Object Detection

Ali Farhadi
Mohammad Rastegari
CSE 576

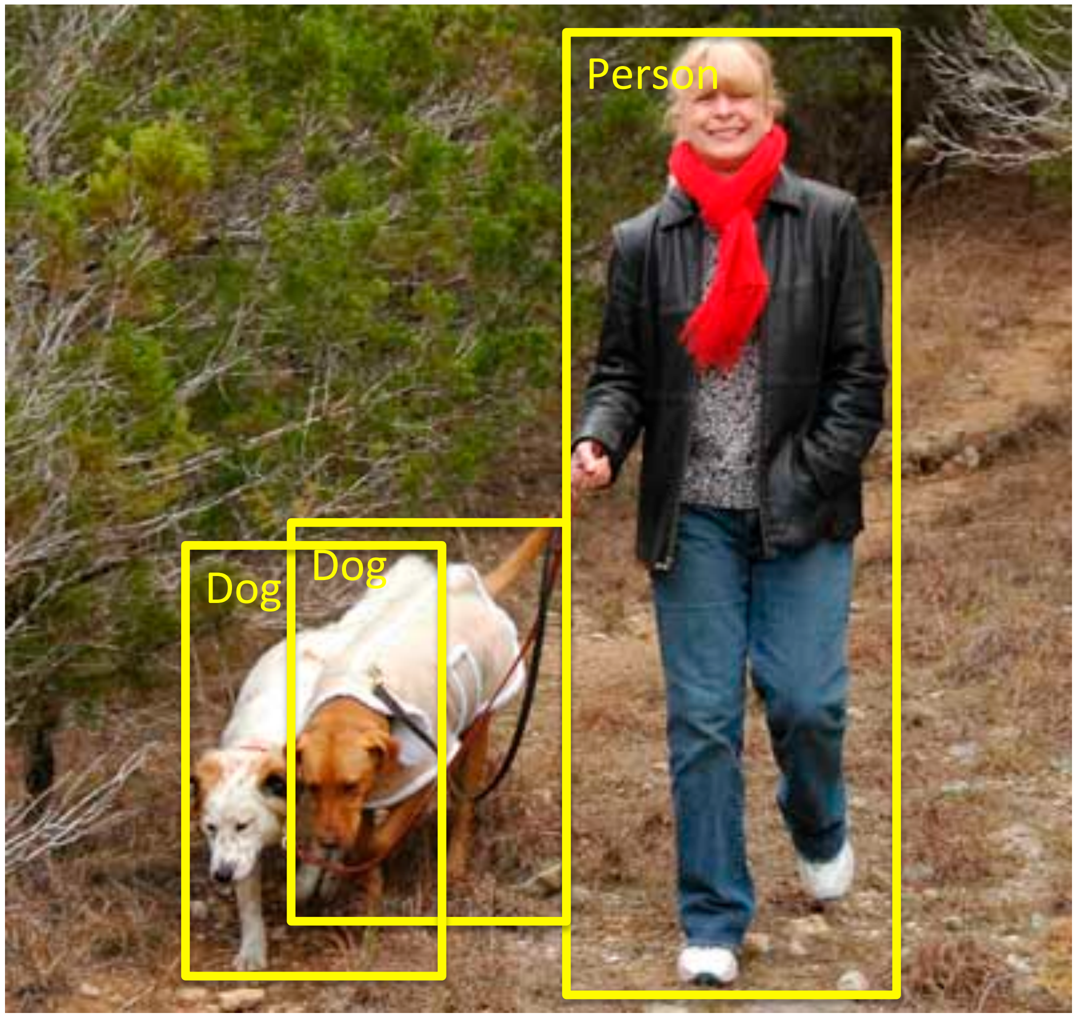
Object Recognition

Person

Dog

Chair

Object Detection



Sliding Window

Sliding Window

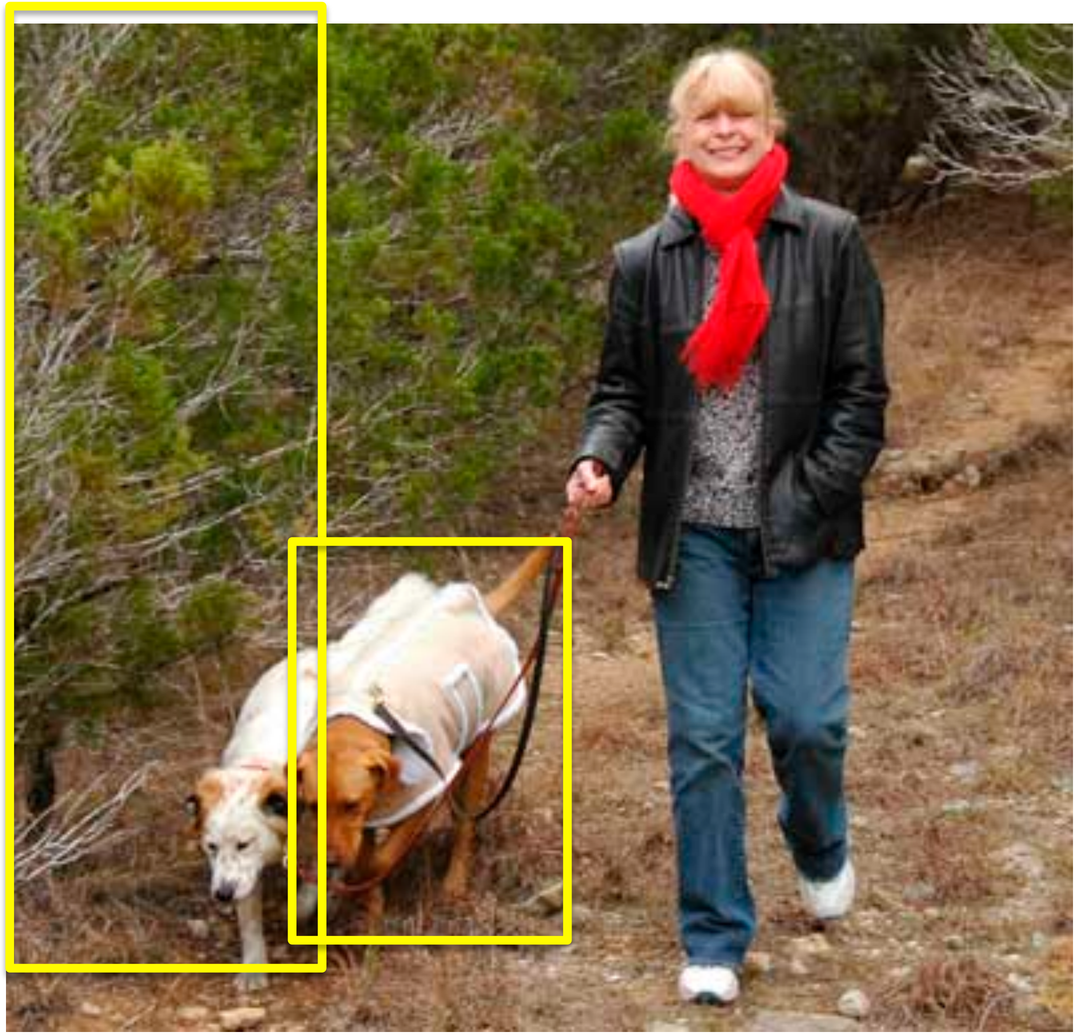


Image Categorization Pipelines

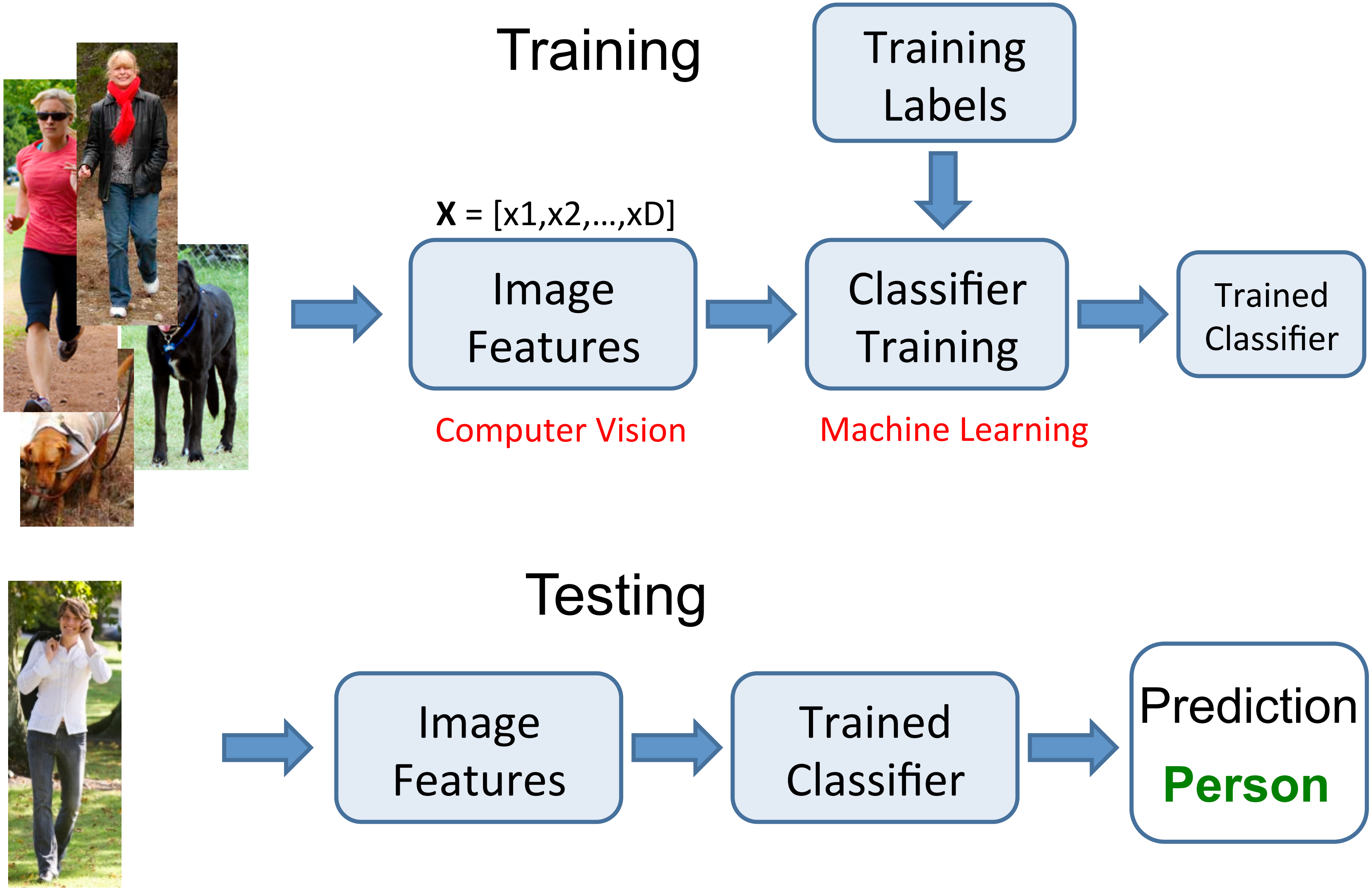


Image Categorization Pipelines

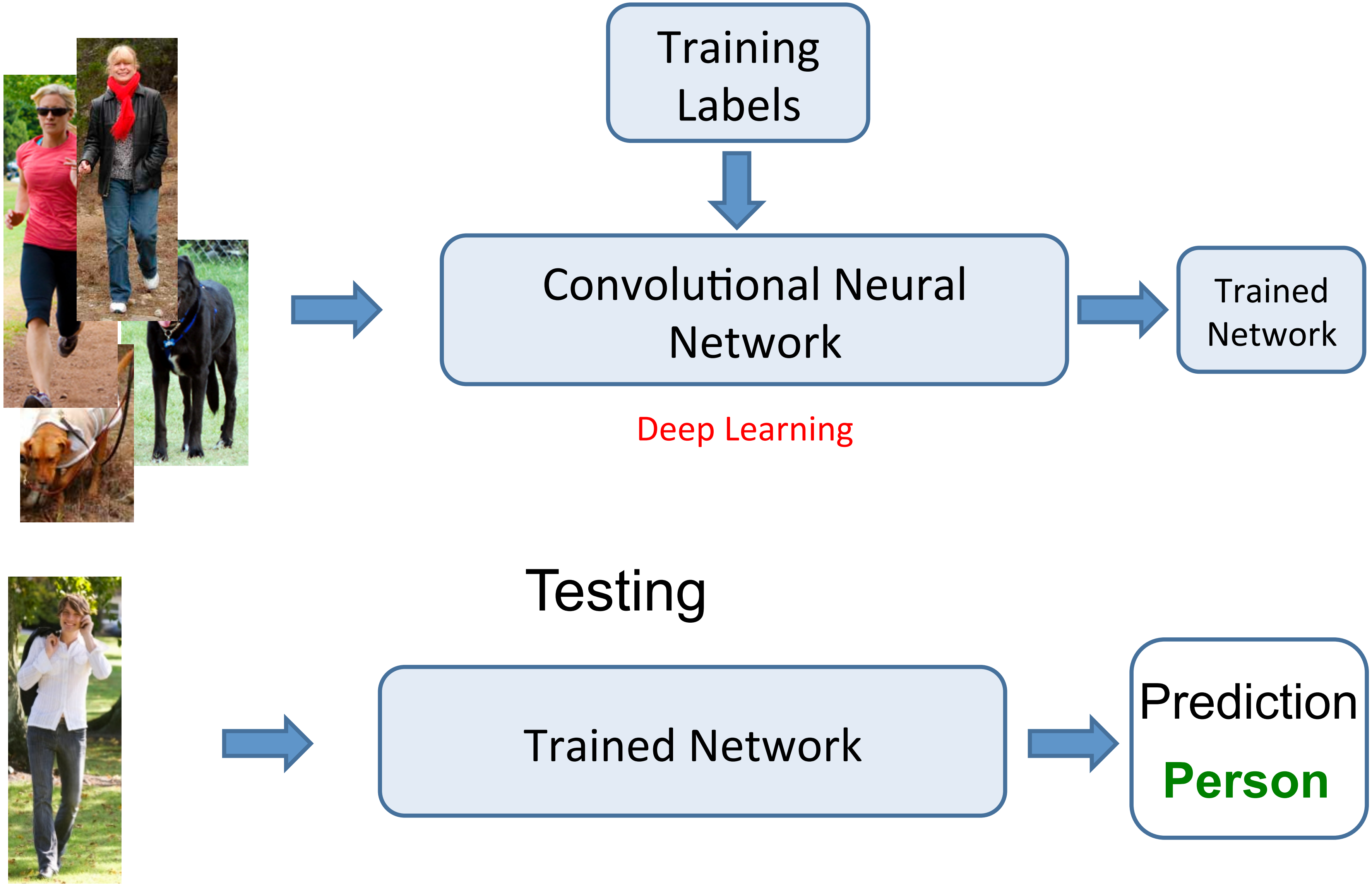
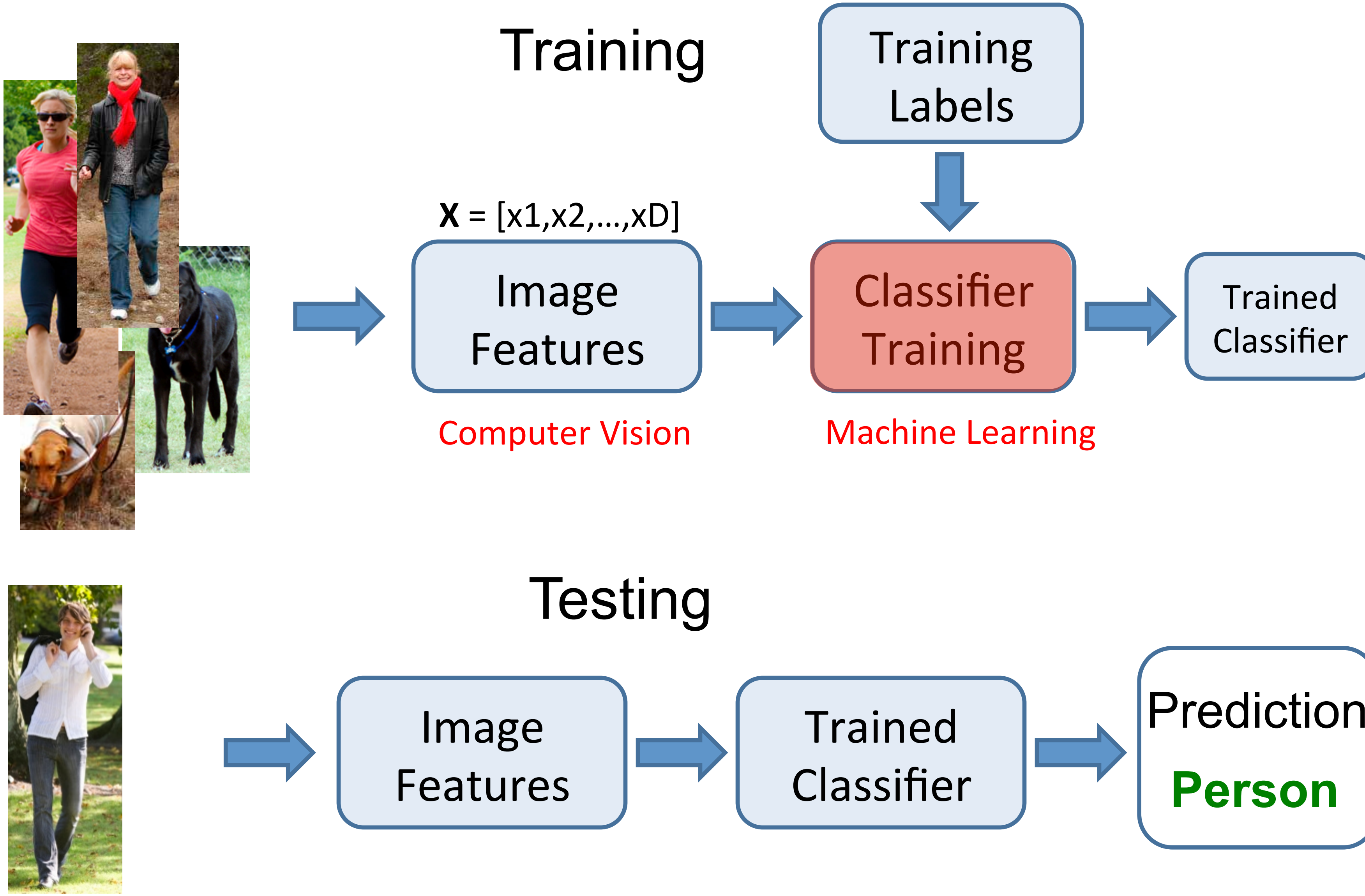


Image Categorization Pipelines



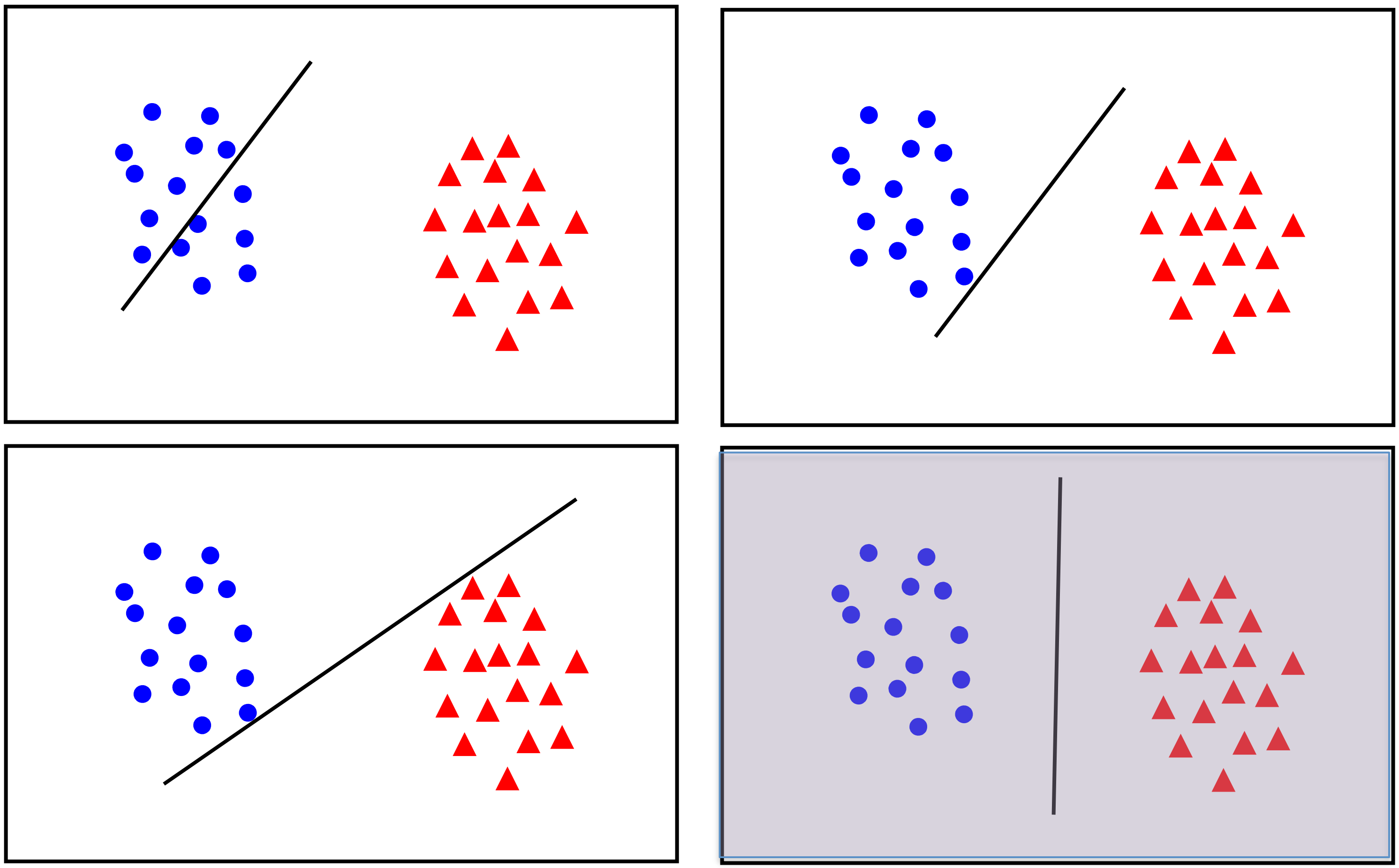
We have talked about

- Nearest Neighbor
- Naïve Bayes
- Logistic Regression
- Boosting

- We saw face detection

Support Vector Machines (SVM)

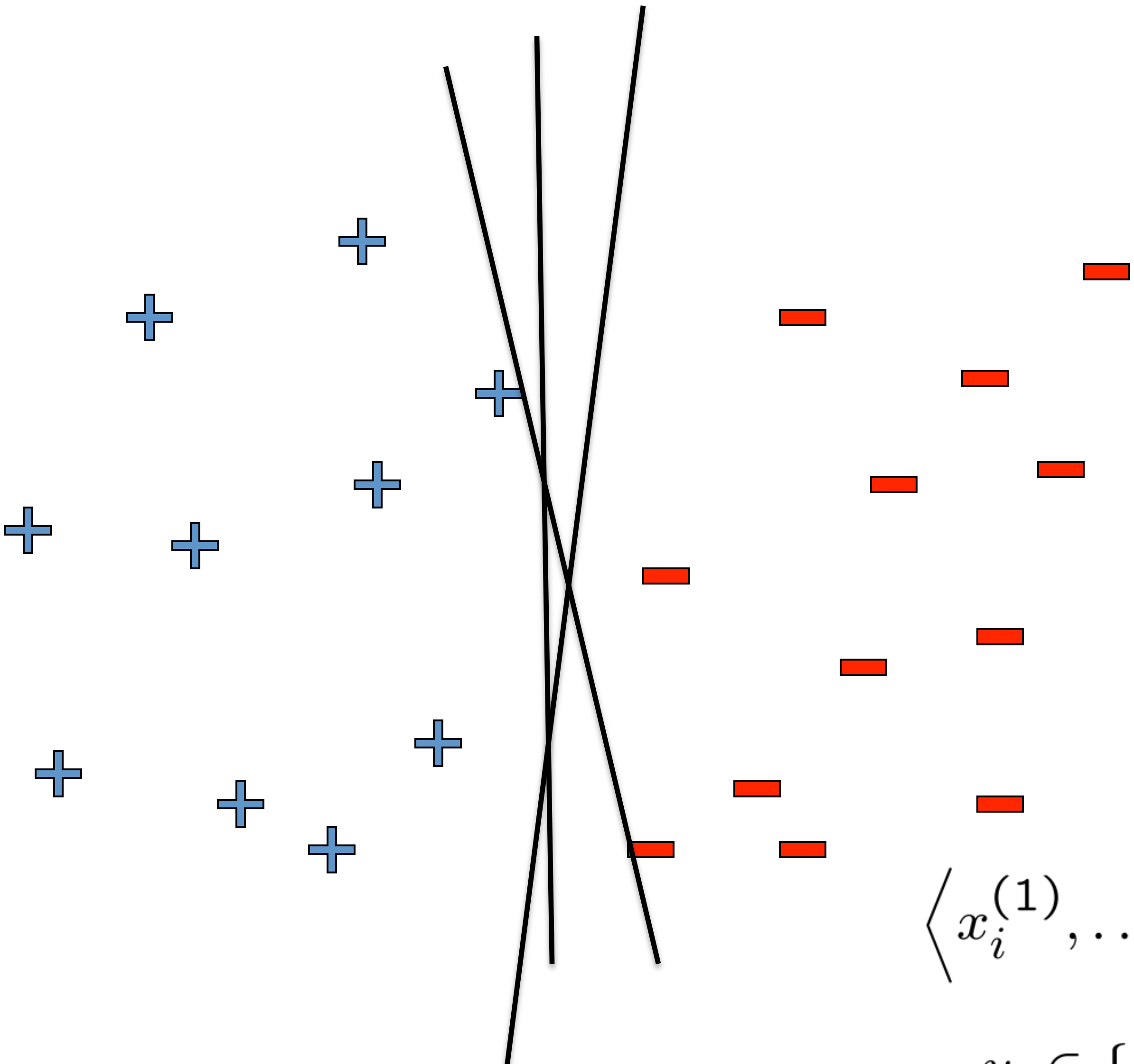
Which one is the best?



• maximum margin solution: most stable under perturbations of the inputs

Linear classifiers – How to find the best?

$$\mathbf{v}. = \sum_j v^{(j)}\, x^{(j)}$$



Data

$$\left\langle x_1^{(1)}, \ldots, x_1^{(m)}, y_1 \right\rangle$$

$$\vdots$$

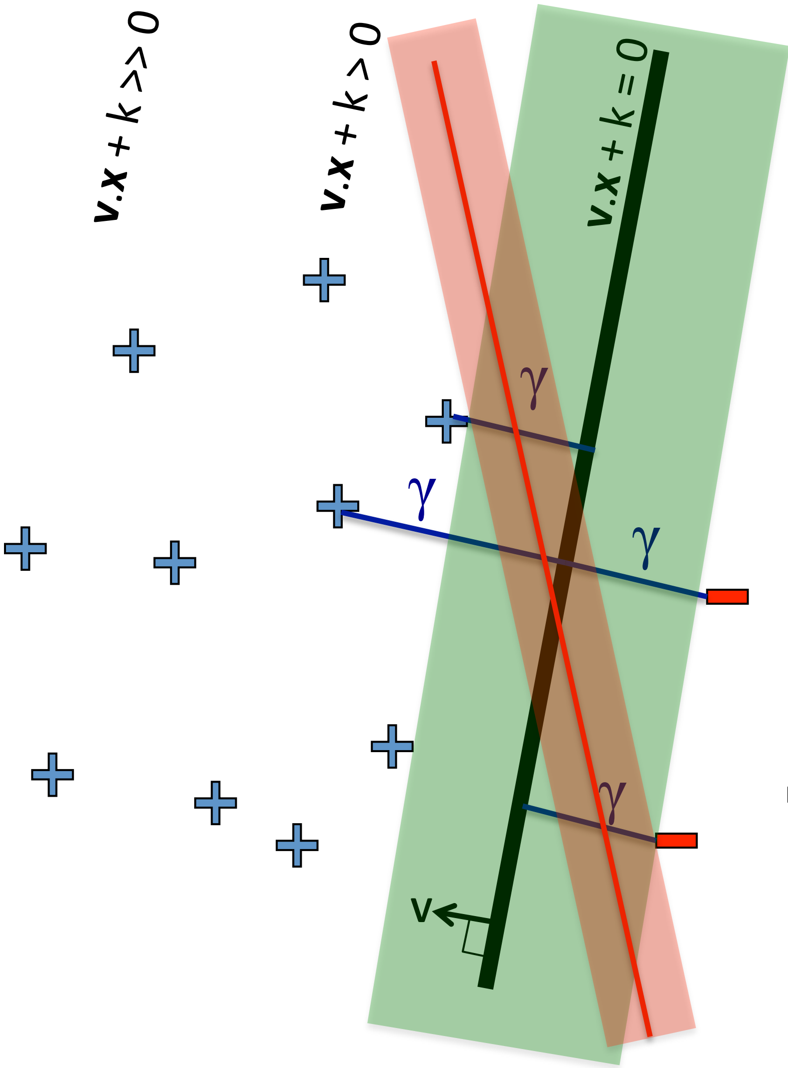
$$\left\langle x_n^{(1)}, \ldots, x_n^{(m)}, y_n \right\rangle$$

Example i

$$\left\langle x_i^{(1)}, \ldots, x_i^{(m)} \right\rangle \quad \text{— } m \text{ features}$$

$$y_i \in \{-1, +1\} \quad \text{— class}$$

Max Margin



v.x = $\sum_j$ v$^{(j)}$ x$^{(j)}$

$$\gamma_j = (v.x_j + k)y_j$$

$$\|v\|_2 = 1$$

Margin: Minimum distance between the data points and the plain

Max Margin: two equivalent forms

(1) $\max_{v,k} \min_j \gamma_j$

(2) $\max_{\gamma,v,k} \gamma$

$$\forall j \ (v.x_j + k)y_j > \gamma$$

Solution

$$\max_{\gamma, v, k} \gamma$$
$$\forall j \ (v.x_j + k)y_j > \gamma$$
$$\|v\|_2 = 1$$

- Non convex formulation

$$\forall j (v^*.x_j + k^*)y_j > \gamma$$

$$\forall j (2v^*.x_j + 2k^*)y_j > \gamma$$
$$\forall j (10v^*.x_j + 10k^*)y_j > \gamma$$
$$\vdots$$
$$\forall j (100v^*.x_j + 100k^*)y_j > \gamma$$

Solution

$$\max_{\gamma,v,k} \gamma$$
$$\forall j \ (v.x_j + k)y_j > \gamma$$
$$\|v\|_2 = 1$$

$$\forall j \ (w.x_j + b) > 1$$
$$\gamma = \frac{1}{\|w\|}$$

$$\forall j \ (\frac{v}{\gamma}.x_j + \frac{k}{\gamma})y_j > \frac{\gamma}{\gamma}$$

$$w = \frac{v}{\gamma} \qquad b = \frac{k}{\gamma}$$

$$\|w\|_2 = \frac{\|v\|_2}{\gamma}$$

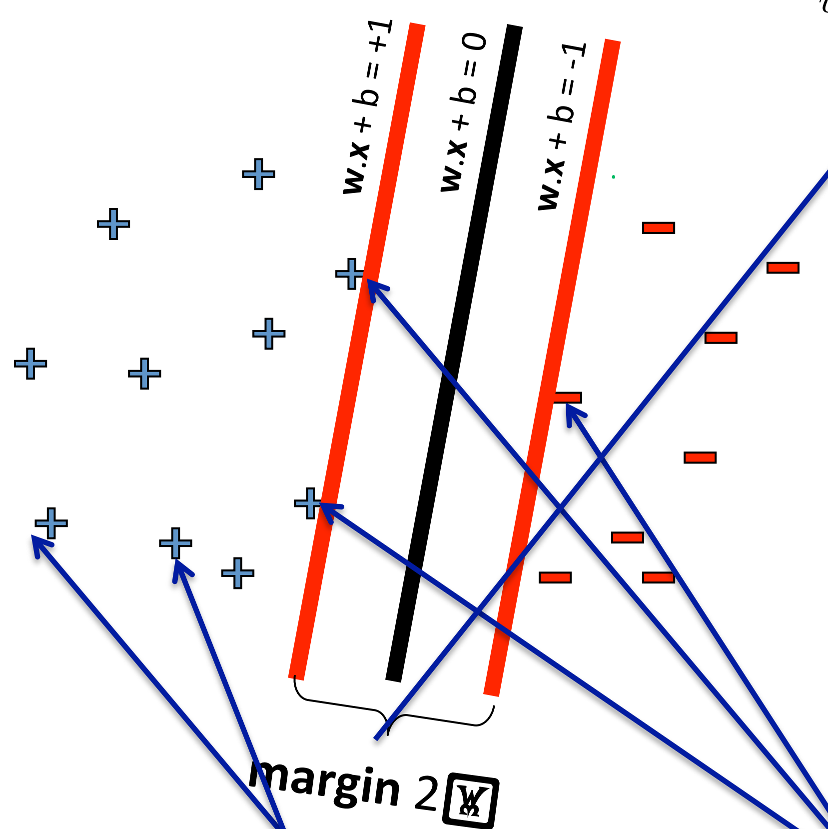
$$\min_{w,b} \|w\|$$
$$\forall j \ (w.x_j + b)y_j > 1$$

Support vector machines (SVMs)

$$\min_{w,b} \frac{1}{2} \|w\|^2$$

$$\min_{w,b} \|w\|$$

$$\forall j \ (w.x_j + b)y_j > 1$$



- Solve efficiently by quadratic programming (QP)
  - Well-studied solution algorithms
  - Not simple gradient ascent, but close
- Hyperplane defined by support vectors
  - Could use them as a lower-dimension basis to write down line, although we haven't seen how yet
  - More on this later

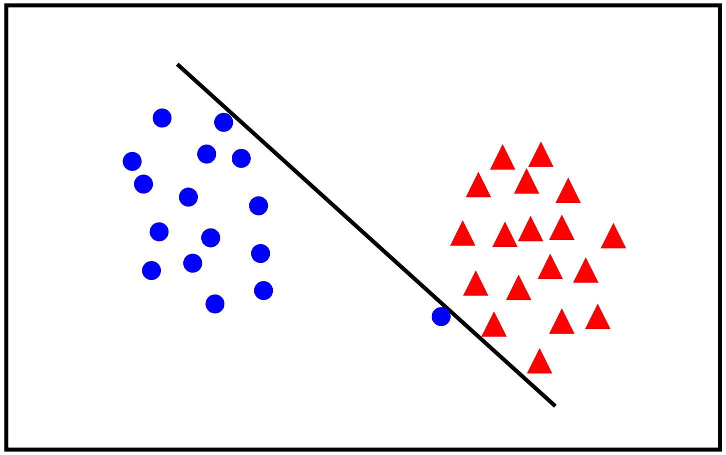
Non-support Vectors:

- everything else
- moving them will not change w

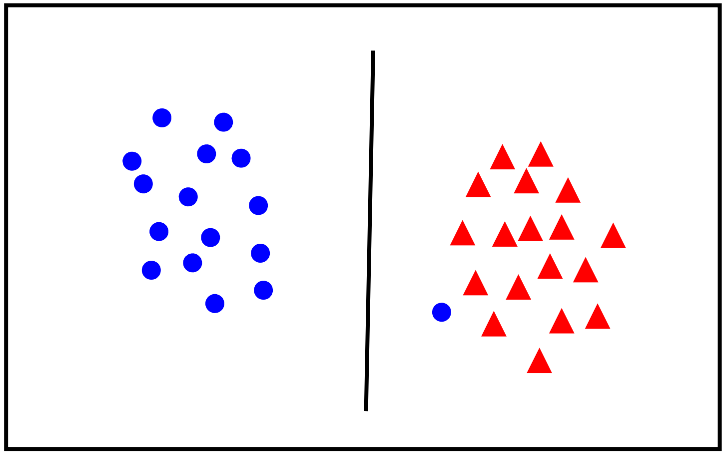
Support Vectors:

- data points on the canonical lines

Soft Margin



• the points can be linearly separated but there is a very narrow margin



• but possibly the large margin solution is better, even though one constraint is violated

In general there is a trade off between the margin and the number of mistakes on the training data

Introducing Slack Variables

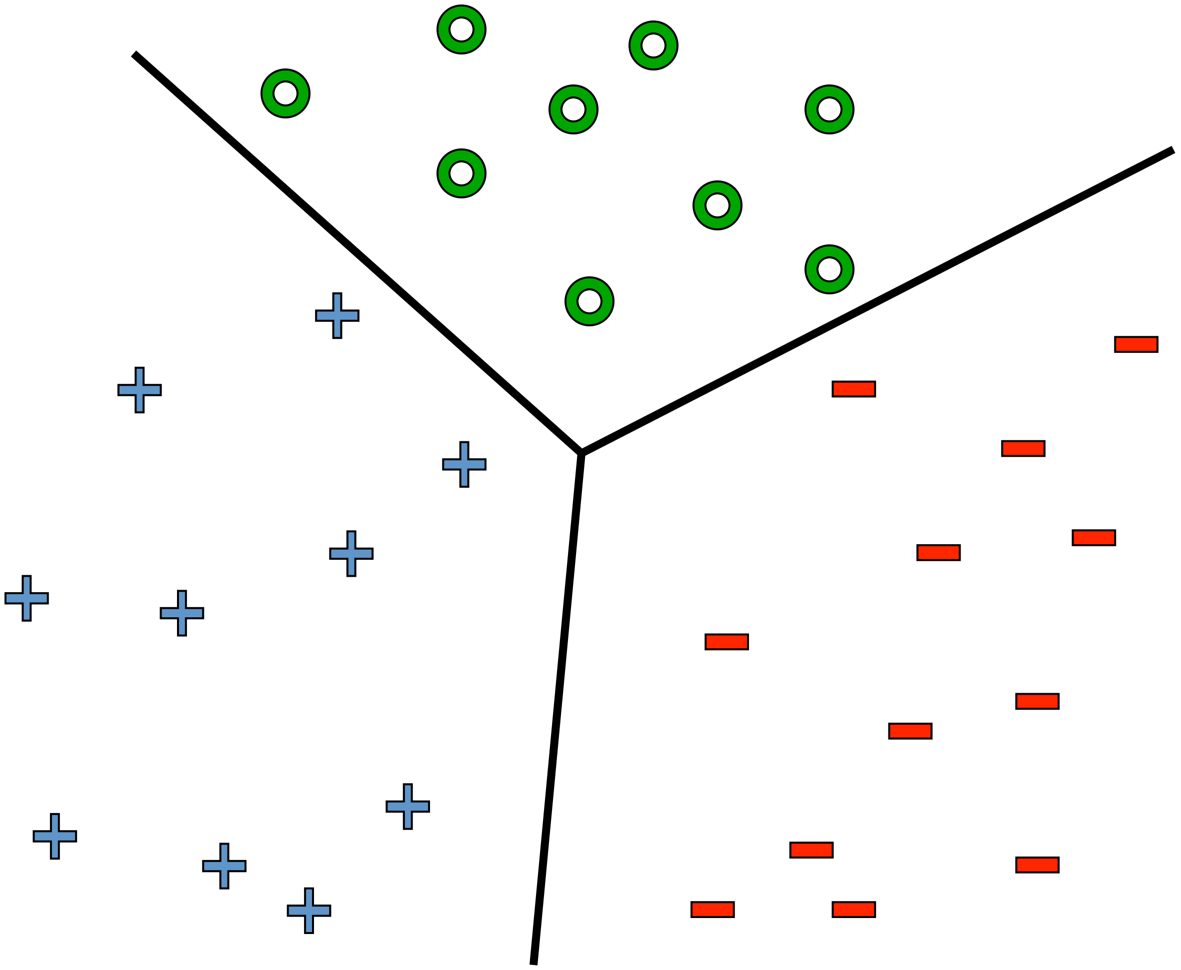
$$\min_{w,b,\xi} \|w\| + C \sum_j \xi_j$$

$$\forall j \;\; (w.x_j + b) y_j > 1 - \xi_j$$

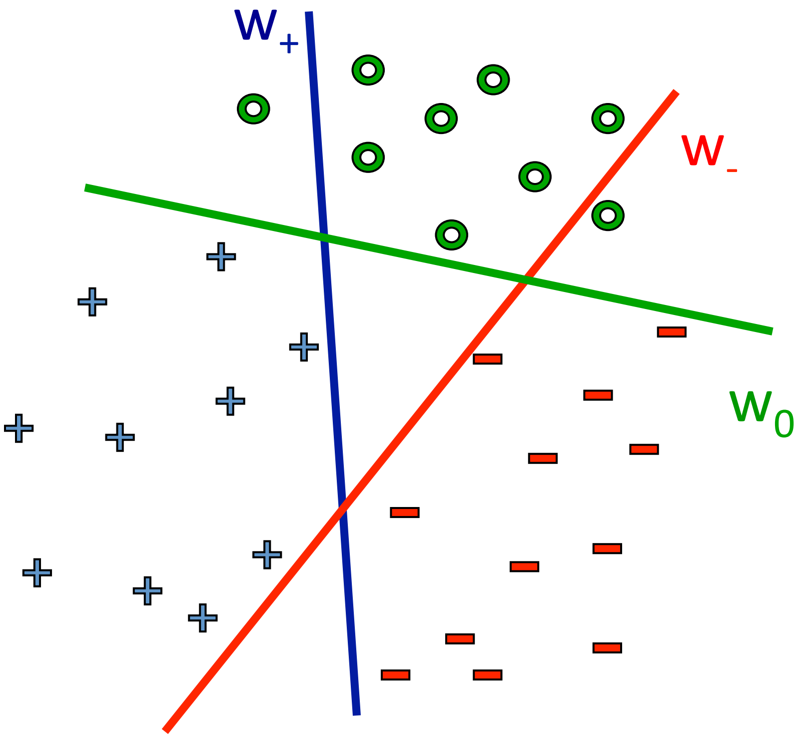
$$\forall j \;\; \xi_j \geq 0$$

- Every constraint can be satisfied if ξ_i is sufficiently large
- C is a regularization parameter:
  - small C allows constraints to be easily ignored $\rightarrow$ large margin
  - large C makes constraints hard to ignore $\rightarrow$ narrow margin
  - $C = \infty$ enforces all constraints: hard margin
- This is still a quadratic optimization problem and there is a unique minimum. Note, there is only one parameter, C.

What about multiple classes?



