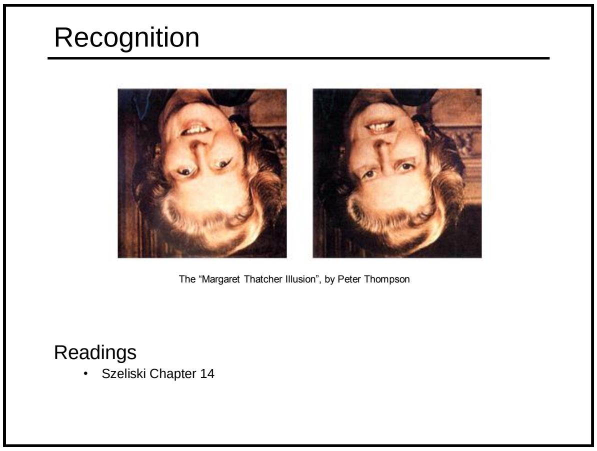
Object recognition (part 1)

CSE P 576
Larry Zitnick (larryz@microsoft.com)

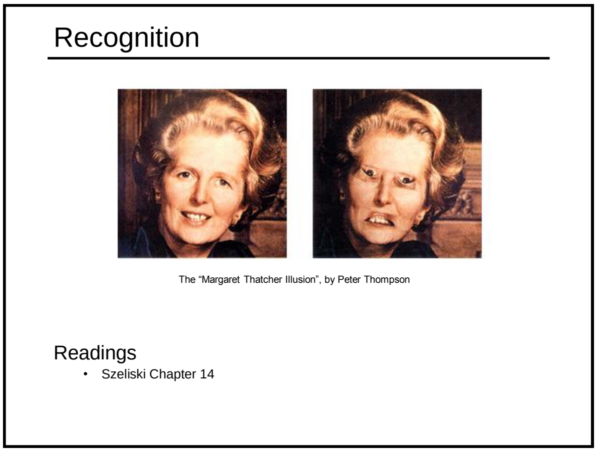


Recognition

The "Margaret Thatcher Illusion", by Peter Thompson

Readings

- Szeliski Chapter 14

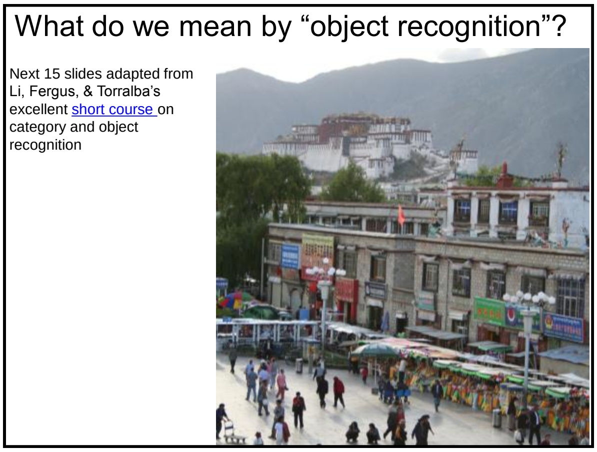


Recognition

The "Margaret Thatcher Illusion", by Peter Thompson

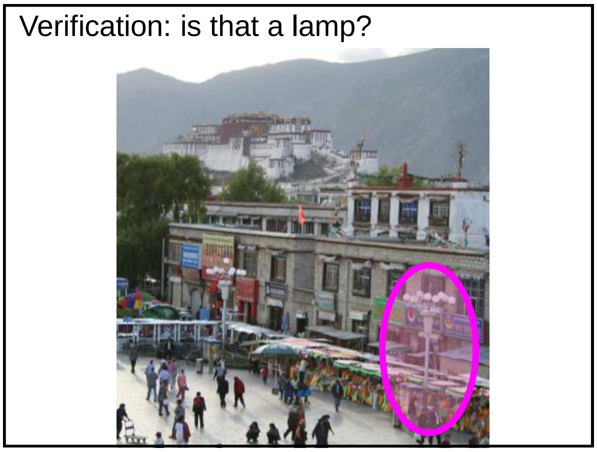
Readings

- Szeliski Chapter 14

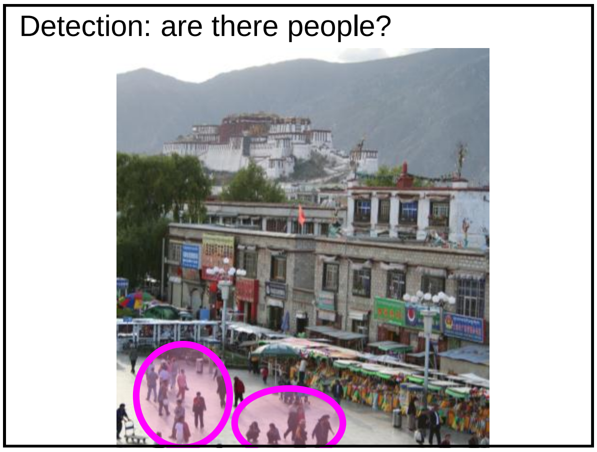


What do we mean by "object recognition"?

