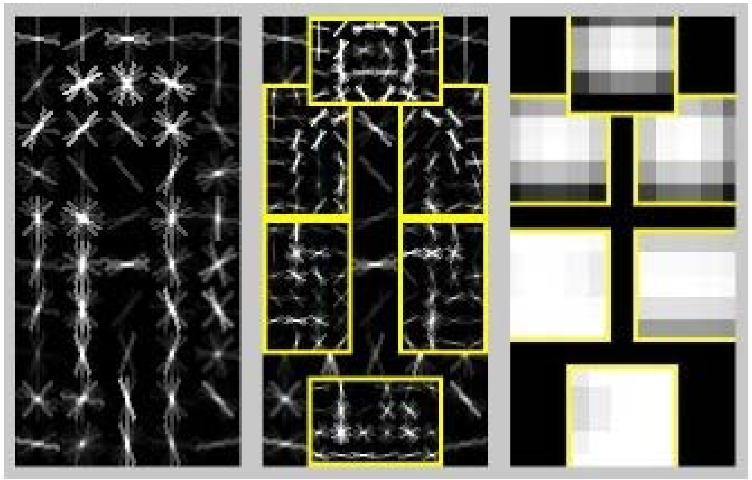
A Discriminatively Trained, Multiscale, Deformable Part Model

P. Felzenszwalb, D. McAllester, and D. Ramanan



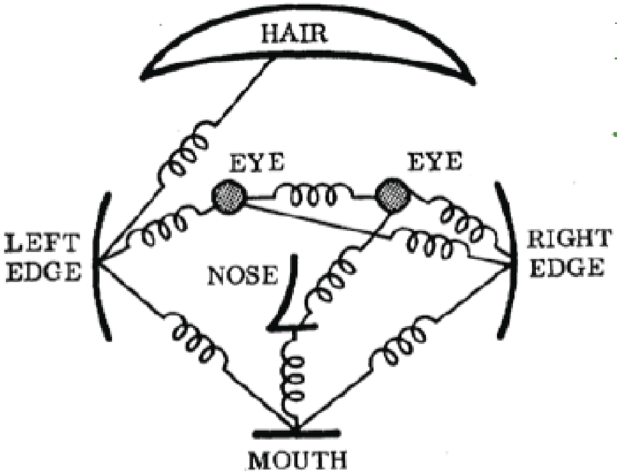
Edward Hsiao

16-721 Learning Based Methods in Vision

February 16, 2009

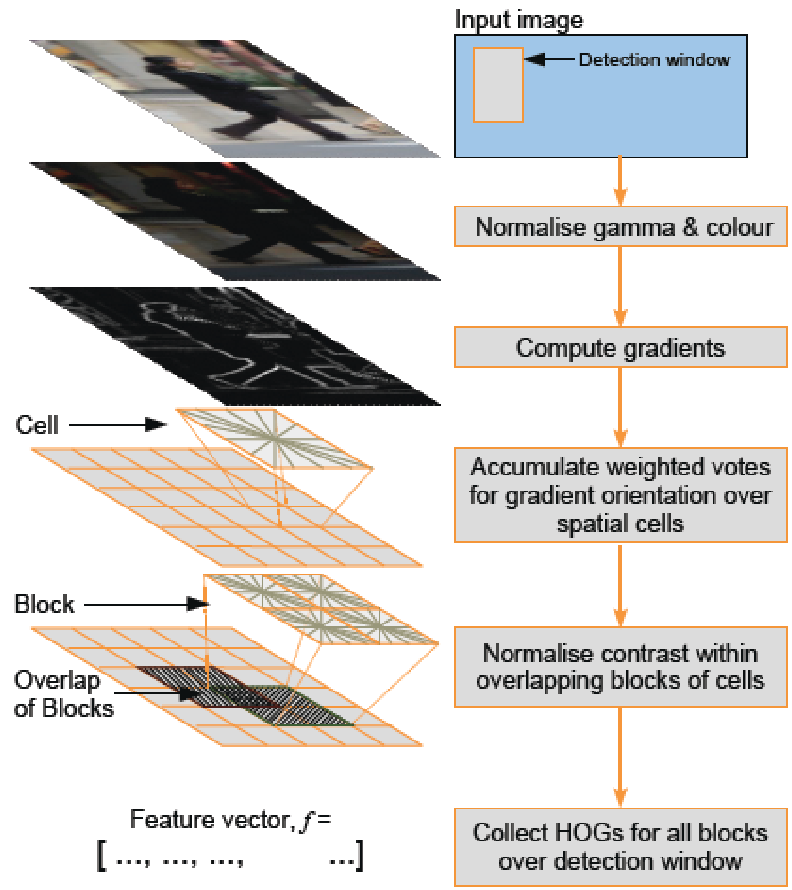
Images taken from P. Felzenszwalb, D. Ramanan, N. Dalal and B. Triggs

Overview

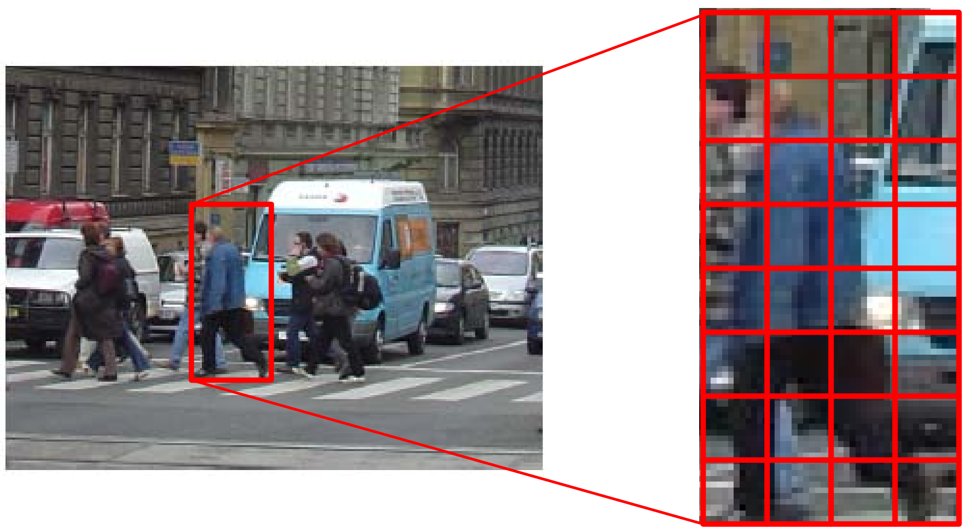


Histogram of Oriented Gradients (HOG)

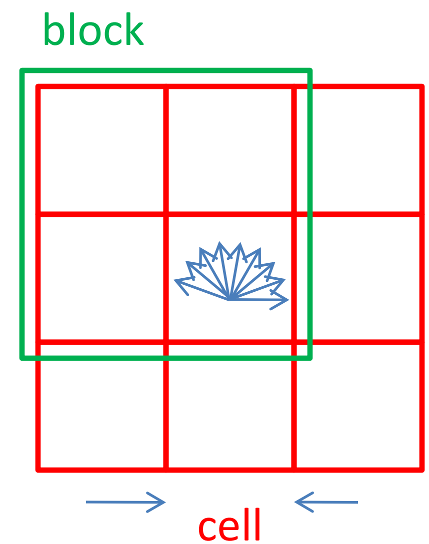
- Split detection window into 8x8 non-overlapping pixel regions called cells
- Compute 1D histogram of gradients in each cell and discretize into 9 orientation bins
- Normalize histogram of each cell with the total energy in the four 2x2 blocks that contain that cell -> 9x4 feature vector
- Apply a linear SVM classifier



Histogram of Oriented Gradients (HOG)



Feature vector
f = [...,...,...,   ,...]

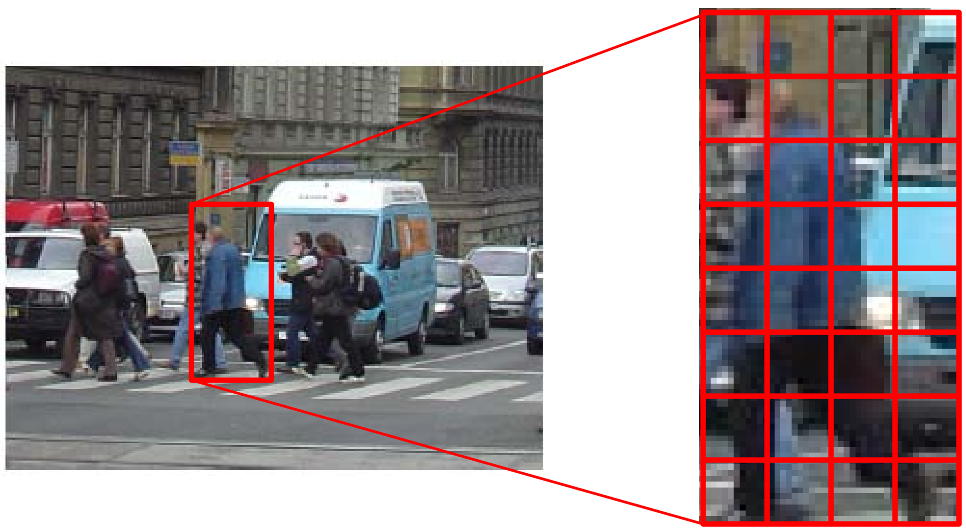


9 orientation bins
0 - 180° degrees

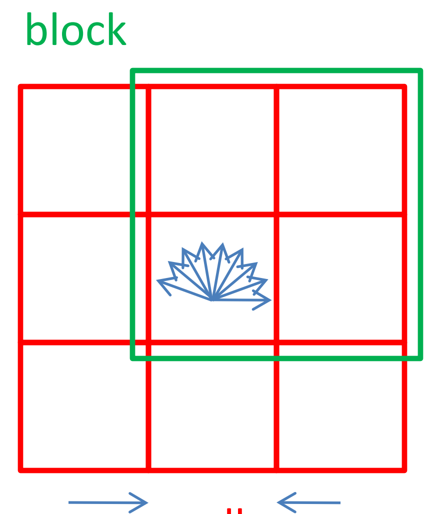
normalize

9x4 feature vector per cell

Histogram of Oriented Gradients (HOG)



Feature vector
f = [...,...,...,   ,...]

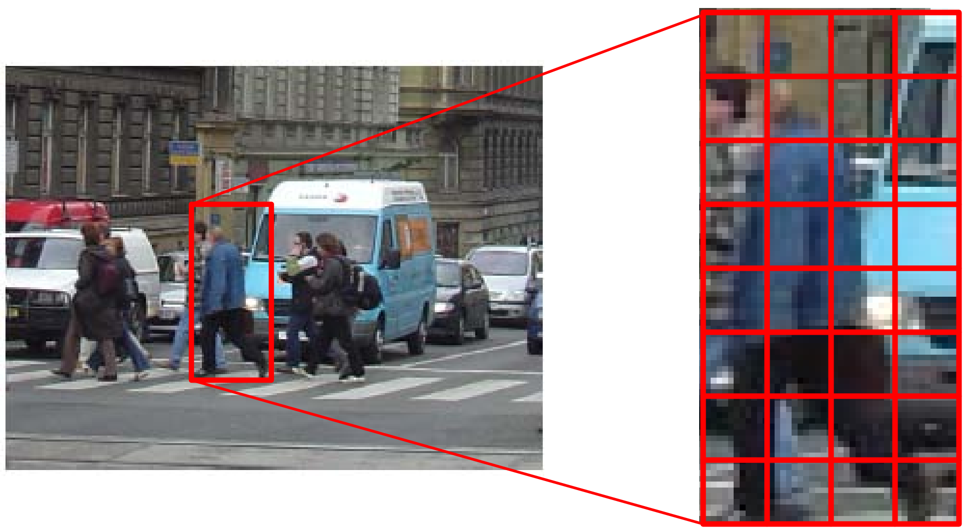


9 orientation bins
0 - 180° degrees

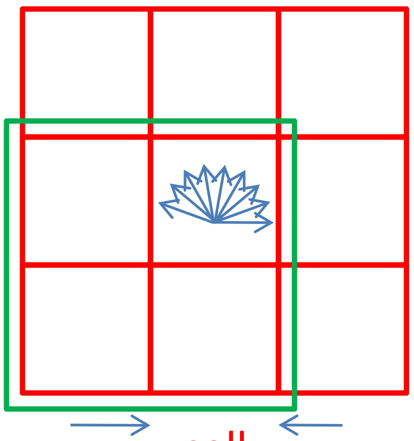
normalize

9x4 feature vector per cell

Histogram of Oriented Gradients (HOG)



Feature vector
f = [...,...,...,   ,...]

