Visual Object Recognition Tutorial

Visual Object Recognition

Bastian Leibe & **Kristen Grauman**

Computer Vision Laboratory
ETH Zurich

Department of Computer Sciences
University of Texas in Austin

Chicago, 14.07.2008

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

THE UNIVERSITY OF TEXAS AT AUSTIN
Department of Computer Sciences

Outline

1. Detection with Global Appearance & Sliding Windows
2. Local Invariant Features: Detection & Description
3. Specific Object Recognition with Local Features

– Coffee Break –

4. Visual Words: Indexing, Bags of Words Categorization
5. Matching Local Features
6. Part-Based Models for Categorization
7. Current Challenges and Research Directions

Detection via classification: Main idea

Basic component: a binary classifier

Car/non-car Classifier

No, not a car.

Yes, car.

Detection via classification: Main idea

If object may be in a cluttered scene, slide a window around looking for it.

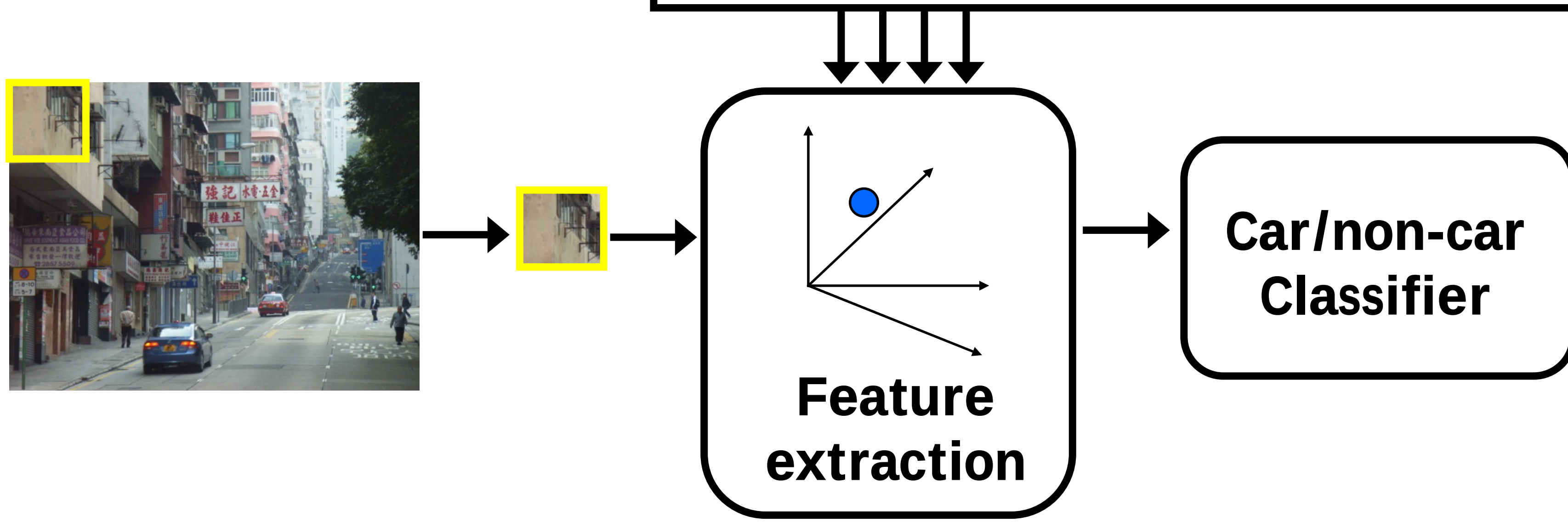
→  →

Car/non-car Classifier

Detection via classification: Main idea

Fleshing out this pipeline a bit more, we need to:

1. **Obtain training data**
2. **Define features**
3. **Define classifier**



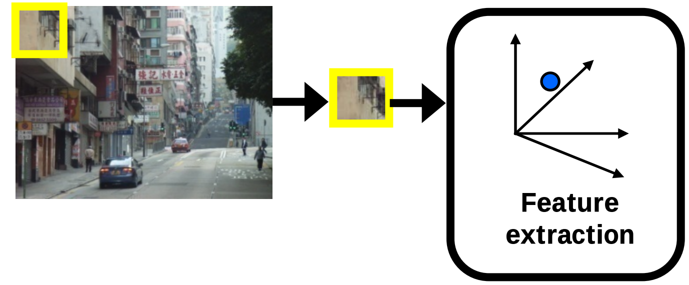
Detection via classification: Main idea

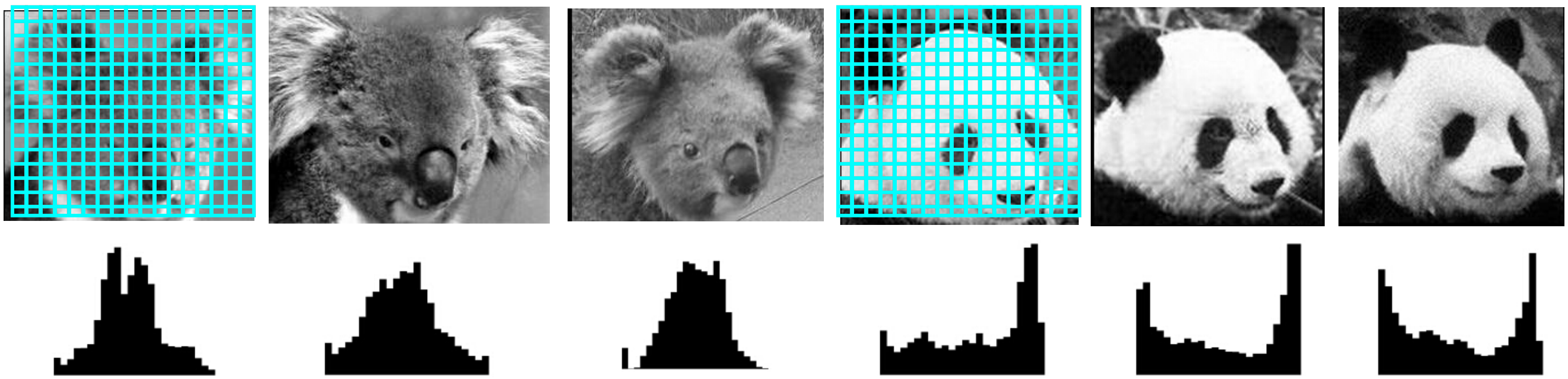
- **Consider all subwindows in an image**
  - **Sample at multiple scales and positions**

- **Make a decision per window:**
  - **“Does this contain object category X or not?”**

- **In this section, we’ll focus specifically on methods using a global representation (i.e., not part-based, not local features).**

Feature extraction: global appearance





Simple holistic descriptions of image content

- grayscale / color histogram
- vector of pixel intensities

K. Grauman, B. Leibe

Eigenfaces: global appearance description

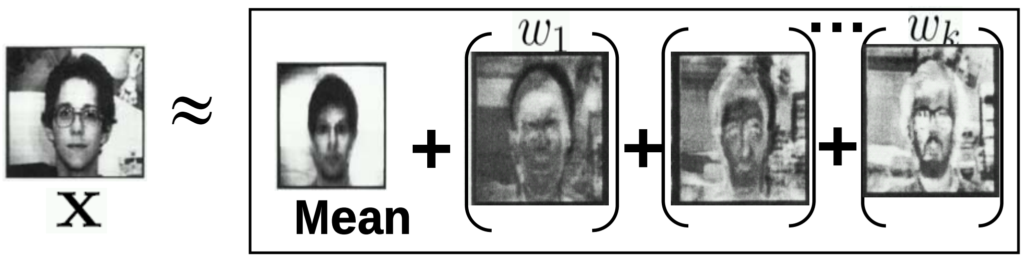
An early appearance-based approach to face recognition

Training images

Mean

Eigenvectors computed from covariance matrix

Generate low-dimensional representation of appearance with a linear subspace.



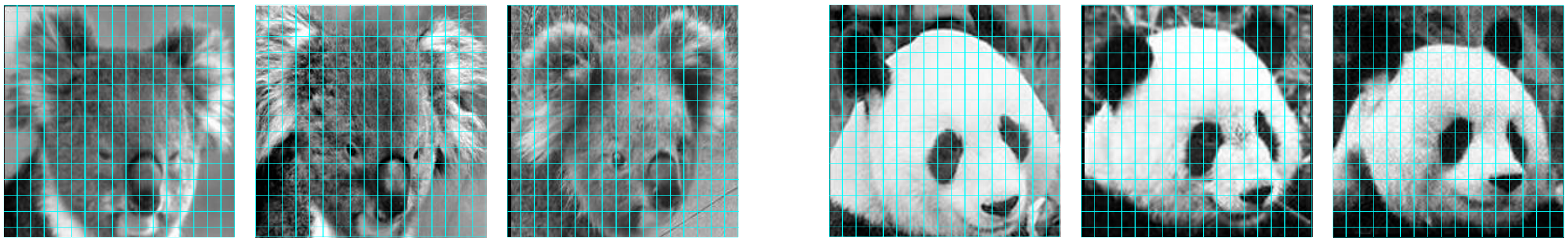
Project new images to "face space".

Recognition via nearest neighbors in face space

Turk & Pentland, 1991

Feature extraction: global appearance

- **Pixel-based representations sensitive to small shifts**

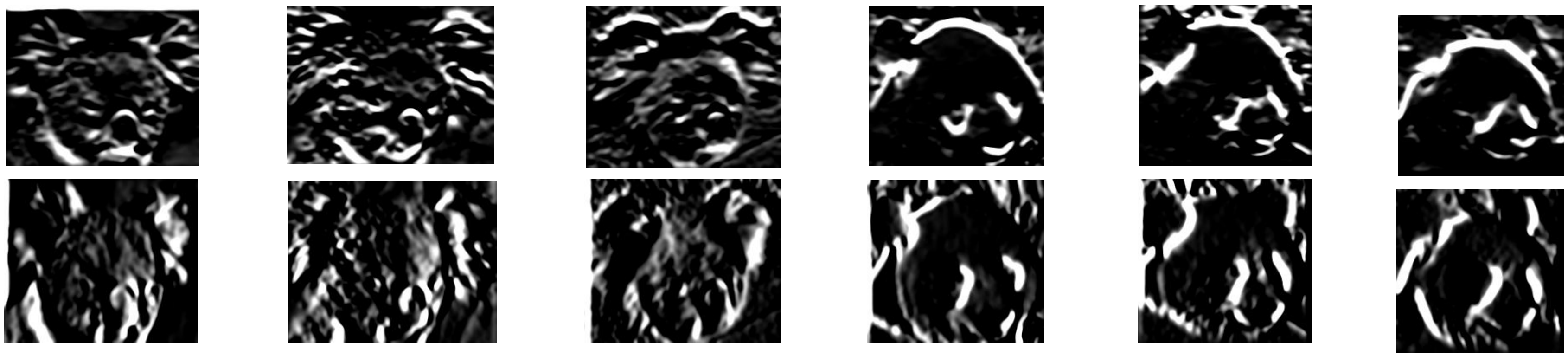


- **Color or grayscale-based appearance description can be sensitive to illumination and intra-class appearance variation**

