Visual Search and Recognition

Kristen Grauman
Dept. of Computer Sciences
University of Texas at Austin

First Bytes CS Teachers Workshop

July 9, 2008

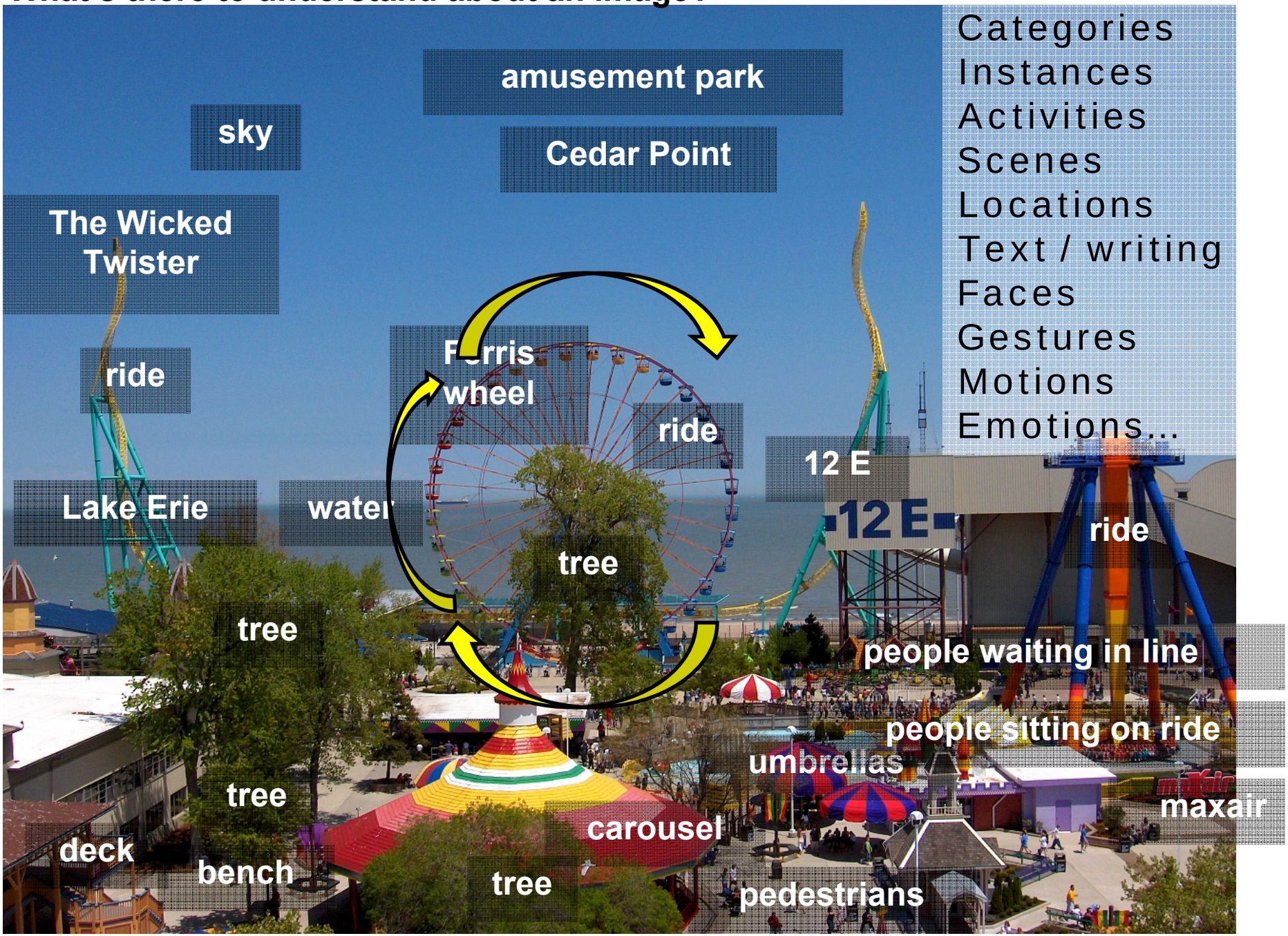
What is computer vision?

Does this computer have vision?

Computer vision

- Automatic understanding of images and video
- Computing properties of the 3D world from visual data
- Algorithms and representations to allow a machine to recognize objects, people, scenes, and activities.

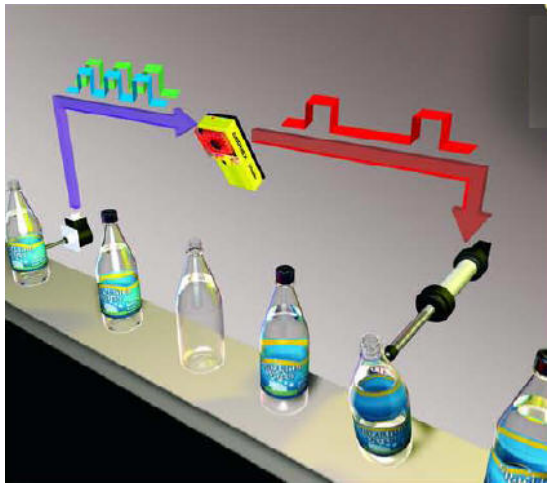
What's there to understand about an image?



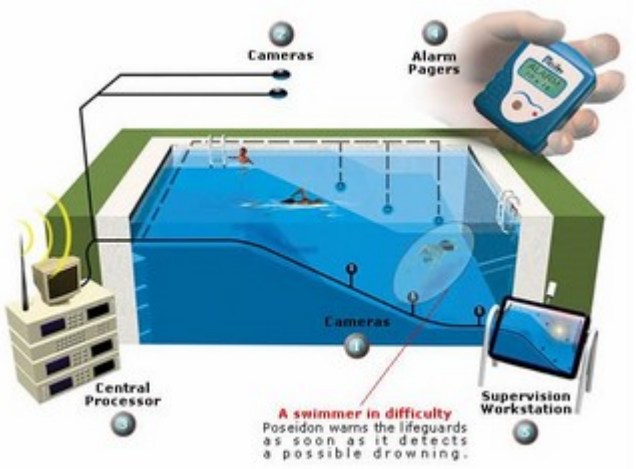
Why vision?

- As image sources multiply, so do applications
  - Relieve humans of boring, easy tasks
  - Enhance human abilities
  - Advance human-computer interaction, visualization
  - Perception for robotics / autonomous agents
- Computational models to test theories about human visual system; possible insights into human vision?

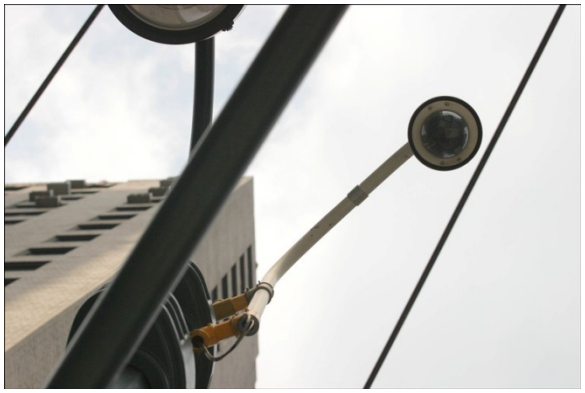
Some applications



Factory – inspection
(Cognex)



Monitoring for safety
(Poseidon)



Surveillance

Visualization
and tracking

License plate reading

Visualization

Some applications

Autonomous robots

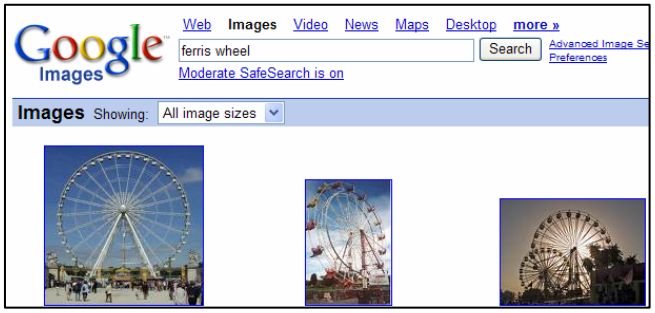
Navigation, driver safety

Assistive technology

Visual effects
(the Matrix)

Medical
imaging

Some applications



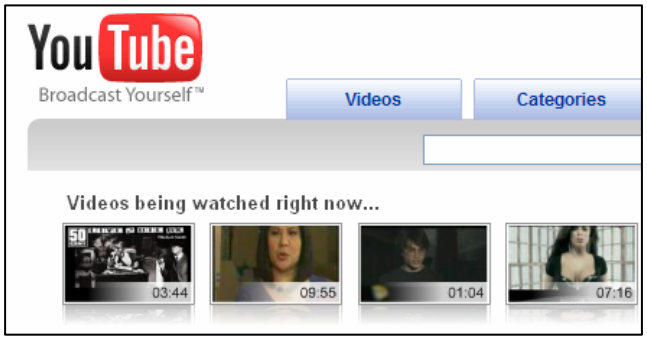
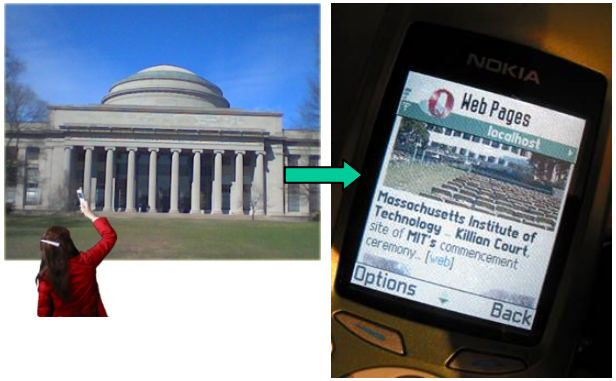


Image and video databases - CBIR

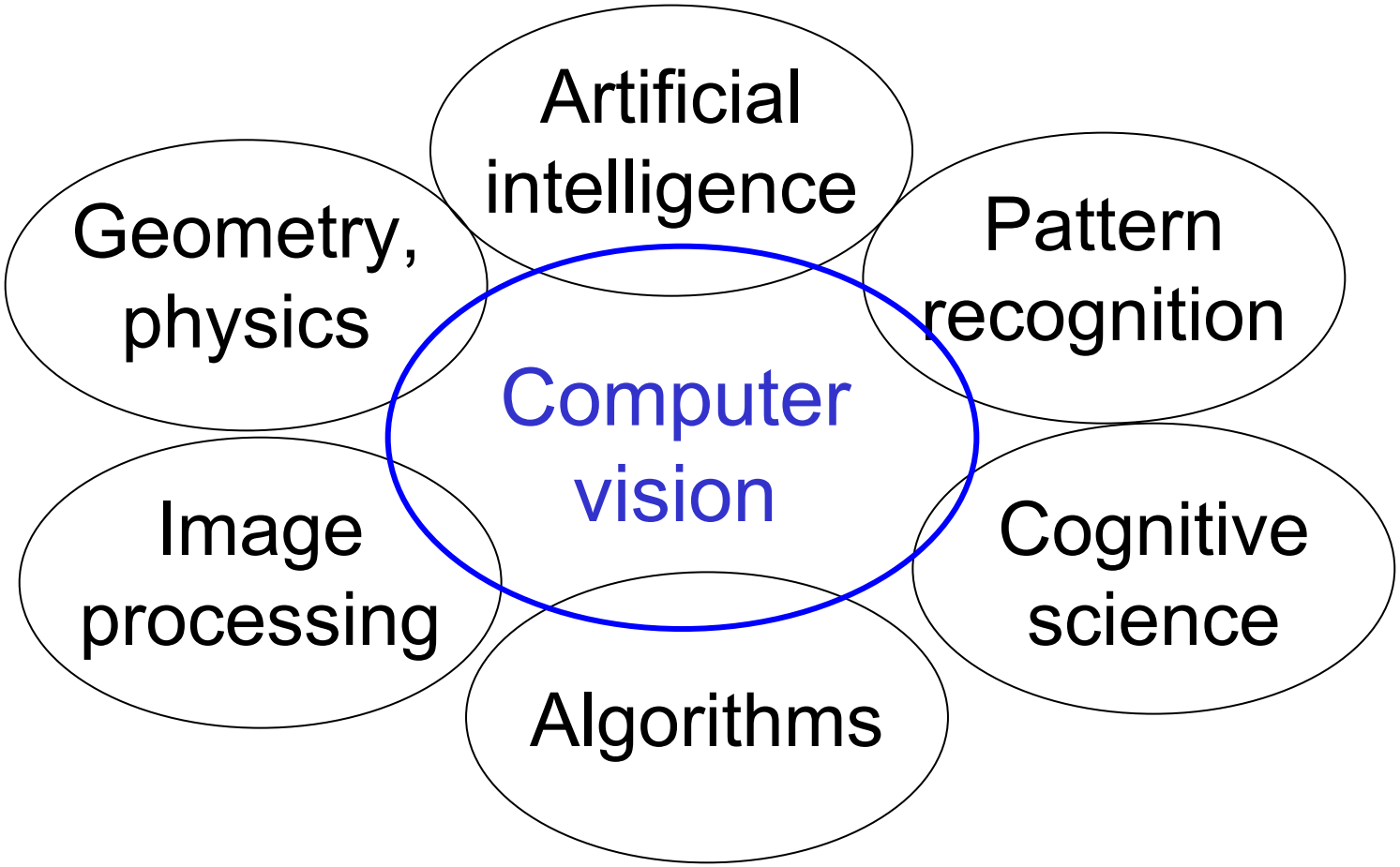
Multi-modal interfaces



Situated search

Tracking, activity recognition

Related disciplines



Why is vision difficult?

- For starters, it's an ill-posed problem: real world much more complex than what we can measure in images
  - 3D → 2D
- Impossible to literally "invert" image formation process

Challenges: context and human experience

Context cues

Challenges: context and human experience

Context cues






Function

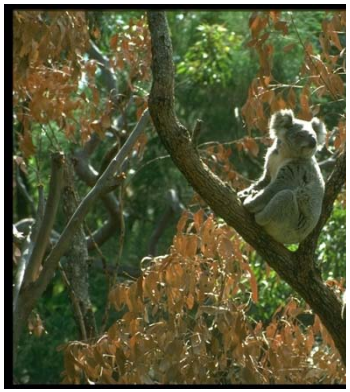
Dynamics

Video credit: J. Davis

Challenges: robustness

Illumination

Object pose



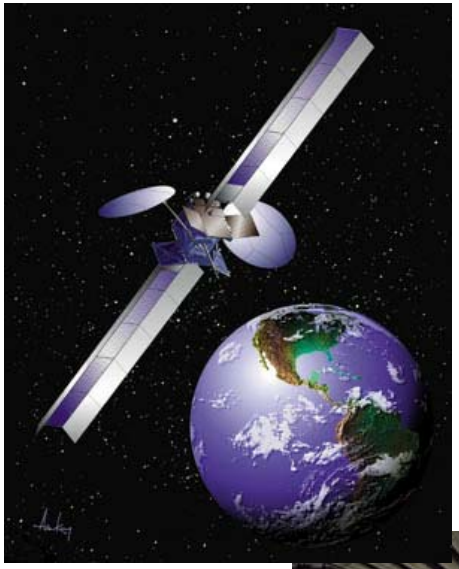
Clutter

Occlusions

Intra-class appearance

Viewpoint











Challenges: scale, efficiency

- Thousands to millions of pixels in an image
- 3,000-30,000 human recognizable object categories
- 30+ degrees of freedom in the pose of articulated objects (humans)
- Estimated 30 Gigapixels of image/video content generated per second
- About half of the cerebral cortex in primates is devoted to processing visual information [Felleman and van Essen 1991]
- Billions of images indexed by Google Image Search
- 18 billion+ prints produced from digital camera images in 2004
- 295.5 million camera phones sold in 2005

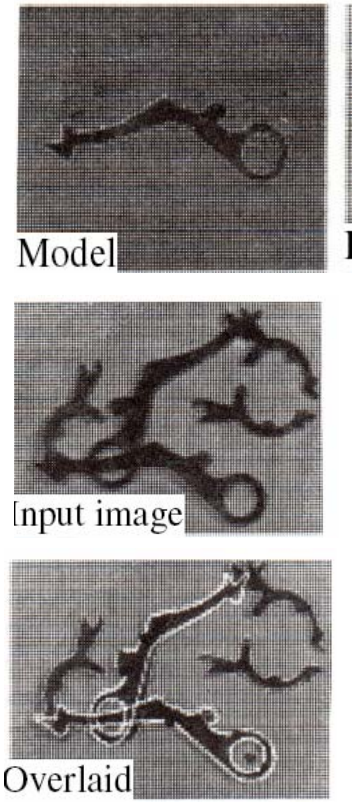
Challenges: learning with minimal or weak supervision

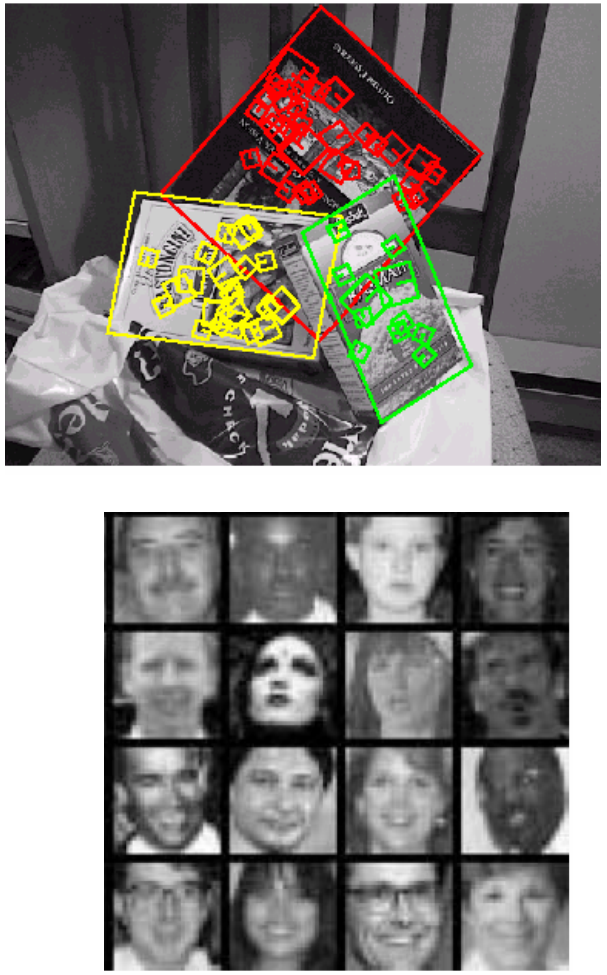
- Providing carefully labeled data is expensive, can be biased anyway
- Human visual system suggests exorbitant supervision not realistic
- Linked to the scale problem

…So what are some things that work well today?

- Frontal face detection
- Finding textured flat objects (from collections of manageable scale)
- Barcode readers
- Fingerprint recognition/matching
- Various medical vision applications: e.g. visualization for surgery, aid in detecting tumors
- Multi-view 3d reconstruction leveraged by various special effects
- …
- In general, most robustness for systems that can exploit constraints or domain knowledge.

Evolution of recognition focus



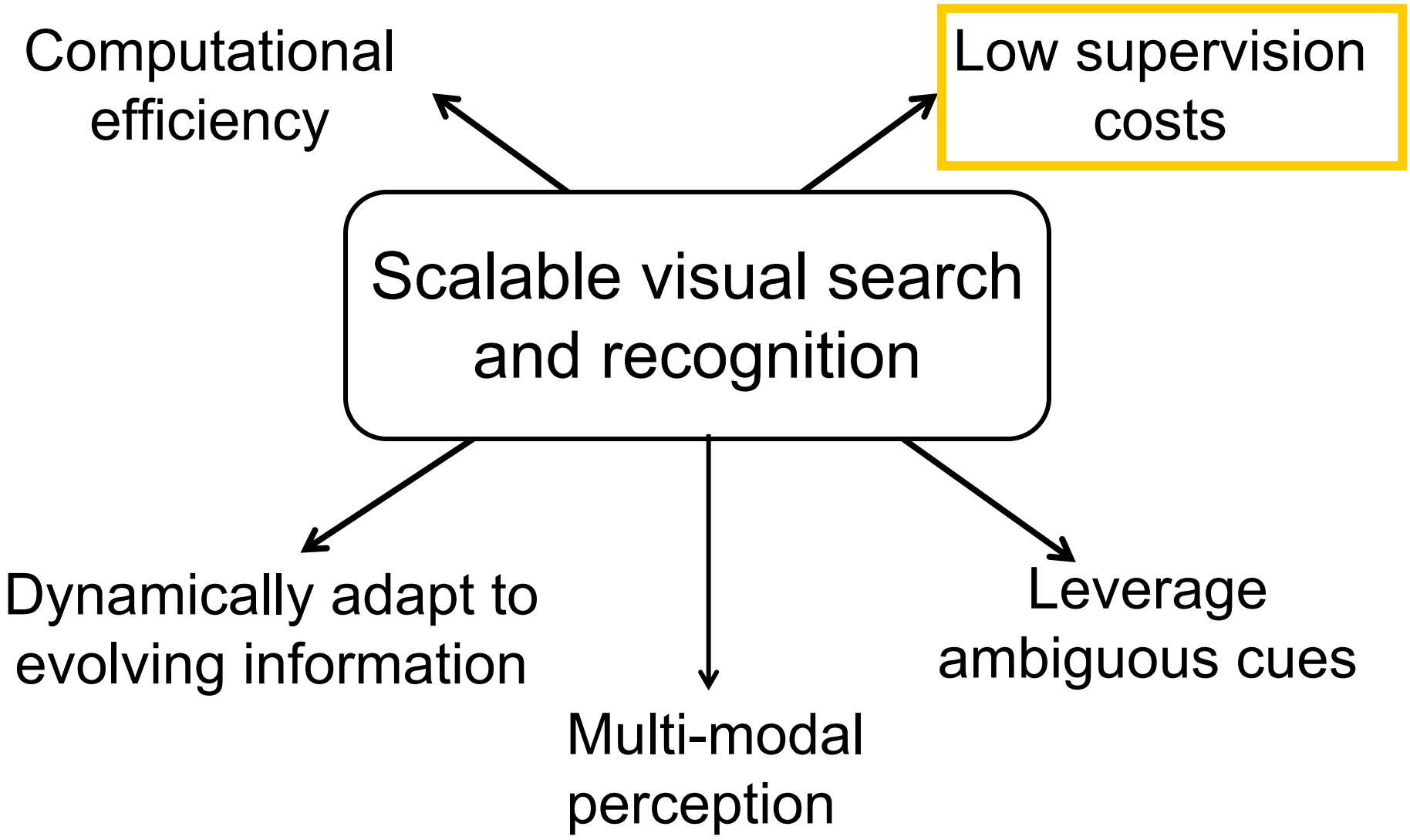


1980s

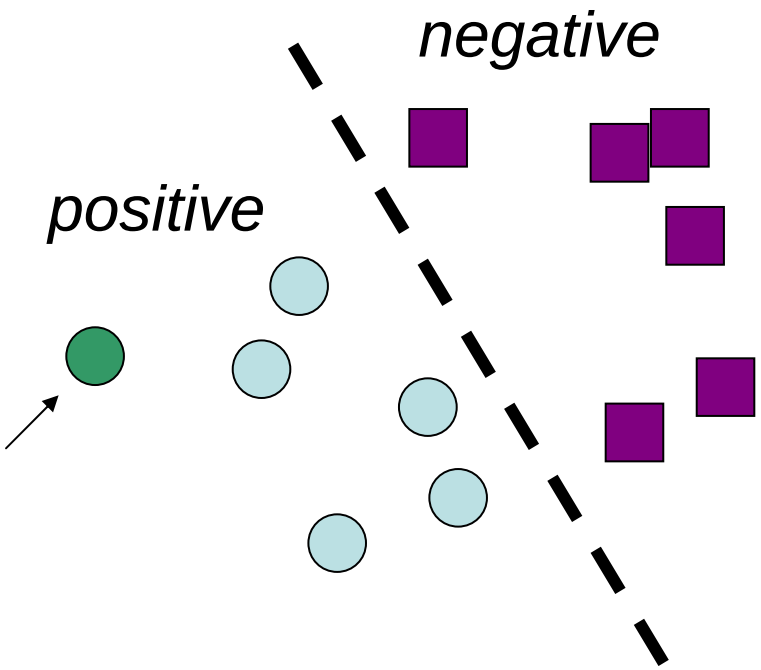
1990s to early 2000s

Currently

Our research tracks



Supervised learning



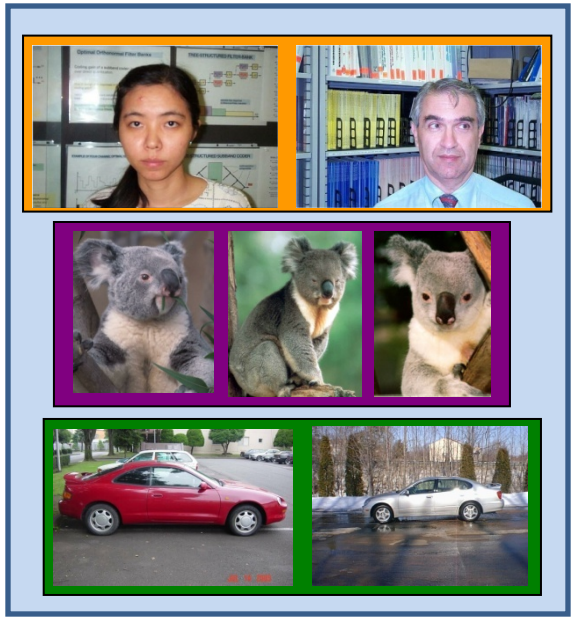
Want to predict the label of new unseen “test” example

