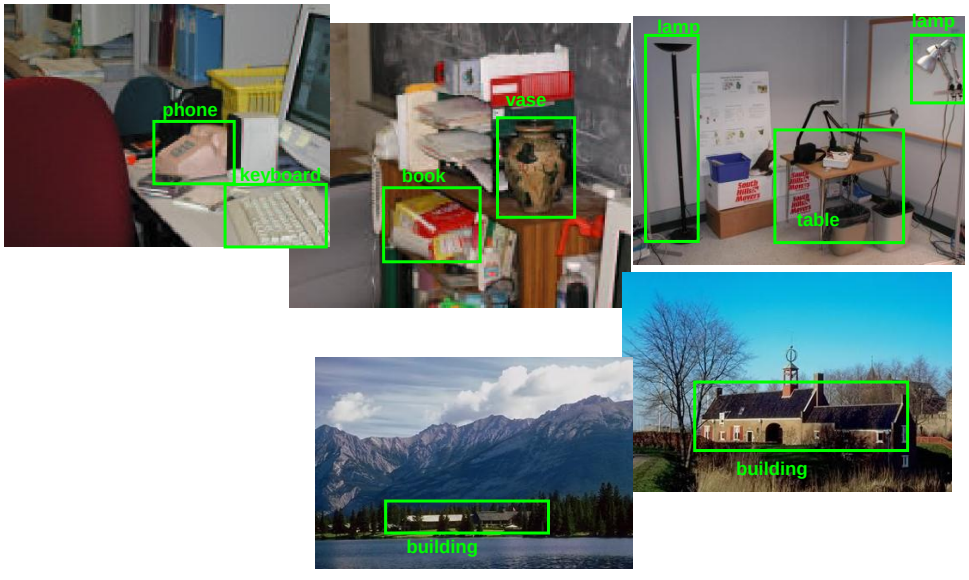
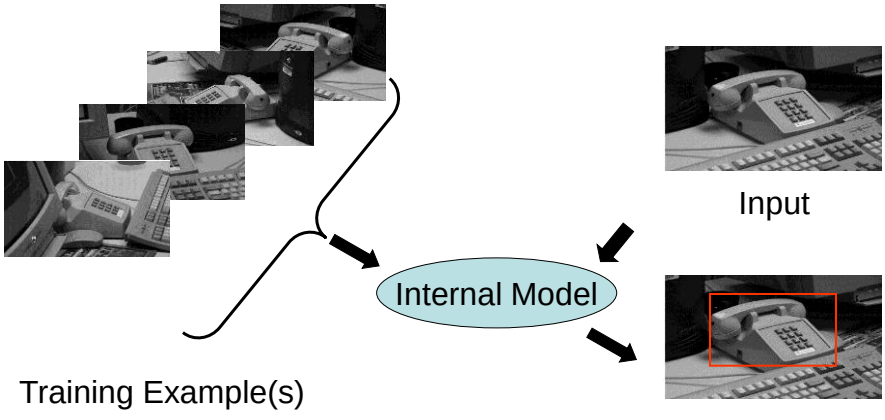
Recognition

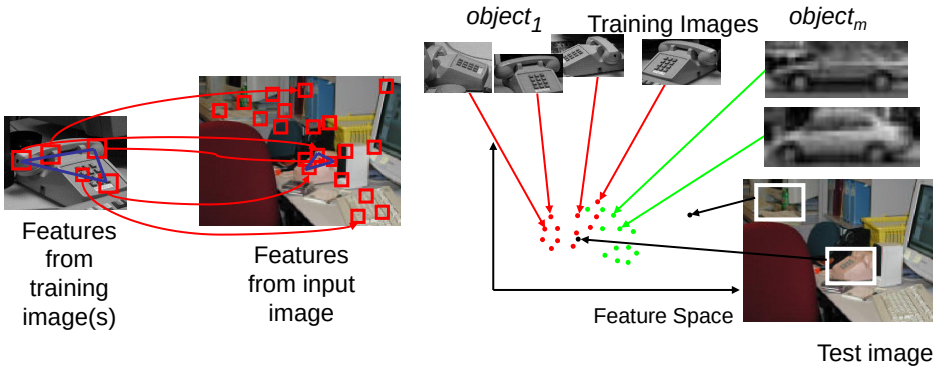
Some of the material from: Fei-Fei Li, Antonio Torralba, Szeliski&Seitz, Rob Fergus

What Is Recognition?





Recognition: Simplified story



Approaches based on using feature matches and geometric relations

Approaches based on classifying/matching image patches (windows)

Example Datasets for Category Recognition

Collecting datasets (towards 10^{6-7} examples)

- ESP game (CMU)
Luis Von Ahn and Laura Dabbish 2004
- LabelMe (MIT)
Russell, Torralba, Freeman, 2005
- StreetScenes (CBCL-MIT)
Bileschi, Poggio, 2006
- WhatWhere (Caltech)
Perona et al, 2007
- PASCAL challenge
2006, 2007
- Lotus Hill Institute
Song-Chun Zhu et al 2007
- Tiny Images (MIT)
Torralba, Fergus & Freeman 2007
- ImageNet
Fei Fei Li 2009

The PASCAL Visual Object Classes Challenge 2007

The twenty object classes that have been selected are:

Person: person
Animal: bird, cat, cow, dog, horse, sheep
Vehicle: aeroplane, bicycle, boat, bus, car, motorbike, train
Indoor: bottle, chair, dining table, potted plant, sofa, tv/monitor

M. Everingham, Luc van Gool , C. Williams, J. Winn, A. Zisserman 2007

ImageNet (http://www.image-net.org/)

- Total number of non-empty synsets: 15589
- Total number of images: 11,231,732
- Number of images with bounding box annotations: 195,331
- Number of synsets with SIFT features: 1000
- Number of images with SIFT features: 1.2 million

Lotus Hill Research Institute image corpus

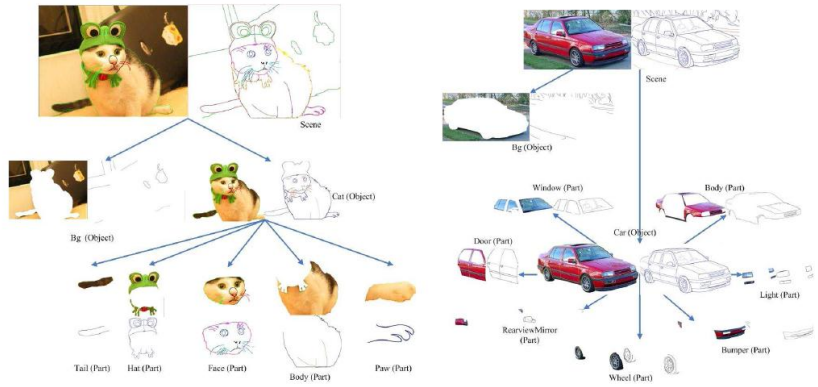


Figure 5: Two examples of the parse trees (cat and car) in the Lotus Hill Research Institute image corpus. From [87].

Z.Y. Yao, X. Yang, and S.C. Zhu, 2007

Evaluation

- Terminology:
  - Classification: Is the object in the image?
    - One-vs.-all
    - Forced choice among *N*
  - Detection: Is the object in the image and *where*?

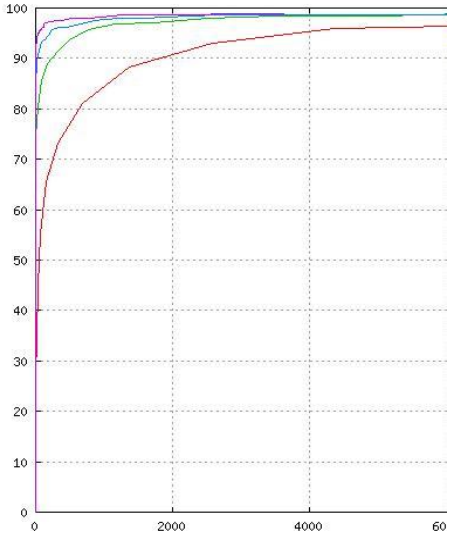
Correct detection:

$$\frac{|GT \cap D|}{|GT \cup D|} > T$$

- Terminology:
  - True positive TP=number of examples containing the object with object detected correctly
  - True negative TN=number of examples *not* containing the object with object *not* detected
  - False positive FP=number of examples *not* containing the object with object incorrectly detected
  - True positive FN=number of examples containing the object with object *not* detected

ROC curve

Detection Rate
TP/(TP+FN)



Most useful in classification tasks

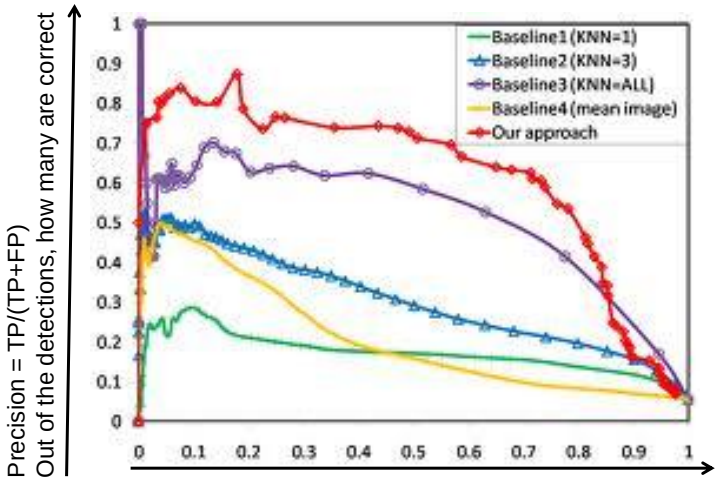
Requires TN to be known

EER = Equal error rate = As many false positives as false negatives

AUC = Area under the curve

False Detections FP or
False detection rate FP/(TN+FP)
Or FPPI = False Positives Per Image

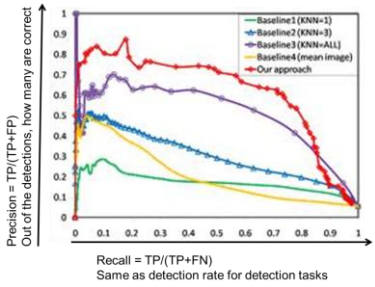
Precision-Recall curve



Recall = TP/(TP+FN)
Same as detection rate for detection tasks

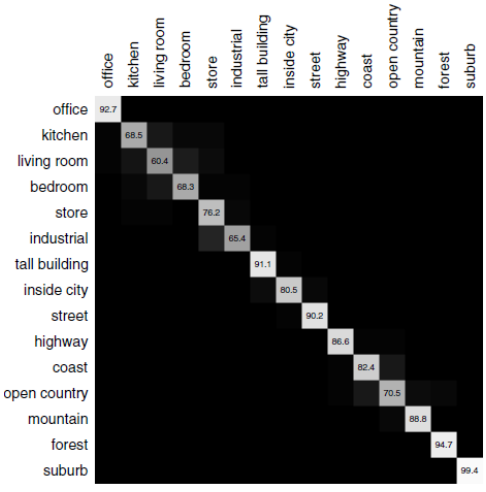
Davis and Goadrich. The Relationship Between Precision-Recall and ROC Curves. ICML (Intern. Conf. Machine Learning) 2006.

Precision-Recall curve



- Invented for retrieval tasks
- Do *not* need to know TN!
- Summary performance numbers:
  - AUPRC = Area under the curve
  - AP = Average precision
  - F-measure = $\frac{2PR}{P+R}$ (1 is perfect performance)

Confusion matrix

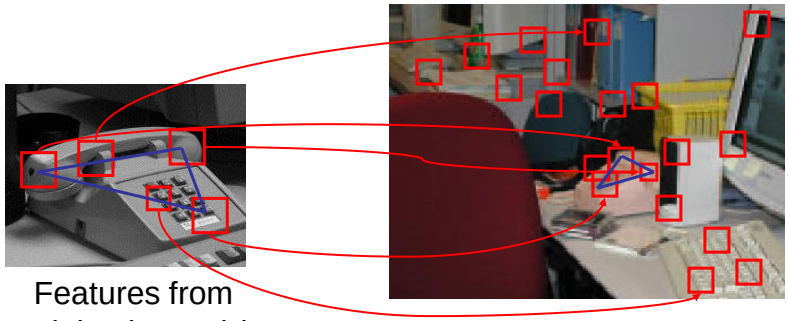


	Build-ing	Grass	Tree	Cow	Sky	Aero-plane	Face
Building	22			2		1	1
Grass		62	3				
Tree			28				
Cow				22			1
Sky					44		
Aeroplane	1					14	
Face							15

- For forced choice classification tasks
- Accuracy = (correct samples)/(total samples)
- Different effect of class imbalance:
  - Per-class accuracy
  - Total overall accuracy

Feature matching

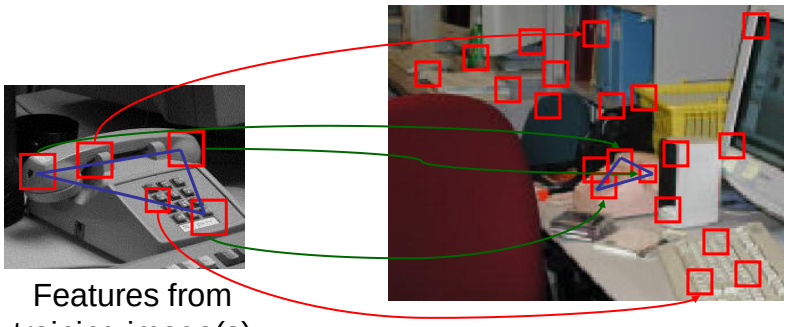
Feature Matching & Geometric Relations



Features from training image(s)

Features from input image

Feature Matching & Geometric Relations



Features from training image(s)

Features from input image

Aspect is different between training and test images → “invariant” features
Local feature similarity is not sufficient → use global geometric consistency
Large number of features → define distance in feature space + efficient indexing

- Image gradients are sampled over 16x16 array of locations in scale space
- Create array of orientation histograms
- 8 orientations x 4x4 histogram array = 128 dimensions

