

HOG + SVM in Pedestrian Detection and Face detection

Alex Lin



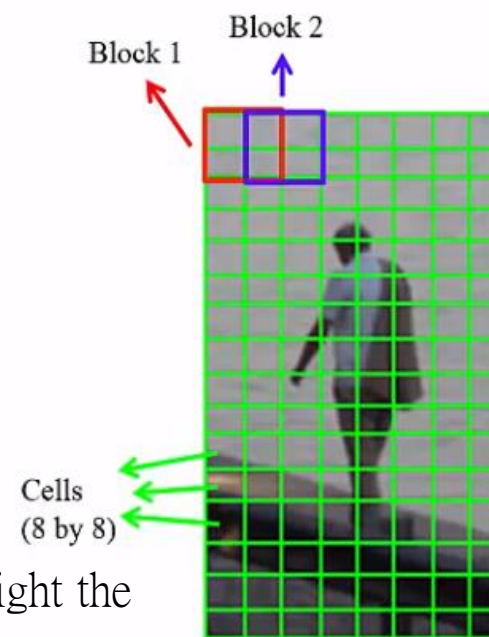
Outline

- ***What is HOG***
- ***Where is SVM***
- ***Multi-Class Detection using Binary Classifiers***
- ***How to do multi-scale detection using HOG+SVM***
- ***Hard Negative Mining***
- ***Part-based method***
- ***Conclusion***

HOG steps



- HOG feature extraction
 - Computer centered horizontal and vertical gradients with no smoothing
 - Compute gradient orientation and magnitudes
 - For color image, pick the color channel with the highest gradient magnitude for each pixel
 - For a 64x128 image,
 - divide the image into 16x16 blocks of 50% overlap
 - $7 \times 15 = 105$ blocks in total
 - Each block should consist 2x2 cells with size 8x8
 - Quantize the gradient orientation into 9 bins
 - The vote is the gradient magnitude
 - Interpolate votes between neighboring bin center
 - The vote can also be weighted with Gaussian to downweight the pixels near the edges of the block.
 - Concatenate histograms (Feature Dimension: $105 \times 4 \times 9 = 3780$)



Computing Gradients

- Centered:
$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x-h)}{2h}$$

- Filter masks in x and y directions

- Centered:

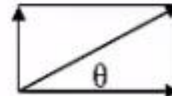
-1	0	1
----	---	---

-1
0
1

- Gradient

- Magnitude:

$$s = \sqrt{s_x^2 + s_y^2}$$



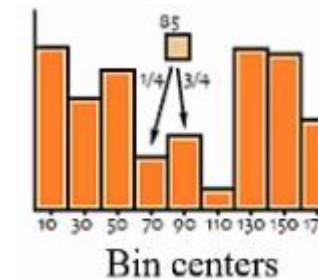
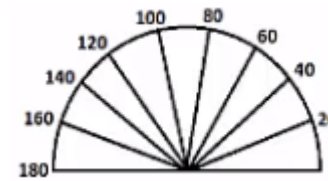
- Orientation:

$$\theta = \arctan\left(\frac{s_y}{s_x}\right)$$

Votes



- Each block consists of 2×2 cells with size 8×8
- Quantize the gradient orientation into 9 bins (0–180)
 - The vote is the gradient magnitude
 - Interpolate votes linearly between neighboring bin centers.
 - Example: if $\theta = 85$ degrees.
 - Distance to the bin center Bin-70 and Bin-90 are 15 and 5 degrees, respectively
 - Hence, ratios are $5/20 = 1/4$, $15/20 = 3/4$
- The vote can also be weighted with Gaussian to downweight the pixels near the edges of the block



Support Vector Machine

- Two-category separable case (cont.)

