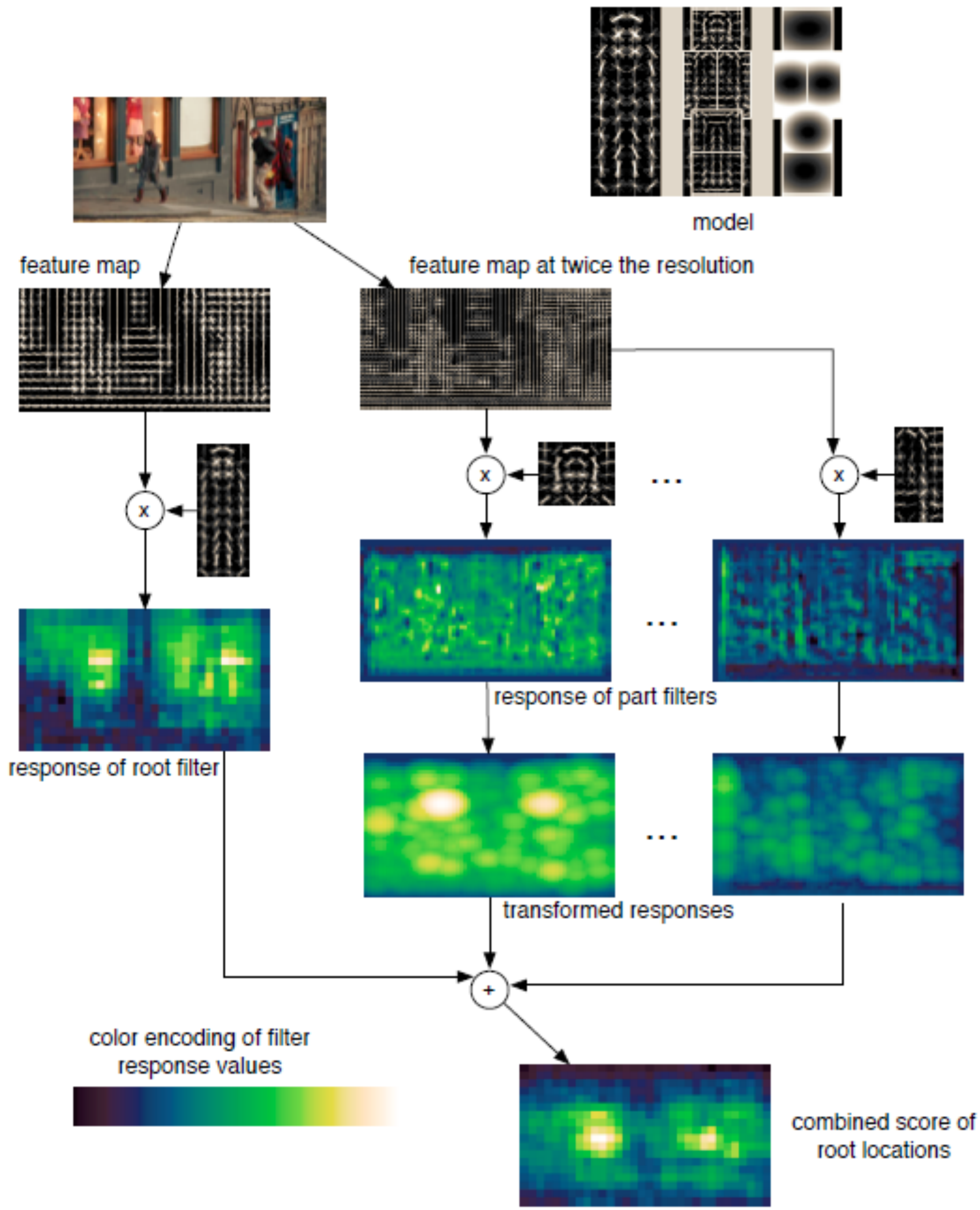

feature map
x
response of root filter



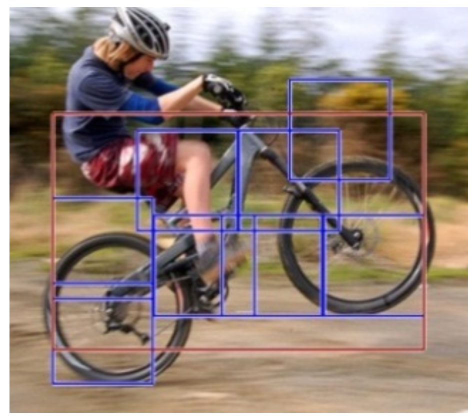

model
feature map at twice the resolution
x
x
...
...
response of part filters

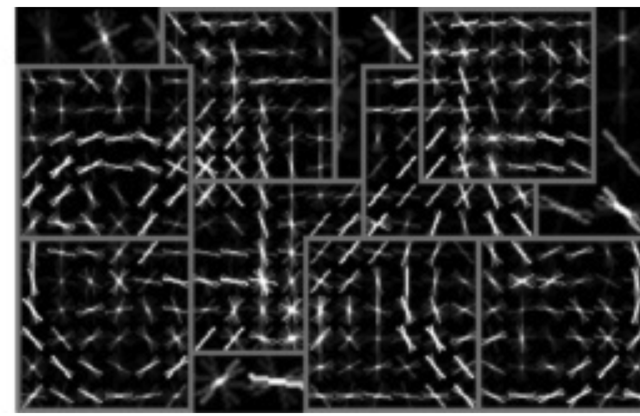


$$\left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right)$$


model
feature map
feature map at twice the resolution
x
x
x
...
...
...
response of part filters
response of root filter
transformed responses
+
color encoding of filter response values
combined score of root locations

State-of-the-art Detector: Deformable Parts Model (DPM)

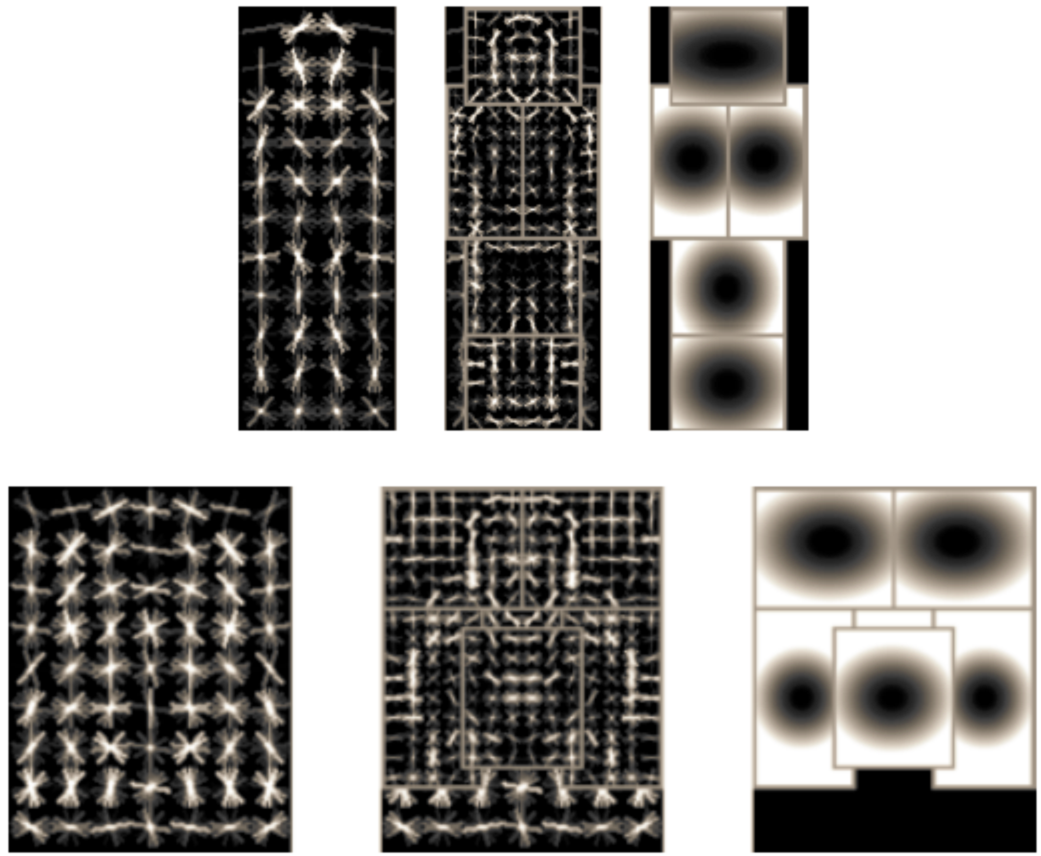




1. Strong low-level features based on HOG
2. Efficient matching algorithms for deformable part-based models (pictorial structures)
3. Discriminative learning with latent variables (latent SVM)

Felzenszwalb et al., 2008, 2010, 2011, 2012

Person model



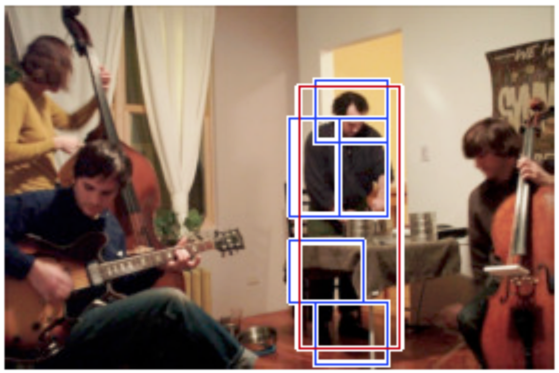
root filters
coarse resolution

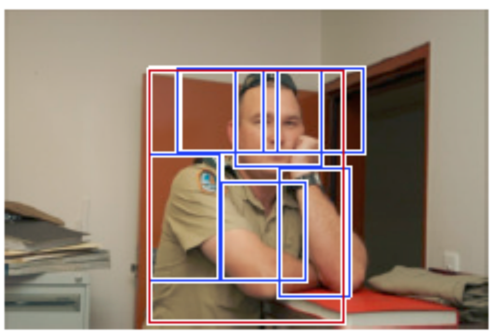
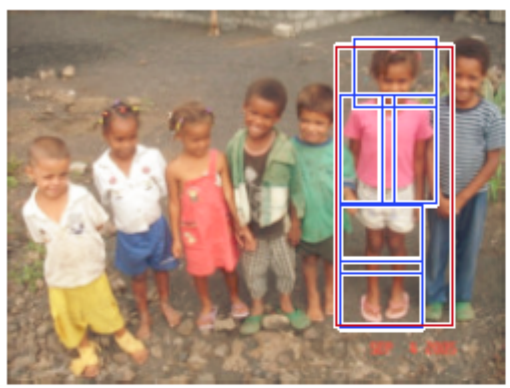
part filters
finer resolution

deformation
models

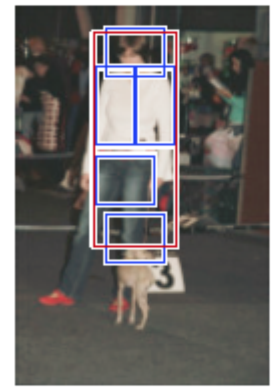
Person detections

high scoring true positives

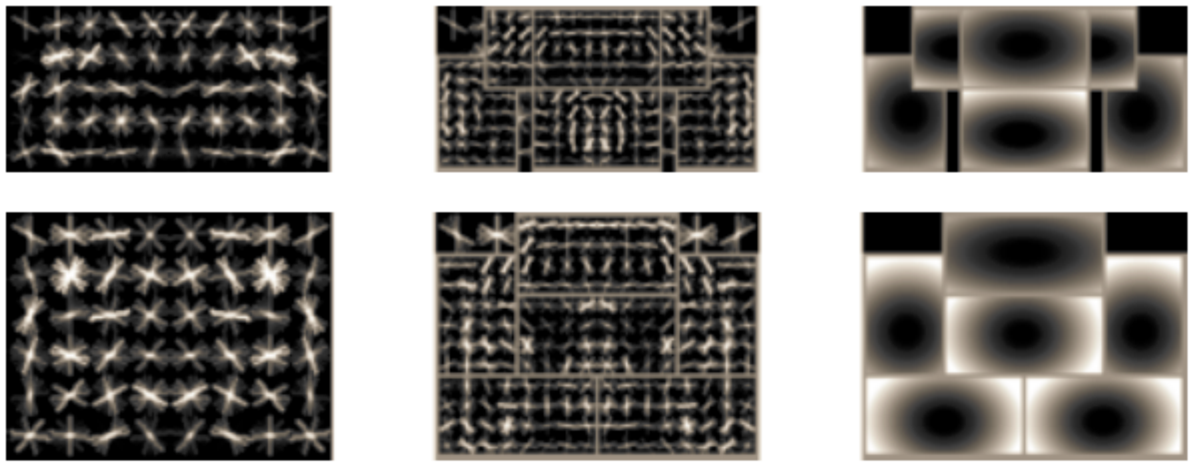


high scoring false positives
(not enough overlap)




