Window- versus part- based representation for Object Recognition

VBM 686 – Bilgisayarli Goru

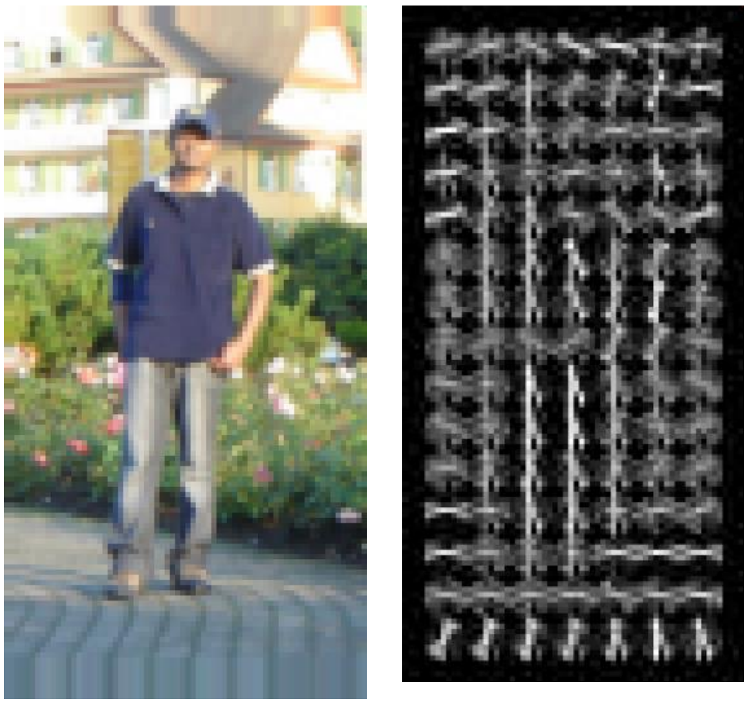
Pinar Duygulu

Hacettepe University

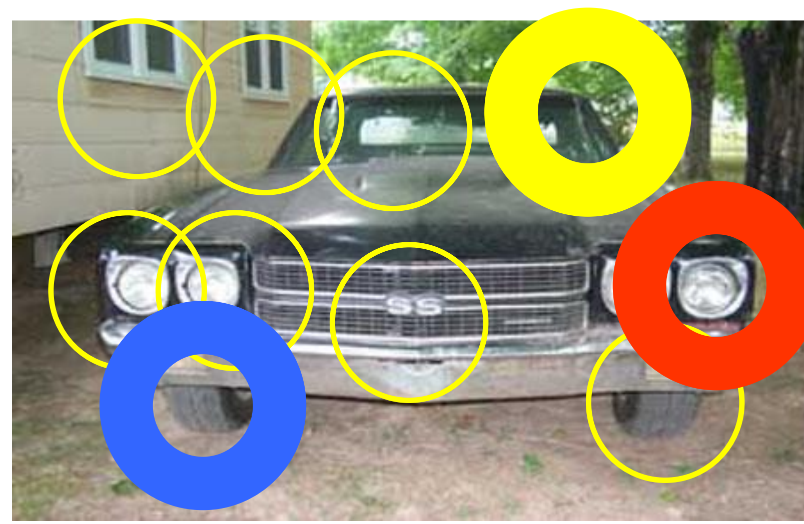
(Slide credits:

Kristen Grauman, Fei fei Li, Antonio Torralba, Hames Hays)

Generic category recognition: representation choice

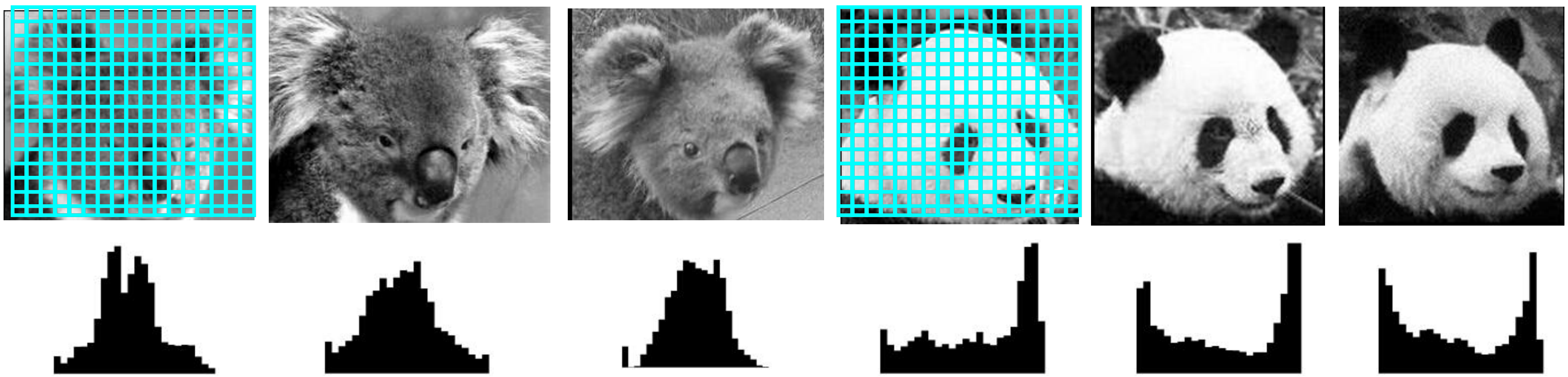


Window-based



Part-based

Window-based models
Building an object model

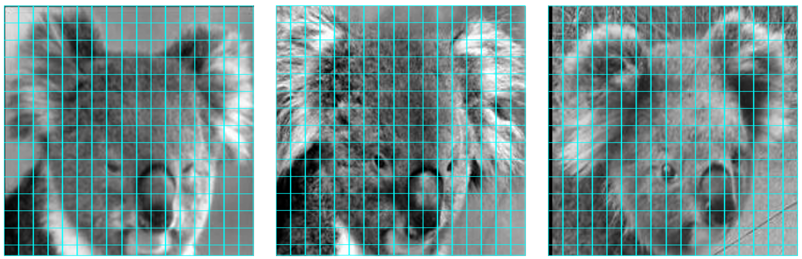


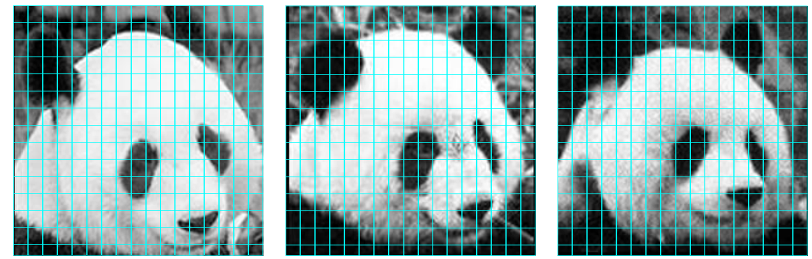
Simple holistic descriptions of image content

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

Window-based models
Building an object model

- Pixel-based representations sensitive to small shifts



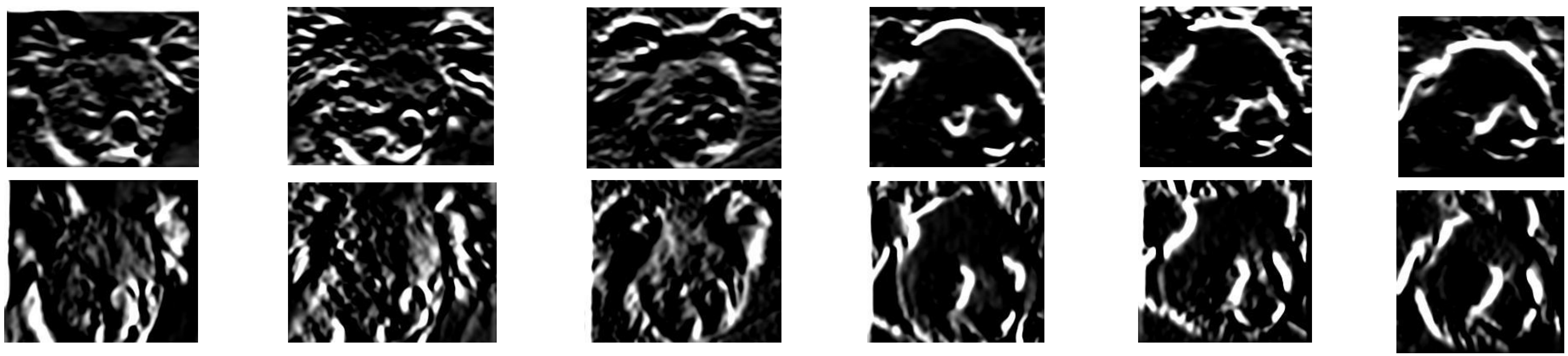


- Color or grayscale-based appearance description can be sensitive to illumination and intra-class appearance variation

Kristen Grauman

Window-based models
Building an object model

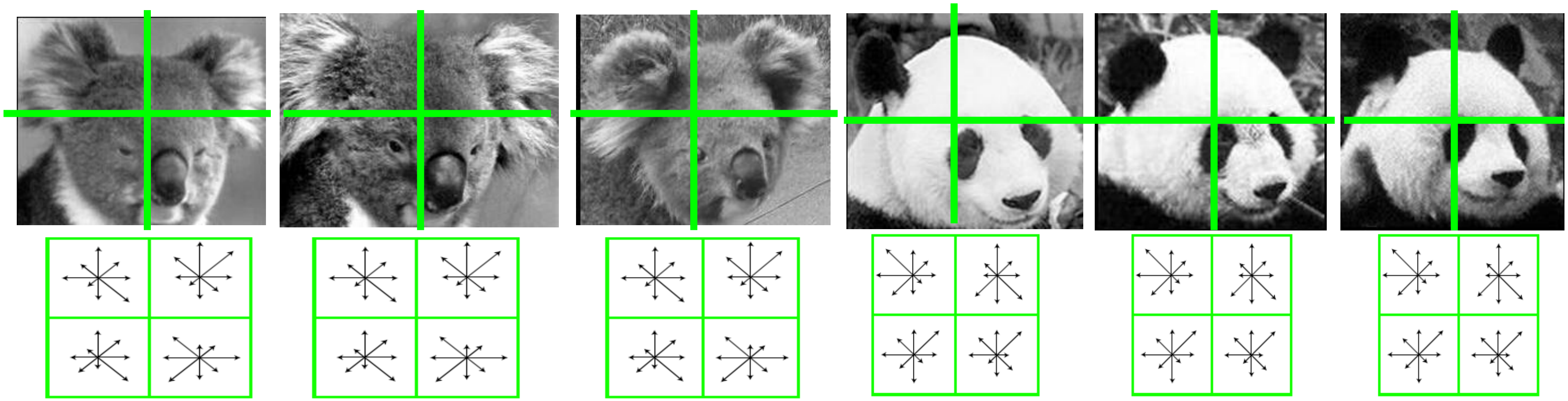
- Consider edges, contours, and (oriented) intensity gradients



Kristen Grauman

Window-based models
Building an object model

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



- Summarize local distribution of gradients with histogram
  - Locally orderless: offers invariance to small shifts and rotations
  - Contrast-normalization: try to correct for variable illumination

Kristen Grauman

Window-based models
Building an object model

Given the representation, train a binary classifier

Car/non-car
Classifier

No, not a car.

Yes, car.

Kristen Grauman

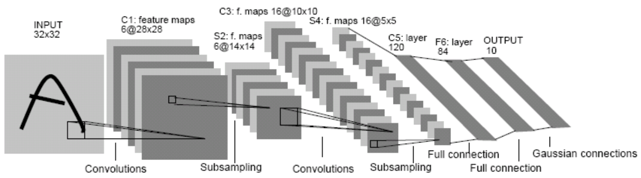
Discriminative classifier construction

Nearest neighbor

10^6 examples

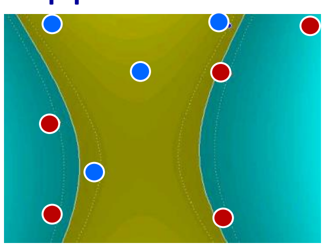
Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005...

Neural networks



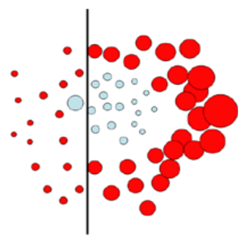
LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998
…

Support Vector Machines



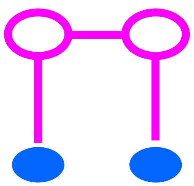
Guyon, Vapnik
Heisele, Serre, Poggio, 2001,…

Boosting



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

Conditional Random Fields



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

Slide adapted from Antonio Torralba

Influential Works in Detection

- Sung-Poggio (1994, 1998) : ~1450 citations
  - Basic idea of statistical template detection (I think), bootstrapping to get “face-like” negative examples, multiple whole-face prototypes (in 1994)
- Rowley-Baluja-Kanade (1996-1998) : ~2900
  - “Parts” at fixed position, non-maxima suppression, simple cascade, rotation, pretty good accuracy, fast
- Schneiderman-Kanade (1998-2000,2004) : ~1250
  - Careful feature engineering, excellent results, cascade
- Viola-Jones (2001, 2004) : ~6500
  - Haar-like features, Adaboost as feature selection, hyper-cascade, very fast, easy to implement
- Dalal-Triggs (2005) : ~2000
  - Careful feature engineering, excellent results, HOG feature, online code
- Felzenszwalb-Huttenlocher (2000): ~800
  - Efficient way to solve part-based detectors
- Felzenszwalb-McAllester-Ramanan (2008)? ~350
  - Excellent template/parts-based blend

Slide: Derek Hoiem

Generic category recognition: basic framework

- Build/train object model
  - Choose a representation
  - Learn or fit parameters of model / classifier
- **Generate candidates in new image**
- **Score the candidates**

Window-based models
Generating and scoring candidates

Car/non-car Classifier

Kristen Grauman

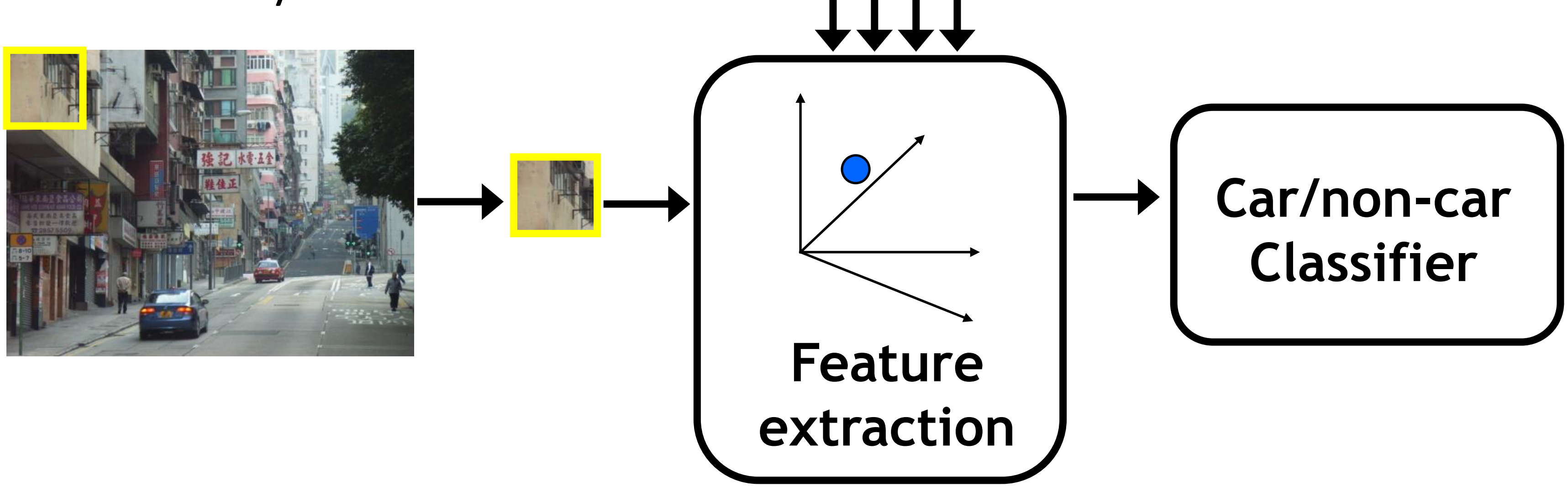
Window-based object detection: recap

Training:

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

Given new image:

1. Slide window
2. Score by classifier



Kristen Grauman

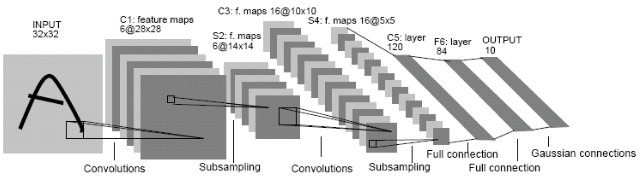
Discriminative classifier construction

Nearest neighbor

10^6 examples

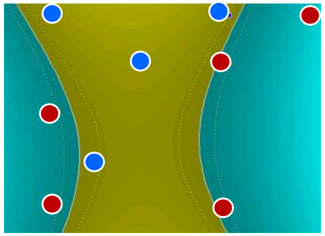
Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005...

Neural networks



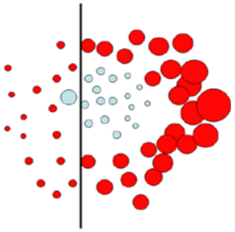
LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998
…

Support Vector Machines



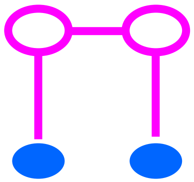
Guyon, Vapnik
Heisele, Serre, Poggio, 2001,…

