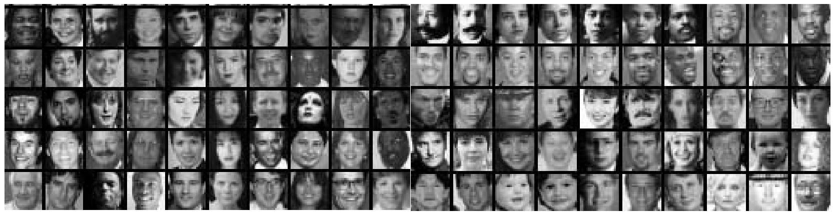


Sliding window detection

January 29, 2009

Kristen Grauman

UT-Austin

Schedule

- http://www.cs.utexas.edu/~grauman/courses/spring2009/schedule.htm
- http://www.cs.utexas.edu/~grauman/courses/spring2009/papers.htm

Plan for today

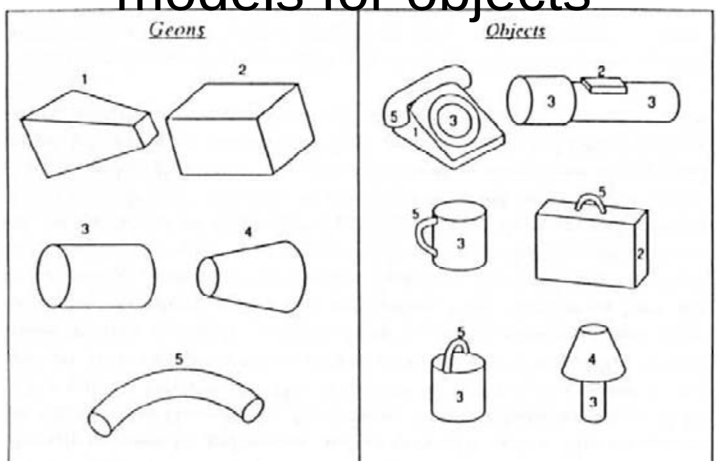
- Lecture
  - Sliding window detection
  - Contrast-based representations
  - Face and pedestrian detection via sliding window classification
- Papers: HoG and Viola-Jones
- Demo
  - Viola-Jones detection algorithm

Tasks

- Detection: Find an object (or instance of object category) in the image.
- Recognition: Name the particular object (or category) for a given image/subimage.

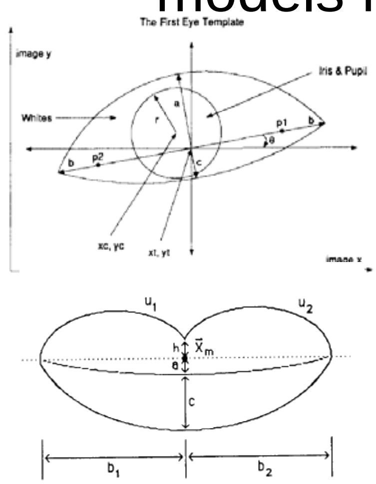
- How is the object (class) going to be modeled or learned?
- Given a new image, how to make a decision?

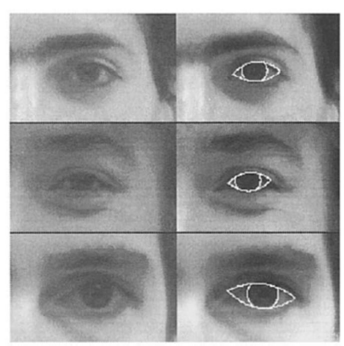
Earlier: Knowledge-rich models for objects



Irving Biederman, Recognition-by-Components: A Theory of Human Image Understanding. Psychological Review, 1987.

Earlier: Knowledge-rich models for objects

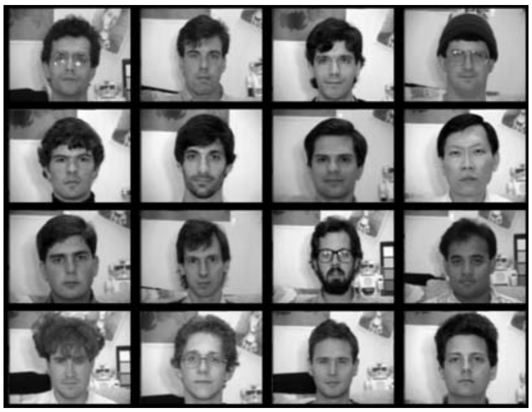


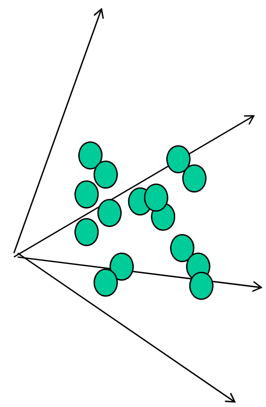


Alan L. Yuille, David S. Cohen, Peter W. Hallinan. Feature extraction from faces using deformable templates,1989.

Later: Statistical models of appearance

- Objects as appearance patches
  - E.g., a list of pixel intensities
- Learning patterns directly from image features

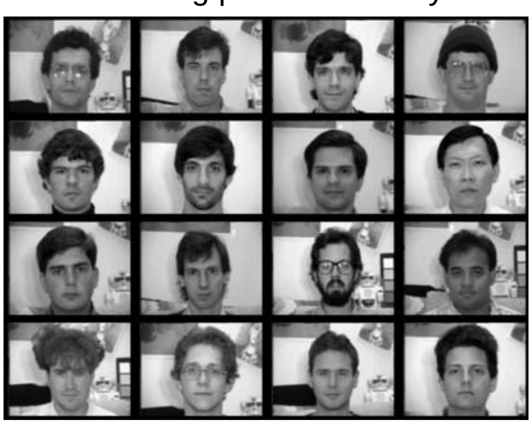


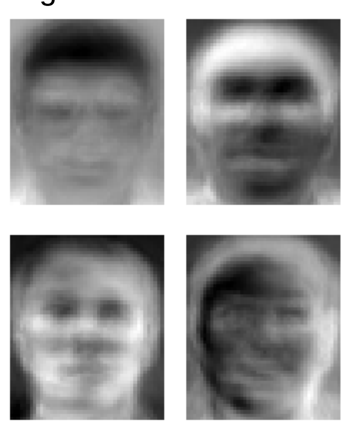


Eigenfaces (Turk & Pentland, 1991)

Later: Statistical models of appearance

- Objects as appearance patches
  - E.g., a list of pixel intensities
- Learning patterns directly from image features





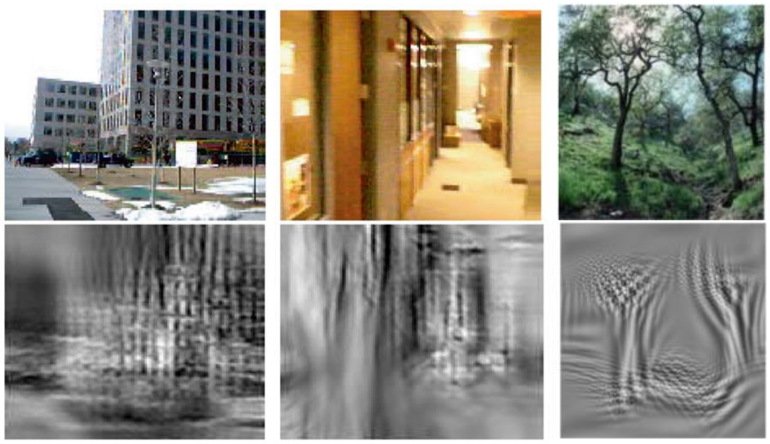
Eigenfaces (Turk & Pentland, 1991)

For what kinds of recognition tasks is a holistic description of appearance suitable?