Car



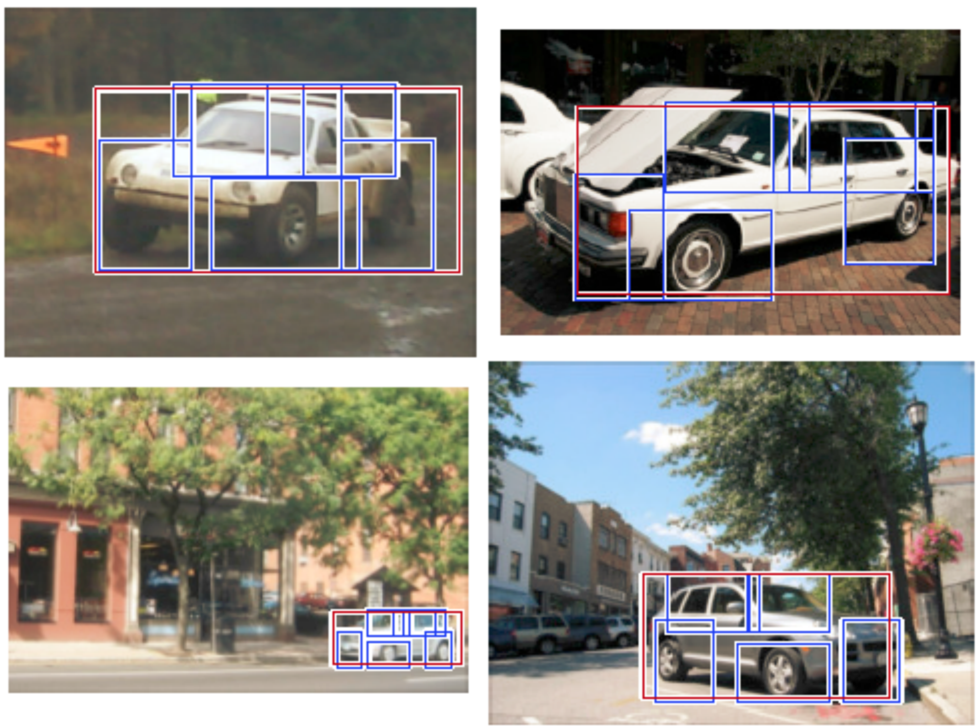
root filters
coarse resolution

part filters
finer resolution

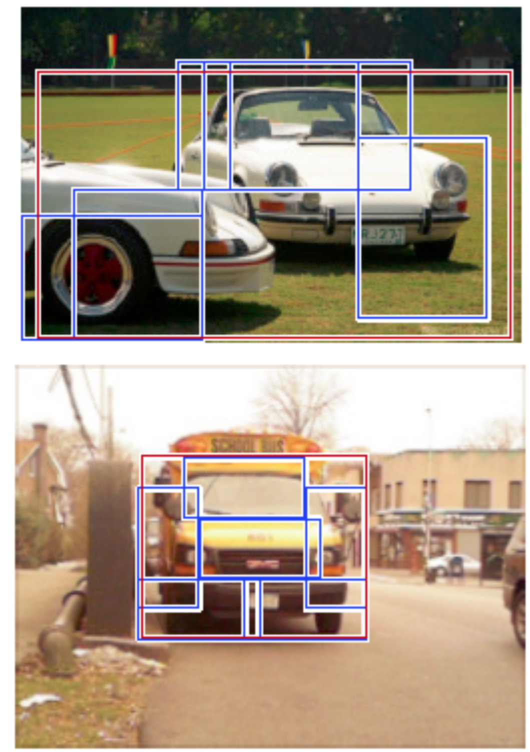
deformation
models

Car detections

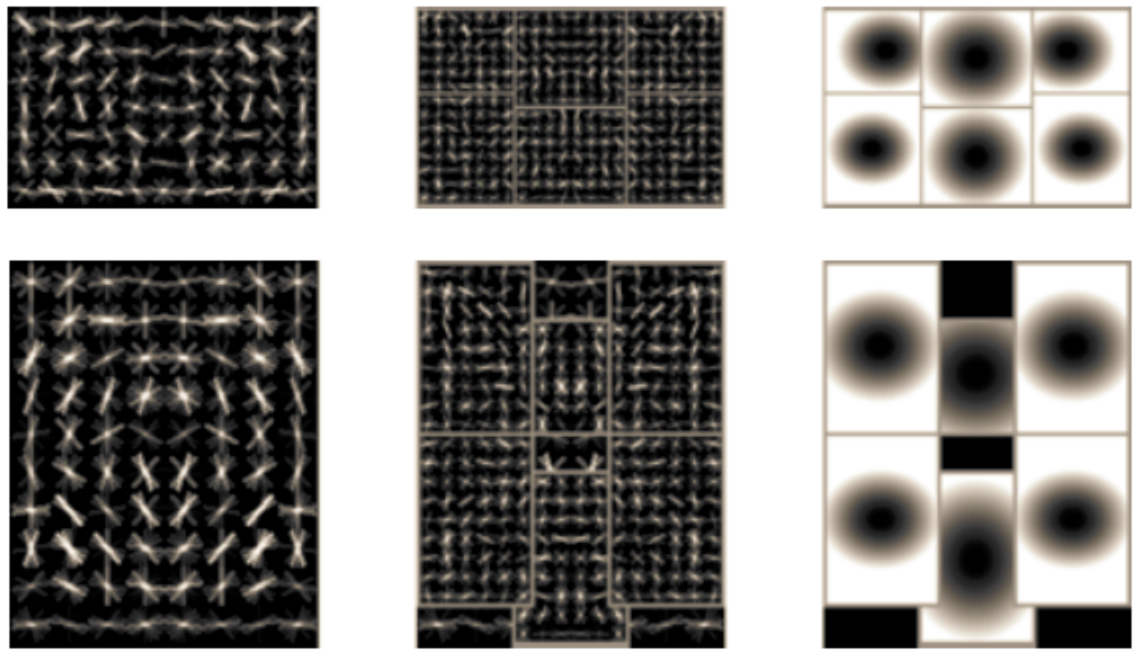
high scoring true positives



high scoring false positives



Cat



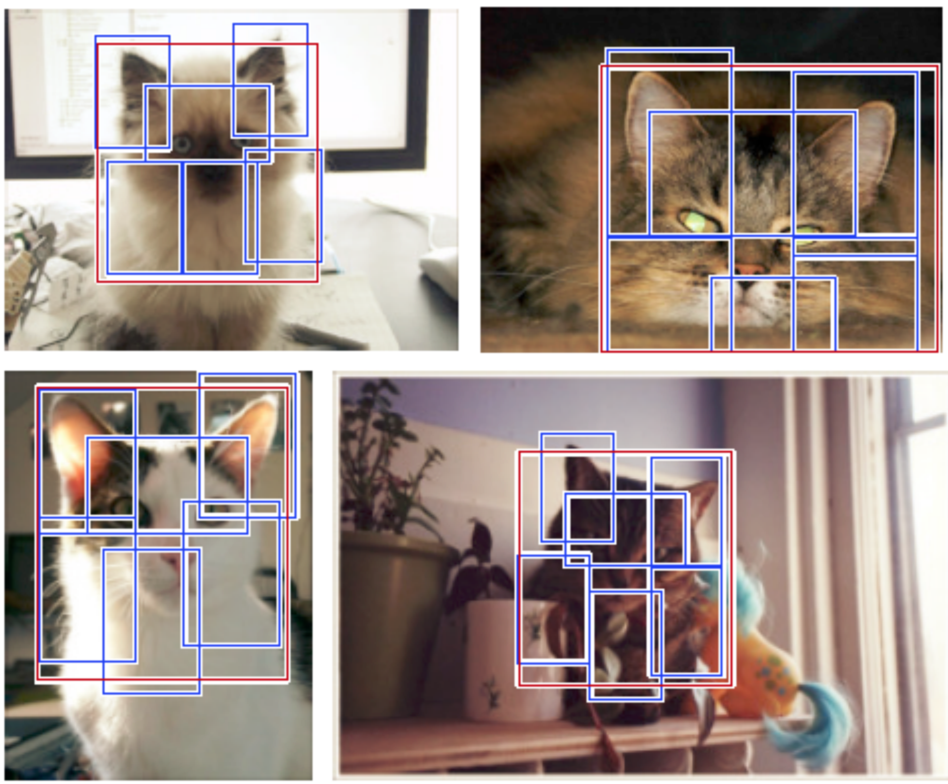
root filters
coarse resolution

part filters
finer resolution

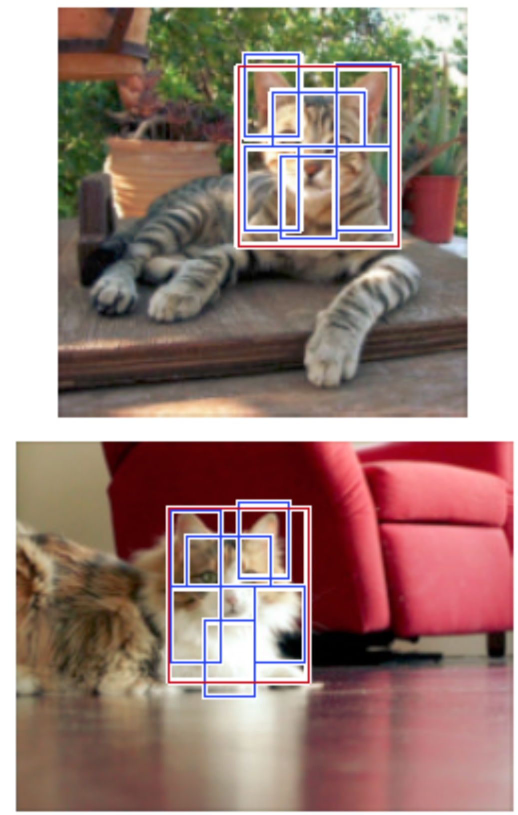
deformation
models

Cat detections

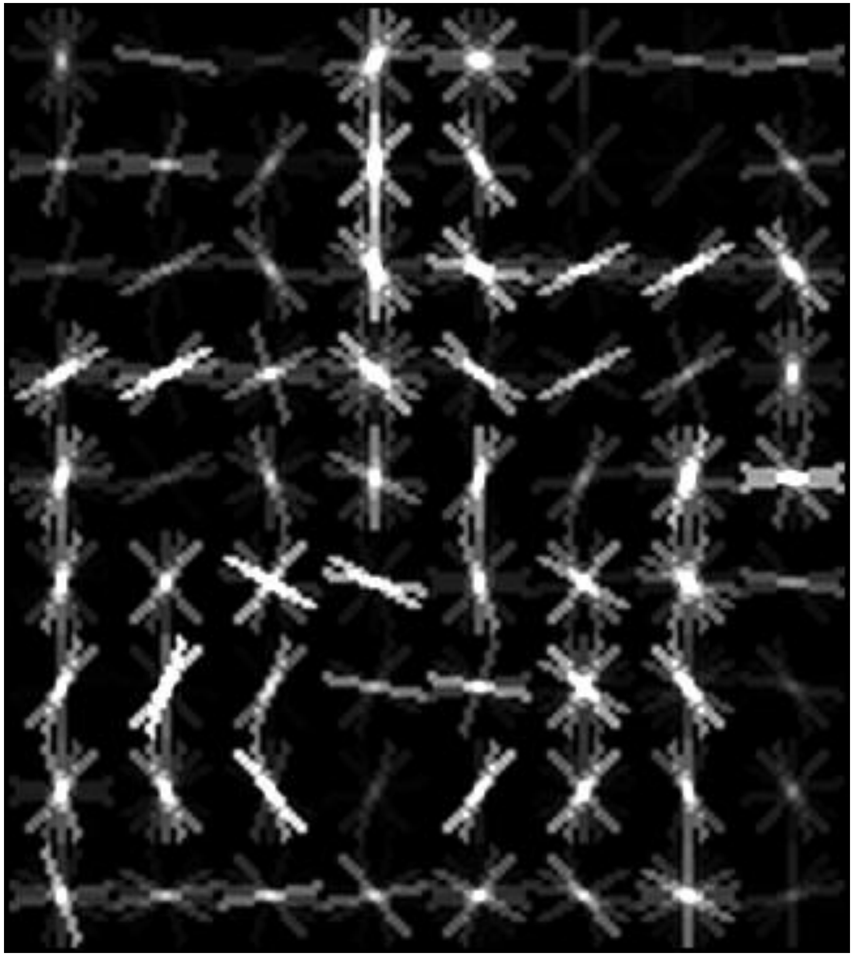
high scoring true positives



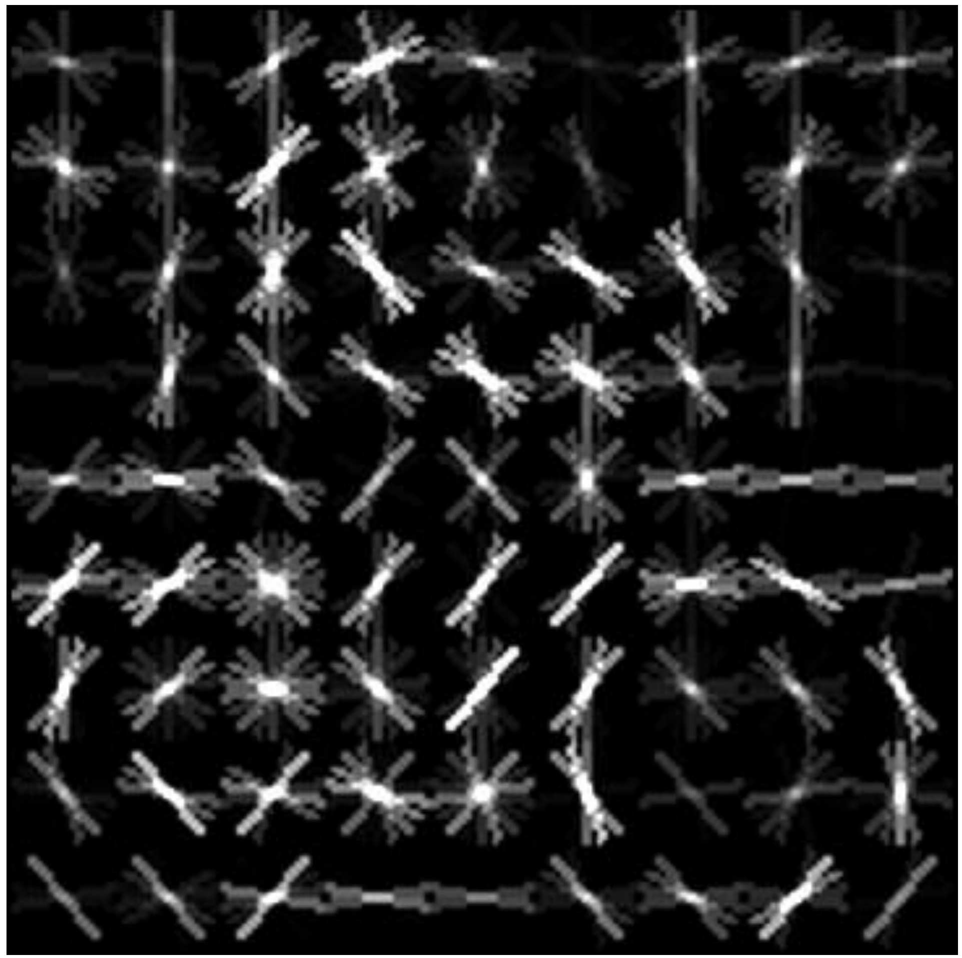
high scoring false positives
(not enough overlap)