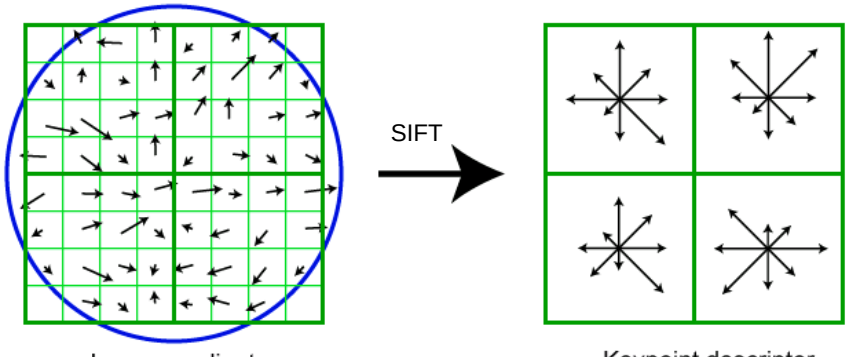


Image gradients

Keypoint descriptor

Outline

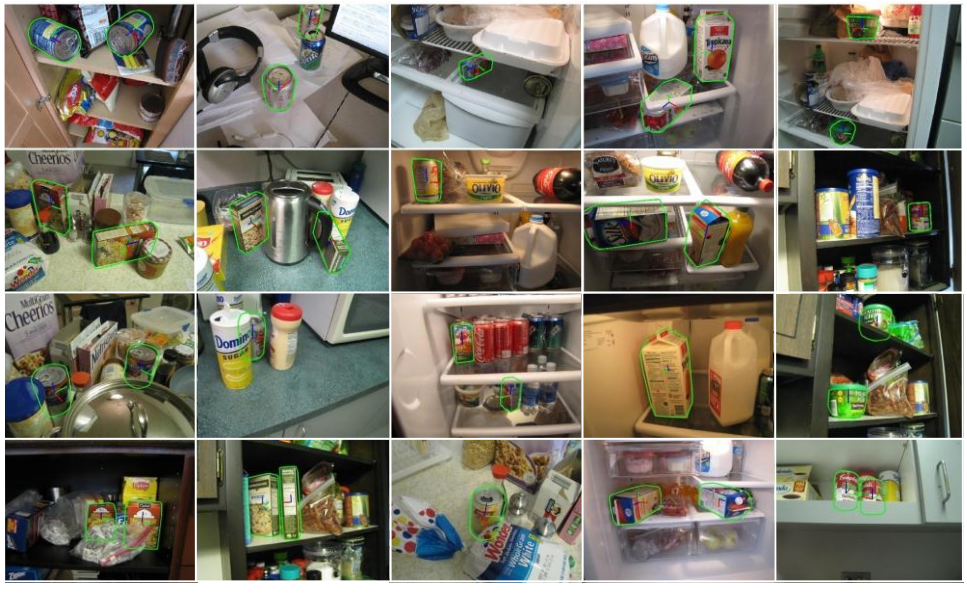
- For each SIFT feature, find the closest feature in the reference set
- Keep match if distance is greater than (1+δ) times distance to next neighbor
- Apply RANSAC to set of potential correspondences to verify geometric consistency

Examples

Model: Limited set of views of the objects

Run-time input: Object in arbitrary pose and illumination conditions in uncontrolled environments

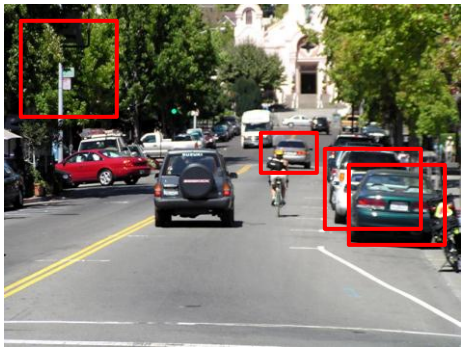
Examples



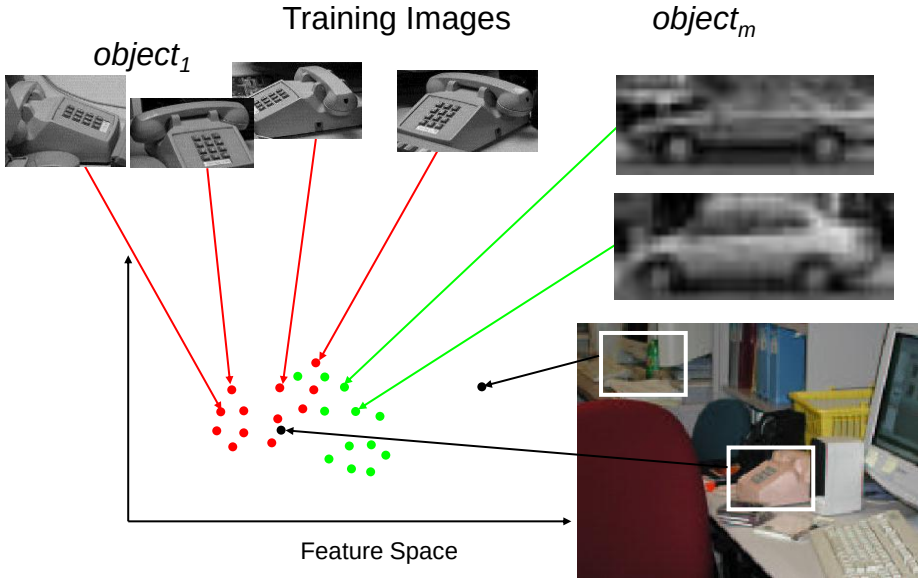
Feature Matching & Geometric Relations

- *Positive*:
  - Relatively simple implementation
  - Well-defined, efficient operations: Indexing, geometric verification (RANSAC), etc.
- *Negative*:
  - Information reduced to a relatively small set of discrete features
  - Generalization issues for broad categories How do I deal with recognizing the class of all the chairs?
- *Alternative*: Use the pixels in the image windows directly

Window-based techniques



Window-based techniques: The short story

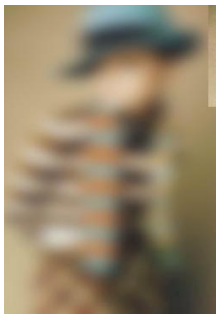


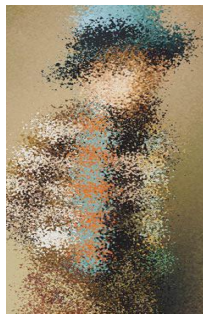
What representation should we use?

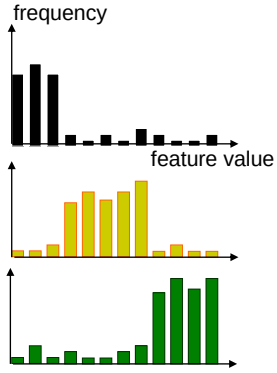
Test image

Window-based techniques: The short story

Example training window :







1. Template classification: Techniques inspired from template matching: Compare directly with pixels from training image. Perhaps use blurring to be robust to small shifts, etc.

2. Locally orderless structures: Intermediate solution: Use local histograms of features instead of a single global histogram. Potentially combines the advantages of templates (strong spatial information) and histograms (robustness to distortions of the image)

3. Histograms: Use histograms of features to compare the windows. Example: Bags Of Words techniques. Problem: All local spatial information is lost since the representation is global. Will be robust to changes but much less discriminative.

Example from: The Structure of Locally Orderless Images, Jan J. Koenderink AND A. J. Van Doorn, IJCV 1999.

Template classification

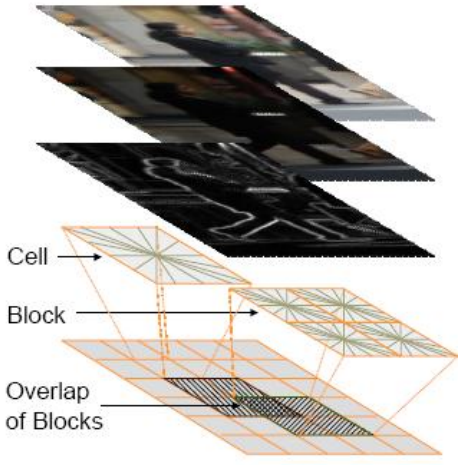
- SVMs
- Combination of simple classifiers (boosting)
- Neural networks, deep learning

Template classification

- SVMs
- Combination of simple classifiers (boosting)
- Neural networks, deep learning

Example: Pedestrian detection

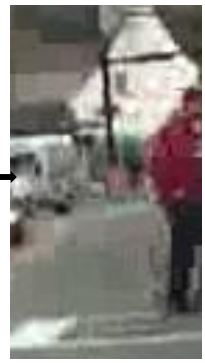
- Compute the gradients of the input image in a detection window
- Compute histograms of the gradients over 16 directions in overlapping blocks
- Combine all the histograms into a single feature vector ***f***
- Apply a classifier (linear SVM) trained off-line on a large training set to ***f***



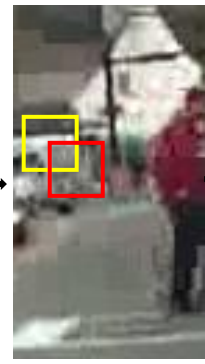
N. Dalal and B. Triggs . ***Histograms of Oriented Gradients for Human Detection.*** **CVPR, 2005**

HOG Descriptor = Histogram of Gradients

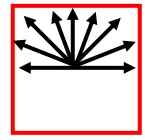
Scan input image



Resize window to 64x128



Divide into overlapping 16x16 blocks

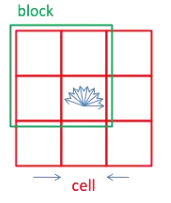


Compute histogram of gradient over 9 directions

Final feature vector ***f*** = 7x15x9 = 945

Histogram of Gradients

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

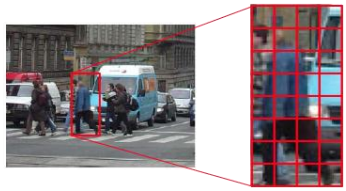


9 orientation bins
0 - 180° degrees

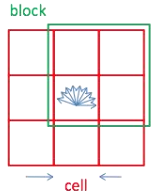
normalize

9x4 feature vector per cell

Histogram of Gradients



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



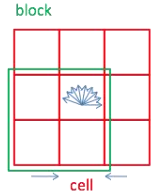
9 orientation bins
0 - 180° degrees

normalize

9x4 feature vector per cell

Histogram of Gradients

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

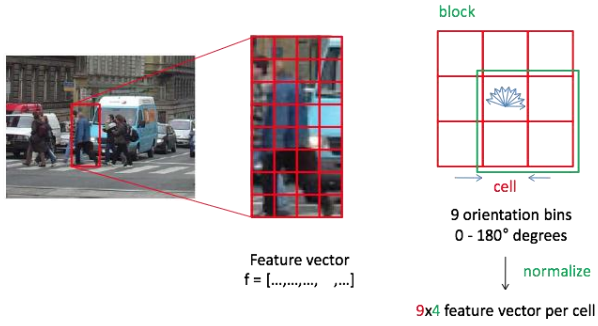


9 orientation bins
0 - 180° degrees

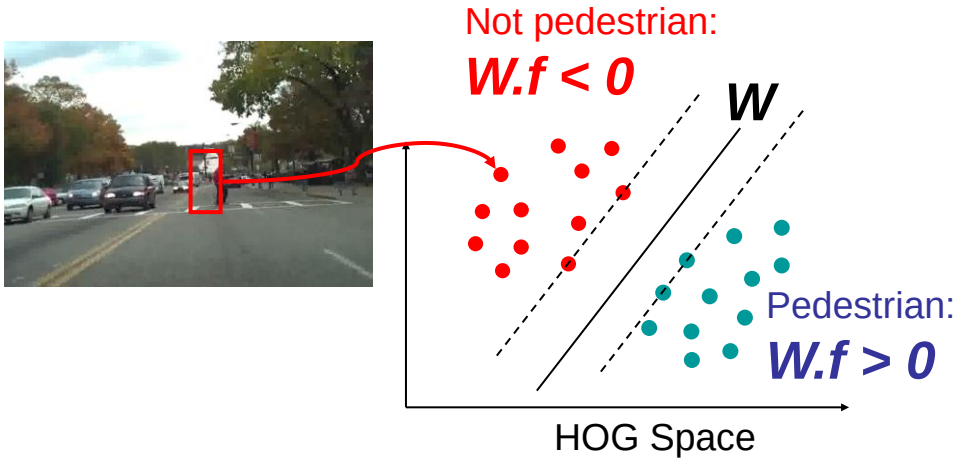
normalize

9x4 feature vector per cell

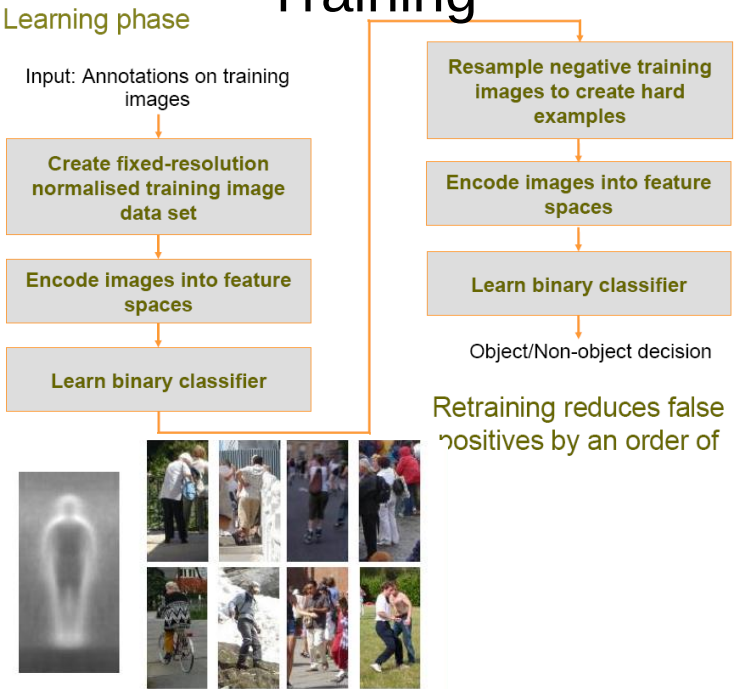
Histogram of Gradients



HOG + Linear SVM



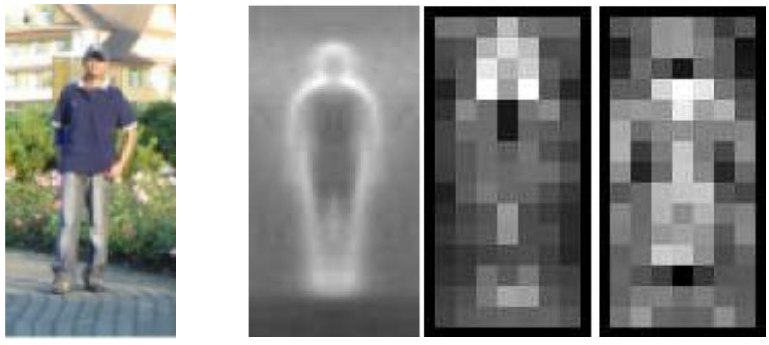
Training



Histogram of Oriented Gradients (HOG)

