CS598:Visual information Retrieval

Lecture IV: Image Representation:
Semantic and Attribute based Representation

Rcapt of Lecture IV

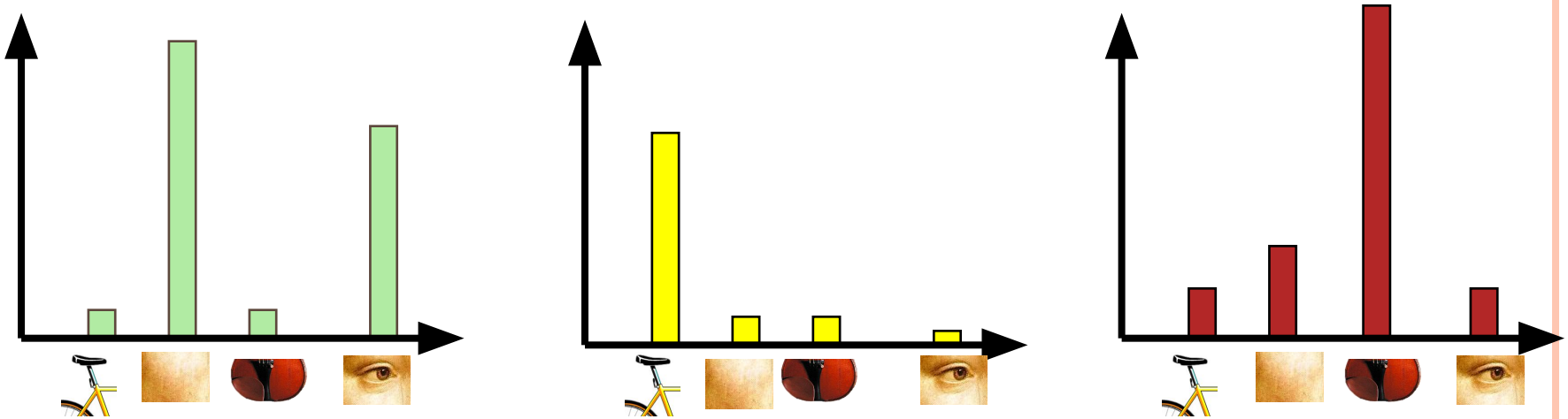
- Histogram of local features
- Bag of words model
- Soft quantization and sparse coding
- Supervector with Gaussian mixture model

Lecture V: Part I

Image classification basics

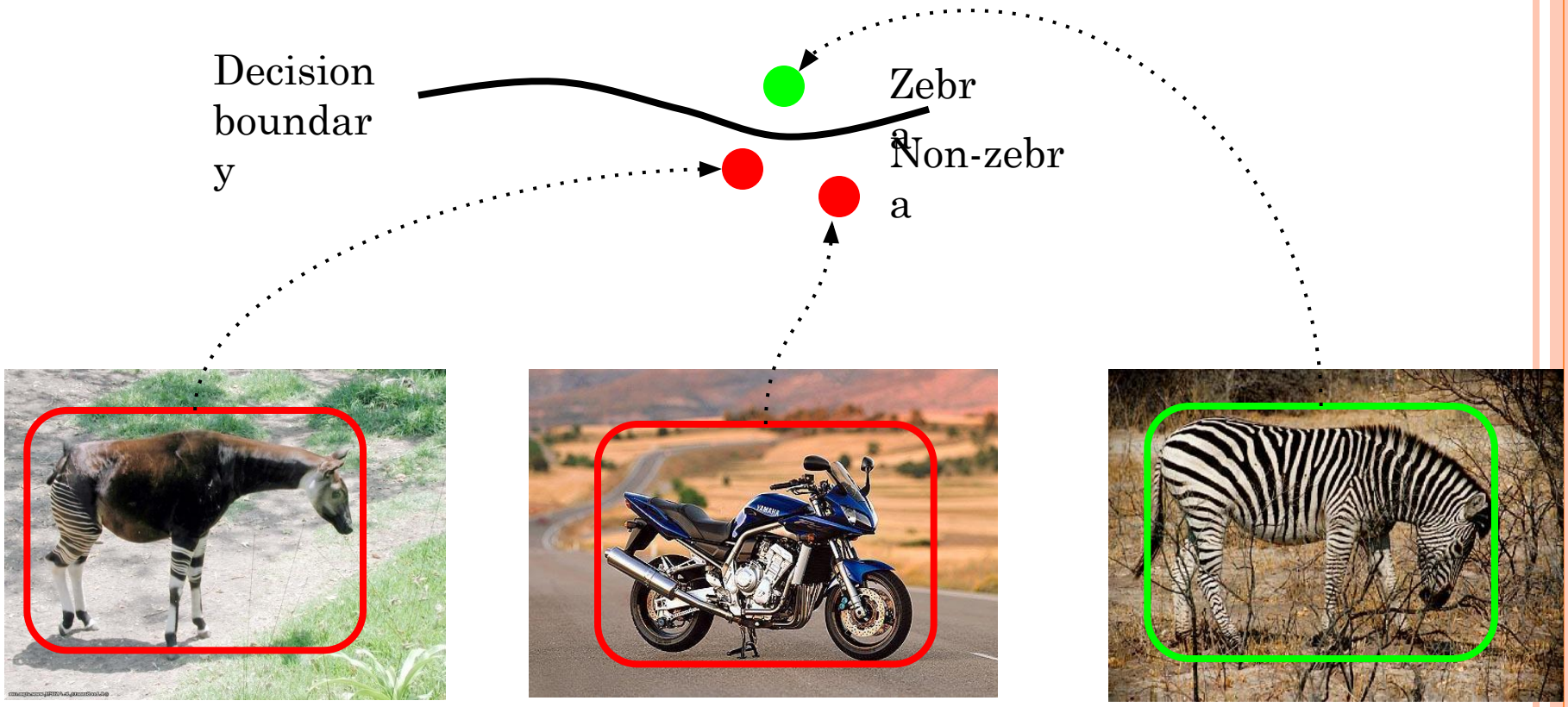
Image classification

- Given the image representation, e.g., color, texture, shape, local descriptors, of images from different classes, how do we learn a model for distinguishing them?



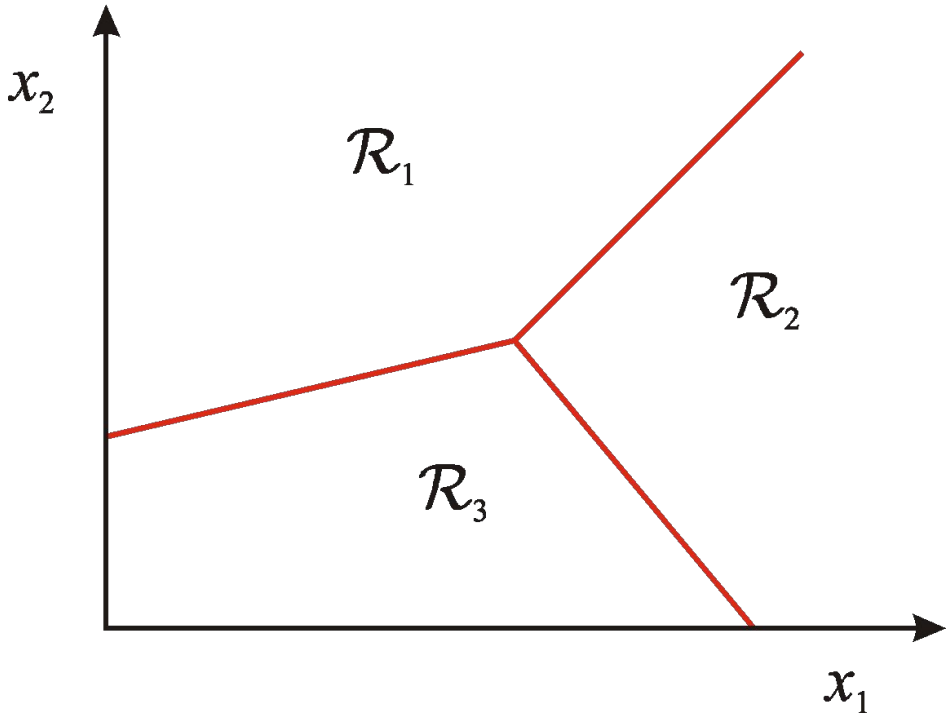
Classifiers

- Learn a decision rule assigning bag-of-features representations of images to different classes



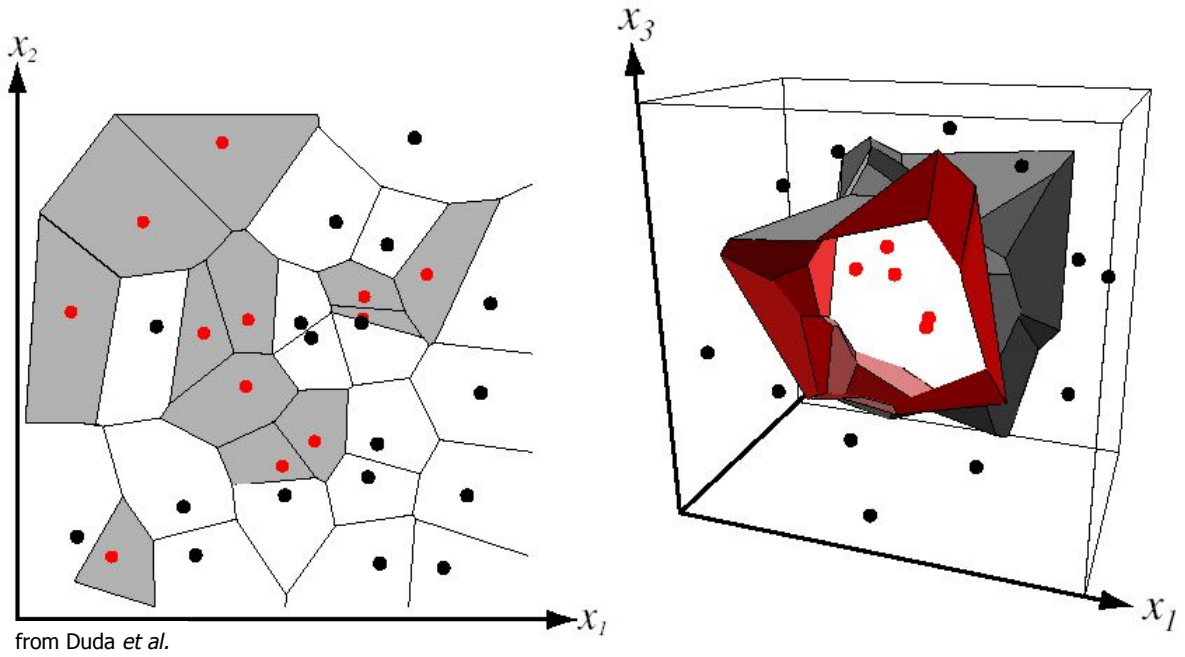
Classification

- Assign input vector to one of two or more classes
- Any decision rule divides input space into *decision regions* separated by *decision boundaries*