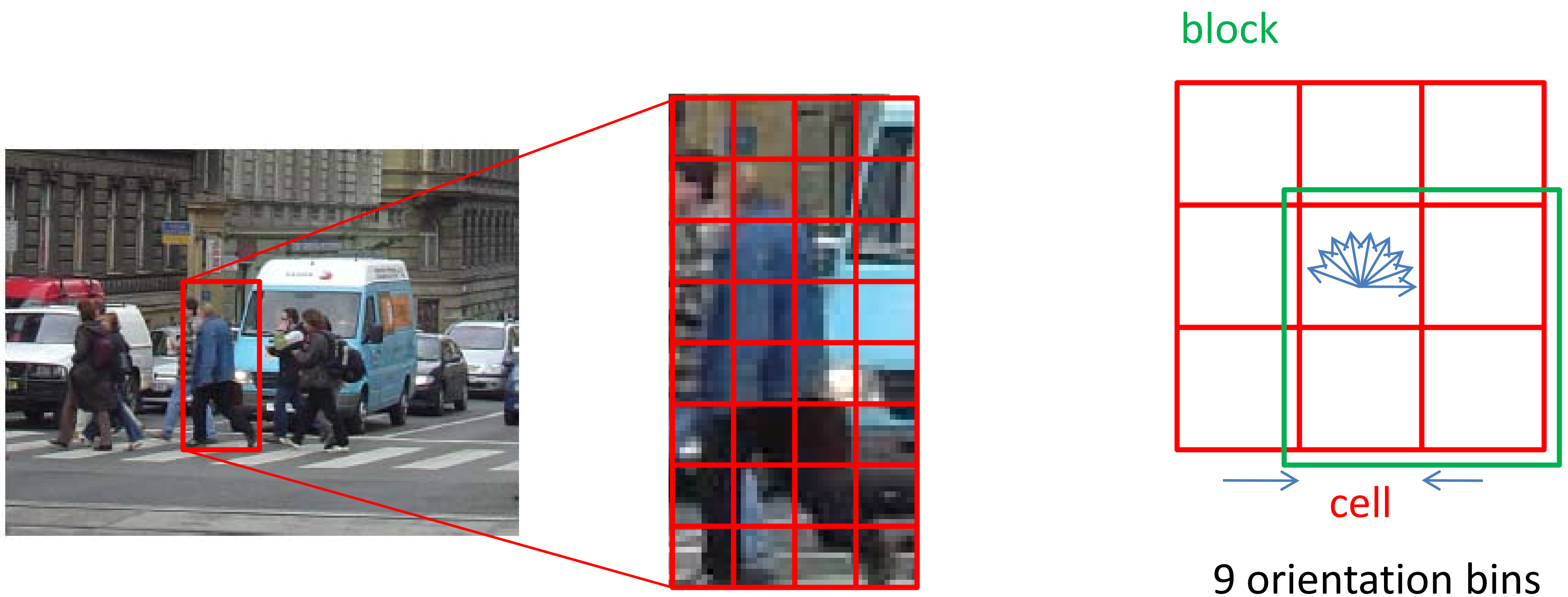


9 orientation bins
0 - 180° degrees

normalize

9x4 feature vector per cell

Histogram of Oriented Gradients (HOG)



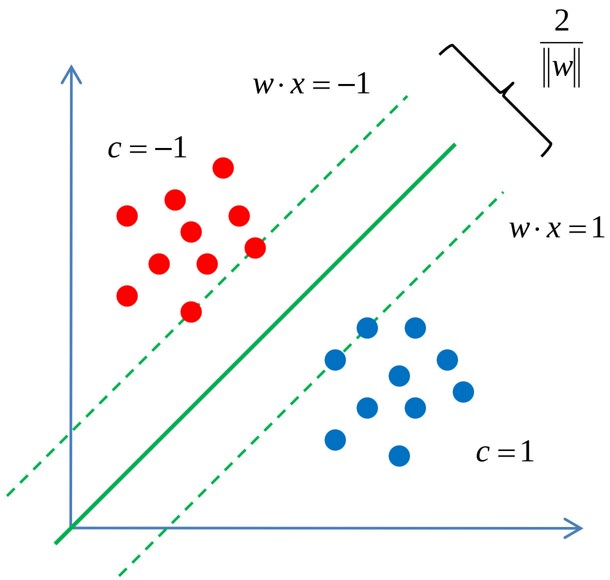
9 orientation bins
0 - 180° degrees

Feature vector
f = [...,...,...,   ,...]

normalize

9x4 feature vector per cell

SVM Review

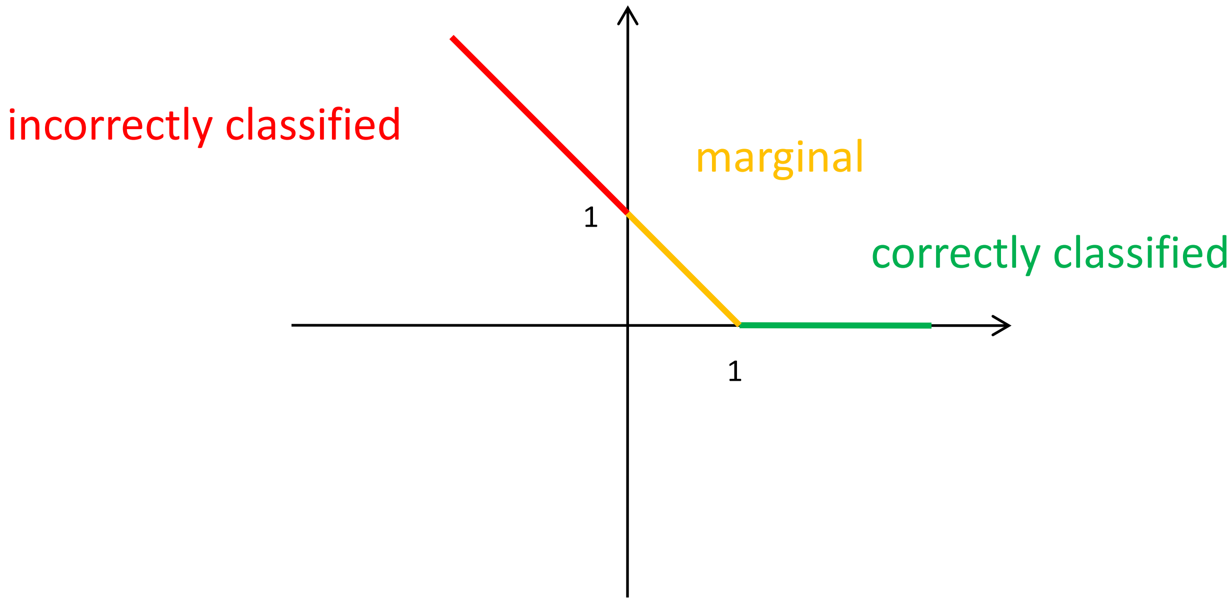


$$c_i(w \cdot x_i) \geq 1$$

$$\text{minimize } \frac{1}{2}\|w\|^2 \text{ subject to } c_i(w \cdot x_i) \geq 1$$

Hinge Loss

$$\max(0, 1 - c_i(w \cdot x_i))$$

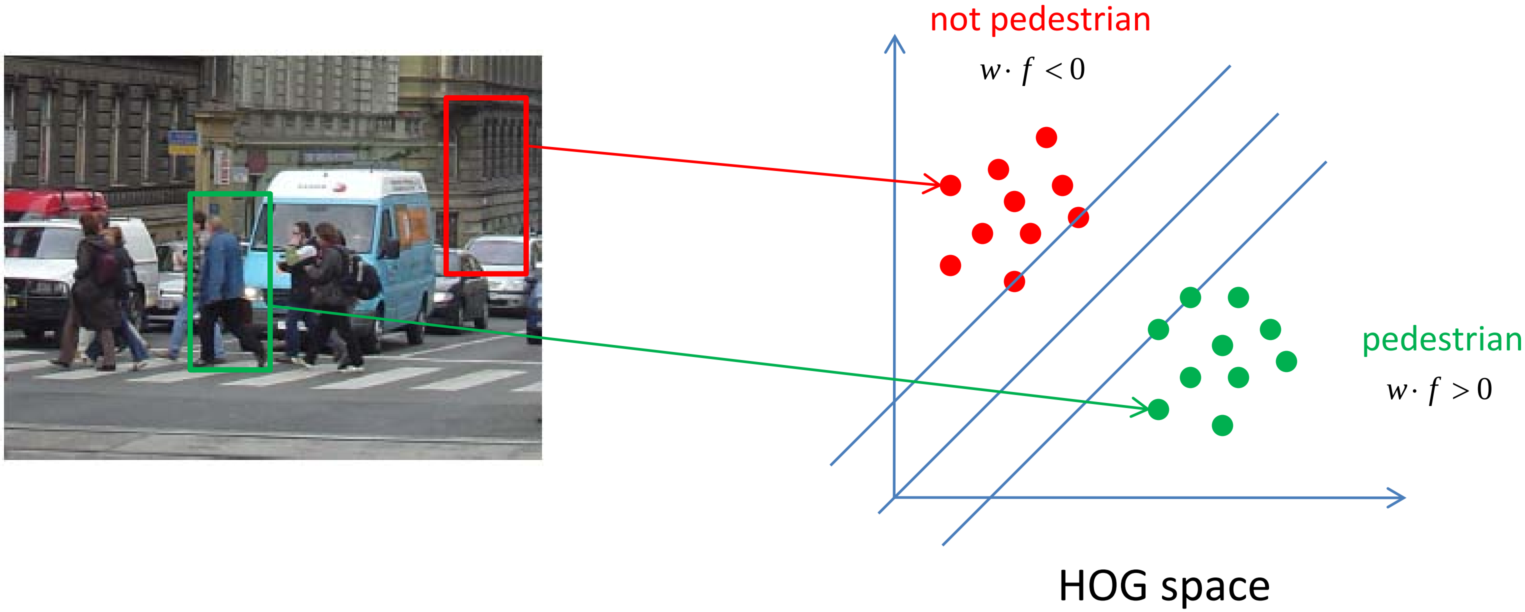


$$1 - c_i(w \cdot x_i) > 0$$

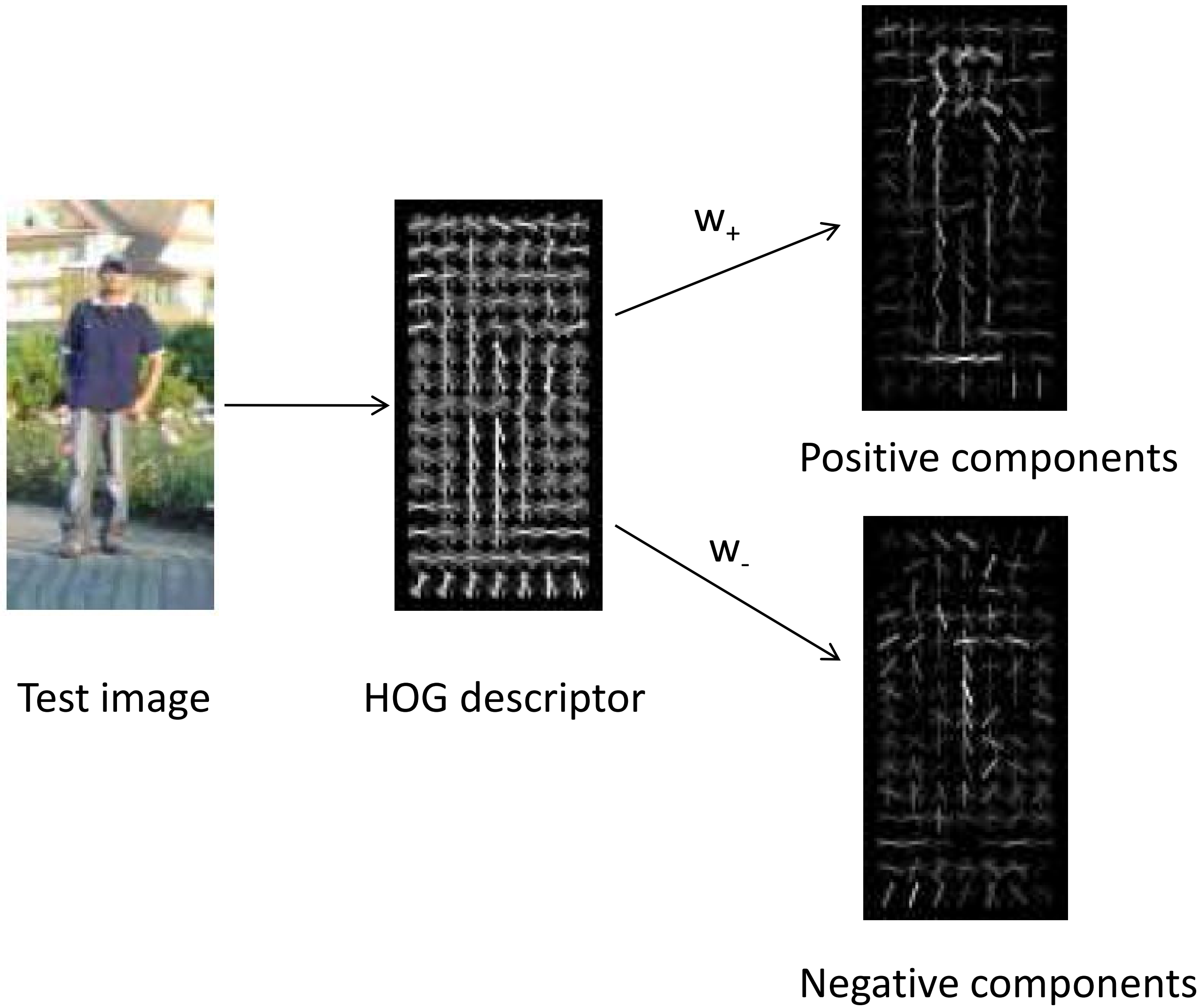
if incorrectly classified
or inside margin

$$\arg\min_{w} \lambda \|w\|^2 + \sum_{i=1}^{n} \max(0, 1 - c_i(w \cdot x_i))$$

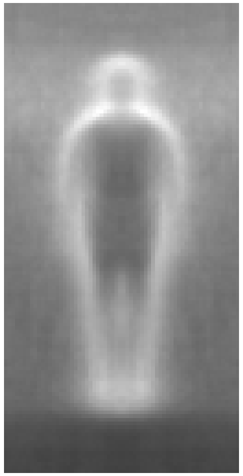
HOG & Linear SVM



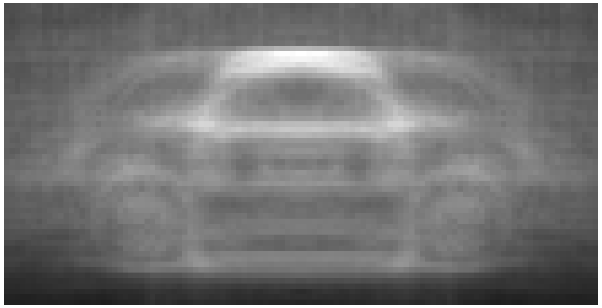
Histogram of Oriented Gradients (HOG)