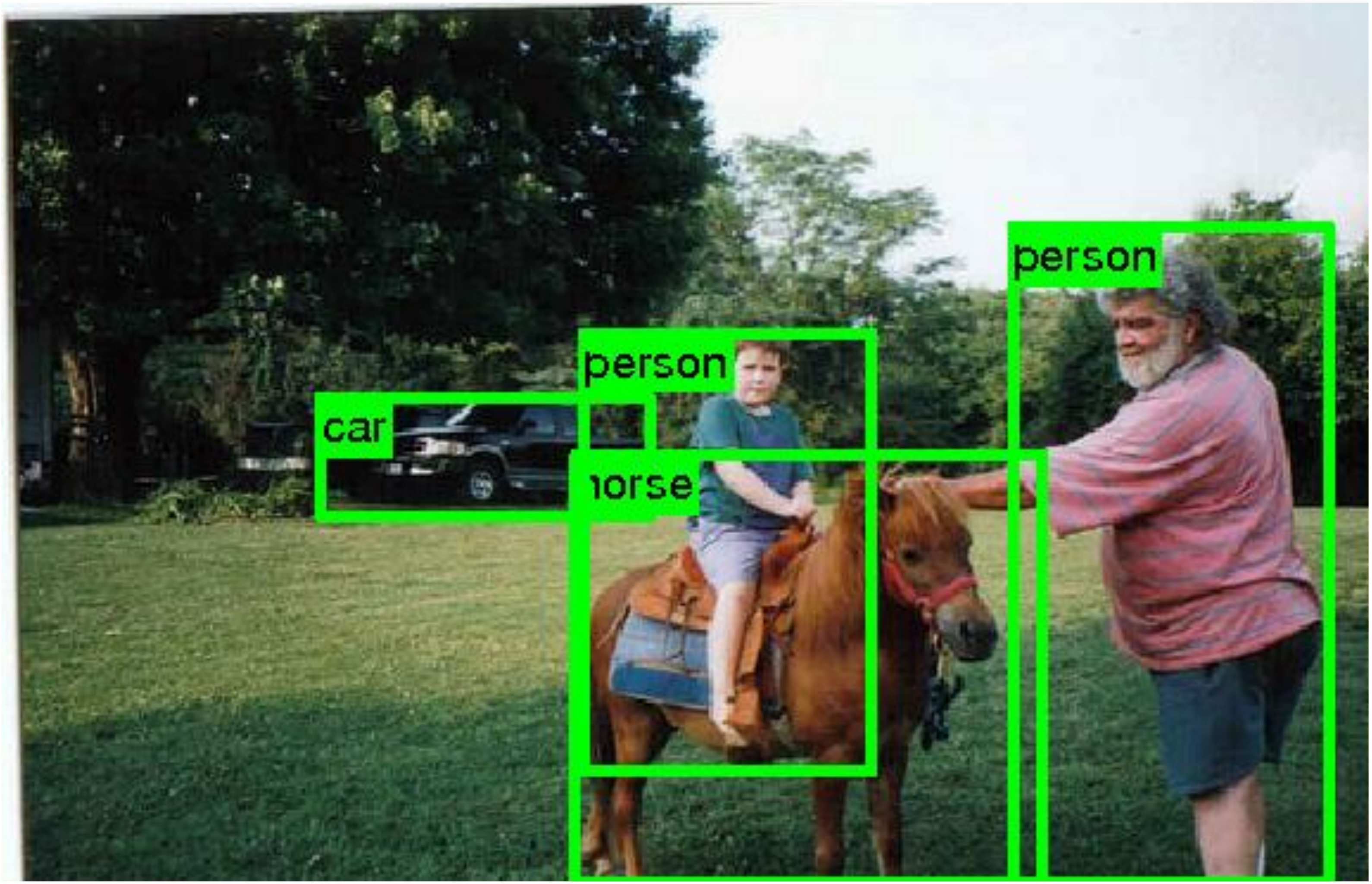


Person riding horse

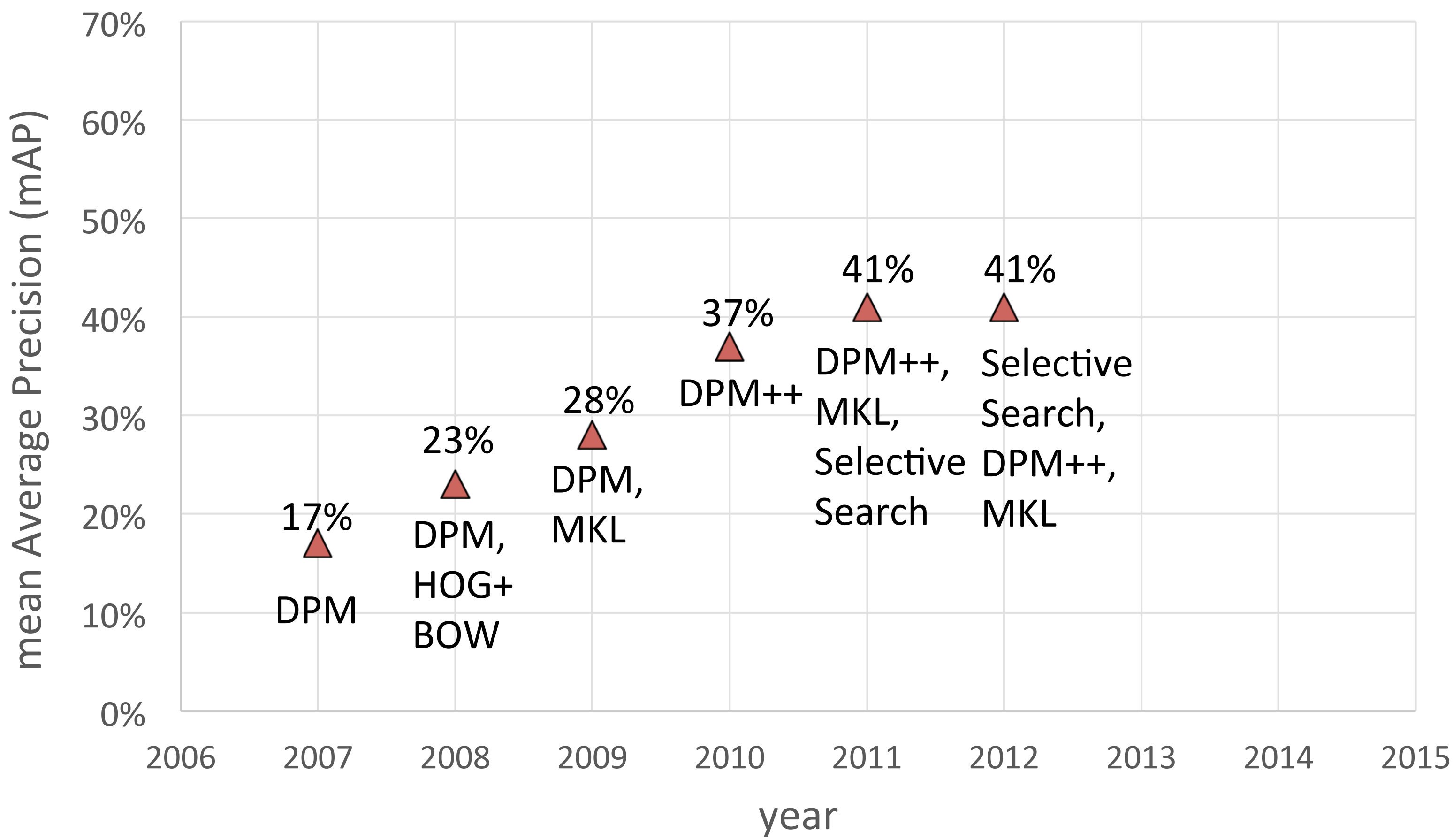


Person riding bicycle

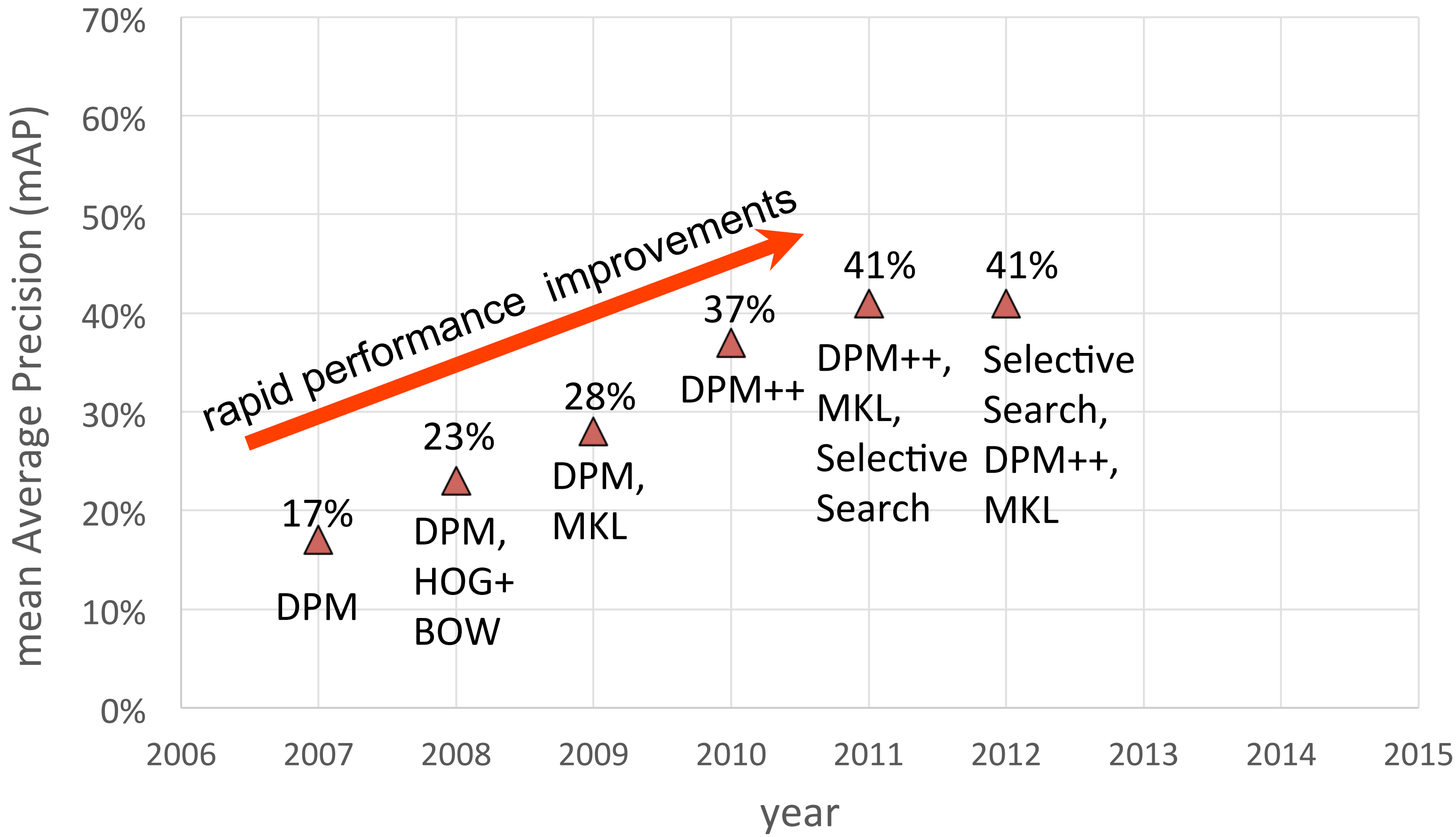



person
person
car
horse

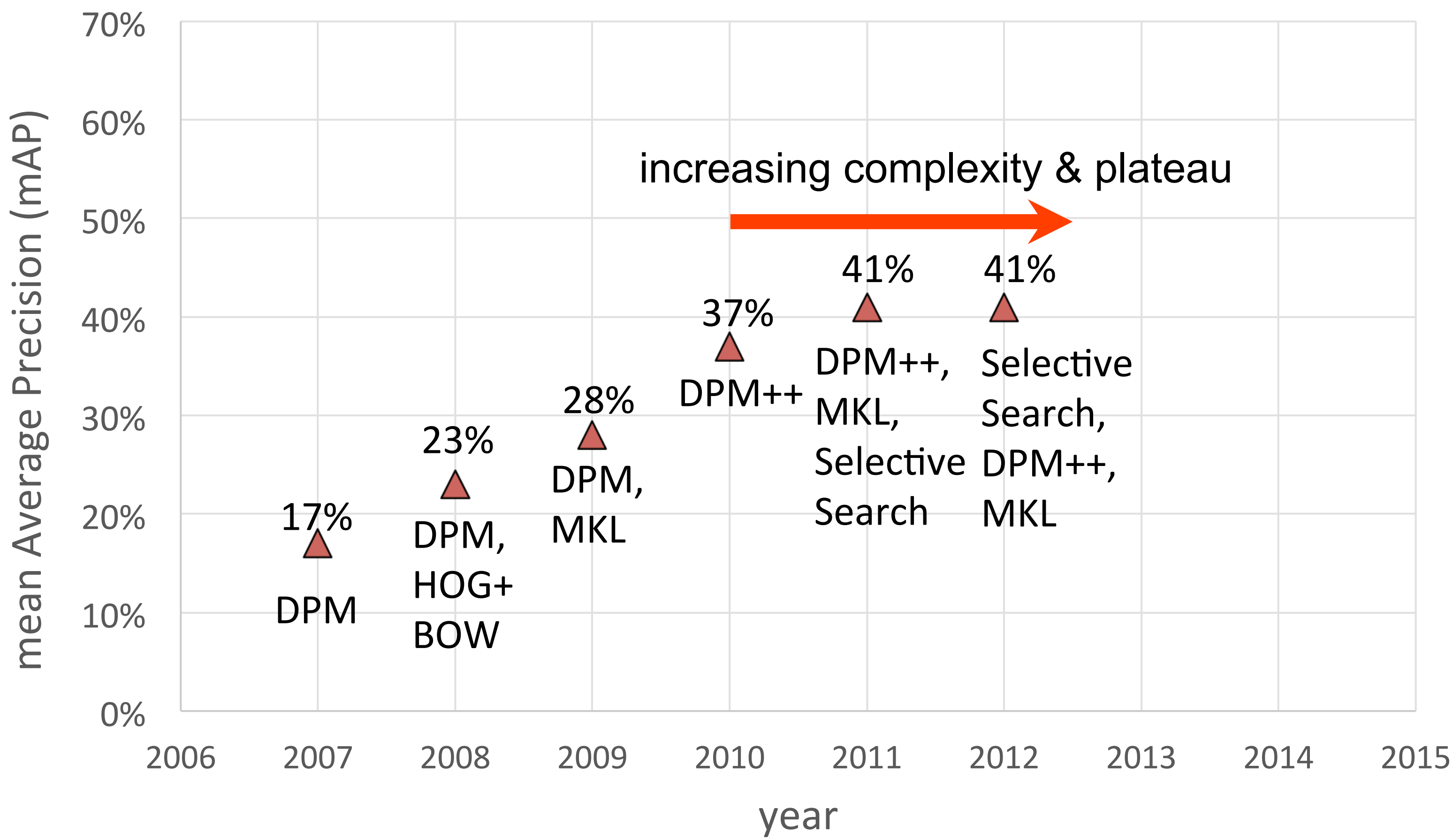
PASCAL VOC detection history



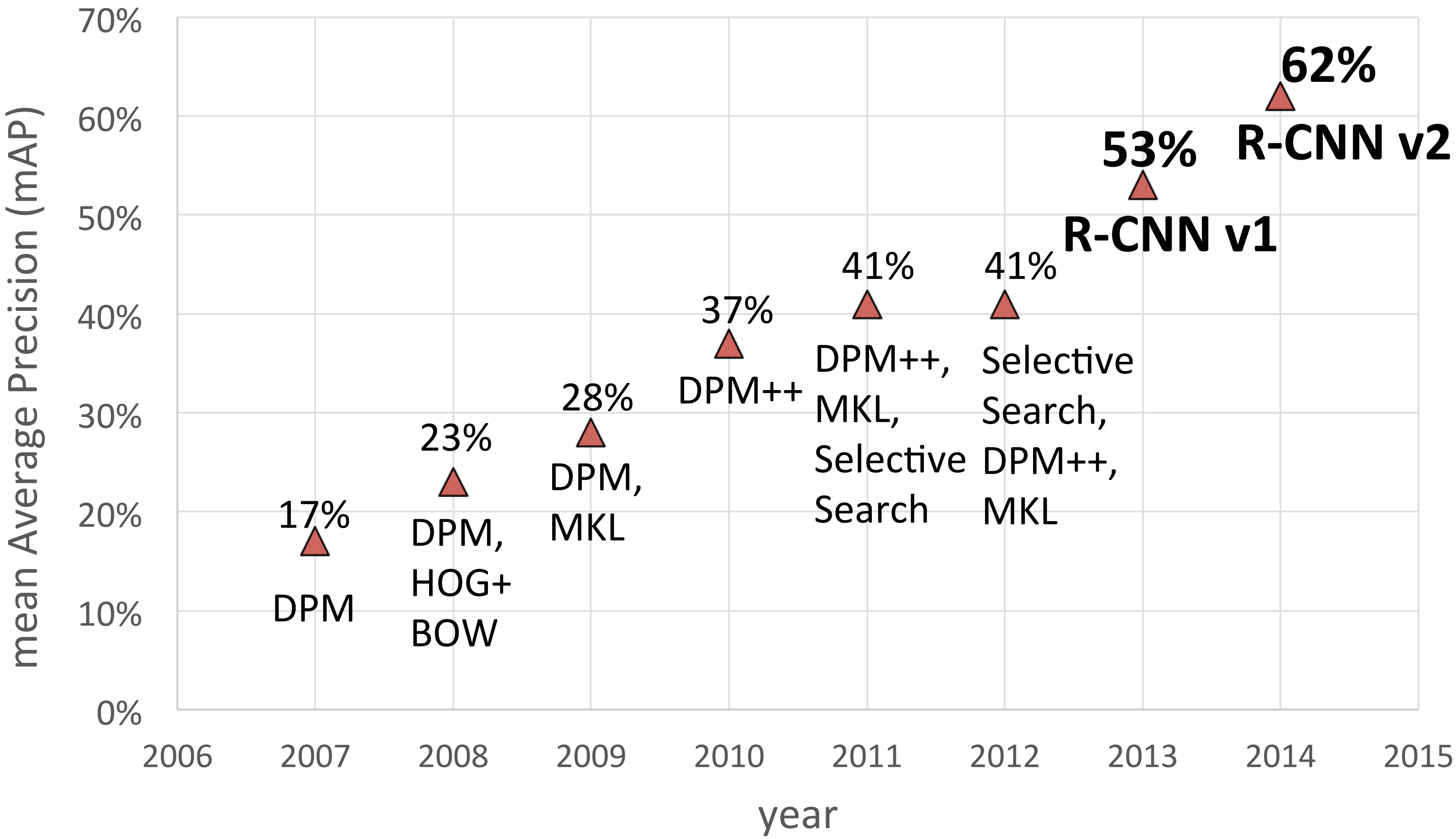
Part-based models & multiple features (MKL)



Kitchen-sink approaches

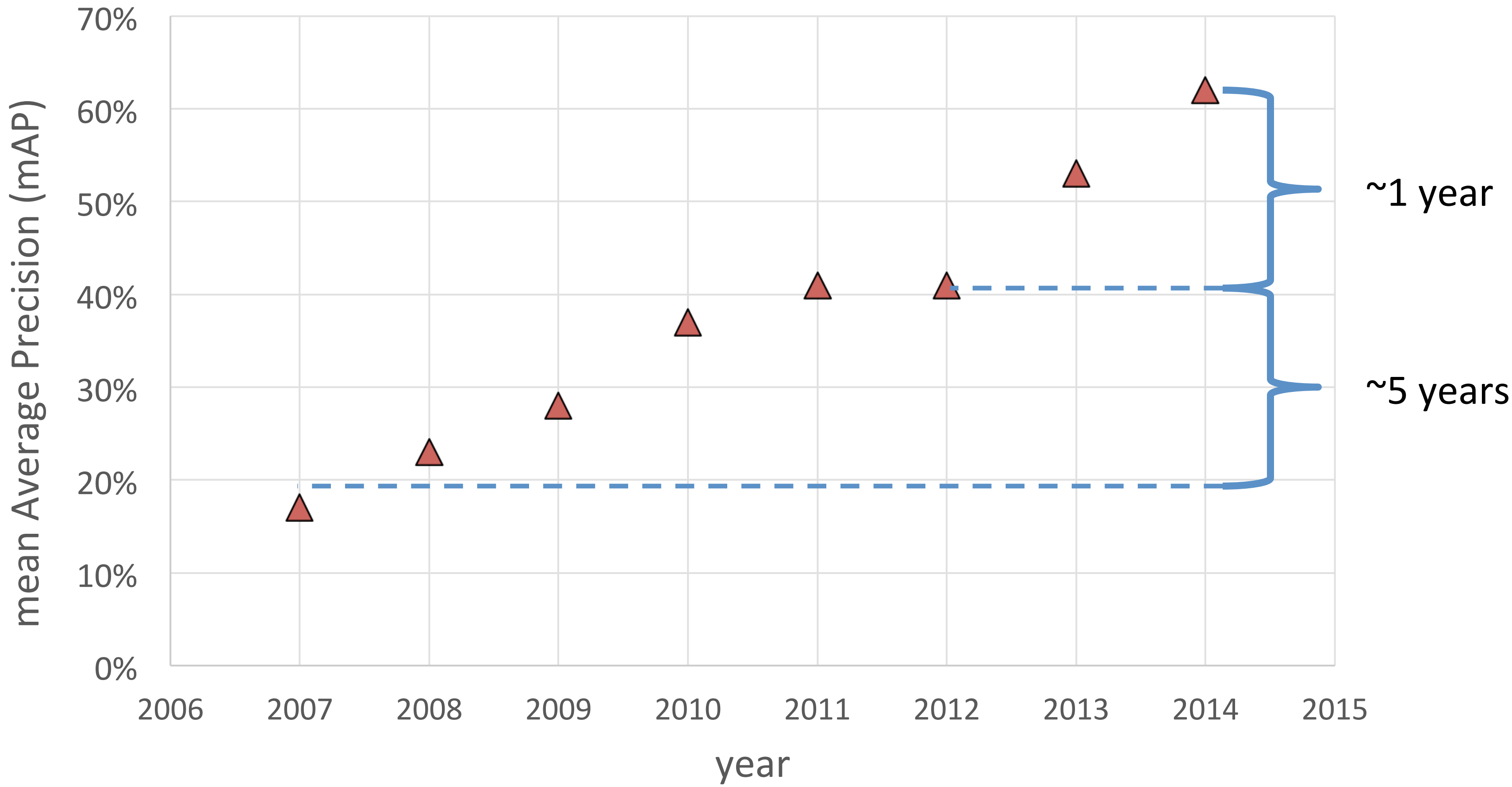


Region-based Convolutional Networks (R-CNNs)