Nearest Neighbor Classifier

- Assign label of nearest training data point to each test data point



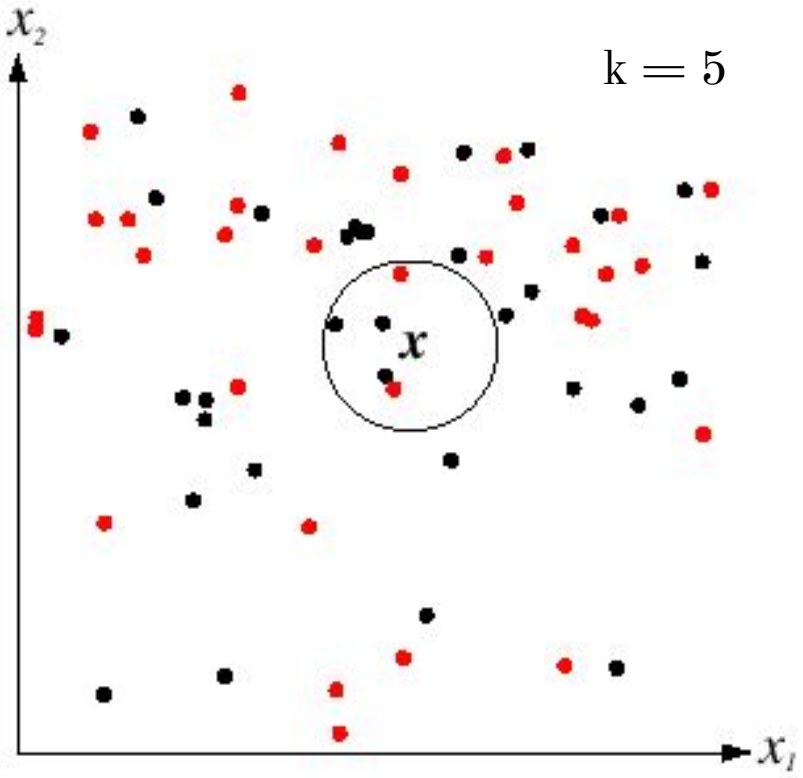
from Duda *et al.*

Voronoi partitioning of feature space for two-category 2D and 3D data

Source: D. Lowe

K-Nearest Neighbors

- For a new point, find the k closest points from training data
- Labels of the k points “vote” to classify
- Works well provided there is lots of data and the distance function is good



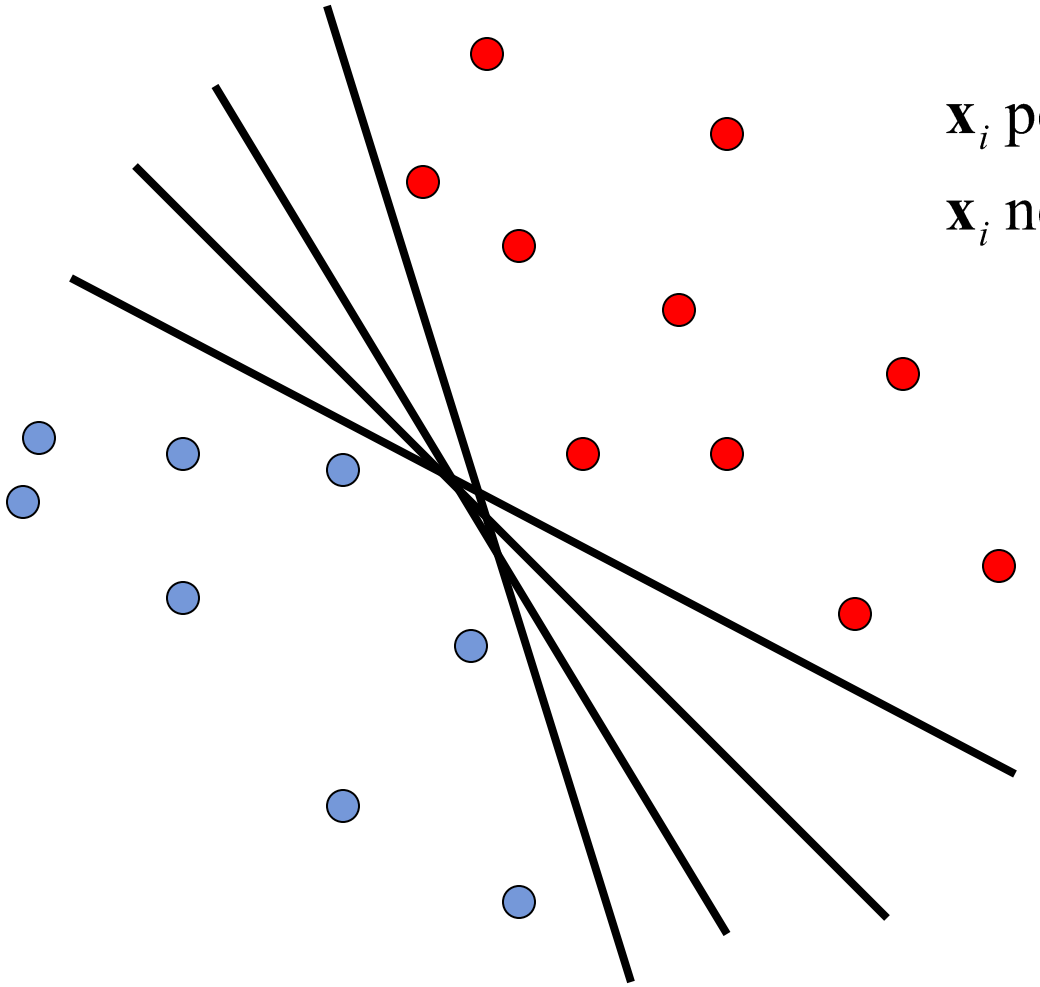
Source: D. Lowe

Functions for comparing histograms

- L1 distance: $$D(h_1, h_2) = \sum_{i=1}^{N} | h_1(i) - h_2(i) |$$
- χ^2 distance: $$D(h_1, h_2) = \sum_{i=1}^{N} \frac{\left(h_1(i) - h_2(i)\right)^2}{h_1(i) + h_2(i)}$$
- Quadratic distance (*cross-bin distance*): $$D(h_1, h_2) = \sum_{i,j} A_{ij} \left(h_1(i) - h_2(j)\right)^2$$
- Histogram intersection (similarity function): $$I(h_1, h_2) = \sum_{i=1}^{N} \min(h_1(i), h_2(i))$$

Linear classifiers

- Find linear function (*hyperplane*) to separate positive and negative examples



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

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

Which hyperplane is best?

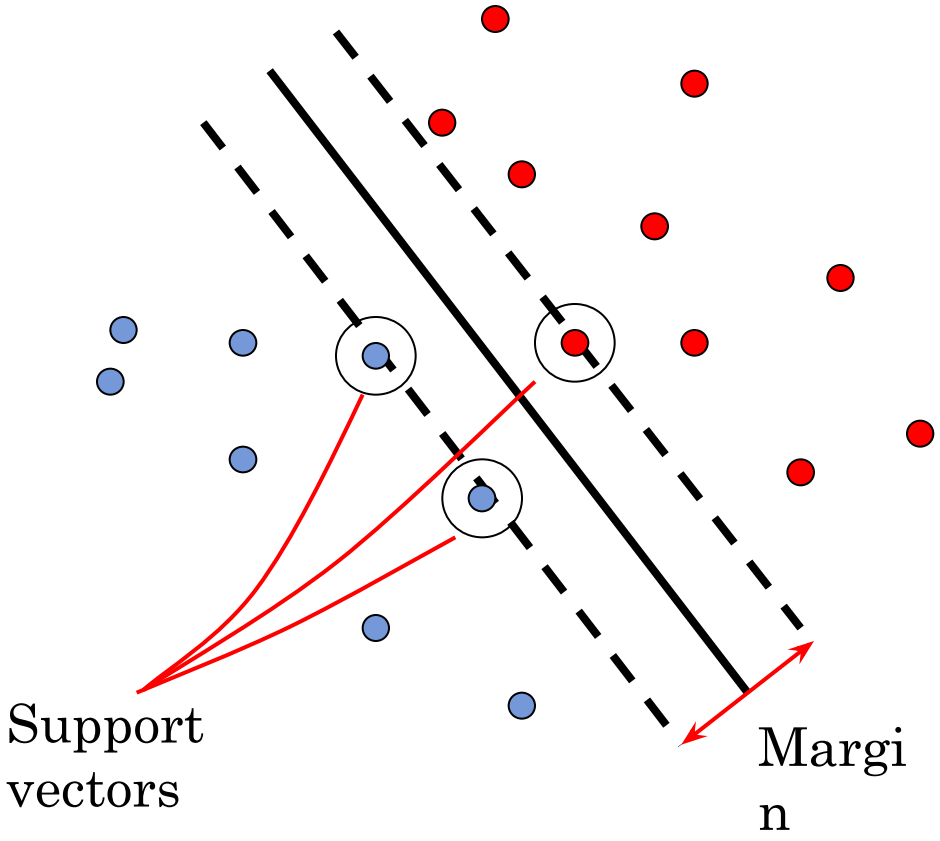
Support vector machines

- Find hyperplane that maximizes the *margin* between the positive and negative examples

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

Support vector machines

- Find hyperplane that maximizes the *margin* between the positive and negative examples



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

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

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

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

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

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

Finding the maximum margin hyperplane

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

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

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

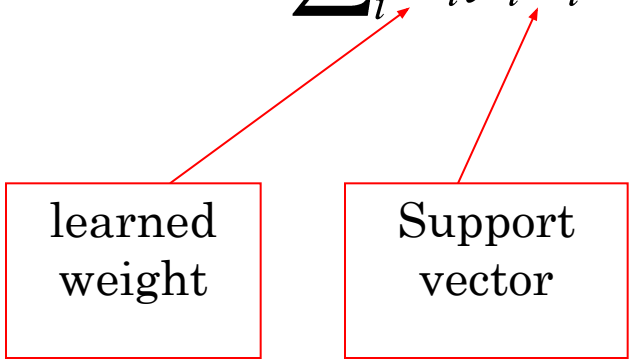
- *Quadratic optimization problem*:

- $$\text{Minimize} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w}$$
  $$\text{Subject to} \quad y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1$$

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

Finding the maximum margin hyperplane

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



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

Finding the maximum margin hyperplane

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

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

- Classification function (decision boundary):

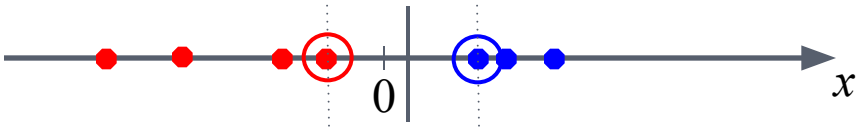
$$\mathbf{w} \cdot \mathbf{x} + b = \sum_i \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b$$

- Notice that it relies on an *inner product* between the test point $\boldsymbol{x}$ and the support vectors $\boldsymbol{x}_i$
- Solving the optimization problem also involves computing the inner products $\boldsymbol{x}_i \cdot \boldsymbol{x}_j$ between all pairs of training points

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

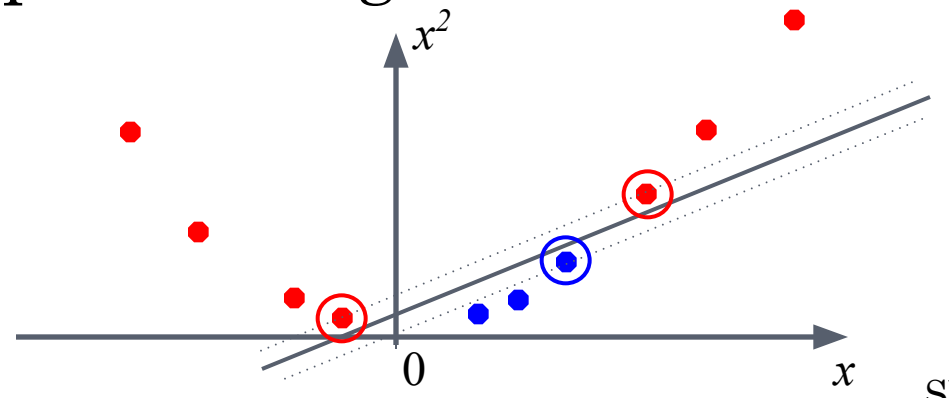
Nonlinear SVMs

- Datasets that are linearly separable work out great:



- But what if the dataset is just too hard?