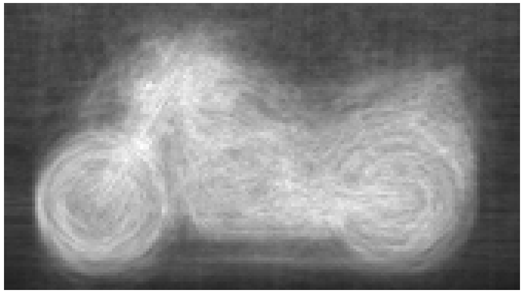
Average Gradients



person



car

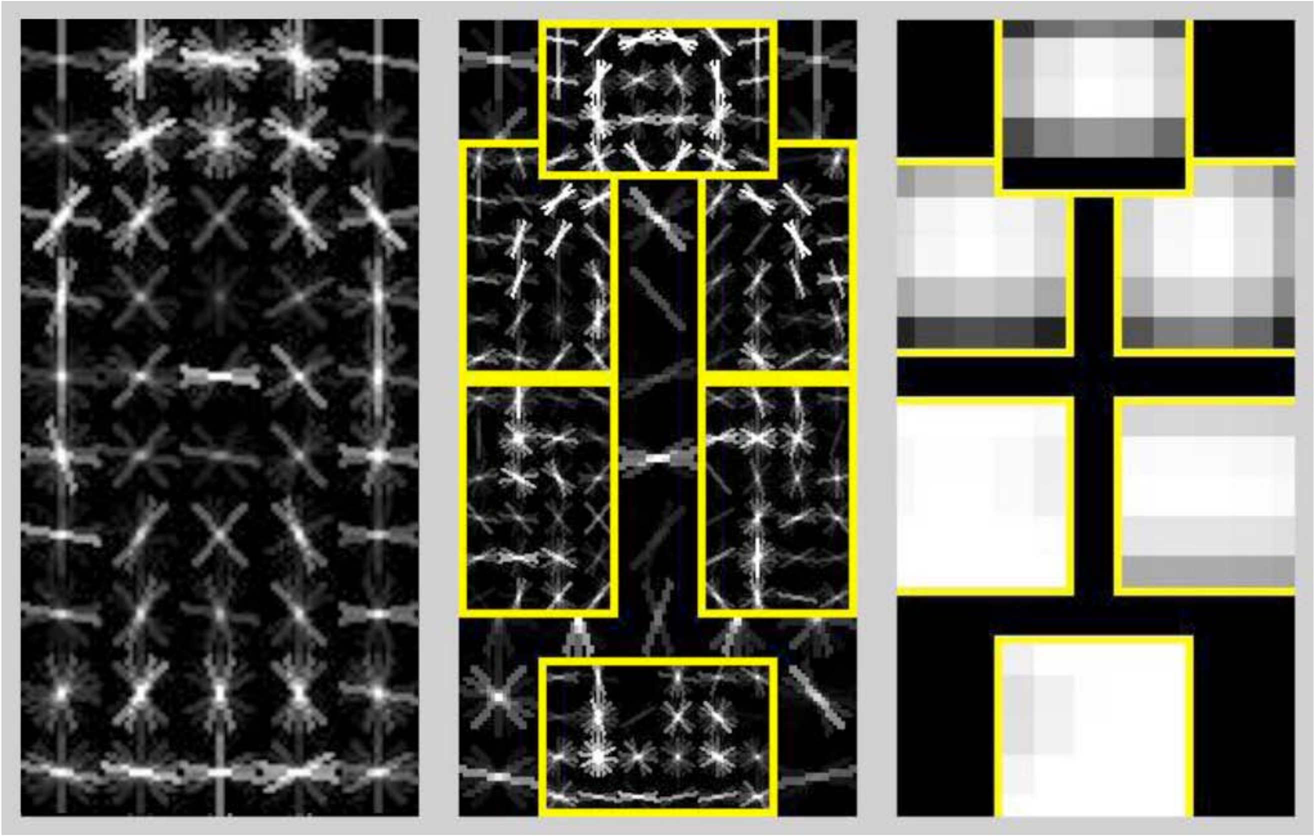


motorbike

Deformable Part Models

Root filter
8x8 resolution

Deformable Part Models



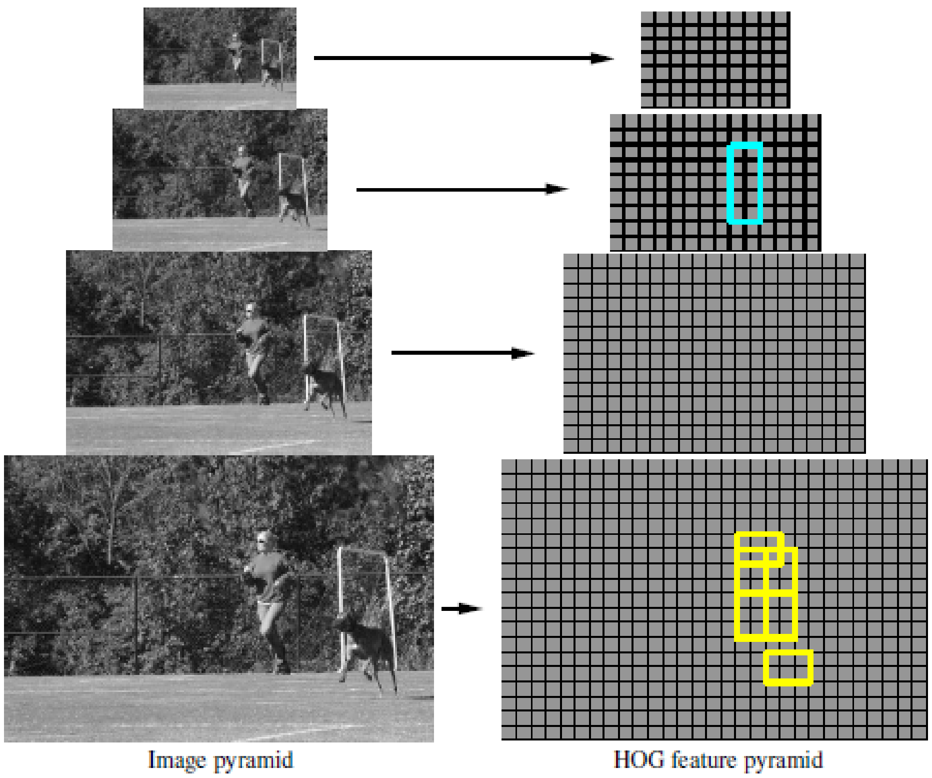
Root filter
8x8 resolution

Part filter
4x4 resolution

Quadratic
spatial model

$$a_{x,i}x_i + a_{y,i}y_i + b_{x,i}x_i^2 + b_{y,i}y_i^2$$

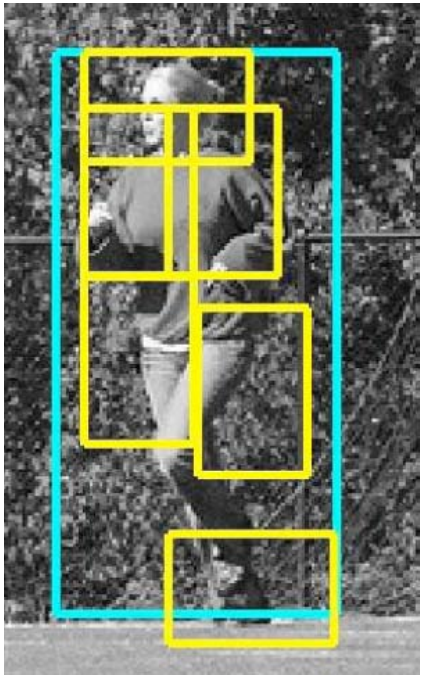
HOG Pyramid



$\phi(H, p)$ concatenation of HOG features in a subwindow of the HOG pyramid H at position p = (x,y,l)

Deformable Part Models

Root filter F_0

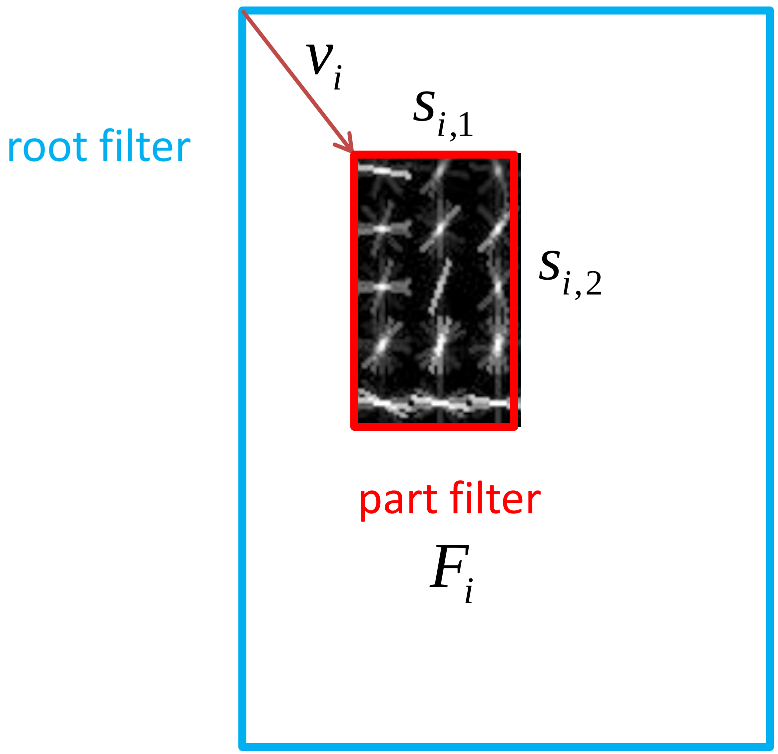


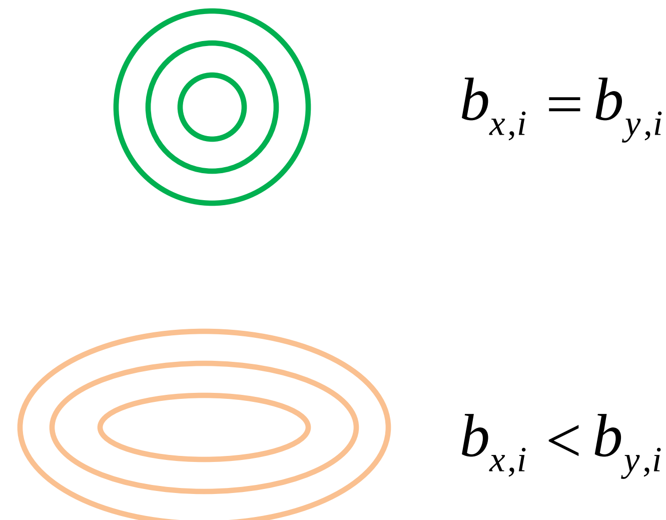
Part filters $P_1 \ldots P_n$

$$P_i = (F_i, v_i, s_i, a_i, b_i)$$

$$\text{score} = \underbrace{\sum_{i=0}^{n} F_i \cdot \phi(H, p_i)}_{\text{filter response}} + \underbrace{\sum_{i=1}^{n} a_i \cdot (\tilde{x}_i, \tilde{y}_i) + b_i \cdot (\tilde{x}_i^2, \tilde{y}_i^2)}_{\text{part placement}}$$

Part Models





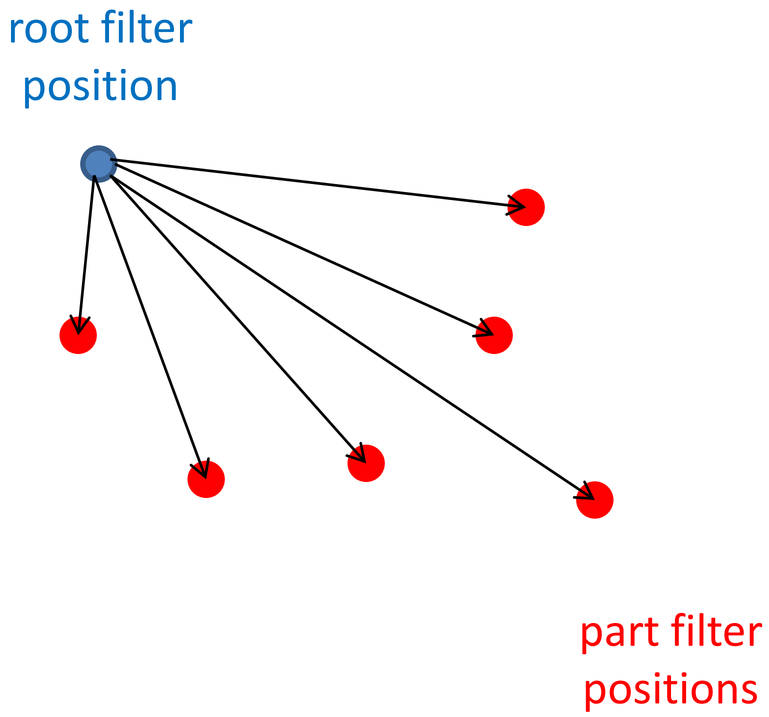
$$P_i = (F_i, v_i, s_i, a_i, b_i)$$

Quadratic spatial model

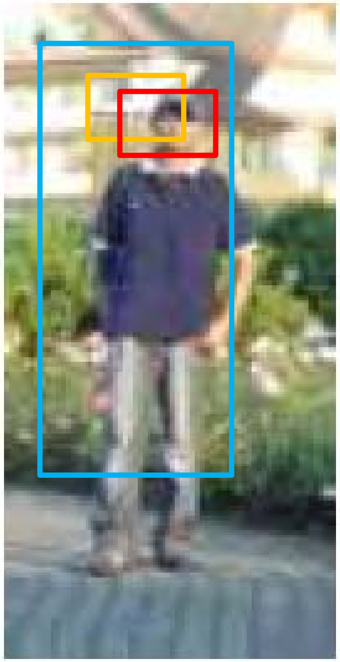
$$a_{x,i}x_i + a_{y,i}y_i + b_{x,i}x_i^2 + b_{y,i}y_i^2$$