One against All



Learn 3 classifiers:

- + vs {0,-}, weights w_+
- - vs {0,+}, weights w_-
- 0 vs {+,-}, weights w_0

Output for x:

$y = \text{argmax}_i \; w_i.x$

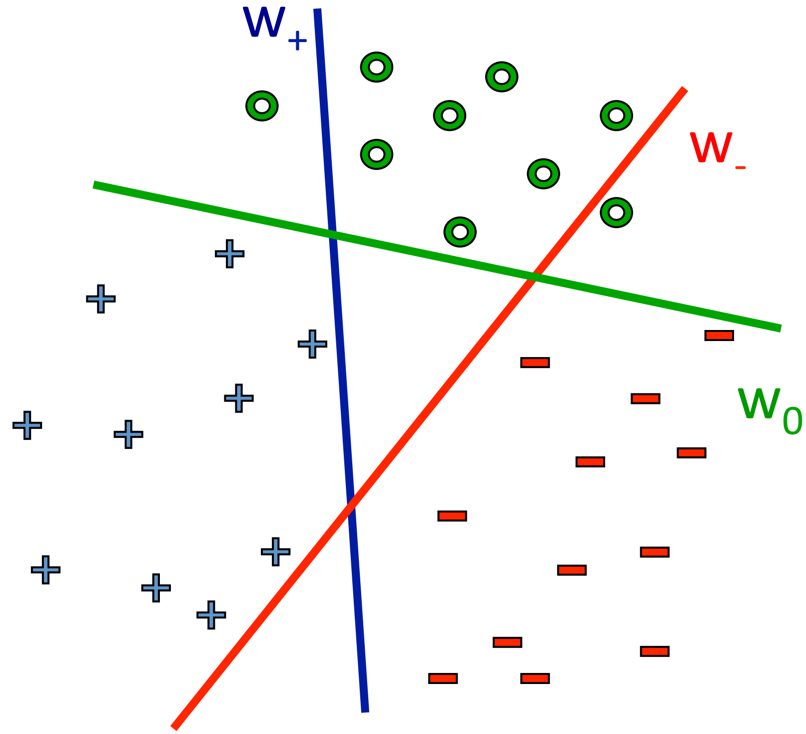
Any other way?

Any problems?

Learn 1 classifier: Multiclass SVM

Simultaneously learn 3 sets of weights:

- How do we guarantee the correct labels?
- Need new constraints!



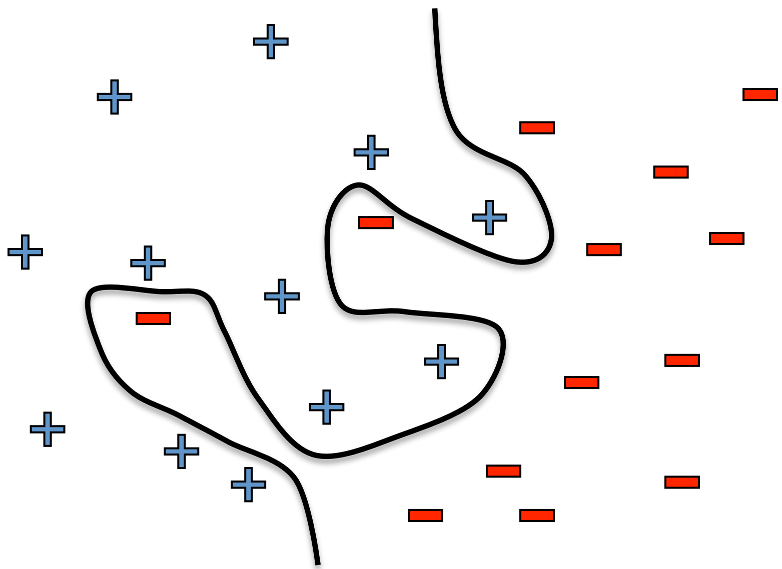
For all possible classes:

$$\mathbf{w}^{(y_j)}.\mathbf{x}_j + b^{(y_j)} \geq \mathbf{w}^{(y')}.\mathbf{x}_j + b^{(y')} + 1, \ \forall y' \neq y_j, \ \forall j$$

What if the data is not linearly separable?

$$\left\langle x_i^{(1)}, \ldots, x_i^{(m)} \right\rangle \quad \text{— } m \text{ features}$$

$$y_i \in \{-1, +1\} \quad \text{— class}$$



Add More Features!!!

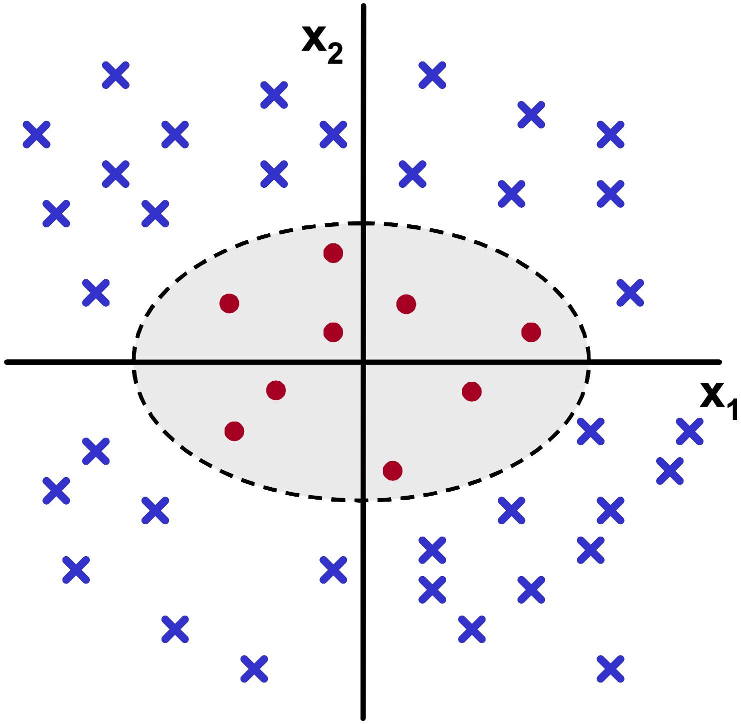
$$\phi(x) = \begin{pmatrix} x^{(1)} \\ \ldots \\ x^{(n)} \\ x^{(1)}x^{(2)} \\ x^{(1)}x^{(3)} \\ \ldots \\ e^{x^{(1)}} \\ \ldots \end{pmatrix}$$

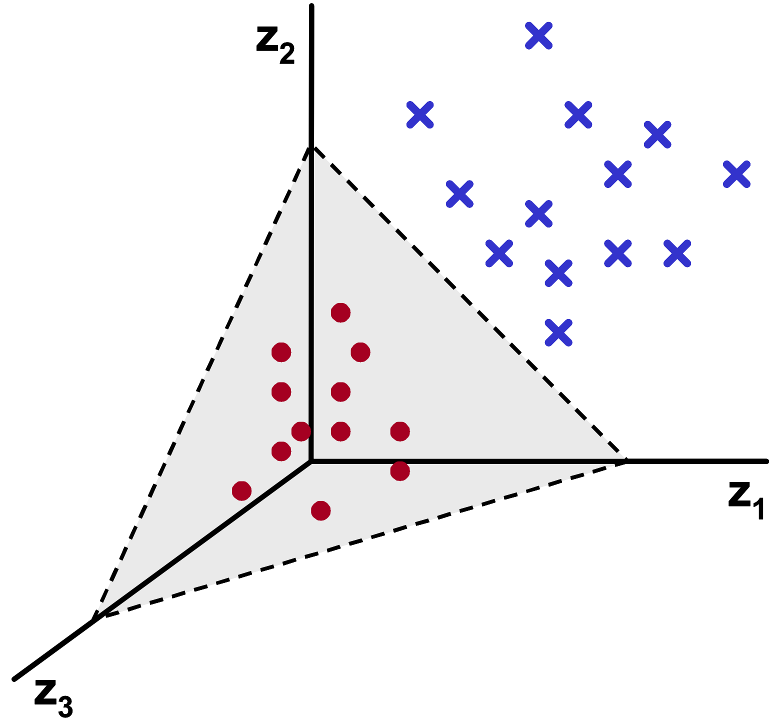
SVM with a polynomial Kernel visualization

Created by:
Udi Aharoni

Non-Linear SVM

$$\psi : R^2 \rightarrow R^3 \quad \psi(\mathbf{x}) = (z_i, z_2, z_3) = \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right)$$





So What?!!!

- Logistic Regression

$$l(w) = \sum_j (w\psi(x_j) + b) y_j - \ln(1 + e^{\sum_j w\psi(x_j) + b})$$

  - No Large Margin
  - No Quadratic Programming
  - Concave Optimization

Dual Form (Lagrange Multiplier)

$$\min_{\theta} f(\theta)$$
$$\forall j \;\; g_j(\theta) \geq 0$$

$$\max_{\alpha:\alpha_j \geq 0} \min_{\theta} \mathcal{L}(\theta, \alpha) = f(\theta) - \left[\sum_j \alpha_j g_j(\theta) \right]$$

$$\min_{w,b} \frac{1}{2} \|w\|^2$$
$$\forall j \;\; (w.x_j + b) - 1 \geq 0$$

$$\max_{\alpha:\alpha_i \geq 0} \min_{w,b} \mathcal{L}(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \left[\sum_i \alpha_i (w.x_i + b - 1) \right]$$

$$\nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0 \quad \Longrightarrow \quad \boxed{\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i}$$

$$\frac{\partial}{\partial b} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i y_i = 0$$

Dual Form

- Plug in the new definition of $\mathbf{w}$ into the Lagrangian and simplify

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n} y_i y_j \alpha_i \alpha_j \mathbf{x}_i^T \mathbf{x}_j - b\sum_{i=1}^{n} \alpha_i y_i$$

but $\sum_{i=1}^{n} \alpha_i y_i = 0$. Thus

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n} y_i y_j \alpha_i \alpha_j \mathbf{x}_i^T \mathbf{x}_j$$

- Putting everything together get the dual problem optimization problem

$$\max_{\boldsymbol{\alpha}} \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n} y_i y_j \alpha_i \alpha_j \mathbf{x}_i^T \mathbf{x}_j \right\}$$

subject to $\alpha_i \geq 0$ for $i = 1, \ldots, n$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$

Implicit Mapping

Recall that the SVM solution depends only on the dot product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ between training examples.

Non-linear separable: $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle$

$$\psi(\mathbf{x}) = \begin{pmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \end{pmatrix}$$

$$\begin{aligned} K(\mathbf{x}, \mathbf{z}) &= \psi(\mathbf{x})^T \psi(\mathbf{z}) = \sum_{i=1}^{d} \sum_{j=1}^{d} (x_i x_j)(z_i z_j) \\ &= \left(\sum_{i=1}^{d} x_i z_i \right) \left(\sum_{j=1}^{d} x_j z_j \right) = (\mathbf{x}^T \mathbf{z})^2 \end{aligned}$$

$$K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T \mathbf{z})^2$$

Kernel Function

Popular Kernel Functions

Polynomial kernels

$$K(\mathbf{x}, \mathbf{z}) = \left(\mathbf{x}^T \mathbf{z} + 1\right)^p$$

The degree of the polynomial is a user-specified parameter.

Radial basis function kernels

$$K(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2}\right) = \sum_{k=0}^{\infty} \frac{1}{k!} \left(-\frac{\|\mathbf{x} - \mathbf{z^2}\|}{2\sigma^2}\right)^k$$

The width σ is a user-specified parameter. This kernel corresponds to an infinite dimensional feature mapping ψ.

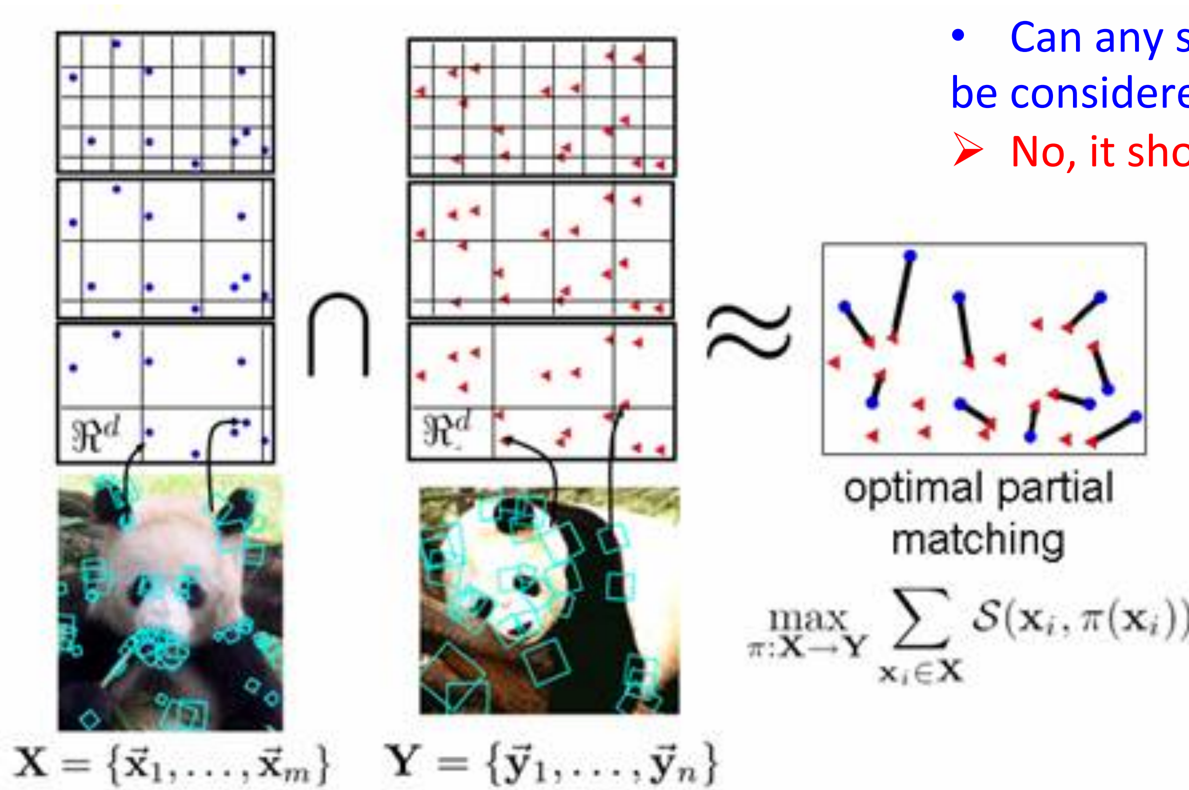
Sigmoid Kernel

$$K(\mathbf{x}, \mathbf{z}) = \tanh\left(\beta_0 \mathbf{x}^T \mathbf{z} + \beta_1\right)$$

Active Research !!

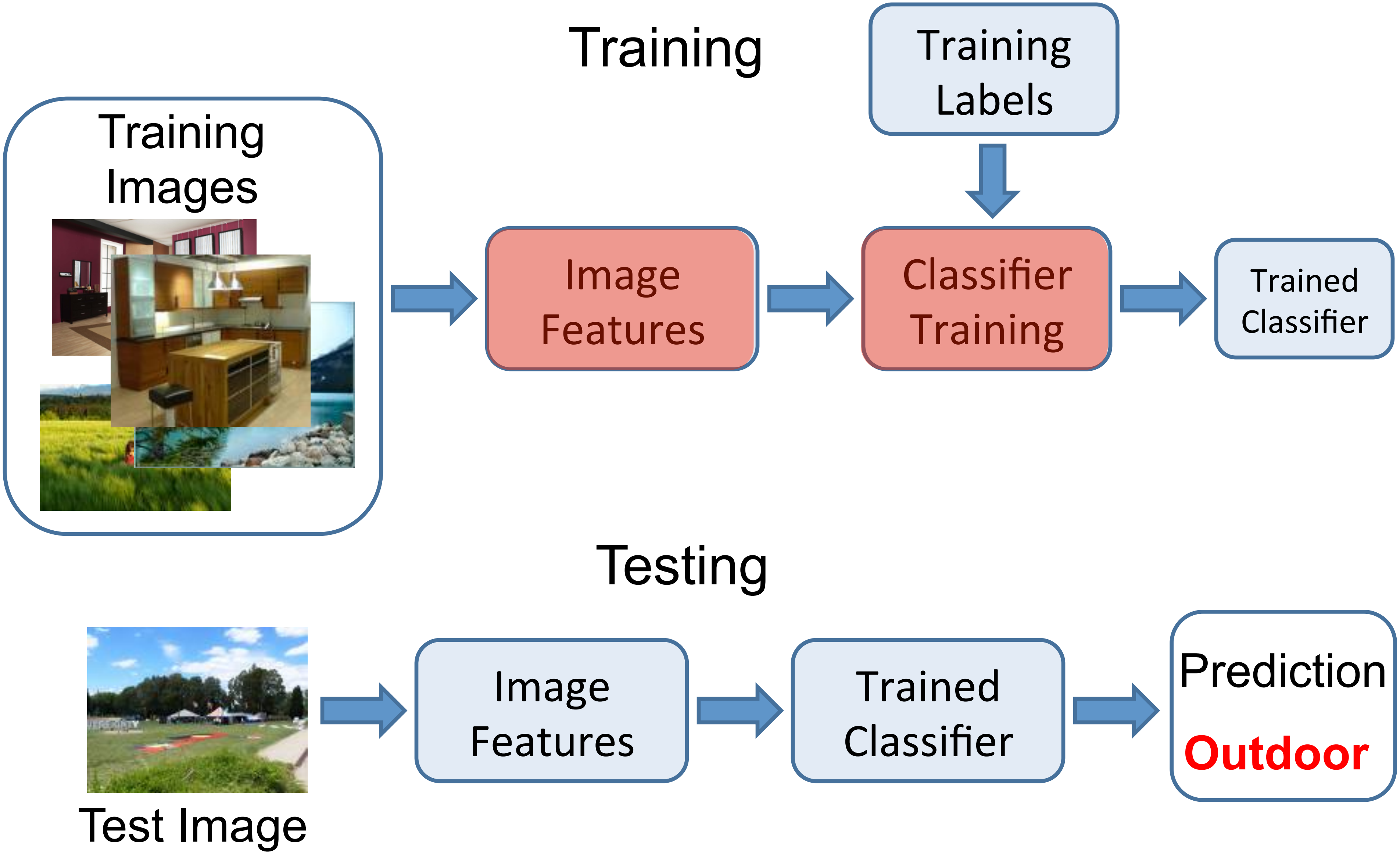
Visual Kernels

Pyramid Match Kernel [Graumen et al. 03]



- Can any similarity measure be considered as kernel?
  - ➢ No, it should satisfy Mercer conditions.

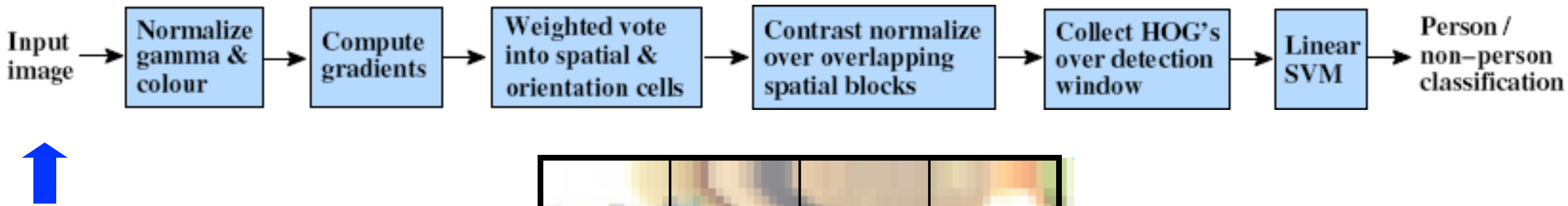
Image Categorization

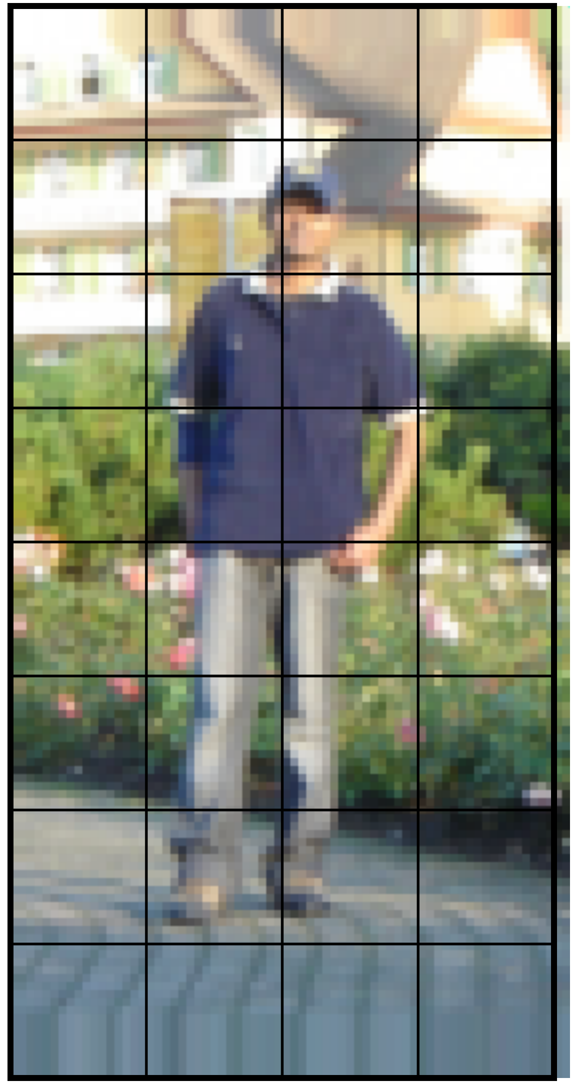


Example: Dalal-Triggs pedestrian

1. Extract fixed-sized (64x128 pixel) window at each position and scale
2. Compute HOG (histogram of gradient) features within each window
3. Score the window with a linear SVM classifier
4. Perform non-maxima suppression to remove overlapping detections with lower scores

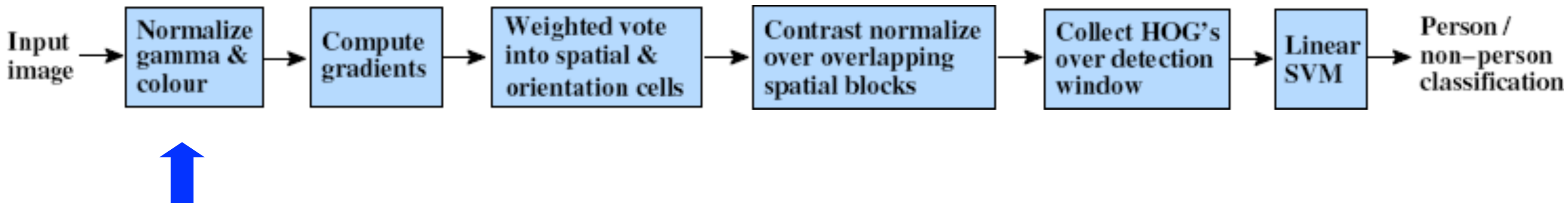
Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05





Slides by Pete Barnum

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



- Tested with
  - RGB
  - LAB
  - Grayscale

Slightly better performance vs. grayscale

Outperforms

-1	0	1

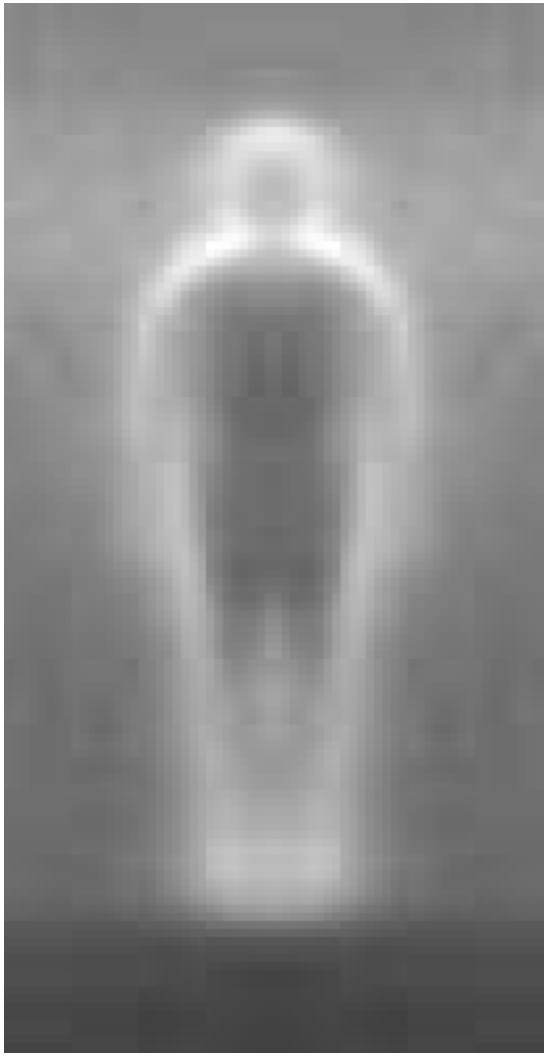
centered

-1	1

uncentered

1	-8	0	8	-1

cubic-corrected



0	1
-1	0

diagonal

-1	0	1
-2	0	2
-1	0	1

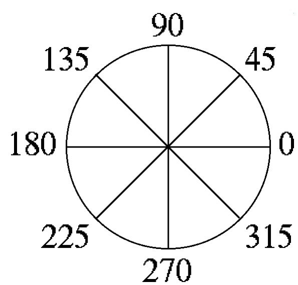
Sobel

Slides by Pete Barnum

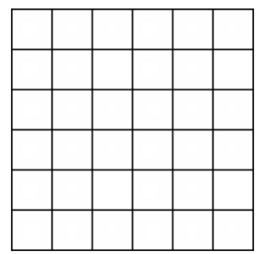
Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

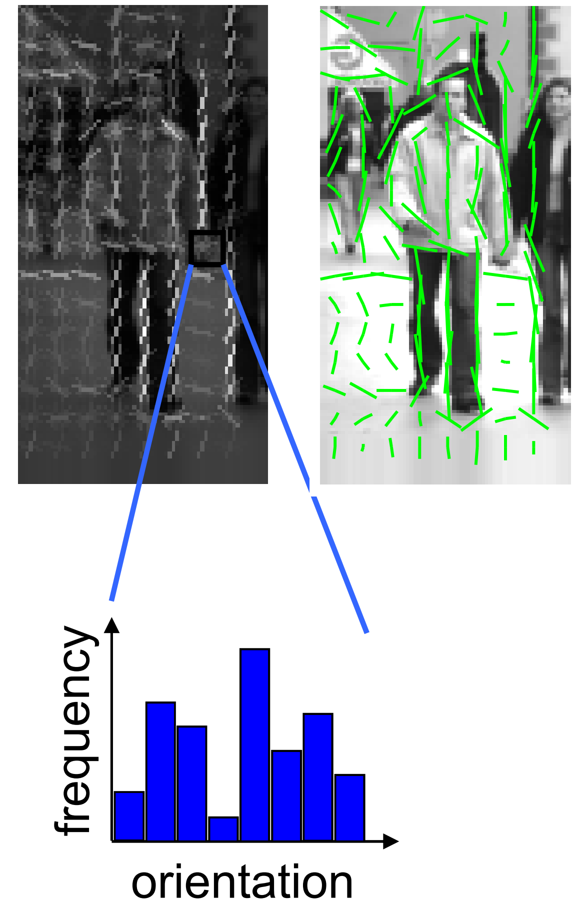
- Histogram of gradient orientations

Orientation: 9 bins (for unsigned angles)



Histograms in 8x8 pixel cells

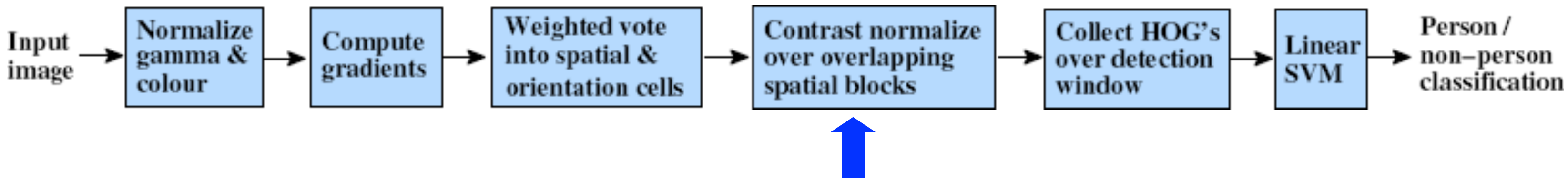




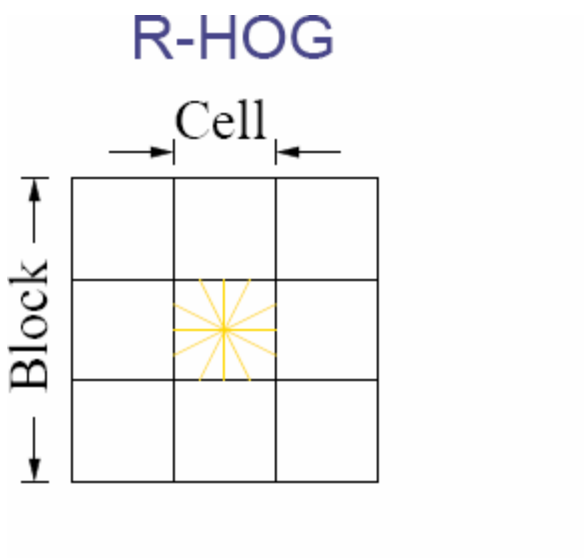
– Votes weighted by magnitude

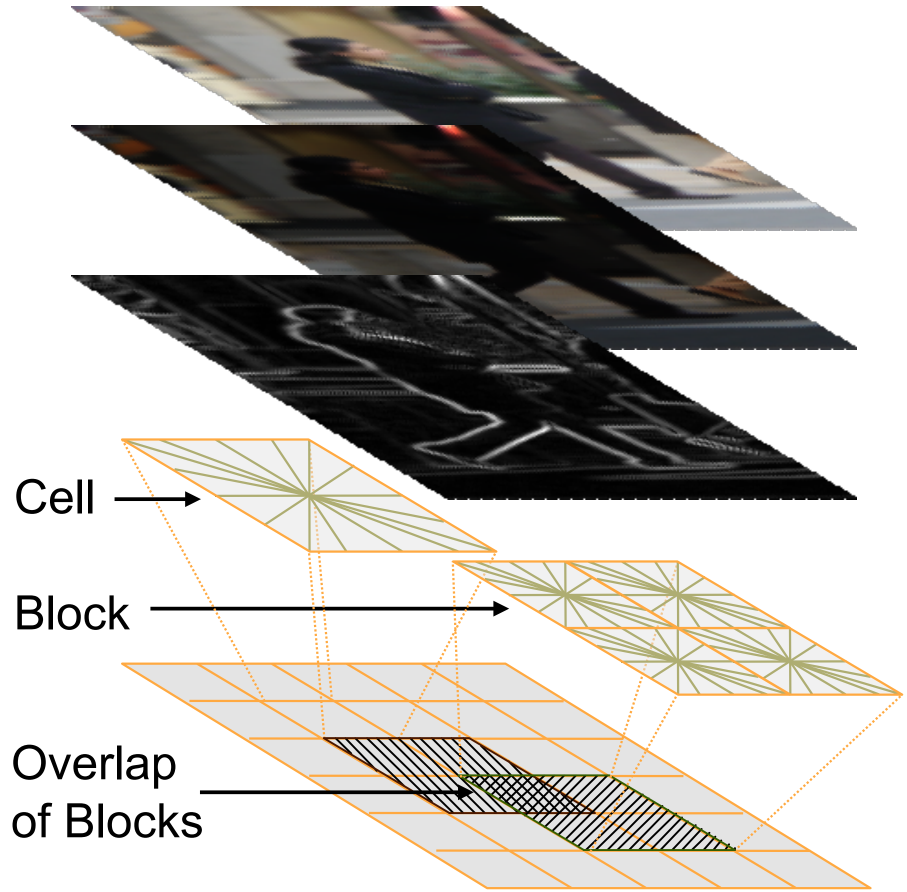
Slides by Pete Barnum

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



Normalize with respect to surrounding cells

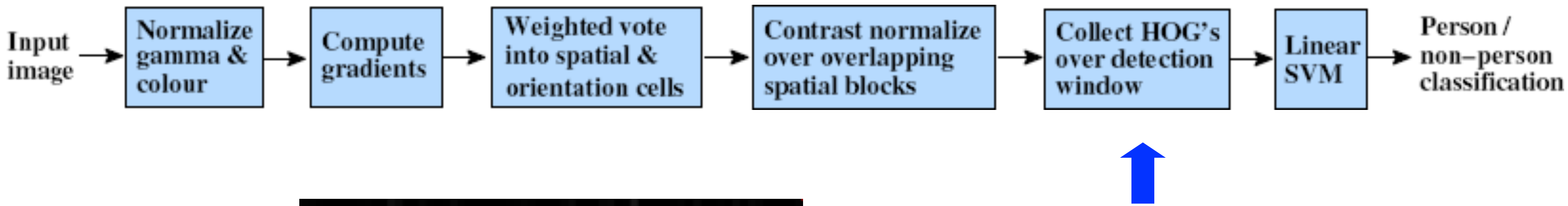




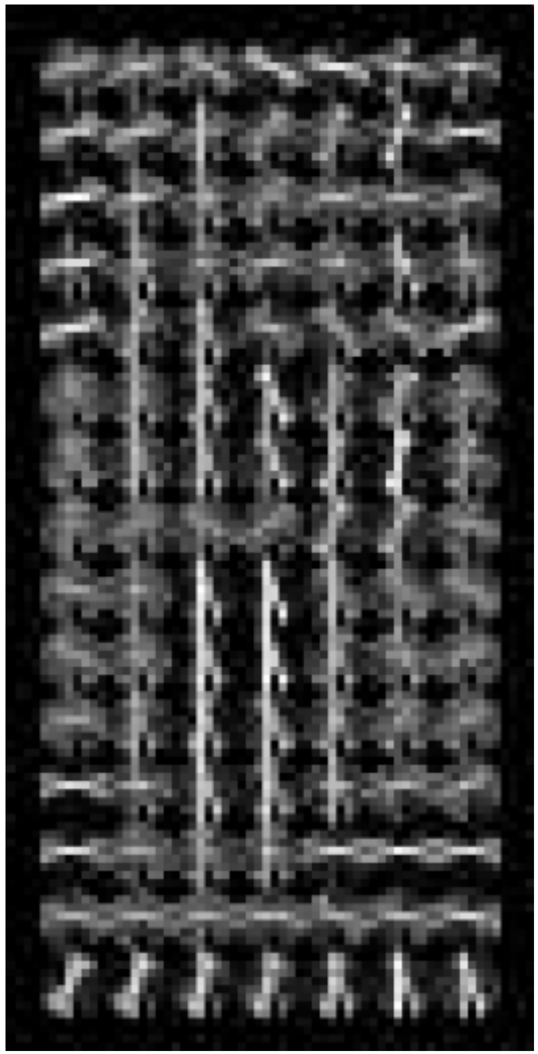
$$L2-norm : v \longrightarrow v/\sqrt{||v||_2^2 + \epsilon^2}$$