Appearance-based descriptions

- Appropriate for classes with more rigid structure, and when good training examples available

Appearance-based descriptions



Scene recognition based on global texture pattern.
[Oliva & Torralba (2001)]

What if the object of interest may be embedded in "clutter"?

Sliding window object detection

Car/non-car Classifier

No, not a car.

Yes, car.

Sliding window object detection

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

Car/non-car Classifier

Detection via classification

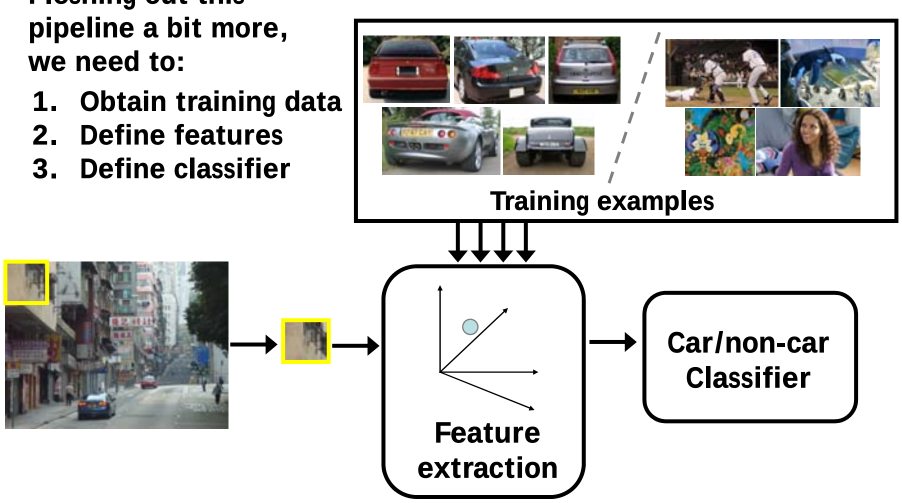
- Consider all subwindows in an image
  – Sample at multiple scales and positions

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

Detection via classification

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

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



Detector evaluation

How to evaluate a detector?

When do we have a **correct** detection?

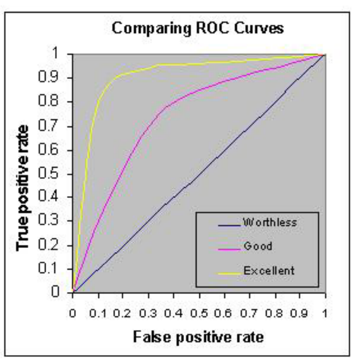
Is this correct?

$$\frac{\text{Area intersection}}{\text{Area union}} > 0.5$$

Slide credit: Antonio Torralba

Detector evaluation

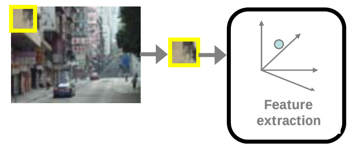
How to evaluate a detector?

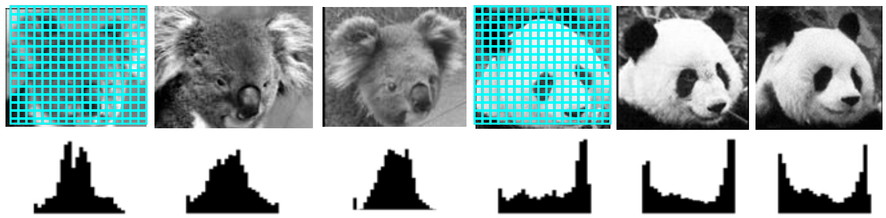


Summarize results with an **ROC curve**: show how the number of correctly classified positive examples varies relative to the number of incorrectly classified negative examples.

- Image: gim.unmc.edu/dxtests/ROC3.htm

Feature extraction: global appearance



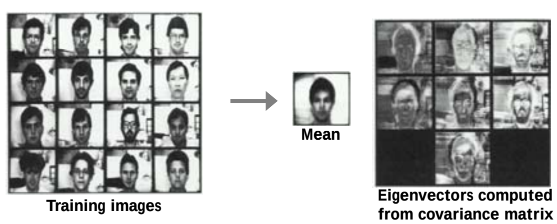


Simple holistic descriptions of image content

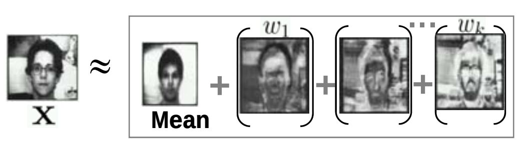
- **grayscale / color histogram**
- **vector of pixel intensities**

Eigenfaces: global appearance description

An early appearance-based approach to face recognition



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



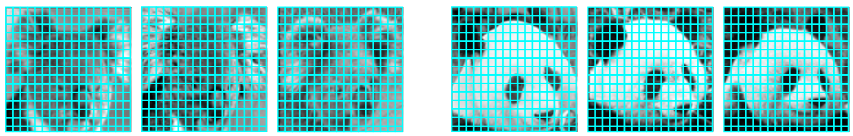
Project new images to “face space”.

Recognition via nearest neighbors in face space

Turk & Pentland, 1991

Feature extraction: global appearance

- Pixel-based representations sensitive to small shifts

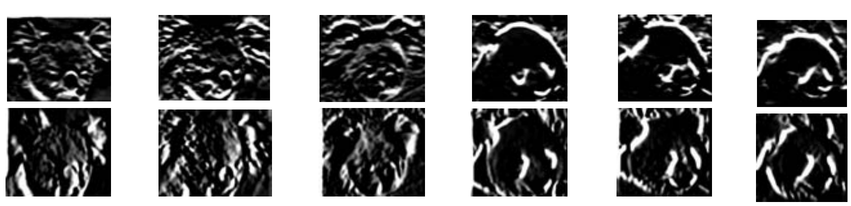


- Color or grayscale-based appearance description can be sensitive to illumination and intra-class appearance variation

Cartoon example: an albino koala

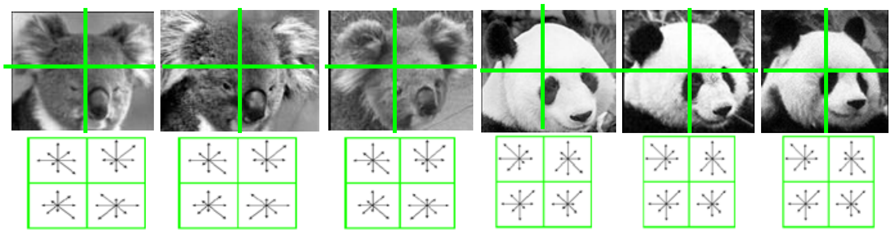
Gradient-based representations

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



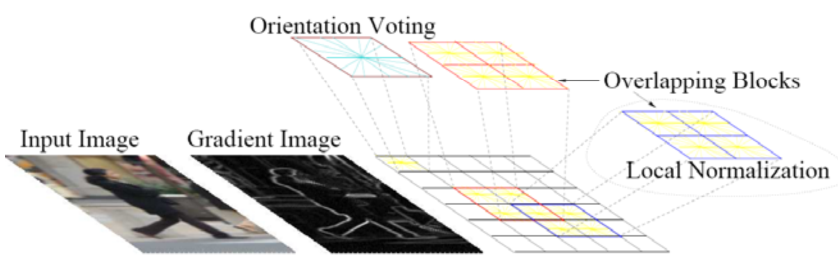
Gradient-based representations

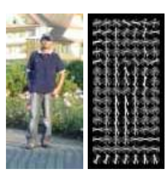
- Consider edges, contours, and (oriented) intensity gradients



- 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

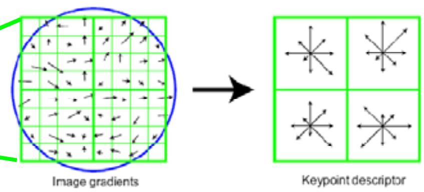




Map each grid cell in the input window to a histogram counting the gradients per orientation.

