Object Detection I

COS 429

Princeton University

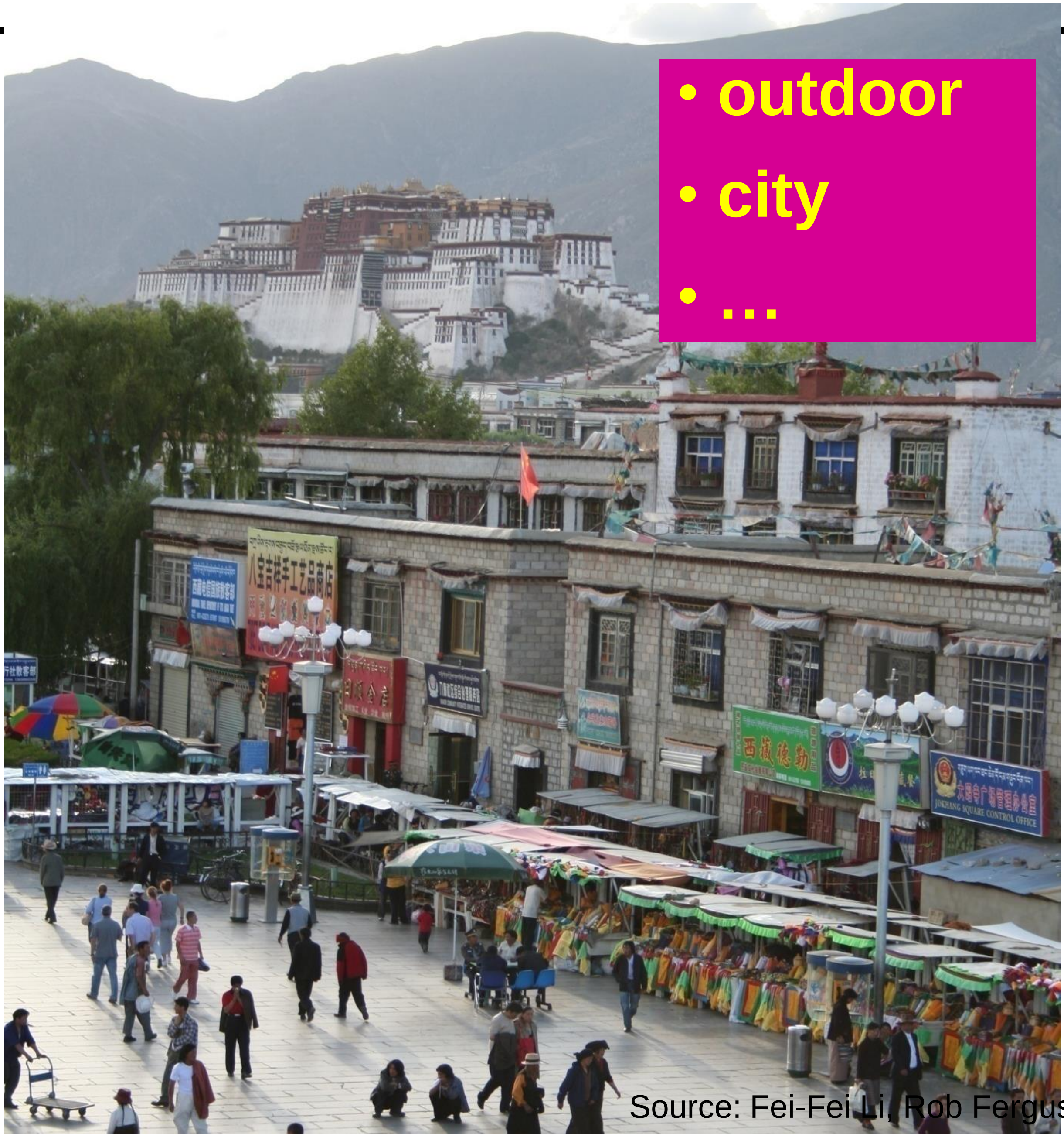
Goal: scene understanding

Source: Fei-Fei Li, Rob Fergus, Antonio Torralba.

Types of scene understanding problems

Source: Fei-Fei Li, Rob Fergus, Antonio Torralba.

Scene categorization

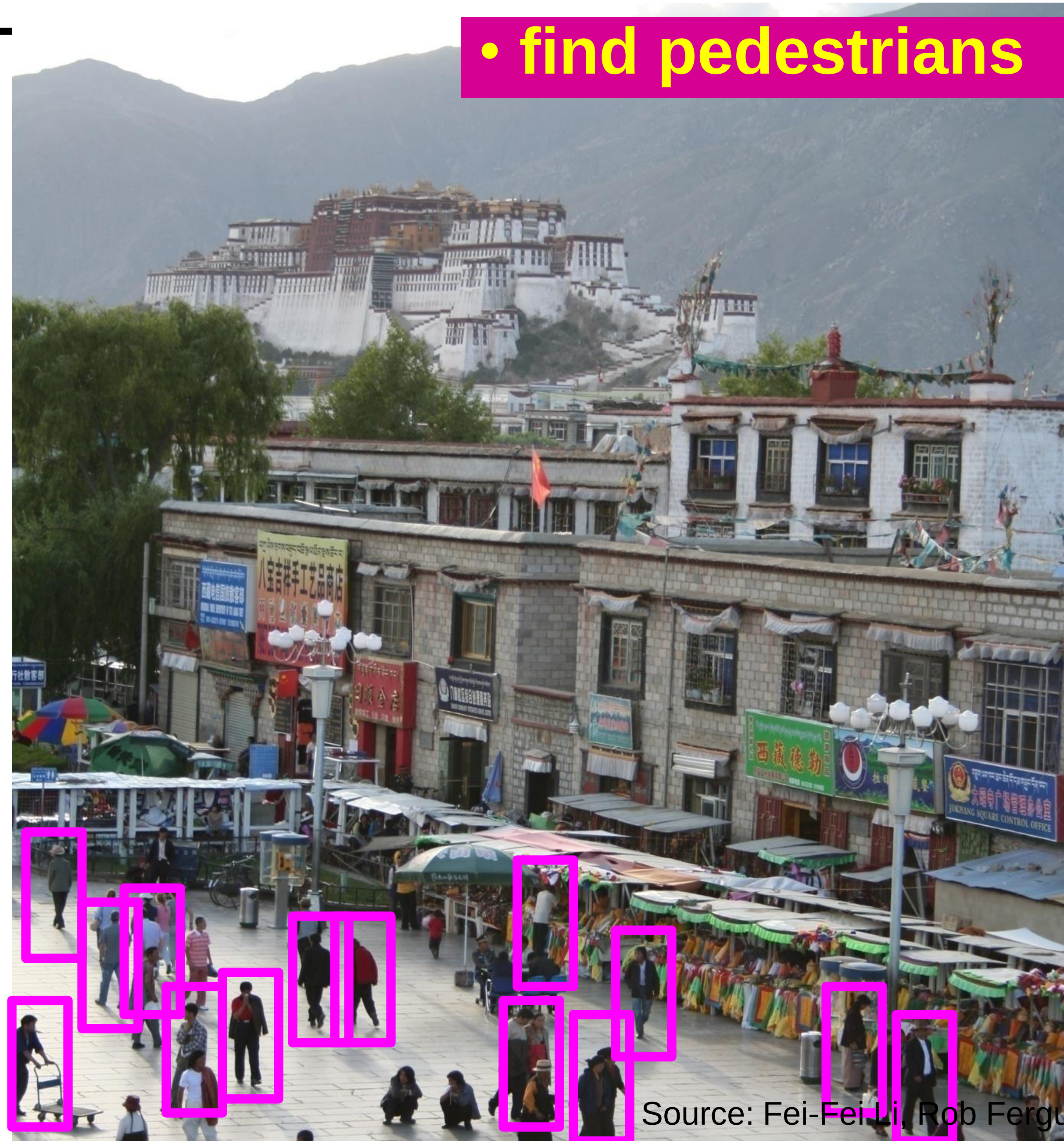


- **outdoor**
- **city**
- **…**

Source: Fei-Fei Li, Rob Fergus, Antonio Torralba.

Object detection

- **find pedestrians**



Source: Fei-Fei Li, Rob Fergus, Antonio Torralba.

Image parsing

Source: Fei-Fei Li, Rob Fergus, Antonio Torralba.

Activity recognition

- **walking**
- **shopping**
- **rolling a cart**
- **sitting**
- **talking**
- **…**

Source: Fei-Fei Li, Rob Fergus, Antonio Torralba.

Identification: is that Potala Palace?

Source: Fei-Fei Li, Rob Fergus, Antonio Torralba.

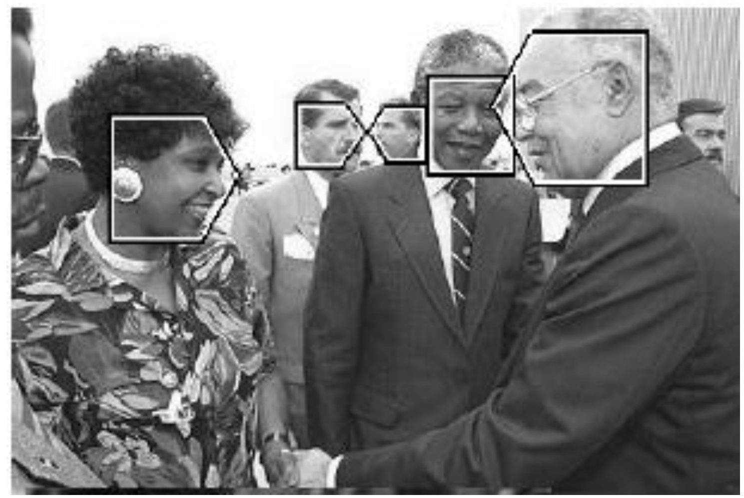
Localization: where was the picture taken?

Source: Fei-Fei Li, Rob Fergus, Antonio Torralba.

Today: object detection

Given an image, find all instances of a basic object category (e.g., car, face, etc.)

- Report the object locations (e.g., bounding boxes) or report that there is none

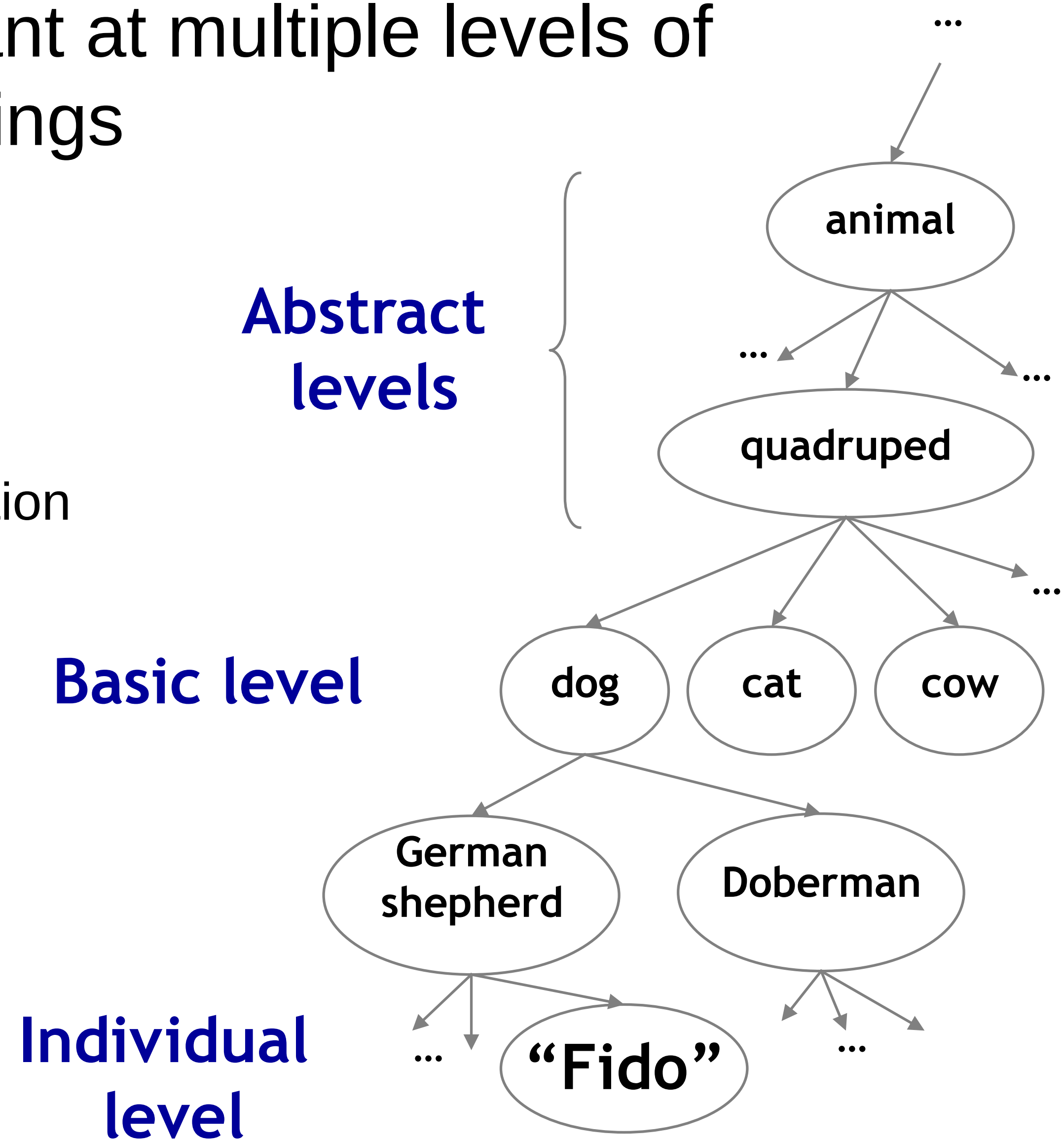


Applications?

Object detection

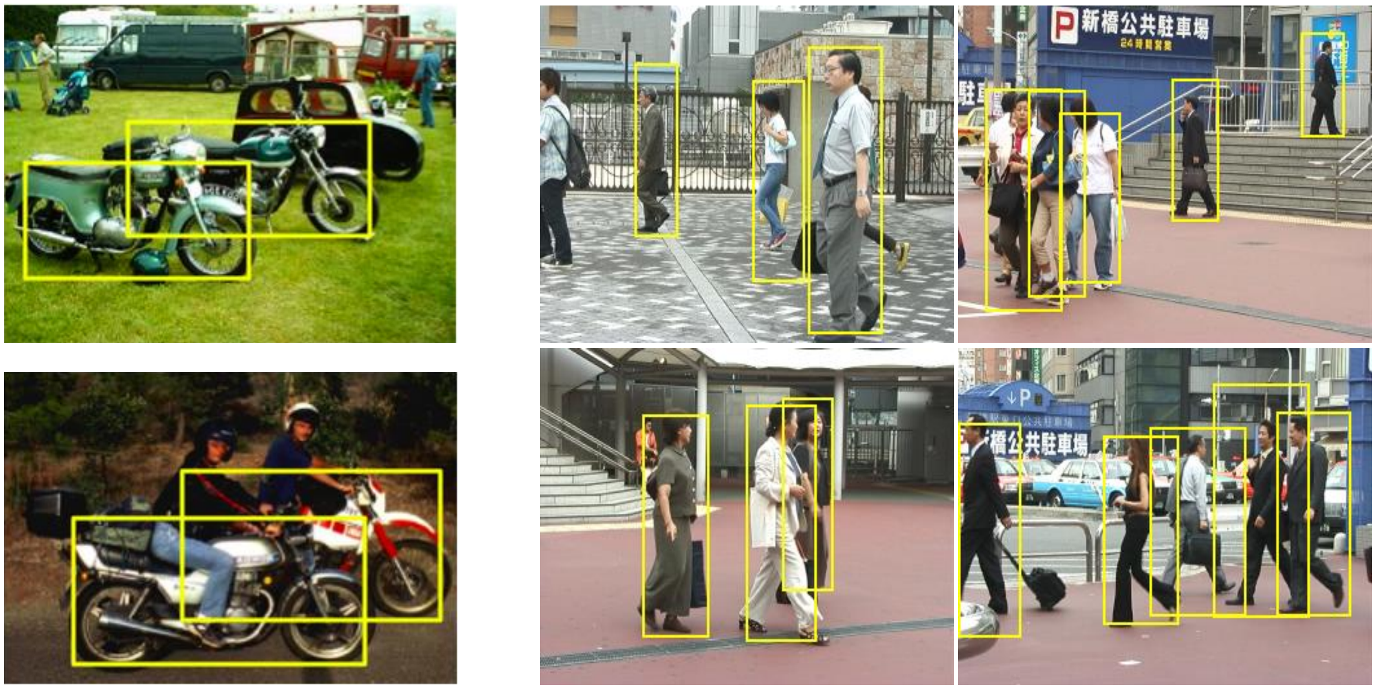
Detection is important at multiple levels of categorical groupings

There is evidence that humans (usually) start with basic-level categorization *before* identification of individuals



K. Grauman, B. Leibe

Challenges: occlusion & clutter

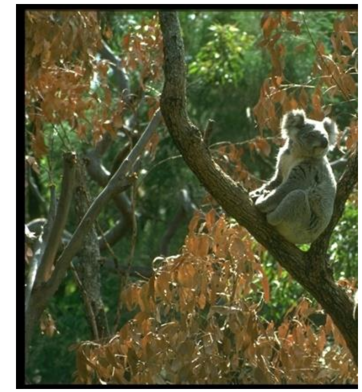


Realistic scenes are crowded, cluttered, have overlapping objects.