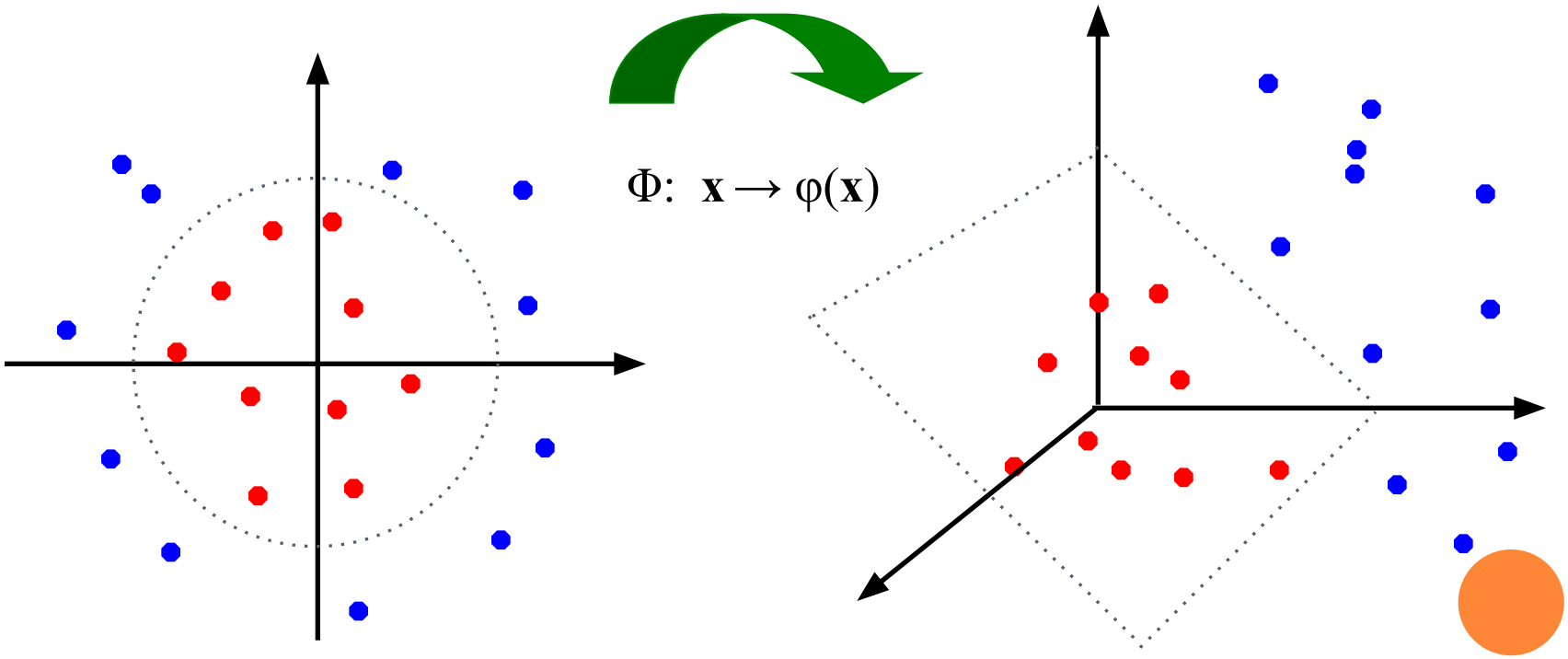
- We can map it to a higher-dimensional space:



Slide credit: Andrew Moore

Nonlinear SVMs

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



Slide credit: Andrew Moore

Nonlinear SVMs

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

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

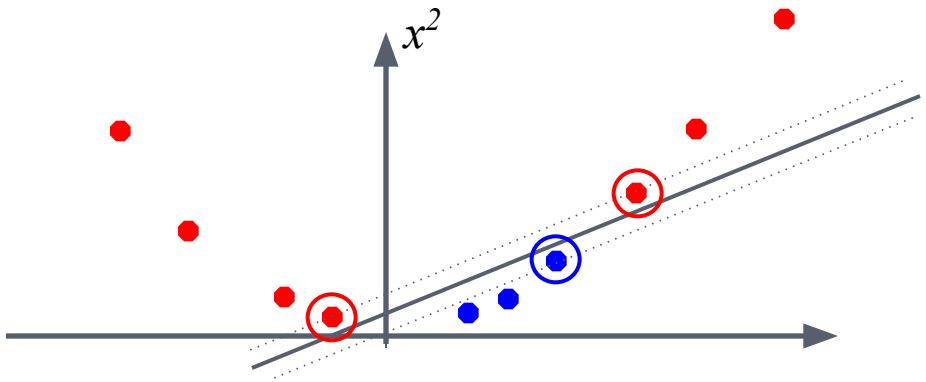
- (to be valid, the kernel function must satisfy *Mercer's condition*)
- This gives a nonlinear decision boundary in the original feature space:

$$\sum_i \alpha_i y_i \varphi(\boldsymbol{x}_i) \cdot \varphi(\boldsymbol{x}) + b = \sum_i \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b$$

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

Nonlinear kernel: Example

- Consider the mapping $\varphi(x) = (x, x^2)$



$$\varphi(x) \cdot \varphi(y) = (x, x^2) \cdot (y, y^2) = xy + x^2 y^2$$

$$K(x, y) = xy + x^2 y^2$$

Kernels for bags of features

- Histogram intersection kernel:

$$I(h_1, h_2) = \sum_{i=1}^{N} \min(h_1(i), h_2(i))$$

- Generalized Gaussian kernel:

$$K(h_1, h_2) = \exp\left(-\frac{1}{A} D(h_1, h_2)^2 \right)$$

- D can be L1 distance, Euclidean distance, χ^2 distance, etc.

J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, Local Features and Kernels for Classifcation of Texture and Object Categories: A Comprehensive Study, IJCV 2007

Summary: SVMs for image classification

1. Pick an image representation (in our case, bag of features)
2. Pick a kernel function for that representation
3. Compute the matrix of kernel values between every pair of training examples
4. Feed the kernel matrix into your favorite SVM solver to obtain support vectors and weights
5. At test time: compute kernel values for your test example and each support vector, and combine them with the learned weights to get the value of the decision function

What about multi-class SVMs?

- Unfortunately, there is no "definitive" multi-class SVM formulation
- In practice, we have to obtain a multi-class SVM by combining multiple two-class SVMs
- One vs. others
  - Traning: learn an SVM for each class vs. the others
  - Testing: apply each SVM to test example and assign to it the class of the SVM that returns the highest decision value
- One vs. one
  - Training: learn an SVM for each pair of classes
  - Testing: each learned SVM "votes" for a class to assign to the test example

SVMs: Pros and cons

- Pros
  - Many publicly available SVM packages: http://www.kernel-machines.org/software
  - Kernel-based framework is very powerful, flexible
  - SVMs work very well in practice, even with very small training sample sizes

- Cons
  - No "direct" multi-class SVM, must combine two-class SVMs
  - Computation, memory
    - During training time, must compute matrix of kernel values for every pair of examples
    - Learning can take a very long time for large-scale problems

Summary: Classifiers

- Nearest-neighbor and k-nearest-neighbor classifiers
  - L1 distance, χ^2 distance, quadratic distance, histogram intersection
- Support vector machines
  - Linear classifiers
  - Margin maximization
  - The kernel trick
  - Kernel functions: histogram intersection, generalized Gaussian, pyramid match
  - Multi-class
- Of course, there are many other classifiers out there
  - Neural networks, boosting, decision trees, …

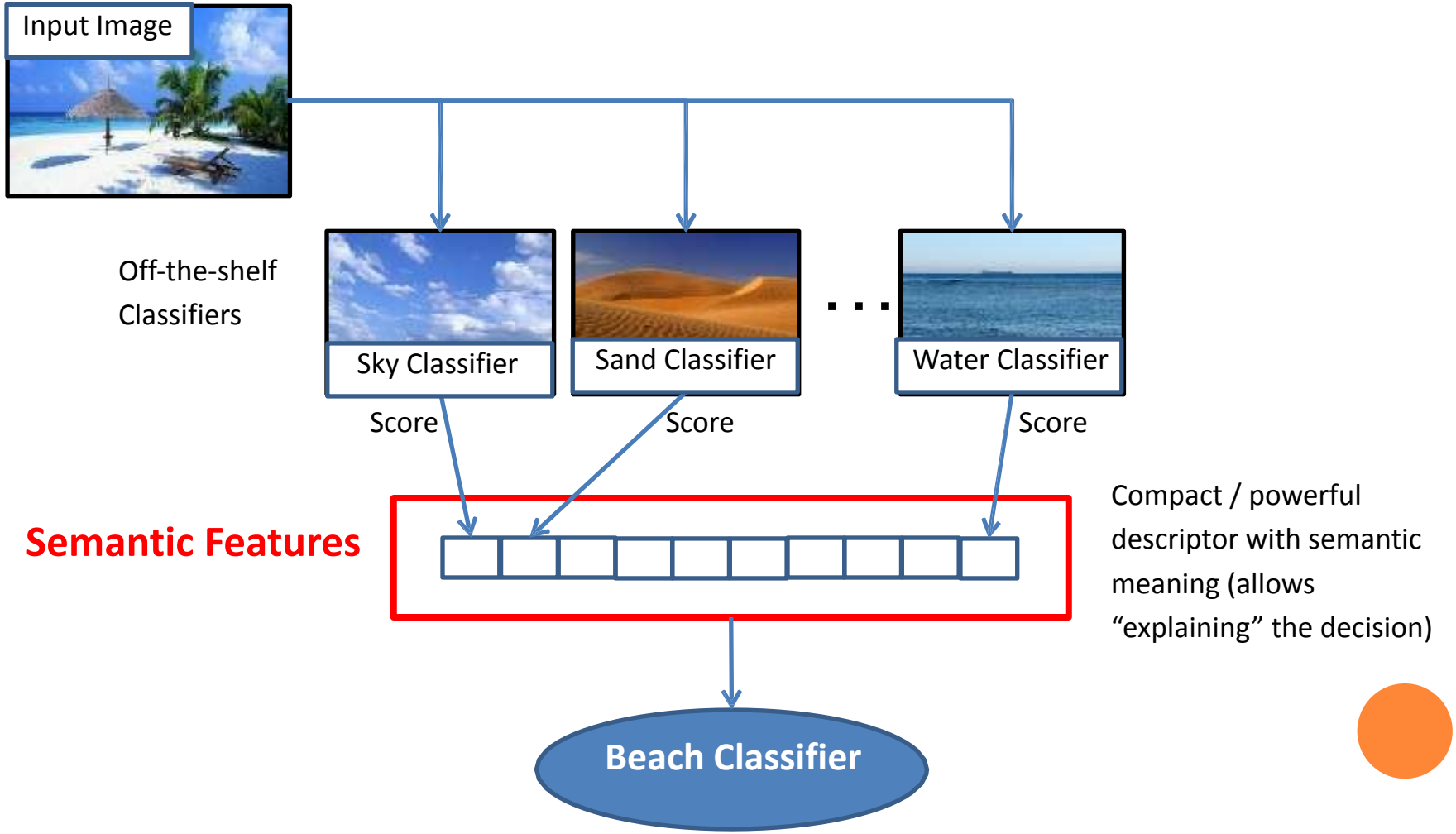
Lecture IV: Part II

Semantic and Attribute Features

Slides adapted from Feris.