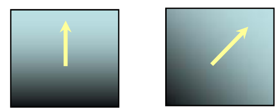
Dalal & Triggs, CVPR 2005

Gradient-based representations: SIFT descriptor



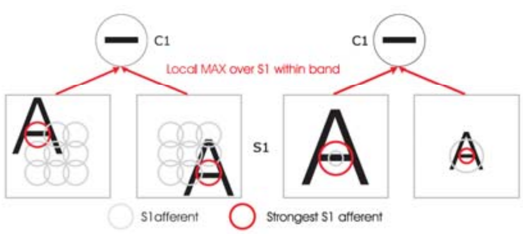
Local patch descriptor

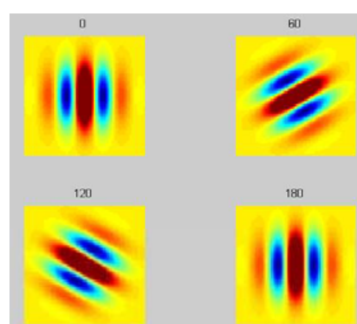
Rotate according to dominant gradient direction



Lowe, ICCV 1999

Gradient-based representations: biologically inspired features





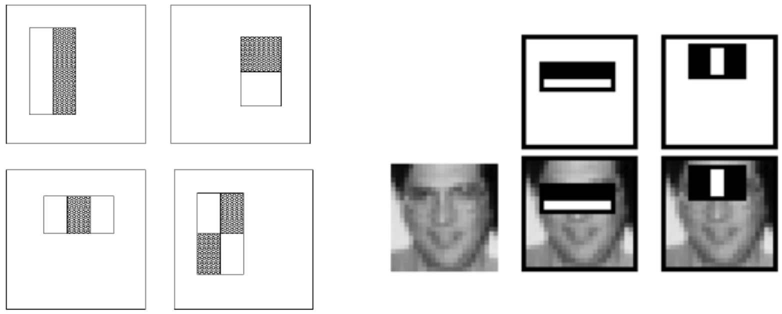
Convolve with Gabor filters at multiple orientations

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



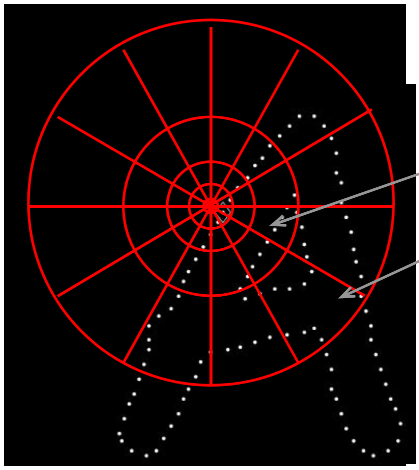
Compute differences between sums of pixels in rectangles

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

Belongie, Malik & Puzicha, ICCV 2001

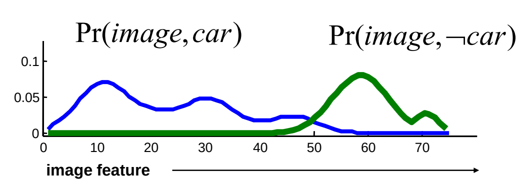
Classifier construction

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

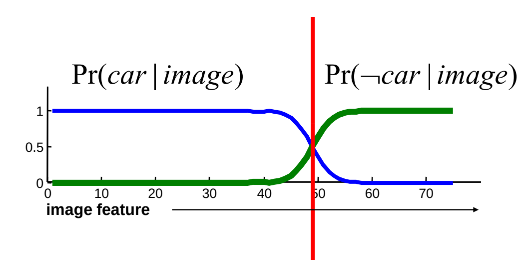
Image feature

Visual Object Recognition Tutorial

Discriminative vs. generative models



Generative: separately model class-conditional and prior densities



Discriminative: directly model posterior

Plots from Antonio Torralba 2007

Visual Object Recognition Tutorial

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**

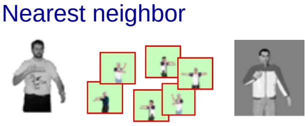
K. Grauman, B. Leibe

31

Visual Object Recognition Tutorial

Discriminative methods

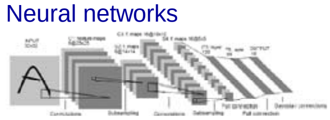
Nearest neighbor



10^6 examples

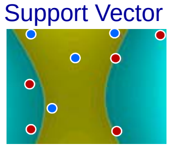
Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005...

Neural networks



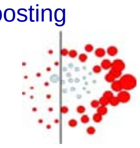
LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998
…

Support Vector Machines



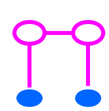
Guyon, Vapnik
Heisele, Serre, Poggio, 2001,…

Boosting



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

Conditional Random Fields



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

K. Grauman, B. Leibe

Slide adapted from Antonio Torralba

Visual Object Recognition Tutorial

Boosting

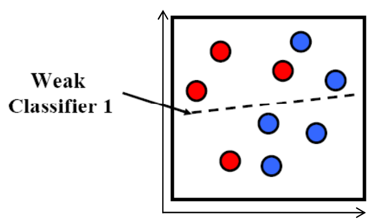
- Build a strong classifier by combining number of "weak classifiers", which need only be better than chance
- Sequential learning process: at each iteration, add a weak classifier
- Flexible to choice of weak learner
  - including fast simple classifiers that alone may be inaccurate
- We'll look at Freund & Schapire's AdaBoost algorithm
  - Easy to implement
  - Base learning algorithm for Viola-Jones face detector

K. Grauman, B. Leibe

33

Visual Object Recognition Tutorial

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.

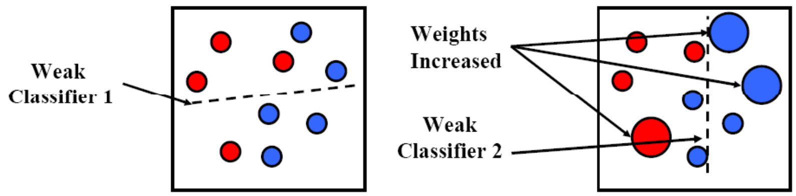
Examples misclassified by a previous weak learner are given more emphasis at future rounds.