We are given labeled “training” examples

Spectrum of supervision: learning from images

Less — **More**

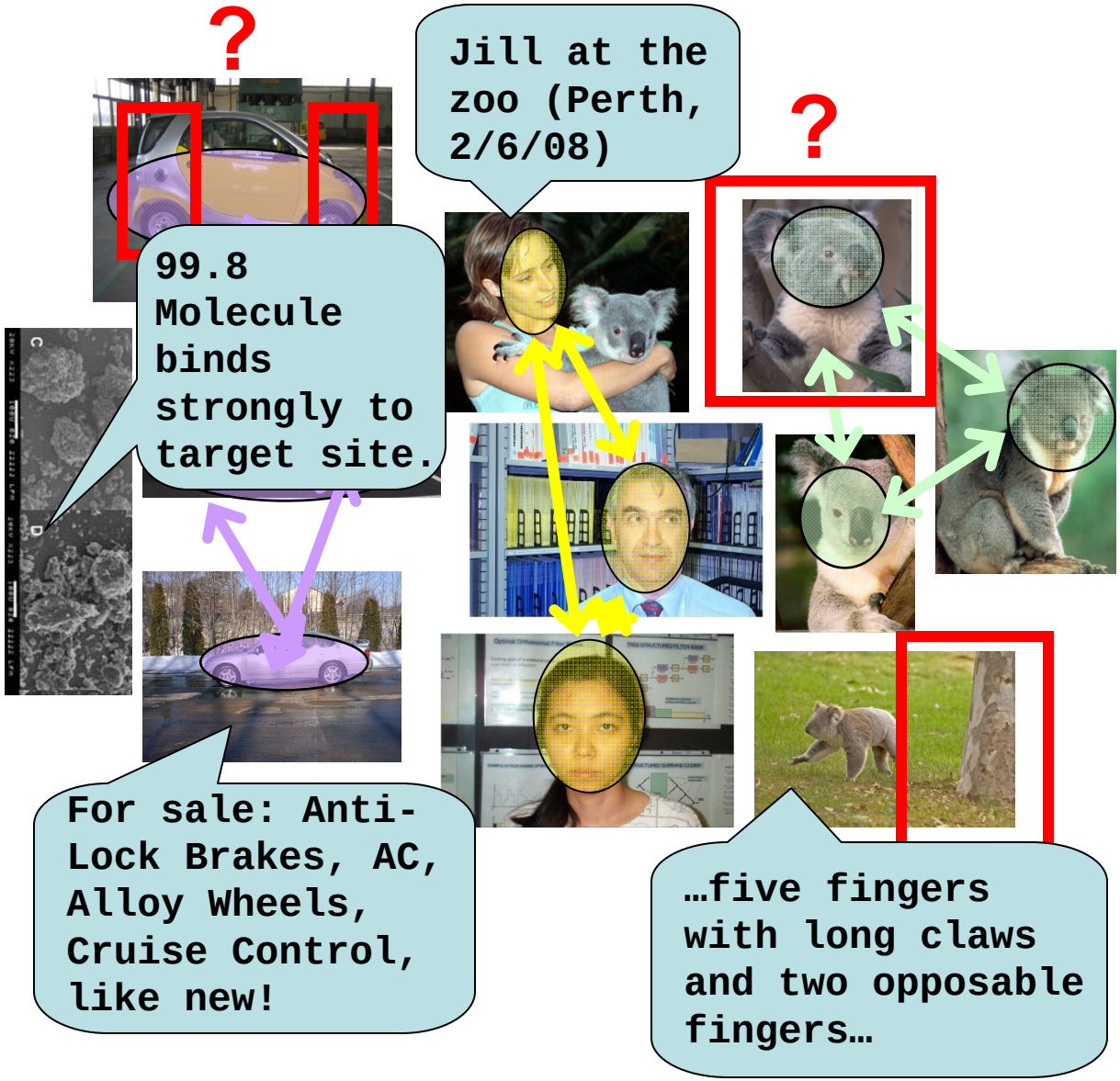
Unlabeled, multiple objects



Classes labeled, some clutter

Cropped to object, parts and classes labeled

Unraveling unlabeled image data



Discovering visual patterns

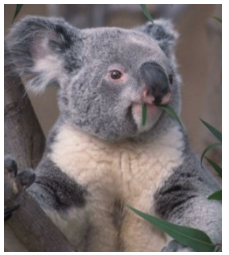
Actively learning

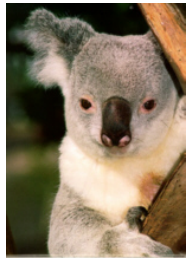
Guiding questions to the right expertise

Leveraging “loose” annotations

Unsupervised category discovery



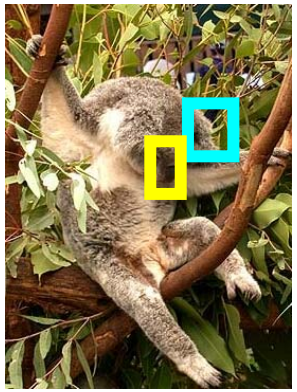



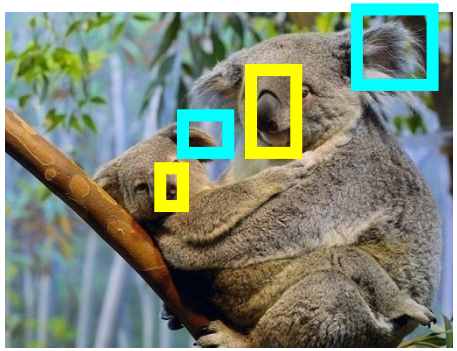





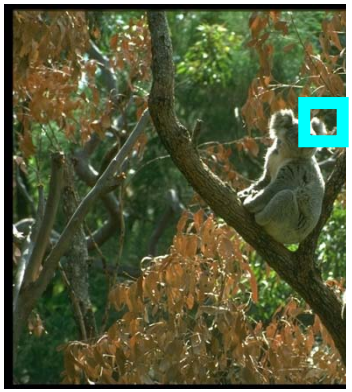
Local image features

Illumination



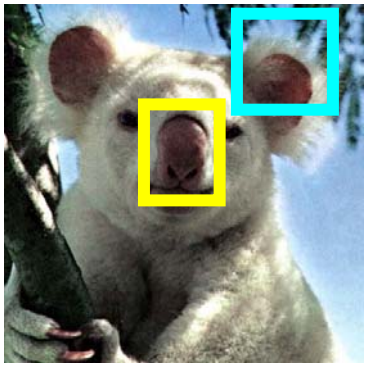


Object pose

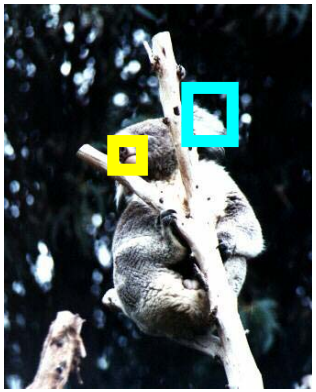


Clutter

Occlusions



Intra-class appearance

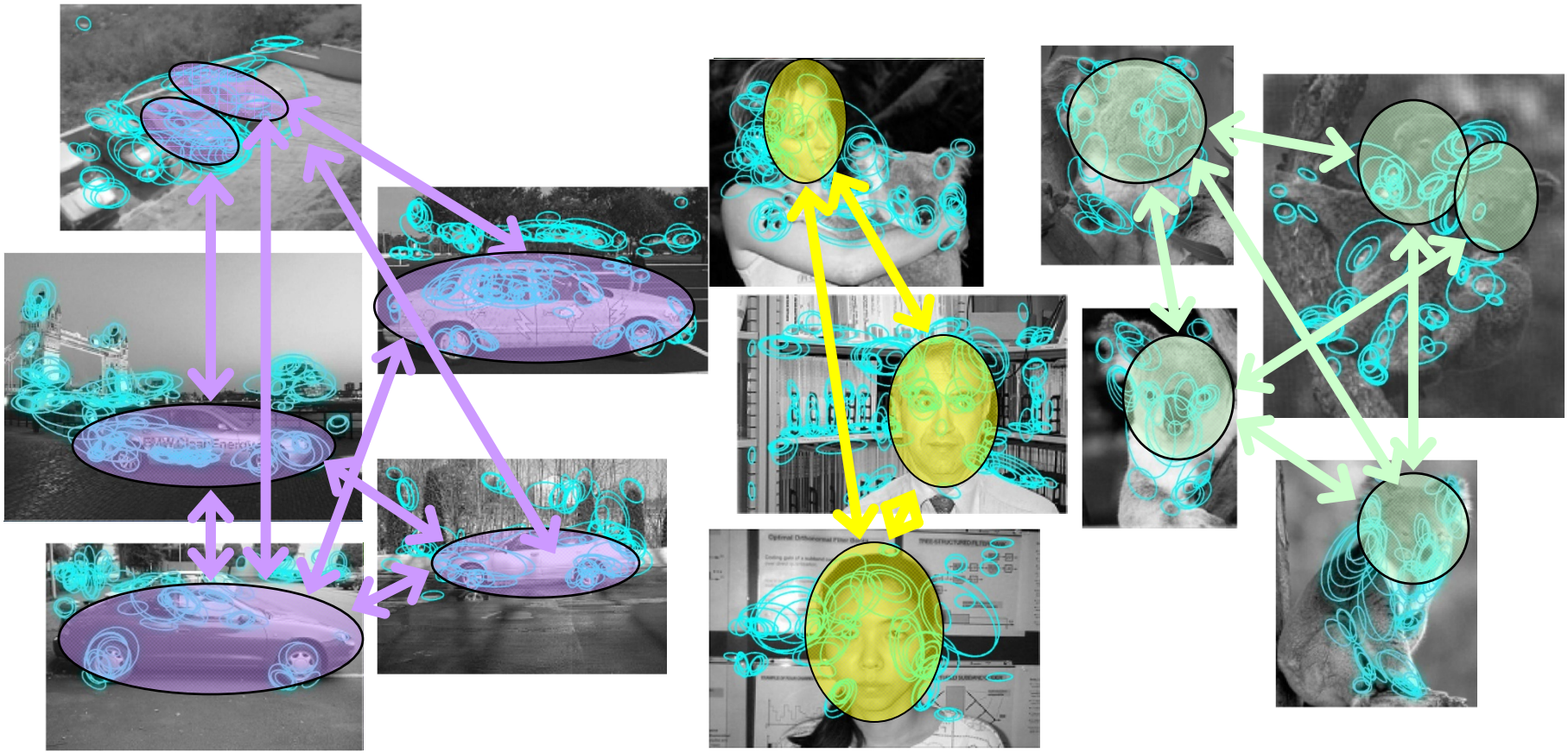


Viewpoint

Partial match graph

[Grauman and Darrell, CVPR 2006]

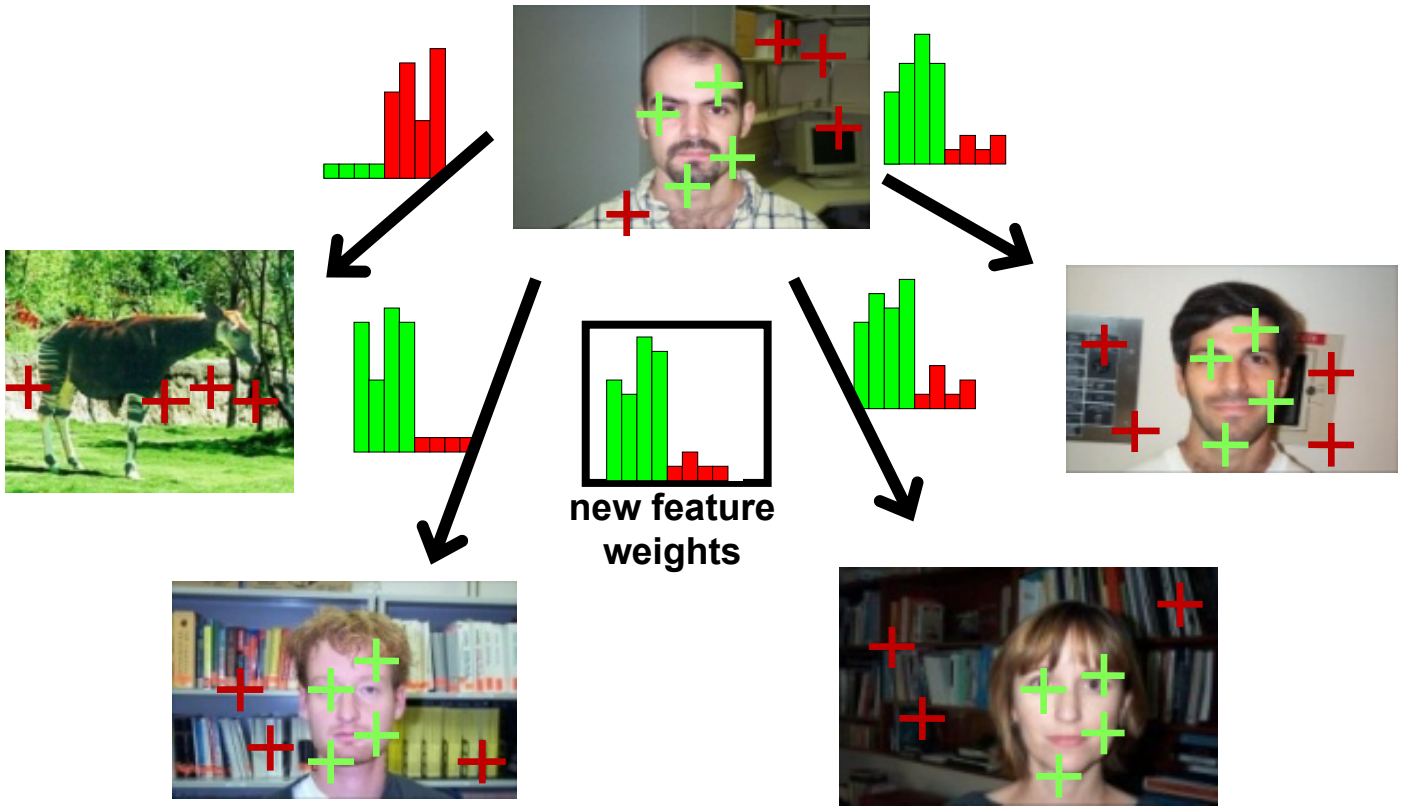
Graph partitioning



Efficiently solve graph partitioning problem to identify initial clusters.

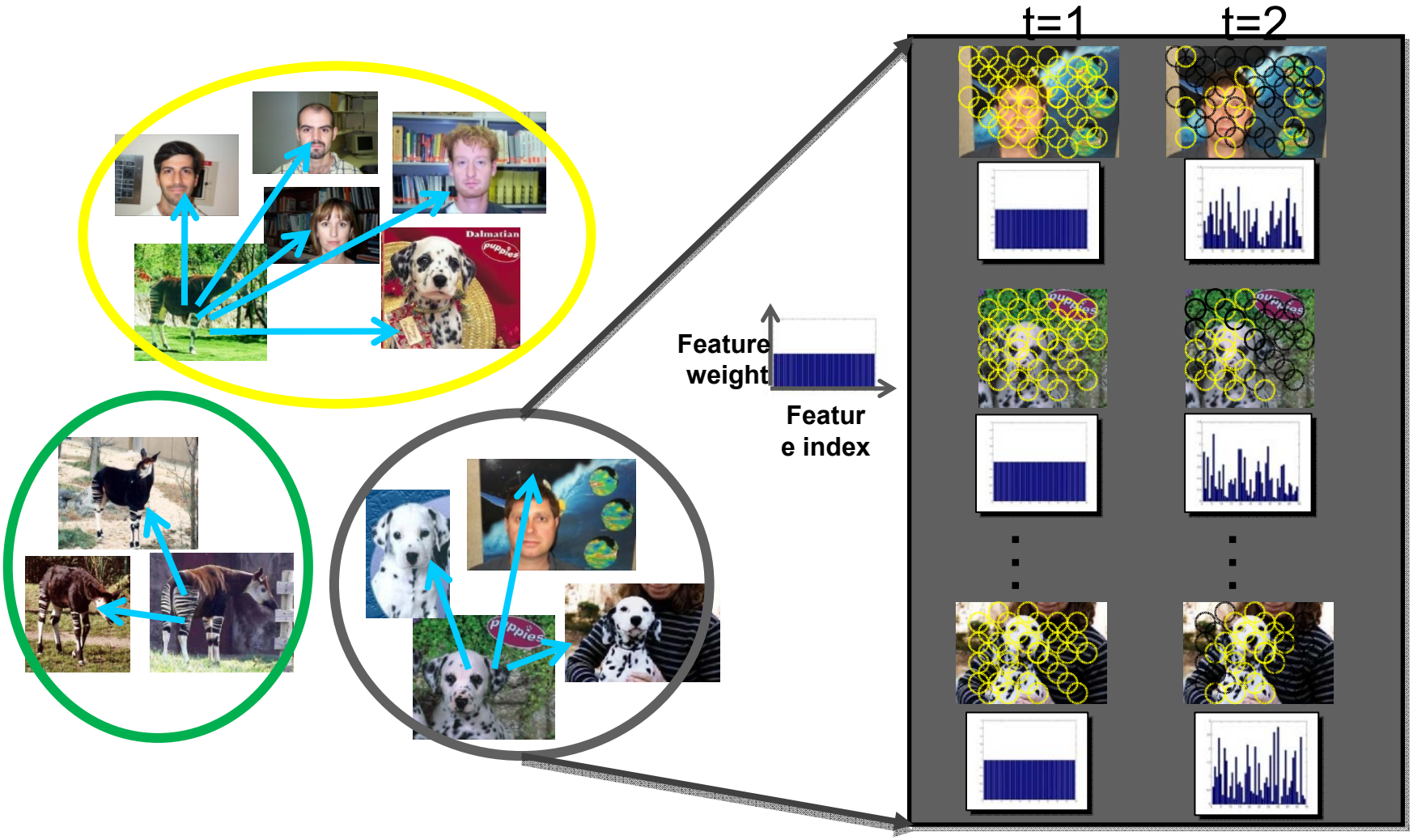
[Grauman and Darrell, CVPR 2006]

Foreground focus



Update the weights (importance) attached to each feature by leveraging any current regions of agreement among the intra-cluster images. *[Lee & Grauman, BMVC 2008]*

Foreground focus

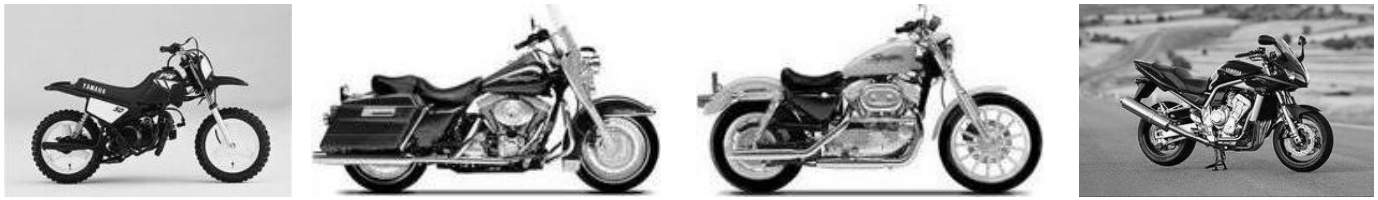


[Lee & Grauman, BMVC 2008]

Unsupervised category discovery

Caltech-4 data set: 3,188 images

Class	Face	Car	Airplane	Motorcycle
Face	99.76	0.00	0.00	0.23
Car	2.47	81.94	0.00	15.5
Airplane	0.33	0.81	81.41	17.44
Motorcycle	2.31	2.78	1.48	93.44



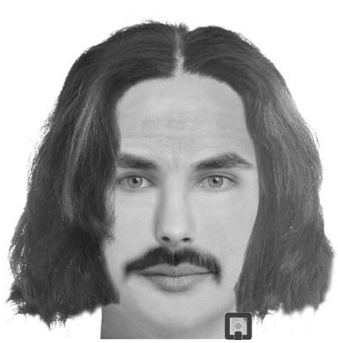
Leveraging text annotations

Search engines already index images based on their proximity to keywords

- \+ easy to collect examples automatically
- \+ lots of data, efficiently indexed
- \- mixed success relying on keywords
- \- more variety than typical recognition datasets

Caltech101 training images: Face category

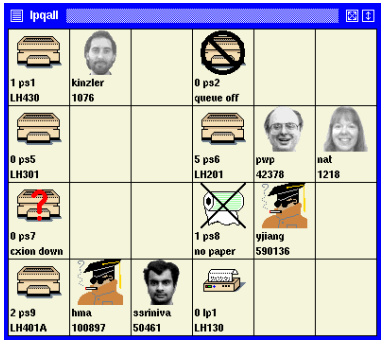
Keyword-based image search : “Face”

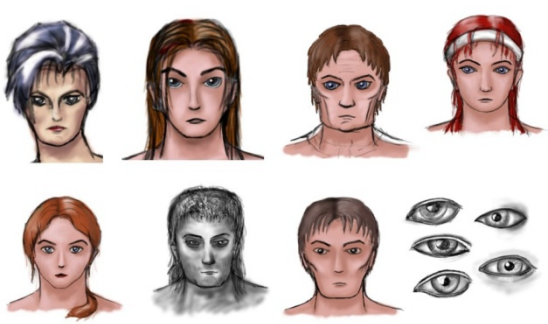











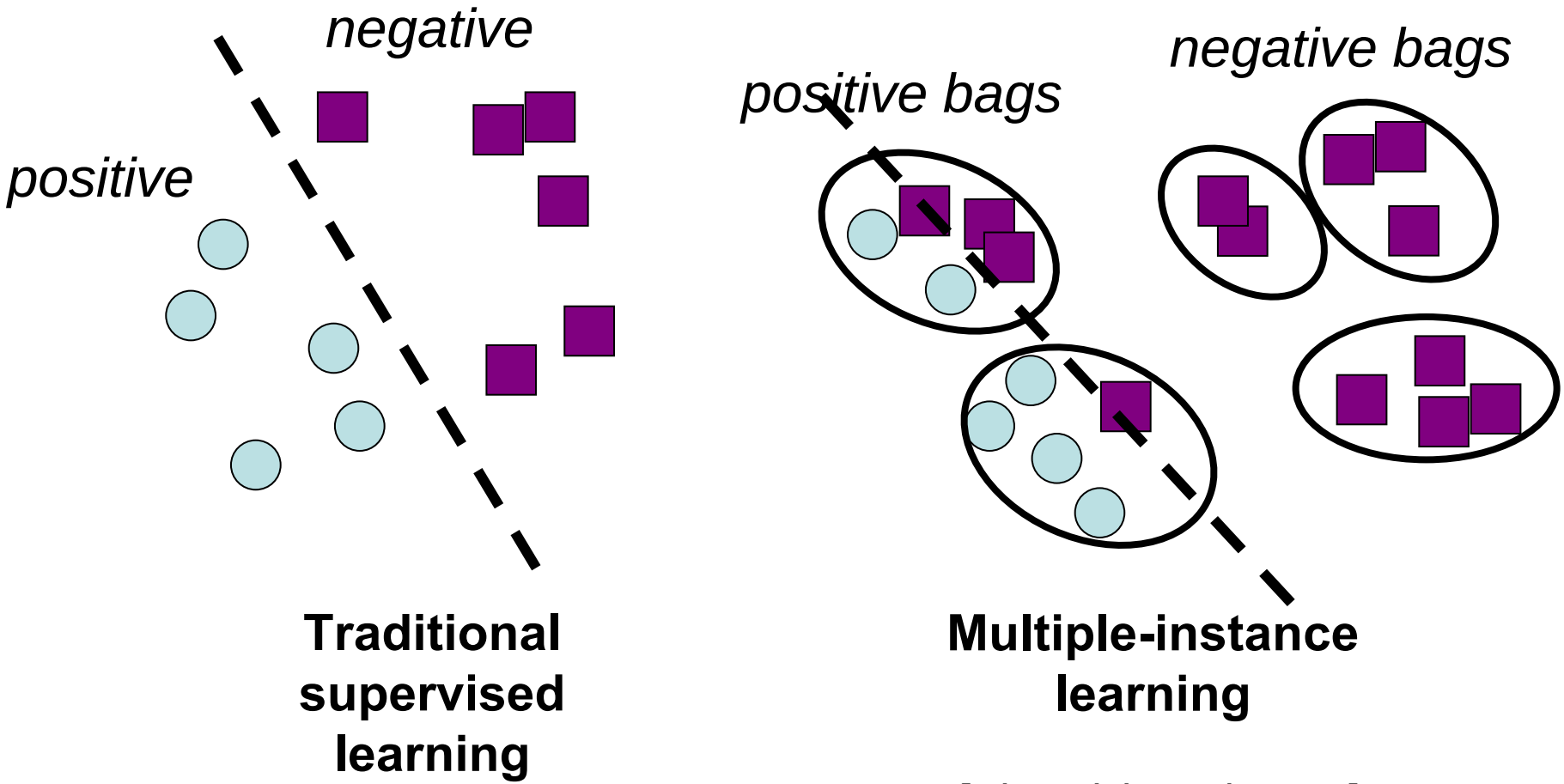



(page 1, Google Image Search)