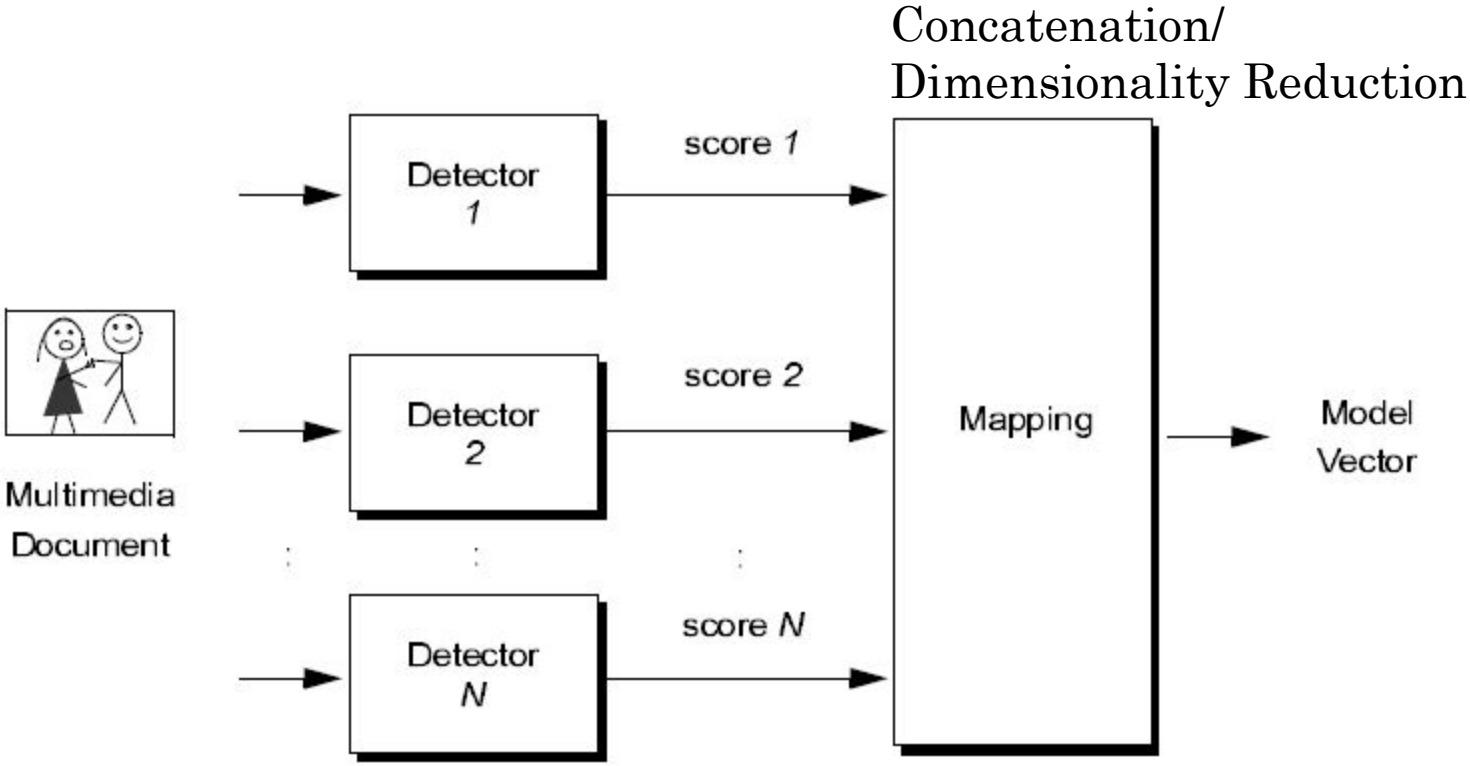
Semantic Features

- Use the scores of semantic classifiers as high-level features



Frame level semantic features (1)

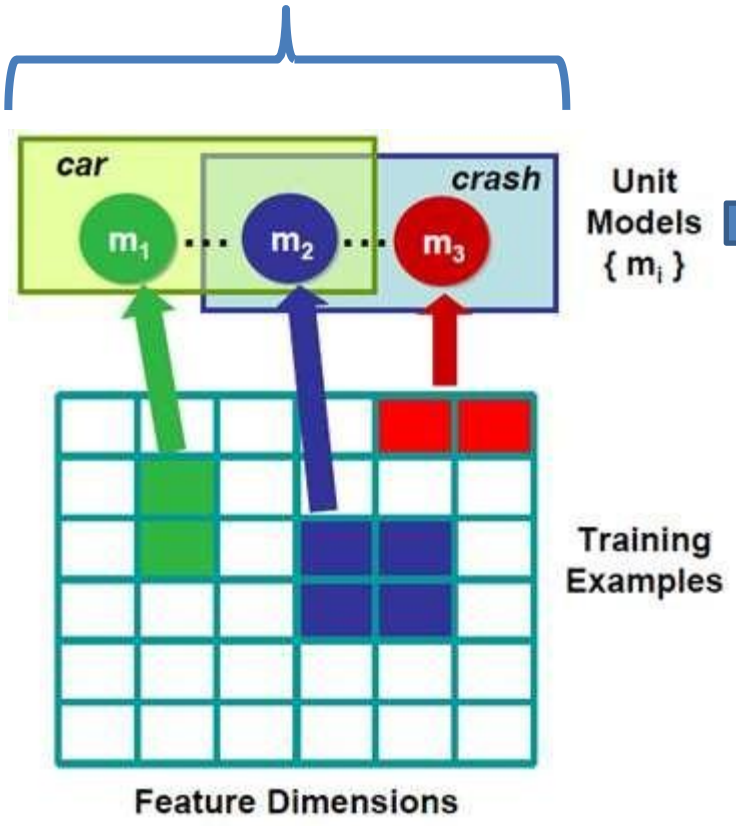
- Early IBM work from multimedia community
  - [Smith et al., ICME2003]



Frame level semantic features (2)

- IMARS: IBM Multimedia Analysis and Retrieval System

Discriminative semantic basis



[Yan et al, KDD 2007]

Ensemble Learning

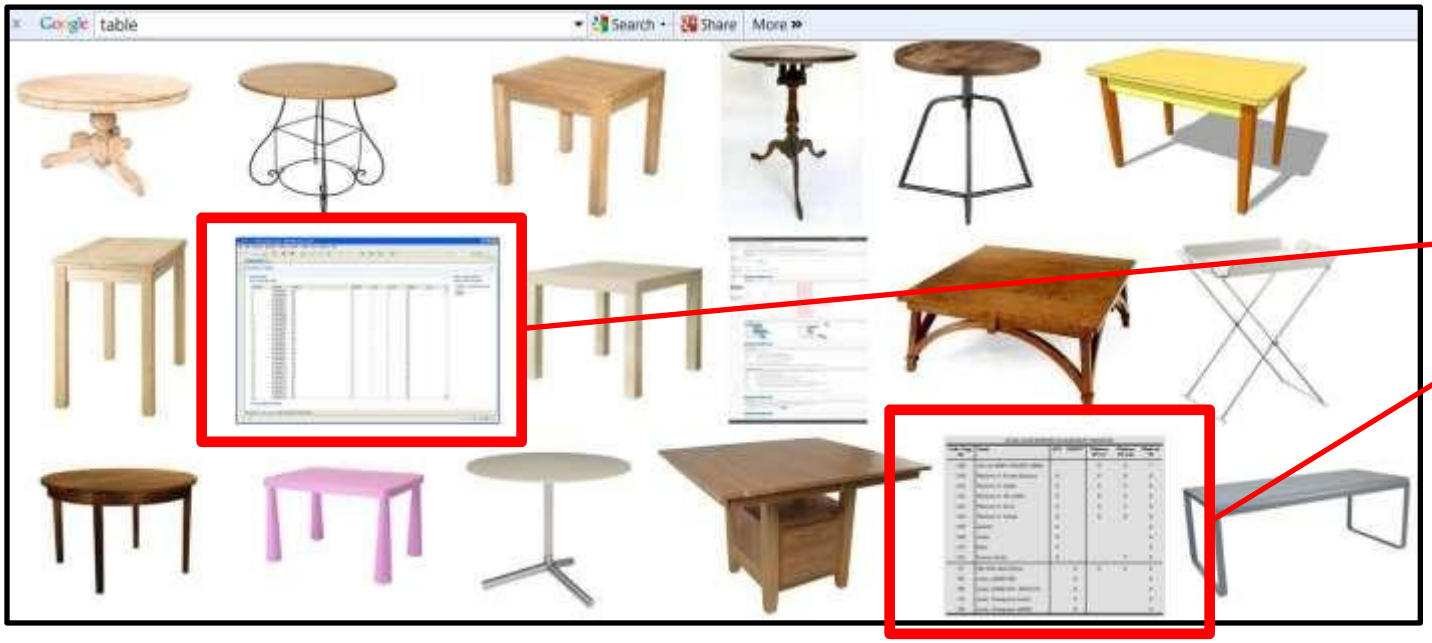
Rapid event modeling, e.g., "accident with high-speed skidding"

Frame level Semantic features (3)

- ClASSME: descriptor is formed by concatenating the outputs of weakly trained classifiers with noisy labels

[Torresani et al, ECCV2010]

Images used to train the “table” classeme (from Google image search)



Frame level semantic features (3)

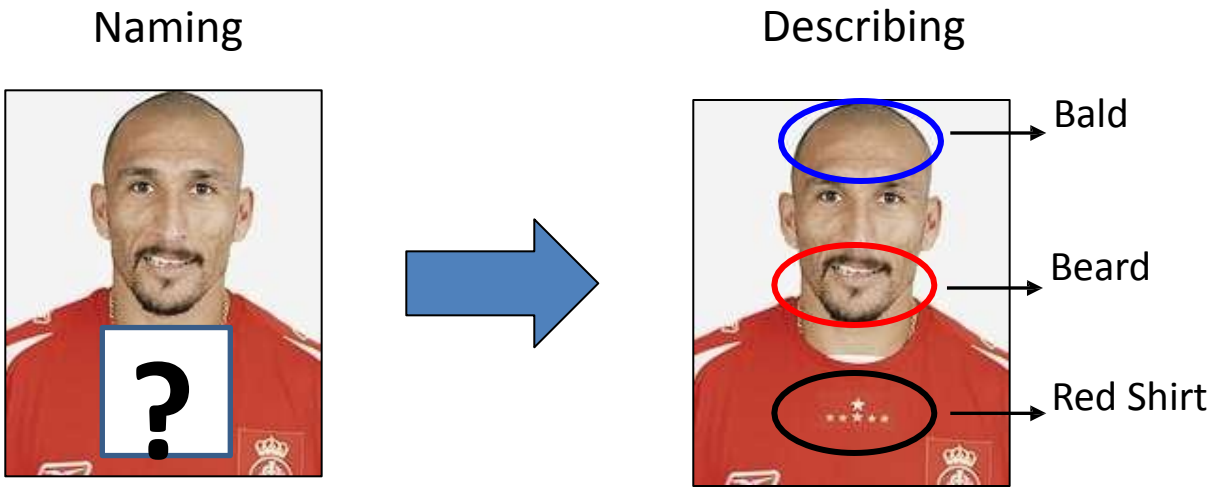
- CLASSMES

New category	Highly weighted classemes				
cowboy-hat	helmet	sports_ track	cake_ pan	collectible	muffin_ pan
duck	bomber_ plane	body_ of_ water	swimmer	walking	straight
elk	figure_ skater	bull_ male_ herd_ animal	cattle	gravesite	dead_ body
frisbee	watercraft_ surface	scsi_ cable	alarm_ clock	hindu	serving_ tray
trilobite-101	convex_ thing	mined_ area	cdplayer	roasting_ pan	western_ hemisphere_ person
wheelbarrow	taking_ care_ of_ something	baggage_ porter	canopy_ closure_ open	rowing_ shell	container_ pressure_ barrier

Compact and Efficient Descriptor, useful for large-scale classification

Features are not really semantic!