Challenges: image variation

Illumination

Object pose



Clutter

Occlusions

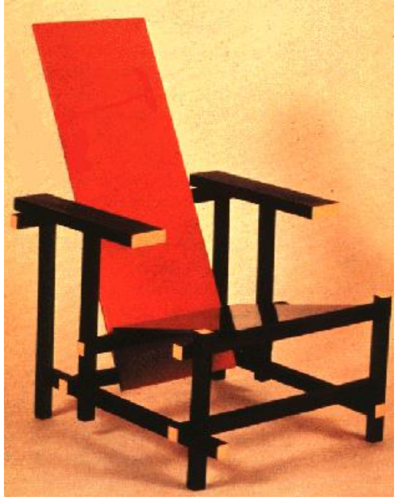
Intra-class appearance

Viewpoint

Kristen Grauman

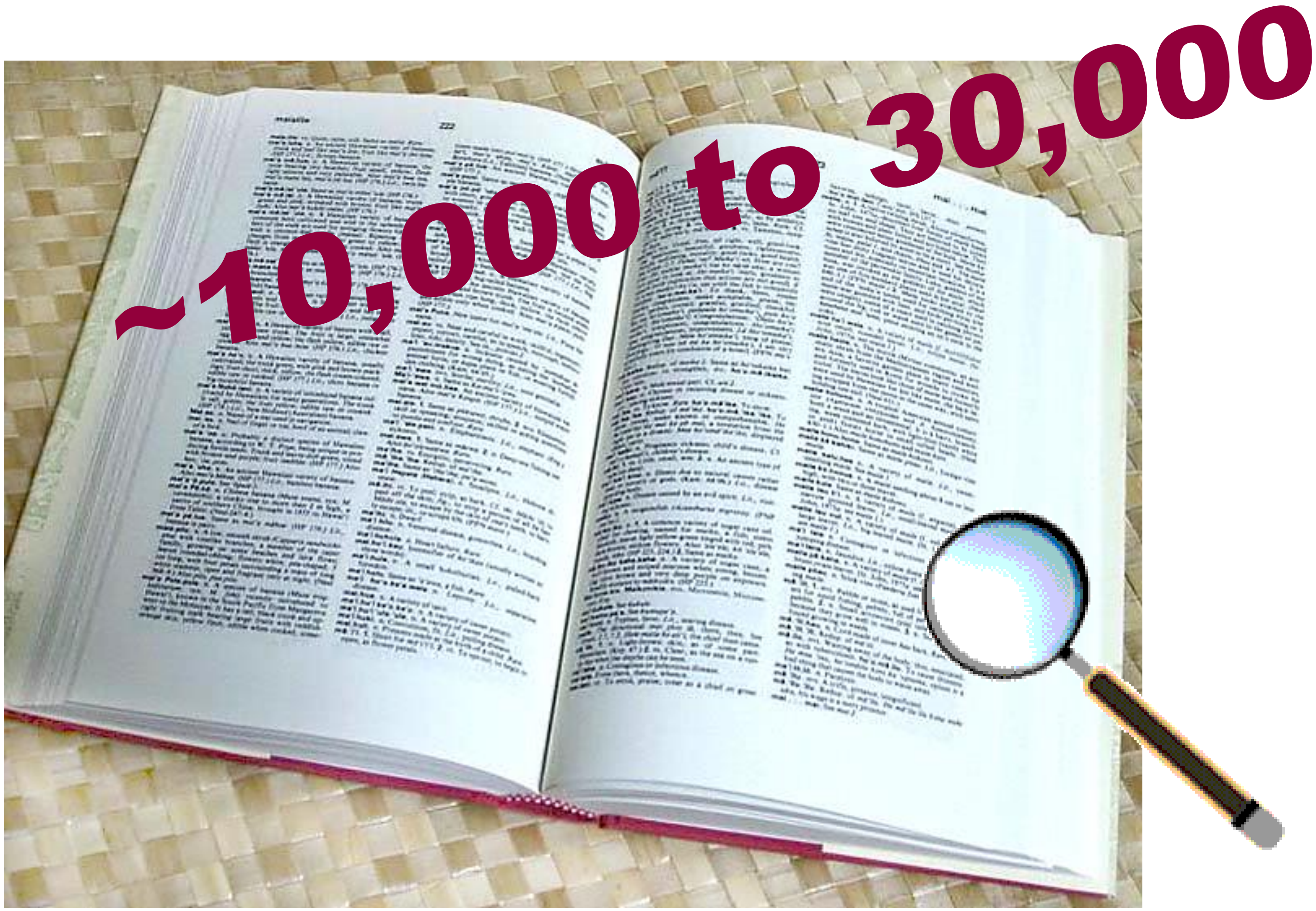
Challenges: intra-class variation



Challenges: complexity

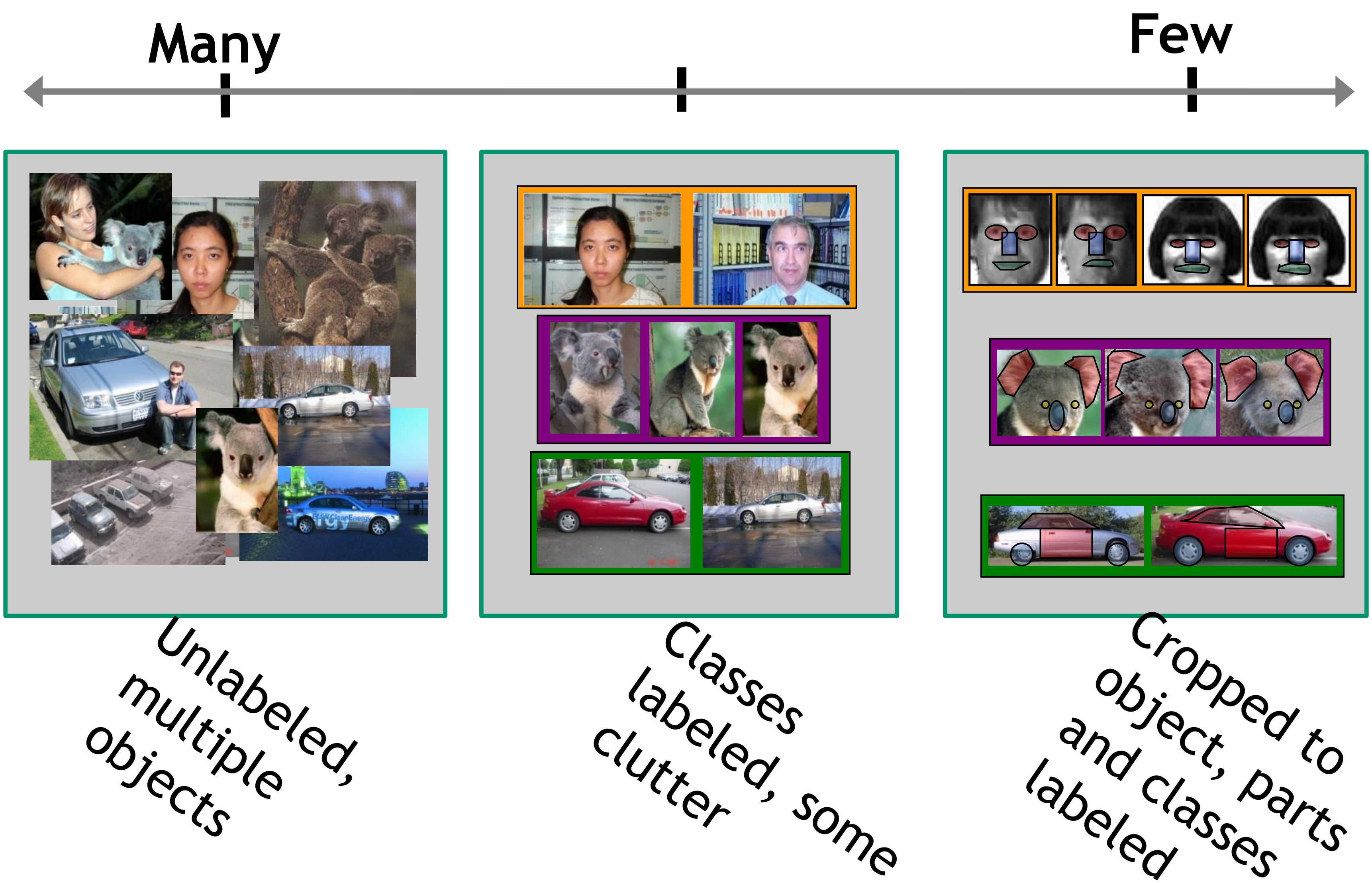
There are many different categories



Source: Fei-Fei Li, Rob Fergus, Antonio Torralba.

Biederman 1987

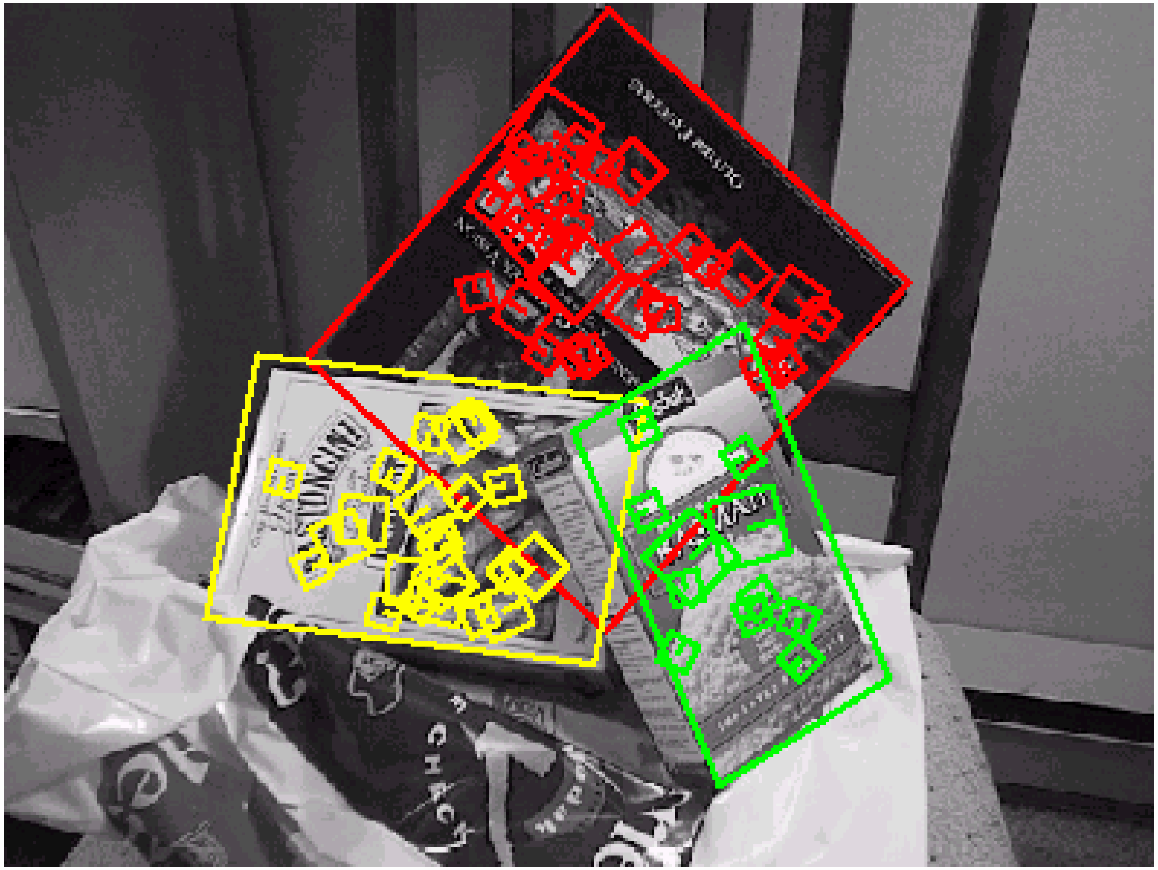
Challenges: limited examples



Kristen Grauman

What works most reliably today

Recognition of flat textured objects



Source: Lana Lazebnik

What works most reliably today

Recognition of flat textured objects

Reading license plates, zip codes, checks

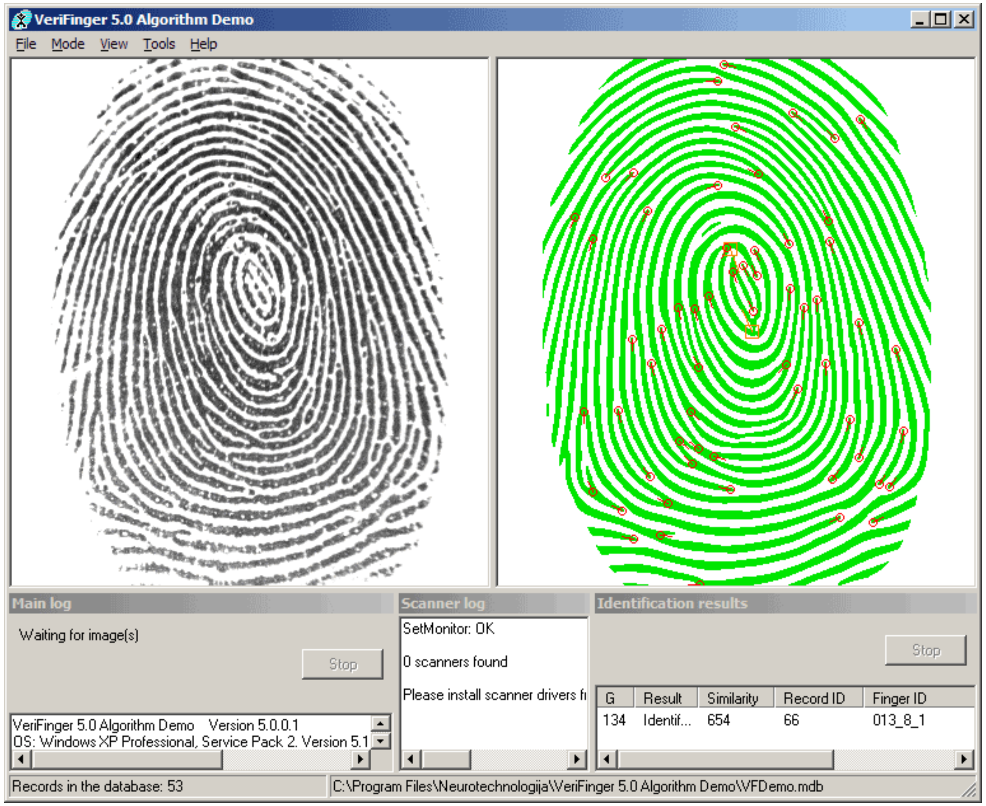
Source: Lana Lazebnik

What works most reliably today

Recognition of flat textured objects

Reading license plates, zip codes, checks

Fingerprint recognition



Source: Lana Lazebnik

What works most reliably today

Recognition of flat textured objects

Reading license plates, zip codes, checks

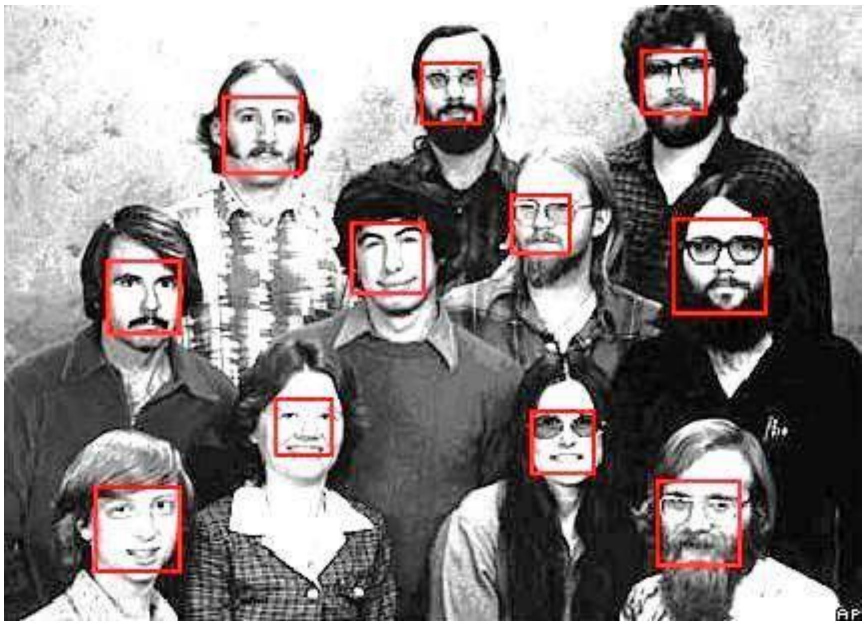
Fingerprint recognition

Face detection

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

Source: Lana Lazebnik

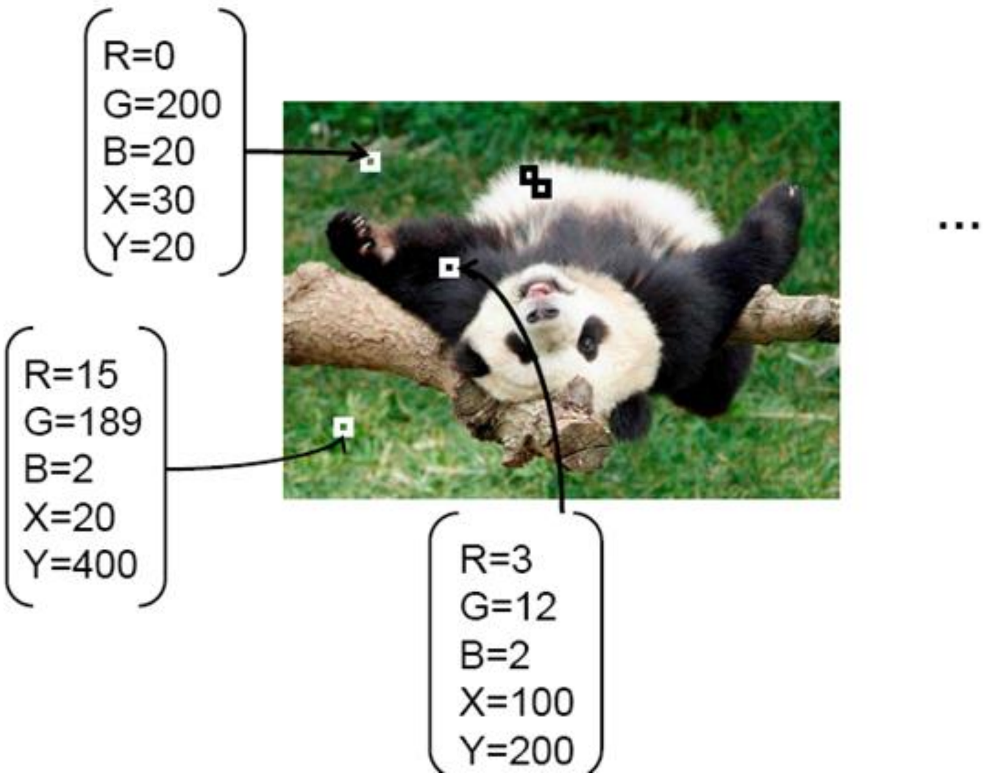
Face detection



Face Detection Methods?