Boosting



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

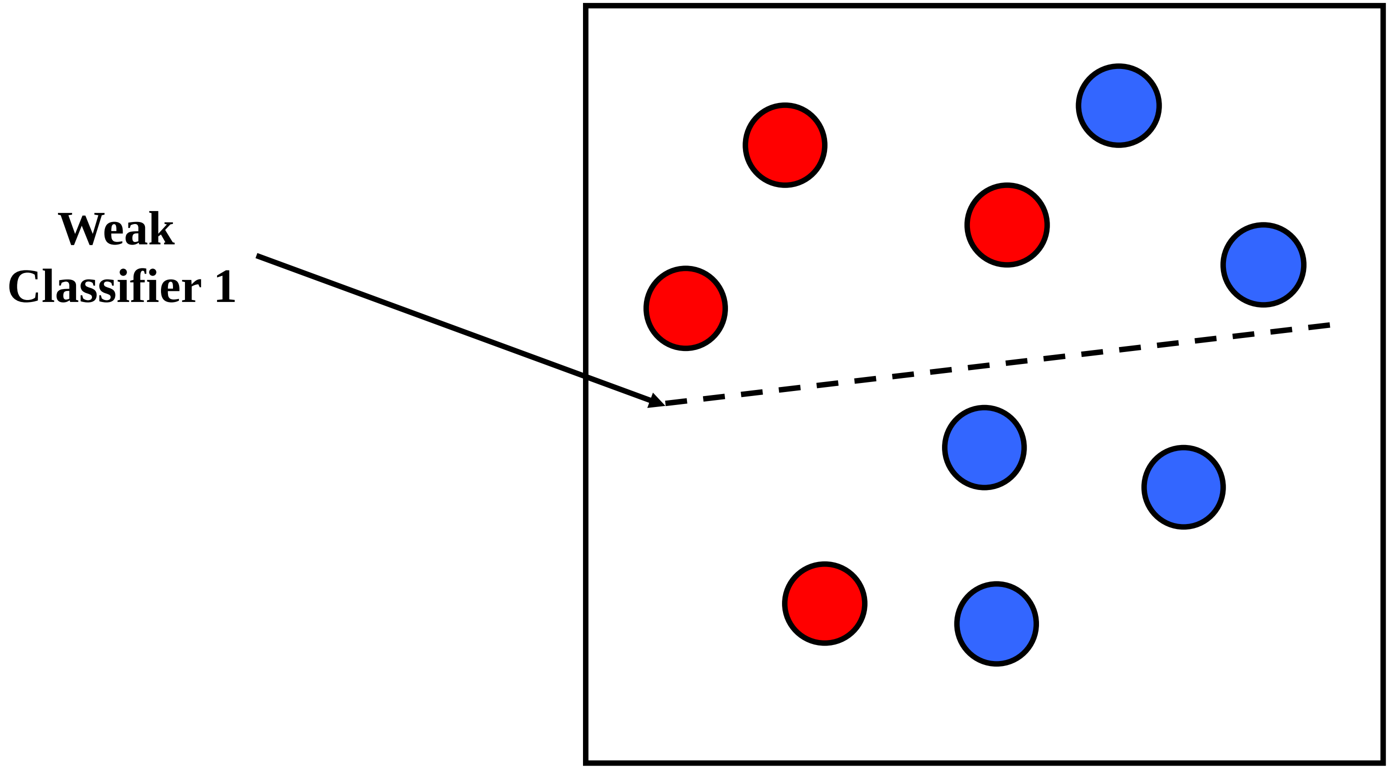
Conditional Random Fields



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

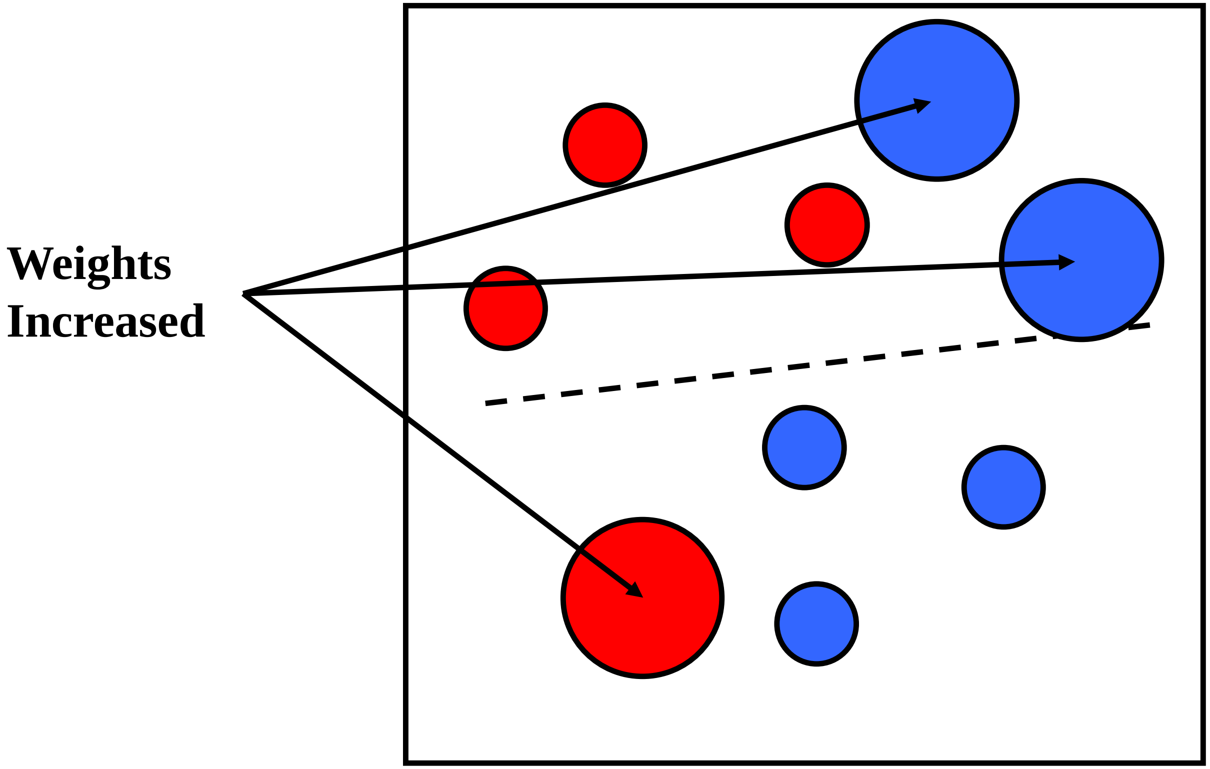
Slide adapted from Antonio Torralba

Boosting intuition

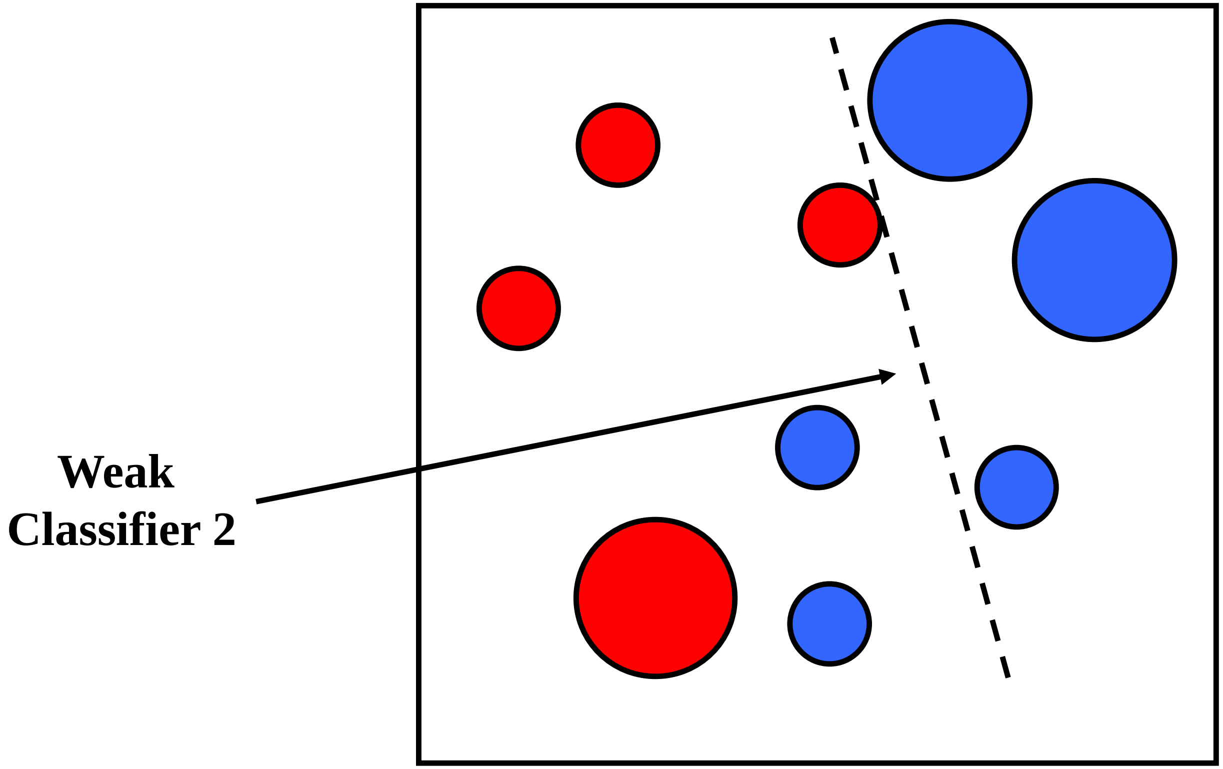


Slide credit: Paul Viola

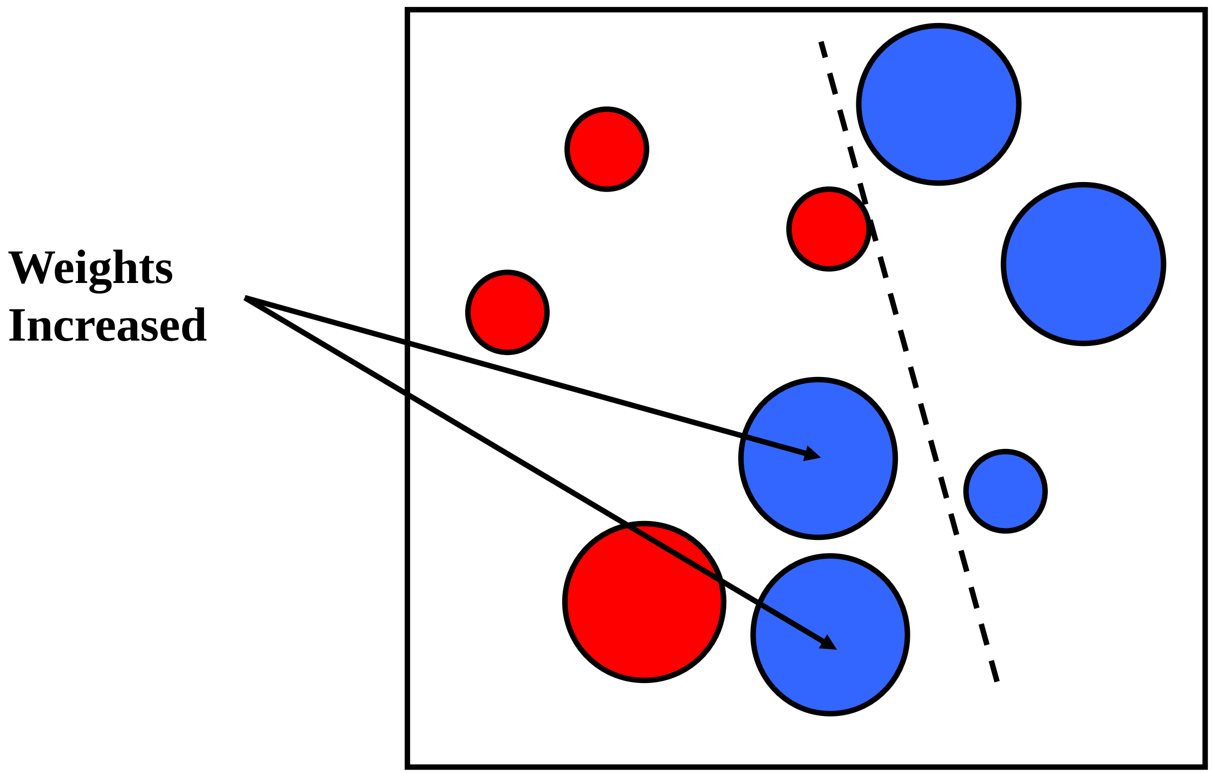
Boosting illustration



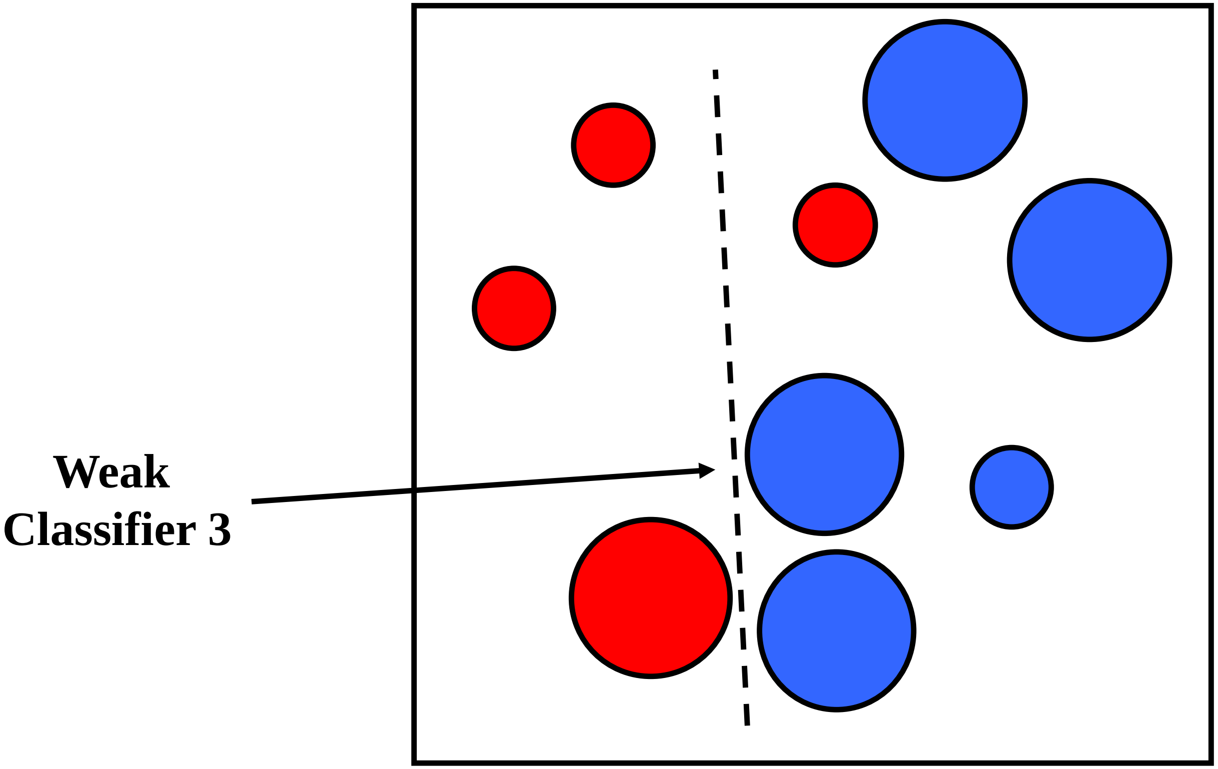
Boosting illustration



Boosting illustration

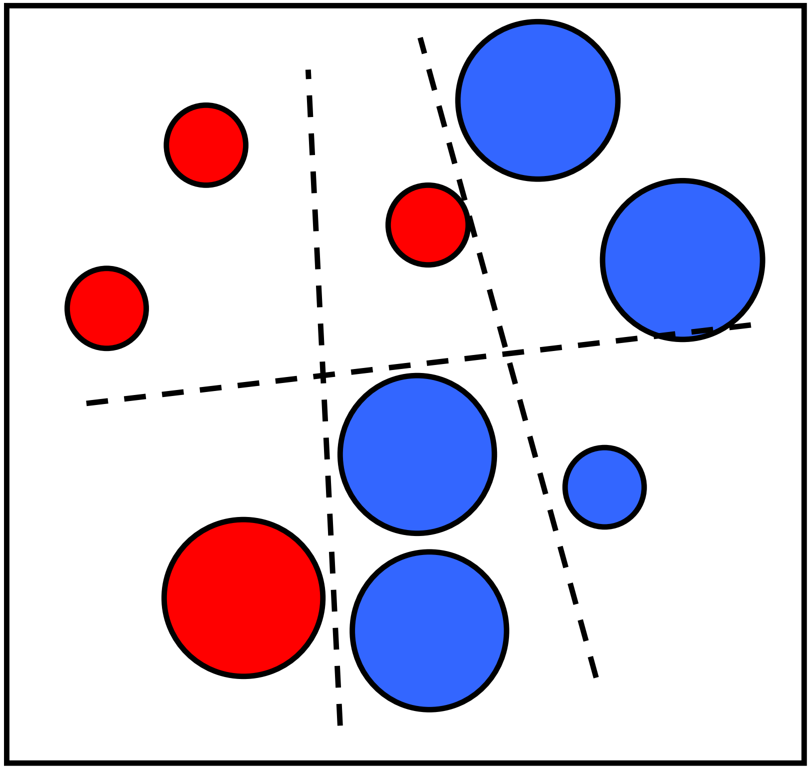


Boosting illustration



Boosting illustration

Final classifier is a combination of weak classifiers



Boosting: training

- Initially, weight each training example equally
- In each boosting round:
  - Find the weak learner that achieves the lowest *weighted* training error
  - Raise 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)

Slide credit: Lana Lazebnik

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

Viola-Jones face 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,

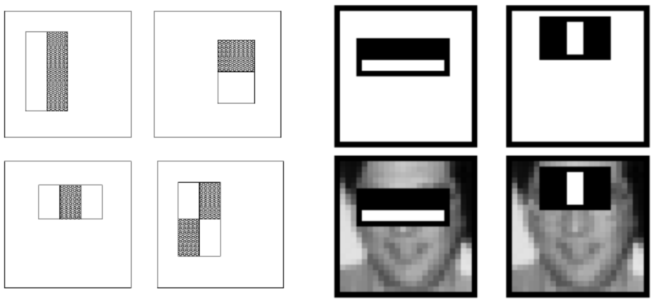
Viola-Jones face detector

Main idea:

- 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

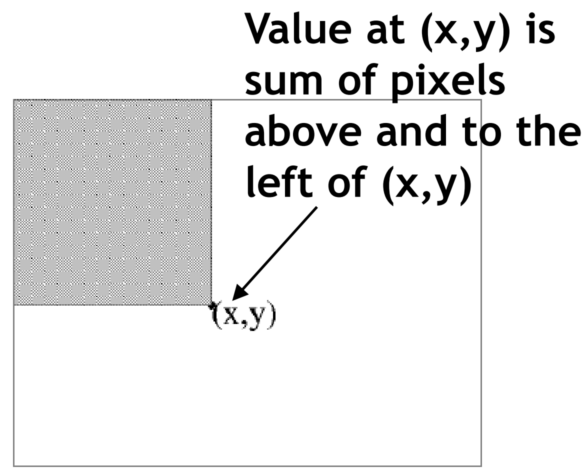
Viola-Jones detector: features



“Rectangular” filters

Feature output is difference between adjacent regions

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



Integral image

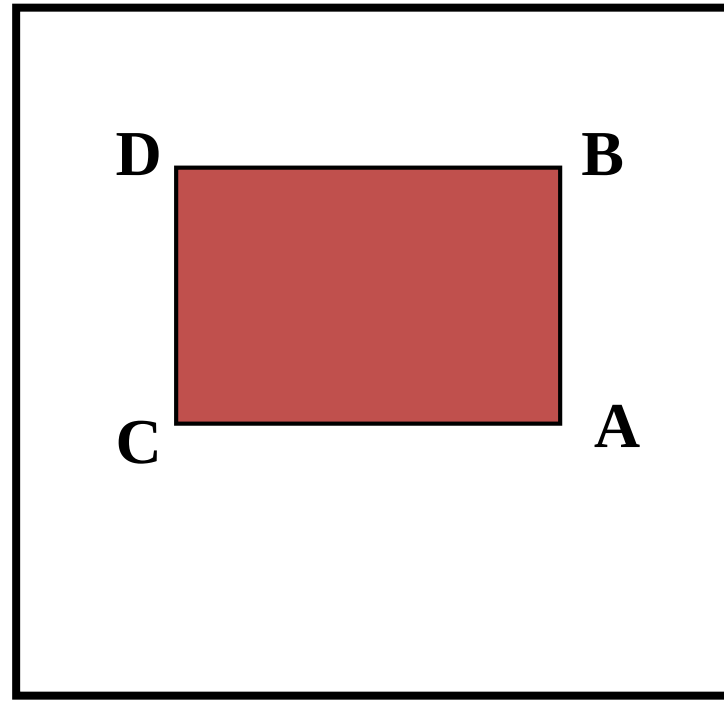
Kristen Grauman

Computing sum within a rectangle

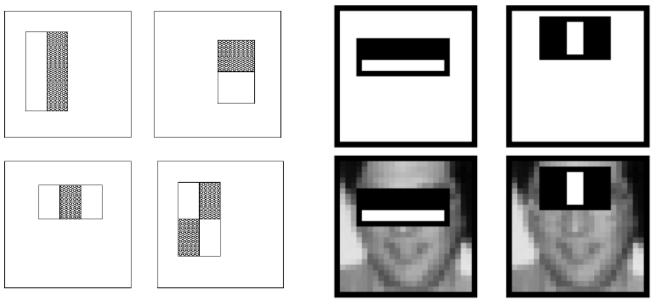
- Let A,B,C,D be the values of the integral image at the corners of a rectangle
- Then the sum of original image values within the rectangle can be computed as:

  sum = A – B – C + D

- Only 3 additions are required for any size of rectangle!



Viola-Jones detector: features



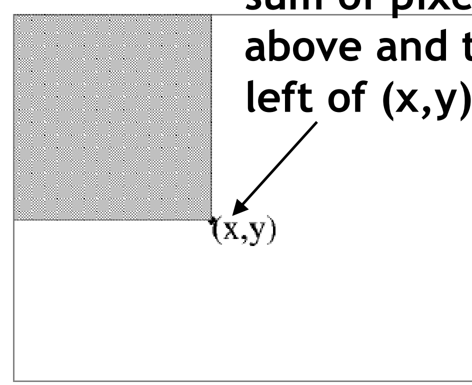
“Rectangular” filters

Feature output is difference between adjacent regions

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

Avoid scaling images → scale features directly for same cost

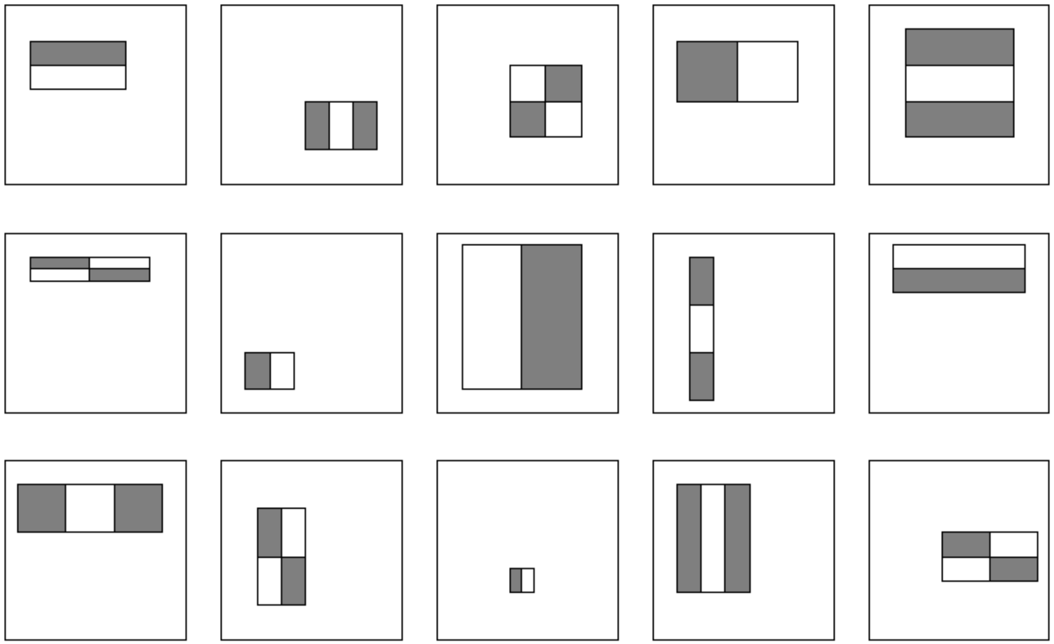
Value at (x,y) is sum of pixels above and to the left of (x,y)



Integral image

Kristen Grauman

Viola-Jones detector: features



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

180,000+ possible features associated with each 24 x 24 window

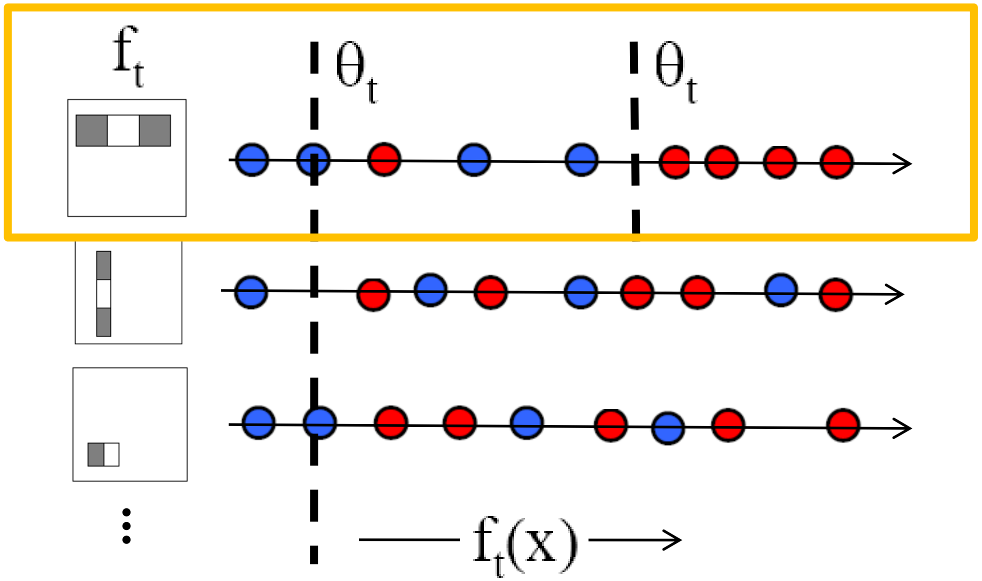
Which subset of these features should we use to determine if a window has a face?

Use AdaBoost both to select the informative features and to form the classifier

Kristen Grauman

Viola-Jones detector: AdaBoost

- **Want to select the single rectangle feature and threshold that best separates positive (faces) and negative (non-faces) training examples, in terms of *weighted* error.**



Outputs of a possible rectangle feature on faces and non-faces.

Resulting weak classifier:

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

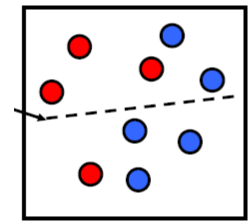
For next round, reweight the examples according to errors, choose another filter/threshold combo.

Kristen Grauman

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

AdaBoost Algorithm

Start with uniform weights on training examples



$\{x_1, \ldots x_n\}$

For T rounds

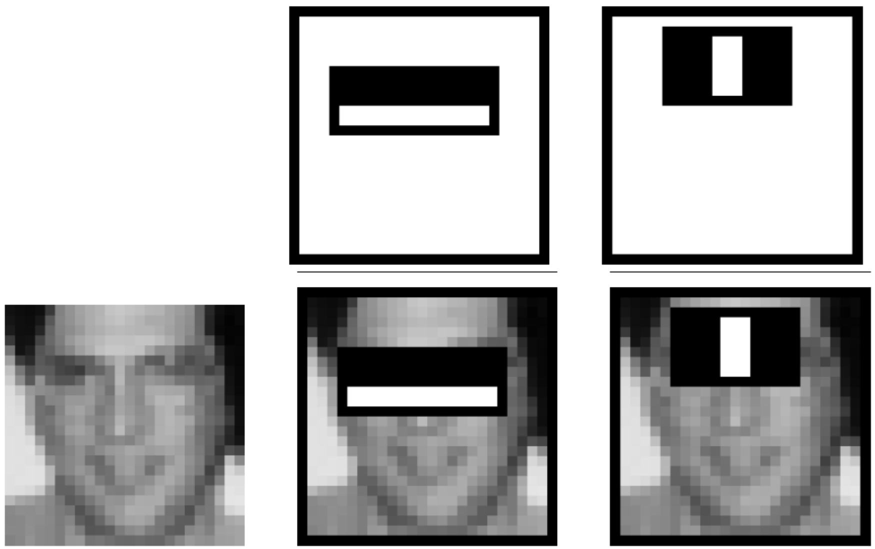
Evaluate *weighted* error for each feature, pick best.

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

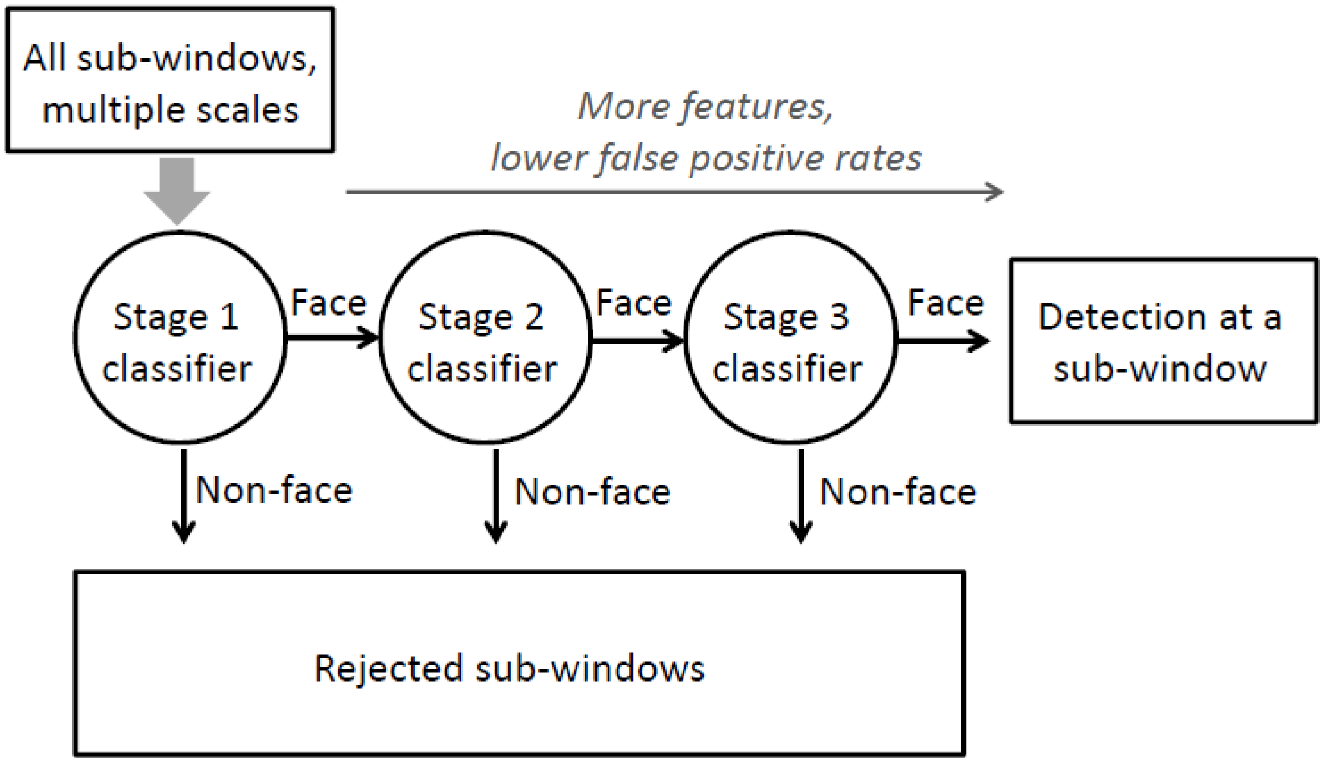
Viola-Jones Face Detector: Results



First two features selected

- 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?

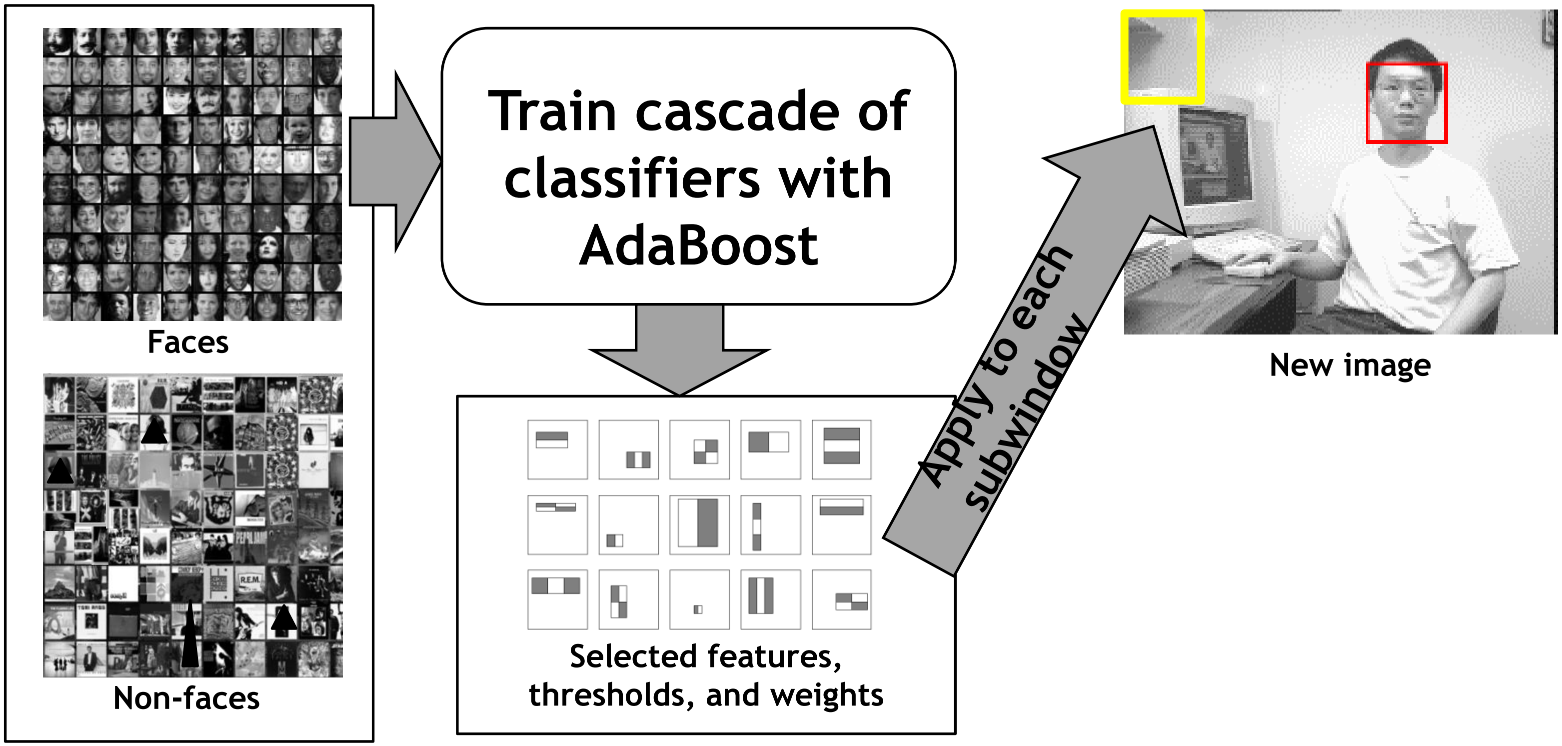
Cascading classifiers for detection



- 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 detector: summary



Train with 5K positives, 350M negatives

Real-time detector using 38 layer cascade

6061 features in all layers

[Implementation available in OpenCV: http://www.intel.com/technology/computing/opencv/]

Kristen Grauman

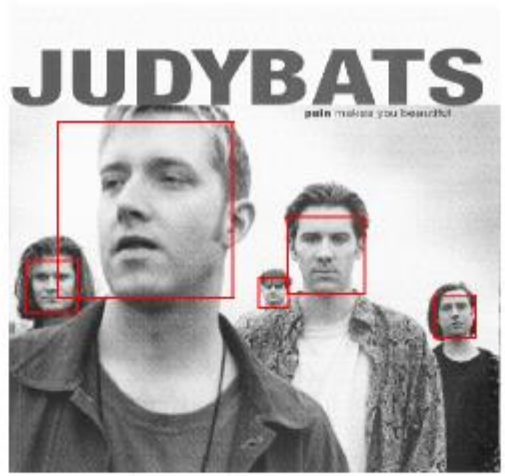
Viola-Jones detector: summary

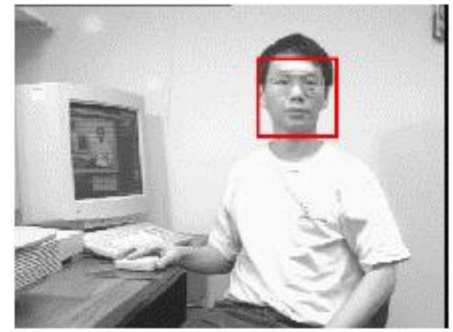
- A seminal approach to real-time object detection
- Training is slow, but detection is very fast
- Key ideas
  - Features which can be evaluated very quickly with *Integral Images*
  - Cascade model which rejects unlikely faces quickly
  - Mining hard negatives

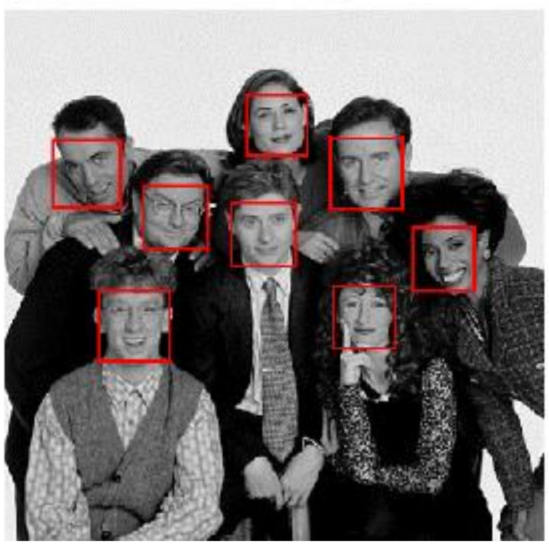
P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features.* CVPR 2001.

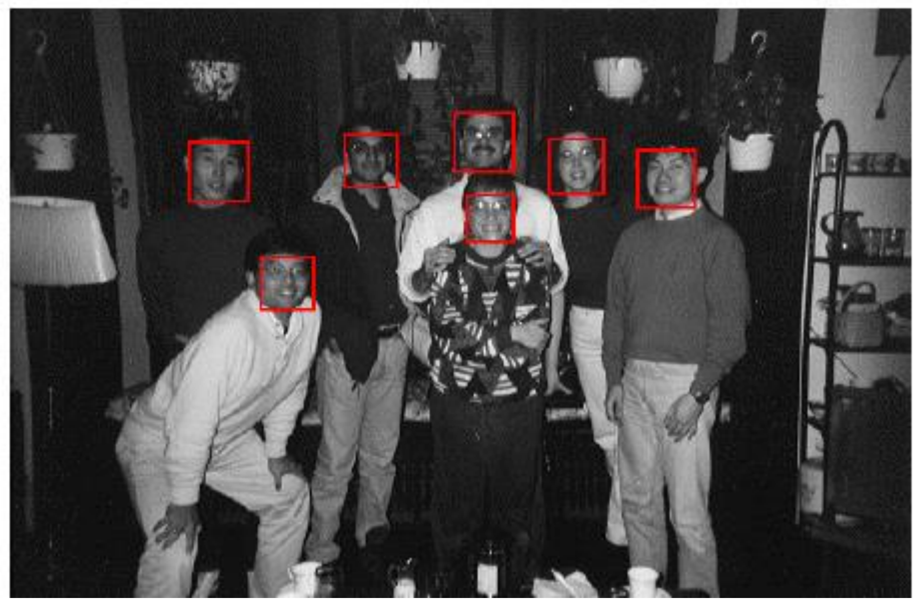
P. Viola and M. Jones. *Robust real-time face detection.* IJCV 57(2), 2004.