Figure adapted from Freund and Schapire

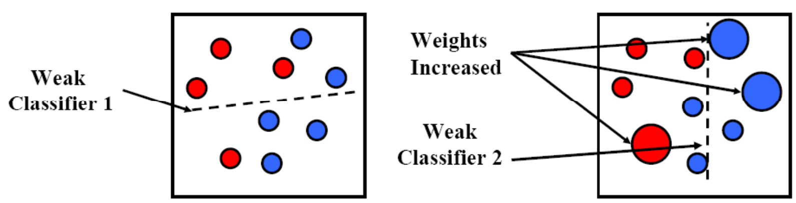
K. Grauman, B. Leibe

34

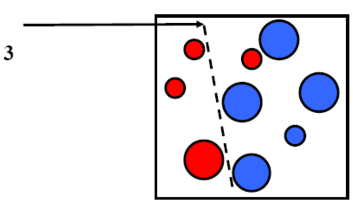
AdaBoost: Intuition



AdaBoost: Intuition



Weak
classifier 3

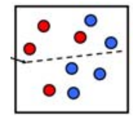


Final classifier is combination of the weak classifiers

- 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:
$$\boxed{h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}}$$
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 $\Rightarrow$ more weight
Correctly classified $\Rightarrow$ less weight

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

[Freund & Schapire 1995]

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**

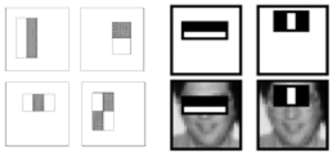
Visual Object Recognition Tutorial

K. Grauman, B. Leibe

38

Feature extraction

"Rectangular" filters

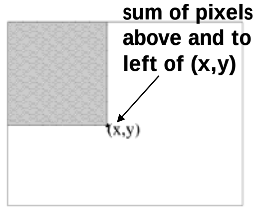


Feature output is difference between adjacent regions

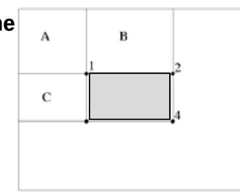
Efficiently computable with integral image: any sum can be computed in constant time

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)



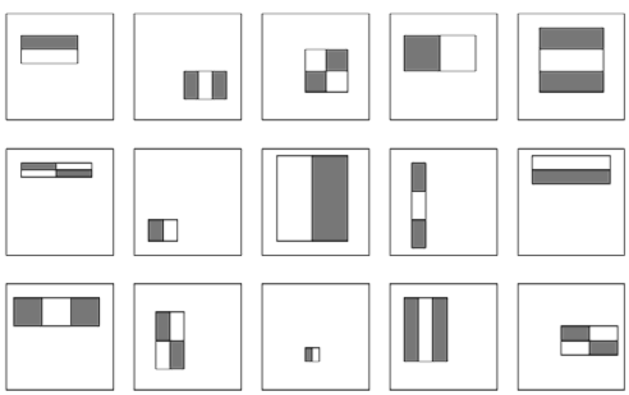
Integral image



$$D = 1 + 4 - (2 + 3)$$
$$= A + (A + B + C + D) - (A + C + A + B)$$
$$= D$$

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

Visual Object Recognition Tutorial

AdaBoost for Efficient Feature Selection

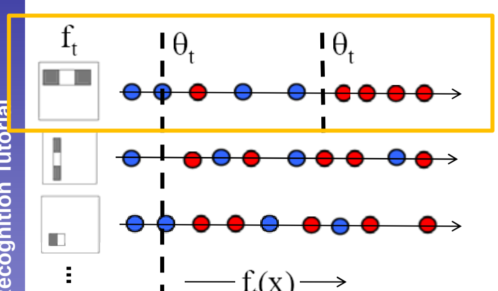
- **Image features = weak classifiers**
- **For each round of boosting:**
  - **Evaluate each rectangle filter on each example**
  - **Sort examples by filter values**
  - **Select best threshold for each filter (min error)**
    - **Sorted list can be quickly scanned for the optimal threshold**
  - **Select best filter/threshold combination**
  - **Weight on this features is a simple function of error rate**
  - **Reweight examples**

P. Viola, M. Jones, Robust Real-Time Face Detection, IJCV, Vol. 57(2), 2004.
(first version appeared at CVPR 2001)

Visual Object Recognition Tutorial

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

K. Grauman, B. Leibe

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

Visual Object Recognition Tutorial

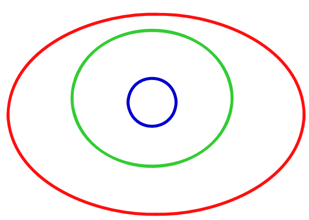
K. Grauman, B. Leibe

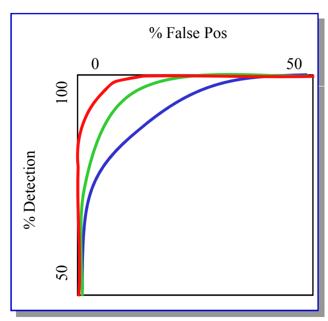
Figure from Viola & Jones CVPR 2001

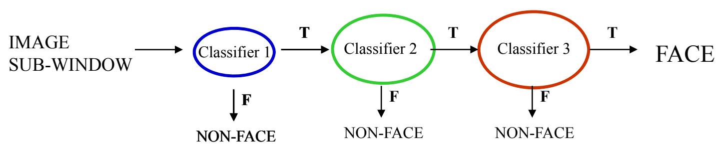
43

Cascading classifiers for detection

- Given a nested set of classifier hypothesis classes



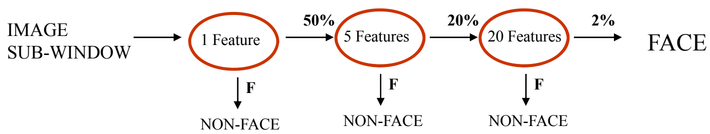




Slide credit: Paul Viola

Viola 2003

Cascading classifiers for detection



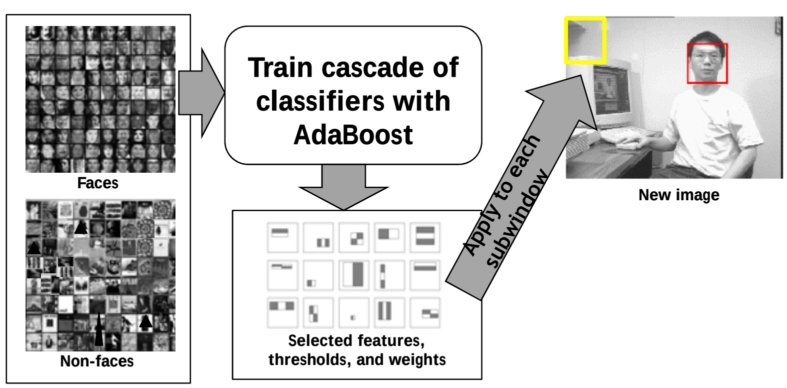
- A 1 feature classifier achieves 100% detection rate and about 50% false positive rate.
- A 5 feature classifier achieves 100% detection rate and 40% false positive rate (20% cumulative)
  - using data from previous stage.
- A 20 feature classifier achieve 100% detection rate with 10% false positive rate (2% cumulative)

