

note:
black & white



Object Category Detection: Sliding Windows

Computer Vision
CS 143, Brown

James Hays

Previously

- Category recognition (proj3)
 - Bag of words over *not-so-invariant* local features.
- Instance recognition
 - Local invariant features: interest point detection and feature description
 - Local feature matching, spatial verification
 - Scalable indexing

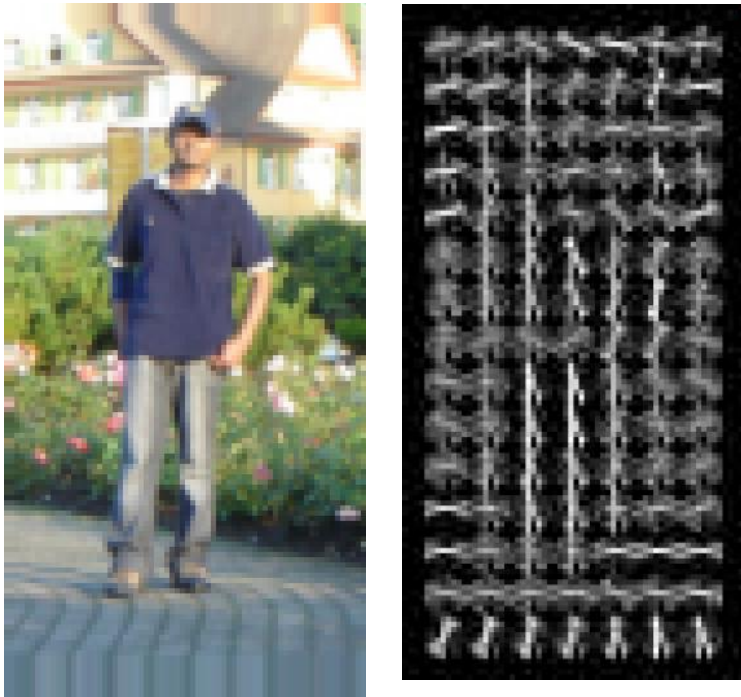
Today

- Window-based generic object detection
 - basic pipeline
 - boosting classifiers
 - face detection as case study

Generic category recognition: basic framework

- Build/train object model
 - Choose a representation
 - Learn or fit parameters of model / classifier
- Generate candidates in new image
- Score the candidates

Generic category recognition: representation choice



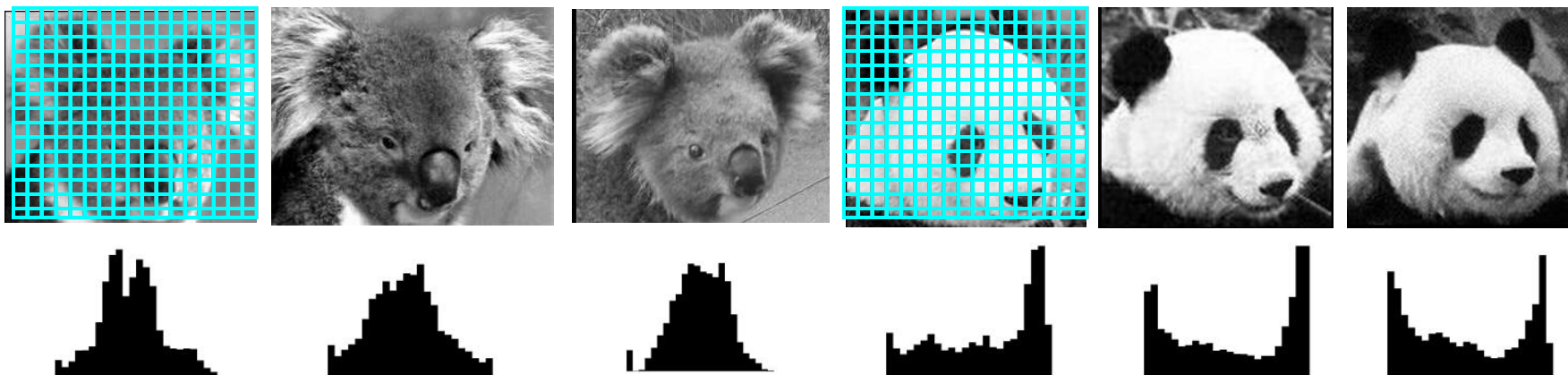
Window-based



Part-based

Window-based models

Building an object model



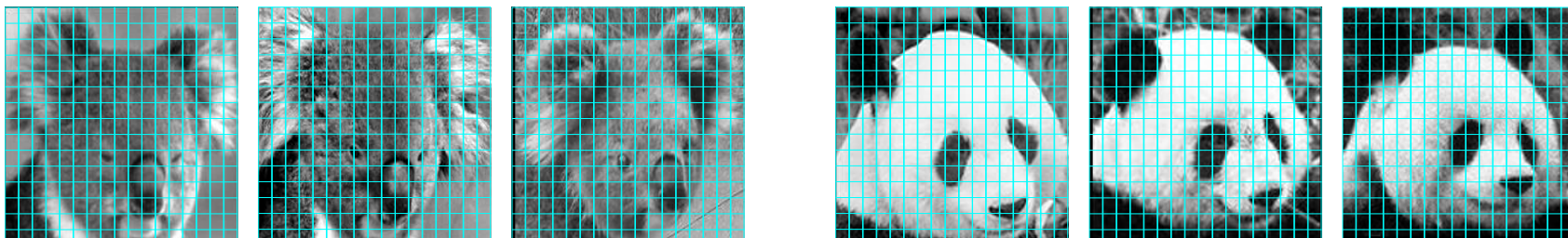
Simple holistic descriptions of image content

- grayscale / color histogram
- vector of pixel intensities

Window-based models

Building an object model

- Pixel-based representations sensitive to small shifts

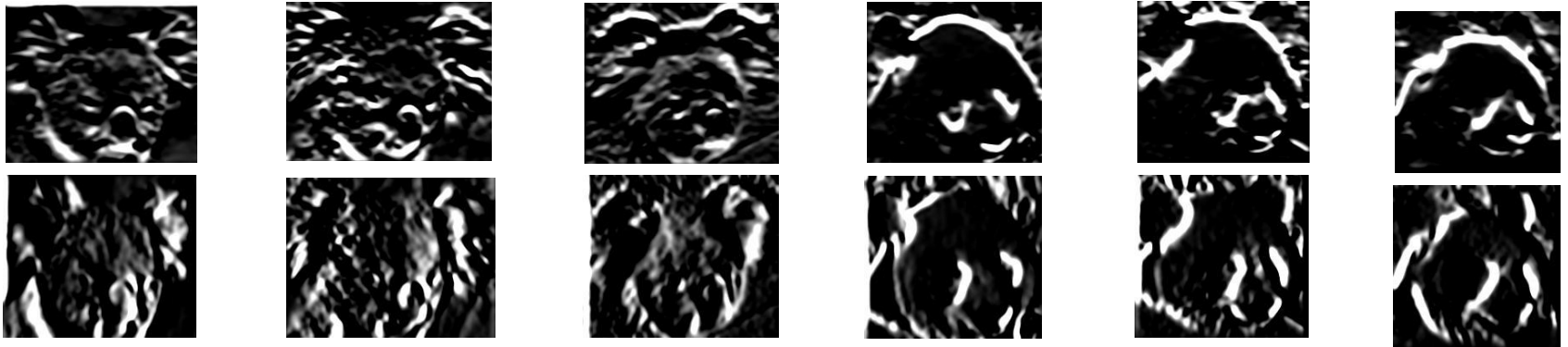


- Color or grayscale-based appearance description can be sensitive to illumination and intra-class appearance variation

Window-based models

Building an object model

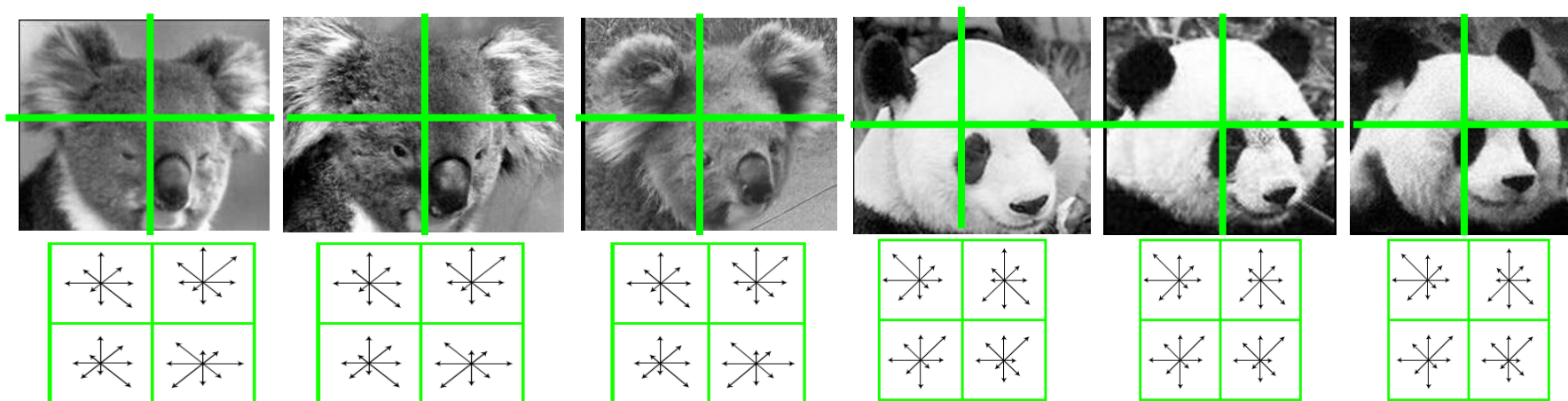
- Consider edges, contours, and (oriented) intensity gradients