Viola-Jones Face Detector: Results

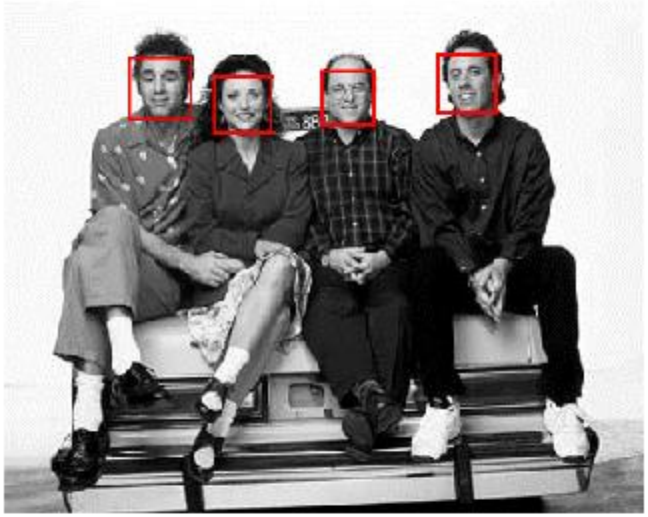


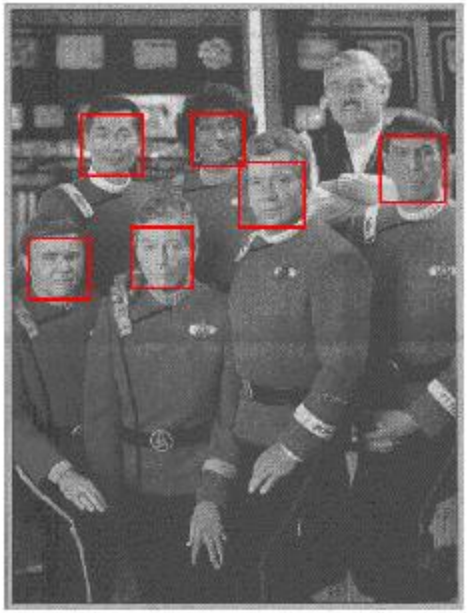


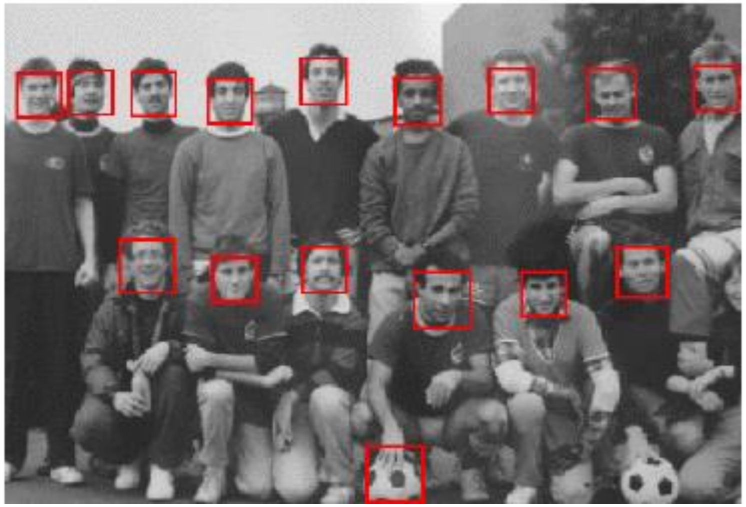


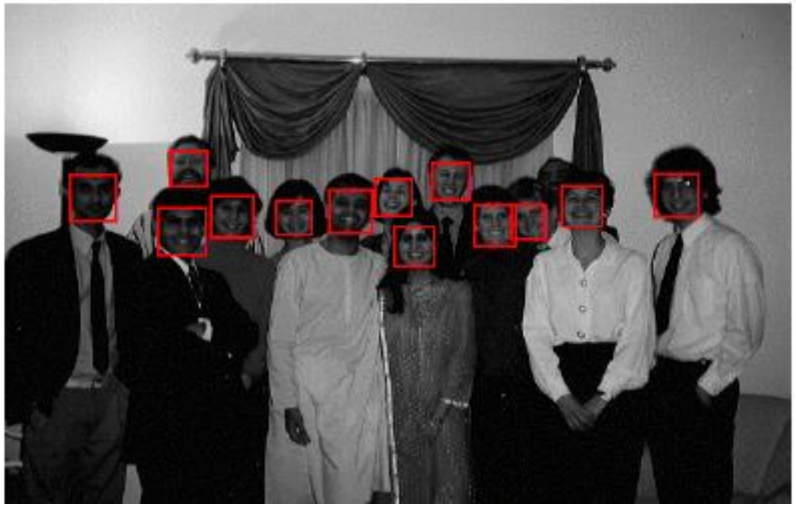


Viola-Jones Face Detector: Results





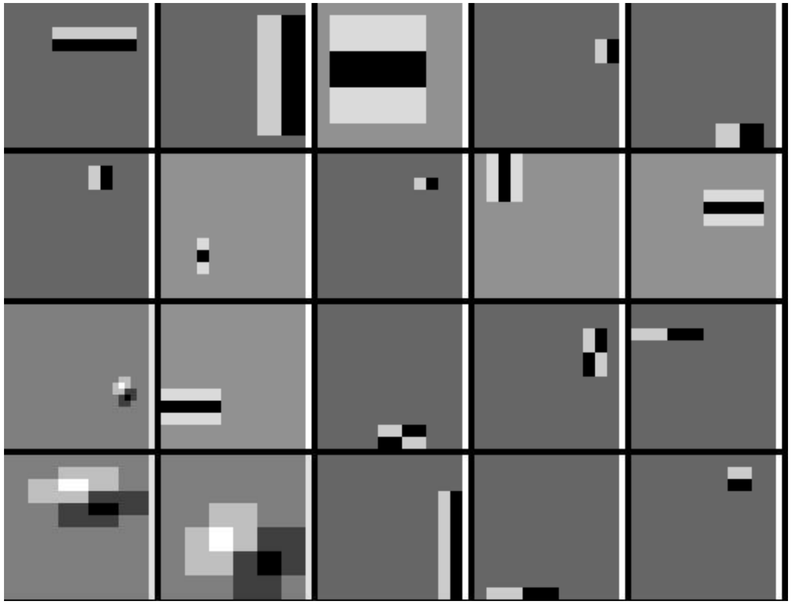




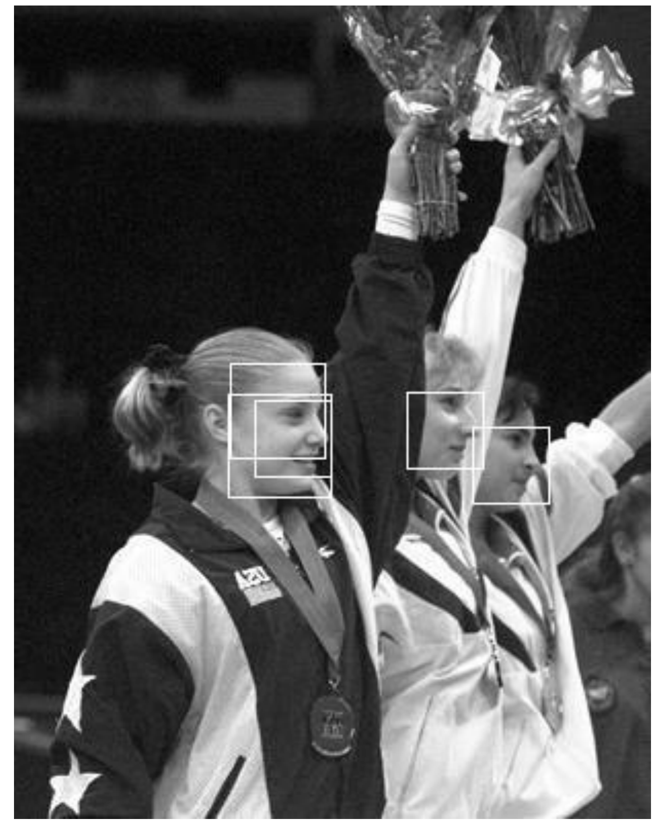
Viola-Jones Face Detector: Results

Detecting profile faces?

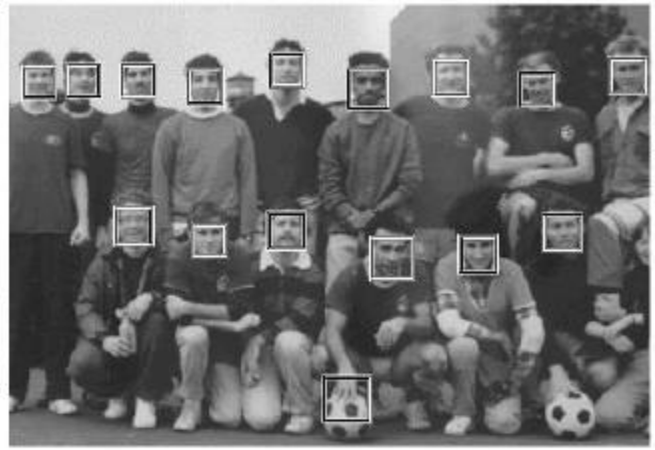
Can we use the same detector?



Viola-Jones Face Detector: Results



Viola Jones Results


Detector \ False detections	10	31	50	65	78	95	167
Viola-Jones	76.1%	88.4%	91.4%	92.0%	92.1%	92.9%	93.9%
Viola-Jones (voting)	81.1%	89.7%	92.1%	93.1%	93.1%	93.2 %	93.7%
Rowley-Baluja-Kanade	83.2%	86.0%	-	-	-	89.2%	90.1%
Schneiderman-Kanade	-	-	-	94.4%	-	-	-
Roth-Yang-Ahuja	-	-	-	-	(94.8%)	-	-

MIT + CMU face dataset

Slide: Derek Hoiem

Schneiderman later results

Schneiderman 2004

Viola-Jones 2001

Roth et al. 1999

Schneiderman-Kanade 2000

	89.7%	93.1%	94.4%	94.8%	95.7%
Bayesian Network *	1	8	19	36	56
Semi-Naïve Bayes*	6	19	29	35	46
[6]	31	65	--	--	--
[7]*	--	--	--	78	--
[16]*	--	--	65	--	--

Table 2. False alarms as a function of recognition rate on the MIT-CMU Test Set for Frontal Face Detection. * indicates exclusion of the 5 images of hand-drawn faces.

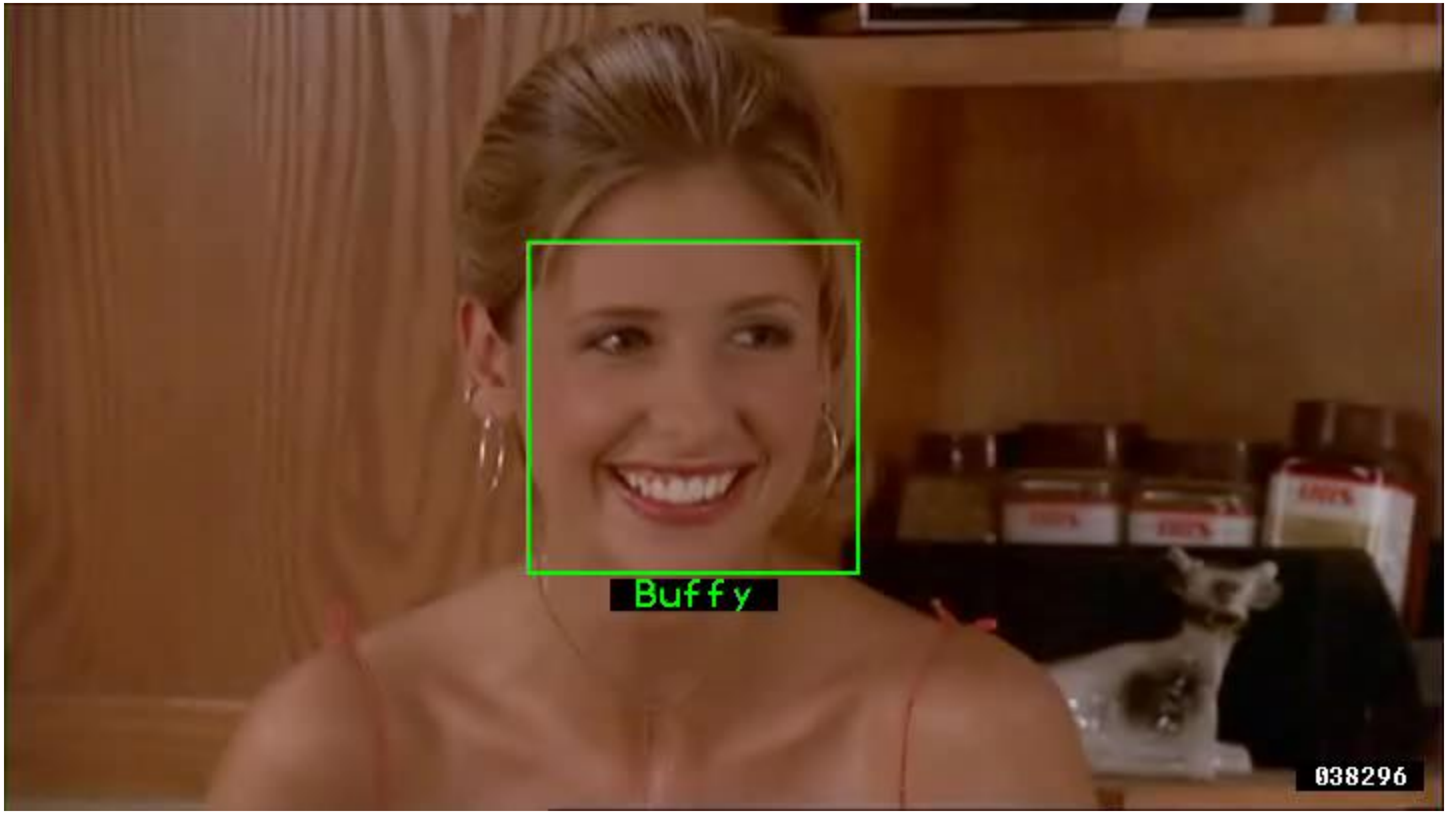
Slide: Derek Hoiem

Speed: frontal face detector

- Schneiderman-Kanade (2000): 5 seconds
- Viola-Jones (2001): 15 fps

Slide: Derek Hoiem

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

ZDNet Asia
Where Technology Means Business

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."

Consumer application: iPhoto 2009

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

Slide credit: Lana Lazebnik

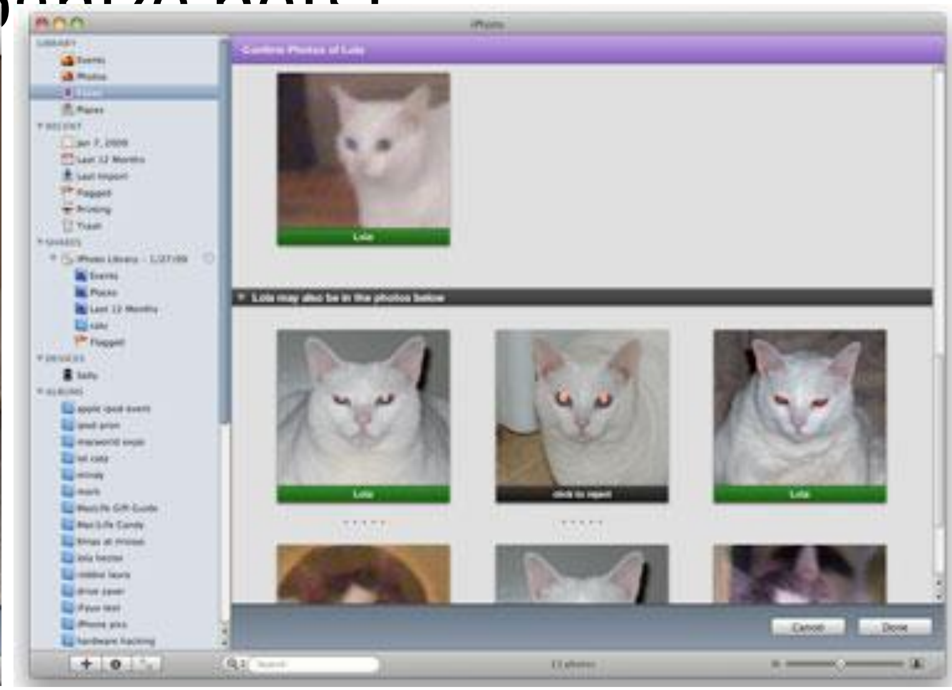
Consumer application: iPhoto 2009

- 

Slide credit: Lana Lazebnik

Consumer application: iPhoto 2009

- Can be trained to recognize pets!

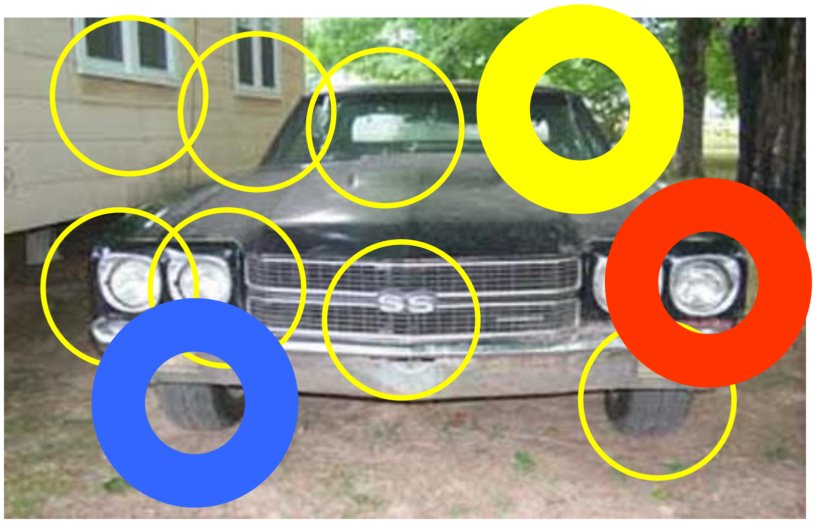


http://www.maclife.com/article/news/iphotos_faces_recognizes_cats

Slide credit: Lana Lazebnik

- Part-based and local feature models for generic object recognition

Part-based and local feature models for recognition



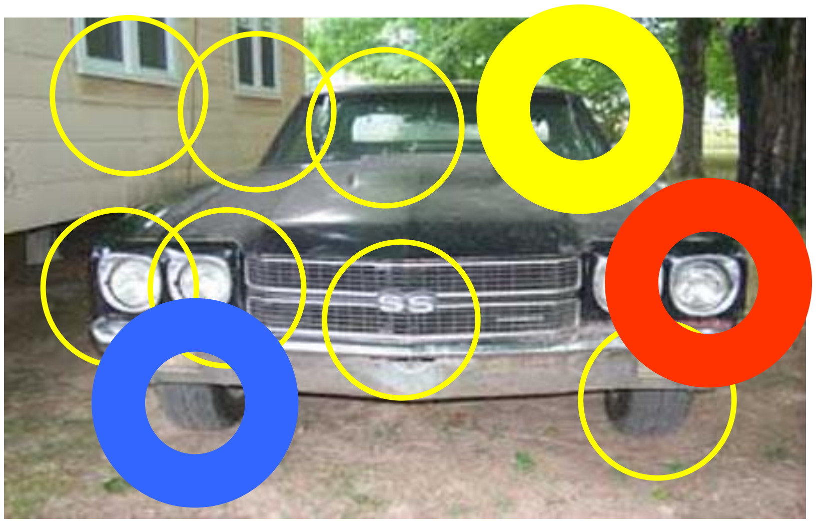
Main idea:

Rather than a representation based on holistic appearance, decompose the image into:

- local parts or patches, and
- their relative spatial relationships

Kristen Grauman

Part-based and local feature models for recognition



We'll look at three forms:

1. **Bag of words** (no geometry)
2. **Implicit shape model** (star graph for spatial model)
3. **Constellation model** (fully connected graph for spatial model)

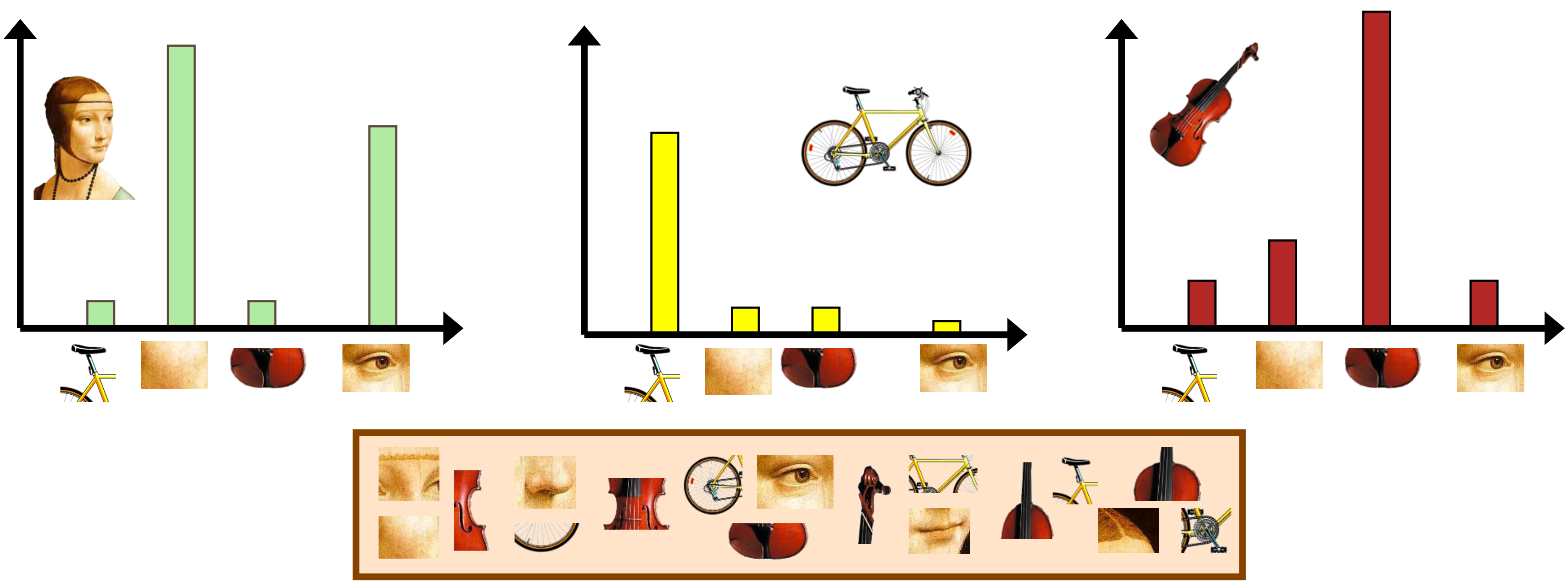
Kristen Grauman

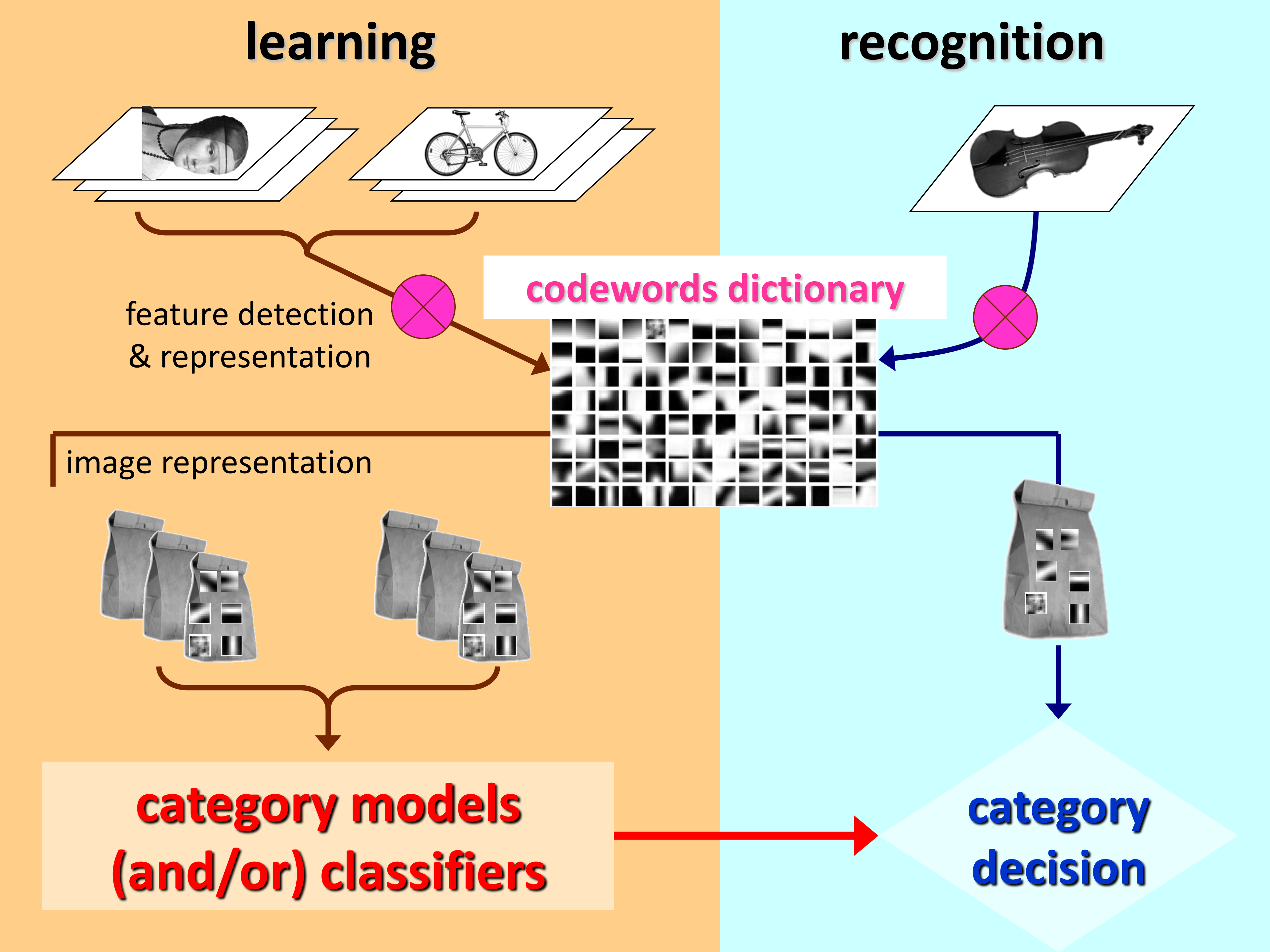
Bag of Words Models

Object → **Bag of ‘words’**

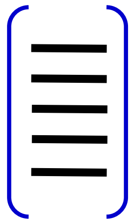
Bag of Words

- Independent features
- Histogram representation



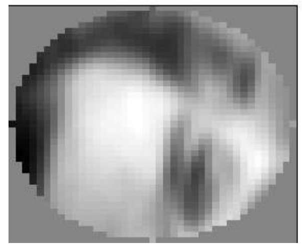

learning
recognition
codewords dictionary
feature detection
& representation
image representation
category models
(and/or) classifiers
category
decision

