# Lecture-18

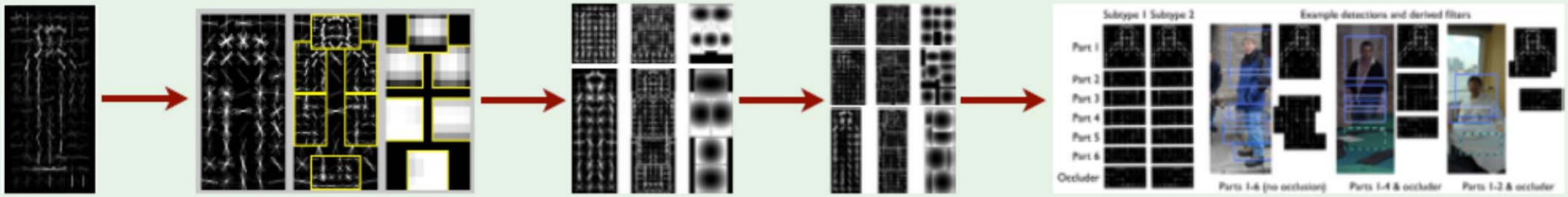
Deformable Part Model (DPM)

# Object Detection with Deformable Part Models

Cited by 2333

(Slides Credits: Girshick, R. B, Yicong Tian and many others)

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. *Object detection with discriminatively trained part-based models*. In PAMI 2010. http://www.cs.berkeley.edu/~rbg/latent/




| | | | | | |
|---|---|---|---|---|---|
| AP | 12% | 27% | 36% | 45% | 49% |
| | 2005 | 2008 | 2009 | 2010 | 2011 |

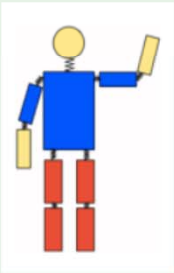
Person detection performance on PASCAL VOC 2007

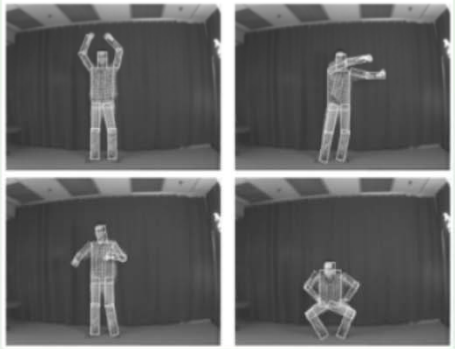
Slide Credit: ., Girshick, R. B

## The Origins of DPM

**Dalal & Triggs '05**

- Histrogram of Oriented Gradients (HOG)
- SVM training
- Sliding window detection





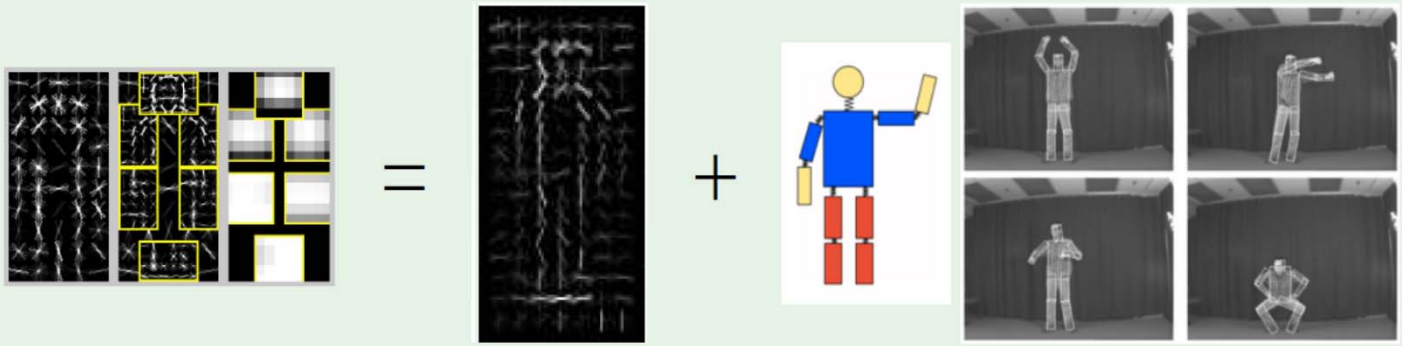
**Fischler & Elschlager '73**
**Felzenszwalb & Huttenlocher '00**

- Pictorial structures
- Weak appearance models
- Non-discriminative training

Slide Credit: ., Girshick, R. B

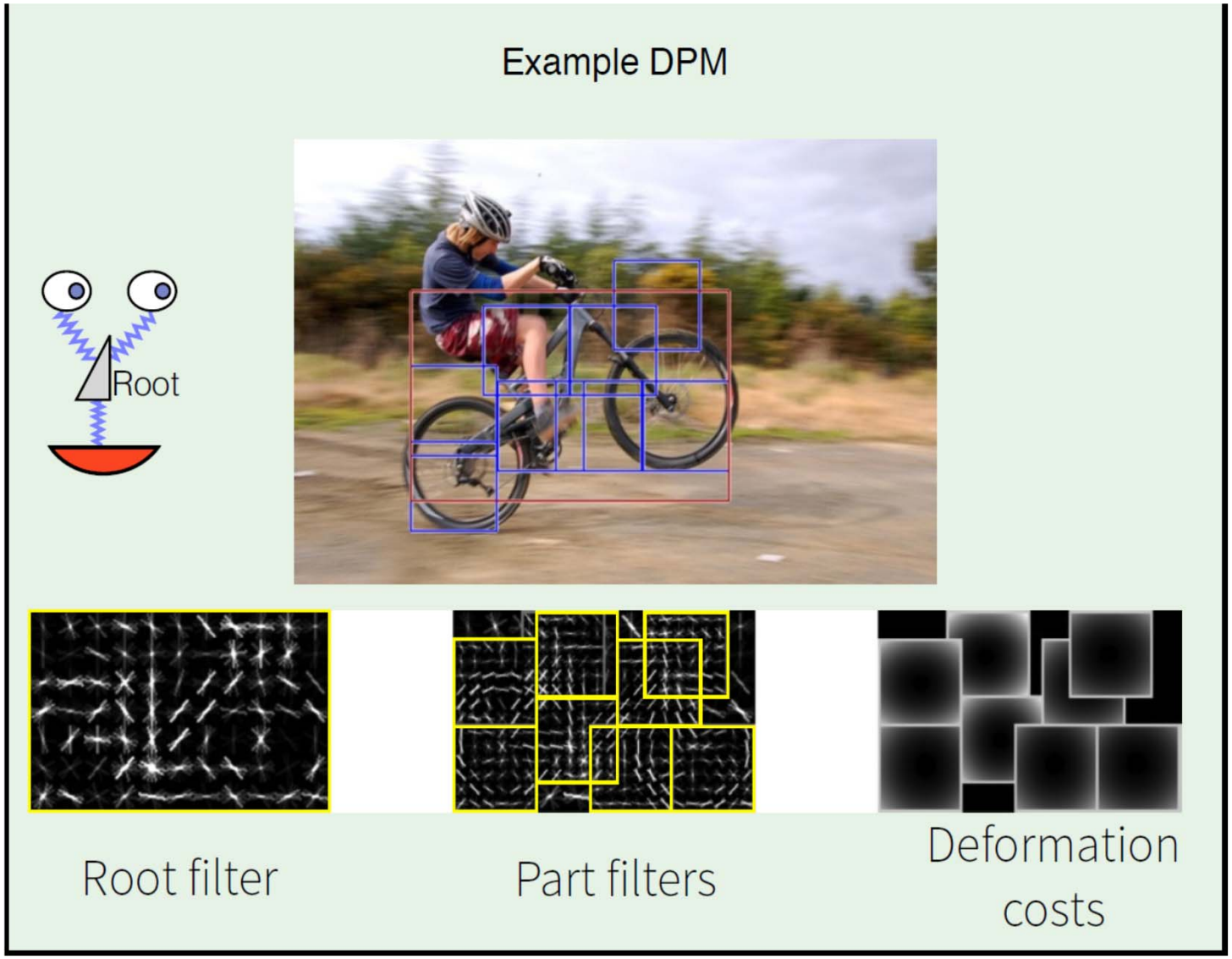
## DPM key idea

Port the success of Dalal & Triggs
into a part-based model




DPM D&T PS

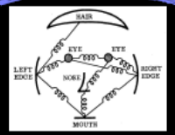
Slide Credit: ., Girshick, R. B



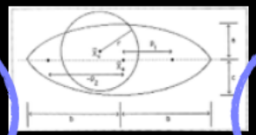

Slide Credit: ., Girshick, R. B

# DPM in the design trade space

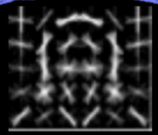
What "deforms"?



