מבוא לראייה ממוחשבת – 22928
2016א

מנחה: אמיר אגוזי
egozi5@gmail.com

מפגש מס' 4

בפעם שעברה

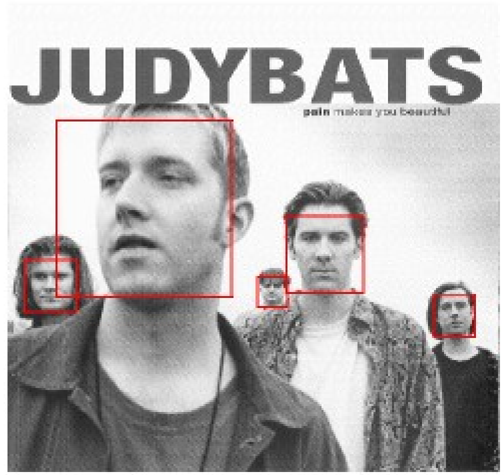
- PCA
- עקרונות לימוד מכונה
  - Feature extraction
  - Training
  - Validation
  - Test
  - Performance evaluation

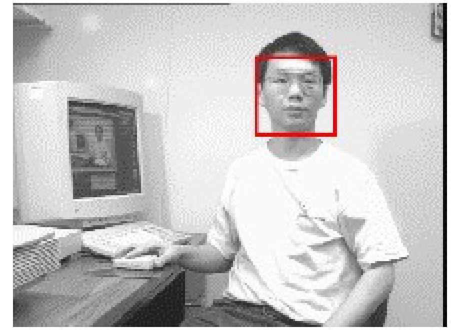
היום

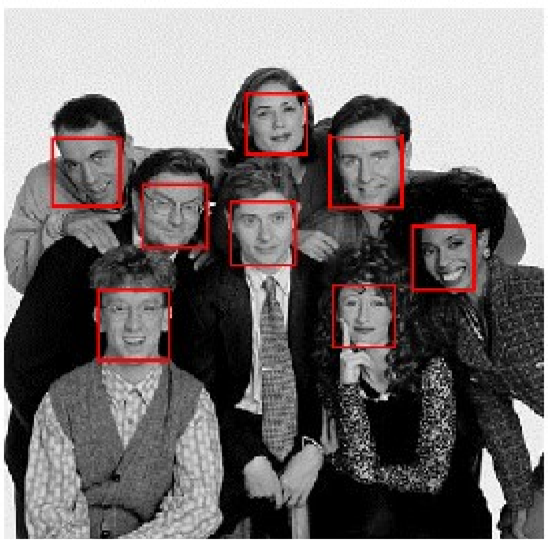
- חזרה על סיווג – BOW + SVM
- זיהוי ע"י סיווג
- זיהוי פנים בשיטת Viola-jones
- Boosting
- Cascade classifier