1.Feature detection and representation

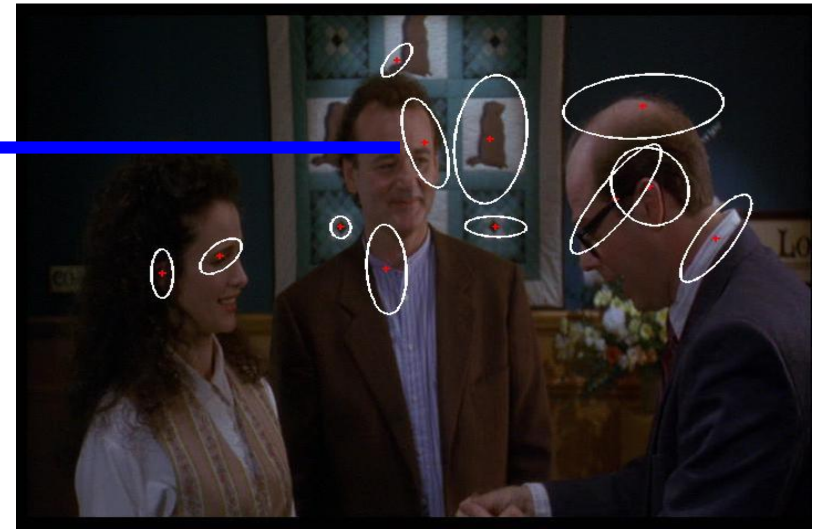


Compute descriptor

e.g. **SIFT** [Lowe’99]



Normalize patch



Detect patches

[Mikojaczyk and Schmid ’02]

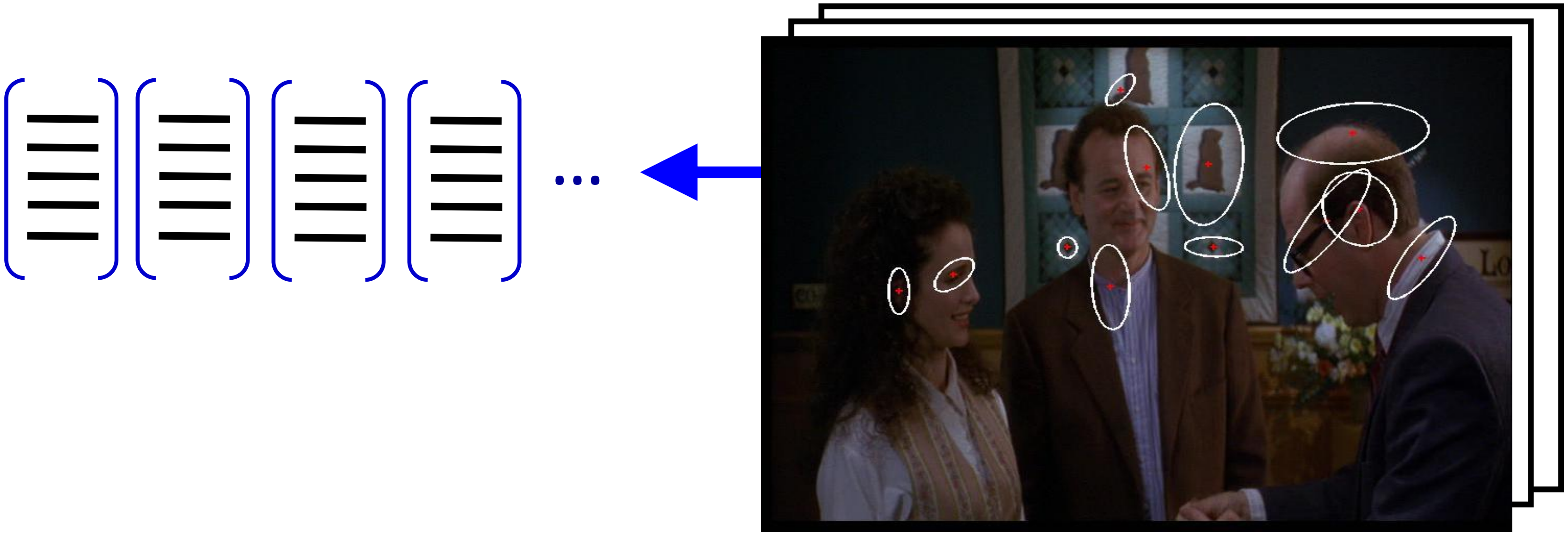
[Mata, Chum, Urban & Pajdla, ’02]

[Sivic & Zisserman, ’03]

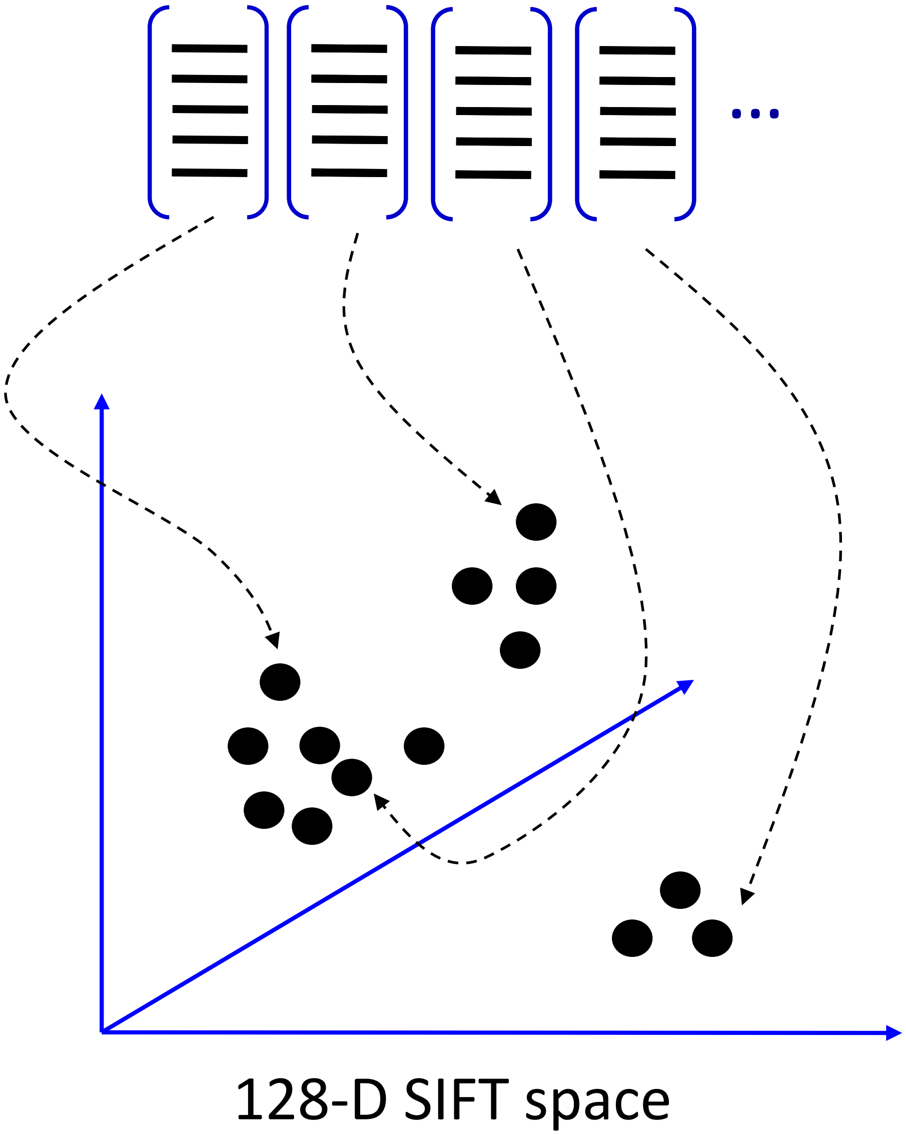
Local interest operator
or
Regular grid

Slide credit: Josef Sivic

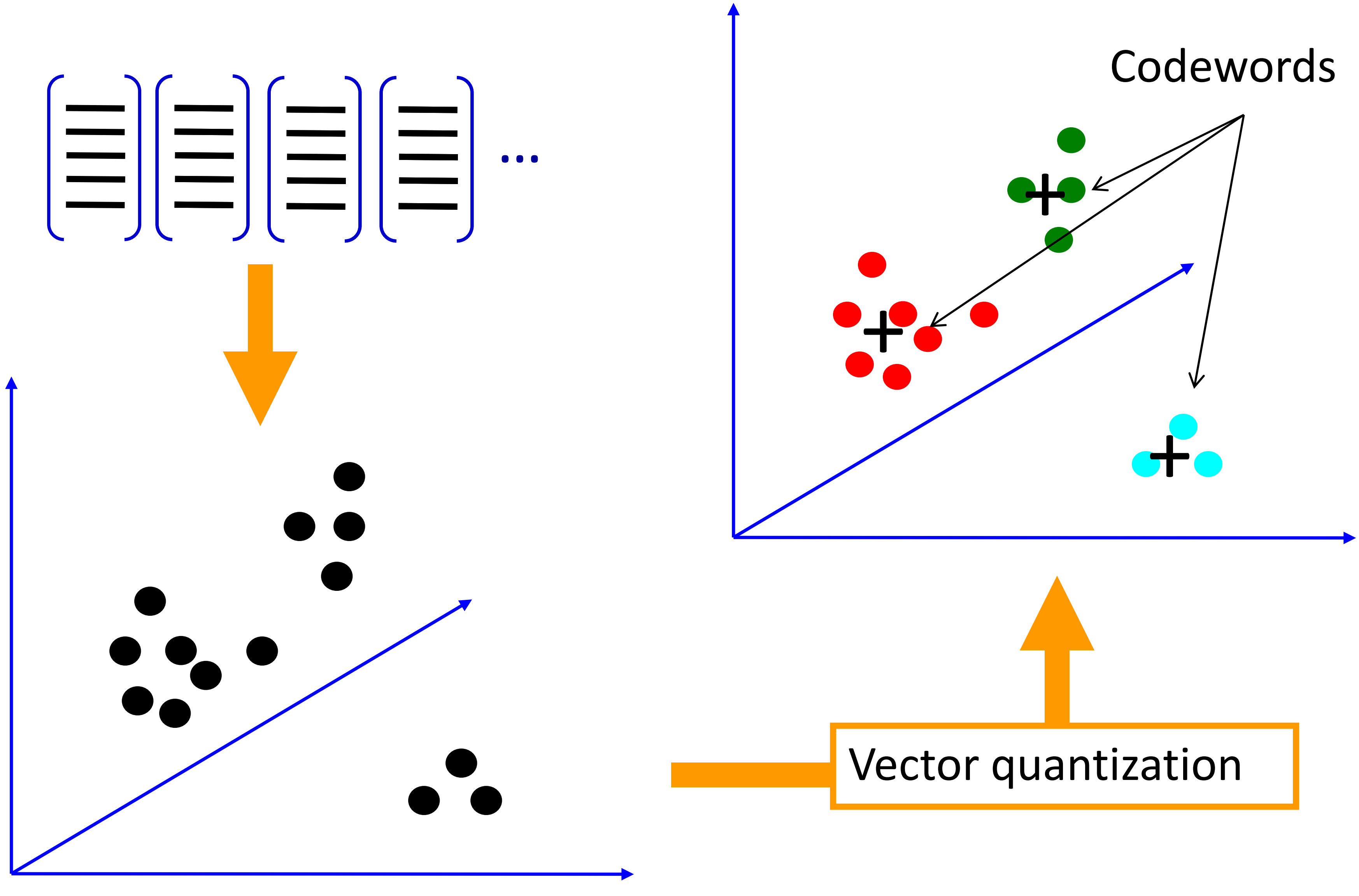
1.Feature detection and representation



2. Codewords dictionary formation

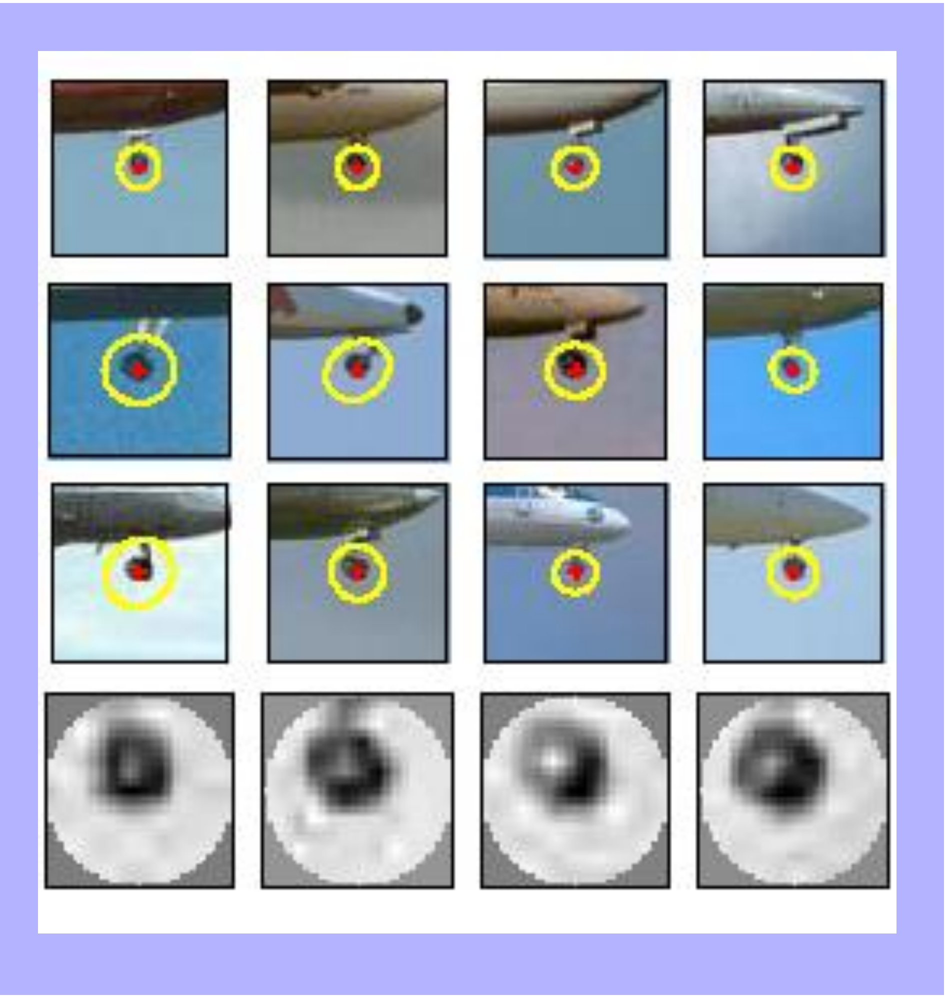


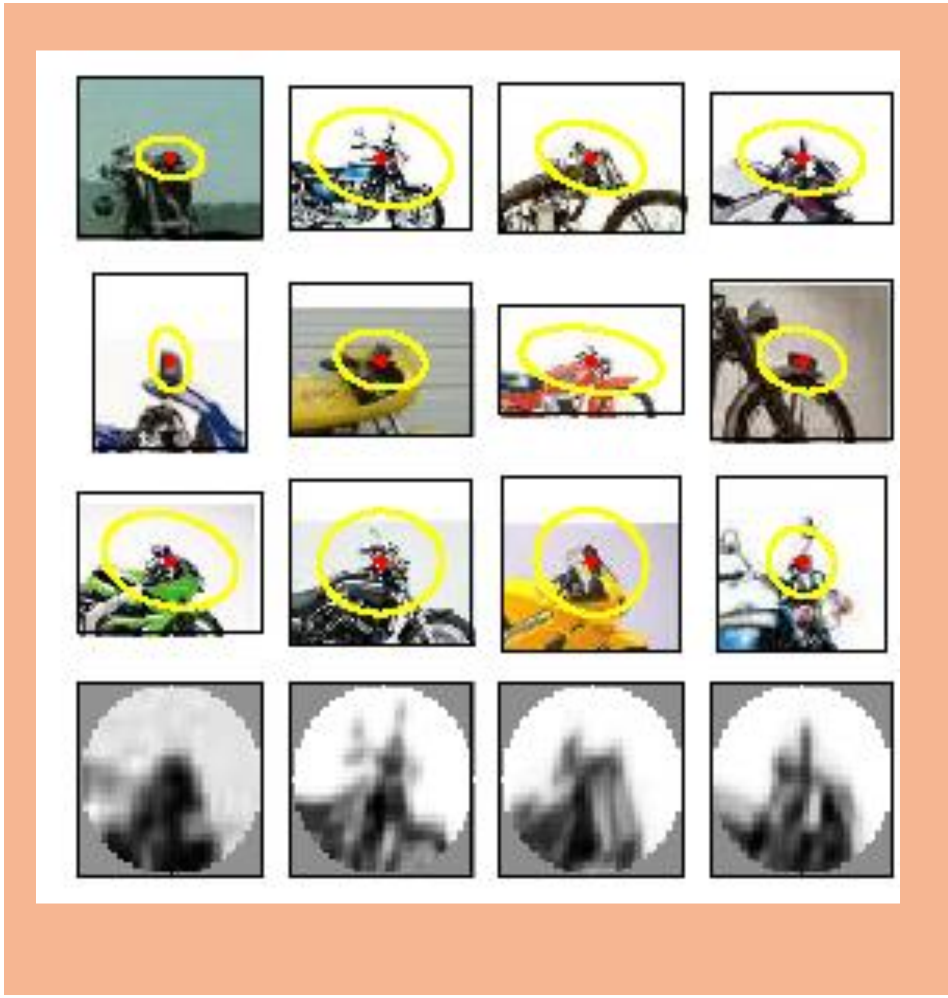
2. Codewords dictionary formation



Slide credit: Josef Sivic

Image patch examples of codewords

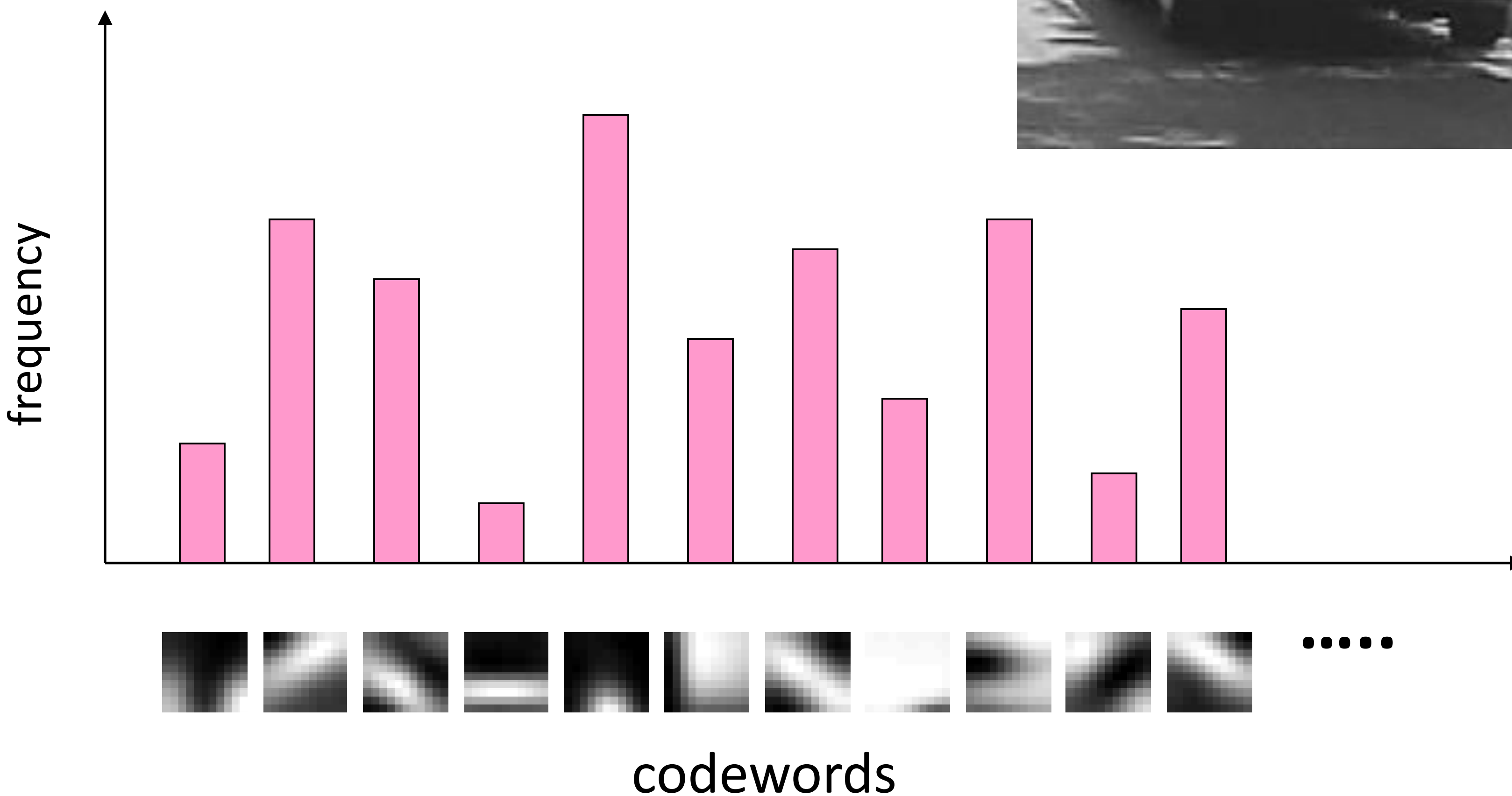




Sivic et al. 2005

Image representation

Histogram of features assigned to each cluster

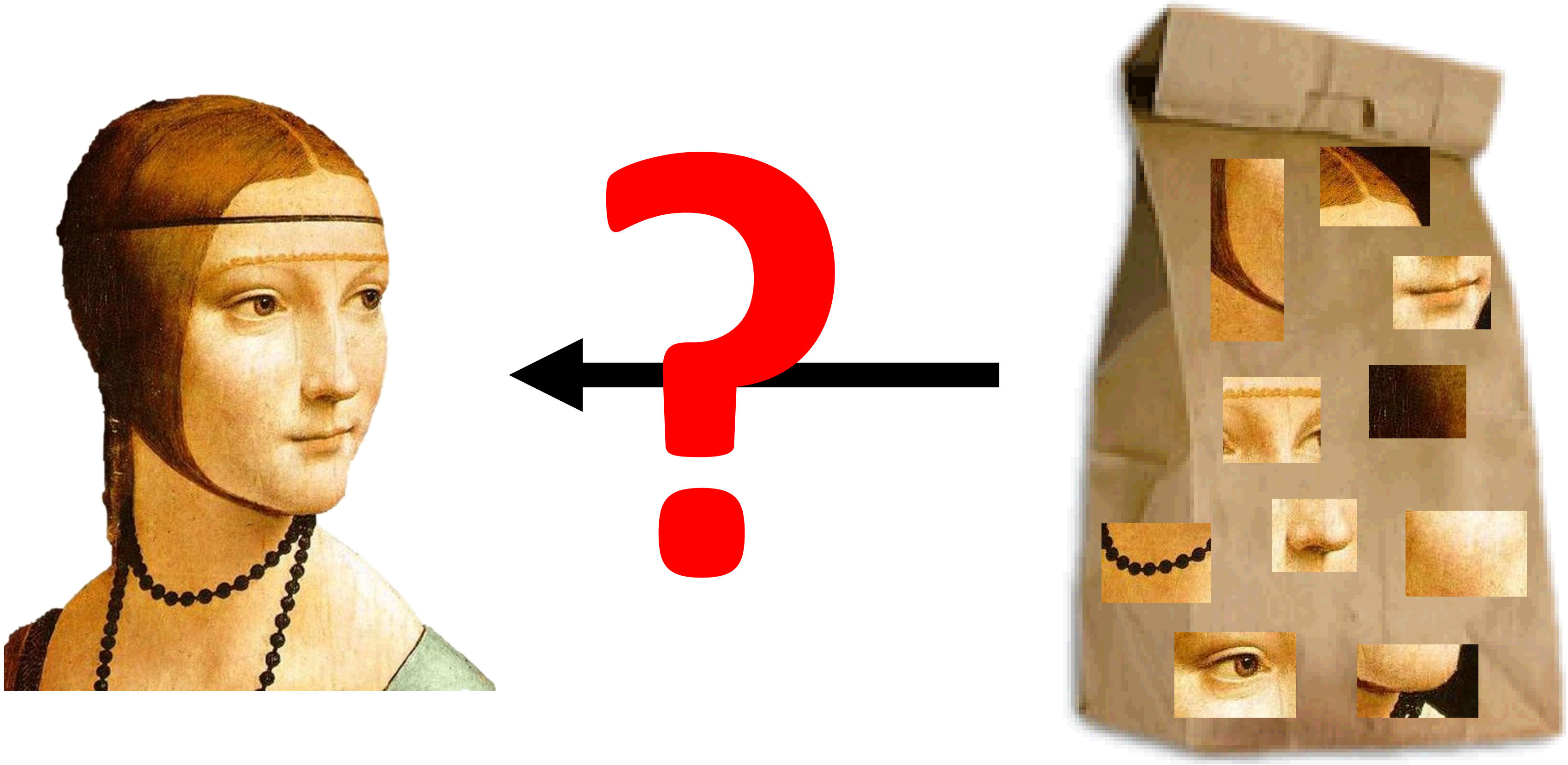


Uses of BoW representation

- Treat as feature vector for standard classifier
  - e.g SVM
- Cluster BoW vectors over image collection
  - Discover visual themes
- Hierarchical models
  - Decompose scene/object

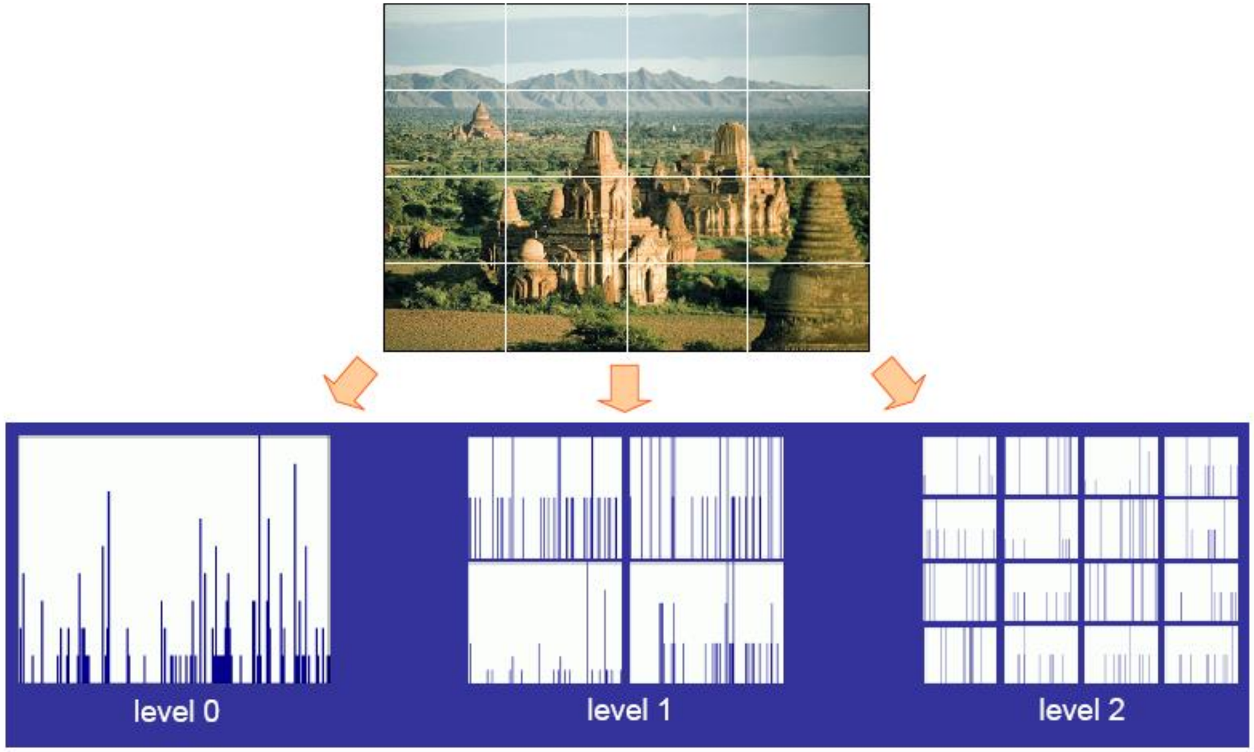
- Scene

What about spatial info?