- ***f*** = vector of gradient histograms
- Input detection window is classified as pedestrian if:

$$\boldsymbol{w}.\boldsymbol{f} > 0$$



Example detection window

Average gradients over the training data

Positive components: Pedestrian class

Negative components: Non-pedestrian class

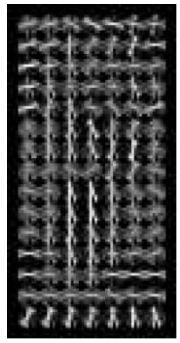
N. Dalal and B. Triggs . ***Histograms of Oriented Gradients for Human Detection.*** **CVPR, 2005**

Histogram of Oriented Gradients (HOG)

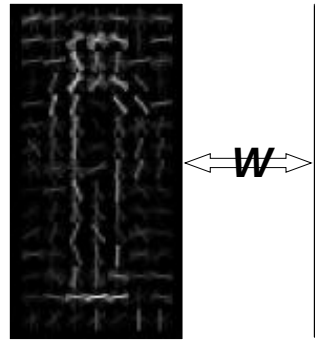
- ***f*** = vector of gradient histograms
- Input detection window is classified as pedestrian if:

$$\boldsymbol{w}.\boldsymbol{f} > 0$$

Example detection window



HOG descriptor



Positive components: Pedestrian class

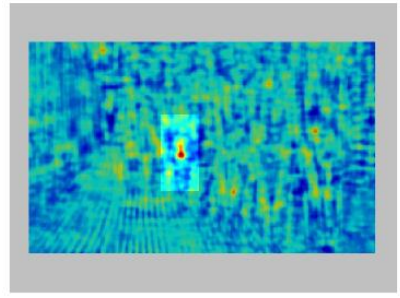
Negative components: Non-pedestrian class

N. Dalal and B. Triggs . ***Histograms of Oriented Gradients for Human Detection.*** **CVPR, 2005**

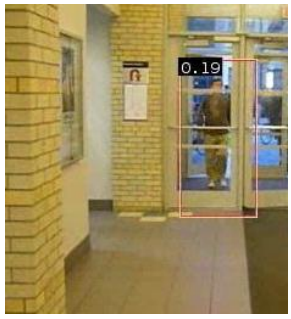
Finding the actual detections

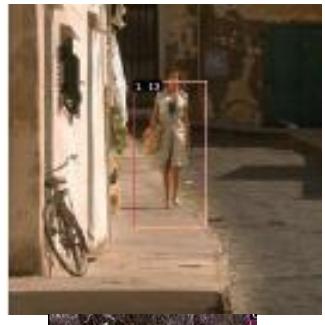
Input image + final detection

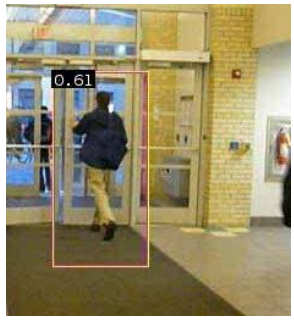
Detection confidence image: Need to find the local maxima and suppress the non-local maxima



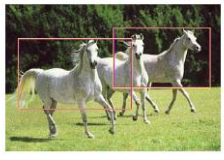
Examples





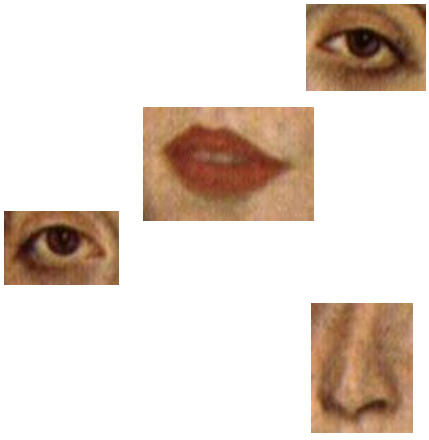


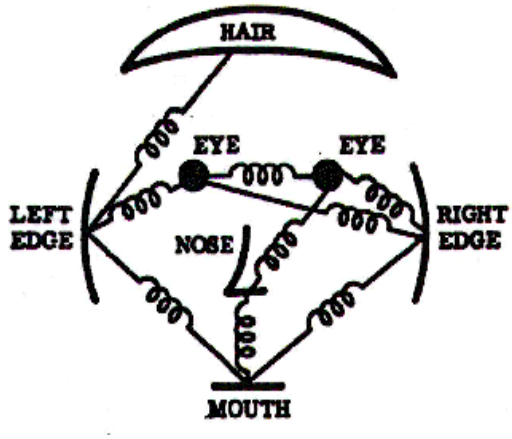
Other objects



Dealing with deformable objects

Parts have similar appearance
Parts appear at similar relative locations on instances of the object category



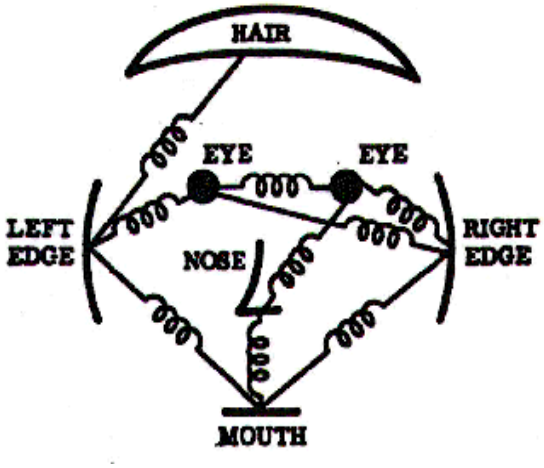


Is this a face?

[Fischler & Elschlager 73]

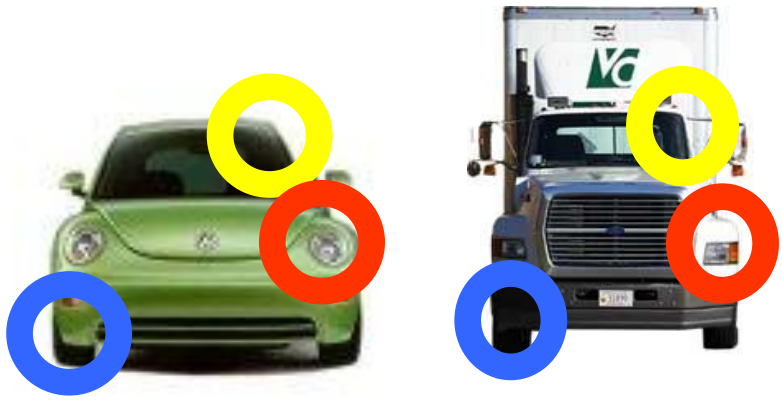
Bags of words and related representations

Note: Classical paper



Martin Fischler and Robert Eschlager. The Representation and Matching of Pictorial Structures. IEEE Transactions on Computers. 1973.

Parts have similar appearance
Parts appear at similar relative locations on instances of the object category

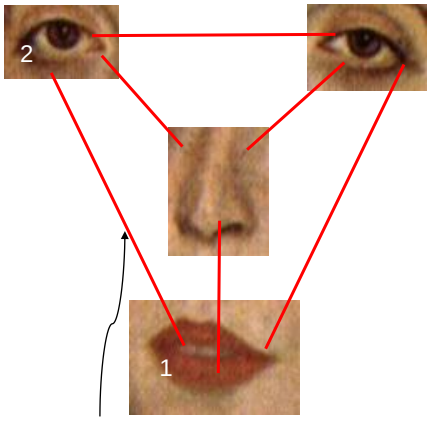


- *M* parts: $p_i = (a_i, l_i)$ ai = appearance l_i = location
- a_i = 128 SIFT vector or local patch…
- $l_i = x, y, \theta$ *N* possible locations

- $L = \{l_1, .., l_M\}$ N^M possible combinations of locations
- Most likely location *L* is found by maximizing:

$$P(L|I) \propto P(I|L)P(L)$$

- $P(I|L)$: How likely is it to observe image I given that the *P* parts are at locations *L*
- Evaluated by comparing the model of each part a_i with the image content at l_i
- $P(L)$: spatial prior controls the geometric configuration of the parts. How to represent $P(L)$??

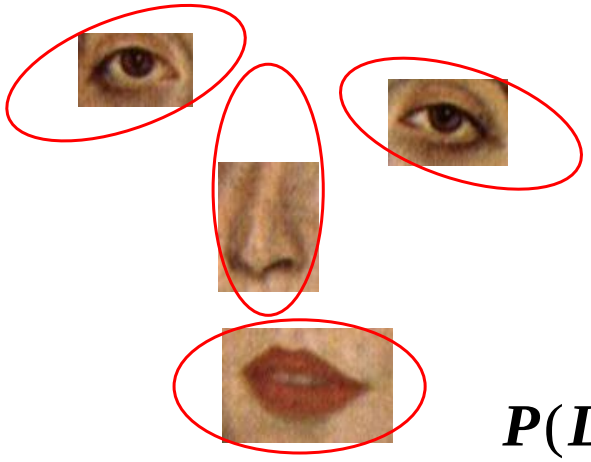


Edge means that the position of part 1 depends on the position of part 2

- In principle, the position of every part depend on the positions of all the other parts
- Combinatorial issue: $O(N^M)$
- Tractability: Factor $P(L)$ into smaller terms

$$P(L) = \prod P(L_j)$$

Naive



Similar to bags of features

Each feature is allowed to move independently

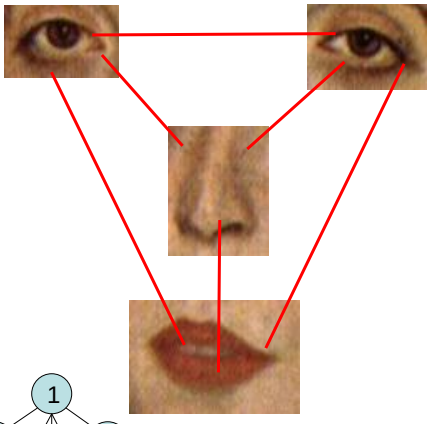
Linear *O(NM)*

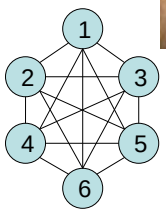
Poor model: Does not model the relative location of parts at all

$$P(L) = \prod_{1}^{N} P(l_i)$$

Csurka '04
Vasconcelos '00

Fully connected: Constellation





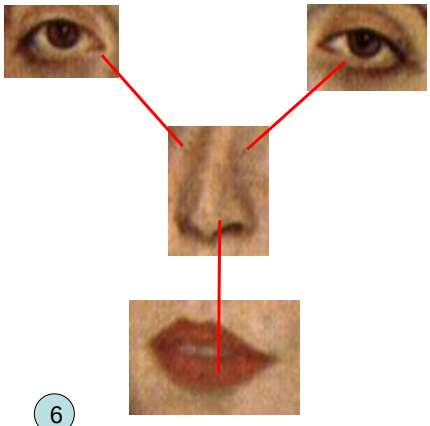
Represent explicitly the joint distribution of locations

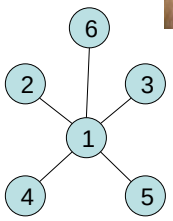
Joint distribution of *P* parts is in principle $O(N^M)$

Good model: Does model the relative location of parts but intractable for moderate number of parts….

$$P(L) = P(l_1, \ldots, l_N)$$

Intermediate: Star (1-Fan)



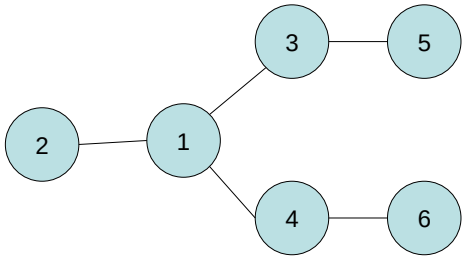


- Represent the location of all the parts relative to a *single* reference part
- Complexity reduced to $O(M^2N)$
- Ok model: Assumes that one reference part is defined.

$$P(L) = P(l_R)\prod P(l_i \mid l_R)$$

Reference part

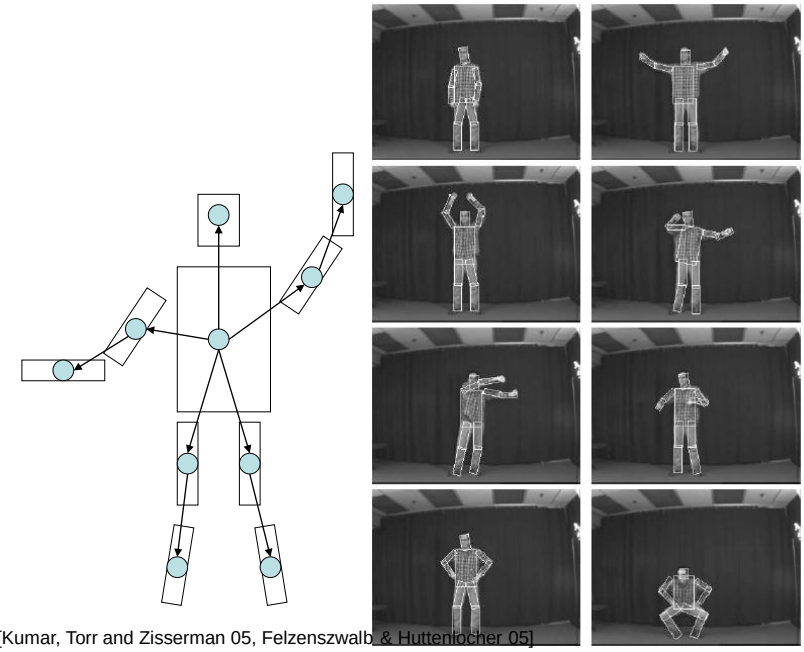
Tree structure



- No cycles in the graph
- O(MN^2)
- Pictorial structures

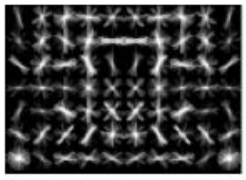
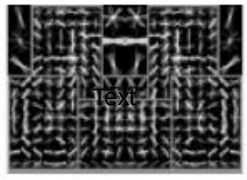
$$P(L) = \prod P(l_i \mid parent(l_i))$$