deformable configuration of parts

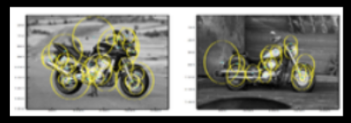


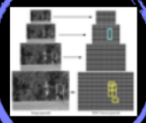
parts that deform



rigid parts on deformation invariant features

Sparse vs. dense

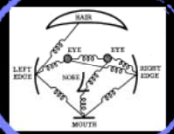




Generative vs. discriminative

Probabilistic vs. non-probabilistic

Explicit vs. implicit structure

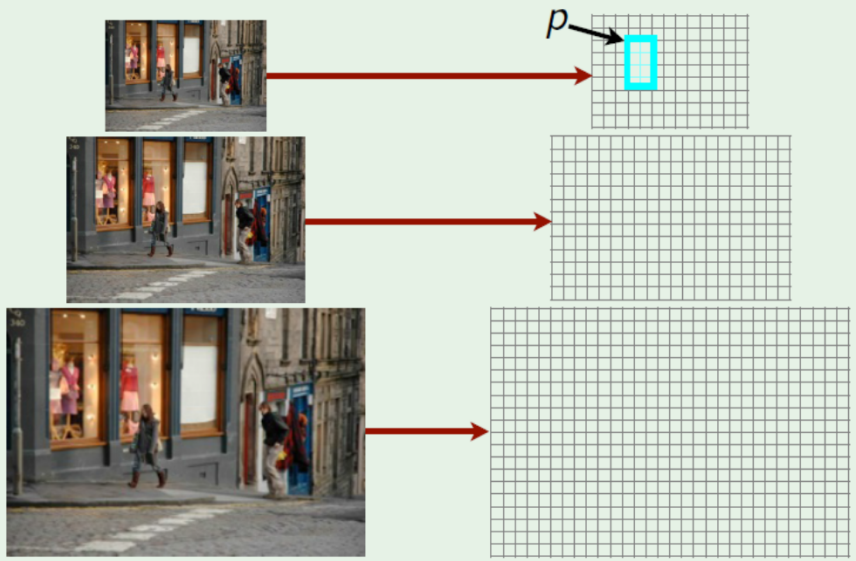


Local appearance model: templates, densities, …

Translation, scale & rotation invariance

Slide Credit: ., Girshick, R. B

# The Dalal & Triggs detector




$$score(I, p) = \mathbf{w} \cdot \phi(I, p)$$

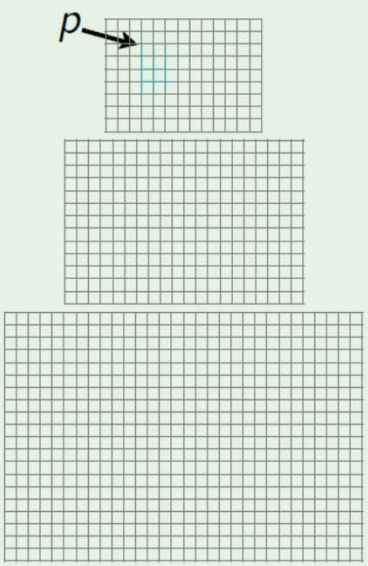
Image pyramid

HOG feature pyramid

- Compute HOG of the whole image at multiple resolutions
- Score every subwindow of the feature pyramid
- Apply non-maximal suppression

Slide Credit: ., Girshick, R. B

# Detection




number of locations $p$ ~ 250,000 per image

test set has ~ 5000 images

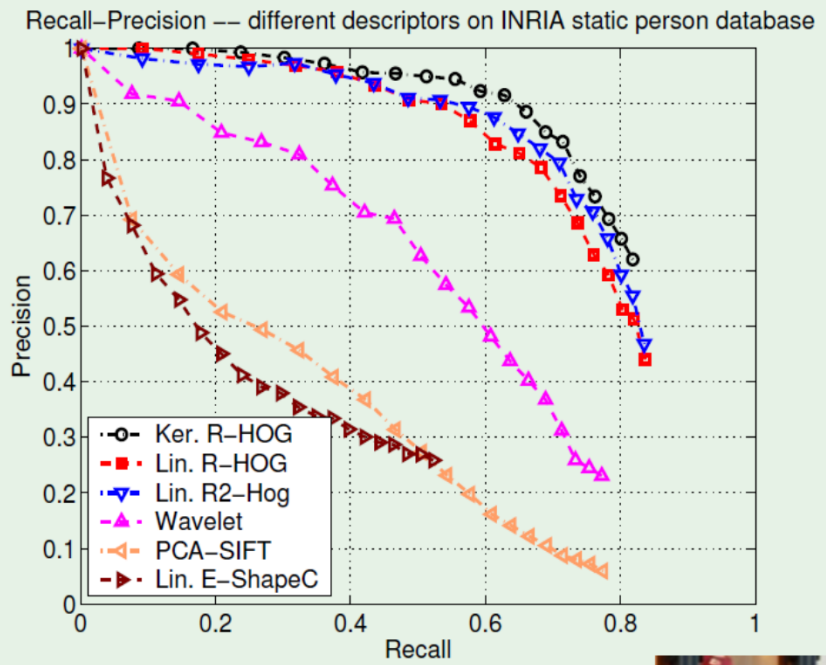
>> $1.3 \times 10^9$ windows to classify

typically only ~ 1,000 true positive locations

*Extremely unbalanced binary classification*

Slide Credit: ., Girshick, R. B

# Dalal & Triggs detector on INRIA pedestrians

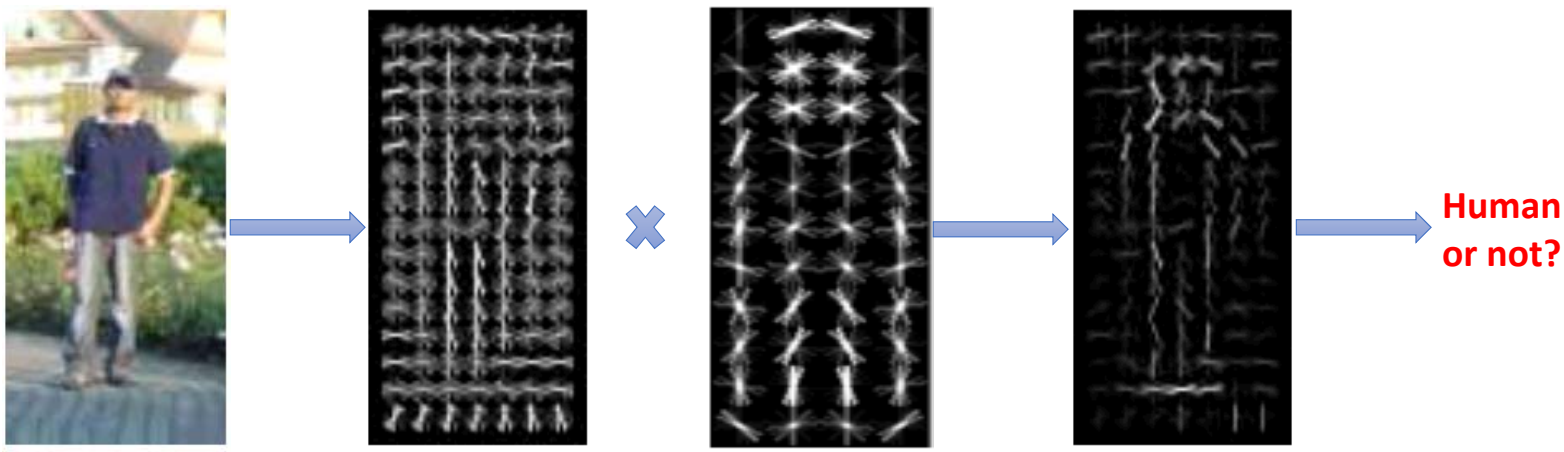



- AP = 75%
  - (79% in my implementation)
- Very good
  - **Declare victory and go home?**

Slide Credit: ., Girshick, R. B

# Object Detection with HOG

- Dalal & Triggs detector: HOG + linear SVM



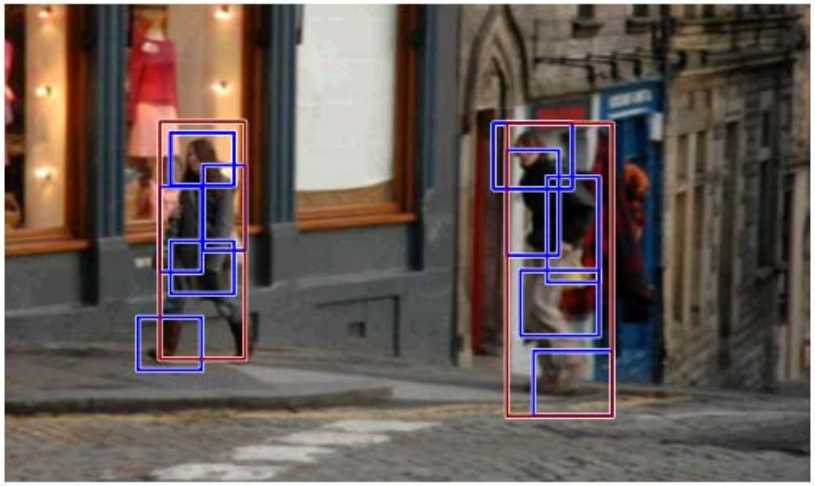

Test image | Extracted HOG feature | The human model (*W*) | HOG feature weighted by the human model

Dalal, N., & Triggs, B. *Histograms of oriented gradients for human detection*. In CVPR 2005.

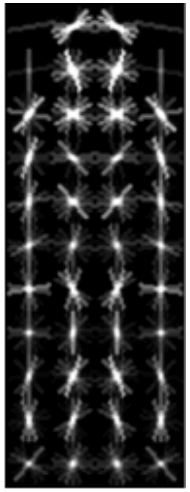
# Why DPM?

- Great variations in objects
  - Non-rigid deformations, e.g. human in different poses
  - Intra-class variability, e.g. cars in various shape
  - Variations caused by different viewpoints and illumination
- A better representation of objects – deformable part model (DPM)
  - The Dalal & Triggs detector acts as the root filter in DPM