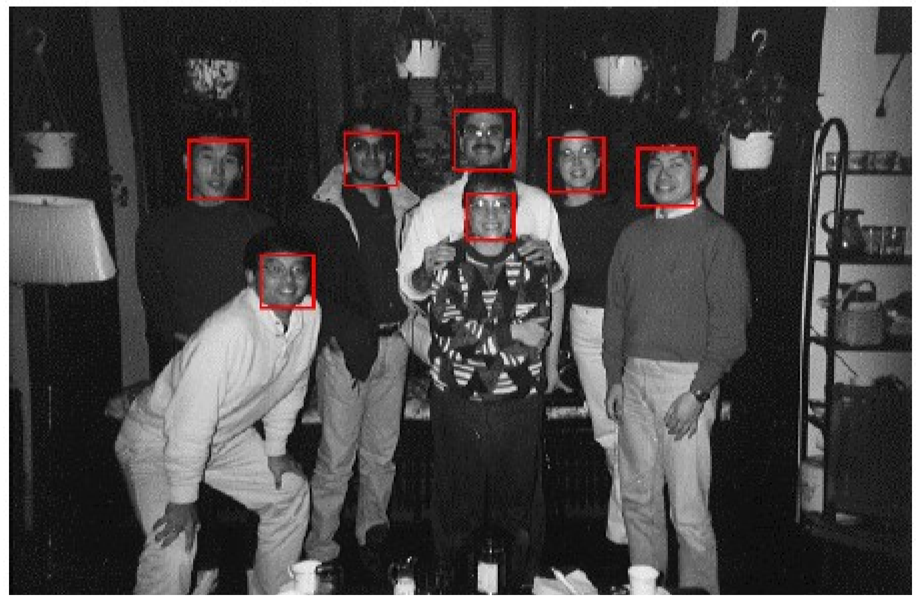
Detection via classification: Main idea

- **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).**









איתור פנים בשיטה של ויאולה וג'ונס

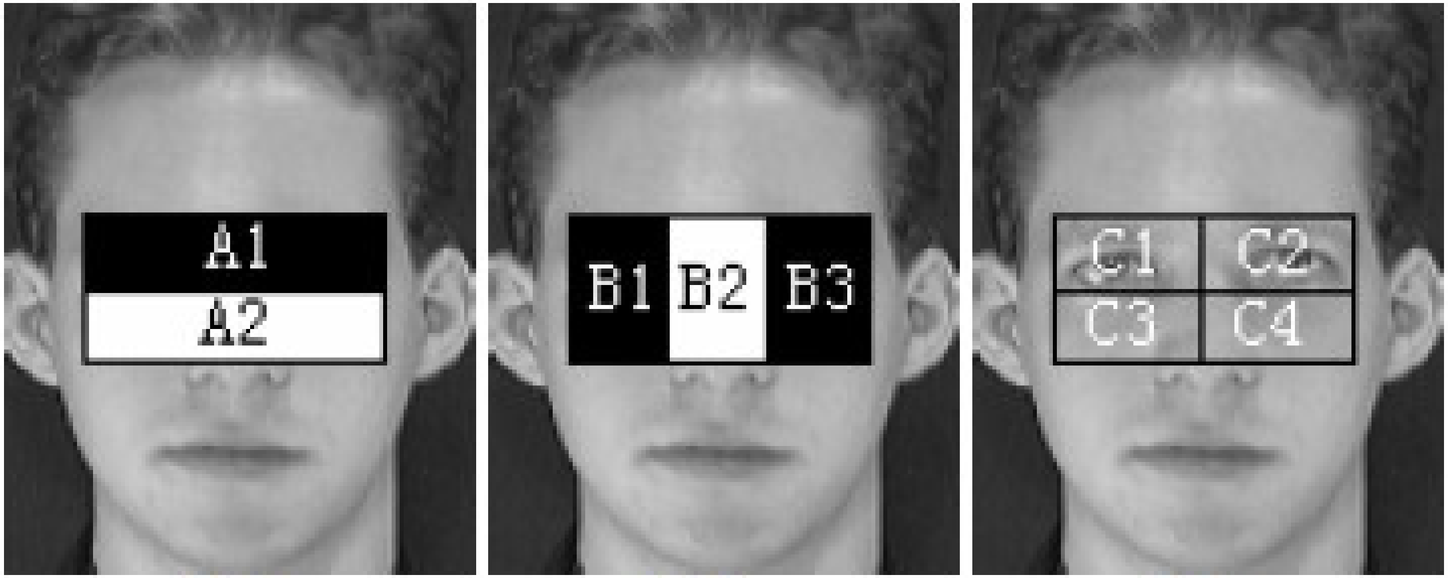
Three Main Contributions

1. **Integral image**, rectangle features can be computed efficiently
2. **Boosting** - constructing a classifier using AdaBoost - learning algorithm developed by Fruend and Schapire, selects a small set of features and build a classifier based on them
3. **Combining** successively more **complex classifiers** in a "**cascade**" – focus attention on promising regions of the picture

בראשי פרקים

- Rectangle features
- Integral image
- Weak learner
- Strong learner
- The boosting algorithm
- The cascade

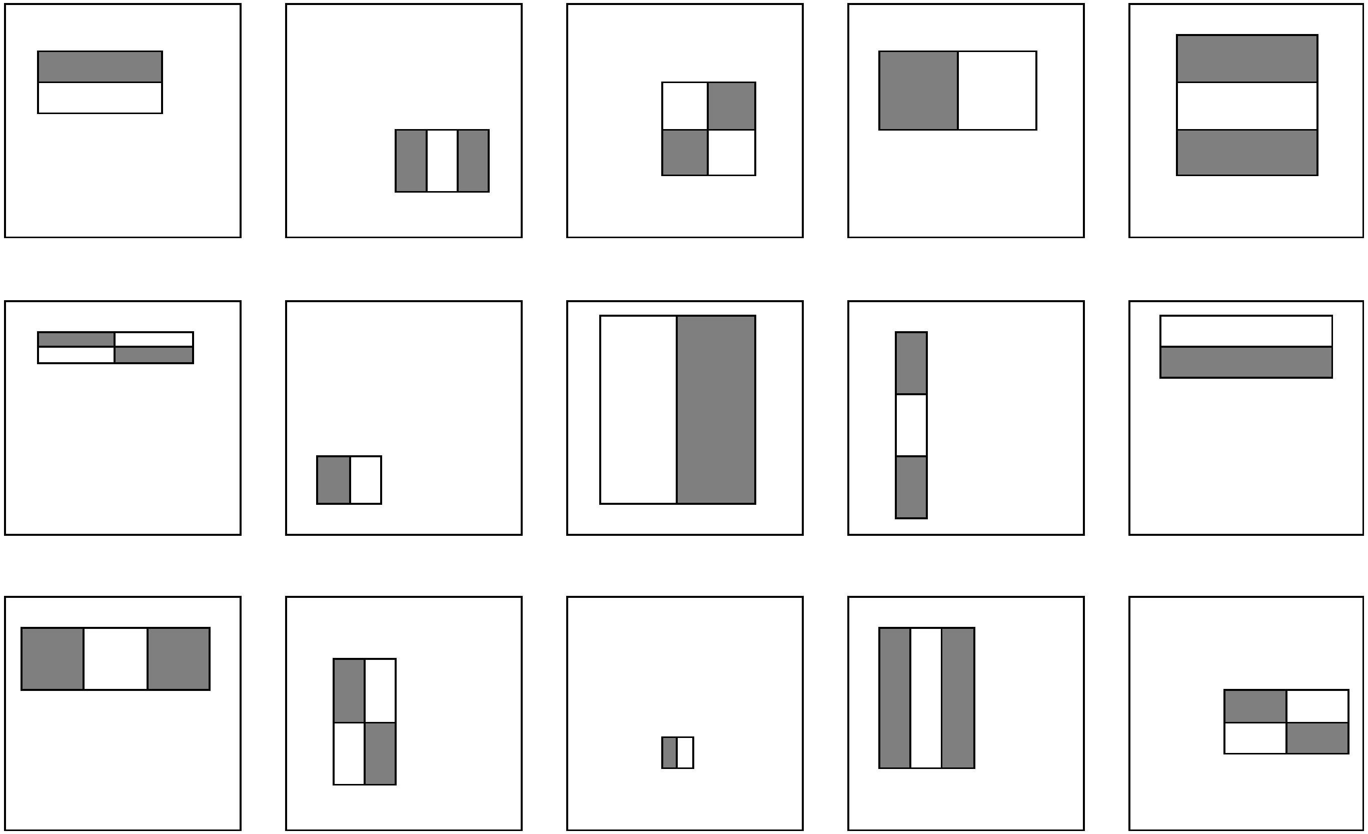
Rectangle Features



(a) (b) (c)

Rectangle Features

For a 24x24 sub-window, the number of possible rectangle features is ~180,000! Select using AdaBoost…



Integral Image Representation

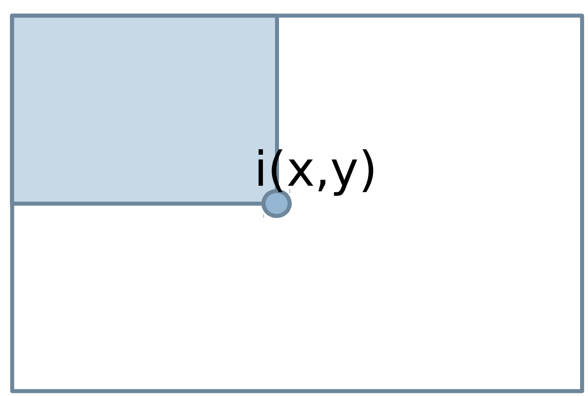
- Each pixel contains the **sum of the pixels above and to the left** of it:

$$ii(x,y) = \sum_{x' \le x,\, y' \le y} i(x',y')$$

The value in the integral image

The value in the original image

- Example:



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

A Formal View of Boosting

- given training set $(x_1, y_1), \ldots, (x_m, y_m)$
- $y_i \in \{-1, +1\}$ correct label of instance $x_i \in X$
- for $t = 1, \ldots, T$:
  - construct distribution D_t on $\{1, \ldots, m\}$
  - find weak hypothesis ("rule of thumb")

    $$h_t : X \to \{-1, +1\}$$

    with small error ϵ_t on D_t:

    $$\epsilon_t = \Pr_{D_t}[h_t(x_i) \neq y_i]$$
- output final hypothesis H_{final}

