Face detection

Slides adapted Grauman & Liebe's tutorial

- http://www.vision.ee.ethz.ch/~bleibe/teaching/tutorial-aaai08/

Also see Paul Viola's talk (video)

- http://www.cs.washington.edu/education/courses/577/04sp/contents.html#DM

Limitations of Eigenfaces

Eigenfaces are cool.

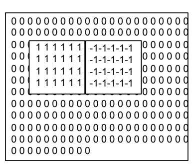
But they're not great for face detection.

Chief Limitations

- not very accurate
- not very fast

To make it work on the camera, we need ~30fps, and near-perfect accuracy.

Rectangle filters

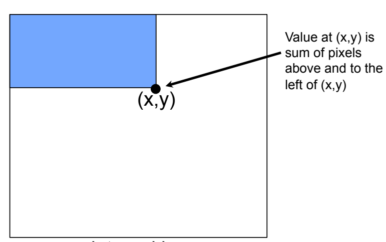


P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

Why rectangles?

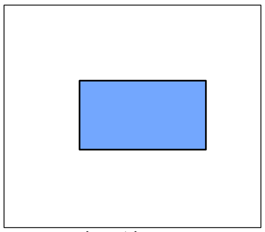
Answer: very very fast to compute

- Trick: integral images (aka summed-area-tables)

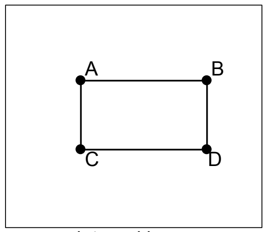


Integral images

What's the sum of pixels in the blue rectangle?

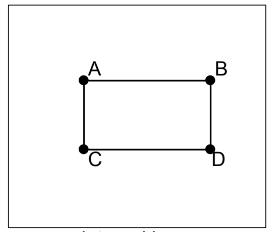


input image



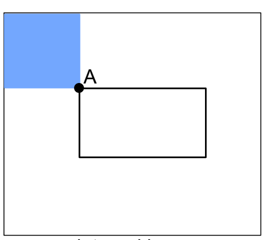
integral image

Integral images



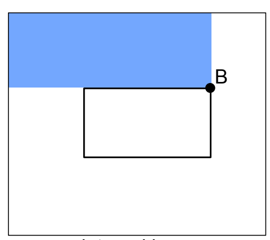
integral image

Integral images



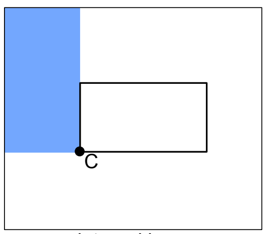
integral image

Integral images



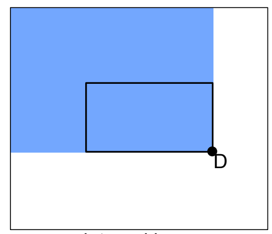
integral image

Integral images



integral image

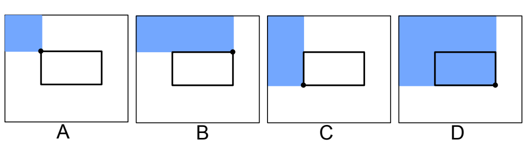
Integral images



integral image

Integral images

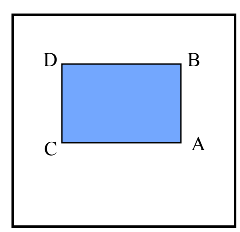
What's the sum of pixels in the rectangle?



Computing sum within a rectangle

- Let A,B,C,D be the values of the integral image at the corners of a rectangle
- Then the sum of original image values within the rectangle can be computed as:

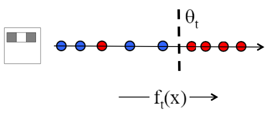
  sum = A – B – C + D
- Only 3 additions are required for any size of rectangle!



Lana Lazebnik

Filter as a classifier

How to convert the filter into a classifier?

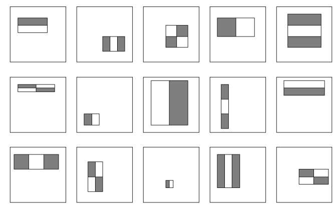


Outputs of a rectangle feature on faces and non-faces.

Resulting weak classifier:

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

Finding the best filters...



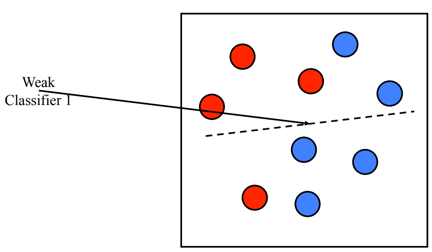
Considering all possible filter parameters: position, scale, and type:

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

Which of these filters(s) should we use to determine if a window has a face?