[R-CNN. Girshick et al. CVPR 2014]

Region-based Convolutional Networks (R-CNNs)



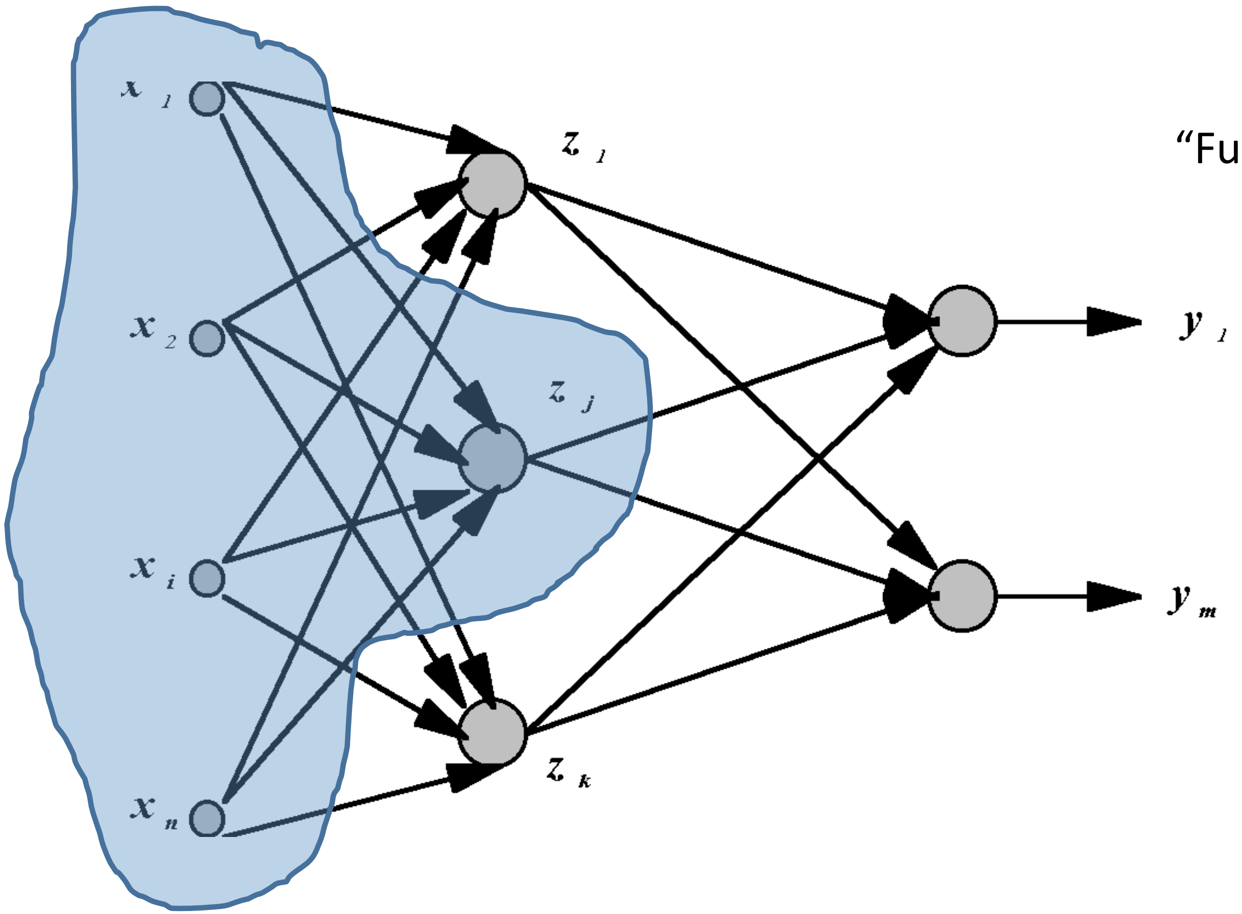
[R-CNN. Girshick et al. CVPR 2014]

Convolutional Neural Networks

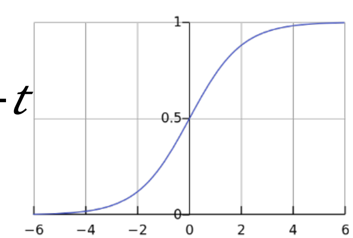
- Overview

Standard Neural Networks

"Fully connected"



$\boldsymbol{x} = (x_1, \ldots, x_{784})^T \quad z_j = g(\boldsymbol{w}_j^T \boldsymbol{x}) \quad g(t) = \frac{1}{1+e^{-t}}$

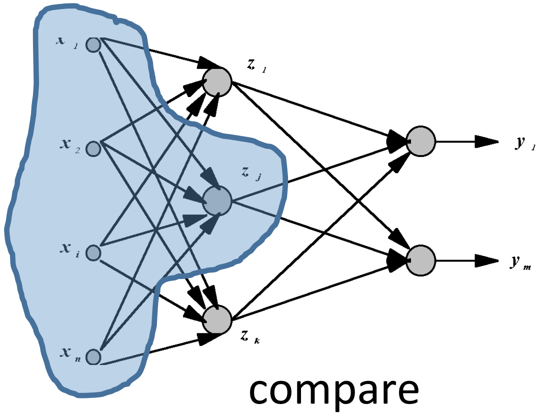


From NNs to Convolutional NNs

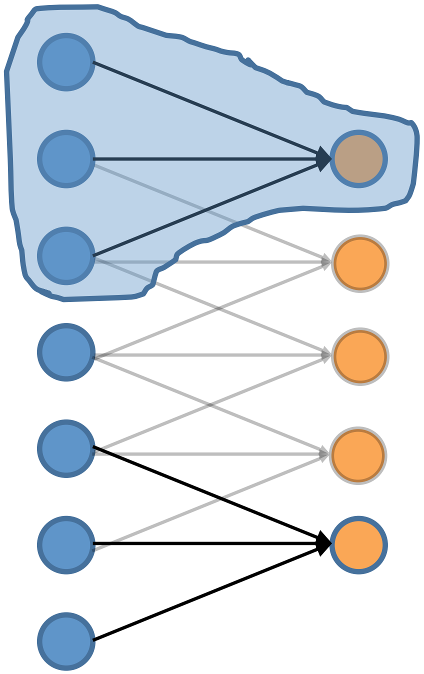
- Local connectivity
- Shared (“tied”) weights
- Multiple feature maps
- Pooling