, From – A tutorial on Boosting
Yoav Freund and Rob Schapire

AdaBoost

[Freund & Schapire]

- constructing D_t:
  - $D_1(i) = 1/m$
  - given D_t and h_t:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \cdot \begin{cases} e^{-\alpha_t} & \text{if } y_i = h_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq h_t(x_i) \end{cases}$$

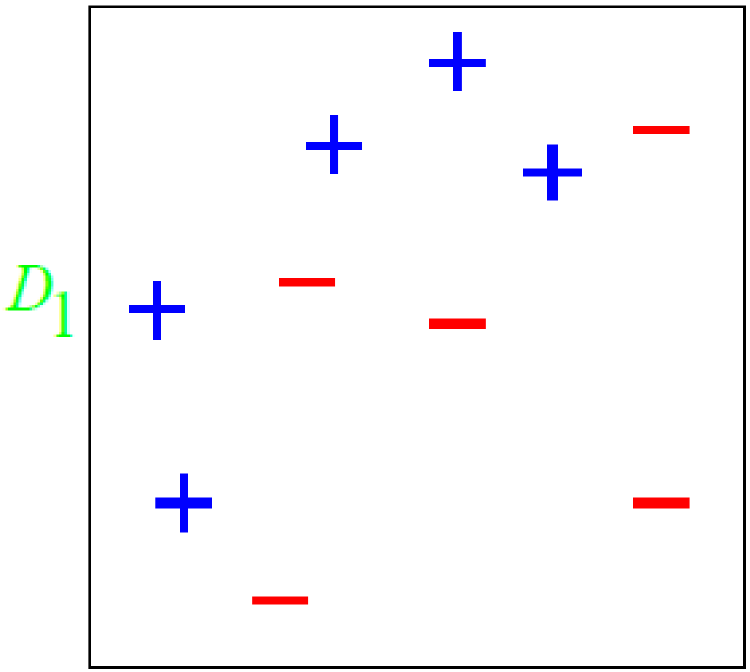
$$= \frac{D_t(i)}{Z_t} \cdot \exp(-\alpha_t \, y_i \, h_t(x_i))$$

where Z_t = normalization constant

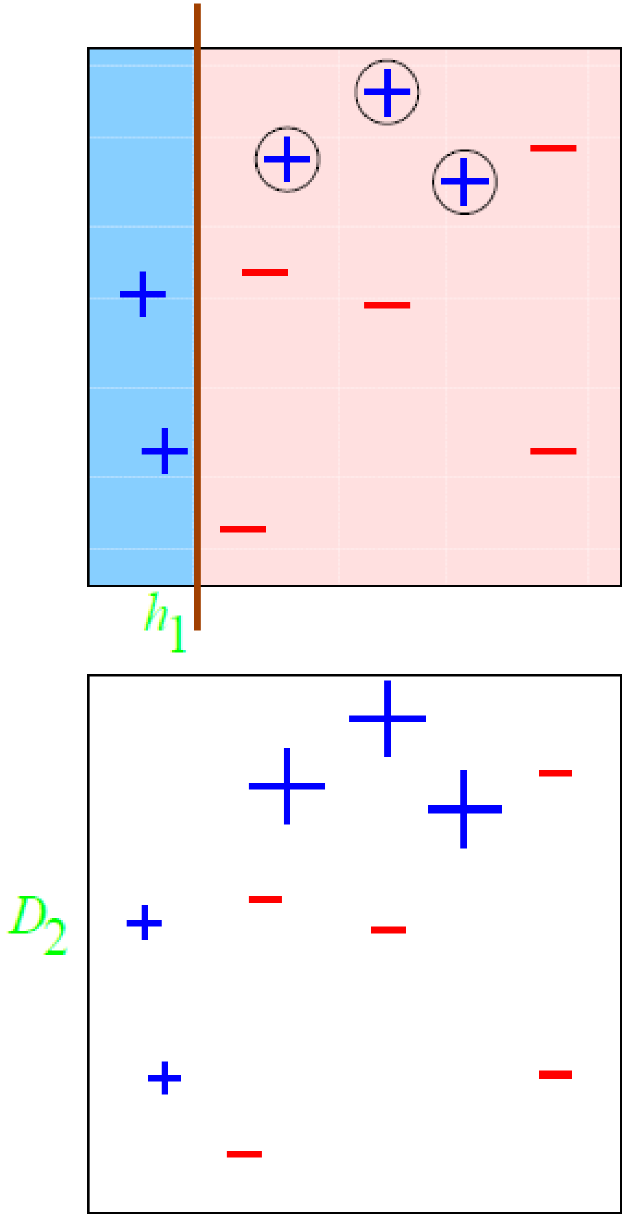
$$\alpha_t = \tfrac{1}{2} \ln\left(\frac{1 - \epsilon_t}{\epsilon_t}\right) > 0$$

- final hypothesis:
  - $H_{\text{final}}(x) = \text{sign}\left(\sum_t \alpha_t h_t(x)\right)$