Adding spatial info. to BoW

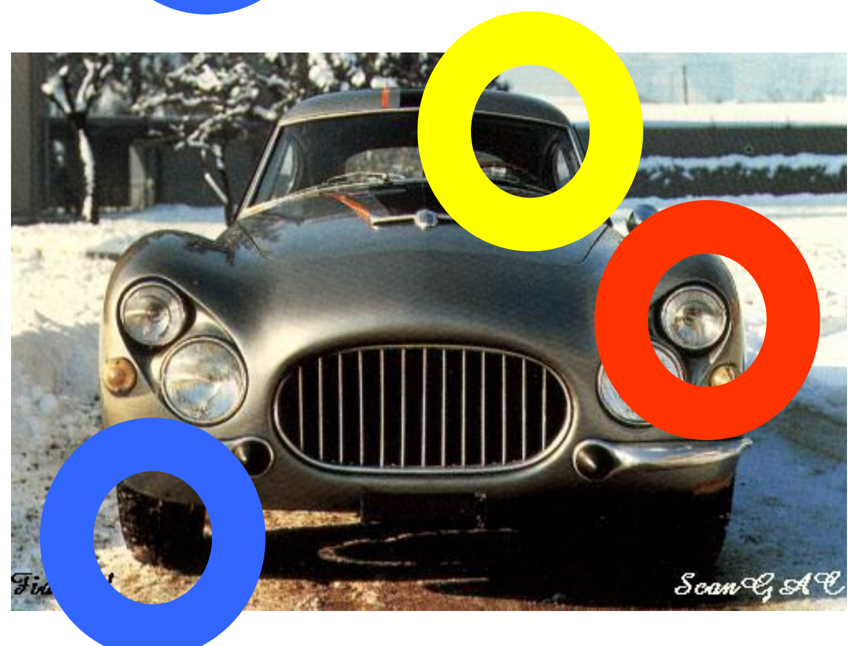
- Feature level
- Generative models
- Discriminative methods
  - Lazebnik, Schmid & Ponce, 2006



Problem with bag-of-words

- All have equal probability for bag-of-words methods
- Location information is important
- BoW + location still doesn't give correspondence

Model: Parts and Structure



Representation

- Object as set of parts
  - Generative representation
- Model:
  - Relative locations between parts
  - Appearance of part
- Issues:
  - How to model location
  - How to represent appearance
  - How to handle occlusion/clutter

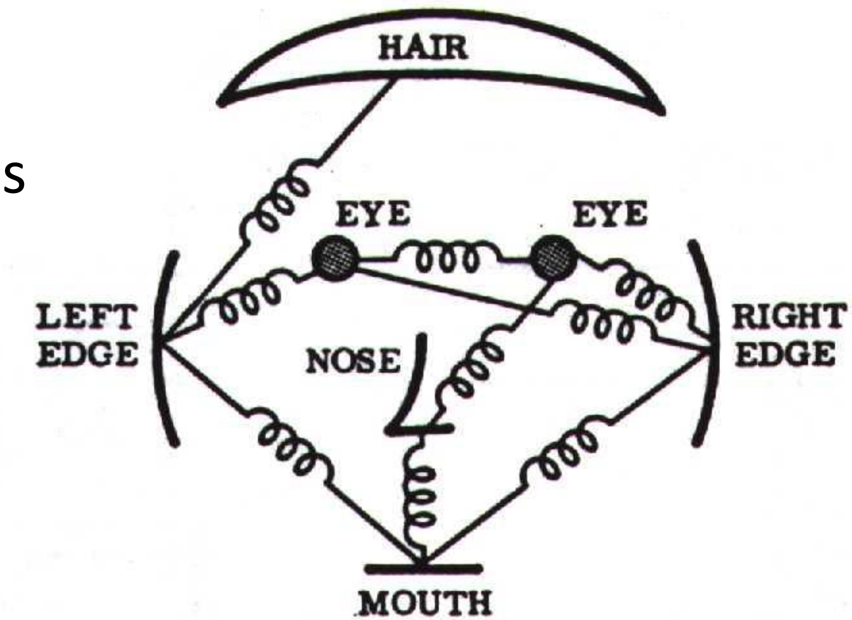
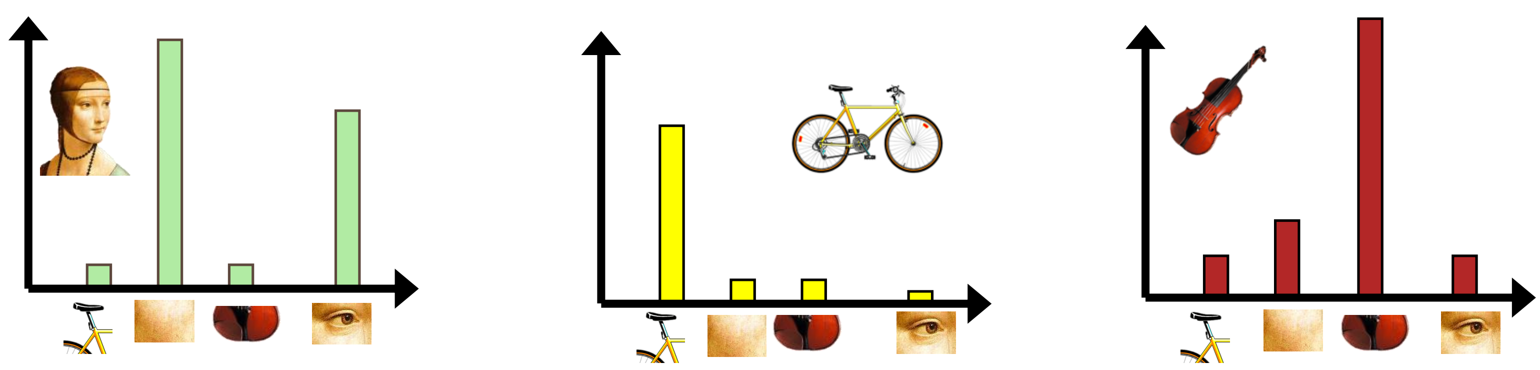


Figure from [Fischler & Elschlager 73]

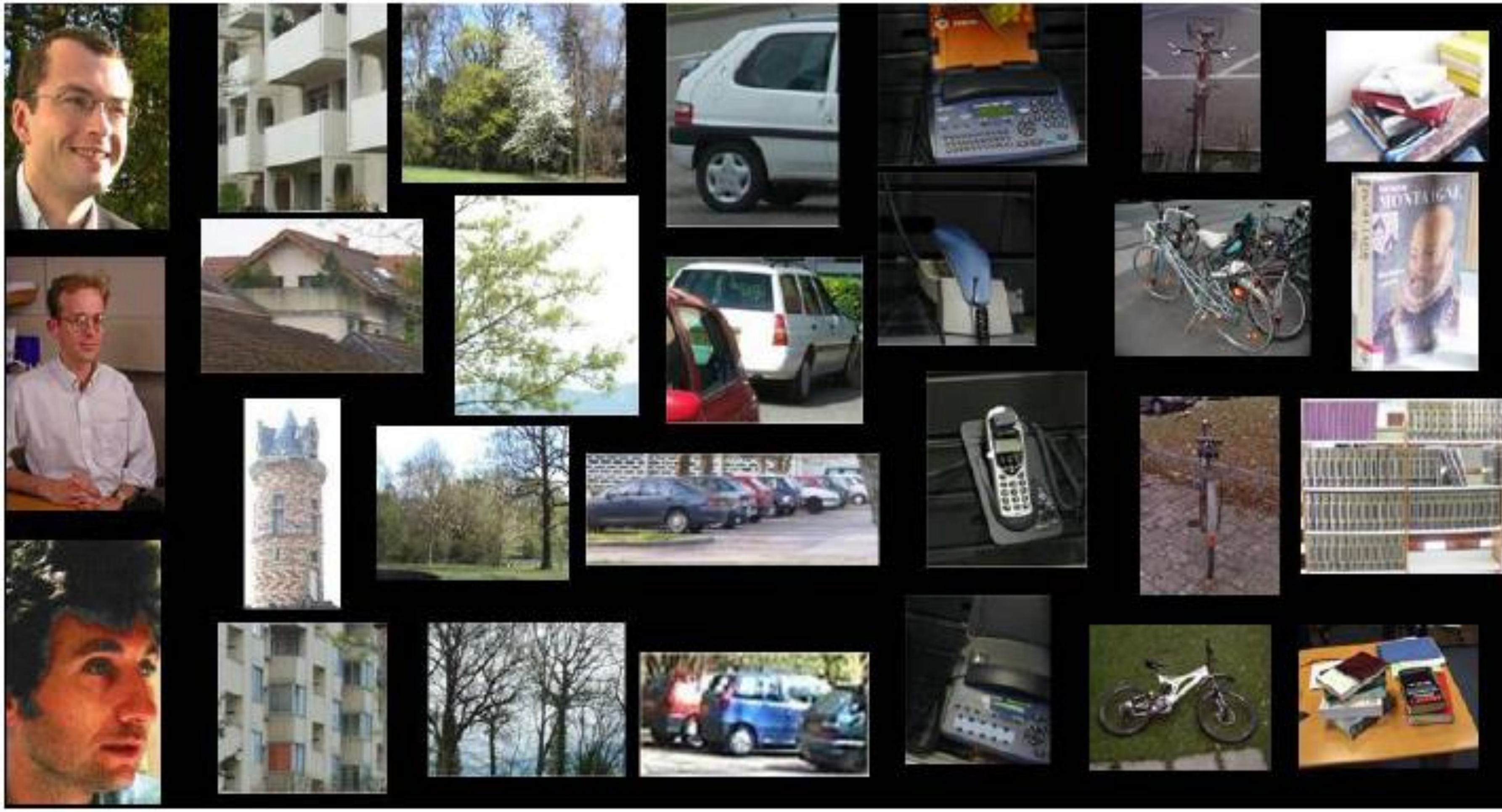
Bag-of-words model

- Summarize entire image based on its distribution (histogram) of word occurrences.
  - Total freedom on spatial positions, relative geometry.
  - Vector representation easily usable by most classifiers.



Kristen Grauman

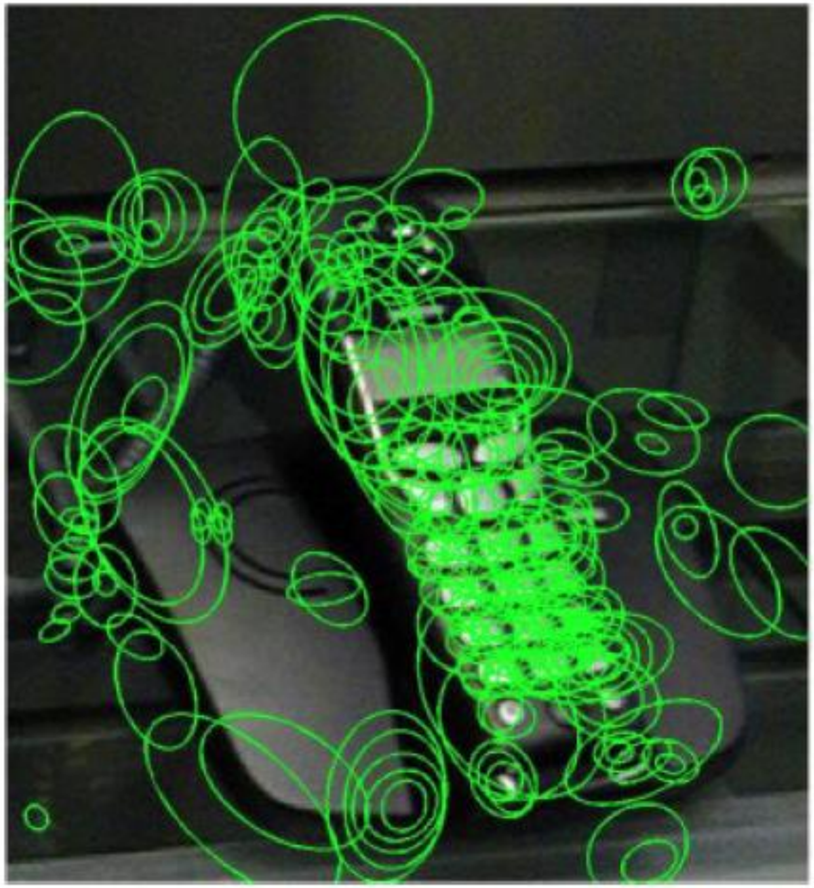
Bag-of-words model



Our in-house database contains 1776 images in seven classes[1]: faces, buildings, trees, cars, phones, bikes and books. Fig. 2 shows some examples from this dataset.

Csurka et al. Visual Categorization with Bags of Keypoints, 2004

Words as parts



All local features

Local features from two selected clusters occurring in this image

Csurka et al. 2004

Naïve Bayes model for classification

$$c^* = \arg\max_c \; p(c \mid w) \propto p(c)\, p(w \mid c) = p(c) \prod_{n=1}^{N} p(w_n \mid c)$$

Object class decision

Prior prob. of the object classes

Image likelihood given the class

N patches

What assumptions does the model make, and what are their significance?

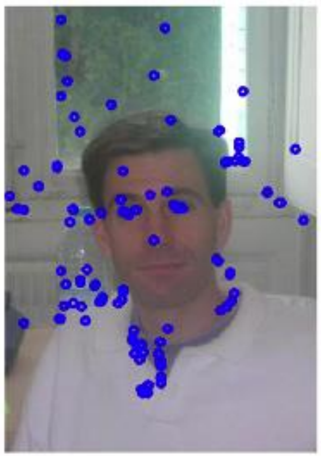
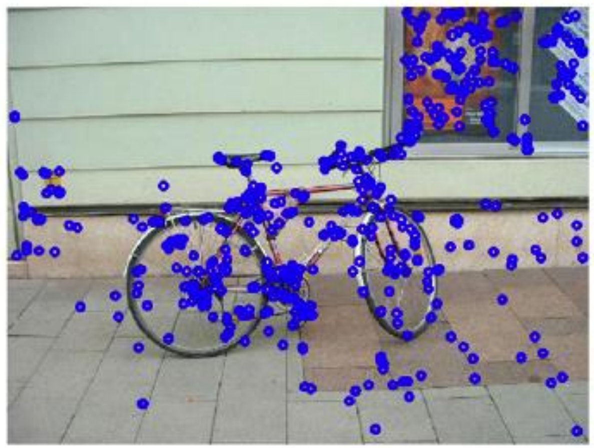
Confusion matrix

True classes →	*faces*	*buildings*	*trees*	*cars*	*phones*	*bikes*	*books*
faces	**76**	4	2	3	4	4	13
buildings	2	**44**	5	0	5	1	3
trees	3	2	**80**	0	0	5	0
cars	4	1	0	**75**	3	1	4
phones	9	15	1	16	**70**	14	11
bikes	2	15	12	0	8	**73**	0
books	4	19	0	6	7	2	**69**

Example bag of words + Naïve Bayes classification results for generic categorization of objects

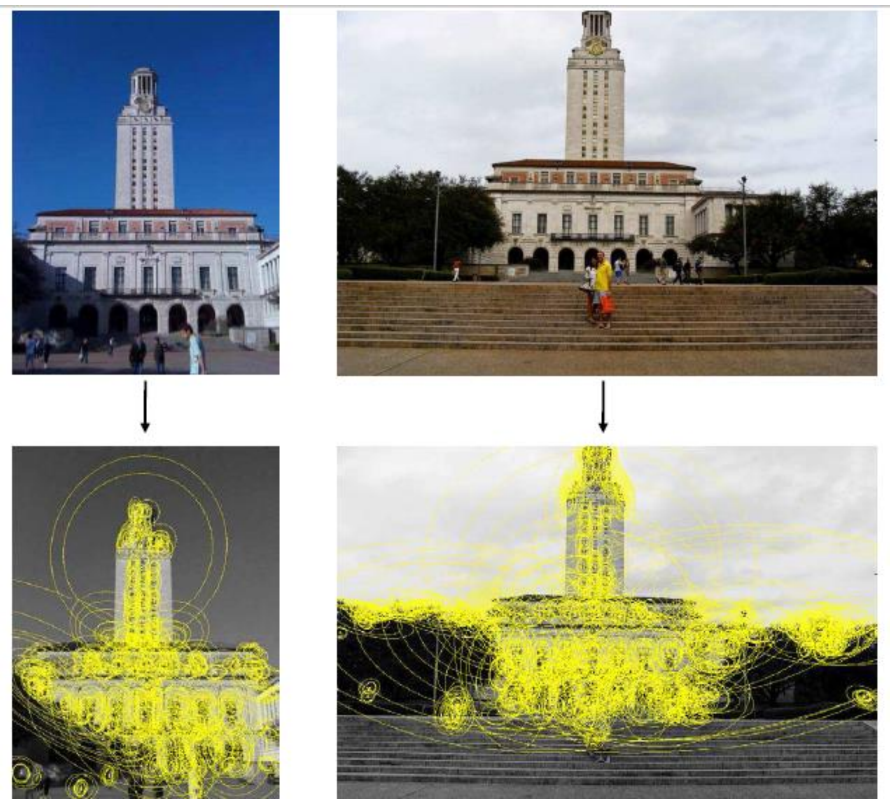
Csurka et al. 2004

Clutter…or context?

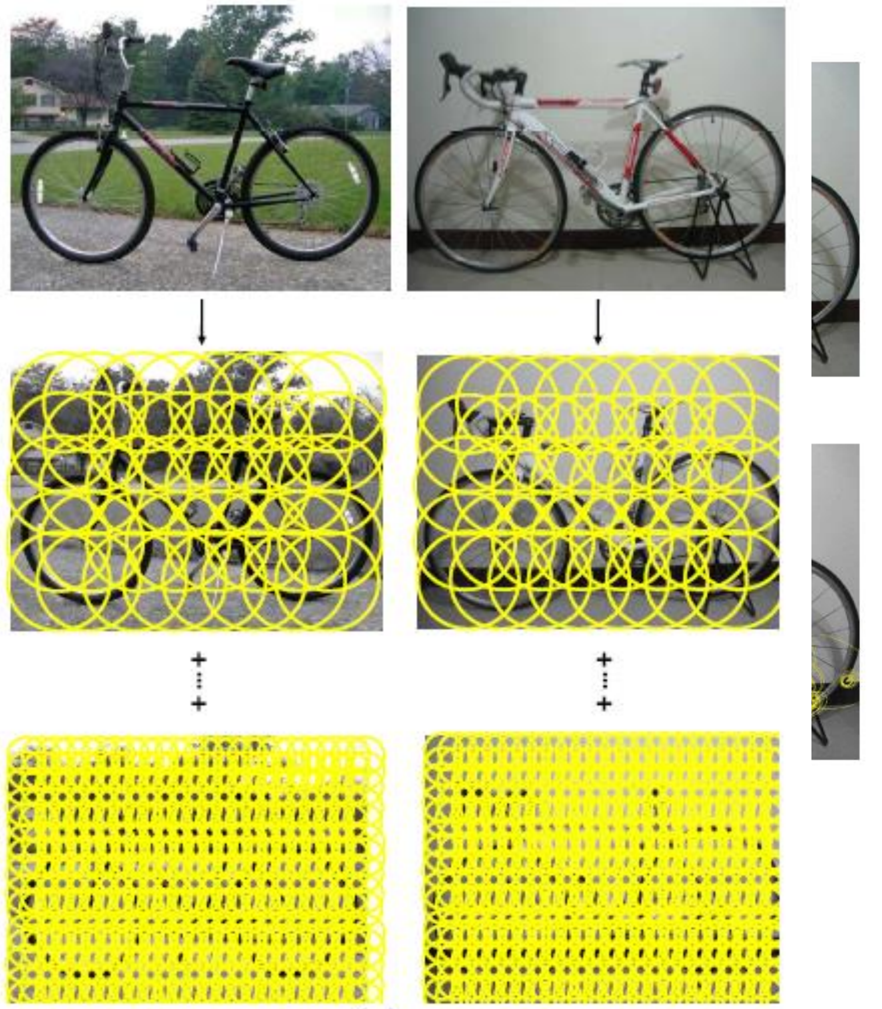
  

Kristen Grauman

Sampling strategies



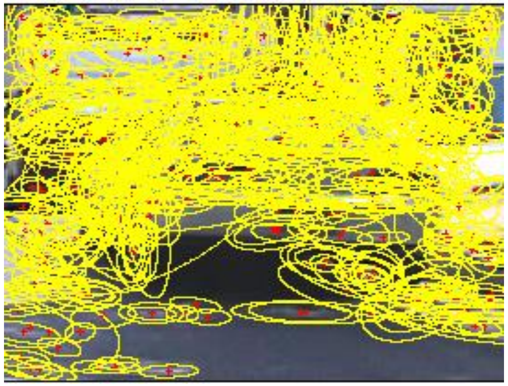
Specific object



Category

Kristen Grauman

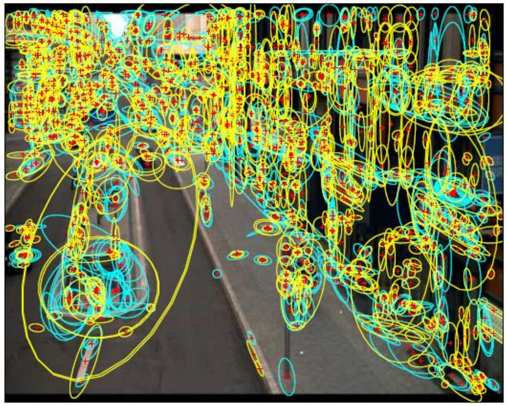
Sampling strategies



Sparse, at interest points

Dense, uniformly

Randomly



Multiple interest operators

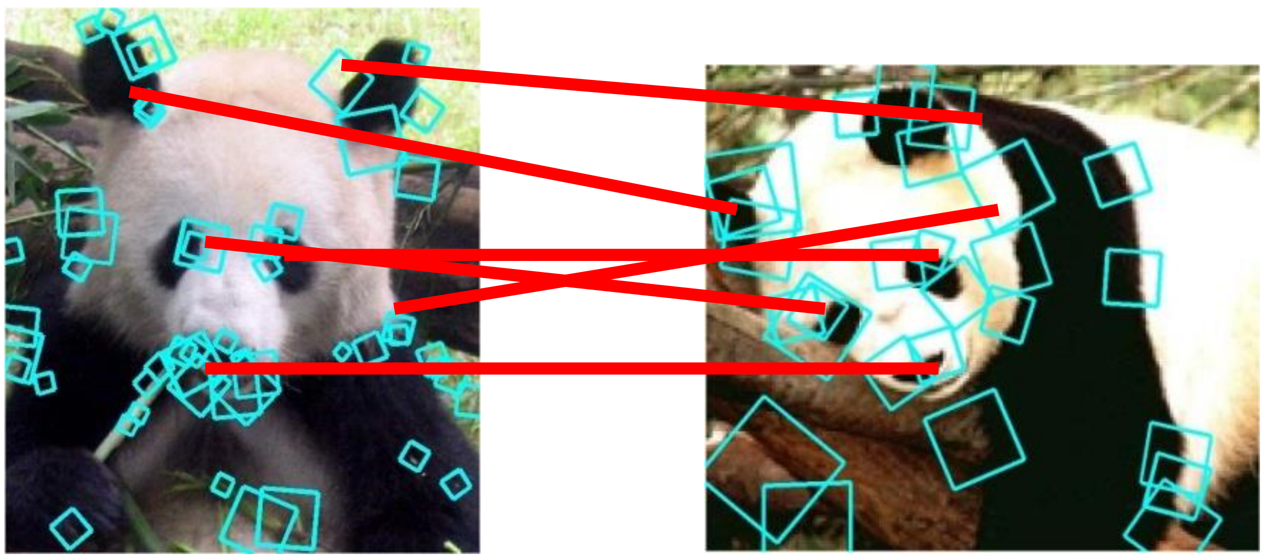
- To find specific, textured objects, sparse sampling from interest points more reliable.
- Multiple complementary interest operators offer more image coverage.
- For object categorization, dense sampling offers better coverage.

Image credits: F-F. Li, E. Nowak, J. Sivic

[See Nowak, Jurie & Triggs, ECCV 2006]

Kristen Grauman

Local feature correspondence for generic object categories

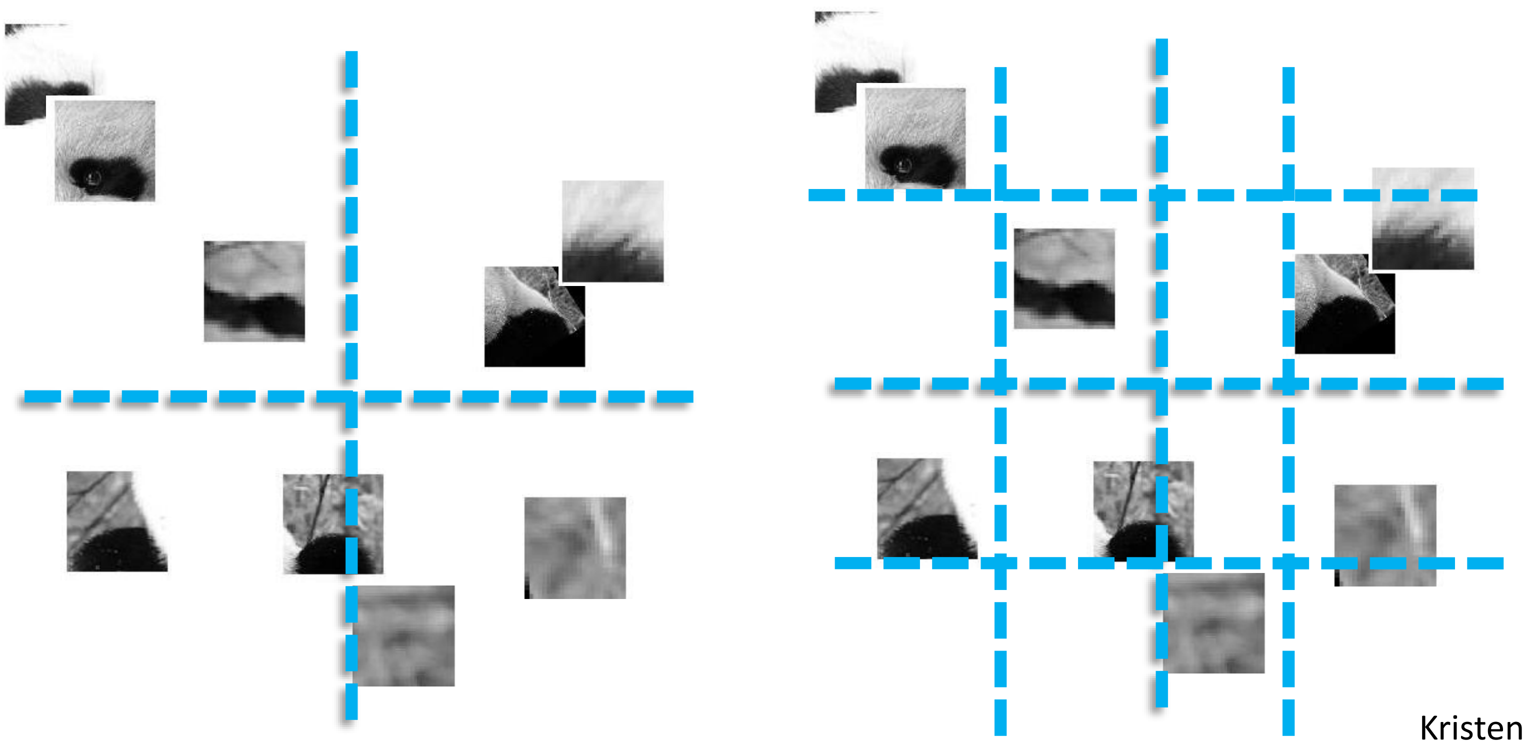


$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$ $\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$

Kristen Grauman

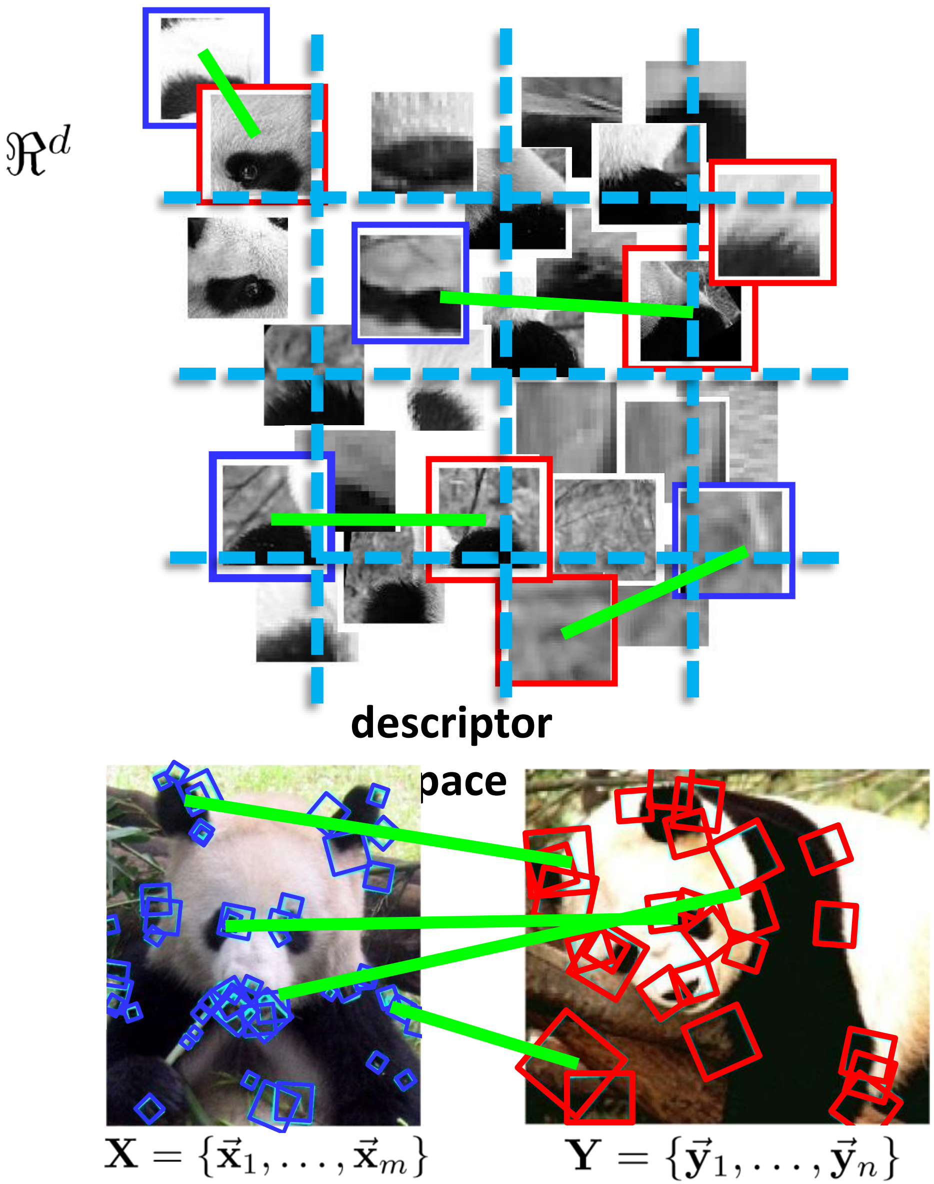
Local feature correspondence for generic object categories