Pixel-based classification

- Basic idea: classify pixels individually as face or not based on their properties (features)

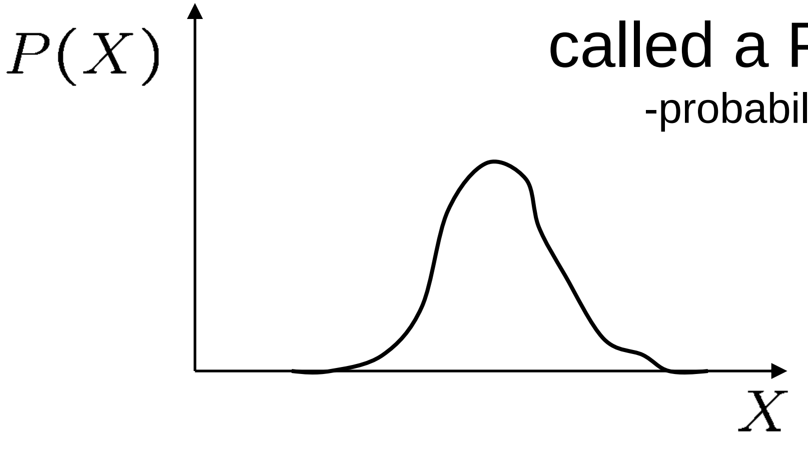


Kristen Grauman

Pixel-based classification

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

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

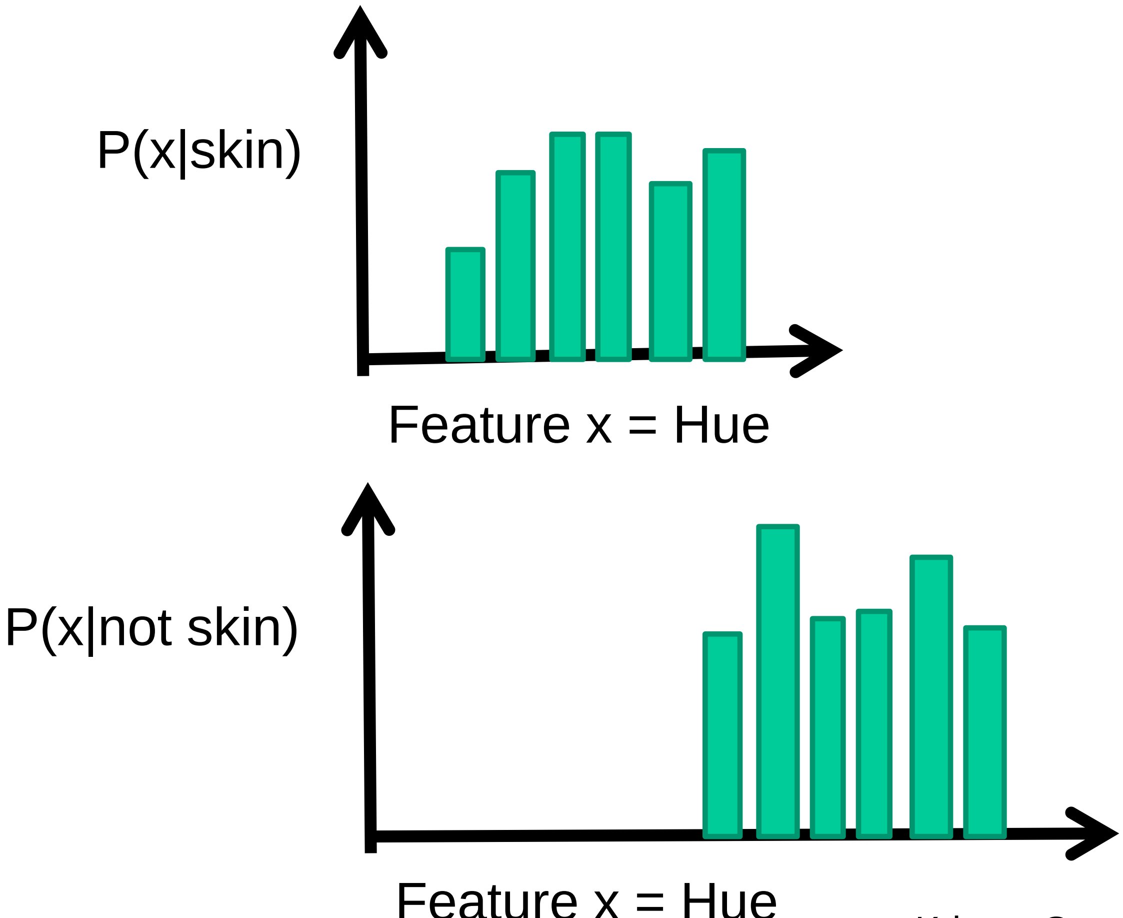
continuous X discrete X

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

Source: Steve Seitz

Example: detecting skin

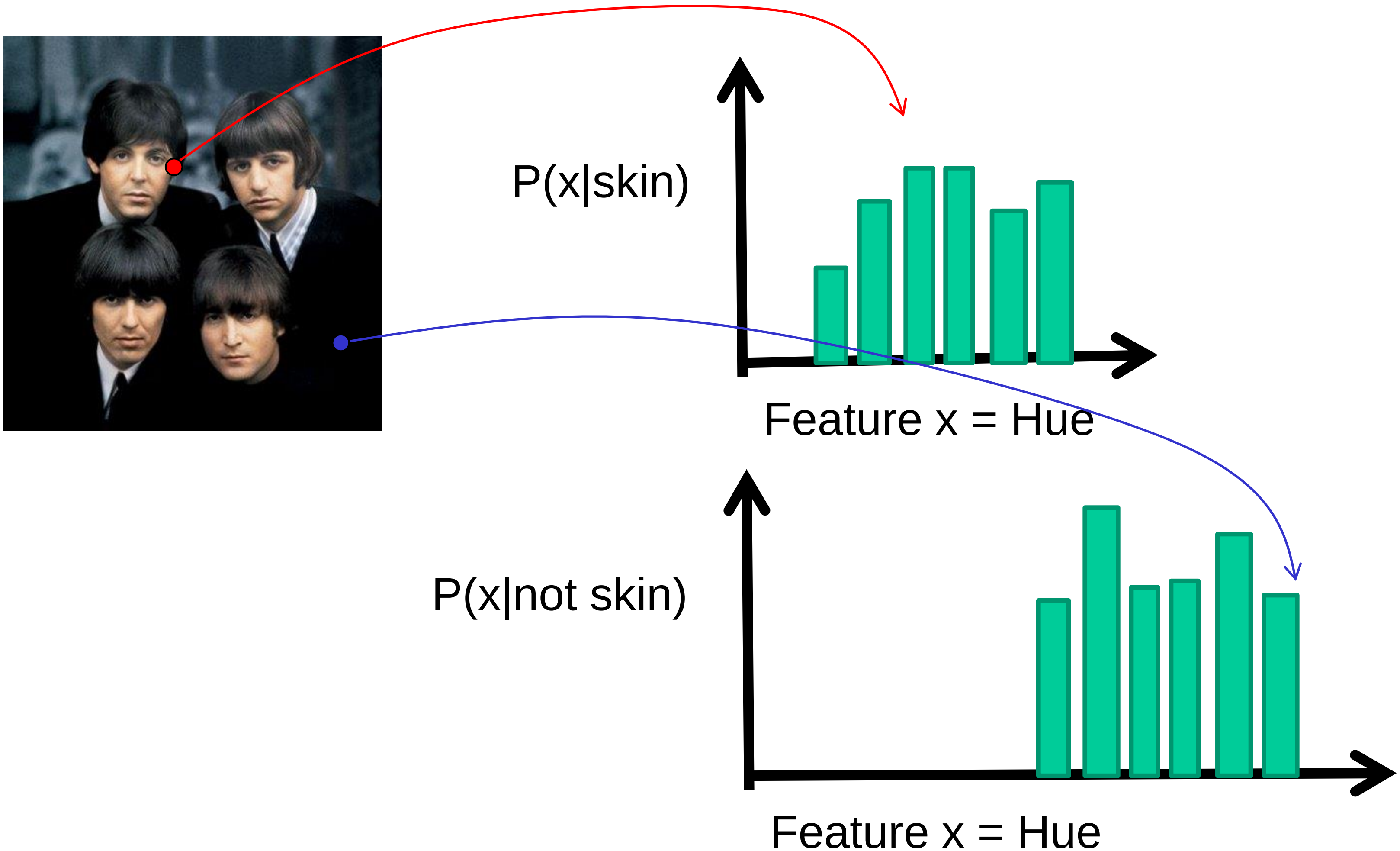
Probability distributions of hues for pixels that are skin and are not skin



Kristen Grauman

Example: detecting skin

For new images, use probability distributions to classify pixels as skin or not



Kristen Grauman

Example: detecting skin

Bayes rule:

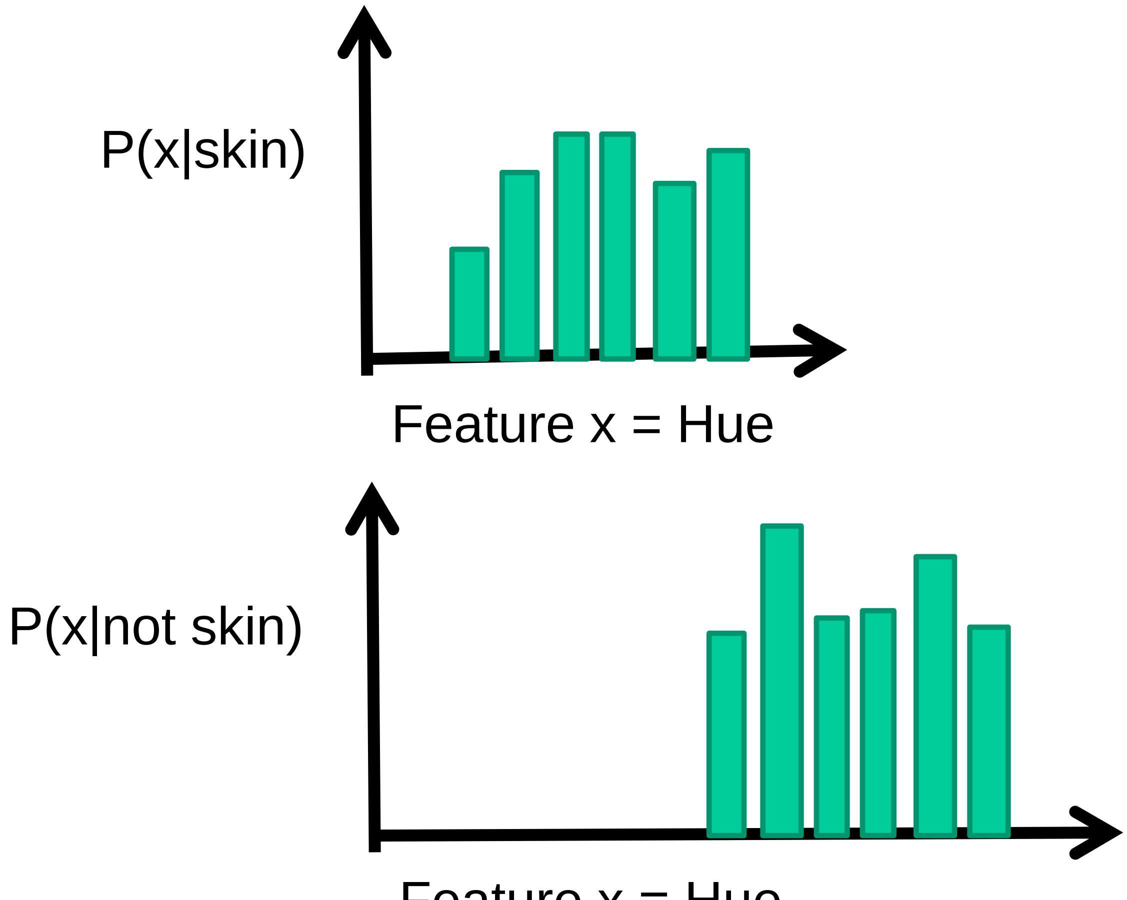
$$\overbrace{P(skin \mid x)}^{\text{posterior}} = \frac{\overbrace{P(x \mid skin)}^{\text{likelihood}}\;\overbrace{P(skin)}^{\text{prior}}}{P(x)}$$

$$P(skin \mid x) \;\alpha\; P(x \mid skin)P(skin)$$

Kristen Grauman

Example: detecting skin

How build likelihood and prior distributions?



Kristen Grauman

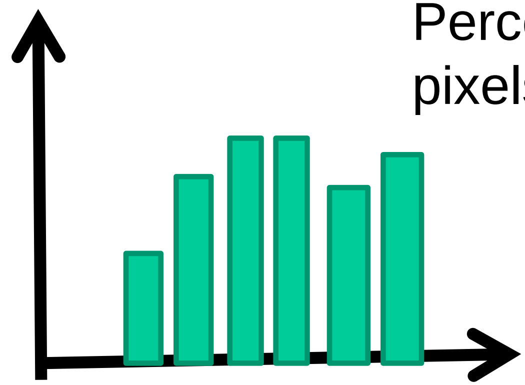
Example: detecting skin

How build likelihood and prior distributions?

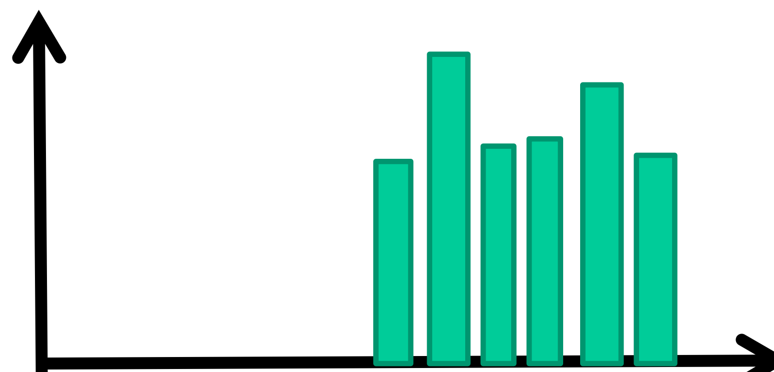
Learn from examples

Labeled examples

Percentage of skin pixels in each bin



Feature x = Hue



Feature x = Hue

Kristen Grauman

Questions

What features should we measure?

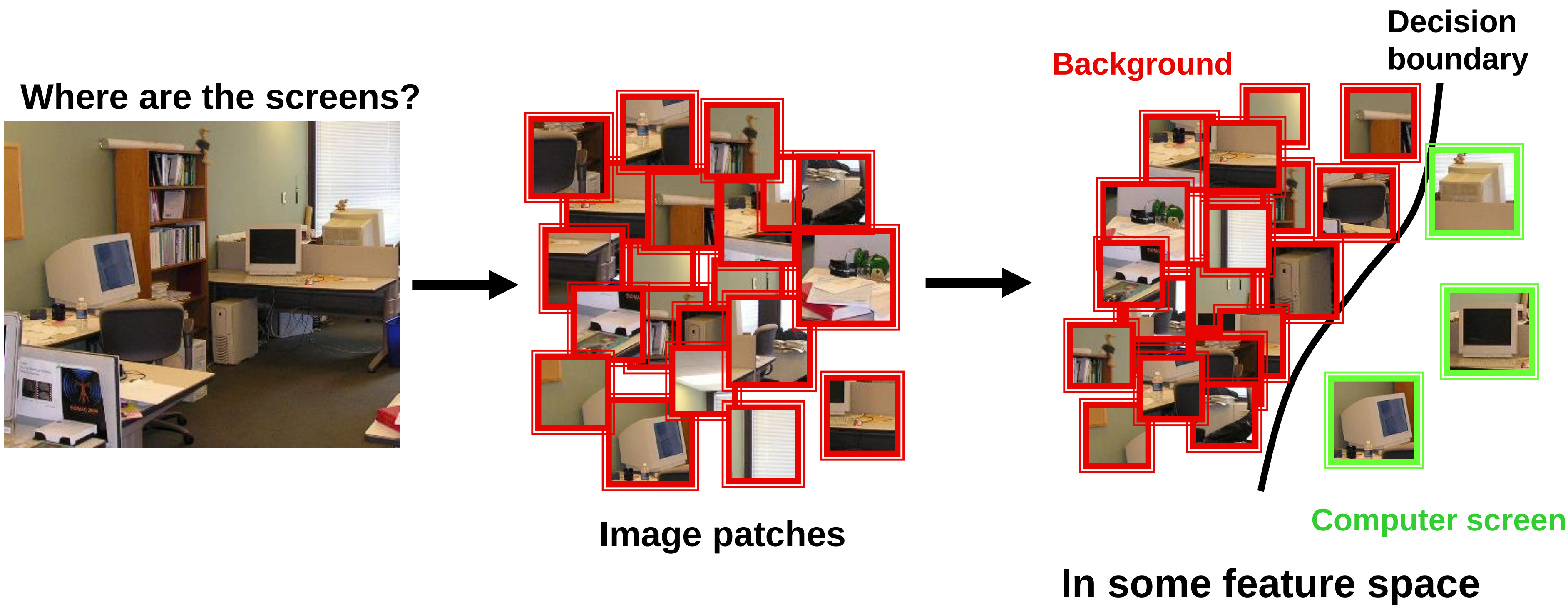
Which positions in the image should we consider?

How should we estimate probability distributions from a limited set of examples?

How can we compute everything quickly?

Window-based classification

The image is partitioned into a set of overlapping windows, features are detected for each window, and then a classifier is used to decide if each window contains an object or not.