Window-based models

Building an object model

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

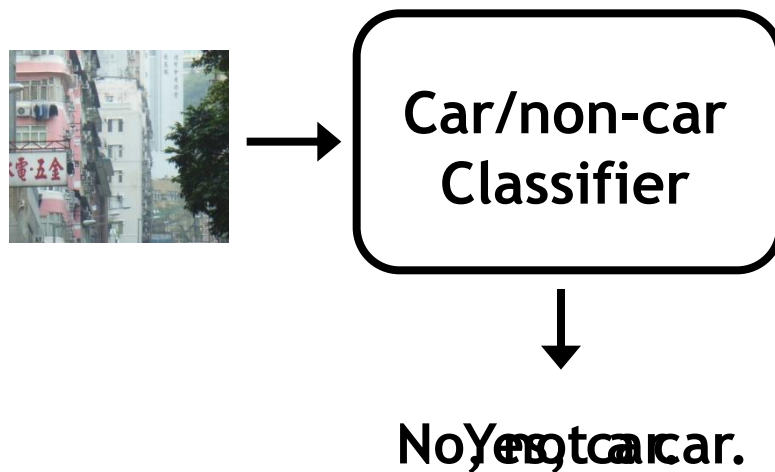


- Summarize local distribution of gradients with histogram
 - Locally orderless: offers invariance to small shifts and rotations
 - Contrast-normalization: try to correct for variable illumination

Window-based models

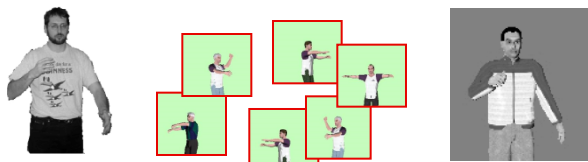
Building an object model

Given the representation, train a binary classifier



Discriminative classifier construction

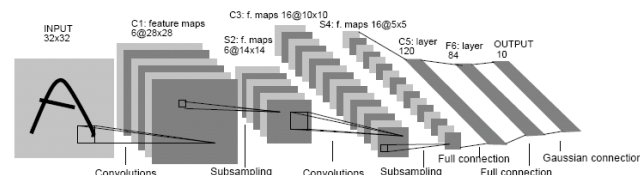
Nearest neighbor



10^6 examples

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

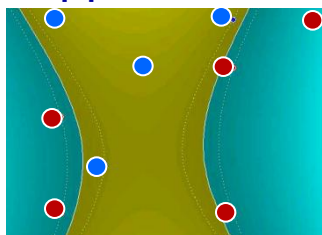
Neural networks



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

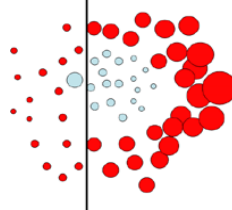
...

Support Vector Machines



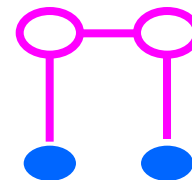
Guyon, Vapnik
Heisele, Serre, Poggio,
2001,...

Boosting



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

Conditional Random Fields



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

Influential Works in Detection

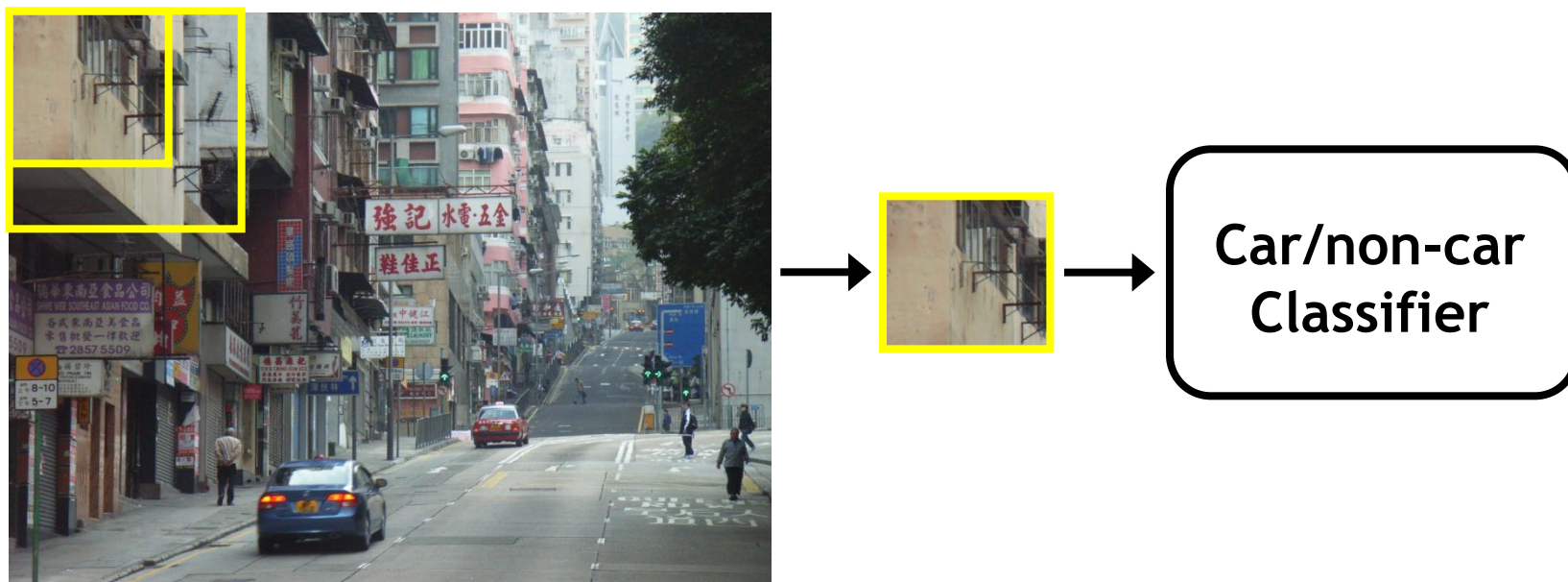
- Sung-Poggio (1994, 1998) : ~1450 citations
 - Basic idea of statistical template detection (I think), bootstrapping to get “face-like” negative examples, multiple whole-face prototypes (in 1994)
- Rowley-Baluja-Kanade (1996-1998) : ~2900
 - “Parts” at fixed position, non-maxima suppression, simple cascade, rotation, pretty good accuracy, fast
- Schneiderman-Kanade (1998-2000,2004) : ~1250
 - Careful feature engineering, excellent results, cascade
- Viola-Jones (2001, 2004) : ~6500
 - Haar-like features, Adaboost as feature selection, hyper-cascade, very fast, easy to implement
- Dalal-Triggs (2005) : ~2000
 - Careful feature engineering, excellent results, HOG feature, online code
- Felzenszwalb-Huttenlocher (2000): ~800
 - Efficient way to solve part-based detectors
- Felzenszwalb-McAllester-Ramanan (2008)? ~350
 - Excellent template/parts-based blend

Generic category recognition: basic framework

- Build/train object model
 - Choose a representation
 - Learn or fit parameters of model / classifier
- **Generate candidates in new image**
- **Score the candidates**

Window-based models

Generating and scoring candidates



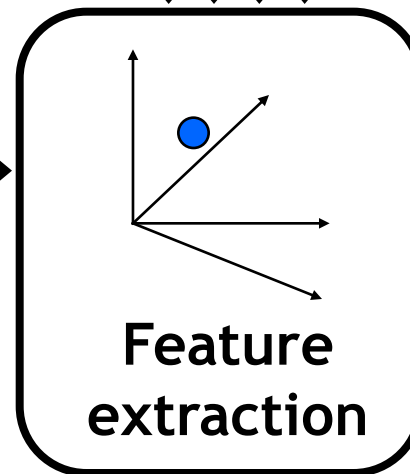
Window-based object detection: recap

Training:

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

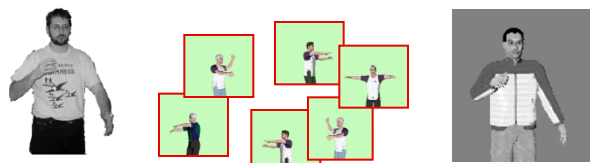
Given new image:

1. Slide window
2. Score by classifier



Discriminative classifier construction

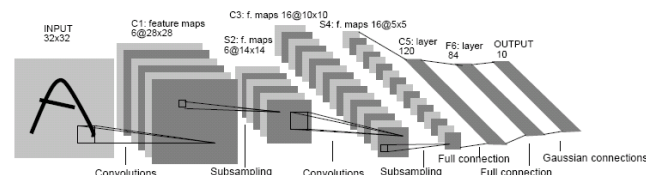
Nearest neighbor



10^6 examples

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

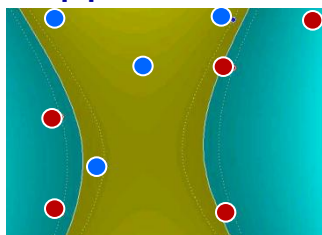
Neural networks



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

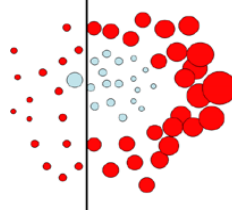
...

Support Vector Machines



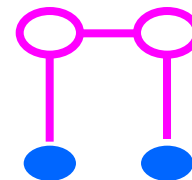
Guyon, Vapnik
Heisele, Serre, Poggio,
2001,...