$$b_i \geq 0$$

Star Graph / 1-fan



Distance Transforms



part anchor location

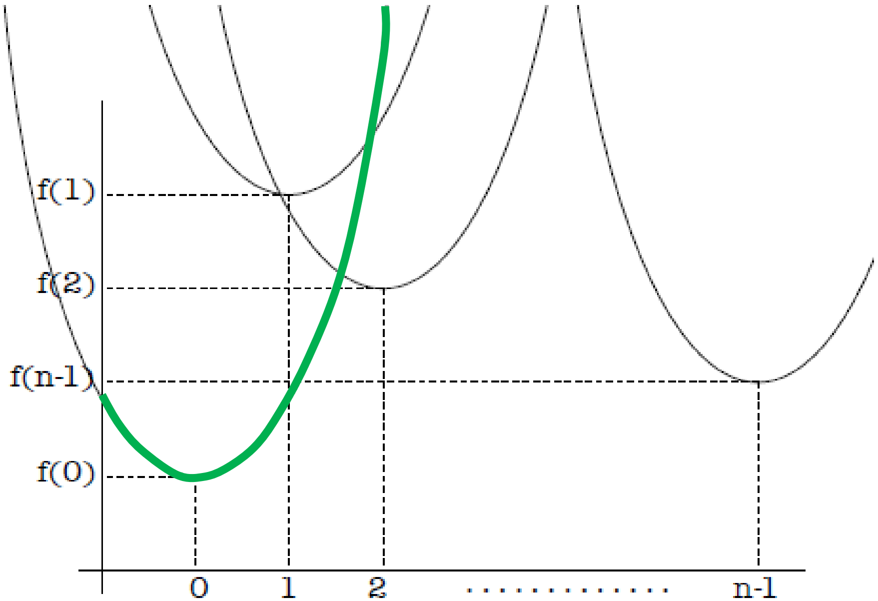
part position

$$\mathcal{D}_f(p) = \min_{q \in \mathcal{G}} (d(p,q) + f(q))$$

quadratic distance specified by a_i and b_i

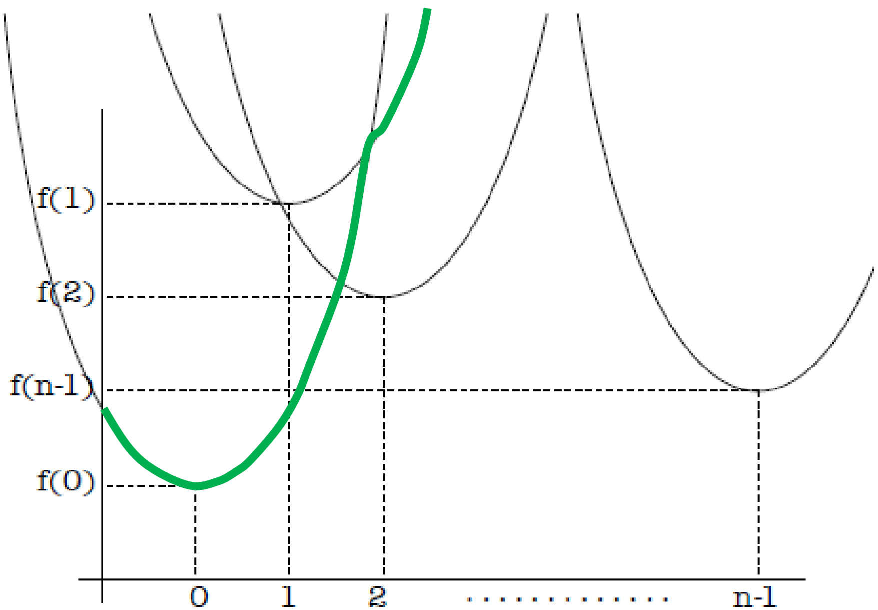
filter response

Quadratic 1-D Distance Transform



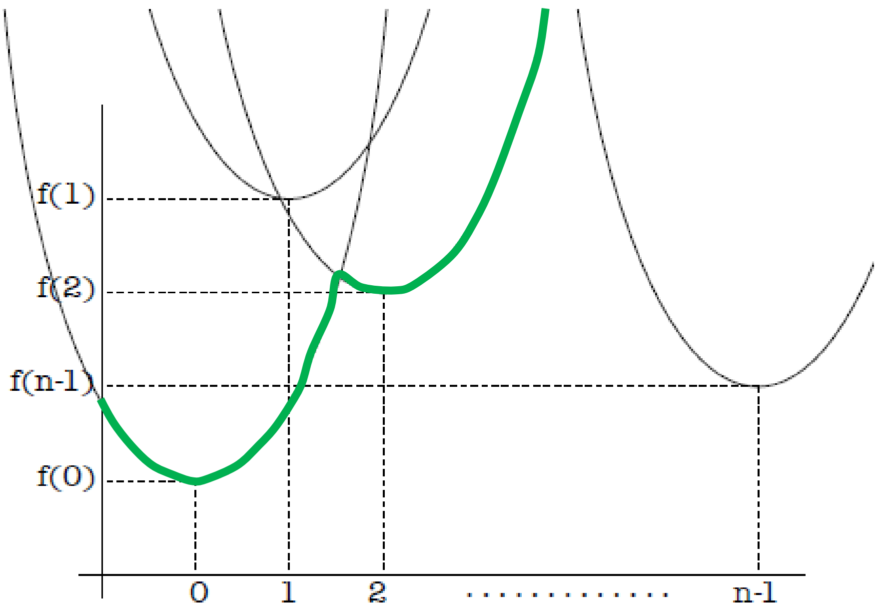
$$\mathcal{D}_f(p) = \min_{q \in \mathcal{G}} ((p-q)^2 + f(q))$$

Quadratic 1-D Distance Transform



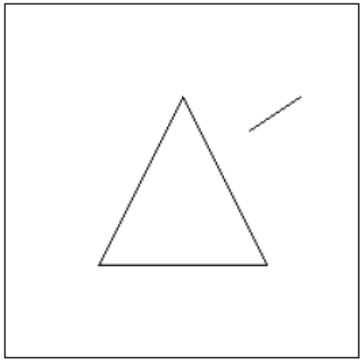
$$\mathcal{D}_f(p) = \min_{q \in \mathcal{G}} ((p-q)^2 + f(q))$$

Quadratic 1-D Distance Transform

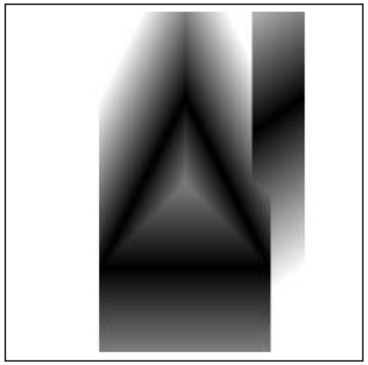


$$\mathcal{D}_f(p) = \min_{q \in \mathcal{G}}((p-q)^2 + f(q))$$

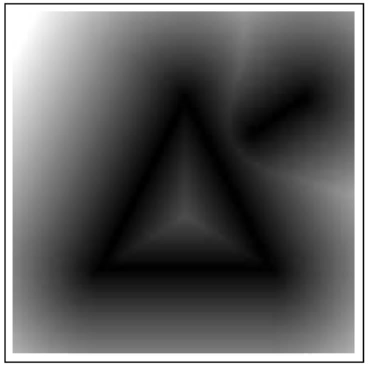
Distance Transforms in 2-D



input

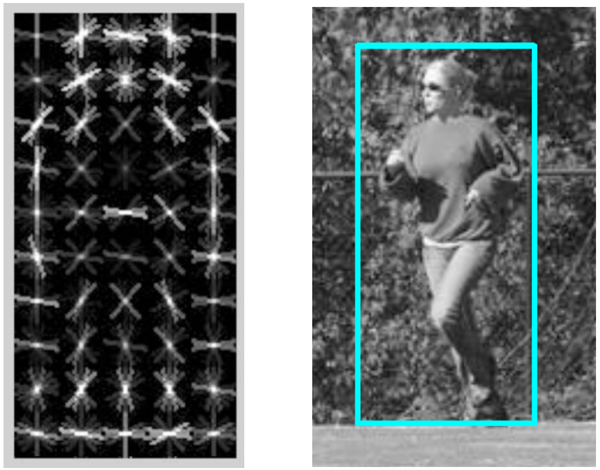


column
distance transform



full
distance transform

Latent SVM



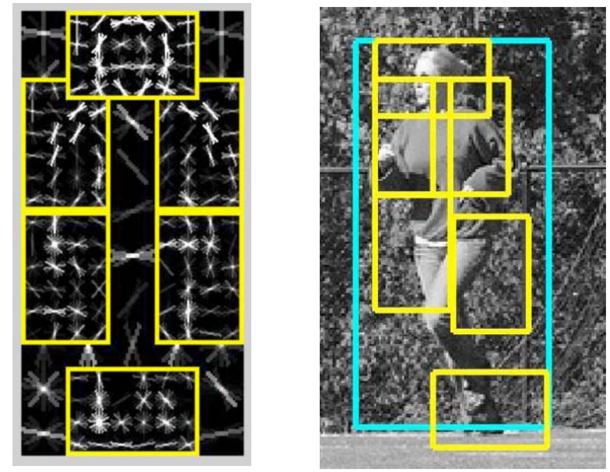
HOG & Linear SVM

$$f_w(x) = w \cdot \Phi(x)$$

$$w = F_0$$

$$\Phi(x) = \phi(H(x), p_0)$$

$$w^* = \arg\min_w \lambda \|w\|^2 + \sum_{i=1}^{n} \max(0, 1 - y_i f_w(x_i))$$



Deformable Parts & Latent SVM

$$f_w(x) = \max_{z \in Z(x)} w \cdot \Phi(x, z)$$

$$w = (F_0, \ldots, F_n, a_1, b_1, \ldots, a_n, b_n)$$

$$\Phi(x,z) = (\phi(H(x), p_0), \phi(H(x), p_1), \ldots, \phi(H(x), p_n), \tilde{x}_1, \tilde{y}_1, \tilde{x}_1^2, \tilde{y}_1^2, \ldots, \tilde{x}_n, \tilde{y}_n, \tilde{x}_n^2, \tilde{y}_n^2)$$

$$w^* = \arg\min_w \lambda \|w\|^2 + \sum_{i=1}^{n} \max(0, 1 - y_i f_w(x_i))$$

Semi-convexity

$$f_w(x) = \max_{z \in Z(x)} w \cdot \Phi(x, z)$$ convex in w

$$w^* = \arg\min_w \lambda \|w\|^2 + \sum_{i \in pos} \max(0, 1 - f_w(x_i)) + \sum_{i \in neg} \max(0, 1 + f_w(x_i))$$

- If $f_w(x)$ is linear in w, this is a standard SVM (convex)
- If $f_w(x)$ is arbitrary, this is in general not convex
- If $f_w(x)$ is convex in w, the hinge loss is convex for negative examples (semi-convex)
  - hinge loss is convex in w if positive examples are restricted to single choice of Z(x)

$$\hat{w} = \arg\min_w \lambda \|w\|^2 + \sum_{i \in pos} \max(0, 1 - w \cdot \Phi(x_i, z_i)) + \sum_{i \in neg} \max(0, 1 + f_w(x_i))$$ convex

Optimization is now convex!

Coordinate Descent

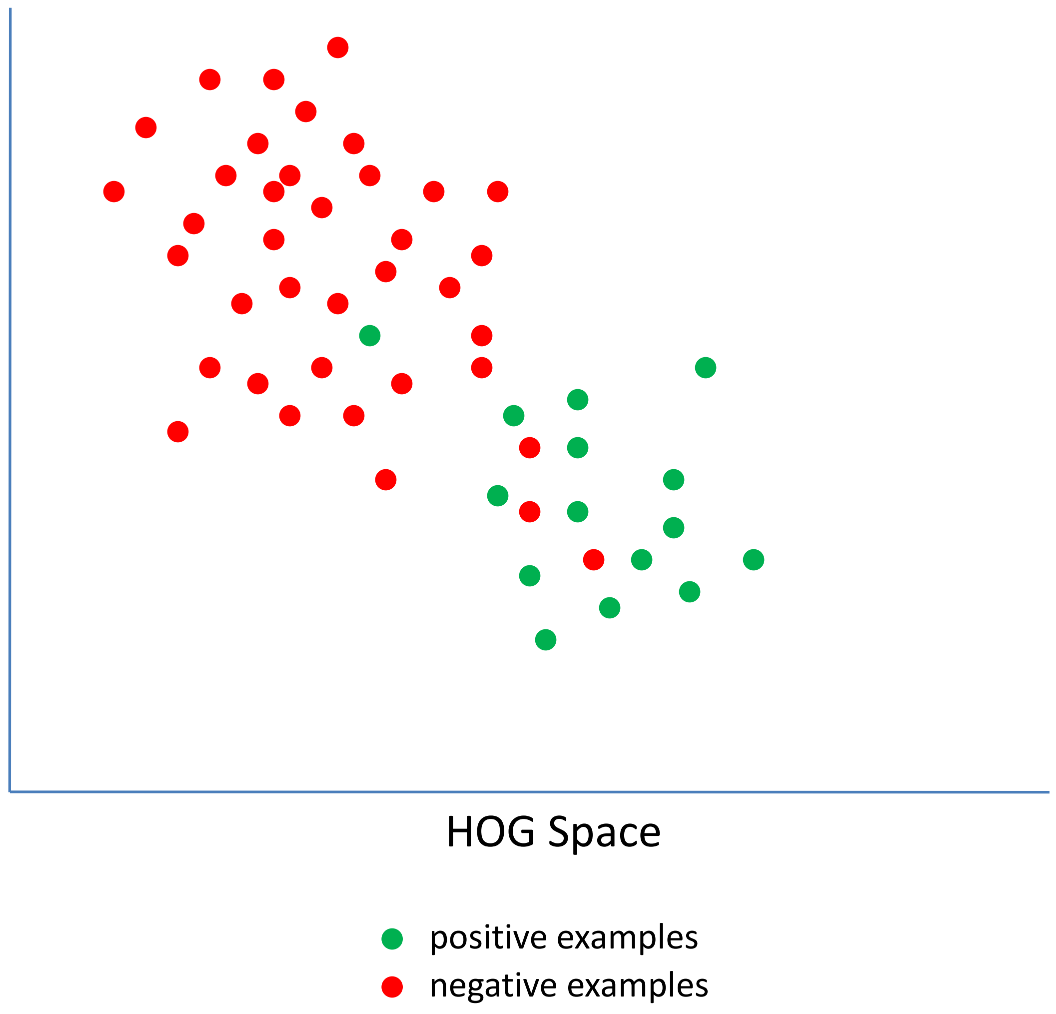
1. Hold w fixed, and optimize the latent values for the positive examples