Toy Example



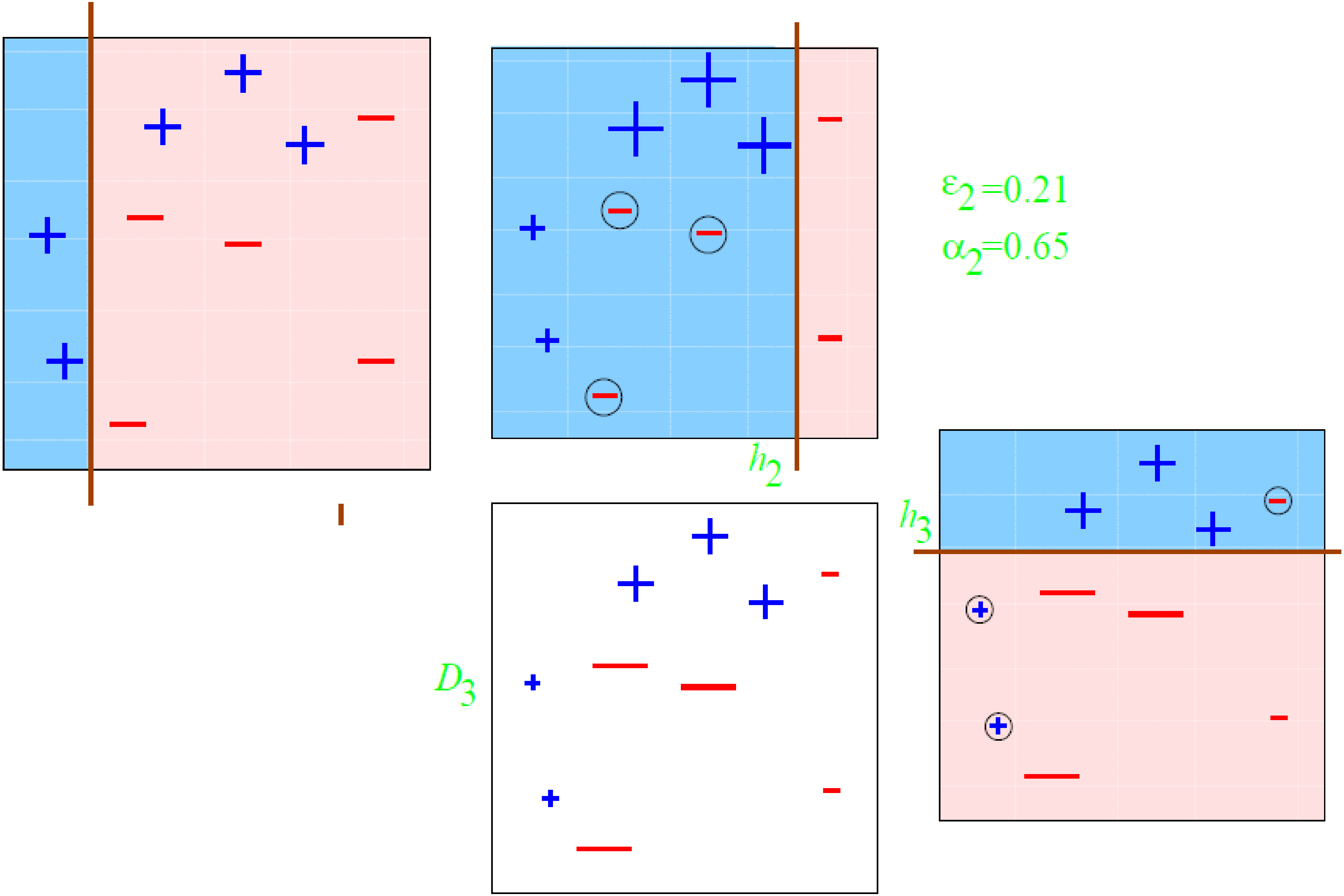
Round 1



$\varepsilon_1 = 0.30$

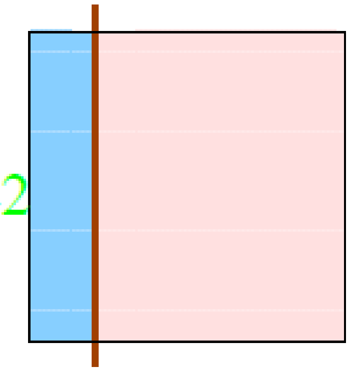
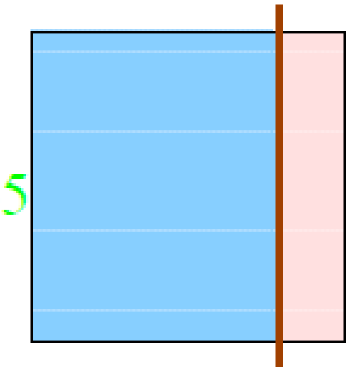
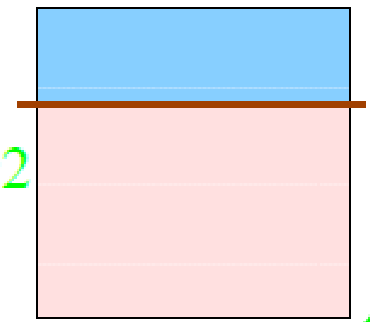
$\alpha_1 = 0.42$

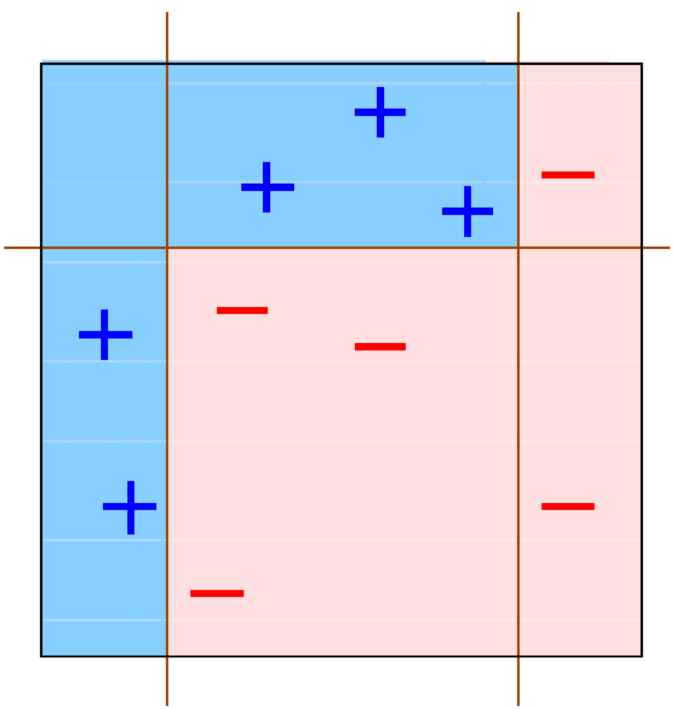
Round 2



Final Hypothesis

H_{final}

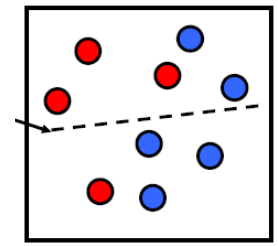
$= \text{sign}\Big($ 0.42  $+\ 0.65$  $+\ 0.92$  $\Big)$