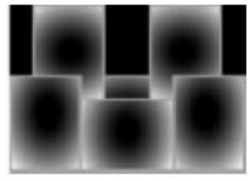
Tree example: Pictorial structures



Images from [Kumar, Torr and Zisserman 05, Felzenszwalb & Huttenlocher 05]

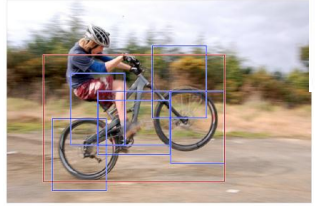
Deformable Part Models (DPM)

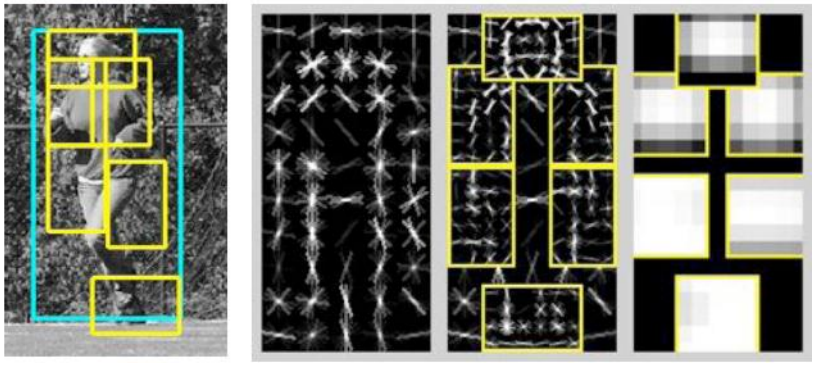


Deformable Objects





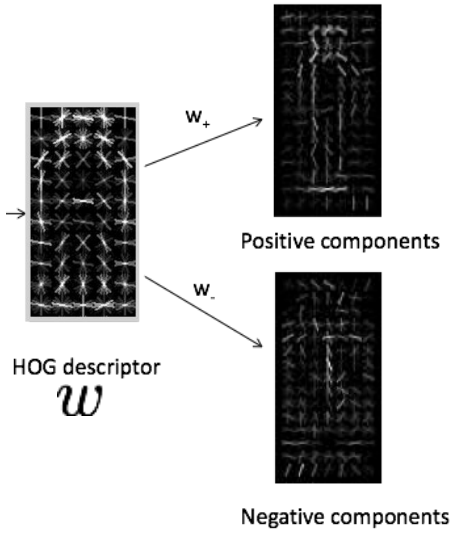

How to deal with geometric deformations?



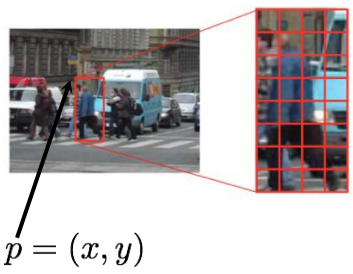
Coarse model | Higher-resolution parts | Model of parts locations

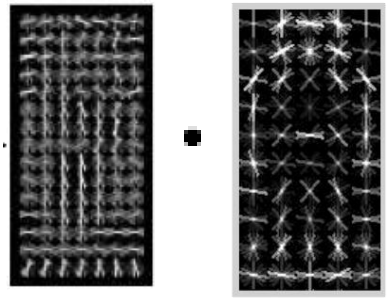
- A Discriminatively Trained, Multiscale, Deformable Part Model with Pedro Felzenszwalb and Deva Ramanan, 2008-2010

Learned SVM weight



Detection





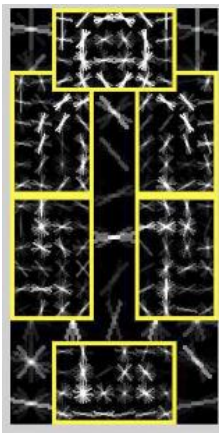
$\phi(p)$ $\cdot$ w

Deformable Objects

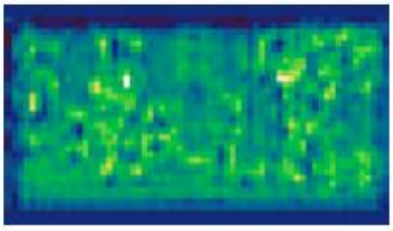
Why is it hard?

- Significant Variability

- Photometric variation
- Viewpoint variation
- Intra-class variation

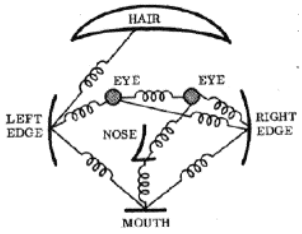


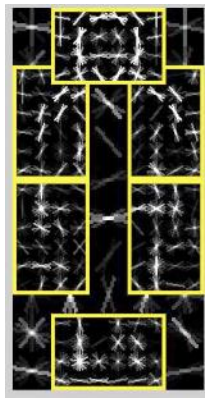
Part Scores



Part Scores

Overview





Scoring a detection

HOG Person Detection

$$\max_p \left(w^T \phi(p)\right)$$

Weight learned by SVM

HOG feature at location p

$$p = (x, y)$$

Note: Next set of 30 slides on deformable parts model adapted from Varun Ramakrishna and Abhinav Shrivastava

Part Configurations

$$p = (p_1, p_2, p_3, \ldots p_N)$$

Configuration

Location of part 1

Location of part 2

Scoring a Configuration

$$score(\mathbf{p}) = \sum_{i=1}^{N} w_i^T \psi(p_i) + \sum_{ij} w_{ij}^T \phi(p_i, p_j)$$

Weight learned by SVM

HOG feature at location $\mathbf{p_i}$

Deformation parameter between part i & j

Quadratic function

Best Scoring Configuration

$$\max_p score(p) = \max_p \left(\sum_{i=1}^{N} w_i^T \psi(p_i) + \sum_{i,j} w_{ij}^T \phi(p_i, p_j) \right)$$

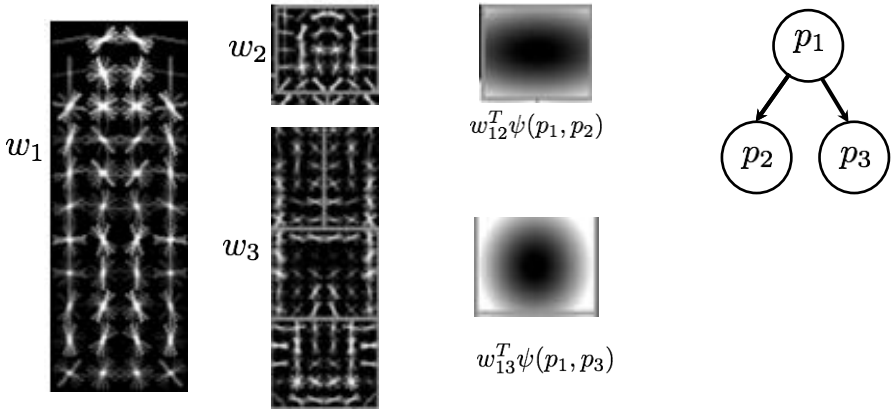
How many configurations?

For a 100x100 image, there are (10^4) possible locations for each part
No of configurations = (10^4)^N

The trick

$$\max(a+b, a+c) = a + max(b, c)$$

Three Part Example



Three Part Example

HOG feature for part 1

HOG feature for part 2

$$score(\mathbf{p}) = w_1^T\psi(p_1) + w_2^T\psi(p_2) + w_3^T\psi(p_3)$$
$$w_{12}^T\phi(p_1,p_2) + w_{13}^T\phi(p_1,p_3)$$

Deformation score between part 1 & 2

Deformation score between part 1 & 3

$$\max_{\mathbf{p}} score(\mathbf{p}) = \max_{p_1}\max_{p_2}\max_{p_3}(w_1^T\psi(p_1) + w_2^T\psi(p_2) + w_3^T\psi(p_3)$$
$$w_{12}^T\phi(p_1,p_2) + w_{13}^T\phi(p_1,p_3))$$

$$\max_{\mathbf{p}} score(\mathbf{p}) = \max_{p_1} \max_{p_2} \max_{p_3} (w_1^T \psi(p_1) + w_2^T \psi(p_2) + w_3^T \psi(p_3) \\ w_{12}^T \phi(p_1, p_2) + w_{13}^T \phi(p_1, p_3))$$

$$\max(a + b, a + c) = a + max(b, c)$$

$$\max_{\mathbf{p}} \texttt{score}(\mathbf{p}) = \max_{p_1} \left(w_1^T \phi(p_1) + \max_{p_2} \left(w_2^T \phi(p_2) + w_{12}^T \psi(p_1, p_2) \right) + \max_{p_3} \left(w_3^T \phi(p_3) + w_{13}^T \psi(p_1, p_3) \right) \right)$$

Three Part Example

$$\max_{\mathbf{p}} \texttt{score}(\mathbf{p}) = \max_{p_1} \left(w_1^T \phi(p_1) + \max_{p_2} \left(w_2^T \phi(p_2) + w_{12}^T \psi(p_1, p_2) \right) + \max_{p_3} \left(w_3^T \phi(p_3) + w_{13}^T \psi(p_1, p_3) \right) \right)$$

Three Part Example

$$\max_{\mathbf{p}} \texttt{score}(\mathbf{p}) = \max_{p_1} \left(w_1^T \phi(p_1) + \max_{p_2} \left(w_2^T \phi(p_2) + w_{12}^T \psi(p_1, p_2) \right) + \max_{p_3} \left(w_3^T \phi(p_3) + w_{13}^T \psi(p_1, p_3) \right) \right)$$

Three Part Example

$$\max_{\mathbf{p}} \texttt{score}(\mathbf{p}) = \max_{p_1} \left(w_1^T \phi(p_1) + \max_{p_2} \left(w_2^T \phi(p_2) + w_{12}^T \psi(p_1, p_2) \right) + \max_{p_3} \left(w_3^T \phi(p_3) + w_{13}^T \psi(p_1, p_3) \right) \right)$$

w_2

head filter

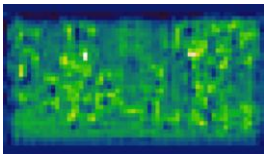
·

Three Part Example

$$\max(a+b, a+c) = a + \max(b, c)$$

$$\max_{\mathbf{p}} \texttt{score}(\mathbf{p}) = \max_{p_1} \left(w_1^T \phi(p_1) + \underbrace{\max_{p_2} \left(w_2^T \phi(p_2) + w_{12}^T \psi(p_1, p_2) \right)}_{w_2} + \max_{p_3} \left(w_3^T \phi(p_3) + w_{13}^T \psi(p_1, p_3) \right) \right)$$

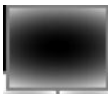
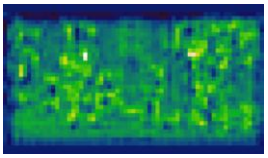
head filter



Three Part Example

$$\max_{\mathbf{p}} \texttt{score}(\mathbf{p}) = \max_{p_1} \left(w_1^T \phi(p_1) + \underbrace{\max_{p_2} \left(w_2^T \phi(p_2) + w_{12}^T \psi(p_1, p_2) \right)}_{w_2} + \max_{p_3} \left(w_3^T \phi(p_3) + w_{13}^T \psi(p_1, p_3) \right) \right)$$