Keyword-based image search: “Normal face”

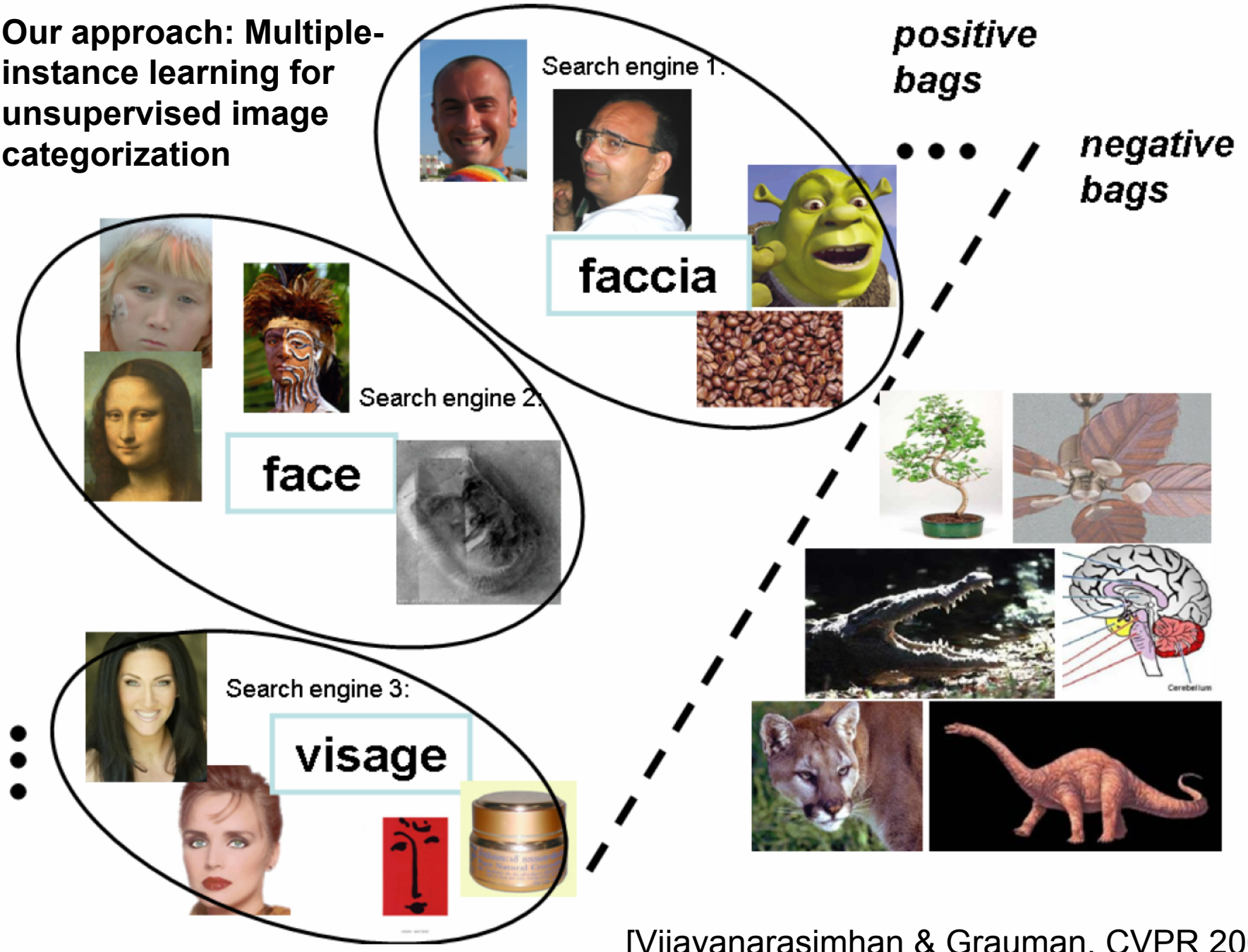
(page 1, Google Image Search)

Multiple-instance learning (MIL)



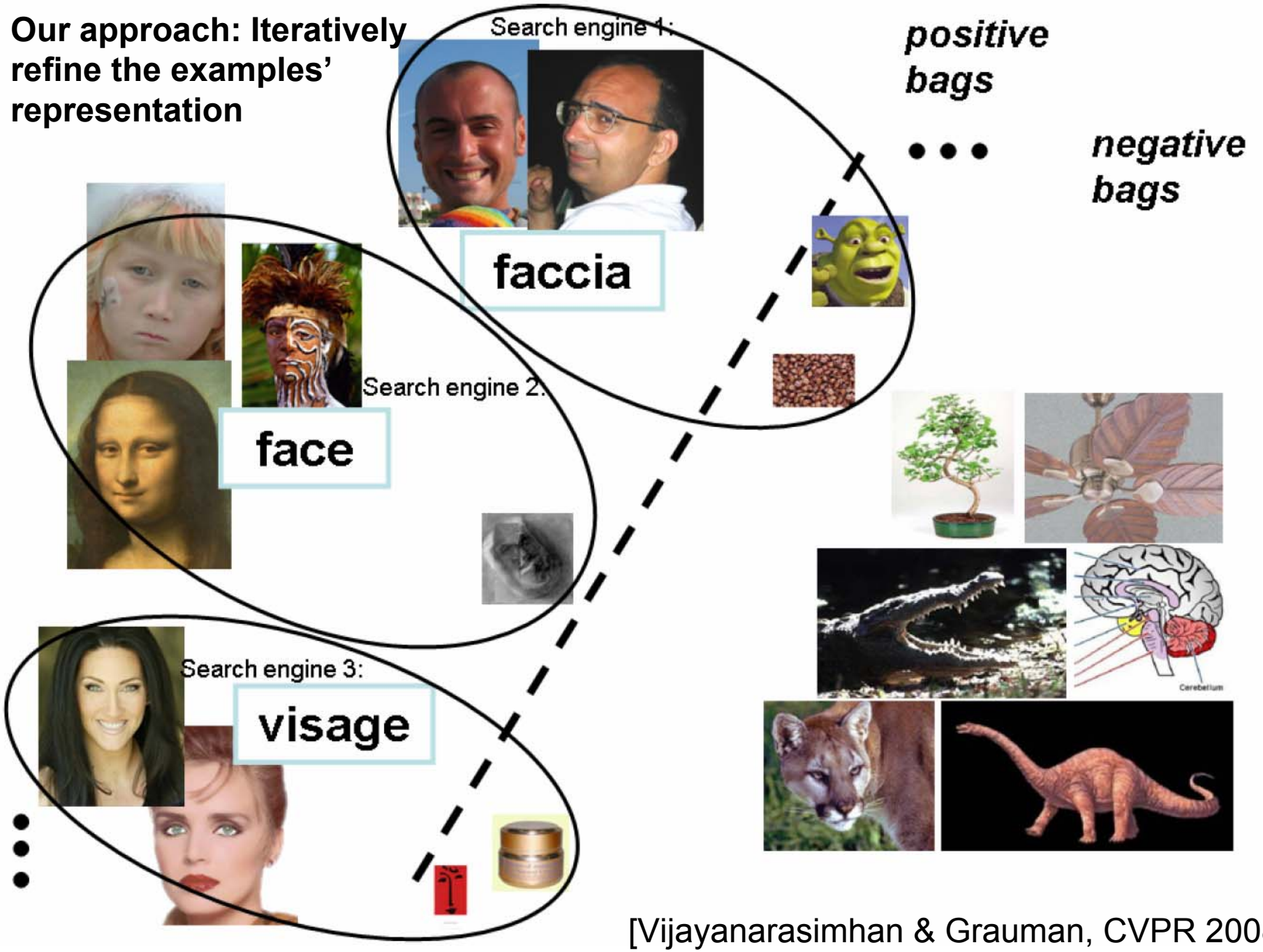
[Dietterich et al. 1997]

Our approach: Multiple-instance learning for unsupervised image categorization



[Vijayanarasimhan & Grauman, CVPR 2008]

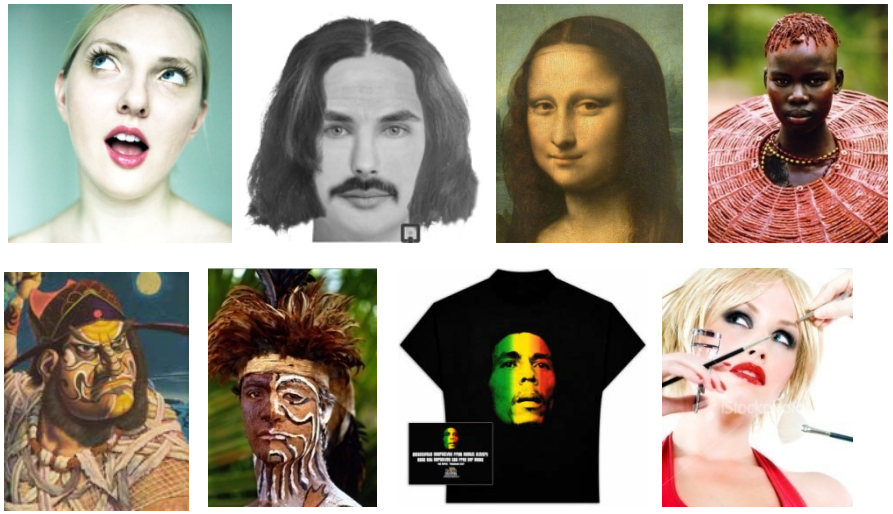
Our approach: Iteratively refine the examples' representation



[Vijayanarasimhan & Grauman, CVPR 2008]

Results: supervised vs. unsupervised

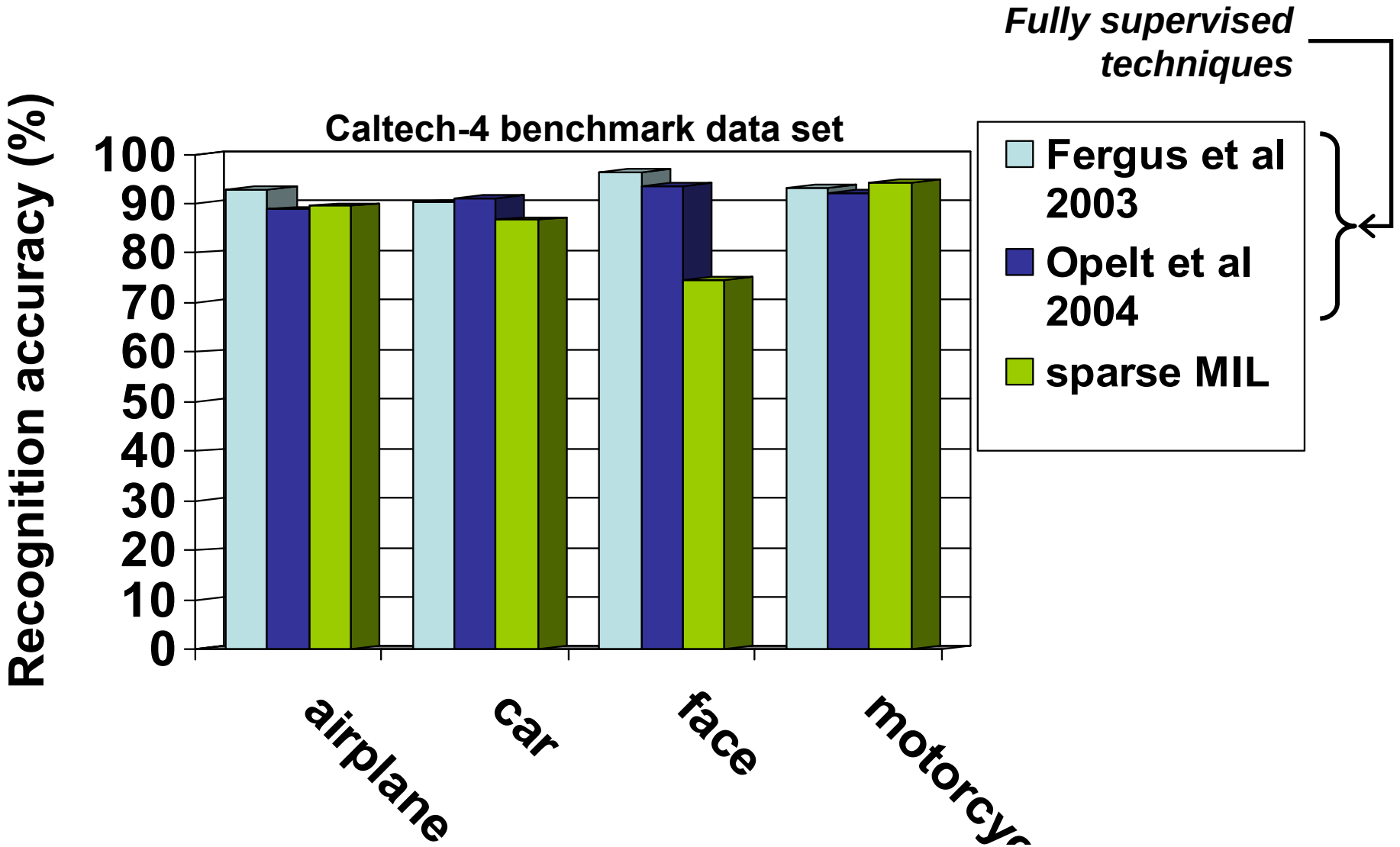
Positive training examples for supervised methods look like this.



Positive training examples for our method look like this.

Positive test examples for all approaches look like this.

Example result: learning from Web images vs. prepared images



Semantic Robot Vision Challenge

A scavenger hunt designed for robots!

Fully automatic training/learning:

System must find out about new categories on the fly, by downloading images from the Web.

Example list:

1. scientific calculator
2. Ritter Sport Marzipan
3. book “Harry Potter and the Deathly Hallows”
4. DVD “Shrek”
5. DVD “Gladiator”
6. CD “Hey Eugene” by Pink Martini
7. fork
8. electric iron
9. banana
10. green apple
11. red bell pepper
12. Lindt Madagascar
13. rolling suitcase
14. Spam
15. Twix candy bar
16. Tide detergent
17. Pepsi bottle
18. yogurt Kettle Chips
19. upright vacuum cleaner

Semantic Robot Vision Challenge

Object List

Google

Live Search

picsearch

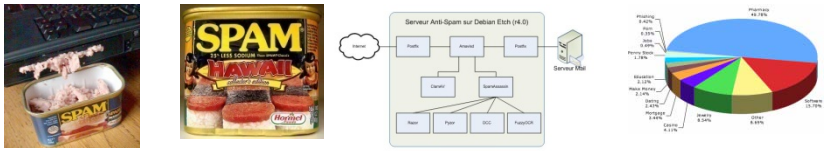
Crawl the Web for data

Robot Images

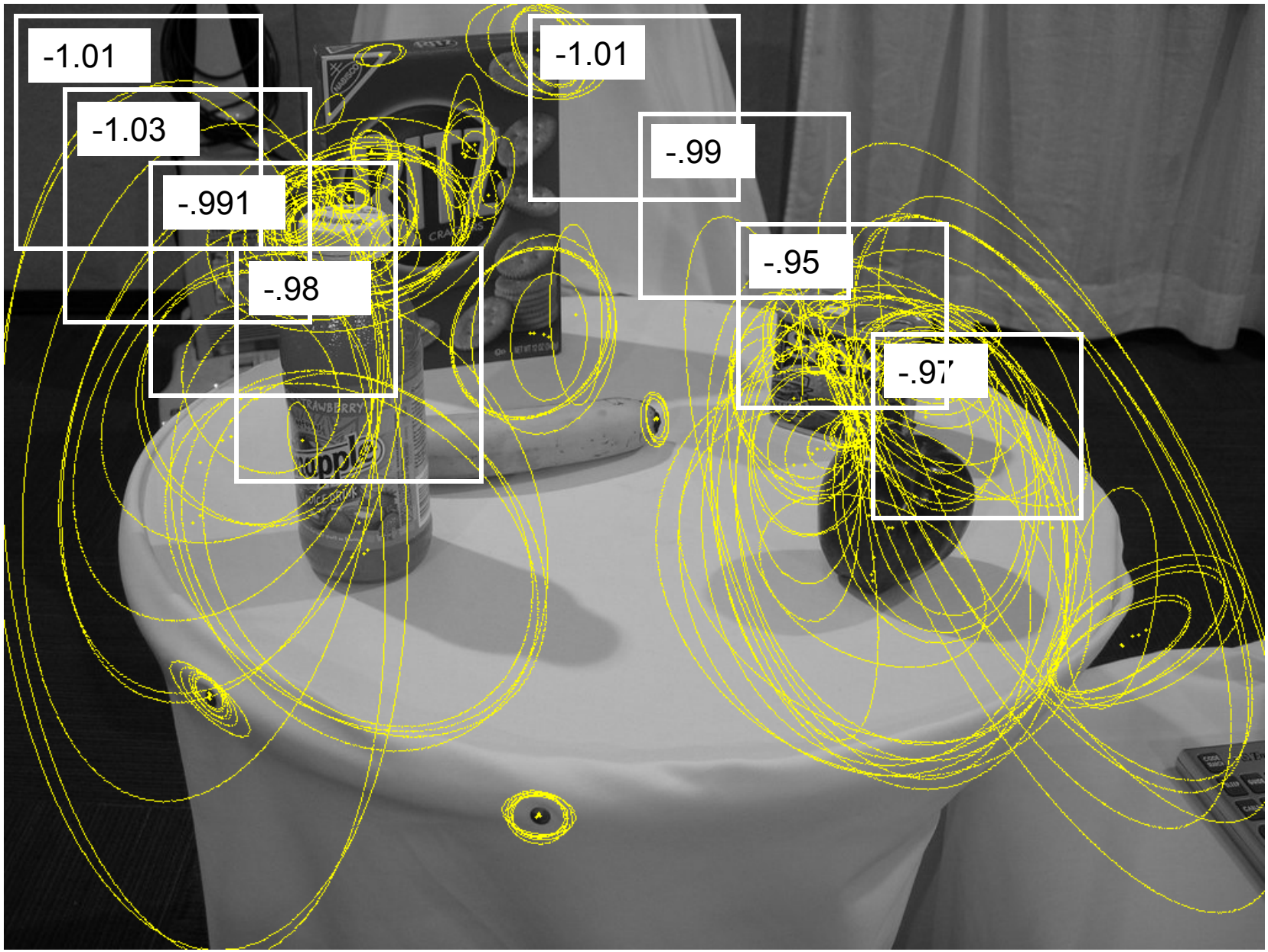
Classifier

Slide credit: Fei-Fei Li

Example bags (Spam category)

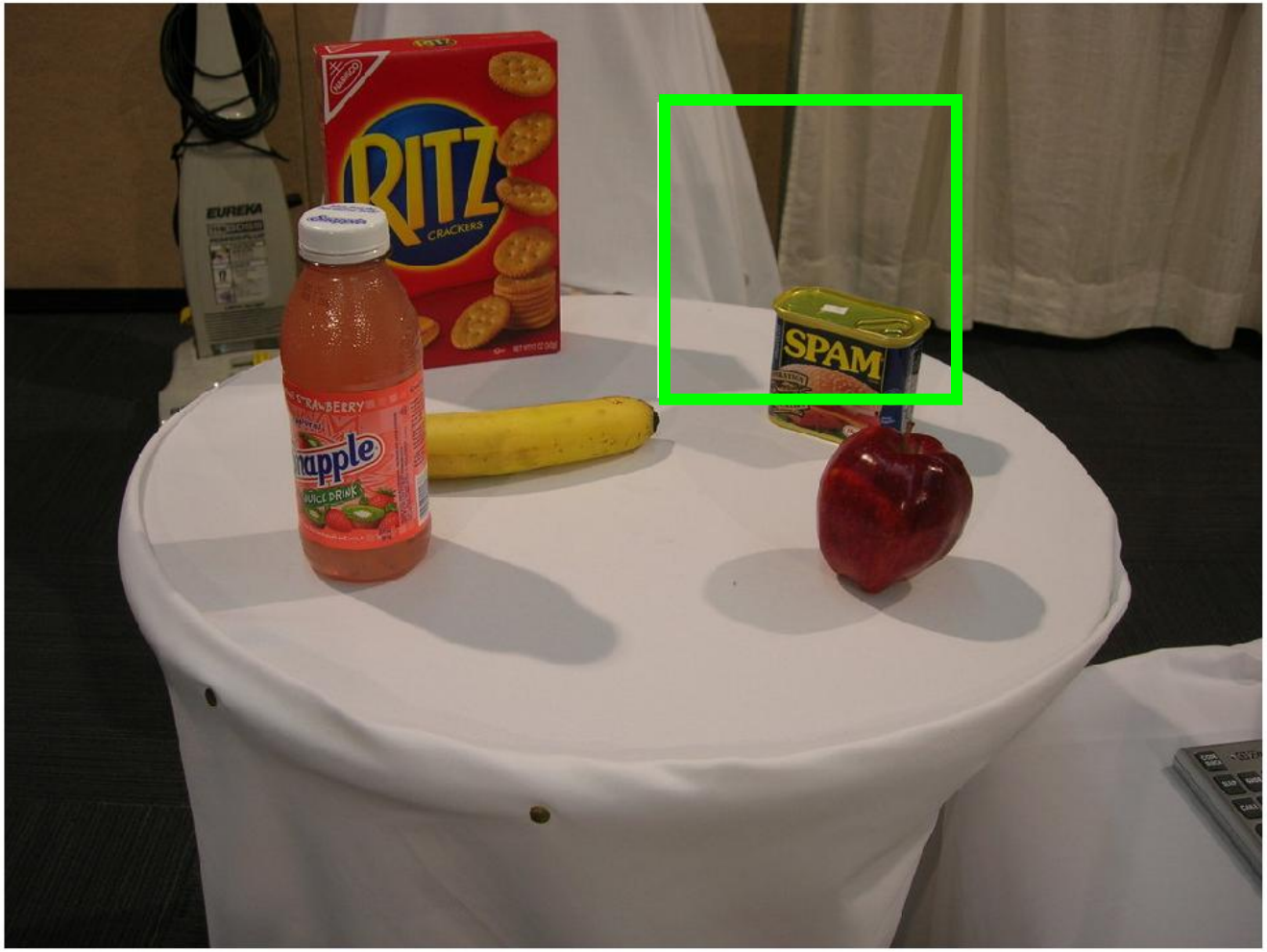
Engine	Language	Bag
Google	English	
Google	French	
Google	German	
Yahoo	English	
MSN	English	
MSN	French	
MSN	German	