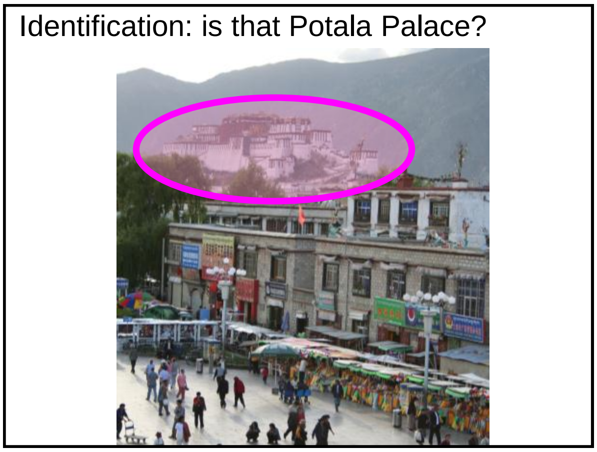
Next 15 slides adapted from Li, Fergus, & Torralba's excellent short course on category and object recognition



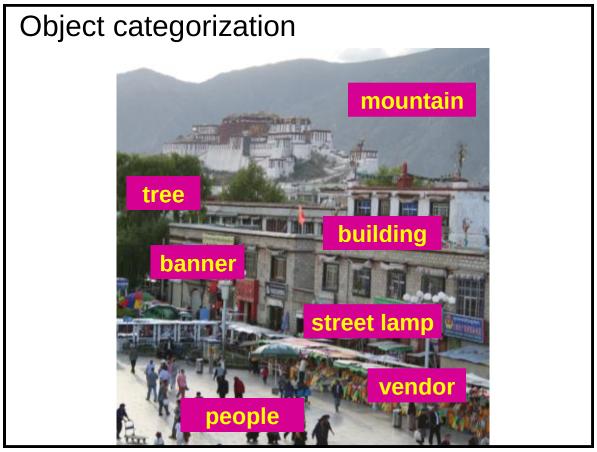
Verification: is that a lamp?



Detection: are there people?

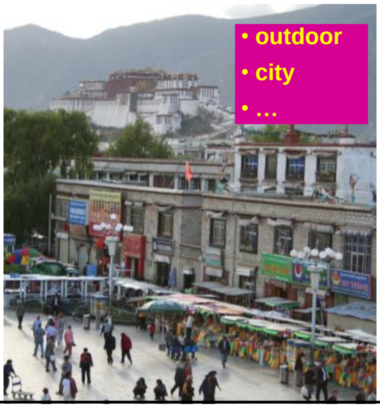


Identification: is that Potala Palace?



Object categorization

Scene and context categorization

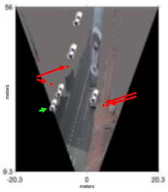


Applications: Computational photography

[Face priority AE] When a bright part of the face is too bright

Applications: Assisted driving

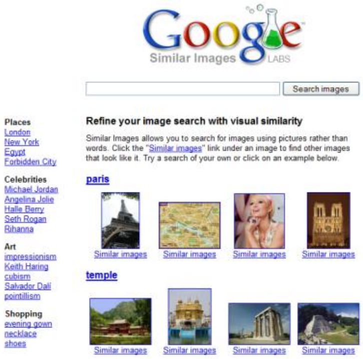
Pedestrian and car detection



Lane detection

- Collision warning systems with adaptive cruise control,
- Lane departure warning systems,
- Rear object detection systems,

Applications: image search



Places
London
New York
Egypt
Forbidden City

Celebrities
Michael Jordan
Angelina Jolie
Halle Berry
Seth Rogan
Rihanna

Art
impressionism
Keith Haring
cubism
Salvador Dali
pointillism

Shopping
evening gown
necklace
shoes

Refine your image search with visual similarity

Similar Images allows you to search for images using pictures rather than words. Click the "Similar images" link under an image to find other images that look like it. Try a search of your own or click on an example below.

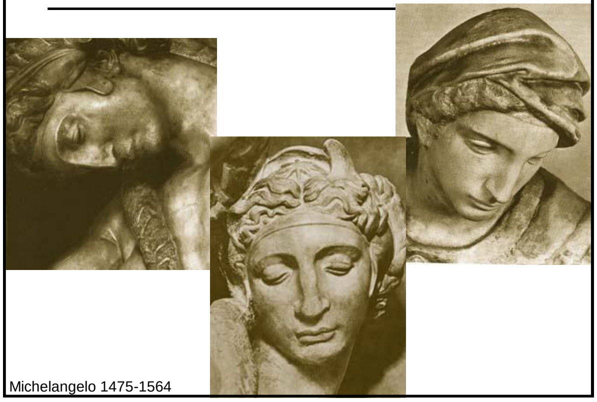
paris

Similar images Similar images Similar images Similar images

temple

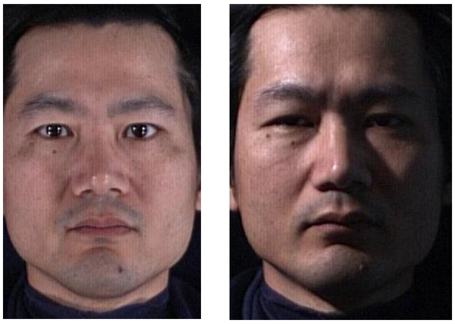
Similar images Similar images Similar images Similar images