- Margin

- The smallest distance to two parallel hyperplanes on each side of the separating hyperplane

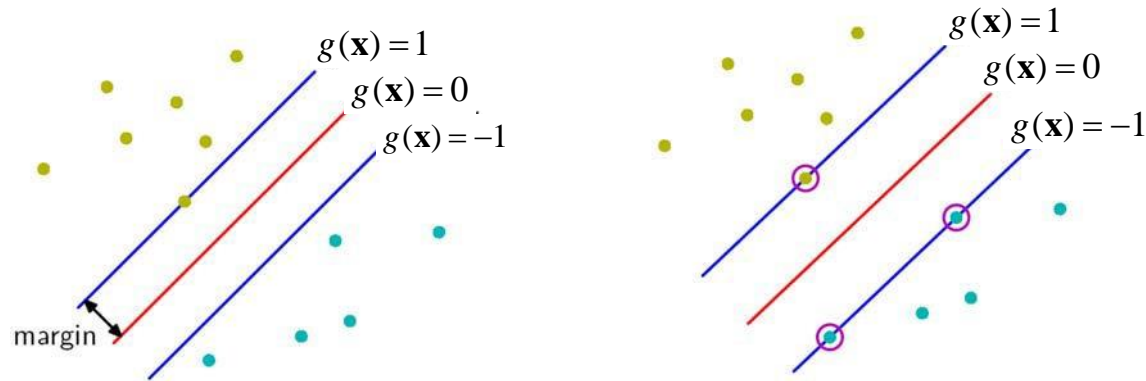


Fig. 7.1 [Bishop 06]

Location of the decision boundary is determined by a subset of the data points, known as support vectors

Support Vector Machine

- Two-category separable case (cont.)

- Maximal margin classifier

- To search for the hyperplane that gives the maximum possible margin

$$\text{maximize the margin: } \frac{1}{\|\mathbf{w}\|} + \frac{1}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$$

$$\text{requiring that } \mathbf{w}^T \mathbf{x}_i + w_0 \geq +1 \quad \forall \mathbf{x}_i \in \omega_1$$

$$\mathbf{w}^T \mathbf{x}_i + w_0 \leq -1 \quad \forall \mathbf{x}_i \in \omega_2$$

- Let y_i be the class indicator

- Then the problem becomes

$$\text{minimize } J(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2}$$

$$\text{subject to } y_i (\mathbf{w}^T \mathbf{x}_i + w_0) \geq +1 \quad i = 1, \dots, N$$

$$\text{where } y_i = \begin{cases} +1, & \text{if } \mathbf{x}_i \in \omega_1 \\ -1, & \text{if } \mathbf{x}_i \in \omega_2 \end{cases}$$

Nonlinear optimization with
linear inequality constraints

Support Vector Machine

- Optimization for constrained problem
(Appendix C.4 [Theodoridis 09])

- The primal problem

$$\begin{array}{ll}\text{minimize} & J(\boldsymbol{\theta}) \\ \text{subject to} & f_i(\boldsymbol{\theta}) \geq 0, \quad i = 1, \dots, N\end{array}$$

- Lagrangian function

- To augment the objective function with a weighted sum of the constraint functions

$$L(\boldsymbol{\theta}, \boldsymbol{\lambda}) = J(\boldsymbol{\theta}) - \sum_{i=1} \lambda_i f_i(\boldsymbol{\theta})$$

- $\{\lambda_i, i = 1, \dots, N\}$: the Lagrange multipliers

- $\boldsymbol{\lambda}$: the dual variables or Lagrange multiplier vectors

Support Vector Machine

- Optimization for constrained problem (cont.)