Semantic attributes



- Modifiers rather than (or in addition to) nouns
- Semantic properties that are shared among objects
- Attributes are category independent and trasferrable

People search in surveillance videos

- Traditional approaches: face recognition (naming)
  - Confronted by lighting, pose, and low SNR in surveillance

- Attribute based people search (describing)
  - Semantic attributes based search
  - Example:
    - Show me all bald people at the 42nd street station last month with dark skin, wearing sunglasses, wearing a red jacket

People search in surveillance videos

PENNSYLVANIA CAPITOL POLICE SUSPECT DESCRIPTION

SEX	RACE	AGE	HEIGHT	WEIGHT	TYPE OF WEAPON

HAIR/FACIAL HAIR

GLASSES (type)

TATTOOS

COMPLEXION

SCARS/MARKS

HAT (color, type)

TIE

COAT

SHIRT

PANTS/SHOES

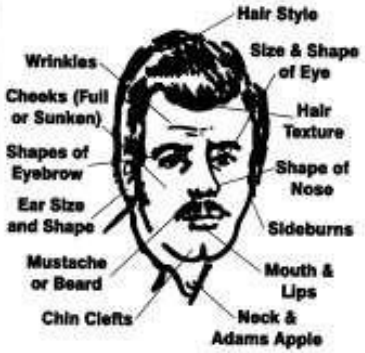
HARRISBURG EMERGENCY
DIAL 1-911
POLICE FIRE
MEDICAL

NON-EMERGENCY
717-787-3199

DON'T HANG UP

AUTO MAKE MODEL, COLOR	LICENSE NUMBER	DIRECTION OF ESCAPE	TIME OF DEPARTURE

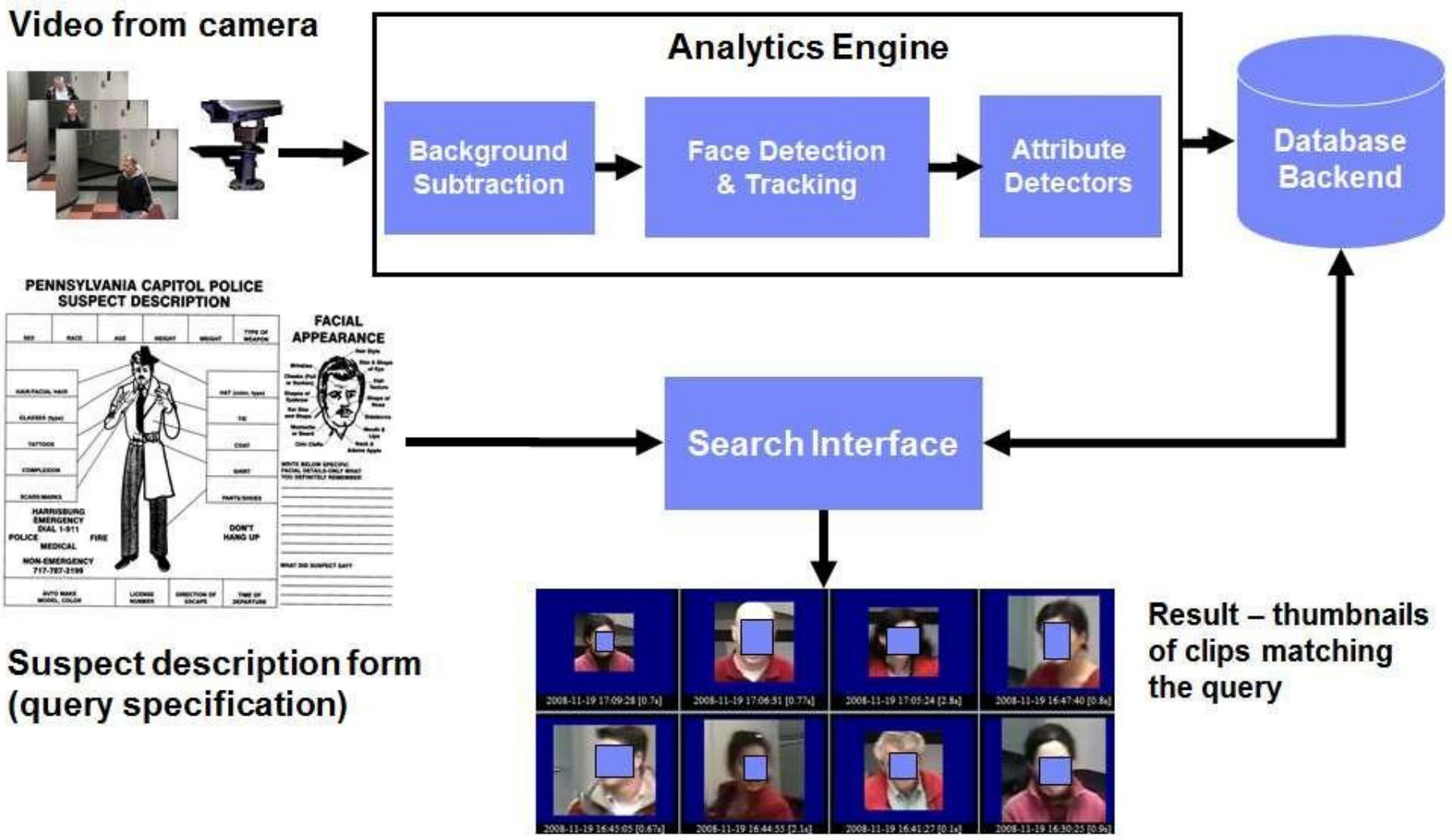
FACIAL APPEARANCE



WRITE BELOW SPECIFIC FACIAL DETAILS-ONLY WHAT YOU DEFINITELY REMEMBER

WHAT DID SUSPECT SAY?

People search in surveillance videos



People search in surveillance videos

People Search based on textual descriptions - It does not require training images for the target suspect.

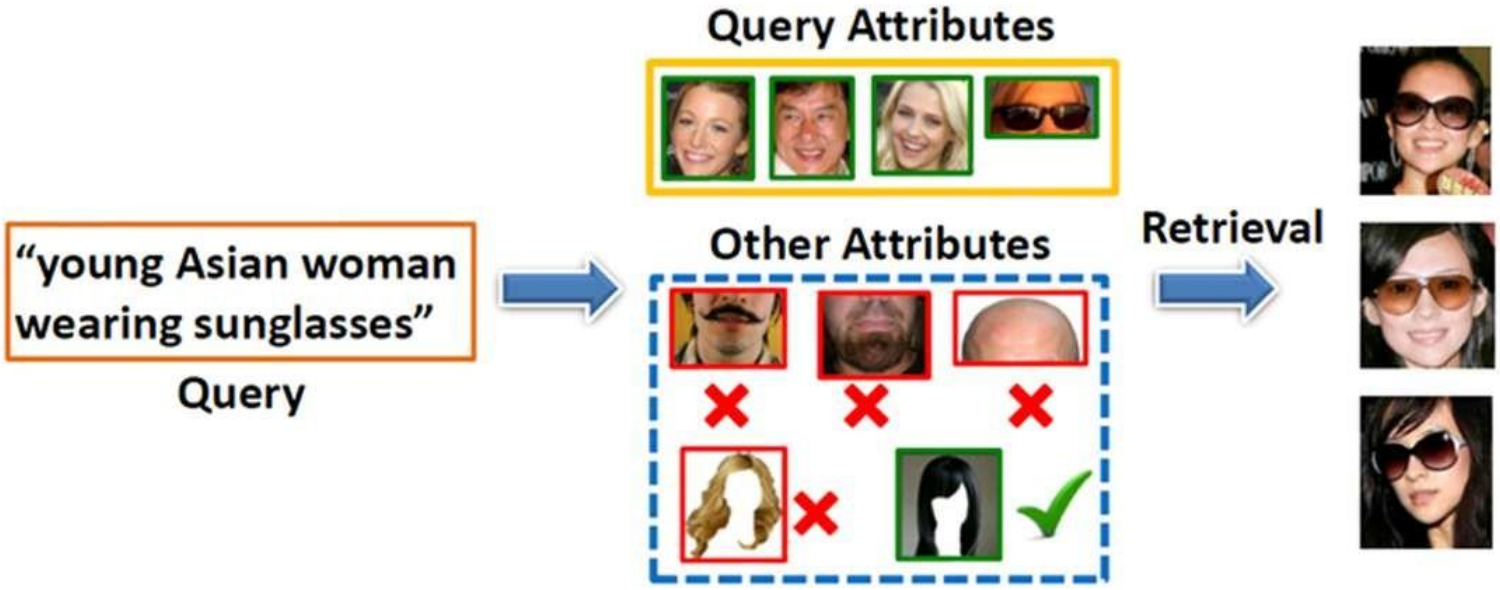
Robustness: attribute detectors are trained using lots of training images covering different lighting conditions, pose variation, etc.

Works well in low-resolution imagery (typical in video surveillance scenarios)

People search in surveillance videos

Modeling attribute correlations

[Siddiquie et al., "Image Ranking and Retrieval Based on Multi-Attribute Queries", CVPR 2011]



Attribute-Based Classification

Attribute-based classification

Recognition of Unseen Classes (Zero-Shot Learning)

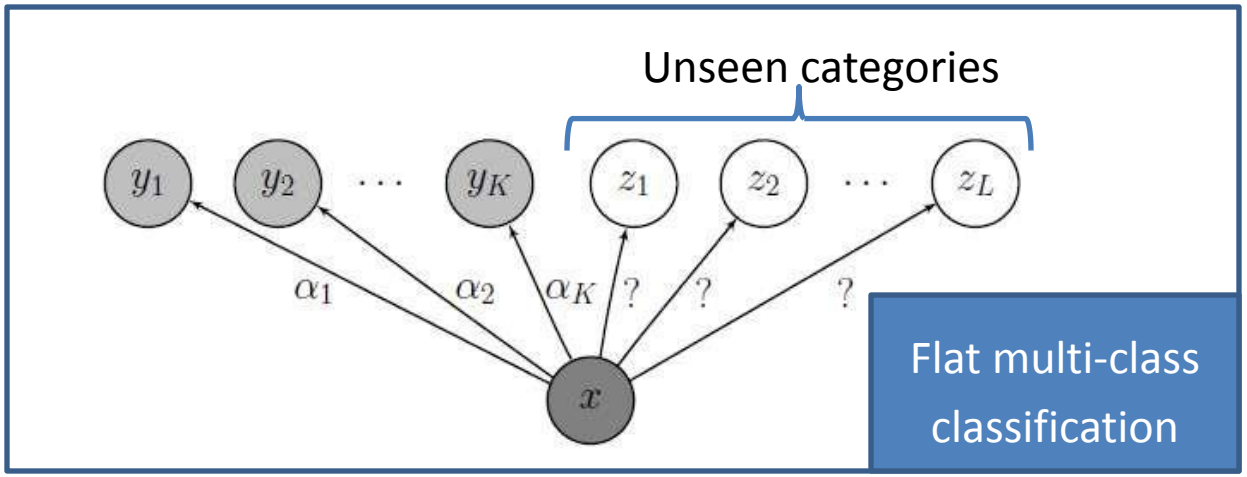
[Lampert et al., Learning To Detect Unseen Object Classes by Between-Class Attribute Transfer, CVPR 2009]

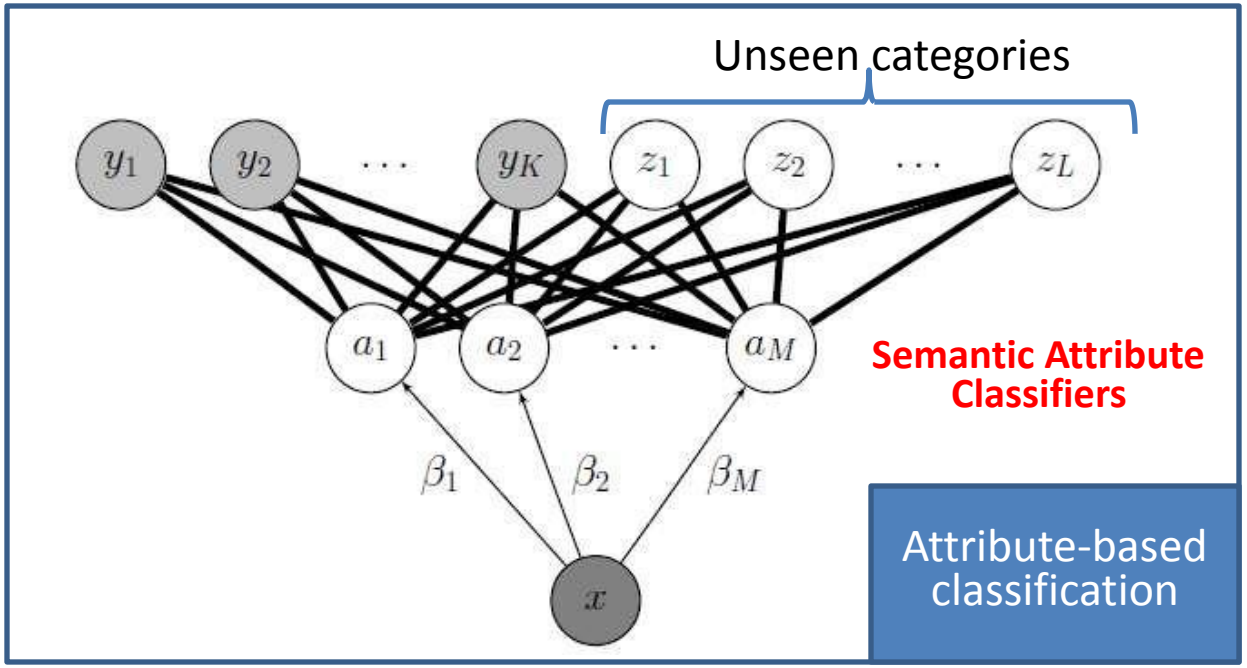
otter	
black:	yes
white:	no
brown:	yes
stripes:	no
water:	yes
eats fish:	yes
polar bear	
black:	no
white:	yes
brown:	no
stripes:	no
water:	yes
eats fish:	yes
zebra	
black:	yes
white:	yes
brown:	no
stripes:	yes
water:	no
eats fish:	no