Sliding window detection

- Basic idea: slide a window across image and evaluate a detection model at every location

Car/non-car Classifier

Kristen Grauman

Sliding window detection

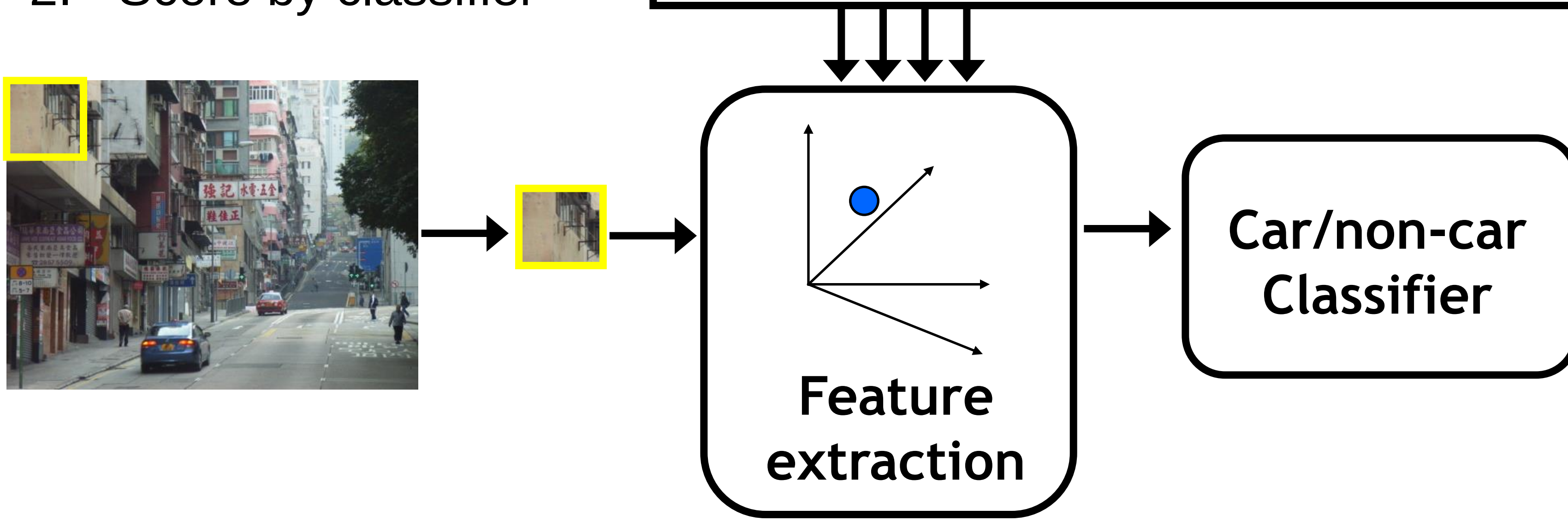
Consider every location at every scale

What the Detector Sees

Sliding window detection

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

Given new image:
1. Slide window
2. Score by classifier

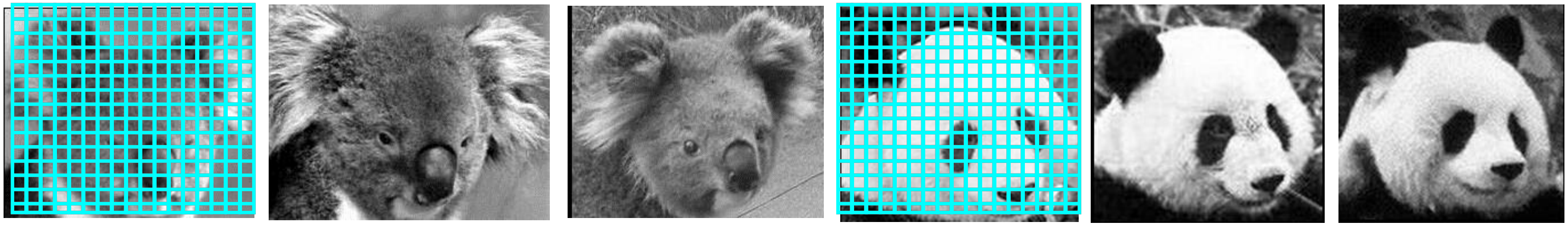


Kristen Grauman

Image features

Intensity-based features:

Pixel-based regions

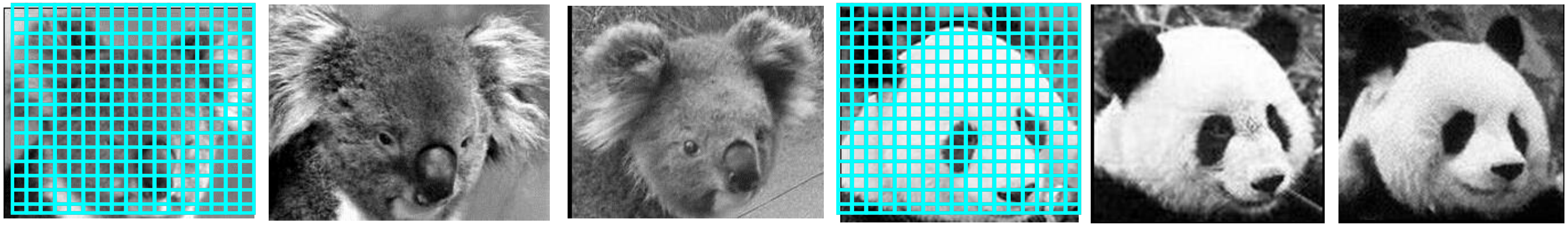


Kristen Grauman

Image features

Intensity-based features:← sensitive to illumination changes

Pixel-based regions

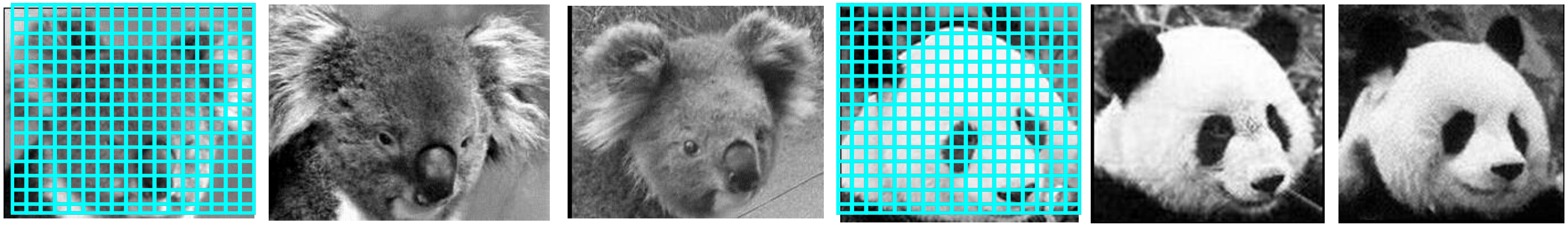


Kristen Grauman

Image features

Intensity-based features: ← sensitive to illumination changes

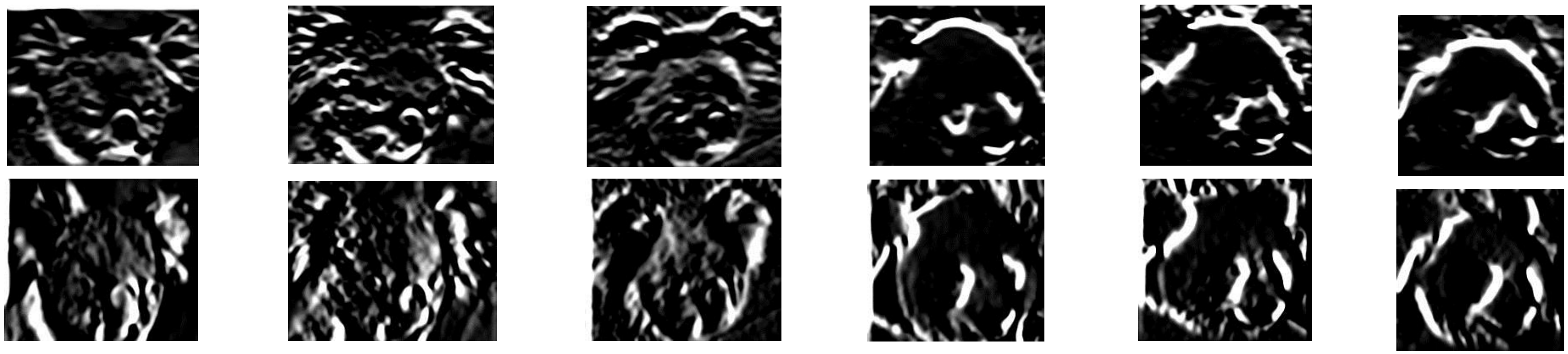
Pixel-based regions ← sensitive to small shifts



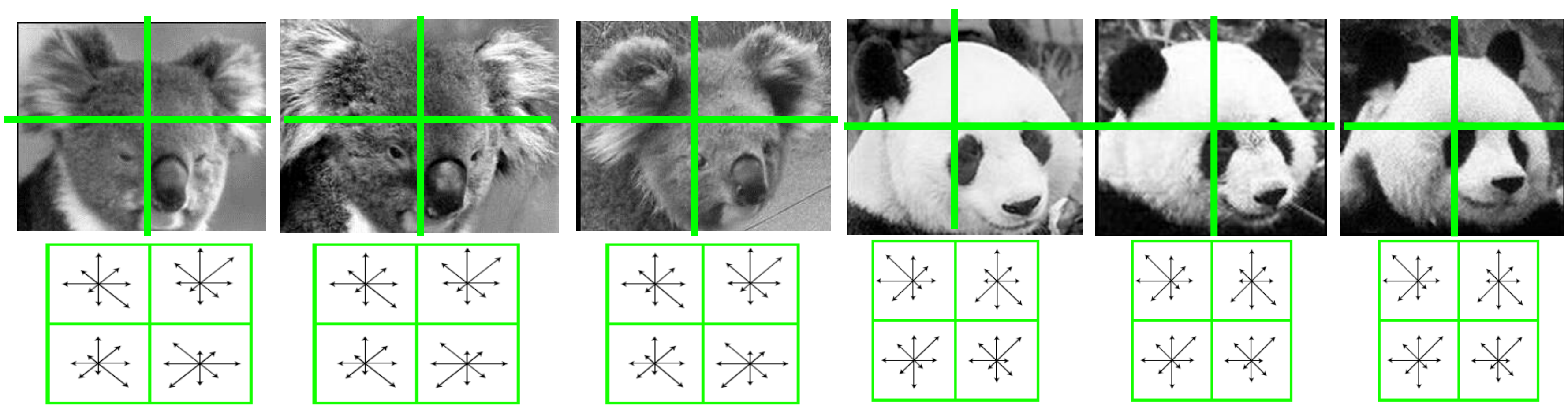
Kristen Grauman

Image features

Better: edges, contours, and (oriented) gradients



Better: block-based features



Kristen Grauman

Challenges of sliding window detection

- Sliding window detector must evaluate tens of thousands of location/scale combinations
  - Need fast computation of features
- Objects are rare: 0–10 per image
  - Try to spend as little time as possible on the non-object windows
  - A megapixel image has ~10^6 pixels and a comparable number of candidate object locations
  - To avoid having a false positive in every image image, our false positive rate has to be less than 10^{-6}

Viola-Jones object 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 object detector

Main ideas:

- 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

Kristen Grauman

Viola-Jones object detector

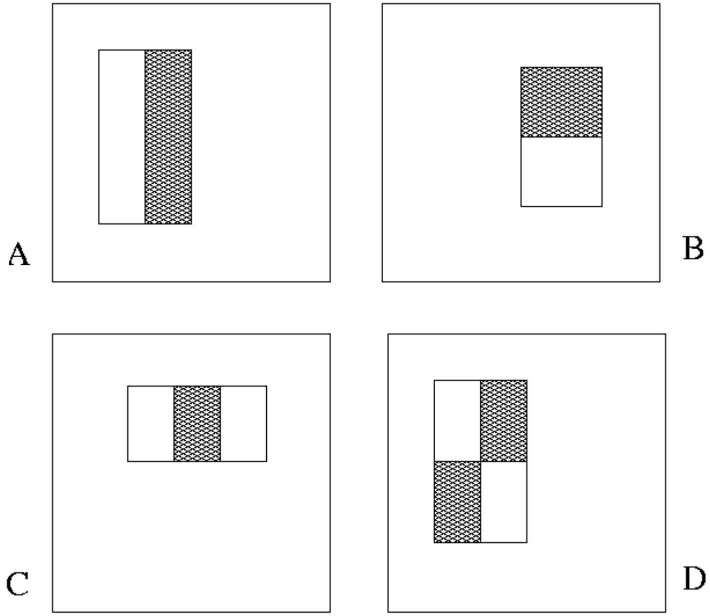
Main ideas:

- 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

Kristen Grauman

Viola-Jones: features

“Rectangle filters”

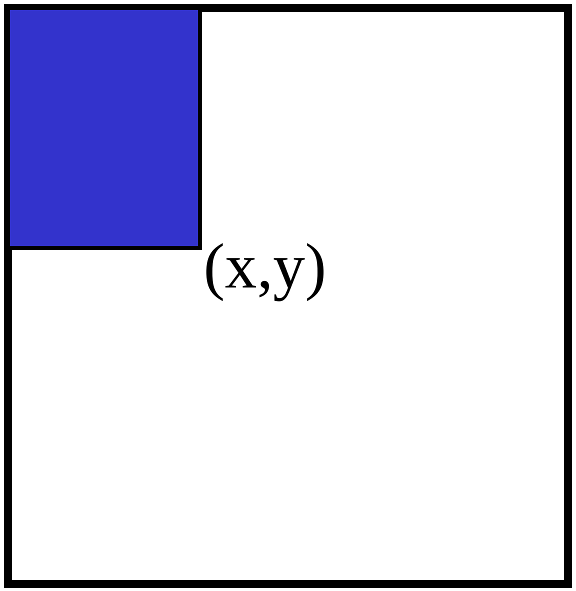


Value = $\sum$ *(pixels in white area) –* $\sum$ *(pixels in black area)*

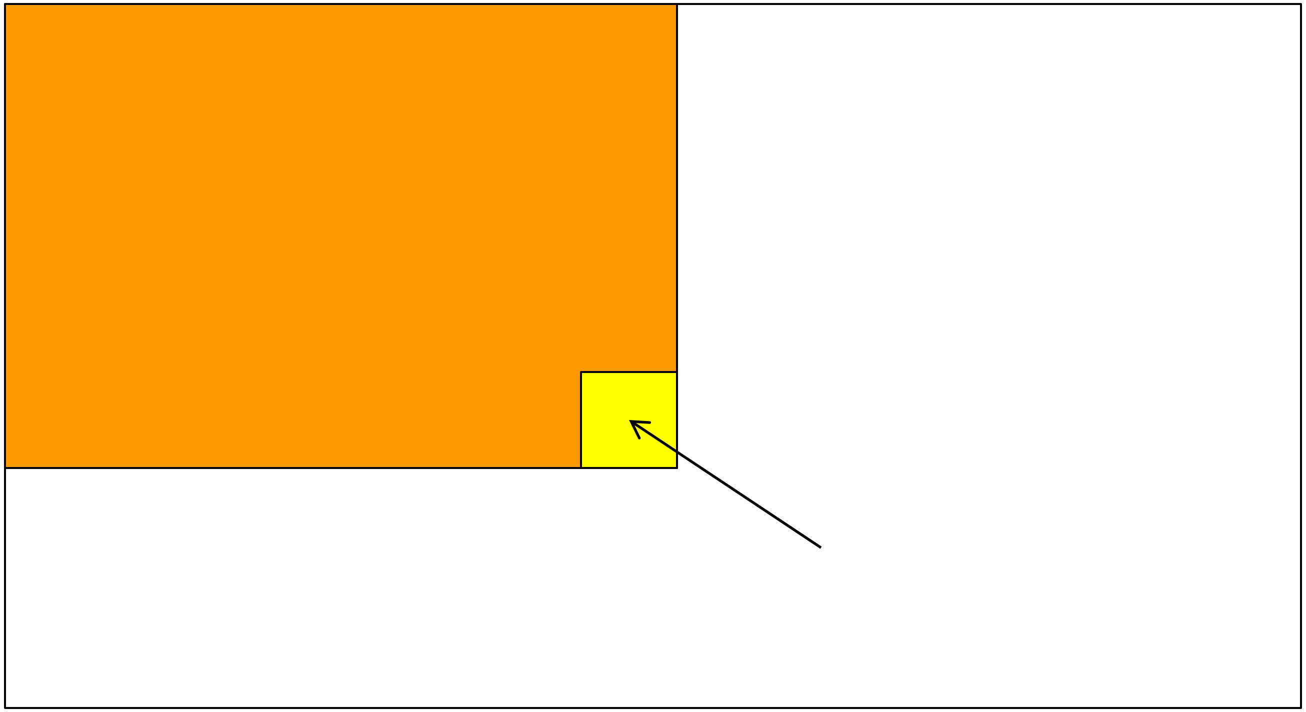
Viola-Jones: features

Viola-Jones: feature computation

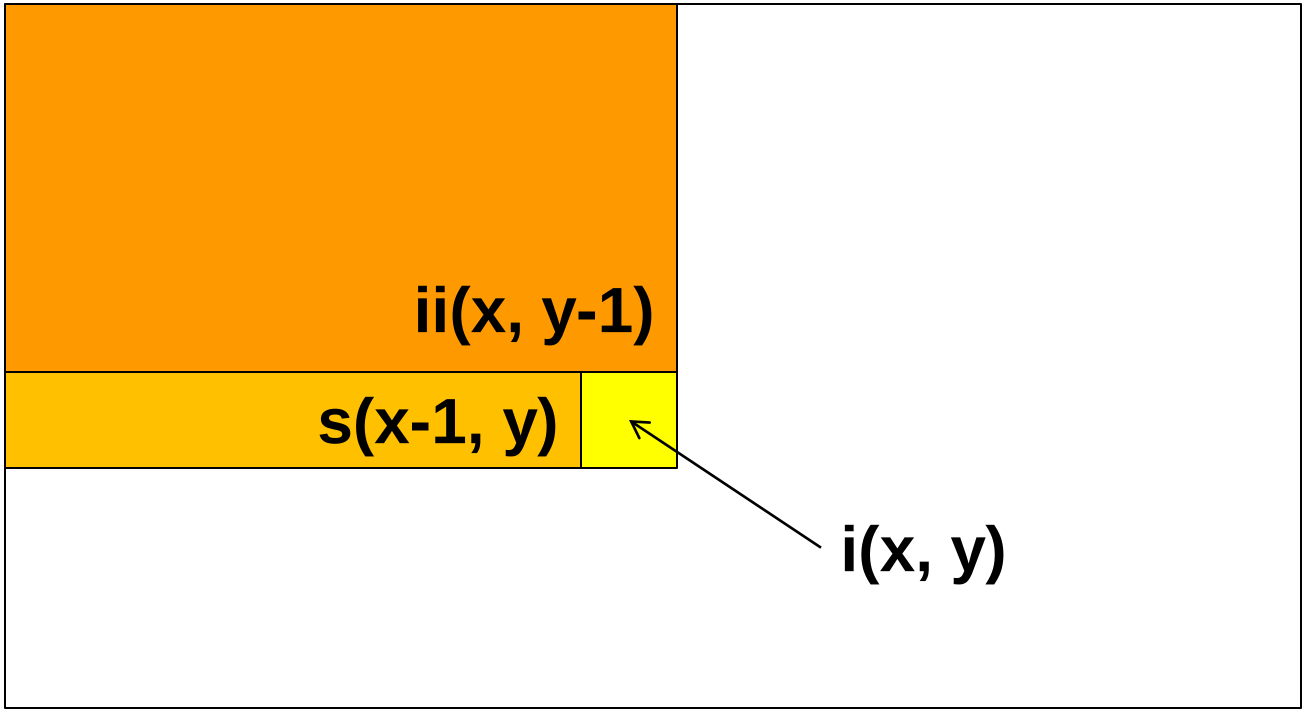
- The *integral image* computes a value at each pixel (*x*,*y*) that is the sum of the pixel values above and to the left of (*x*,*y*), inclusive
- This can quickly be computed in one pass through the image
- Allows evaluating rectangle features quickly at any scale