$=$ 

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

AdaBoost Algorithm

Start with uniform weights on training examples



{x1,…xn}

For T rounds

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

Re-weight the examples:
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

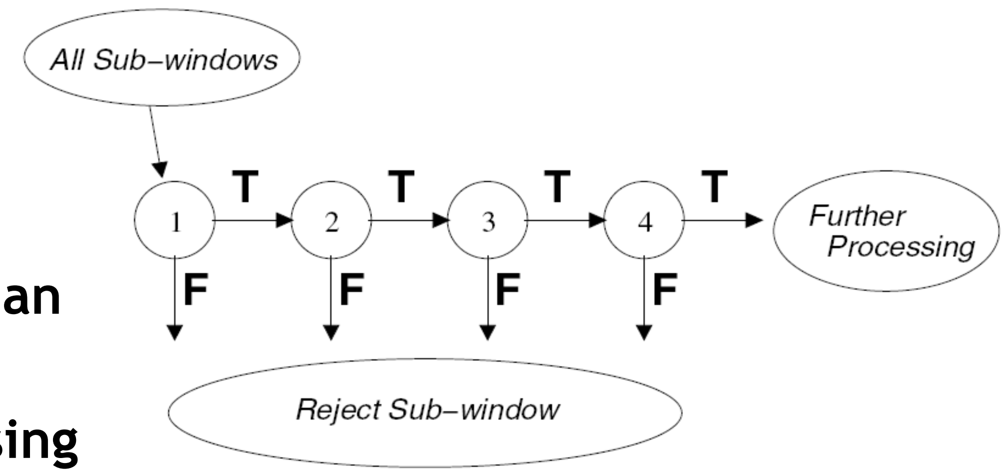
Learning to Detect

- Image Features + thresholds = Weak Classifiers
- For each round of Boosting
  - Evaluate each rectangle filter on each example
  - Sort examples by filter values
  - Select best threshold for each filter (min. err.)
    - Sorted list can be quickly scanned for optimal threshold
  - Select best filter / threshold combo.
  - Weight on this feature is a simple function of err. rate.
  - Rewight examples
- Output: selected features and their weights

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



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

Figure from Viola & Jones CVPR 2001

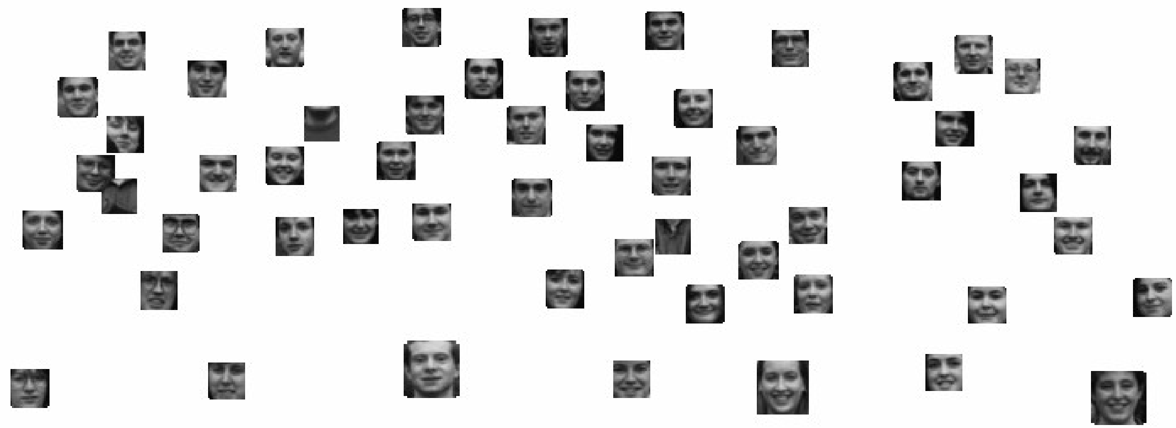
9.8 % patches remaining

0.74 % patches remaining

0.06 % patches remaining

0.01 % patches remaining

0.007 % patches remaining



Highlights

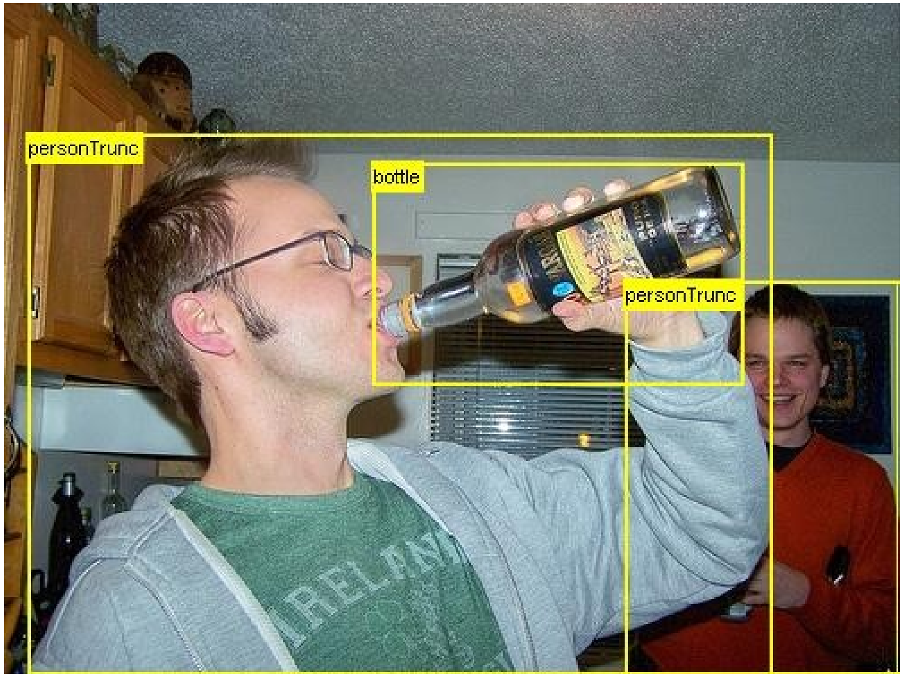
- **Sliding window detection and global appearance descriptors:**
  - Simple detection protocol to implement
  - Good feature choices critical
  - Past successes for certain classes

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)