Slide credit: Paul Viola

Viola 2003

Visual Object Recognition Tutorial

Viola-Jones Face Detector: Summary



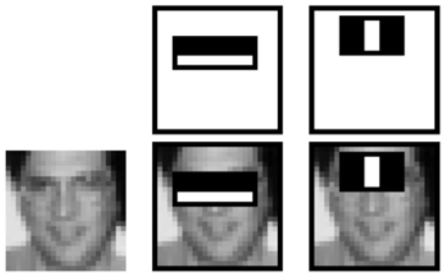
- **Train with 5K positives, 350M negatives**
- **Real-time detector using 38 layer cascade**
- **6061 features in final layer**
- [Implementation available in OpenCV: http://www.intel.com/technology/computing/opencv/]

K. Grauman, B. Leibe

46

Visual Object Recognition Tutorial

Viola-Jones Face Detector: Results



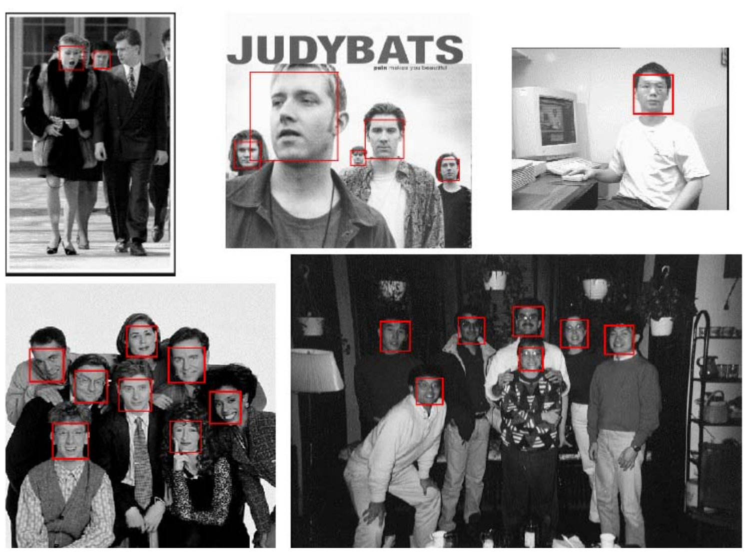
First two features selected

K. Grauman, B. Leibe

47

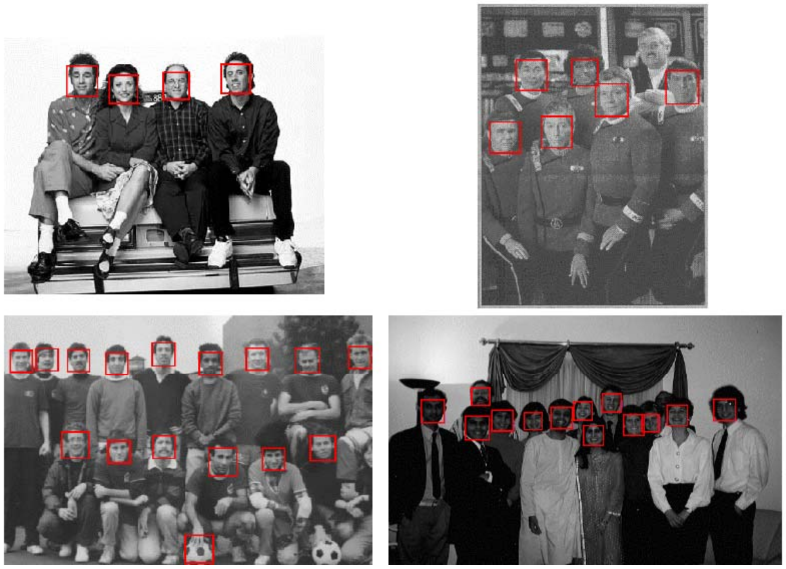
Visual Object Recognition Tutorial

Viola-Jones Face Detector: Results



K. Grauman, B. Leibe

Viola-Jones Face Detector: Results



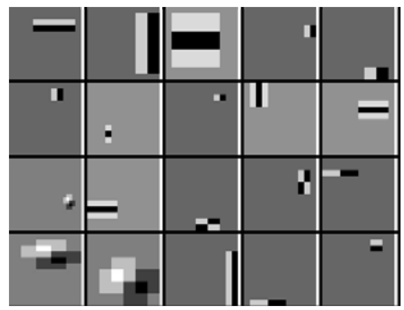
K. Grauman, B. Leibe

Viola-Jones Face Detector: Results

K. Grauman, B. Leibe

Profile Features

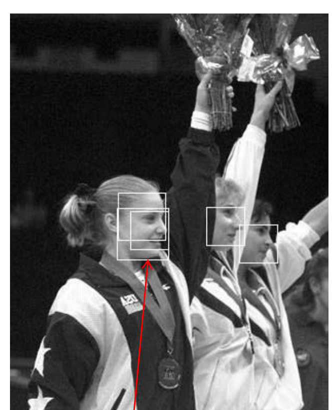
Detecting profile faces requires training separate detector with profile examples.



Visual Object Recognition Tutorial

K. Grauman, B. Leibe

Viola-Jones Face Detector: Results



Postprocess: suppress non-maxima

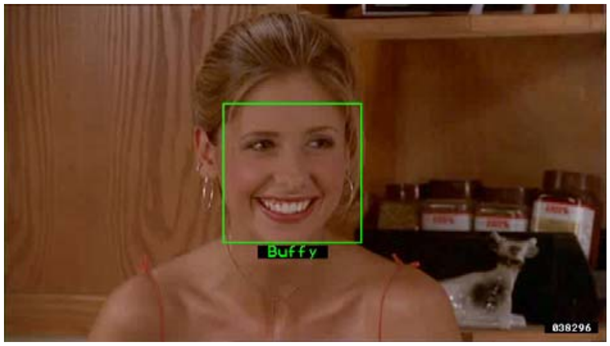
Visual Object Recognition Tutorial

Paul

K. Grauman, B. Leibe

Visual Object Recognition Tutorial

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

K. Grauman, B. Leibe

53

Fast face detection: Viola & Jones

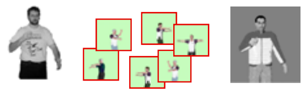
Key points:

- Huge library of features
- Integral image – efficiently computed
- AdaBoost to find best combo of features
- Cascade architecture for fast detection

Visual Object Recognition Tutorial

Discriminative methods

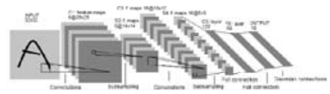
Nearest neighbor



10^6 examples

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

Neural networks

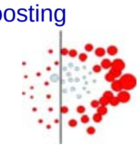


LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998
…

Support Vector Machines

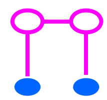
Guyon, Vapnik
Heisele, Serre, Poggio, 2001,…