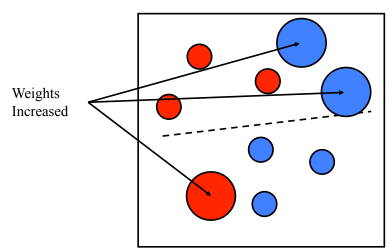
14

Boosting

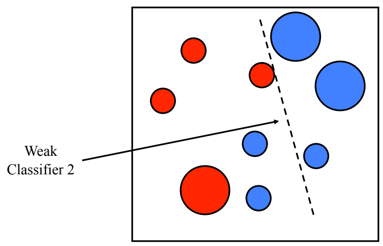


Slide credit: Paul Viola

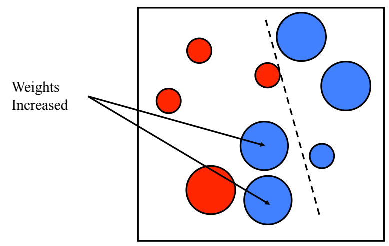
Boosting



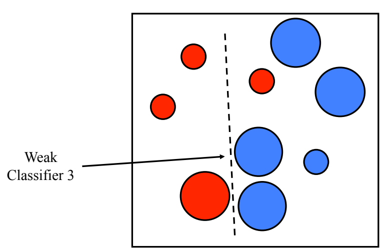
Boosting



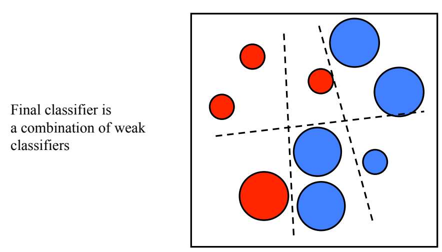
Boosting



Boosting



Boosting



Boosting: training

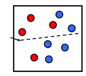
- Initially, weight each training example equally
- In each boosting round:
  - find the weak classifier with lowest weighted training error
  - raise weights of training examples misclassified by current weak classifier
- Final classifier is linear combination of all weak classifiers
  - weight of each learner is directly proportional to its accuracy)
- Exact formulas for re-weighting and combining weak

Slide credit: Lana Lazebnik

- Given example images $(x_1, y_1), \dots, (x_n, y_n)$ 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 $t = 1, \dots, T$:
  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



$\{x_1, \dots x_n\}$

For T rounds

Evaluate weighted error for each feature, pick best.

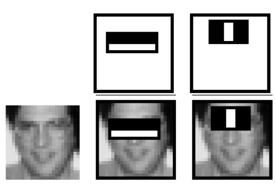
Re-weight the examples:
Incorrectly classified -> more weight
Correctly classified -> less weight

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

Freund & Schapire 1995

Visual Object Recognition Tutorial

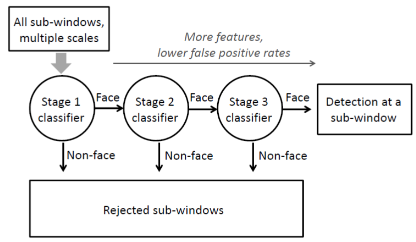
Viola-Jones Face Detector: Results



First two features selected

- Even if the filters are fast to compute, each new image has a lot of possible windows to search.
- How to make the detection more efficient?

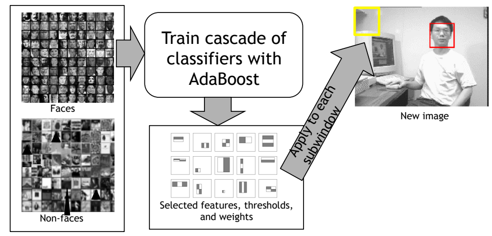
Cascading classifiers for detection



- Form a cascade with low false negative rates early on
- Apply less accurate but faster classifiers first to immediately discard windows that clearly appear to be negative

Kristen Grauman

Viola-Jones detector: summary



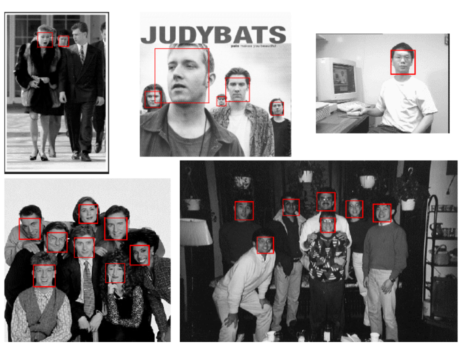
•Train with 5K positives, 350M negatives
•Real-time detector using 38 layer cascade
•6061 features in all layers

•[Implementation available in OpenCV: http://www.intel.com/technology/computing/opencv/]