$$z_i = \arg\max_{z \in Z(x_i)} w \cdot \Phi(x, z)$$

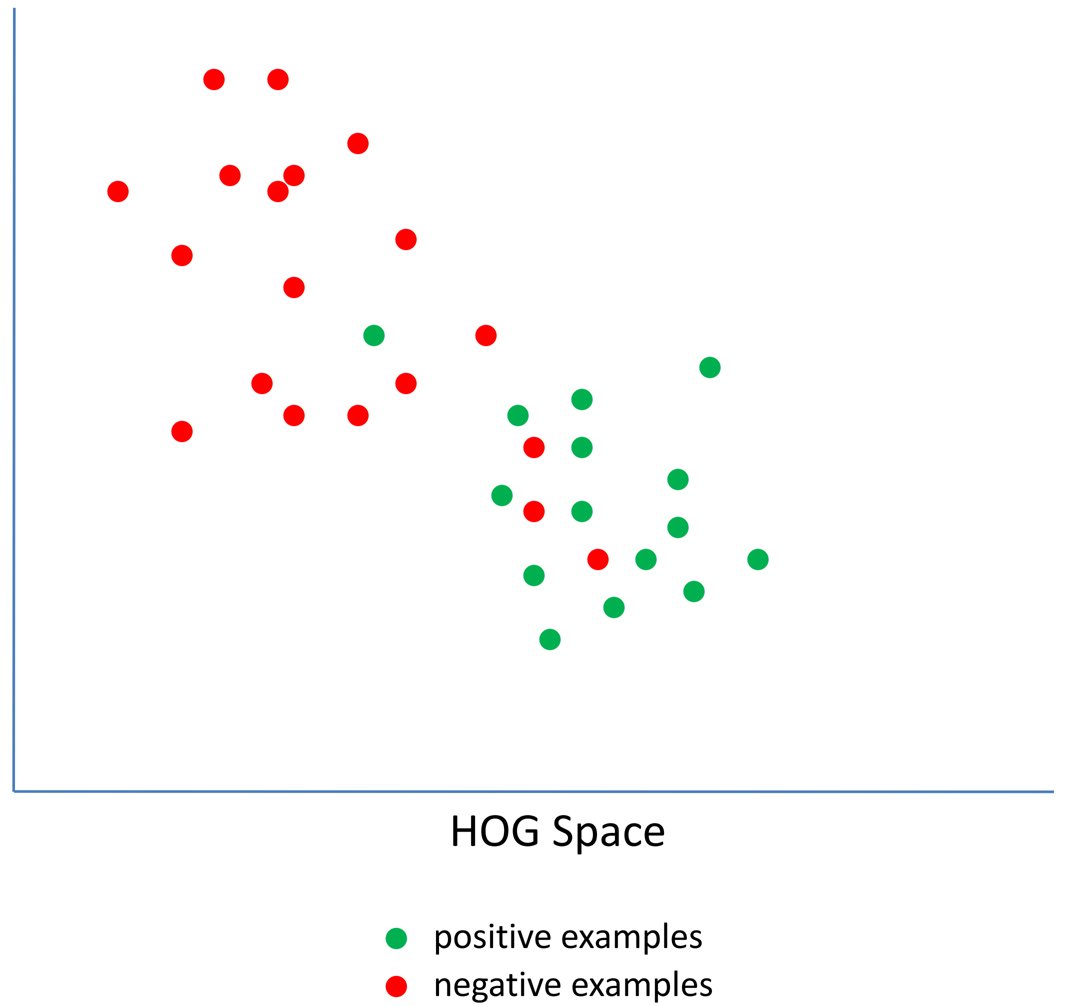
2. Hold $\{z_i\}$ fixed for positive examples, optimize w by solving the convex problem

$$\hat{w} = \arg\min_{w} \lambda \|w\|^2 + \sum_{i \in pos} \max(0, 1 - w \cdot \Phi(x_i, z_i)) + \sum_{i \in neg} \max(0, 1 + f_w(x_i))$$