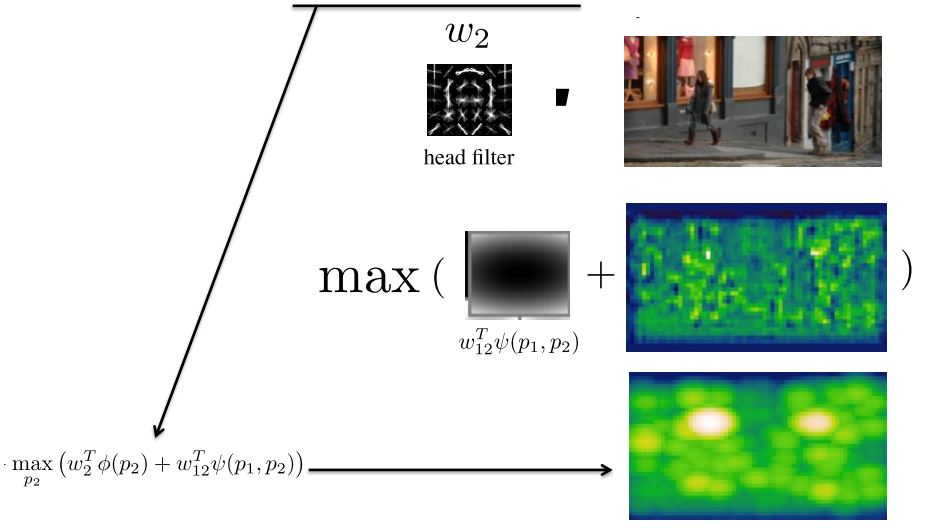
head filter

$\max($  $+$  $)$

$w_{12}^T \psi(p_1, p_2)$

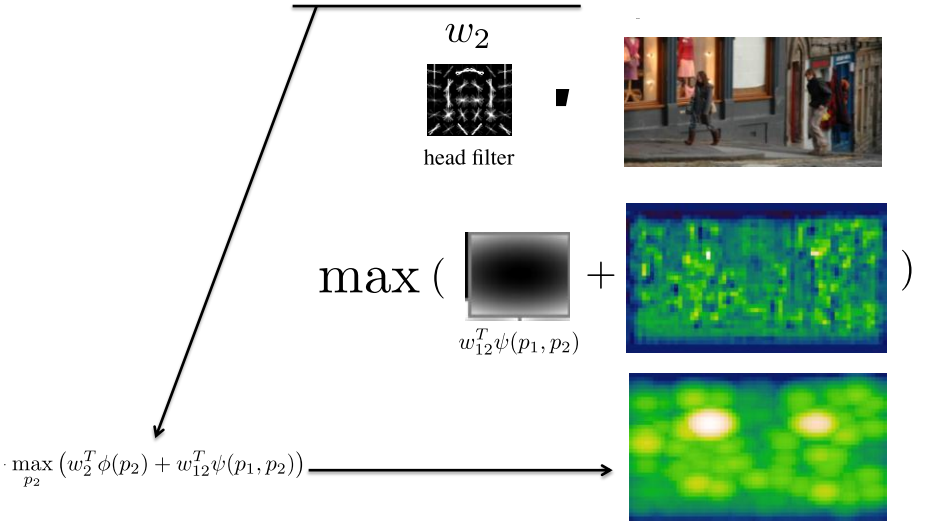
Three Part Example

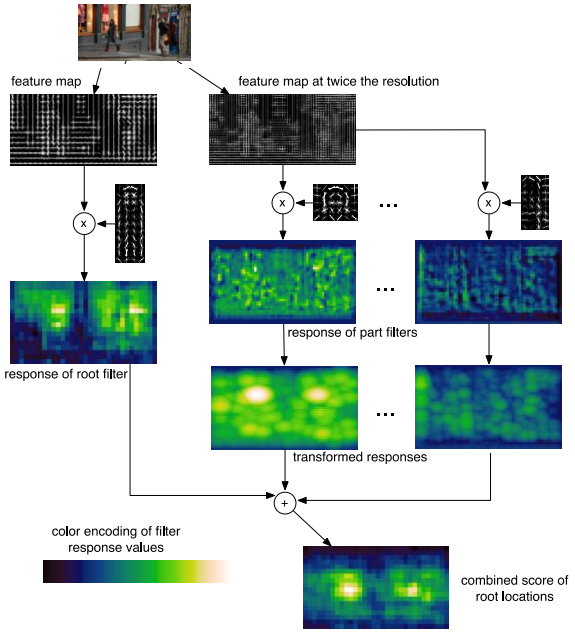
$$\max_{\mathbf{p}} \texttt{score}(\mathbf{p}) = \max_{p_1} \left(w_1^T \phi(p_1) + \max_{p_2} \left(w_2^T \phi(p_2) + w_{12}^T \psi(p_1, p_2) \right) + \max_{p_3} \left(w_3^T \phi(p_3) + w_{13}^T \psi(p_1, p_3) \right) \right)$$



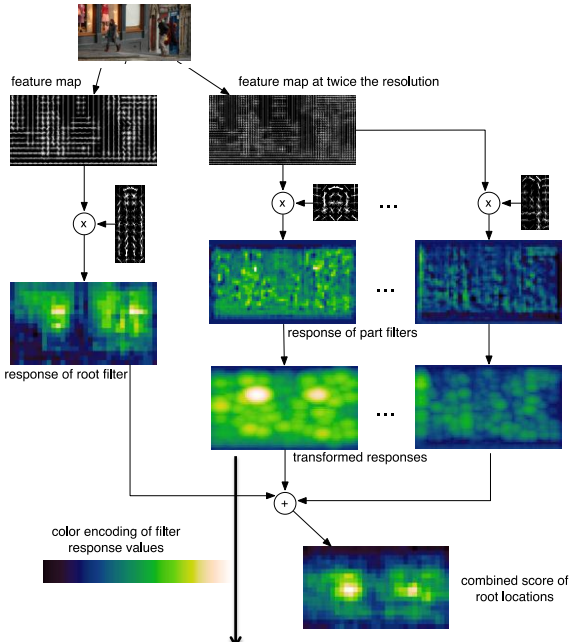
Three Part Example

$$\max_{\mathbf{p}} \texttt{score}(\mathbf{p}) = \max_{p_1} \left(w_1^T \phi(p_1) + \max_{p_2} \left(w_2^T \phi(p_2) + w_{12}^T \psi(p_1, p_2) \right) + \max_{p_3} \left(w_3^T \phi(p_3) + w_{13}^T \psi(p_1, p_3) \right) \right)$$

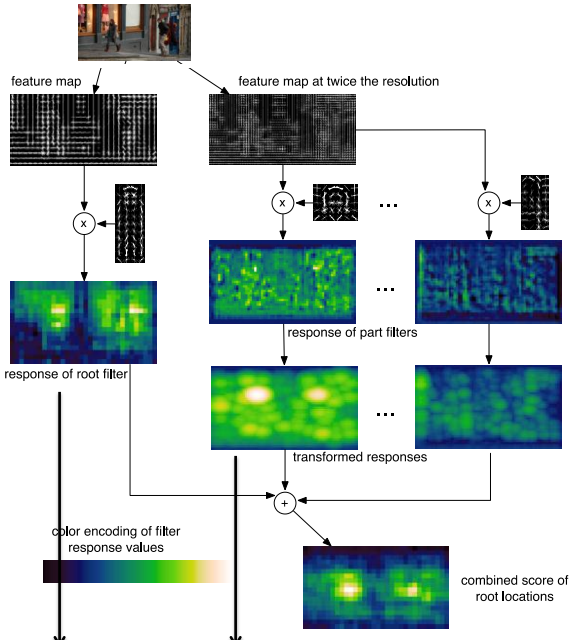




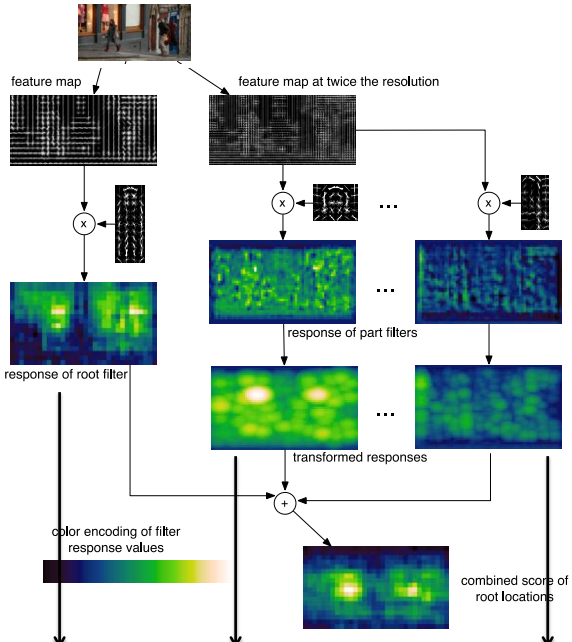
$$\max_{\mathbf{p}} \texttt{score}(\mathbf{p}) = \max_{p_1} \left(w_1^T \phi(p_1) + \max_{p_2} \left(w_2^T \phi(p_2) + w_{12}^T \psi(p_1, p_2) \right) + \max_{p_3} \left(w_3^T \phi(p_3) + w_{13}^T \psi(p_1, p_3) \right) \right)$$



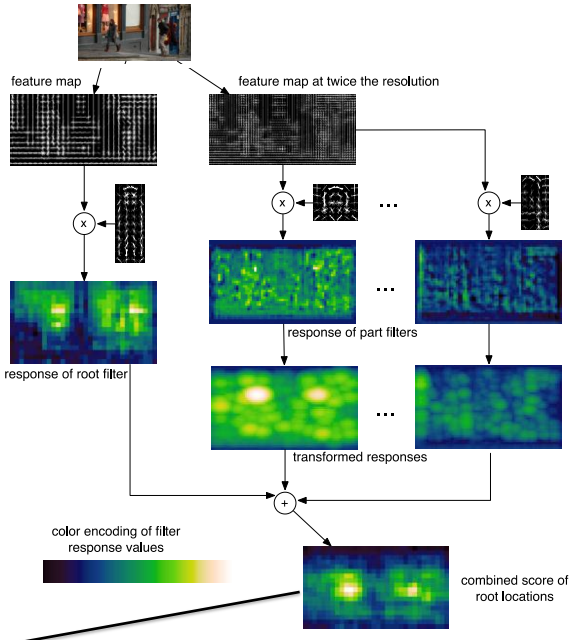
$$\max_{\mathbf{p}} \texttt{score}(\mathbf{p}) = \max_{p_1} \left(w_1^T \phi(p_1) + \max_{p_2} \left(w_2^T \phi(p_2) + w_{12}^T \psi(p_1, p_2) \right) + \max_{p_3} \left(w_3^T \phi(p_3) + w_{13}^T \psi(p_1, p_3) \right) \right)$$



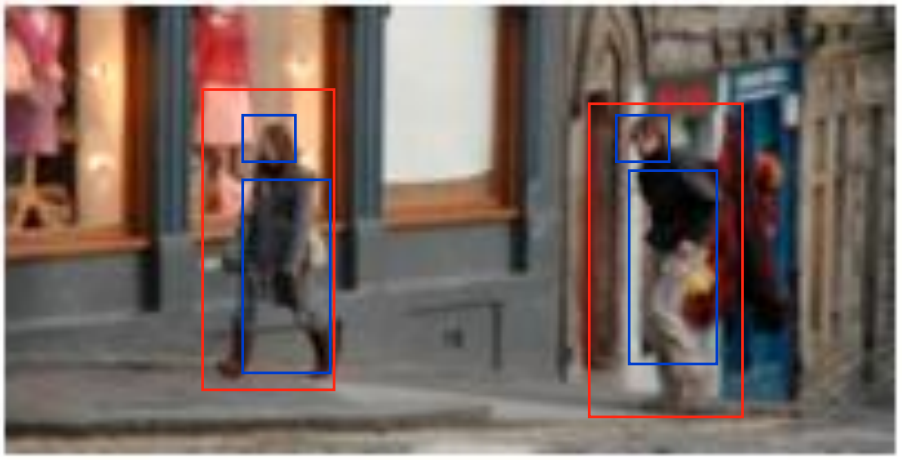
$$\max_{\mathbf{p}} \texttt{score}(\mathbf{p}) = \max_{p_1} \left(w_1^T \phi(p_1) + \max_{p_2} \left(w_2^T \phi(p_2) + w_{12}^T \psi(p_1, p_2) \right) + \max_{p_3} \left(w_3^T \phi(p_3) + w_{13}^T \psi(p_1, p_3) \right) \right)$$



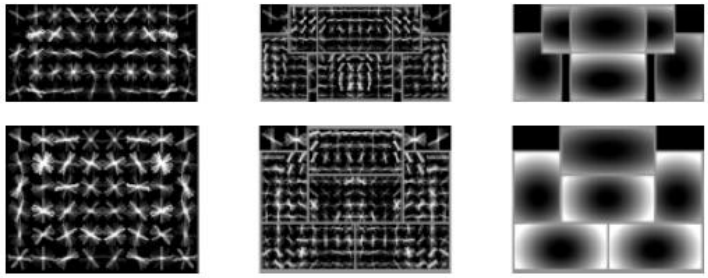
$$\max_{\mathbf{p}} \texttt{score}(\mathbf{p}) = \max_{p_1} \left(w_1^T \phi(p_1) + \max_{p_2} \left(w_2^T \phi(p_2) + w_{12}^T \psi(p_1, p_2) \right) + \max_{p_3} \left(w_3^T \phi(p_3) + w_{13}^T \psi(p_1, p_3) \right) \right)$$



$$\max_{\mathbf{p}} \texttt{score}(\mathbf{p}) = \max_{p_1} \left(w_1^T \phi(p_1) + \max_{p_2} \left(w_2^T \phi(p_2) + w_{12}^T \psi(p_1, p_2) \right) + \max_{p_3} \left(w_3^T \phi(p_3) + w_{13}^T \psi(p_1, p_3) \right) \right)$$