Convolutional NNs

- Local connectivity



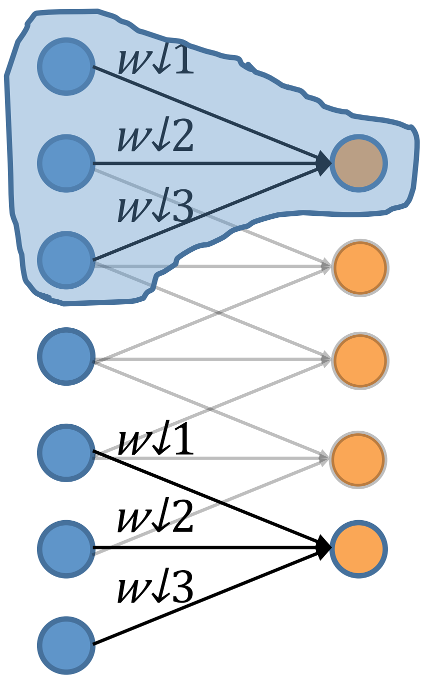
compare



- Each orange unit is only connected to (3) **neighboring** blue units

Convolutional NNs

- Shared ("tied") weights

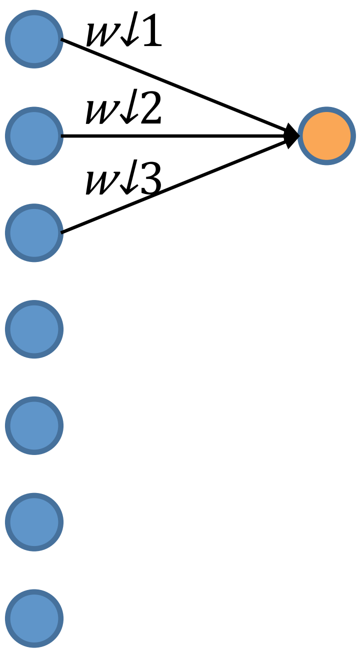


All orange units share the same parameters

Each orange unit computes the same function but with a different input window

Convolutional NNs

- Convolution with 1-D filter: $[w_3, w_2, w_1]$

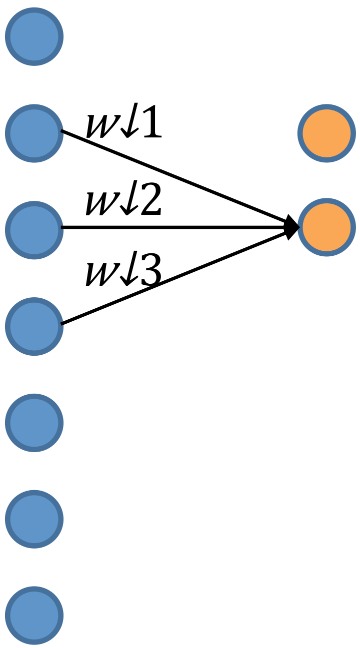


All orange units share the same parameters

Each orange unit computes the same function but with a different input window

Convolutional NNs

- Convolution with 1-D filter: $[w_3, w_2, w_1]$

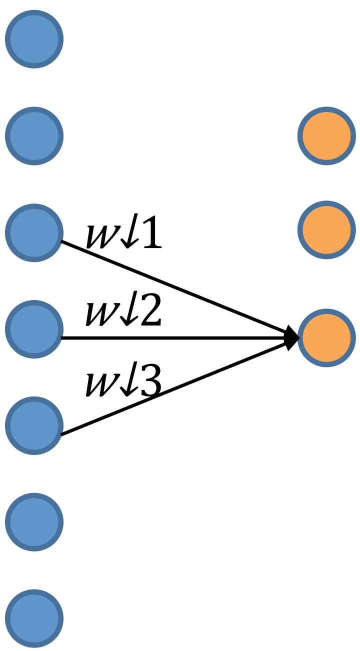


All orange units share the same parameters

Each orange unit computes the same function but with a different input window

Convolutional NNs

- Convolution with 1-D filter: $[w_3, w_2, w_1]$

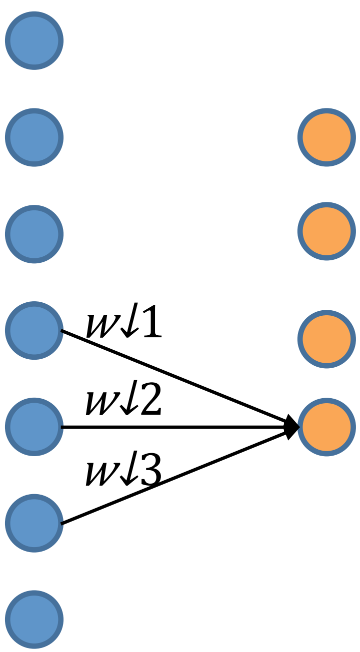


All orange units share the same parameters

Each orange unit computes the same function but with a different input window

Convolutional NNs

- Convolution with 1-D filter: $[w_3, w_2, w_1]$

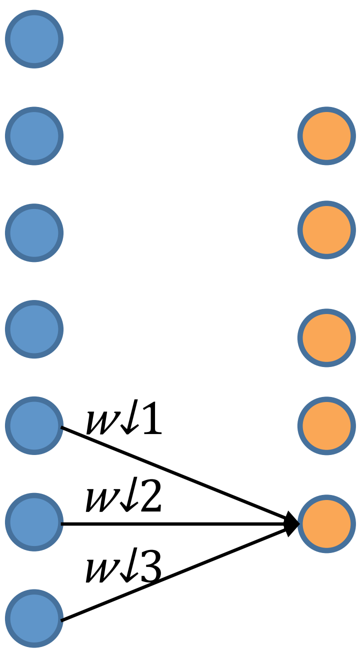


All orange units share the same parameters

Each orange unit computes the same function but with a different input window

Convolutional NNs

- Convolution with 1-D filter: $[w_3, w_2, w_1]$

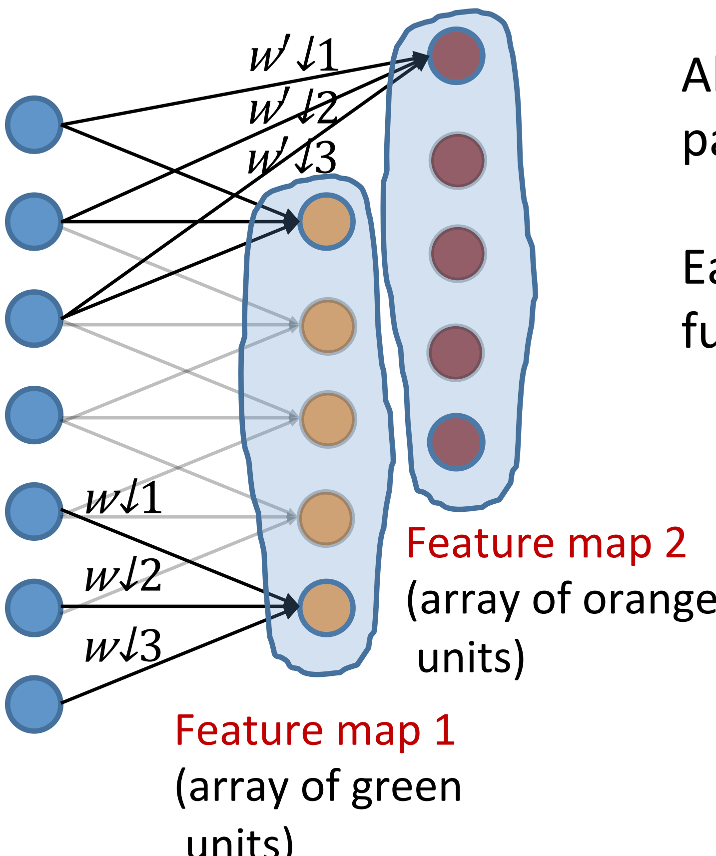


All orange units share the same parameters

Each orange unit computes the same function but with a different input window

Convolutional NNs

- Multiple feature maps

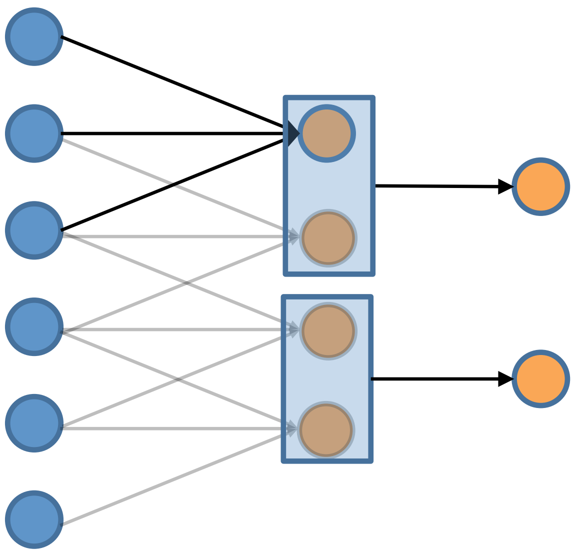


All orange units share the same parameters

Each orange unit computes the same function but with a different input window

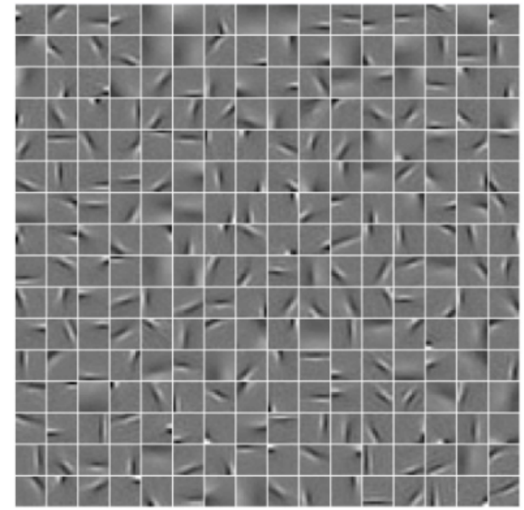
Convolutional NNs

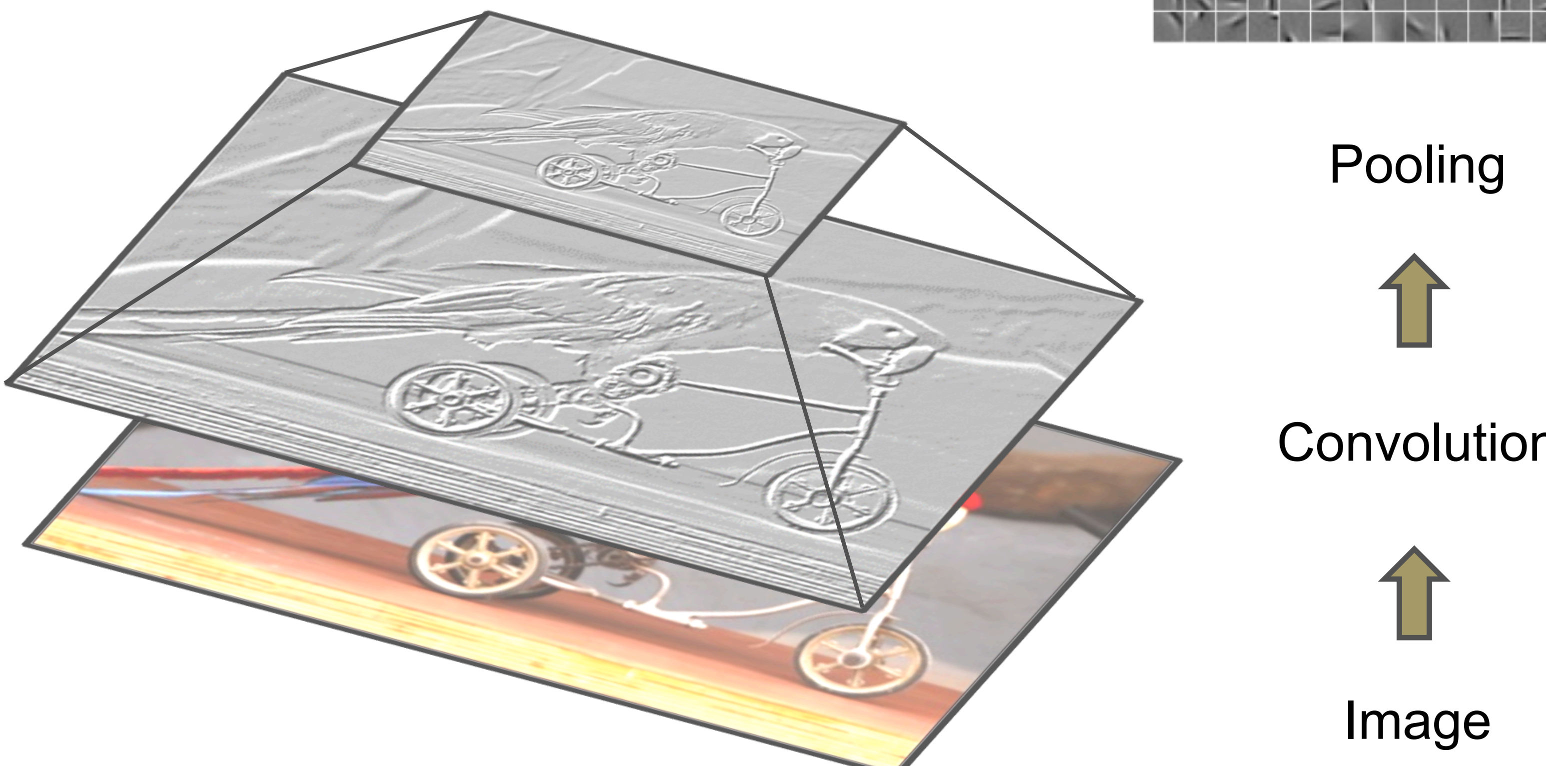
- Pooling (max, average)



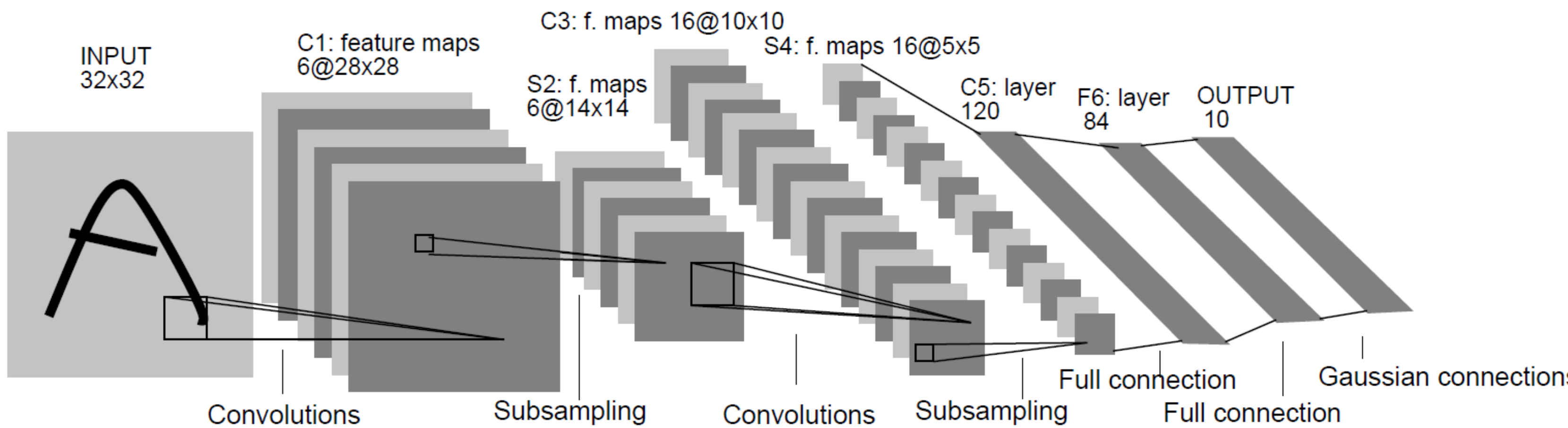
- Pooling area: 2 units
- Pooling stride: 2 units
- **Subsamples** feature maps

2D input





Practical ConvNets



Gradient-Based Learning Applied to Document Recognition,
Lecun et al., 1998

Demo

- http://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html
- ConvNetJS by Andrej Karpathy (Ph.D. student at Stanford)

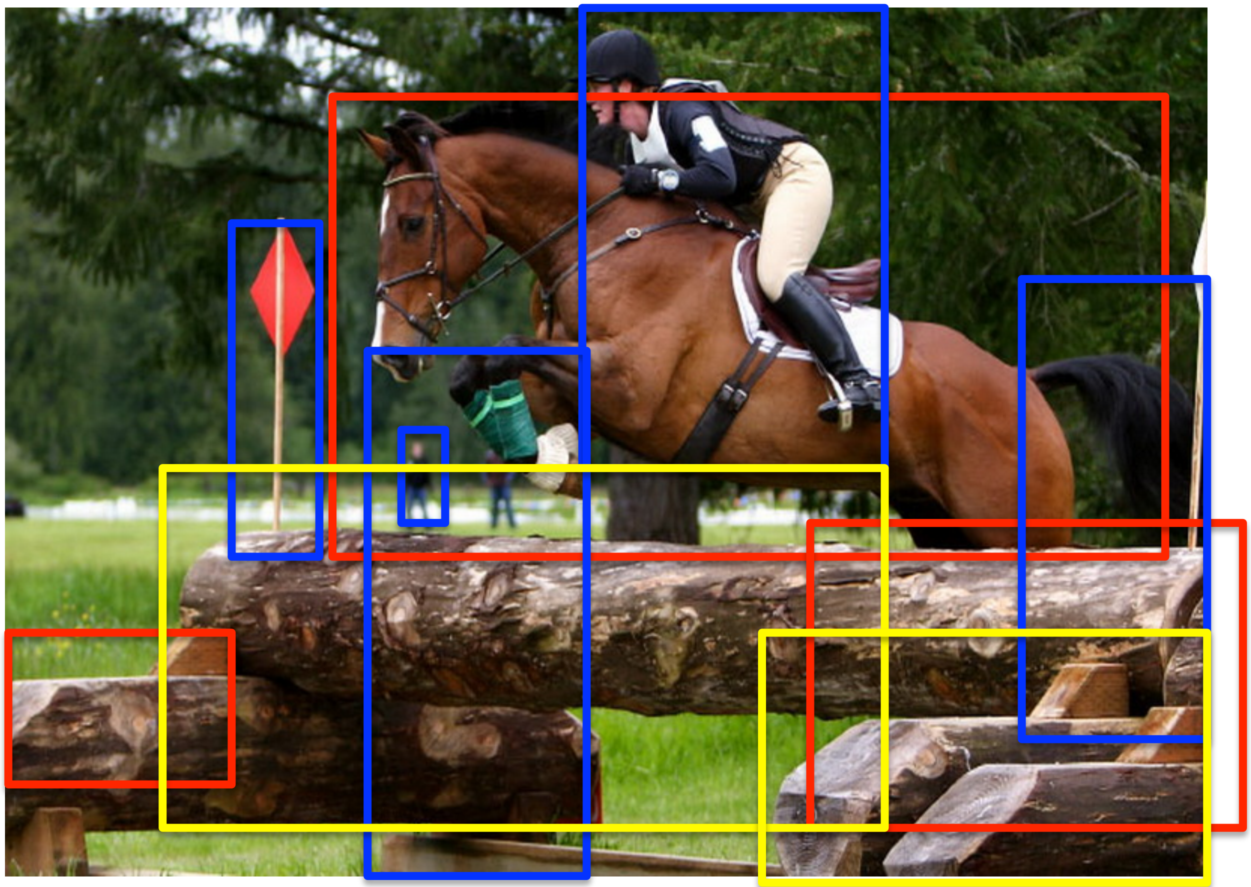
Software libraries

- Caffe (C++, python, matlab)
- Torch7 (C++, lua)
- Theano (python)

Core idea of “deep learning”

- Input: the “*raw*” signal (image, waveform, …)
- Features: hierarchy of features is *learned* from the raw input

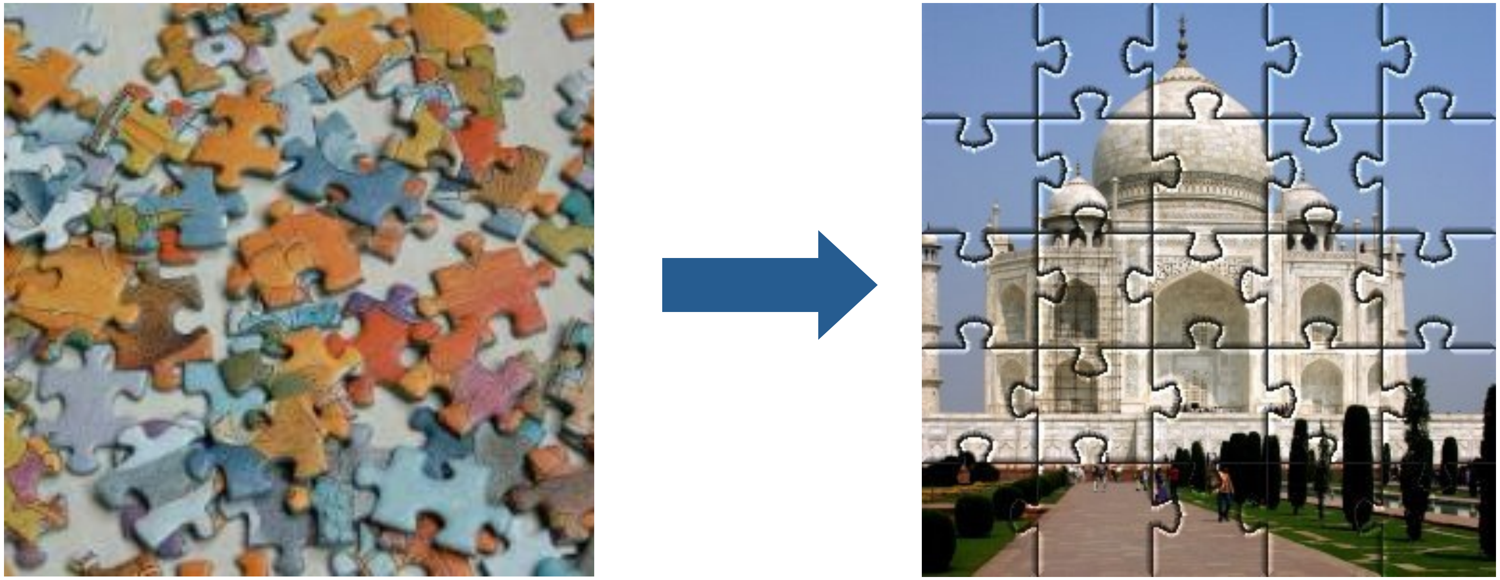
Structure



Recognition using Visual Phrases , CVPR 2011

Structured Prediction

- Prediction of complex outputs
  - Structured outputs: multivariate, correlated, constrained