Test phase



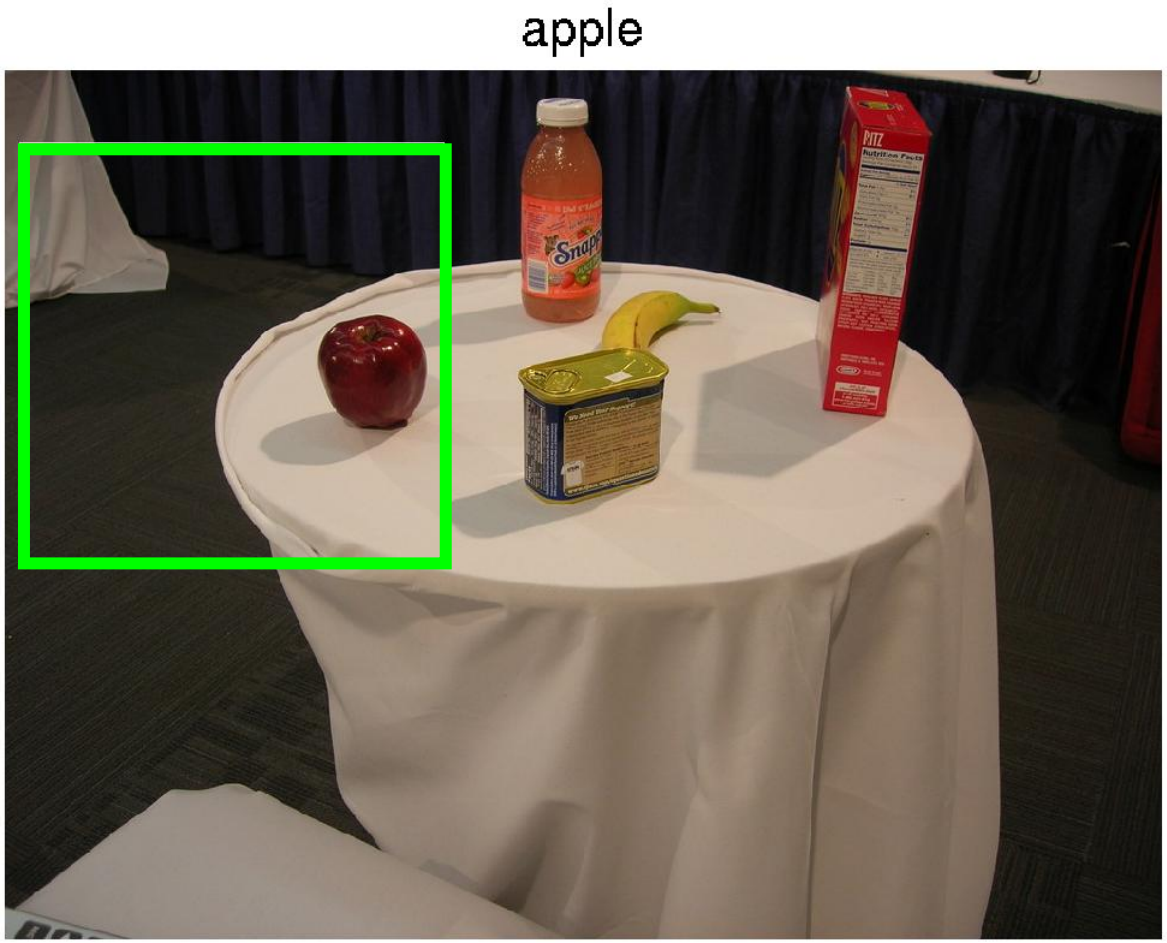
Challenge results

Spam



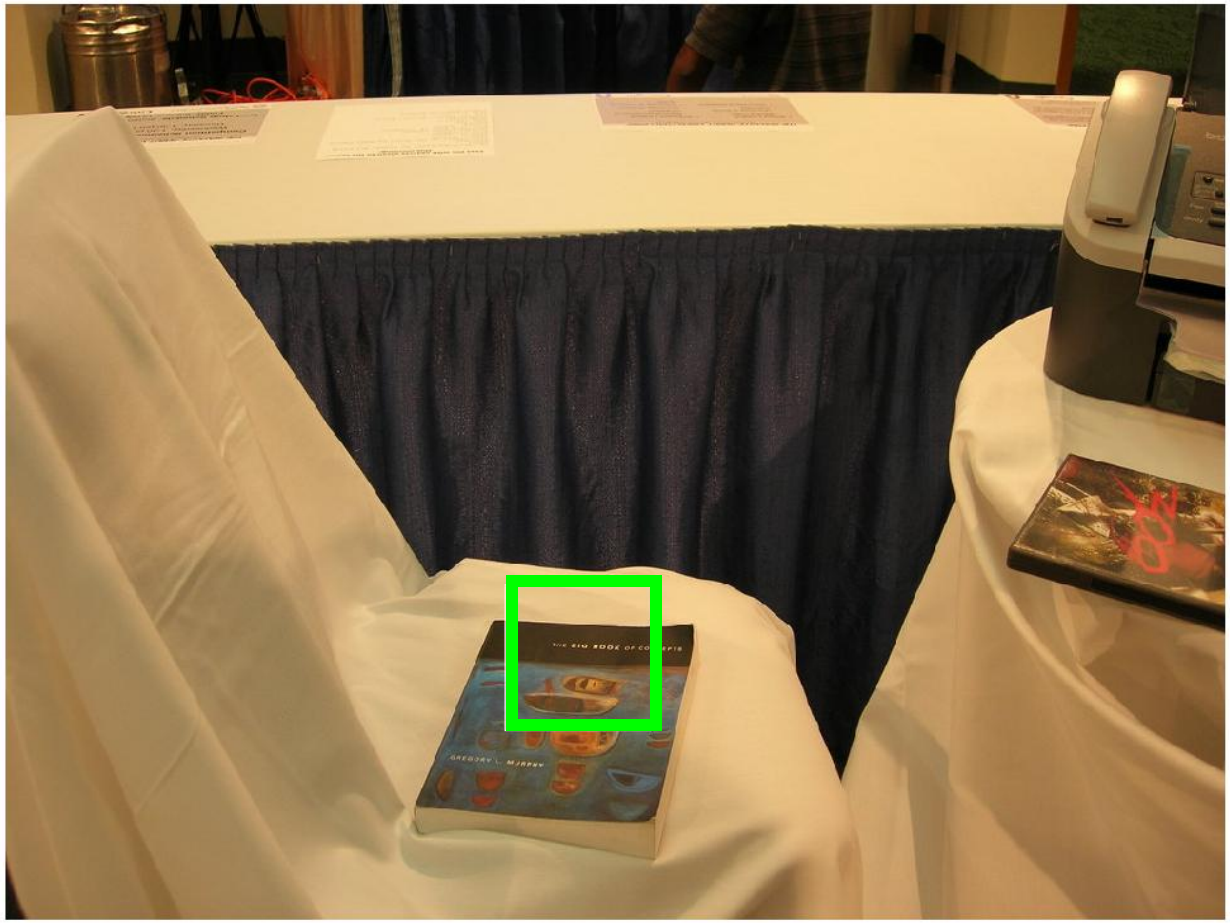
(sMIL Spam Filter)

Challenge results



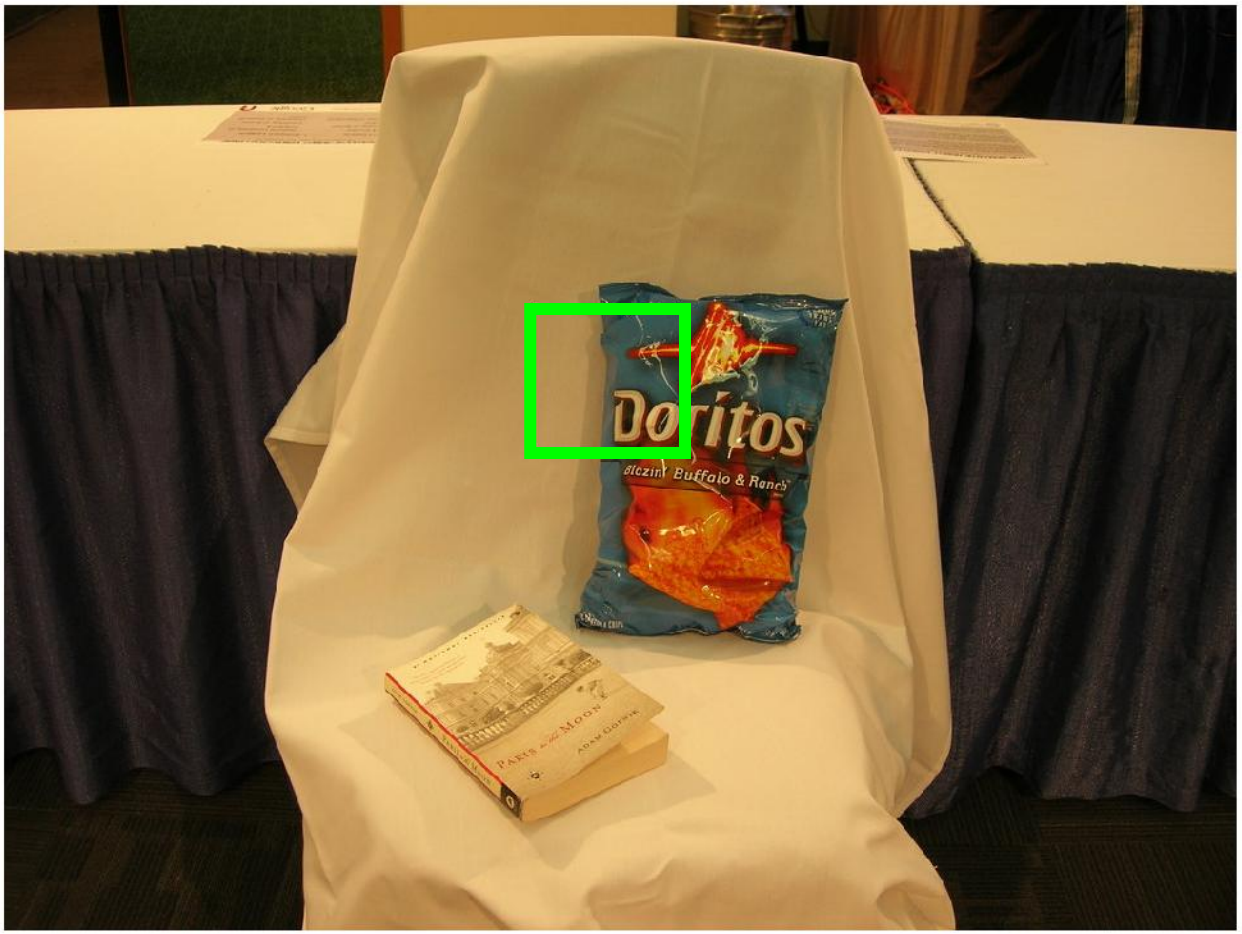
Challenge results

book "Big Book of Concepts"



Challenge results

Doritos Blazin' Buffalo Ranch



Challenge results

fax machine

Practice round results

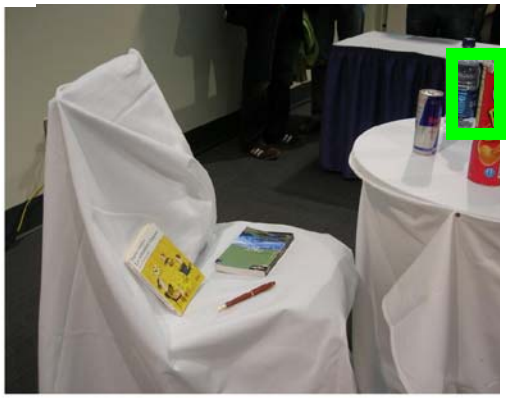
Upright Vacuum cleaner

Brown pen

Nescafe Taster's Choice

Pellegrino bottle

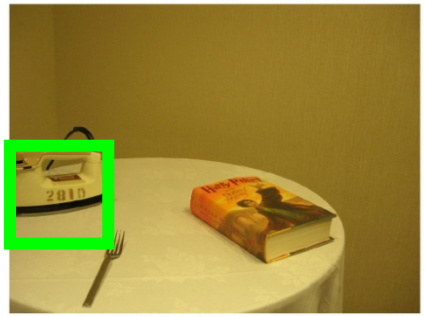
Pringles



Red sport bottle

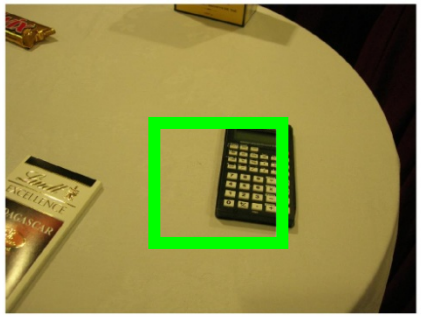
Qualification round results

Electric iron

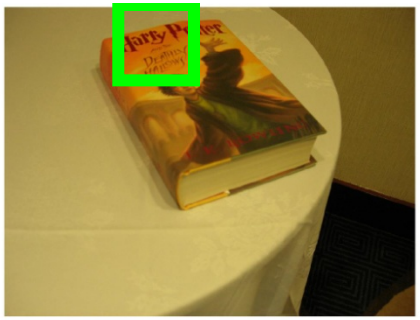


Upright vacuum cleaner

Scientific calculator



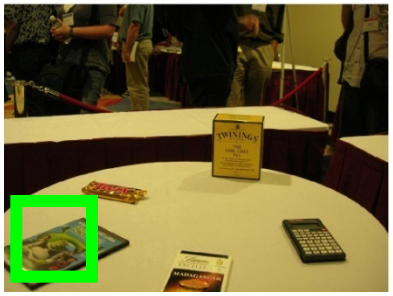
Harry potter and the deathly hallows



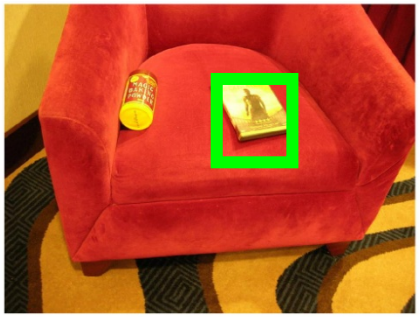
Lindt Madagaskar

Twix candy bar

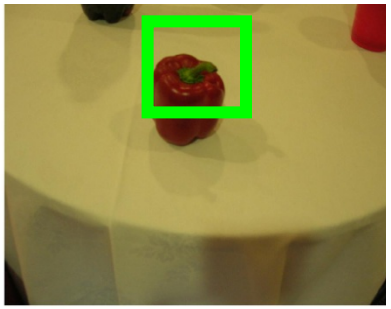
DVD “shrek”



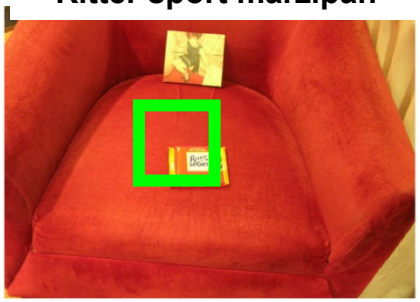
DVD “gladiator”



Red bell pepper



Ritter sport marzipan



Tide detergent

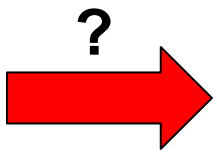
Reducing costs

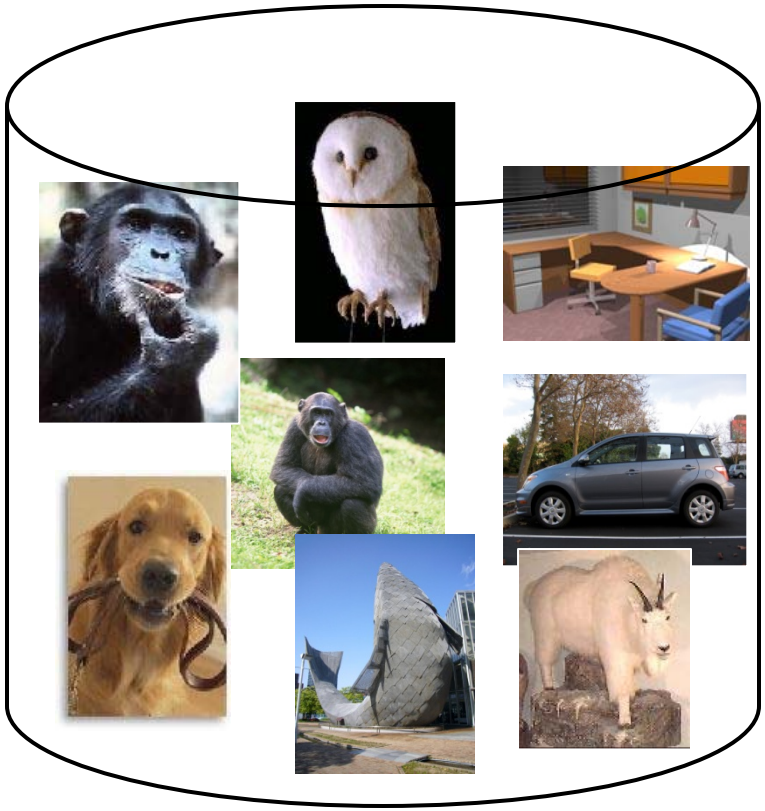
- Removing heavy reliance on **human effort** is important to make recognition scalable
- **Computational cost** reductions are also critical

Motivation

- Fast image search is a useful component for a number of vision problems.

Object categorization

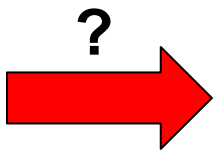


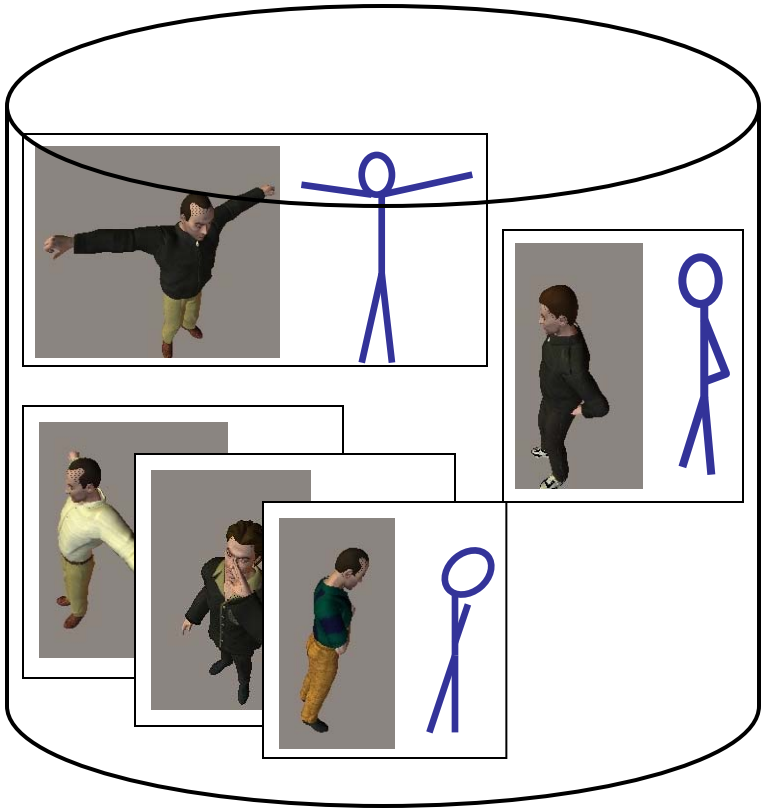


Motivation

- Fast image search is a useful component for a number of vision problems.

Example-based pose estimation

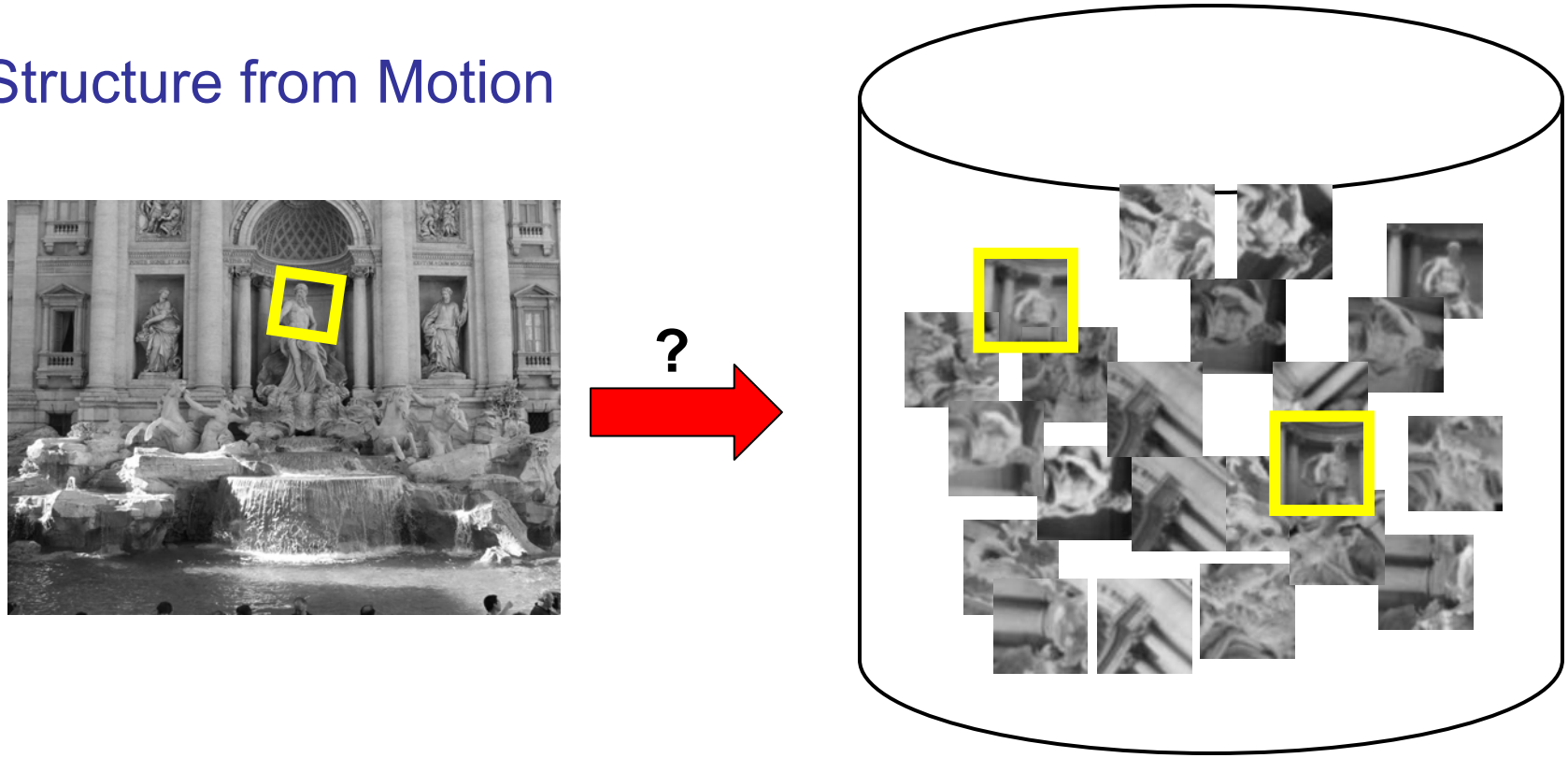




Motivation

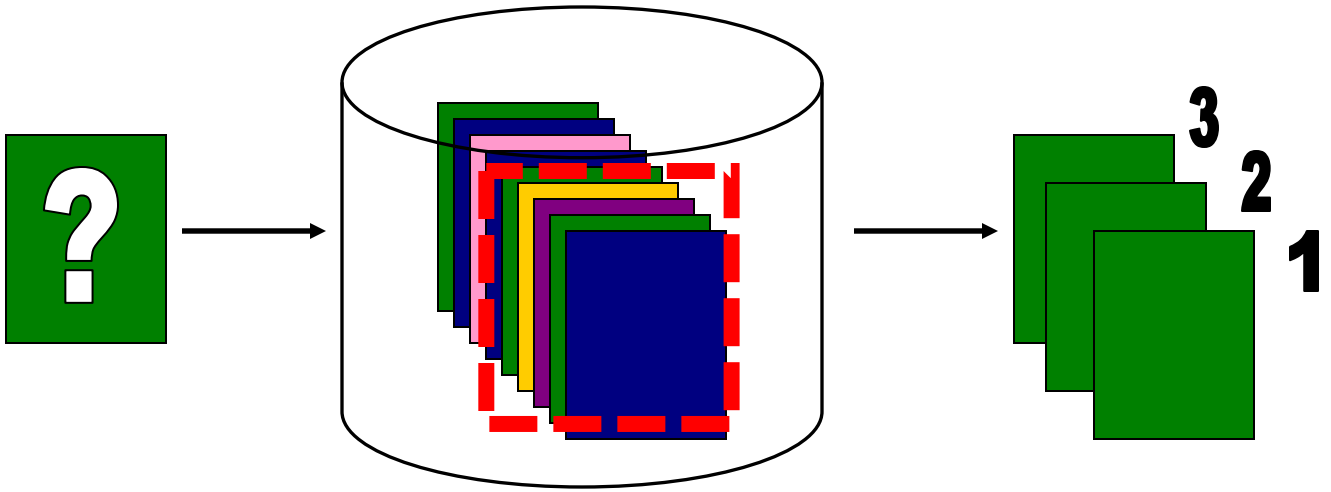
- Fast image search is a useful component for a number of vision problems.