Boosting



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

Conditional Random Fields

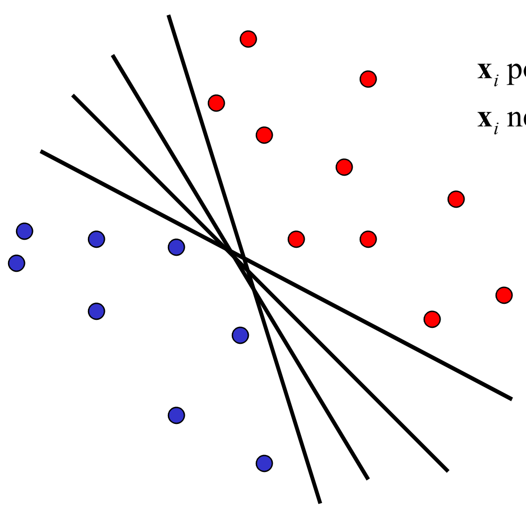


McCallum, Freitag, Pereira 2000; Kumar, Hebert 2003
…

Slide adapted from Antonio Torralba

Linear classifiers

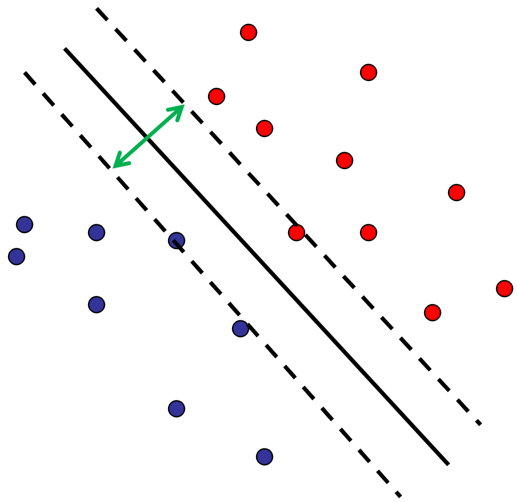
- Find linear function to separate positive and negative examples



$$\mathbf{x}_i \text{ positive}: \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0$$

$$\mathbf{x}_i \text{ negative}: \quad \mathbf{x}_i \cdot \mathbf{w} + b < 0$$

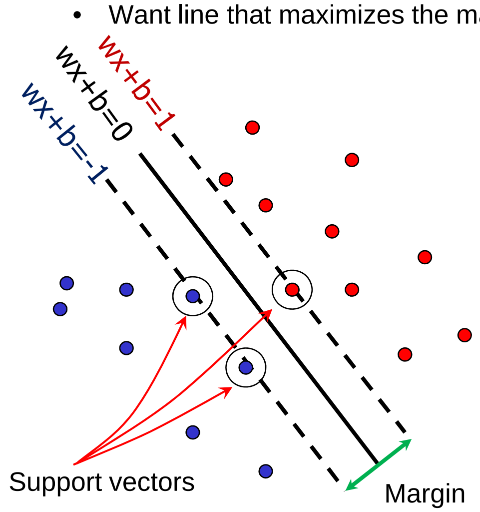
Support Vector Machines (SVMs)



- Discriminative classifier based on *optimal separating hyperplane*
- Maximize the *margin* between the positive and negative training examples

Support vector machines

- Want line that maximizes the margin.



$\mathbf{x}_i$ positive $(y_i = 1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

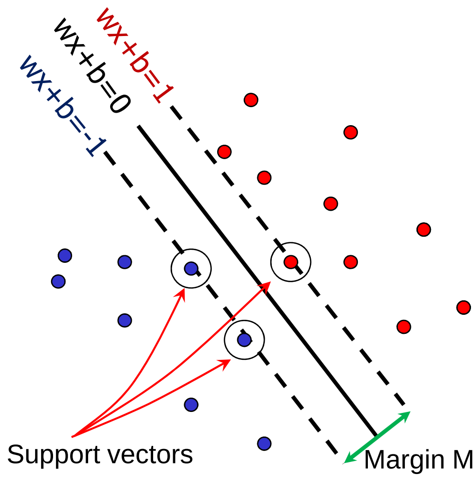
$\mathbf{x}_i$ negative $(y_i = -1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

For support, vectors, $\quad \mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Support vector machines

- Want line that maximizes the margin.



$\mathbf{x}_i$ positive $(y_i = 1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

$\mathbf{x}_i$ negative $(y_i = -1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

For support, vectors, $\quad \mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

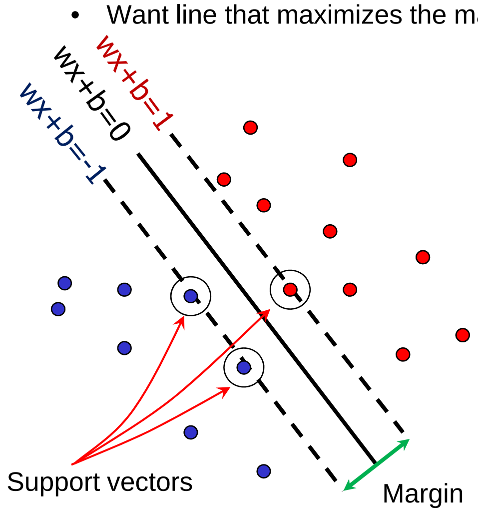
Distance between point and line: $\quad \dfrac{|\mathbf{x}_i \cdot \mathbf{w} + b|}{\|\mathbf{w}\|}$

For support vectors:

$$\frac{\mathbf{w}^T\mathbf{x} + b}{\|\mathbf{w}\|} = \frac{\pm 1}{\|\mathbf{w}\|} \qquad M = \left|\frac{1}{\|\mathbf{w}\|} - \frac{-1}{\|\mathbf{w}\|}\right| = \frac{2}{\|\mathbf{w}\|}$$

Support vector machines

- Want line that maximizes the margin.



$\mathbf{x}_i$ positive $(y_i = 1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

$\mathbf{x}_i$ negative $(y_i = -1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

For support, vectors, $\quad \mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

Distance between point and line: $\quad \dfrac{|\mathbf{x}_i \cdot \mathbf{w} + b|}{\|\mathbf{w}\|}$

Therefore, the margin is $2 / \|\mathbf{w}\|$

Finding the maximum margin line

1. Maximize margin $2/||\mathbf{w}||$
2. Correctly classify all training data points:

$$\mathbf{x}_i \text{ positive } (y_i = 1): \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$$
$$\mathbf{x}_i \text{ negative } (y_i = -1): \quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$$

Quadratic optimization problem:

Minimize $\frac{1}{2}\mathbf{w}^T\mathbf{w}$

Subject to $y_i(\mathbf{w}\cdot\boldsymbol{x}_i+b) \geq 1$

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1

Finding the maximum margin line

- Solution: $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$

learned weight

Support vector

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1

Finding the maximum margin line

- Solution: $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$

  $b = y_i - \mathbf{w}\cdot\mathbf{x}_i$ (for any support vector)

  $$\mathbf{w}\cdot\mathbf{x} + b = \sum_i \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b$$
- Classification function:

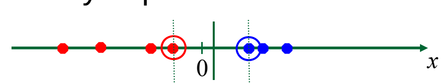
  $$f(x) = \text{sign}\,(\mathbf{w}\cdot\mathbf{x} + \text{b})$$
  $$= \text{sign}\left(\sum_{\text{i}} \alpha_{\text{i}} \boxed{\mathbf{x}_i \cdot \mathbf{x}} + b\right)$$

  If f(x) < 0, classify as negative, if f(x) > 0, classify as positive
- Notice that it relies on an *inner product* between the test point $\boldsymbol{x}$ and the support vectors $\boldsymbol{x}_i$
- (Solving the optimization problem also involves computing the inner products $\boldsymbol{x}_i \cdot \boldsymbol{x}_j$ between all pairs of training points)

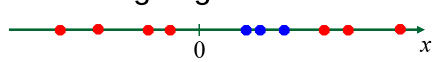
C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1

Non-linear SVMs

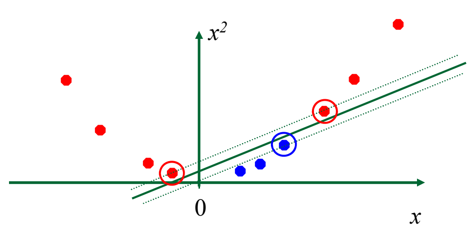
- Datasets that are linearly separable with some noise work out great:



- But what are we going to do if the dataset is just too hard?