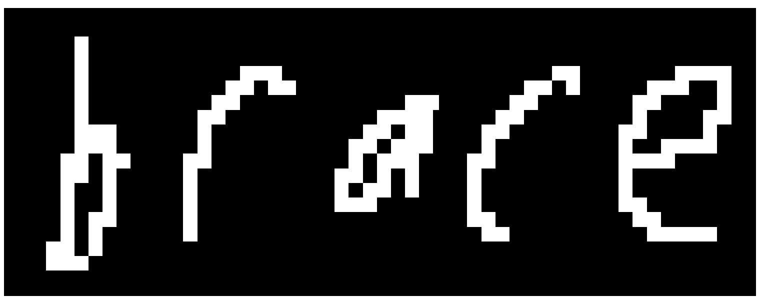


- Novel, general way to solve many learning problems

Handwriting Recognition

x

y

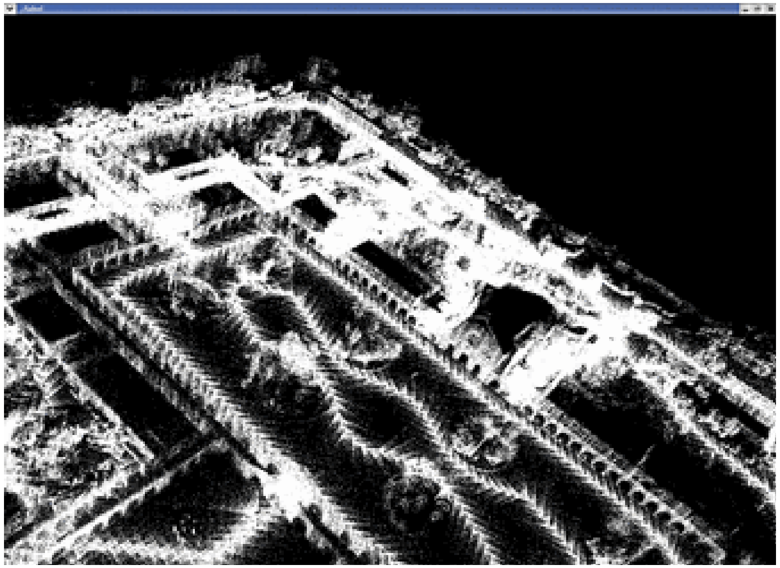


brace

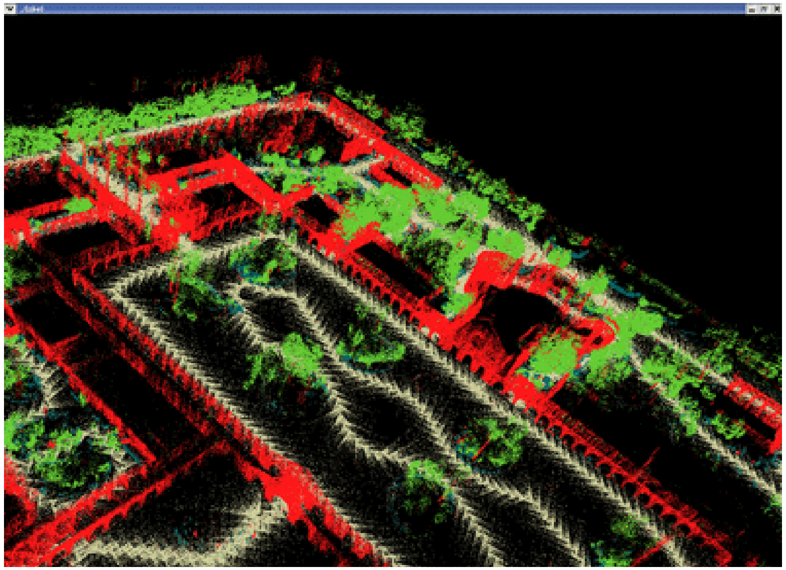
Sequential structure

Object Segmentation

x

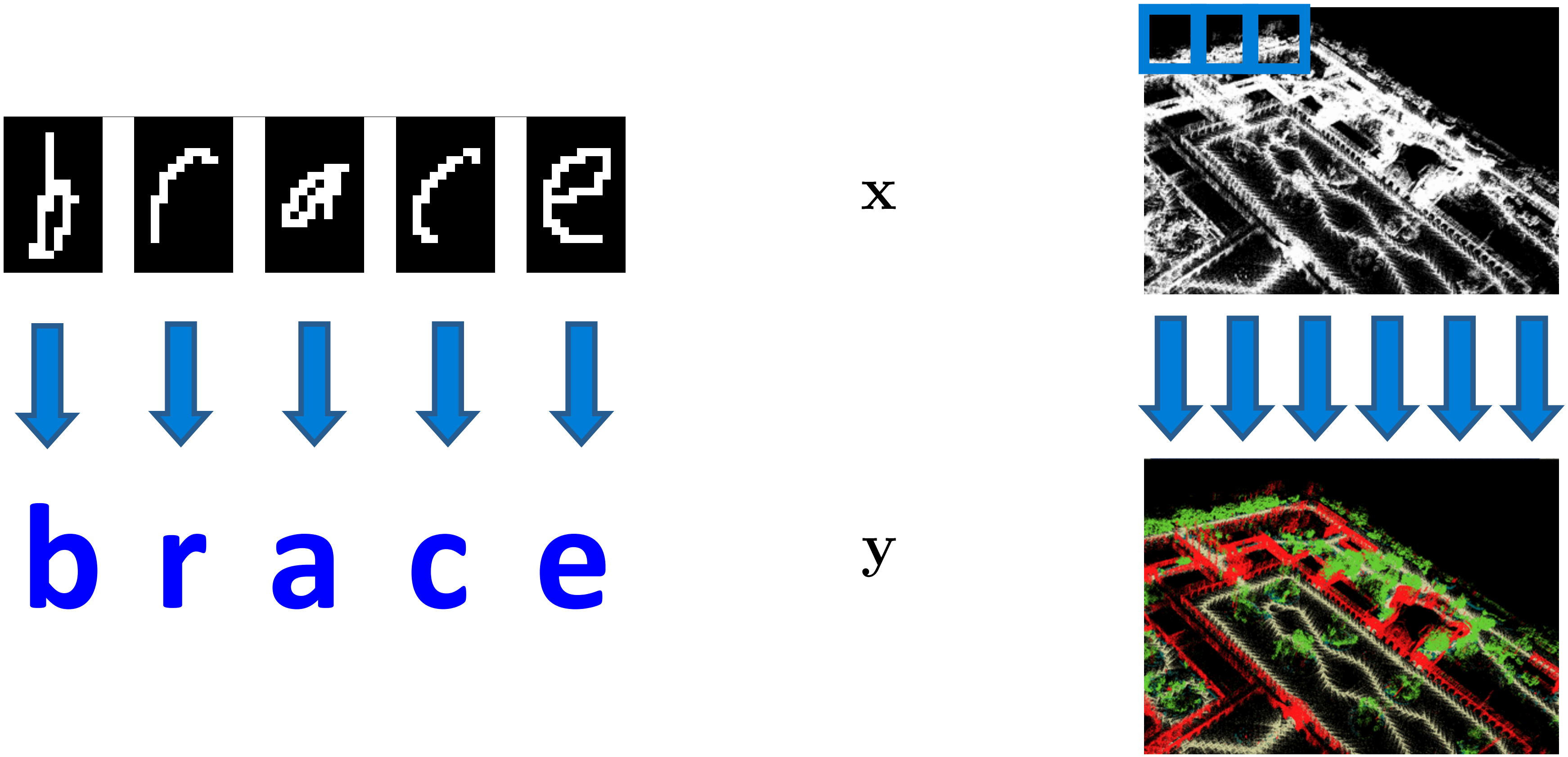


y



Spatial structure

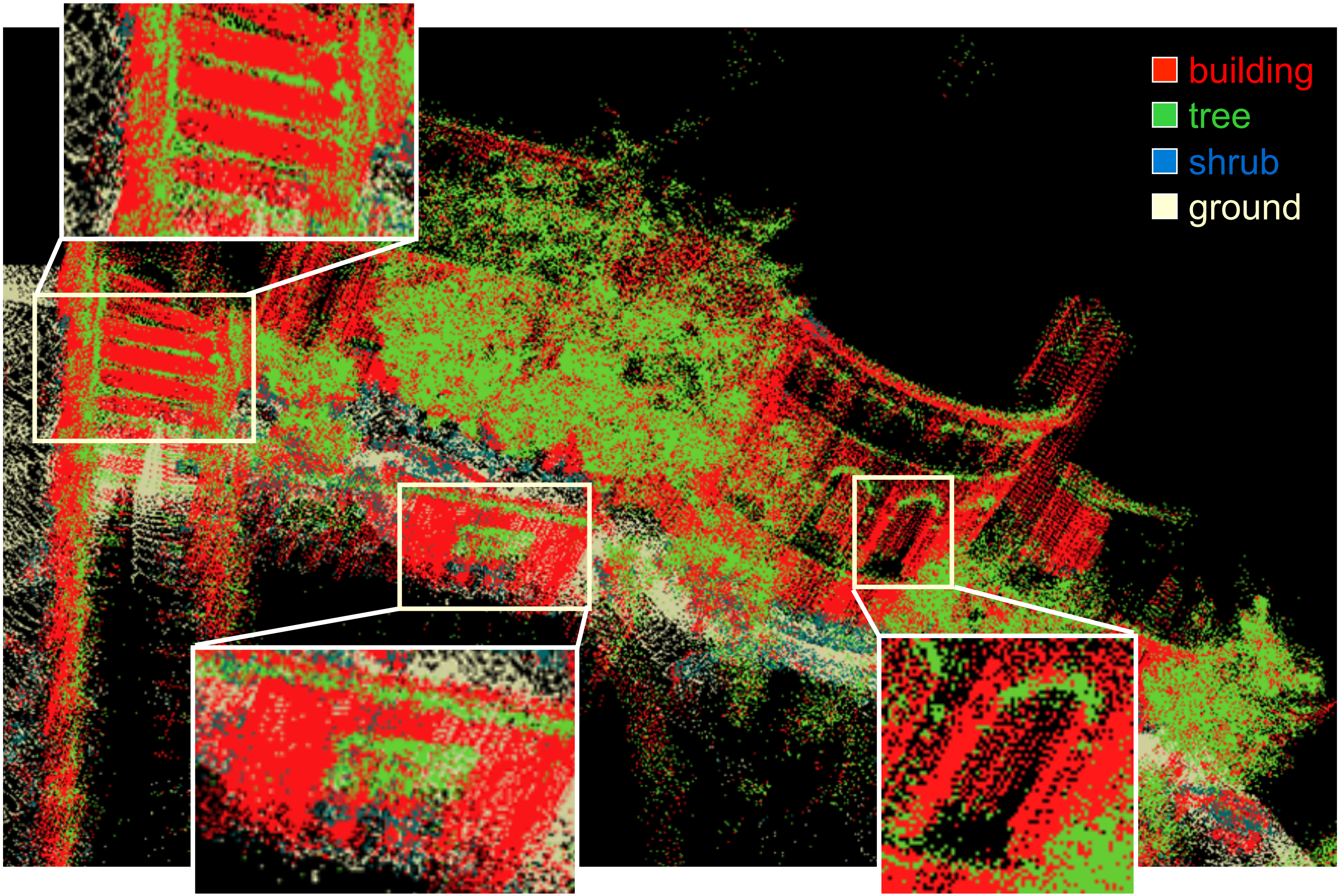
Local Prediction



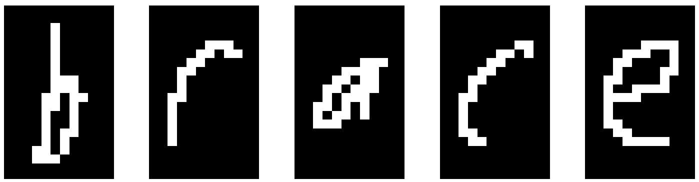
Classify using local information

$\Rightarrow$ Ignores correlations & constraints!

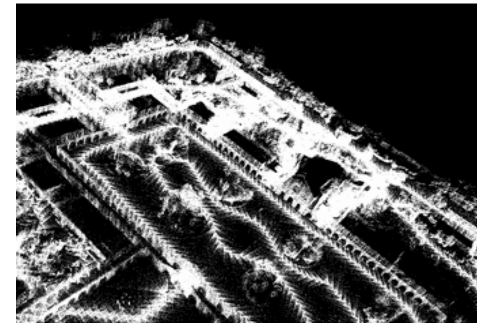
Local Prediction



Structured Prediction

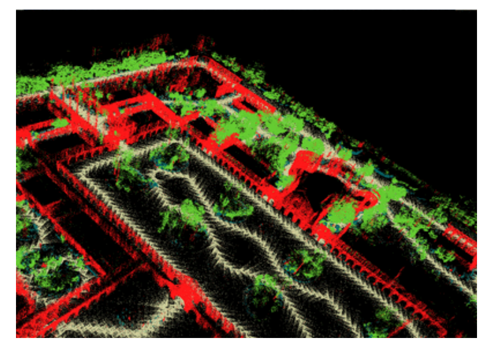


$\mathbf{x}$



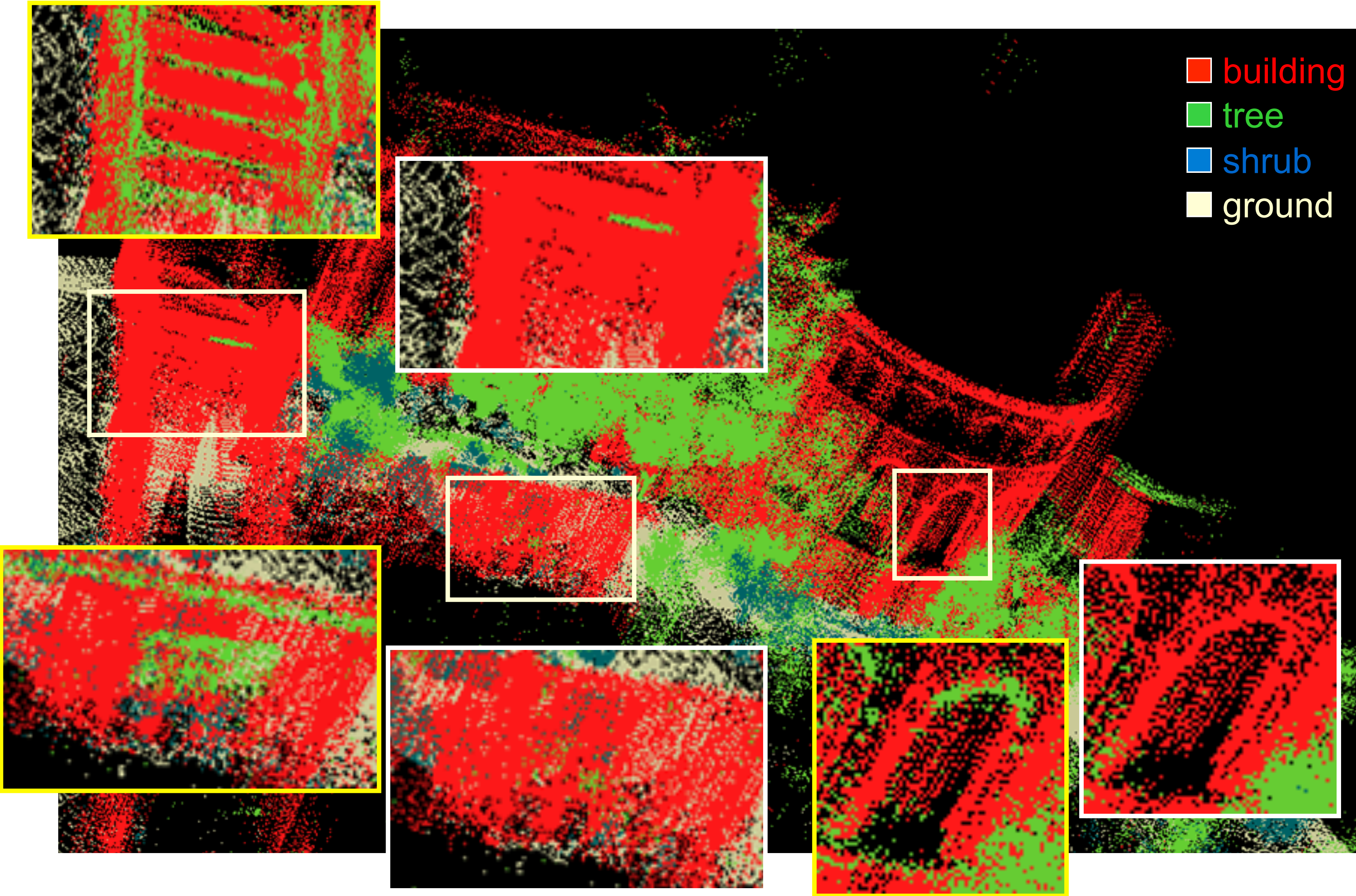
b r a c e

$\mathbf{y}$



- Use local information
- Exploit correlations

Structured Prediction



Structured Models

$$h(\mathbf{x}) = \underset{\mathbf{y} \in \mathcal{Y}(\mathbf{x})}{\arg\max} \; score(\mathbf{x}, \mathbf{y})$$

scoring function

space of feasible outputs

Mild assumptions:

$$score(\mathbf{x}, \mathbf{y}) = \mathbf{w}^\top \mathbf{f}(\mathbf{x}, \mathbf{y}) = \sum_p \mathbf{w}^\top \mathbf{f}(\mathbf{x}_p, \mathbf{y}_p)$$

linear combination

sum of part scores

Supervised Structured Prediction

Model: $$P_{\mathbf{w}}(\mathbf{y} \mid \mathbf{x}) \propto \exp\{\mathbf{w}^{\top}\mathbf{f}(\mathbf{x},\mathbf{y})\}$$

Data

$$(\mathbf{x}^1, \mathbf{y}^1)$$
$$\vdots$$
$$(\mathbf{x}^n, \mathbf{y}^n)$$

Learning

Estimate w

Local
(ignores structure)

Likelihood
(can be intractable)

Margin

Prediction

$$\arg\max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} P_{\mathbf{w}}(\mathbf{y} \mid \mathbf{x})$$

Example:
Weighted matching

Generally:
Combinatorial optimization

Local Estimation

Model: $$\mathsf{P}_{\mathbf{w}}(\mathbf{y} \mid \mathbf{x}) \propto \prod_{jk} \mathsf{exp}\{\mathbf{w}^\top \mathbf{f}(y_{jk}, \mathbf{x})\}$$

Data

$$\boxed{\begin{array}{c}(\mathbf{x}^1, \mathbf{y}^1) \\ \vdots \\ (\mathbf{x}^n, \mathbf{y}^n)\end{array}}$$

- Treat edges as independent decisions
- Estimate **w** locally, use globally
  - E.g., naïve Bayes, SVM, logistic regression
  - Cf. [Matusov+al, 03] for matchings

  - Simple and cheap
  - Not well-calibrated for matching model
  - Ignores correlations & constraints

Conditional Likelihood Estimation

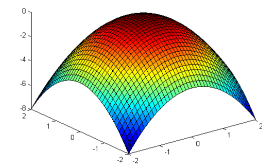
Model: $$P_{\mathbf{w}}(\mathbf{y}\,|\,\mathbf{x}) = \frac{\prod_{jk} \exp\{\mathbf{w}^\top \mathbf{f}(y_{jk}, \mathbf{x})\}}{\sum_{\mathbf{y}' \in \mathcal{Y}(\mathbf{x})} \prod_{jk} \exp\{\mathbf{w}^\top \mathbf{f}(y'_{jk}, \mathbf{x})\}}$$

Data

$$(\mathbf{x}^1, \mathbf{y}^1)$$
$$\vdots$$
$$(\mathbf{x}^n, \mathbf{y}^n)$$

- Estimate **w** jointly:

$$\sum_i \log P_{\mathbf{w}}(\mathbf{y}^i\,|\,\mathbf{x}^i)$$



$$\mathbf{w} \in \mathcal{W}$$

- Denominator is #P-complete
  [Valiant 79, Jerrum & Sinclair 93]
- Tractable model, intractable learning
- Need tractable learning method
  $\Longrightarrow$ margin-based estimation

Structured large margin estimation

- We want:

$\arg\max_{\mathbf{y}}\ \mathbf{w}^\top \mathbf{f}($  $, \mathbf{y}) \quad = \quad$ "brace"

- Equivalently:

$\mathbf{w}^\top \mathbf{f}($  $,$ "brace"$) \quad > \quad \mathbf{w}^\top \mathbf{f}($  $,$ "aaaaa"$)$

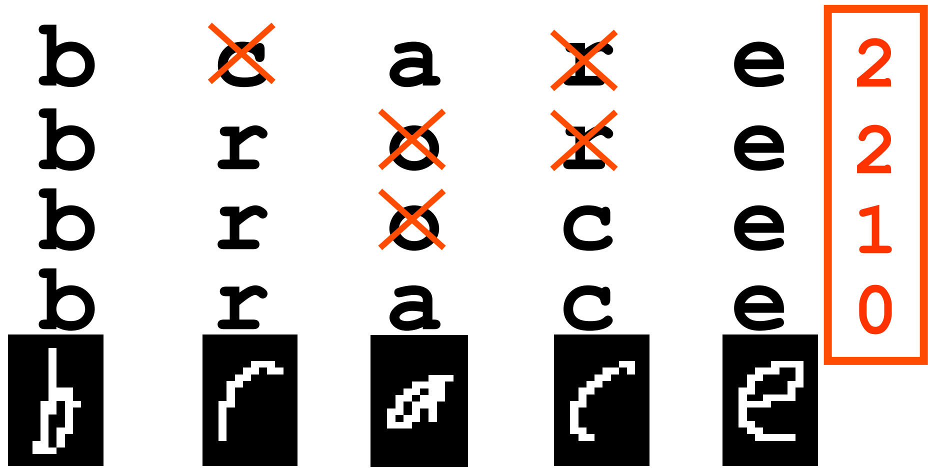
$\mathbf{w}^\top \mathbf{f}($  $,$ "brace"$) \quad > \quad \mathbf{w}^\top \mathbf{f}($  $,$ "aaaab"$)$

…

$\mathbf{w}^\top \mathbf{f}($  $,$ "brace"$) \quad > \quad \mathbf{w}^\top \mathbf{f}($  $,$ "zzzzz"$)$

a lot!