Cartoon example: an albino koala

Gradient-based representations

- **Consider edges, contours, and (oriented) intensity gradients**

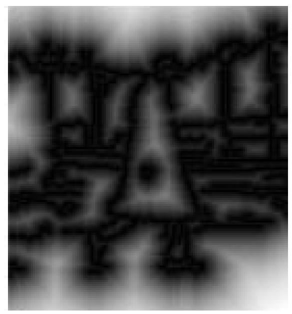


Gradient-based representations: Matching edge templates

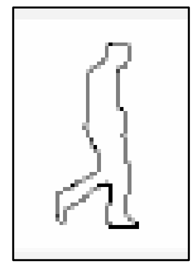
- **Example: Chamfer matching**

Input image

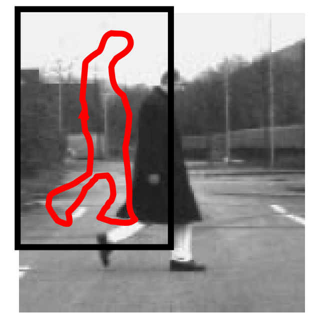
Edges detected



Distance transform



Template shape



Best match

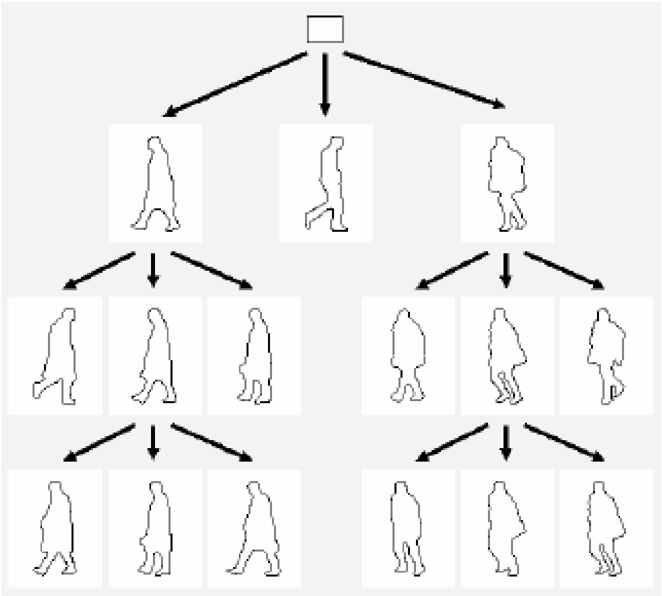
At each window position, compute average min distance between points on template (T) and input (I).

$$D_{chamfer}(T, I) \equiv \frac{1}{|T|} \sum_{t \in T} d_I(t)$$

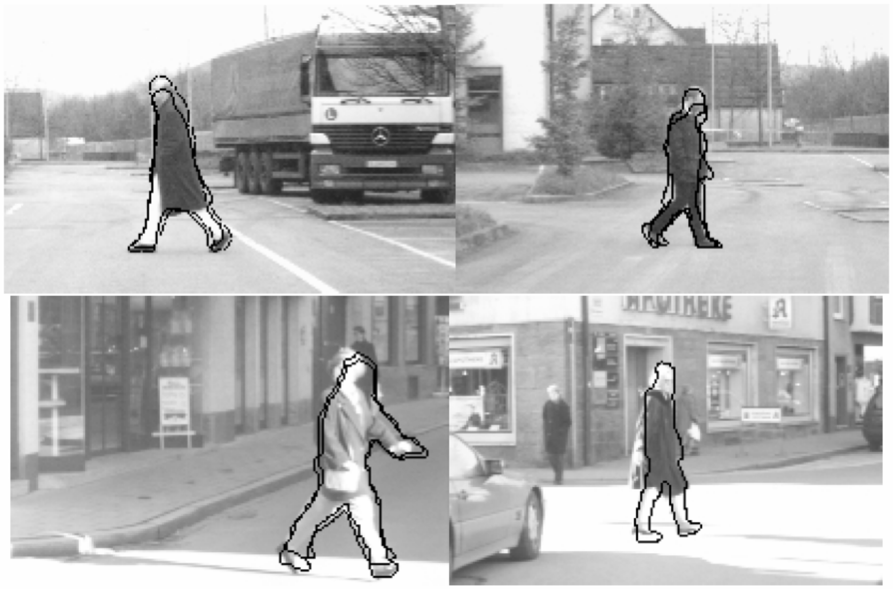
Gavrila & Philomin ICCV 1999

Gradient-based representations: Matching edge templates

- **Chamfer matching**



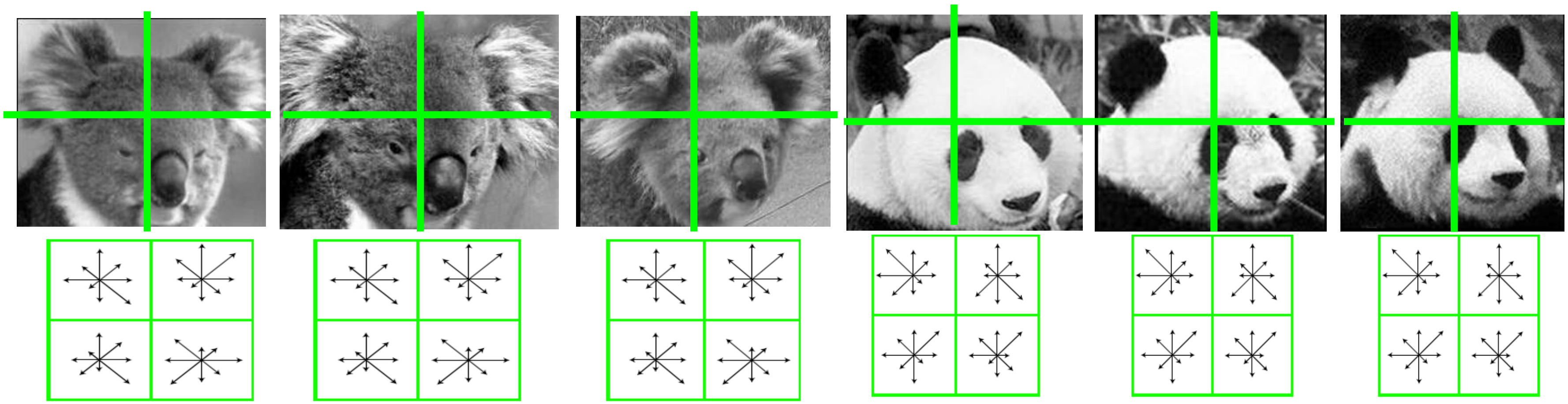
Hierarchy of templates



Gavrila & Philomin ICCV 1999

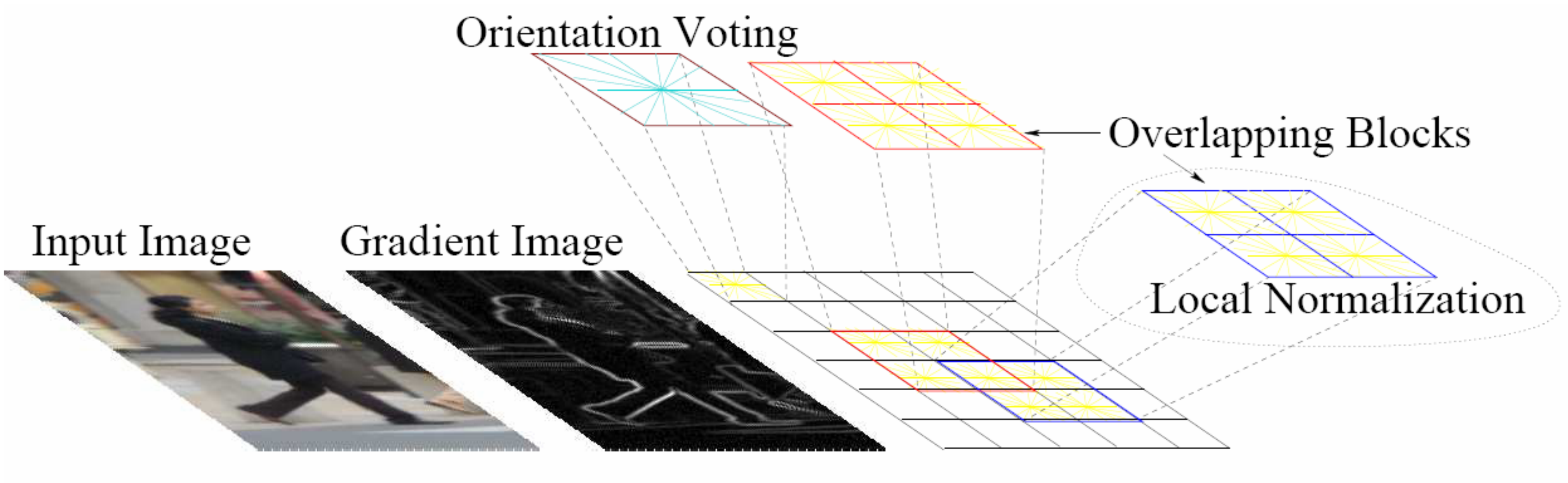
Gradient-based representations

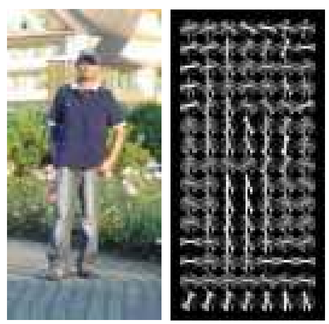
- **Consider edges, contours, and (oriented) intensity gradients**



- **Summarize local distribution of gradients with histogram**
  - **Locally orderless: offers invariance to small shifts and rotations**
  - **Contrast-normalization: try to correct for variable illumination**

Gradient-based representations: Histograms of oriented gradients (HoG)



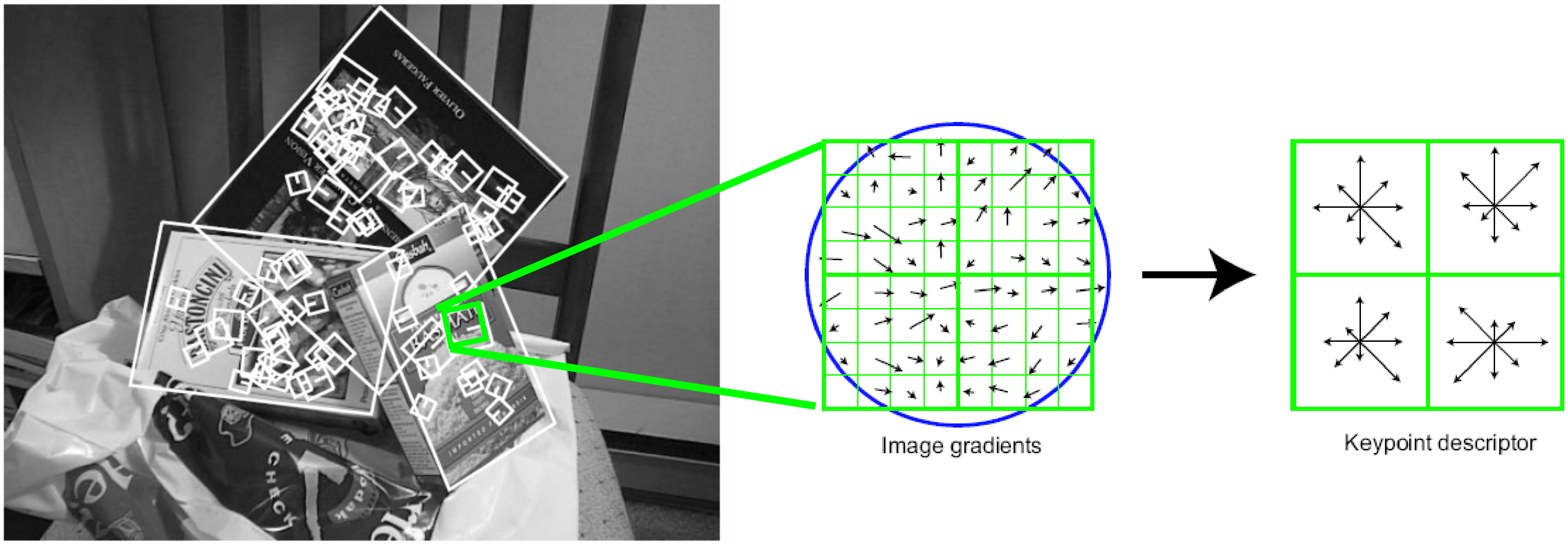


Map each grid cell in the input window to a histogram counting the gradients per orientation.