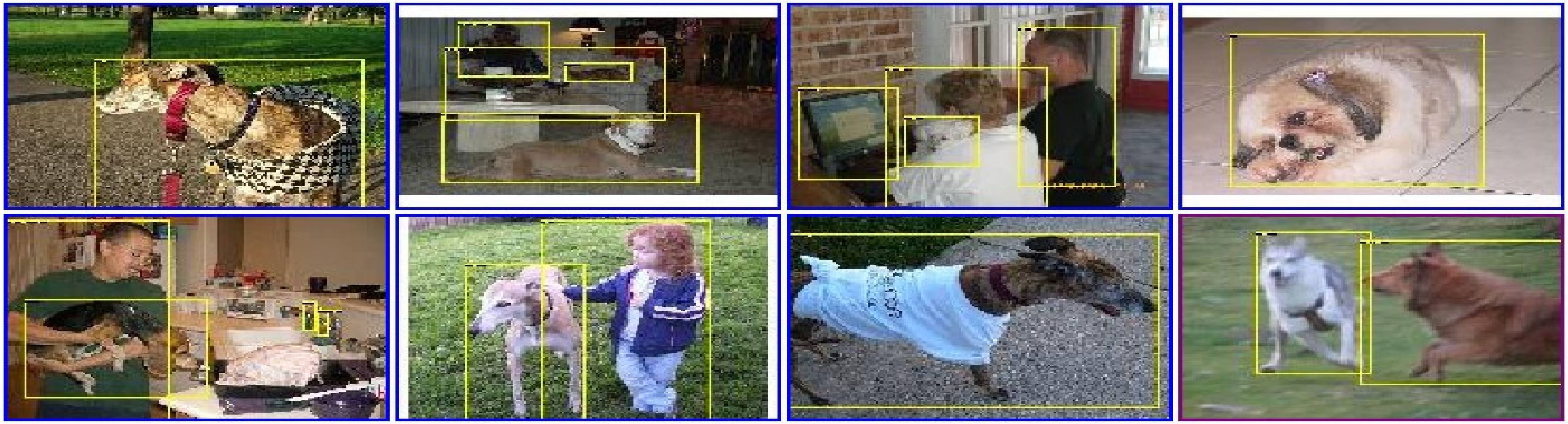
- **Not all objects are "box" shaped**



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

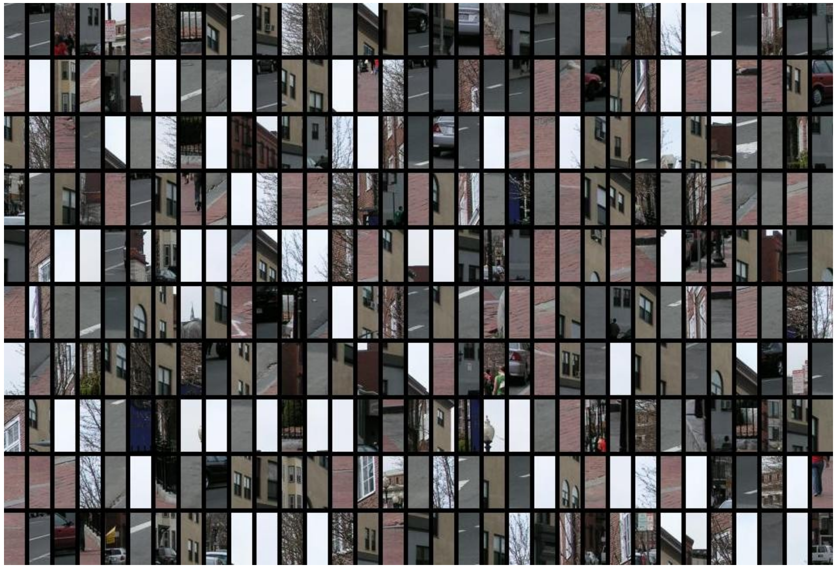
Dogs - *all images contain at least one dog.*



Limitations (continued)

- If considering windows in isolation, context is lost

Sliding window



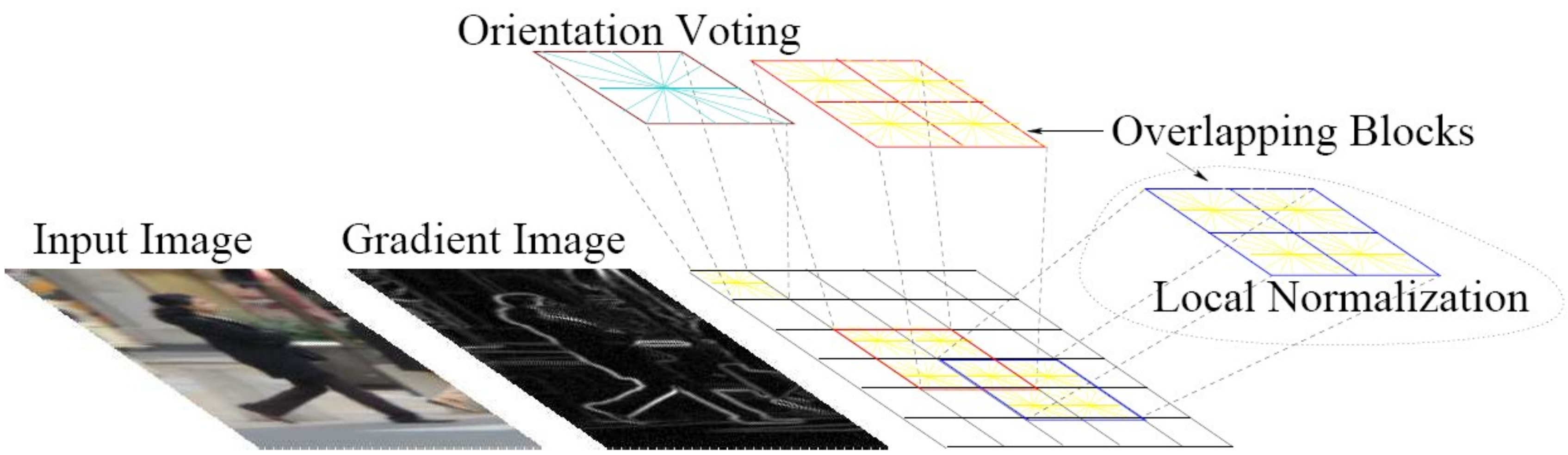
Detector's view

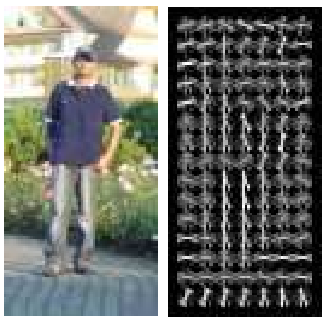
Limitations (continued)

- 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

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





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

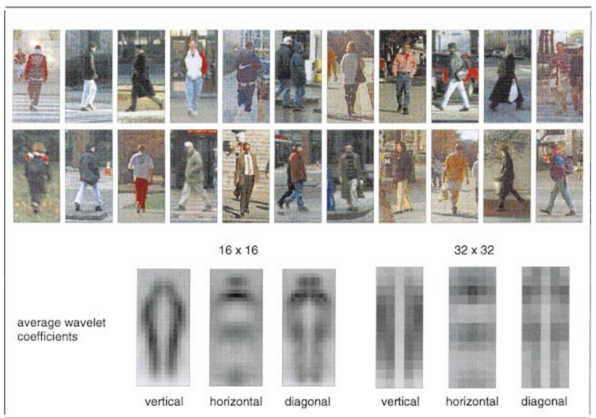
Code available: http://pascal.inrialpes.fr/soft/olt/