- The dual function

$$D(\boldsymbol{\lambda}) = \inf_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \inf_{\boldsymbol{\theta}} \left(J(\boldsymbol{\theta}) - \sum_{i=1}^N \lambda_i f_i(\boldsymbol{\theta}) \right)$$

- Suppose $\tilde{\boldsymbol{\theta}}$ is a feasible point of the primal problem and $\lambda_i \geq 0$

$$\text{i.e., } f_i(\tilde{\boldsymbol{\theta}}) \geq 0, \quad \forall i$$

- Then

$$-\sum_{i=1}^N \lambda_i f_i(\tilde{\boldsymbol{\theta}}) \leq 0$$

$$L(\tilde{\boldsymbol{\theta}}, \boldsymbol{\lambda}) = J(\tilde{\boldsymbol{\theta}}) - \sum_{i=1}^N \lambda_i f_i(\tilde{\boldsymbol{\theta}}) \leq J(\tilde{\boldsymbol{\theta}})$$

- Thus

$$D(\boldsymbol{\lambda}) = \inf_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \boldsymbol{\lambda}) \leq L(\tilde{\boldsymbol{\theta}}, \boldsymbol{\lambda}) \leq J(\tilde{\boldsymbol{\theta}})$$

- The dual function yields lower bounds on the optimal value of the primal problem

$$D(\boldsymbol{\lambda}) \leq p^*$$

Support Vector Machine

- Optimization for constrained problem (cont.)
 - The Lagrange dual problem
 - maximize $D(\lambda)$
 - subject to $\lambda \geq 0$
 - The Lagrange dual problem is a convex optimization problem, whether or not the primal problem is convex
 - The objective is concave
 - The constraint is convex

How to do multi-class classification using multiple binary classifiers?

- Multicategory case
 - One-against-the-rest
 - Use M two-class linear discriminants
 - Assign \mathbf{x} to ω_i or not ω_i
 - Class imbalance problem
 - #negative \gg #positive
 - Pairwise separation:
 - Use $M(M-1)/2$ linear discriminants
 - For every pair of classes
 - Decision is made by majority vote
 - Could lead to nonlinear separation of classes

Both approaches lead to undefined regions in the feature space

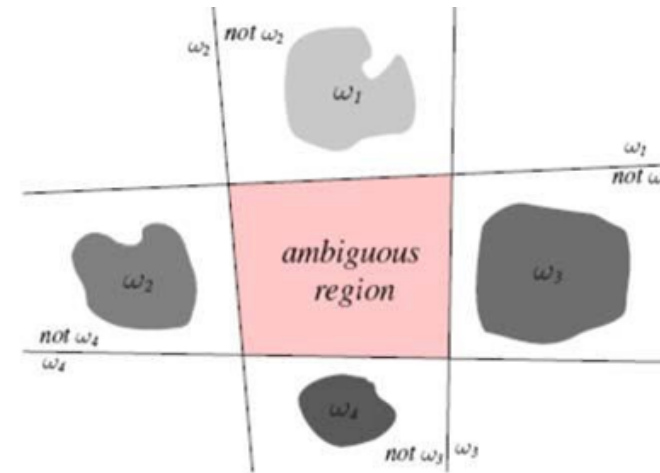
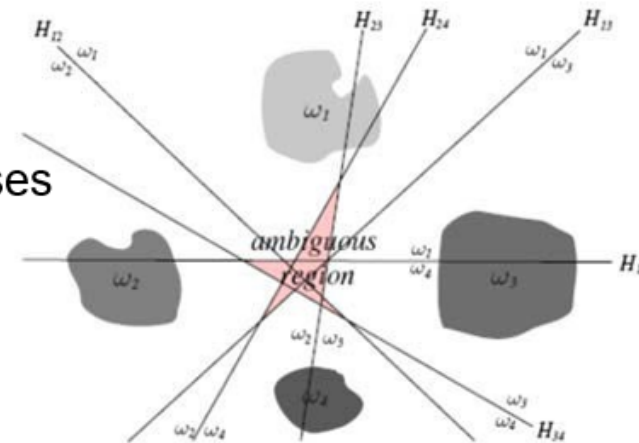


Fig. 5.3 [Duda 01]



From Image classification to object detection



.

.



Sliding window



HOG feature
extraction



SVM
classification

How to apply HOG in multi-scale detection?