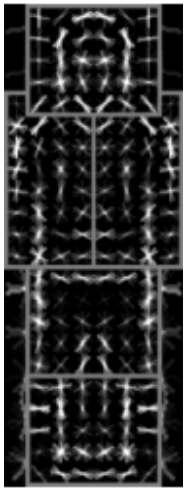
# Overview



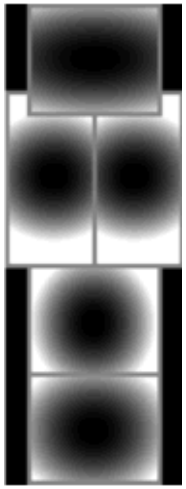
Example detection result



Coarse root filter

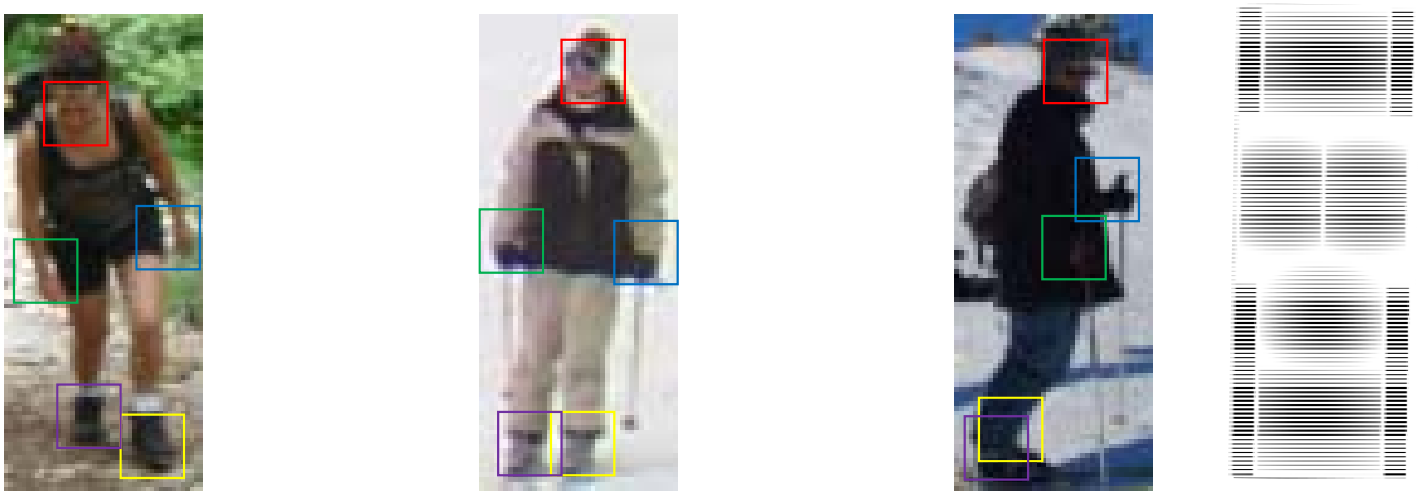


Higher resolution part filters



Deformation models

# Starting Point

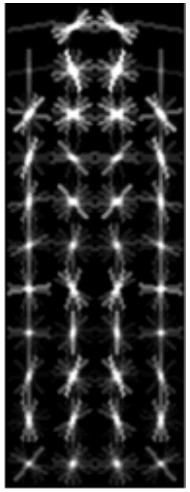


Deformation models

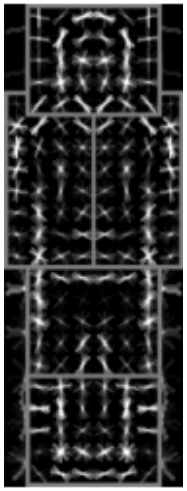
- Represent human by several parts
- The positions of the parts are not fixed, which means the model is deformable
- Penalize the parts that are far away from where they are supposed to be

# Model Representation

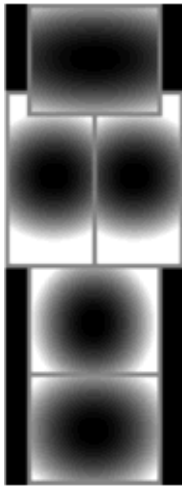
- A model has a root filter $F_0$ and $n$ part models $(F_i, v_i, d_i)$
  - $F_i$: $i$-th part filter
  - $v_i$: anchor position of $i$-th part relative to the root
  - $d_i$: deformation parameters for $i$-th part



Coarse root filter



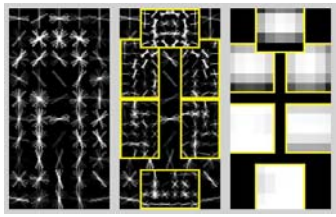
Higher resolution part filters

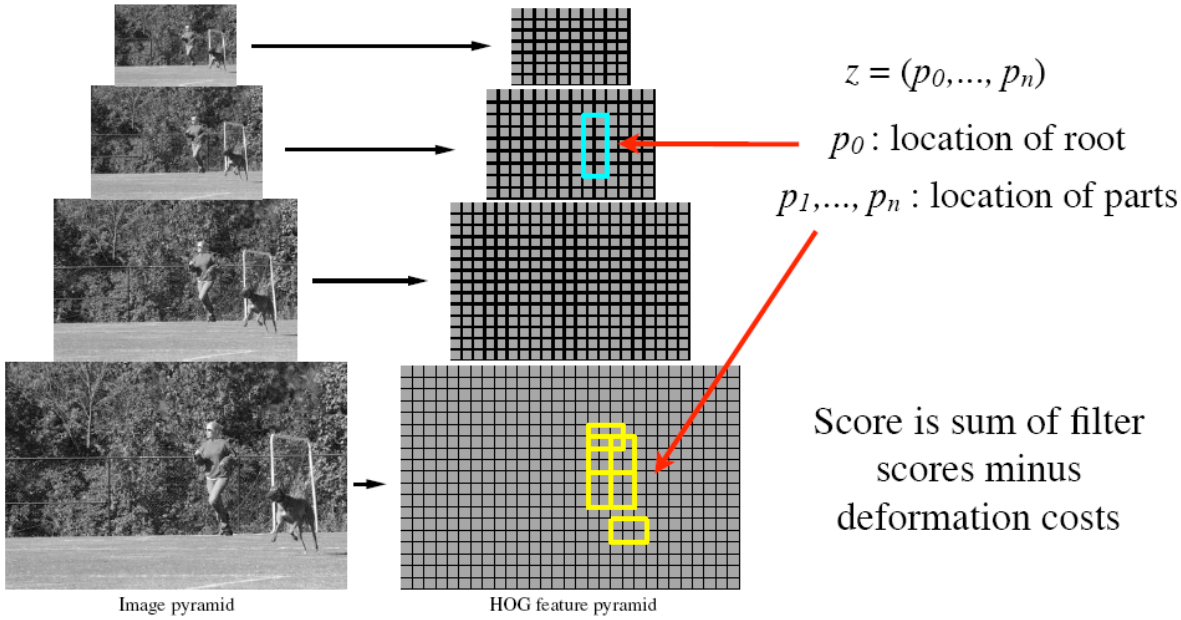


Deformation models

# Object Hypothesis

- In HOG feature pyramid
  - root filter - coarser scale
  - part filters - finer scale





# Score of a Hypothesis

$$score(p_0, \dots, p_n) = \underbrace{\boxed{\sum_{i=0}^{n} F_i \cdot \phi(H, p_i)}}_{\text{data term}} - \underbrace{\boxed{\sum_{i=1}^{n} d_i \cdot \phi_d(dx_i, dy_i)}}_{\text{spatial prior}} + b$$

Filters

Feature of subwindow at location $p_i$

Deformation parameters

Displacement of part i relative to its anchor position

Bias

Score of a hypothesis $z$ is

$$score(z) = \beta \cdot \psi(H, z)$$

where

$$\beta = (F_0, \dots, F_n, d_1, \dots, d_n, b)$$

Unknown

$$\psi(H, z) = (\phi(H, p_0), \dots, \phi(H, p_n), -\phi_d(dx_1, dy_1), \dots, -\phi_d(dx_n, dy_n), 1)$$

Known

# Score of a Hypothesis

Bias

data term

spatial prior

$$score(p_0, \dots, p_n) = \sum_{i=0}^{n} F_i \cdot \phi(H, p_i) - \sum_{i=1}^{n} d_i \cdot \phi_d(dx_i, dy_i) + b$$

Filters

Feature of subwindow at location $p_i$

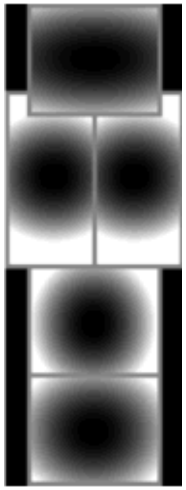
Deformation parameters

Displacement of part i relative to its anchor position

$$\phi_d(dx, dy) = (dx, dy, dx^2, dy^2)$$

$$d_i = (0, 0, 1, 1)$$

Initial Value (to be learned)



Deformation models

# Detection

- The overall score of a root location is computed according to the best possible placement of the parts

$$score(p_0) = \max_{p_1,\dots,p_n} score(p_0, \dots, p_n)$$

- High-scoring root locations define detections
- Sliding-window approach
- Efficient computation (O(nk)): dynamic programming + generalized distance transforms

# Detection

- Distance transform
  - Response of the $i$-th part filter in the $l$-th level of the feature pyramid

$$R_{i,l}(x,y) = F_i \cdot \phi(H,(x,y,l))$$

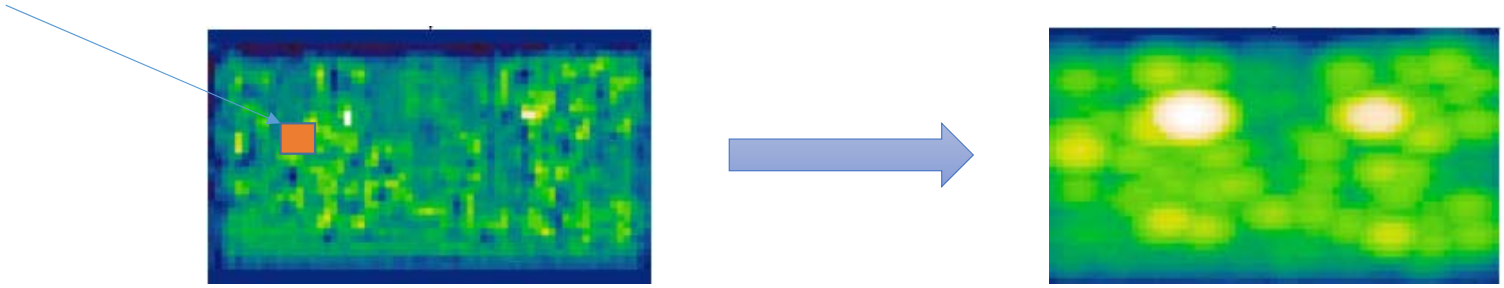
$$\phi_d(dx,dy) = (dx, dy, dx^2, dy^2)$$