Challenges: viewpoint variation



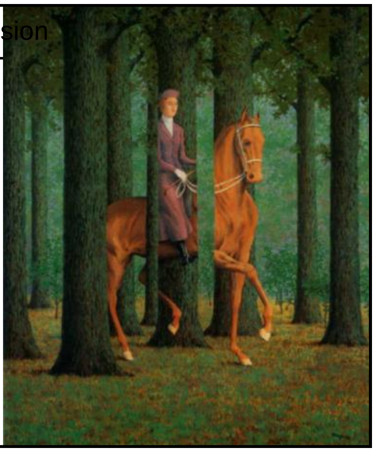
Michelangelo 1475-1564

Challenges: illumination variation



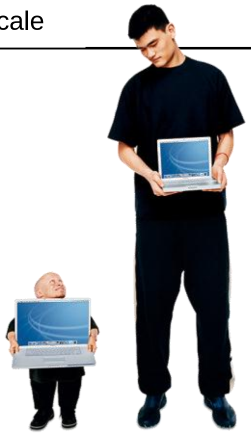
slide credit: S. Ullman

Challenges: occlusion

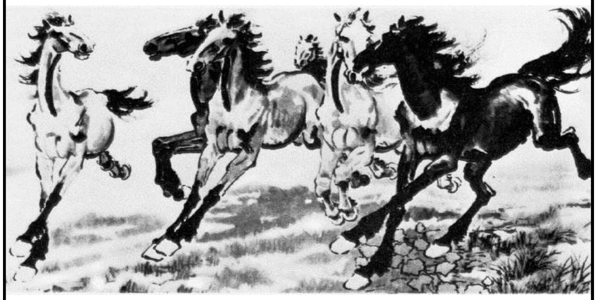


Magritte, 1957

Challenges: scale

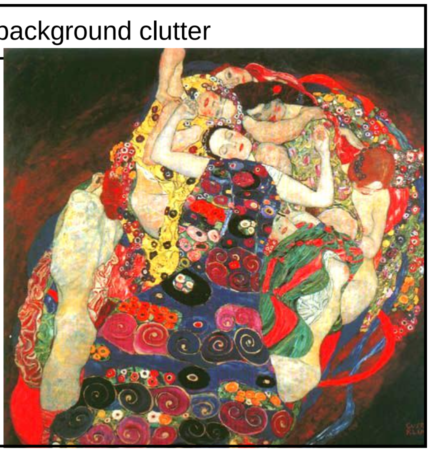


Challenges: deformation



Xu, Beihong 1943

Challenges: background clutter



Klimt, 1913

Challenges: intra-class variation

Let's start simple

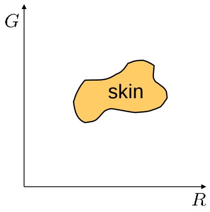
Today

- skin detection
- face detection with adaboost

Face detection

How to tell if a face is present?

One simple method: skin detection



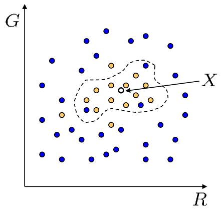
Skin pixels have a distinctive range of colors

- Corresponds to region(s) in RGB color space
  - for visualization, only R and G components are shown above

Skin classifier

- A pixel X = (R,G,B) is skin if it is in the skin region
- But how to find this region?

Skin detection



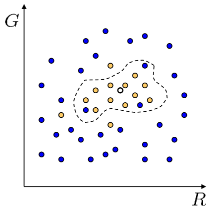
Learn the skin region from examples

- Manually label pixels in one or more “training images” as skin or not skin
- Plot the training data in RGB space
  - skin pixels shown in orange, non-skin pixels shown in blue
  - some skin pixels may be outside the region, non-skin pixels inside. Why?

Skin classifier

- Given X = (R,G,B): how to determine if it is skin or not?

Skin classification techniques



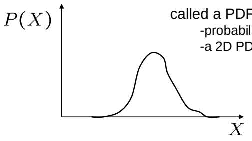
Skin classifier

- Given X = (R,G,B): how to determine if it is skin or not?
- Nearest neighbor
  - find labeled pixel closest to X
  - choose the label for that pixel
- Data modeling
  - fit a model (curve, surface, or volume) to each class
- Probabilistic data modeling
  - fit a probability model to each class

Probability

Basic probability

- X is a random variable
- P(X) is the probability that X achieves a certain value



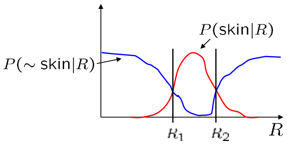
called a PDF
-probability distribution/density function
-a 2D PDF is a surface, 3D PDF is a volume

- $0 \leq P(X) \leq 1$
- $\int_{-\infty}^{\infty} P(X)dX = 1$ or $\sum P(X) = 1$

  continuous X discrete X

- Conditional probability: P(X | Y)
  - probability of X given that we already know Y

Probabilistic skin classification



Now we can model uncertainty

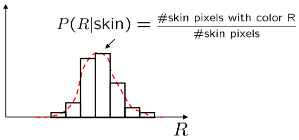
- Each pixel has a probability of being skin or not skin
  - $P(\sim \text{skin}|R) = 1 - P(\text{skin}|R)$

Skin classifier

- Given X = (R,G,B): how to determine if it is skin or not?
- Choose interpretation of highest probability
  - set X to be a skin pixel if and only if $R_1 < X \leq R_2$

Where do we get $P(\text{skin}|R)$ and $P(\sim \text{skin}|R)$?

Learning conditional PDF's

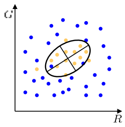


$P(R|\text{skin}) = \frac{\#\text{skin pixels with color R}}{\#\text{skin pixels}}$

We can calculate P(R | skin) from a set of training images

- It is simply a histogram over the pixels in the training images
  - each bin R_i contains the proportion of skin pixels with color R_i

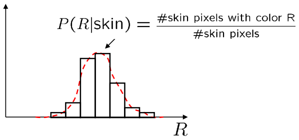
This doesn't work as well in higher-dimensional spaces. Why not?