Slides by Pete Barnum

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



X=



orientations

features = 15 x 7 x 9 x 4 = 3780

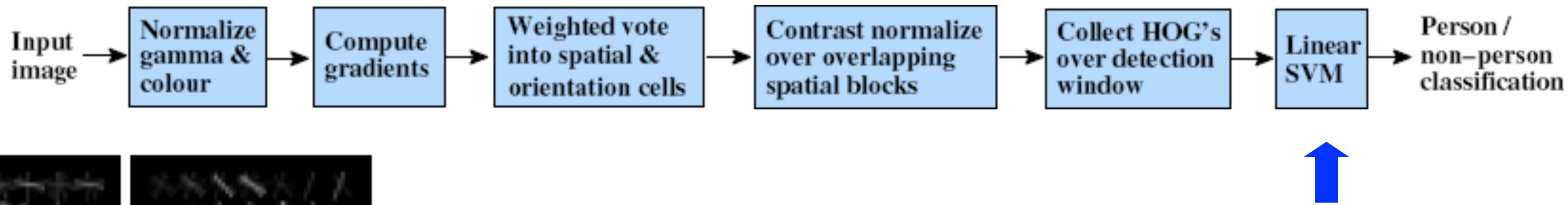
cells

normalizations by neighboring cells

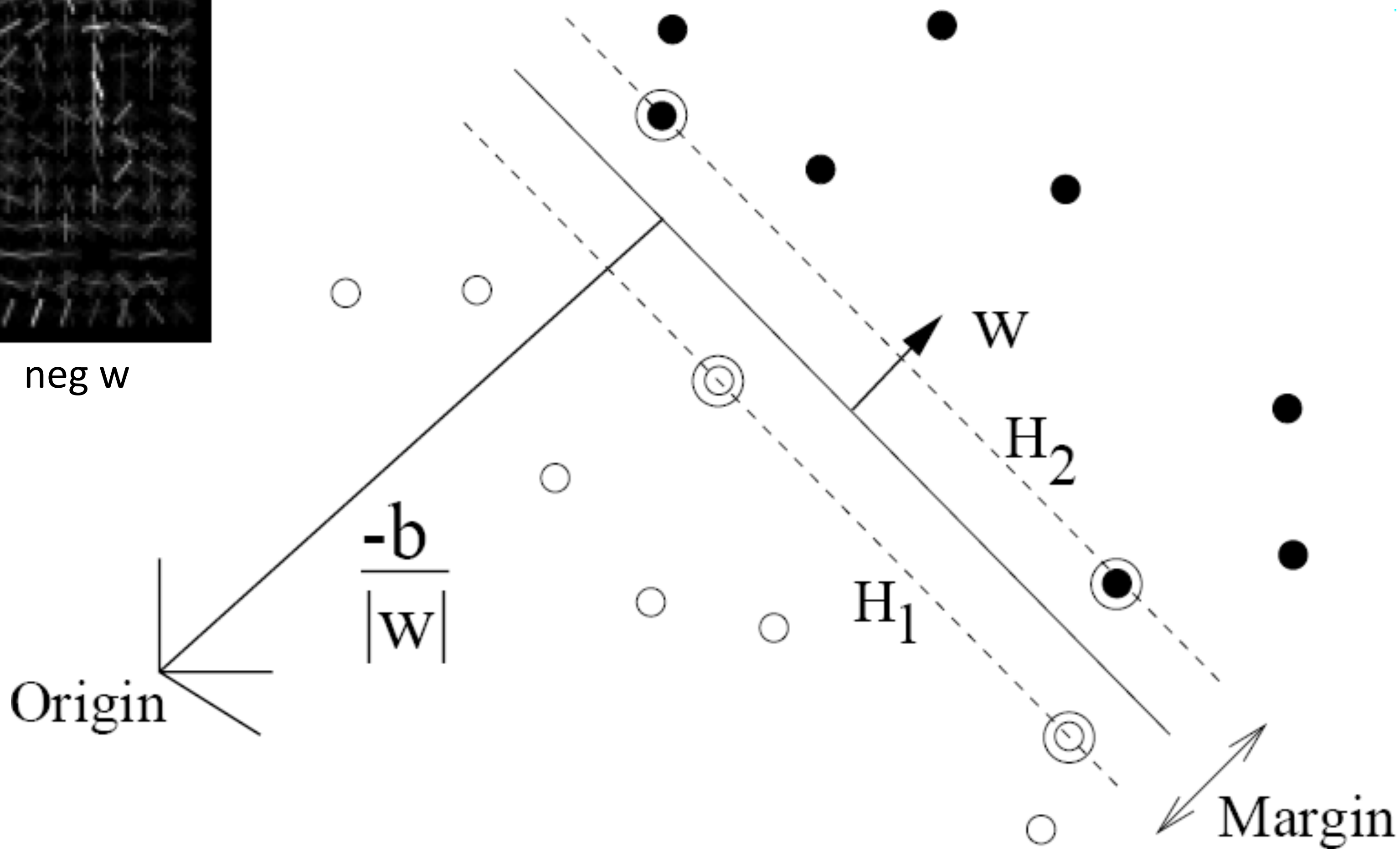
Slides by Pete Barnum

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

Training set

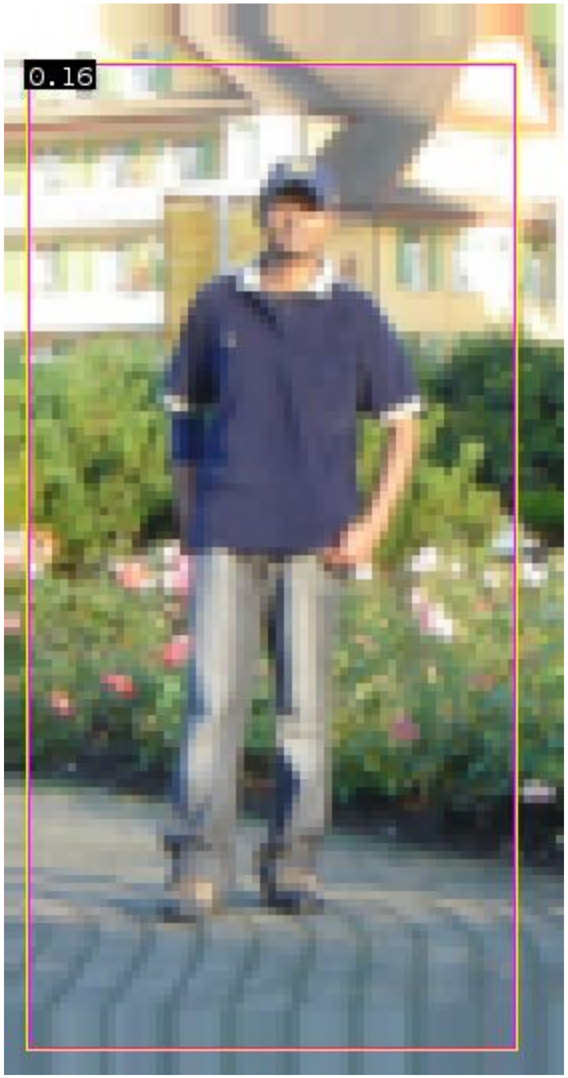


pos w   neg w



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

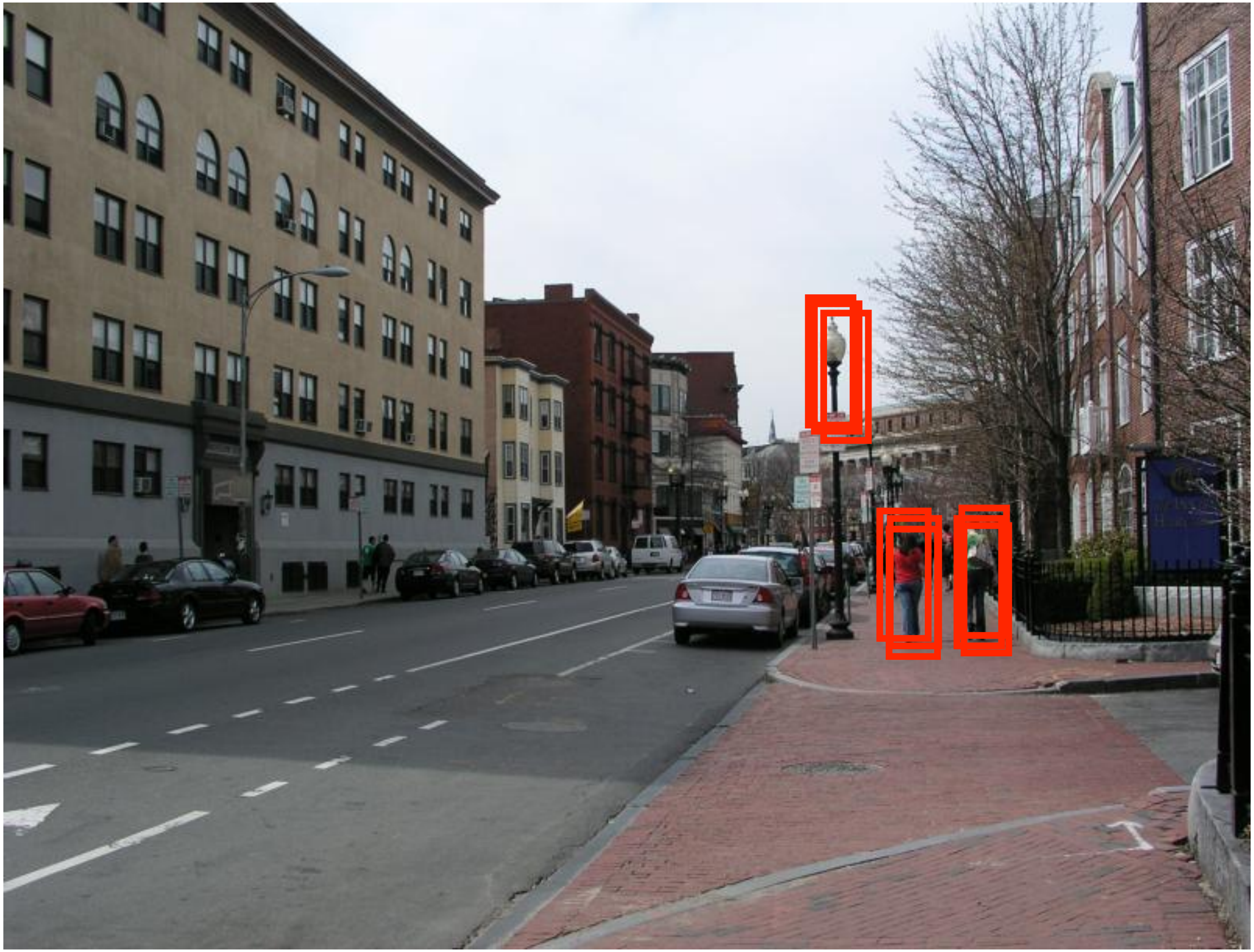
Slides by Pete Barnum

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

Detection examples

Each window is separately classified

Each window is separately classified

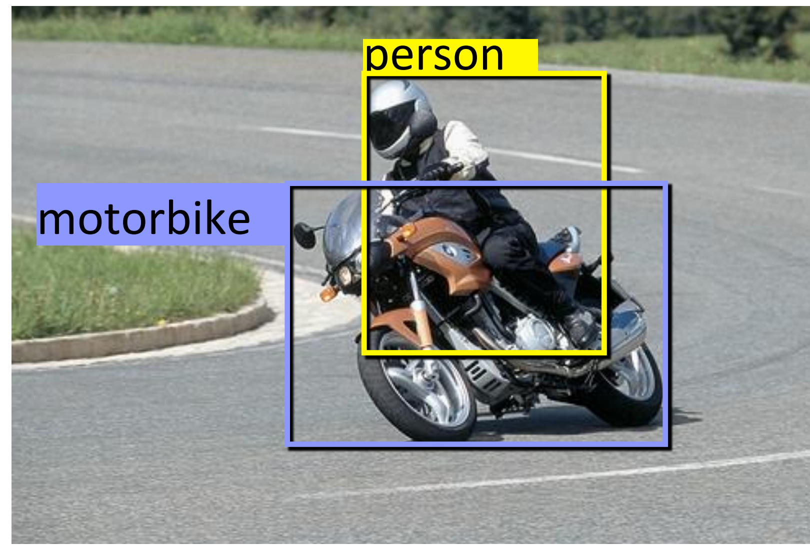


Non-Max Suppression

Problem formulation

{ airplane, bird, motorbike, person, sofa }

Input



Desired output

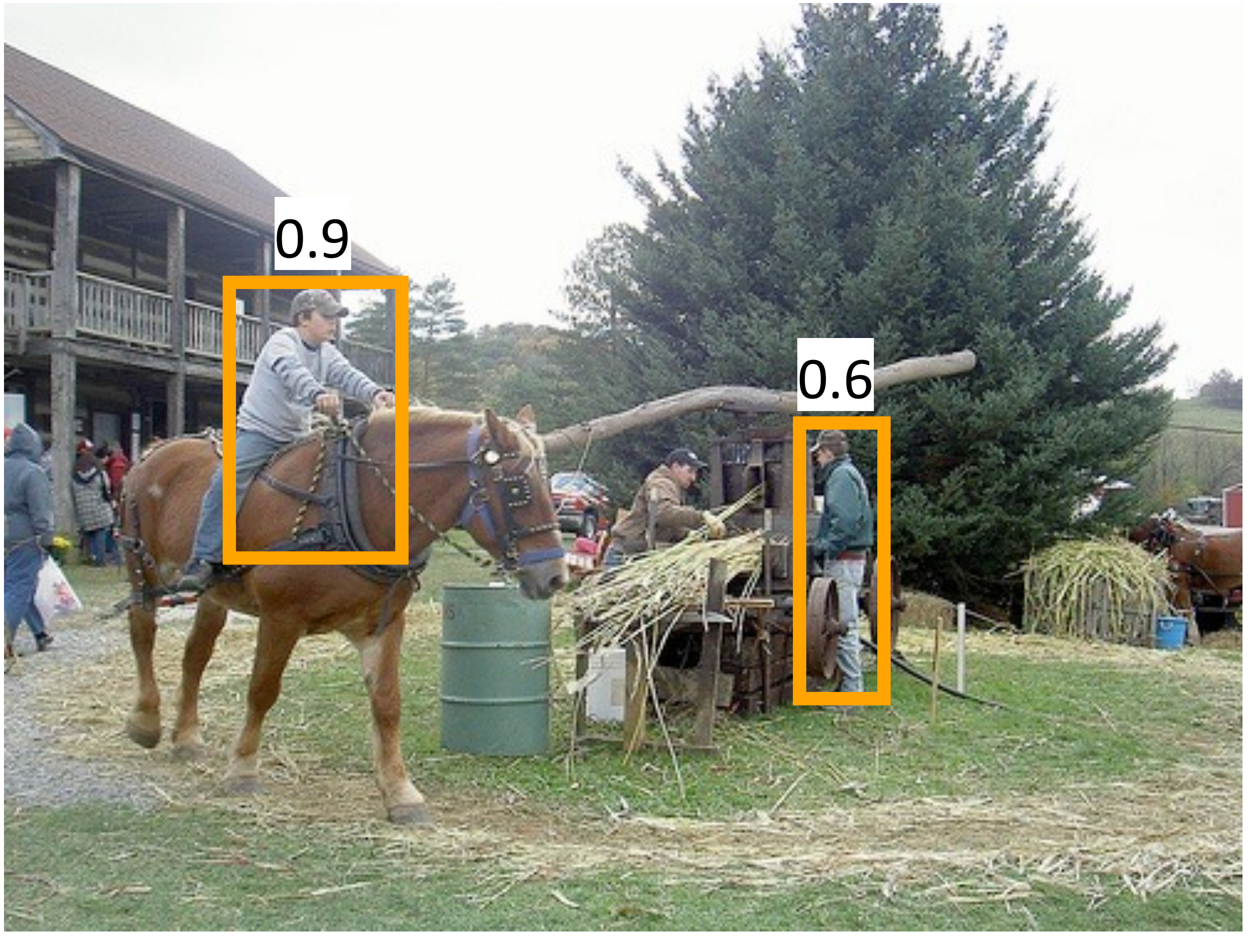
Evaluating a detector

Test image (previously unseen)

First detection ...

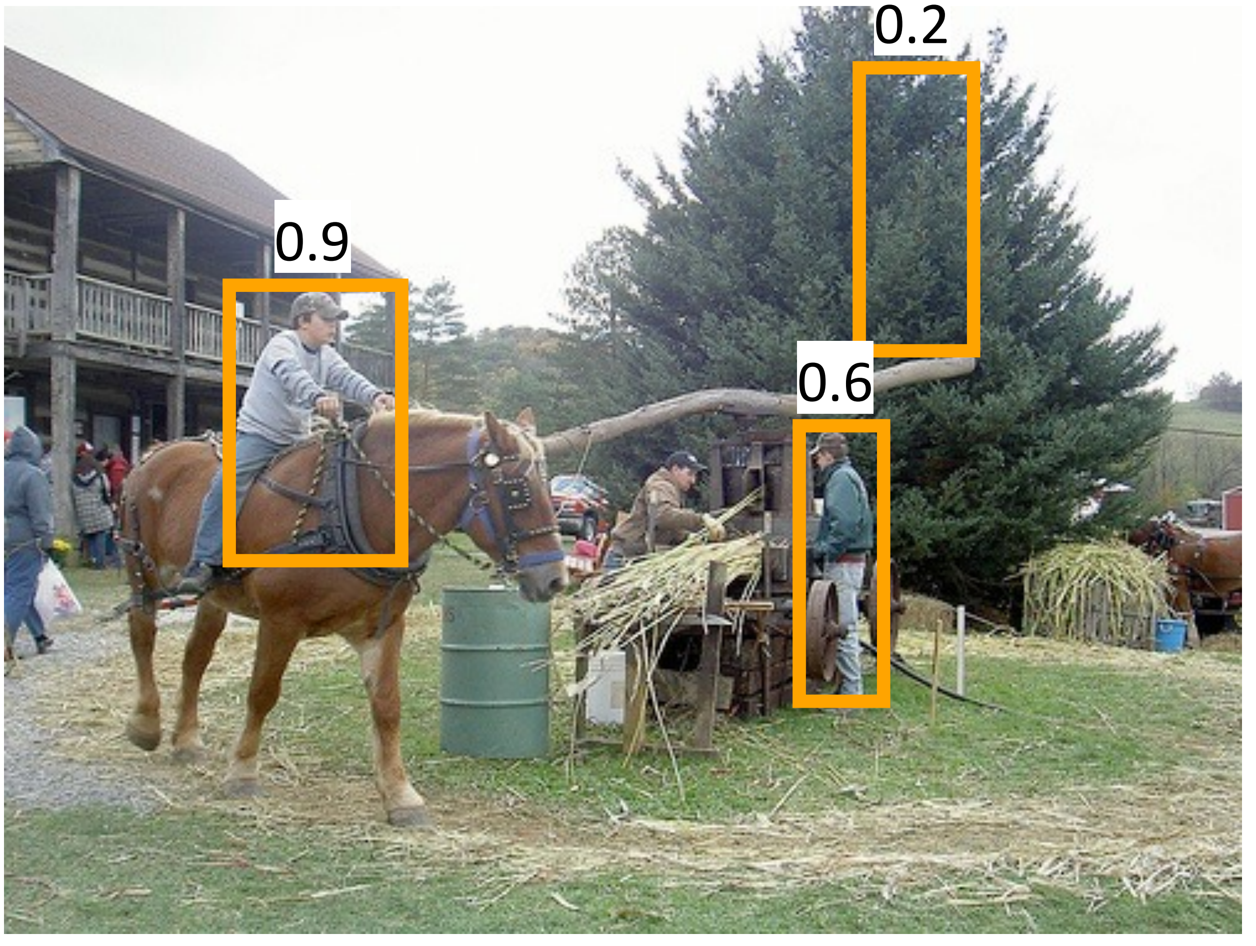
‘person’ detector predictions

Second detection ...



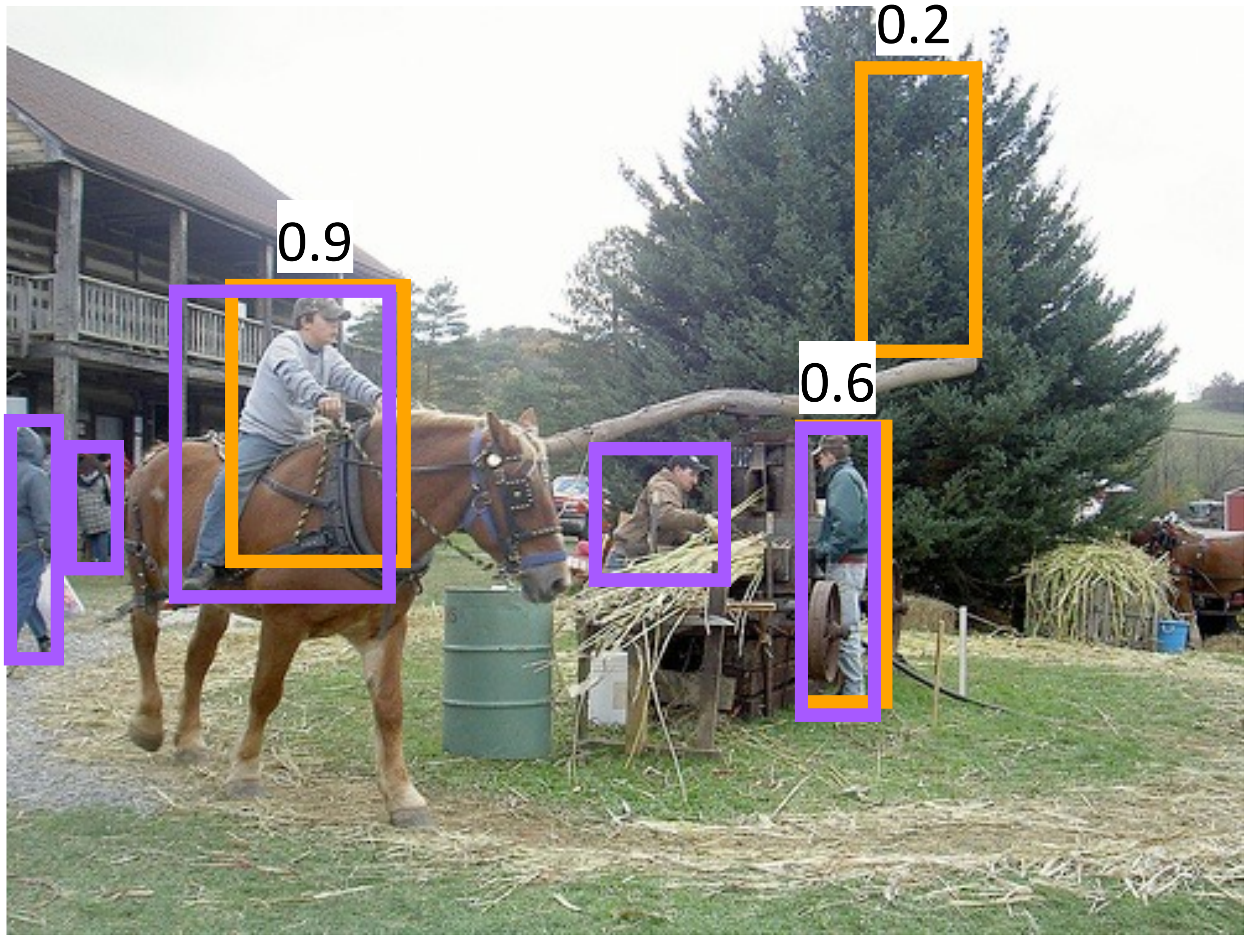
'person' detector predictions

Third detection ...



'person' detector predictions

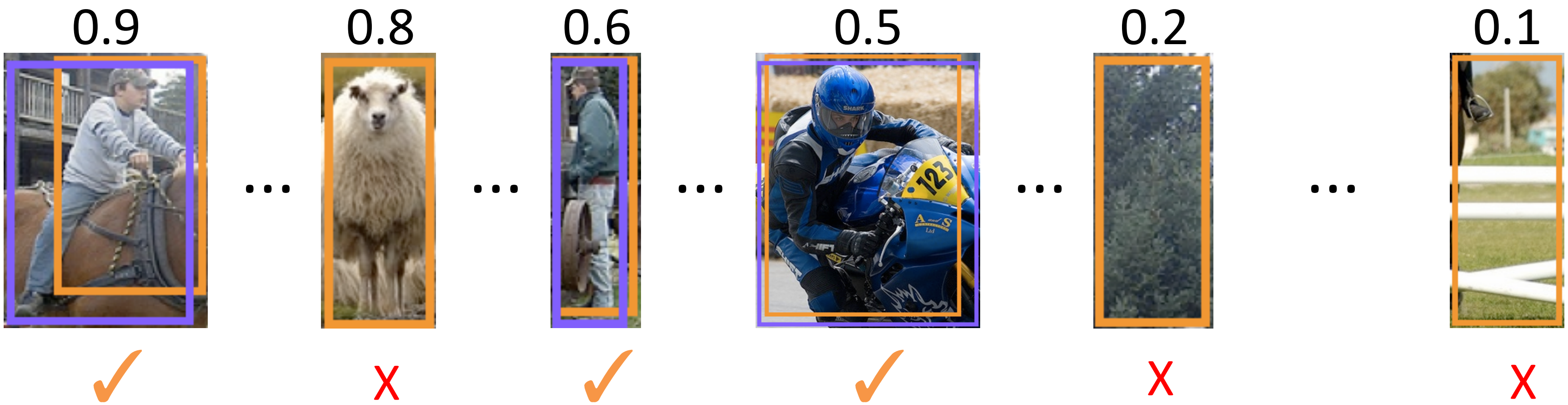
Compare to ground truth



‘person’ detector predictions

ground truth ‘person’ boxes

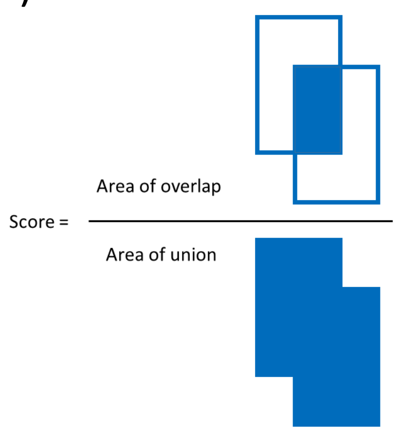
Sort by confidence

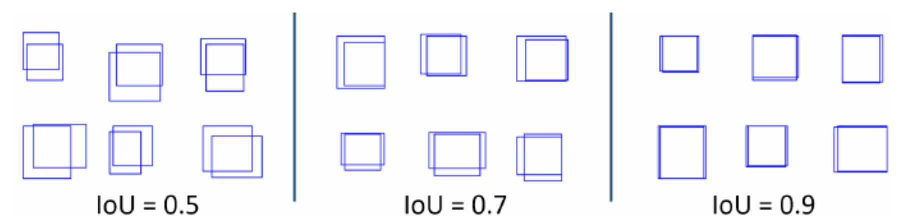


true positive (IOU>=0.5)

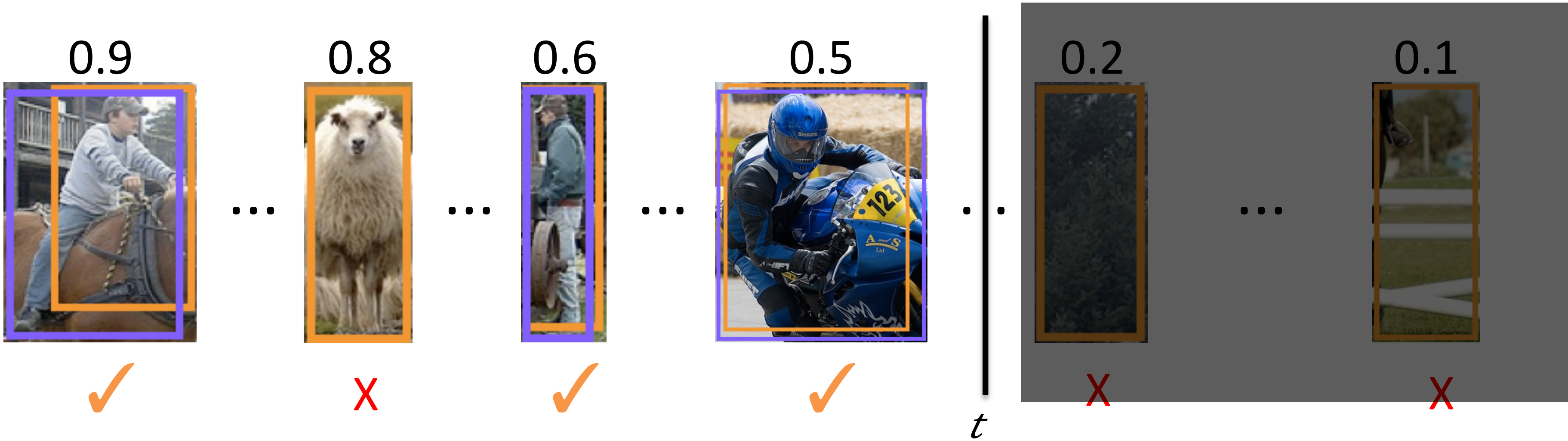
false positive (IOU<0.5)

Intersection Over Union (IOU)





Evaluation metric

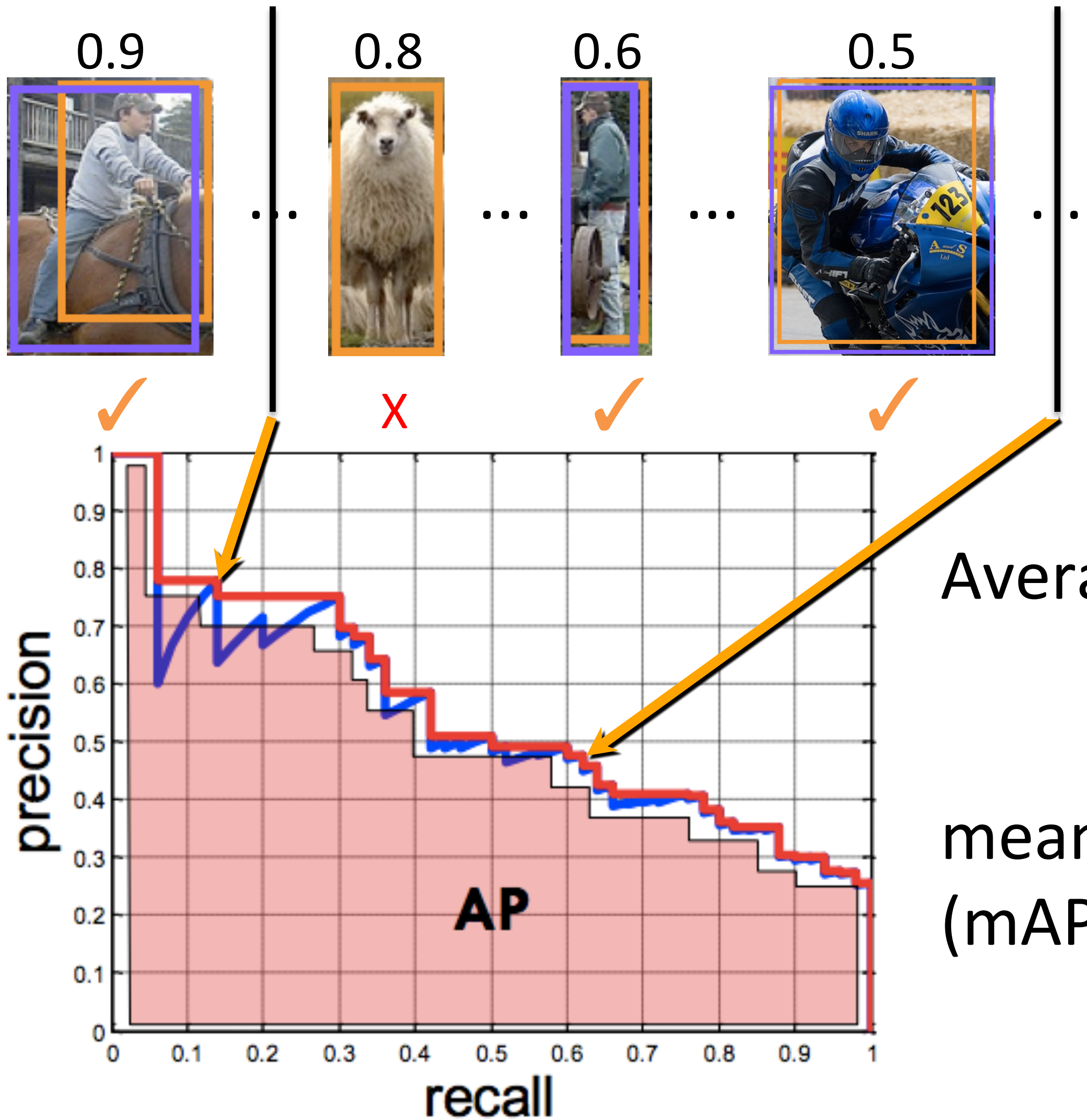


$$precision@t = \#true\ positives@t / \#true\ positives@t + \#false\ positives@t$$

✓ / (✓ + X)

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

Evaluation metric



Average Precision (AP)

mean AP over classes (mAP)

What about this one?

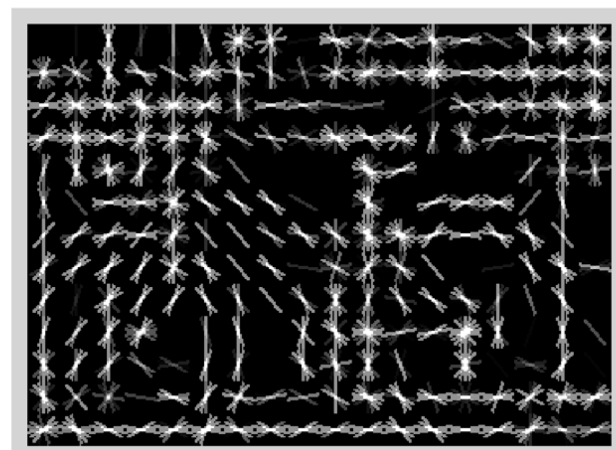
Can the model we trained for pedestrians detect the person in this image?

Specifying an object model

Statistical Template in Bounding Box

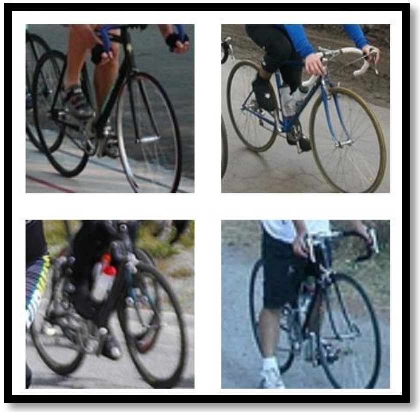
- Object is some (x,y,w,h) in image
- Features defined wrt bounding box coordinates

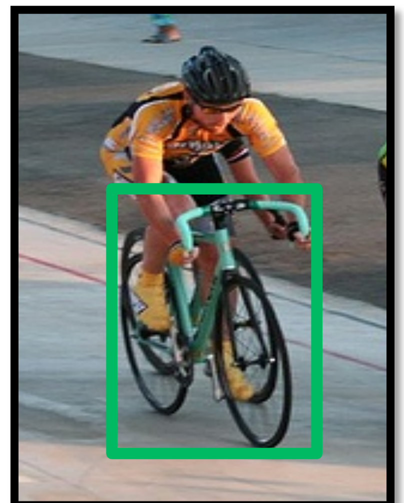
Image

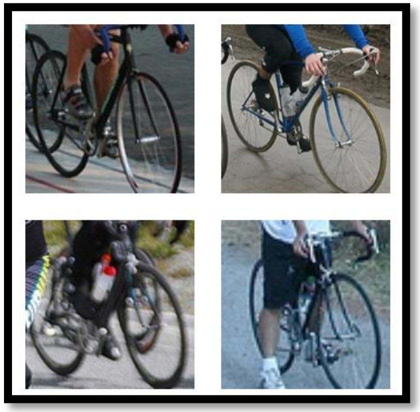
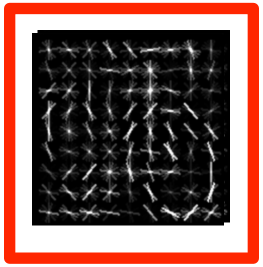
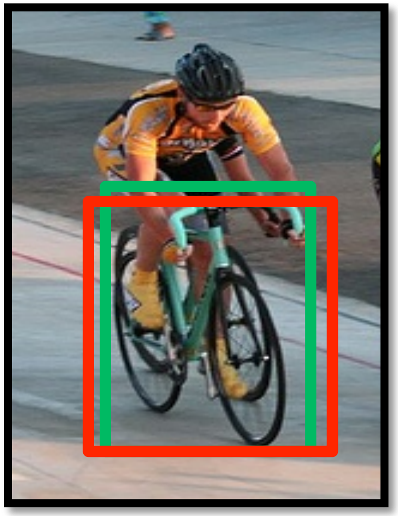


Template Visualization

Images from Felzenszwalb

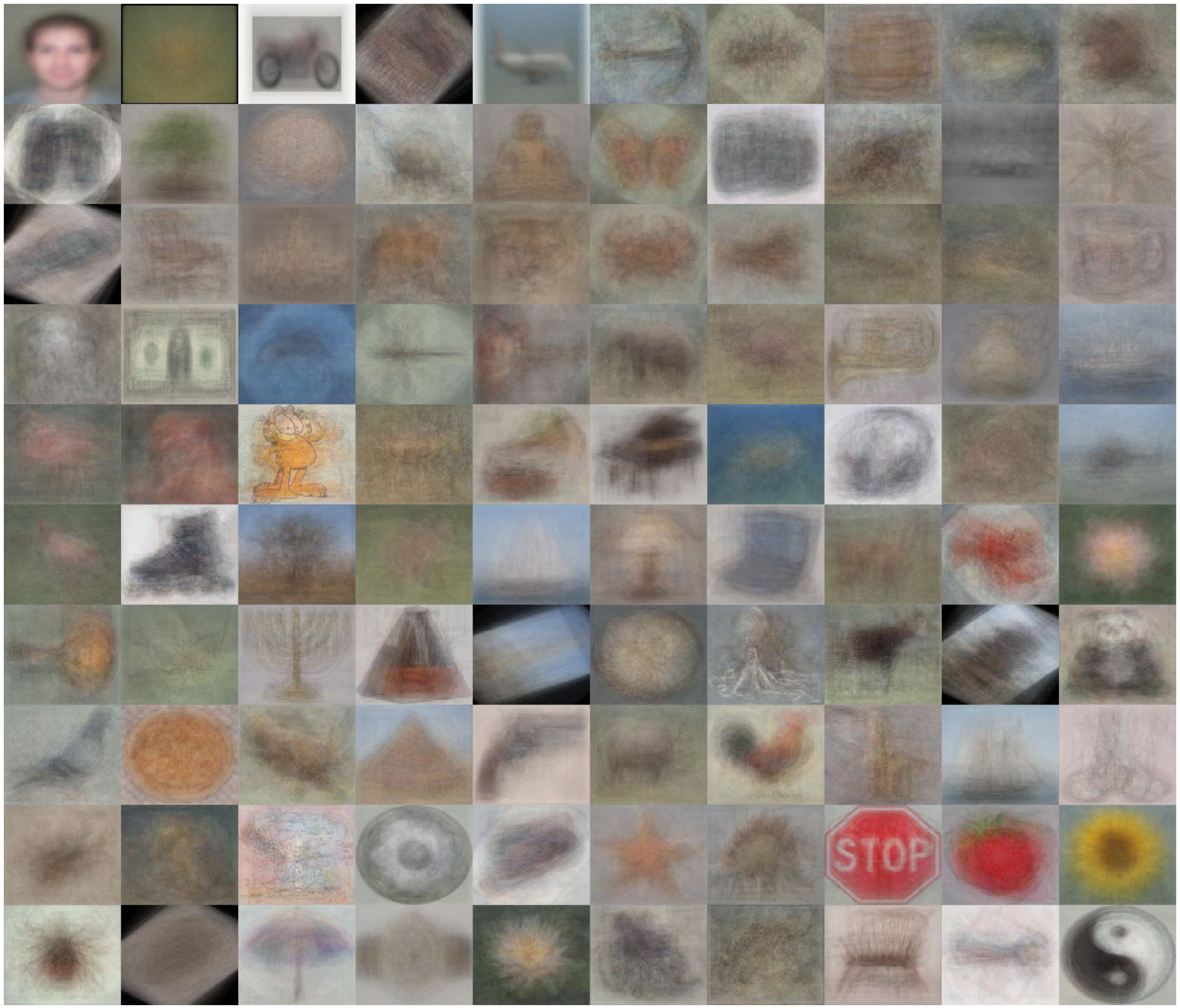




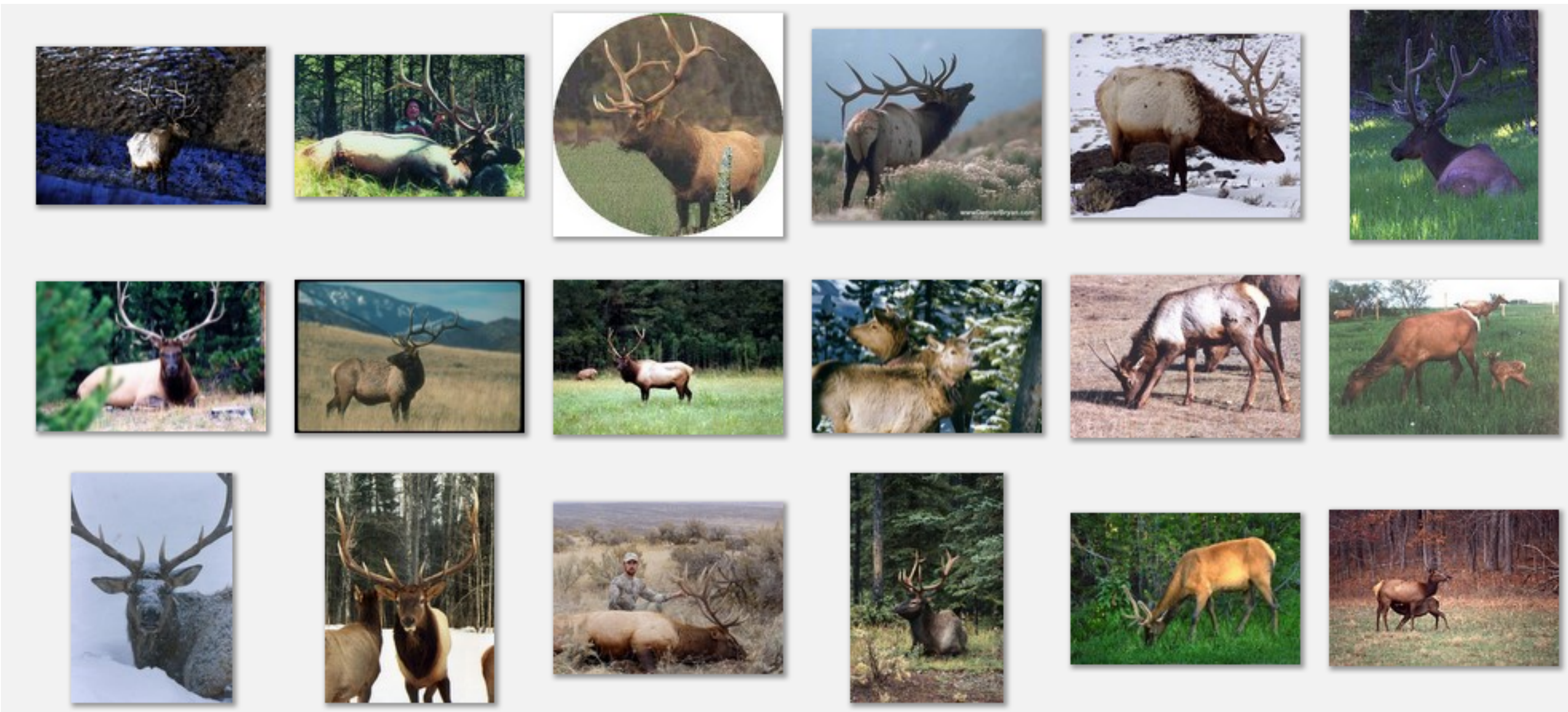

Octane

When do statistical templates make sense?



Caltech 101 Average Object Images

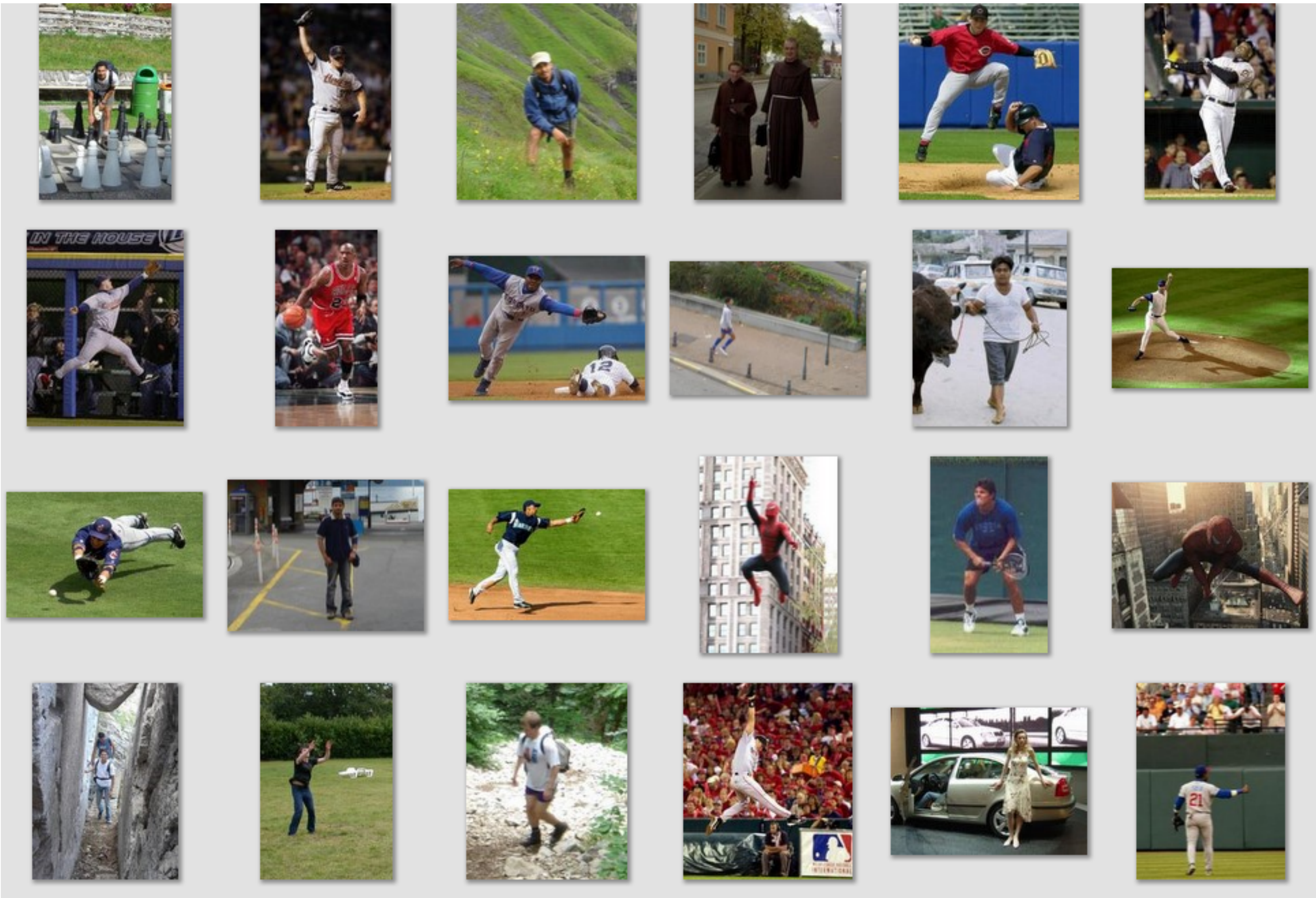
Deformable objects



Images from Caltech-256

Slide Credit: Duan Tran

Deformable objects



Images from D. Ramanan's dataset

Slide Credit: Duan Tran

Parts-based Models

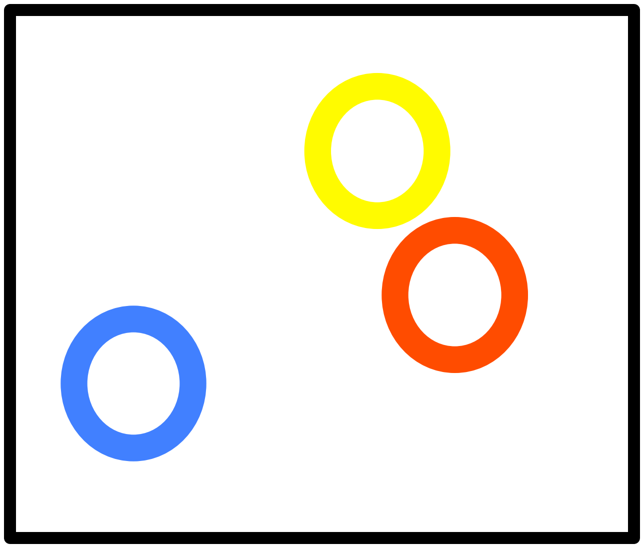
Define objects by collection of parts modeled by

1. Appearance
2. Spatial configuration

Slide credit: Rob Fergus

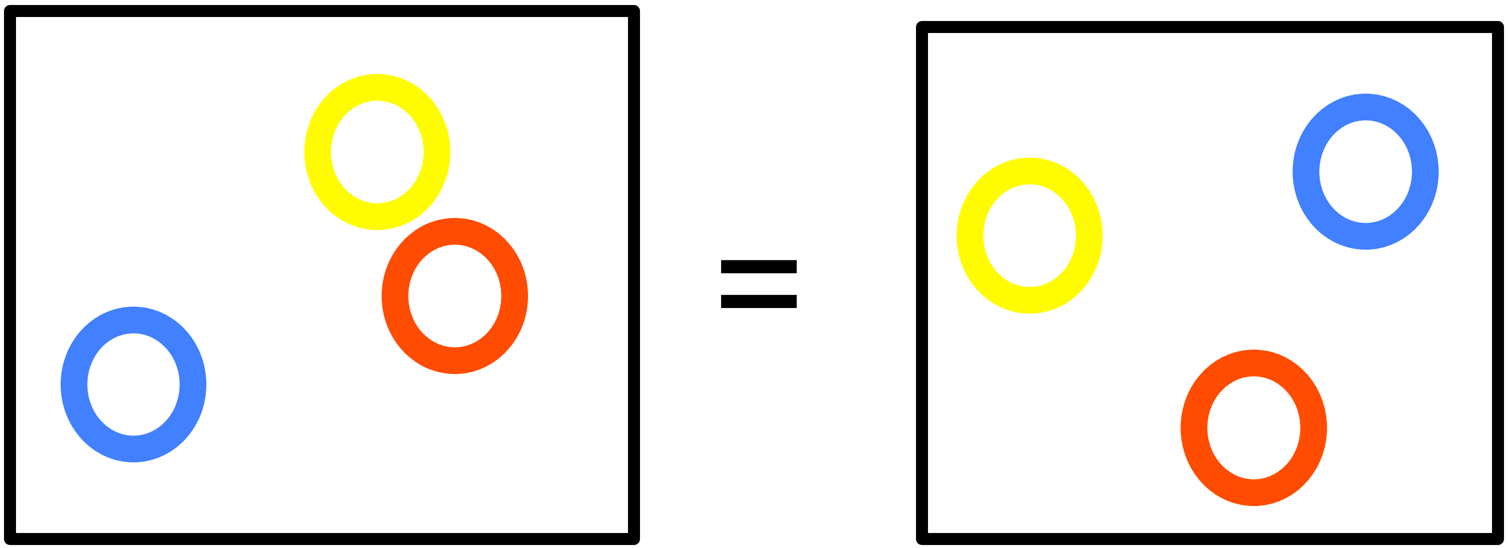
How to model spatial relations?