Structure from Motion



Goal: sub-linear time search

- Content-based search and retrieval

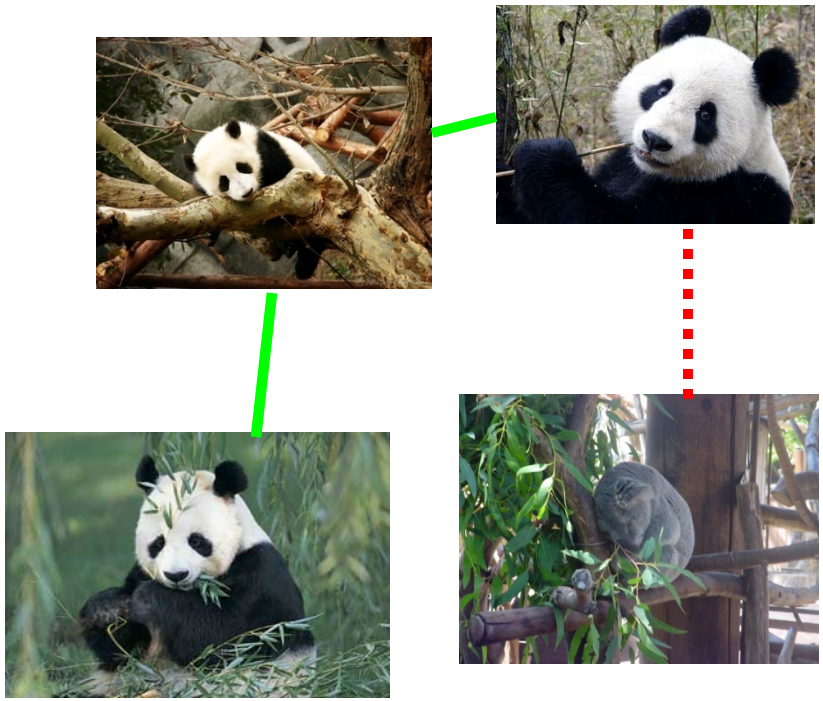


Query image or video

Database

Examples that are most relevant to the query

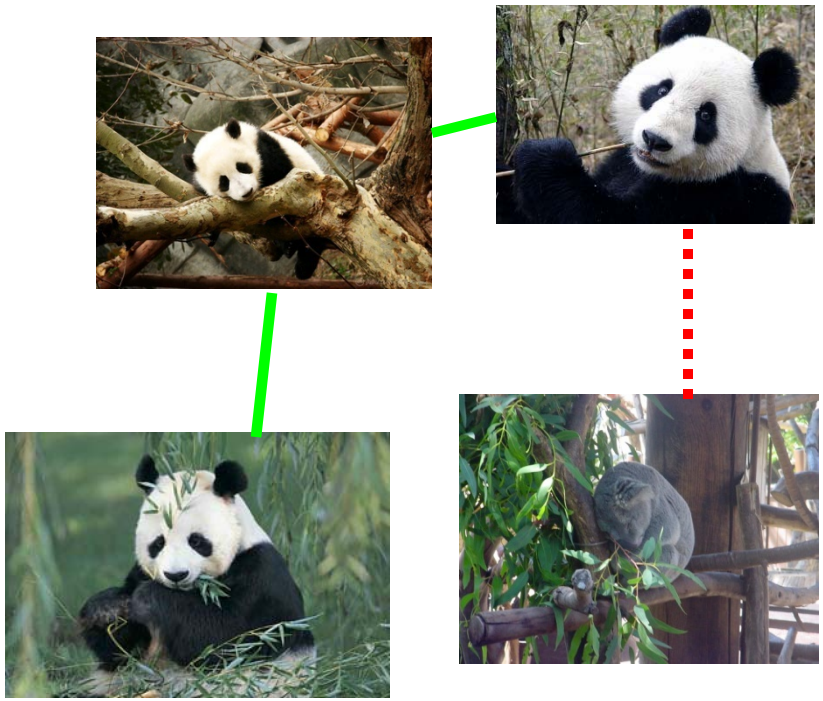
Metric learning



There are various ways to judge appearance/shape similarity…

but often we know more about (some) data than just their appearance.

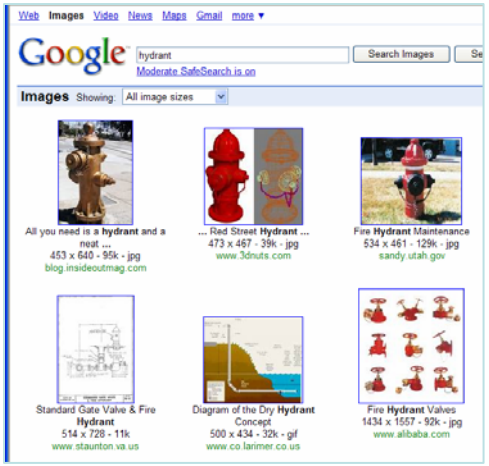
Metric learning



- Exploit partially labeled data and/or (dis)similarity constraints to construct more useful distance function
- Various existing techniques

Example sources of similarity constraints

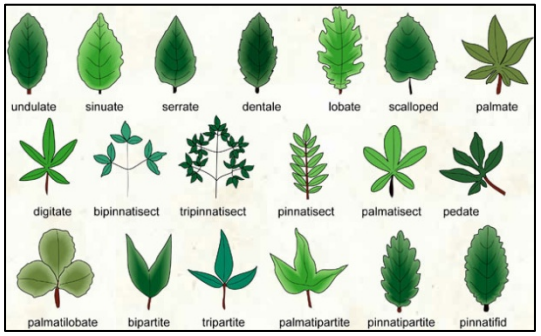
Partially labeled image databases



User feedback

Detected video shots, tracked objects

Fully labeled image databases

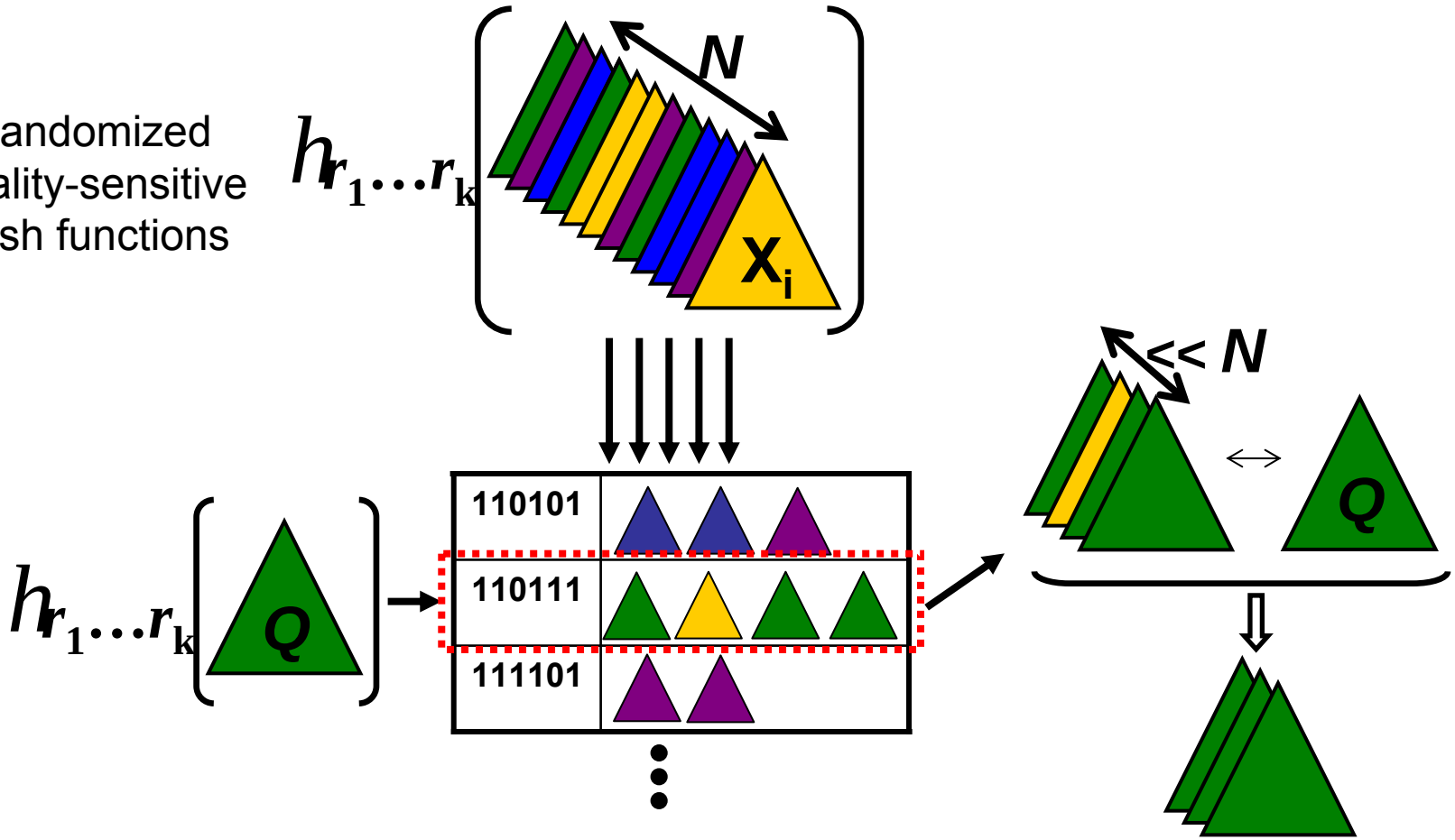


Problem-specific knowledge

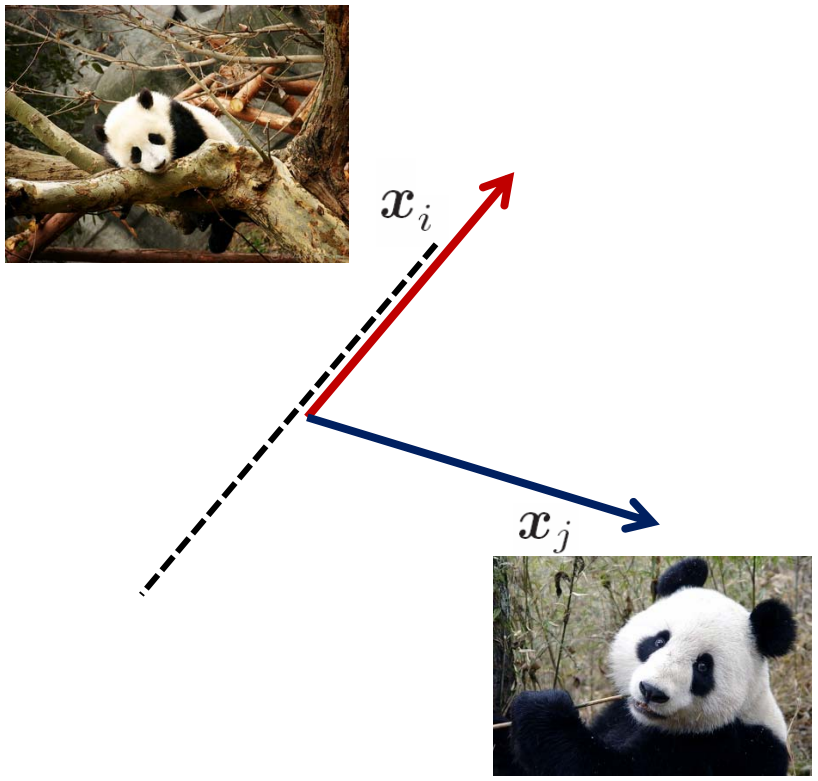
Sub-linear time search

$$\Pr_{h \in \mathcal{F}} [h(x) = h(y)] = sim(x, y)$$

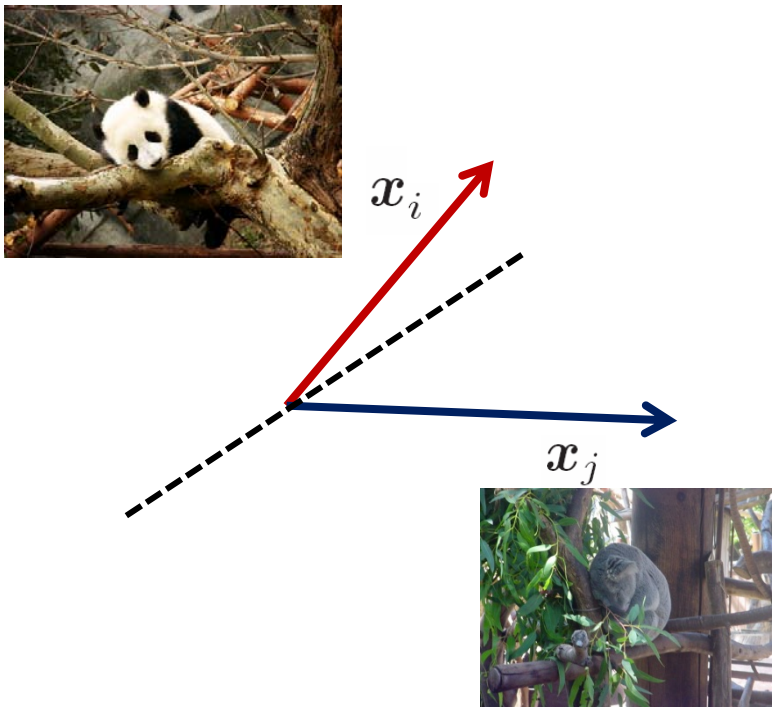
Randomized locality-sensitive hash functions



Hash functions for learned metrics



It should be unlikely that a hash function will split examples like those having similarity constraints…



…but likely that it splits those having dissimilarity constraints.

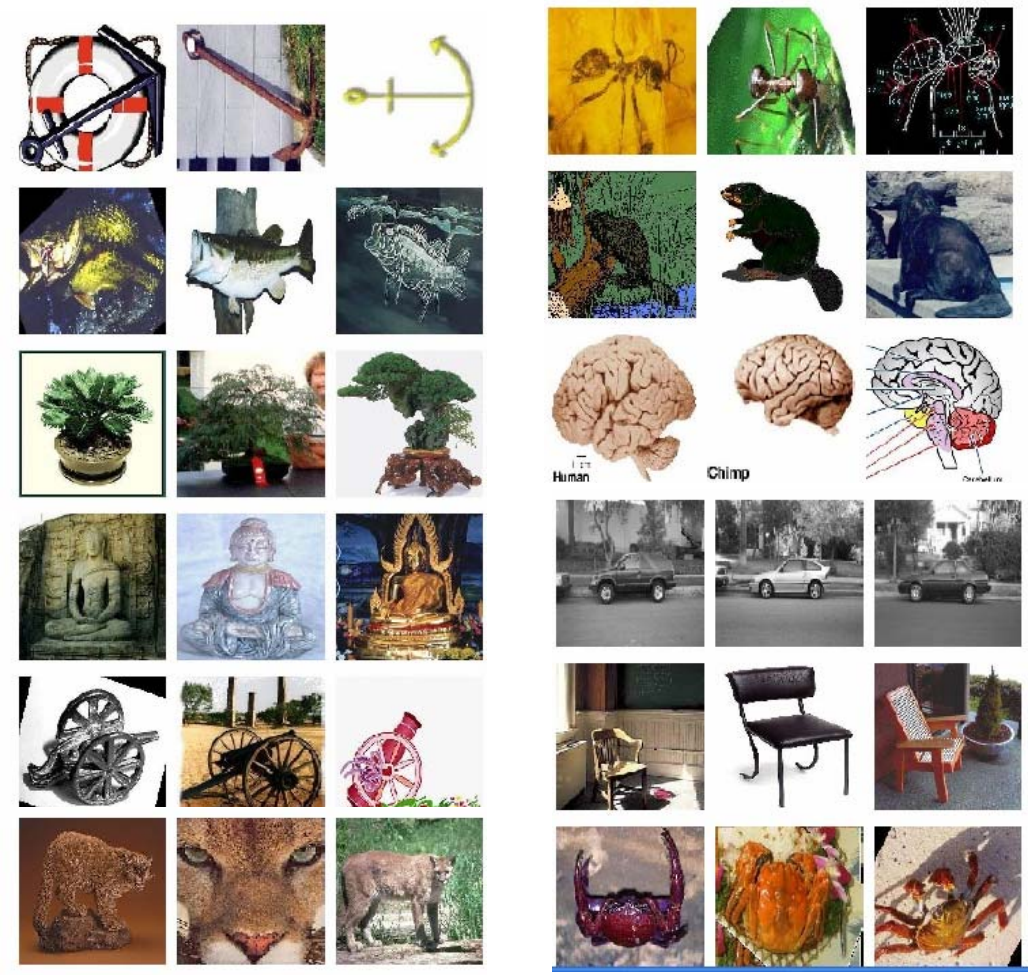
[Jain, Kulis, & Grauman, CVPR 2008]

Caltech-101 dataset

- **Caltech101 data set**

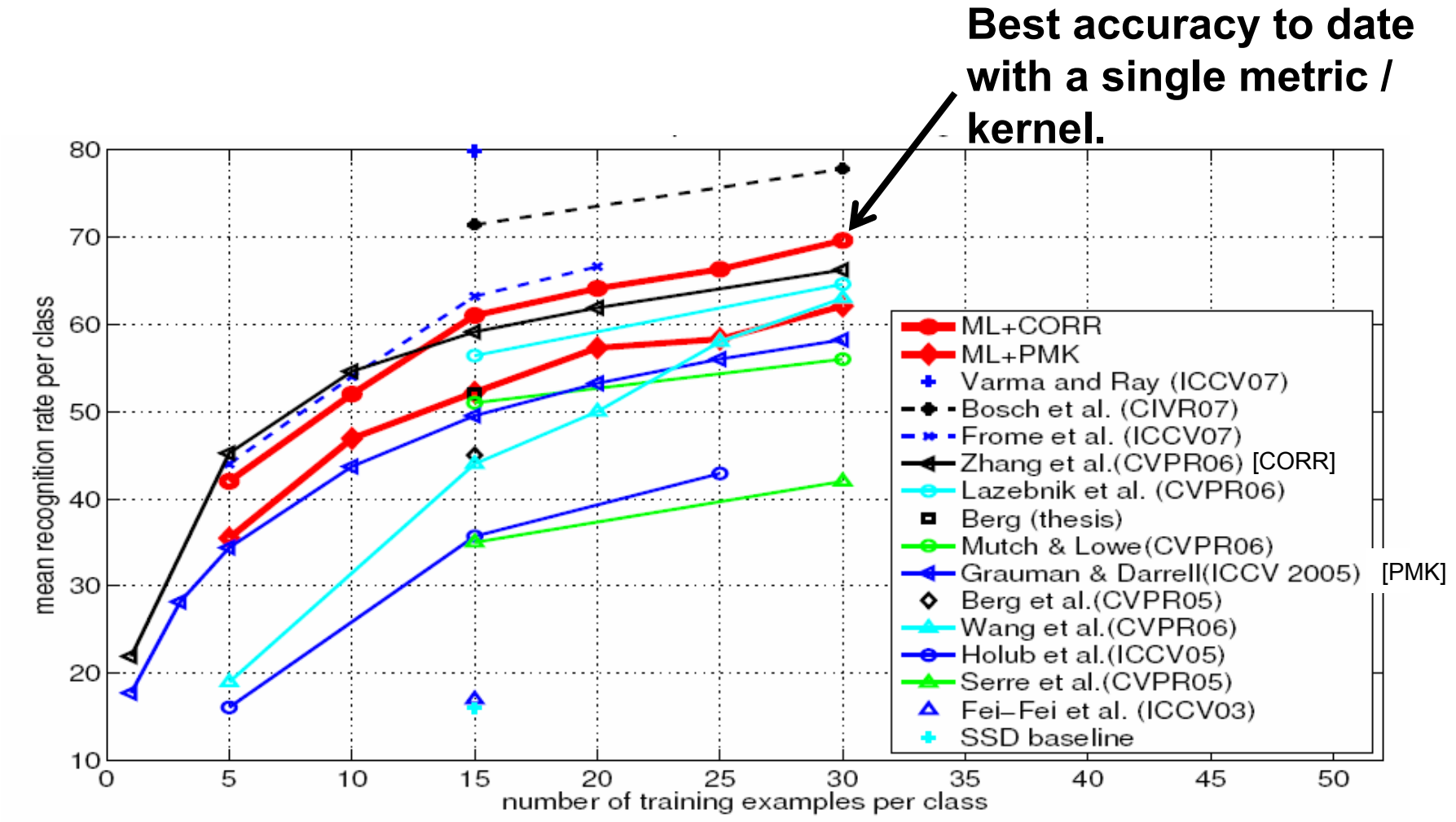
  101 categories
  40-800 images per class

- **Features**:
  - Densely sampled
  - SIFT descriptor + spatial
  - Average *m*=1140 per set



Data provided by Fei-Fei, Fergus, and Perona

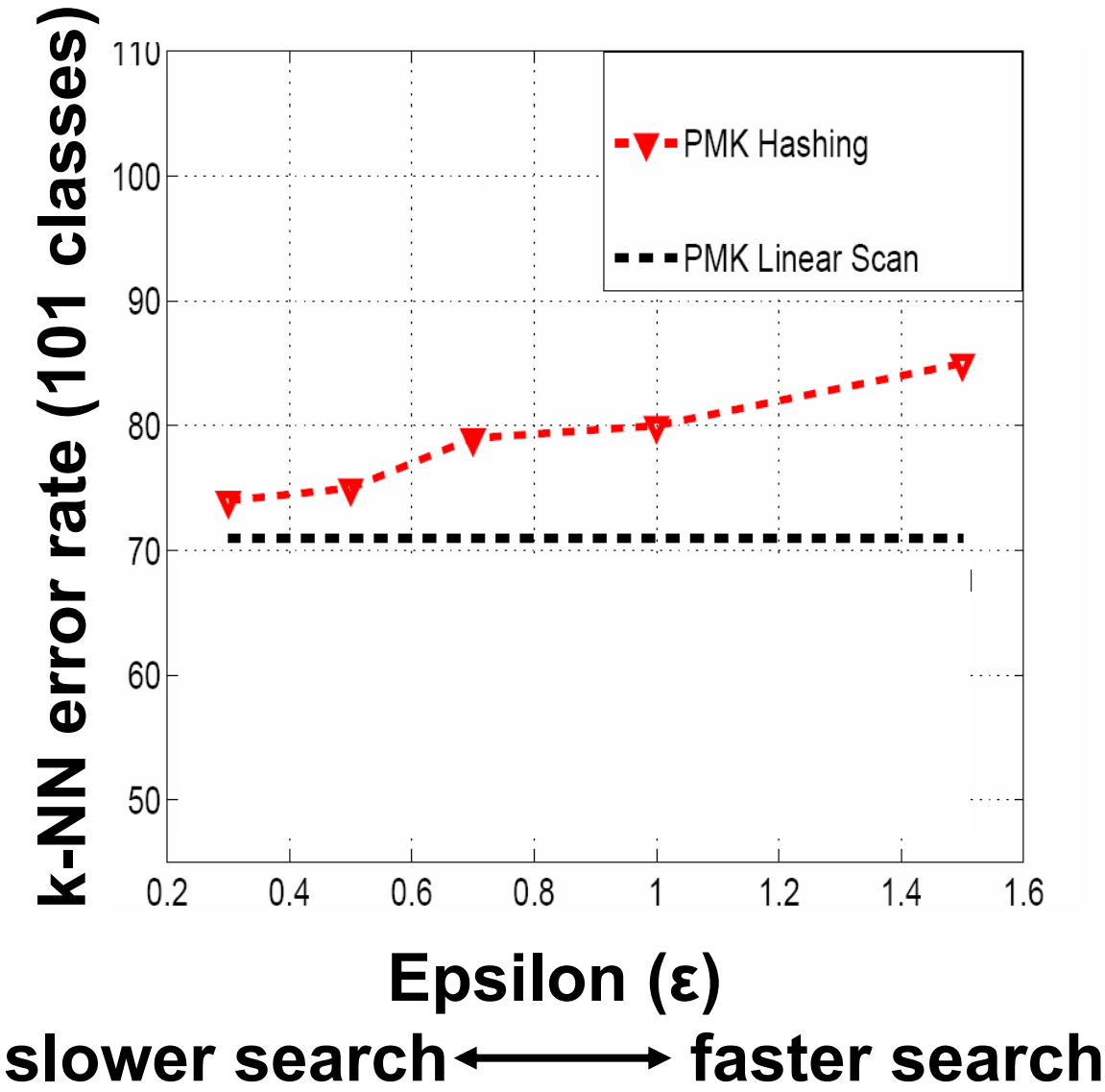
Results: object categorization



Caltech-101 database

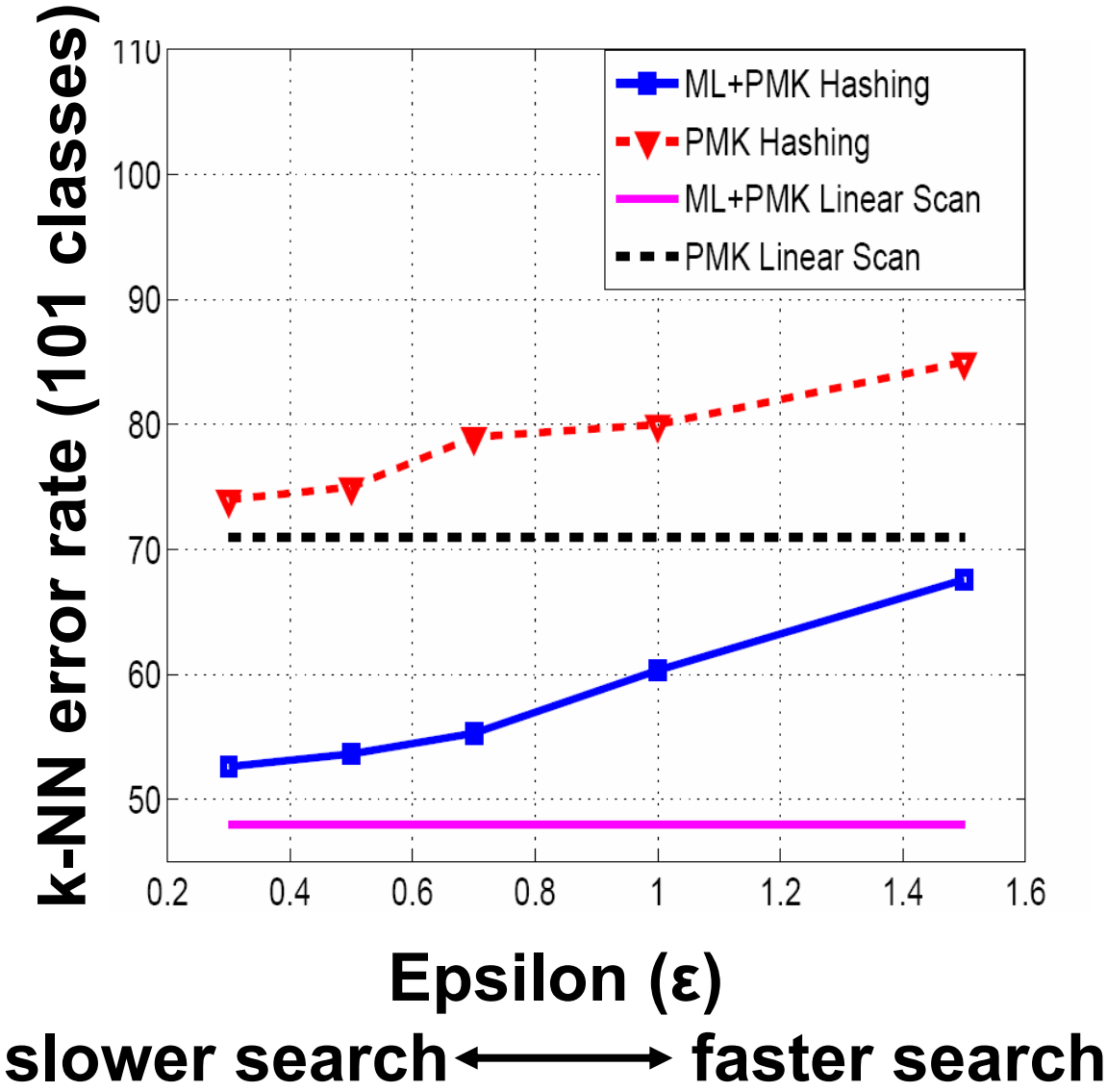
ML = metric learning

Results: object categorization



- Query time controlled by required accuracy
- e.g., search less than 2% of database examples for accuracy close to linear scan