  - Transformed response, given root is at *(x,y)*

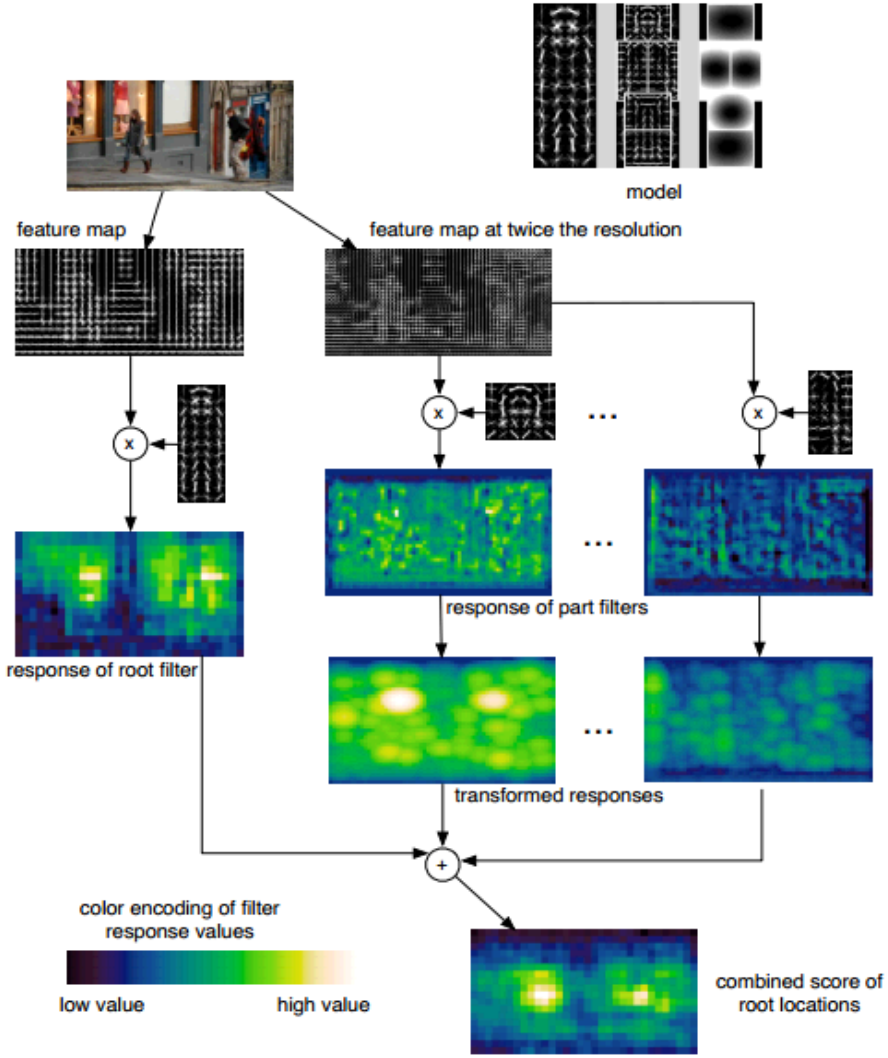
$$D_{i,l}(x,y) = \max_{dx,dy}(R_{i,l}(x+dx, y+dy) - d_i \cdot \phi_d(dx,dy))$$

*(x,y)*

$$d_i = (0,0,1,1)$$

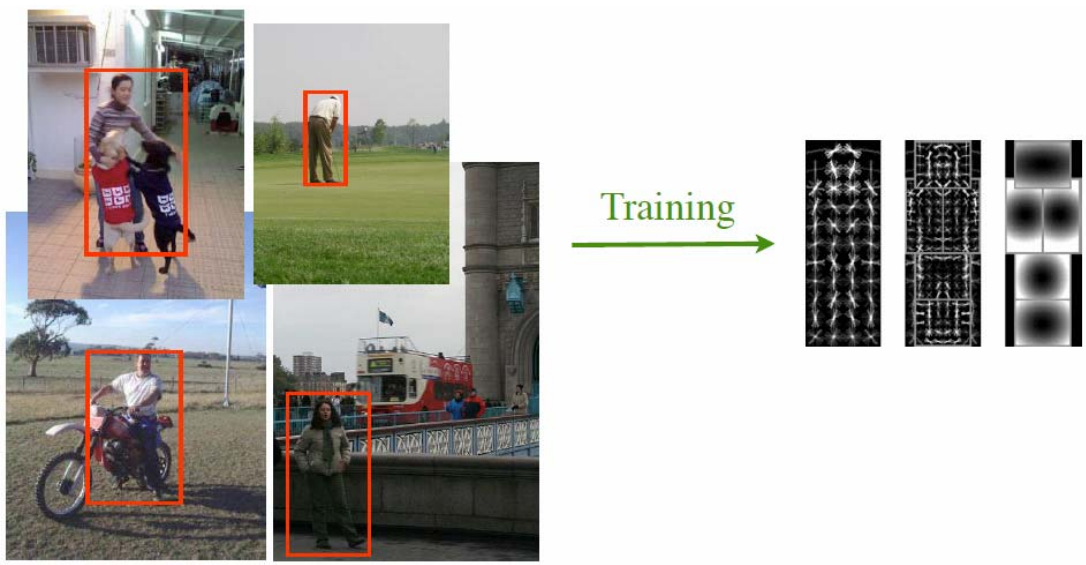


# Detection

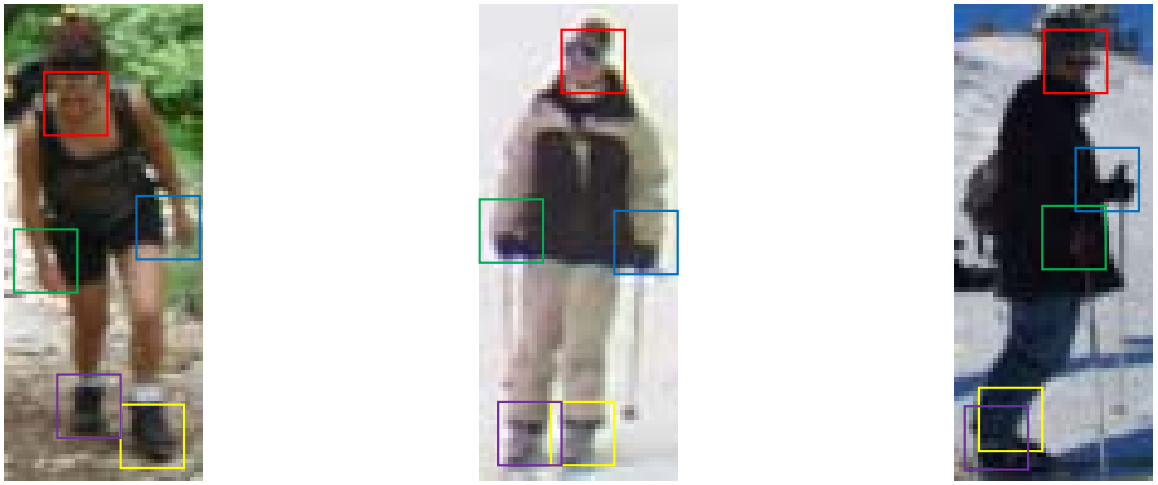

model
feature map
feature map at twice the resolution
response of part filters
response of root filter
transformed responses
color encoding of filter
response values
low value
high value
combined score of
root locations

# Training

- Positive training examples are labeled with bounding boxes
- No part location is available during training (latent)
- Aim: learn model parameters $\beta = (F_0, \ldots, F_n, d_1, \ldots, d_n, b)$



# Latent Variables



- The positions of the parts are not given in both the training and the testing images
- The variables that exist but not known in training samples are called latent variables
- The learning algorithm must be able to find/discover the optimal values for the latent variables, namely the position of the parts.

# Training

- The classifier scores an example $x$ by

$$f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z)$$

  - $\beta$: the model parameters
  - $z$: latent values
  - $Z(x)$: the possible latent values for example $x$

# Training

- Minimize the objective function

$$L_D(\beta) = \frac{1}{2} \parallel \beta \parallel^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i))$$

  - Labeled training examples $D = (\langle x_1, y_1 \rangle, \ldots, \langle x_n, y_n \rangle)$
  - $y_i \in \{-1, 1\}$

# Optimization

Learning an SVM has been formulated as a constrained optimization problem over $\mathbf{w}$ and $\xi$

$$\min_{\mathbf{w}\in\mathbb{R}^d, \xi_i\in\mathbb{R}^+} ||\mathbf{w}||^2 + C\sum_i^N \xi_i \text{ subject to } y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1 - \xi_i \text{ for } i = 1\ldots N$$

The constraint $y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1 - \xi_i$, can be written more concisely as

$$y_i f(\mathbf{x}_i) \geq 1 - \xi_i$$

which, together with $\xi_i \geq 0$, is equivalent to

$$\xi_i = \max\left(0, 1 - y_i f(\mathbf{x}_i)\right)$$

Hence the learning problem is equivalent to the unconstrained optimization problem over $\mathbf{w}$

$$\min_{\mathbf{w}\in\mathbb{R}^d} \underbrace{||\mathbf{w}||^2}_{\text{regularization}} + C\sum_i^N \underbrace{\max\left(0, 1 - y_i f(\mathbf{x}_i)\right)}_{\text{loss function}}$$

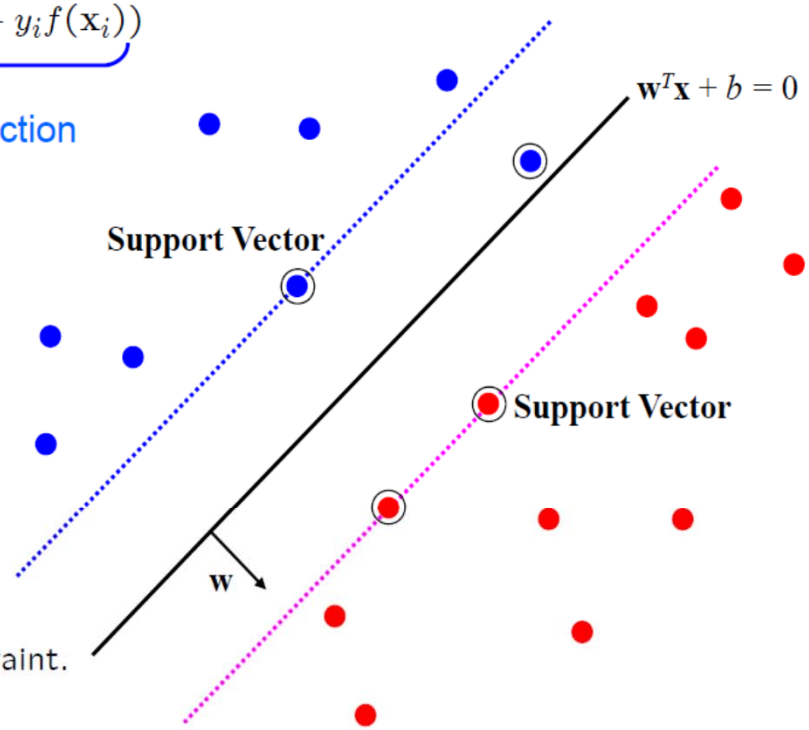
Slide Credit: A. Zisserman

# Loss function

$$\min_{\mathbf{w}\in\mathbb{R}^d} ||\mathbf{w}||^2 + C\sum_i^N \underbrace{\max(0, 1 - y_i f(\mathbf{x}_i))}_{\text{loss function}}$$