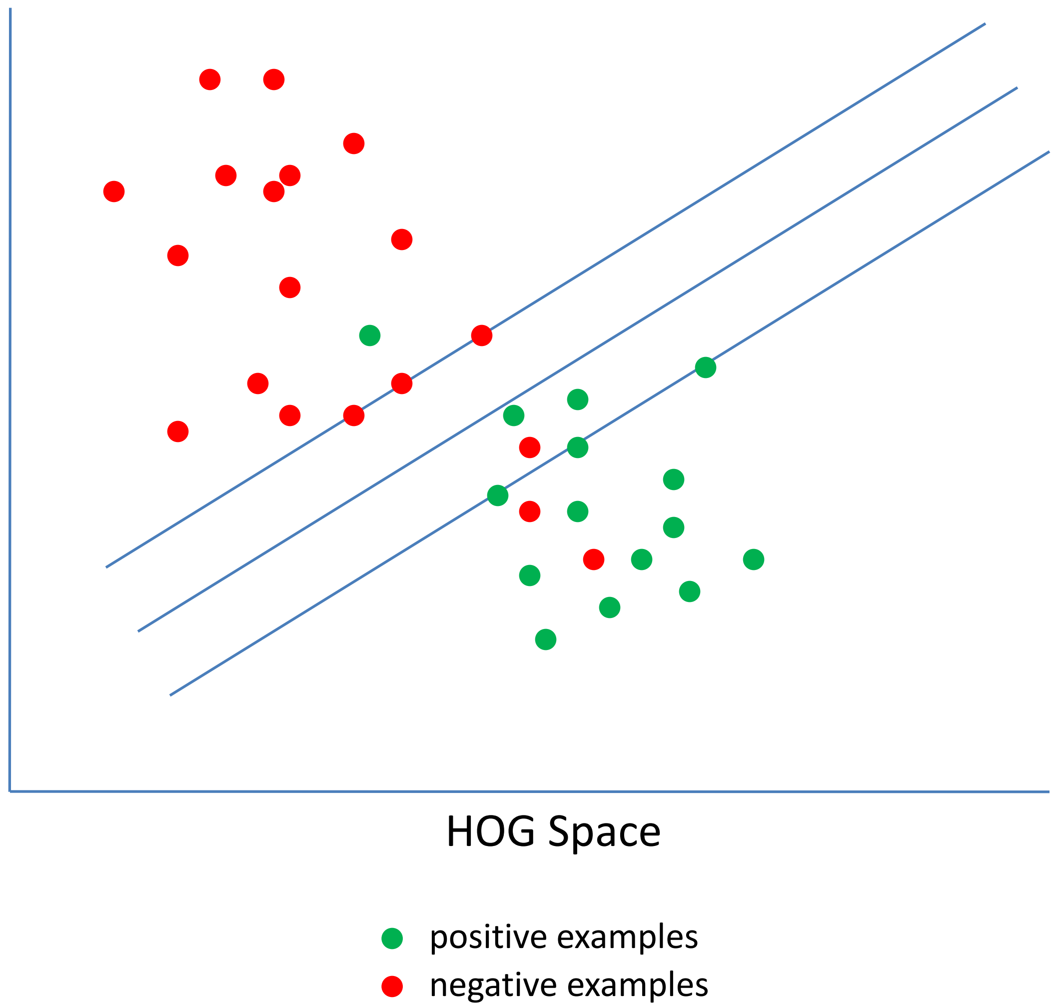
Data Mining Hard Negatives



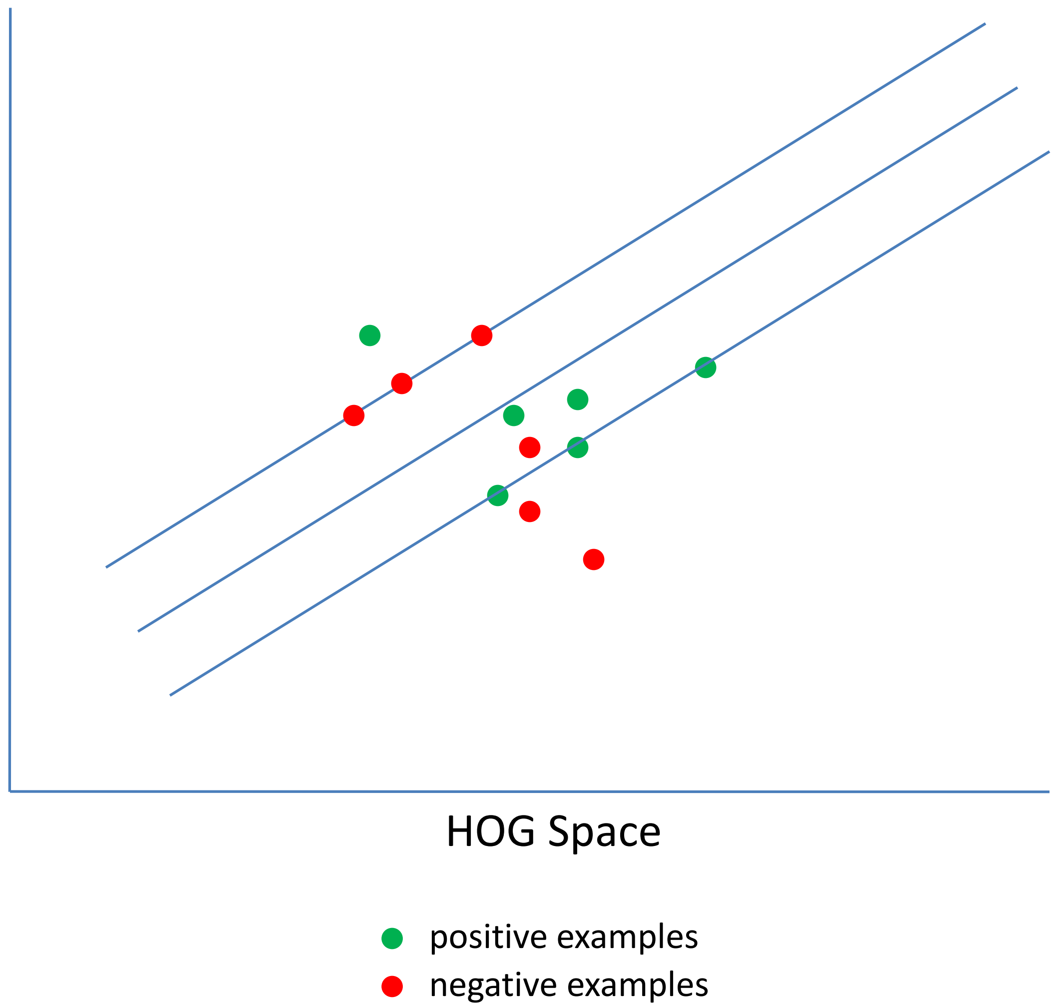
Data Mining Hard Negatives



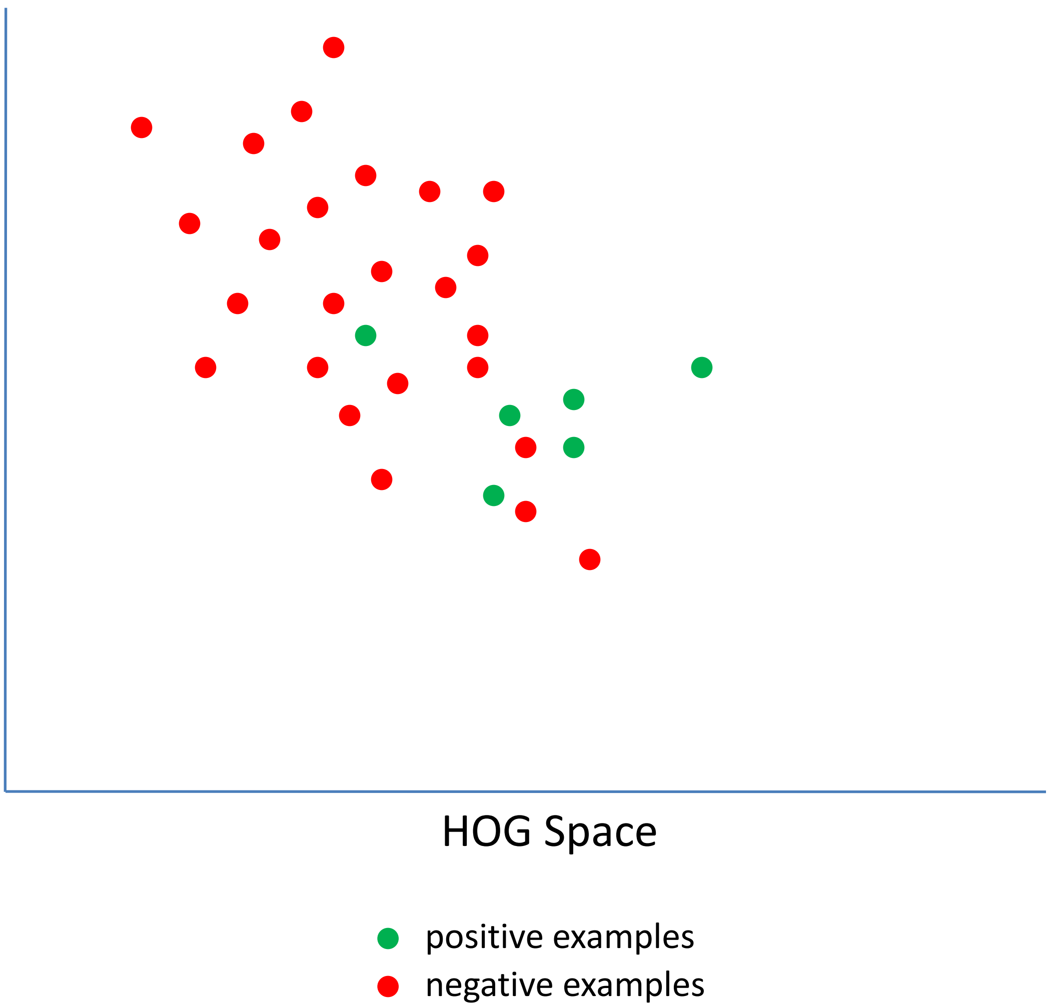
Data Mining Hard Negatives



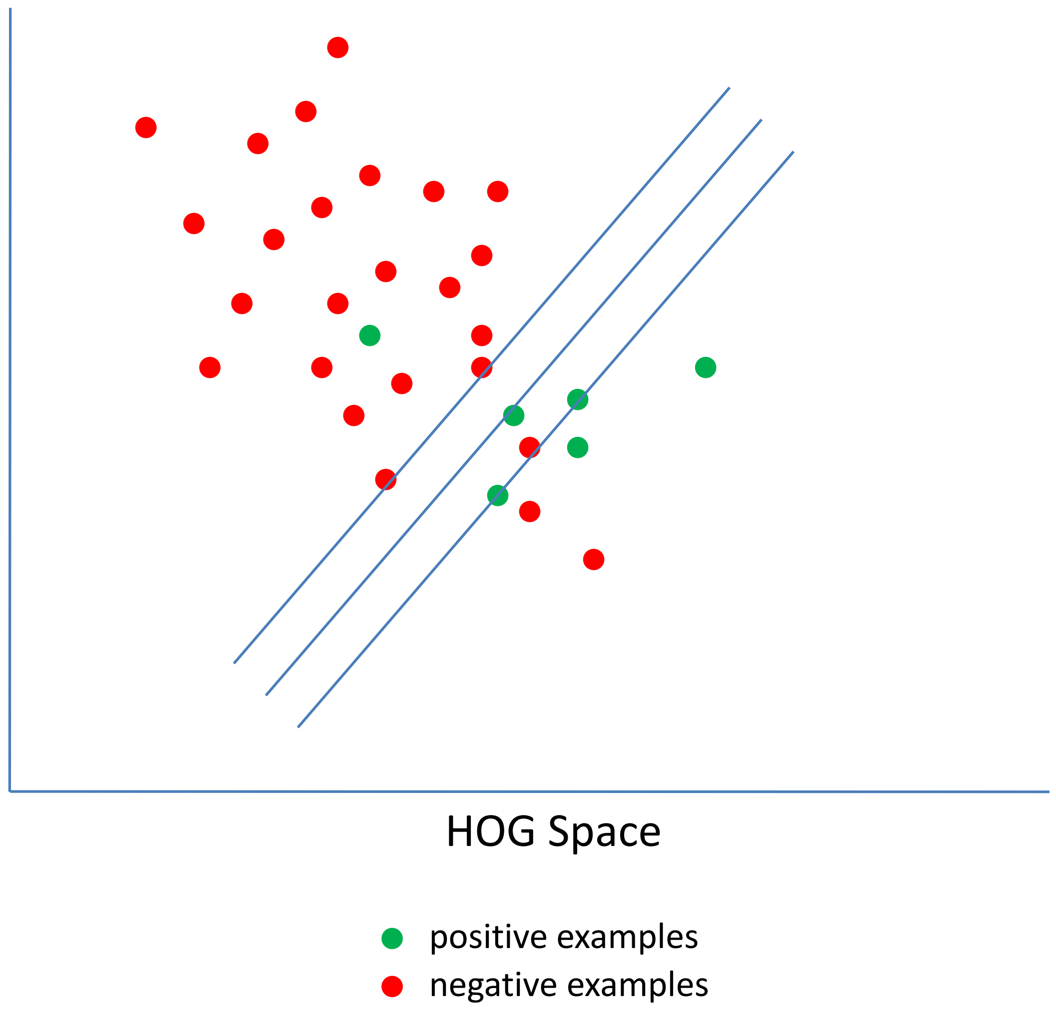
Data Mining Hard Negatives



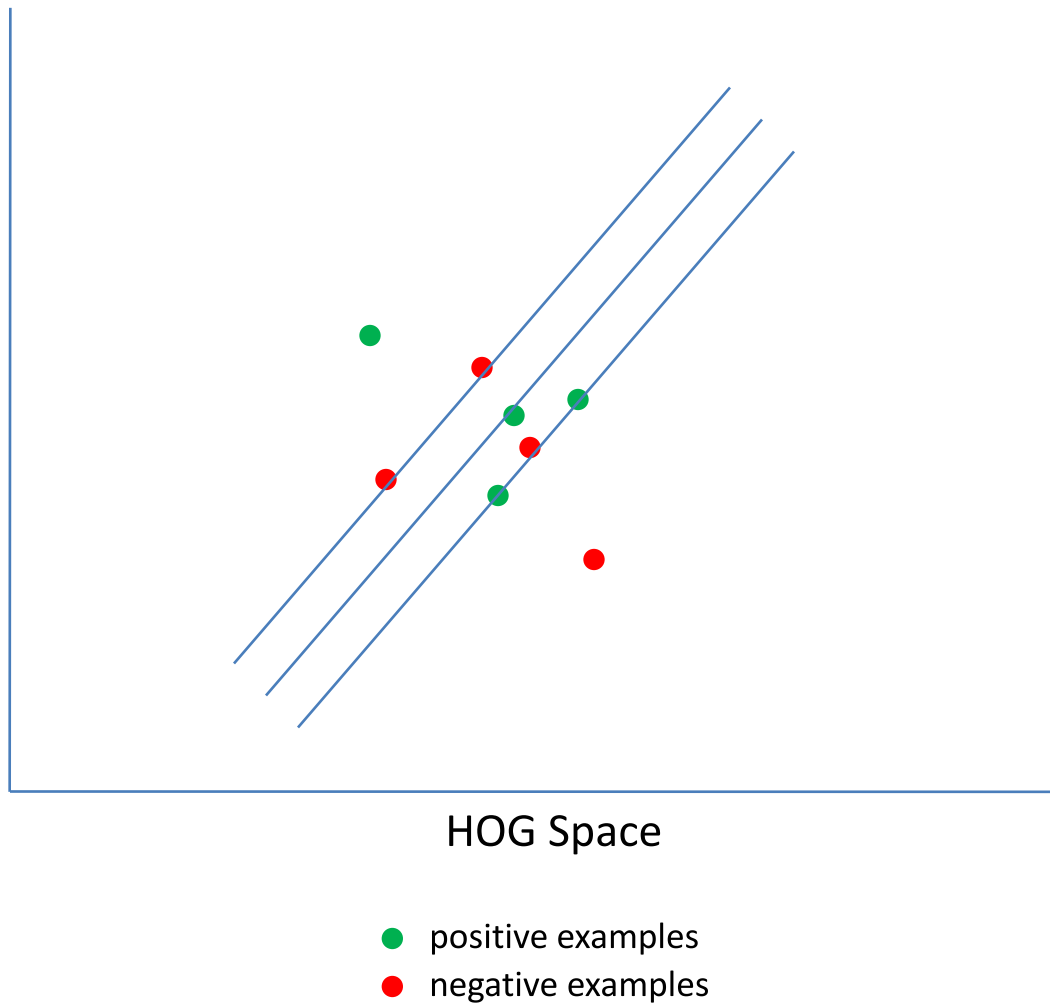
Data Mining Hard Negatives



Data Mining Hard Negatives

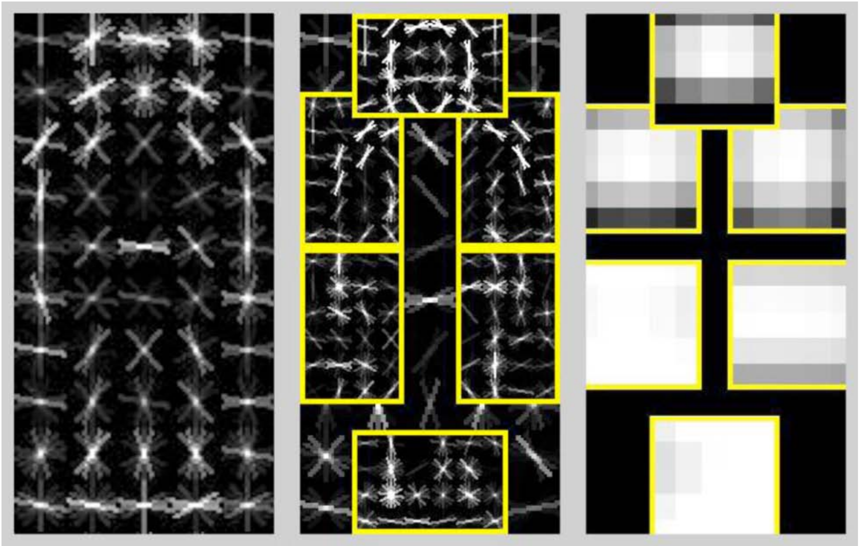


Data Mining Hard Negatives



Model Learning Algorithm

- Initialize root filter
- Update root filter
- Initialize parts
- Update model



Root Filter Initialization

- Select aspect ratio and size by using a heuristic

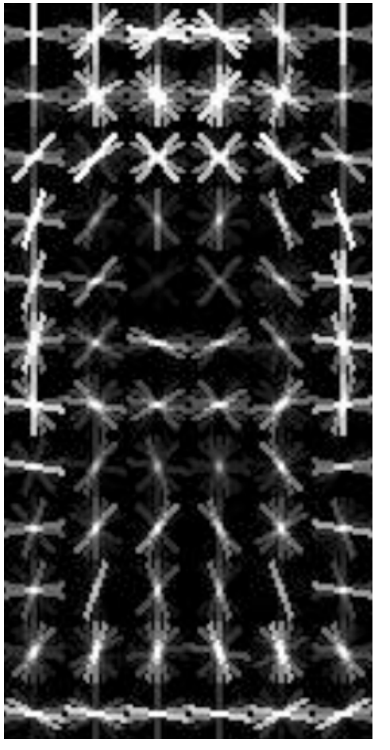
  - model aspect is the mode of data

  - model size is largest size > 80% of the data

- Train initial root filter F_0 using an SVM with no latent variables

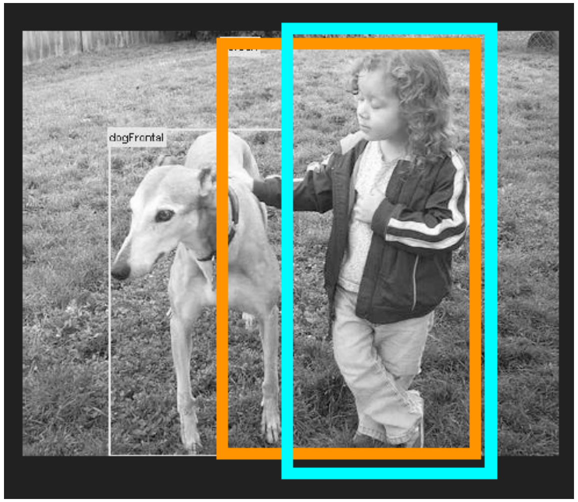
  - positive examples anisotropically scaled to aspect and size of filter

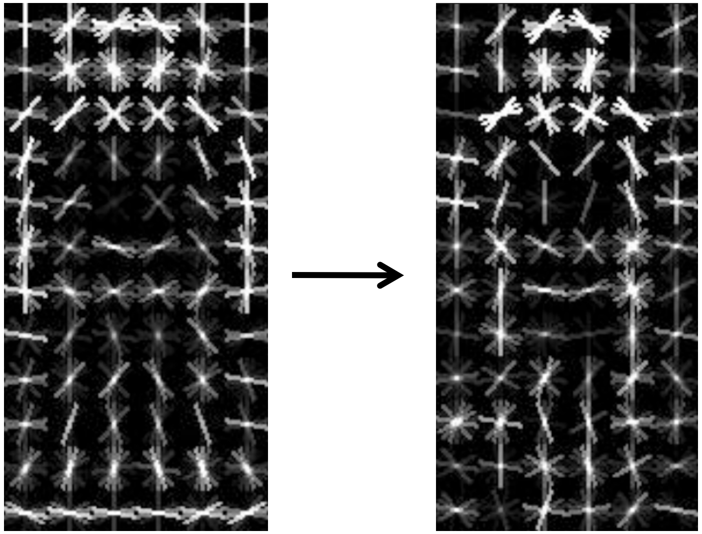
  - random negative examples



Root Filter Update

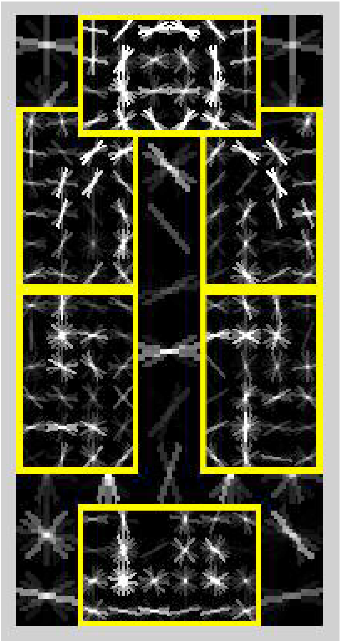
- Find best scoring placement of root filter that significantly overlaps the bounding box
- Retrain F_0 with new positive set

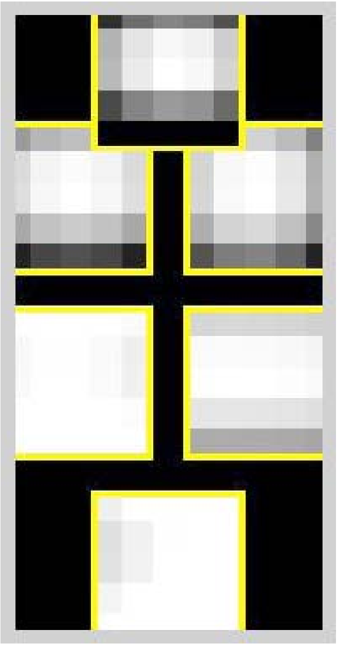




Part Initialization

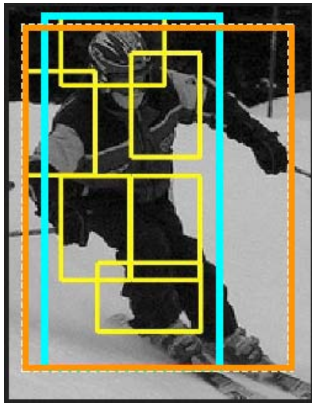
- Greedily select regions in root filter with most energy
- Part filter initialized to subwindow at twice the resolution
- Quadratic deformation cost initialized to weak Gaussian



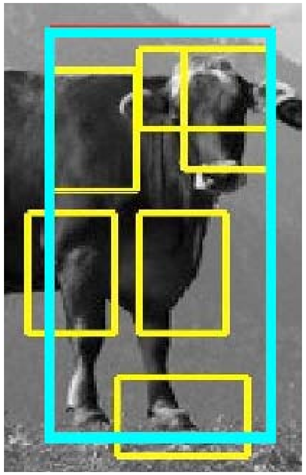


Model Update

- Positive examples – highest scoring placement with > 50% overlap with bounding box
- Negative examples – high scoring detections with no target object (add as many as can fit in memory)
- Train a new model using SVM
- Keep only hard examples and add more negative examples
- Iterate 10 times