Boosting



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

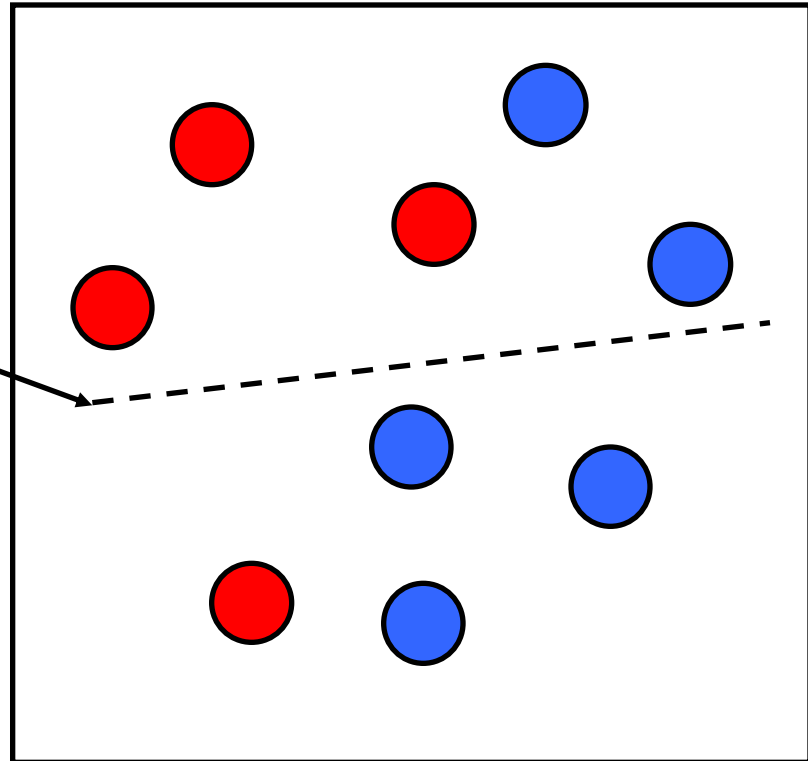
Conditional Random Fields



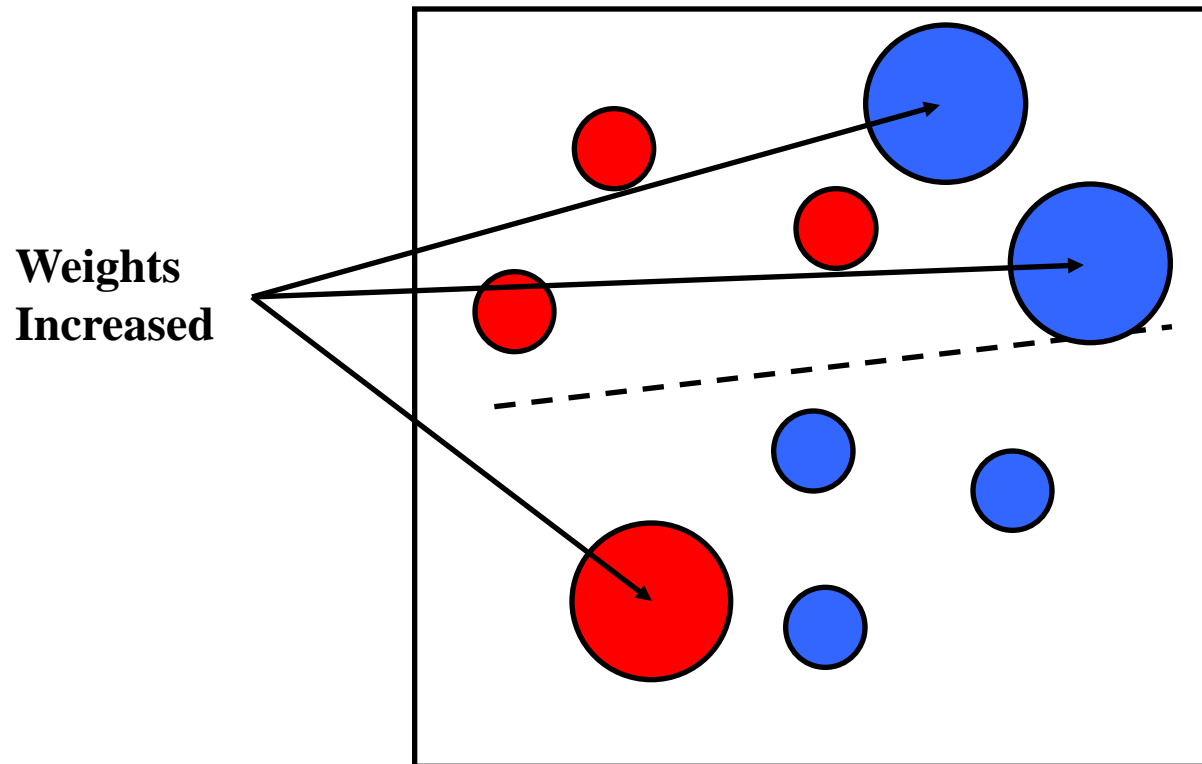
McCallum, Freitag, Pereira
2000; Kumar, Hebert 2003
...