- One extreme: fixed template

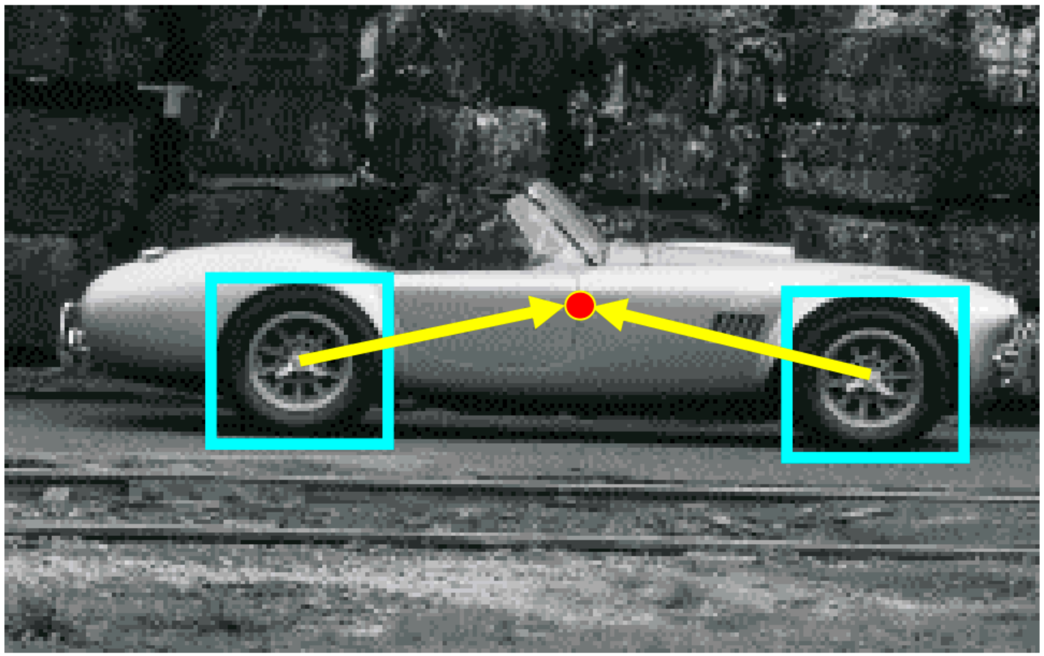


How to model spatial relations?

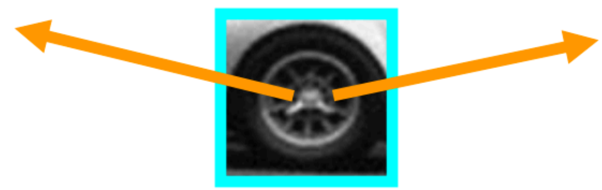
- Another extreme: bag of words



ISM:Implicit Shape Model for Detection



training image



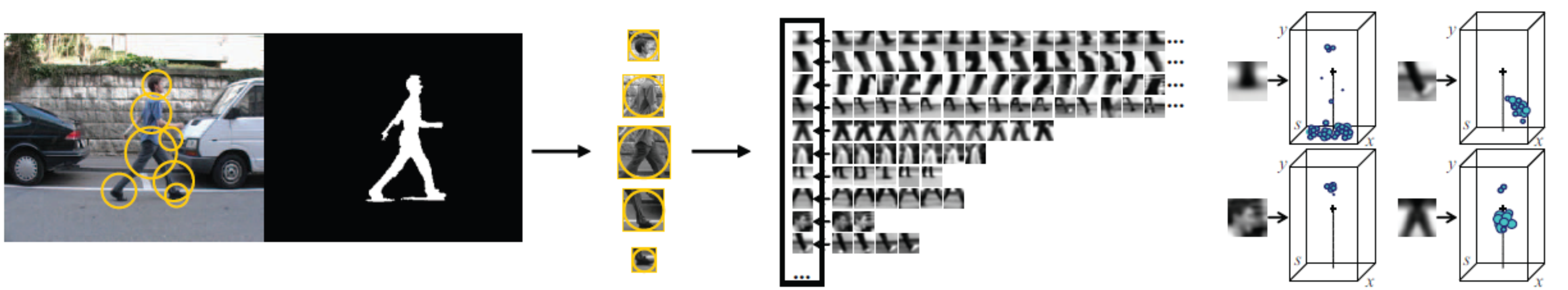
visual codeword with displacement vectors

Leibe. et al [ECCV 2004]

ISM: Implicit Shape Model

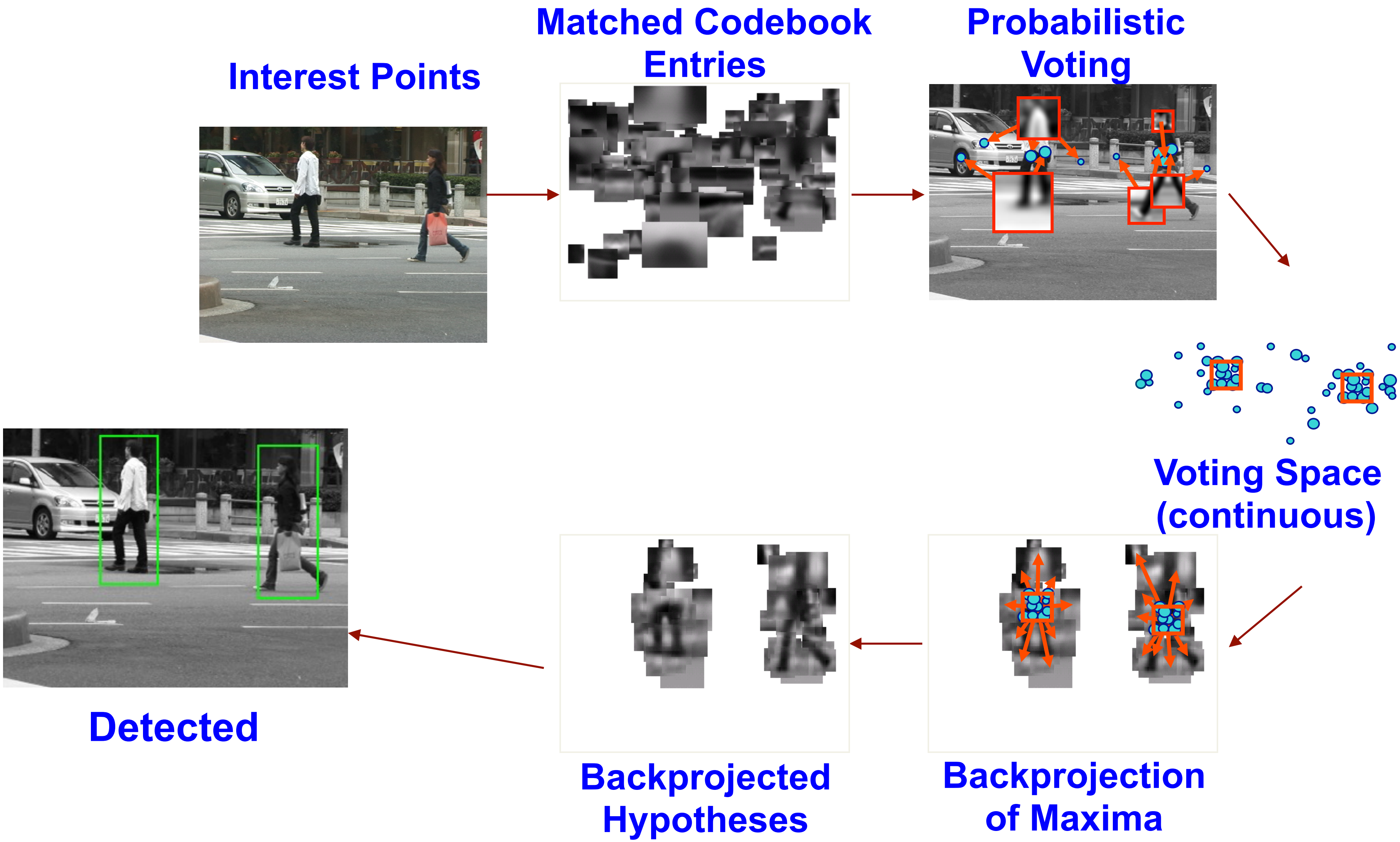
Training overview

- 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



Training Images (+Reference Segmentations) — Local Features — Appearance Codebook (Cluster Centers) — Spatial Occurrence Distributions (non-parametric)

Implicit Shape Model for Detection



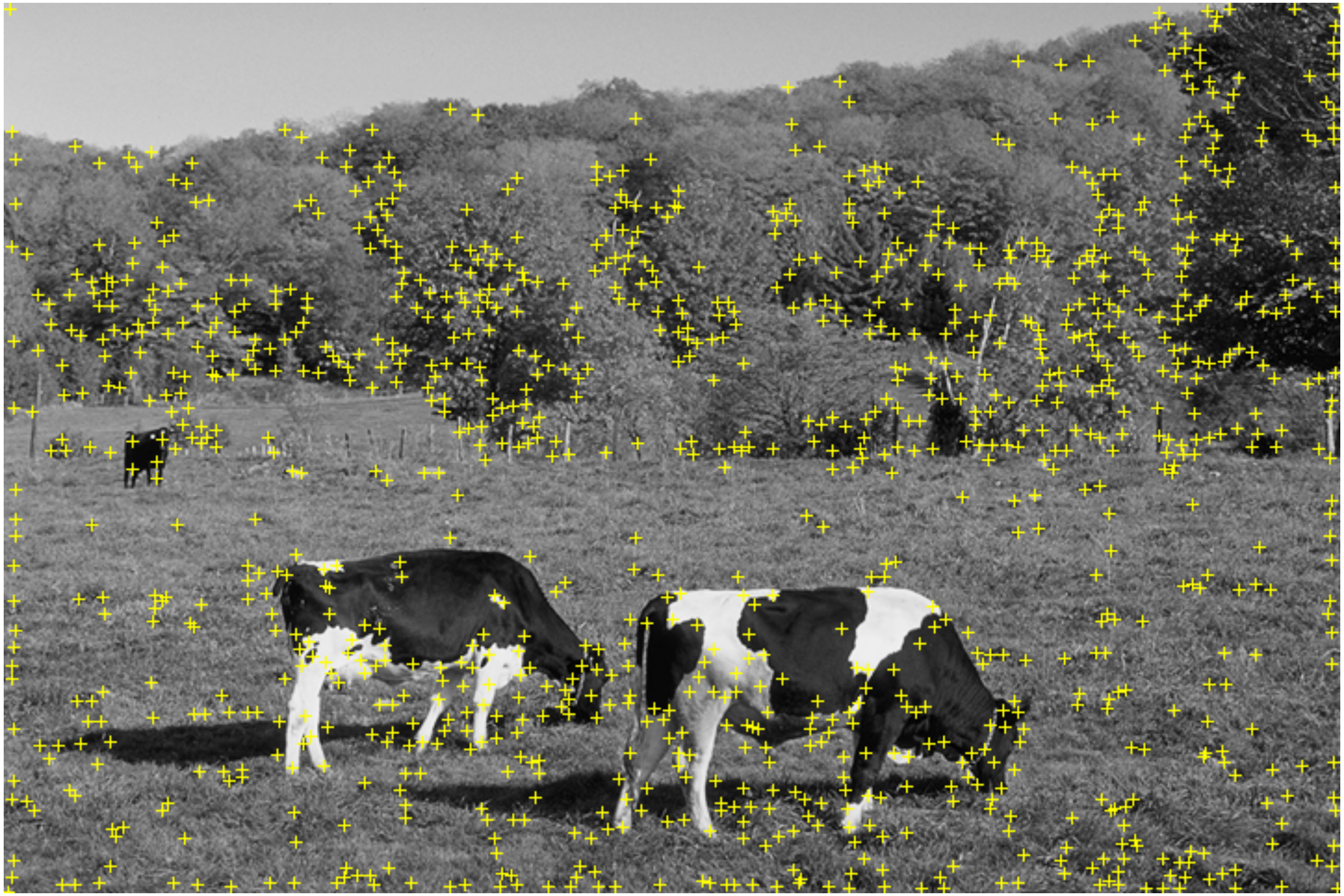
Liebe and Schiele, 2003, 2005

Example: Results on Cows

[redacted]

K. Grauman, B. Leibe

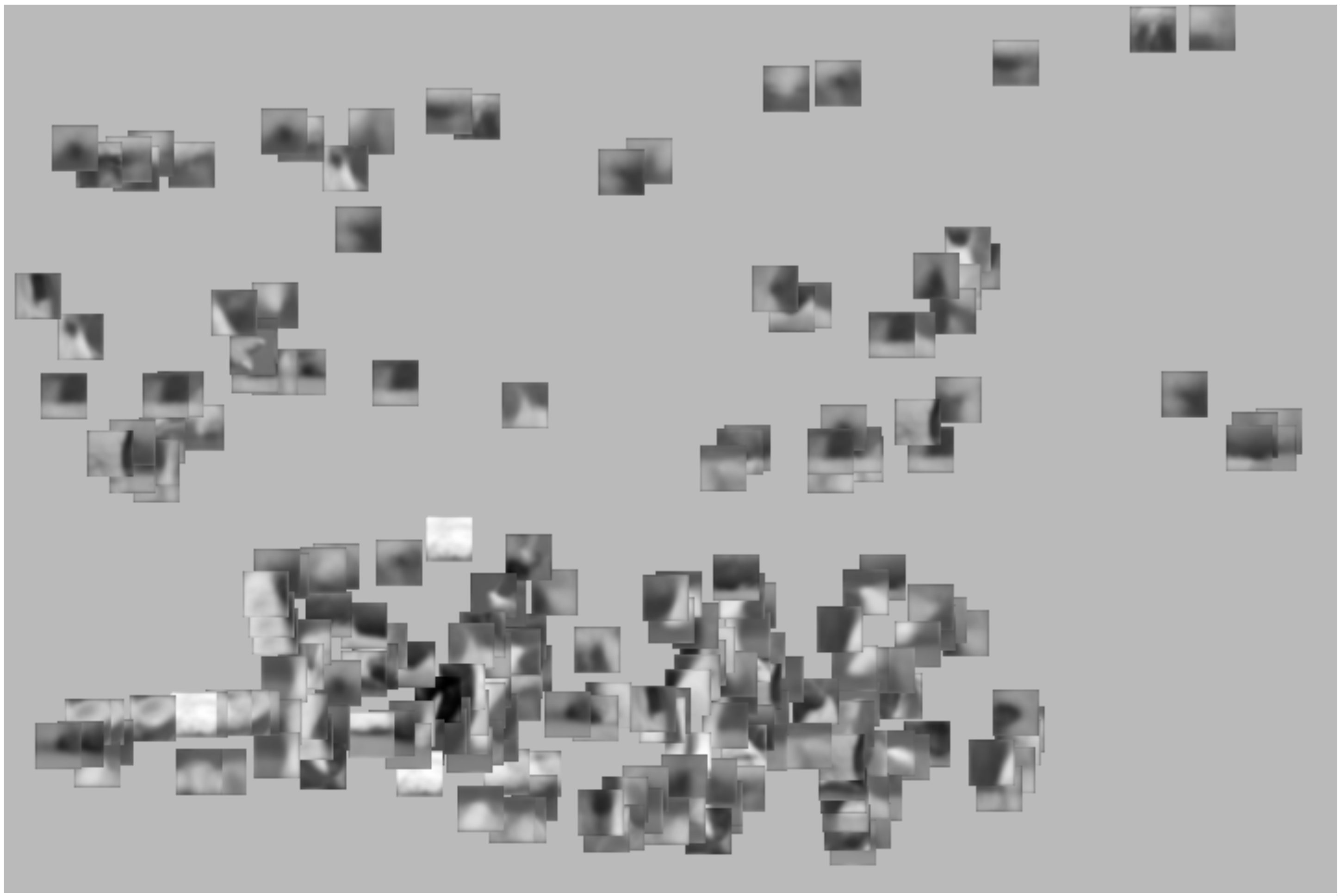
Example: Results on Cows



[redacted]

K. Grauman, B. Leibe

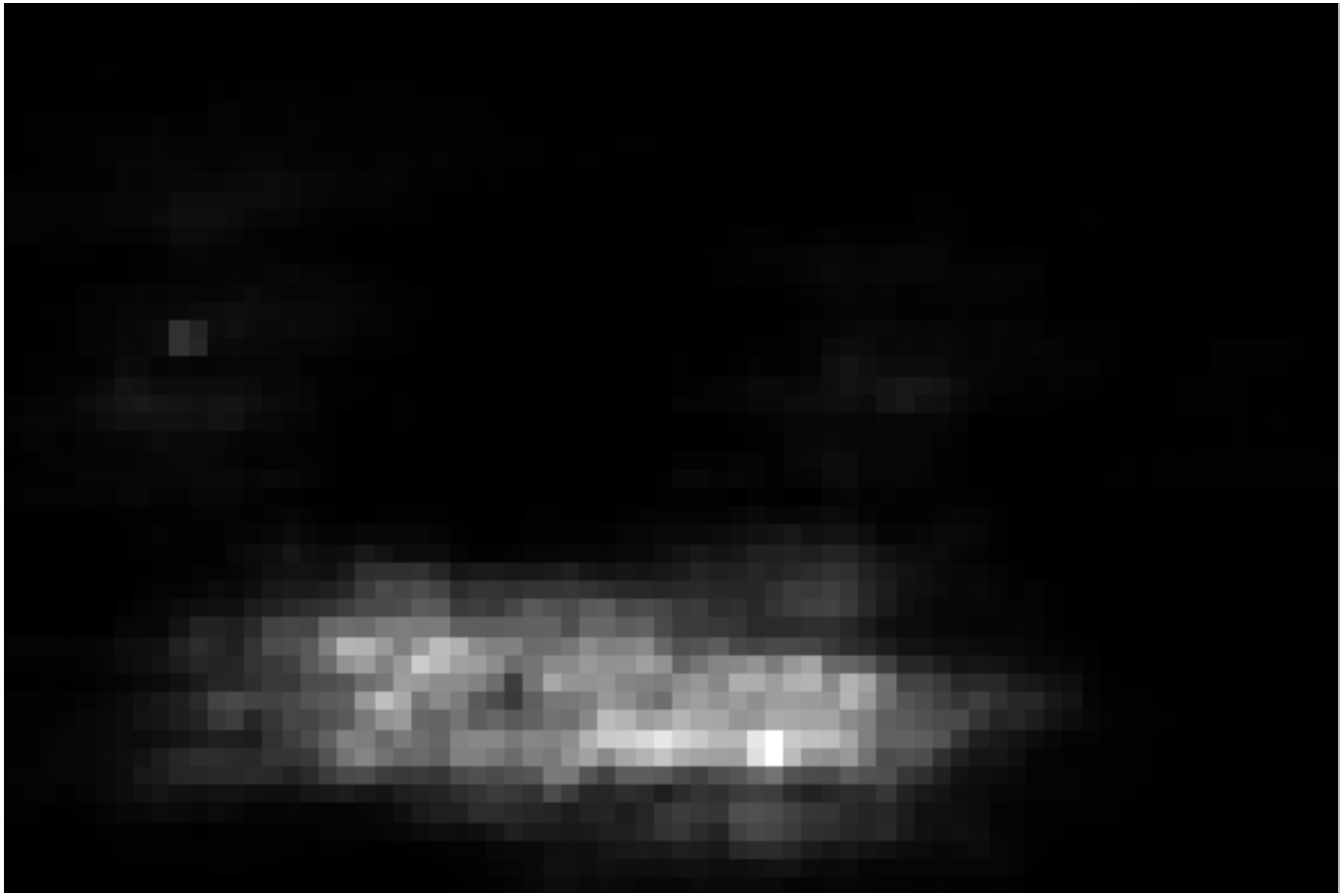
Example: Results on Cows



[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

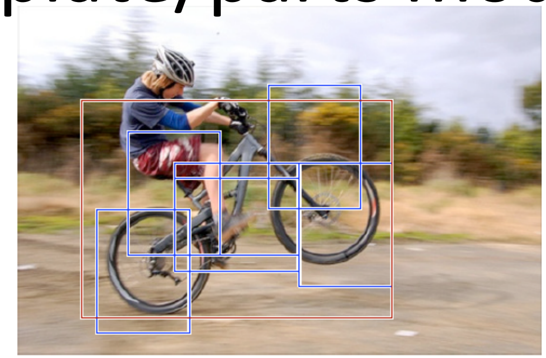
- Qualitative Performance
  - Robust to clutter, occlusion, noise, low contrast

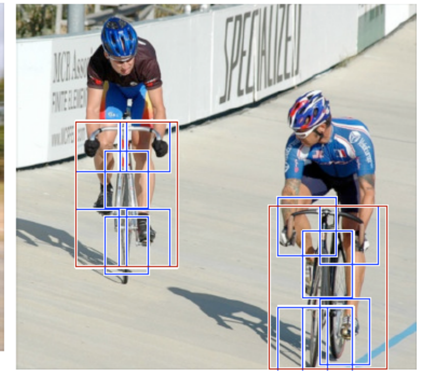
K. Grauman, B. Leibe

Explicit Models

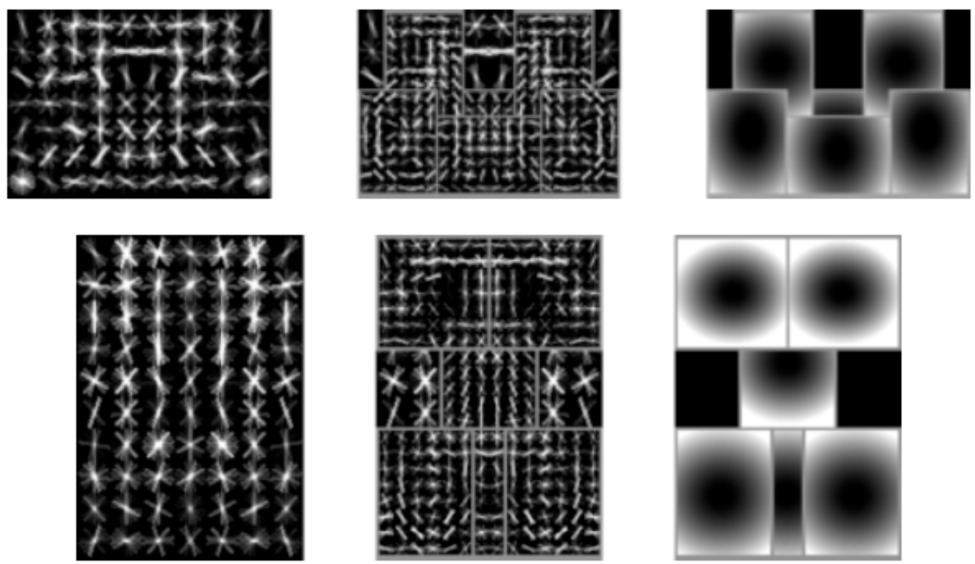
Hybrid template/parts model

Detections





Template Visualization



root filters
coarse resolution

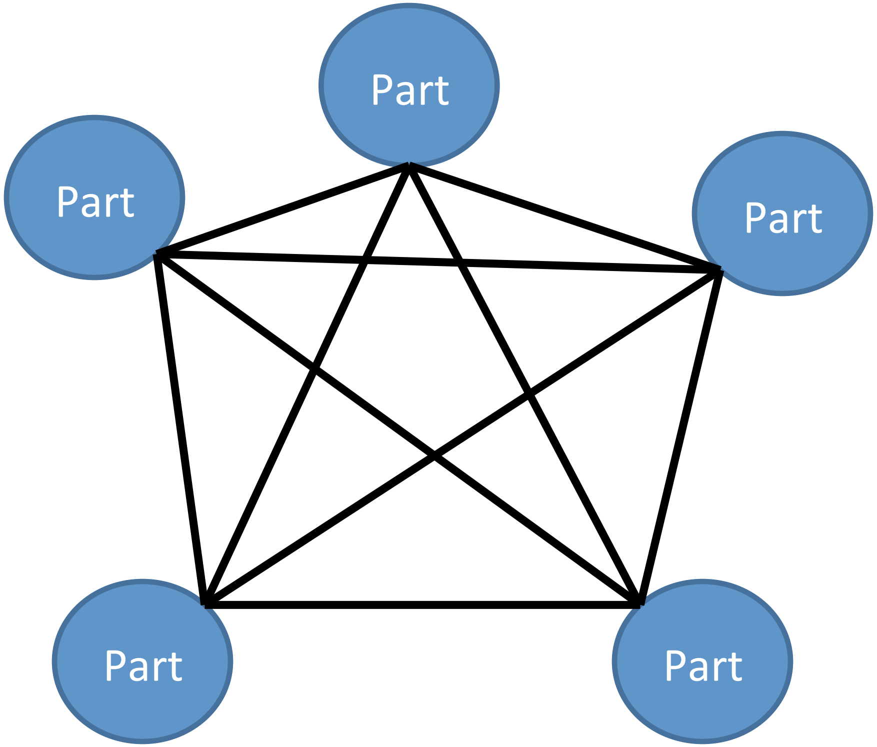
part filters
finer resolution

deformation
models

Felzenszwalb et al. 2008

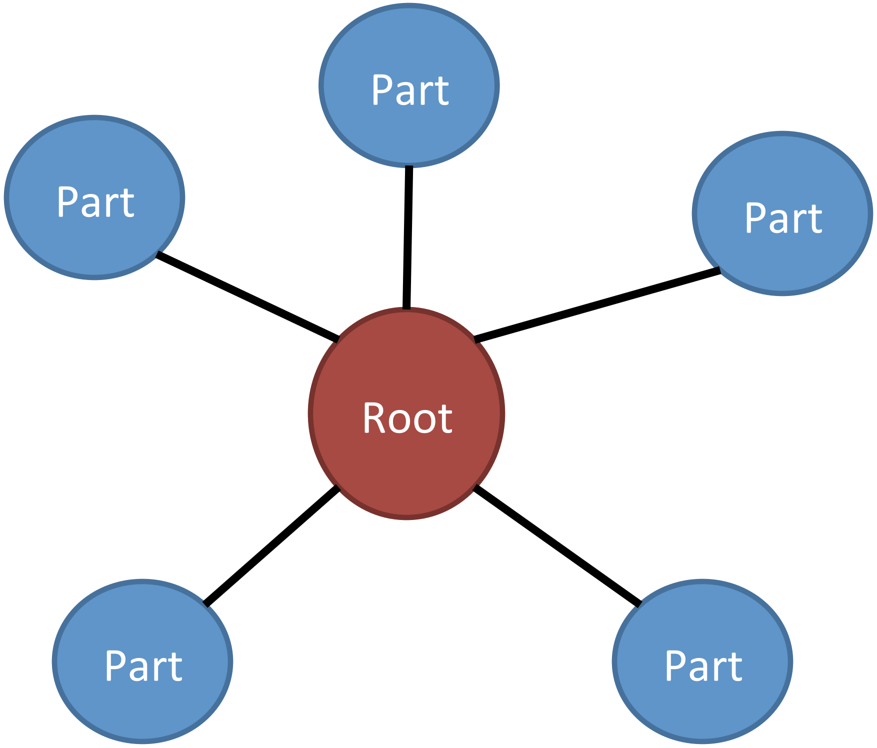
How to model spatial relations?

- Explicit Models
- Too expensive



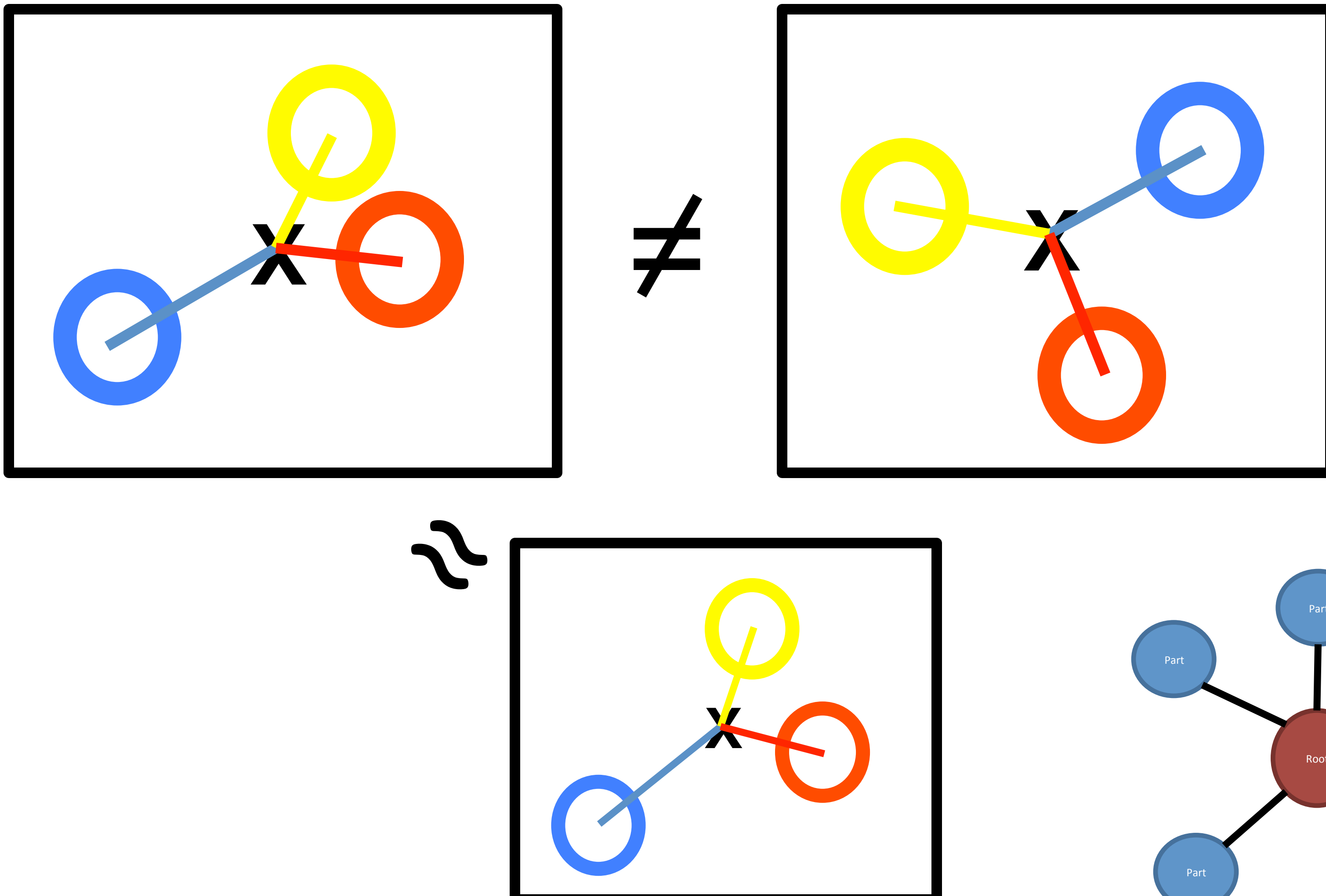
How to model spatial relations?

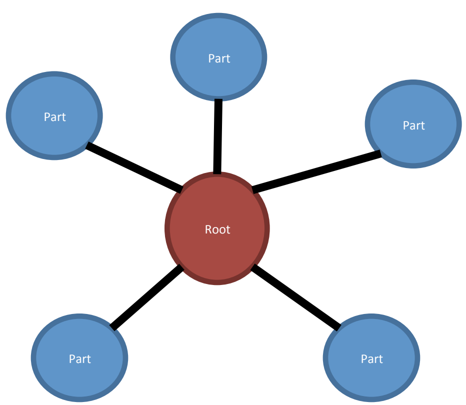
- Star-shaped model



How to model spatial relations?

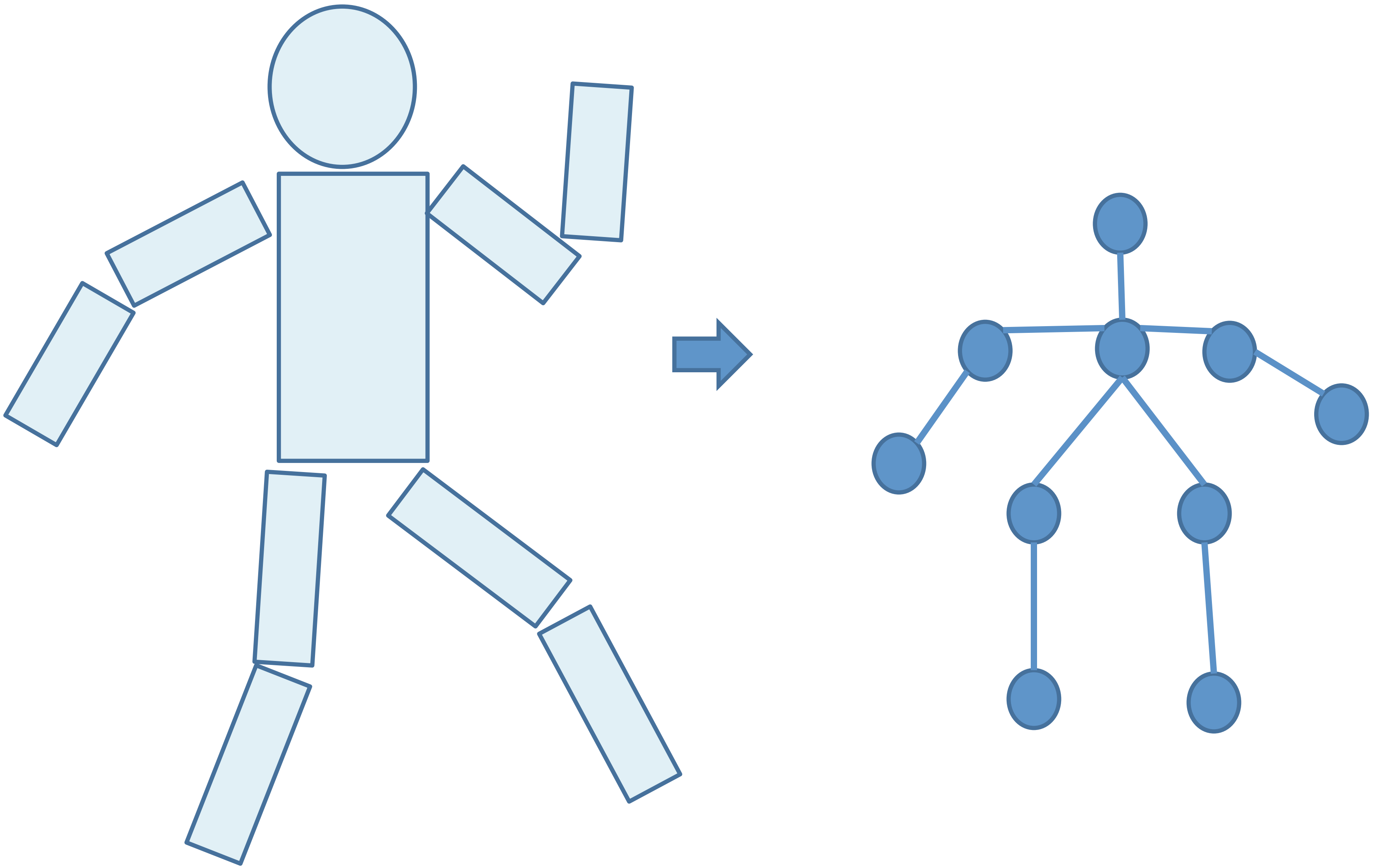
- Star-shaped model





How to model spatial relations?

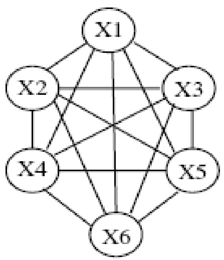
- Tree-shaped model



How to model spatial relations?

- Many others...

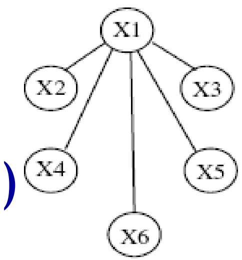
$O(N^6)$



a) Constellation

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

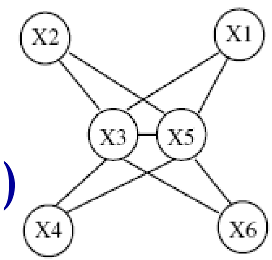
$O(N^2)$



b) Star shape

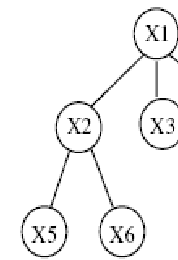
Leibe et al. '04, '08
Crandall et al. '05
Fergus et al. '05

$O(N^3)$



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

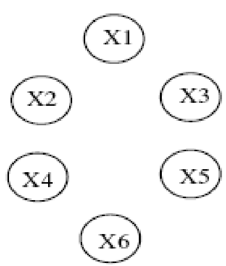
Crandall et al. '05



$O(N^2)$

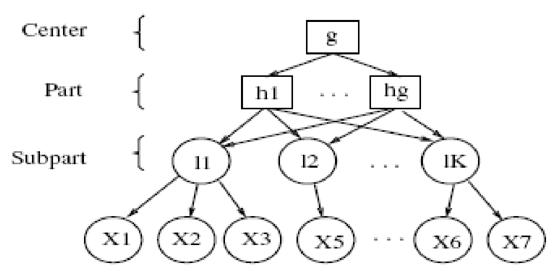
d) Tree

Felzenszwalb &
Huttenlocher '05



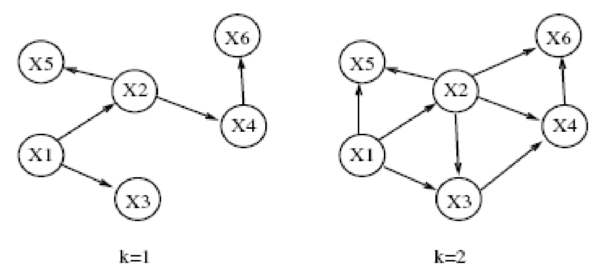
e) Bag of features

Csurka '04
Vasconcelos '00



f) Hierarchy

Bouchard & Triggs '05

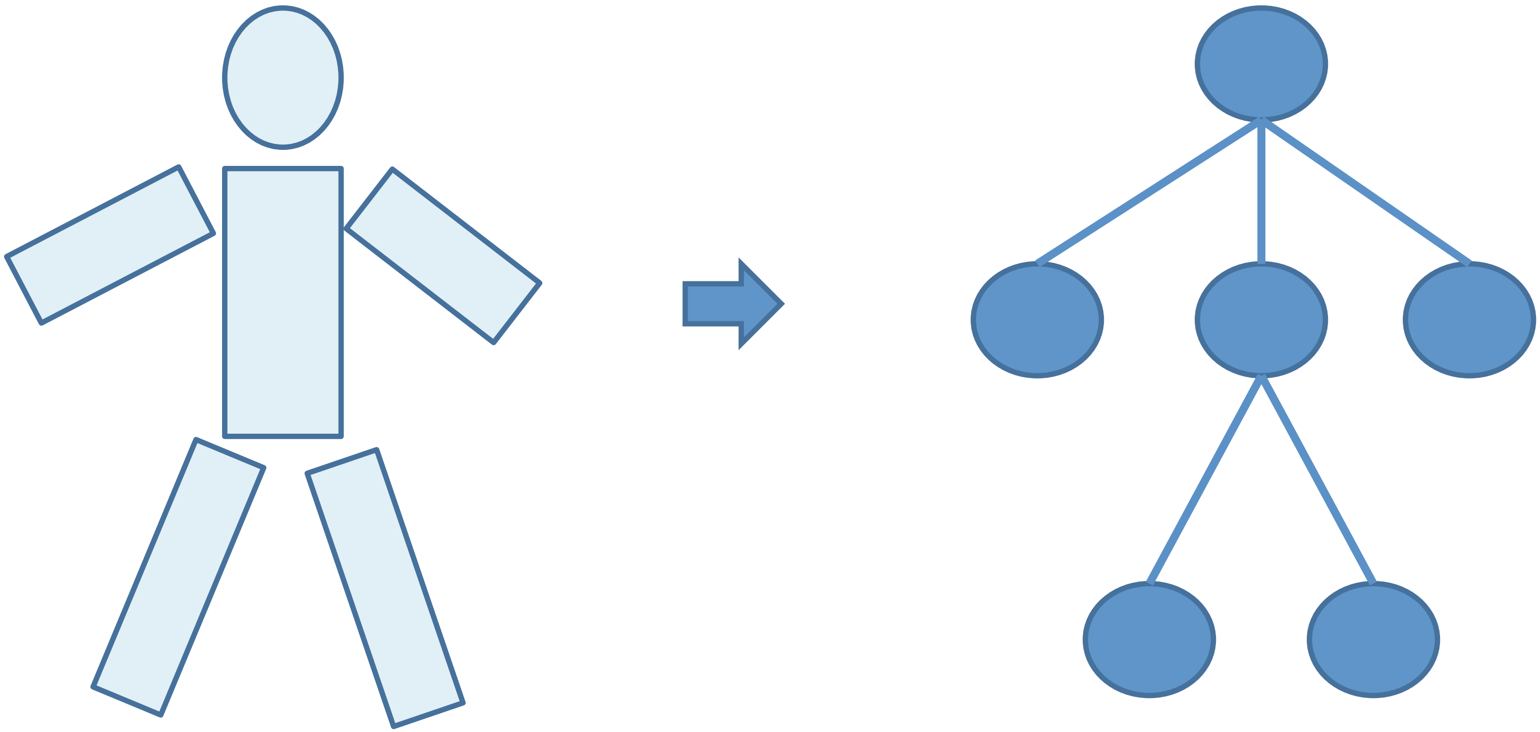


g) Sparse flexible model

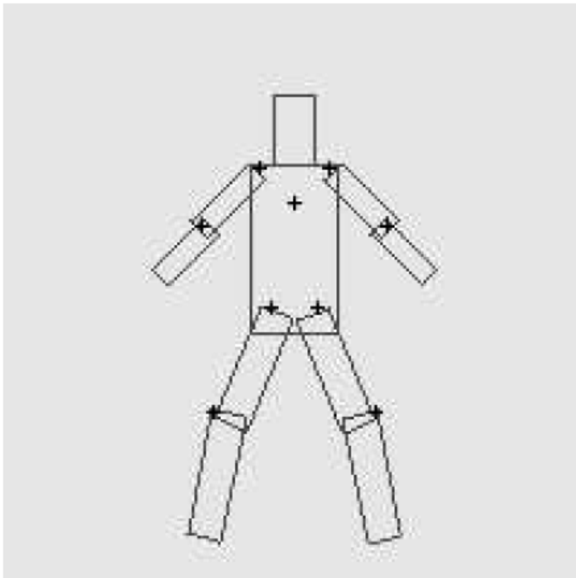
Carneiro & Lowe '06

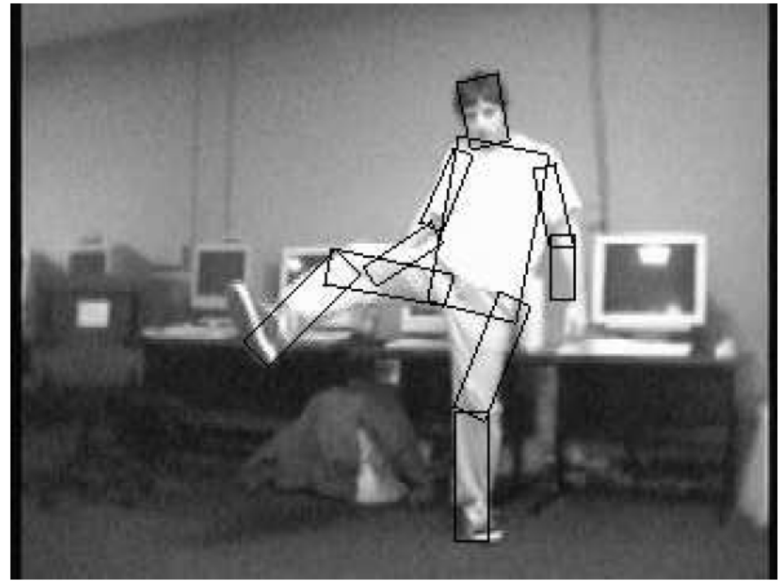
from [Carneiro & Lowe, ECCV'06]