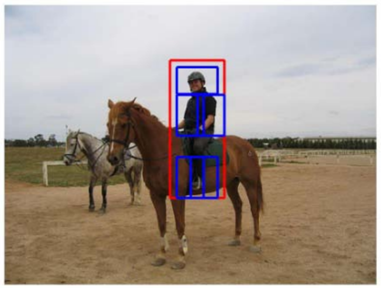


positive example



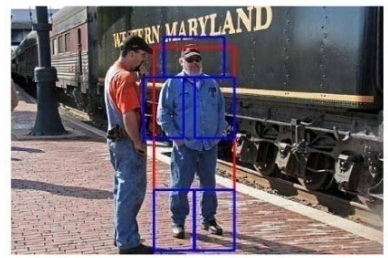
hard negative example

Results – PASCAL07 - Person



0.9562

0.9519



0.8720

0.8298

0.7723

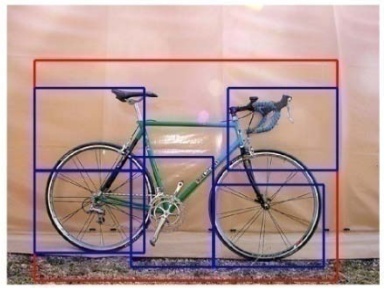
0.7536

0.7186

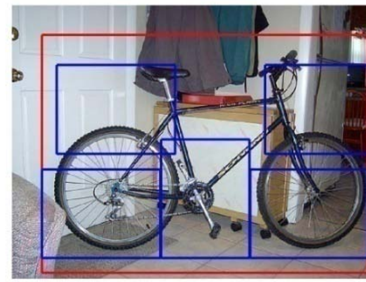
0.6865

Results – PASCAL07 - Bicycle

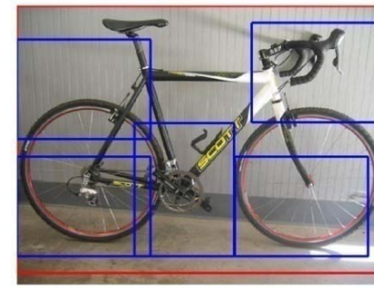
2.1838



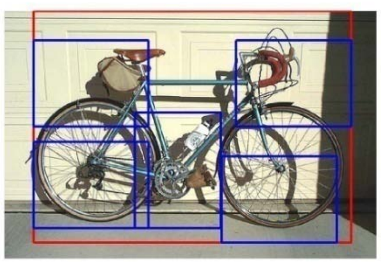
2.1014



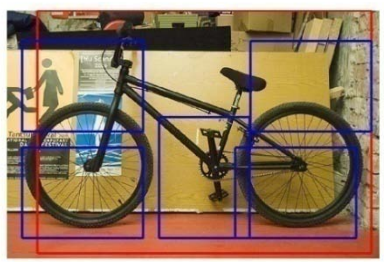
1.8149



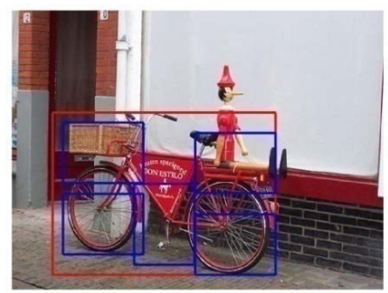
1.6054



1.4806



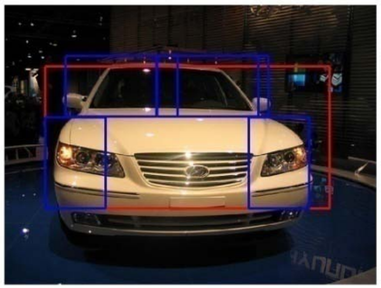
1.4282



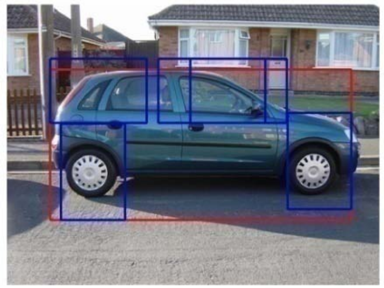
1.3662

1.3189

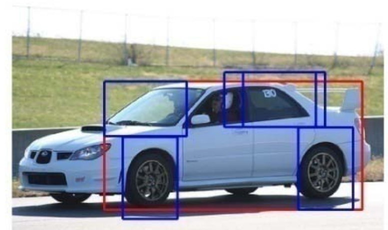
Results – PASCAL07 - Car



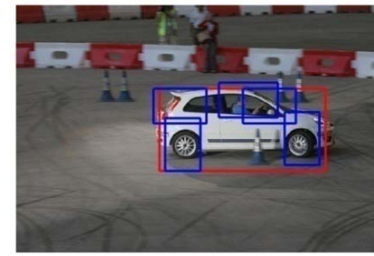
1.5663



1.3875

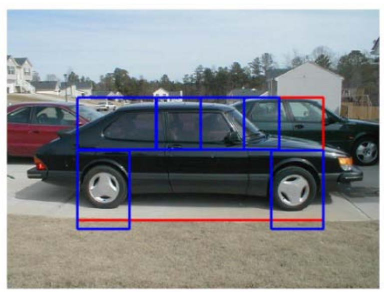


1.2594

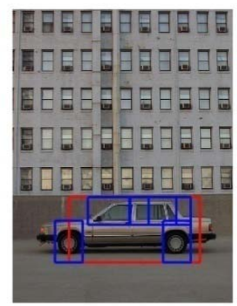


1.1390

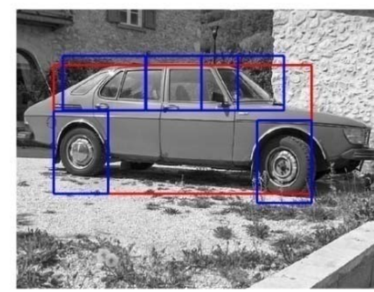
1.1035



1.0645

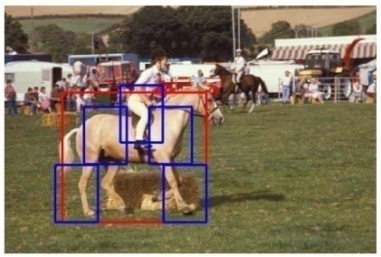


1.0623

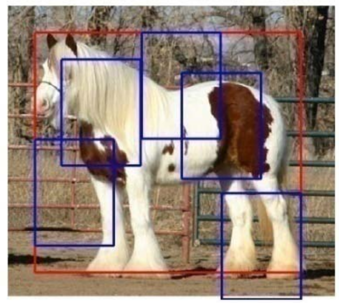


1.0525

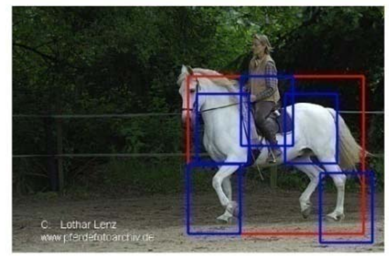
Results – PASCAL07 - Horse



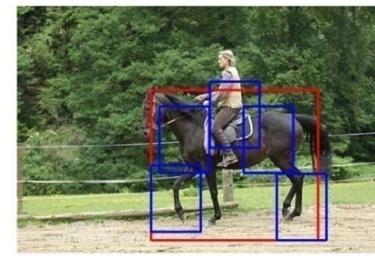
-0.3007



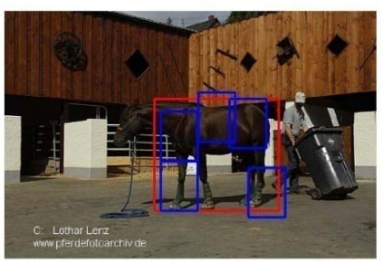
-0.3946



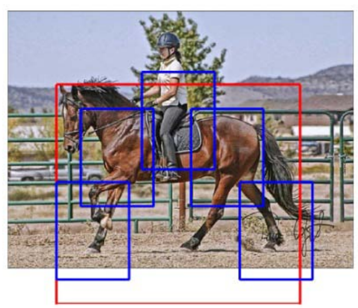
-0.4138



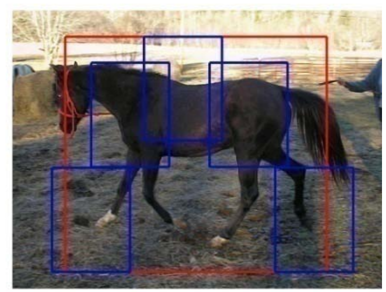
-0.4254



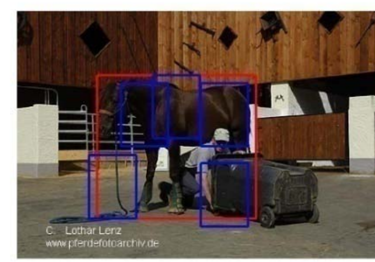
-0.4573



-0.5014

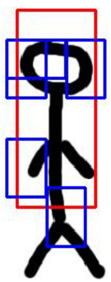


-0.5106

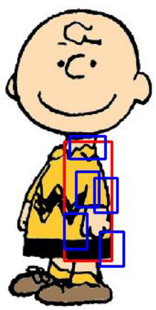


-0.5499

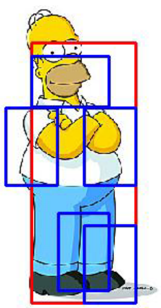
Results - Person



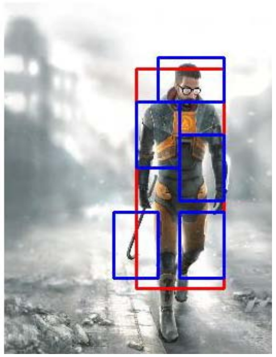
-1.1999



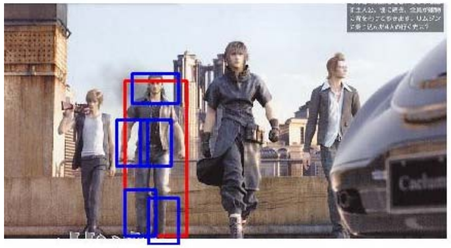
-0.7230



-0.0189



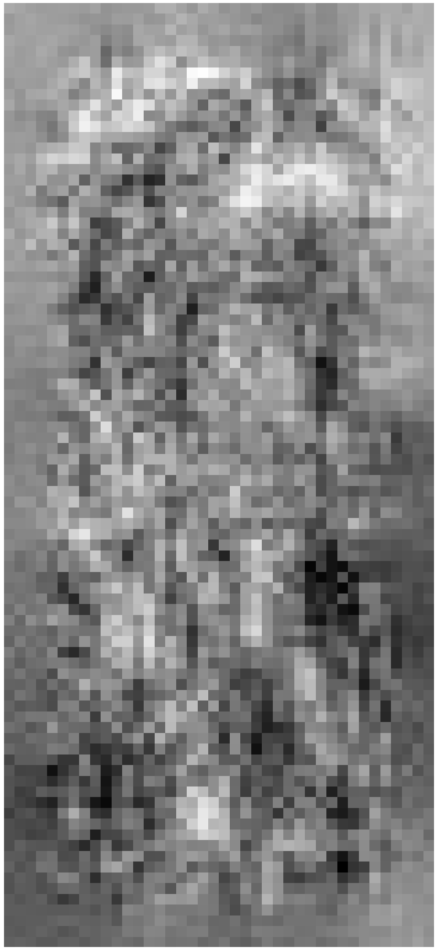
0.1432



0.3267

So do more realistic images give higher scores?

Superhuman

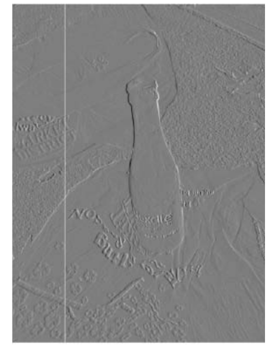


2.56!

Gradient Domain Editing

g_x

g_y



g_x'

g_y'

f(x,y)

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right)$$

$$\nabla f = g$$

$$\nabla^2 f = \nabla g$$

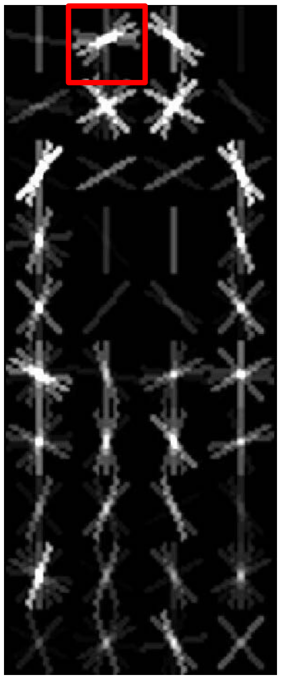
Decompose

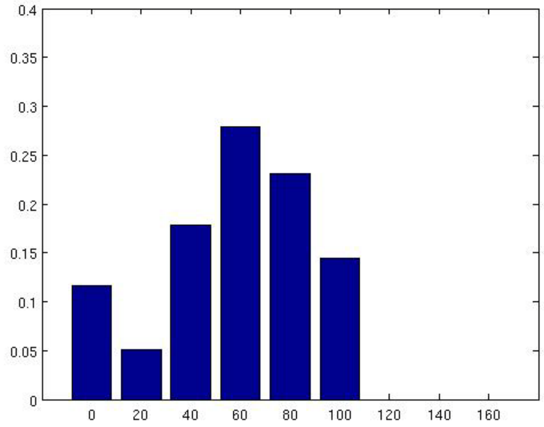
Edit

Reconstruct

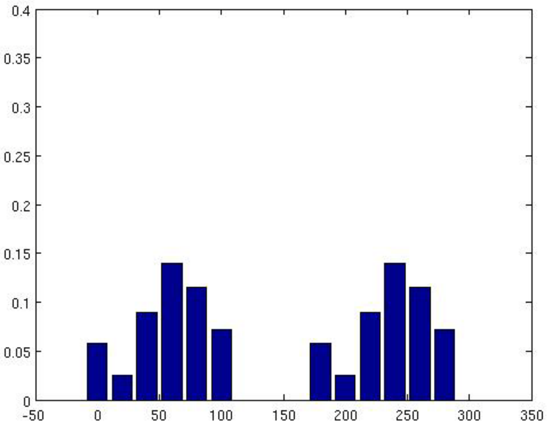
(Solve Poisson Equation)

Generating a “person”