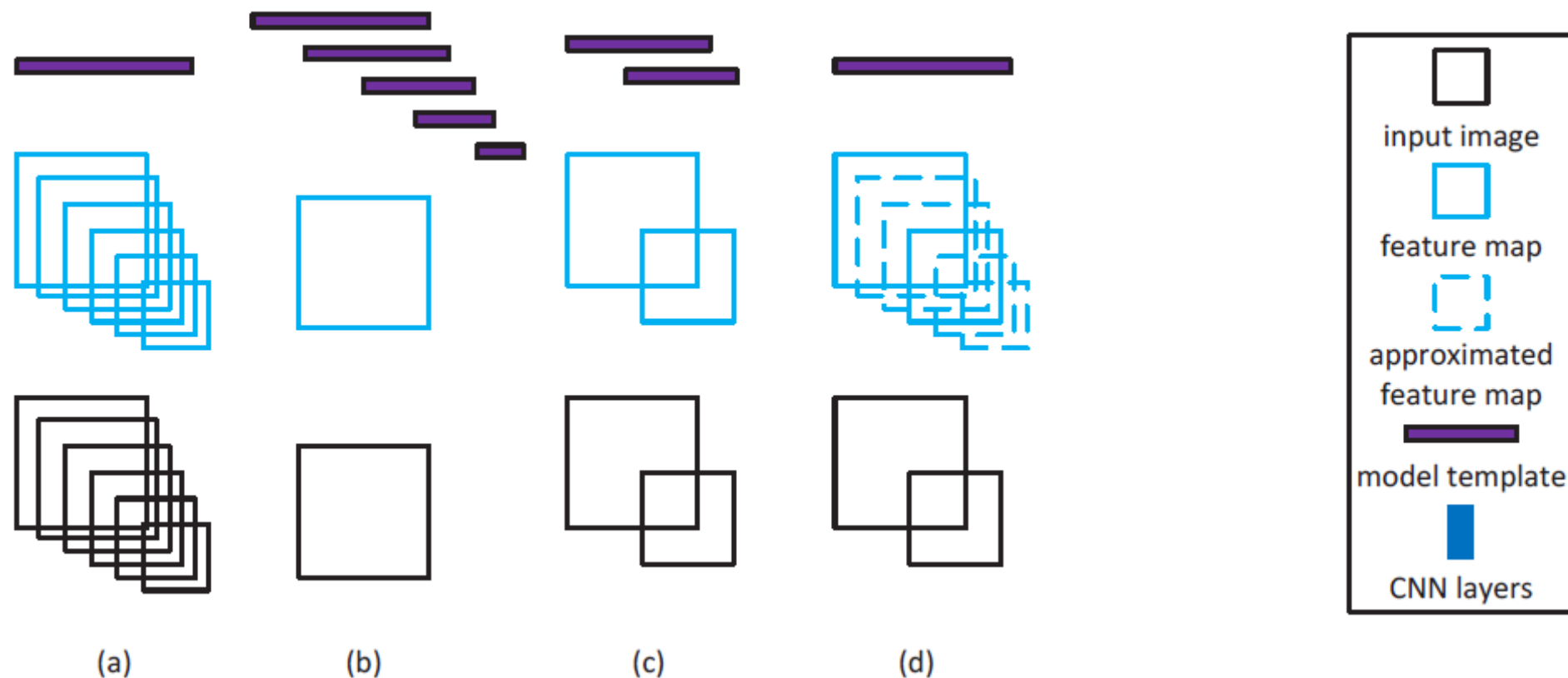
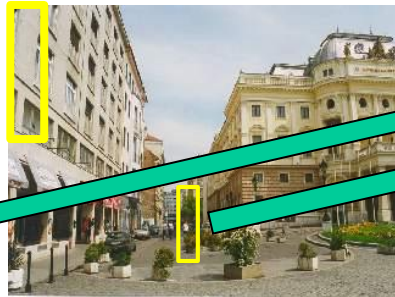


Fig. 2. Different strategies for multi-scale detection. The length of model template represents the template size.