Tree-shaped model



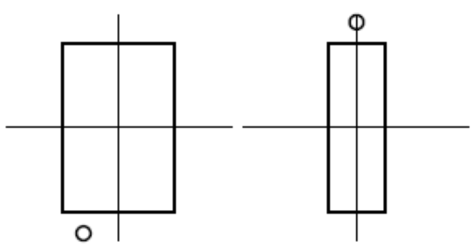
Pictorial Structures Model



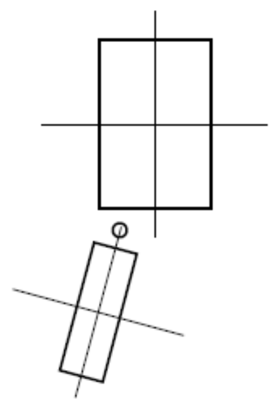


Part = oriented rectangle

Spatial model = relative size/orientation



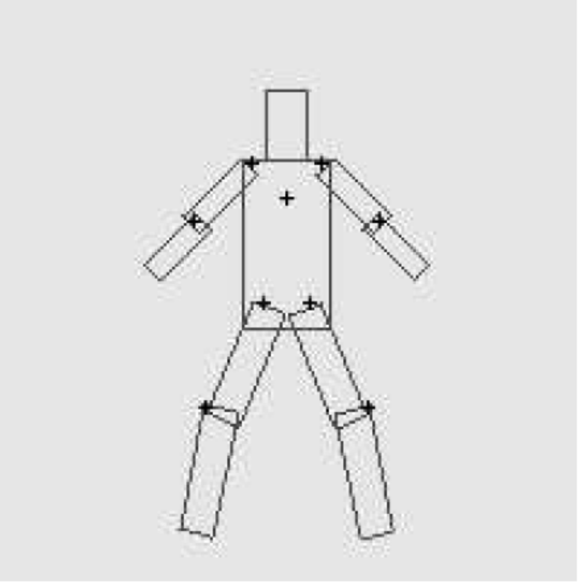
a



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)

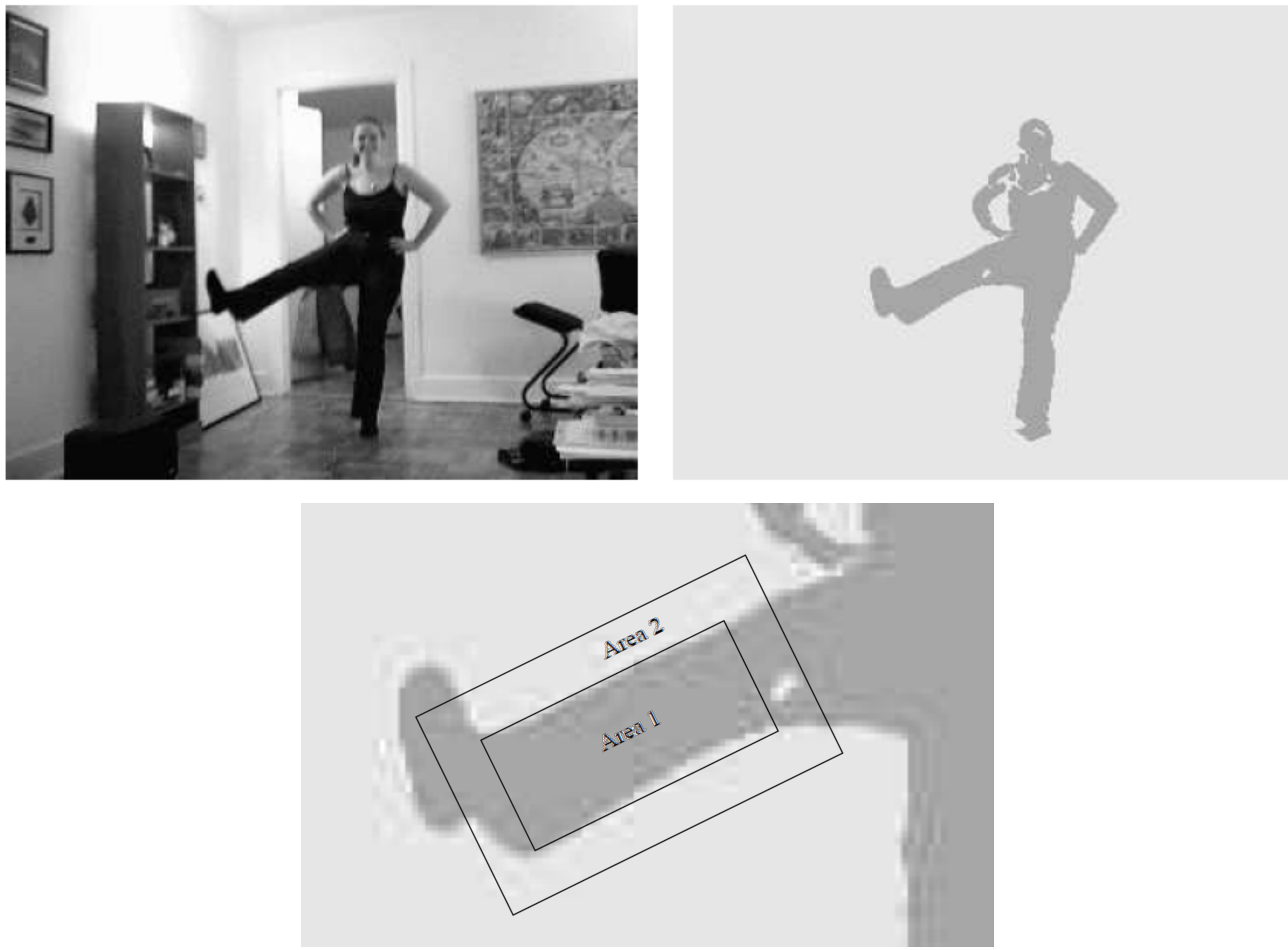
$$P(L|I,\theta) \propto \left(\prod_{i=1}^{n} p(I|l_i, u_i) \prod_{(v_i,v_j)\in E} p(l_i, l_j|c_{ij}) \right)$$

Appearance likelihood

Geometry likelihood

Part representation

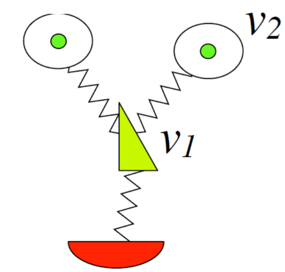
- Background subtraction



Pictorial structures model

Optimization is tricky but can be efficient

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

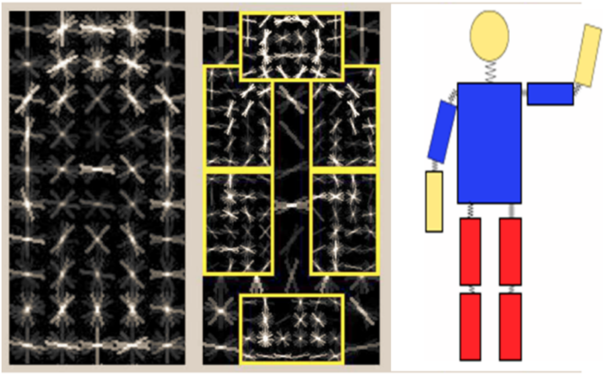


- For each l_1, find best l_2:

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

- 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

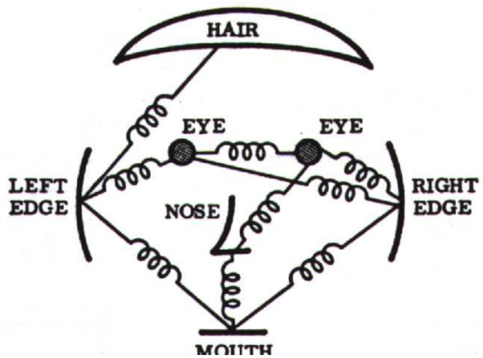


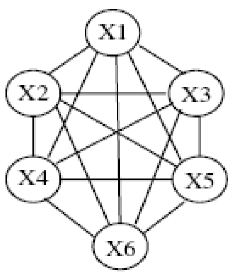
- Model is represented by a graph $G = (V, E)$.
  - $V = \{v_1, \ldots, v_n\}$ are the parts.
  - $(v_i, v_j) \in E$ indicates a connection between parts.
- $m_i(l_i)$ is the cost of placing part i at location l_i.
- $d_{ij}(l_i, l_j)$ is a deformation cost.
- Optimal location for object is given by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

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

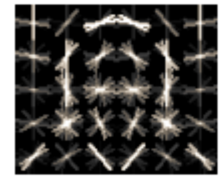
$$L^* = \operatorname*{argmin}_L \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right)$$

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

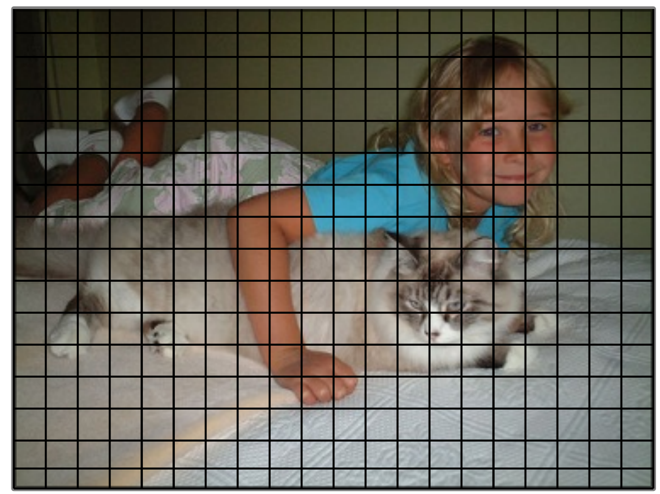




a) Constellation [13]



head filter



Complexity $O(h^n)$

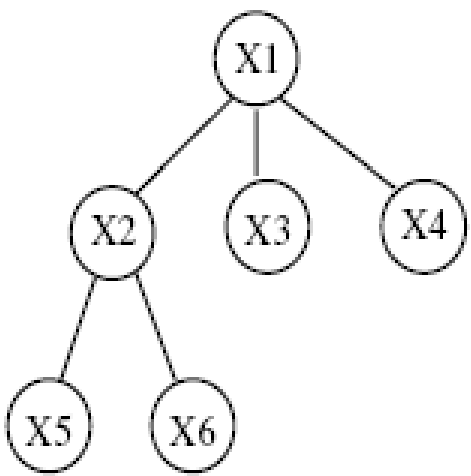
h: number of possible part placements

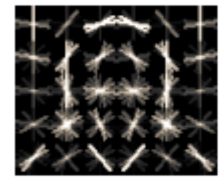
n: number of parts

Efficient minimization

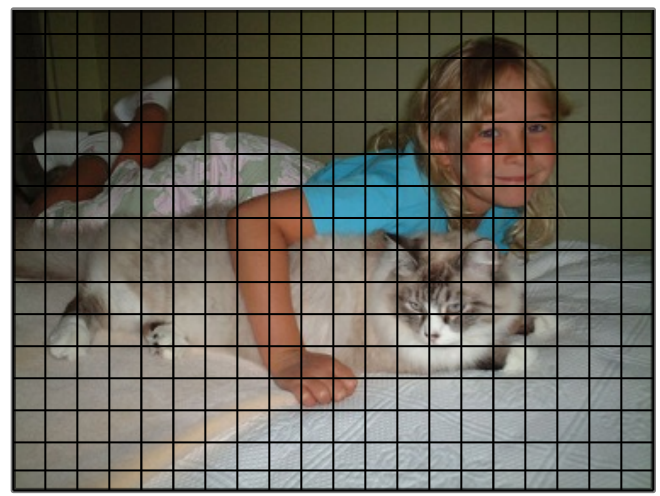
$$L^* = \underset{L}{\text{argmin}} \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right)$$

- 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.





head filter



Complexity $O(nh^2)$

Efficient minimization

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

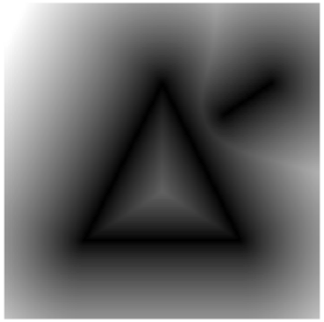
- n parts and h locations gives h^n configurations.
- If graph is a tree we can use dynamic programming.
  - $O(nh^2)$, much better but still slow.
- If $d_{ij}(l_i, l_j) = ||T_{ij}(l_i) - T_{ji}(l_j)||^2$ can use DT.
  - $O(nh)$, as good as matching each part separately!!

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

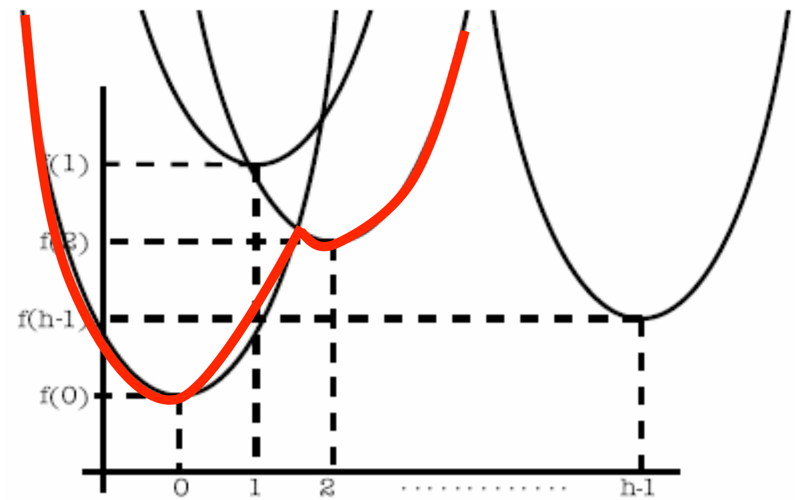
Given a function $f : \mathcal{G} \to \mathbb{R}$,

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

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

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

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.

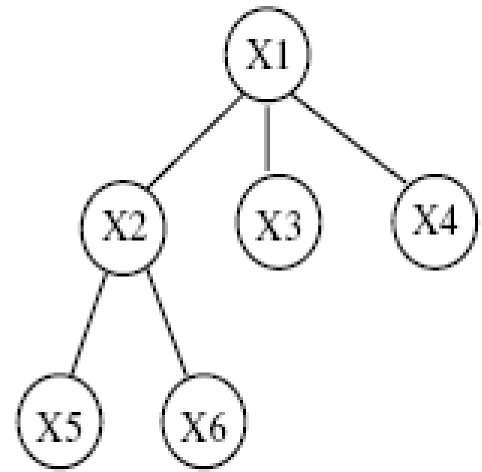
The 2D case is "separable", it can be solved by sequential 1D transformations along rows and columns of the grid.

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

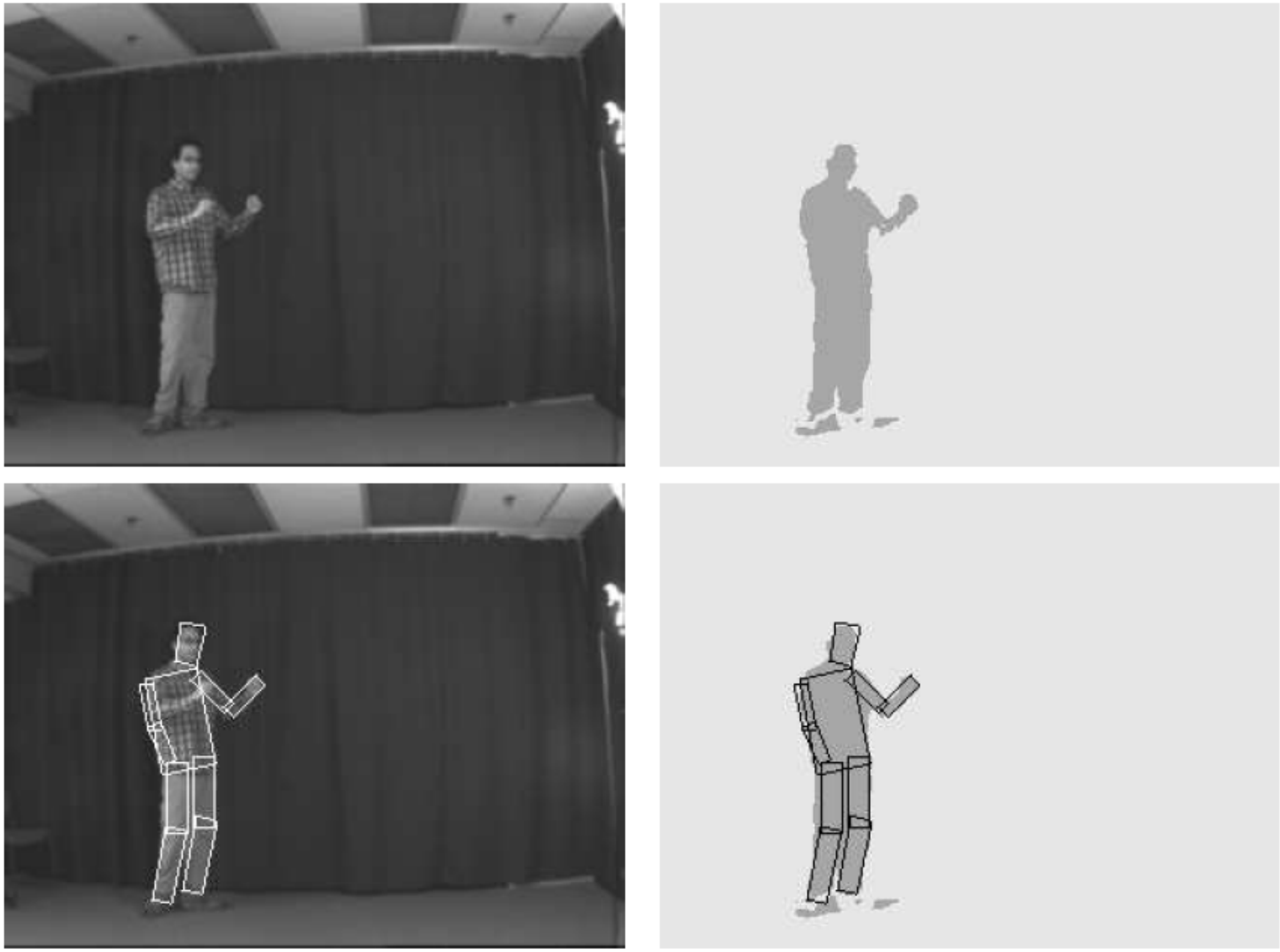
Pictorial Structures: Summary

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

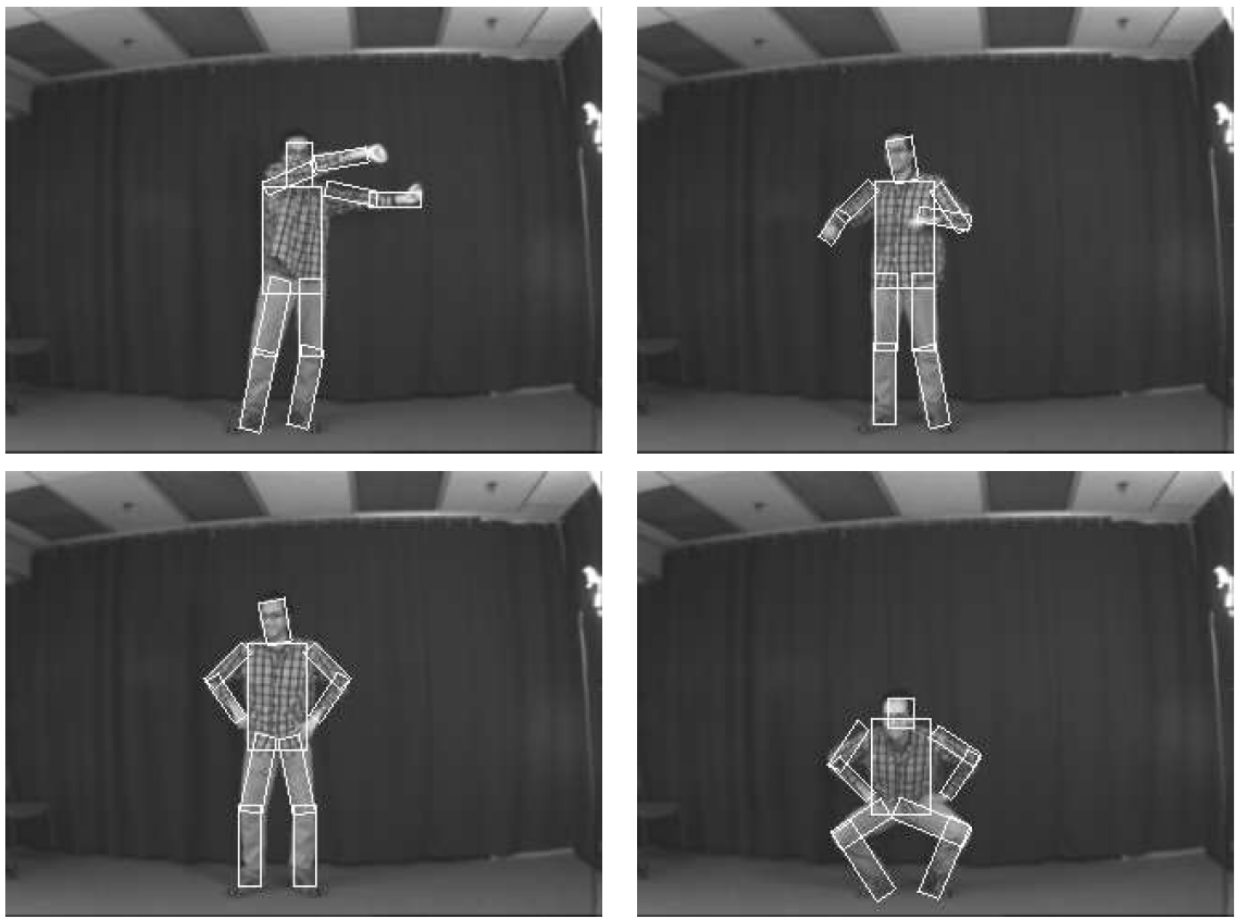
$$d_{ij}(l_i, l_j) = ||T_{ij}(l_i) - T_{ji}(l_j)||^2$$



Results for person matching



Results for person matching



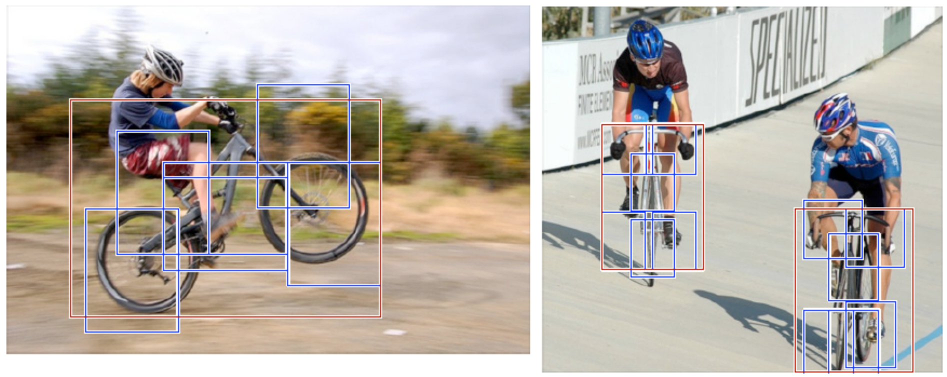
Enhanced pictorial structures

BMVC 2009

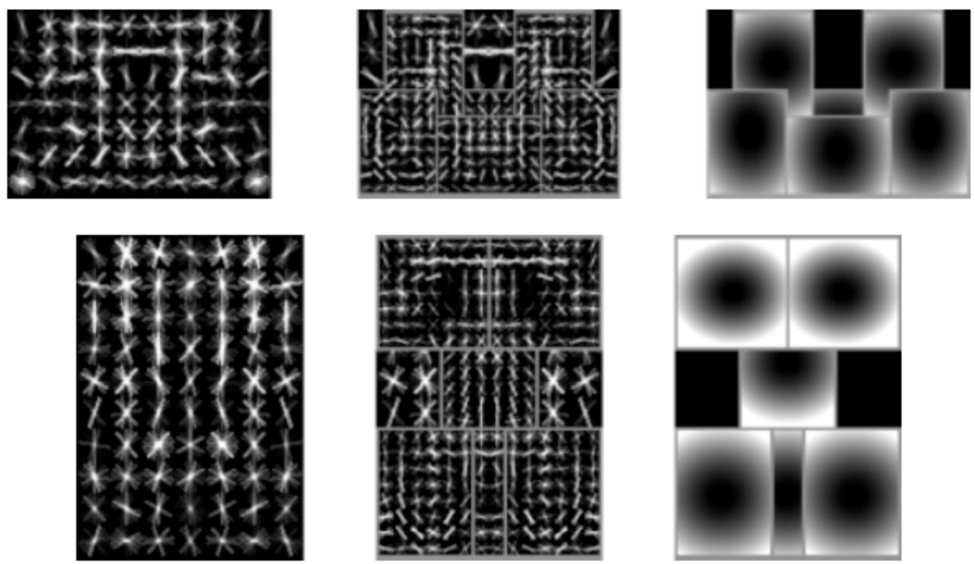
Deformable Latent Parts Model

Useful parts discovered during training

Detections



Template Visualization



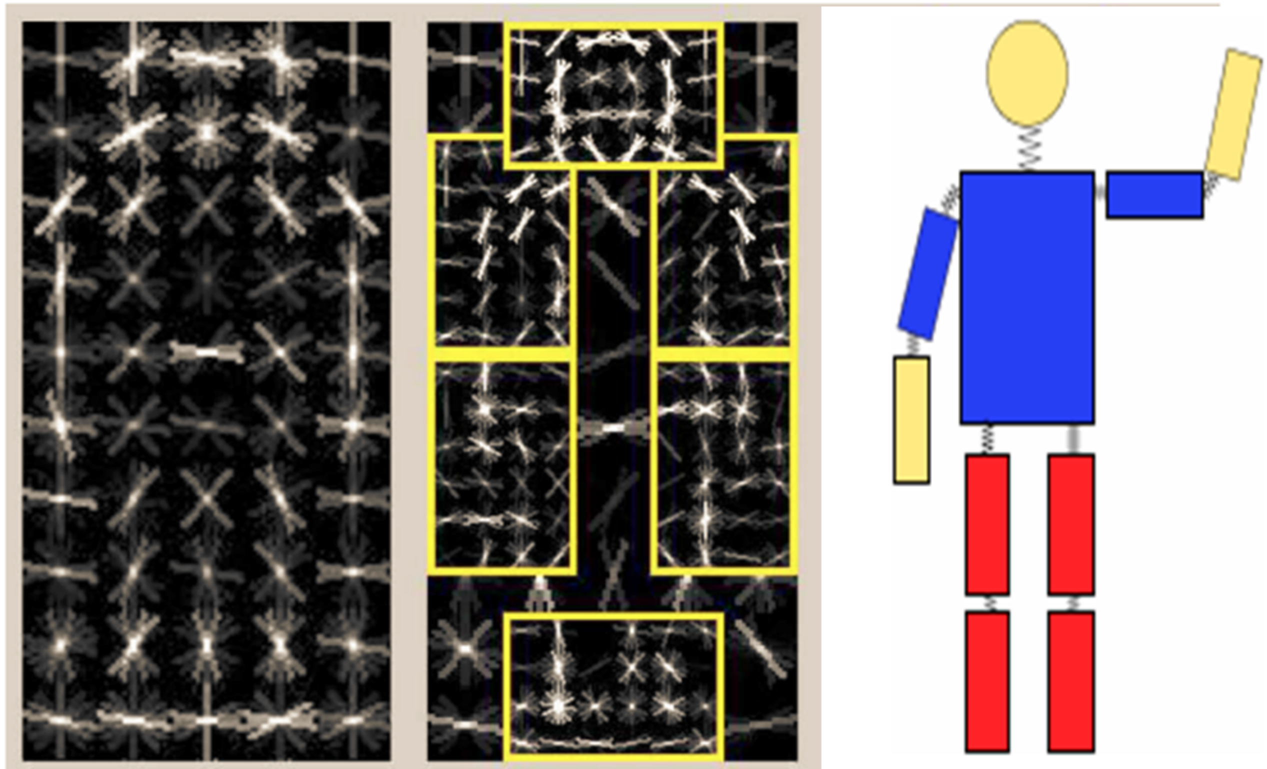
root filters
coarse resolution

part filters
finer resolution

deformation
models

Felzenszwalb et al. 2008

Deformable Part Models



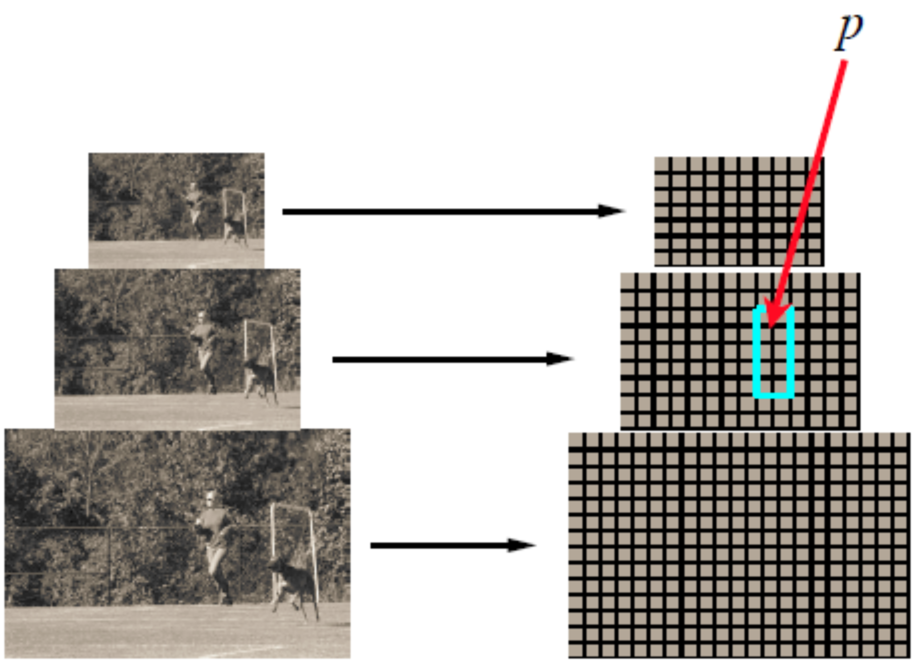
Root filter | Part filter | Quadratic

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

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

HOG Filters

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



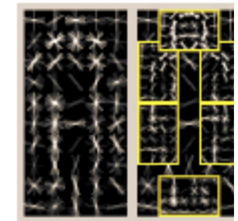
HOG pyramid H

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

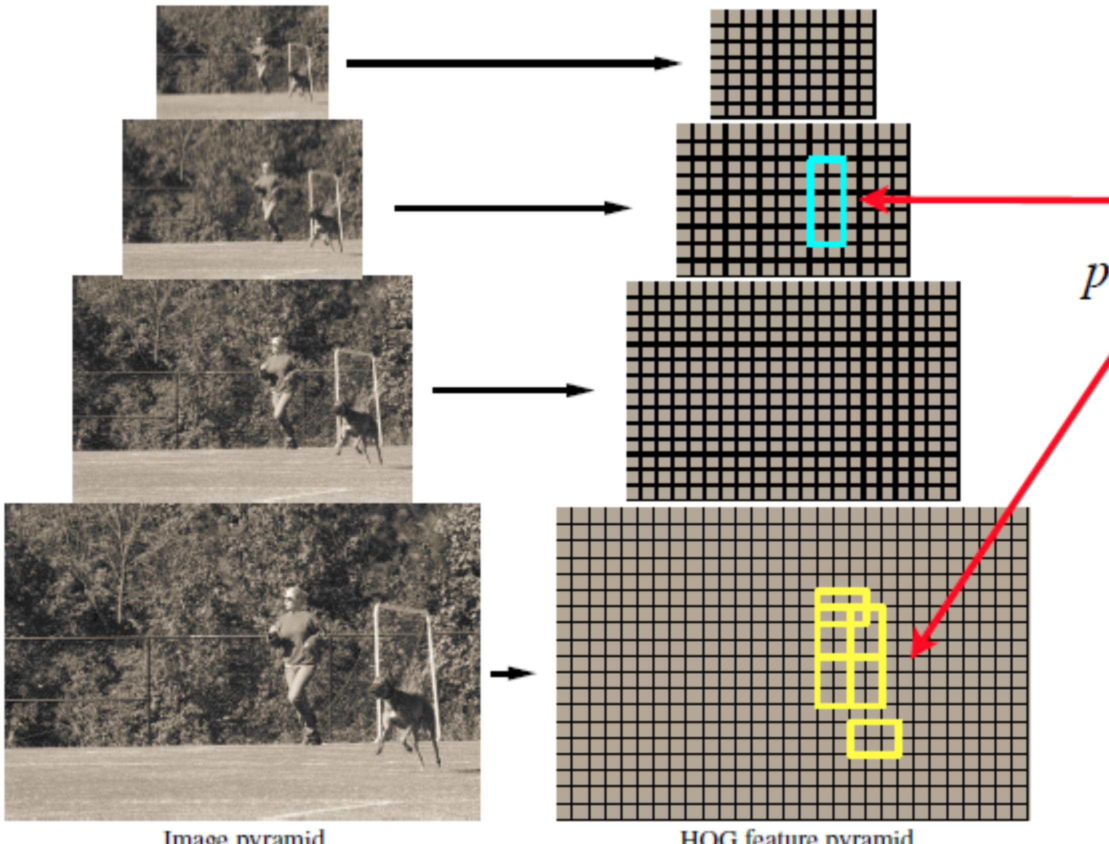
Object hypothesis



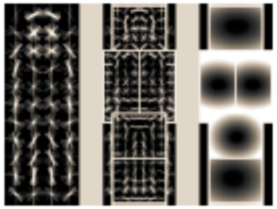
$z = (p_0, \ldots, p_n)$

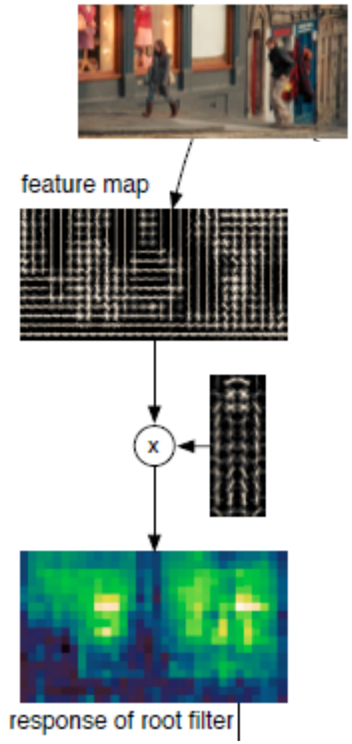
p_0 : location of root

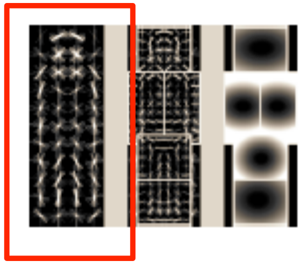
$p_1, \ldots, p_n$: location of parts

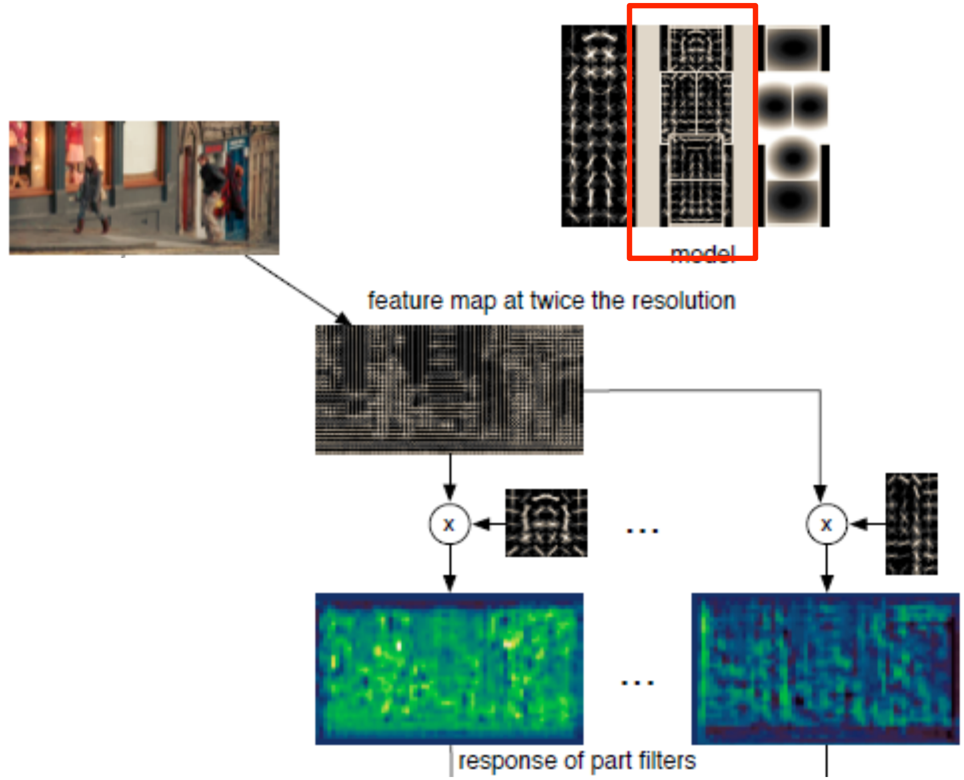


Multiscale model captures features at two-resolutions



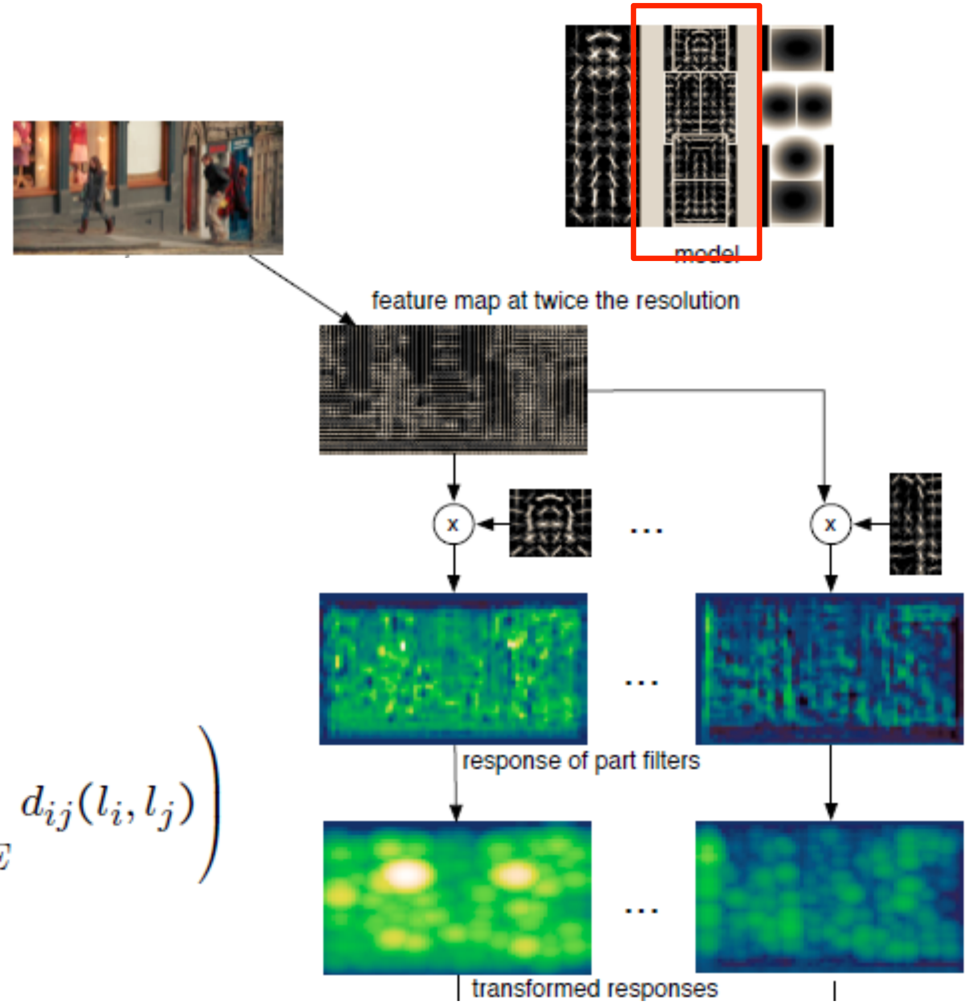

feature map
x
response of root filter

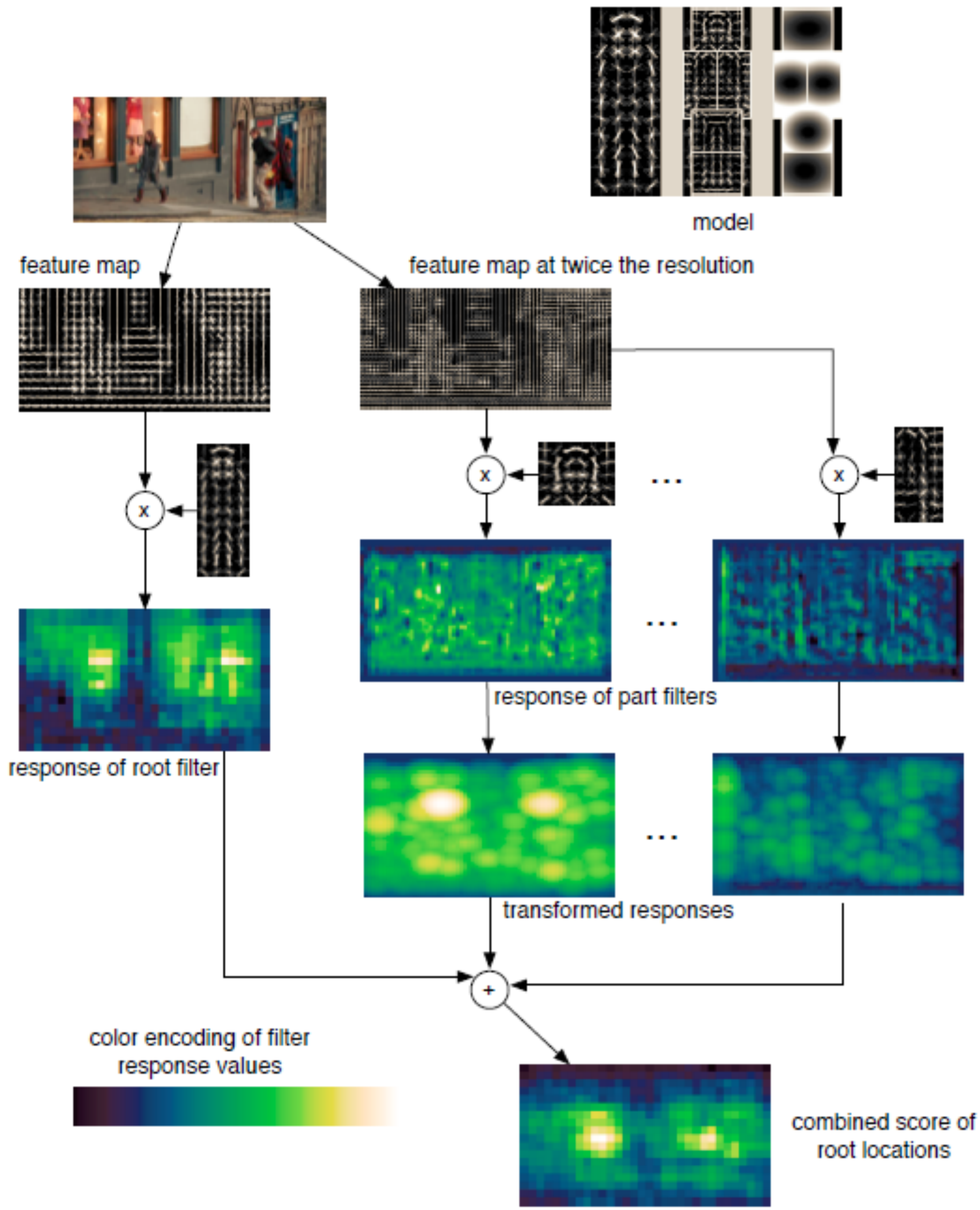




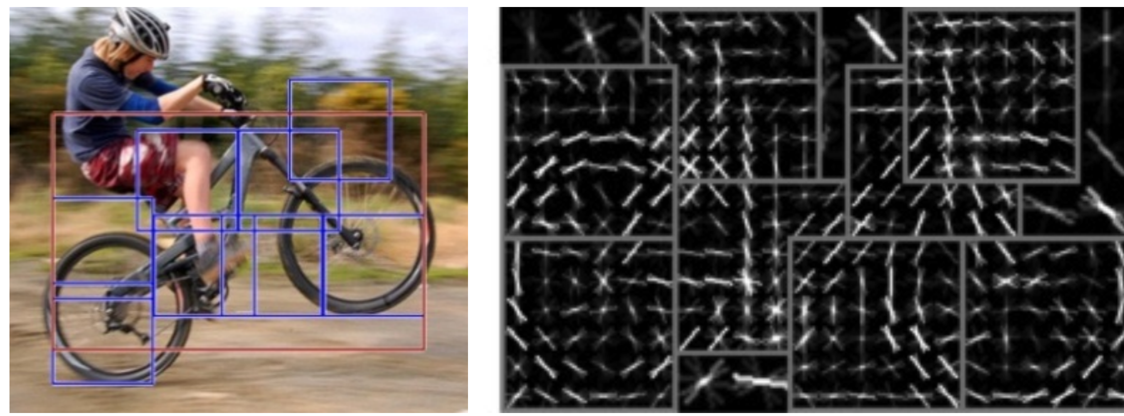
model
feature map at twice the resolution
x
x
...
...
response of part filters