Code available:
http://pascal.inrialpes.fr/soft/olt/

Dalal & Triggs, CVPR 2005

Gradient-based representations: SIFT descriptor



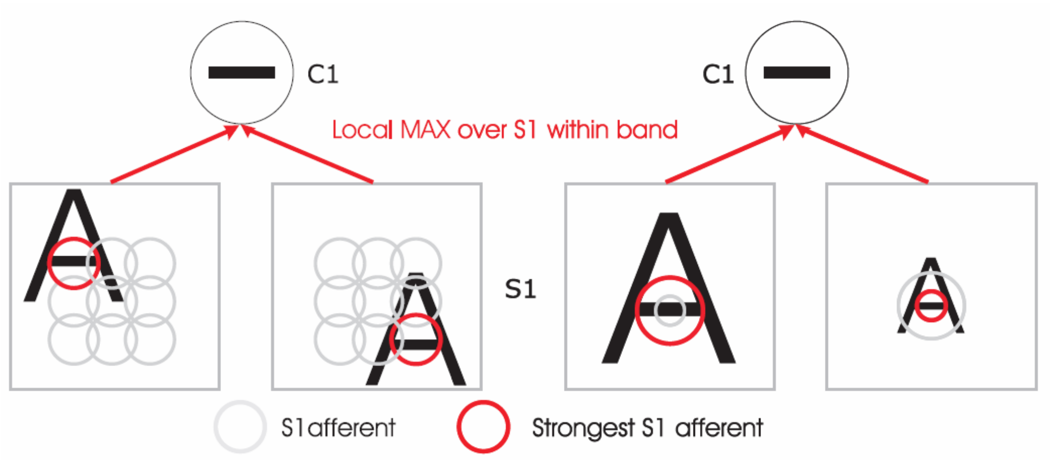
Local patch descriptor (more on this later)

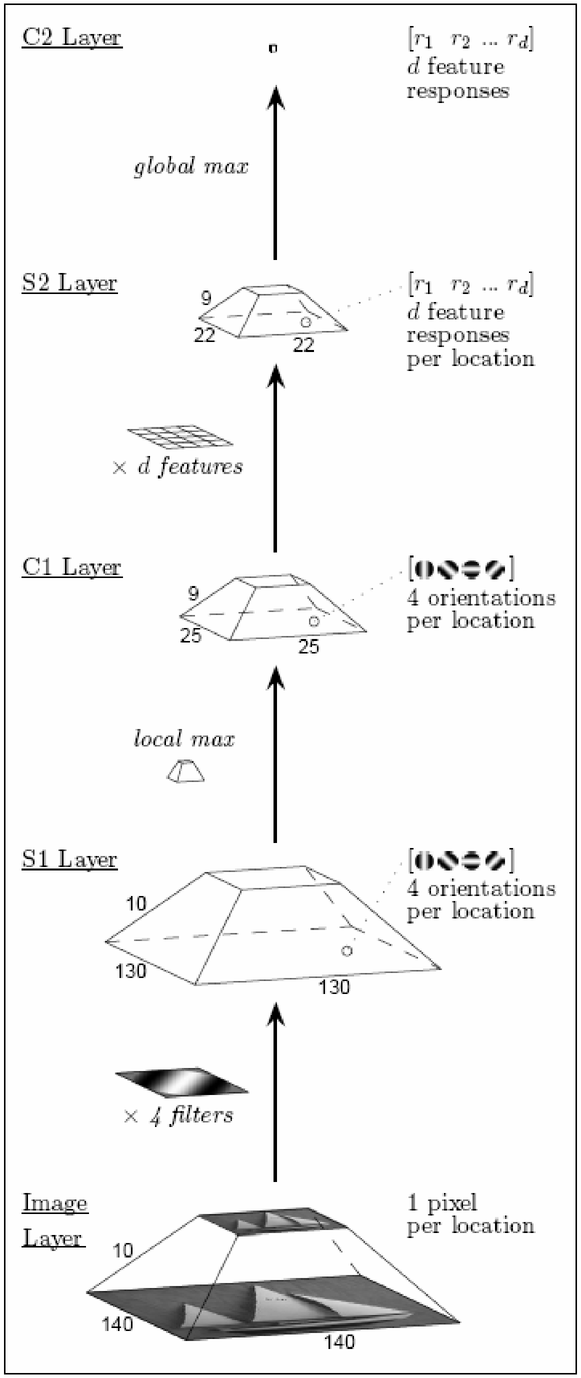
Code: http://vision.ucla.edu/~vedaldi/code/sift/sift.html

Binary: http://www.cs.ubc.ca/~lowe/keypoints/

Lowe, ICCV 1999

Gradient-based representations: Biologically inspired features





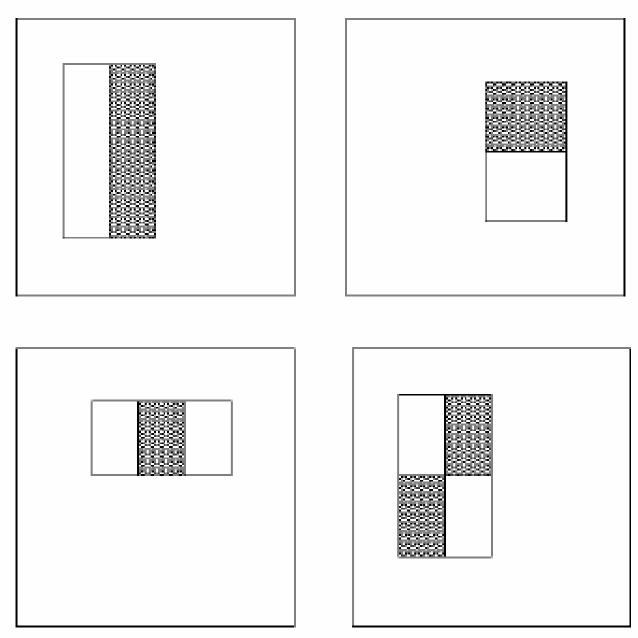
Convolve with Gabor filters at multiple orientations

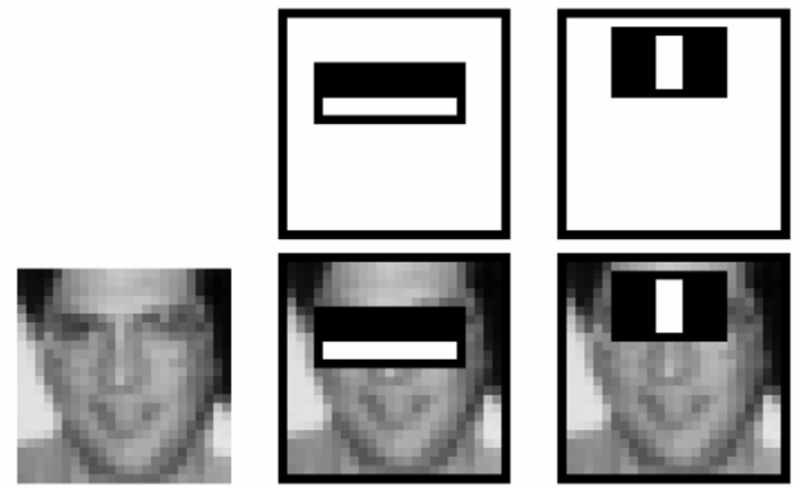
Pool nearby units (max)

Intermediate layers compare input to prototype patches

Serre, Wolf, Poggio, CVPR 2005
Mutch & Lowe, CVPR 2006

Gradient-based representations: Rectangular features





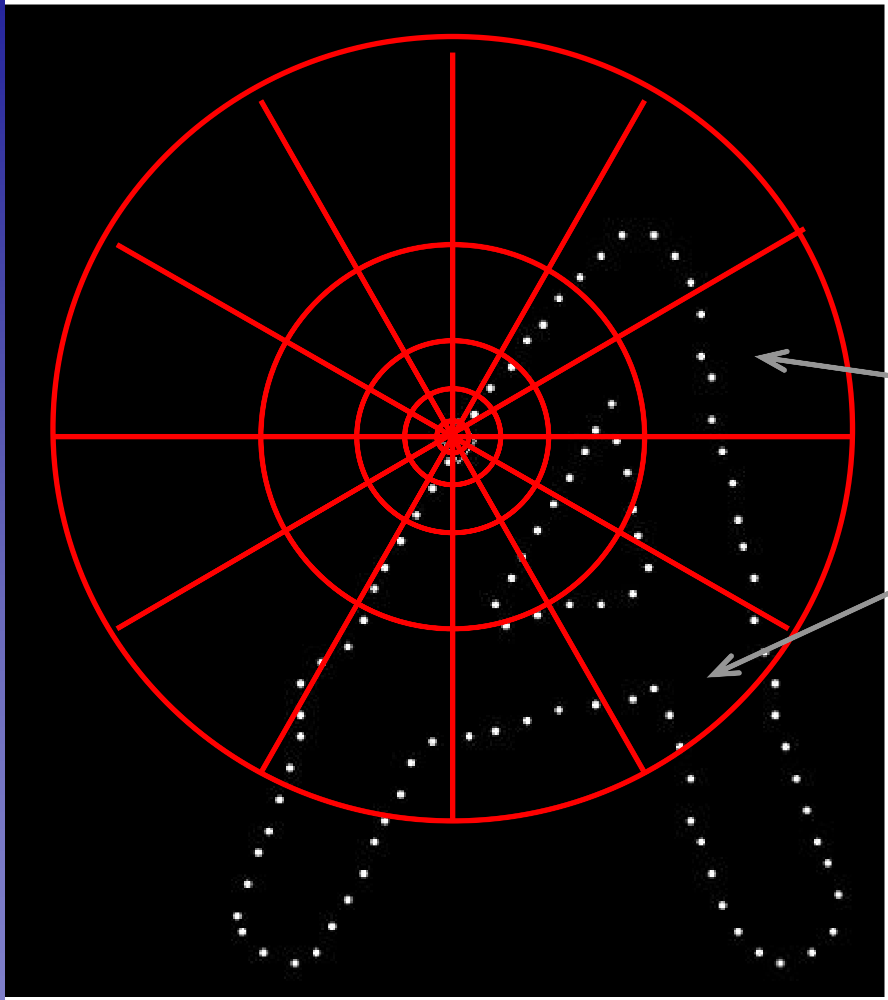
Compute differences between sums of pixels in rectangles

Captures contrast in adjacent spatial regions

Similar to Haar wavelets, efficient to compute

Viola & Jones, CVPR 2001

Gradient-based representations: Shape context descriptor



Count the number of points inside each bin, e.g.:

Count = 4

⋮

Count = 10

Log-polar binning: more precision for nearby points, more flexibility for farther points.