Kristen Grauman

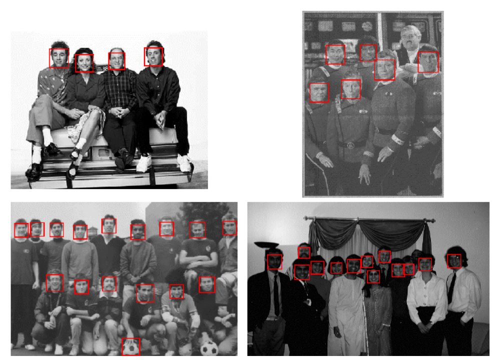
Viola-Jones Face Detector: Results

Visual Object Recognition Tutorial



Viola-Jones Face Detector: Results

Visual Object Recognition Tutorial

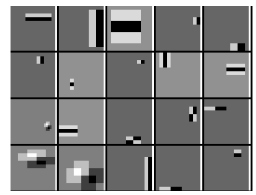


Viola-Jones Face Detector: Results

Visual Object Recognition Tutorial

Detecting profile faces?

Can we use the same detector?



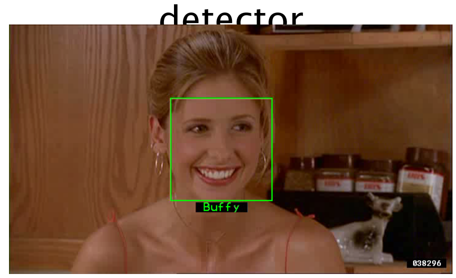
Visual Object Recognition Tutorial

Viola-Jones Face Detector: Results

Visual Object Recognition Tutorial

Paul

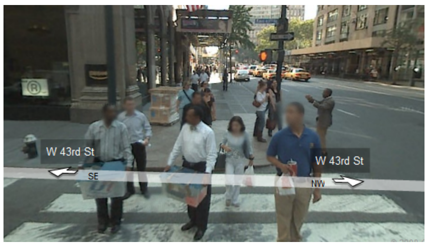
Example using Viola-Jones detector



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

Application: streetview



Application: streetview

34

Application: streetview

35

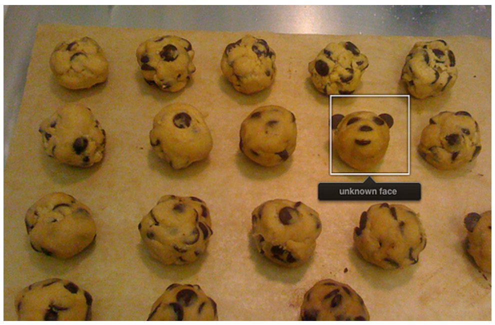
Consumer application: iPhoto 2009

http://www.apple.com/ilife/iphoto/

Slide credit: Lana Lazebnik

Consumer application: iPhoto 2009

[Things iPhoto thinks are faces]



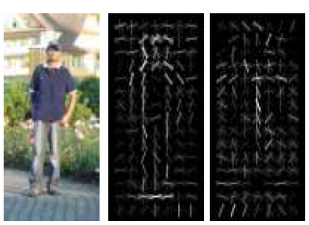
Slide credit: Lana Lazebnik

What other categories are amenable to window-based representation?

Visual Object Recognition Tutorial

Pedestrian detection

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



SVM with HoG [Dalal & Triggs, CVPR 2005]

Kristen Grauman

Visual Object Recognition Tutorial

Window-based detection: strengths

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

Kristen Grauman

Window-based detection: Limitations

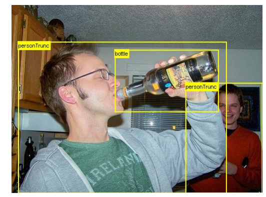
- High computational complexity
  - For example: 250,000 locations x 30 orientations x 4 scales = 30,000,000 evaluations!
  - If training binary detectors independently, means cost increases linearly with number of classes
- With so many windows, false positive rate better be low

Visual Object Recognition Tutorial

Kristen Grauman

Limitations (continued)

- Not all objects are “box” shaped

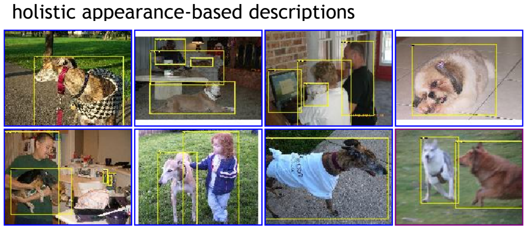


Visual Object Recognition Tutorial

Kristen Grauman

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



Visual Object Recognition Tutorial

Kristen Grauman

Limitations (continued)

- If considering windows in isolation, context is lost

Sliding window

Detector’s view

Visual Object Recognition Tutorial

Figure credit: Derek Hoiem

Kristen Grauman

Visual Object Recognition Tutorial

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

Kristen Grauman