Dalal & Triggs, CVPR 2005

K. Grauman, B. Leibe

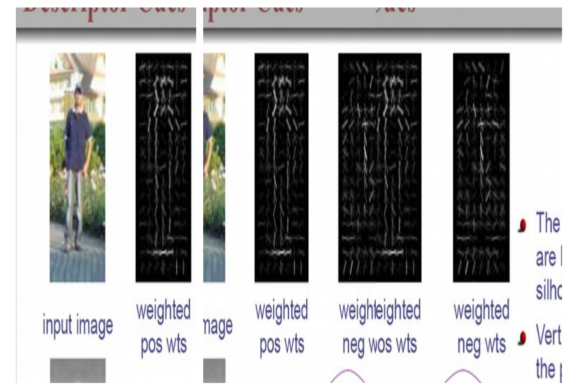
Pedestrian detection

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



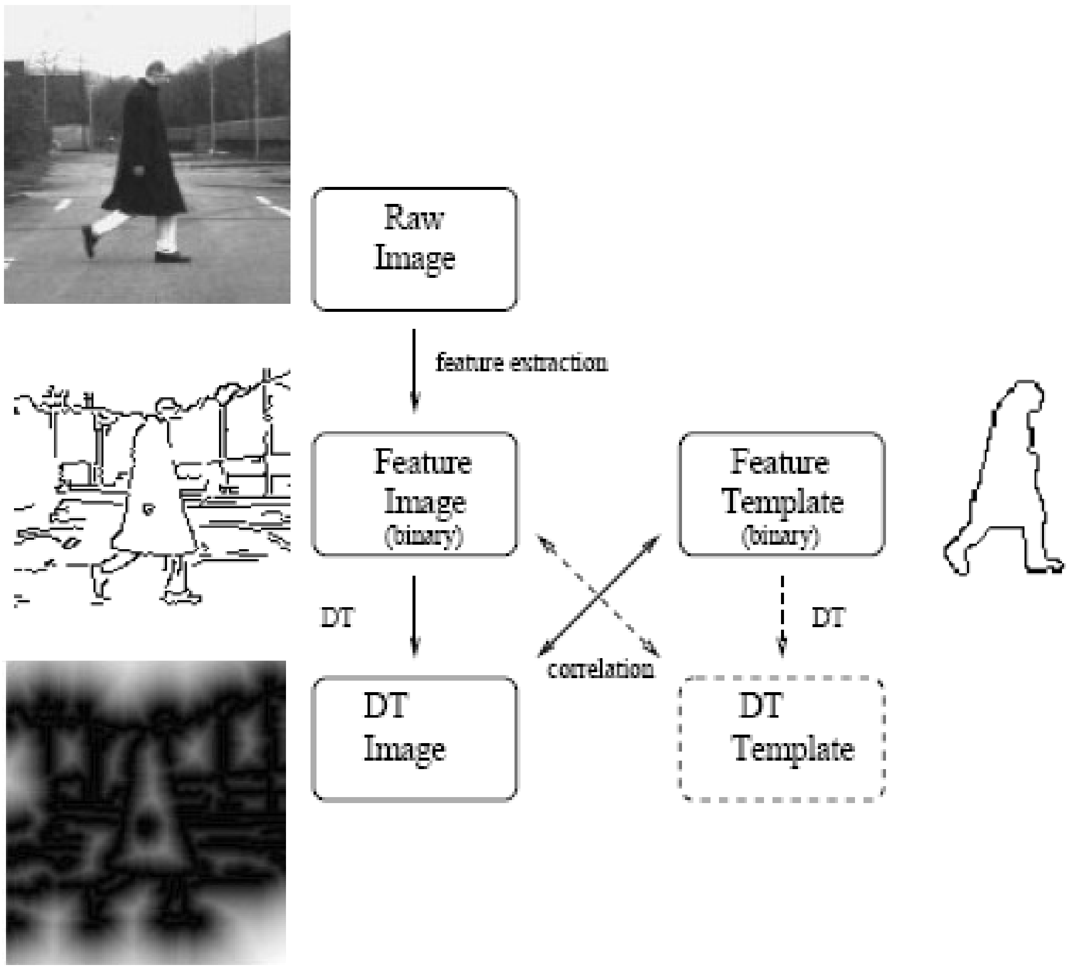
SVM with Haar wavelets [Papageorgiou & Poggio, IJCV 2000]

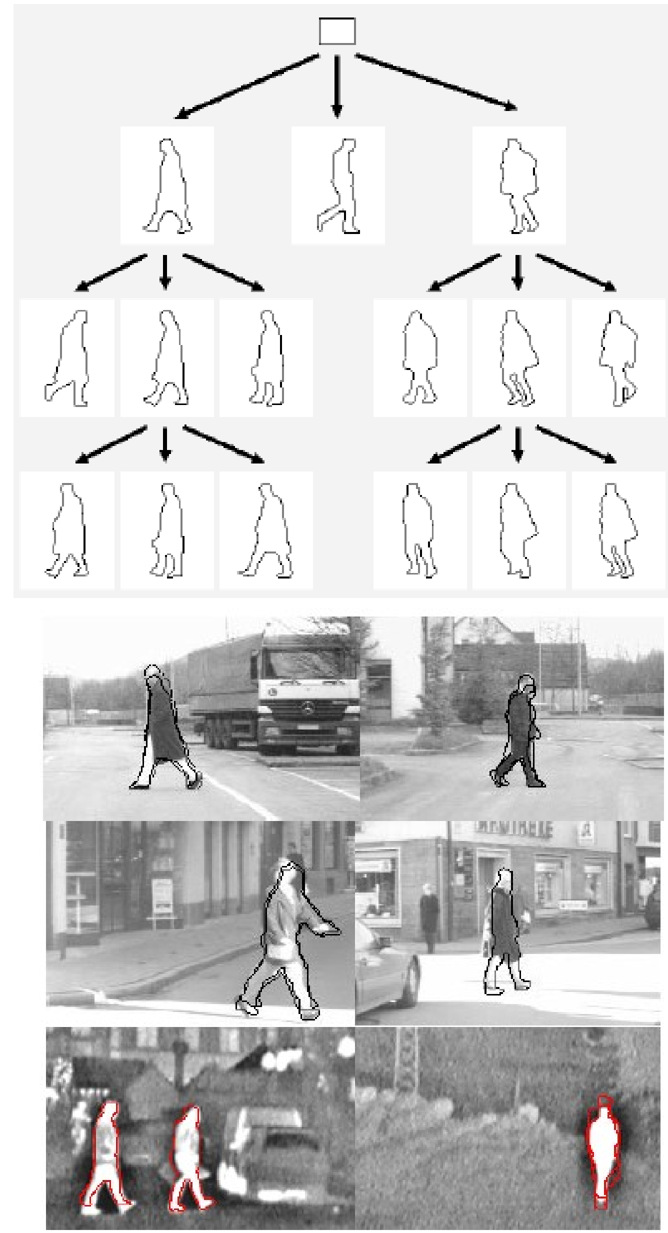
Space-time rectangle features [Viola, Jones & Snow, ICCV 2003]



SVM with HoGs [Dalal & Triggs, CVPR 2005]