Local descriptor (more on this later)

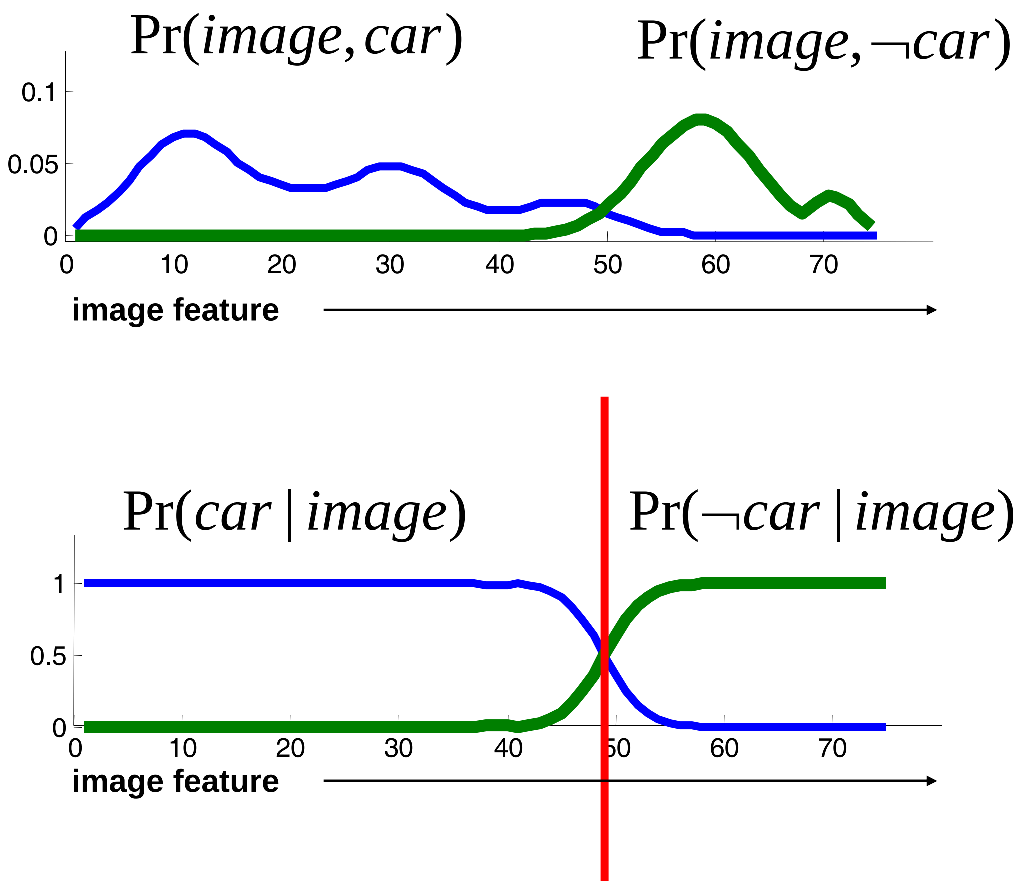
Belongie, Malik & Puzicha, ICCV 2001

Classifier construction

- **How to compute a decision for each subwindow?**

Image feature

Discriminative vs. generative models



Generative: separately model class-conditional and prior densities

Discriminative: directly model posterior

Plots from Antonio Torralba 2007

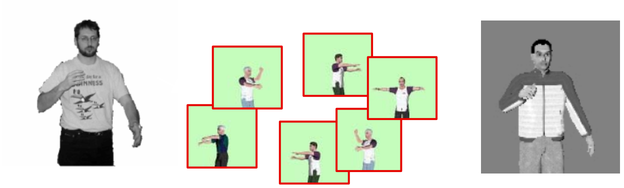
Discriminative vs. generative models

- **Generative:**
  - **+ possibly interpretable**
  - **+ can draw samples**
  - **- models variability unimportant to classification task**
  - **- often hard to build good model with few parameters**

- **Discriminative:**
  - **+ appealing when infeasible to model data itself**
  - **+ excel in practice**
  - **- often can't provide uncertainty in predictions**
  - **- non-interpretable**

Discriminative methods

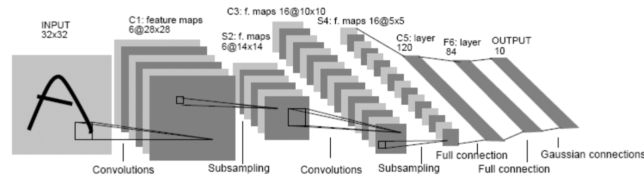
Nearest neighbor



10^6 examples

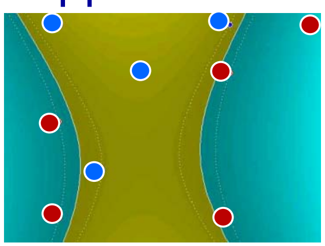
Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005...

Neural networks



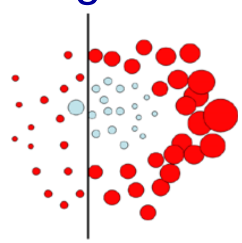
LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998
…

Support Vector Machines



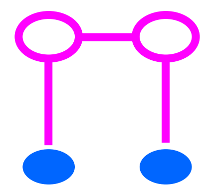
Guyon, Vapnik
Heisele, Serre, Poggio, 2001,…

Boosting



Viola, Jones 2001,
Torralba et al. 2004,
Opelt et al. 2006,…

Conditional Random Fields



McCallum, Freitag, Pereira 2000; Kumar, Hebert 2003
…

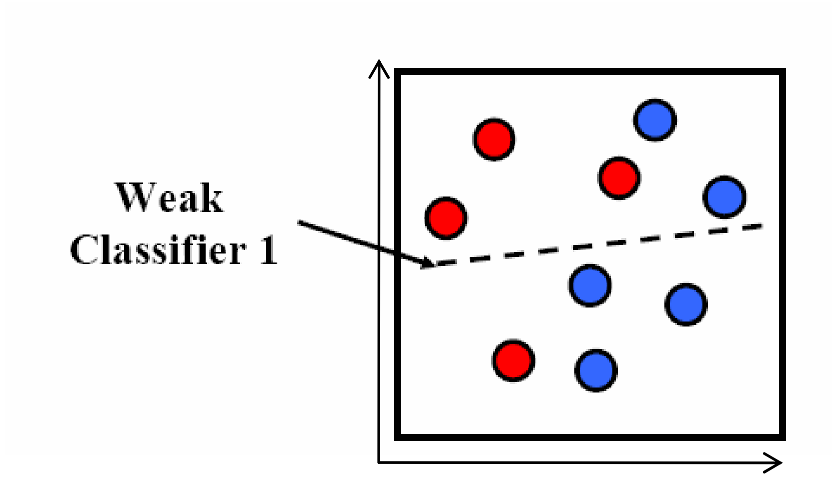
Slide adapted from Antonio Torralba

Boosting

- **Build a strong classifier by combining number of "weak classifiers", which need only be better than chance**
- **Sequential learning process: at each iteration, add a weak classifier**
- **Flexible to choice of weak learner**
  - **including fast simple classifiers that alone may be inaccurate**

- **We'll look at Freund & Schapire's AdaBoost algorithm**
  - **Easy to implement**
  - **Base learning algorithm for Viola-Jones face detector**

AdaBoost: Intuition



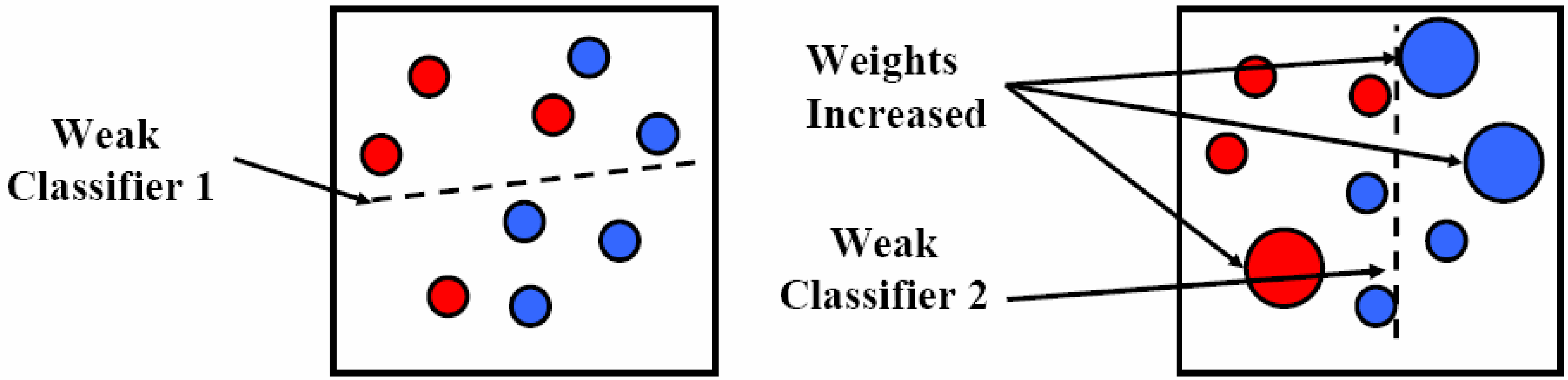
Consider a 2-d feature space with positive and negative examples.

Each weak classifier splits the training examples with at least 50% accuracy.