Boosting intuition

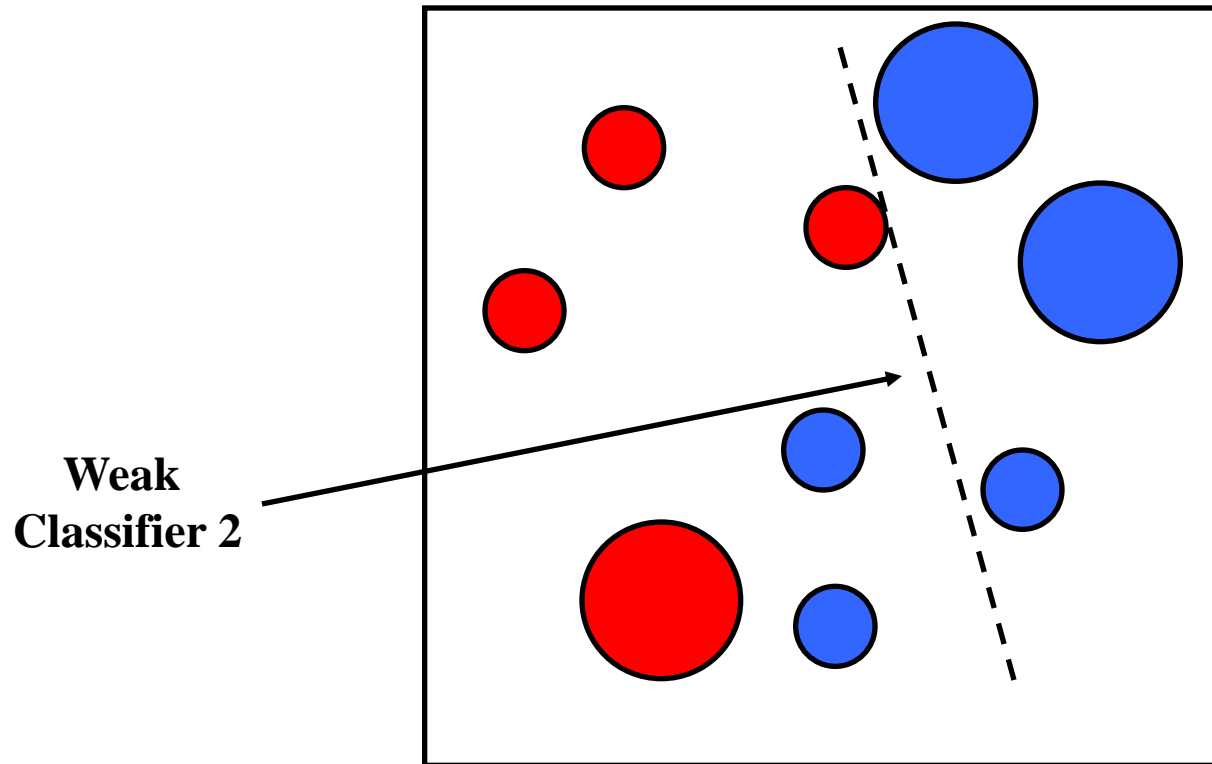
**Weak
Classifier 1**



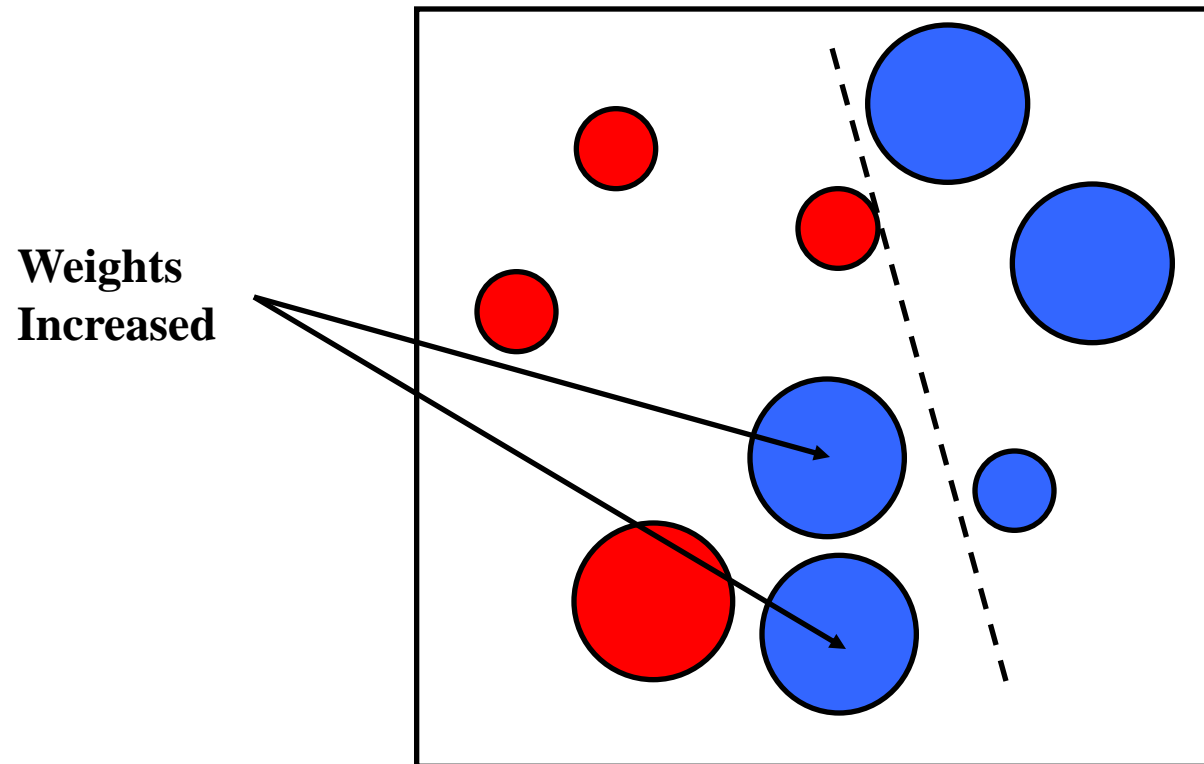
Boosting illustration



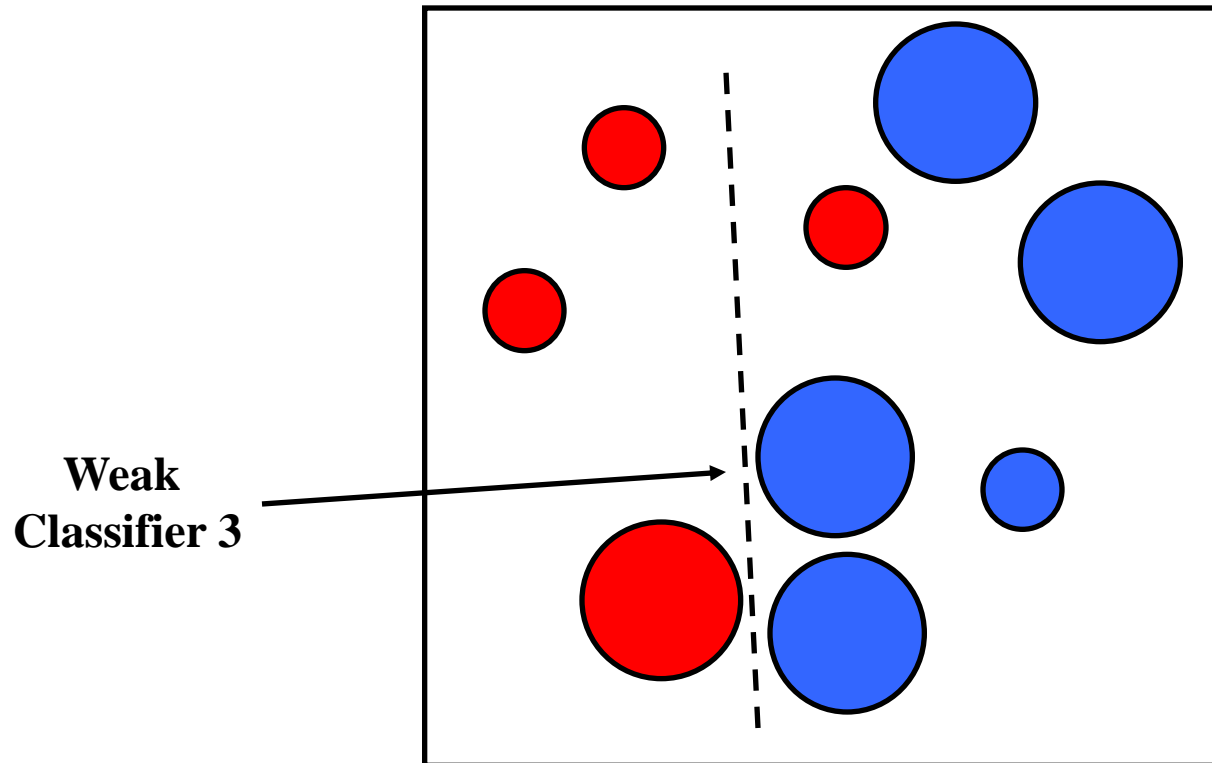
Boosting illustration



Boosting illustration

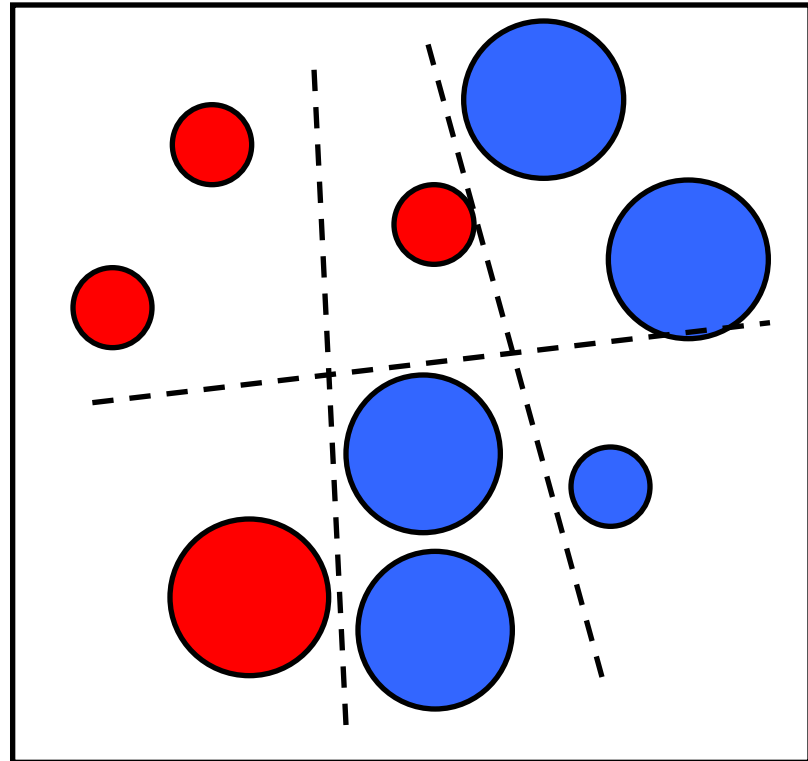


Boosting illustration



Boosting illustration

**Final classifier is
a combination of weak
classifiers**



Boosting: training

- Initially, weight each training example equally
- In each boosting round:
 - Find the weak learner that achieves the lowest *weighted* training error
 - Raise weights of training examples misclassified by current weak learner
- Compute final classifier as linear combination of all weak learners (weight of each learner is directly proportional to its accuracy)
- Exact formulas for re-weighting and combining weak learners depend on the particular boosting scheme (e.g., AdaBoost)

Boosting: pros and cons

- Advantages of boosting
 - Integrates classification with feature selection
 - Complexity of training is linear in the number of training examples
 - Flexibility in the choice of weak learners, boosting scheme
 - Testing is fast
 - Easy to implement
- Disadvantages
 - Needs many training examples
 - Often found not to work as well as an alternative discriminative classifier, support vector machine (SVM)
 - especially for many-class problems

Viola-Jones face detector

ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION 2001

Rapid Object Detection using a Boosted Cascade of Simple Features

Paul Viola

viola@merl.com

Mitsubishi Electric Research Labs

201 Broadway, 8th FL

Cambridge, MA 02139

Michael Jones

mjones@crl.dec.com

Compaq CRL

One Cambridge Center

Cambridge, MA 02142

Abstract

This paper describes a machine learning approach for vi-

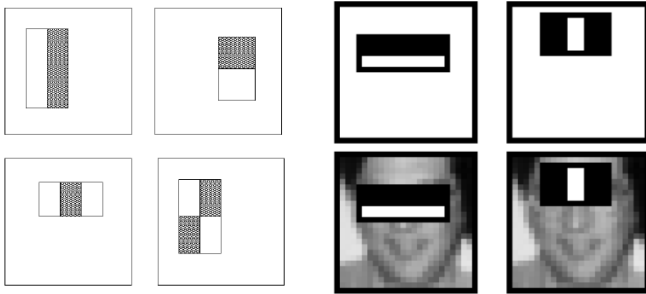
tected at 15 frames per second on a conventional 700 MHz Intel Pentium III. In other face detection systems, auxiliary information, such as image differences in video sequences,

Viola-Jones face detector

Main idea:

- Represent local texture with efficiently computable “rectangular” features within window of interest
- Select discriminative features to be weak classifiers
- Use boosted combination of them as final classifier
- Form a cascade of such classifiers, rejecting clear negatives quickly