Results: object categorization



- Query time controlled by required accuracy
- e.g., search less than 2% of database examples for accuracy close to linear scan

Photo Tourism Data

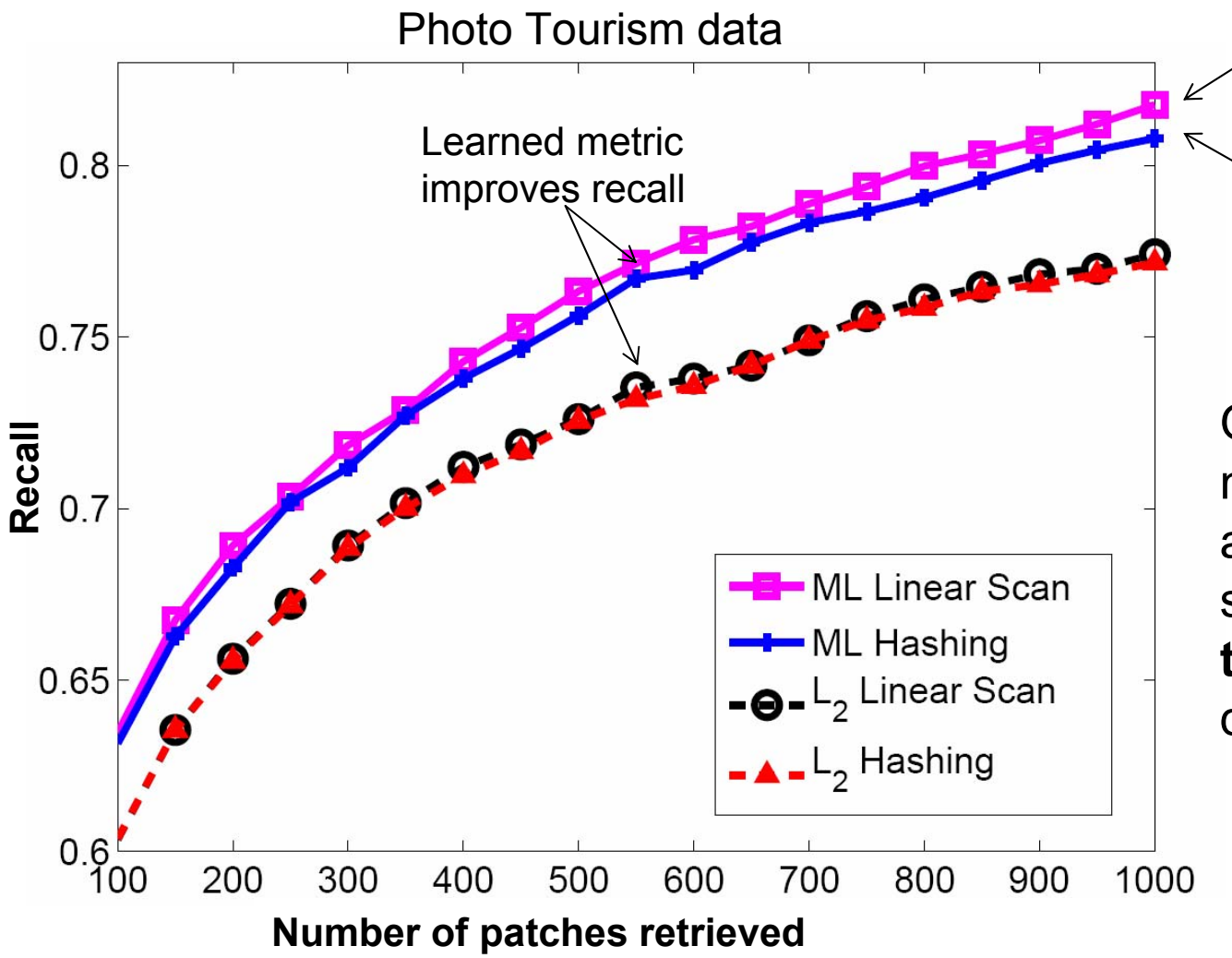
Results: patch indexing

O(10^5) patches

- Photo Tourism data: goal is to match patches that correspond to same point on 3d object
- More accurate matches → better reconstruction
- Huge search pool

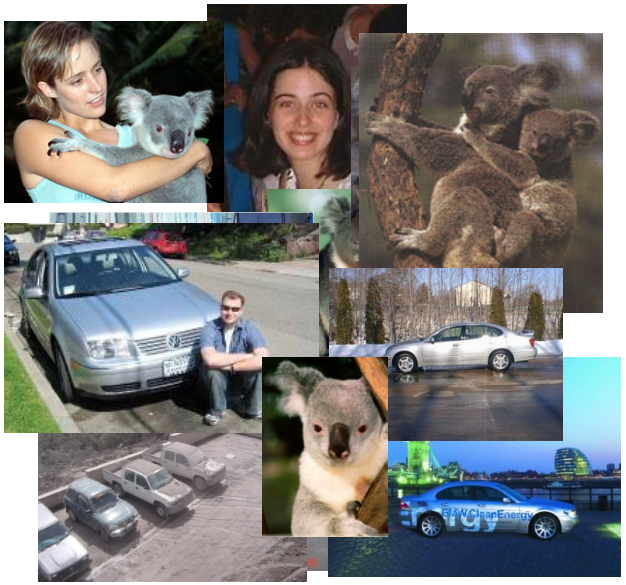
[Photo Tourism data provided by Snavely, Seitz, Szeliski, Hua, Winder & Brown]

Results: patch indexing



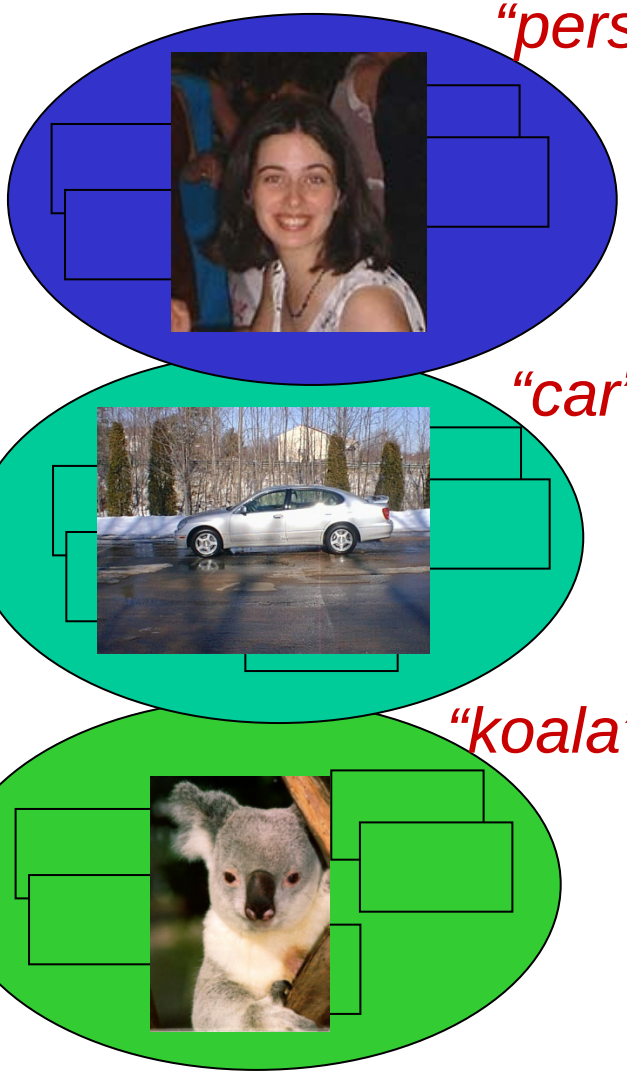
Our technique maintains accuracy while searching **less than 1%** of the database.

1. User takes unstructured videos and photos…



2. Clusters are formed automatically

Soon…



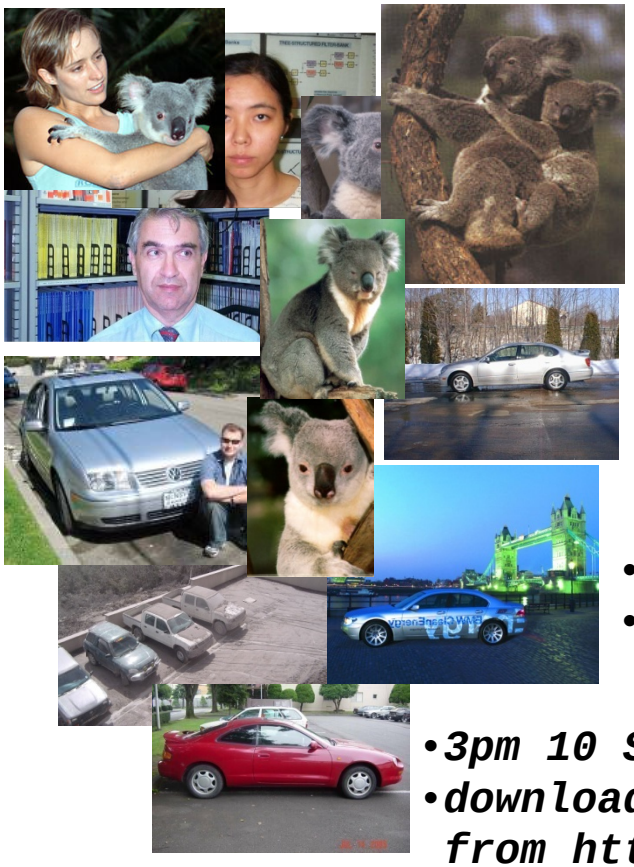
3. A few images in each cluster are interactively labeled

4. Meta-data labels are extrapolated to the entire collection

It's not just vision…

Integrate with mobile sensor information (GPS, time, nearby objects or people), calendar, schedule…

Infer *semantically rich* meta-data labels from joint sources.



- ***10am 7 Sep 05***
- ***Australian park***
- ***Jim, Jill nearby***

- ***4pm 8 Sep 05***
- ***Sydney***

- ***8pm 10 Oct 05***
- ***London***

- ***3pm 10 Sep 05***
- ***downloaded from http://...***

"two koalas seen on nat. park trip with Jim and Jill"

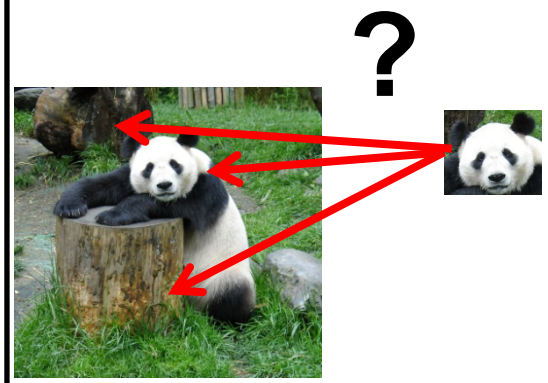
"Jill and koala on nat. park trip"

"John and his new car"

"office parking lot"

"car to consider purchasing"

Possible programming exercises



Template matching

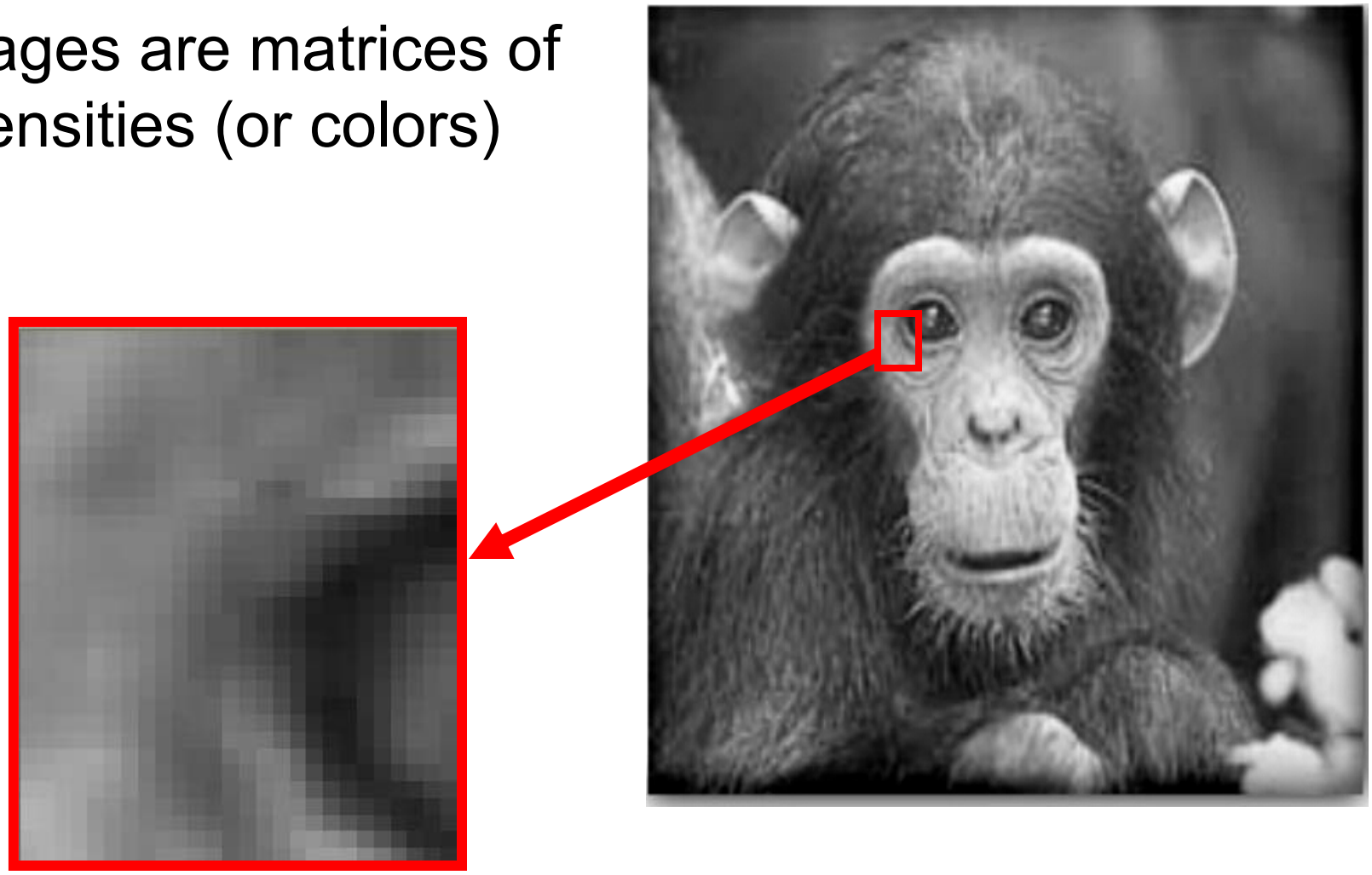
Color-based retrieval

Background subtraction

Main programming requirement: operations on 2-d arrays, loops, file I/O.

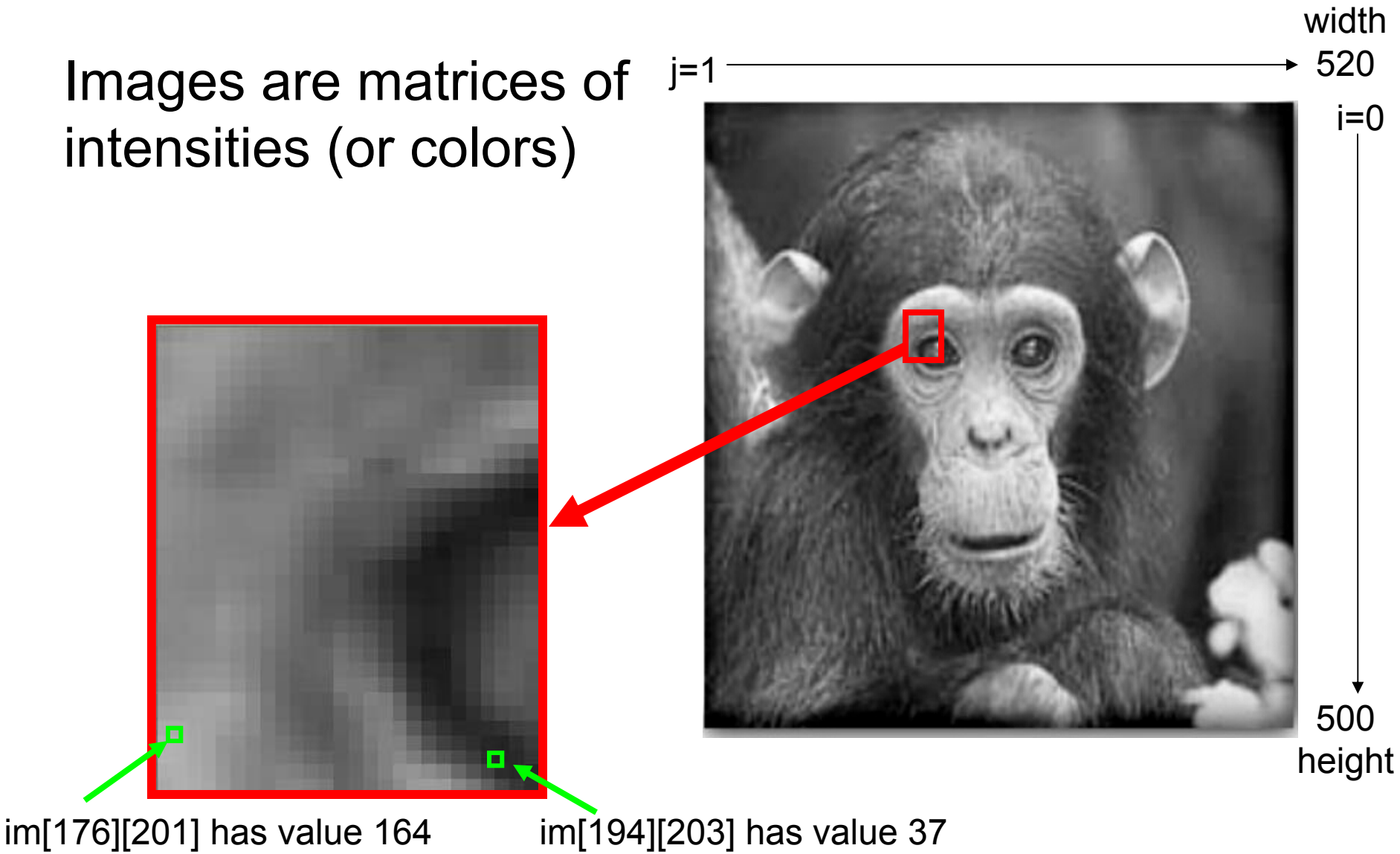
Digital images

Images are matrices of intensities (or colors)

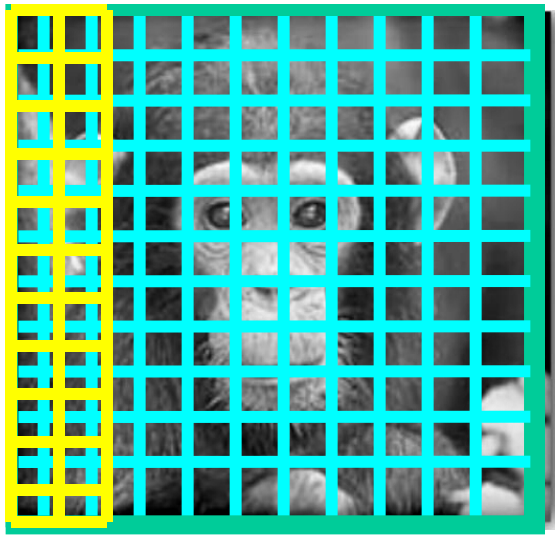


Digital images

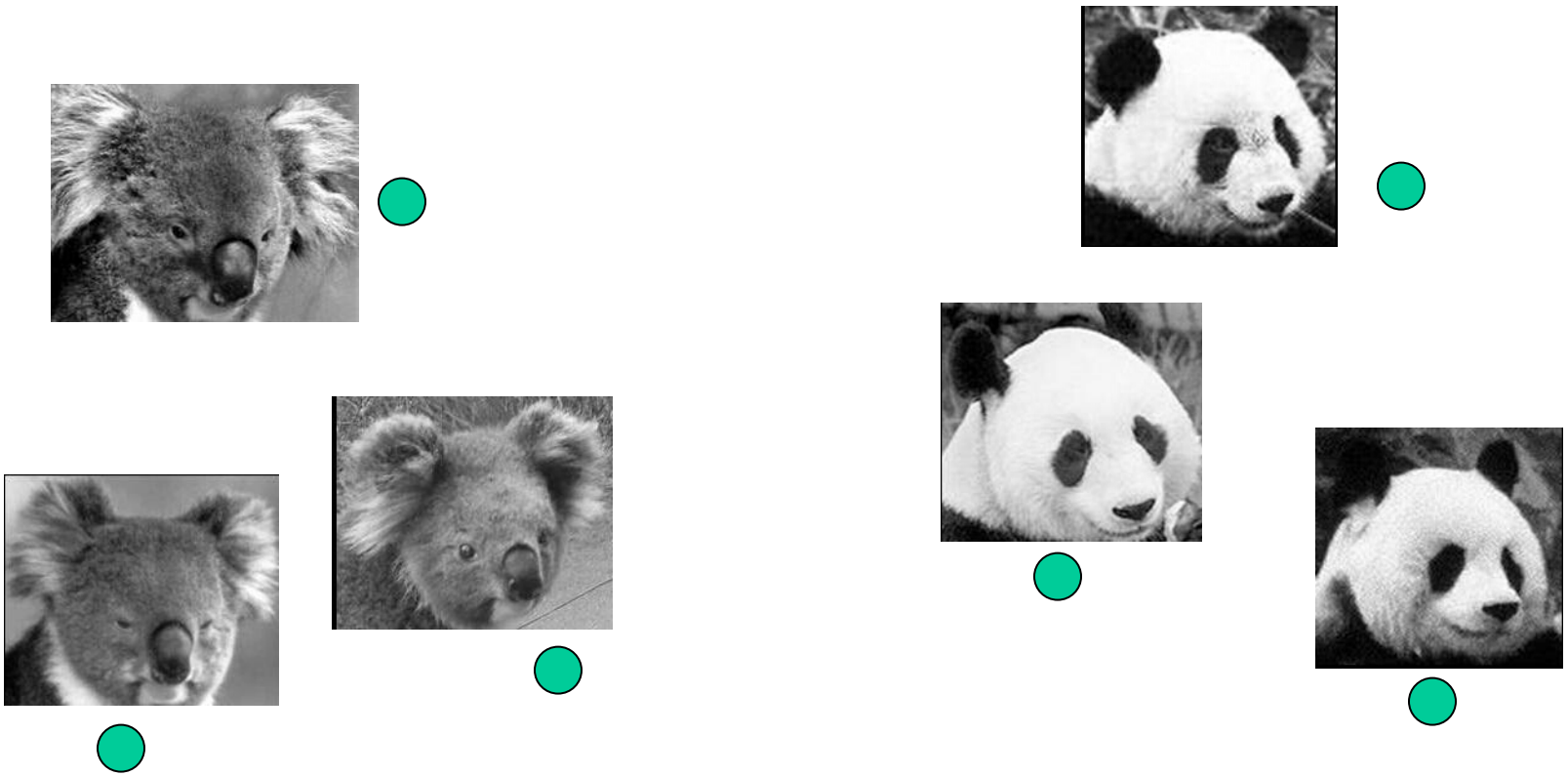
Images are matrices of intensities (or colors)