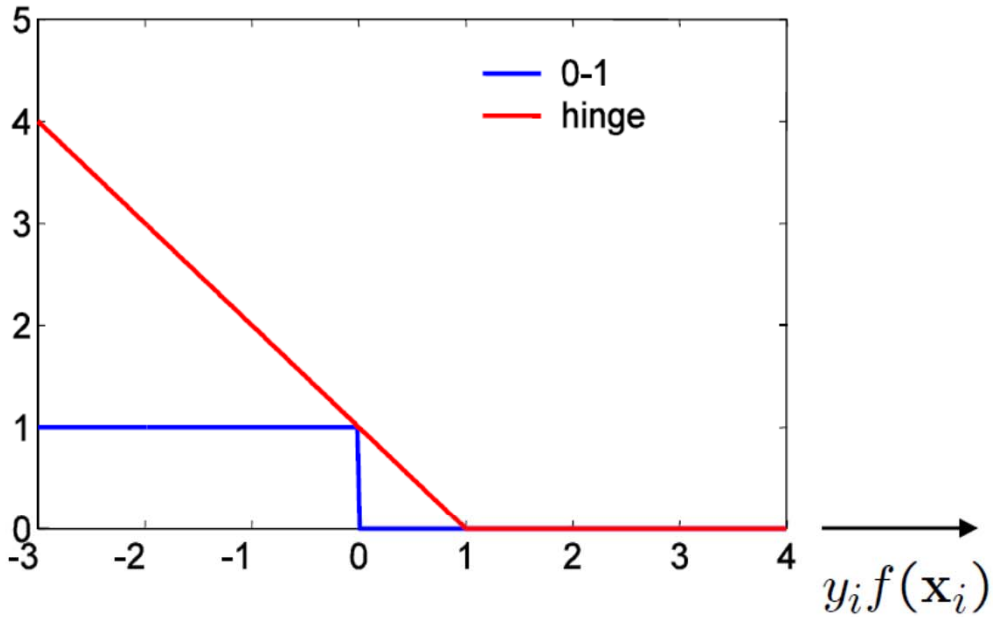
Points are in three categories:

1. $y_i f(x_i) > 1$
   Point is outside margin.
   No contribution to loss
2. $y_i f(x_i) = 1$
   Point is on margin.
   No contribution to loss.
   As in hard margin case.
3. $y_i f(x_i) < 1$
   Point violates margin constraint.
   Contributes to loss




Slide Credit: A. Zisserman

# Loss functions

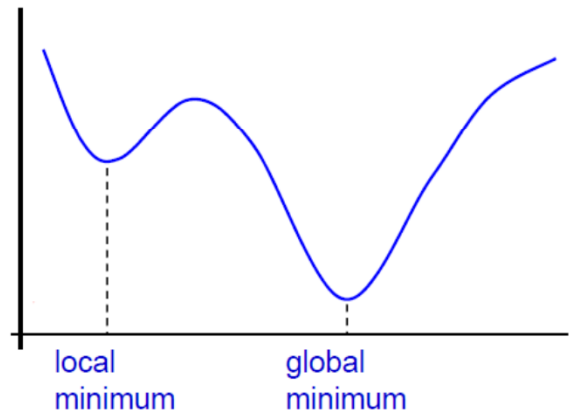



- SVM uses "hinge" loss $\max(0, 1 - y_i f(\mathbf{x}_i))$
- an approximation to the 0-1 loss

Slide Credit: A. Zisserman

# Optimization continued

$$\min_{\mathbf{w}\in\mathbb{R}^d} C\sum_i^N \max\left(0, 1 - y_i f(\mathbf{x}_i)\right) + ||\mathbf{w}||^2$$




- Does this cost function have a unique solution?
- Does the solution depend on the starting point of an iterative optimization algorithm (such as gradient descent)?

If the cost function is convex, then a locally optimal point is globally optimal (provided the optimization is over a convex set, which it is in our case)

Slide Credit: A. Zisserman

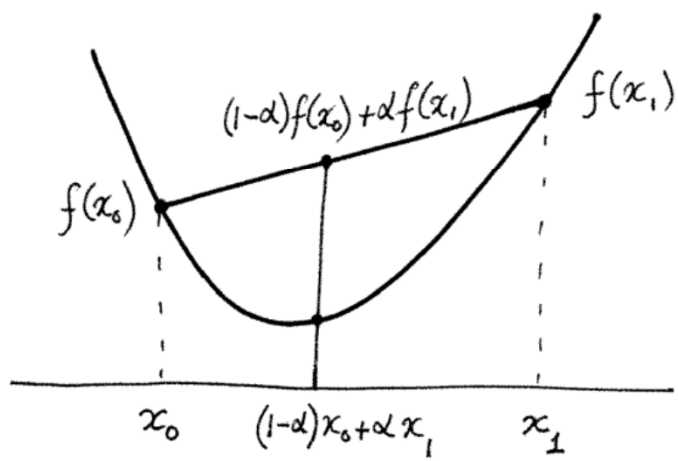
# Convex functions

$D$ – a domain in $\mathbb{R}^n$.

A convex function $f : D \to \mathbb{R}$ is one that satisfies, for any $\mathbf{x}_0$ and $\mathbf{x}_1$ in $D$:

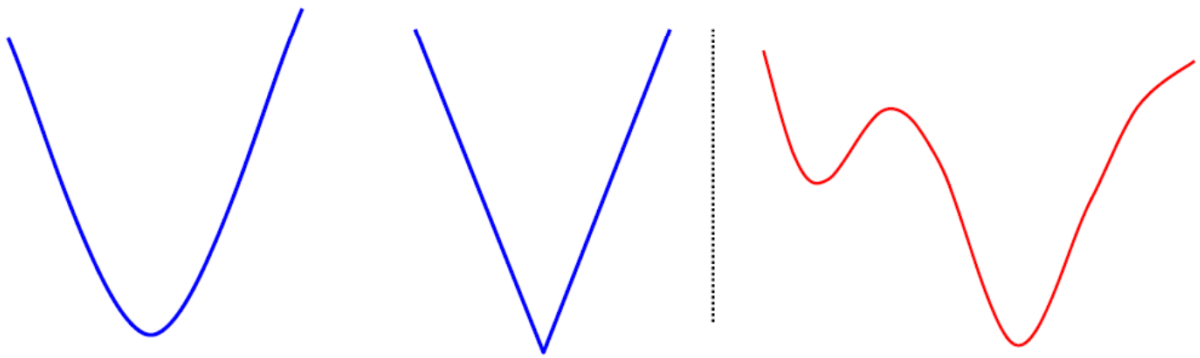
$$f((1-\alpha)\mathbf{x}_0 + \alpha\mathbf{x}_1) \leq (1-\alpha)f(\mathbf{x}_0) + \alpha f(\mathbf{x}_1) \ .$$

Line joining $(\mathbf{x}_0, f(\mathbf{x}_0))$ and $(\mathbf{x}_1, f(\mathbf{x}_1))$ lies above the function graph.




Slide Credit: A. Zisserman

# Convex function examples



convex

Not convex

A non-negative sum of convex functions is convex

Slide Credit: A. Zisserman

SVM

$$\min_{\mathbf{w}\in\mathbb{R}^d} C\sum_i^N \max\left(0, 1 - y_i f(\mathbf{x}_i)\right) + ||\mathbf{w}||^2$$

convex

Slide Credit: A. Zisserman

# Gradient (or steepest) descent algorithm for SVM

To minimize a cost function $\mathcal{C}(\mathbf{w})$ use the iterative update

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \nabla_{\mathbf{w}} \mathcal{C}(\mathbf{w}_t)$$

where $\eta$ is the learning rate.

First, rewrite the optimization problem as an average

$$\begin{aligned} \min_{\mathbf{w}} \mathcal{C}(\mathbf{w}) &= \frac{\lambda}{2}||\mathbf{w}||^2 + \frac{1}{N}\sum_i^N \max(0, 1 - y_i f(\mathbf{x}_i)) \\ &= \frac{1}{N}\sum_i^N \left(\frac{\lambda}{2}||\mathbf{w}||^2 + \max(0, 1 - y_i f(\mathbf{x}_i))\right) \end{aligned}$$