- Comparing bags of words histograms coarsely reflects agreement between local "parts" (patches, words).
- *But* choice of quantization directly determines what we consider to be similar…



Kristen Grauman

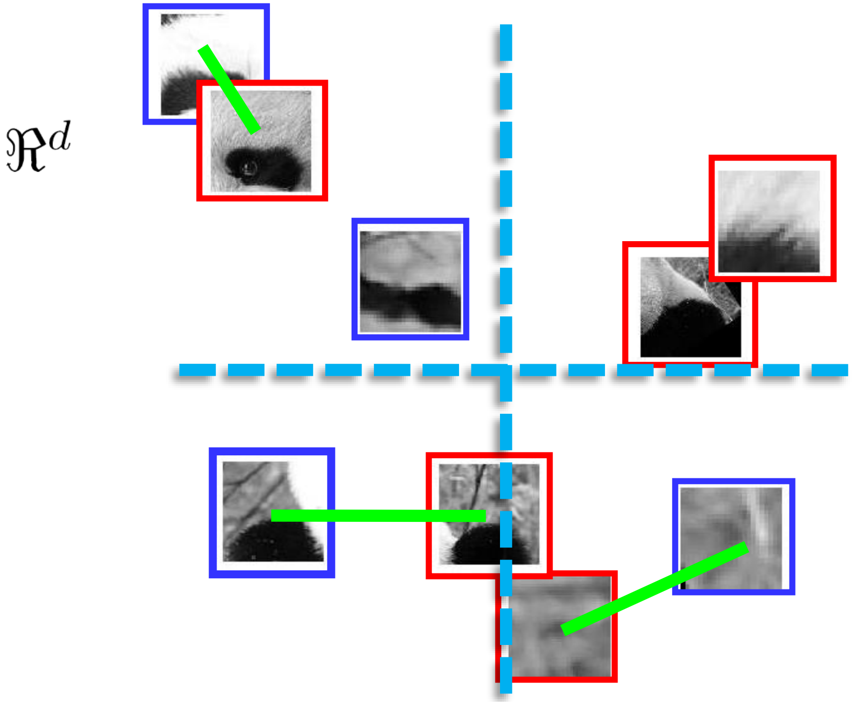
Pyramid match: main idea

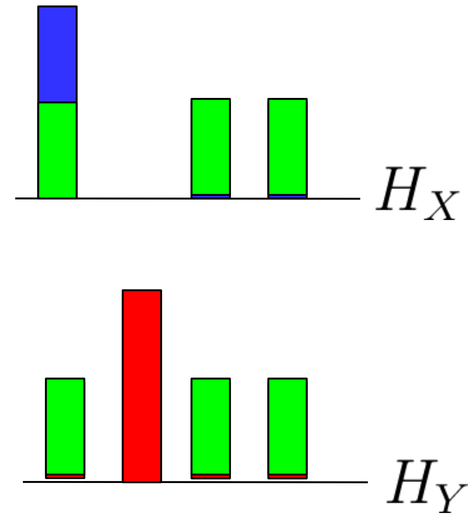


Feature space partitions serve to “match” the local descriptors within successively wider regions.

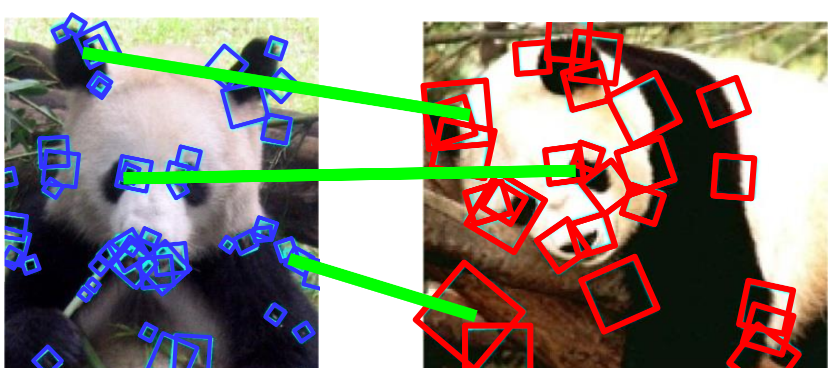
[Grauman & Darrell, ICCV 2005]

Pyramid match: main idea





$$\mathcal{I}(H_X, H_Y) = \sum_j \min\left(H_X(j), H_Y(j)\right)$$
$$= 3$$

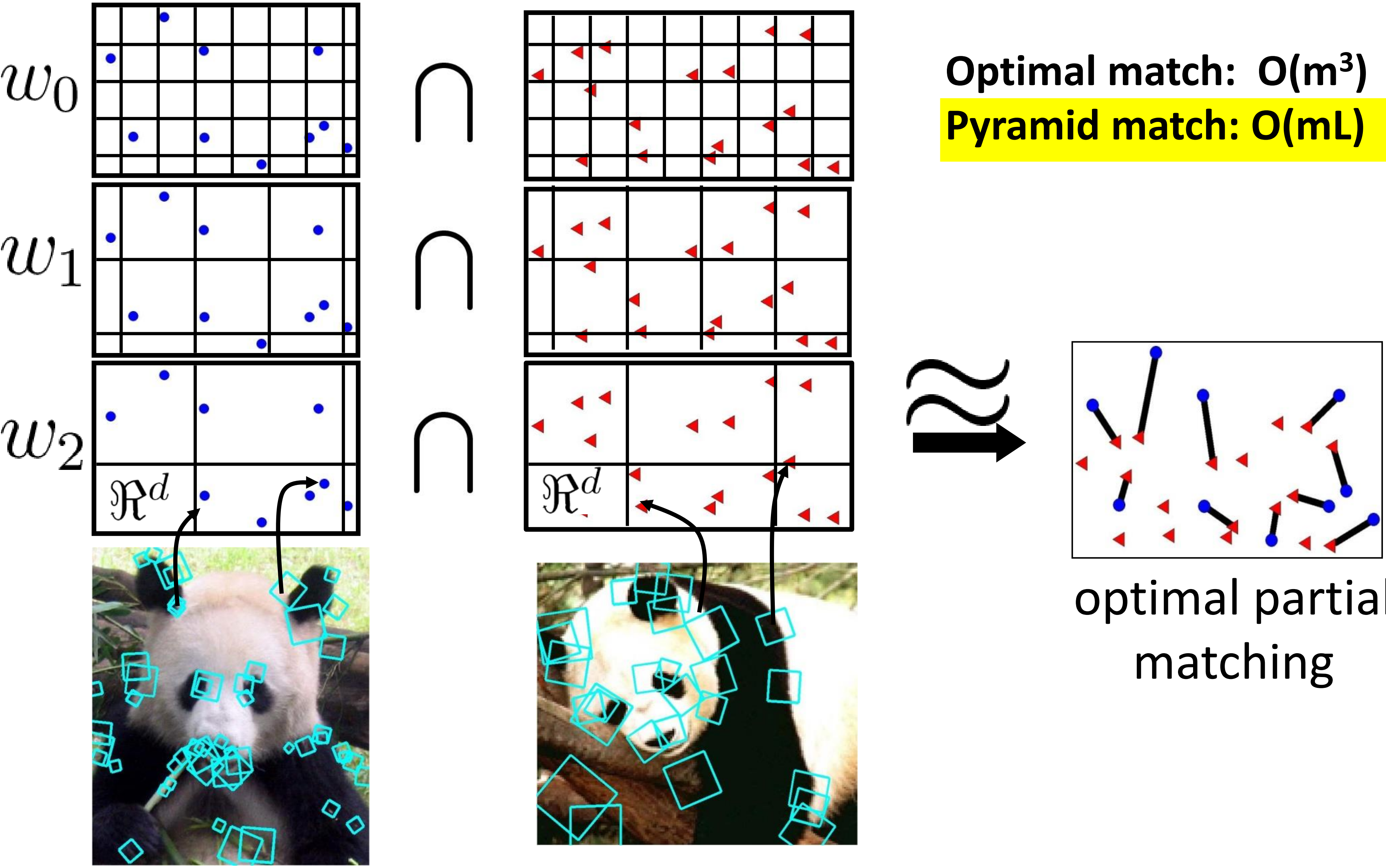


$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$ $\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$

Histogram intersection counts number of possible matches at a given partitioning.

[Grauman & Darrell, ICCV 2005]

Pyramid match kernel



Optimal match: O(m³)

Pyramid match: O(mL)

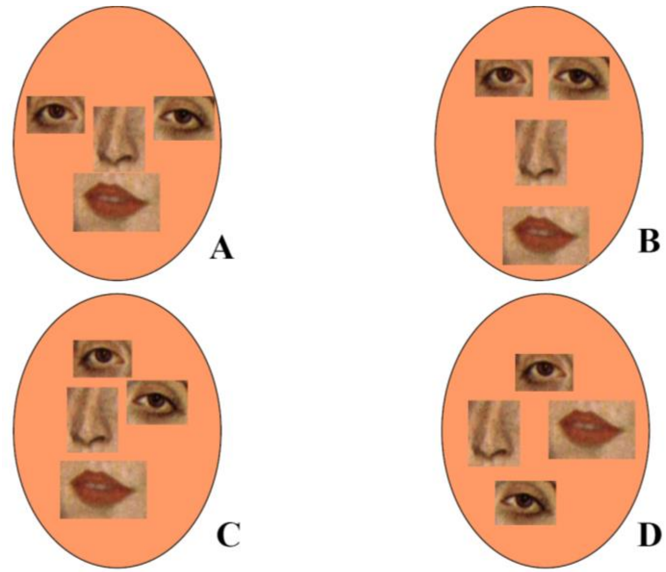
optimal partial matching

$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$ $\quad$ $\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$

[Grauman & Darrell, ICCV 2005]

Unordered sets of local features: **No** spatial layout preserved!

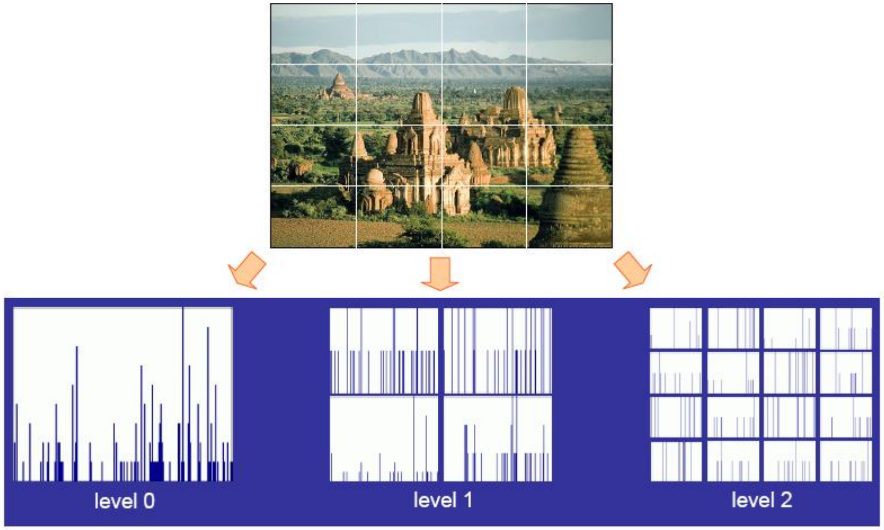
Too much?



Too little?

Spatial pyramid match

- Make a pyramid of bag-of-words histograms.
- Provides some loose (global) spatial layout information



$$K^L(X,Y) = \sum_{m=1}^{M} \kappa^L(X_m, Y_m)$$

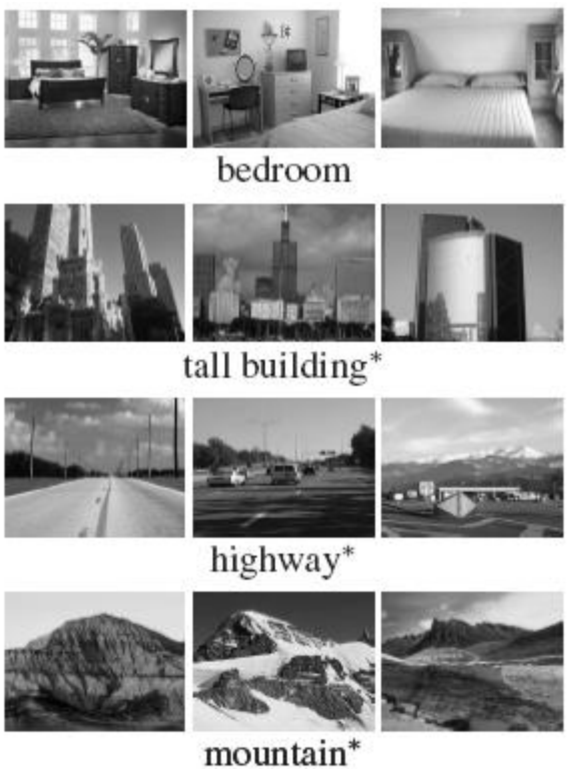
Sum over PMKs computed in *image coordinate* space, one per word.

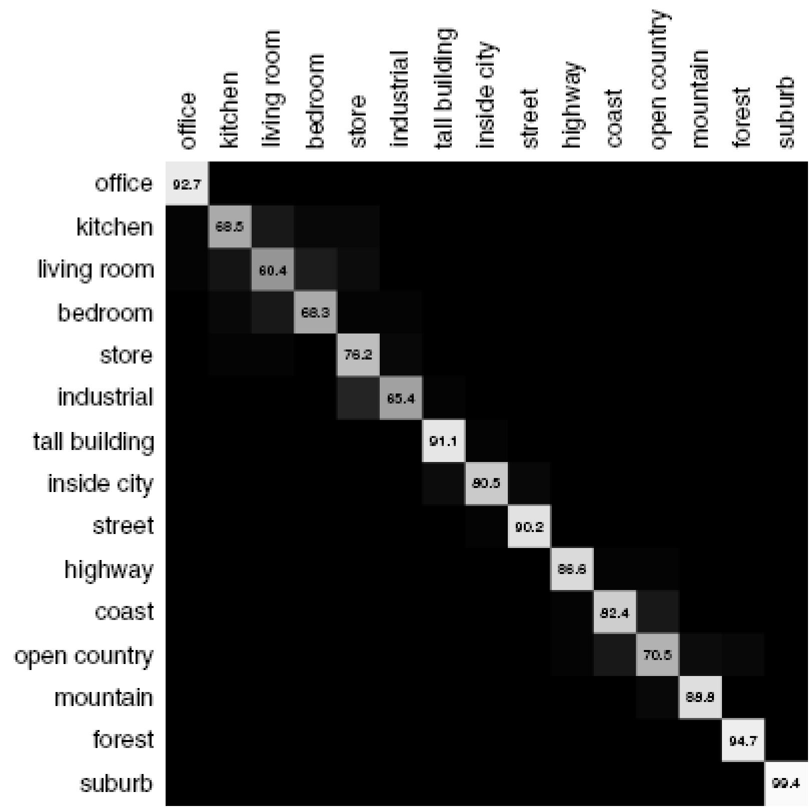
[Lazebnik, Schmid & Ponce, CVPR 2006]

Kristen Grauman

Spatial pyramid match

Captures scene categories well---texture-like patterns but with some variability in the positions of all the local pieces.



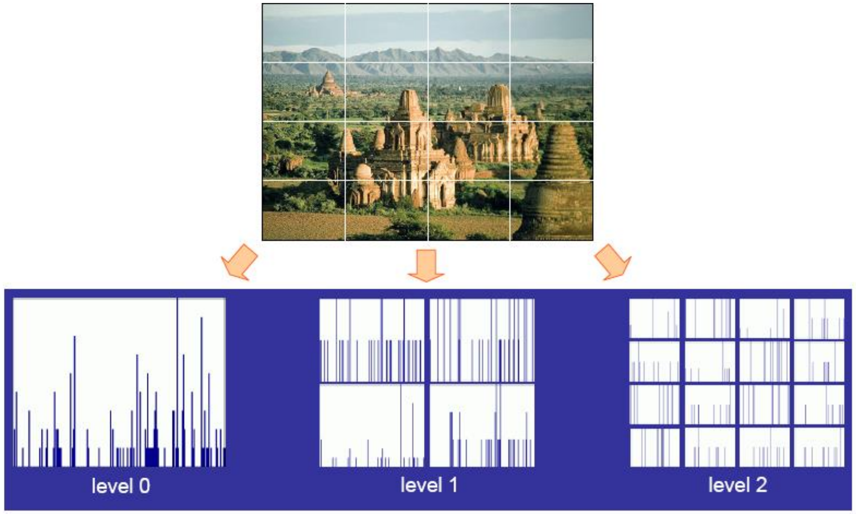


Confusion matrix

Kristen Grauman

Spatial pyramid match

Captures scene categories well---texture-like patterns but with some variability in the positions of all the local pieces.



	Strong features (vocabulary size: 200)	
Level	Single-level	Pyramid
0 (1 × 1)	72.2 ±0.6	
1 (2 × 2)	77.9 ±0.6	79.0 ±0.5
2 (4 × 4)	79.4 ±0.3	**81.1** ±0.3
3 (8 × 8)	77.2 ±0.4	80.7 ±0.3

Kristen Grauman

Part-based and local feature models for recognition

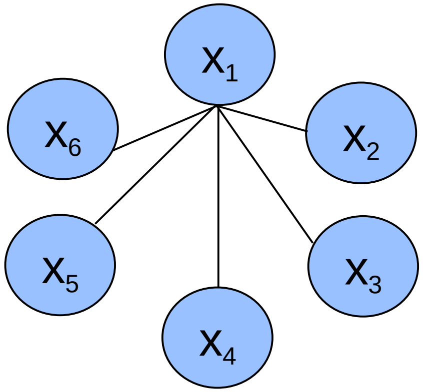
We'll look at three forms:

1. **Bag of words** (no geometry)
2. **Implicit shape model** (star graph for spatial model)
3. **Constellation model** (fully connected graph for spatial model)

Kristen Grauman

Shape representation in part-based models

"Star" shape model

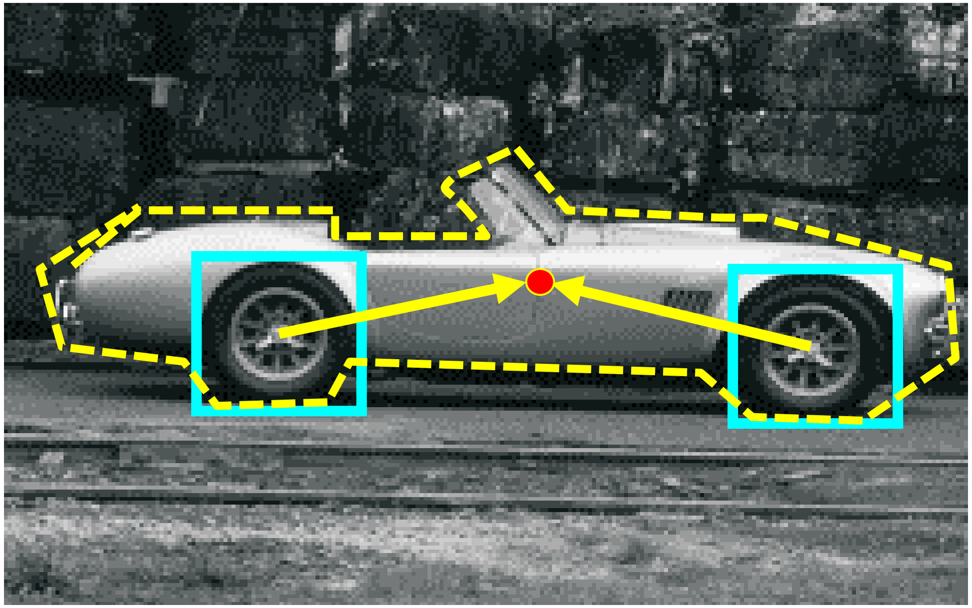


- **e.g. implicit shape model**
- **Parts mutually independent**

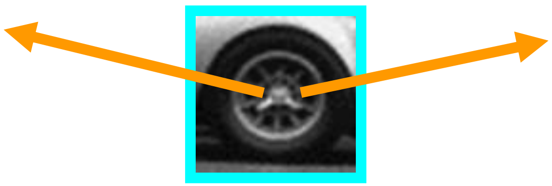
N image features, P parts in the model

Implicit shape models

- Visual vocabulary is used to index votes for object position [a visual word = “part”]



training image annotated with object localization info



visual codeword with displacement vectors

B. Leibe, A. Leonardis, and B. Schiele, [Combined Object Categorization and Segmentation with an Implicit Shape Model](), ECCV Workshop on Statistical Learning in Computer Vision 2004

Implicit shape models

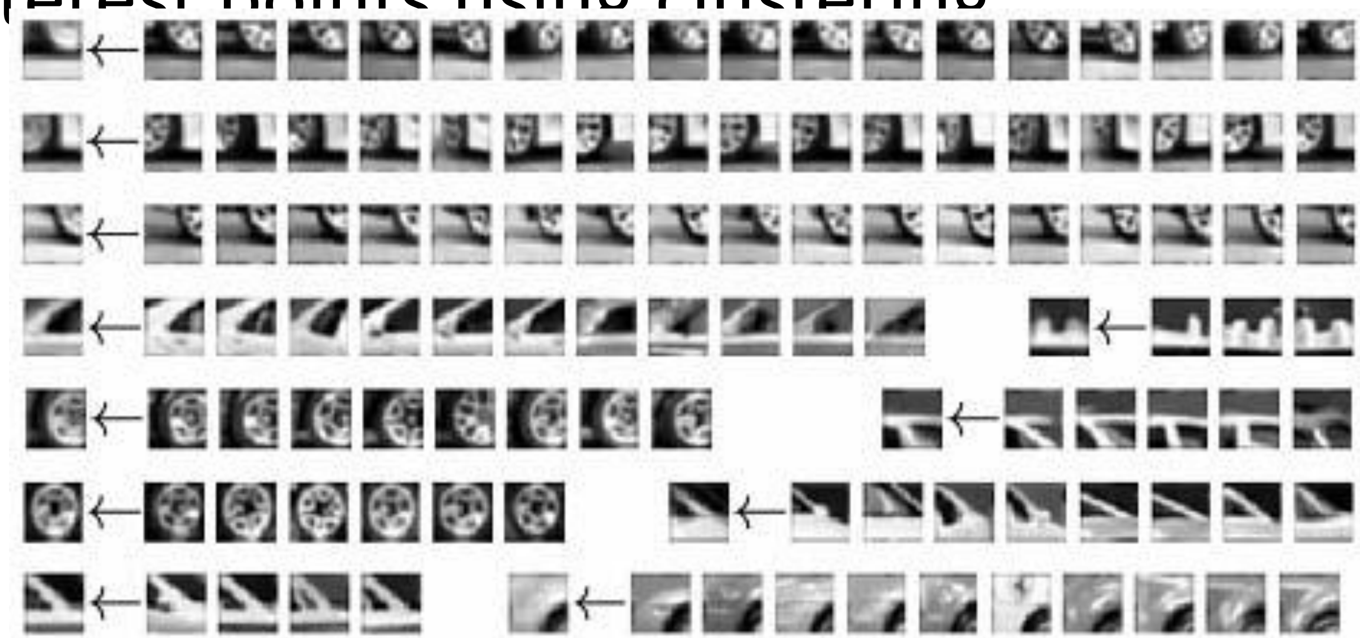
- Visual vocabulary is used to index votes for object position [a visual word = “part”]

test image

B. Leibe, A. Leonardis, and B. Schiele, [Combined Object Categorization and Segmentation with an Implicit Shape Model](), ECCV Workshop on Statistical Learning in Computer Vision 2004

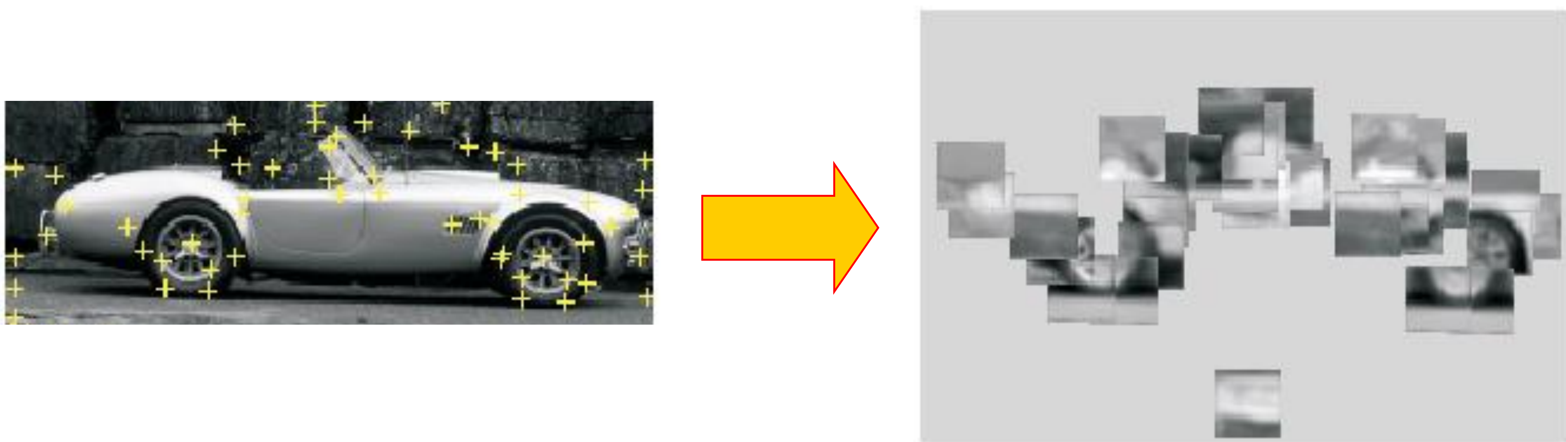
Implicit shape models: Training

1. Build vocabulary of patches around extracted interest points using clustering



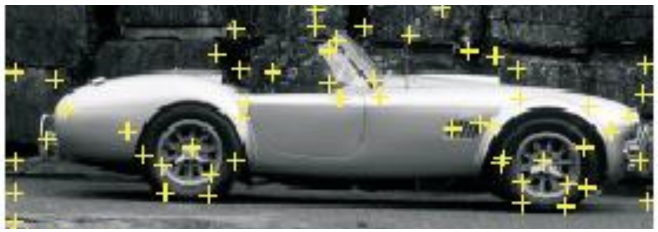
Implicit shape models: Training

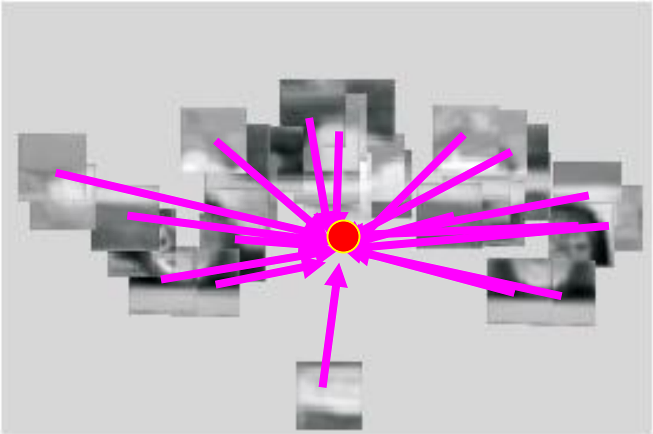
1. Build vocabulary of patches around extracted interest points using clustering
2. Map the patch around each interest point to closest word