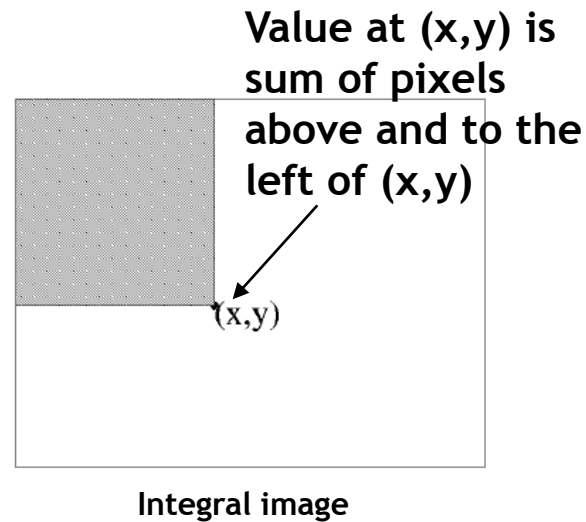
Viola-Jones detector: features



“Rectangular” filters

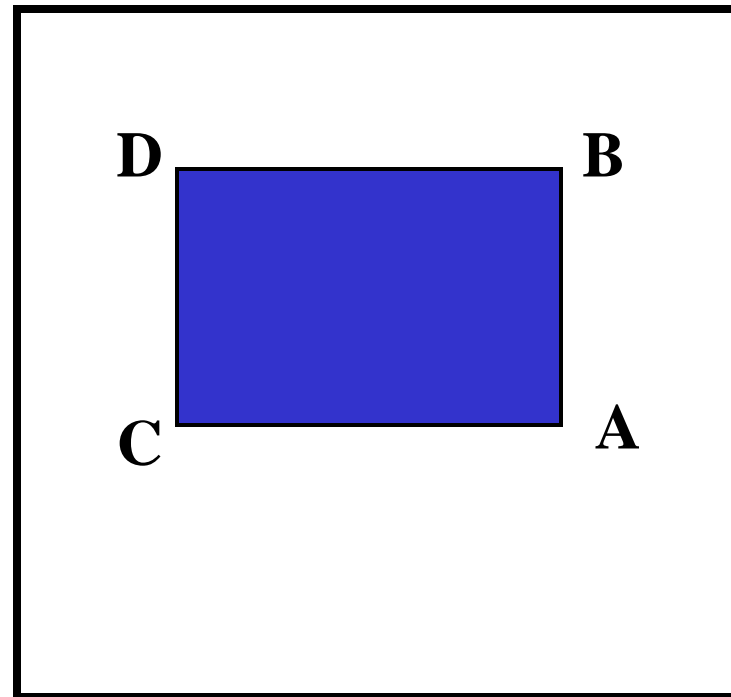
Feature output is difference between adjacent regions

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

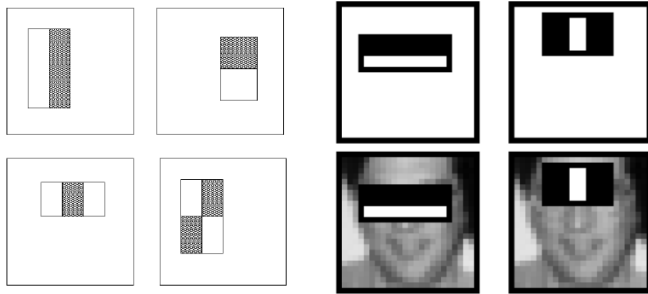


Computing sum within a rectangle

- 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:
$$\text{sum} = A - B - C + D$$
- Only 3 additions are required for any size of rectangle!



Viola-Jones detector: features

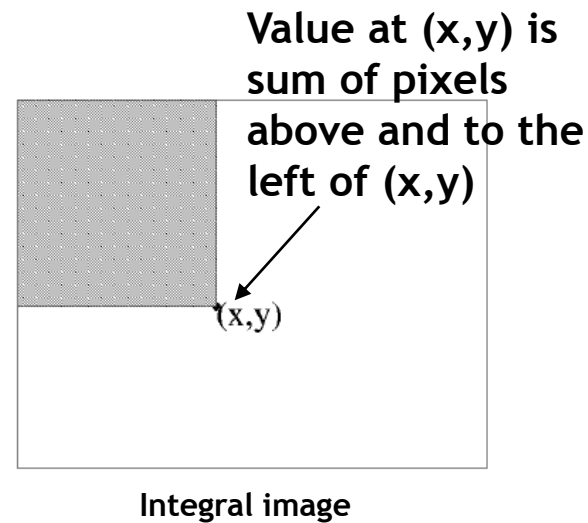


“Rectangular” filters

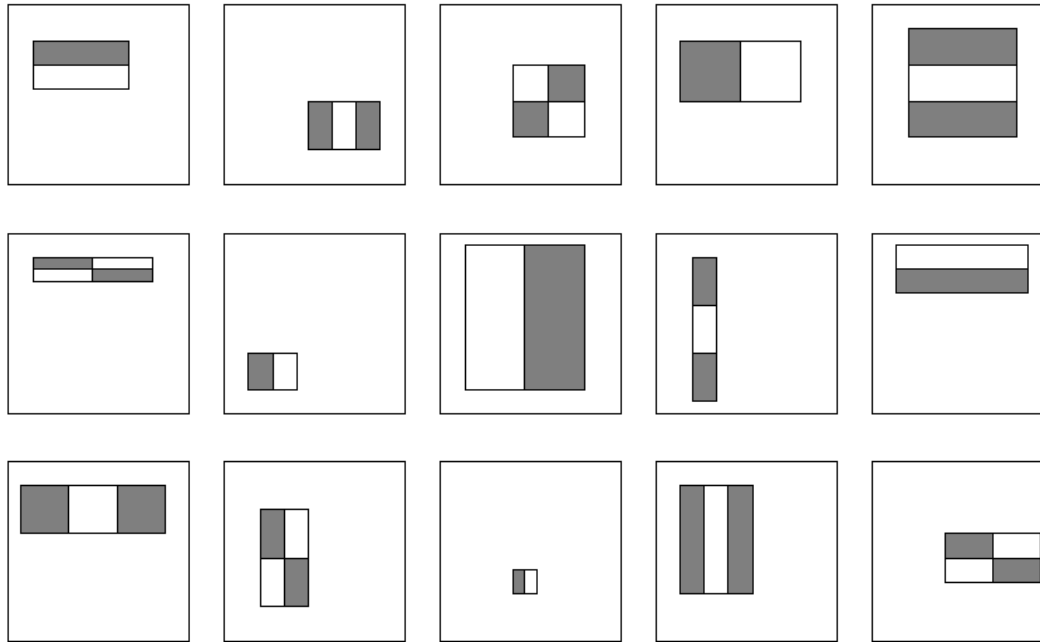
Feature output is difference between adjacent regions

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

Avoid scaling images →
scale features directly
for same cost



Viola-Jones detector: features



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

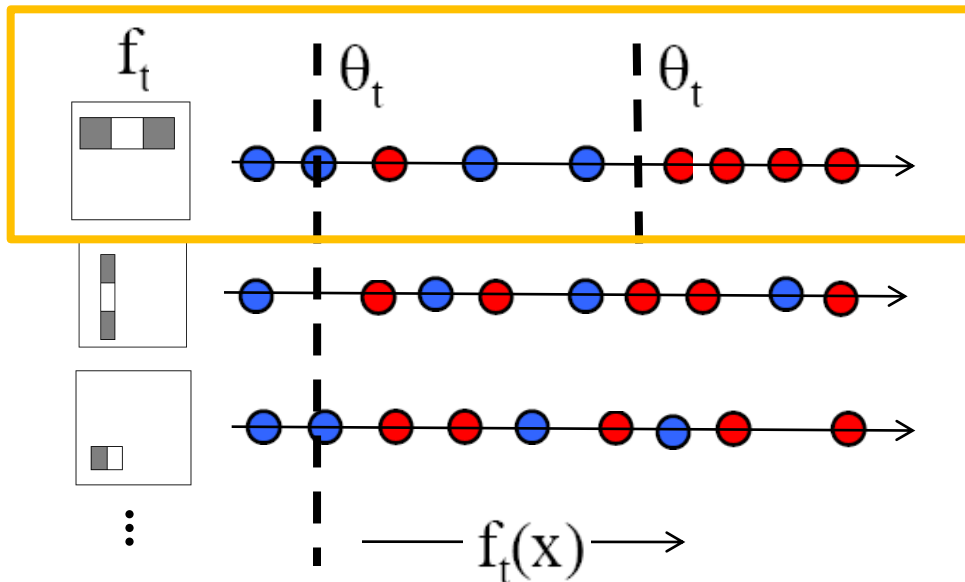
180,000+ possible features associated with each 24 x 24 window

Which subset of these features should we use to determine if a window has a face?

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


Viola-Jones detector: AdaBoost

- 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.

- 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 $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

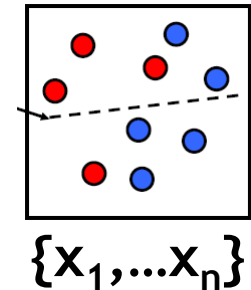
- The final strong classifier is:

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

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

AdaBoost Algorithm

Start with
uniform weights
on training
examples



For T rounds

← Evaluate
weighted error
for each feature,
pick best.