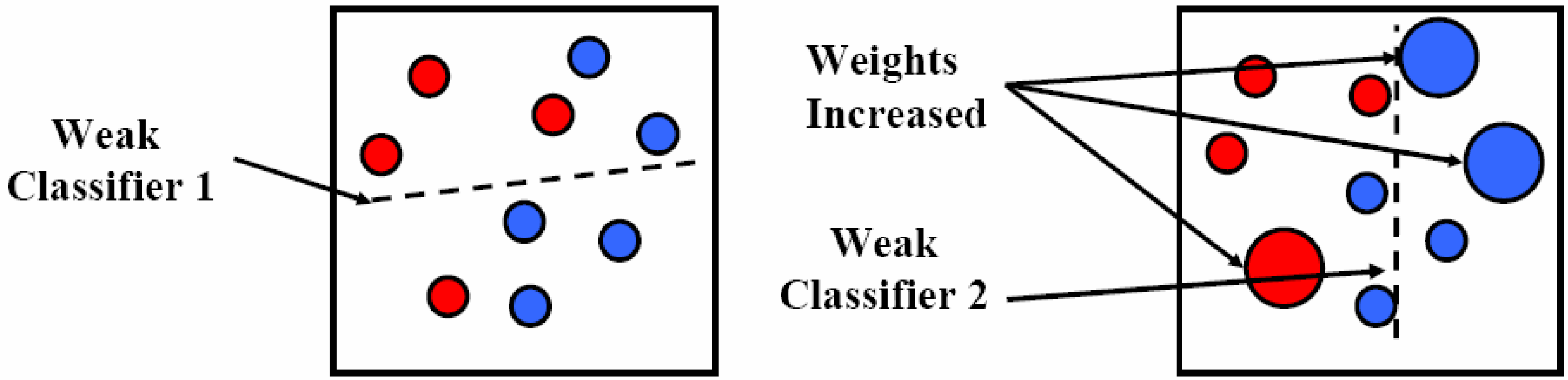
Examples misclassified by a previous weak learner are given more emphasis at future rounds.

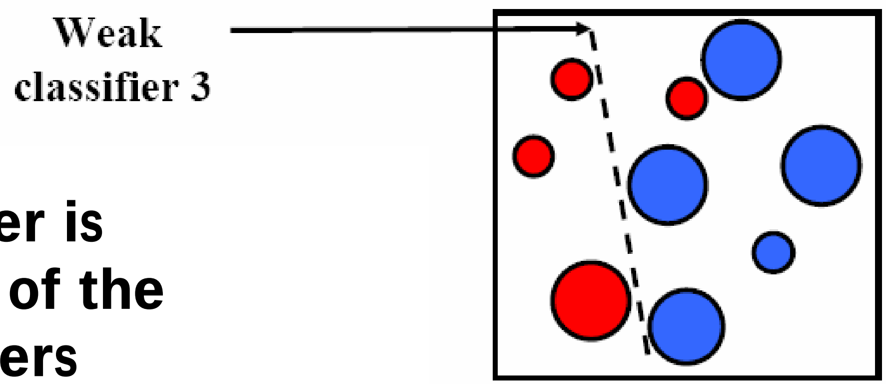
Figure adapted from Freund and Schapire

AdaBoost: Intuition



AdaBoost: Intuition





Final classifier is combination of the weak classifiers

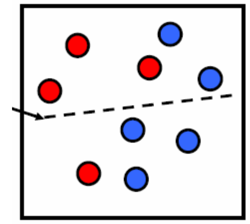
AdaBoost Algorithm

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights,
     $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$
     so that w_t is a probability distribution.
  2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i \, |h_j(x_i) - y_i|$.
  3. Choose the classifier, h_t, with the lowest error ϵ_t.
  4. Update the weights:
     $$w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$$
     where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha_t = \log \frac{1}{\beta_t}$

Start with uniform weights on training examples



$\{x_1, \ldots x_n\}$

Evaluate *weighted* error for each feature, pick best.

Incorrectly classified -> more weight

Correctly classified -> less weight

Final classifier is combination of the weak ones, weighted according to error they had.

Freund & Schapire 1995

Cascading classifiers for detection

For efficiency, apply less accurate but faster classifiers first to immediately discard windows that clearly appear to be negative; e.g.,

- **Filter for promising regions with an initial inexpensive classifier**
- **Build a chain of classifiers, choosing cheap ones with low false negative rates early in the chain**

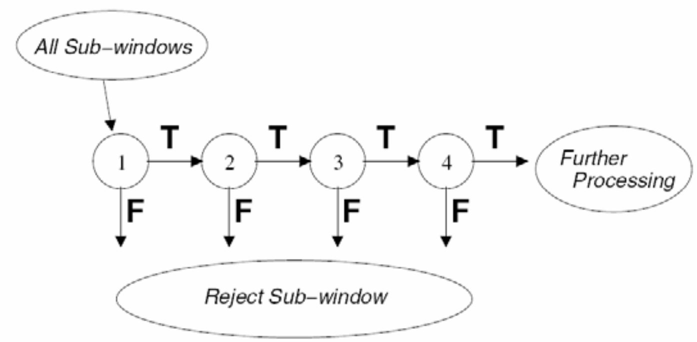
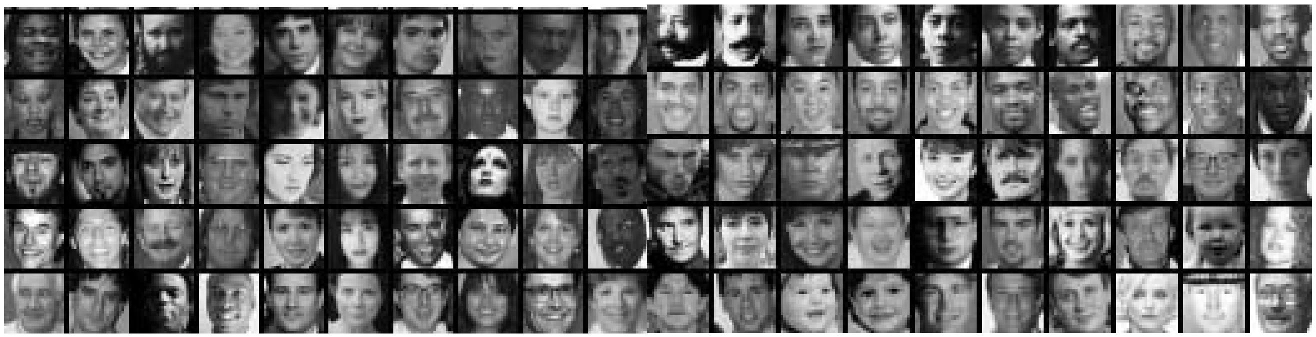


Figure from Viola & Jones CVPR 2001

Fleuret & Geman, IJCV 2001
Rowley et al., PAMI 1998
Viola & Jones, CVPR 2001

Example: Face detection

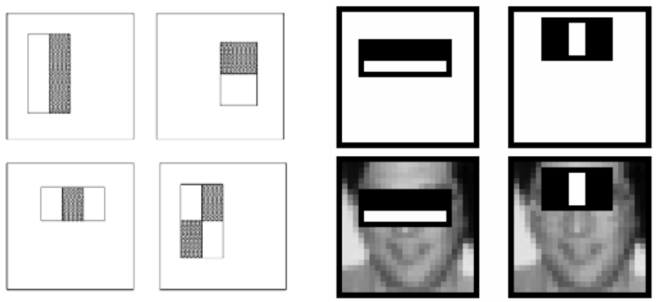
- **Frontal faces are a good example of a class where global appearance models + a sliding window detection approach fit well:**
  - **Regular 2D structure**
  - **Center of face almost shaped like a "patch"/window**



- **Now we'll take AdaBoost and see how the Viola-Jones face detector works**

Feature extraction

"Rectangular" filters

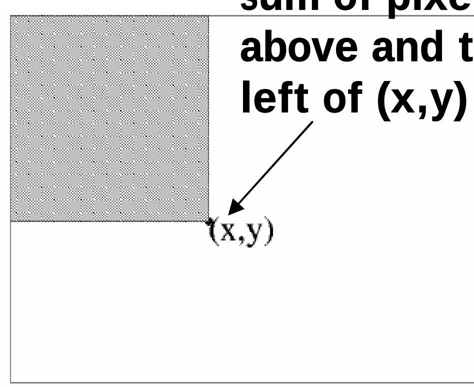


Feature output is difference between adjacent regions

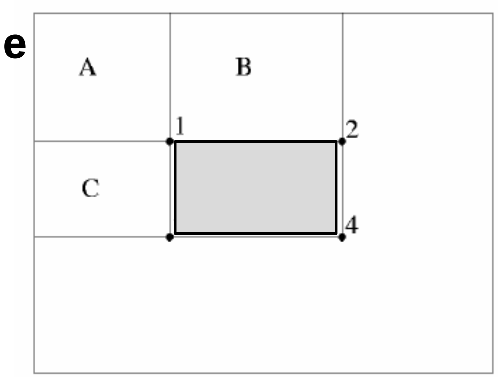
Efficiently computable with integral image: any sum can be computed in constant time

Avoid scaling images → scale features directly for same cost

Value at (x,y) is sum of pixels above and to the left of (x,y)



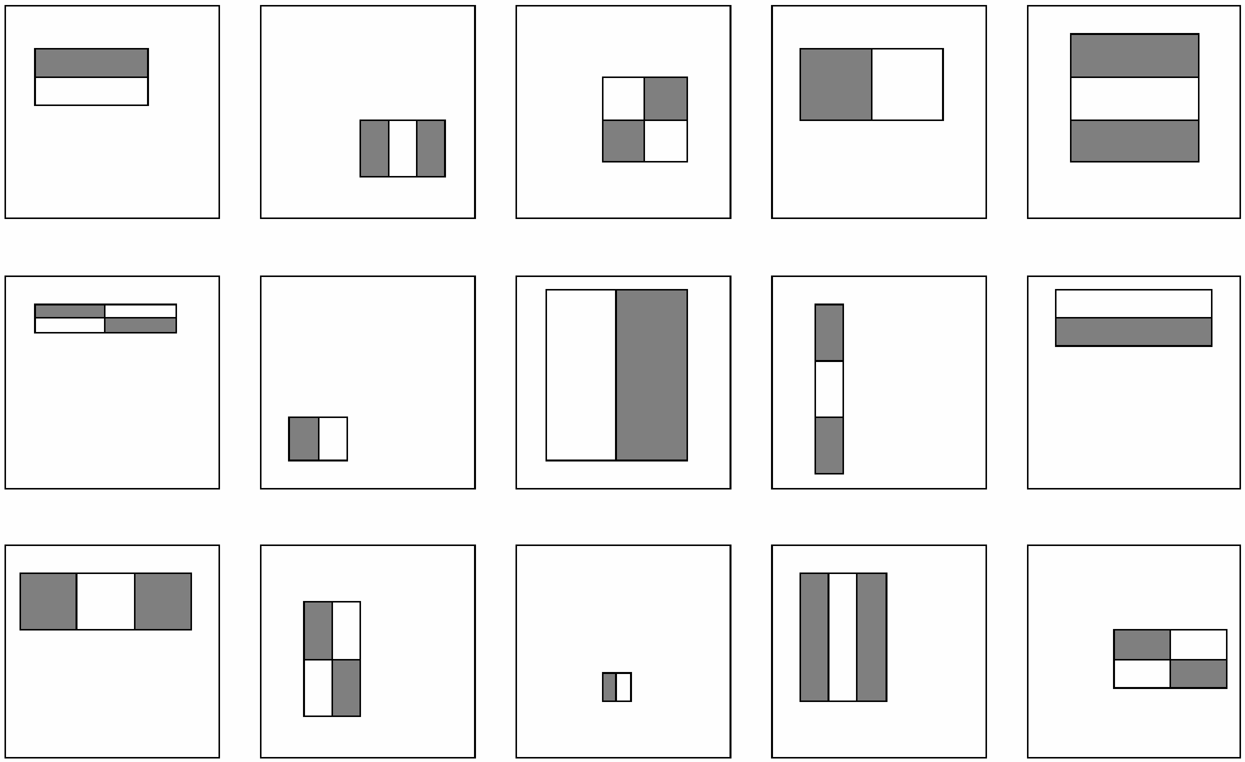
Integral image



$$
\begin{aligned}
D &= 1 + 4 - (2 + 3) \\
&= A + (A + B + C + D) - (A + C + A + B) \\
&= D
\end{aligned}
$$