Viola-Jones: feature computation



Viola-Jones: feature computation



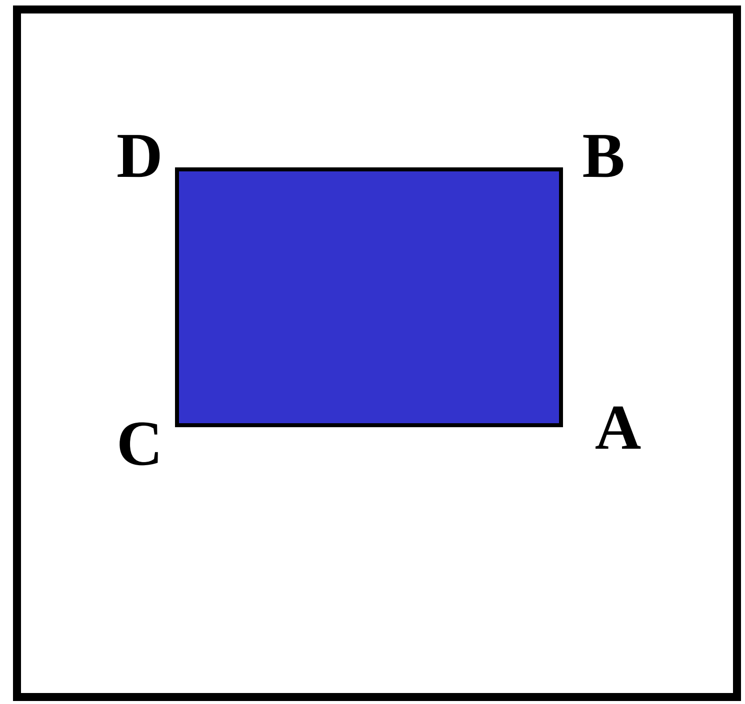
Cumulative row sum: $s(x, y) = s(x-1, y) + i(x, y)$

Integral image: $ii(x, y) = ii(x, y-1) + s(x, y)$

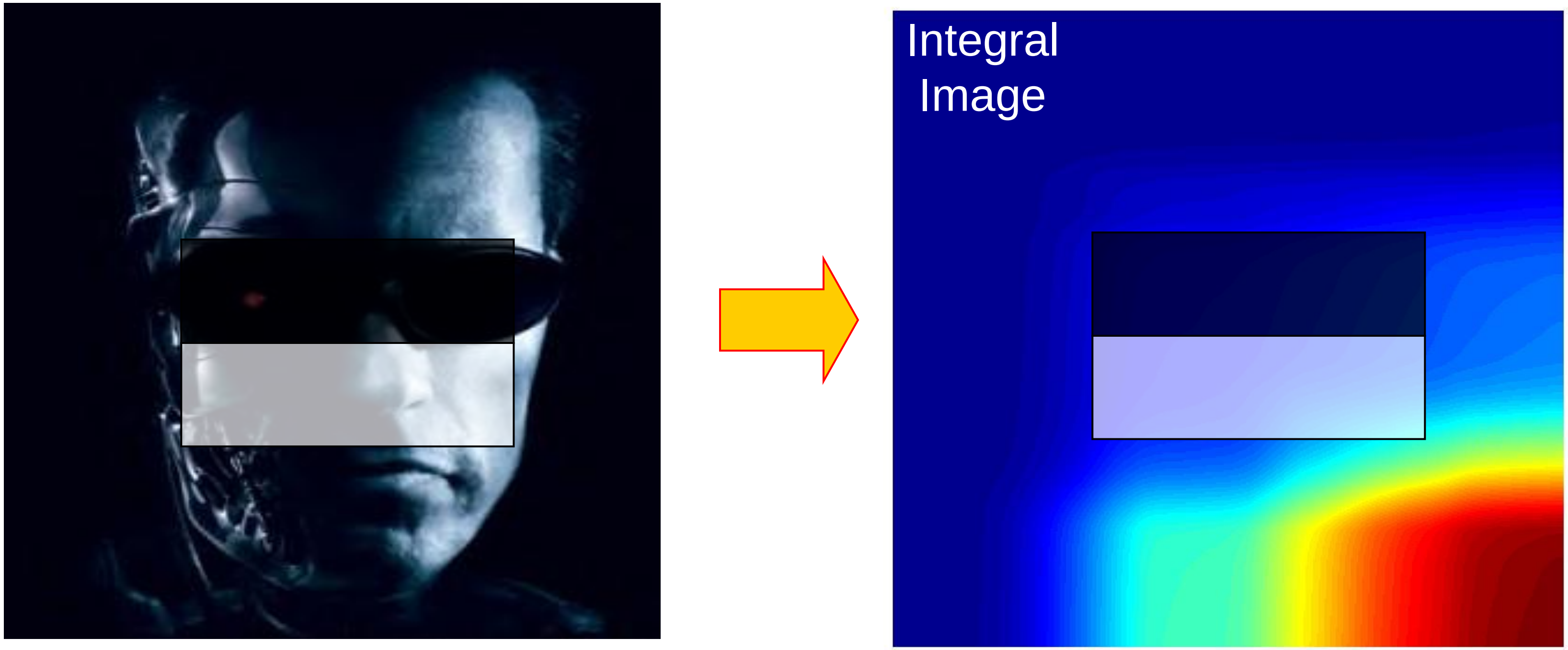
Viola-Jones: feature computation

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

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

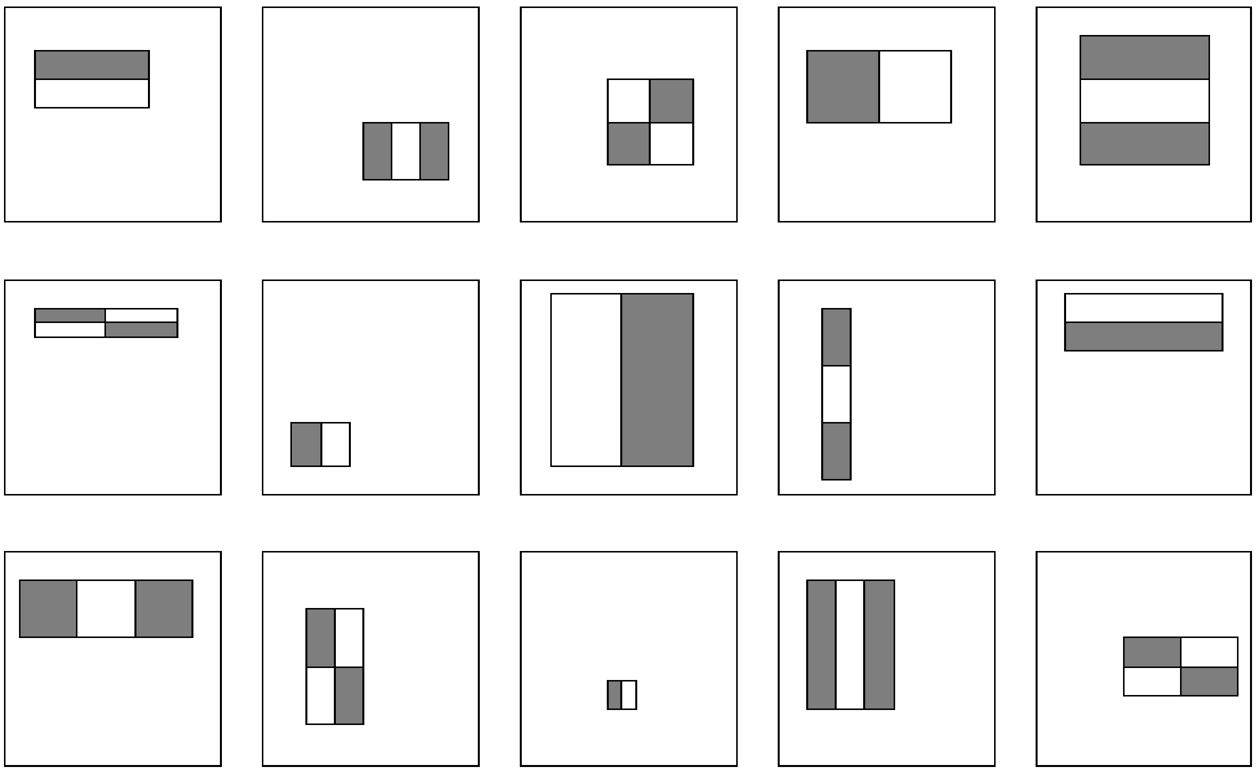


Viola-Jones: feature computation

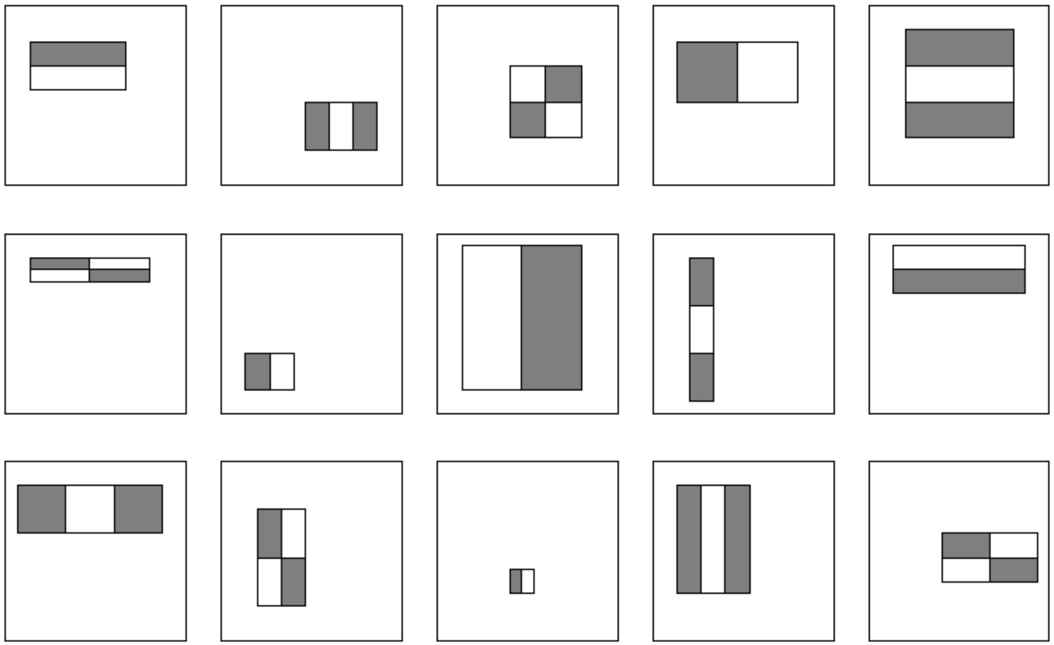


Viola-Jones: feature selection

- For a 24x24 detection region, the number of possible rectangle features is ~160,000!



Viola-Jones: feature selection



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?

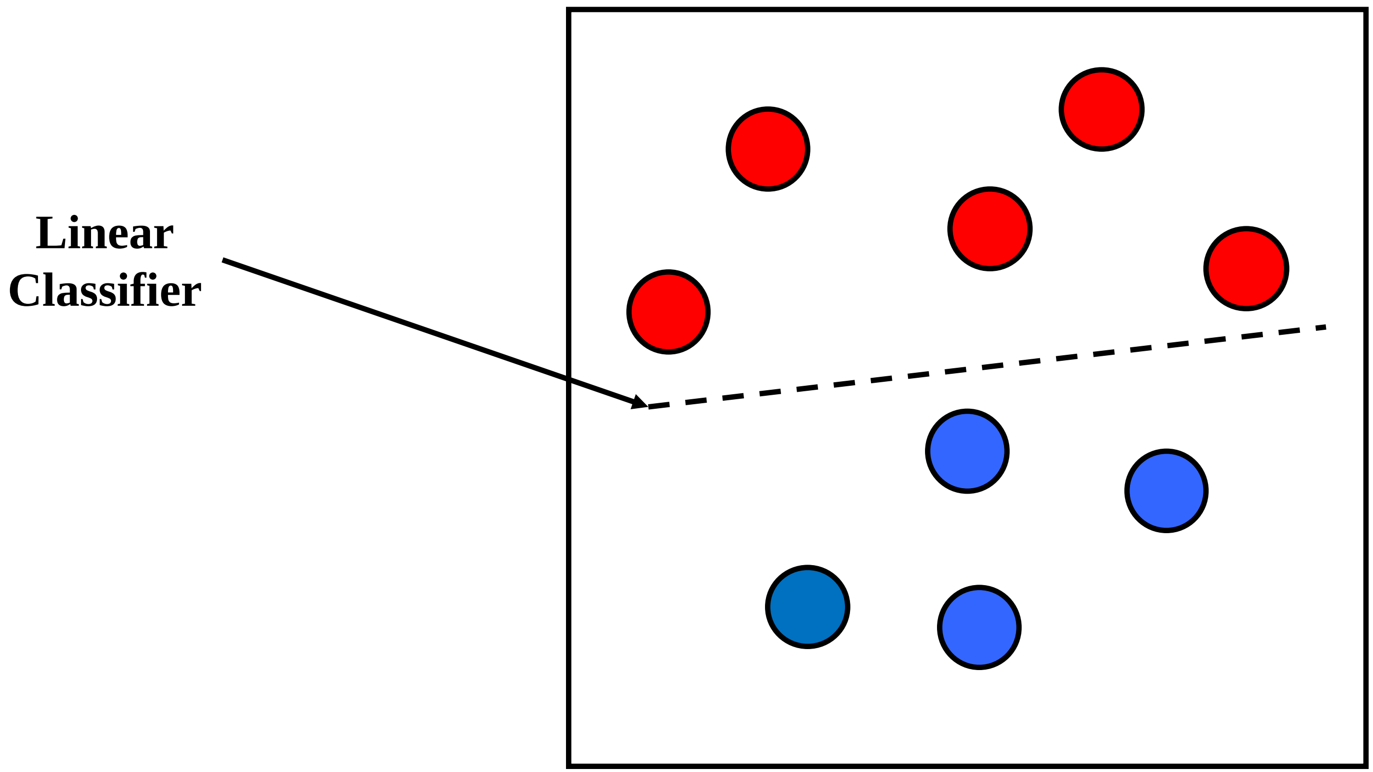
Kristen Grauman

Viola-Jones object detector

Main ideas:

- 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

Classifiers



Feature Space

Boosting

- Boosting is a classification scheme that works by combining *weak learners* into a more accurate ensemble classifier
  - A weak learner need only do better than chance
- Training consists of multiple *boosting rounds*
  - During each boosting round, we select a weak learner that does well on examples that were hard for the previous weak learners
  - “Hardness” is captured by weights attached to training examples

Y. Freund and R. Schapire, A short introduction to boosting, *Journal of Japanese Society for Artificial Intelligence*, 14(5):771-780, September, 1999.

Training procedure

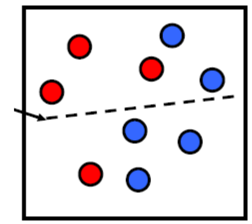
- Initially, weight each training example equally
- In each boosting round:
  - Find the weak learner that achieves the lowest *weighted* training error
  - Raise the weights of training examples misclassified by current weak learner
- Compute final classifier as linear combination of all weak learners (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)

Y. Freund and R. Schapire, A short introduction to boosting, *Journal of Japanese Society for Artificial Intelligence*, 14(5):771-780, September, 1999.

Training procedure

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights,
     $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$
     so that w_t is a probability distribution.
  2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i \, |h_j(x_i) - y_i|$.
  3. Choose the classifier, h_t, with the lowest error ϵ_t.
  4. Update the weights:
     $$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$
     where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:
  $$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$
  where $\alpha_t = \log \frac{1}{\beta_t}$

Start with uniform weights on training examples



$\{x_1, \ldots x_n\}$

For T rounds

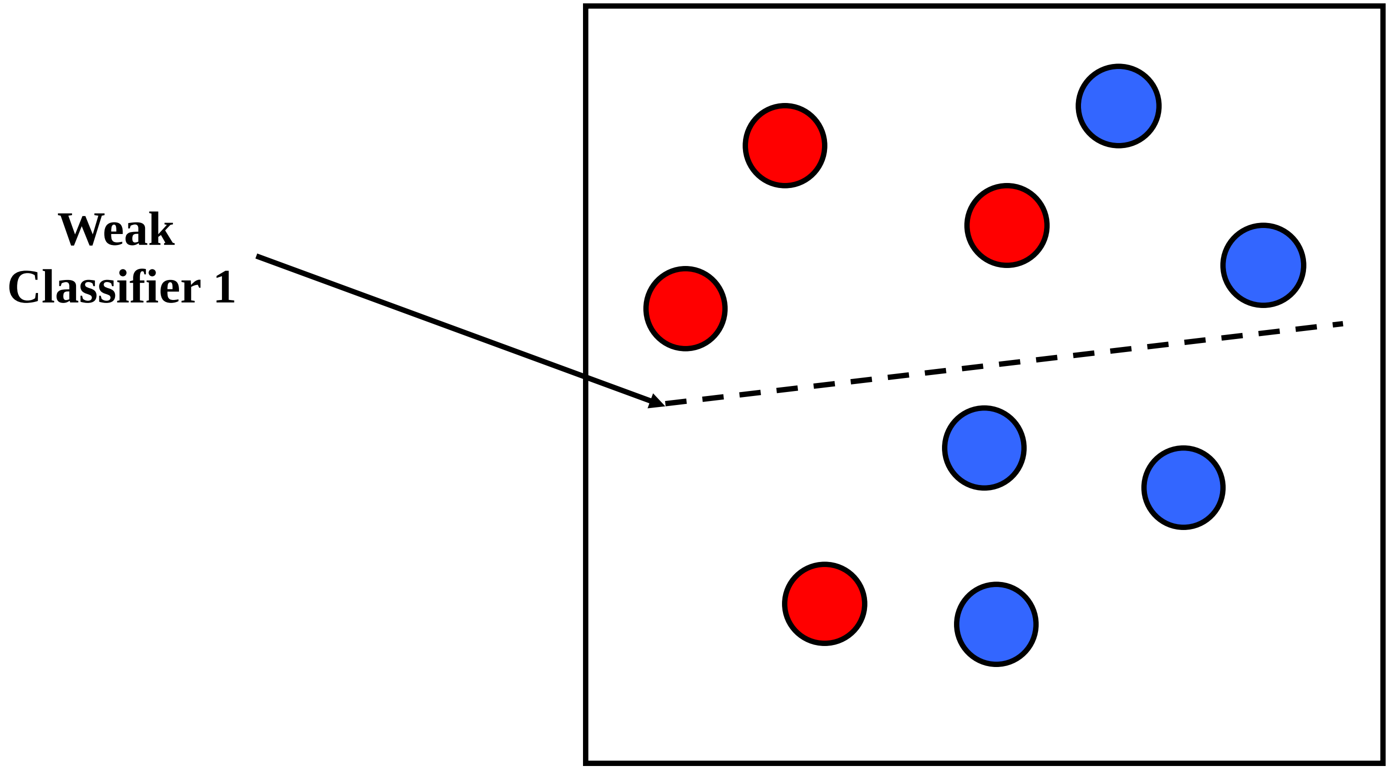
Evaluate *weighted* error for each feature, pick best.