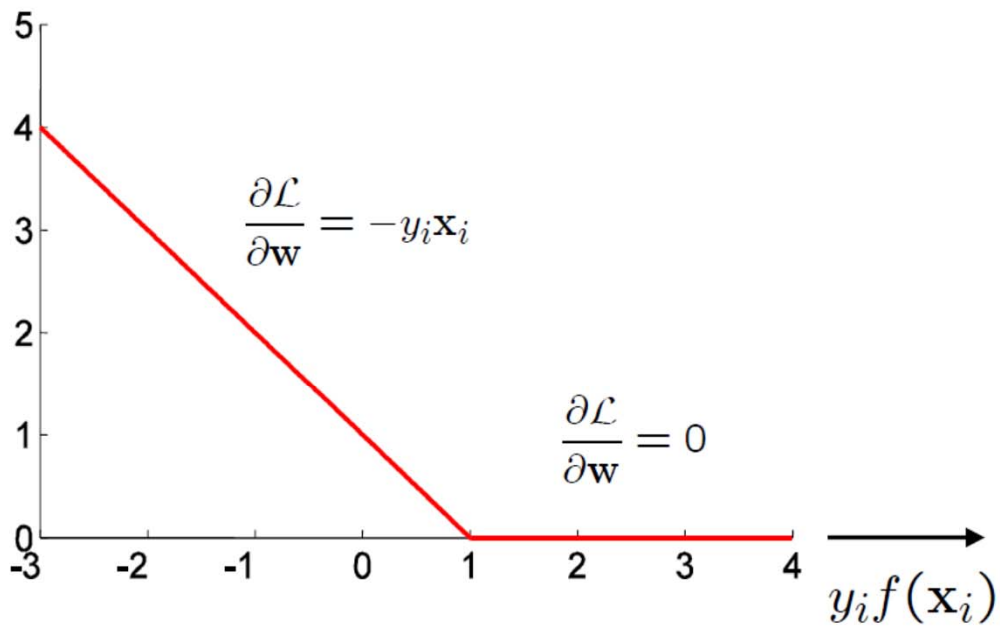
(with $\lambda = 2/(NC)$ up to an overall scale of the problem) and $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$

Because the hinge loss is not differentiable, a sub-gradient is computed

Slide Credit: A. Zisserman

# Sub-gradient for hinge loss

$$\mathcal{L}(\mathbf{x}_i, y_i; \mathbf{w}) = \max(0, 1 - y_i f(\mathbf{x}_i)) \qquad f(\mathbf{x}_i) = \mathbf{w}^\top \mathbf{x}_i + b$$




Slide Credit: A. Zisserman

# Sub-gradient descent algorithm for SVM

$$\mathcal{C}(\mathbf{w}) = \frac{1}{N}\sum_{i}^{N}\left(\frac{\lambda}{2}||\mathbf{w}||^2 + \mathcal{L}(\mathbf{x}_i, y_i; \mathbf{w})\right)$$

The iterative update is

$$\begin{aligned}\mathbf{w}_{t+1} &\leftarrow \mathbf{w}_t - \eta\nabla_{\mathbf{w}_t}\mathcal{C}(\mathbf{w}_t) \\ &\leftarrow \mathbf{w}_t - \eta\frac{1}{N}\sum_{i}^{N}(\lambda\mathbf{w}_t + \nabla_{\mathbf{w}}\mathcal{L}(\mathbf{x}_i, y_i; \mathbf{w}_t))\end{aligned}$$

where $\eta$ is the learning rate.

Then each iteration $t$ involves cycling through the training data with the updates:

$$\begin{aligned}\mathbf{w}_{t+1} &\leftarrow \mathbf{w}_t - \eta(\lambda\mathbf{w}_t - y_i\mathbf{x}_i) && \text{if } y_i f(\mathbf{x}_i) < 1 \\ &\leftarrow \mathbf{w}_t - \eta\lambda\mathbf{w}_t && \text{otherwise}\end{aligned}$$

In the Pegasos algorithm the learning rate is set at $\eta_t = \frac{1}{\lambda t}$

Slide Credit: A. Zisserman

# Latent SVM

- A latent SVM is semi-convex
  - $f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z)$ is convex in $\beta$
  - For negative examples ($y_i = -1$), the hinge loss is convex

$$\max\left(0,1 - y_i f_\beta(x_i)\right) = \max(0,1 + f_\beta(x_i))$$

(the maximum of two convex function)

  - For positive examples ($y_i = 1$), the hinge loss is not convex

$$\max\left(0,1 - y_i f_\beta(x_i)\right) = \max(0,1 - f_\beta(x_i))$$

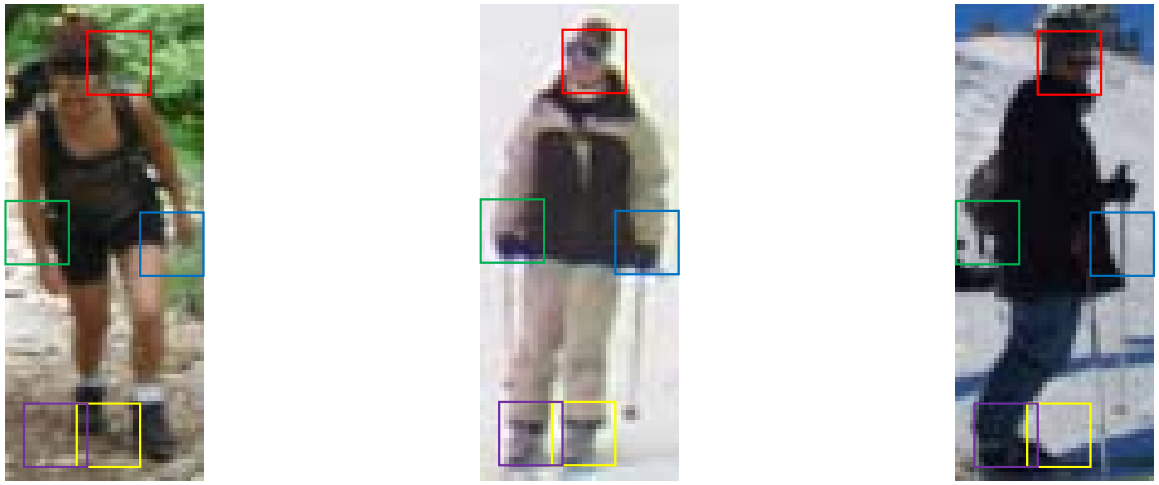
(the maximum of a convex function and a concave function)

  - If the latent value for positive examples are fixed, the hinge loss is convex

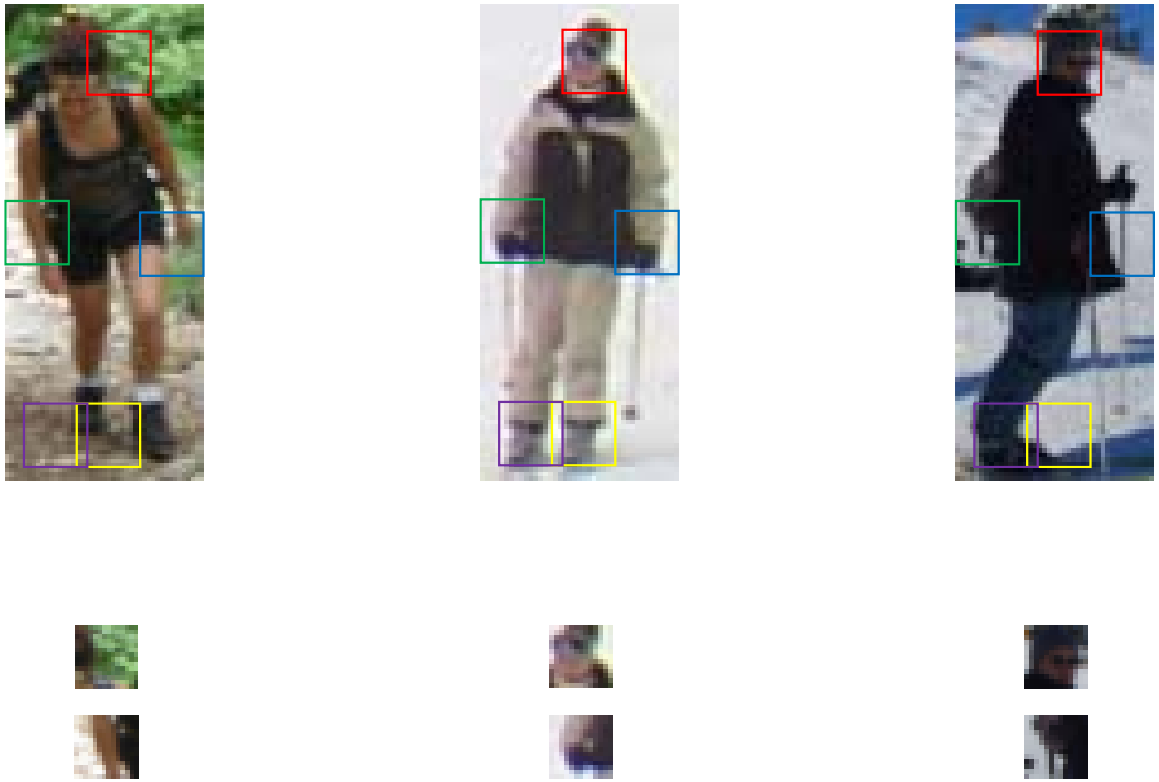
# Latent SVM

- Initialize $\beta$ using standard SVM by assuming the same parts locations for all the positive examples
- Iterative optimization:
  - Relabel positive examples: fix $\beta$, find the best $z$ for each positive example (exactly the same with detection!)
  - Optimize $\beta$: fix $z$, optimize $\beta$ by solving the convex problem

# Training

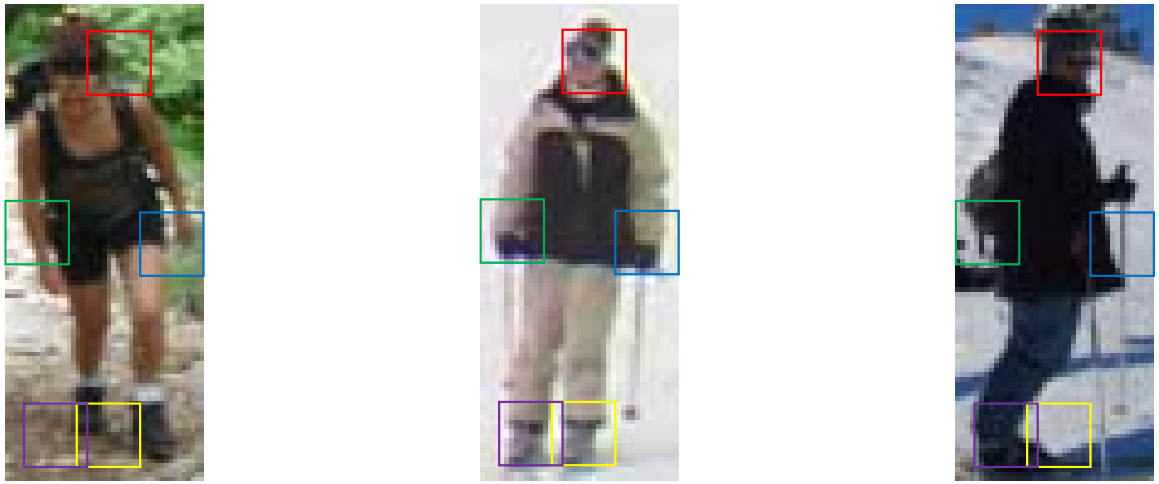


Step 1. we suppose the positions of the parts are fixed for each frame, and train part detector using SVM