Viola & Jones, CVPR 2001

Large library of filters



Considering all possible filter parameters: position, scale, and type:

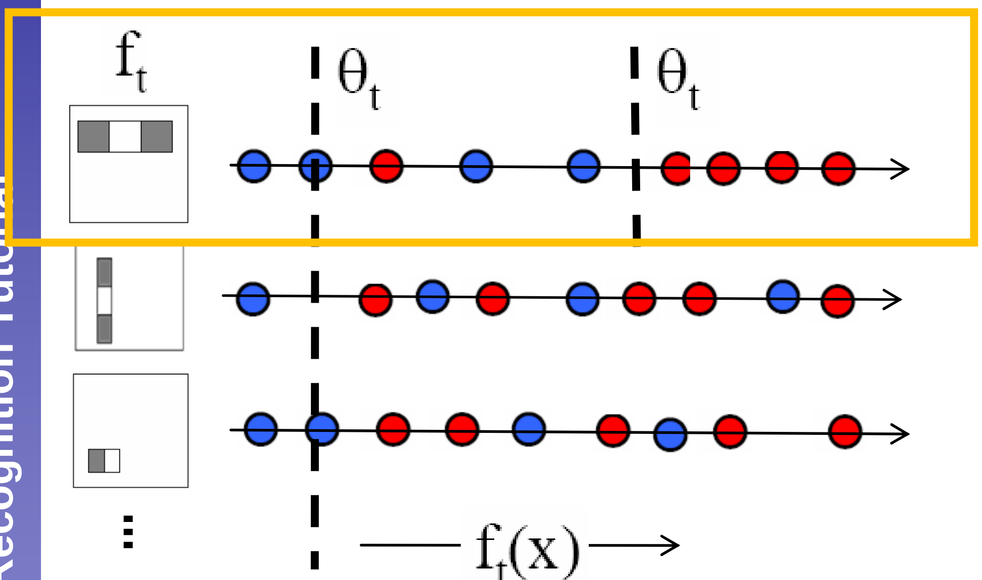
180,000+ possible features associated with each 24 x 24 window

Use AdaBoost both to select the informative features and to form the classifier

Viola & Jones, CVPR 2001

AdaBoost for feature+classifier selection

- Want to select the single rectangle feature and threshold that best separates positive (faces) and negative (non-faces) training examples, in terms of *weighted* error.



Outputs of a possible rectangle feature on faces and non-faces.

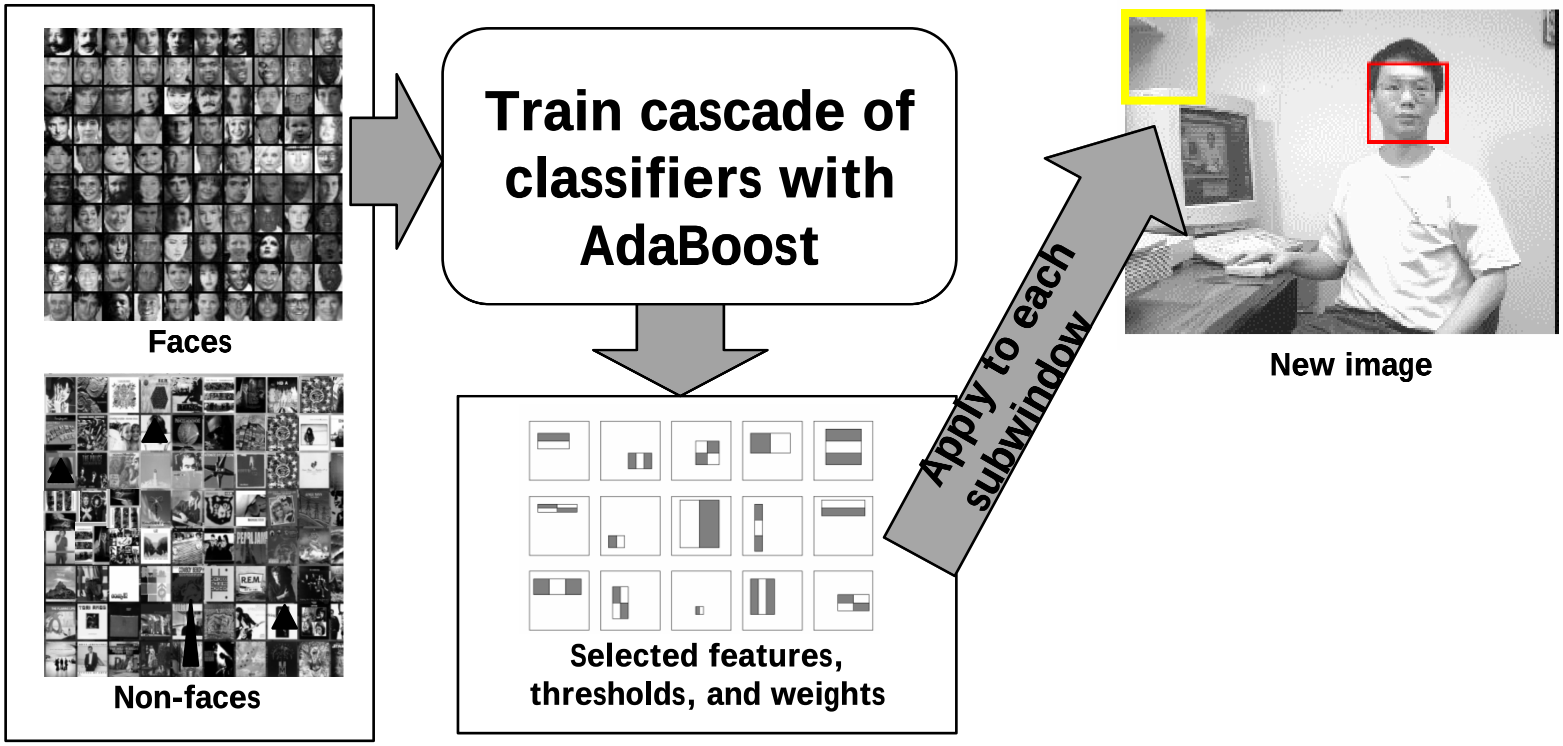
Resulting weak classifier:

$$h_t(x) = \begin{cases} +1 & \text{if } f_t(x) > \theta_t \\ -1 & \text{otherwise} \end{cases}$$

For next round, reweight the examples according to errors, choose another filter/threshold combo.

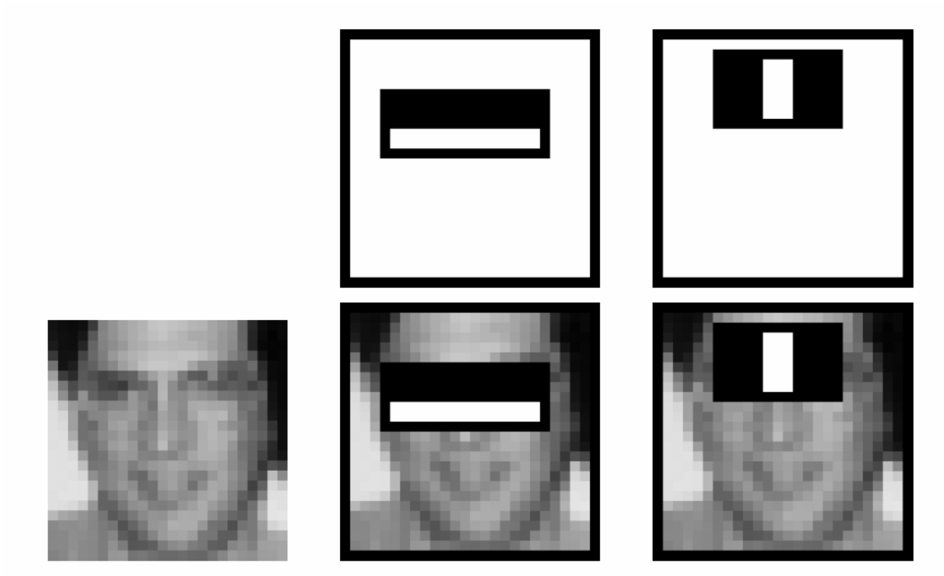
Viola & Jones, CVPR 2001

Viola-Jones Face Detector: Summary



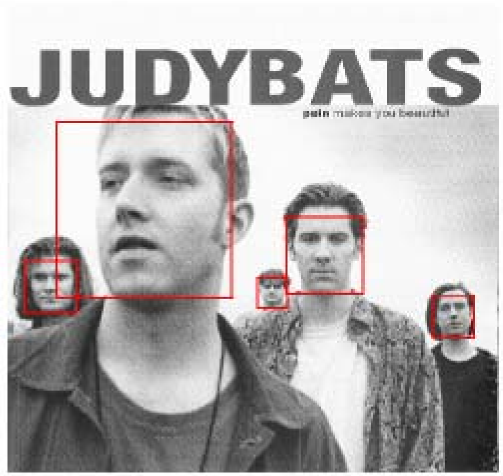
- **Train with 5K positives, 350M negatives**
- **Real-time detector using 38 layer cascade**
- **6061 features in final layer**
- [Implementation available in OpenCV: http://www.intel.com/technology/computing/opencv/]