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

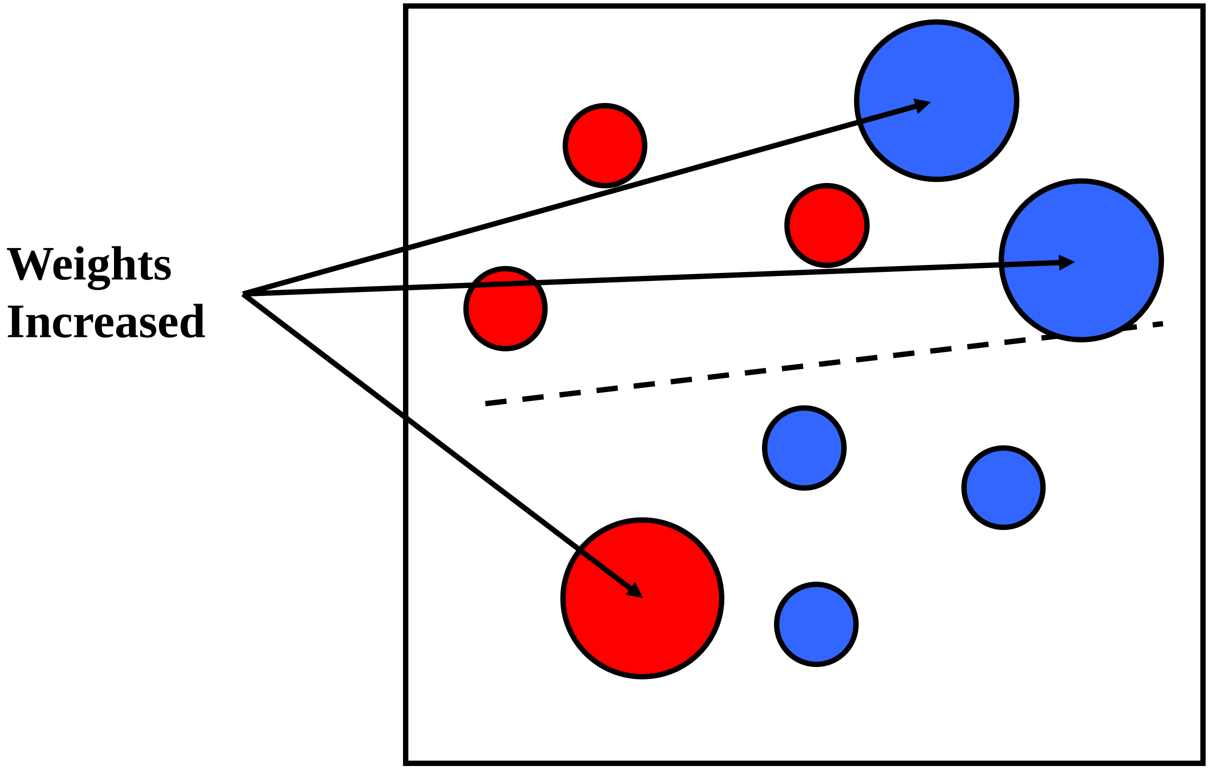
Final classifier is combination of the weak ones, weighted according to error they had.

Freund & Schapire 1995

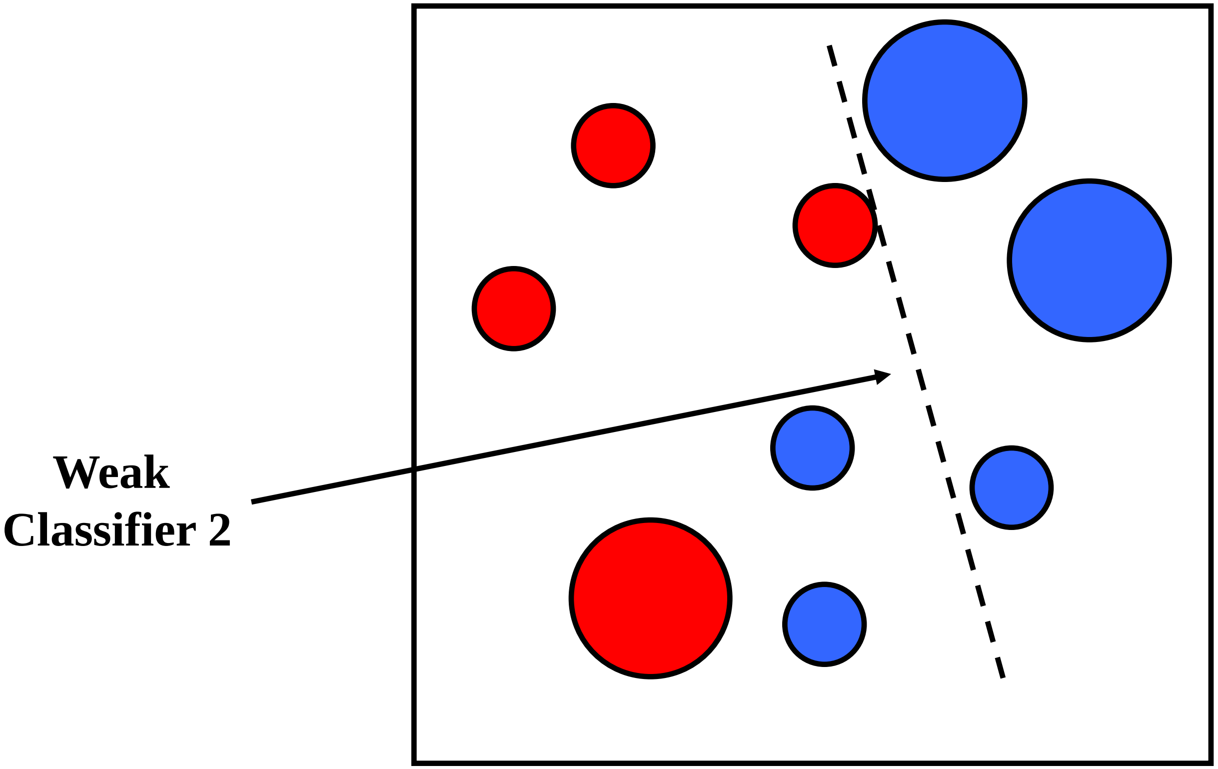
Boosting illustration



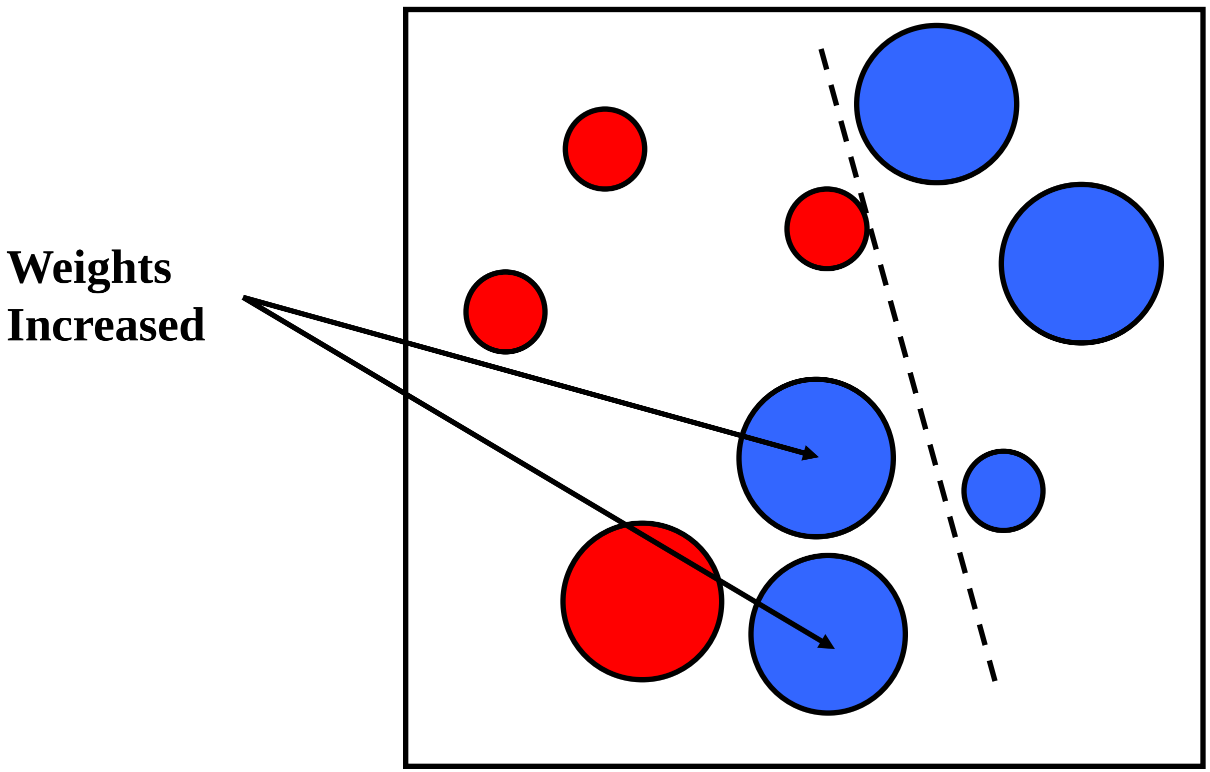
Boosting illustration



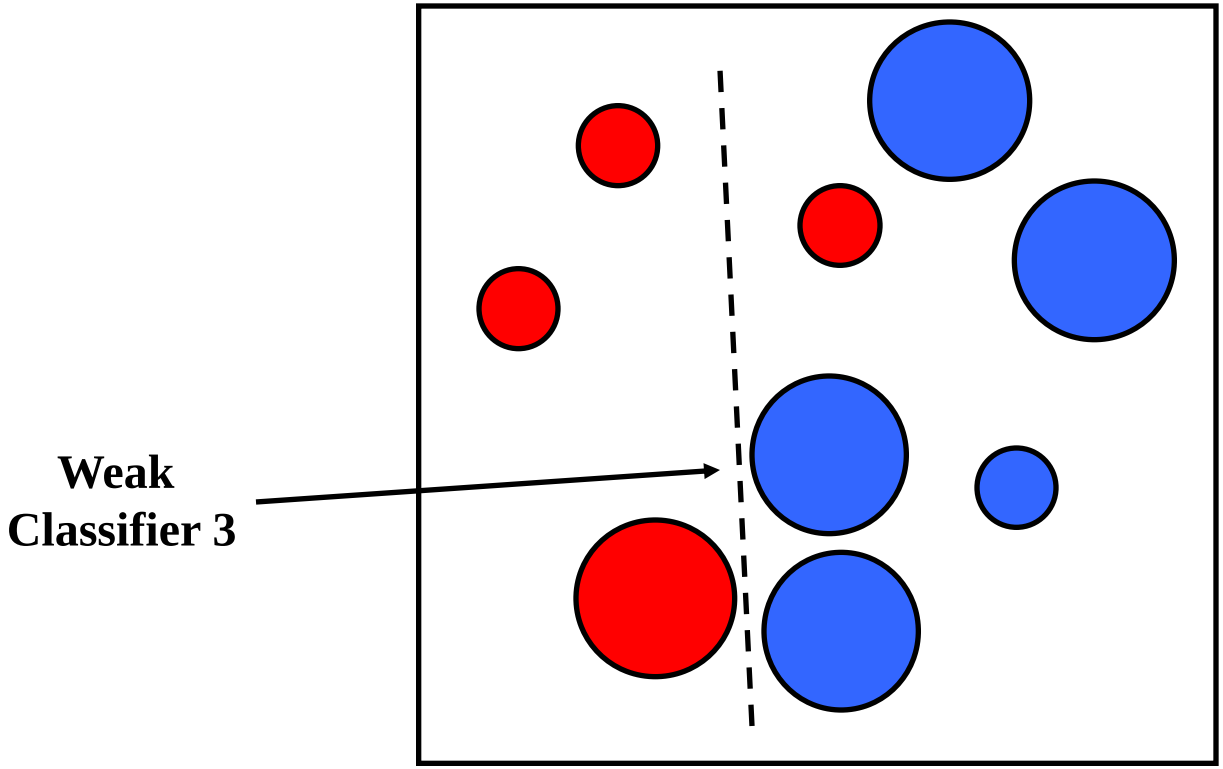
Boosting illustration



Boosting illustration

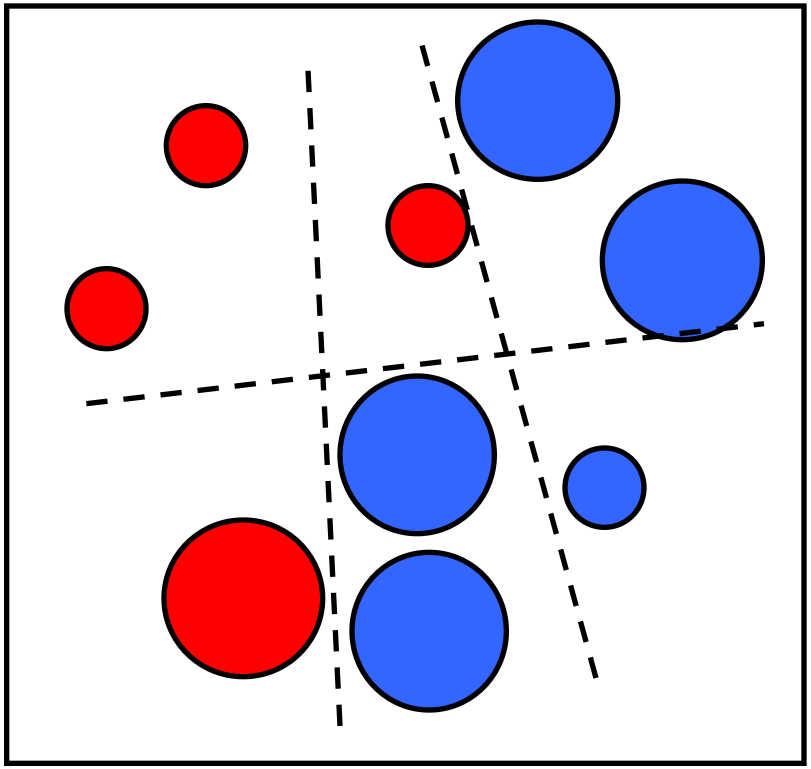


Boosting illustration



Boosting illustration

Final classifier is a combination of weak classifiers



Boosting for object detection

- Define weak learners based on rectangle features

value of rectangle feature

$$h_t(x) = \begin{cases} 1 & \text{if } p_t f_t(x) > p_t \theta_t \\ 0 & \text{otherwise} \end{cases}$$

window

parity

threshold

Boosting for object detection

- Define weak learners based on rectangle features
- For each round of boosting:
  - Evaluate each rectangle filter on each example
  - Select best threshold for each filter
  - Select best filter/threshold combination
  - Reweight examples
- Computational complexity of learning: $O(MNK)$
  - M rounds, N examples, K features

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

Slide credit: Lana Lazebnik

Problem …

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?