Structured Loss



Large margin estimation

- Given training examples $(\mathbf{x}^i, \mathbf{y}^i)$, we want:

$$\mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) > \mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}) \quad \forall \mathbf{y} \neq \mathbf{y}^i$$

- Maximize margin γ

$$\boxed{\mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) \geq \mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}) + \gamma \ell(\mathbf{y}^i, \mathbf{y}) \quad \forall \mathbf{y}}$$

- Mistake weighted margin: $\gamma \ell(\mathbf{y}^i, \mathbf{y})$

$\ell(\mathbf{y}^i, \mathbf{y}) = \sum_p I(y_p^i \neq y_p)$ # of mistakes in **y**

*Collins 02, Altun et al 03, Taskar 03

Large margin estimation

$$\max_{||\mathbf{w}||\leq 1} \gamma$$

$$\mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) \geq \mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}) + \gamma \ell(\mathbf{y}^i, \mathbf{y}), \quad \forall i, \mathbf{y}$$

- Eliminate γ

$$\min_{\mathbf{w}} \quad \frac{1}{2}||\mathbf{w}||^2$$

$$\mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) \geq \mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}) + \ell(\mathbf{y}^i, \mathbf{y}), \quad \forall i, \mathbf{y}$$

- Add slacks ξ_i for inseparable case (hinge loss)

$$\min_{\mathbf{w},\xi} \quad \frac{1}{2}||\mathbf{w}||^2 + C \sum_i \xi_i$$

$$\mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) + \xi_i \geq \mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}) + \ell(\mathbf{y}^i, \mathbf{y}), \quad \forall i, \mathbf{y}$$

Large margin estimation

- Brute force enumeration

$$\min \quad \frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i$$

$$\mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) + \xi_i \geq \mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}) + \ell(\mathbf{y}^i, \mathbf{y}), \quad \forall i, \mathbf{y}$$

- Min-max formulation

$$\min \quad \frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i$$

$$\mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) + \xi_i \geq \max_{\mathbf{y}} [\mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}) + \ell(\mathbf{y}^i, \mathbf{y})], \quad \forall i$$

  - 'Plug-in' linear program for inference

$$\max_{\mathbf{y}} [\mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}) + \ell(\mathbf{y}^i, \mathbf{y})]$$

Min-max formulation

$$\max_{\mathbf{y}} \, [\mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}) + \ell(\mathbf{y}^i, \mathbf{y})]$$

Structured loss (Hamming): $\ell(\mathbf{y}^i, \mathbf{y}) = \sum_p \ell_p(\mathbf{y}^i_p, \mathbf{y}_p)$

Inference $\max_{\mathbf{y}} \left[\sum_p \mathbf{w}^\top \mathbf{f}(\mathbf{x}^i_p, \mathbf{y}_p) + \ell_p(\mathbf{y}^i_p, \mathbf{y}_p) \right]$

LP Inference $\max_{\substack{\mathbf{z} \geq 0; \\ \mathbf{A}\mathbf{z} \leq \mathbf{b};}} \mathbf{q}^\top \mathbf{z}$

Key step: $\max_{\mathbf{y}} \Longleftrightarrow \max_{\mathbf{z}}$

discrete optim. continuous optim.

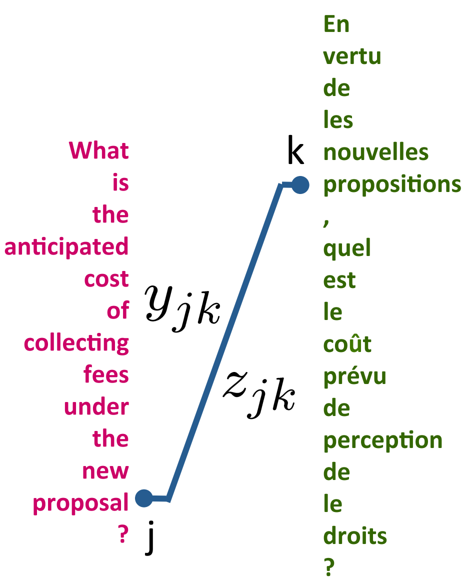
Matching Inference LP

$$\max_{\mathbf{y}} \quad \mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}) + \ell(\mathbf{y}^i, \mathbf{y})$$

Need Hamming-like loss

$$\underbrace{\max_{\mathbf{z}} \quad \sum_{jk} z_{jk} \left[\mathbf{w}^\top \mathbf{f}(\mathbf{x}^i_{jk}) + \ell^i_{jk}\right]}_{\mathbf{q}^\top \mathbf{z}}$$

$$\mathbf{q} = \mathbf{F}^\top \mathbf{w} + \ell$$



s.t. $\quad z_{jk} \geq 0$

degree

$$\left.\begin{aligned} \sum_k z_{jk} &\leq 1 \\ \sum_j z_{jk} &\leq 1 \end{aligned}\right\} A\mathbf{z} \leq \mathbf{b}$$

LP Duality

- Linear programming duality
  - Variables $\Rightarrow$ constraints
  - Constraints $\Rightarrow$ variables
- Optimal values are the same
  - When both feasible regions are bounded

$$\boxed{\begin{aligned} \max_{\mathbf{z}} \quad & \mathbf{c}^\top \mathbf{z} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{z} \le \mathbf{b}; \\ & \mathbf{z} \ge 0. \end{aligned}} \quad \Longleftrightarrow \quad \boxed{\begin{aligned} \min_{\lambda} \quad & \mathbf{b}^\top \lambda \\ \text{s.t.} \quad & \mathbf{A}^\top \lambda \ge \mathbf{c}; \\ & \lambda \ge 0. \end{aligned}}$$

Min-max Formulation

$$\min_{\mathbf{w},\xi} \quad \frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i$$

$$\mathbf{w}^\top \mathbf{f}(\mathbf{x}^i,\mathbf{y}^i)+\xi_i \geq \underbrace{\max_{\mathbf{y}}\,[\mathbf{w}^\top\mathbf{f}(\mathbf{x}^i,\mathbf{y})+\ell(\mathbf{y}^i,\mathbf{y})]}_{\displaystyle \max_{\substack{\mathbf{A}_i\mathbf{z}_i\leq\mathbf{b}_i\\ \mathbf{z}_i\geq 0}} \mathbf{q}_i^\top\mathbf{z}_i}, \quad \forall i$$

$$\boxed{\mathbf{q}_i = \mathbf{F}_i^\top\mathbf{w} + \ell_i}$$

$$\max_{\substack{\mathbf{A}_i\mathbf{z}_i\leq\mathbf{b}_i\\ \mathbf{z}_i\geq 0}} \quad \mathbf{q}_i^\top\mathbf{z}_i \quad \Longleftrightarrow \quad \min_{\substack{\mathbf{A}_i^\top\lambda_i\geq\mathbf{q}_i\\ \lambda_i\geq 0}} \quad \mathbf{b}_i^\top\lambda_i$$

LP duality

$$\boxed{\begin{aligned}
&\min_{\mathbf{w},\xi,\lambda} && \frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i\\
&\text{s.t.} && \mathbf{w}^\top\mathbf{f}(\mathbf{x}^i,\mathbf{y}^i)+\xi_i \geq \mathbf{b}_i^\top\lambda_i,\\
& && \mathbf{A}_i^\top\lambda_i \geq \mathbf{q}_i; \quad \lambda_i \geq 0
\end{aligned}}$$

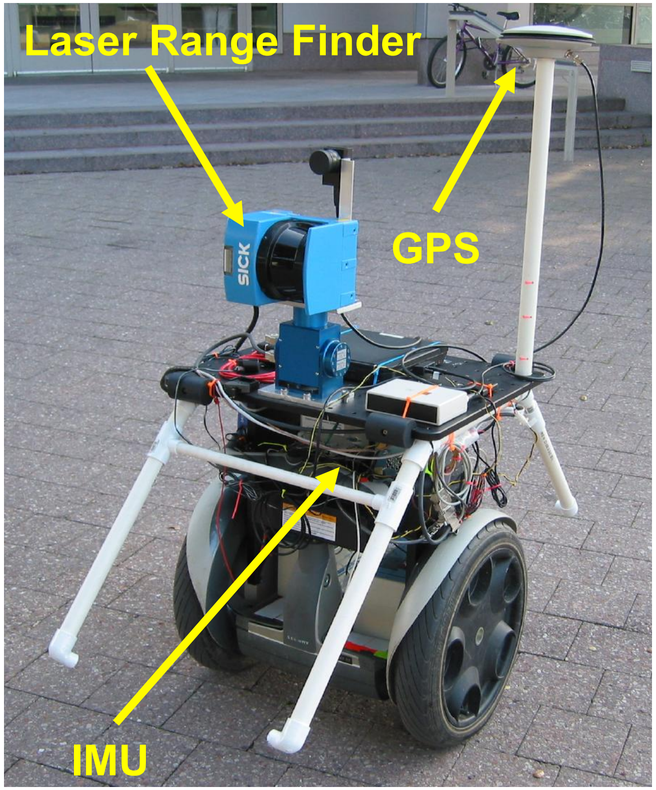
Min-max formulation summary

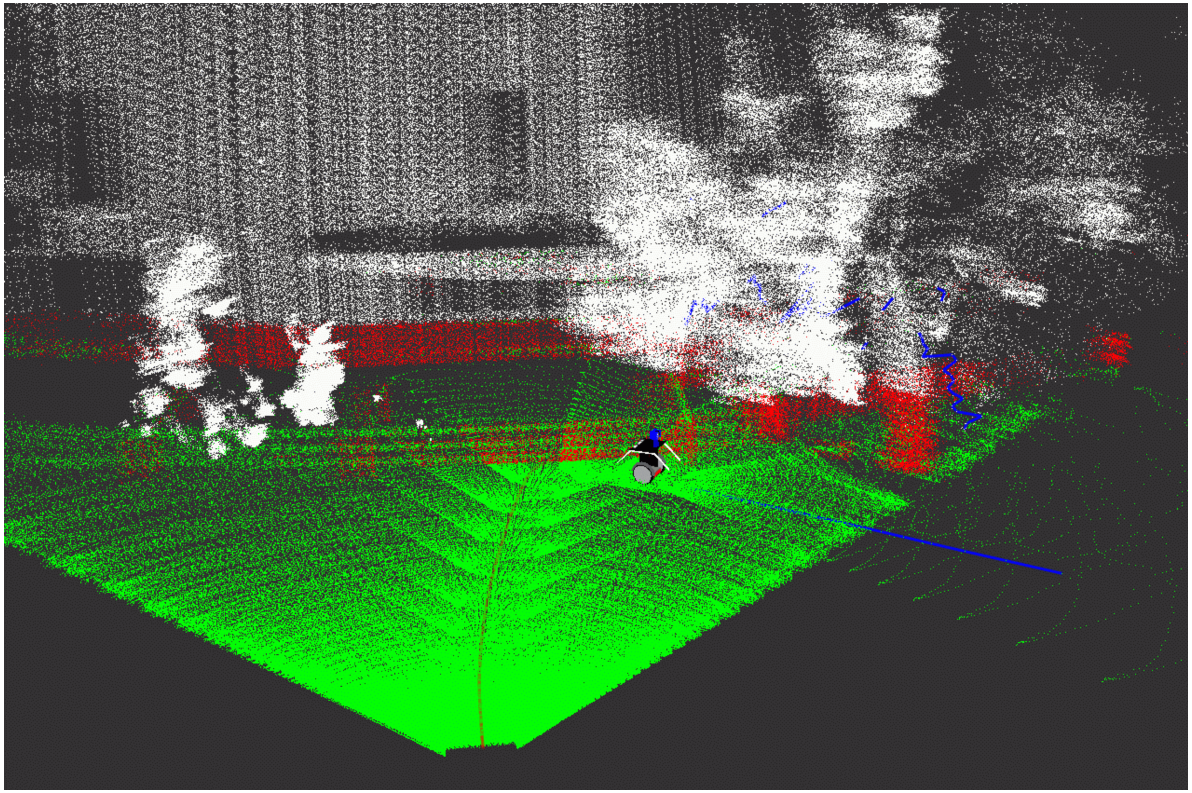
$$\min_{\mathbf{w},\lambda} \quad \frac{1}{2}||\mathbf{w}||^2 + C\left(\sum_i \mathbf{b}_i^\top \lambda_i - \mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i)\right)$$
$$\text{s.t.} \quad \mathbf{A}_i^\top \lambda_i \geq \mathbf{F}_i^\top \mathbf{w} + \ell_i; \qquad \lambda_i \geq 0, \ \forall i.$$

*Taskar et al 04

3D Mapping

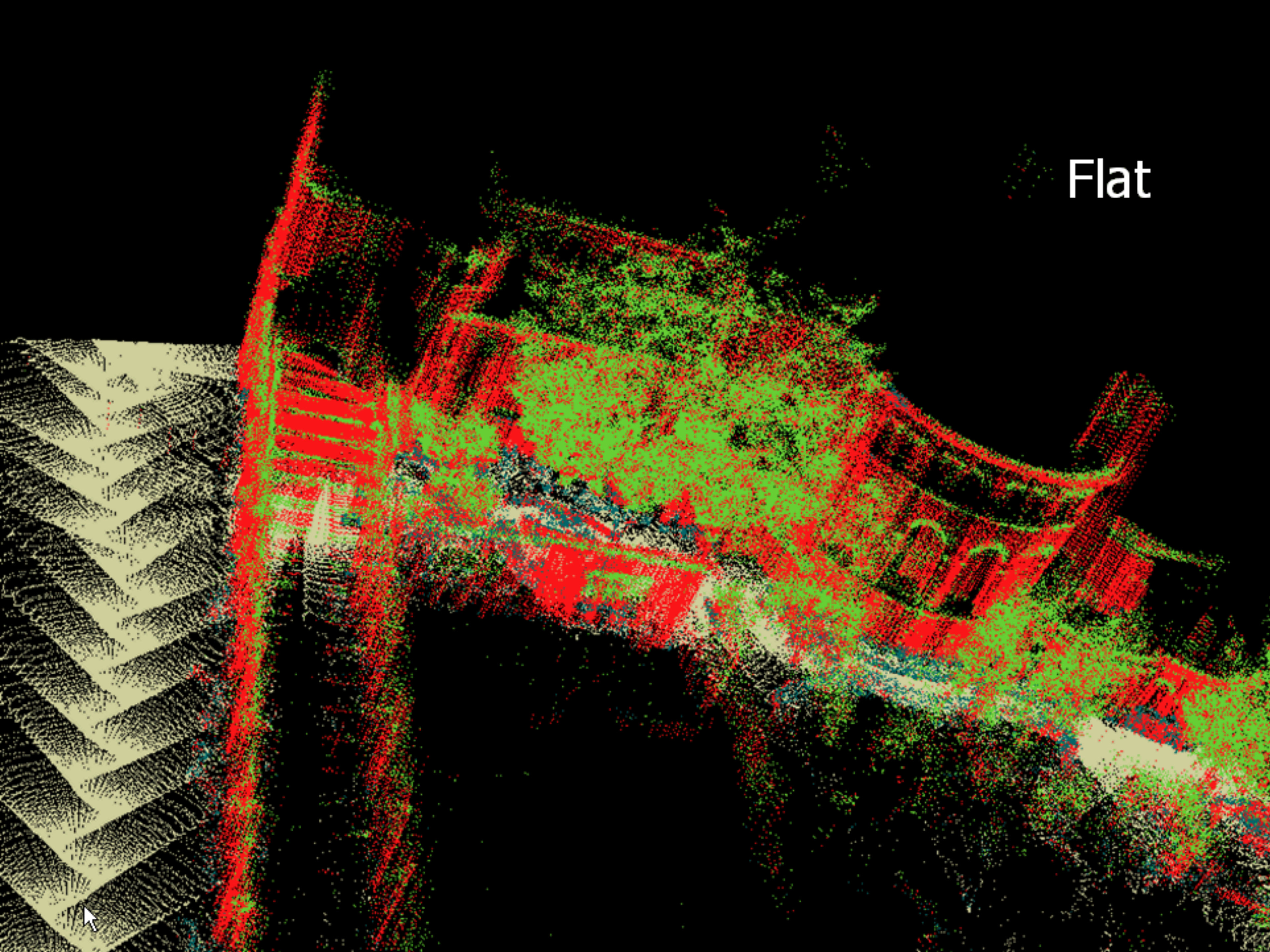
Data provided by: Michael Montemerlo & Sebastian Thrun