Approach: fit parametric PDF functions

- common choice is rotated Gaussian
  - center $\mathbf{c} = \overline{X}$
  - covariance $\sum_X (X - \overline{X})(X - \overline{X})^T$
    - » orientation, size defined by eigenvecs, eigenvals

Learning conditional PDF's



$P(R|\text{skin}) = \frac{\#\text{skin pixels with color R}}{\#\text{skin pixels}}$

We can calculate P(R | skin) from a set of training images

- It is simply a histogram over the pixels in the training images
  - each bin R_i contains the proportion of skin pixels with color R_i

But this isn't quite what we want

- Why not? How to determine if a pixel is skin?
- We want P(skin | R) not P(R | skin)
- How can we get it?

Bayes rule

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}$$

In terms of our problem:

$$P(\text{skin}|R) = \frac{\overset{\substack{\text{what we measure} \\ \textbf{(likelihood)}}}{P(R|\text{skin})}\ \overset{\substack{\text{domain knowledge} \\ \textbf{(prior)}}}{P(\text{skin})}}{\underset{\substack{\textbf{normalization}\text{ term}}}{P(R)}}$$

what we want (**posterior**)

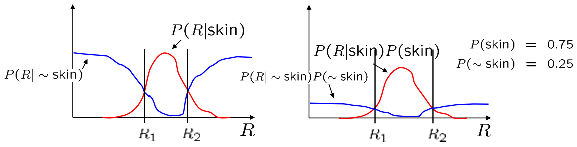
normalization term

$$P(R) = P(R|\text{skin})P(\text{skin}) + P(R|\sim\text{skin})P(\sim\text{skin})$$

The prior: P(skin)

- Could use domain knowledge
  - P(skin) may be larger if we know the image contains a person
  - for a portrait, P(skin) may be higher for pixels in the center
- Could learn the prior from the training set. How?
  - P(skin) may be proportion of skin pixels in training set

Bayesian estimation



likelihood

posterior (unnormalized)

Bayesian estimation = minimize probability of misclassification

- Goal is to choose the label (skin or ~skin) that maximizes the posterior
  - this is called **Maximum A Posteriori (MAP) estimation**
- Suppose the prior is uniform: P(skin) = P(~skin) = 0.5
  - in this case $P(\text{skin}|R) = cP(R|\text{skin}), \quad P(\sim\text{skin}|R) = cP(R|\sim\text{skin})$
  - maximizing the posterior is equivalent to maximizing the likelihood
    - » $P(\text{skin}|R) > P(\sim\text{skin}|R)$ if and only if $P(R|\text{skin}) > P(R|\sim\text{skin})$
  - this is called **Maximum Likelihood (ML) estimation**

Skin detection results

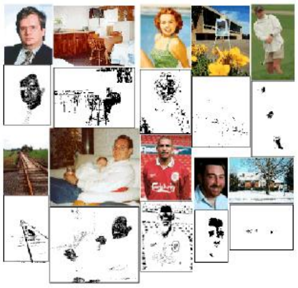
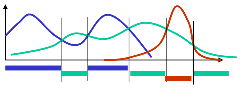


Figure 25.3. The figure shows a variety of images together with the output of the skin detector of Jones and Rehg applied to the image. Pixels marked black are skin pixels, and white are background. Notice that this process is relatively effective, and could certainly be used to focus attention on, say, faces and hands. *Figure from "Statistical color models with application to skin detection," M.J. Jones and J. Rehg, Proc. Computer Vision and Pattern Recognition, 1999 © 1999, IEEE*

General classification

This same procedure applies in more general circumstances

- More than two classes
- More than one dimension



Example: face detection

- Here, X is an image region
  - dimension = # pixels
  - each face can be thought of as a point in a high dimensional space

H. Schneiderman, T. Kanade. "A Statistical Method for 3D Object Detection Applied to Faces and Cars". IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000)
http://www-2.cs.cmu.edu/afs/cs.cmu.edu/user/hws/www/CVPR00.pdf

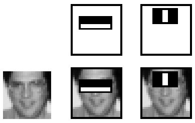
H. Schneiderman and T.Kanade

Issues: metrics

What's the best way to compare images?

- need to define appropriate features
- depends on goal of recognition task

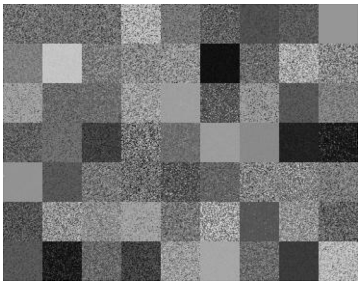
exact matching
complex features work well
(SIFT, MOPS, etc.)



classification/detection
simple features work well
(Viola/Jones, etc.)

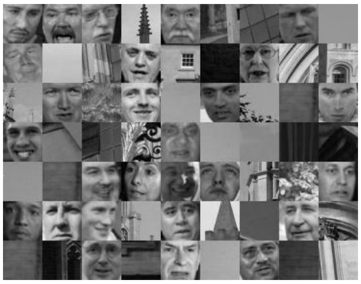
Issues: metrics

What do you see?



Issues: metrics

What do you see?



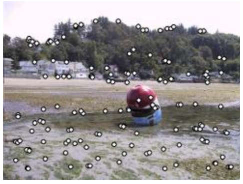
Metrics

Lots more feature types that we haven't mentioned

- moments, statistics
  - metrics: Earth mover's distance, ...
- edges, curves
  - metrics: Hausdorff, shape context, ...
- 3D: surfaces, spin images
  - metrics: chamfer (ICP)
- ...

We'll discuss more in Part 2

Issues: feature selection



If all you have is one image:
non-maximum suppression, etc.