Car model

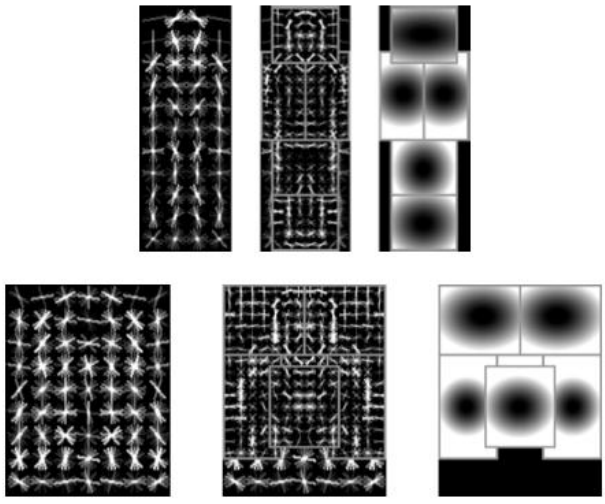


root filters
coarse resolution

part filters
finer resolution

deformation
models

Person model

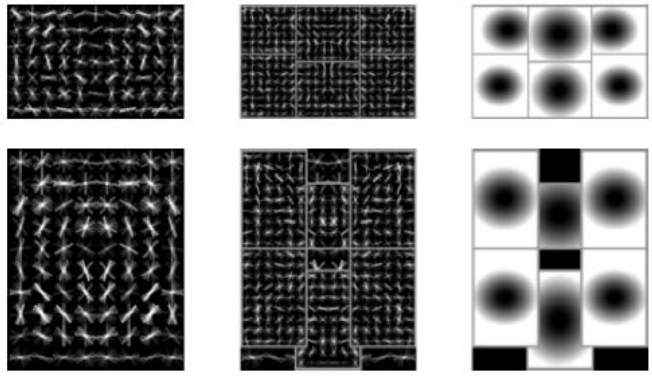


root filters
coarse resolution

part filters
finer resolution

deformation
models

Cat model



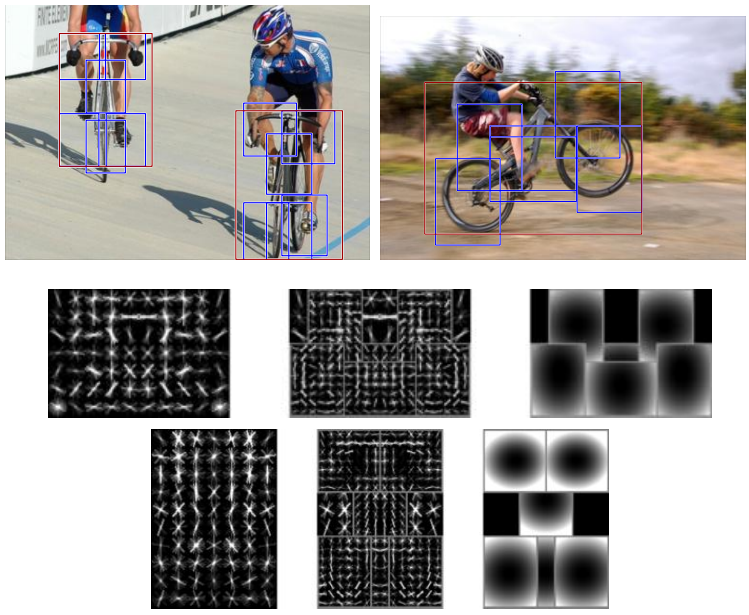
root filters
coarse resolution

part filters
finer resolution

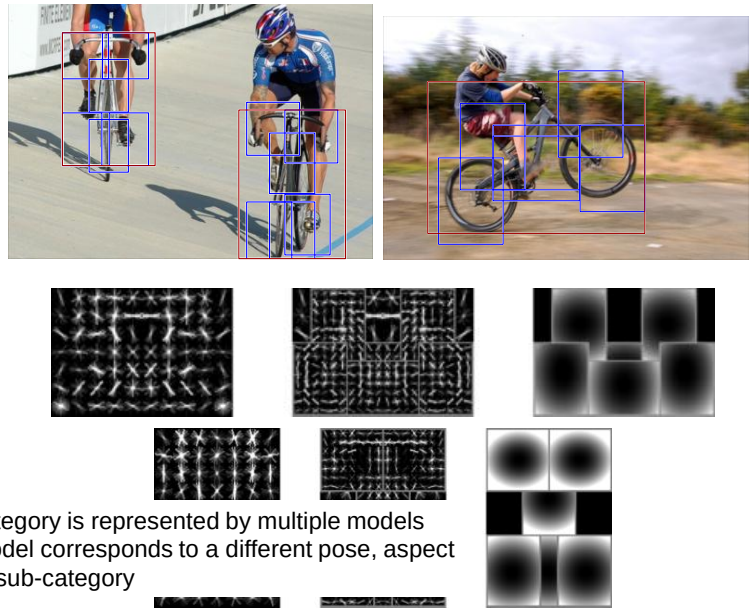
deformation
models

Is One Model Enough?

Mixture Models



Mixture Models



Each category is represented by multiple models
Each model corresponds to a different pose, aspect ratio, or sub-category

Notation Recap…

$$\texttt{score}(\mathbf{p}) = \sum_{i=0}^{N} w_i^T \phi(p_i) + \sum_{ij} w_{ij}^T \psi(p_i, p_j)$$

$$\beta = [w_0, w_1, \ldots, w_N, b]$$

Append all learnt weights

Notation Recap…

$$\texttt{score}(\mathbf{p}) = \sum_{i=0}^{N} w_i^T \phi(p_i) + \sum_{ij} w_{ij}^T \psi(p_i, p_j)$$

$$\beta = [w_0, w_1, \ldots, w_N, b]$$

$$\Psi(\mathbf{p}) = [\phi(p_0), \ldots, \phi(p_N), \psi(p_0, p_1), \ldots, \psi(p_0, p_N)]$$

Append HOG features for all parts and deformation features between parts as a vector.

Notation Recap…

$$\texttt{score}(\mathbf{p}) = \sum_{i=0}^{N} w_i^T \phi(p_i) + \sum_{ij} w_{ij}^T \psi(p_i, p_j)$$

$$\beta = [w_0, w_1, \ldots, w_N, b]$$

$$\Psi(\mathbf{p}) = [\phi(p_0), \ldots, \phi(p_N), \psi(p_0, p_1), \ldots, \psi(p_0, p_N)]$$

$$\texttt{score}(\mathbf{p}) = \beta \bullet \Psi(\mathbf{p})$$

Notation Recap…

$$\texttt{score}(\mathbf{p}) = \sum_{i=0}^{N} w_i^T \phi(p_i) + \sum_{ij} w_{ij}^T \psi(p_i, p_j)$$

$$\beta = [w_0, w_1, \ldots, w_N, b]$$

$$\Psi(\mathbf{p}) = [\phi(p_0), \ldots, \phi(p_N), \psi(p_0, p_1), \ldots, \psi(p_0, p_N)]$$

$$\texttt{score}(\mathbf{p}) = \beta \bullet \Psi(\mathbf{p})$$

Train Linear SVM!
Remember Dalal & Triggs and Hard-mining?

Mixture Models

Configuration: $\mathbf{p} = (p_0, p_1, p_2, \ldots, p_N)$

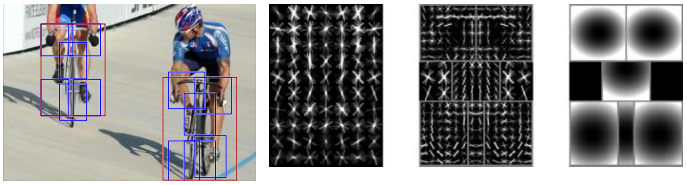
Configuration of a Mixture: $\mathbf{z} = (c, p_0, p_1, p_2, \ldots, p_{N_c})$

Denotes Mixture or Cluster "ID"

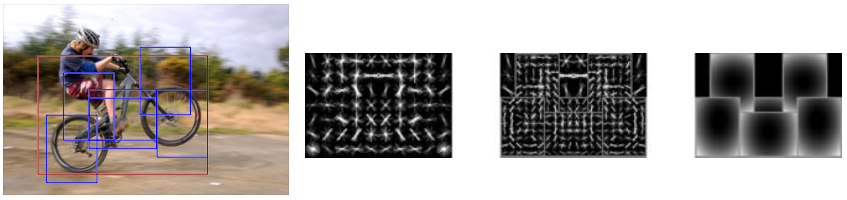
Mixture Models

Configuration of a Mixture: $\mathbf{z} = (c, p_0, p_1, p_2, \ldots, p_{N_c})$

C = 0



C = 1



Mixture Models

$$\beta_i = [w_0, w_1, \ldots, w_N, b]$$

$$\beta = [\beta_0, \beta_1, \ldots, \beta_m]$$

Mixture Models

$$\beta_i = [w_0, w_1, \ldots, w_N, b]$$

$$\beta = [\beta_0, \beta_1, \ldots, \beta_m]$$

$$\Psi(\mathbf{p}) = [\phi(p_0), \ldots, \phi(p_N), \psi(p_0, p_1), \ldots, \psi(p_0, p_N)]$$

$$\Psi(\mathbf{z}) = [0, \ldots, 0, \Psi(\mathbf{p}), 0, \ldots, 0]$$

Mixture Models

$$\beta_i = [w_0, w_1, \ldots, w_N, b]$$

$$\beta = [\beta_0, \beta_1, \ldots, \beta_m]$$

$$\Psi(\mathbf{p}) = [\phi(p_0), \ldots, \phi(p_N), \psi(p_0, p_1), \ldots, \psi(p_0, p_N)]$$

$$\Psi(\mathbf{z}) = [0, \ldots, 0, \Psi(\mathbf{p}), 0, \ldots, 0]$$

$$\texttt{score}(\mathbf{z}) = \max_{\mathbf{z} \in Z} (\beta \bullet \Psi(\mathbf{z}))$$

Mixture Models

$$\texttt{score}(\mathbf{z}) = \max_{\mathbf{z} \in Z} (\beta \bullet \Psi(\mathbf{z}))$$

1. If I knew which sample belongs to which component:
   I could train β using standard training procedure

2. If I knew β:
   I could decide which sample belongs to which component

Membership of samples to classes is *latent*

Solution: Iterate

1. Assign each sample to the component with the maximum score(z)

2. Estimate β using standard training procedure

Training Mixture Models

Relabel positive examples

Optimize beta

Training Mixture Models

Relabel positive examples

- Optimize $\mathtt{L}(\beta, Z_p)$ over Z_p
- Select the highest scoring latent value for each positive example

$$z_i = \arg\max_{z \in Z} \beta \bullet \Psi(I_i, z)$$

Optimize beta

Training Mixture Models

Relabel positive examples

- Optimize $\mathtt{L}(\beta, Z_p)$ over Z_p
- Select the highest scoring latent value for each positive example

$$z_i = \arg\max_{z \in Z} \beta \bullet \Psi(I_i, z)$$

Optimize beta

- Optimize $\mathtt{L}(\beta, Z_p)$ over β
- Standard hard-mining training for $\mathtt{L}(\beta)$

Training Mixture Models

Relabel positive examples

- Optimize $\mathtt{L}(\beta, Z_p)$ over Z_p
- Select the highest scoring latent value for each positive example

$$z_i = \arg\max_{z \in Z} \beta \bullet \Psi(I_i, z)$$

Optimize beta

- Optimize $\mathtt{L}(\beta, Z_p)$ over β
- Standard hard-mining training for $\mathtt{L}(\beta)$

Training Mixture Models

Relabel positive examples

- Optimize $\mathtt{L}(\beta, Z_p)$ over Z_p
- Select the highest scoring latent value for each positive example

$$z_i = \arg\max_{z \in Z} \beta \bullet \Psi(I_i, z)$$

Optimize beta

- Optimize $\mathtt{L}(\beta, Z_p)$ over β
- Standard hard-mining training for $\mathtt{L}(\beta)$

Latent-SVM

- Cluster Positives in K clusters (initialization)

- Train K Mixture Models
- Reassign Positives to K clusters

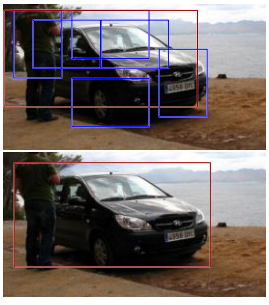
- Cluster Positives in K clusters (initialization)

- Train K Mixture Models
- Reassign Positives to K clusters

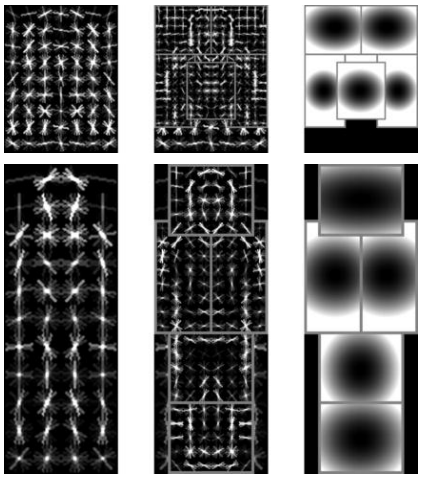
- Cluster Positives in K clusters (initialization)