Implicit shape models: Training

1. Build vocabulary of patches around extracted interest points using clustering
2. Map the patch around each interest point to closest word
3. For each word, store all positions it was found, relative to object center

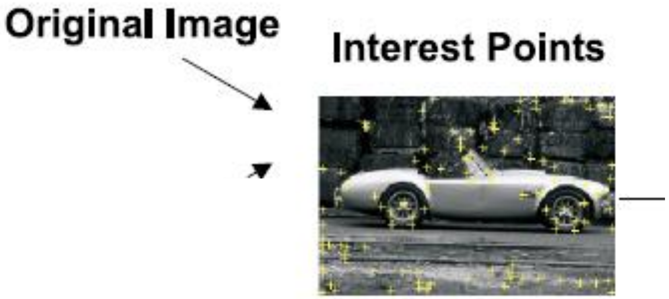




Implicit shape models: Testing

1. Given new test image, extract patches, match to vocabulary words
2. Cast votes for possible positions of object center
3. Search for maxima in voting space
4. (Extract weighted segmentation mask based on stored masks for the codebook occurrences)

Implicit shape models: Testing



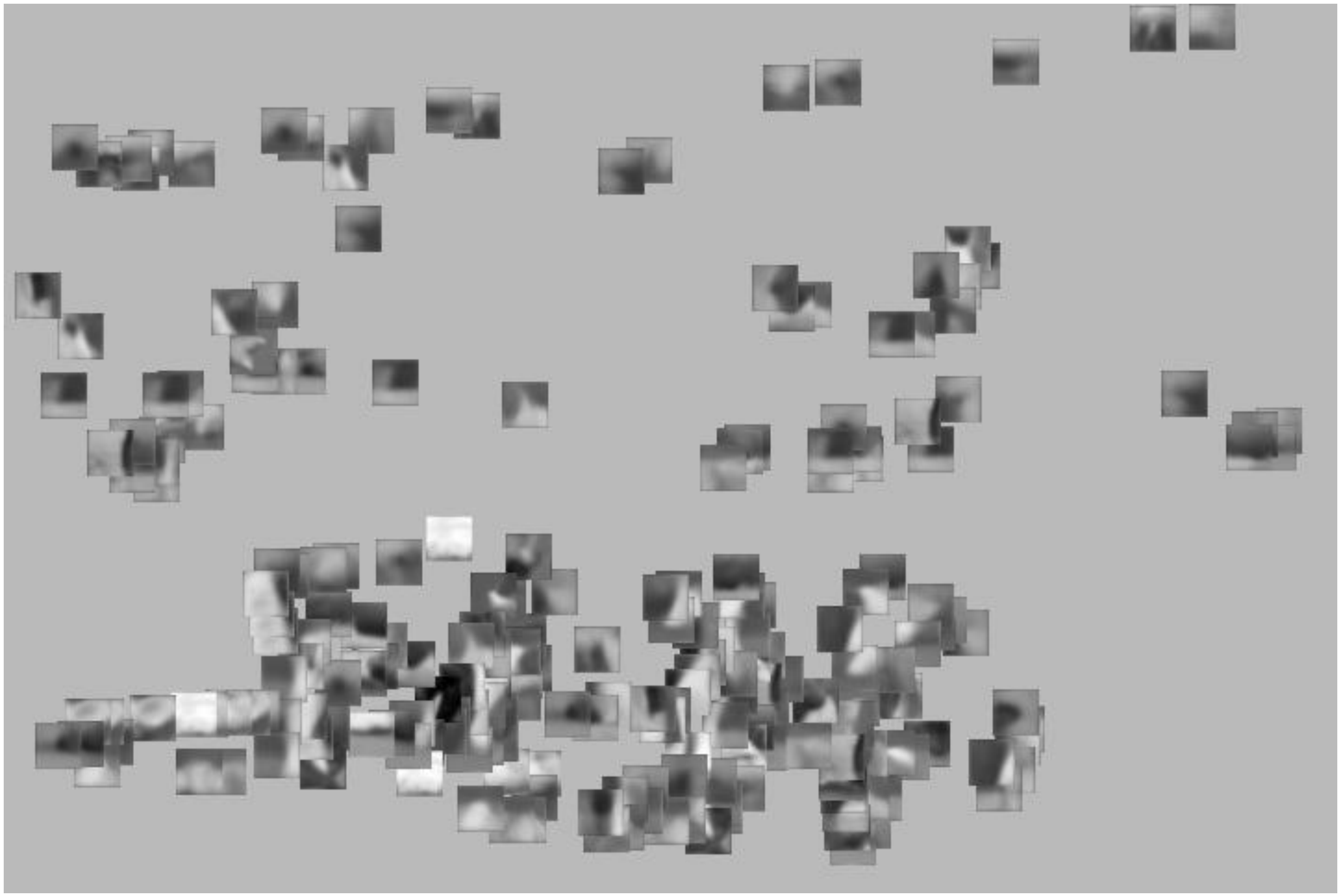
Example: Results on Cows

[redacted]

Example: Results on Cows

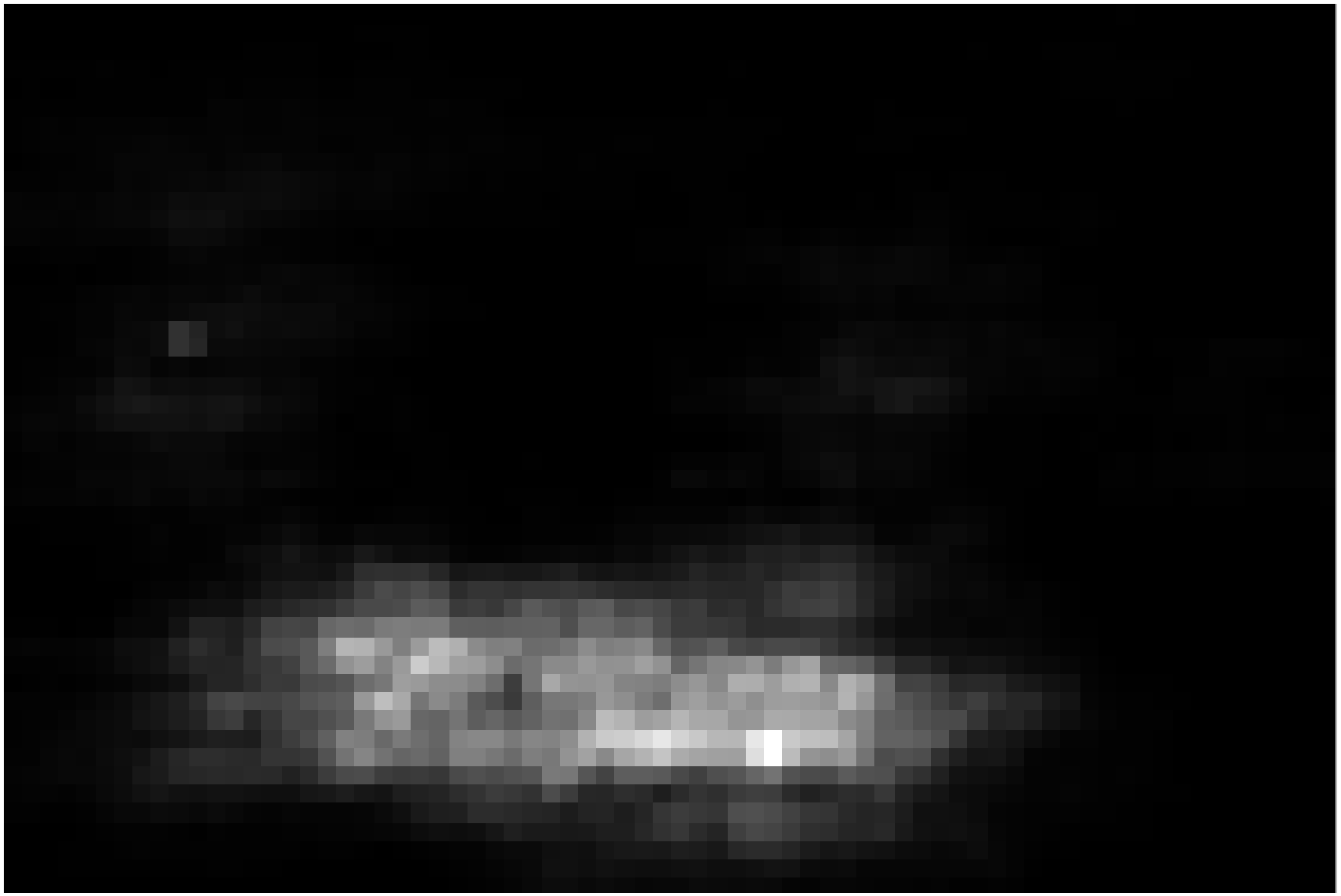
[redacted]

Example: Results on Cows



[redacted]

Example: Results on Cows



[redacted]

Example: Results on Cows

[redacted]

Example: Results on Cows

[redacted]

Example: Results on Cows

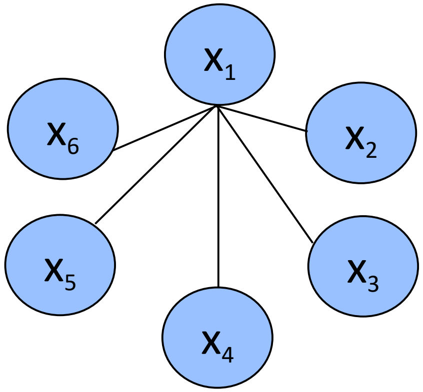
[redacted]

Detection Results

- Qualitative Performance
  - Recognizes different kinds of objects
  - Robust to clutter, occlusion, noise, low contrast

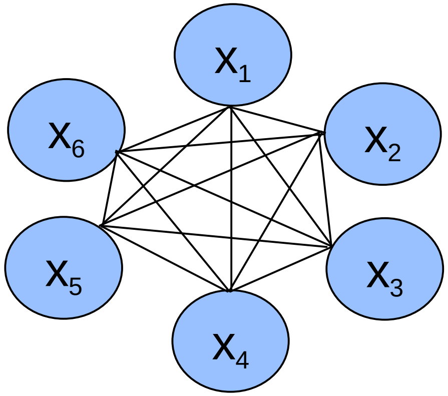
Shape representation in part-based models

"Star" shape model



- **e.g. implicit shape model**
- **Parts mutually independent**

Fully connected constellation model



- **e.g. Constellation Model**
- **Parts fully connected**

N image features, P parts in the model

Slide credit: Rob Fergus

Probabilistic constellation model

$$P(image \mid object) = P(appearance, shape \mid object)$$

Part descriptors

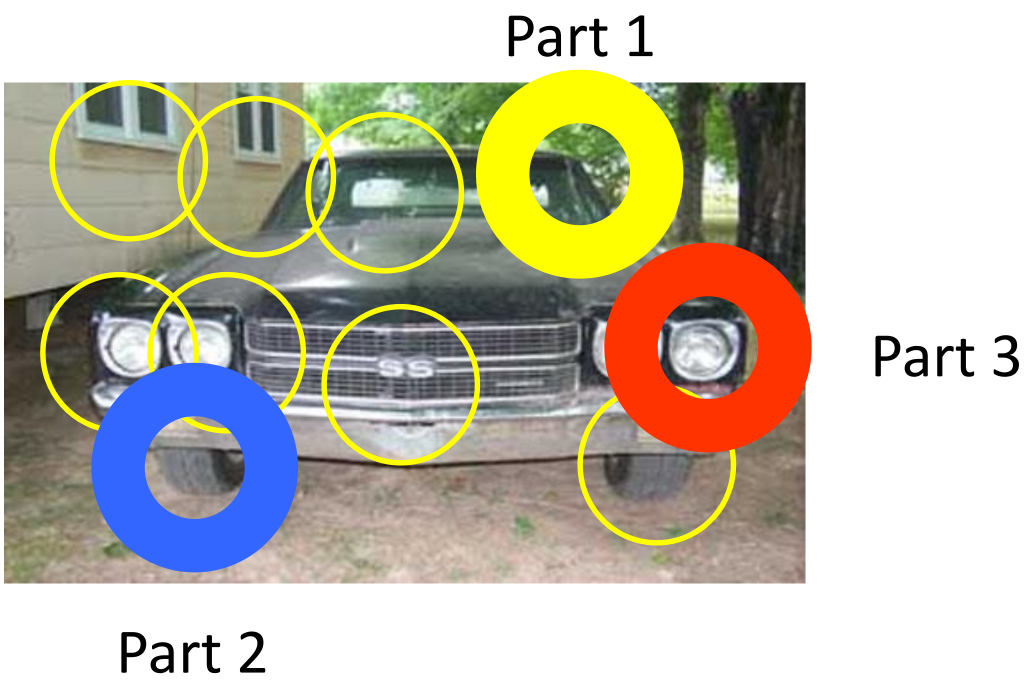
Part locations

Candidate parts

Source: Lana Lazebnik

Probabilistic constellation model

$$P(image \mid object) = P(appearance, shape \mid object)$$



Source: Lana Lazebnik

Probabilistic constellation model

$$P(image \mid object) = P(appearance, shape \mid object)$$
$$= \max_h P(appearance \mid h, object)\, p(shape \mid h, object)\, p(h \mid object)$$

h: assignment of features to parts

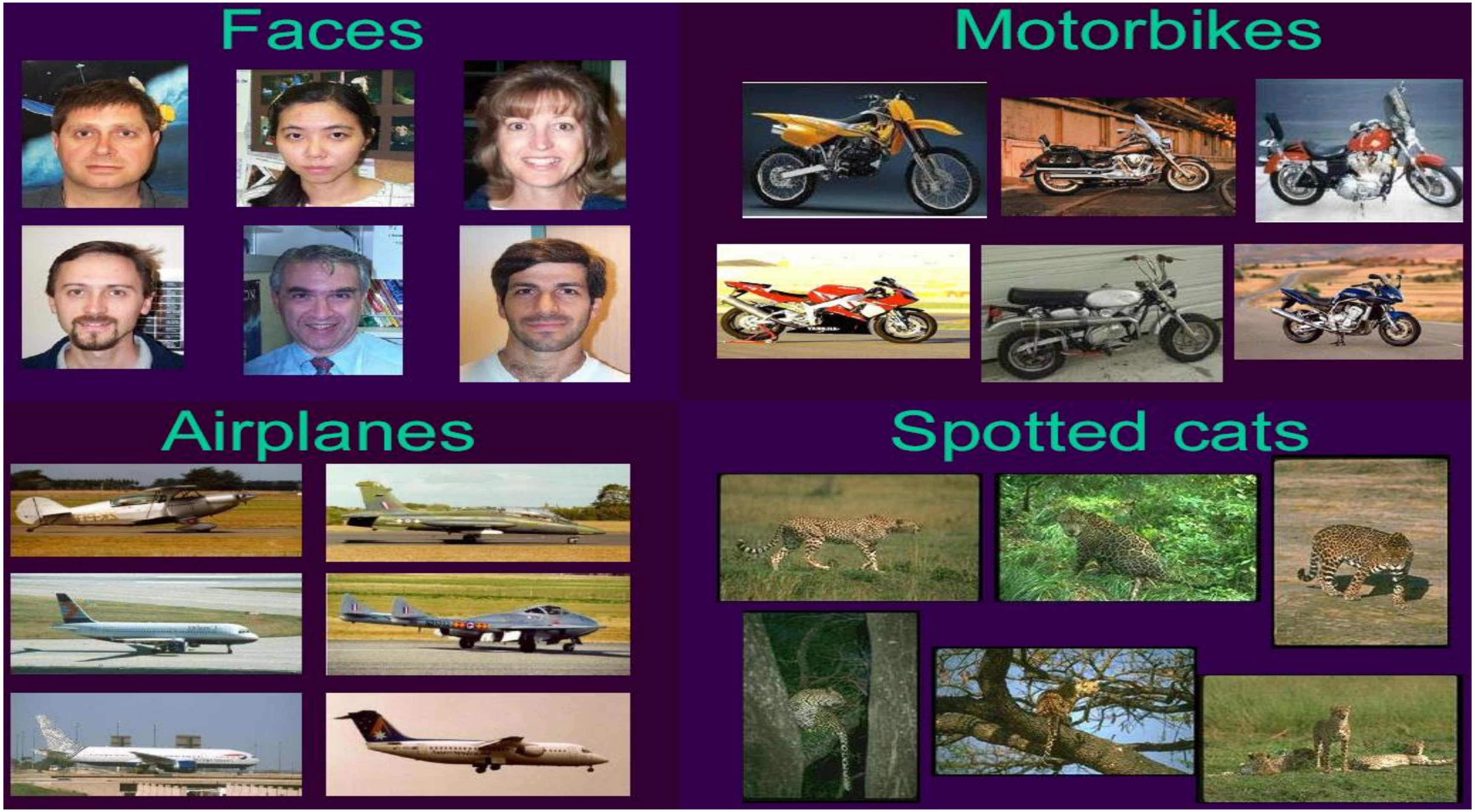
Part 1

Part 3

Part 2

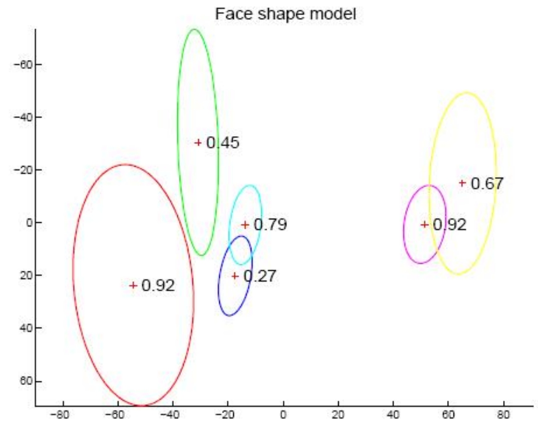
Source: Lana Lazebnik

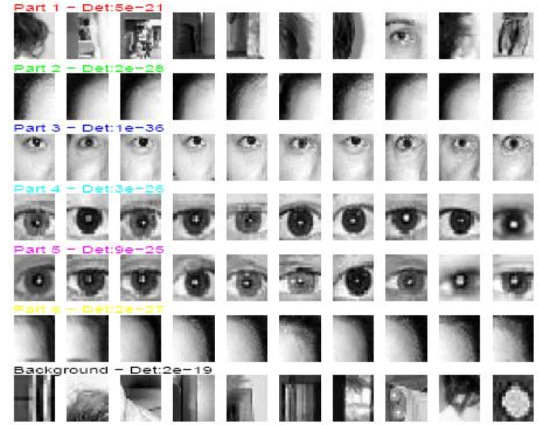
Example results from constellation model: data from four categories



Slide from Li Fei-Fei http://www.vision.caltech.edu/feifeili/Resume.htm

Face model

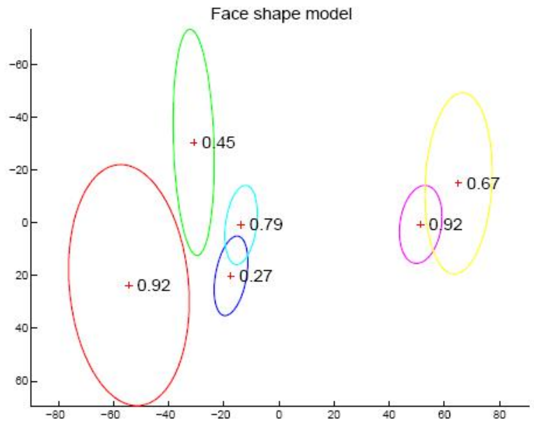


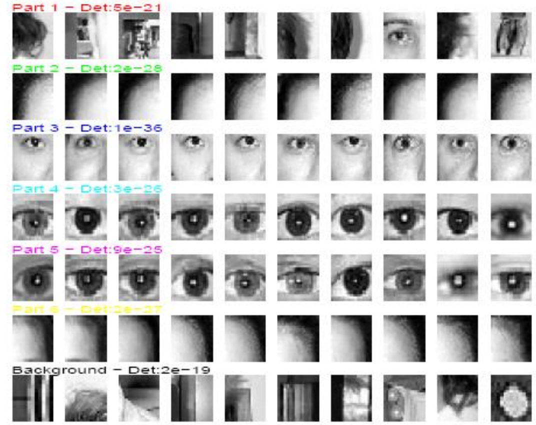


Appearance: 10 patches closest to mean for each part

Fergus et al. CVPR 2003

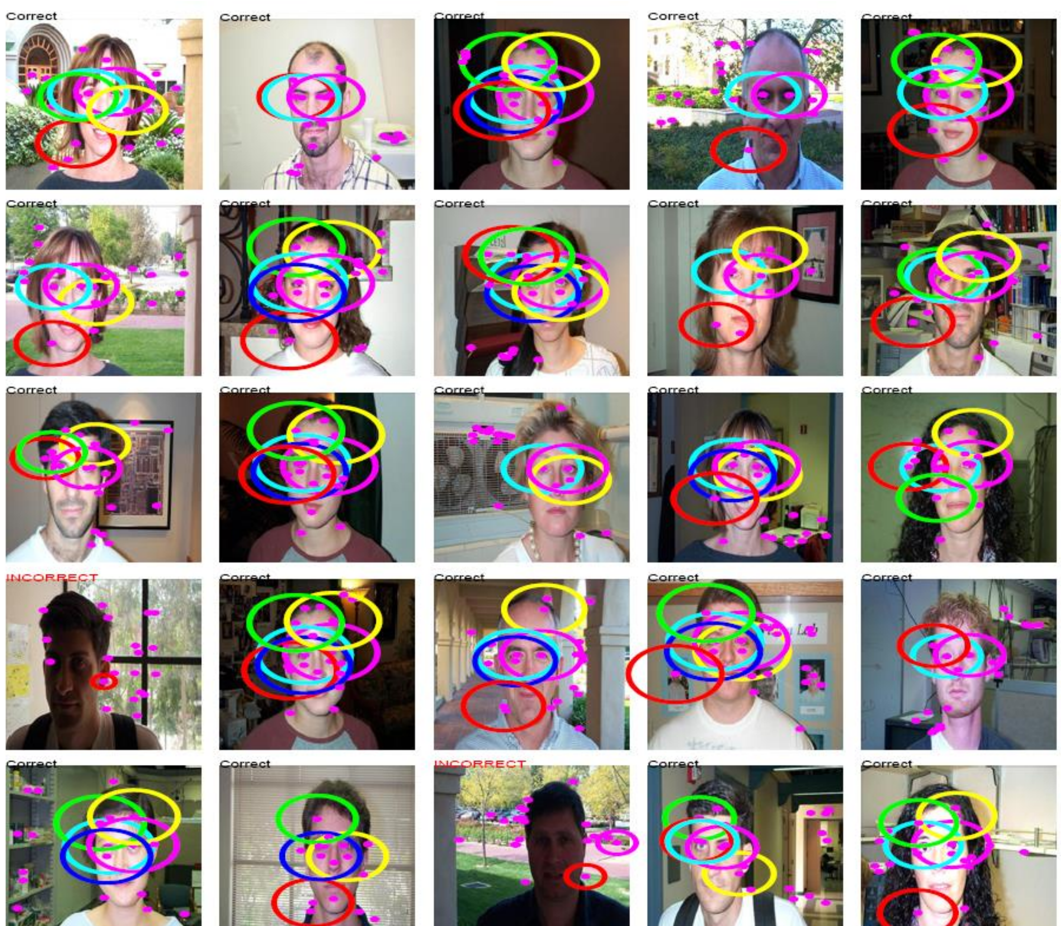
Face model





Appearance: 10 patches closest to mean for each part

Recognition results

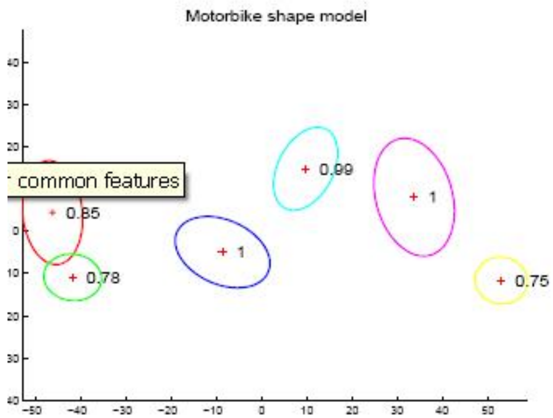


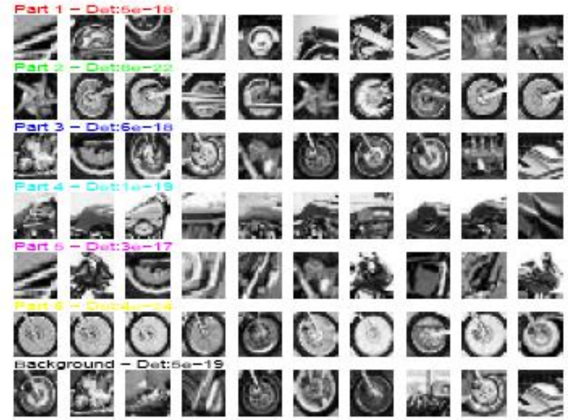
Test images: size of circles indicates score of hypothesis

Kristen Grauman

Fergus et al. CVPR 2003

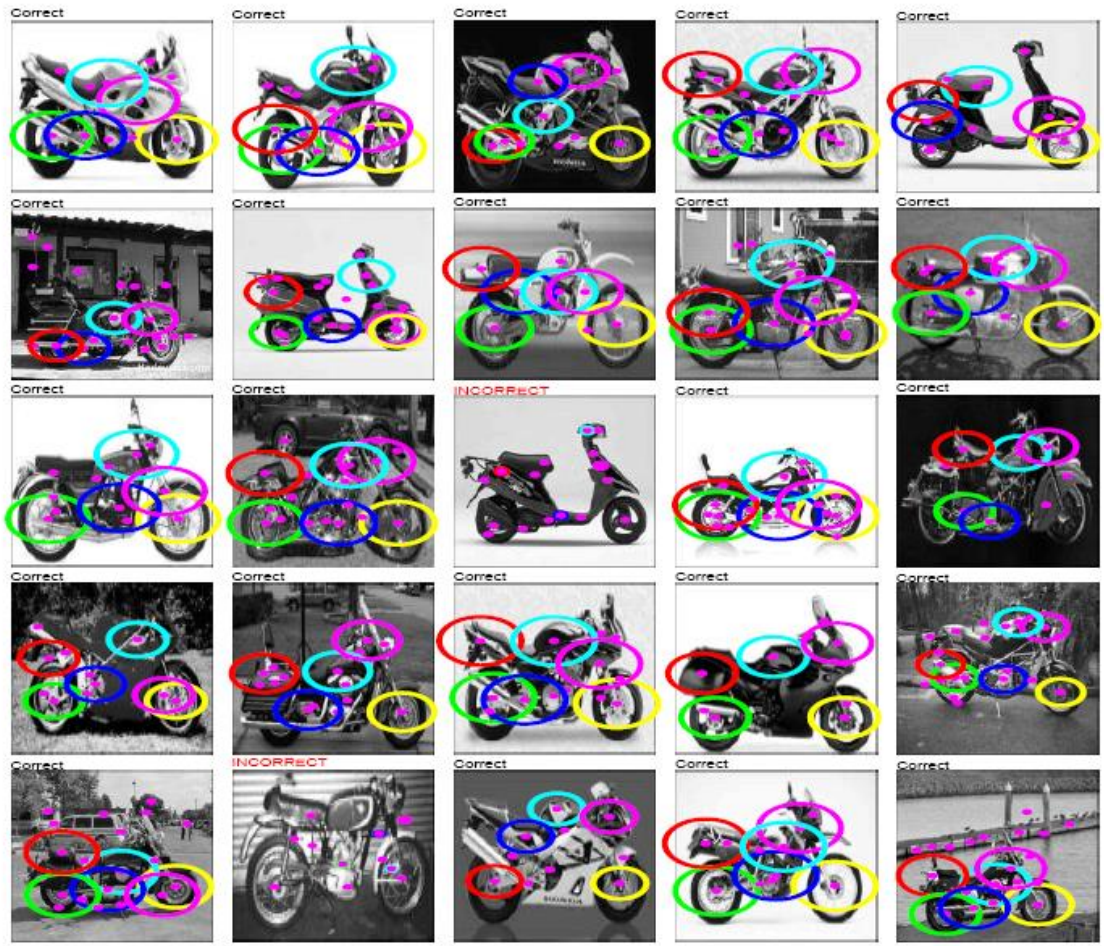
Motorbike model





Appearance: 10 patches closest to mean for each part

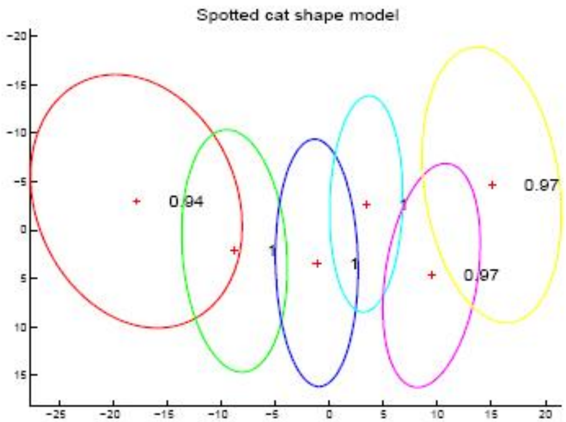
Recognition results

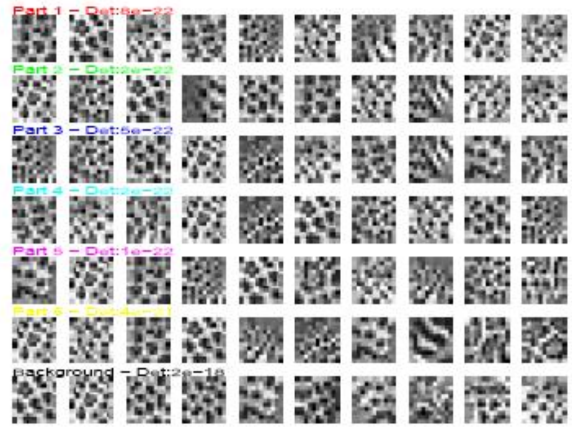


Kristen Grauman

Fergus et al. CVPR 2003

Spotted cat model





Appearance: 10 patches closest to mean for each part

Recognition results

Kristen Grauman

Fergus et al. CVPR 2003

Comparison

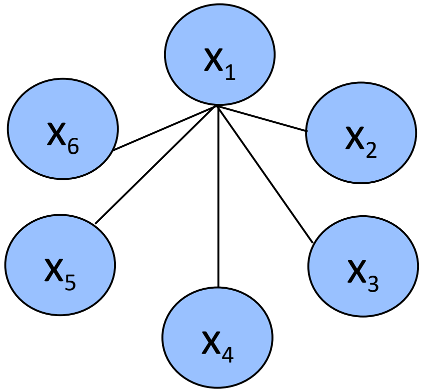
     

class	bag of features Zhang et al. (2005)	bag of features Willamowski et al. (2004)	Part-based model Fergus et al. (2003)
airplanes	**98.8**	97.1	90.2
cars (rear)	98.3	**98.6**	90.3
cars (side)	**95.0**	87.3	88.5
faces	**100**	99.3	96.4
motorbikes	**98.5**	98.0	92.5
spotted cats	**97.0**	—	90.0