Images as arrays, or points in “feature space”



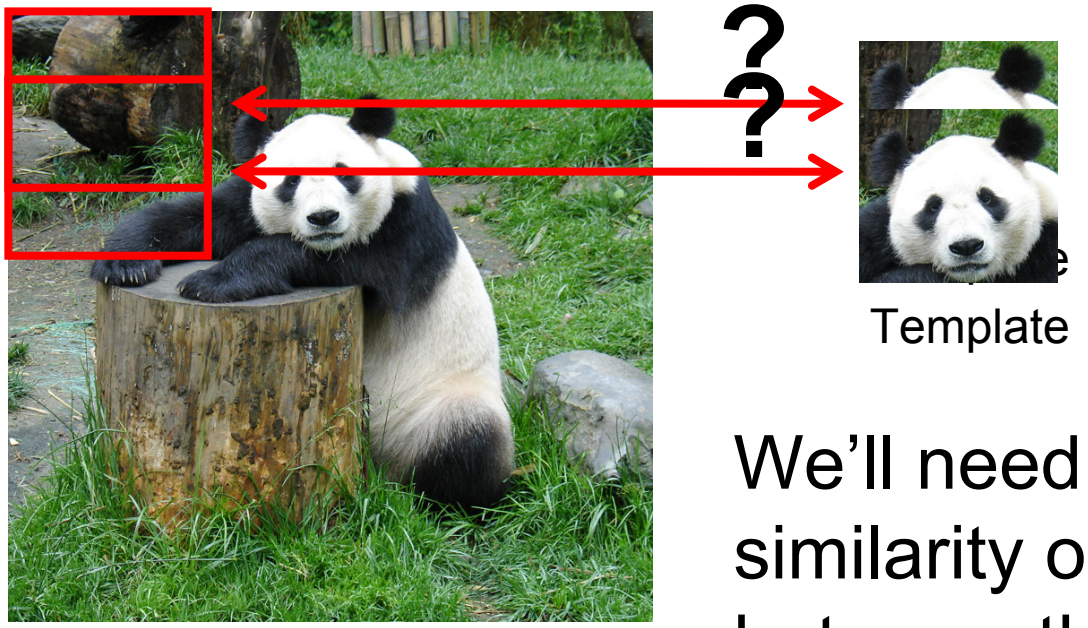
Images as arrays, or points in “feature space”



d = image height * image width

Template matching

- Goal: search all image windows in a scene looking for occurrences of a template

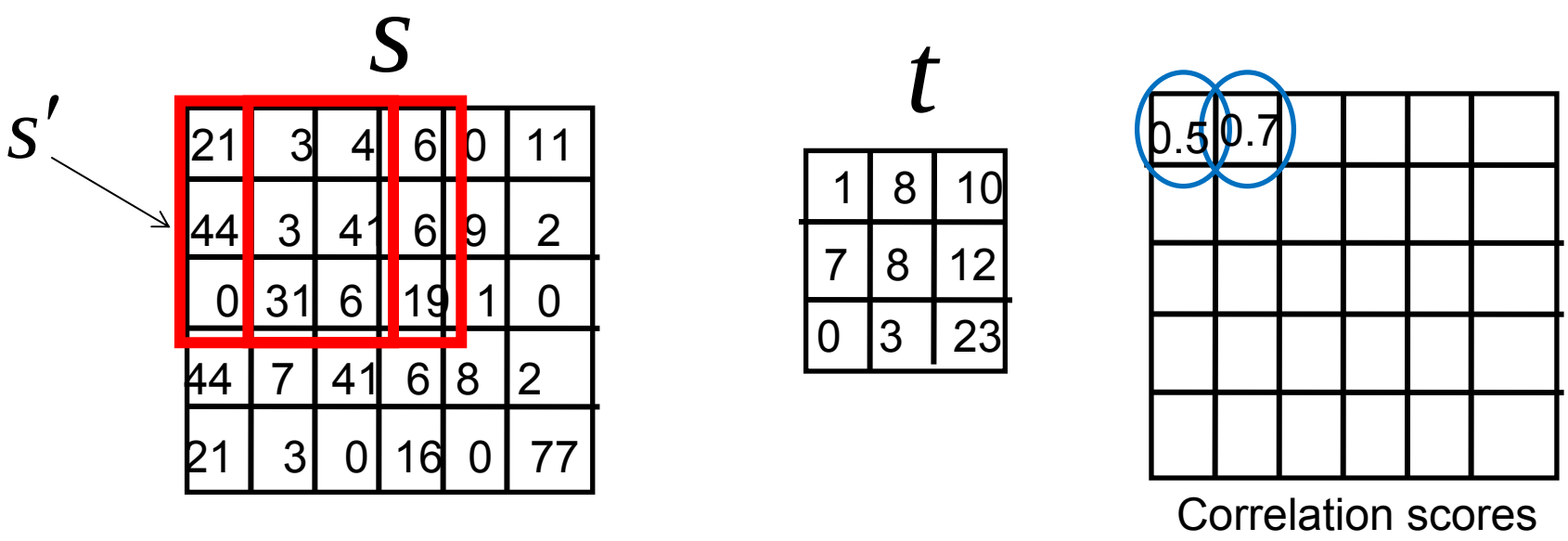


Scene

Template

We'll need to evaluate the similarity or distance between the template and every part of the scene.

Template matching: normalized cross-correlation



$$\gamma = \frac{\sum_{x,y}\left(s(x,y)-\bar{s}'\right)\left(t(x,y)-\bar{t}\right)}{\sigma_{s'}\sigma_t}$$

Template matching

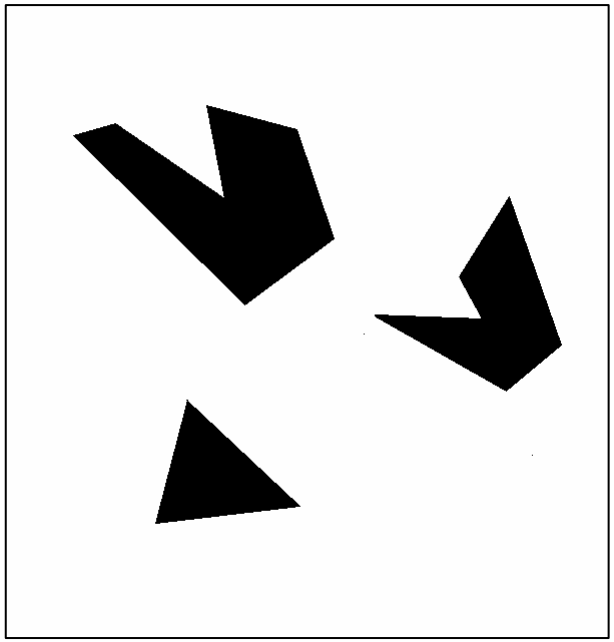
For each subwindow position within scene *s*:

- Compute normalized correlation score between current portion of *s* and template *t*
- Record score in output correlation matrix that's indexed by position of the subwindows

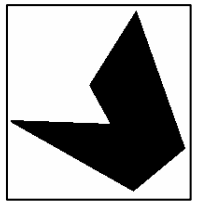
Find maximum value(s) in output correlation matrix

Return position of maximal value(s) as the best template detection

Template matching



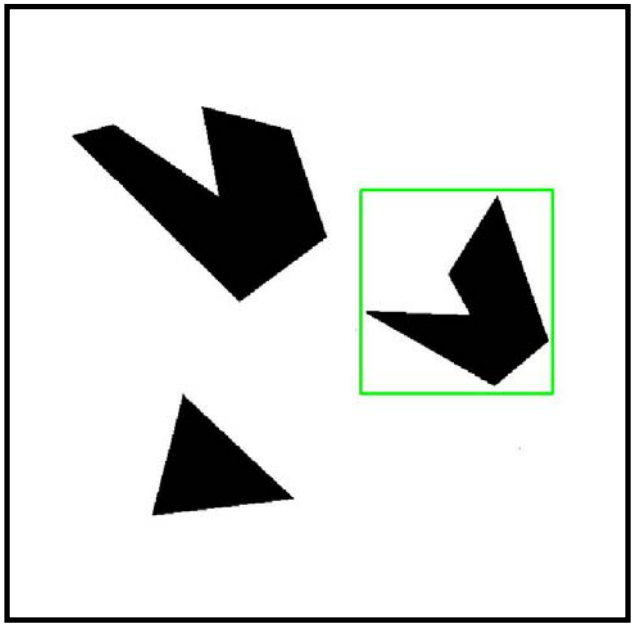
Scene



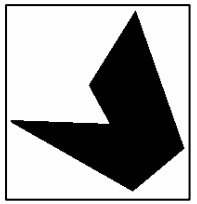
Template

A toy example

Template matching

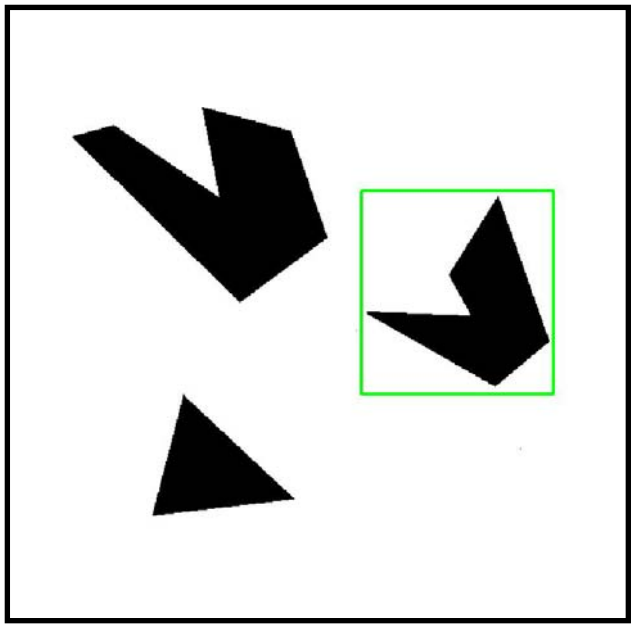


Detected template

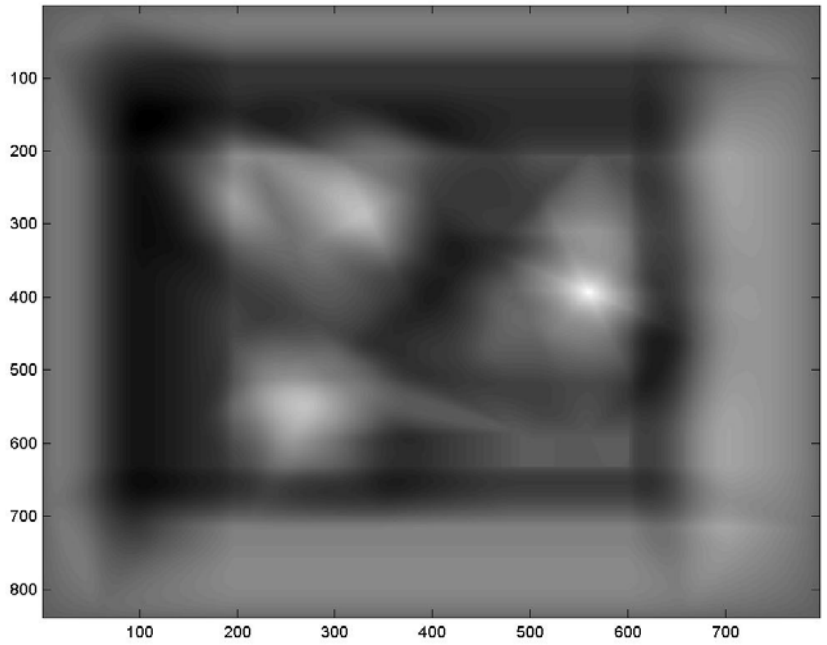


Template

Template matching

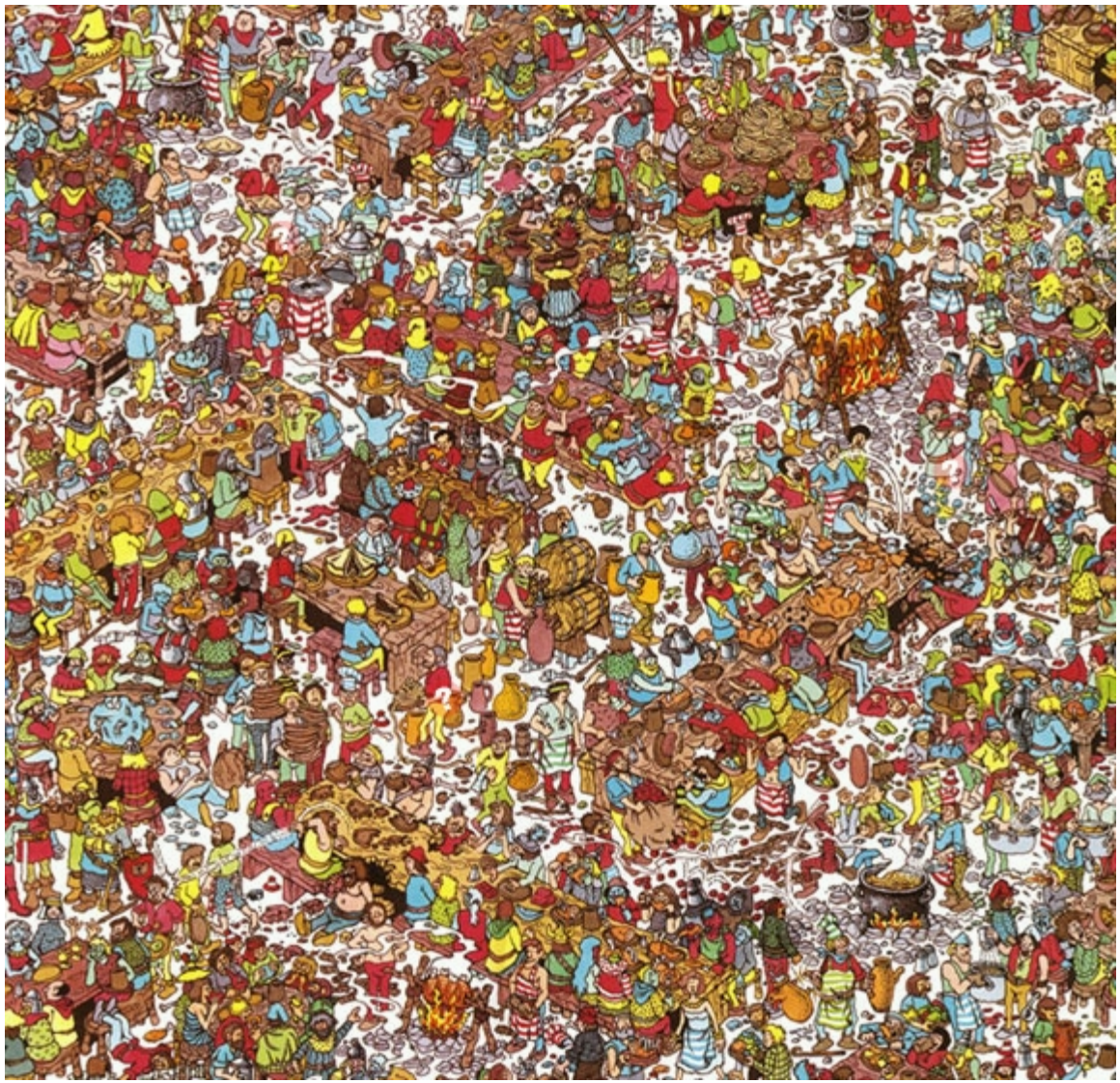


Detected template



Correlation map

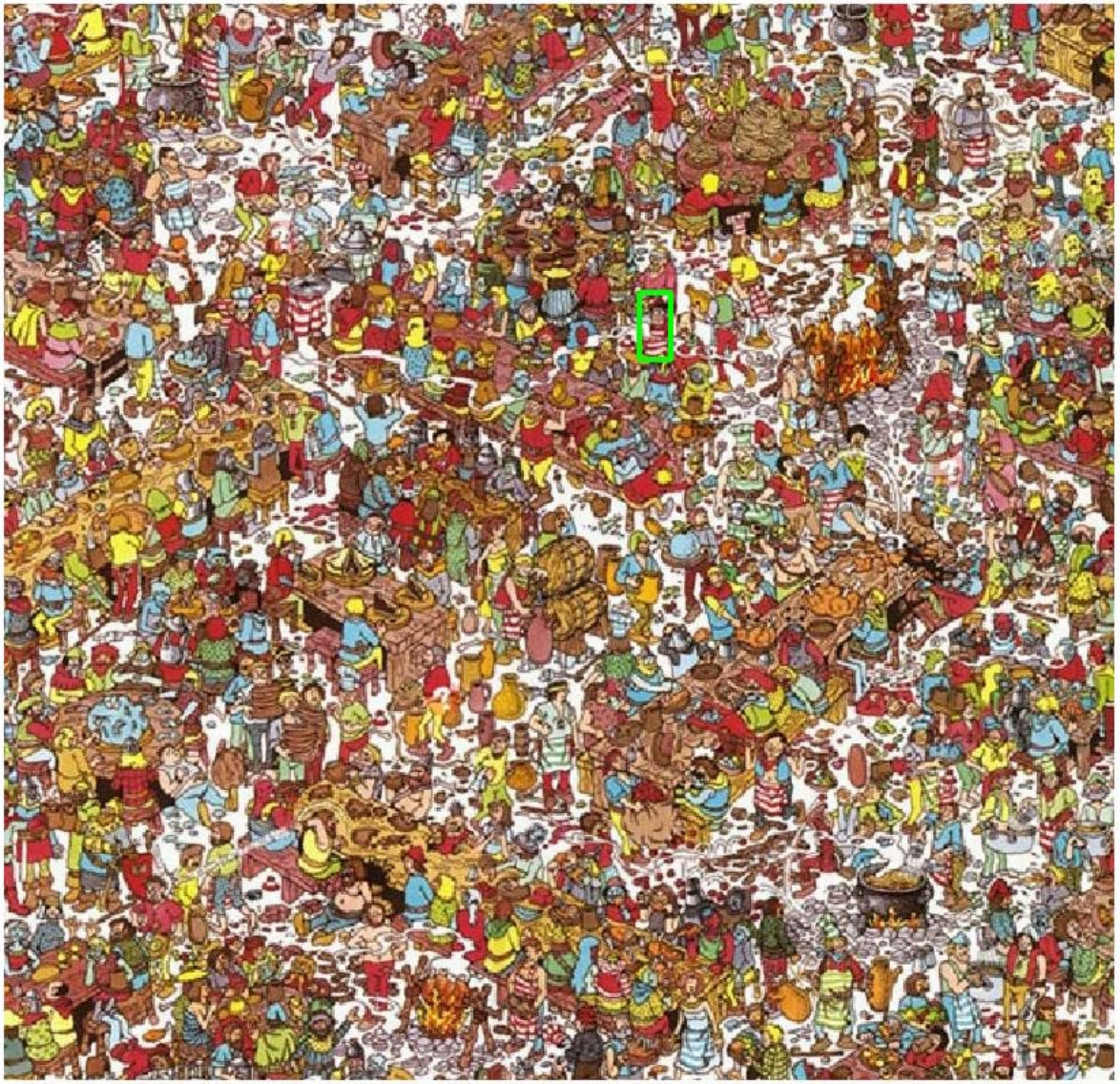
Where's Waldo?



Scene

Template

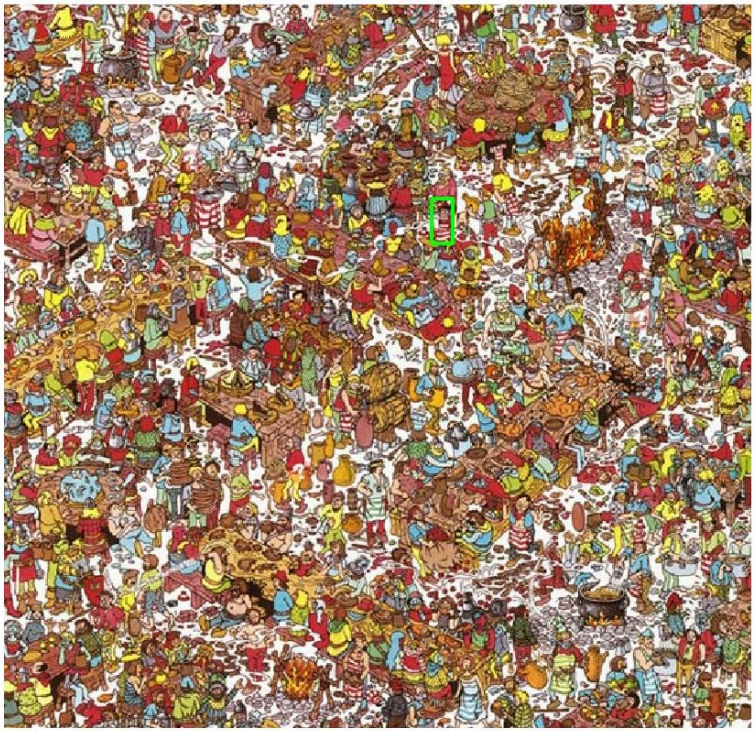
Where's Waldo?



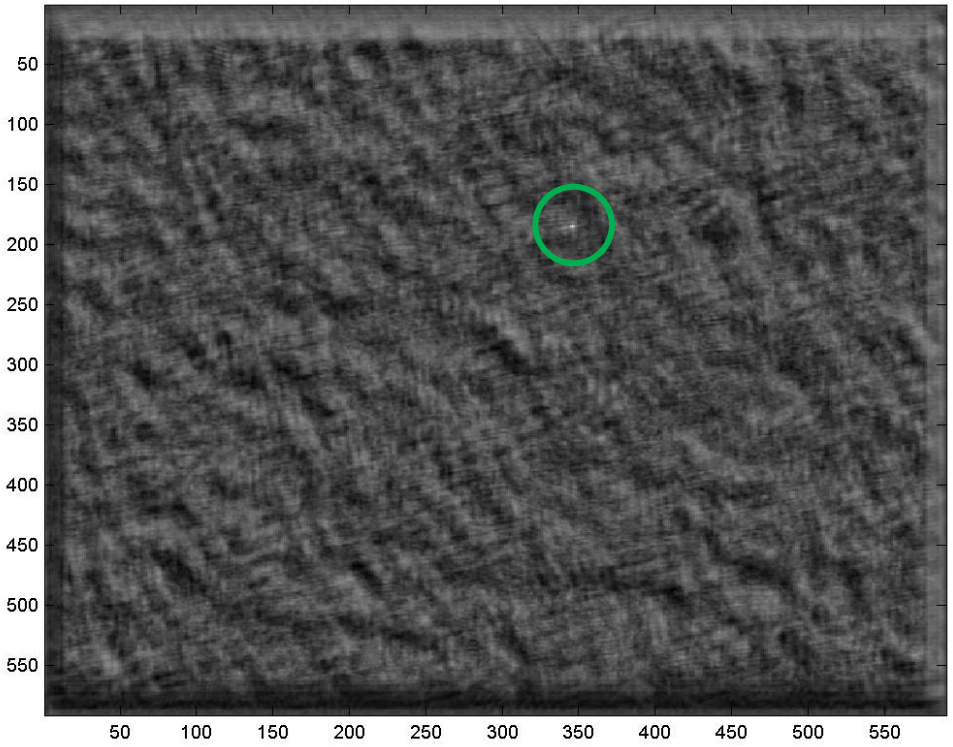
Detected template

Template

Where’s Waldo?



Detected template



Correlation map

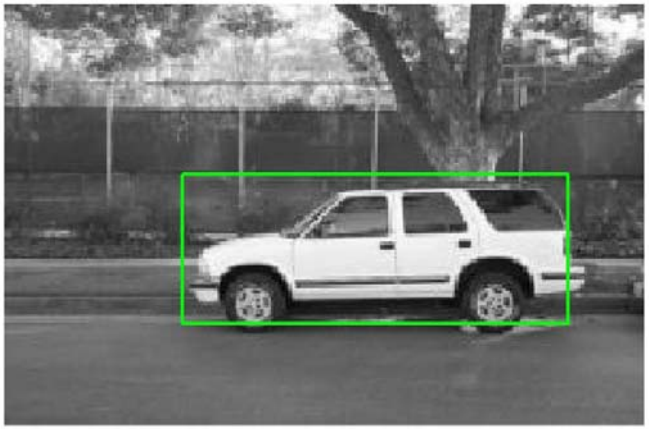
Template matching

Scene

Template

What if the template is not identical to some subimage in the scene?

Template matching



Detected template

Template

Match can still be meaningful, if scale and general appearance is right.