Train SVM Classifier

# Training



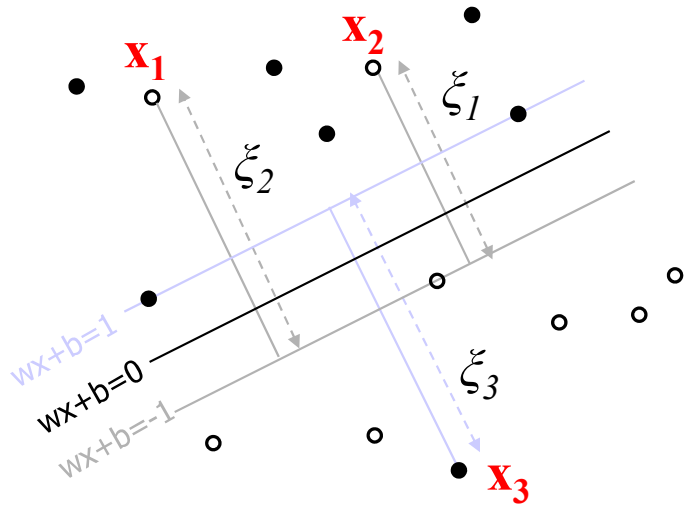
Step 2. Using the trained SVM classifier, find the optimal part position in each training sample

# Negative examples

- A very large number of negative examples
  - Infeasible to consider all of them in training!
- Data-mining hard examples

$$H(\beta, D) = \left\{ \left(i, \Phi(x_i, z_i)\right) \middle| z_i = \underset{z \in Z(x_i)}{\operatorname{argmax}} \beta \cdot \Phi(x_i, z) and y_i \left(\beta \cdot \Phi(x_i, z_i)\right) < 1 \right\}$$

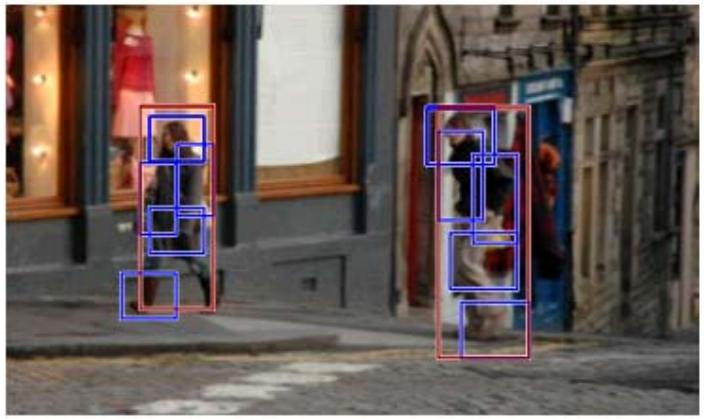
# Hard Negatives

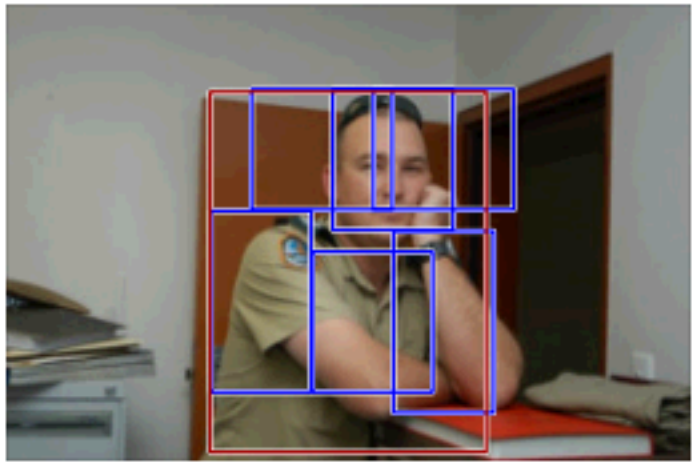


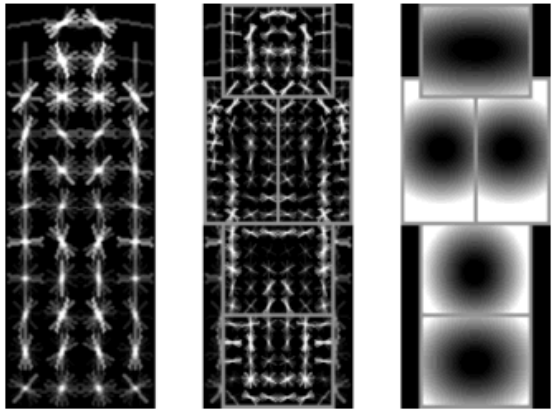
# Mixture Model

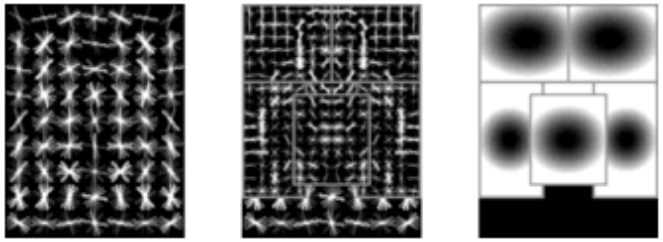
- A mixture model consists of $m$ components
- Captures extreme intra-class variation
- Split the positive bounding boxes into $m$ groups by aspect ratio