(1). Train semantic attribute classifiers

(2). Obtain a classifier for an unseen object (no training samples) by just specifying which attributes it has

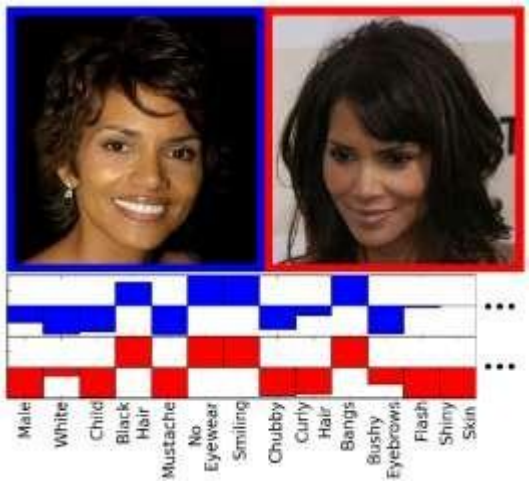
Attribute-based classification



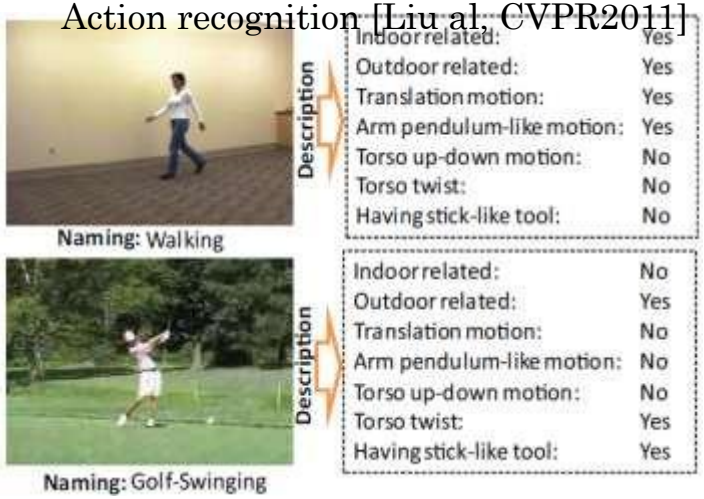


Attribute-based classification

Face verification [Kumar et al, ICCV 2009]



Action recognition [Liu al, CVPR2011]



Animal Recognition
[Lampert et al, CVPR 2009]

Person Re-identification
[Layne et al, BMVC 2012]

Bird Categorization [Farrell et al, ICCV 2011]

Many more! Significant growth in the past few years

Attribute-based classification

Note: Several recent methods use the term “attributes” to refer to non-semantic model outputs

In this case attributes are just mid-level features, like PCA, hidden layers in neural nets, … (non-interpretable splits)

Attribute-based classification

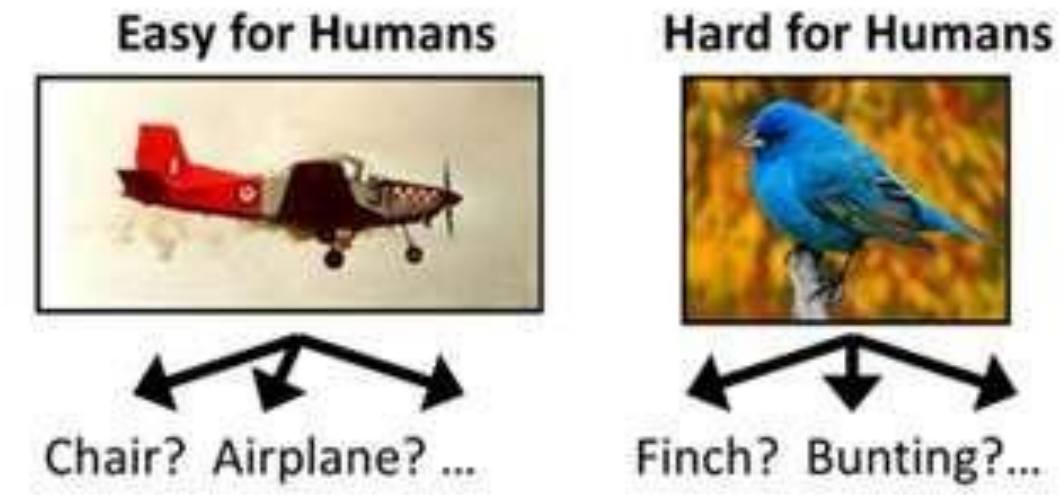
http://rogerioferis.com/VisualRecognitionAndSearch/Resources.html

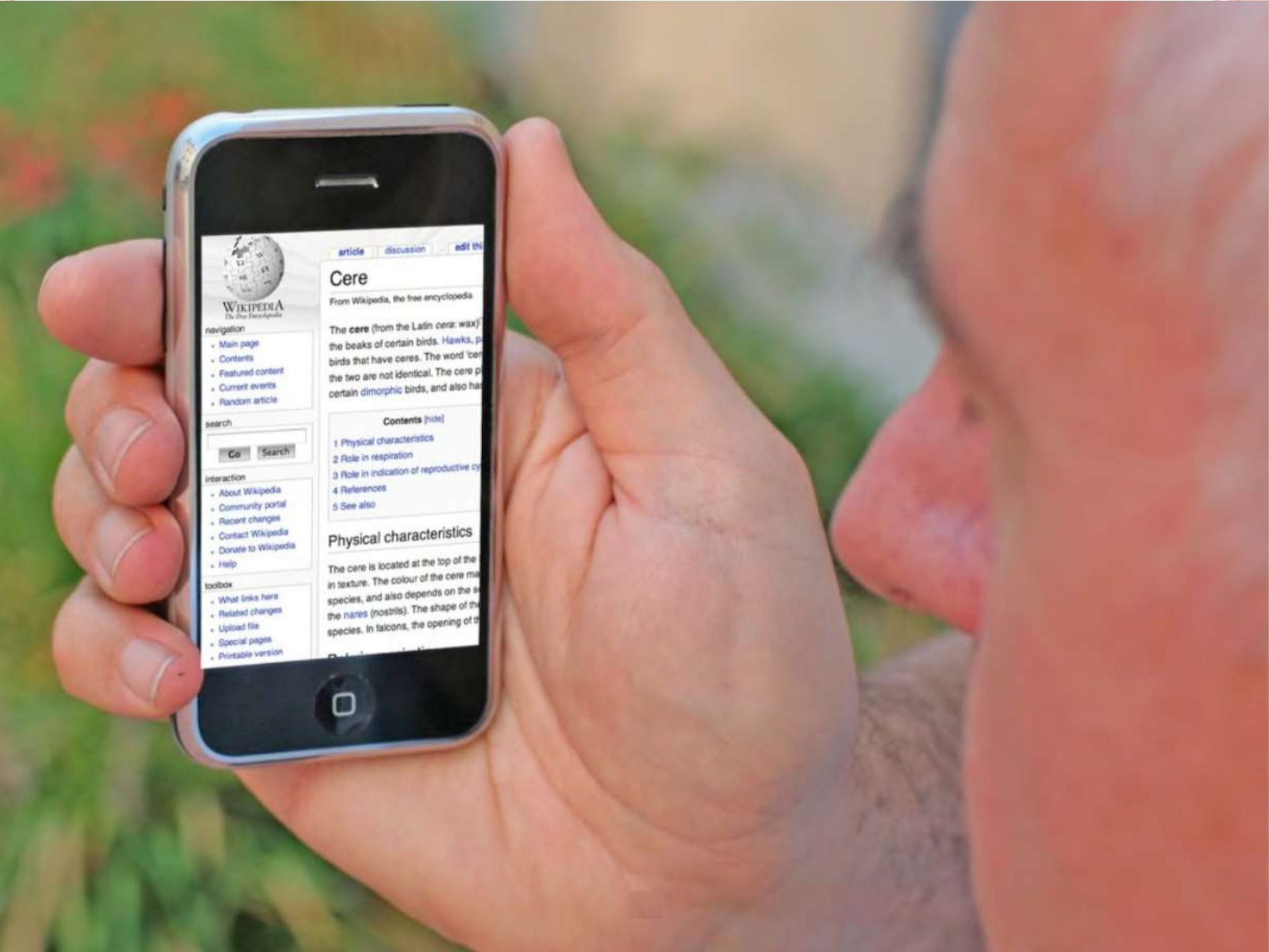
Datasets

Attributes

- Animals with Attributes – 30,475 images of 50 animals classes with 6 pre-extracted feature representations for each image.
- aYahoo and aPascal – Attribute annotations for images collected from Yahoo and Pascal VOC 2008.
- FaceTracer – 15,000 faces annotated with 10 attributes and fiducial points.
- PubFig – 58,797 face images of 200 people with 73 attribute classifier outputs.
- LFW – 13,233 face images of 5,749 people with 73 attribute classifier outputs.
- Human Attributes – 8,000 people with annotated attributes. Check also this link for another dataset of human attributes.
- SUN Attribute Database – Large-scale scene attribute database with a taxonomy of 102 attributes.
- ImageNet Attributes – Variety of attribute labels for the ImageNet dataset.
- Relative attributes – Data for OSR and a subset of PubFig datasets. Check also this link for the WhittleSearch data.
- Attribute Discovery Dataset – Images of shopping categories associated with textual descriptions.
- Caltech-UCSD Birds Dataset – Hundreds of bird categories with annotated parts and attributes.

Attributes for Fine-Grained Categorization




Wikipedia
navigation
Main page
Contents
Featured content
Current events
Random article
search
Go
Search
interaction
About Wikipedia
Community portal
Recent changes
Contact Wikipedia
Donate to Wikipedia
Help
toolbox
What links here
Related changes
Upload file
Special pages
Printable version
article
discussion
Cere
From Wikipedia, the free encyclopedia
The cere (from the Latin cera: wax)
the beaks of certain birds. Hawks,
birds that have ceres. The word 'cer
the two are not identical. The cere
certain dimorphic birds, and also ha
Contents [hide]
1 Physical characteristics
2 Role in respiration
3 Role in indication of reproductive
4 References
5 See also
Physical characteristics
The cere is located at the top of the
in texture. The colour of the cere
species, and also depends on the
the nares (nostrils). The shape of
species. In falcons, the opening of

Fine-grained categorization

Visipedia (~~http://http://visipedia.org /~~)

- Machines collaborating with humans to organize visual knowledge, connecting text to images, images to text, and images to images
- Easy annotation interface for experts (powered by computer vision)

Visual Query: Fine-grained Bird Categorization

Picture credit: Serge Belongie

Fine-grained categorization

African

Is it an African or Indian Elephant?

Indian

Example-based Fine-Grained Categorization is Hard!!

Slide Credit: Christoph Lampert

Fine-grained categorization

African

Larger Ears

Is it an African or Indian Elephant?