- Train K Mixture Models
- Reassign Positives to K clusters

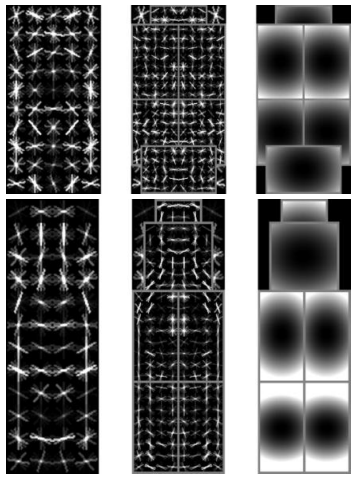
- Optional: Bounding-box Prediction



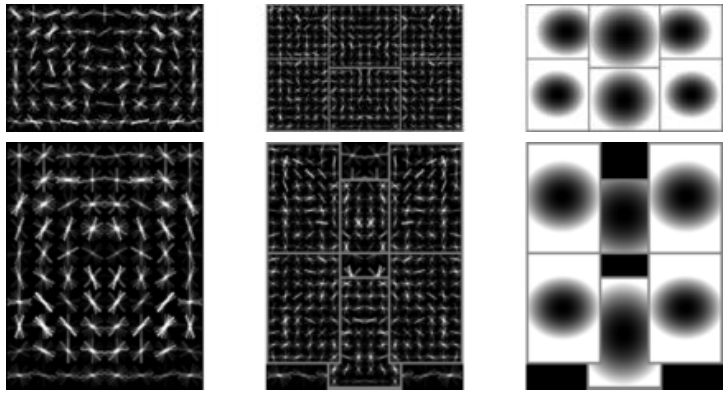
person



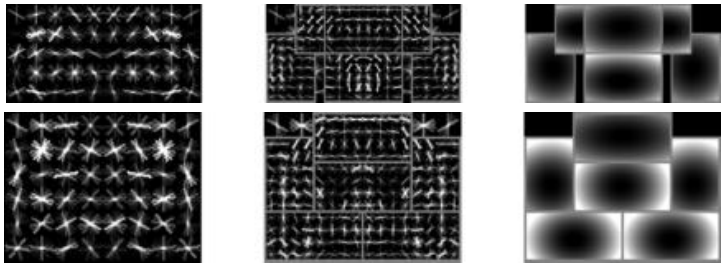
bottle



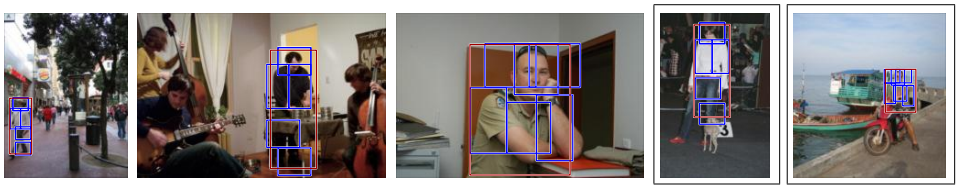
cat



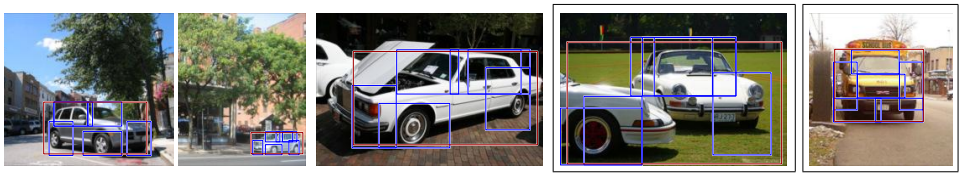
car



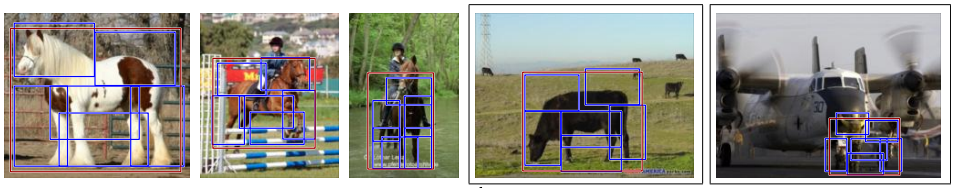
person



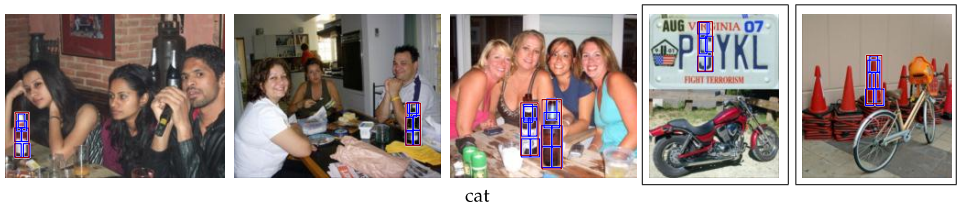
car



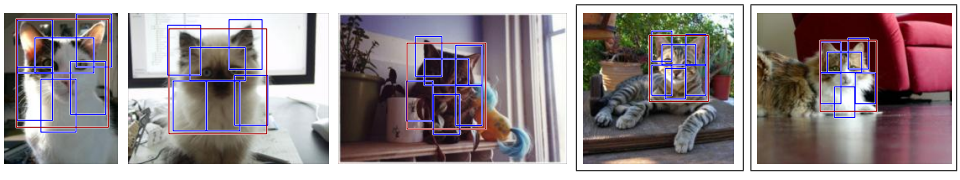
horse

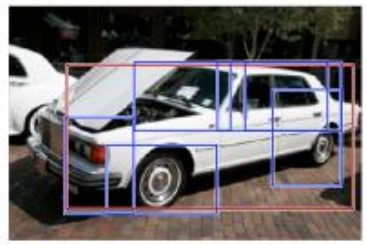


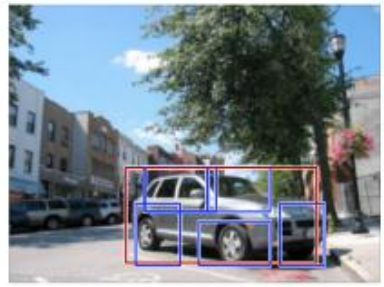
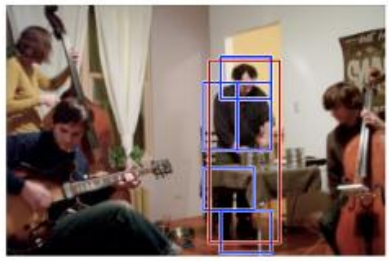
bottle

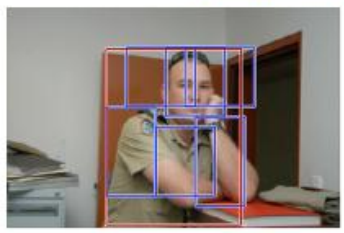


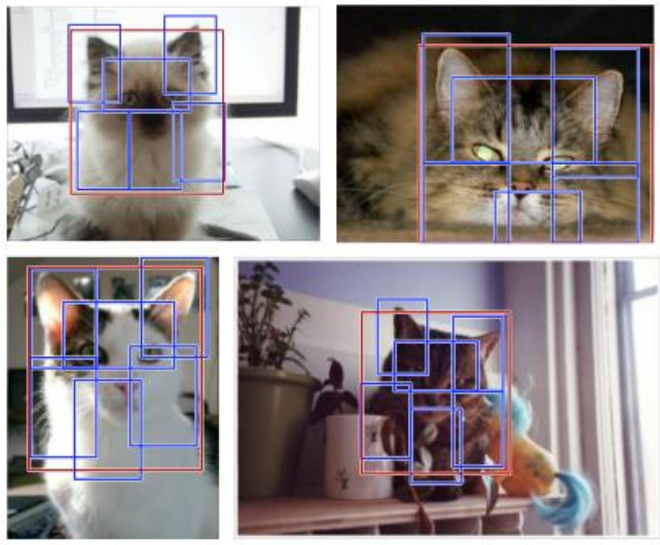
cat



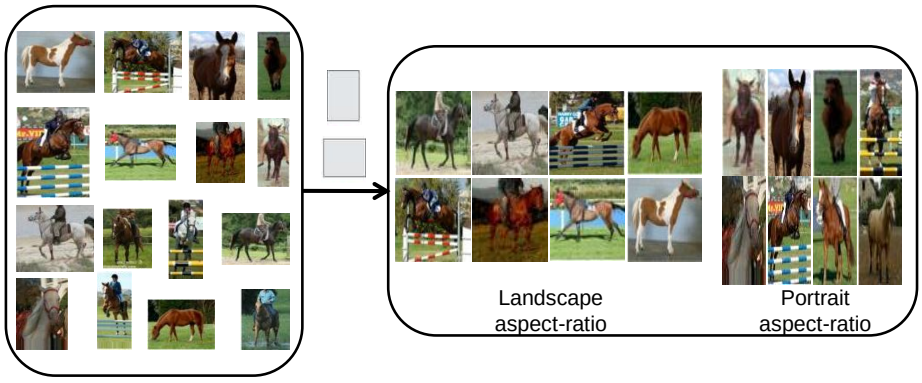


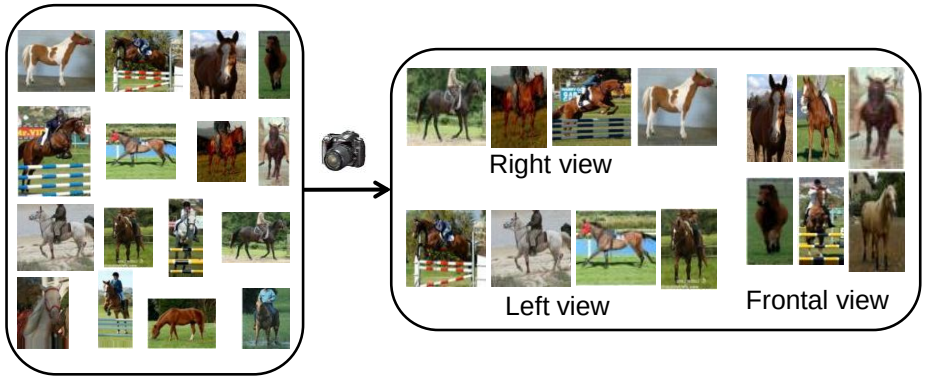
Components

- Many possible sources of division of category into components
- Maybe learned (see above) or generated from other sources

Aspect ratio Components

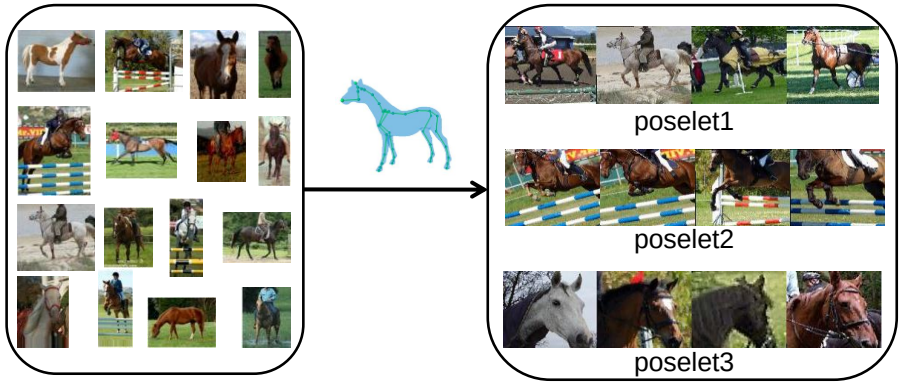


View-point components



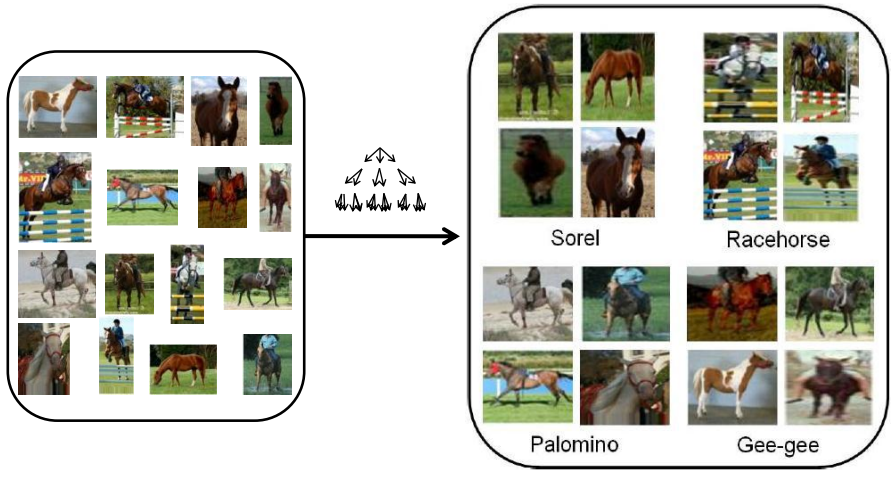
Chum & Zisserman 2007, Harzallah and Schmid 2008

3D configuration components



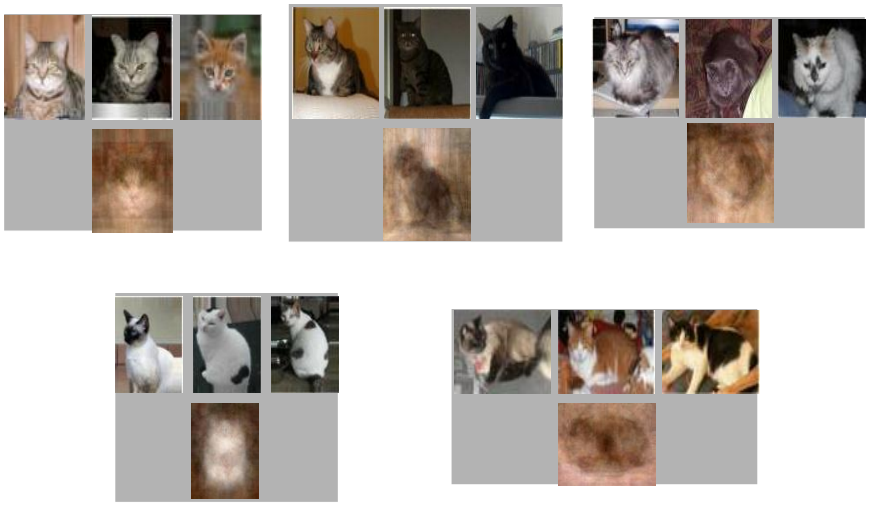
Bourdev & Malik, 2009

Taxonomy components



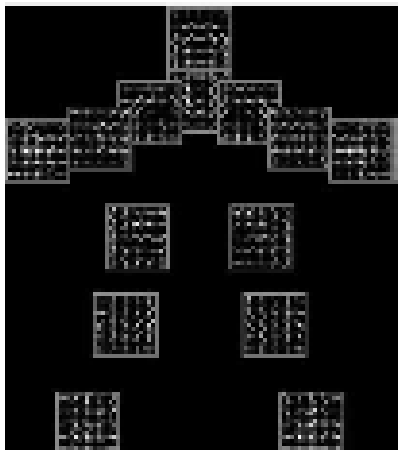
"ImageNet", Deng et al., 2009

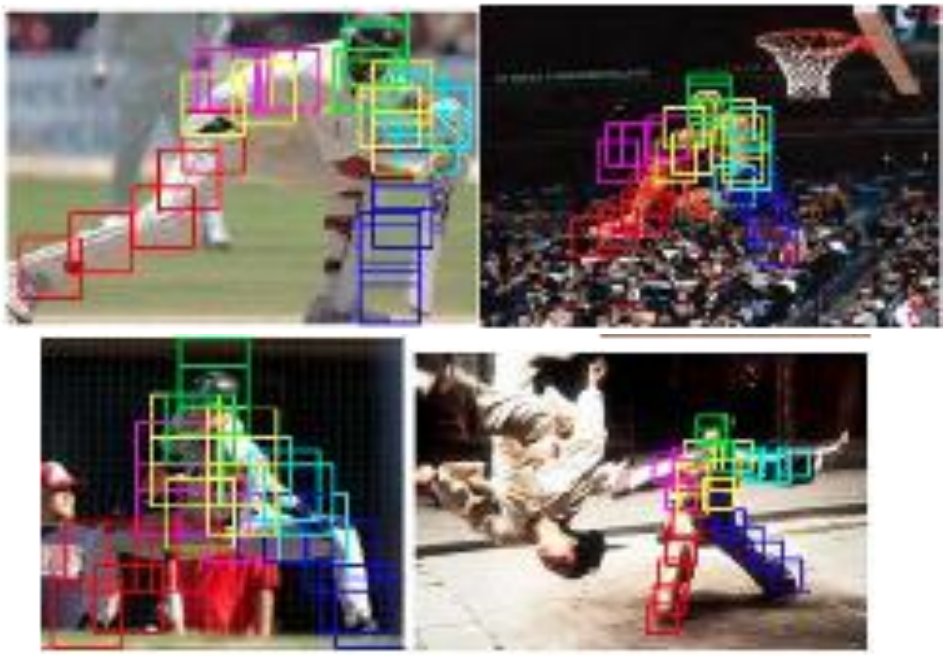
Appearance/Visual components



Divvala et al., 2012

Finer models for articulated human bodies





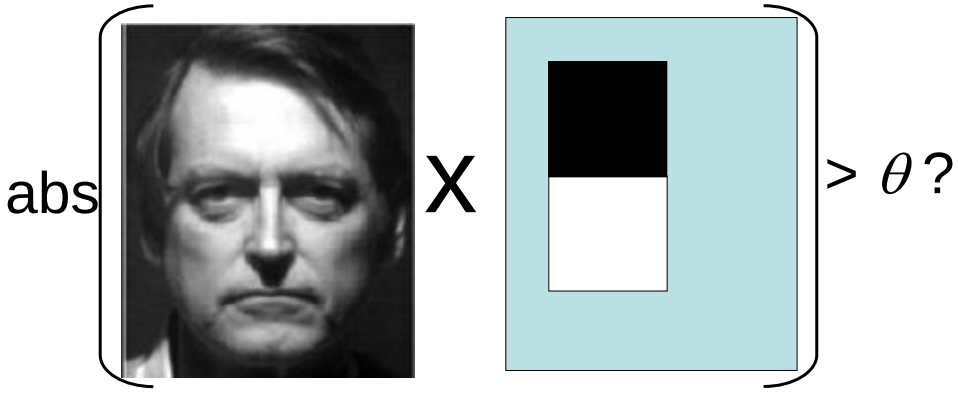
References

- [1] P. Felzenszwalb, D. McAllester, D. Ramaman. A Discriminatively Trained, Multiscale, Deformable Part Model. Proceedings of the IEEE CVPR 2008.
- Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. *CVPR.* 2011
- Slide Credits - Pedro's MLSS tutorial, Ed Hsiao's class presentation in Learning based methods in Vision.