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




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

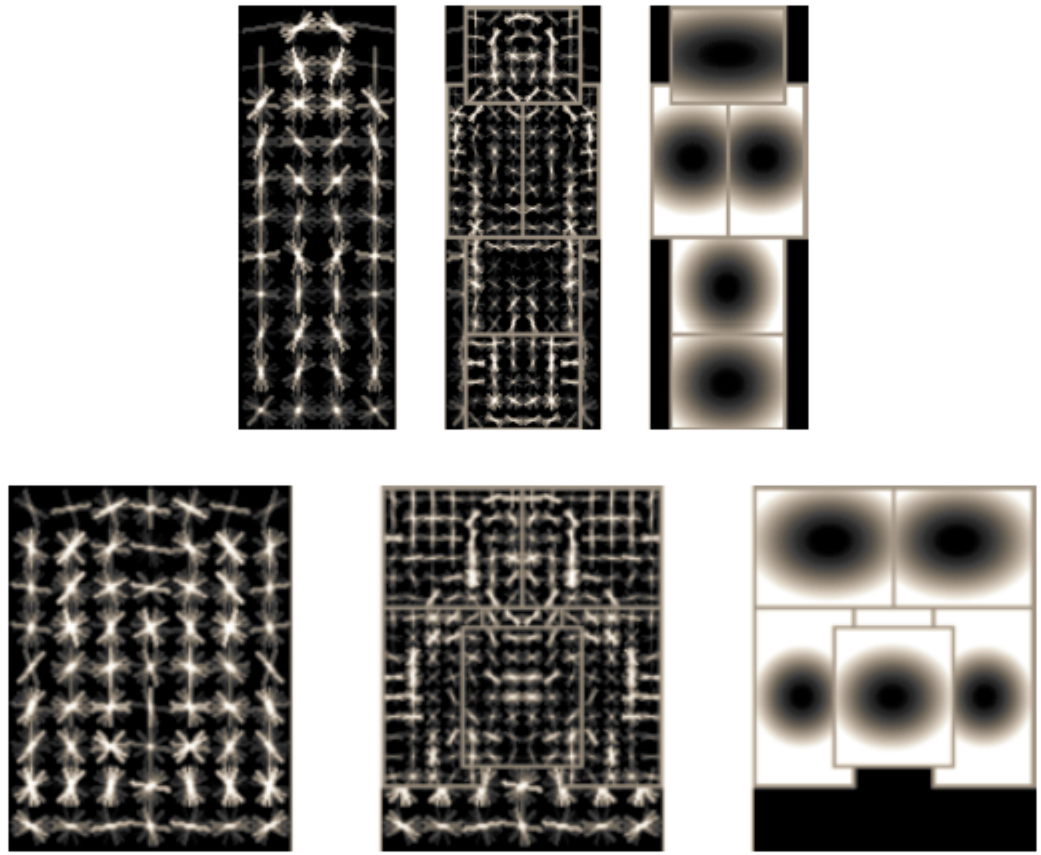
State-of-the-art Detector: Deformable Parts Model (DPM)



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



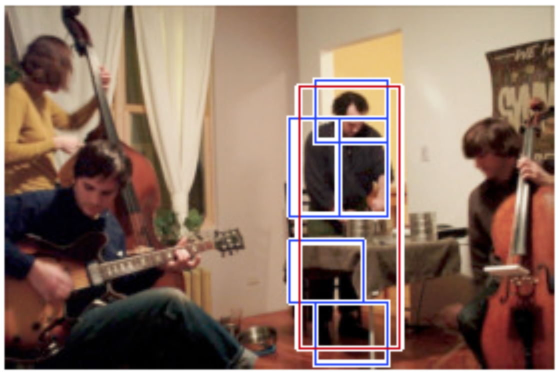
root filters
coarse resolution

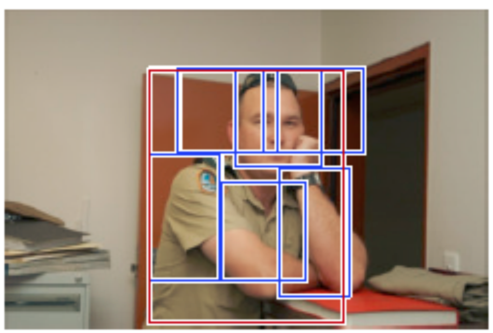
part filters
finer resolution

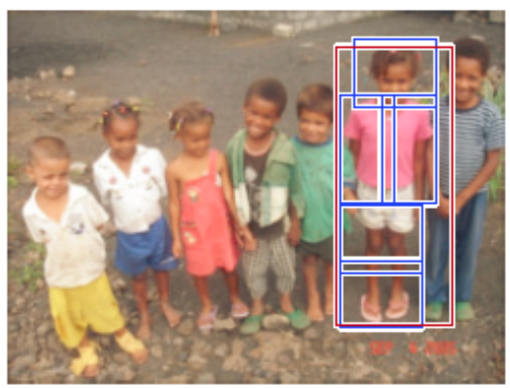
deformation
models

Person detections

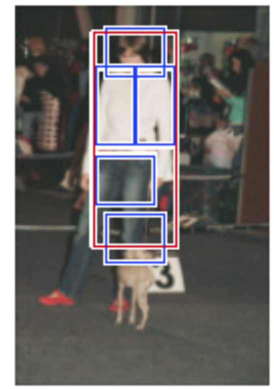
high scoring true positives



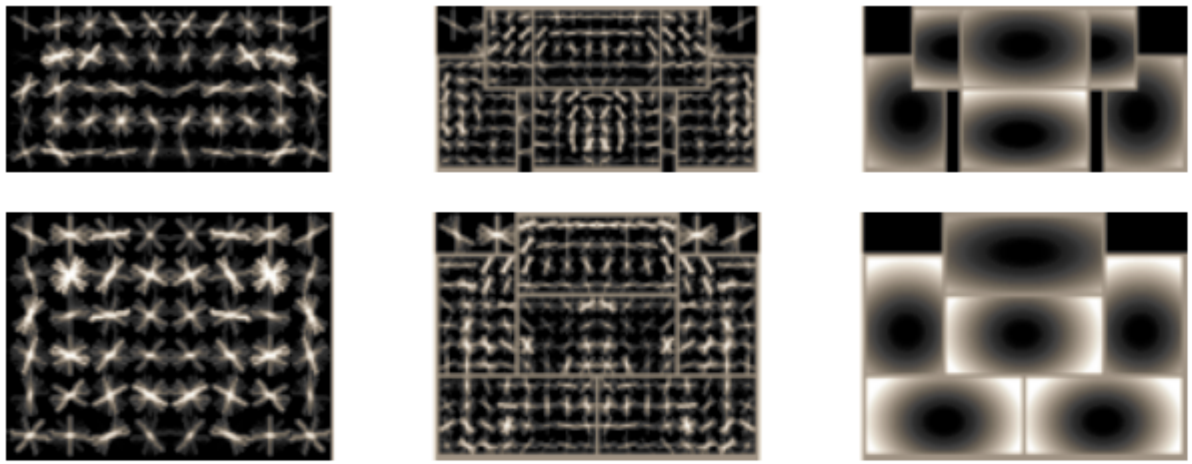




high scoring false positives
(not enough overlap)



Car



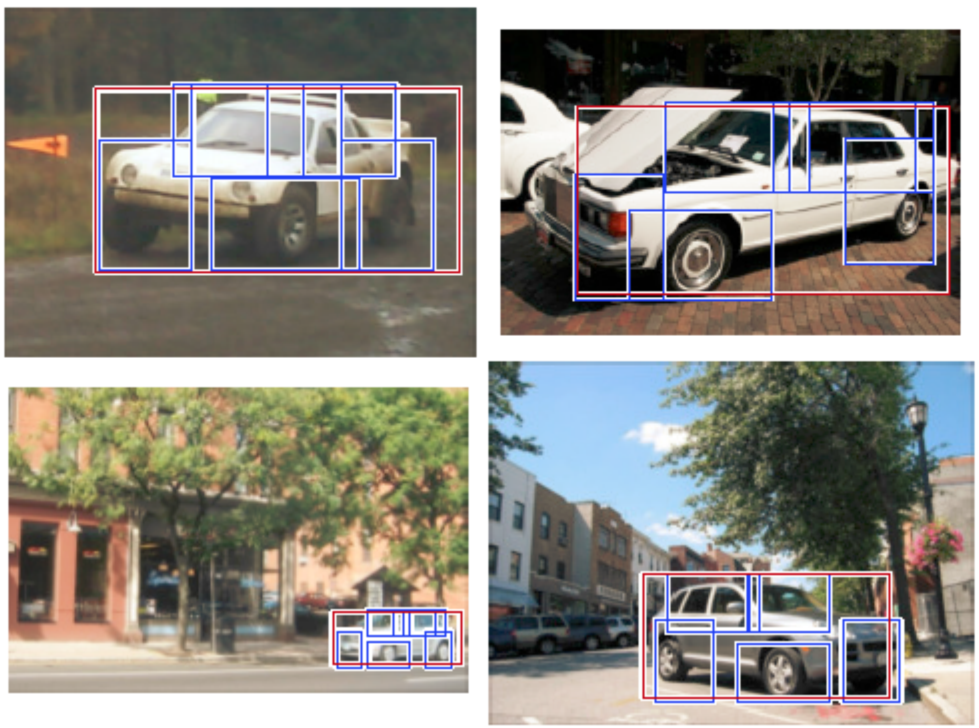
root filters
coarse resolution

part filters
finer resolution

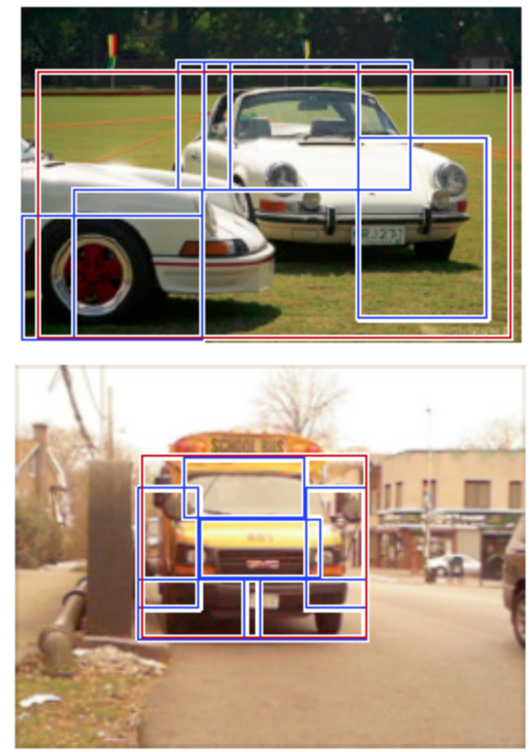
deformation
models

Car detections

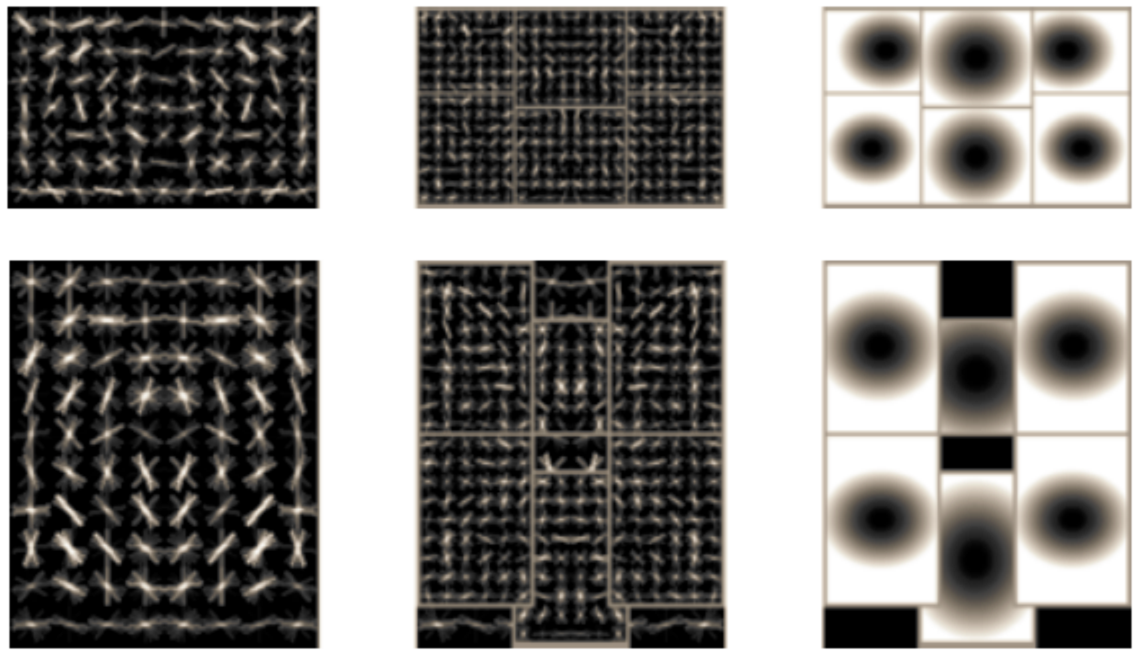
high scoring true positives



high scoring false positives



Cat



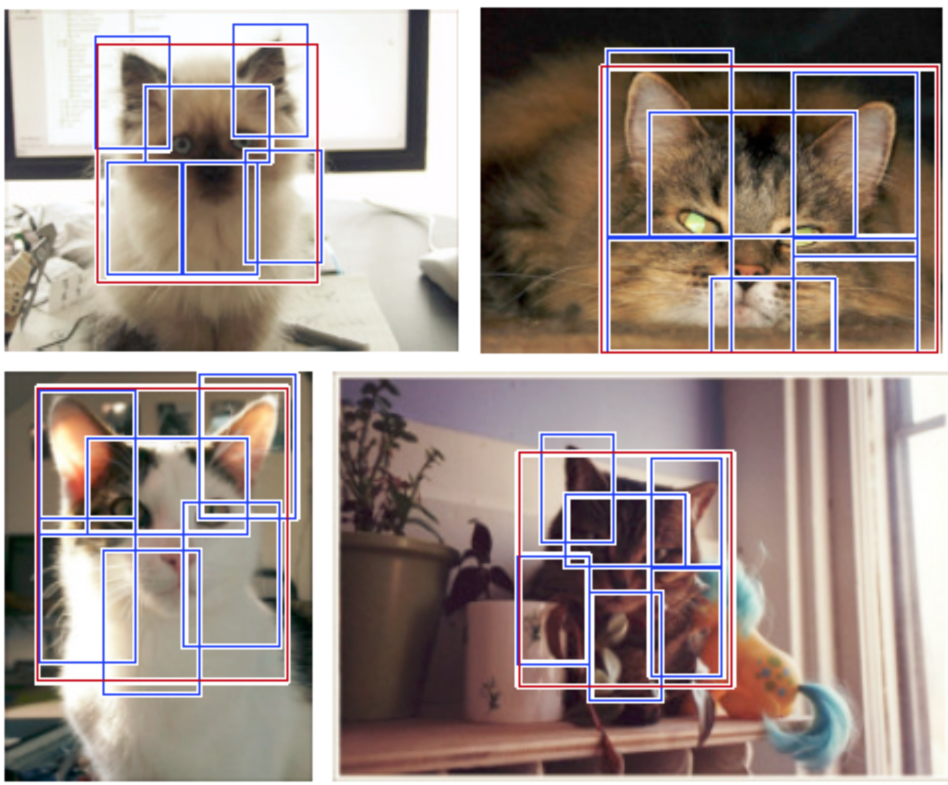
root filters
coarse resolution

part filters
finer resolution

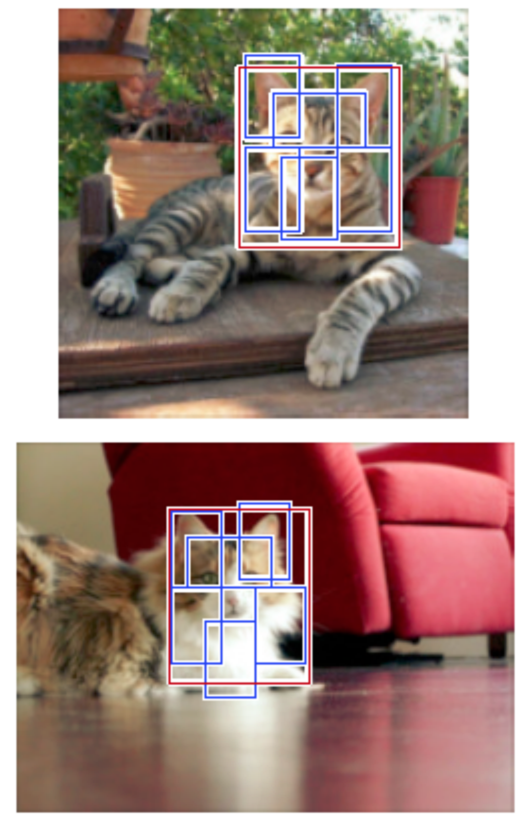
deformation
models

Cat detections

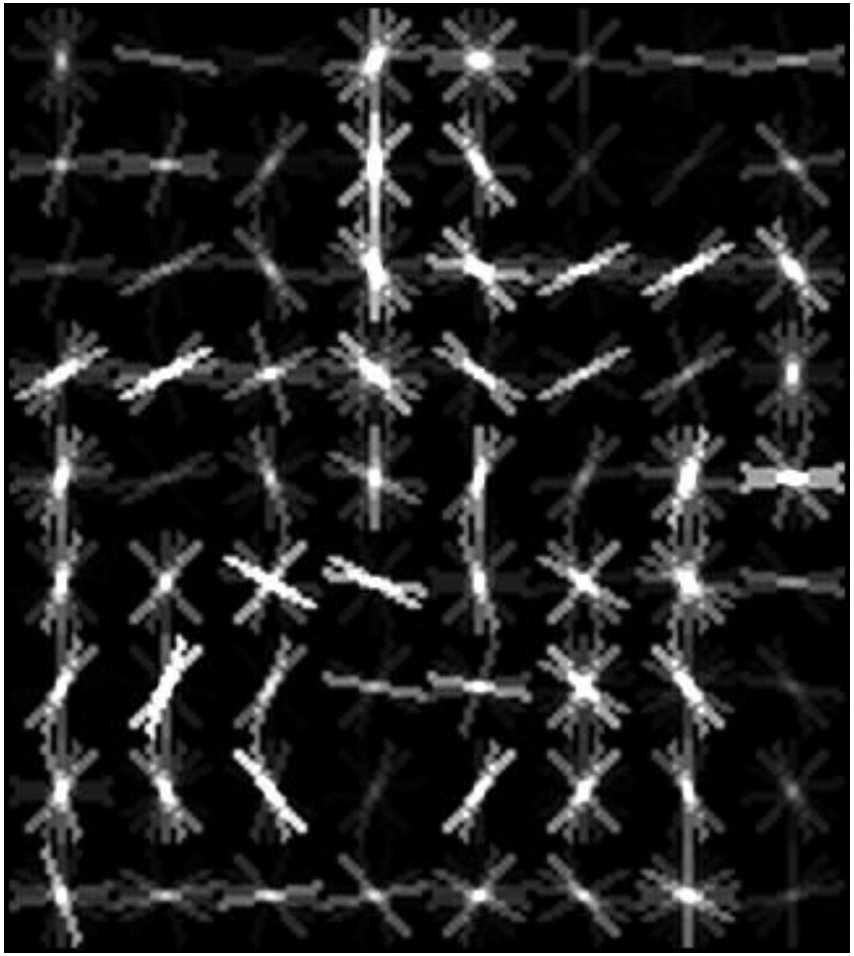
high scoring true positives



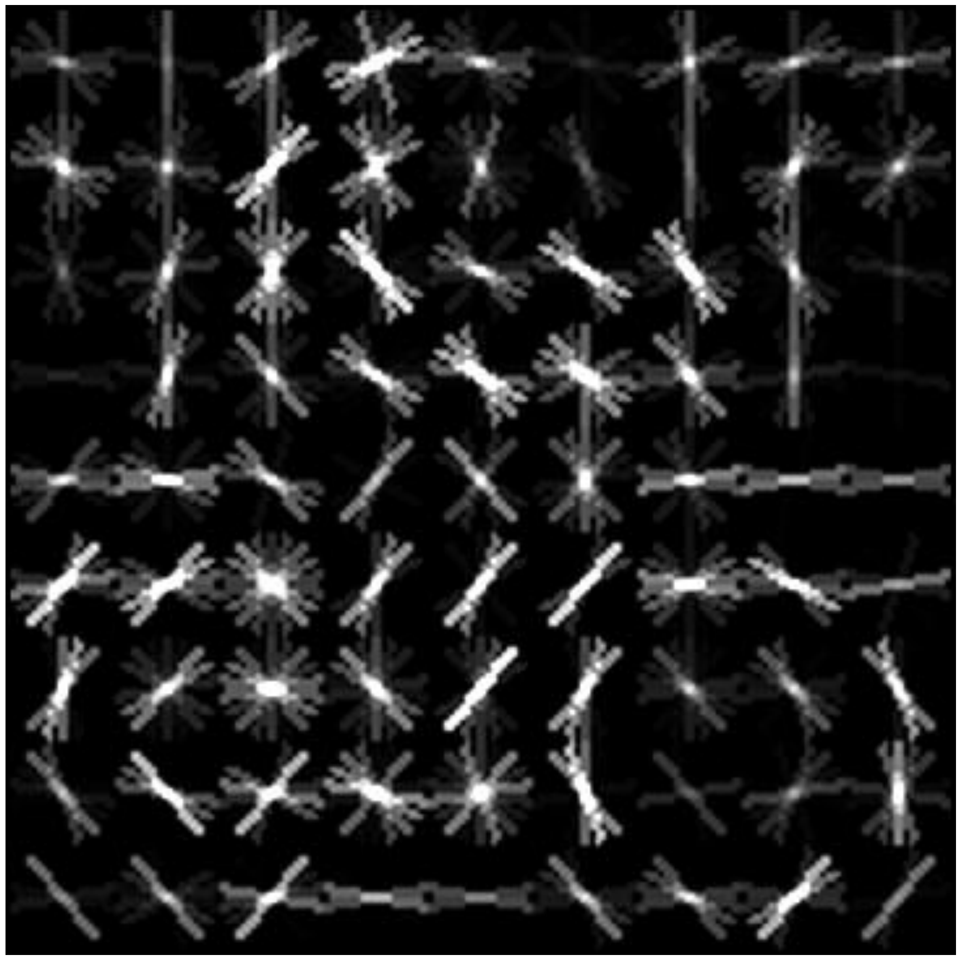
high scoring false positives
(not enough overlap)

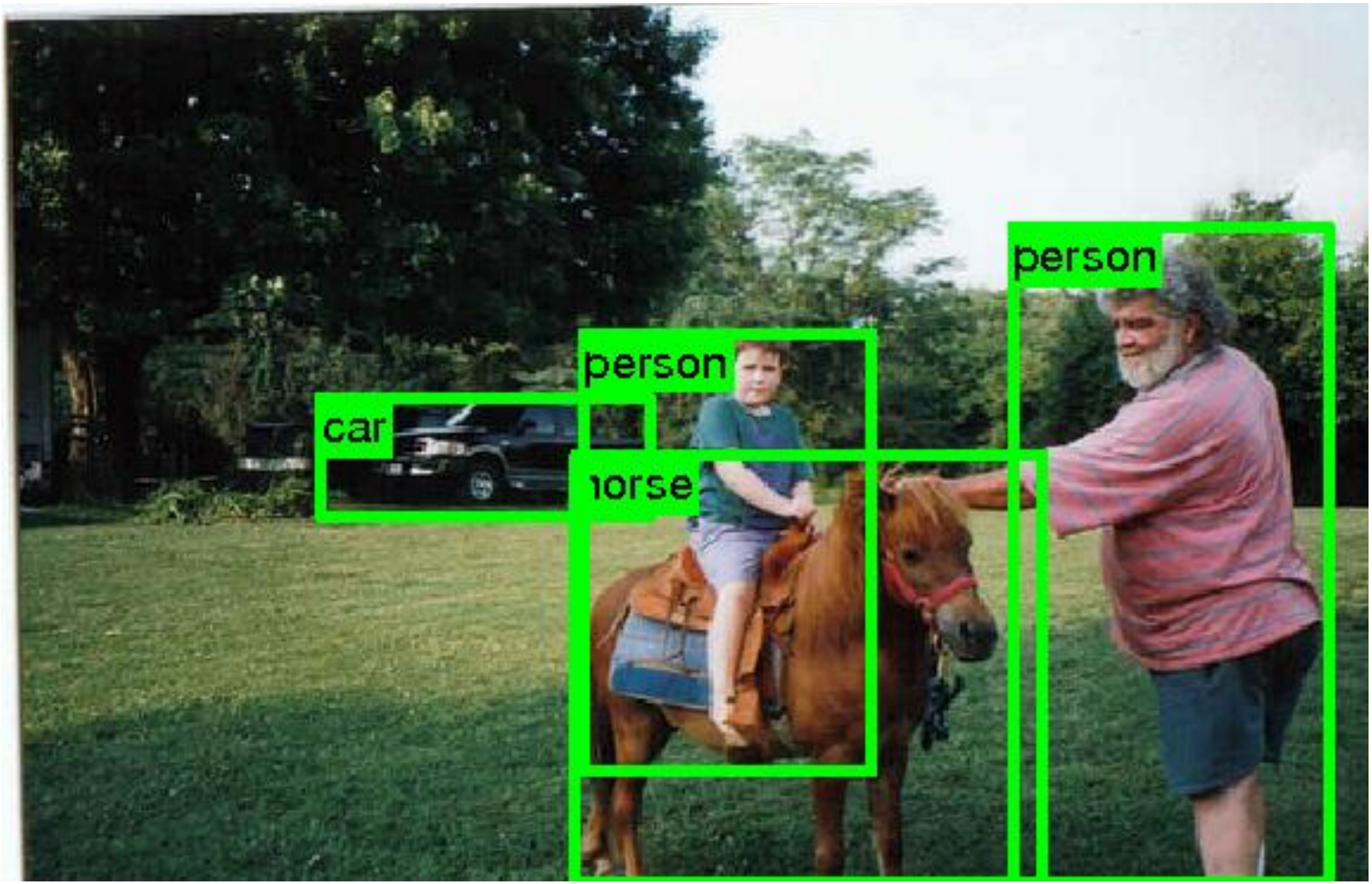


Person riding horse

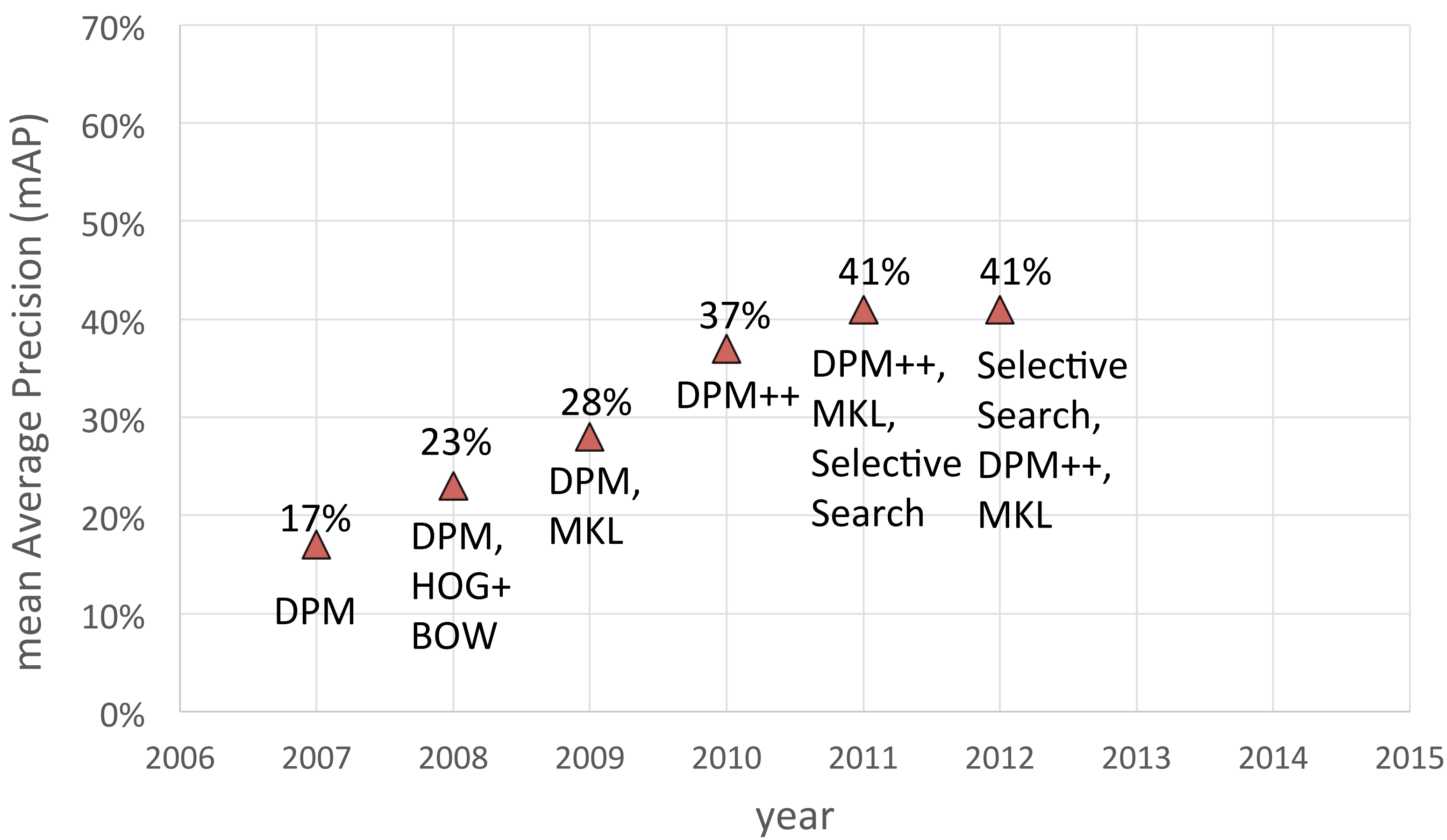


Person riding bicycle

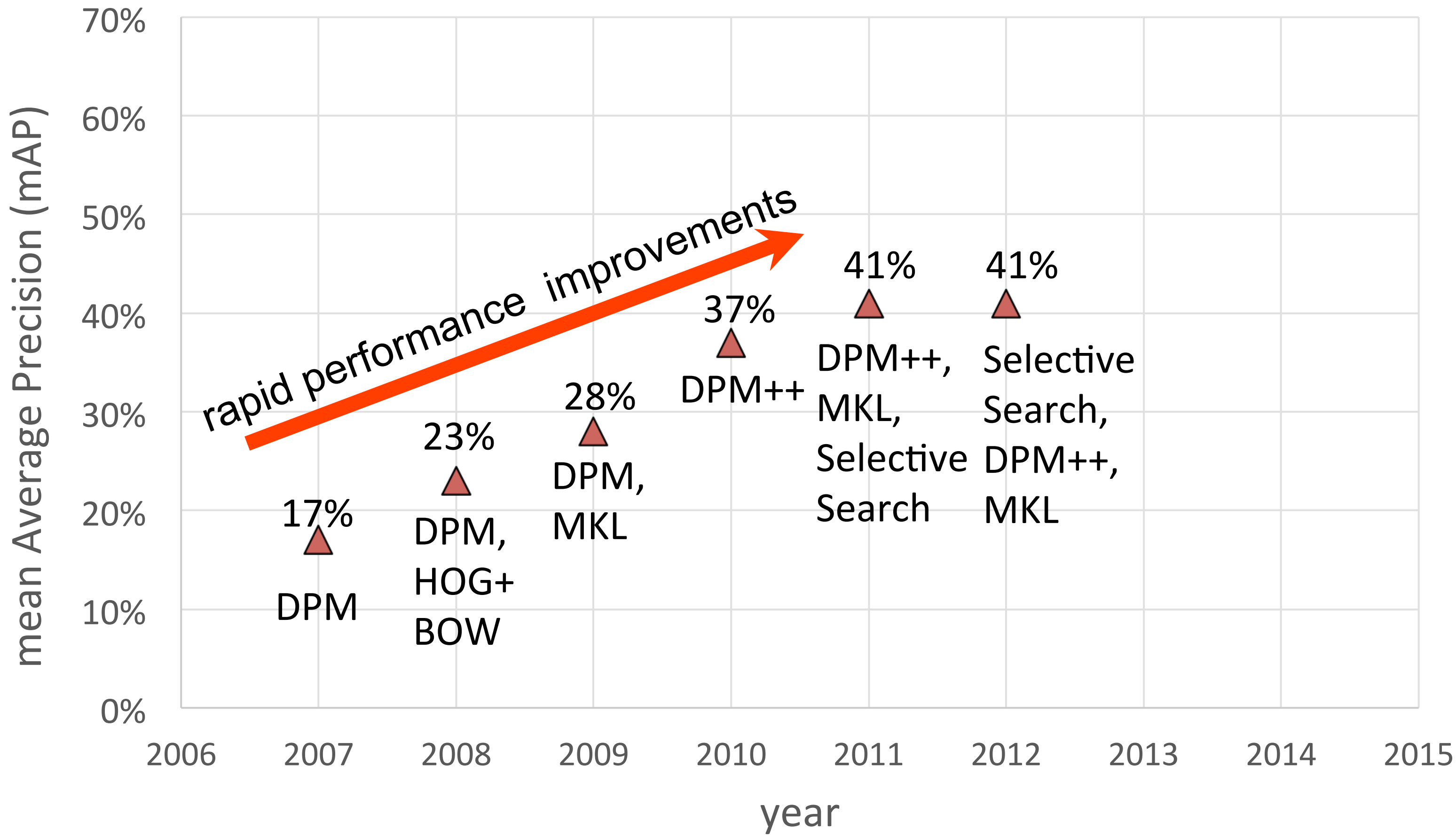



person
person
car
horse

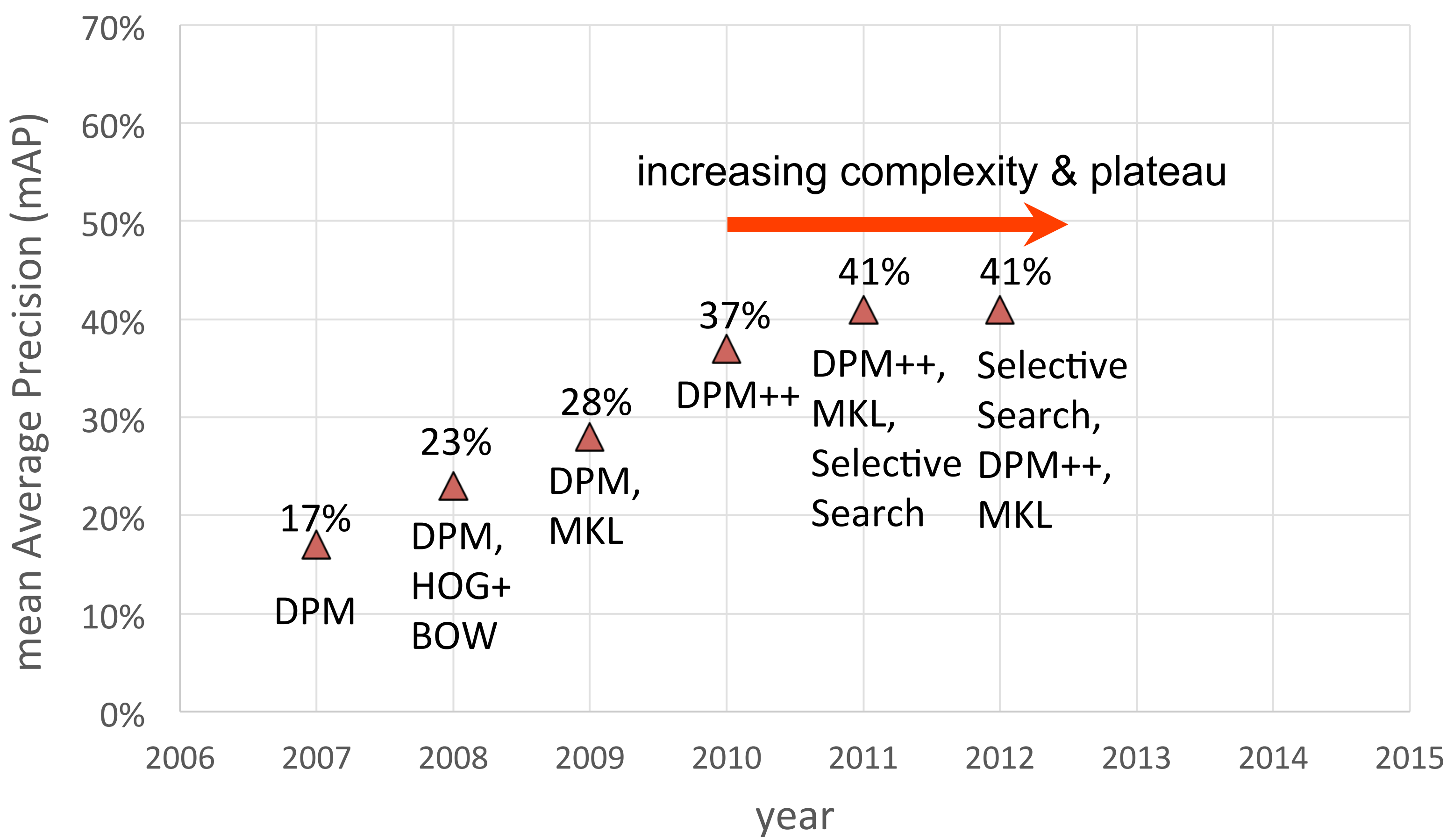
PASCAL VOC detection history



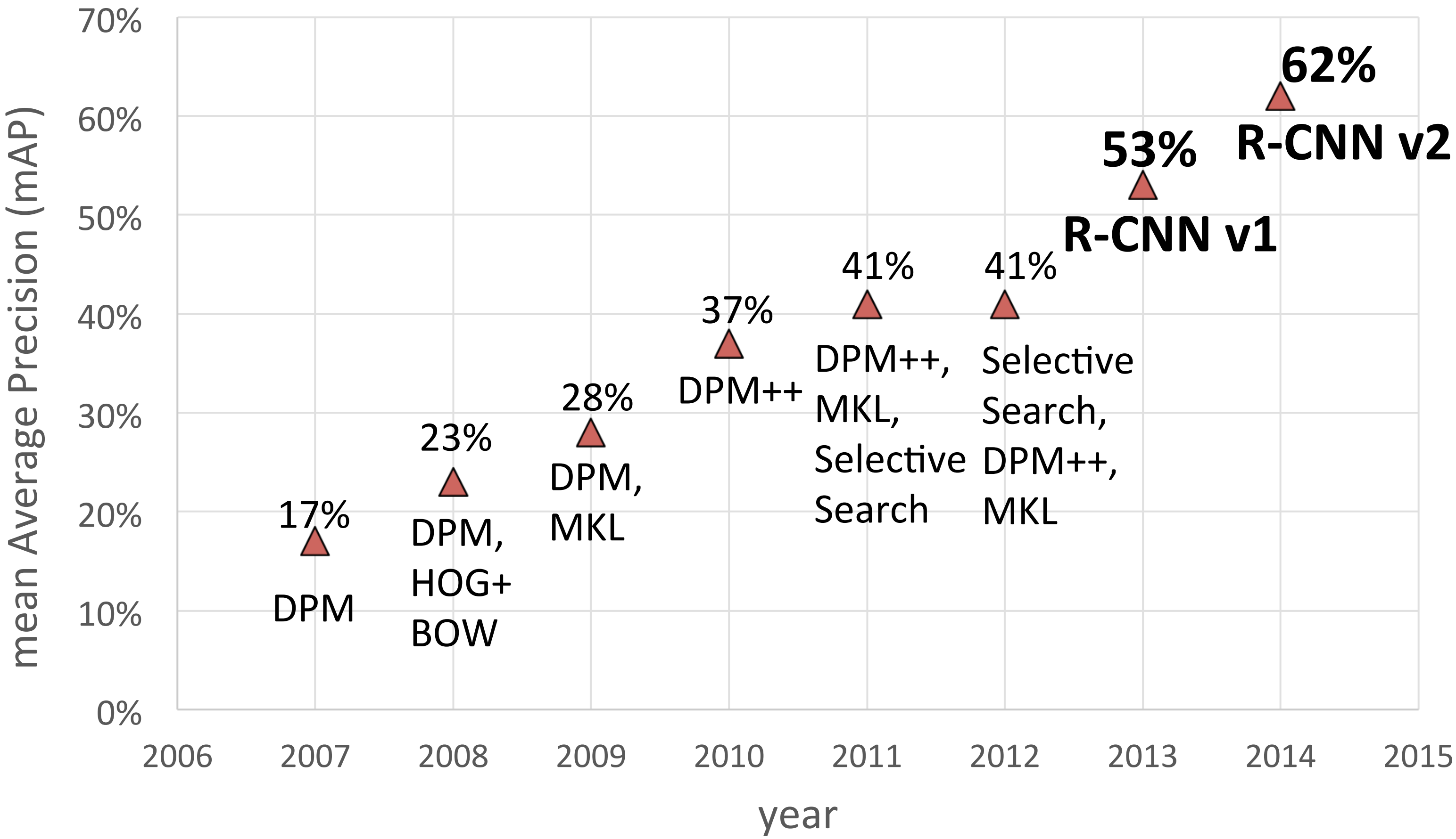
Part-based models & multiple features (MKL)