# Mixture Model Example - Person

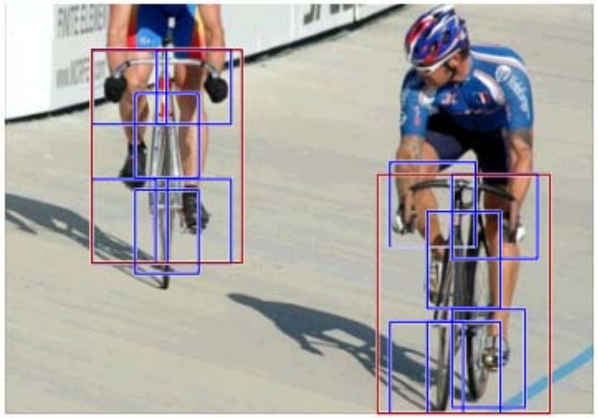


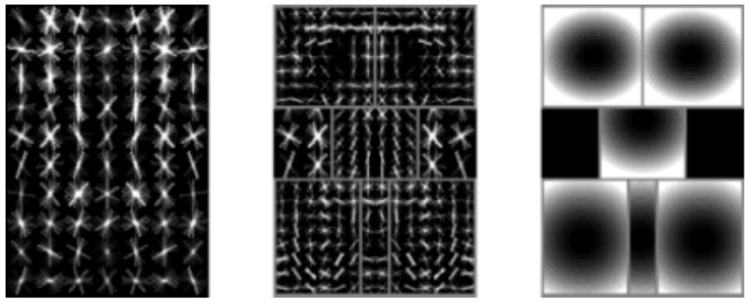


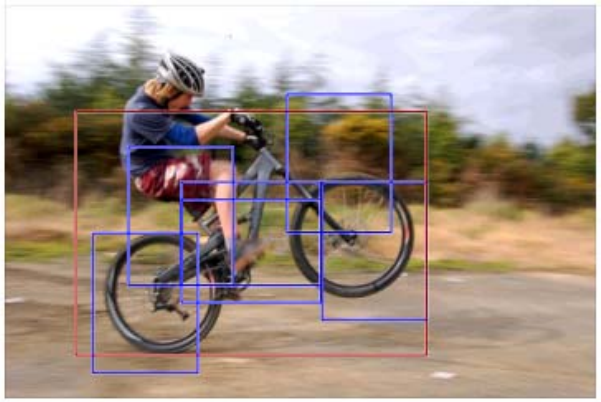


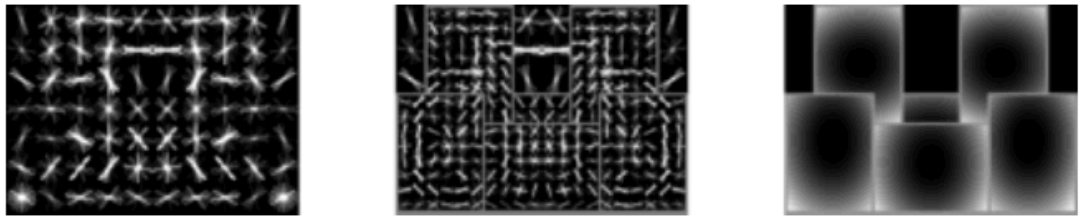


# Mixture Model Example - Bicycle

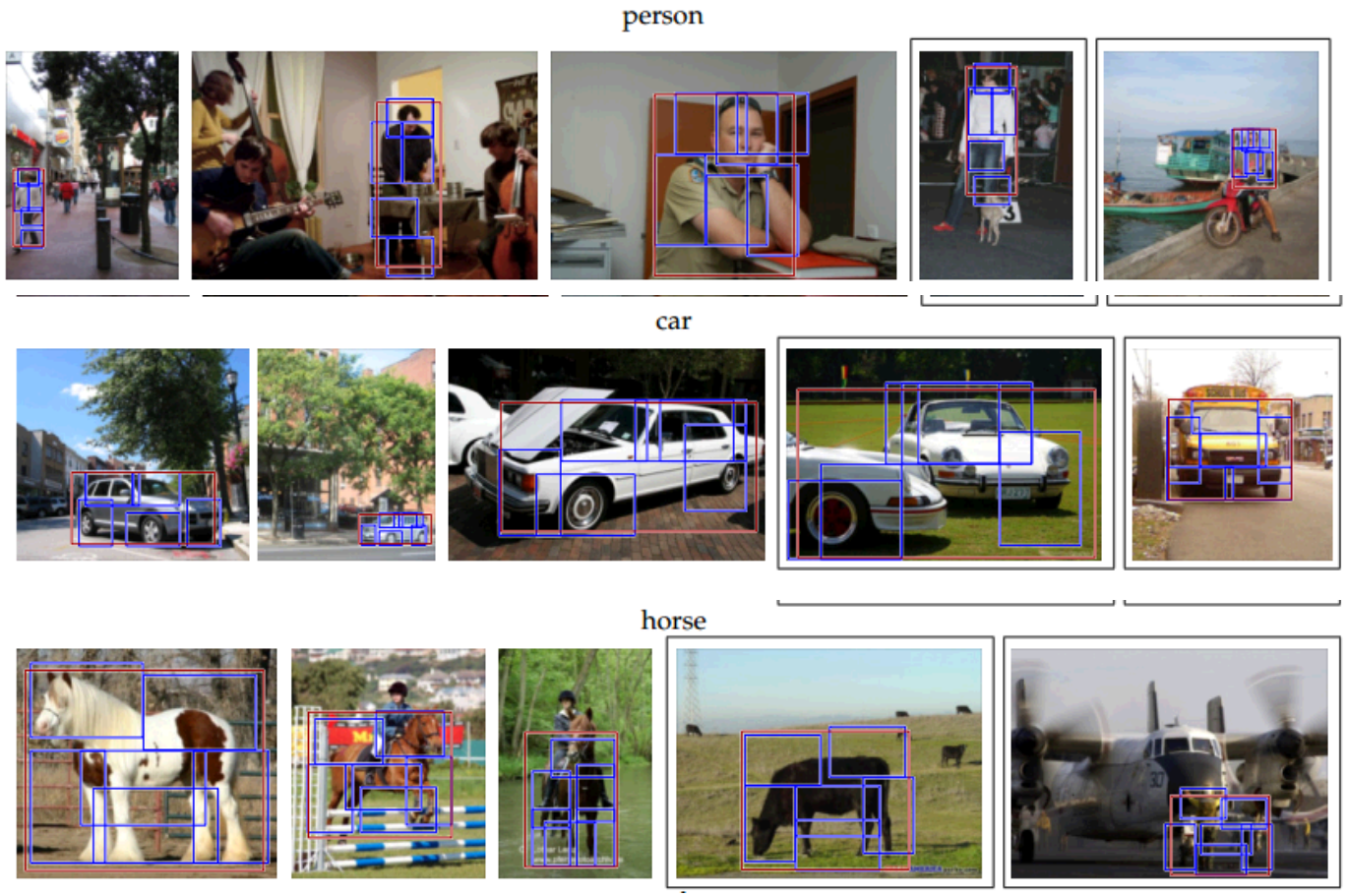








# Qualitative Results (PASCAL 2007)



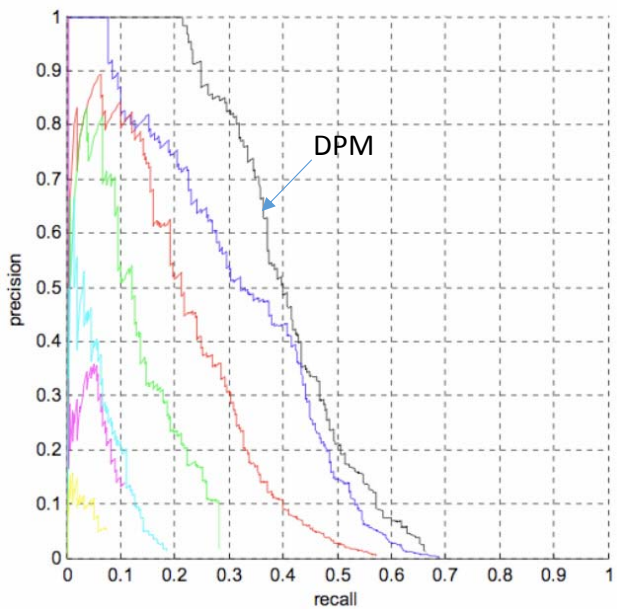
# Qualitative Results (PASCAL 2007)

# Quantitative Results (PASCAL 2008)

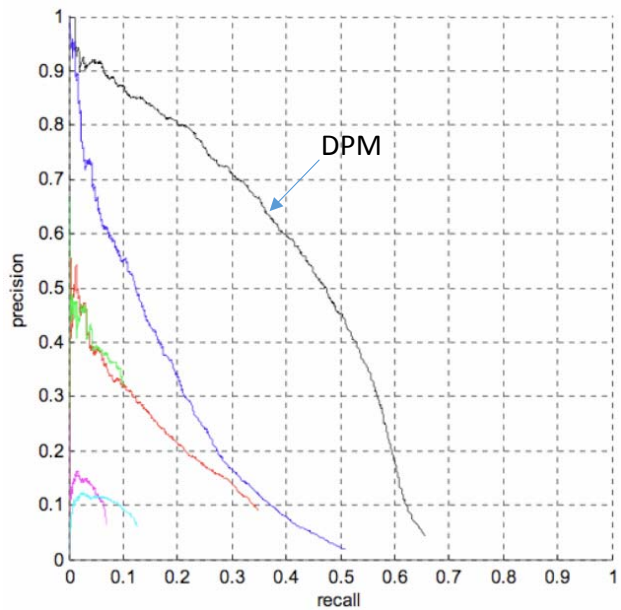
- 7 systems competed in the 2008 challenge
- Out of 20 classes, DPM got
  - First place in 7 classes
  - Second place in 8 classes

# Quantitative Results (PASCAL 2008)

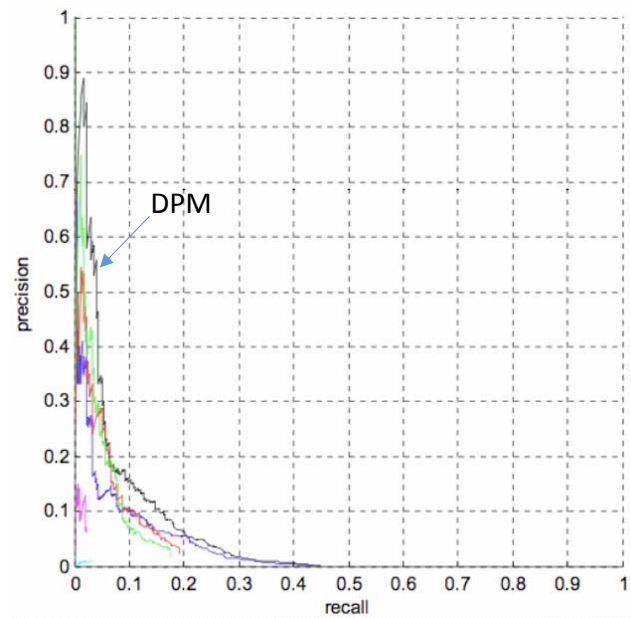
Bicycle

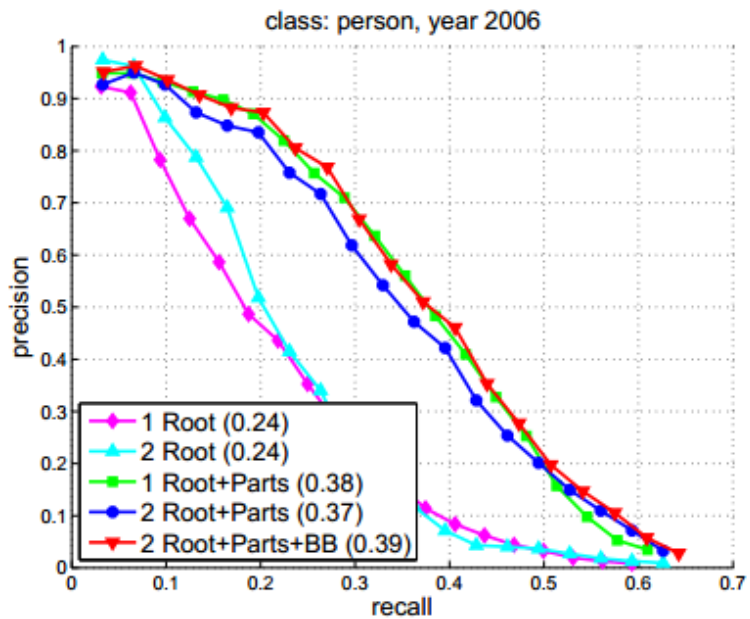
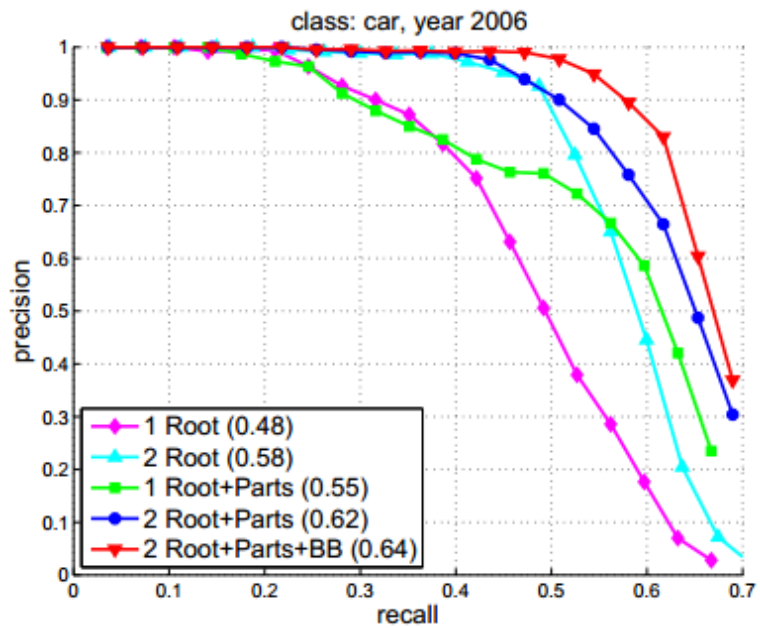



Person




Bird



# Quantitative Results (PASCAL 2006)

- Comparison of different configurations of proposed model