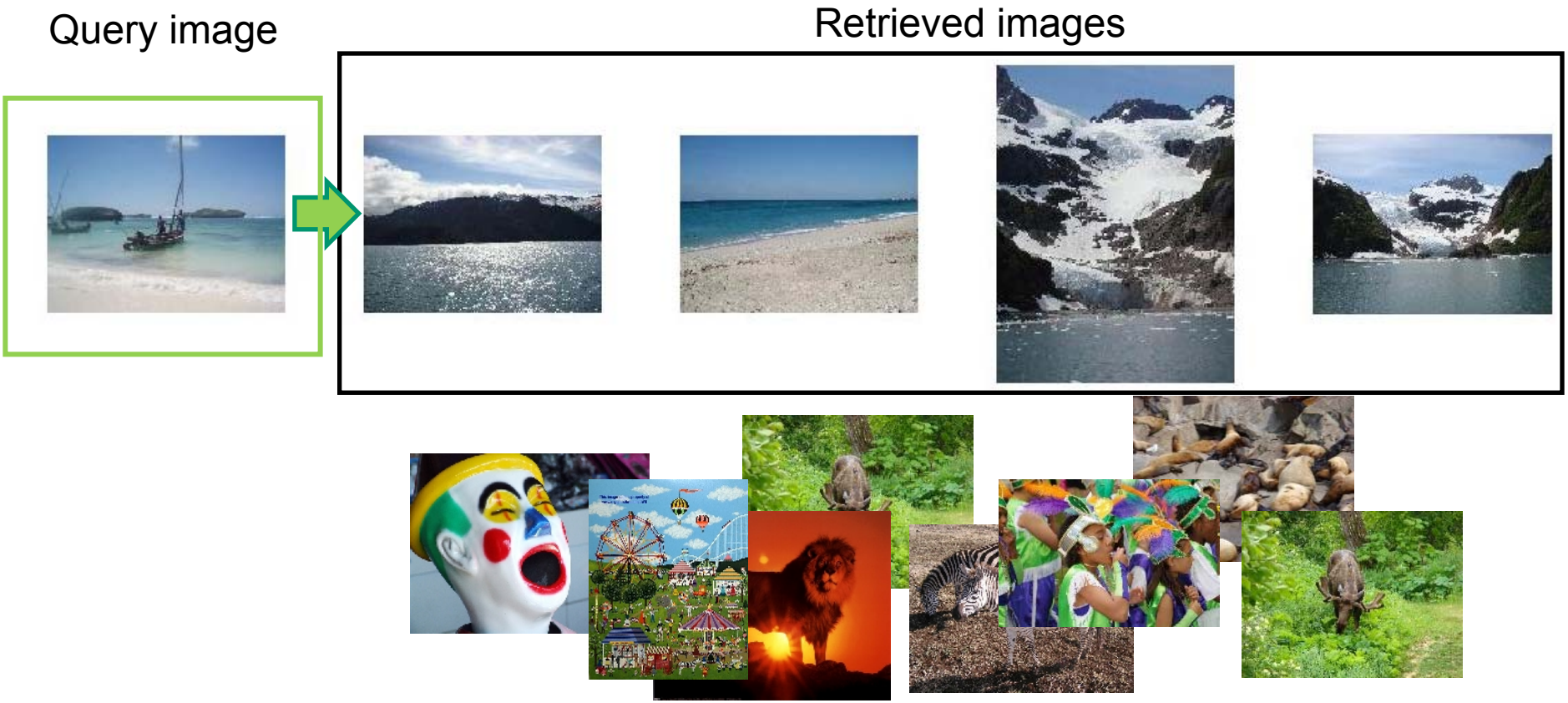
Extensions

- Take a video as the scene input, and *track* the template over time by matching it at each frame.
  - Connect the tracker with Windows' mouse input to build a "camera mouse" video interface.
- Search at multiple scales.

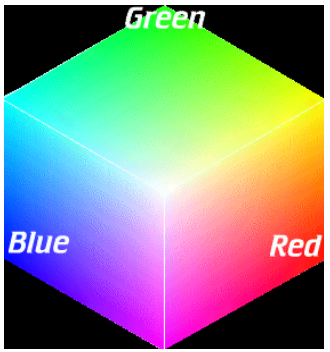
User in the Boston College Camera Mouse project

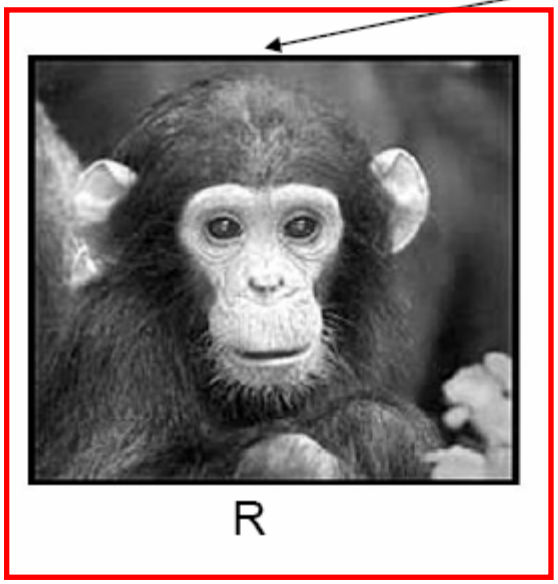
Color-based image retrieval

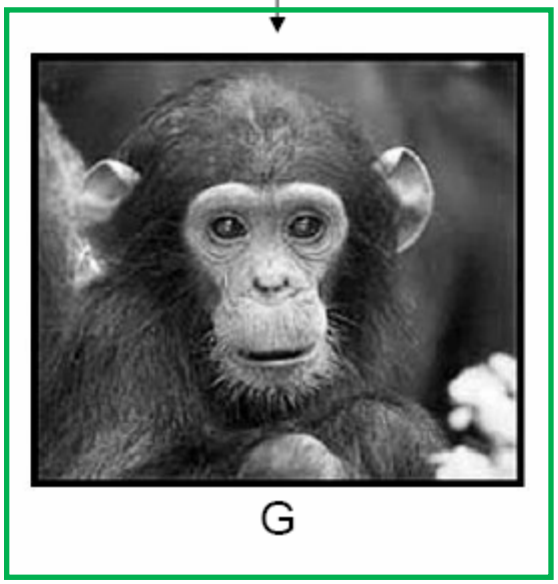
- Goal: give a query image, find images that have similar color distributions.

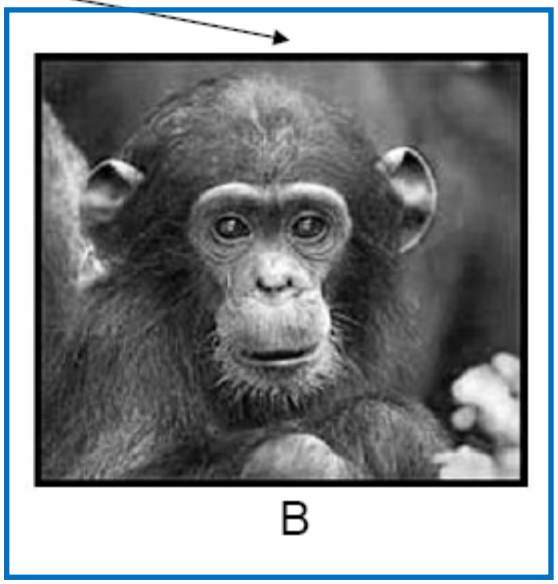


Each pixel is a combination of three primary color channels.
(RGB color space =Red,Green,Blue)

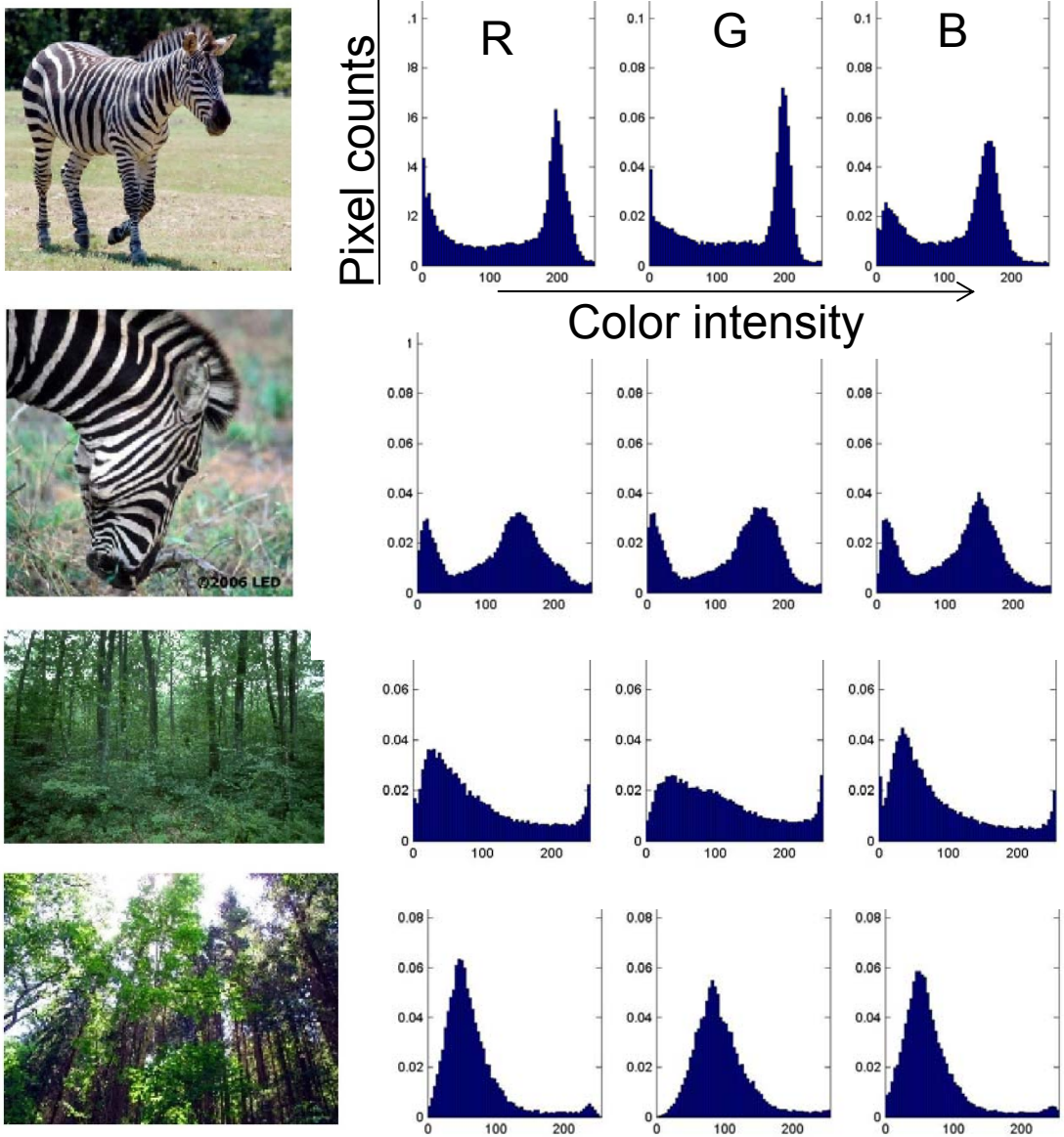






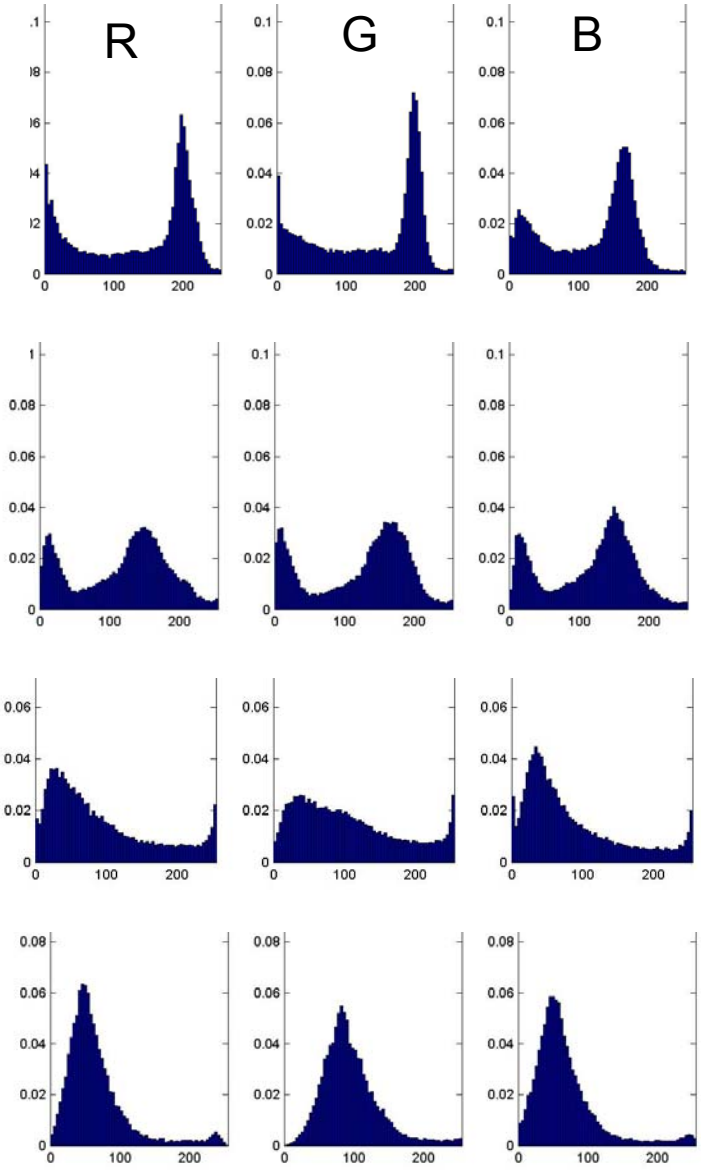


Color histograms



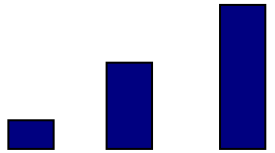
- Use distribution of colors to describe image
- No spatial info – invariant to translation, rotation, scale

Histogram intersection



Given two histogram vectors, sum the minimum counts per bin:

$$I(x,y) = \sum_{i=1}^{n} \min(x_i, y_i)$$



x = [1, 3, 5]

y = [2, 0, 3]

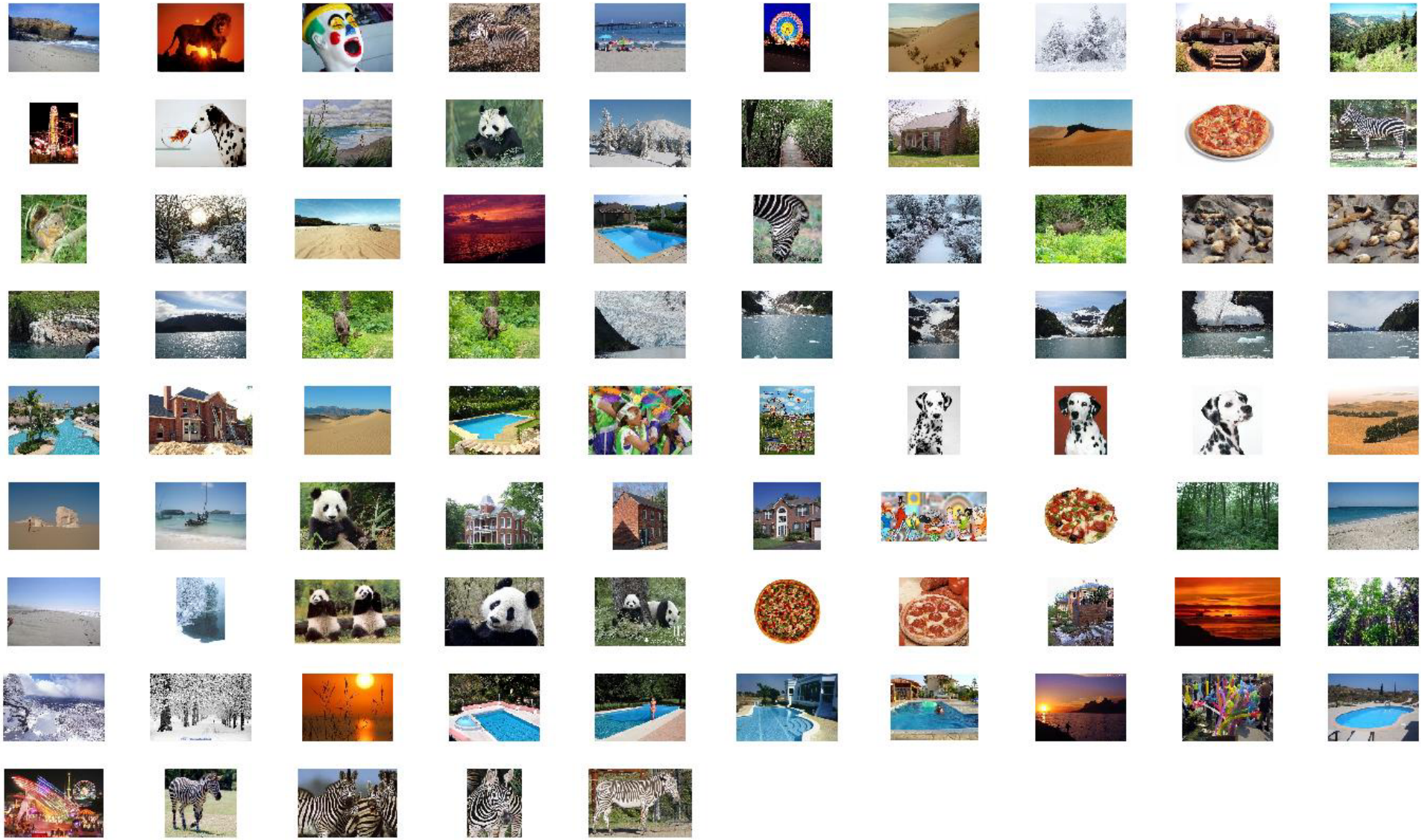
[1, 0, 3]

$$\sum_i \min(x_i, y_i) = 4$$

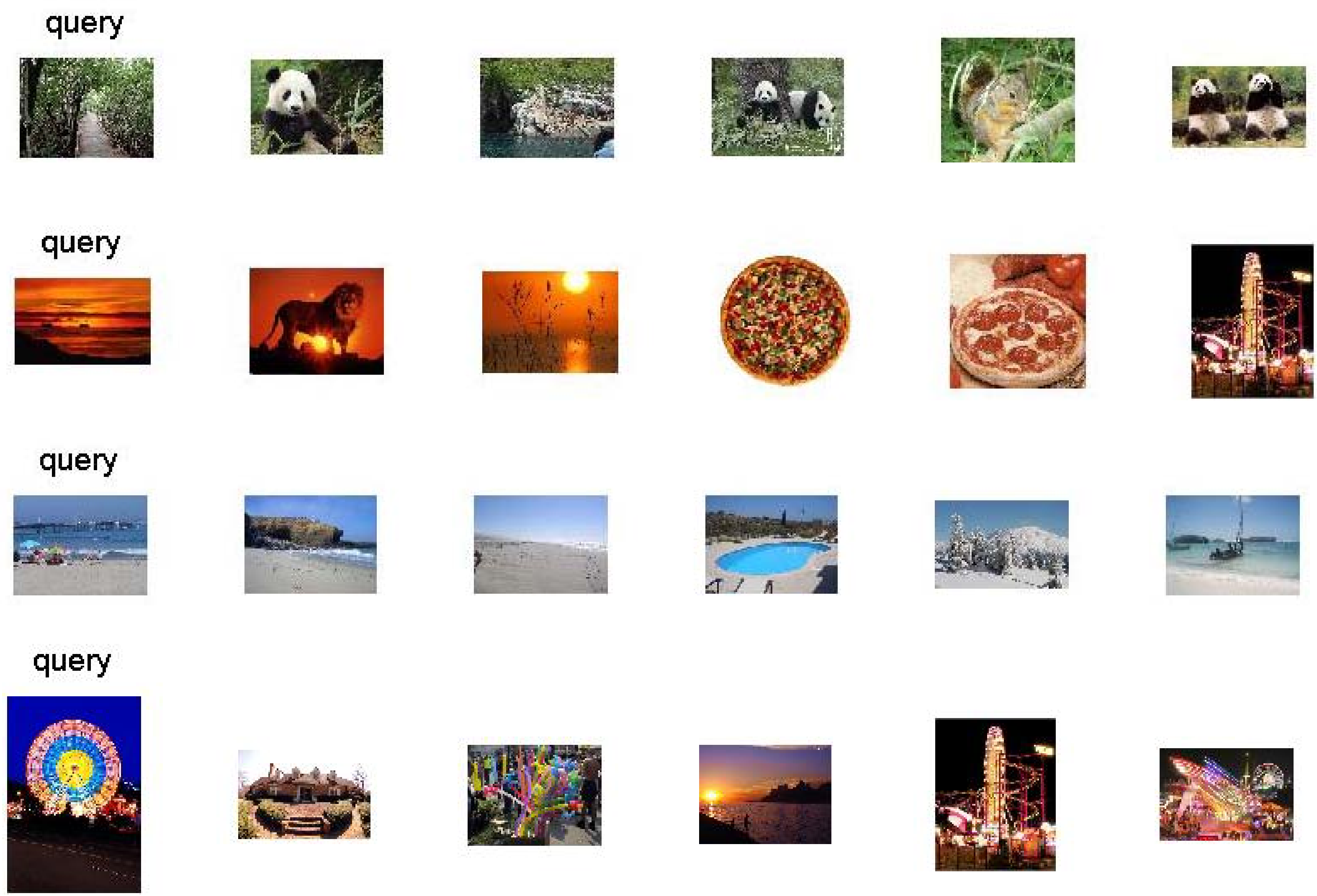
Color-based image retrieval

- Given collection (database) of images:
  - Extract and store one color histogram per image (concatenate R, G, B histogram counts)
- Given new query image:
  - Extract its color histogram
  - For each database image:
    - Compute intersection between query histogram and database histogram
  - Sort intersection values (highest score = most similar)
  - Rank database items relative to query based on this sorted order

Example database



Example retrievals



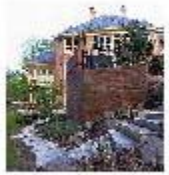
Example retrievals

query






query



query






Extensions

- Explore alternate color spaces (e.g., HSV, Lab)
- Cluster all the images based on their histograms
- Match for skin color to detect faces or hands

Figure from M. Jones and J. Rehg, Statistical Color Models with Application to Skin Detection, IJCV 2002.

Background subtraction

"Background" image

New image

Differences

Thresholded differences

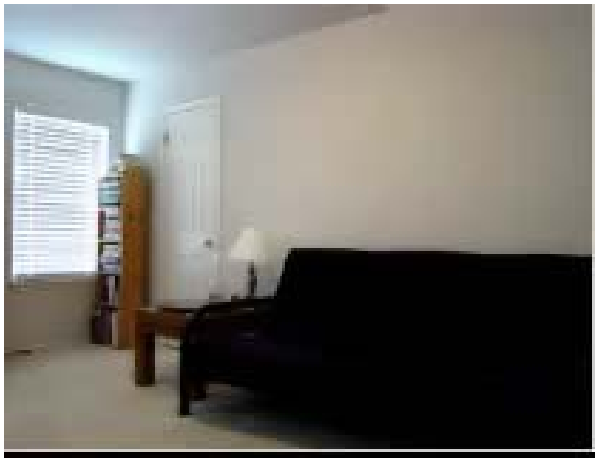
- Use absolute value or squared difference to ignore sign of change.
- Eliminate some noise by only looking at "large" changes.

Example: background subtraction

Original video

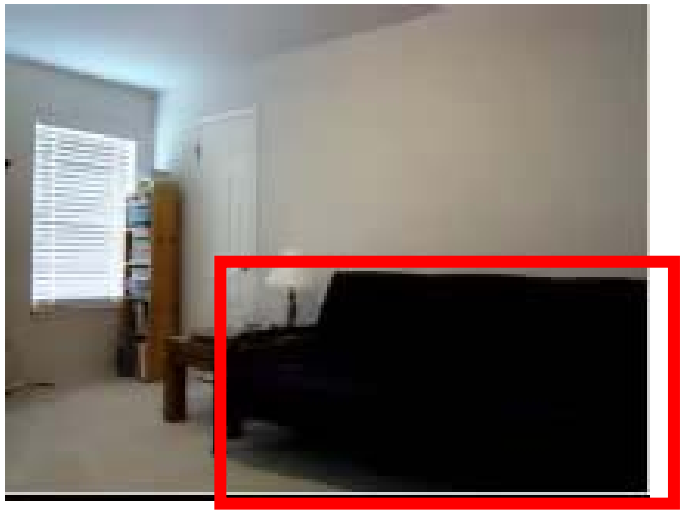
Differences

Output



Example: background subtraction

Original video



Output

Extensions

- Assert regions of interest
- Build more robust background model via average/median over multiple empty frames
- Adaptive background model for changing environments
- Extract “connected components”, and evaluate their area, shape, position, etc.
- Summarize a long video based on the activity (motion) detected

Example of connected components

Note

- Simple techniques can allow students to try some fun things---easy entry, but also possibilities for more advanced level.
- Caveat: Even for simple tasks, *perfect* solutions are hard!
  - Can require some tweaking and good choice of image data

Tools

- The exercises presented here can be done with C/Java with basic image IO functions (see handout).
- For more advanced exercises and building applications, there are a number of existing resources:
  - Open CV library from Intel (C++)
  - Java computer vision commons library
  - Matlab: image processing toolbox