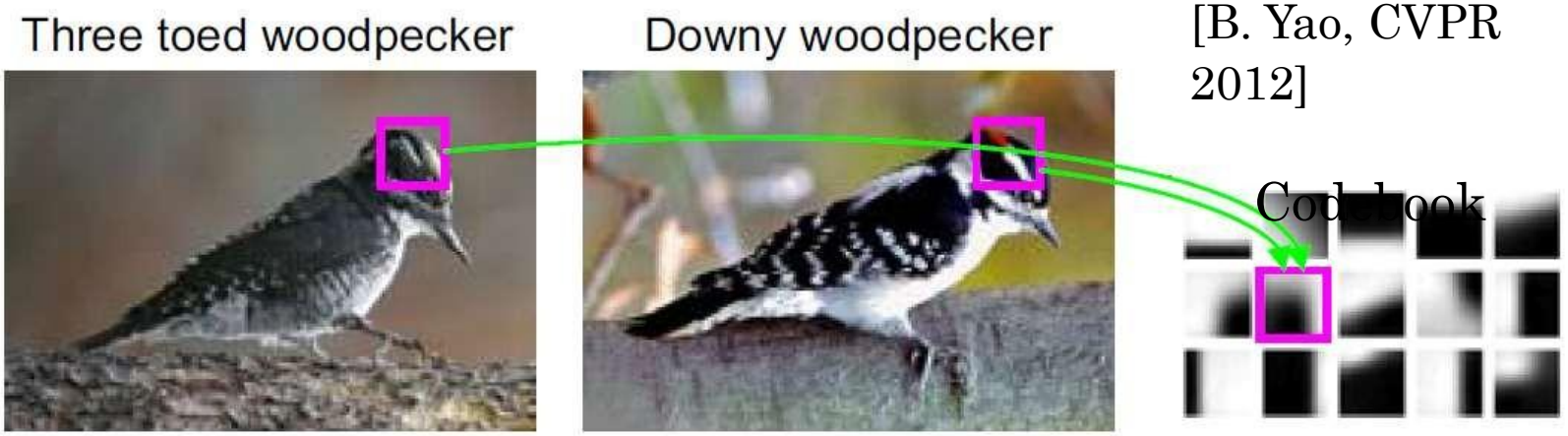
Indian

Smaller Ears

Visual distinction of subordinate categories may be quite subtle, usually based on Attributes

Fine-grained categorization

- Standard classification methods may not be suitable because the variation between classes is small …

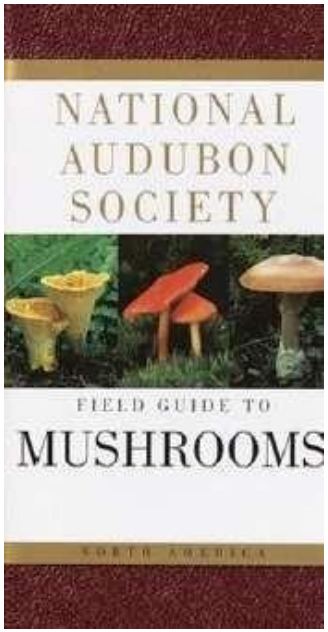


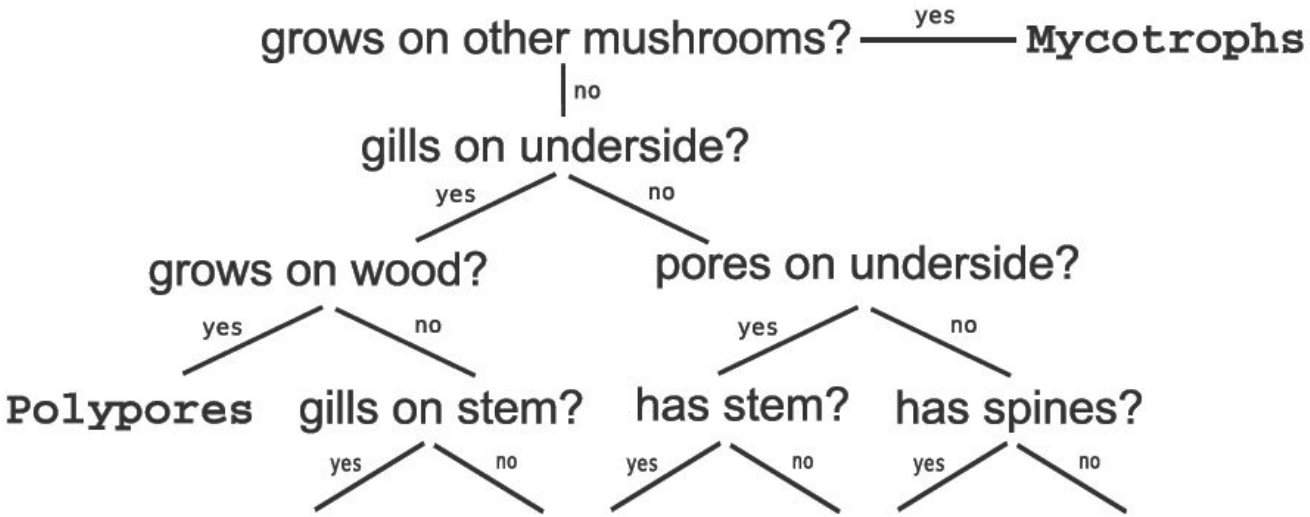
[B. Yao, CVPR 2012]

- … and intra-class variation is still high.

Fine-grained categorization

- Humans rely on field guides!
- Field guides usually refer to parts and attributes of the object





Slide Credit: Pietro Perona

Fine-grained categorization

[Branson et al, Visual Recognition with Humans in the Loop, ECCV 2010]

Visual 20 Questions Game

Blue Belly? no

↓

Cone-shaped Beak? yes

↓

Striped Wing? yes

↓

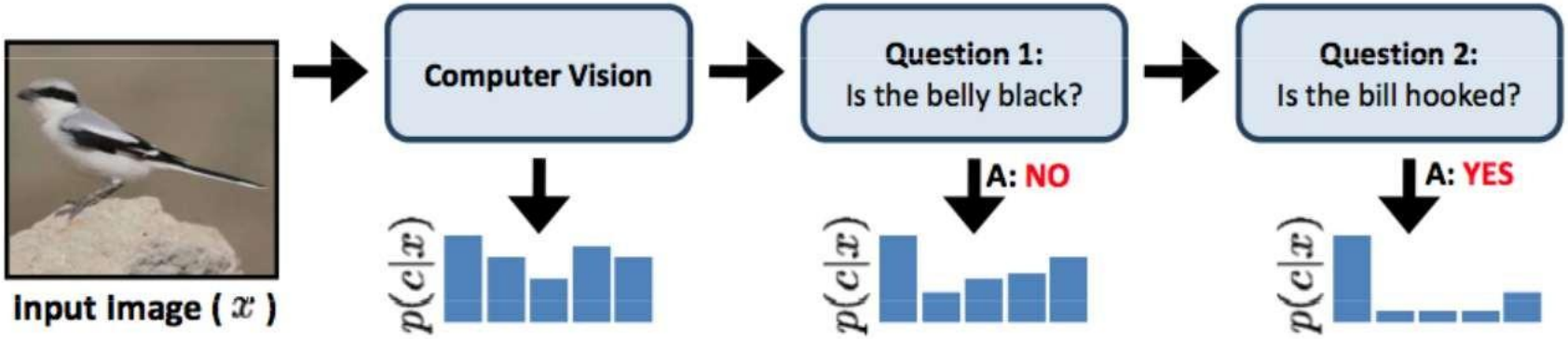
American Goldfinch? yes

Hard classification problems can be turned into a sequence of easy ones

Fine-grained categorization

[Branson et al, Visual Recognition with Humans in the Loop, ECCV 2010]

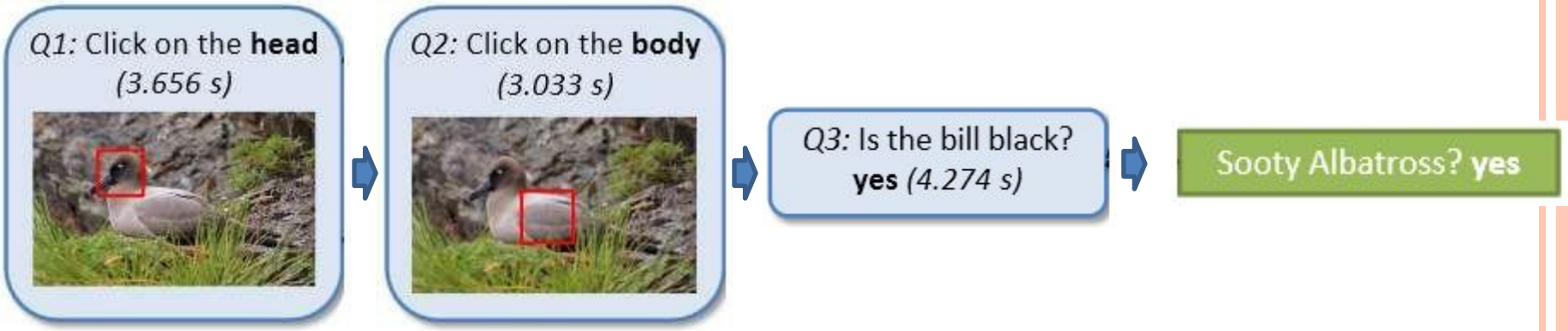
- Computer vision reduces the amount of human-interaction (minimizes the dnumber of questions)



Fine-grained categorization

[Wah et al, Multiclass Recognition and Part Localization with Humans in the Loop, ICCV 2011]

- **Localized part and attribute detectors.**
- Questions include asking the user to localize parts.



Fine-grained categorization

- http://www.vision.caltech.edu/visipedia/CUB-200-2011.html

Caltech-UCSD Birds-200-2011

Browse

Click here to browse the dataset.

Details

Caltech-UCSD Birds-200-2011 (CUB-200-2011) is an extended version of the CUB-200 dataset, with roughly double the number of images per class and new part location annotations. For detailed information about the dataset, please see the technical report linked below.

- **Number of categories:** 200
- **Number of images:** 11,788
- **Annotations per image:** 15 Part Locations, 312 Binary Attributes, 1 Bounding Box

Some related datasets are Caltech-256, the Oxford Flower Dataset, and Animals with Attributes. More datasets are available at the Caltech Vision Dataset Archive.

Fine-grained categorization

Video Demo:

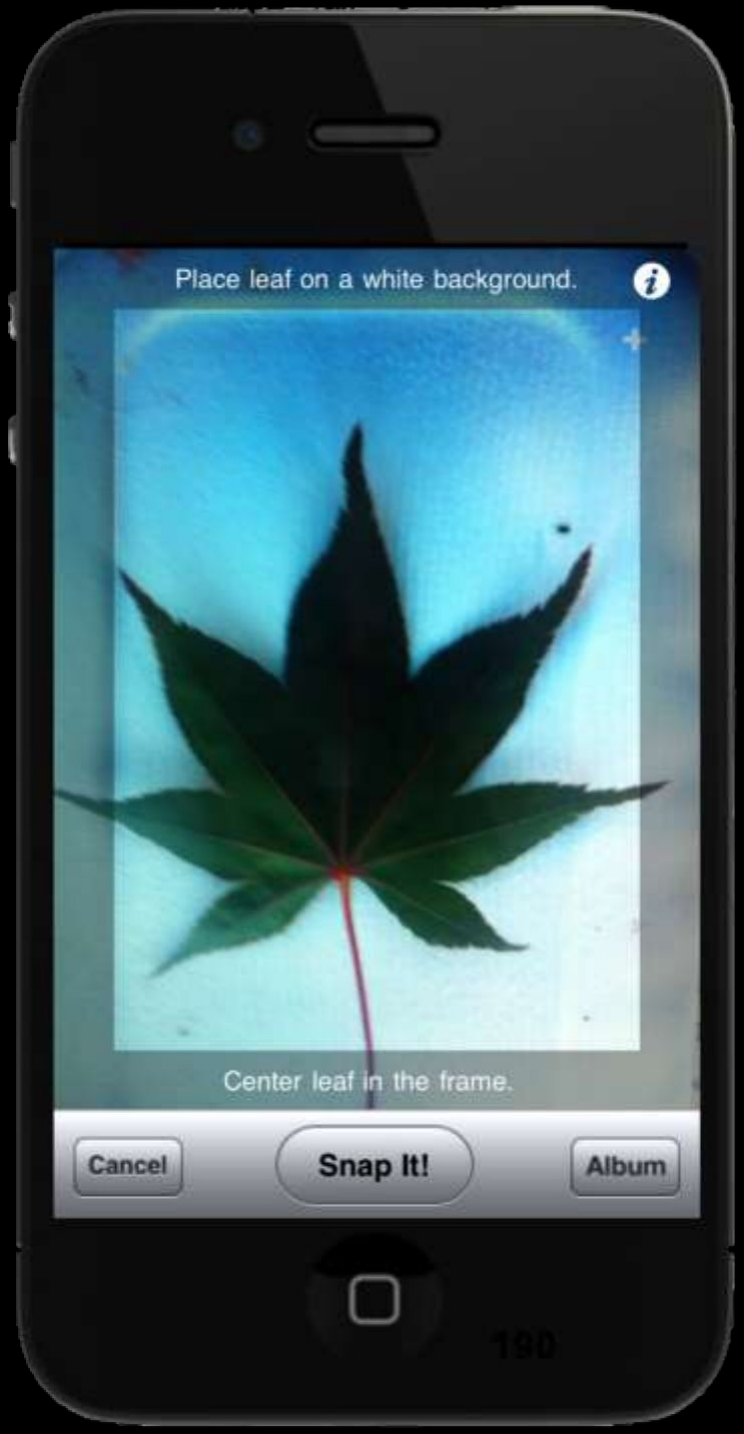
http://www.youtube.com/watch?v=_ReKVqnDXzA

leafsnap

Like a normal field guide…

- that you can search and sort
- and with visual recognition

See N. Kumar et al, "Leafsnap: A Computer Vision System for Automatic Plant Species Identification, ECCV 2012



leafsnap

Available on the App Store

Available on the iPad App Store

- Nearly 1 million downloads
  - 40k new users per month
  - 100k active users
- 1.7 million images taken
  - 100k new images/month
  - 100k users with > 5 images
- Users from all over the world
- Botanists, educators, kids, hobbyists, photographers, …