Kitchen-sink approaches

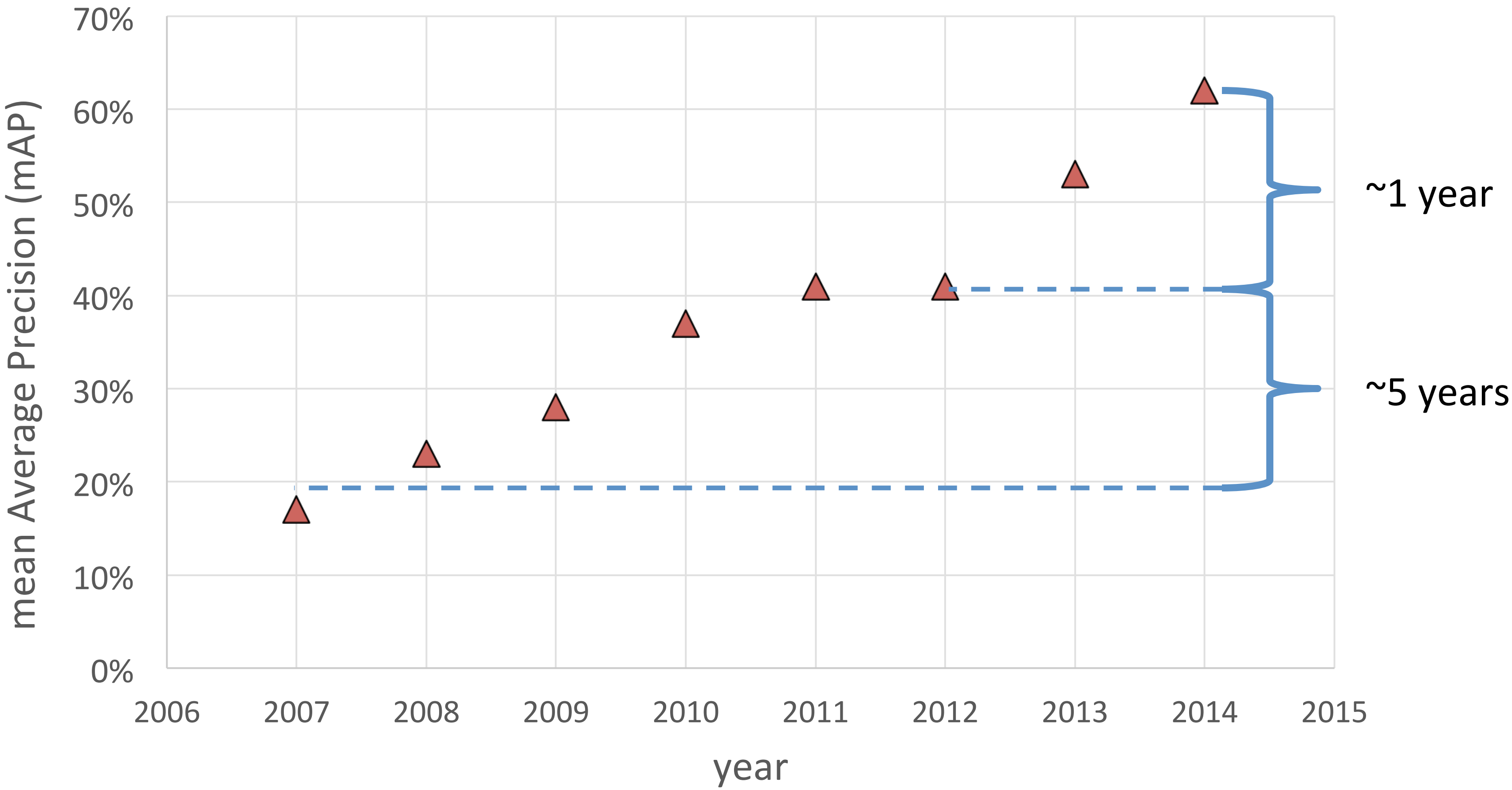


Region-based Convolutional Networks (R-CNNs)



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

Region-based Convolutional Networks (R-CNNs)

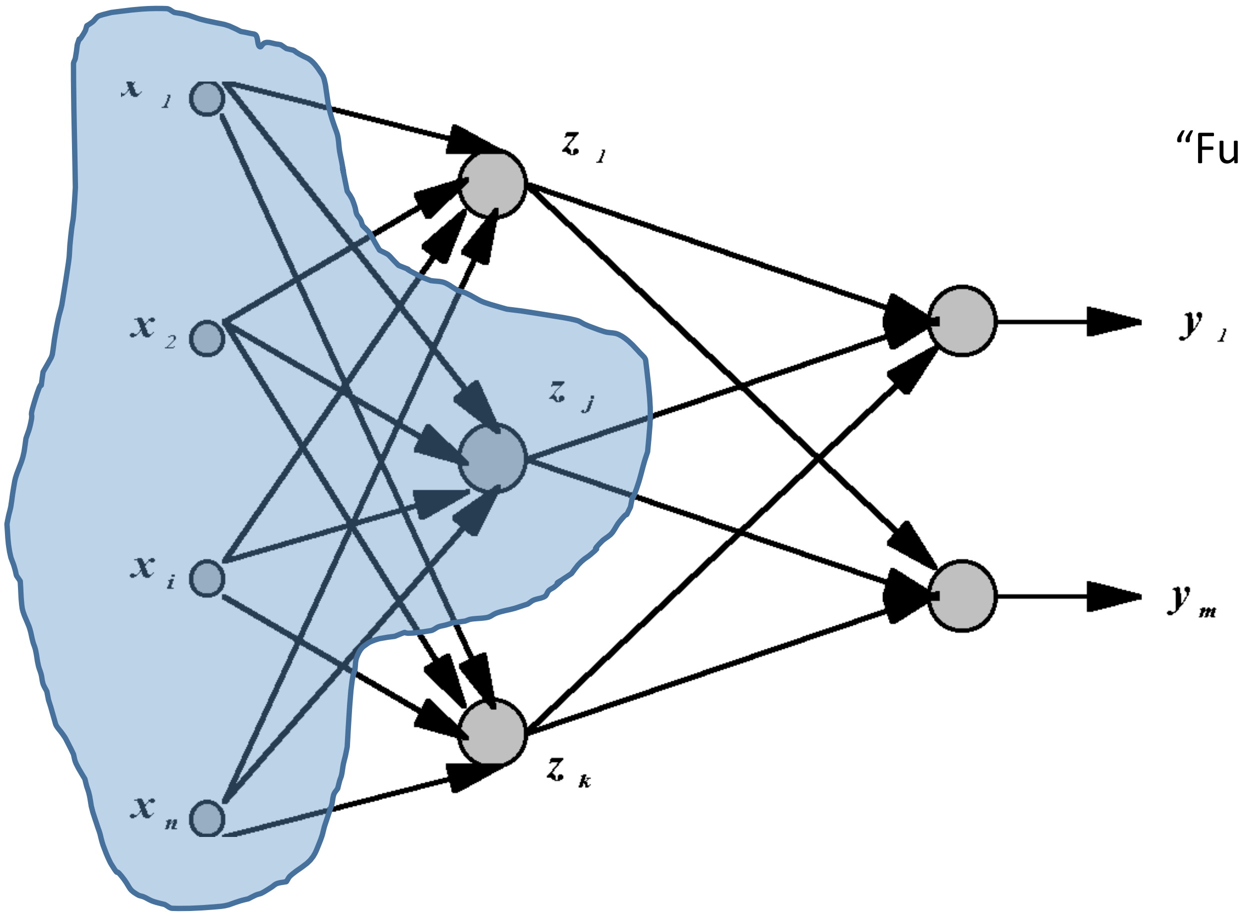


[R-CNN. Girshick et al. CVPR 2014]

Convolutional Neural Networks

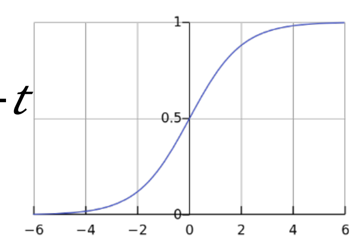
- Overview

Standard Neural Networks



"Fully connected"

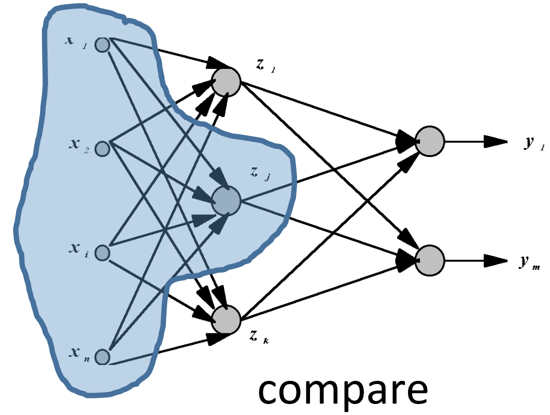
$\boldsymbol{x} = (x_1, \dots, x_{784})^T$ $\quad z_j = g(\boldsymbol{w}_j^T \boldsymbol{x})$ $\quad g(t) = \frac{1}{1+e^{-t}}$



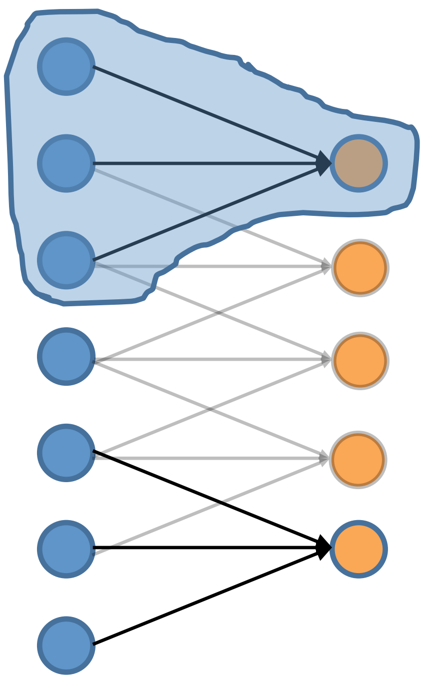
From NNs to Convolutional NNs

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

Convolutional NNs



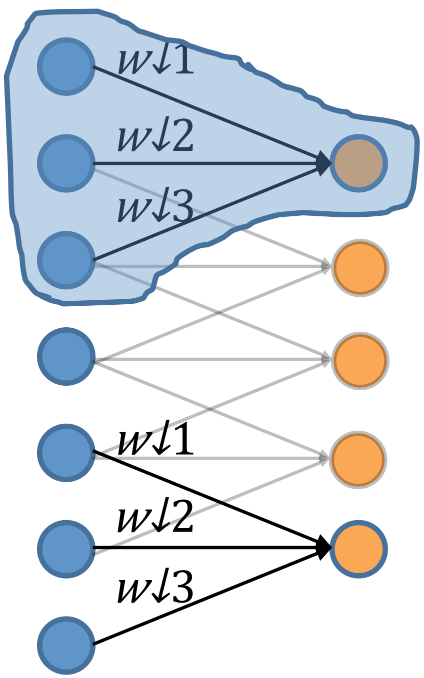
- Local connectivity



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

Convolutional NNs

- Shared ("tied") weights

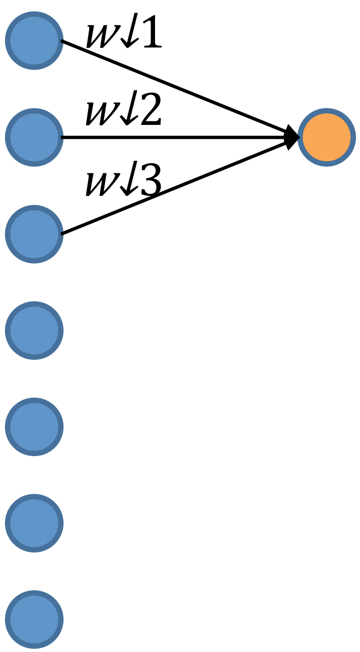


All orange units share the same parameters

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

Convolutional NNs

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

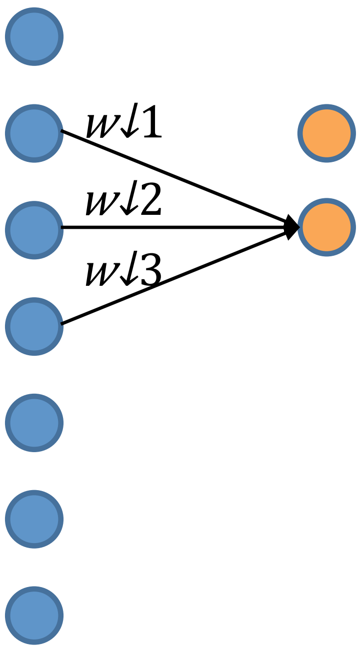


All orange units share the same parameters

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

Convolutional NNs

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

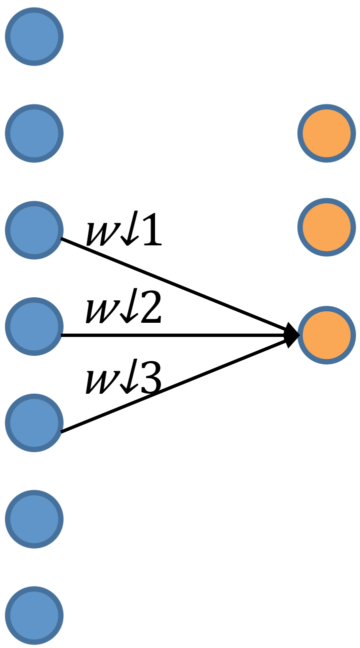


All orange units share the same parameters

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

Convolutional NNs

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

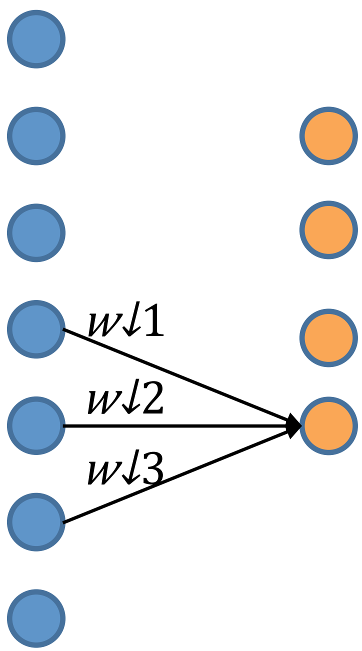


All orange units share the same parameters

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

Convolutional NNs

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

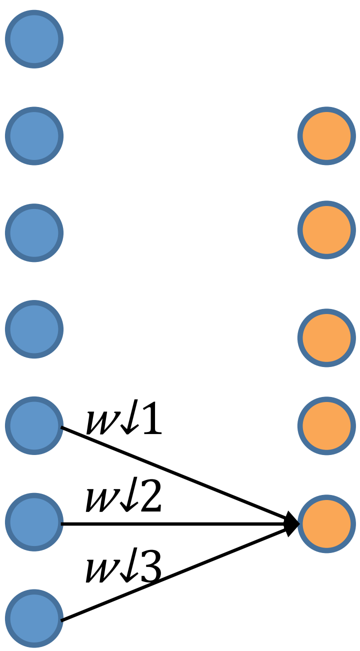


All orange units share the same parameters

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

Convolutional NNs

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

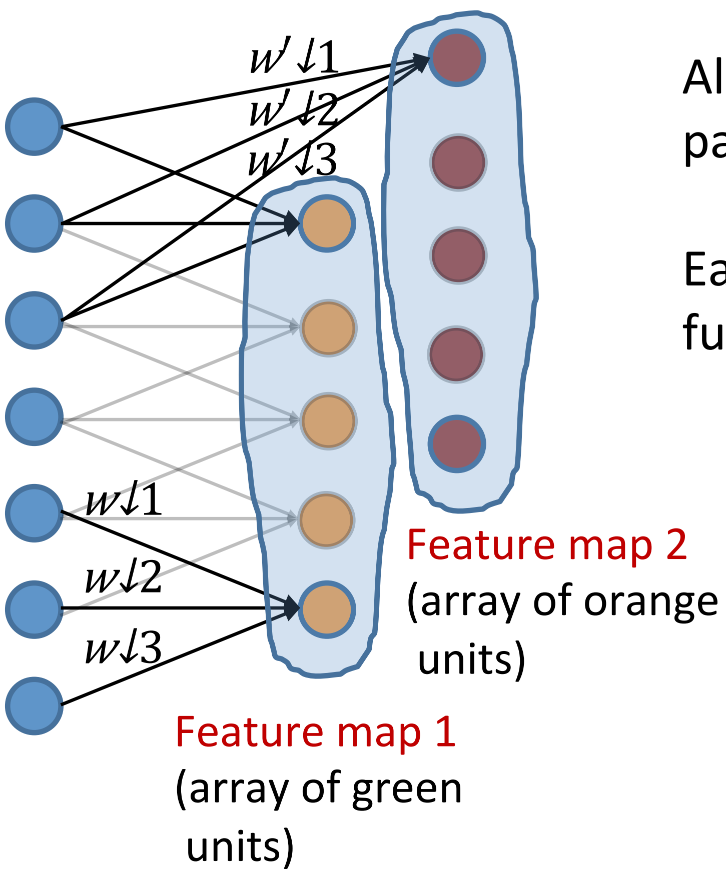


All orange units share the same parameters

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

Convolutional NNs

- Multiple feature maps

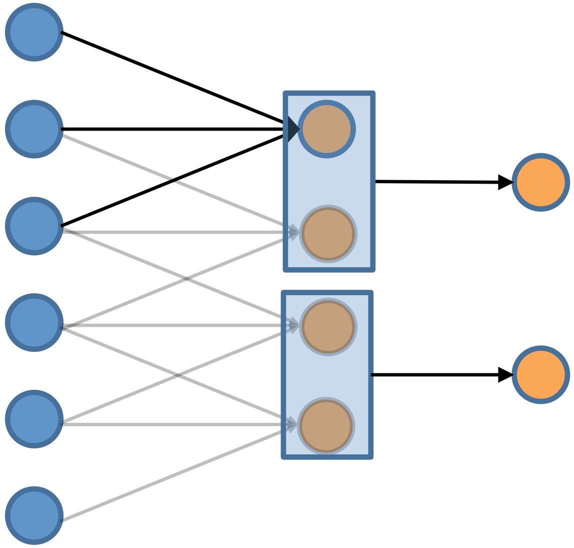


All orange units share the same parameters

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

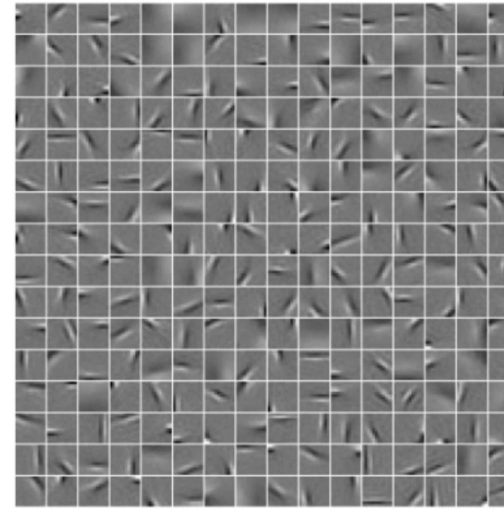
Convolutional NNs

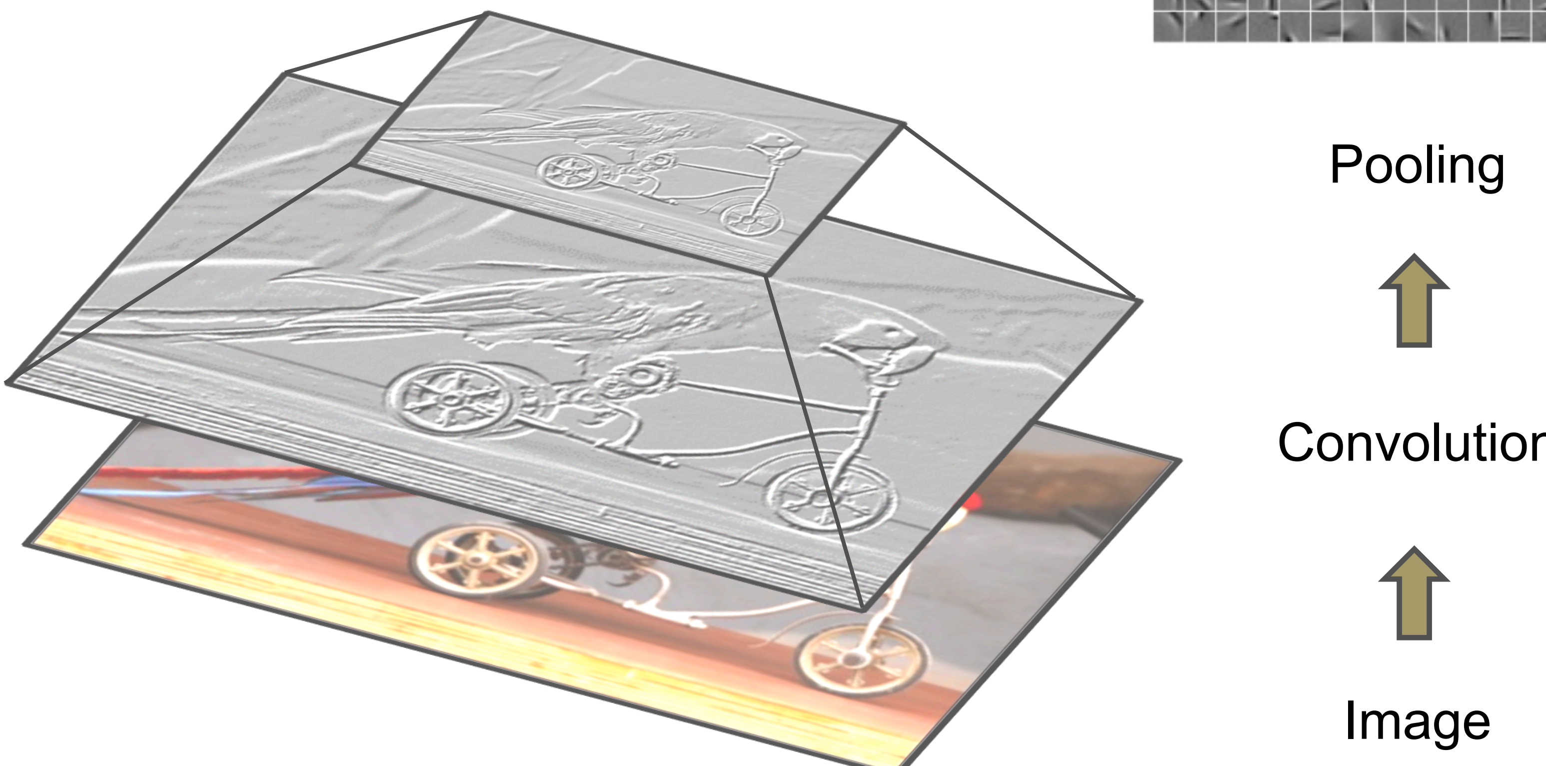
- Pooling (max, average)



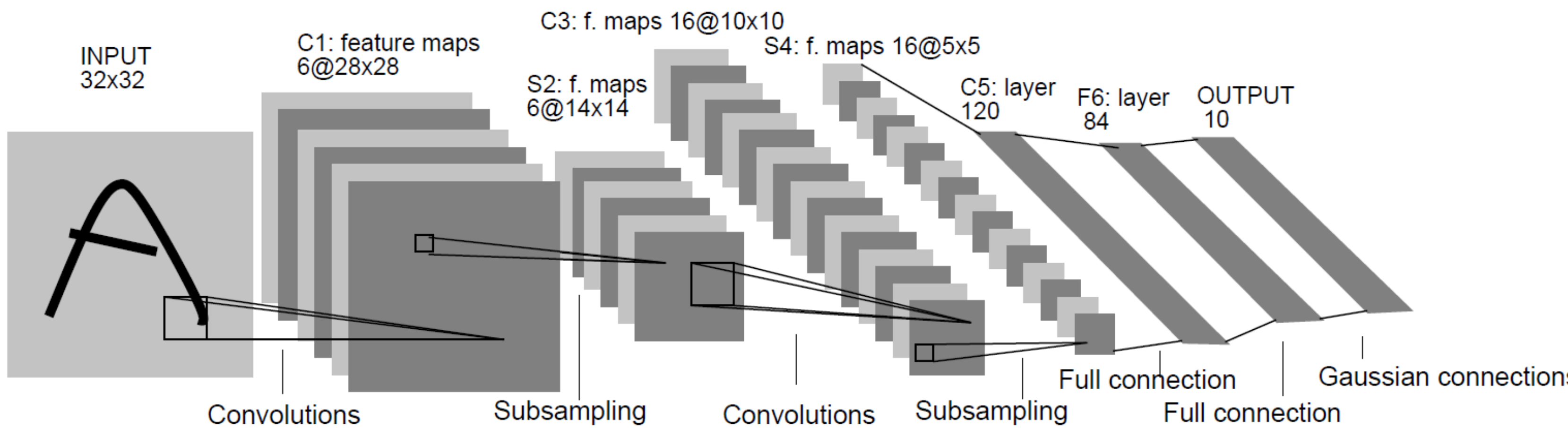
- Pooling area: 2 units
- Pooling stride: 2 units
- **Subsamples** feature maps

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

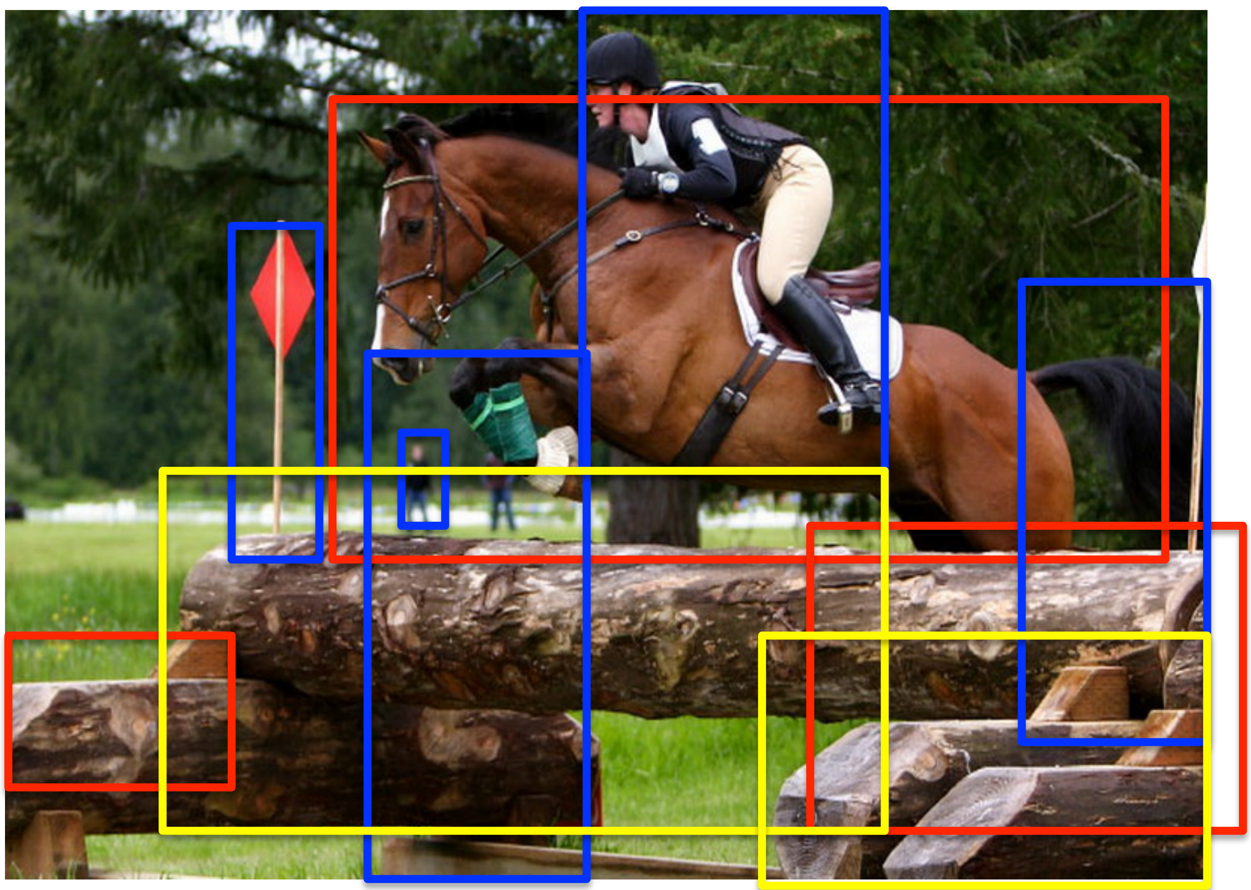
- Caffe (C++, python, matlab)
- Torch7 (C++, lua)
- Theano (python)

Core idea of “deep learning”

- Input: the “*raw*” signal (image, waveform, …)

- Features: hierarchy of features is *learned* from the raw input

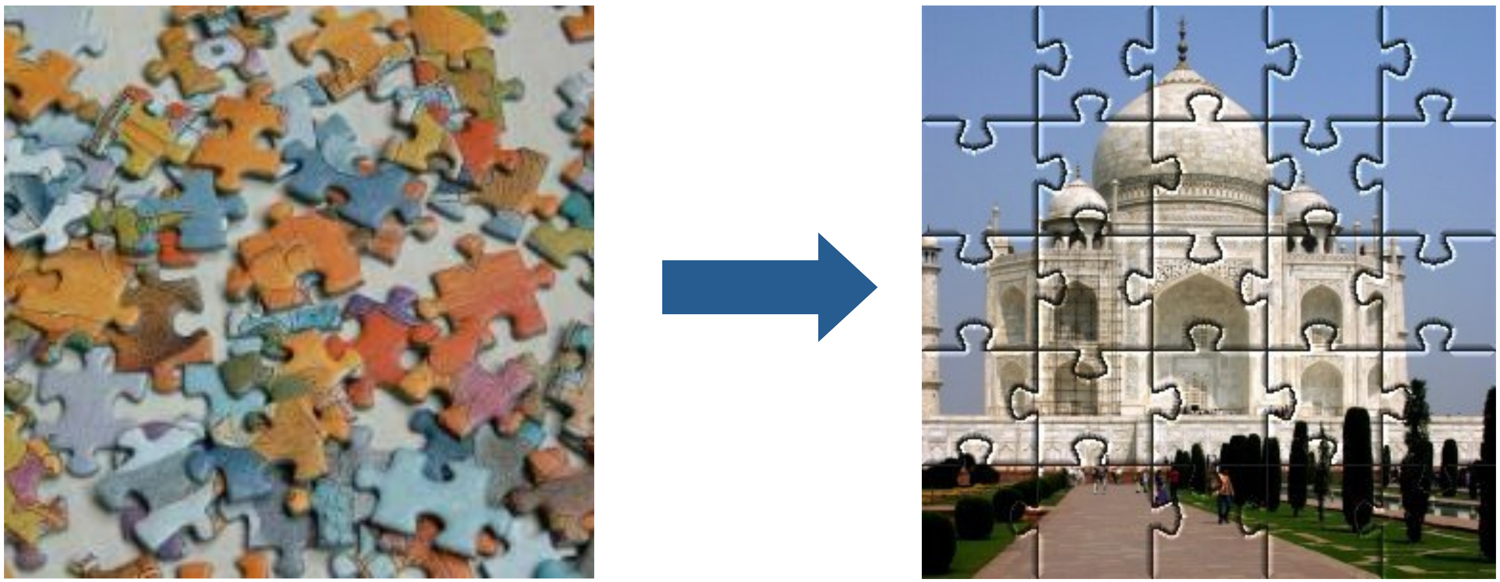
Structure



Recognition using Visual Phrases , CVPR 2011

Structured Prediction

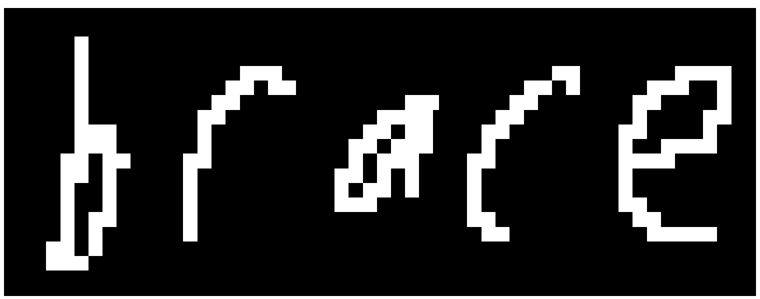
- Prediction of complex outputs
  - Structured outputs: multivariate, correlated, constrained



- Novel, general way to solve many learning problems

Handwriting Recognition

x **y**

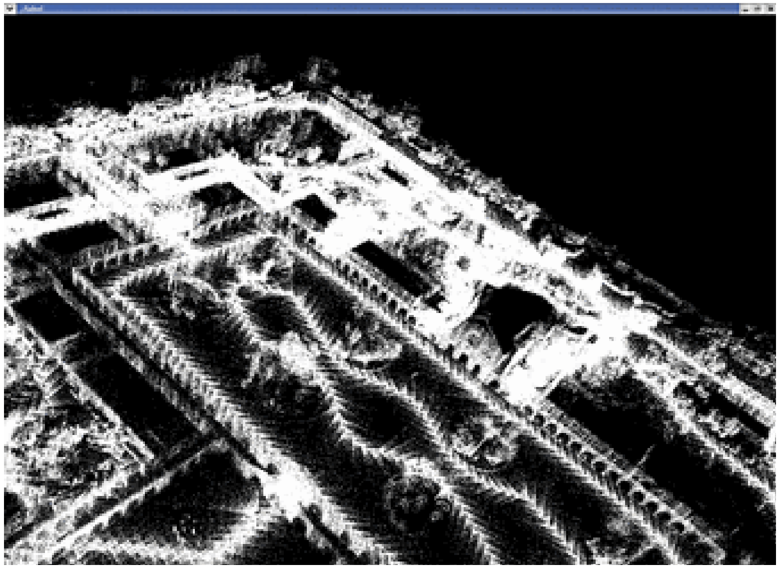


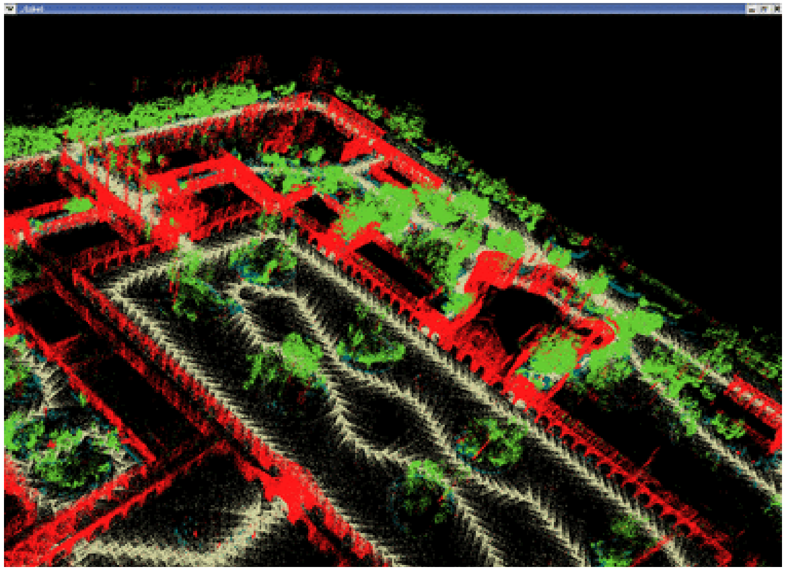
brace

Sequential structure

Object Segmentation

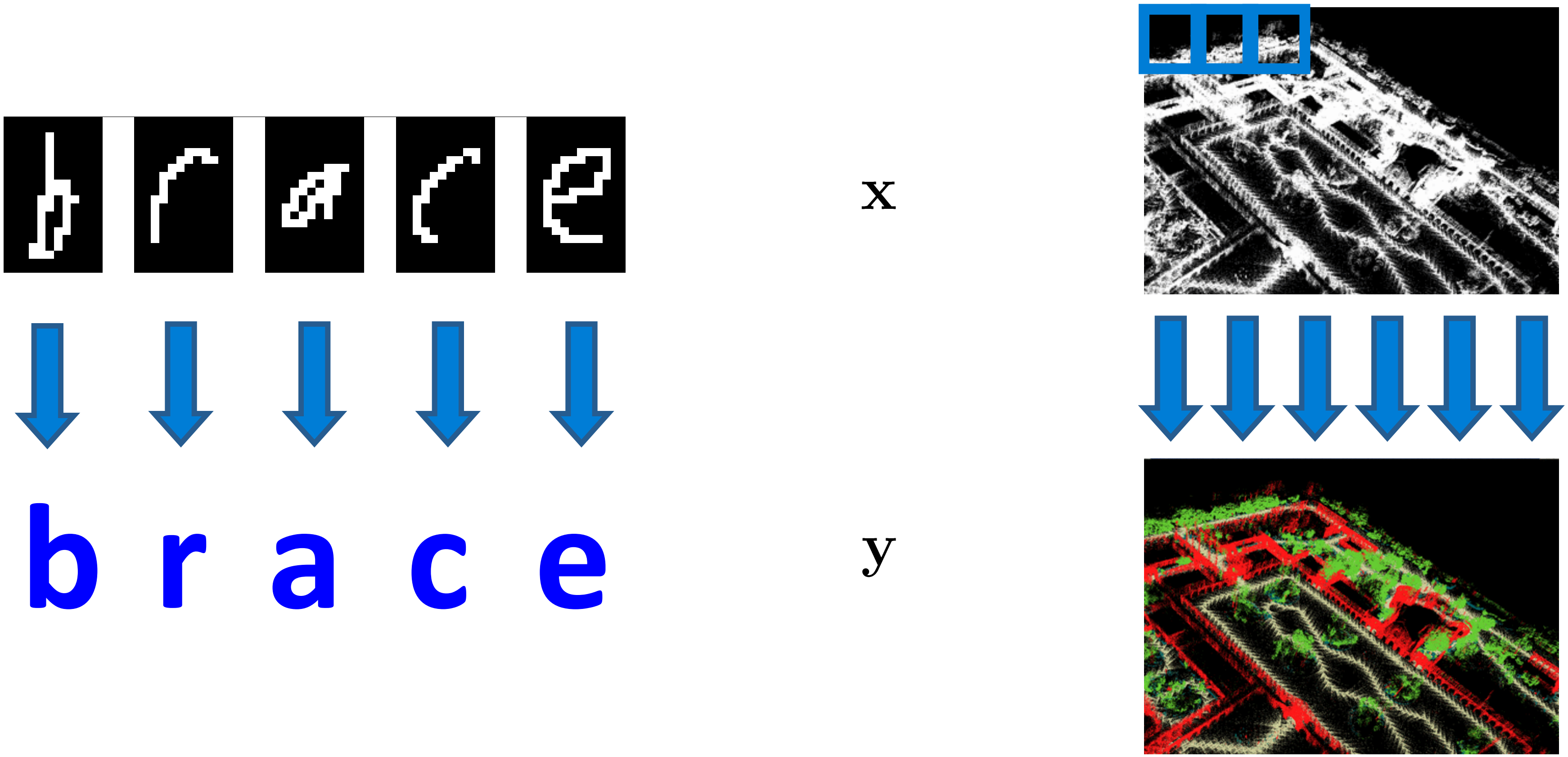
x **y**





Spatial structure

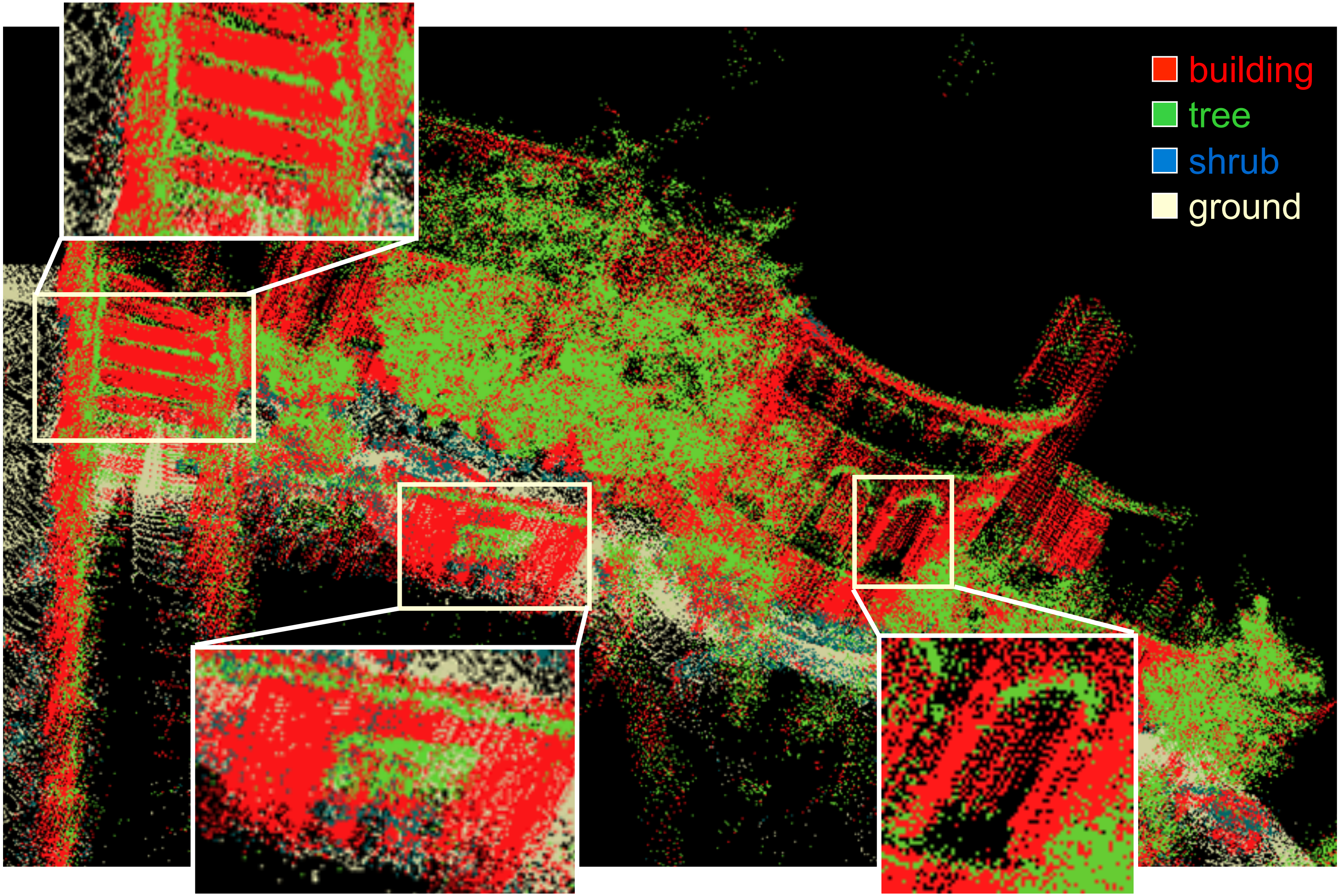
Local Prediction



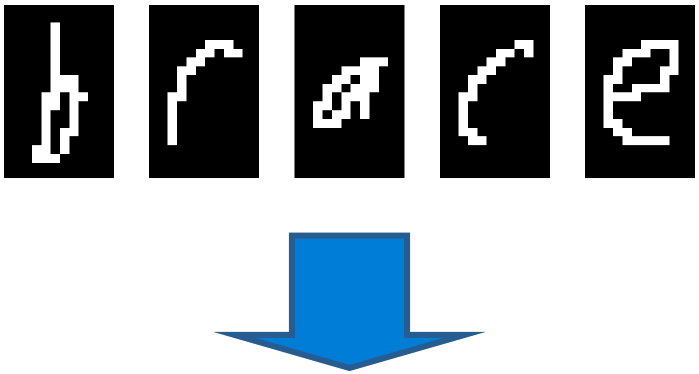
Classify using local information

Ignores correlations & constraints!