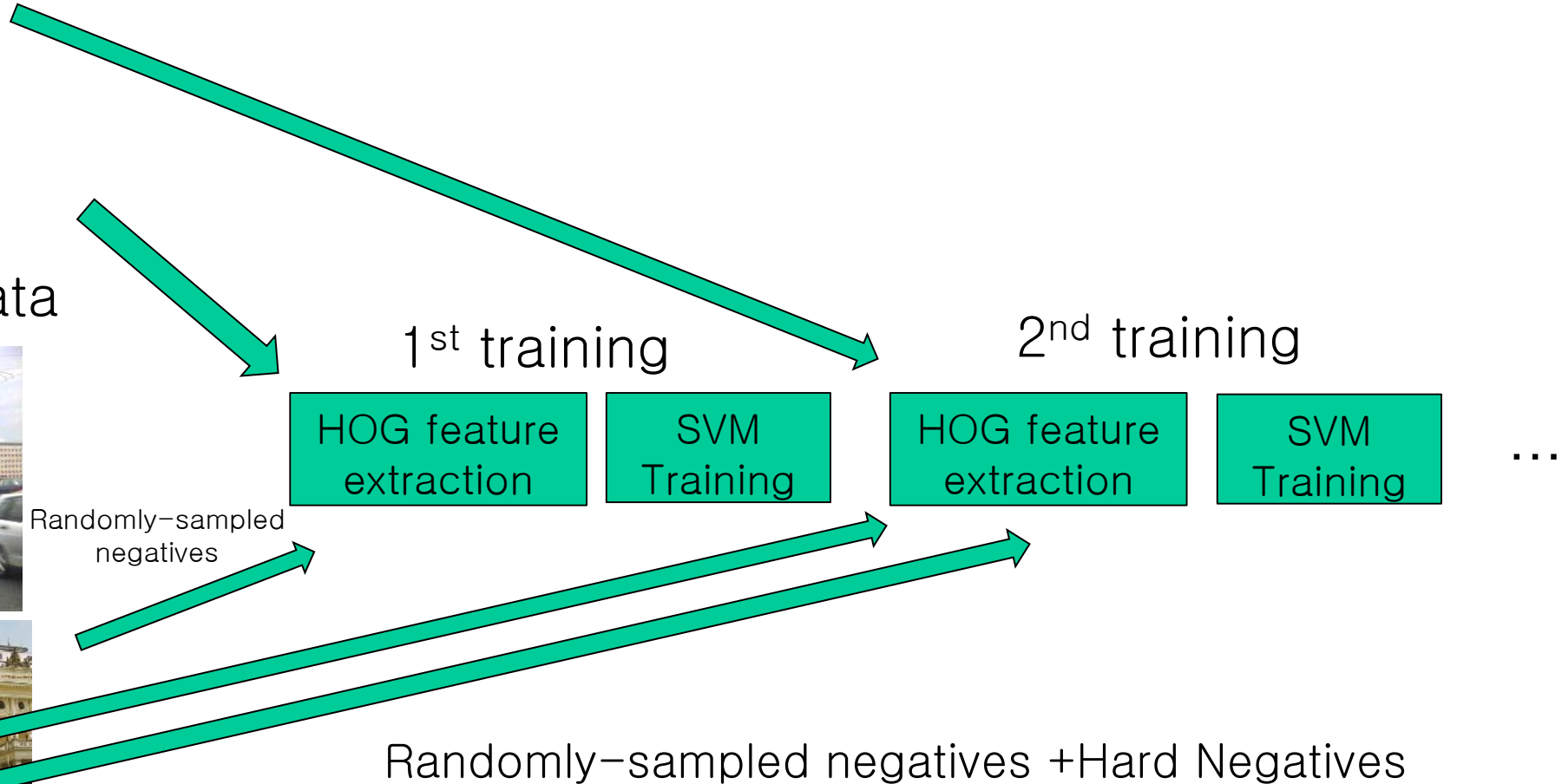
Hard Negative Mining in training a detector