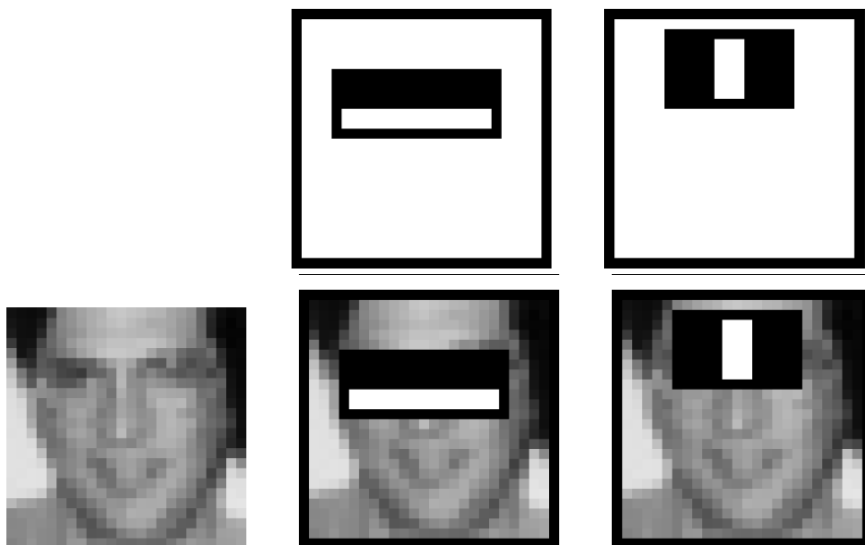
Re-weight the examples:

← 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

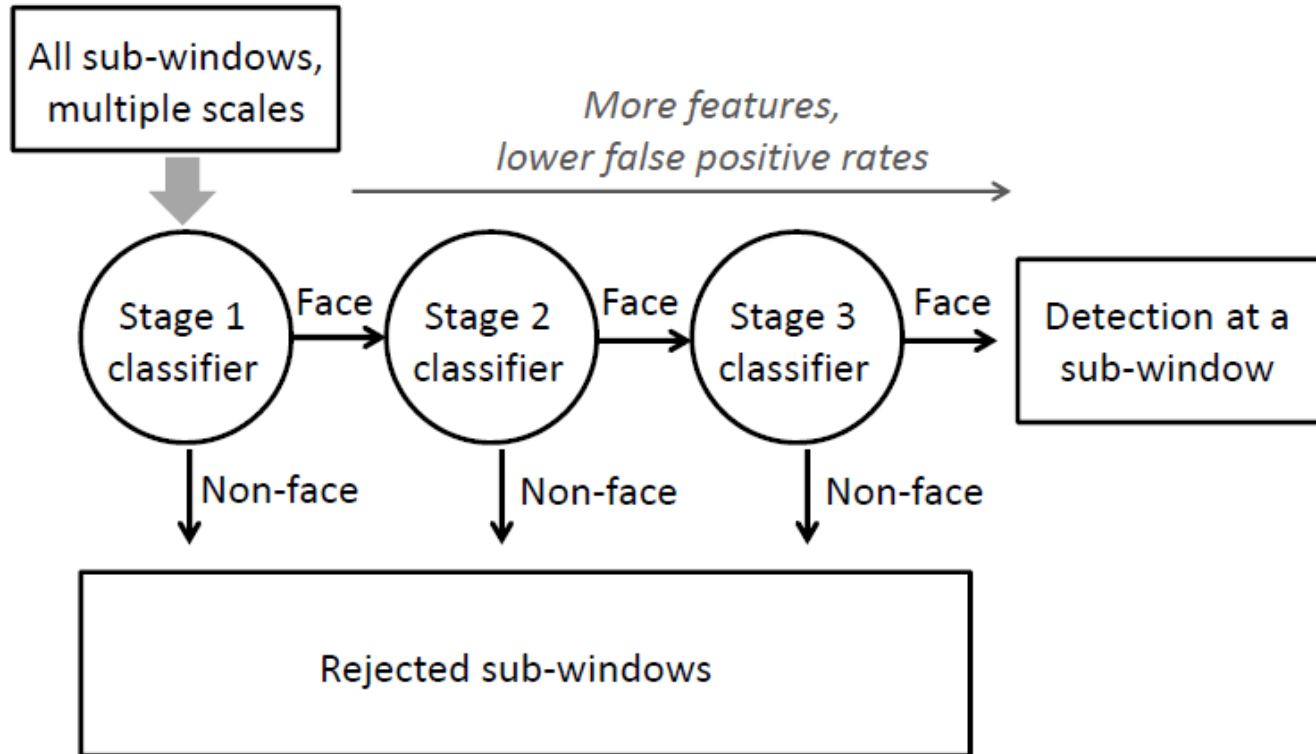
Viola-Jones Face Detector: Results



First two features
selected

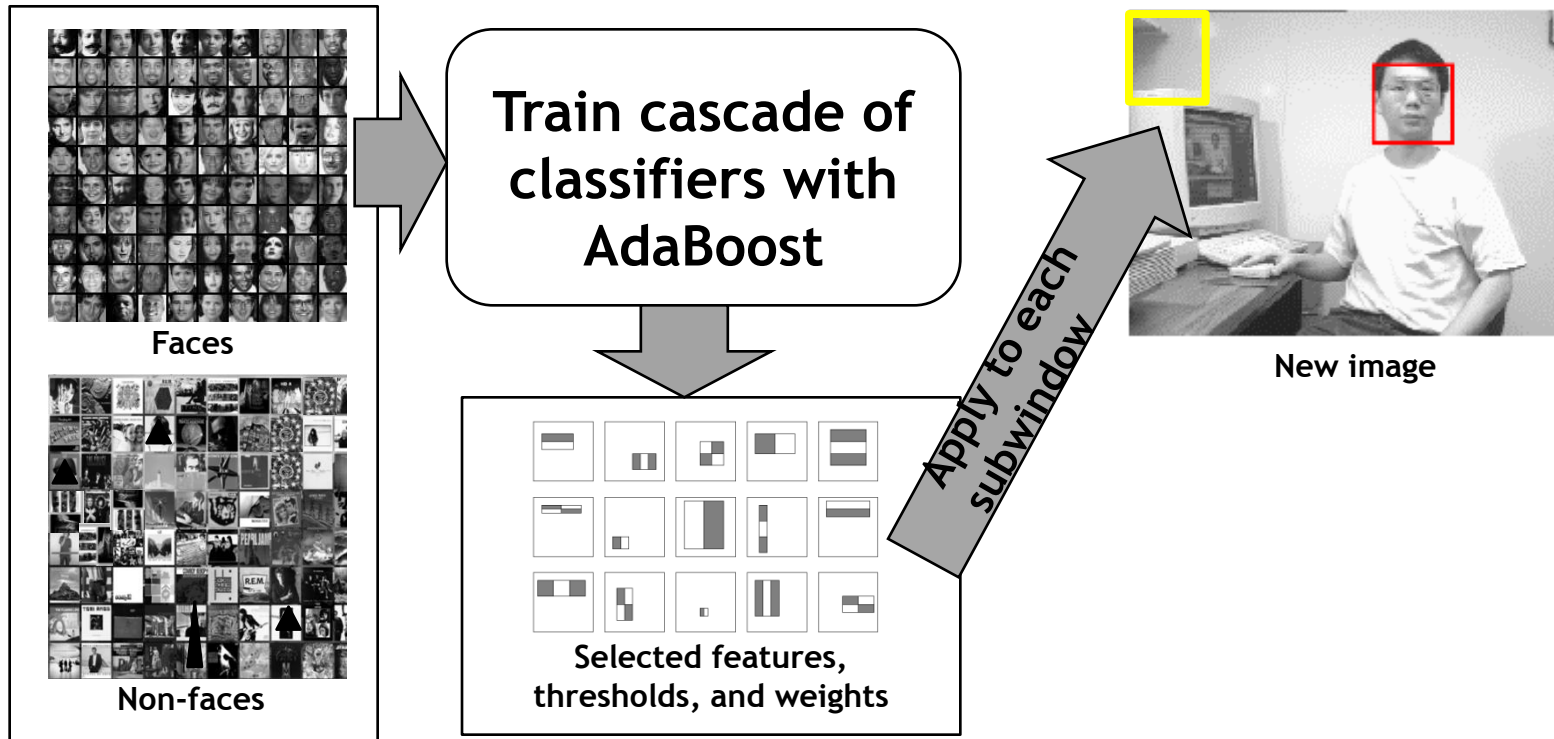
- Even if the filters are fast to compute, each new image has a lot of possible windows to search.
- How to make the detection more efficient?