If you have a training set of images:
AdaBoost, etc.

Issues: data modeling

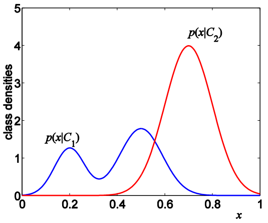
Generative methods

- model the “shape” of each class
  - histograms, PCA, mixtures of Gaussians
  - graphical models (HMM’s, belief networks, etc.)
  - ...

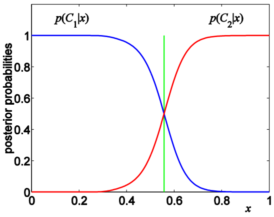
Discriminative methods

- model boundaries between classes
  - perceptrons, neural networks
  - support vector machines (SVM’s)

Generative vs. Discriminative



Generative Approach
model individual classes, priors



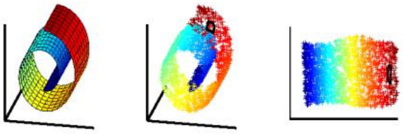
Discriminative Approach
model posterior directly

from Chris Bishop

Issues: dimensionality

What if your space isn’t *flat*?

- PCA may not help



Nonlinear methods
LLE, MDS, etc.

Issues: speed

Case study: Viola Jones face detector

Next few slides adapted Grauman & Liebe's tutorial

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

Also see Paul Viola's talk (video)

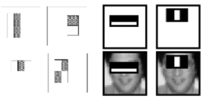
- http://www.cs.washington.edu/education/courses/577/04sp/contents.html#DM

Face detection

Where are the faces? Not who they are, that's recognition or identification.

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

Visual Object Recognition Tutorial

Viola & Jones, CVPR 2001

K. Grauman, B. Leibe

43

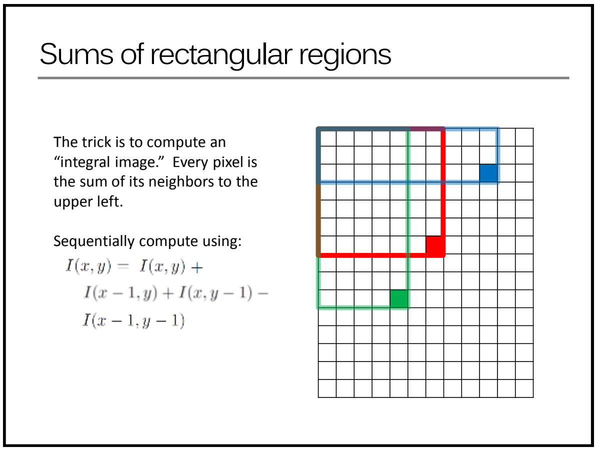
Sums of rectangular regions

How do we compute the sum of the pixels in the red box?

After some pre-computation, this can be done in constant time for any box.

This "trick" is commonly used for computing Haar wavelets (a fundemental building block of many object recognition approaches.)

243	239	240	225	206	185	188	218	211	206	216	225
242	239	218	110	67	31	34	152	213	206	208	221
243	242	123	58	94	82	132	77	108	208	208	215
235	217	115	212	243	236	247	139	91	209	208	211
233	208	131	222	219	226	196	114	74	208	213	214
232	217	131	116	77	150	69	56	52	201	228	223
232	232	182	186	184	179	159	123	93	232	235	235
232	236	201	154	216	133	129	81	175	252	241	240
235	238	230	128	172	138	65	63	234	249	241	245
237	236	247	143	59	78	10	94	255	248	247	251
234	237	245	193	55	33	115	144	213	255	253	251
248	245	161	128	149	109	138	65	47	156	239	255
190	107	39	102	94	73	114	58	17	7	51	137
23	32	33	148	168	203	179	43	27	17	12	8
17	26	12	160	255	255	109	22	26	19	35	24

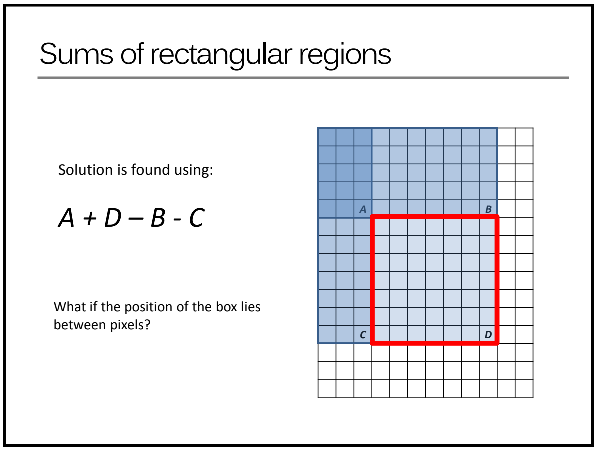


Sums of rectangular regions

The trick is to compute an "integral image." Every pixel is the sum of its neighbors to the upper left.

Sequentially compute using:

$$I(x,y) = I(x,y) + I(x-1,y) + I(x,y-1) - I(x-1,y-1)$$

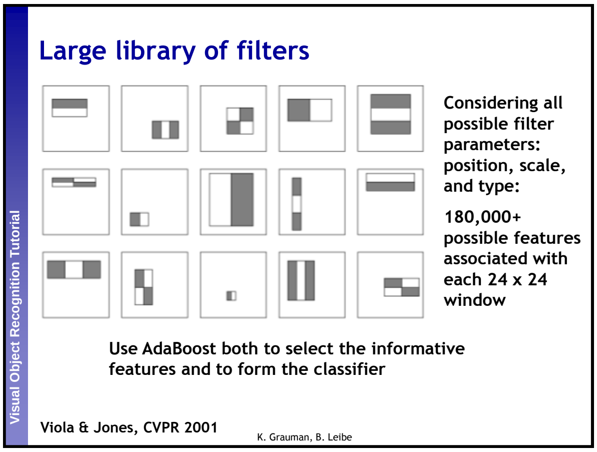


Sums of rectangular regions

Solution is found using:

A + D – B - C

What if the position of the box lies between pixels?