Local Prediction



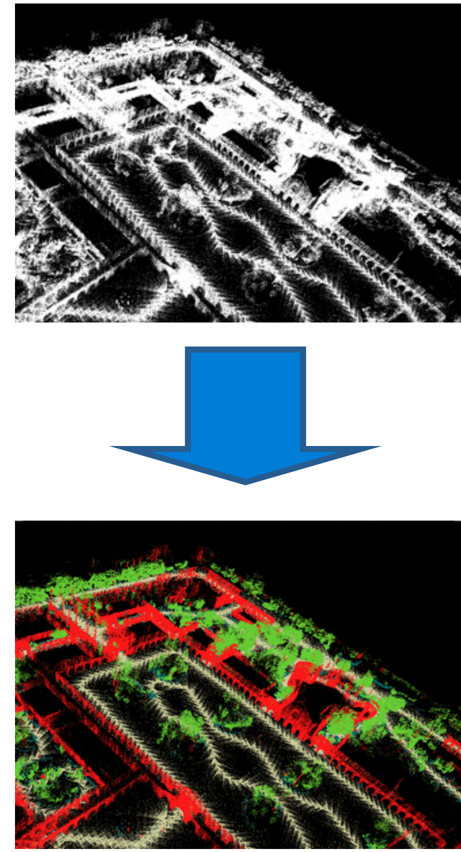
Structured Prediction



b r a c e

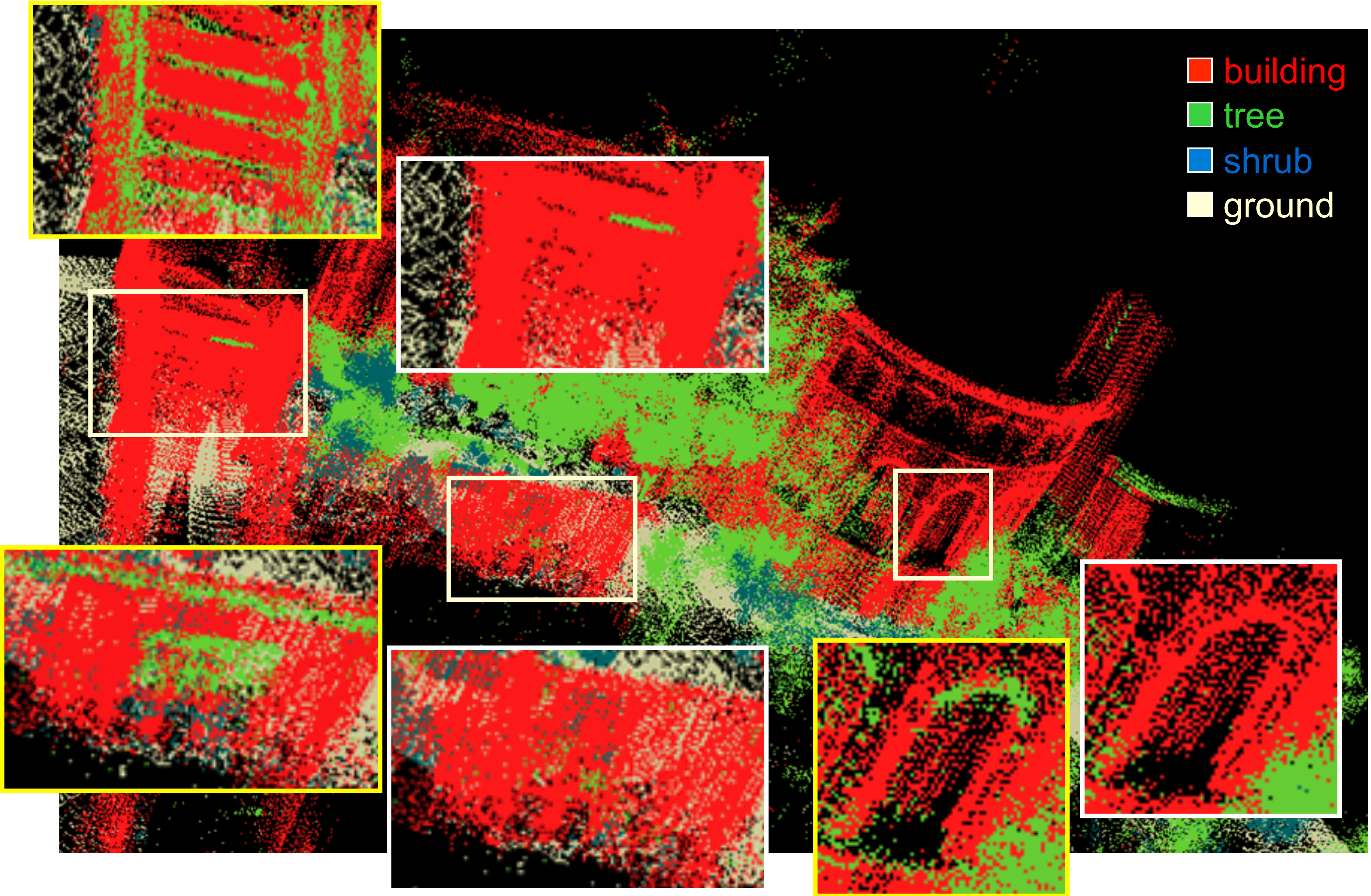
x

y



- Use local information
- Exploit correlations

Structured Prediction



Structured Models

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

scoring function

space of feasible outputs

Mild assumptions:

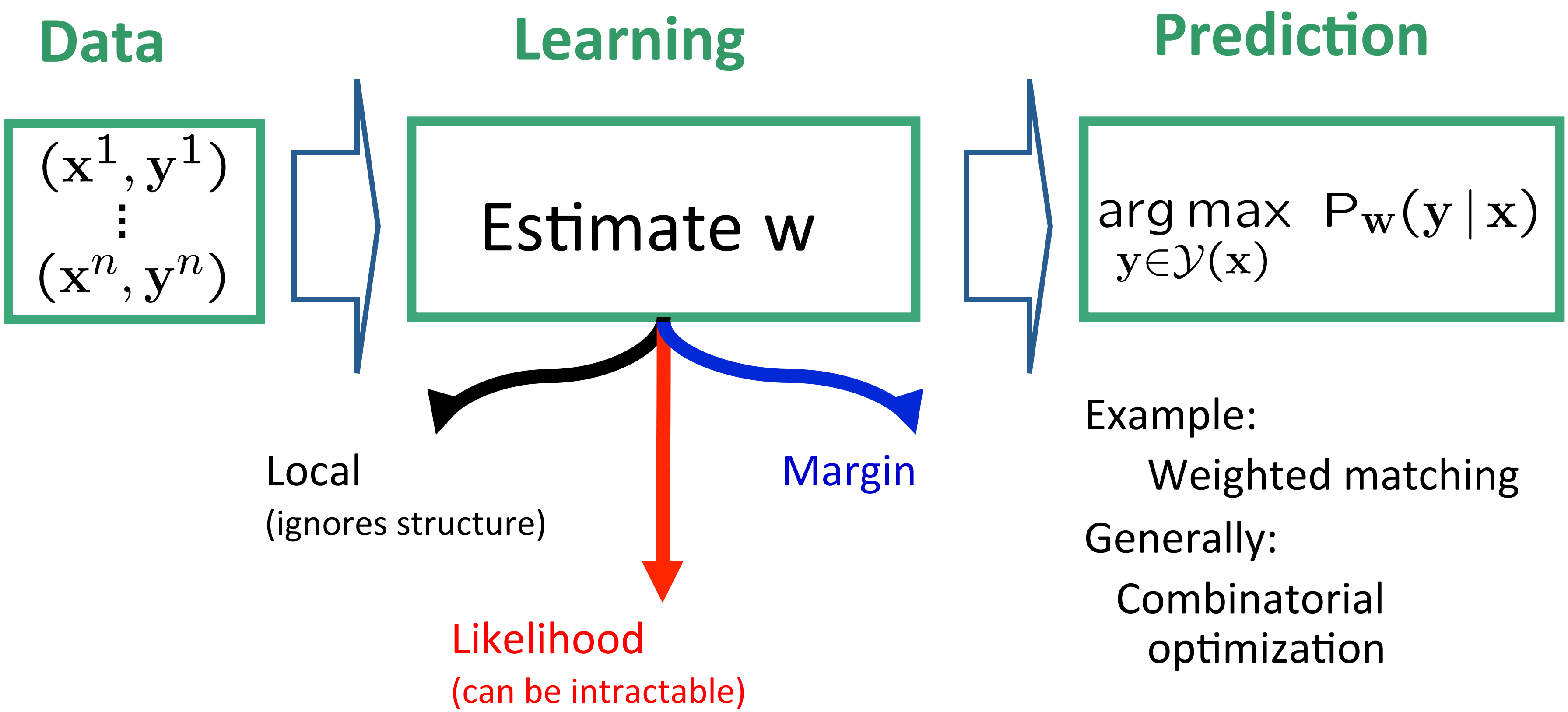
$$score(\mathbf{x}, \mathbf{y}) = \mathbf{w}^\top \mathbf{f}(\mathbf{x}, \mathbf{y}) = \sum_p \mathbf{w}^\top \mathbf{f}(\mathbf{x}_p, \mathbf{y}_p)$$

linear combination

sum of part scores

Supervised Structured Prediction

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



Local Estimation

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

- Treat edges as independent decisions

Data

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

- 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

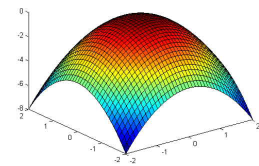
Model: $$\mathrm{P}_{\mathbf{w}}(\mathbf{y} \mid \mathbf{x}) = \frac{\prod_{jk} \exp\{\mathbf{w}^\top \mathbf{f}(y_{jk}, \mathbf{x})\}}{\sum_{\mathbf{y}' \in \mathcal{Y}(\mathbf{x})} \prod_{jk} \exp\{\mathbf{w}^\top \mathbf{f}(y'_{jk}, \mathbf{x})\}}$$

Data

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

- Estimate **w** jointly:

$$\sum_i \log \mathrm{P}_{\mathbf{w}}(\mathbf{y}^i \mid \mathbf{x}^i)$$



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

- Denominator is #P-complete
  [Valiant 79, Jerrum & Sinclair 93]

- Tractable model, intractable learning

- Need tractable learning method
  margin-based estimation

Structured large margin estimation

- We want:

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

- Equivalently:

$\mathbf{w}^\top \mathbf{f}($  $,$ “brace”$) \ > \ \mathbf{w}^\top \mathbf{f}($  $,$ “aaaaa”$)$

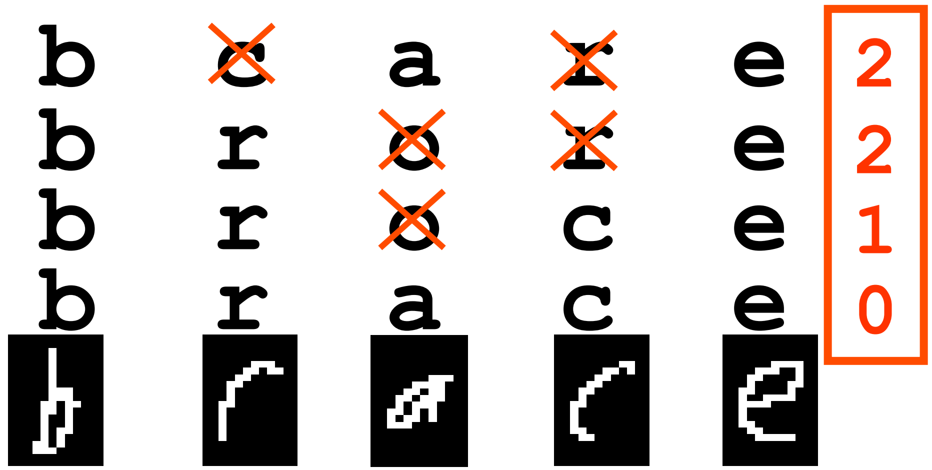
$\mathbf{w}^\top \mathbf{f}($  $,$ “brace”$) \ > \ \mathbf{w}^\top \mathbf{f}($  $,$ “aaaab”$)$

...

$\mathbf{w}^\top \mathbf{f}($  $,$ “brace”$) \ > \ \mathbf{w}^\top \mathbf{f}($  $,$ “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

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

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

  – 'Plug-in' linear program for inference

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

Min-max formulation

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

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

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

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

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

discrete optim. continuous optim.

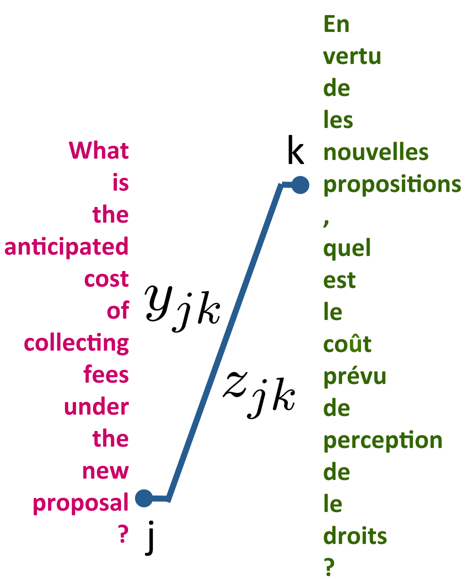
Matching Inference LP

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

Need Hamming-like loss

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

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



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

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 [illegible] constraints
  - Constraints [illegible] variables
- Optimal values are the same
  - When both feasible regions are bounded

$$
\begin{aligned}
\max_{\mathbf{z}} \quad & \mathbf{c}^\top \mathbf{z} \\
\text{s.t.} \quad & \mathbf{A}\mathbf{z} \leq \mathbf{b}; \\
& \mathbf{z} \geq 0.
\end{aligned}
$$

$$\Longleftrightarrow$$

$$
\begin{aligned}
\min_{\lambda} \quad & \mathbf{b}^\top \lambda \\
\text{s.t.} \quad & \mathbf{A}^\top \lambda \geq \mathbf{c}; \\
& \lambda \geq 0.
\end{aligned}
$$

Min-max Formulation

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

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

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

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

LP duality

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

$$\text{s.t.} \quad \mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) + \xi_i \geq \mathbf{b}_i^\top \lambda_i,$$

$$\mathbf{A}_i^\top \lambda_i \geq \mathbf{q}_i; \quad \lambda_i \geq 0$$

Min-max formulation summary

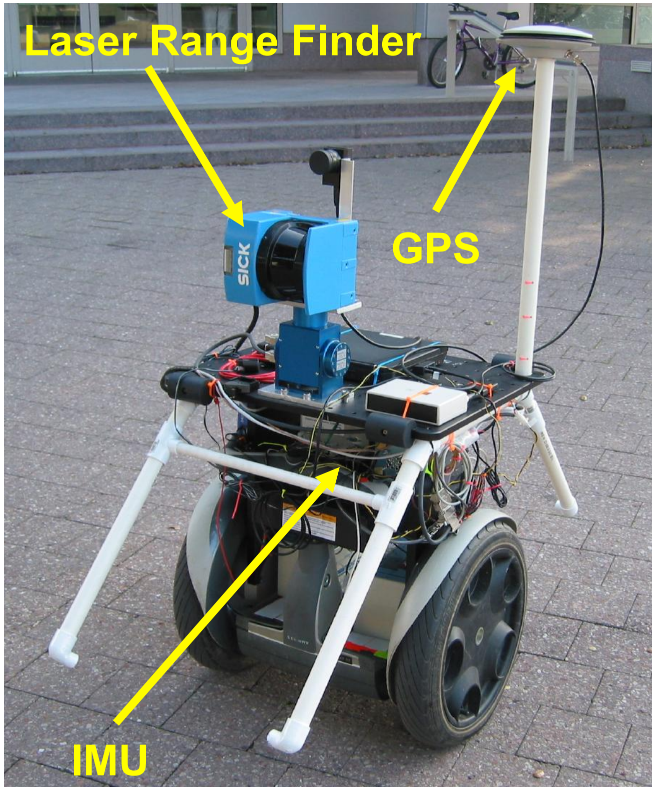
$$\min_{\mathbf{w},\lambda} \quad \frac{1}{2}||\mathbf{w}||^2 + C\left(\sum_i \; \mathbf{b}_i^\top \lambda_i - \mathbf{w}^\top \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i)\right)$$

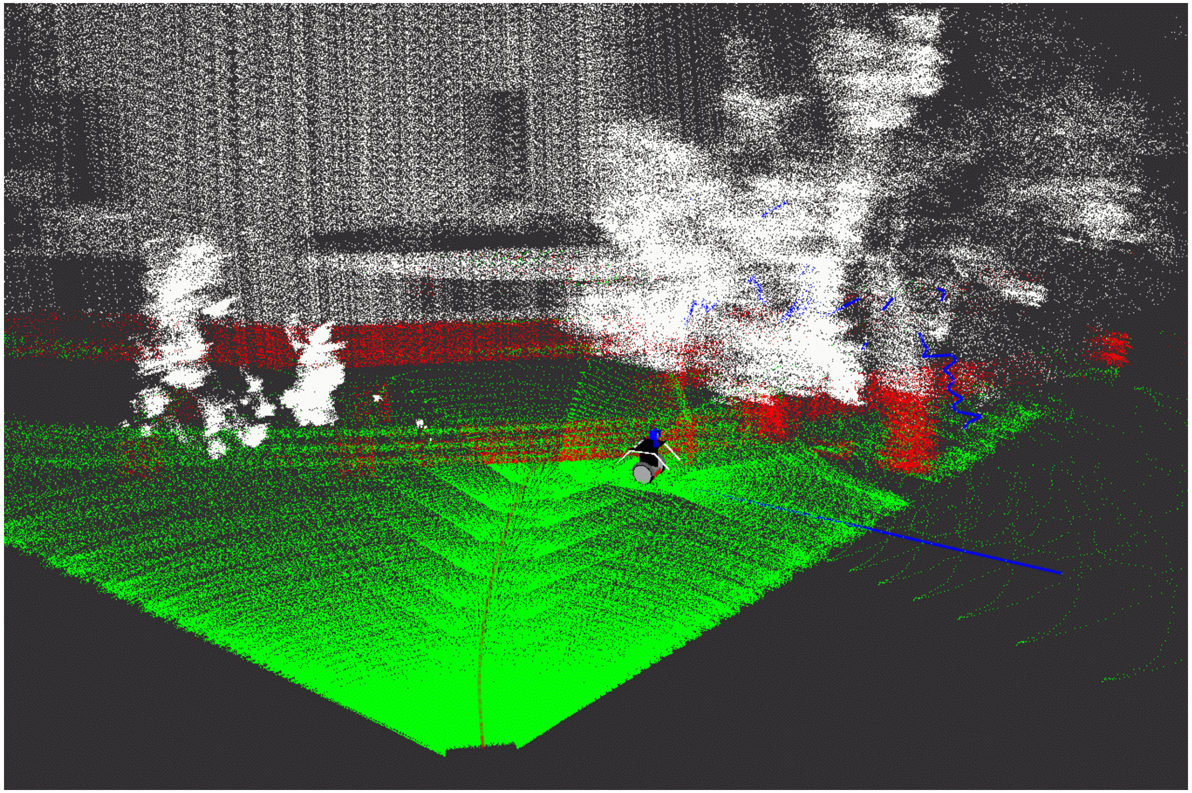
$$\text{s.t.} \quad \mathbf{A}_i^\top \lambda_i \geq \mathbf{F}_i^\top \mathbf{w} + \ell_i; \qquad \lambda_i \geq 0, \ \forall i.$$

*Taskar et al 04

3D Mapping

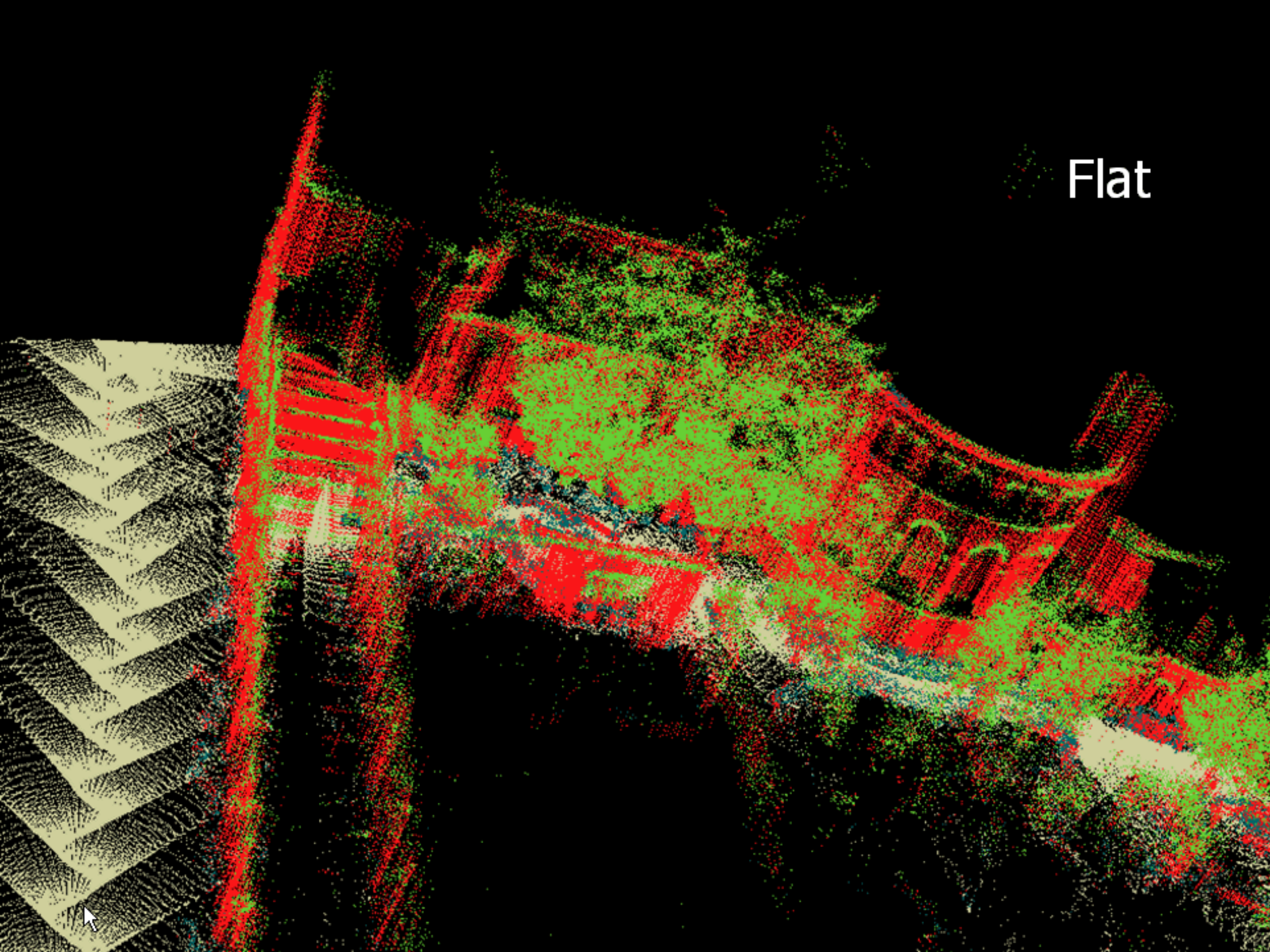
Data provided by: Michael Montemerlo & Sebastian Thrun

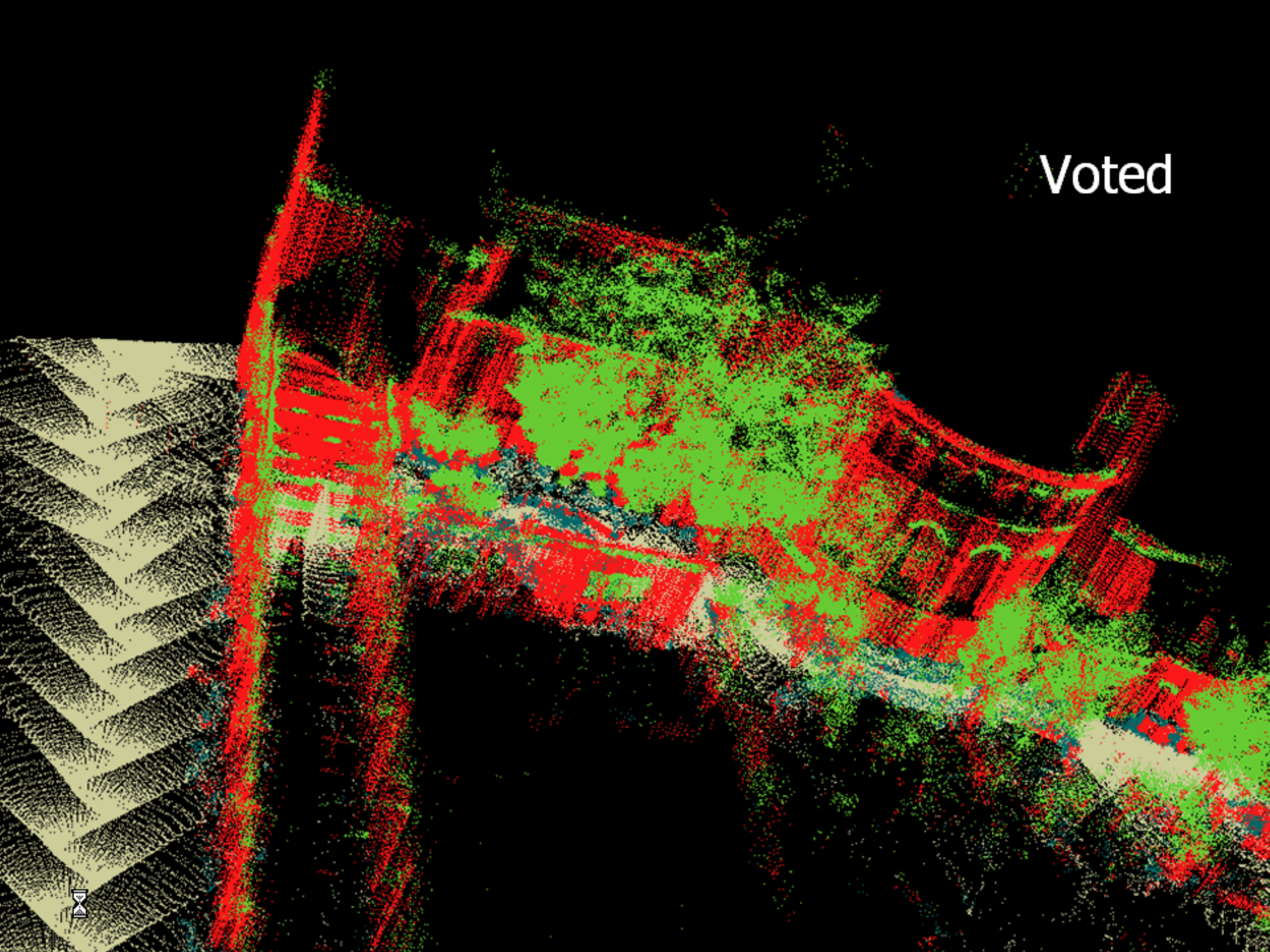


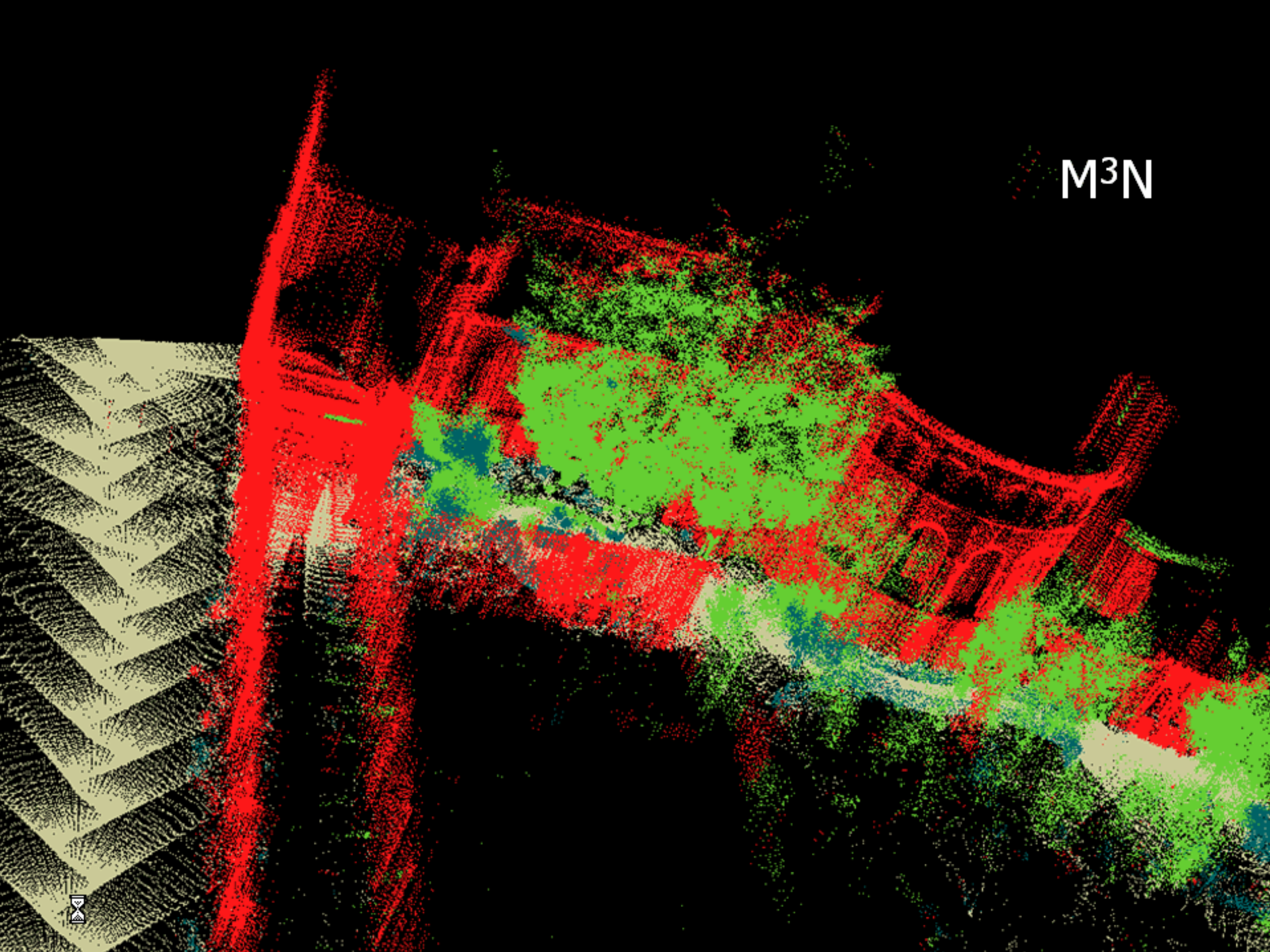


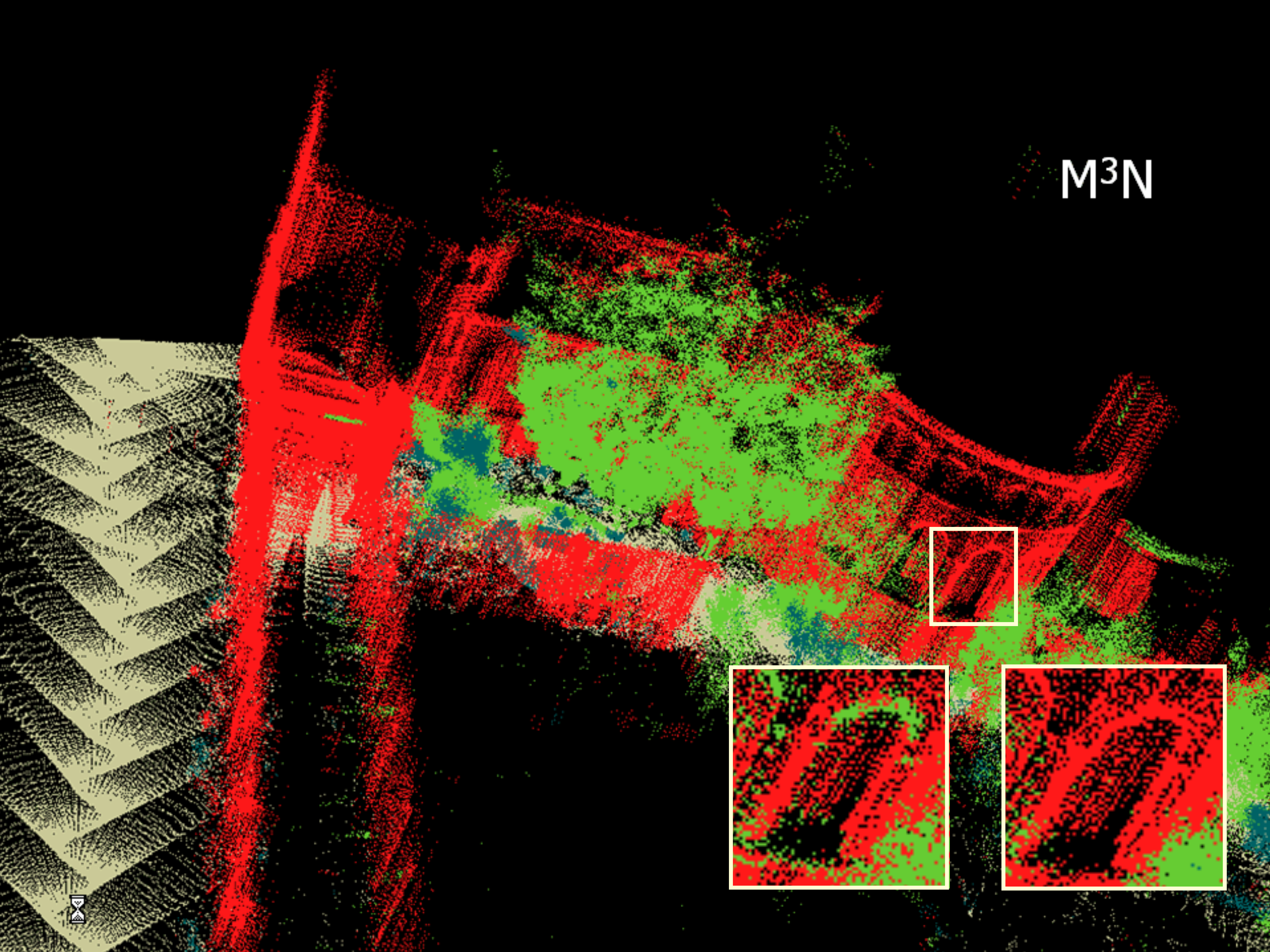
Label: ground, building, tree, shrub

Training: 30 thousand points   Testing: 3 million points


Flat

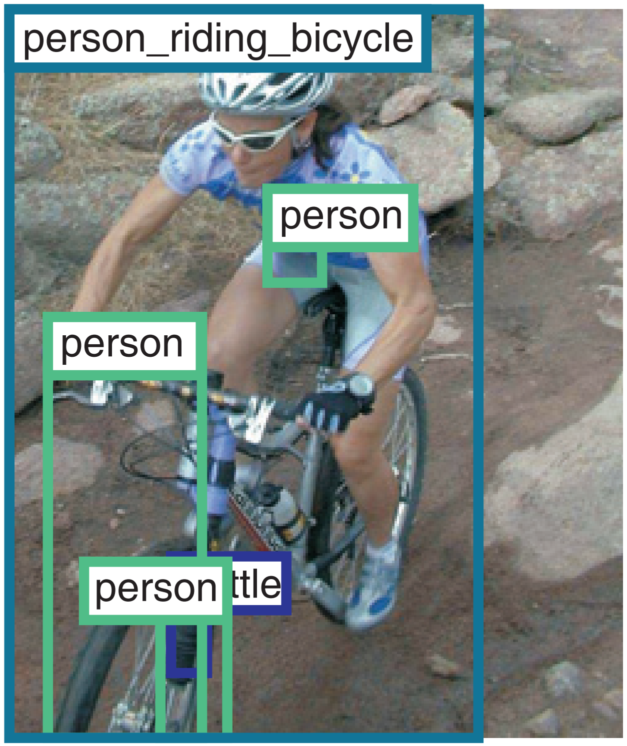

Voted


M³N

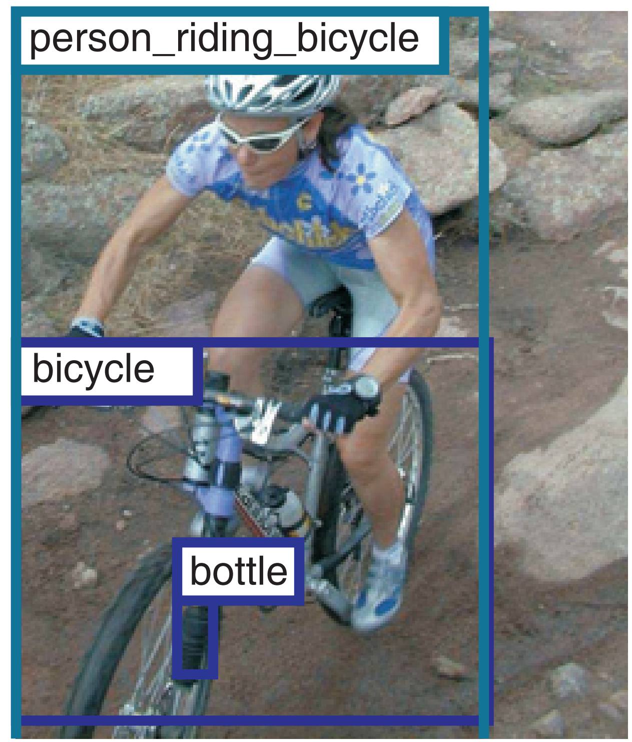

M³N

Before and After

Before Decoding



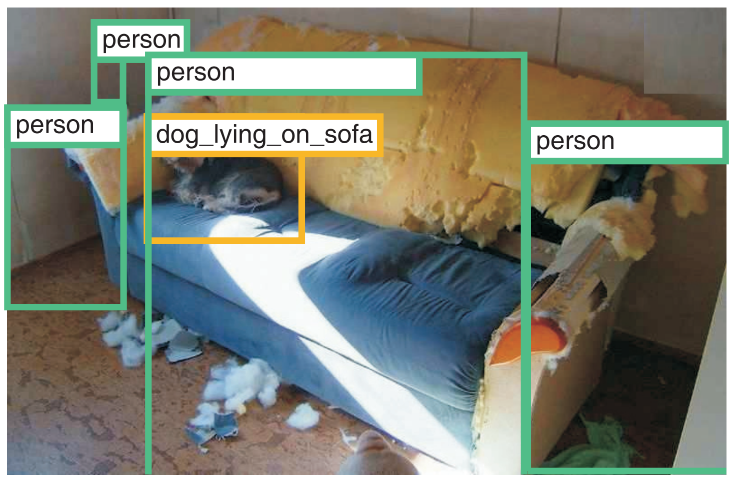
After Decoding



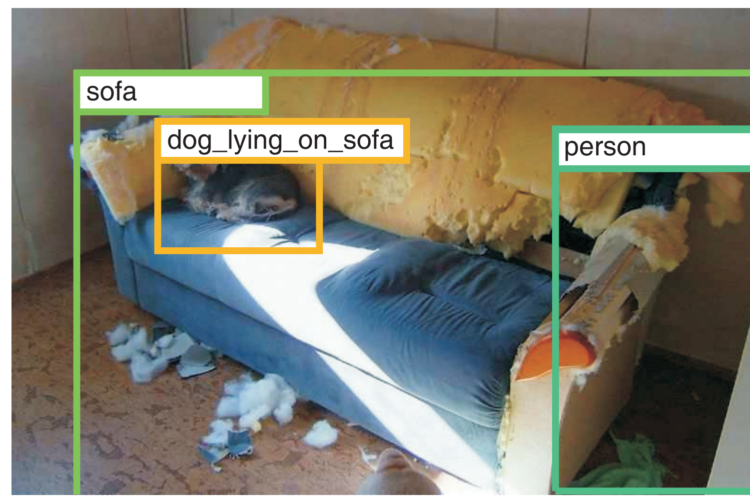
Recognition using Visual Phrases , CVPR 2011

Before and After

Before Decoding



After Decoding



Recognition using Visual Phrases , CVPR 2011