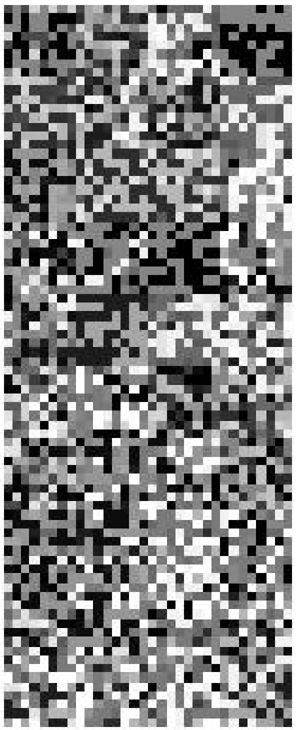


9 orientation bins



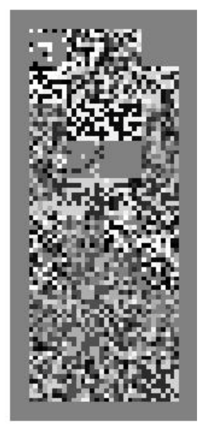
18 orientation bins
for positive and negative

Generating a “person”

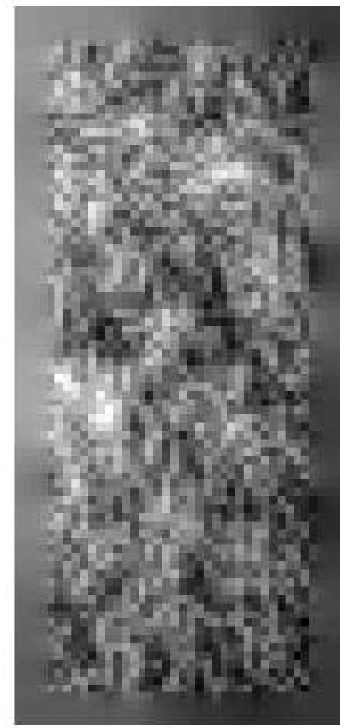


initial orientation
bin assignments

g_x

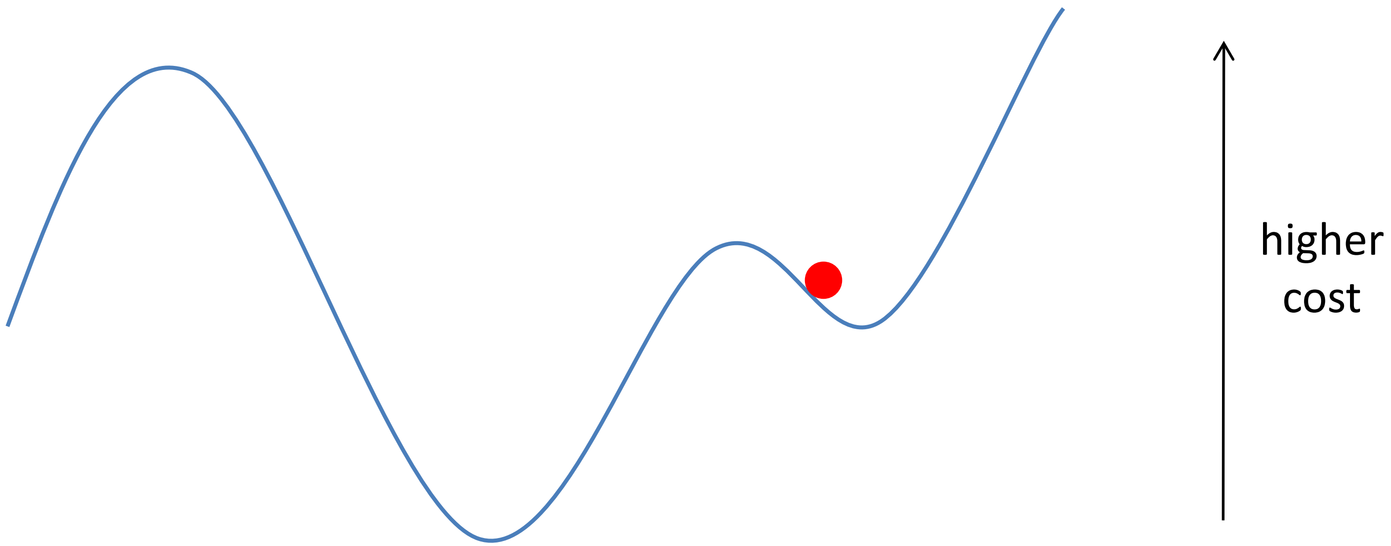


g_y



initial “person”

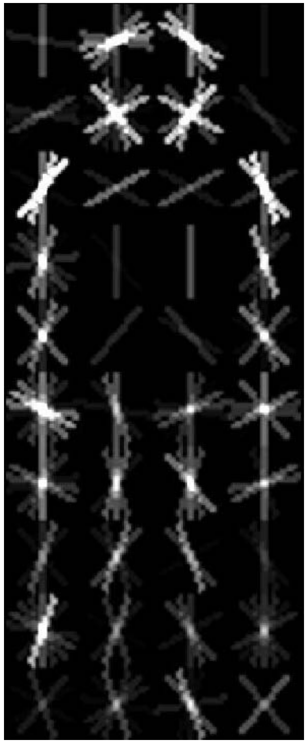
Simulated Annealing

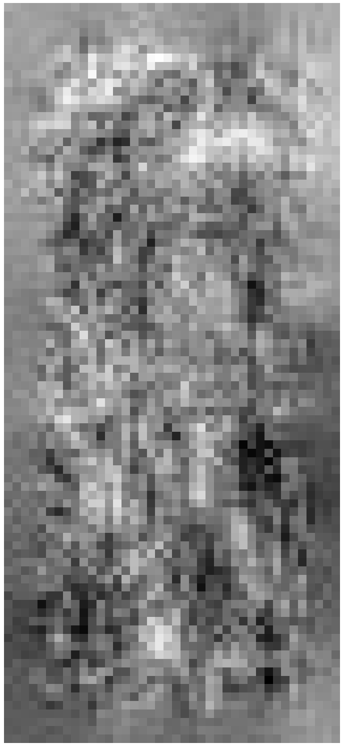


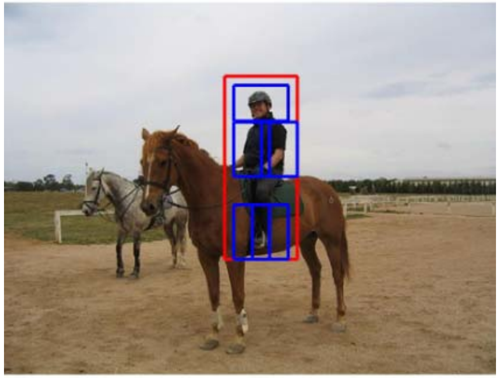
$$P = \exp\left[-\frac{(c_{new} - c_{current})}{T} \right]$$

T is initially high and decreases with number of iterations

Person





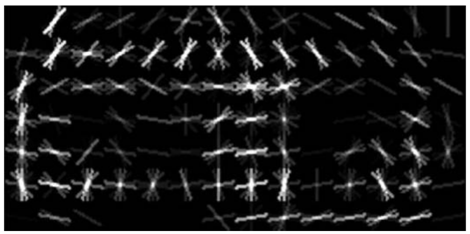


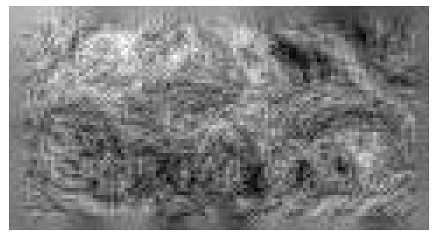
Score: 2.56

Score: 0.96

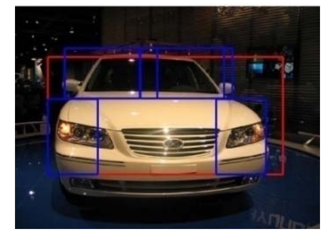
Generated Images

Car





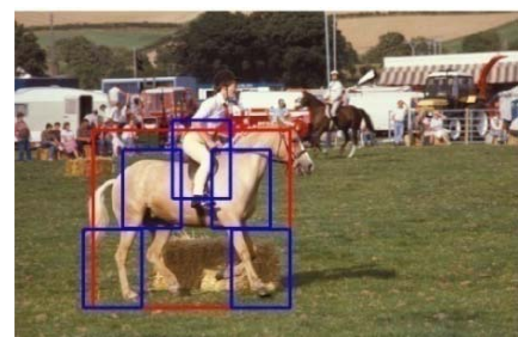
Score: 3.14



Score: 1.57

Horse

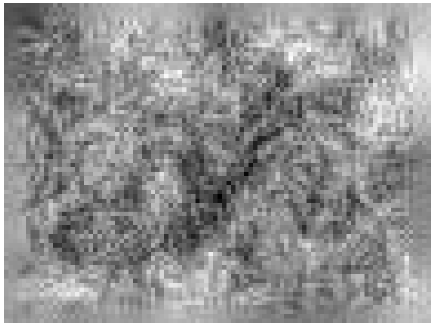
Score: 0.84



Score: -0.30

Generated Images

Bicycle

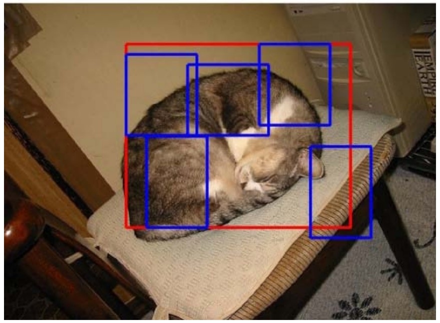


Score: 2.63

Score: 2.18

Cat

Score: 0.80



Score: -0.71

Gradient Erasing

Original
Score: 0.83

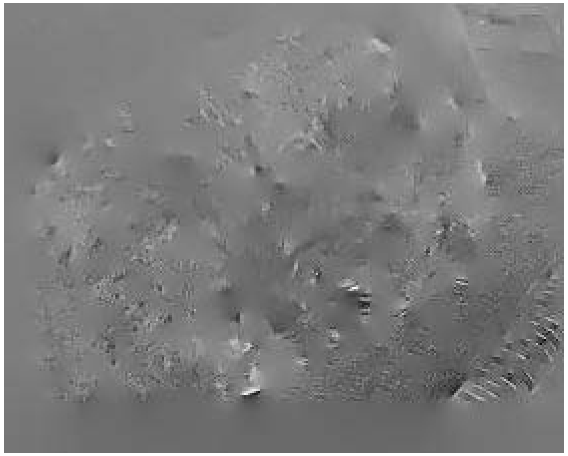
Erased
Score: 2.78

Difference image

Gradient Erasing

Original
Score: -0.76

Erased
Score: 0.26



Difference image

Gradient Addition

Score: 0.83

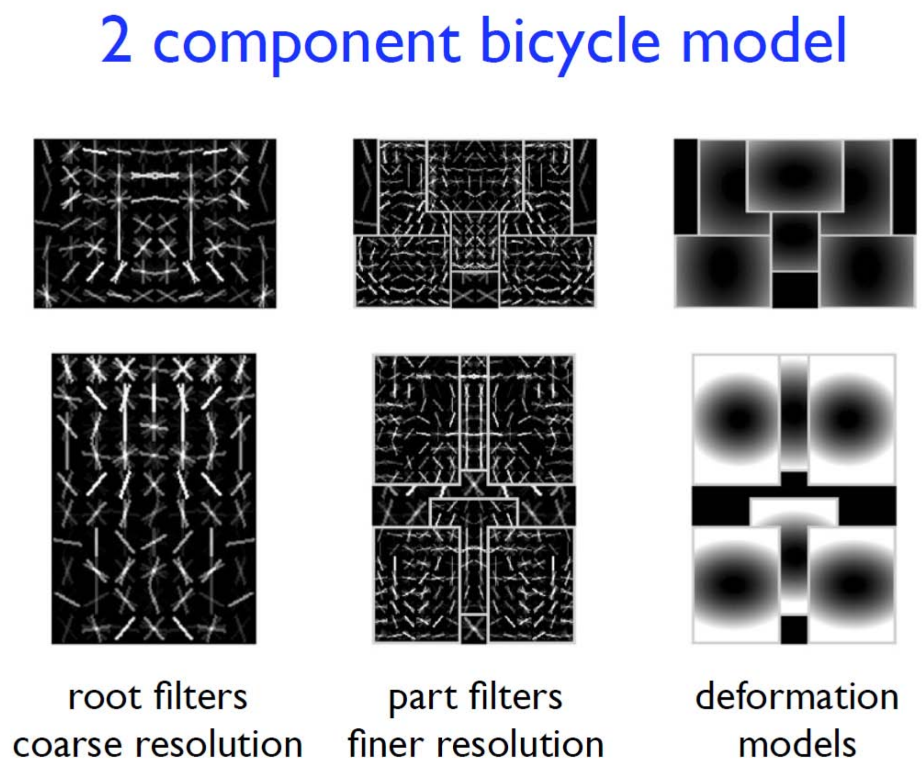
Score: 3.03

Gradient Addition

Score: 2.15

Discriminatively Trained Mixtures of Deformable Part Models

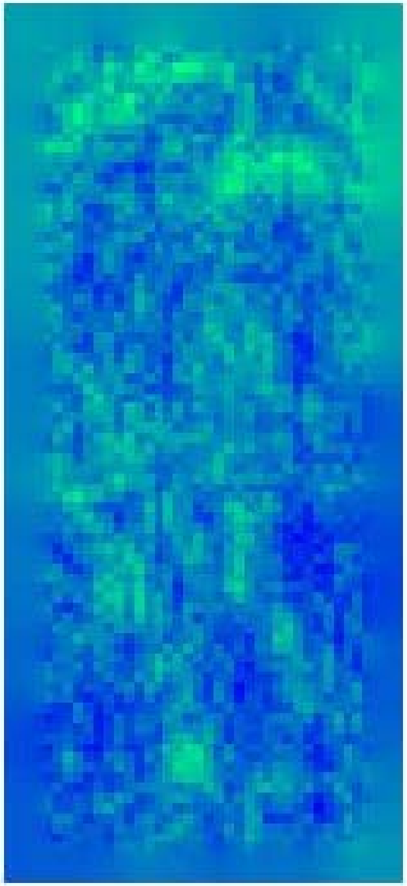
P. Felzenszwalb, D. McAllester, and D. Ramanan



http://www.cs.uchicago.edu/~pff/latent

Slide taken from P. Felzenszwalb

Questions?



Thank You