Large library of filters

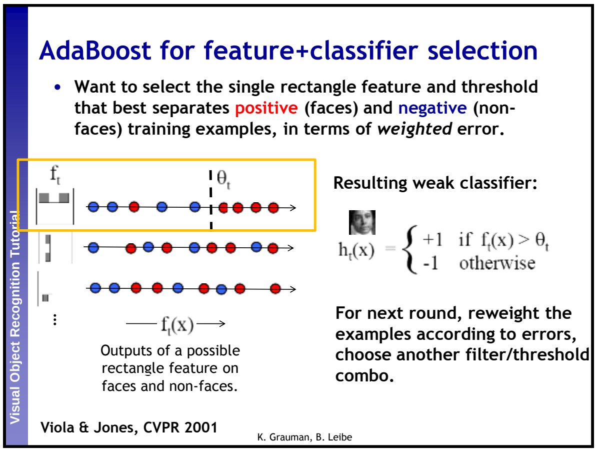
Considering all possible filter parameters: position, scale, and type:

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

K. Grauman, B. Leibe



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.

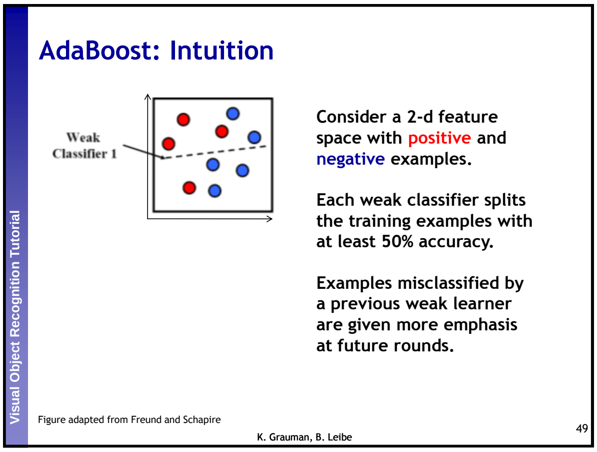
Resulting weak classifier:

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

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

Viola & Jones, CVPR 2001

K. Grauman, B. Leibe



AdaBoost: Intuition

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

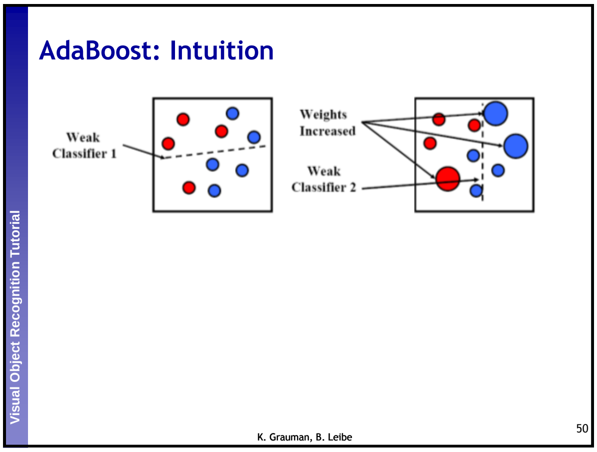
Each weak classifier splits the training examples with at least 50% accuracy.

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

Figure adapted from Freund and Schapire

K. Grauman, B. Leibe

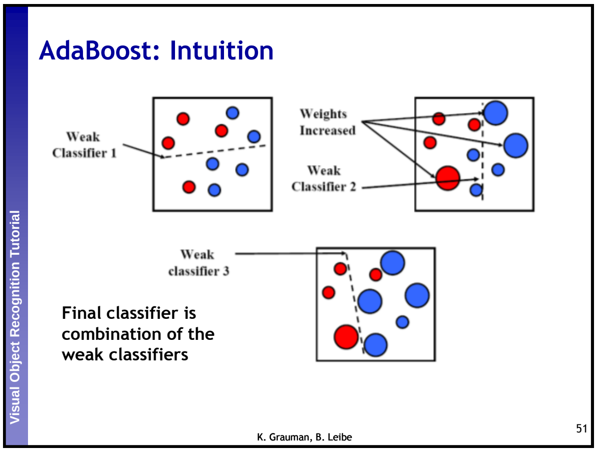
49



AdaBoost: Intuition

K. Grauman, B. Leibe

50

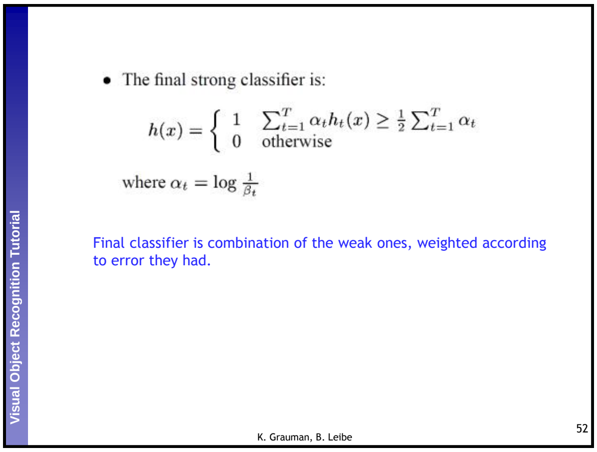


AdaBoost: Intuition

Final classifier is combination of the weak classifiers

K. Grauman, B. Leibe

51



Visual Object Recognition Tutorial

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

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

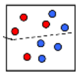
K. Grauman, B. Leibe

52

AdaBoost Algorithm

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

Start with uniform weights on training examples



$\{x_1, \ldots x_n\}$

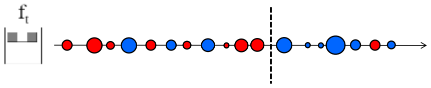
For T rounds

Find the best threshold and polarity for each feature, and return error.