Slide Credit: Neeraj Kumar

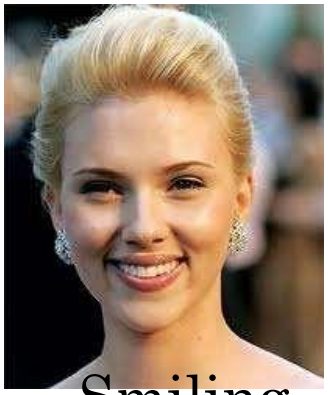
Fine-grained categorization

Check the fine-grained visual categorization workshop:
http://www.fgvc.org/

Relative Attributes

Relative attributes

[Parikh & Grauman, Relative Attributes, ICCV 2011]



Smiling

???

No smiling

t

Natural

???

Not natural

Slide credit: Parikh &Grauman

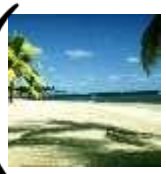
Learning Relative attributes

For each attribute a_m, e.g., "openness"

Supervision consists of:

O_m: $\{($  $\succ$  $), \ldots\}$,

Ordered pairs

S_m: $\{($  $\sim$  $), \ldots\}$

Similar pairs

Slide credit: Parikh &Grauman

Learning Relative attributes

Learn a ranking function

$$r_m(\boldsymbol{x_i}) = \boldsymbol{w_m^T x_i}$$

Image features

Learned parameters

that best satisfies the constraints:

$$\forall (i,j) \in O_m : \boldsymbol{w_m^T x_i} > \boldsymbol{w_m^T x_j}$$

$$\forall (i,j) \in S_m : \boldsymbol{w_m^T x_i} = \boldsymbol{w_m^T x_j}$$

Slide credit: Parikh &Grauman

Learning Relative attributes

Max-margin learning to rank formulation

$$\min \quad \left(\frac{1}{2}||\boldsymbol{w}_m^T||_2^2 + C\left(\sum \xi_{ij}^2 + \sum \gamma_{ij}^2\right)\right)$$

$$\text{s.t} \quad \boldsymbol{w}_m^T(\boldsymbol{x_i} - \boldsymbol{x_j}) \geq 1 - \xi_{ij}, \forall (i,j) \in O_m$$

$$|\boldsymbol{w}_m^T(\boldsymbol{x_i} - \boldsymbol{x_j})| \leq \gamma_{ij}, \forall (i,j) \in S_m$$

$$\xi_{ij} \geq 0; \gamma_{ij} \geq 0$$

Based on [Joachims 2002]

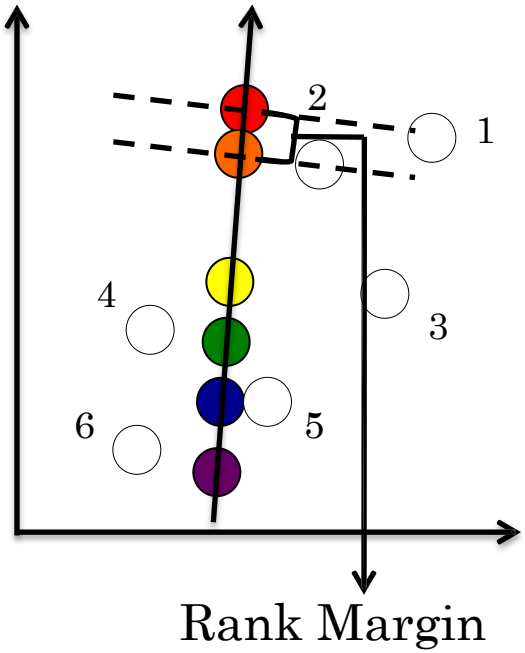


Image → Relative Attribute Score

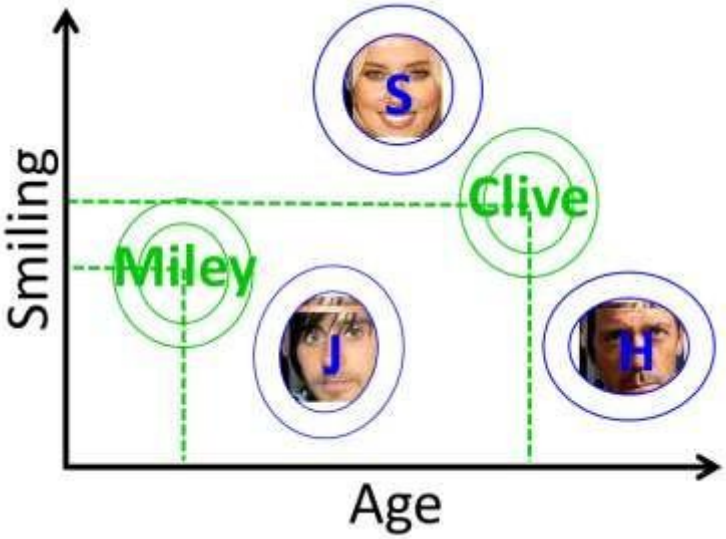
Slide credit: Parikh

Relative Zero-shot learning

- Each image is converted into a vector of relative attribute scores indicating the strength of each attribute
- A Gaussian distribution for each category is built in the relative attribute space. The distribution of unseen categories is estimated based on the specified constraints and the distributions of seen categories
- Max-likelihood is then used for classification

Age: Hugh ≻ Clive ≻ Scarlett
Jared ≻ Miley

Smiling: Miley ≻ Jared



Blue: Seen class Green: Unseen class

Relative image description

Binary:

Not natural

Not open

Has perspective

Relative :

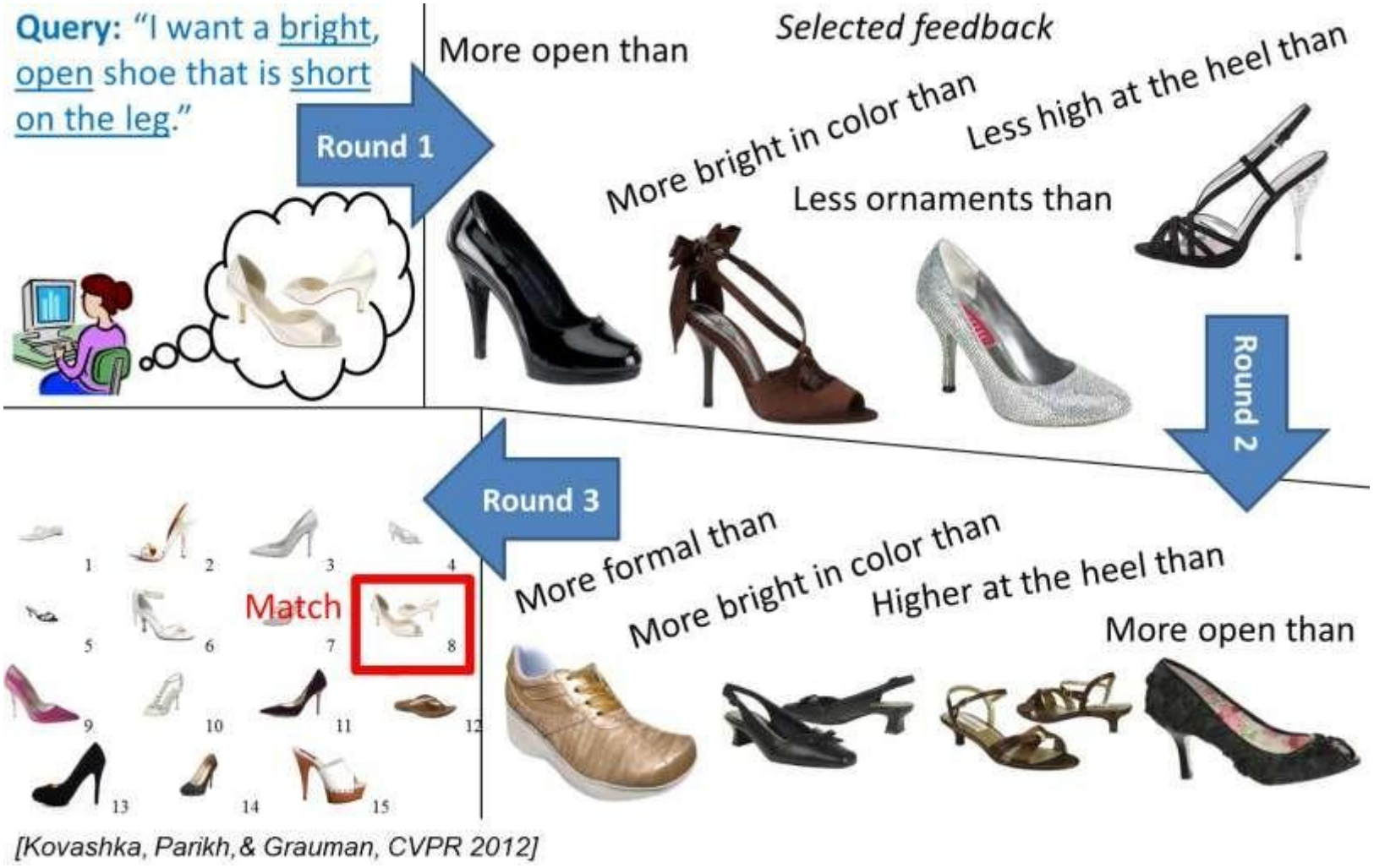
More natural than tallbuilding
Less natural than forest

More open than tallbuilding
Less open than coast

Has more perspective than tallbuilding

Slide credit: Parikh &Grauman

Whittle search



[Kovashka, Parikh, & Grauman, CVPR 2012]

Slide credit: Kristen Grauman

Summary

Semantic attribute classifiers can be useful for:

- Describing images of unknown objects [Farhadi et al, CVPR 2009]
- Recognizing unseen classes [Lampert et al, CVPR 2009]
- Reducing dataset bias (trained across classes)
- Effective object search in surveillance videos [Vaquero et al, WACV 2009]
- Compact descriptors / Efficient image retrieval [Douze et al, CVPR 2011]
- Fine-grained object categorization [Wah et al, ICCV 2011]
- Face verification [Kumar et al, 2009], Action recognition [Liu et al, CVPR 2011], Person re-identification [Layne et al, BMVC 2012] and other classification tasks.
- Other applications, such as sentence generation from images [Kulkarni et al, CVPR 2011], image aesthetics prediction [Dhar et al CVPR 2011], …