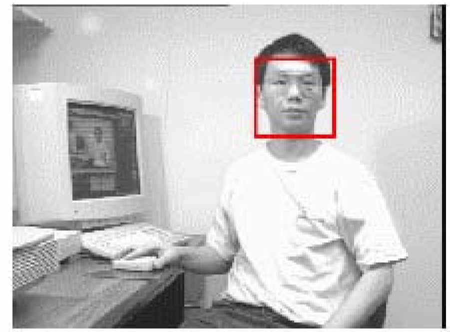
Viola-Jones Face Detector: Results

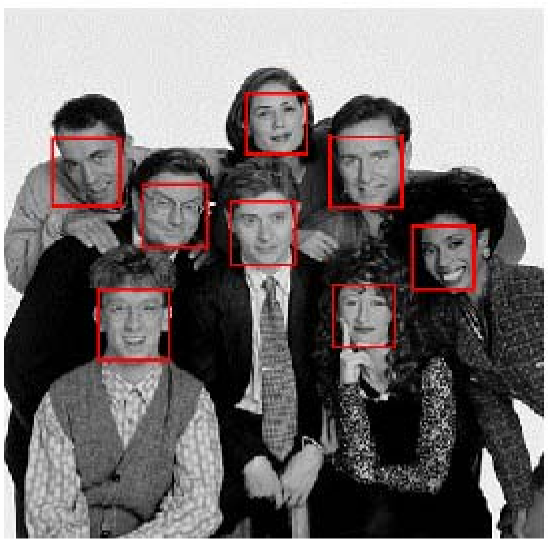


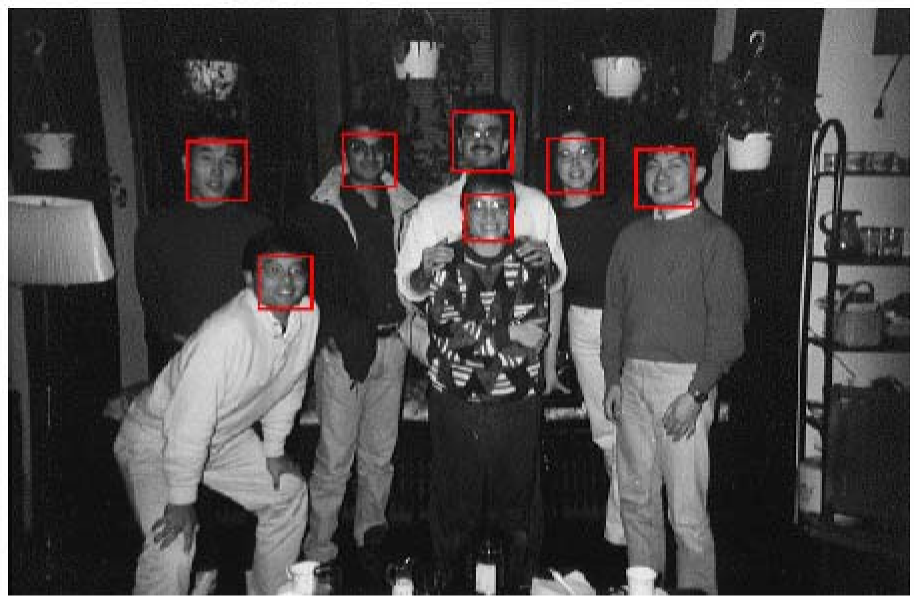
First two features selected

Viola-Jones Face Detector: Results

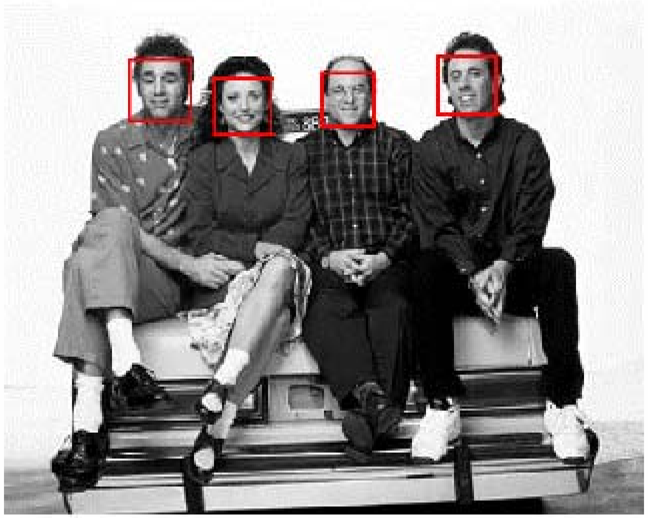


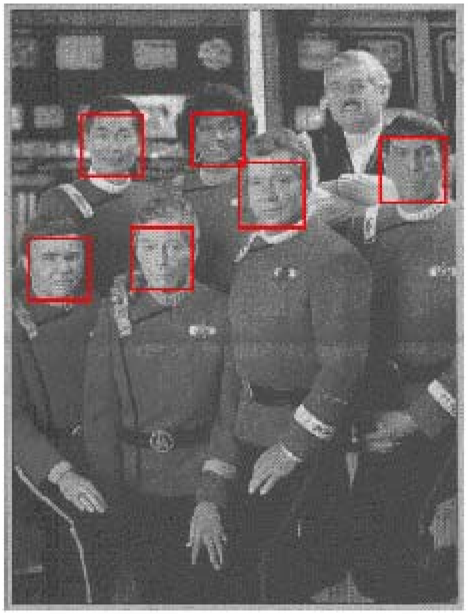


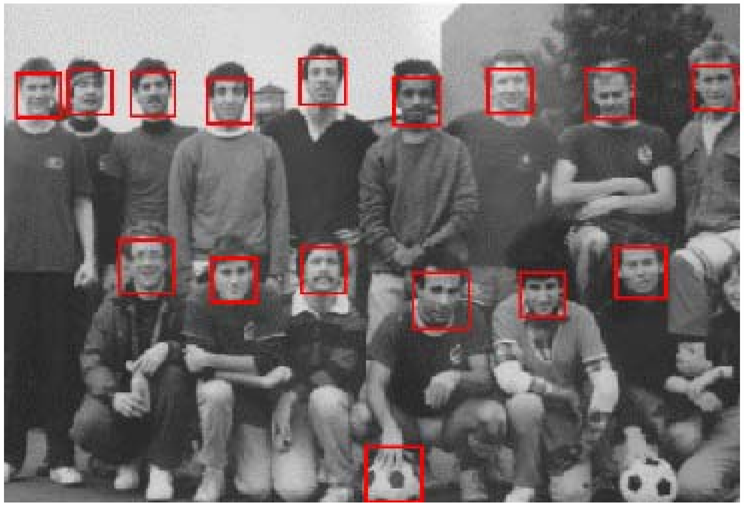


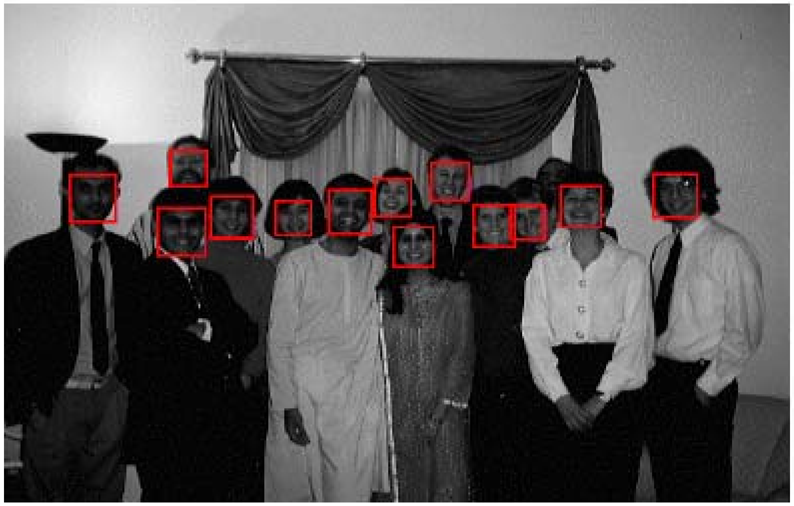


Viola-Jones Face Detector: Results





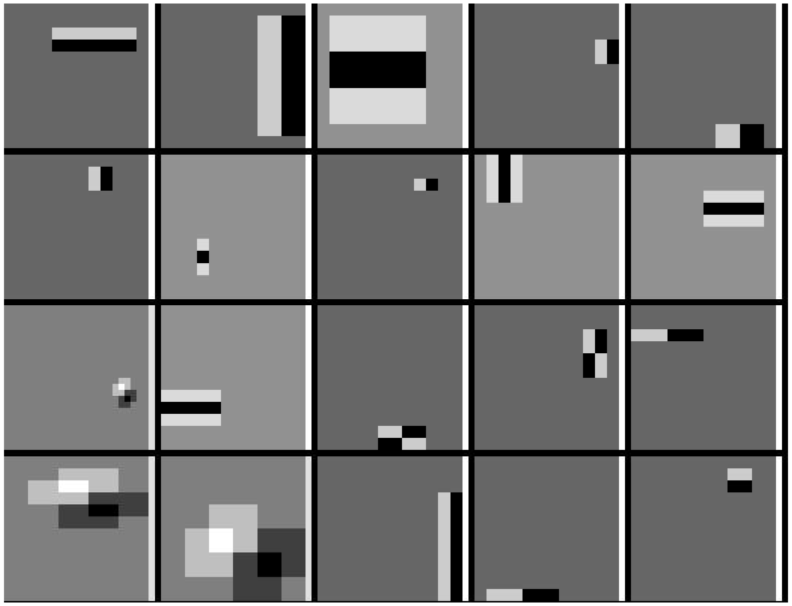




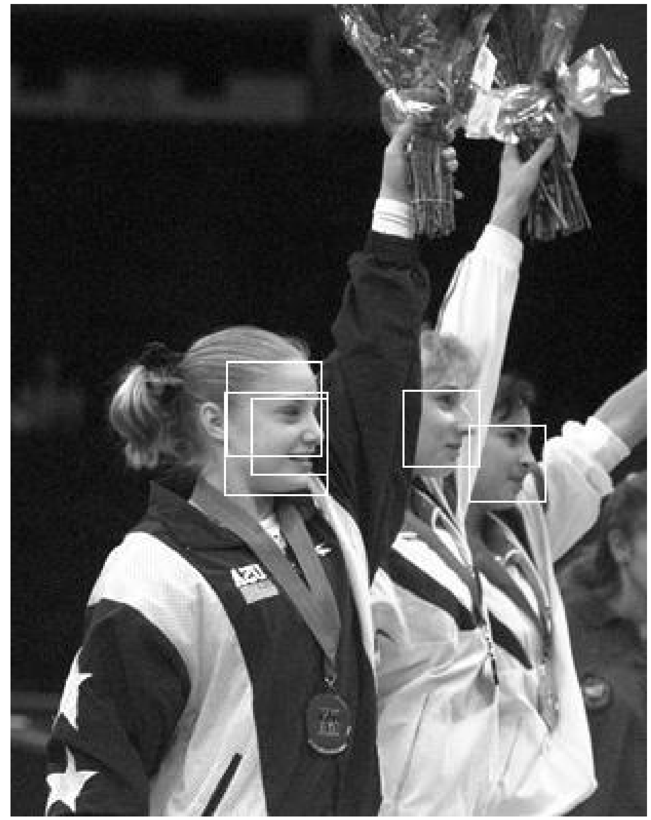
Viola-Jones Face Detector: Results

Profile Features

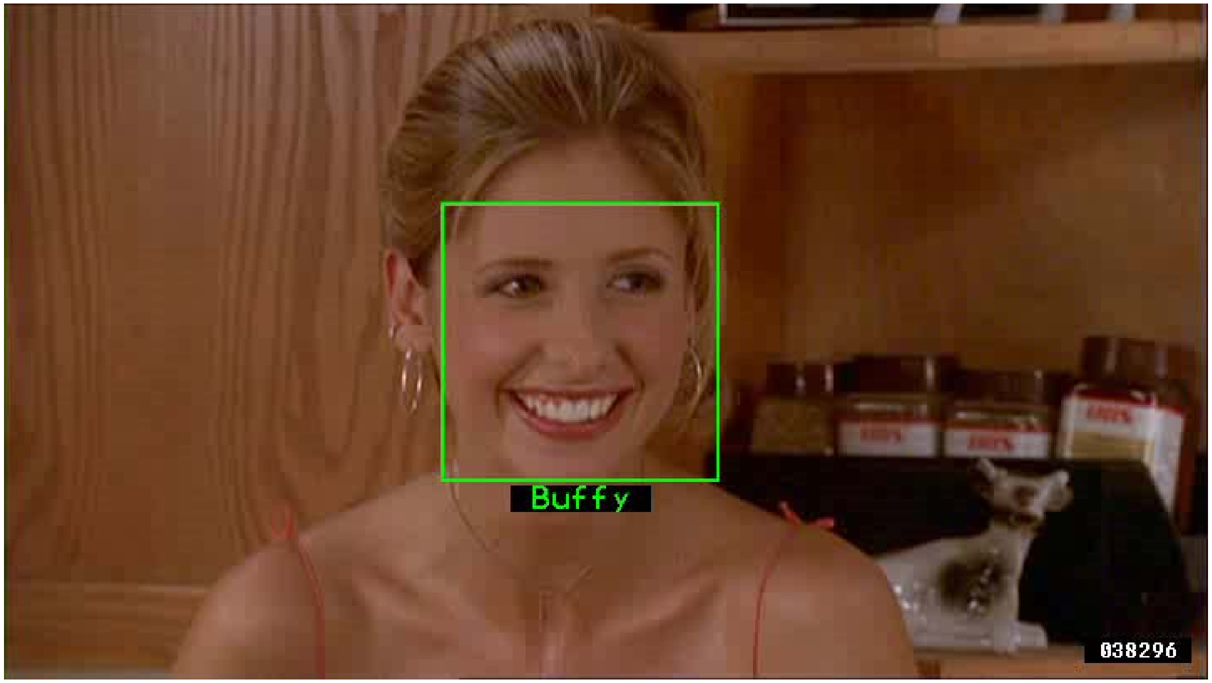
Detecting profile faces requires training separate detector with profile examples.