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

Picking the best classifier

Efficient single pass approach:



At each sample compute:

$$e = \min\left(S + (T - S),\ S + (T - S) \right)$$

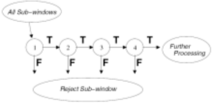
Find the minimum value of e, and use the value of the corresponding sample as the threshold.

S = sum of samples below the current sample
T = total sum of all samples

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.,

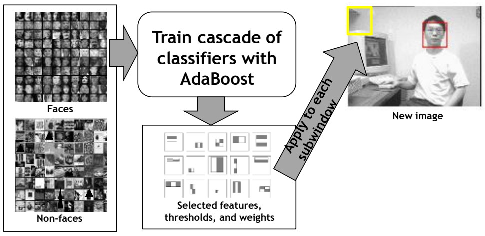
- Filter for promising regions with an initial inexpensive classifier
- Build a chain of classifiers, choosing cheap ones with low false negative rates early in the chain



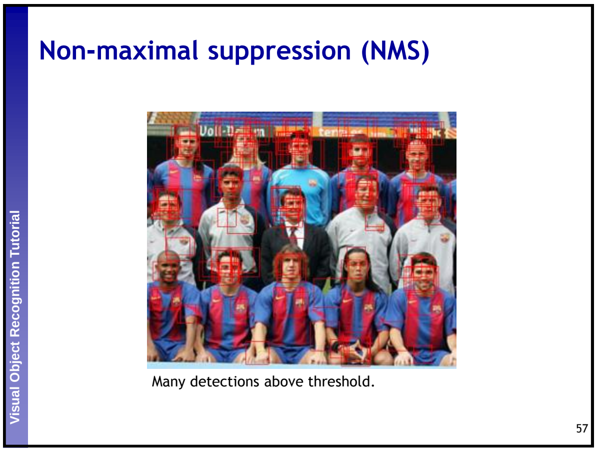
Fleuret & Geman, IJCV 2001
Rowley et al., PAMI 1998
Viola & Jones, CVPR 2001

Figure from Viola & Jones CVPR 2001

Viola-Jones Face Detector: Summary


Faces
Non-faces
Train cascade of classifiers with AdaBoost
Selected features, thresholds, and weights
Apply to each subwindow
New image

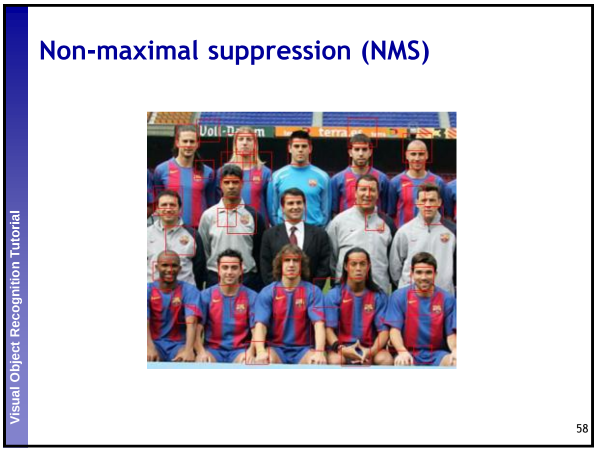
- Train with 5K positives, 350M negatives
- Real-time detector using 38 layer cascade
- 6061 features in final layer
- [Implementation available in OpenCV: http://www.intel.com/technology/computing/opencv/]



Non-maximal suppression (NMS)

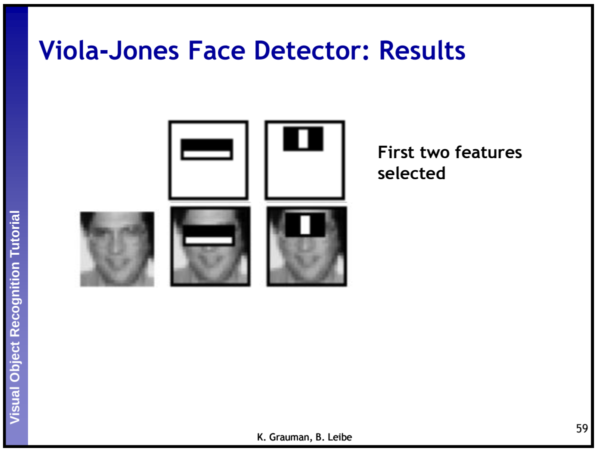
Many detections above threshold.