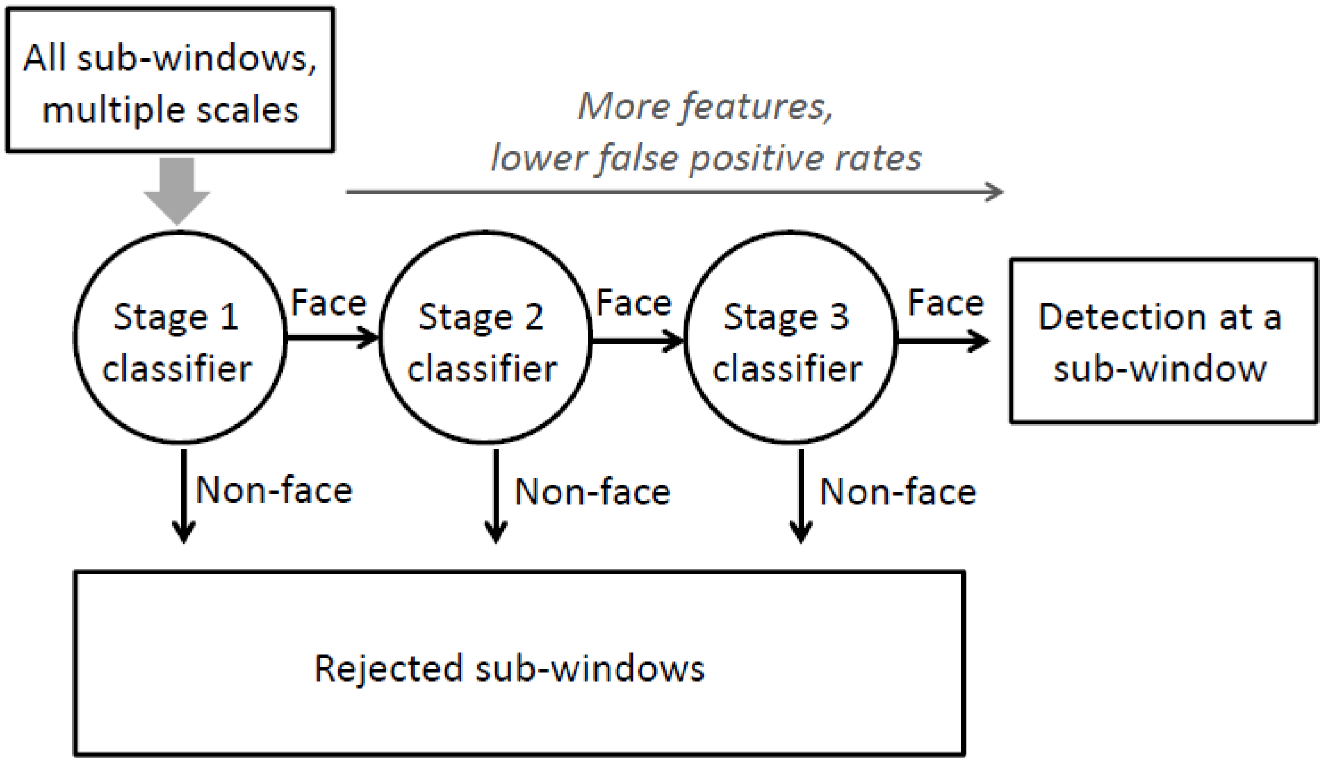
Viola-Jones object detector

Main ideas:

- 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 ⟵

Kristen Grauman

Cascading classifiers for detection

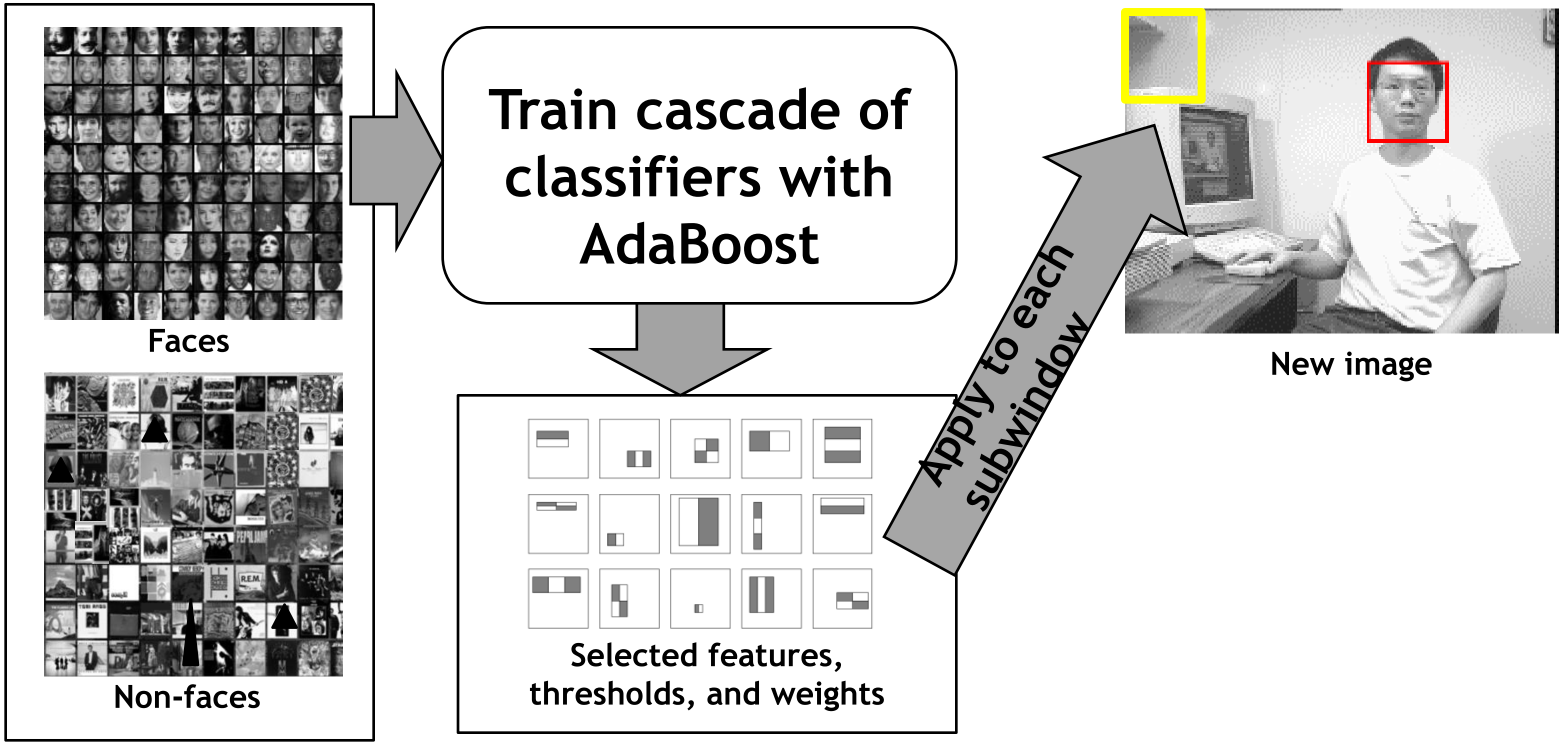


Form a *cascade* with low false negative rates early on

Apply less accurate but faster classifiers first to immediately discard windows that clearly appear to be negative

Kristen Grauman

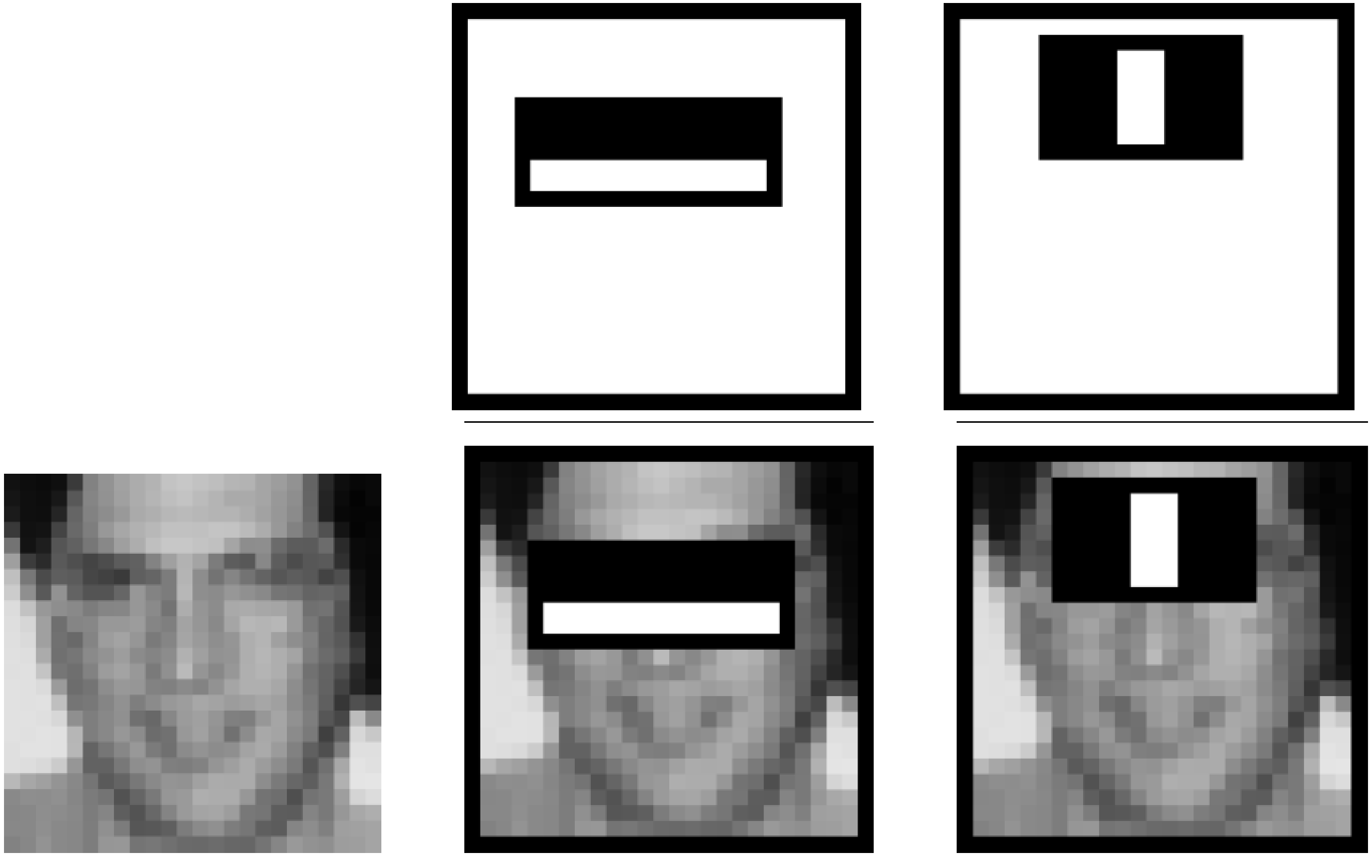
Viola-Jones: summary



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

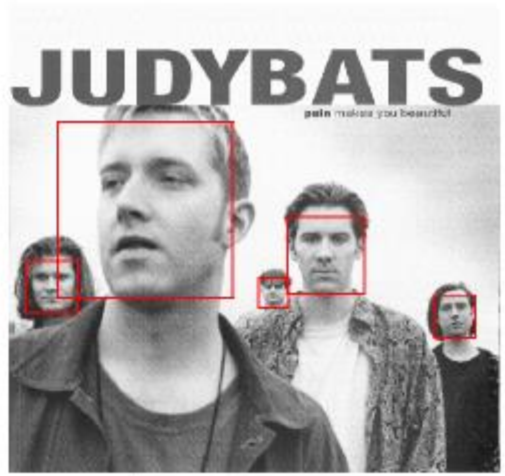
Kristen Grauman

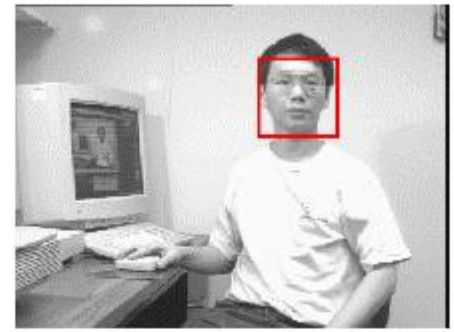
Viola-Jones: results

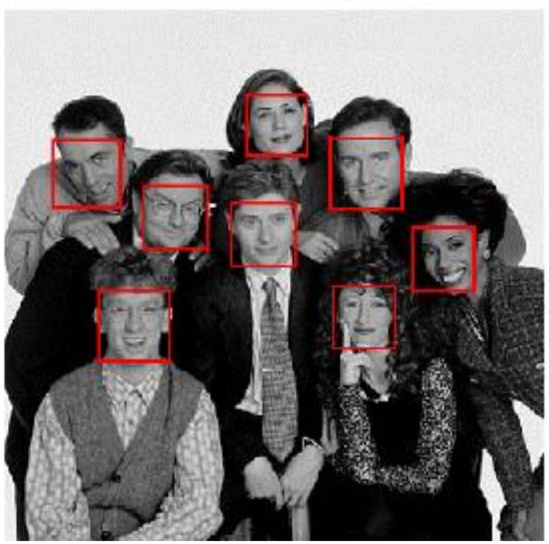


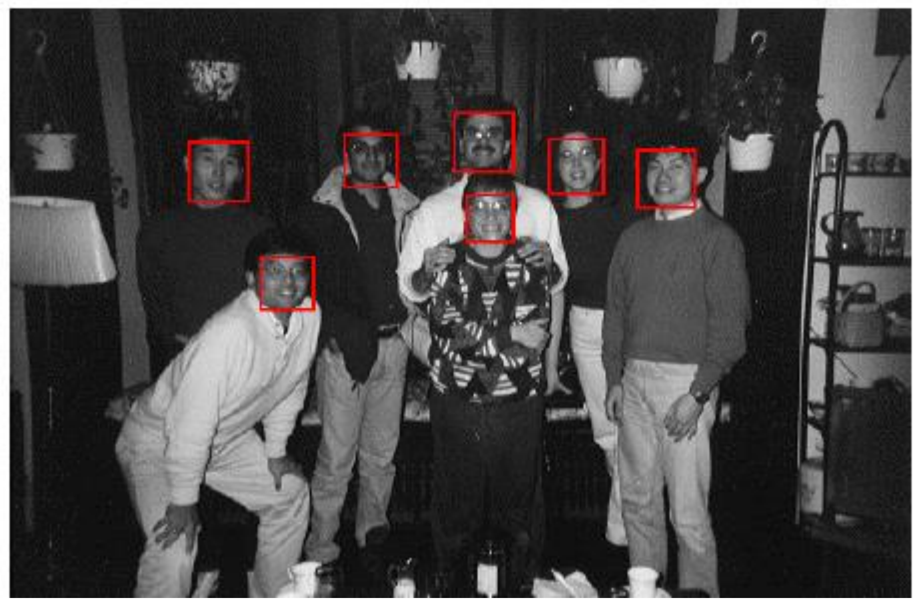
First two features selected

Viola-Jones: results

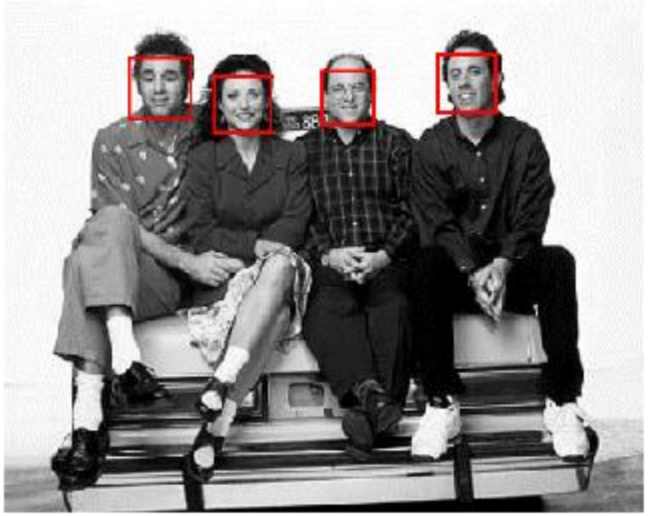


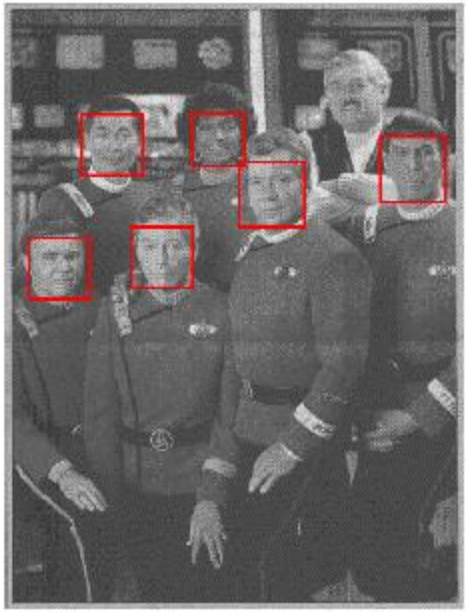


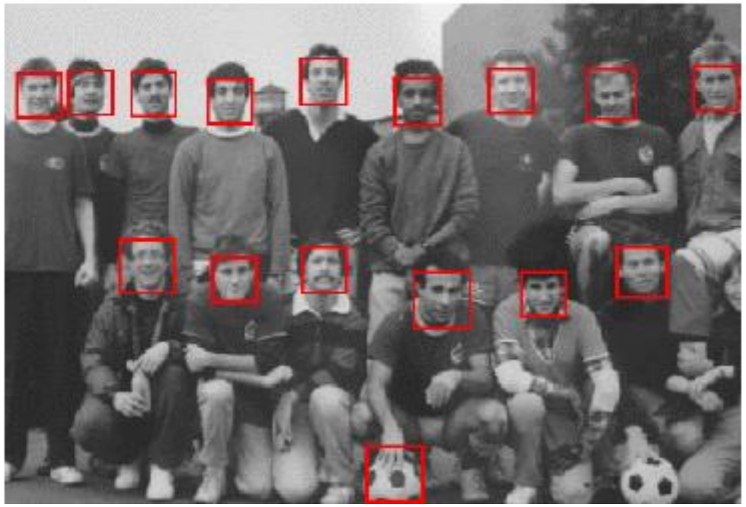


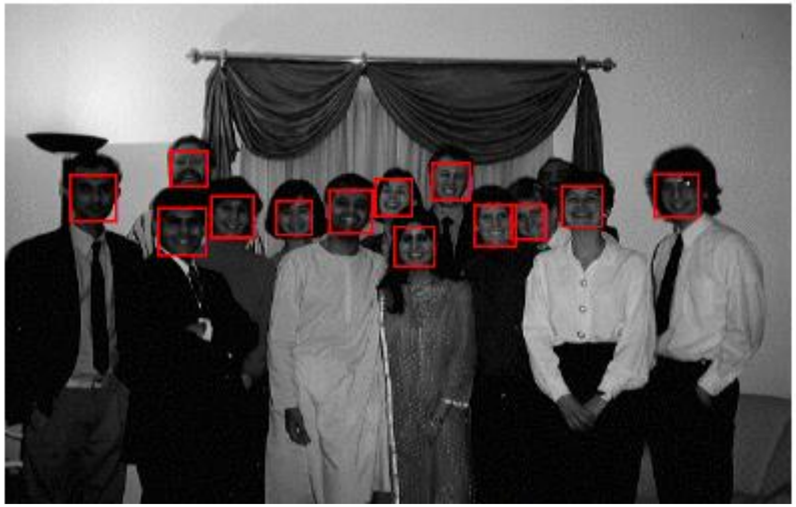


Viola-Jones: results





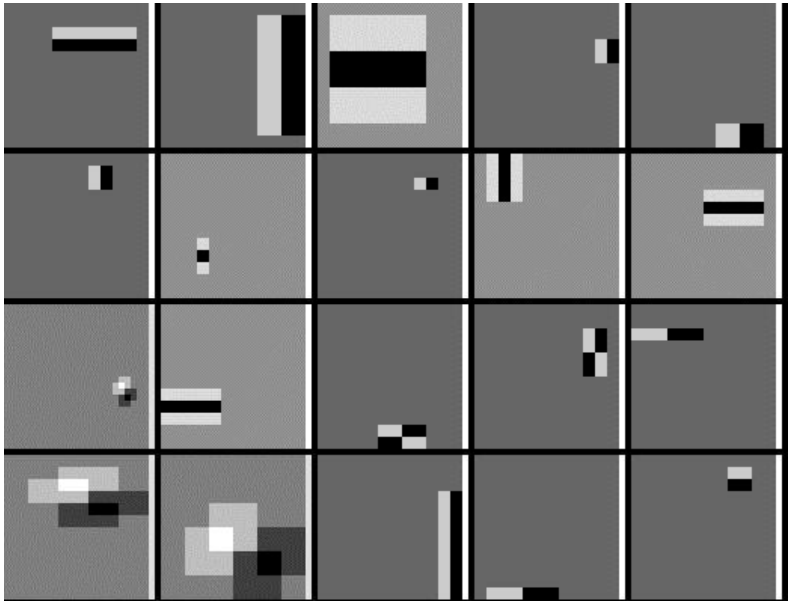




Viola-Jones: results

Viola-Jones: results

If train on profile faces:



Top features

Viola-Jones detector: results

If train on profile faces:

Application: blurring faces

ZDNet Asia
Where Technology Means Business

Home | News | Insight | Reviews | TechGuides | Jobs | Blogs | Videos | Community | Downloads | IT Library

Software | Hardware | Security | Communications | Business | Internet | Photos

Search ZDNet Asia

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/R
» Malaysia » Philippines » ASEAN
» Thailand » Indonesia » Asia Pacifi

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

Kristen Grauman

Consumer application: iPhoto 2009

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

Slide credit: Lana Lazebnik

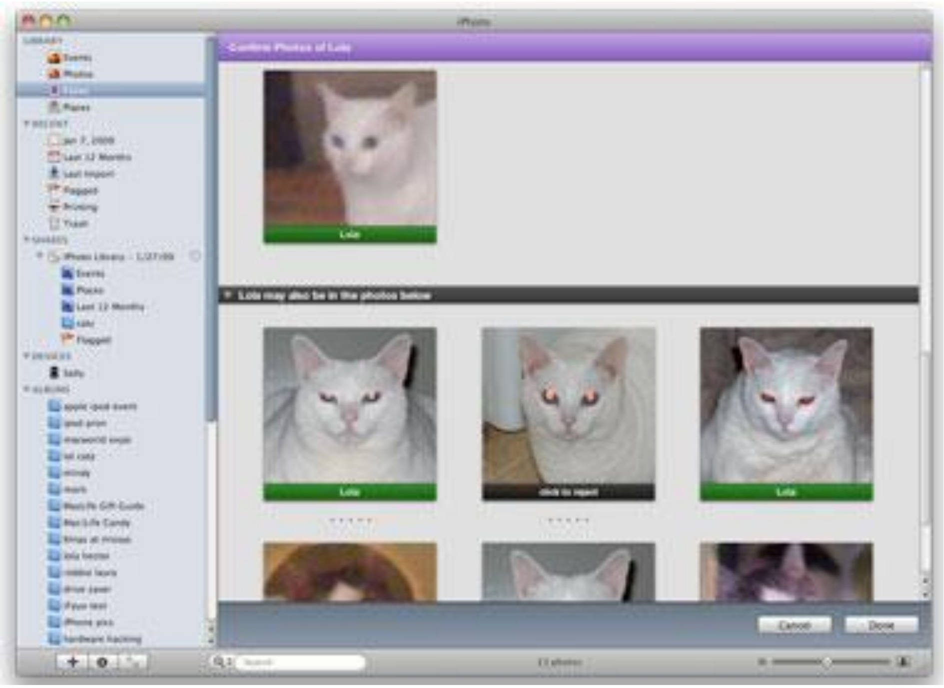
Consumer application: iPhoto 2009

[Things iPhoto thinks are faces]

Slide credit: Lana Lazebnik

Consumer application: iPhoto 2009

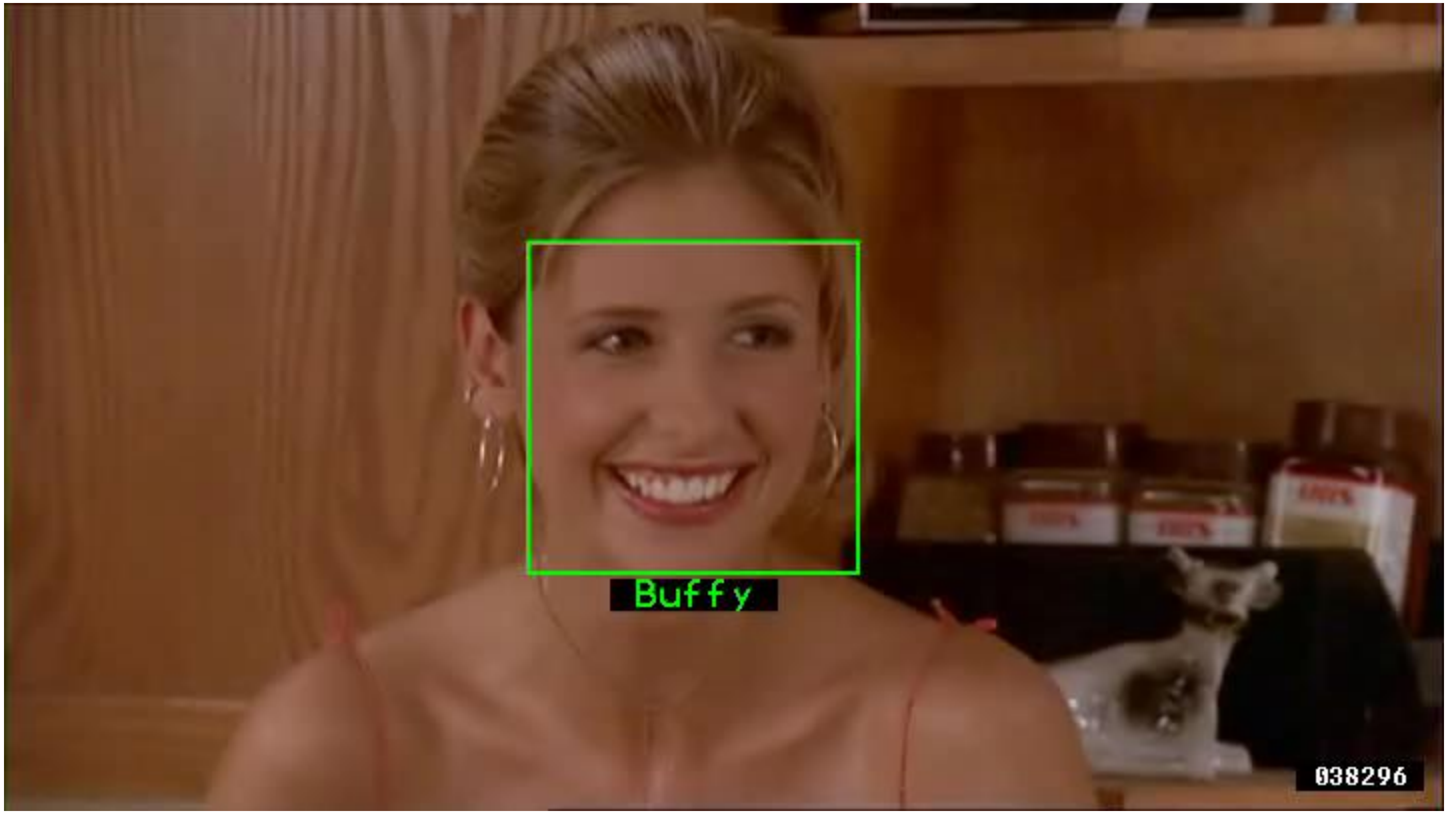
Can be trained to recognize pets!



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

Slide credit: Lana Lazebnik

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

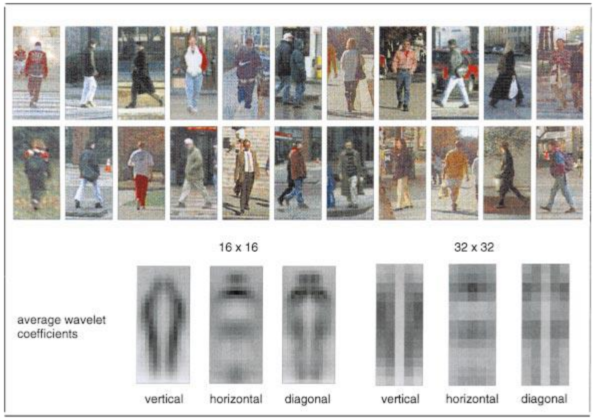
Sliding window detection

What other object categories are amenable to sliding window detection?

Kristen Grauman

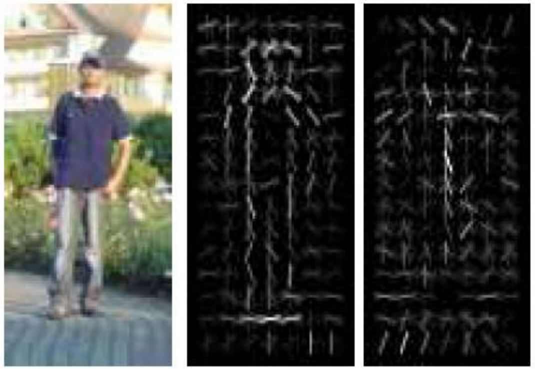
Pedestrian detection

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



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

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



SVM with HoGs [Dalal & Triggs, CVPR 2005]

Kristen Grauman

Window-based detection: strengths

Sliding window detection and global appearance descriptors:

- Simple detection protocol to implement
- Good feature choices critical
- Past successes for certain classes

Kristen Grauman

Window-based detection: Limitations

High computational complexity

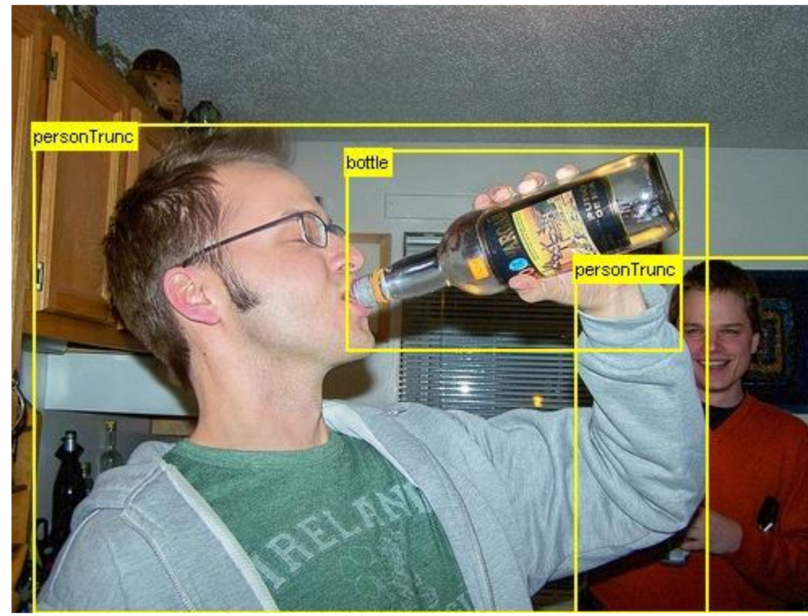
- For example: 250,000 locations x 30 orientations x 4 scales = 30,000,000 evaluations!
- If training binary detectors independently, means cost increases linearly with number of classes

With so many windows, false positive rate better be low

Kristen Grauman

Limitations (continued)

Not all objects are “box” shaped

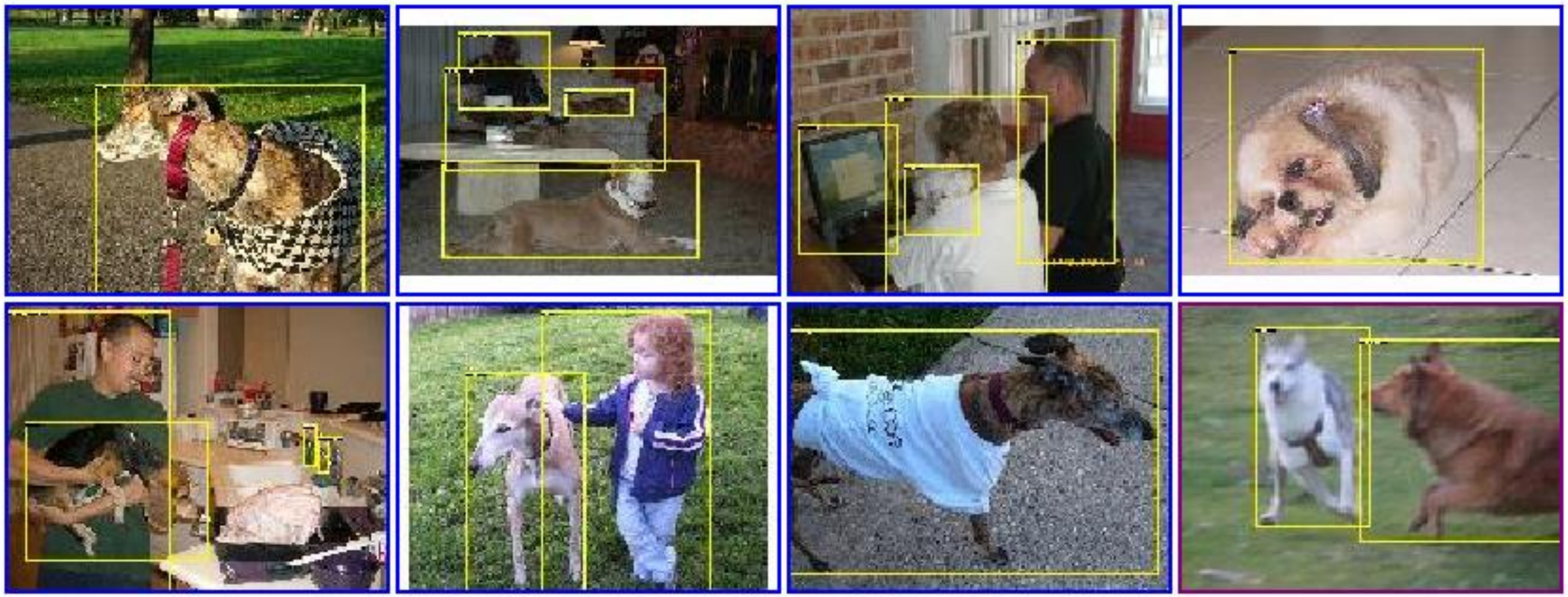


Kristen Grauman

Limitations (continued)

Non-rigid, deformable objects not captured well with representations assuming a fixed 2D structure; or must assume fixed viewpoint

Objects with less-regular textures not captured well with current image features

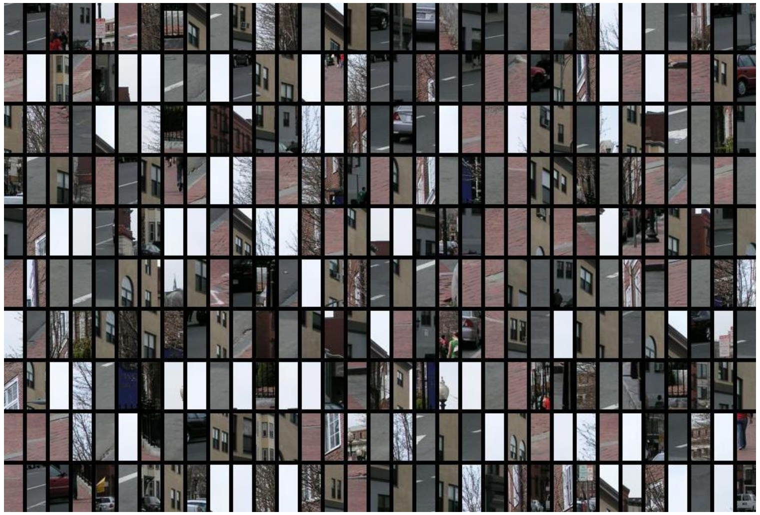


Kristen Grauman

Limitations (continued)

Does not take advantage of contextual cues

Sliding window



Detector's view

Figure credit: Derek Hoiem

Limitations (continued)

In practice, often requires large, training set to handle variations in occlusion, viewpoint, etc. (expensive)

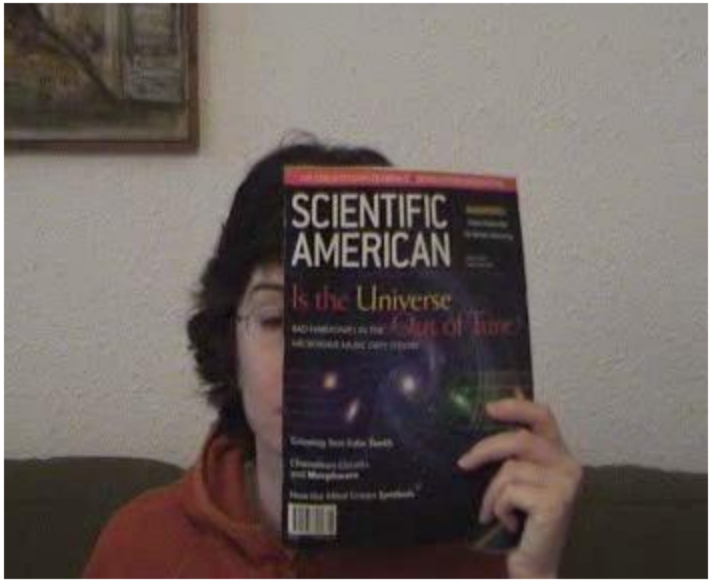


Image credit: Adam, Rivlin, & Shimshoni

Kristen Grauman

Summary

Basic pipeline for window-based detection

- Model/representation/classifier choice
- Sliding window and classifier scoring

Boosting classifiers: general idea

Viola-Jones face detector

- Exemplar of basic paradigm
- Plus key ideas: rectangular features, Adaboost for feature selection, cascade

Pros and cons of window-based detection