Viola-Jones Face Detector: Results



Example application

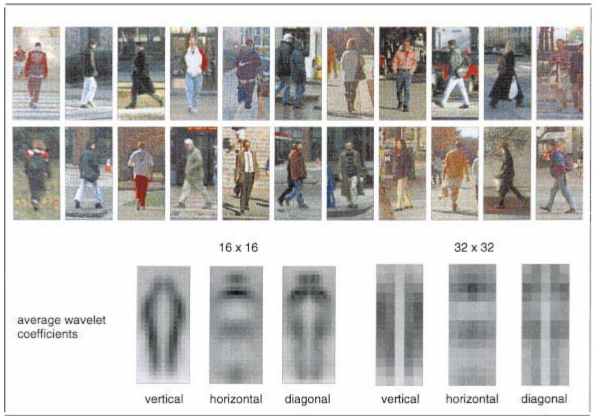


Frontal faces detected and then tracked, character names inferred with alignment of script and subtitles.

Everingham, M., Sivic, J. and Zisserman, A.
"Hello! My name is... Buffy" - Automatic naming of characters in TV video, BMVC 2006.
http://www.robots.ox.ac.uk/~vgg/research/nface/index.html

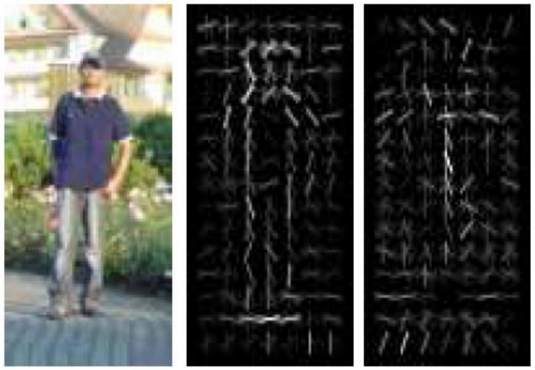
Pedestrian detection

- Detecting upright, walking humans also possible using sliding window's appearance/texture; e.g.,



SVM with Haar wavelets [Papageorgiou & Poggio, IJCV 2000]

Space-time rectangle features [Viola, Jones & Snow, ICCV 2003]



SVM with HoGs [Dalal & Triggs, CVPR 2005]

Highlights

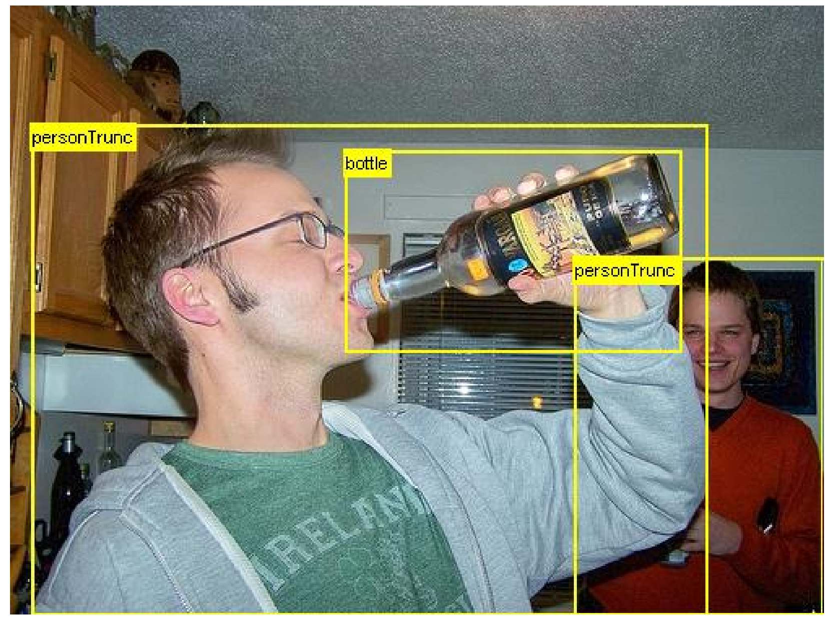
- **Sliding window detection and global appearance descriptors:**
  - **Simple detection protocol to implement**
  - **Good feature choices critical**
  - **Past successes for certain classes**

Limitations

- High computational complexity
  - For example: 250,000 locations x 30 orientations x 4 scales = 30,000,000 evaluations!
  - If training binary detectors independently, means cost increases linearly with number of classes
- With so many windows, false positive rate better be low

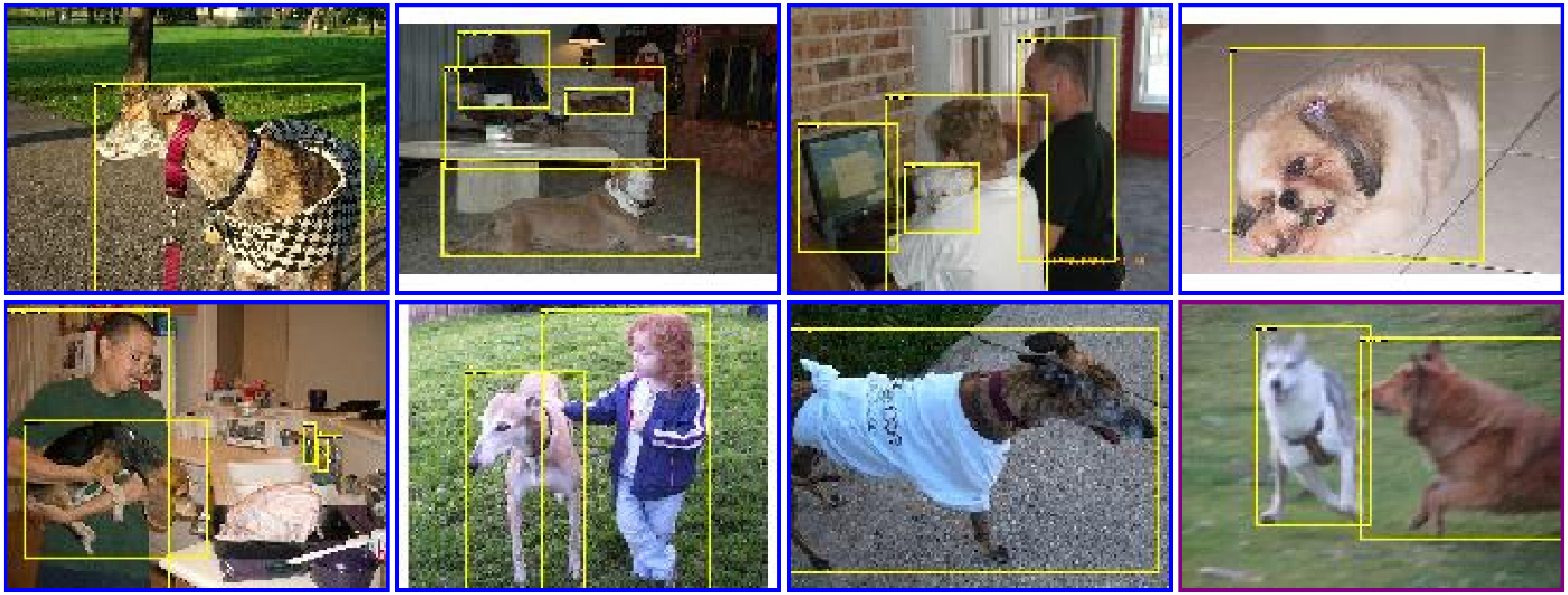
Limitations (continued)

- **Not all objects are "box" shaped**



Limitations (continued)

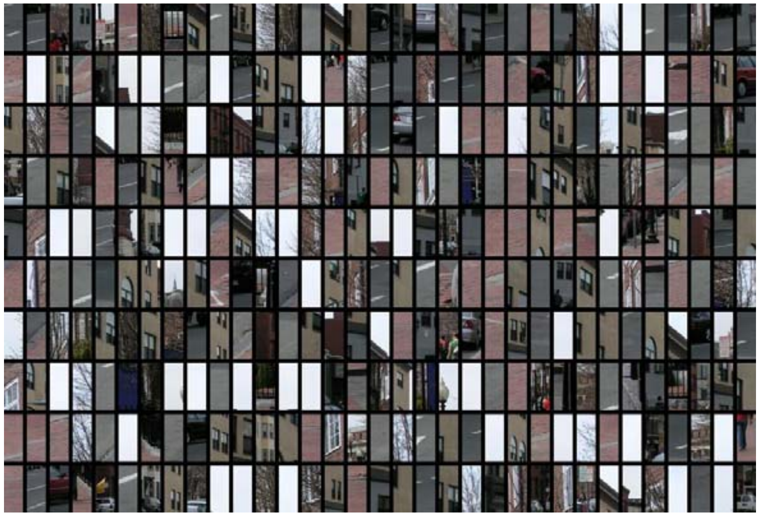
- Non-rigid, deformable objects not captured well with representations assuming a fixed 2d structure; or must assume fixed viewpoint
- Objects with less-regular textures not captured well with holistic appearance-based descriptions



Limitations (continued)

- **If considering windows in isolation, context is lost**

Sliding window



Detector's view

Figure credit: Derek Hoiem

Limitations (continued)

- **In practice, often entails large, cropped training set (expensive)**
- **Requiring good match to a global appearance description can lead to sensitivity to partial occlusions**

Image credit: Adam, Rivlin, & Shimshoni

Outline

1. Detection with Global Appearance & Sliding Windows
2. Local Invariant Features: Detection & Description
3. Specific Object Recognition with Local Features

– *Coffee Break* –

4. Visual Words: Indexing, Bags of Words Categorization
5. Matching Local Features
6. Part-Based Models for Categorization
7. Current Challenges and Research Directions