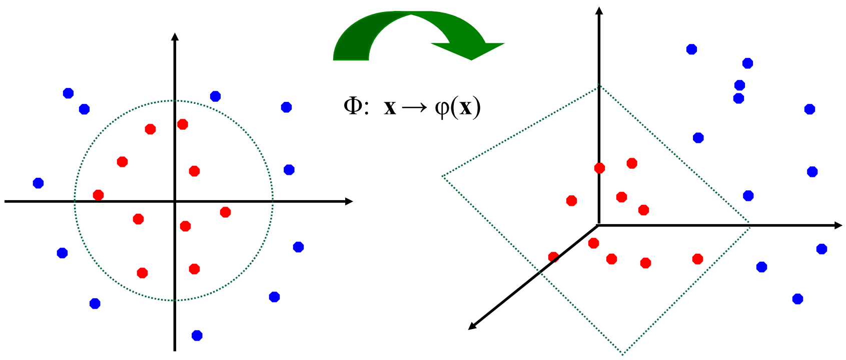
- How about… mapping data to a higher-dimensional space:



Slide from Andrew Moore's tutorial: http://www.autonlab.org/tutorials/svm.html

Non-linear SVMs: Feature spaces

- General idea: the original input space can be mapped to some higher-dimensional feature space where the training set is separable:



Slide from Andrew Moore's tutorial: http://www.autonlab.org/tutorials/svm.html

Nonlinear SVMs

- *The kernel trick*: instead of explicitly computing the lifting transformation $\varphi(\mathbf{x})$, define a kernel function K such that

$$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$$

- This gives a nonlinear decision boundary in the original feature space:

$$\sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

C. Burges, [A Tutorial on Support Vector Machines for Pattern Recognition](), Data Mining and Knowledge Discovery, 1998

Examples of General Purpose Kernel Functions

- Linear: $K(\mathbf{x}_i,\mathbf{x}_j)= \mathbf{x}_i^T\mathbf{x}_j$
- Polynomial of power p: $K(\mathbf{x}_i,\mathbf{x}_j)= (1+ \mathbf{x}_i^T\mathbf{x}_j)^p$
- Gaussian (radial-basis function network):

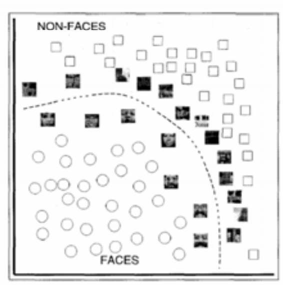
$$K(\mathbf{x}_i,\mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$

More on specialized image kernels -- next class.

Slide from Andrew Moore's tutorial: http://www.autonlab.org/tutorials/svm.html

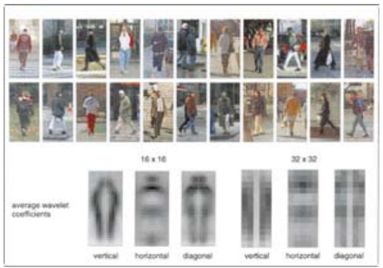
SVMs for recognition

1. Define your representation for each example.
2. Select a kernel function.
3. Compute pairwise kernel values between labeled examples
4. Given this "kernel matrix" to SVM optimization software to identify support vectors & weights.
5. To classify a new example: compute kernel values between new input and support vectors, apply weights, check sign of output.



Pedestrian detection

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



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

SVM with HoGs [Dalal & Triggs, CVPR 2005]

Pedestrian detection

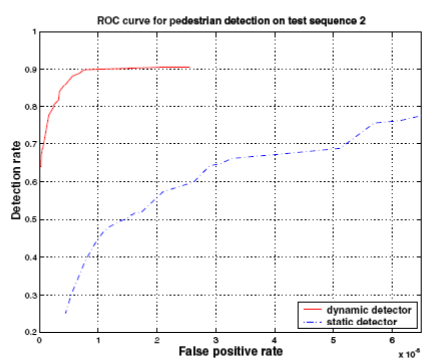
- Navneet Dalal, Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

Moving pedestrians

- What about video? Is pedestrian motion a useful feature?

Detecting Pedestrians Using Patterns of Motion and Appearance, P. Viola, M. Jones, and D. Snow, ICCV 2003.

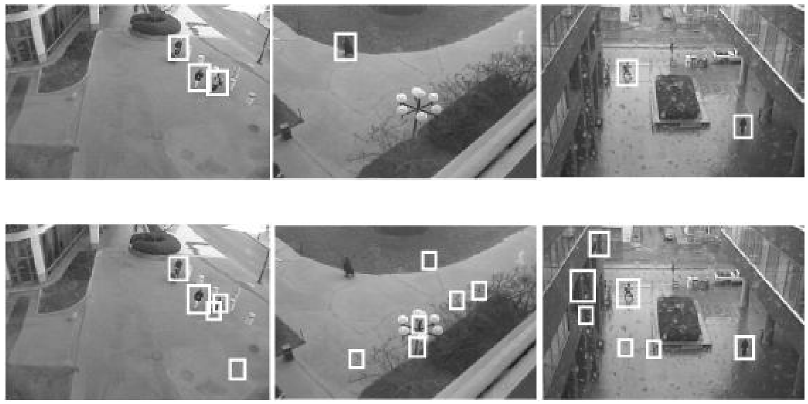
- – Use motion and appearance to detect pedestrians
- – Generalize rectangle features for sequence data
- – Training examples = pairs of images.



- Detecting Pedestrians Using Patterns of Motion and Appearance, P. Viola, M. Jones, and D. Snow, ICCV 2003.

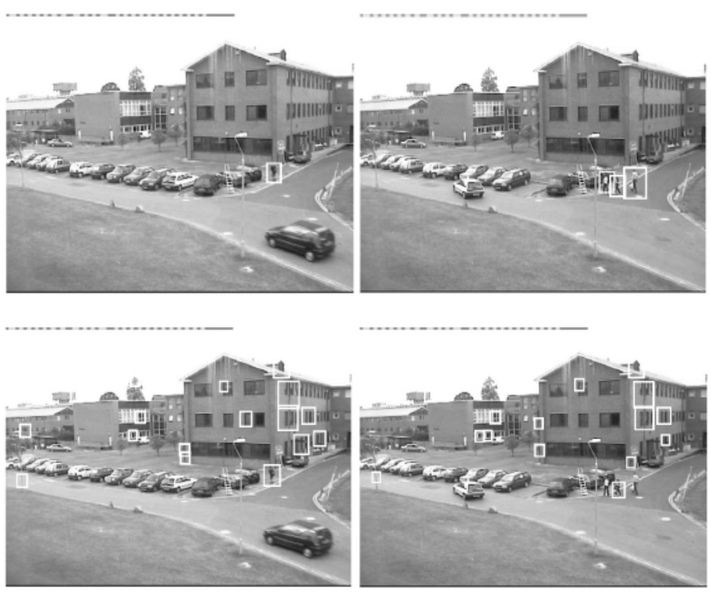
Dynamic detector

Static detector



Dynamic detector

Static detector



Global appearance, windowed detectors: The good things

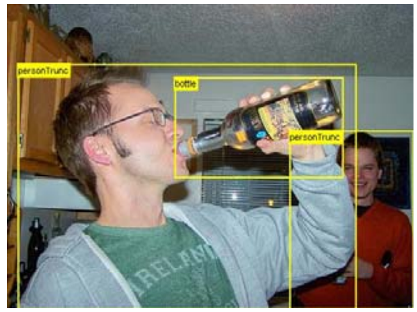
- Some classes well-captured by 2d appearance pattern
- Simple detection protocol to implement
- Good feature choices critical
- Past successes for certain classes