Visual Object Recognition Tutorial



Non-maximal suppression (NMS)

Visual Object Recognition Tutorial

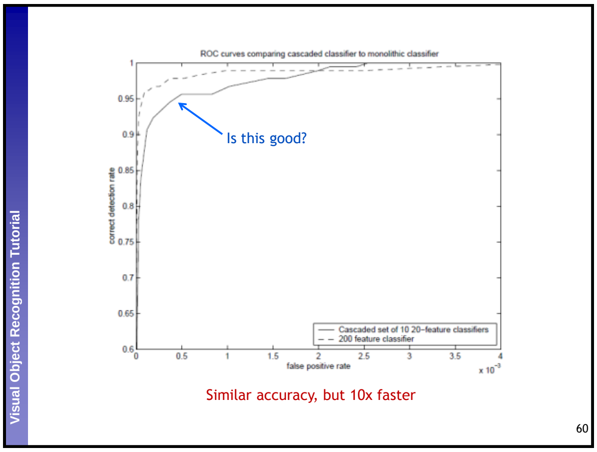


Viola-Jones Face Detector: Results

First two features selected

K. Grauman, B. Leibe

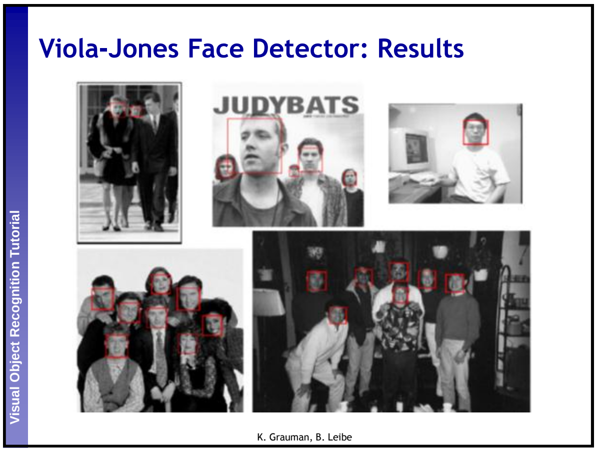
Visual Object Recognition Tutorial



Is this good?

Similar accuracy, but 10x faster

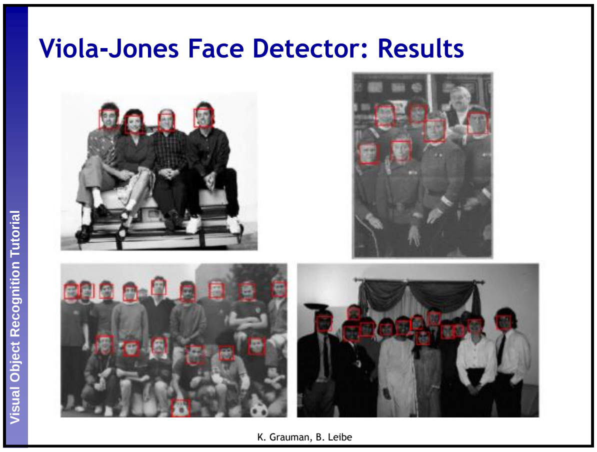
Visual Object Recognition Tutorial



Viola-Jones Face Detector: Results

Visual Object Recognition Tutorial

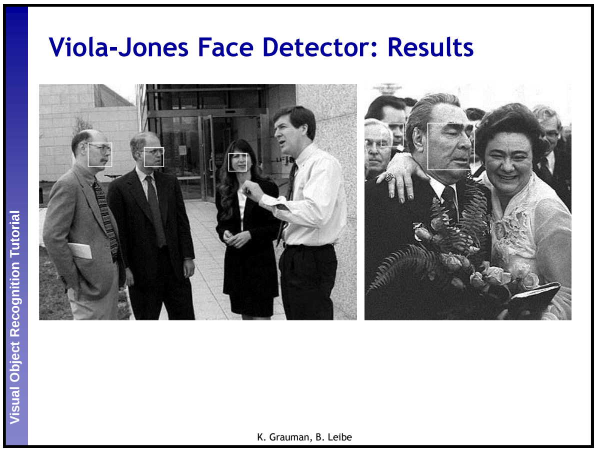
K. Grauman, B. Leibe



Viola-Jones Face Detector: Results

Visual Object Recognition Tutorial

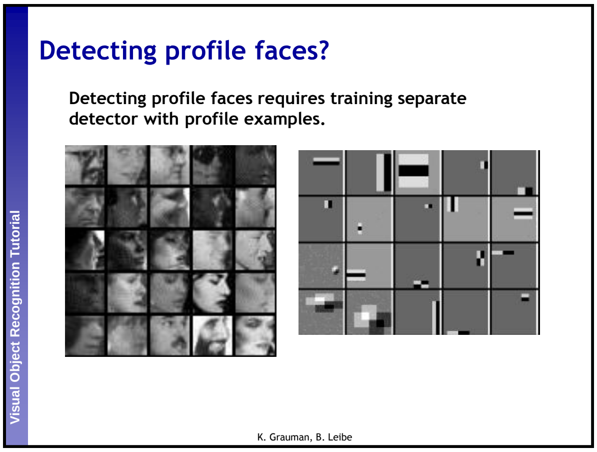
K. Grauman, B. Leibe



Viola-Jones Face Detector: Results

Visual Object Recognition Tutorial

K. Grauman, B. Leibe



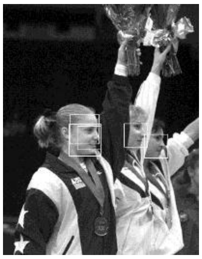
Detecting profile faces?

Detecting profile faces requires training separate detector with profile examples.

Visual Object Recognition Tutorial

K. Grauman, B. Leibe

Viola-Jones Face Detector: Results



Visual Object Recognition Tutorial

Paul

K. Grauman, B. Leibe

Visual Object Recognition Tutorial

http://www.pittpatt.com/face_tracking/

K. Grauman, B. Leibe

66