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

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

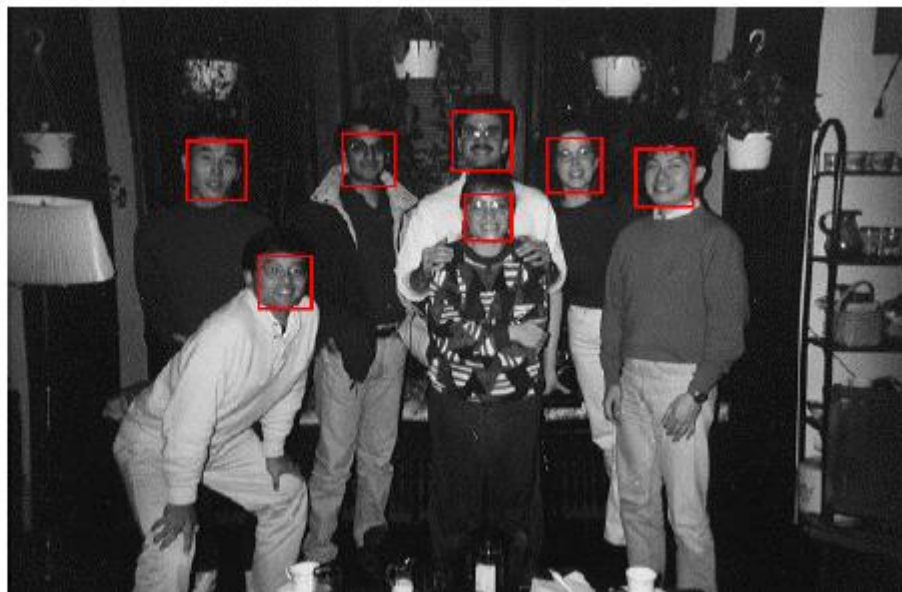
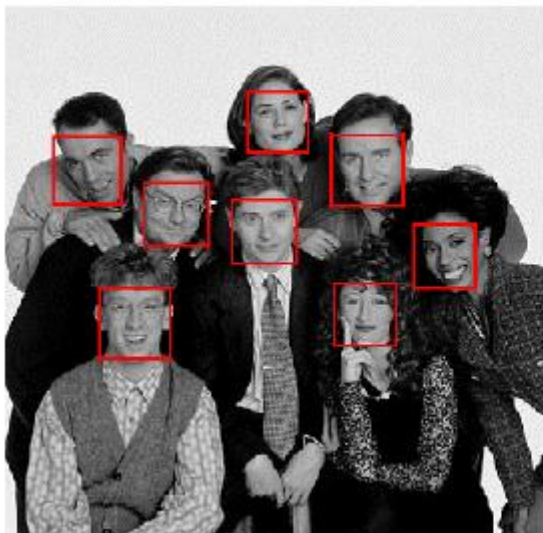
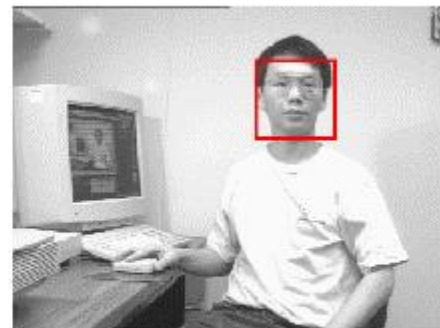
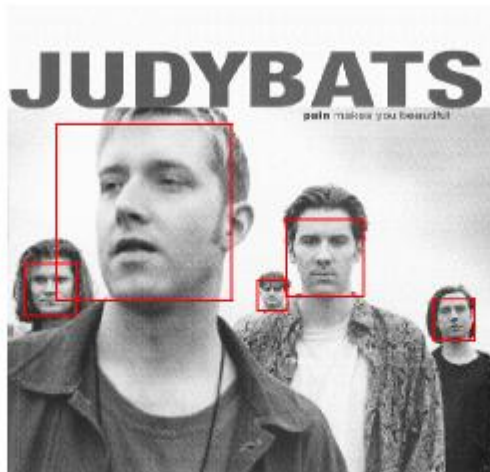
Viola-Jones detector: summary

- A seminal approach to real-time object detection
- Training is slow, but detection is very fast
- Key ideas
 - Features which can be evaluated very quickly with *Integral Images*
 - Cascade model which rejects unlikely faces quickly
 - Mining hard negatives

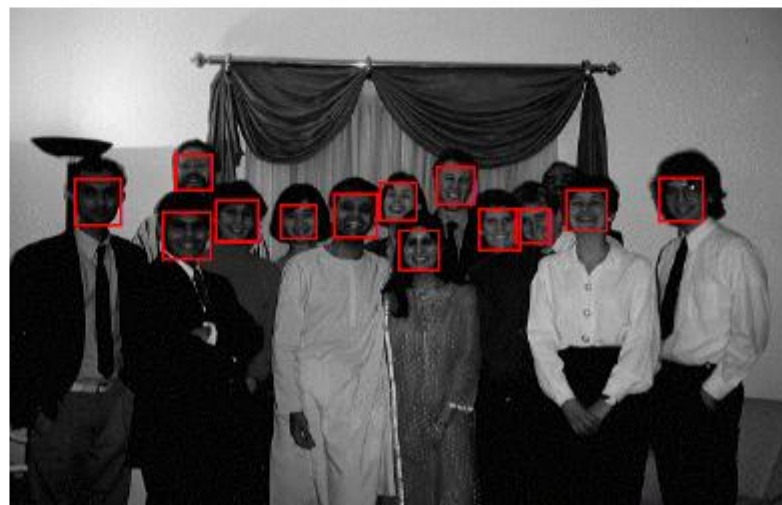
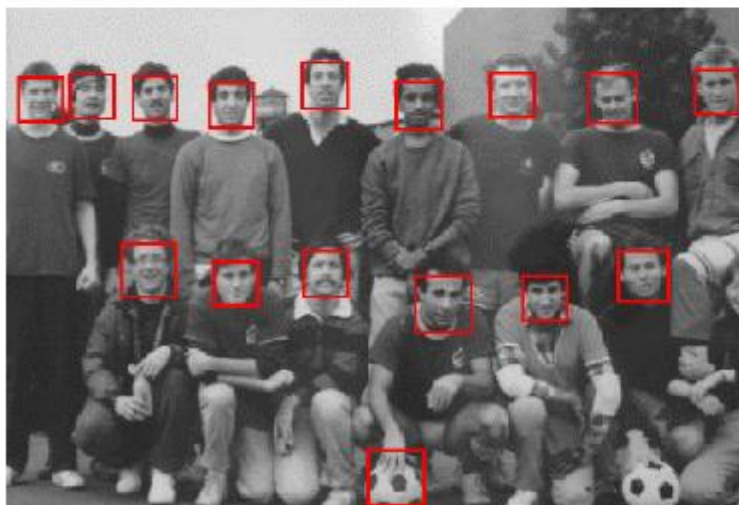
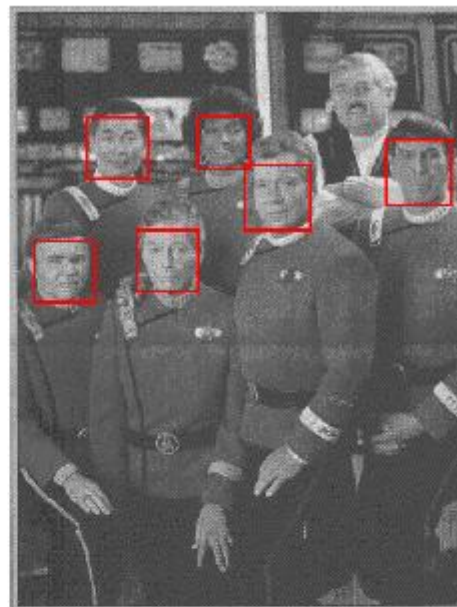
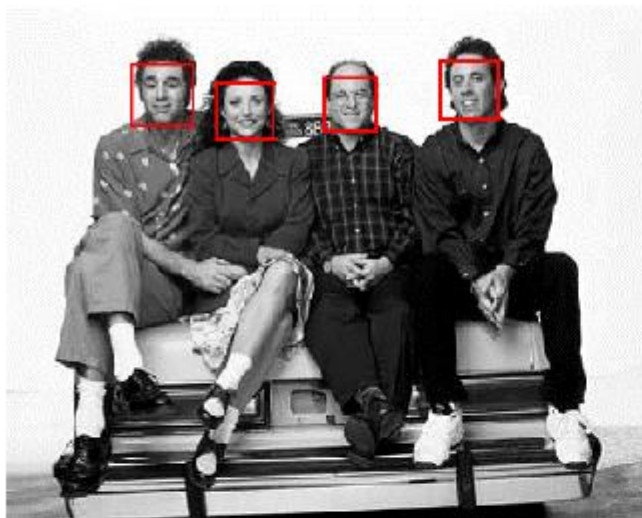
P. Viola and M. Jones. [Rapid object detection using a boosted cascade of simple features.](#) CVPR 2001.

P. Viola and M. Jones. [Robust real-time face detection.](#) IJCV 57(2), 2004.

Viola-Jones Face Detector: Results



Viola-Jones Face Detector: Results

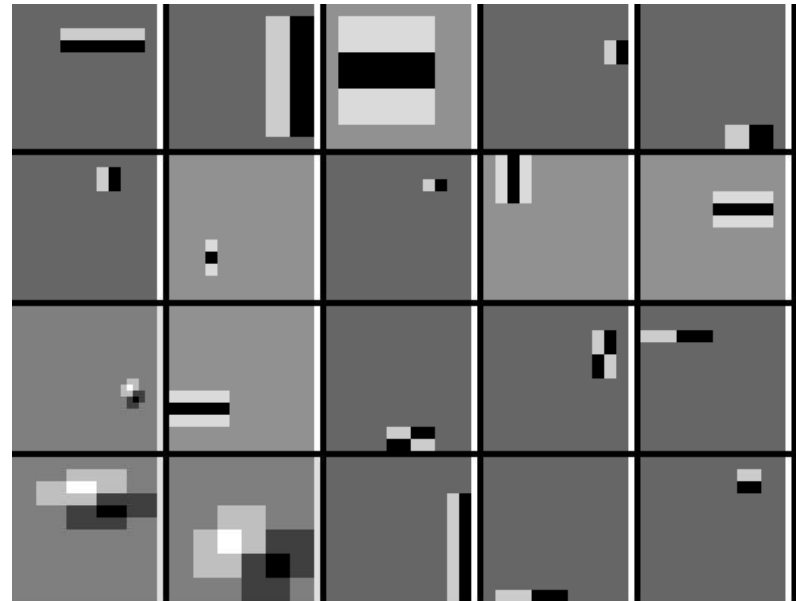


Viola-Jones Face Detector: Results



Detecting profile faces?

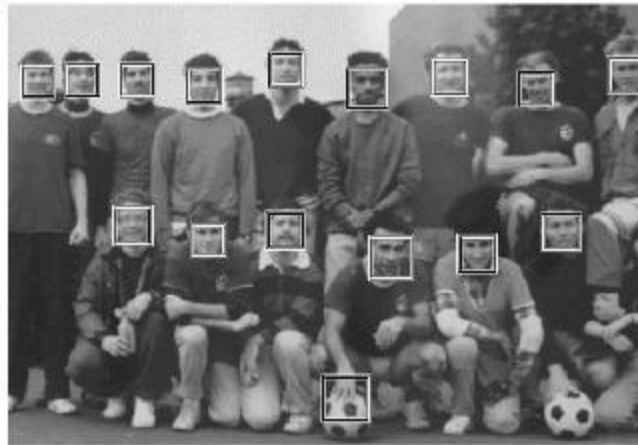
Can we use the same detector?