Limitations

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

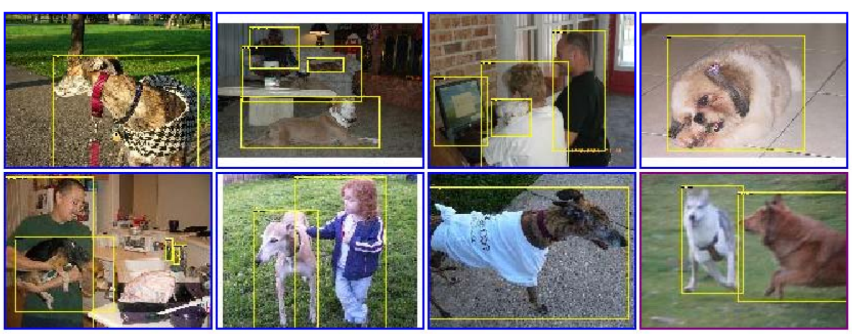
Limitations (continued)

- 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

Visual Object Recognition Tutorial

Figure credit: Derek Hoiem

K. Grauman, B. Leibe

79

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**

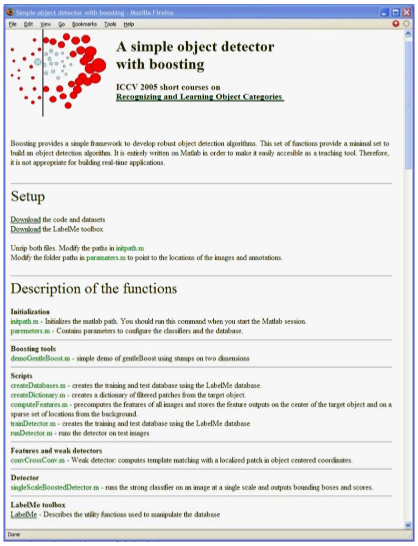
Visual Object Recognition Tutorial

Image credit: Adam, Rivlin, & Shimshoni

K. Grauman, B. Leibe

80

Tools:
A simple object detector with Boosting


Simple object detector with boosting - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

A simple object detector with boosting

ICCV 2005 short courses on Recognizing and Learning Object Categories

Boosting provides a simple framework to develop robust object detection algorithms. This set of functions provide a minimal set to build an object detection algorithm. It is entirely written on Matlab in order to make it easily accesible as a teaching tool. Therefore, it is not appropriate for building real-time applications.

Setup

Download the code and datasets
Download the LabelMe toolbox

Unzip both files. Modify the paths in initpath.m
Modify the folder paths in paramaters.m to point to the locations of the images and annotations.

Description of the functions

Initialization
initpath.m - Initializes the matlab path. You should run this command when you start the Matlab session.
paremeters.m - Contains parameters to configure the classifiers and the database.

Boosting tools
demoGentleBoost.m - simple demo of gentleBoost using stumps on two dimensions

Scripts
createDatabases.m - creates the training and test database using the LabelMe database.
createDictionary.m - creates a dictionary of filtered patches from the target object.
computeFeatures.m - precomputes the features of all images and stores the feature outputs on the center of the target object and on a sparse set of locations from the background.
trainDetector.m - creates the training and test database using the LabelMe database
runDetector.m - runs the detector on test images

Features and weak detectors
convCrossConv.m - Weak detector: computes template matching with a localized patch in object centered coordinates.

Detector
singleScaleBoostedDetector.m - runs the strong classifier on an image at a single scale and outputs bounding boxes and scores.

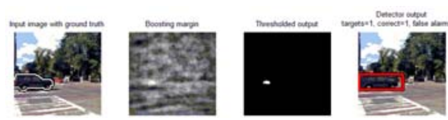
LabelMe toolbox
LabelMe - Describes the utility functions used to manipulate the database

Done

Download
- Toolbox for manipulating dataset
- Code and dataset

Matlab code
- Gentle boosting
- Object detector using a part based model

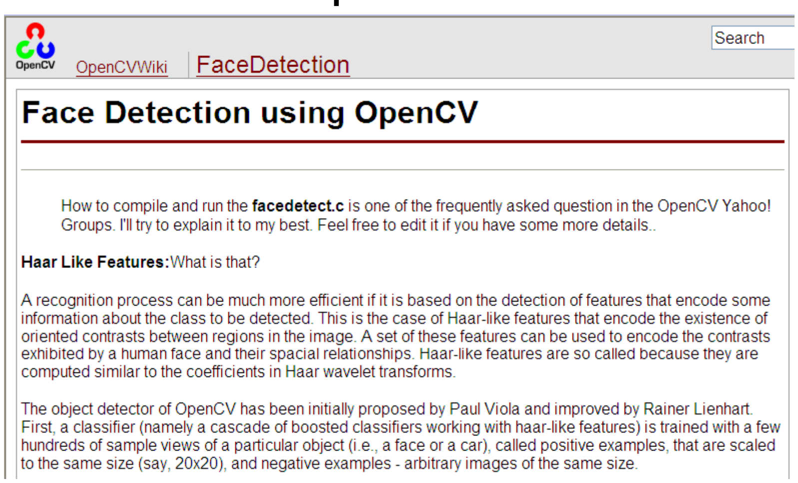
Dataset with cars and computer monitors



http://people.csail.mit.edu/torralba/iccv2005/

From : Antonio Torralba

Tools:
OpenCV


OpenCV

OpenCVWiki | FaceDetection

Search

Face Detection using OpenCV

How to compile and run the **facedetect.c** is one of the frequently asked question in the OpenCV Yahoo! Groups. I'll try to explain it to my best. Feel free to edit it if you have some more details..

Haar Like Features: What is that?

A recognition process can be much more efficient if it is based on the detection of features that encode some information about the class to be detected. This is the case of Haar-like features that encode the existence of oriented contrasts between regions in the image. A set of these features can be used to encode the contrasts exhibited by a human face and their spacial relationships. Haar-like features are so called because they are computed similar to the coefficients in Haar wavelet transforms.

The object detector of OpenCV has been initially proposed by Paul Viola and improved by Rainer Lienhart. First, a classifier (namely a cascade of boosted classifiers working with haar-like features) is trained with a few hundreds of sample views of a particular object (i.e., a face or a car), called positive examples, that are scaled to the same size (say, 20x20), and negative examples - arbitrary images of the same size.

- http://pr.willowgarage.com/wiki/OpenCV

Tools: LibSVM

- http://www.csie.ntu.edu.tw/~cjlin/libsvm/
- C++, Java
- Matlab interface