Positive training data

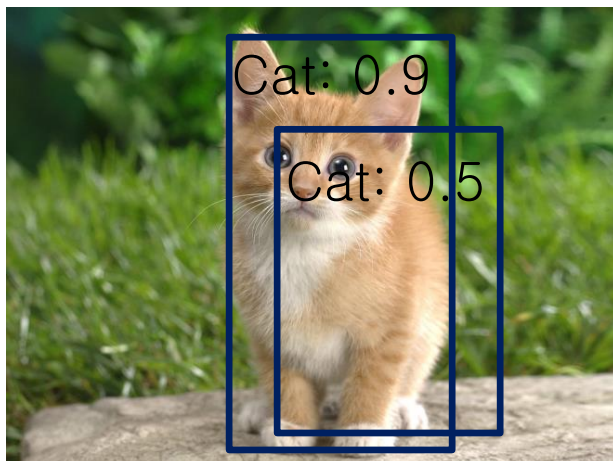



Negative training data pool
(non-pedestrian-containing images)



Non-maximal Suppression

Boxes with lower confidence and IOU < 0.5 with the one with higher confidence should be eliminated

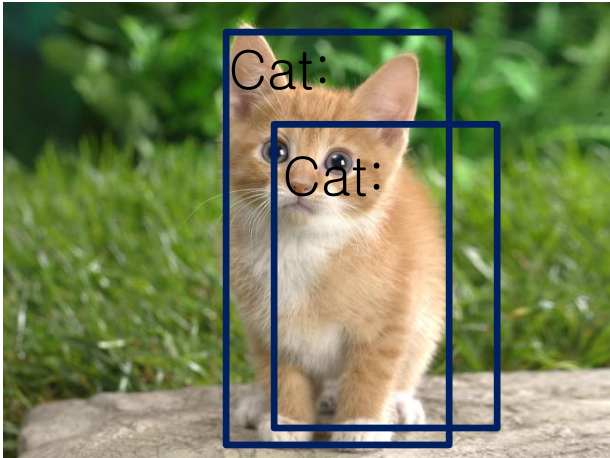


$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$
A diagram illustrating the calculation of Intersection over Union (IoU). It shows two overlapping blue squares. The top square is labeled "Area of Overlap" and the bottom square is labeled "Area of Union". The formula $\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$ is shown to the left of the diagram.

<http://blog.csdn.net/lanchunhui>

Non-maximal Suppression: a Naïve implementation

Pick the 1st bounding box according to some heuristic rules and eliminate others if their IOU is higher than the chosen one



Detection result without using Hard Negative



Detection result with using Hard Negative



The above training and testing dataset in pedestrian detection:

- Training dataset
 - Positive: INRIA Person (1218 images)
 - Negative: INRIA Person (614 images)
- Test image:
 - Random image downloaded from internet.

Example HOG+SVM: face detection

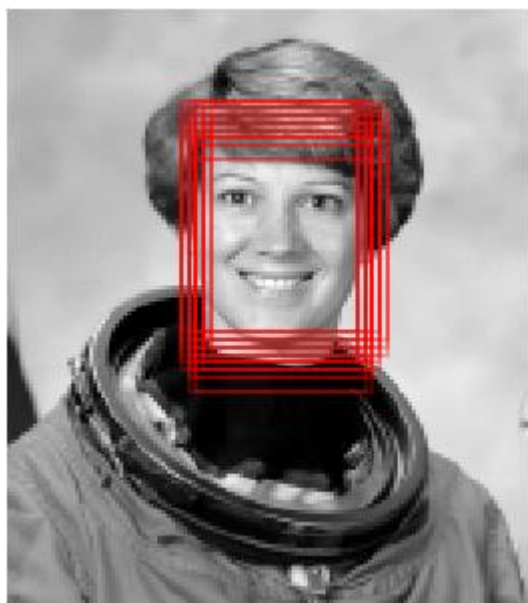
- Training dataset
 - Positive: Labeled faces in the wild (13233 images)
 - Negative: random cropped images from 'camera', 'text', 'coins', 'moon', 'page', 'clock', 'immunohistochemistry', 'chelsea', 'coffee', 'hubble_ deep_ field ' in “The USC-SIPI Image Database”
- Testing image:
 - Test image: astronaut Eileen Collins in The USC-SIPI Image Database”