Viola-Jones Face Detector: Results



Viola Jones Results



<div>False detections</div> <div>Detector</div>	10	31	50	65	78	95	167
Viola-Jones	76.1%	88.4%	91.4%	92.0%	92.1%	92.9%	93.9%
Viola-Jones (voting)	81.1%	89.7%	92.1%	93.1%	93.1%	93.2 %	93.7%
Rowley-Baluja-Kanade	83.2%	86.0%	-	-	-	89.2%	90.1%
Schneiderman-Kanade	-	-	-	94.4%	-	-	-
Roth-Yang-Ahuja	-	-	-	-	(94.8%)	-	-

MIT + CMU face dataset

Slide: Derek Hoiem

Schneiderman later results

Schneiderman 2004

Viola-Jones 2001

Roth et al. 1999

Schneiderman-

Kanade 2000

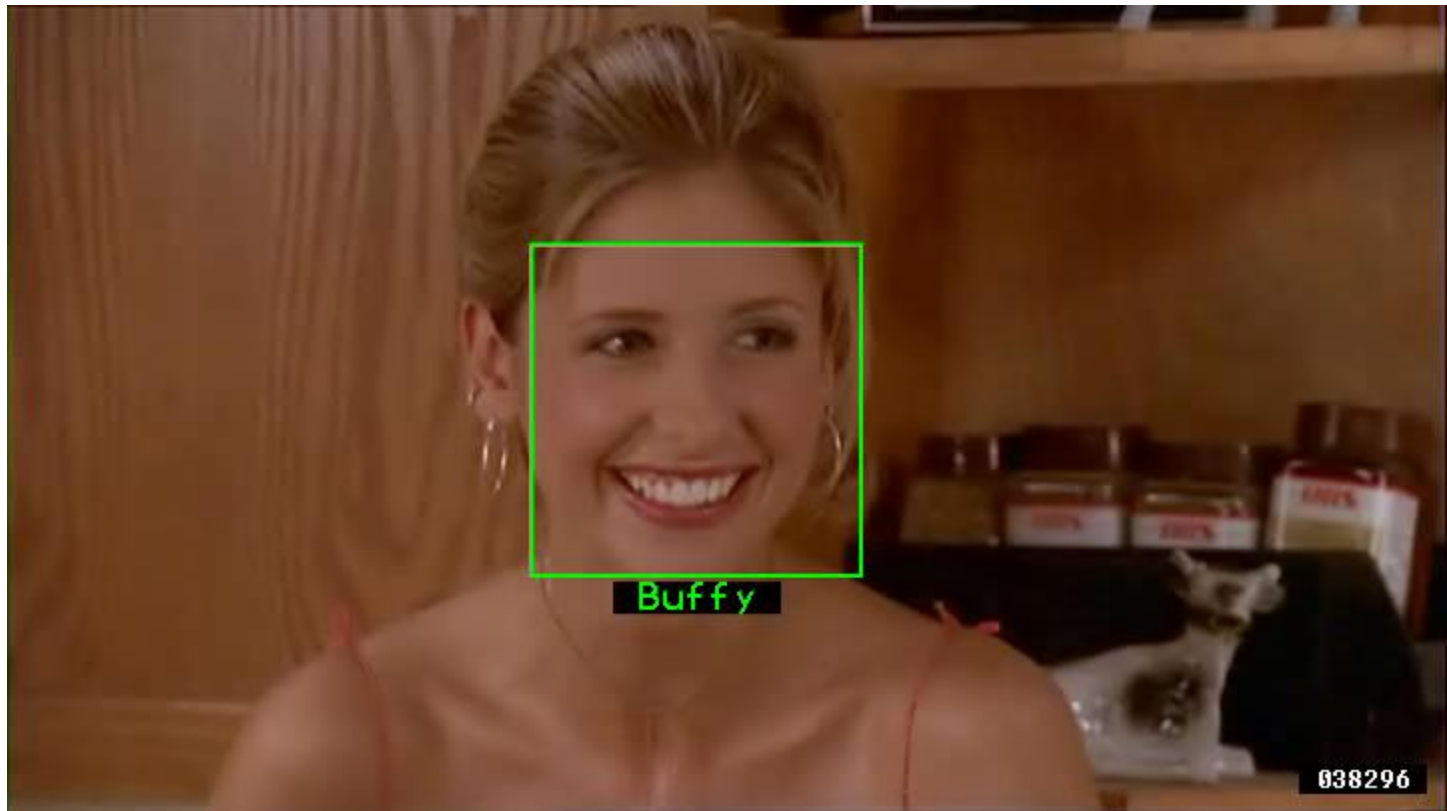
	89.7%	93.1%	94.4%	94.8%	95.7%
Bayesian Network *	1	8	19	36	56
Semi-Naïve Bayes*	6	19	29	35	46
[6]	31	65	--	--	--
[7]*	--	--	--	78	--
[16]*	--	--	65	--	--

Table 2. False alarms as a function of recognition rate on the MIT-CMU Test Set for Frontal Face Detection. * indicates exclusion of the 5 images of hand-drawn faces.

Speed: frontal face detector

- Schneiderman-Kanade (2000): 5 seconds
- Viola-Jones (2001): 15 fps

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>

[News](#) > [Internet](#)

Google now erases faces, license plates on Map Street View

By [Elinor Mills](#), CNET News.com
Friday, August 24, 2007 01:37 PM

Google has gotten a lot of flack from privacy advocates for photographing faces and license plate numbers and displaying them on the Street View in Google Maps. Originally, the company said only people who identified themselves could ask the company to remove their image.

But Google has quietly changed that policy, partly in response to criticism, and now anyone can alert the company and have an image of a license plate or a recognizable face removed, not just the owner of the face or car, says Marissa Mayer, vice president of search products and user experience at Google.

"It's a good policy for users and also clarifies the intent of the product," she said in an interview following her keynote at the Search Engine Strategies conference in San Jose, Calif., Wednesday.

The policy change was made about 10 days after the launch of the product in late May, but was not publicly announced, according to Mayer. The company is removing images only when someone notifies them and not proactively, she said. "It was definitely a big policy change inside."

News from Countries/Region

- » [Singapore](#)
- » [India](#)
- » [China/HK/F](#)
- » [Malaysia](#)
- » [Philippines](#)
- » [ASEAN](#)
- » [Thailand](#)
- » [Indonesia](#)
- » [Asia Pacific](#)

What's Hot [Latest News](#)

- [Is eBay facing seller revolt?](#)
- [Report: Amazon may again be mulling Netflix bu](#)
- [Mozilla maps out Jetpack add-on transition plan](#)
- [Google begins search for Middle East lobbyist](#)
- [Google still thinks it can change China](#)

▼ advertisement



ZDNet Asia
TECH SHOWCASE

Brought to you by **CISCO**

Cisco Collaboration Solutions

Consumer application: iPhoto 2009



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

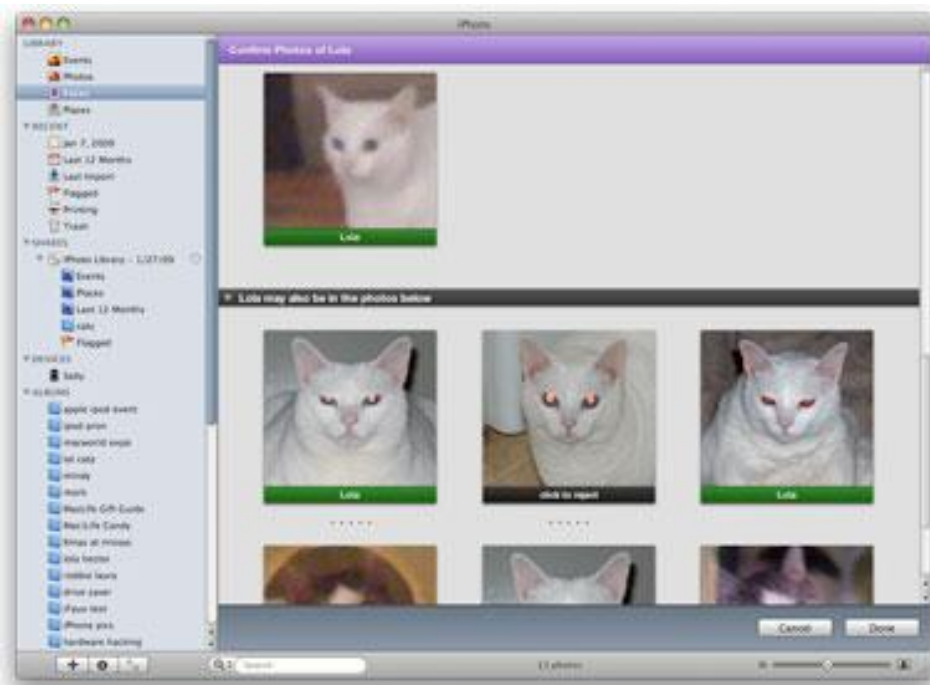
Consumer application: iPhoto 2009

Things iPhoto thinks are faces



Consumer application: iPhoto 2009

Can be trained to recognize pets!



http://www.maclife.com/article/news/iphotos_faces_recognizes_cats