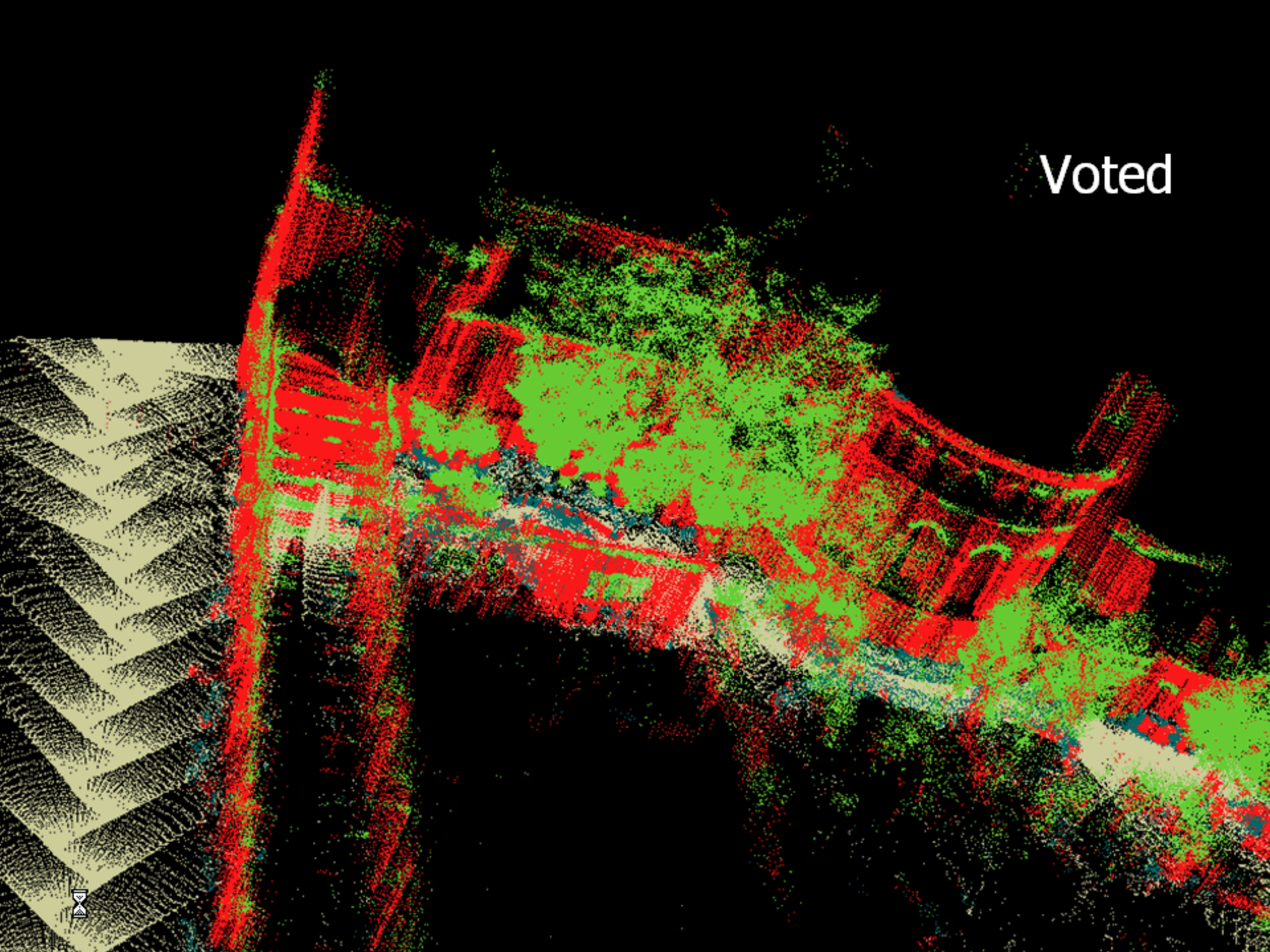


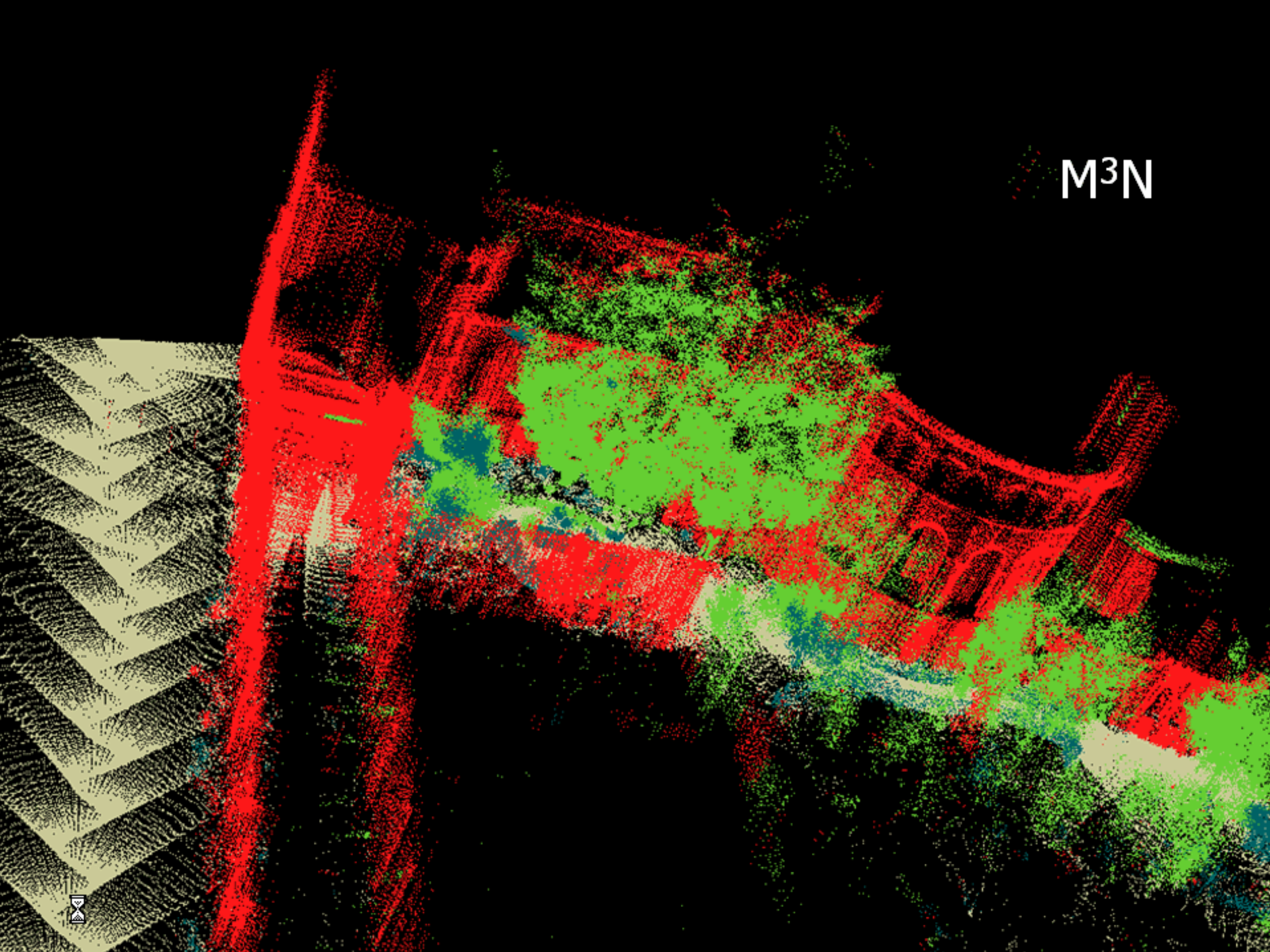


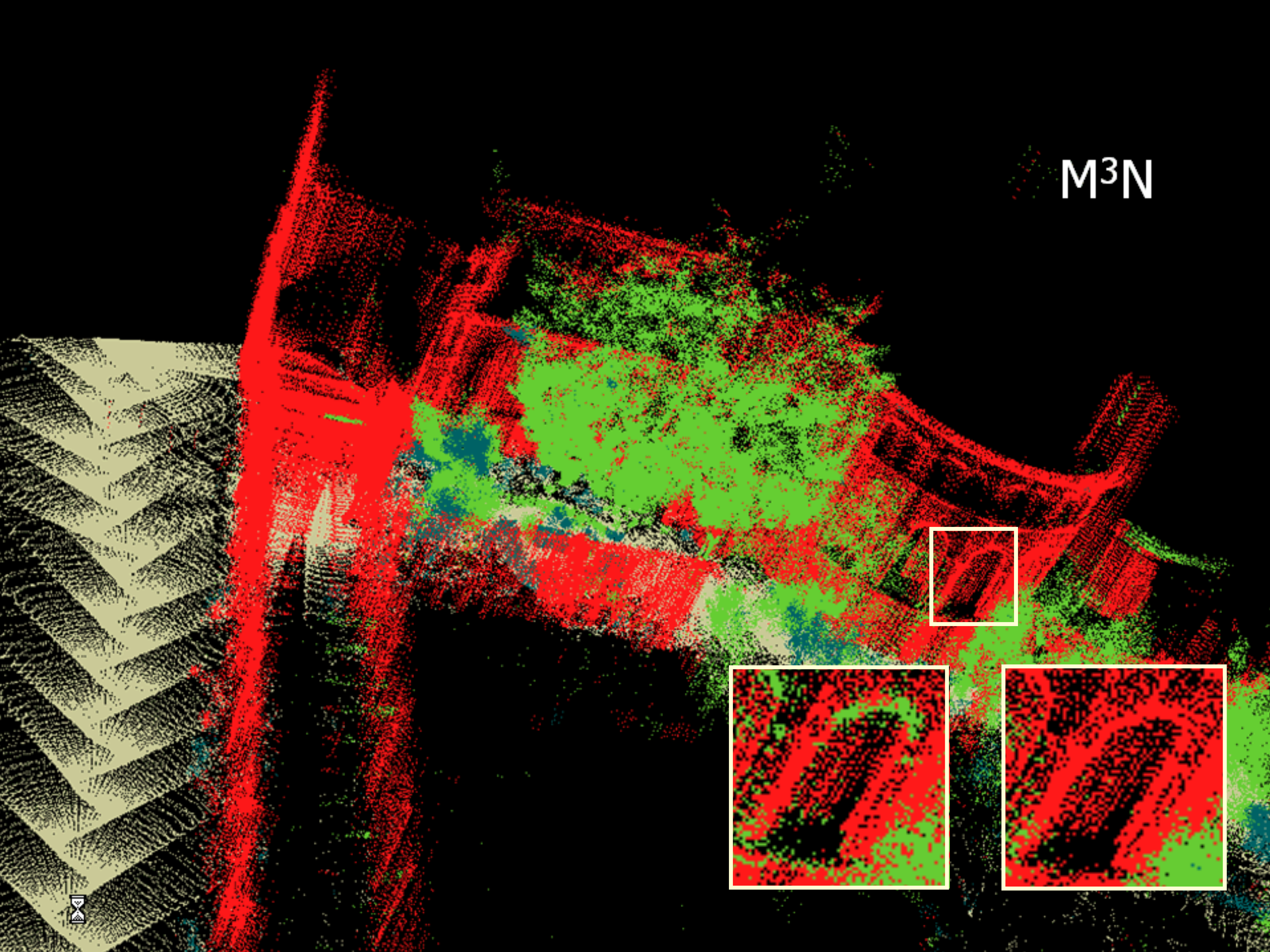
Label: ground, building, tree, shrub

Training: 30 thousand points   Testing: 3 million points


Flat

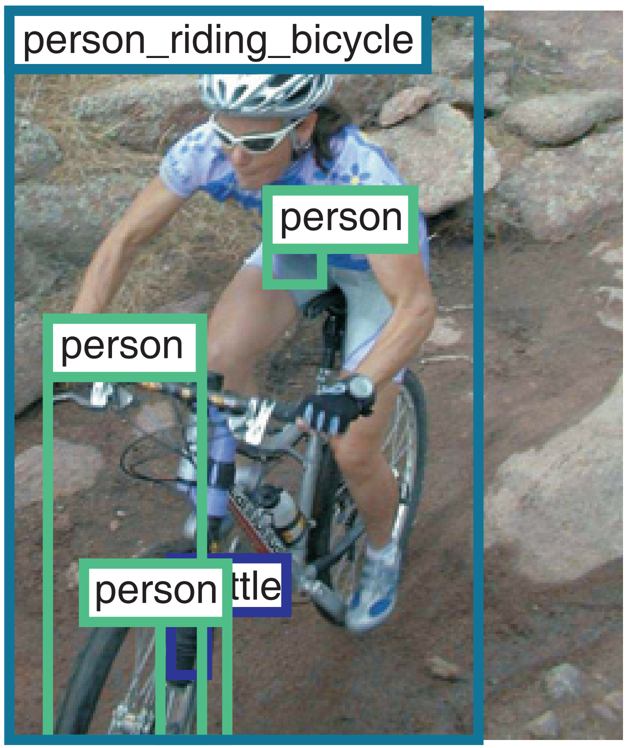

Voted


M³N

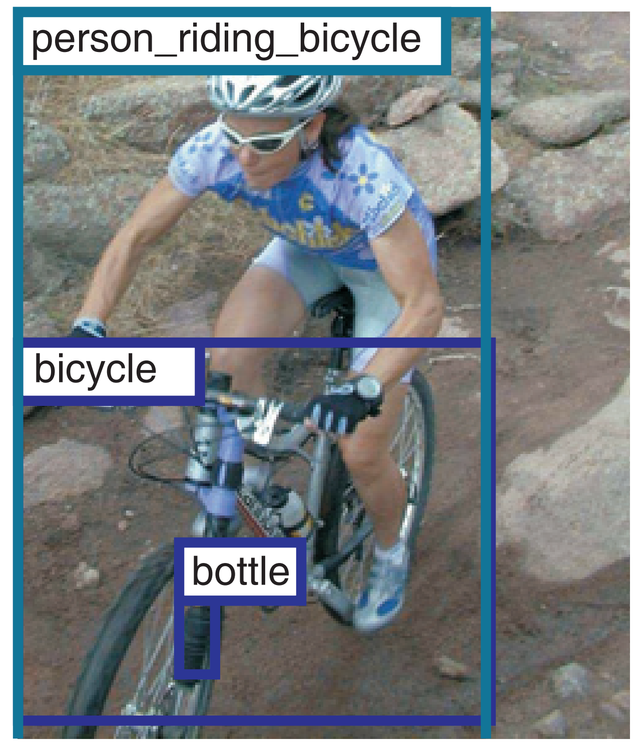

M^{3}N

Before and After

Before Decoding



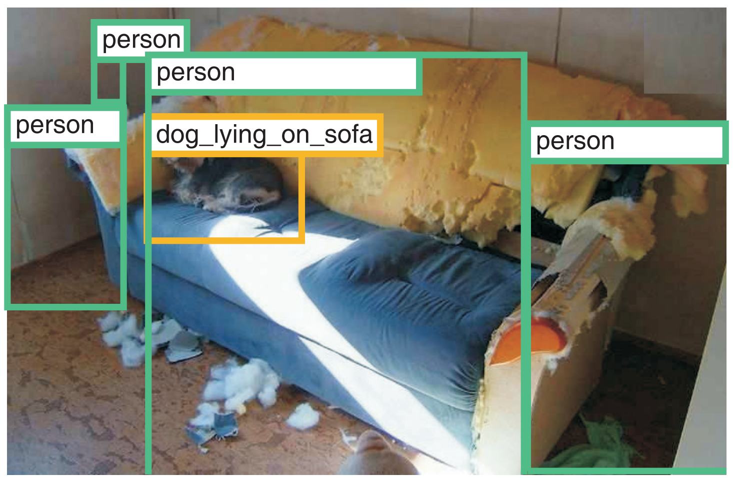
After Decoding



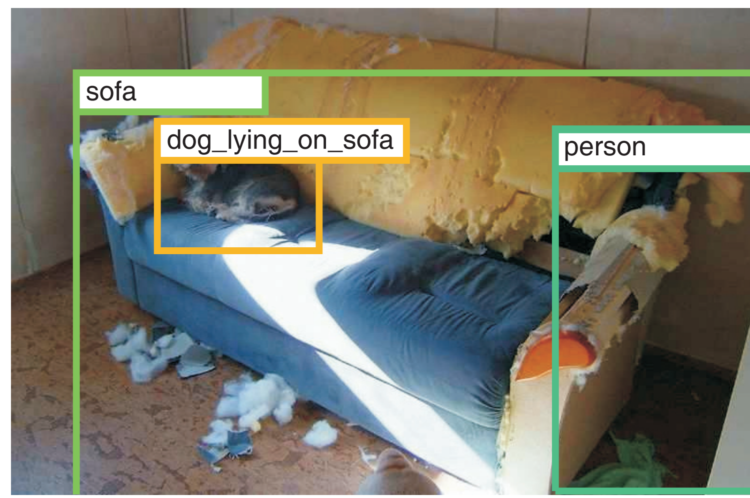
Recognition using Visual Phrases , CVPR 2011

Before and After

Before Decoding



After Decoding



Recognition using Visual Phrases , CVPR 2011