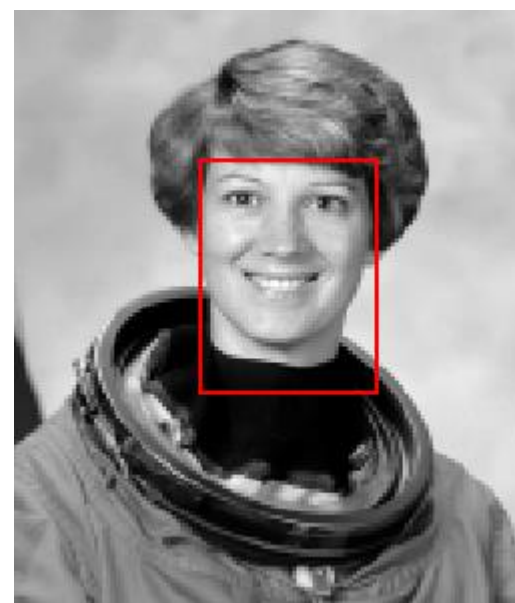
Positive training images



Negative training images

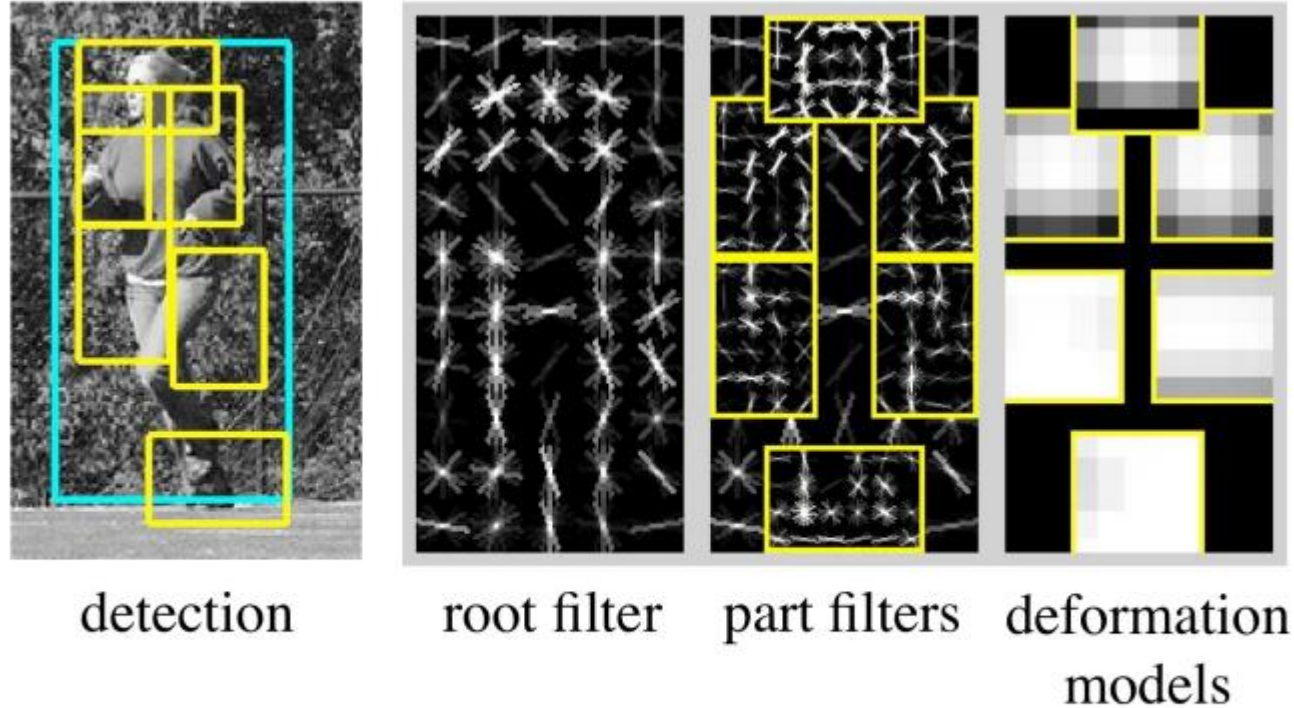


Detection results before NMS



Detection results after NMS

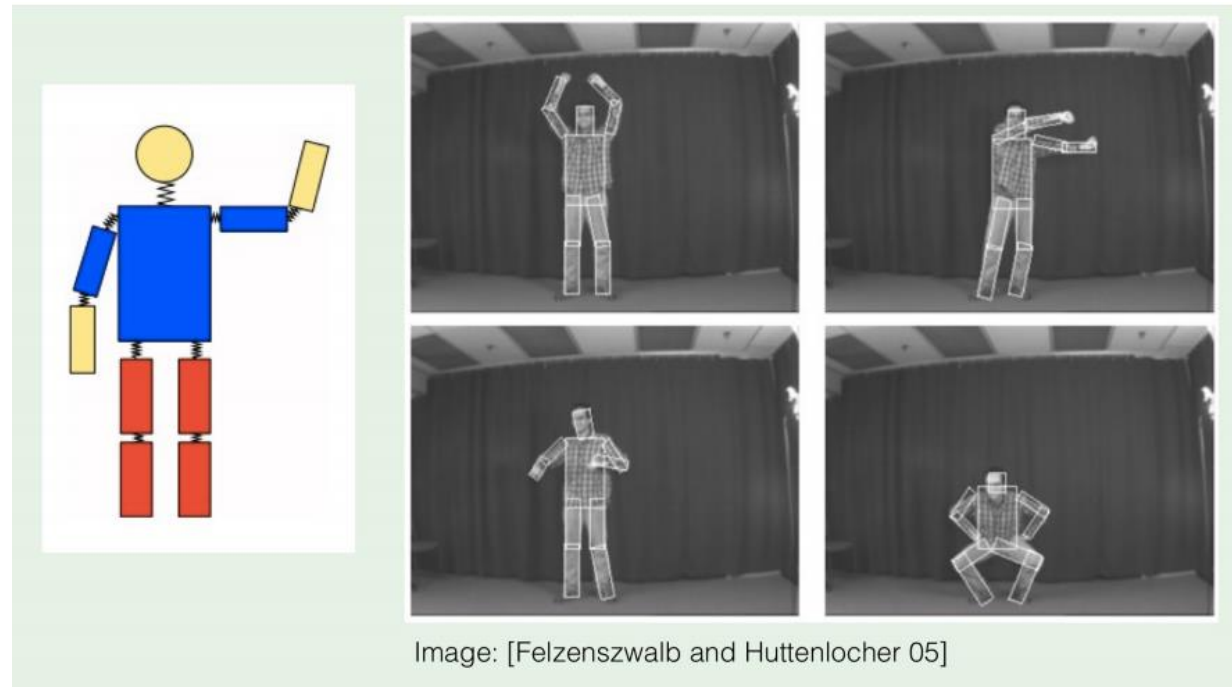
More advanced method: Deformable Part Model



Model has a root filter plus deformable parts

More advanced method: Deformable Part Model

- Problem: More expressive object models are difficult to train because they often use latent (unobserved/unlabeled) information.
- Insight: Part-based model could provide more robustness/unified feature distribution across different viewing-angle, occlusion level, and degree of deformation.



Conclusion



- Shallow feature extraction would only produce features with limited invariance.
- Part-based method could provide better invariance in terms of each distinct part to detect the whole object.
- In the era of deep learning, the sophisticated detection pipeline dealing with feature invariance is internalized in each layer of a neural network.