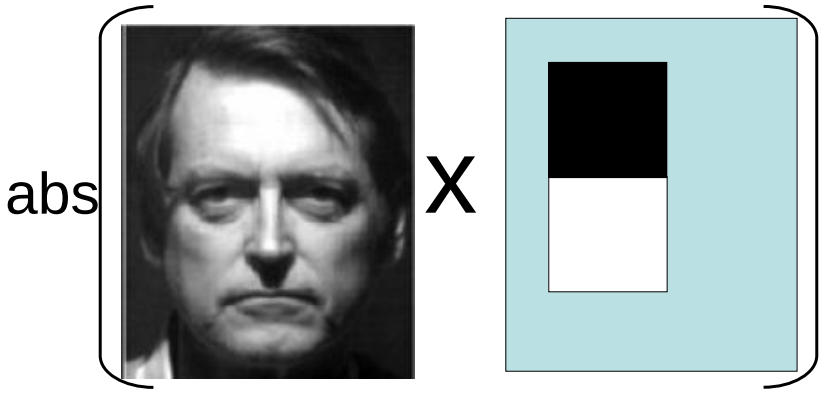
Template classification

- SVMs
- Combination of simple classifiers (boosting)
- Neural networks, deep learning

Using Weak Features



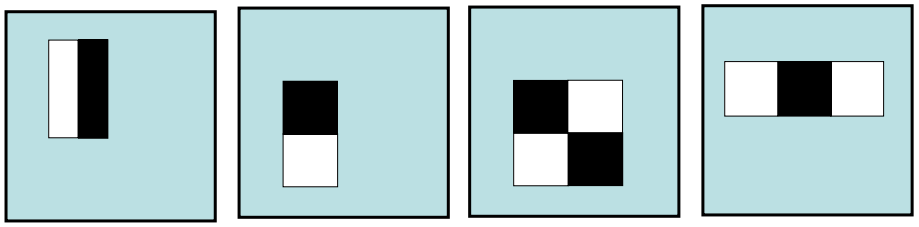
Using Weak Features



- Don't try to design strong features from the beginning, just use really stupid but really fast features (and a lot of them)
- *Weak learner* = Very fast (but very inaccurate) classifier
- Example: Multiply input window by a very simple box operator and threshold output

(Example from Paul Viola, Distributed by Intel as part of the OpenCV library)

Feature Selection



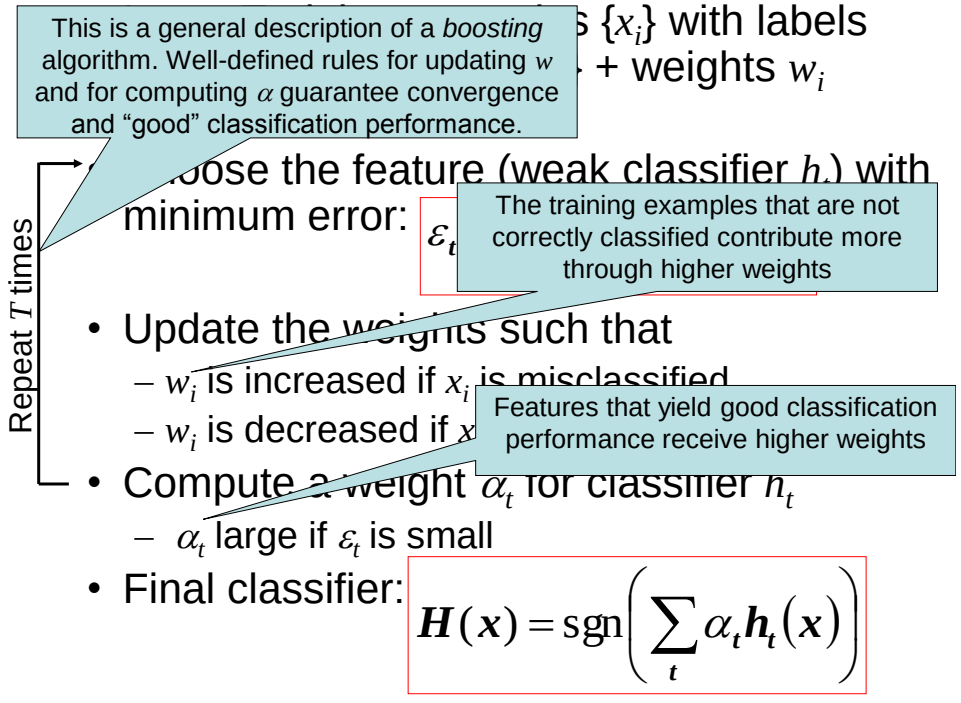
- Operators defined over all possible shapes and positions within the window
- For a 24x24 window → 45,396 combinations!!
- How to select the "useful" features?
- How to combine them into classifiers?

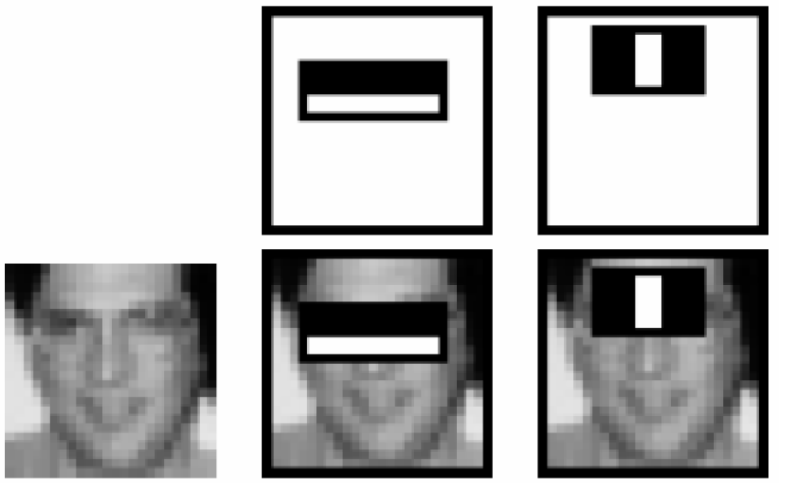
(Example from Paul Viola)

- Input: Training examples $\{x_i\}$ with labels ("face" or "non-face" = +/-1) $\{y_i\}$ + weights w_i (initially $w_i = 1$)
- Choose the feature (weak classifier h_t) with minimum error: $\varepsilon_t = \sum_i w_i [h_t(x_i) \neq y_i]$

Repeat T times

- Update the weights such that
  - w_i is increased if x_i is misclassified
  - w_i is decreased if x_i is correctly classified
- Compute a weight α_t for classifier h_t
  - α_t large if ε_t is small
- Final classifier: $H(x) = \text{sgn}\left(\sum_t \alpha_t h_t(x)\right)$



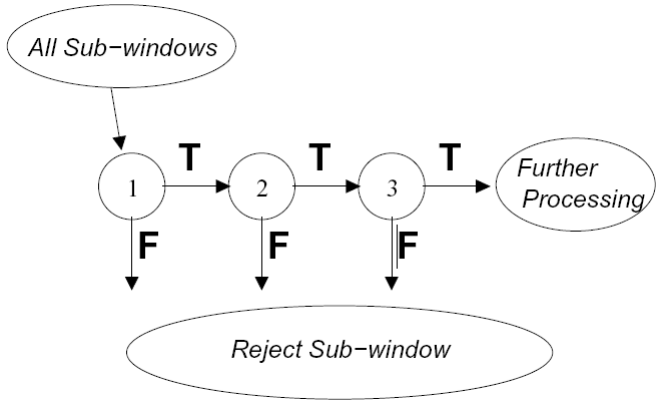


1st feature selected 2nd feature selected

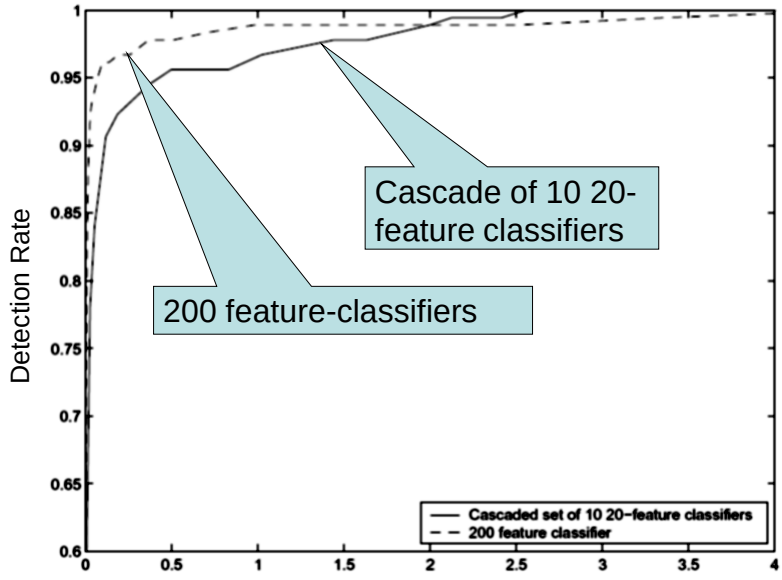
The automatic selection process selects “natural” features

(Example from Paul Viola)

Using a Cascade

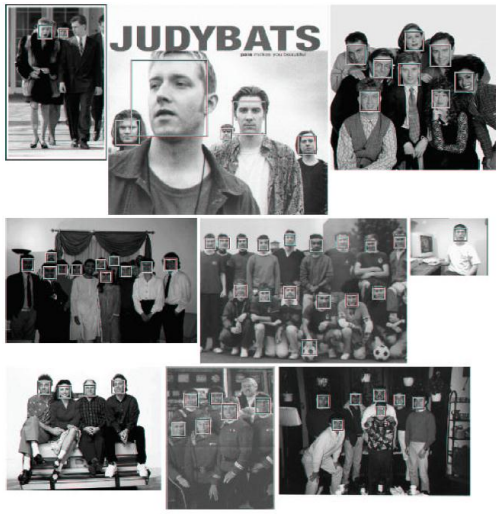


- Same problem as before: It is too hard (or impossible) to build a single accurate classifier
- Key reason: An image containing one face may have 10^5 possible locations but only 1 “correct” location → *rare event detection* → Would require an enormous number of features
- Solution: Use a cascade of classifiers. Each classifier eliminates more of the non-object locations while retaining the “object” locations.

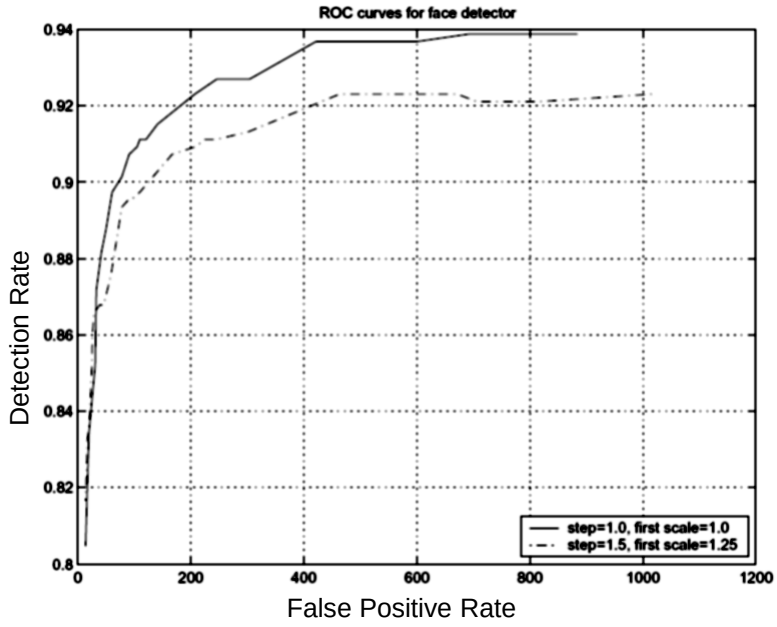


- In this example, the cascade and the single classifier have similar accuracy, but the cascade is 10 times faster

Training: ~5000 face images
+ ~10000 non-face windows



Run-Time: Apply the cascade of classifiers over all positions and scales and return those positions/scales that survive the sequence of classifiers



75,081,800 windows scanned over 130 images (507 faces). (Example from Paul Viola)

Template classification

- SVMs
- Combination of simple classifiers (boosting)
- Neural networks, deep learning