Edges and chamfer distance

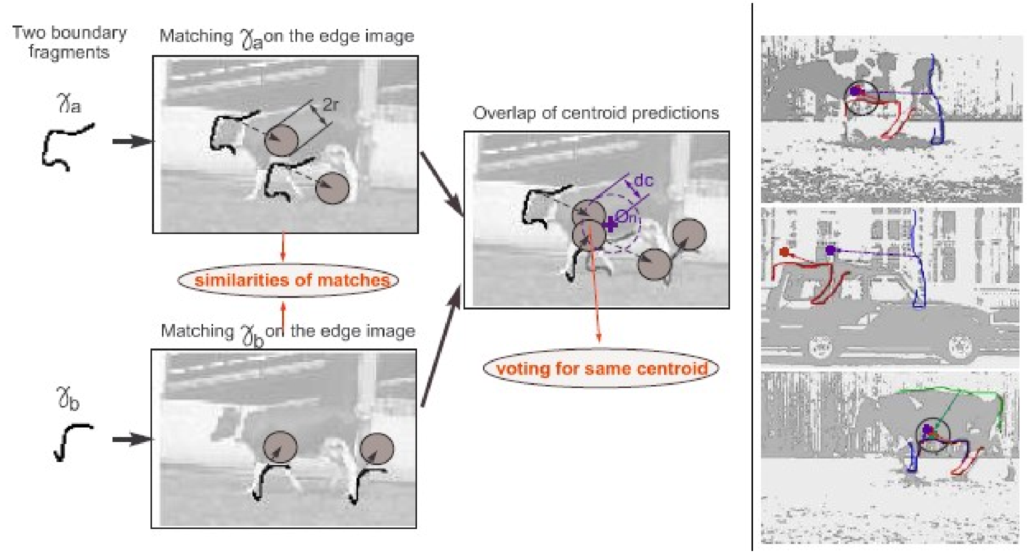




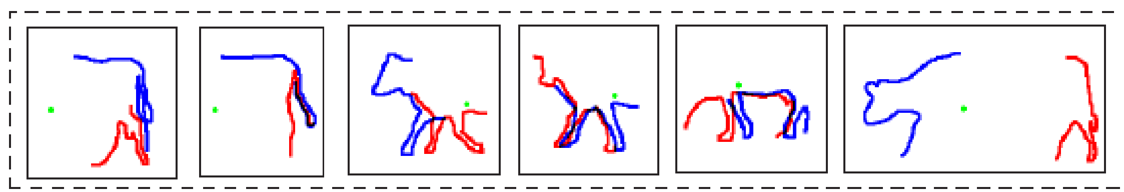
Gavrila, Philomin, ICCV 1999

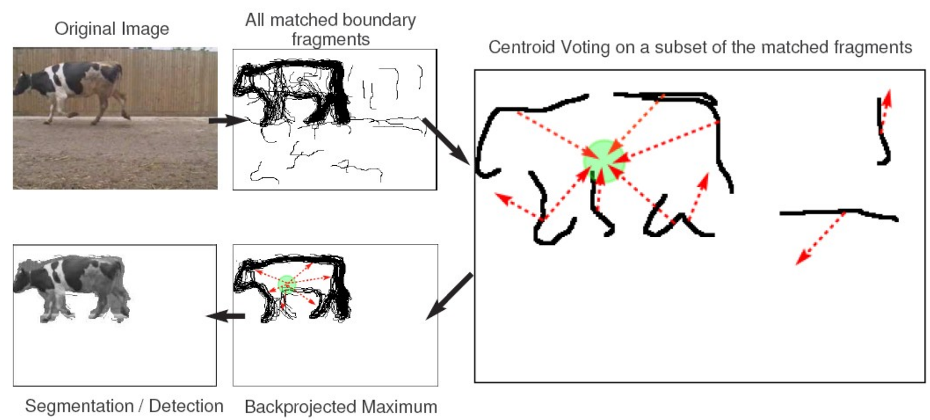
Edge fragments

Opelt, Pinz, Zisserman, ECCV 2006



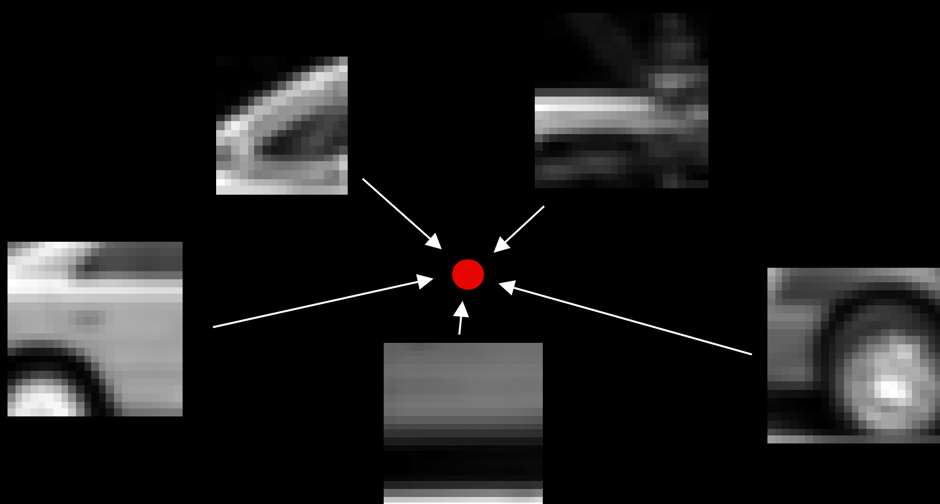
Weak detector = k edge fragments and threshold. Chamfer distance uses 8 orientation planes



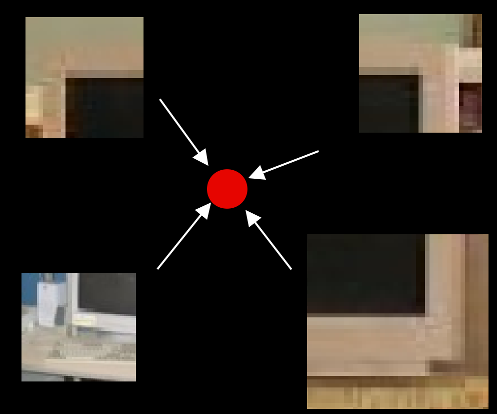


Weak detectors

Part based: similar to part-based generative models. We create weak detectors by using parts and voting for the object center location



Car model



Screen model

These features are used for the detector on the course web site.

סיכום

- זיהוי ע"י סיווג
- Boosting
- Viola & Jones

בפעם הבאה

- סינטוז טקסטורה
- גאומטריה של תמונות