Source: Lana Lazebnik

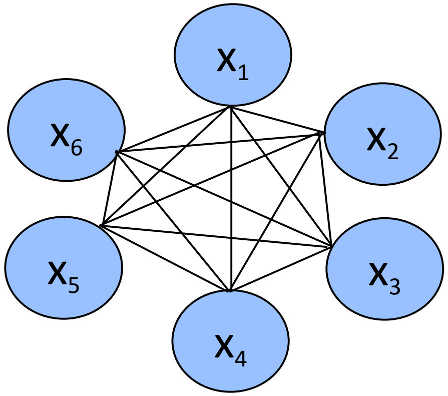
Shape representation in part-based models

"Star" shape model



- **e.g. implicit shape model**
- **Parts mutually independent**
- **Recognition complexity: O(NP)**
- **Method: Gen. Hough Transform**

Fully connected constellation model



- **e.g. Constellation Model**
- **Parts fully connected**
- **Recognition complexity: $O(N^P)$**
- **Method: Exhaustive search**

N image features, P parts in the model

Slide credit: Rob Fergus

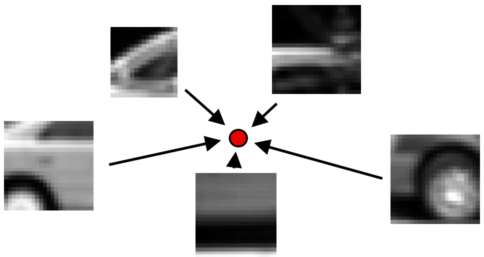
Summary:
part-based and local feature models for generic object recognition

- **Histograms of visual words** to capture global or local layout in the bag-of-words framework
  - Pyramid match kernels
  - Powerful in practice for image recognition

- **Part-based models** encode category's part appearance together with 2d layout and allow detection within cluttered image
  - "**implicit shape model**": shape based on layout of all parts relative to a reference part; Generalized Hough for detection
  - "**constellation model**": explicitly model mutual spatial layout between all pairs of parts; exhaustive search for best fit of features to parts

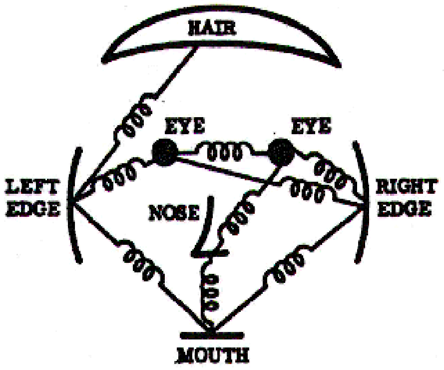
Structure models

Voting models



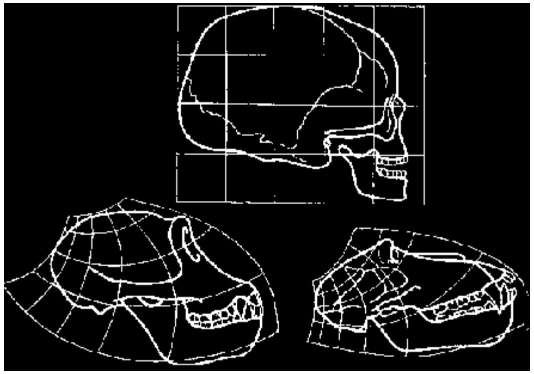
- Many parts (>100)

Constellation models



- Few parts (~6)

Deformable models



- No parts

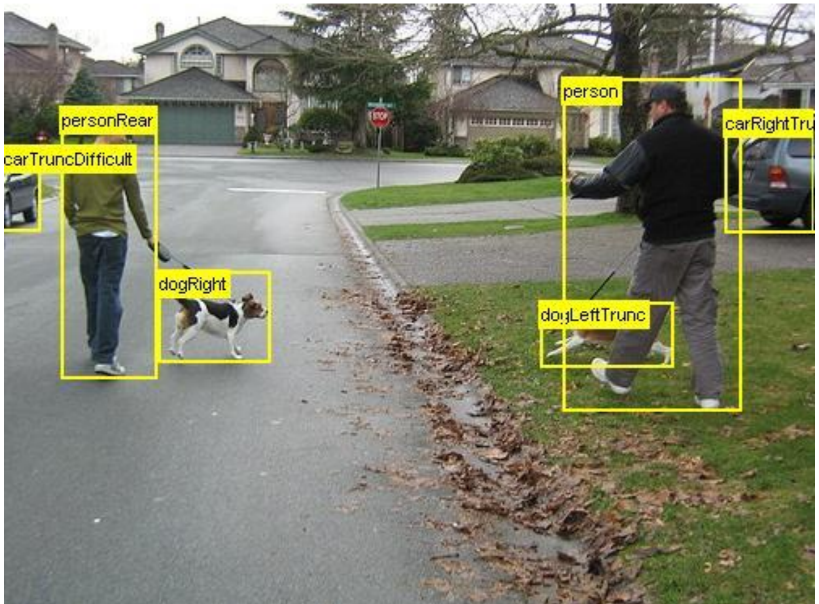
Object Detection with Discriminatively Trained Part Based Models

Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester and Deva Ramanan

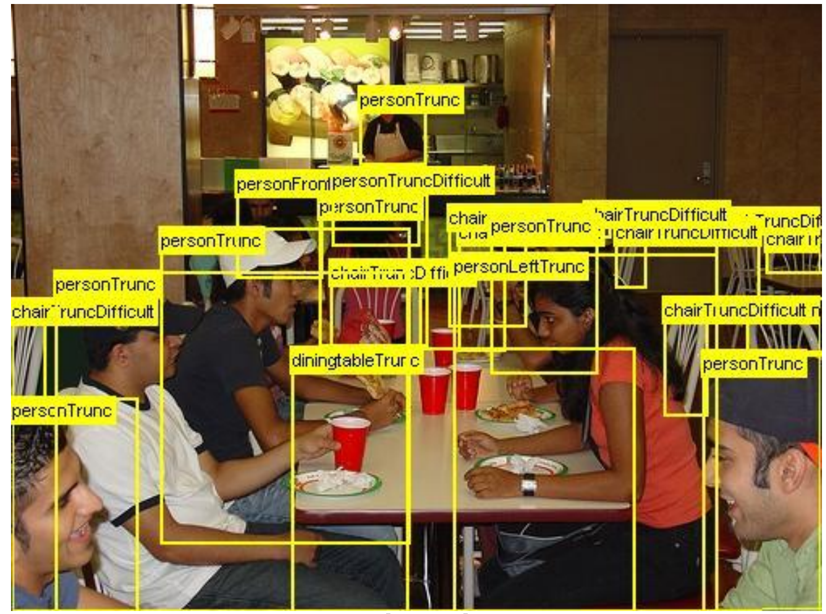
Abstract—We describe an object detection system based on mixtures of multiscale deformable part models. Our system is able to represent highly variable object classes and achieves state-of-the-art results in the PASCAL object detection challenges. While deformable part models have become quite popular, their value had not been demonstrated on difficult benchmarks such as the PASCAL datasets. Our system relies on new methods for discriminative training with partially labeled data. We combine a margin-sensitive approach for data-mining hard negative examples with a formalism we call *latent SVM*. A latent SVM is a reformulation of MI-SVM in terms of latent variables. A latent SVM is semi-convex and the training problem becomes convex once latent information is specified for the positive examples. This leads to an iterative training algorithm that alternates between fixing latent values for positive examples and optimizing the latent SVM objective function.

Index Terms—Object Recognition, Deformable Models, Pictorial Structures, Discriminative Training, Latent SVM

PASCAL Visual Object Challenge



5000 training images



5000 testing images

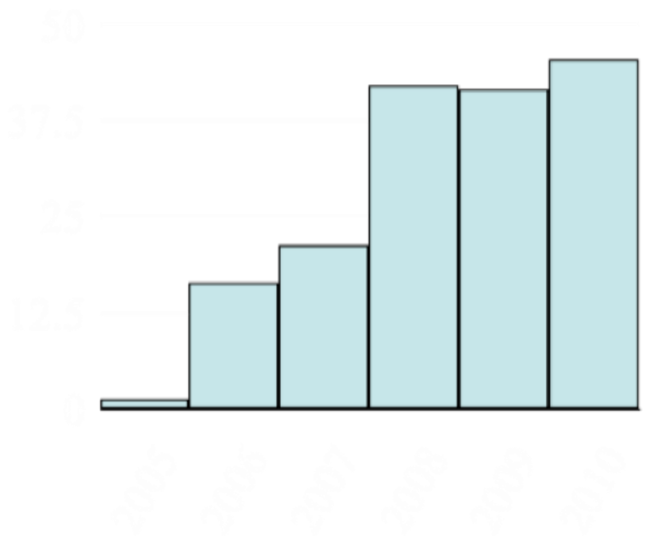
20 everyday object categories

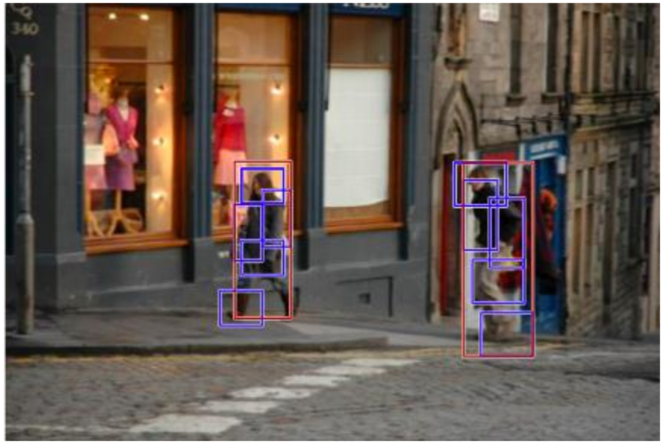
aeroplane bike bird boat bottle bus car cat chair cow table
dog horse motorbike person plant sheep sofa train tv

Source: Deva Ramanan

5 years of PASCAL people detection

average
precision



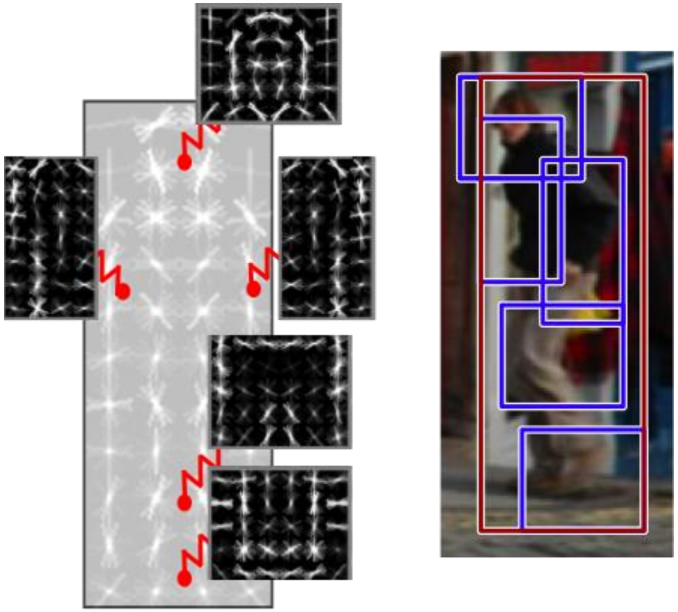


1% to 45% in 5 years

Discriminative mixtures of star models 2007-2010 Felzenszwalb, McAllester, Ramanan *CVPR* 2008
Felzenszwalb, Girshick, McAllester, and Ramanan *PAMI* 2009

Source: Deva Ramanan

Deformable part models



Model encodes local appearance + pairwise geometry

Source: Deva Ramanan

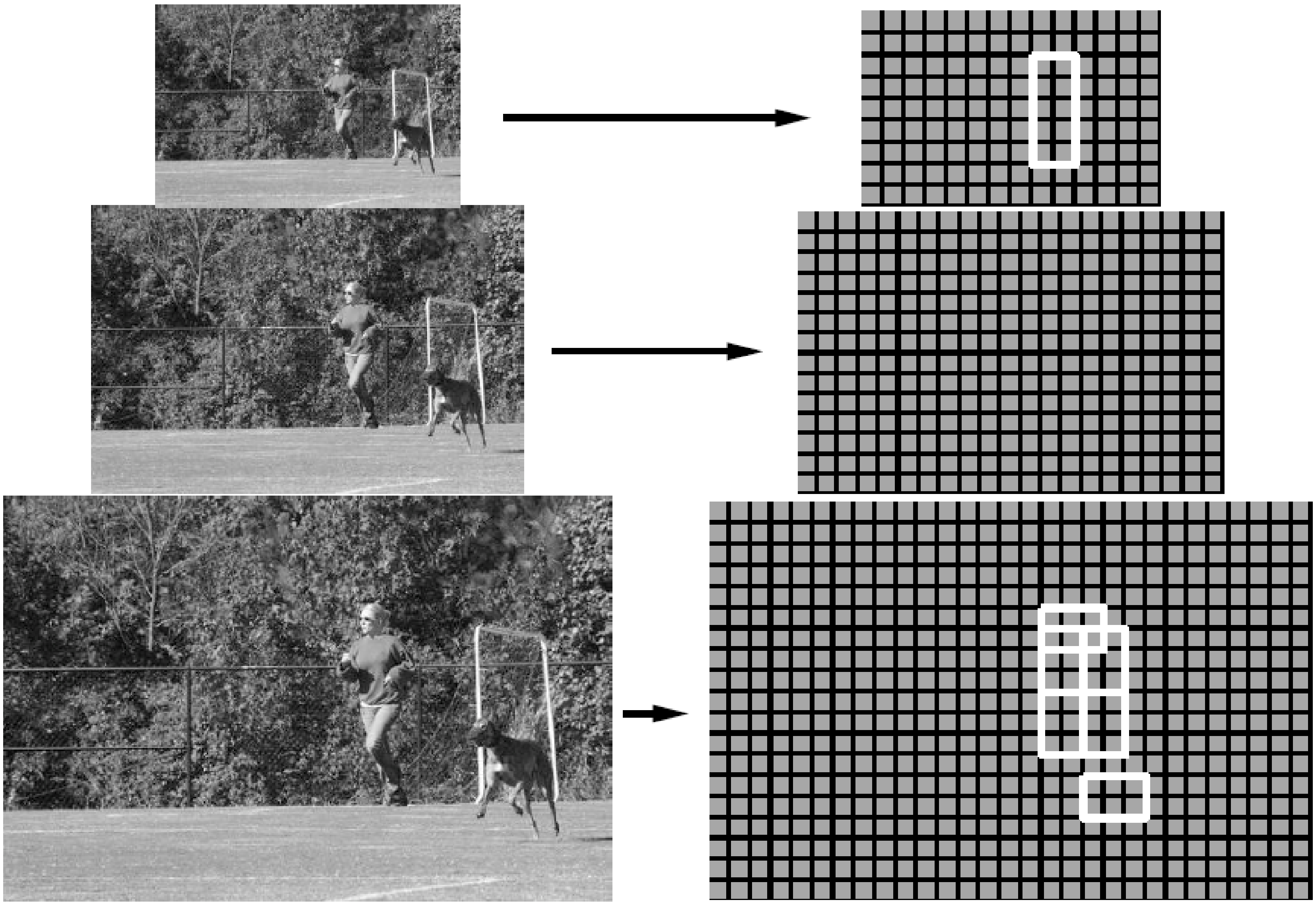
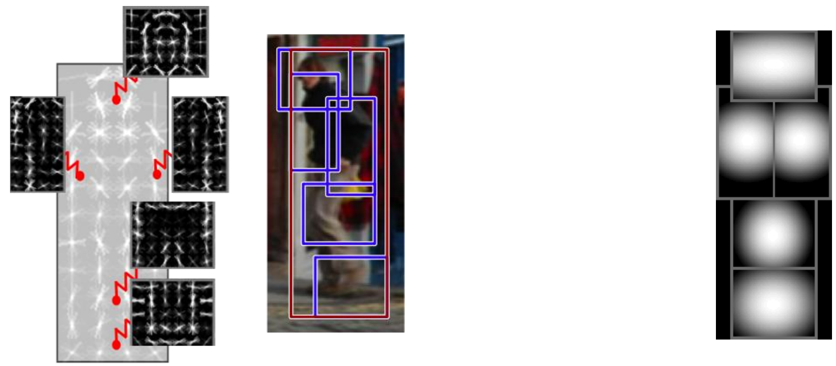


Image pyramid

Feature pyramid

Scoring function



$$\text{score}(x,z) = \sum_i w_i\, \phi(x, z_i) + \sum_{i,j} w_{ij}\, \Psi(z_i, z_j)$$

x = image
$z_i = (x_i, y_i)$
$z = \{z_1, z_2 \ldots\}$

part template scores

spring deformation model

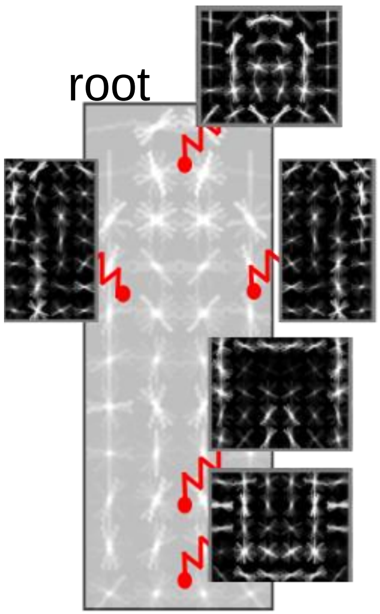
Score is linear in local templates w_i and spring parameters w_{ij}

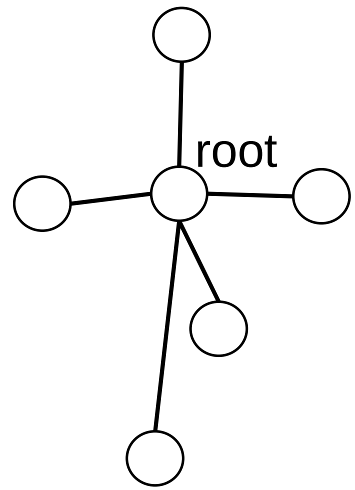
$$\text{score}(x,z) = w \cdot \Phi(x, z)$$

Source: Deva Ramanan

Inference: $\max_{z}$ score(x,z)

Felzenszwalb & Huttenlocher 05



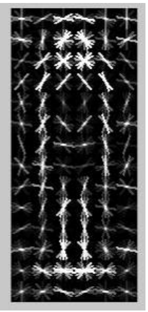


Star model: the location of the root filter is the anchor point
Given the root location, all part locations are independent

Classification

$f_w(x)=w \cdot \Phi(x)$

$f_w(x)>0$

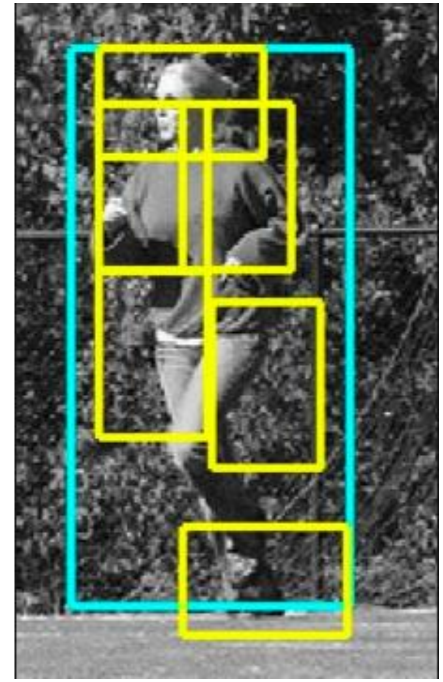


Source: Deva Ramanan

Latent-variable classification

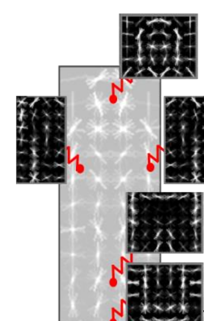
$f_w(x)=w \cdot \Phi(x)$

$f_w(x)>0$



$f_w(x)=\max_z S(x,z)$

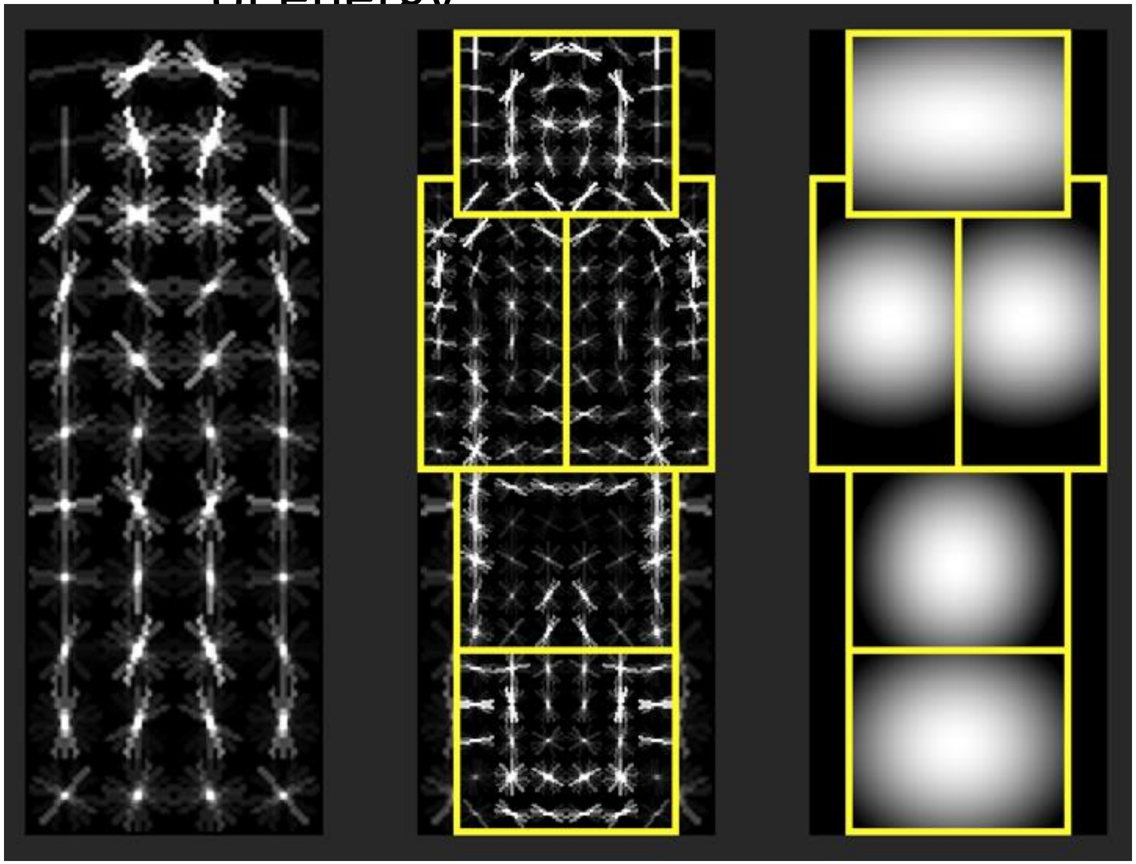
$=\max_z w \cdot \Phi(x, z)$



Source: Deva Ramanan

Learning Initialization

- Learn root filter with SVM
- Initialize part filters to regions in root filter with lots of energy



Source: Deva Ramanan

Coordinate descent

1) Given positive part locations, learn w with a convex program

$$w = \arg\min_{w} L(w) \quad \text{with fixed} \quad \{z_n : n \in \text{pos}\}$$

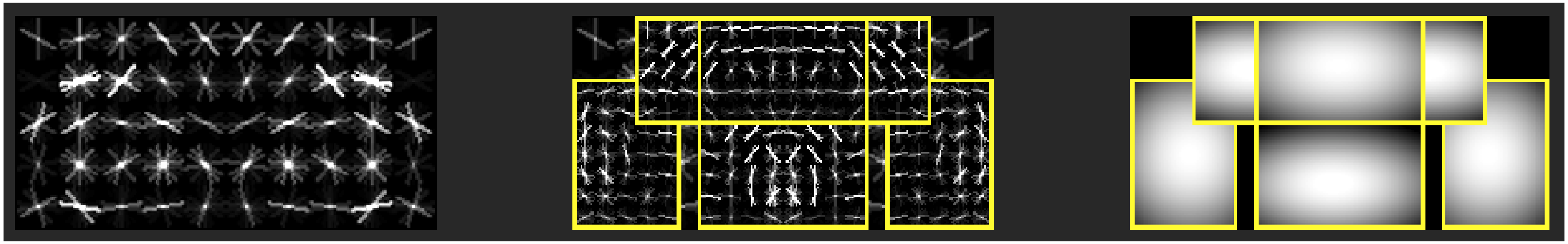
2) Given w, estimate part locations on positives

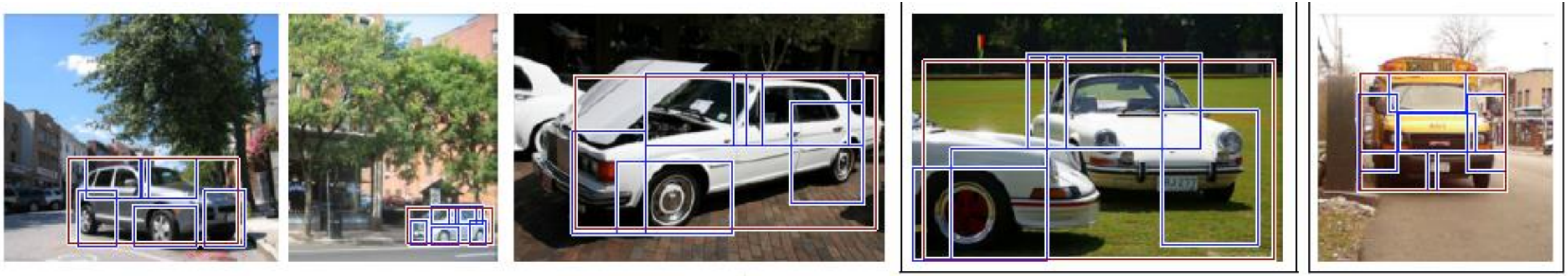
$$z_n = \arg\max_{z} w \cdot \Phi(x_n, z) \quad \forall n \in \text{pos}$$

The above steps perform coordinate descent on a joint loss

Source: Deva Ramanan

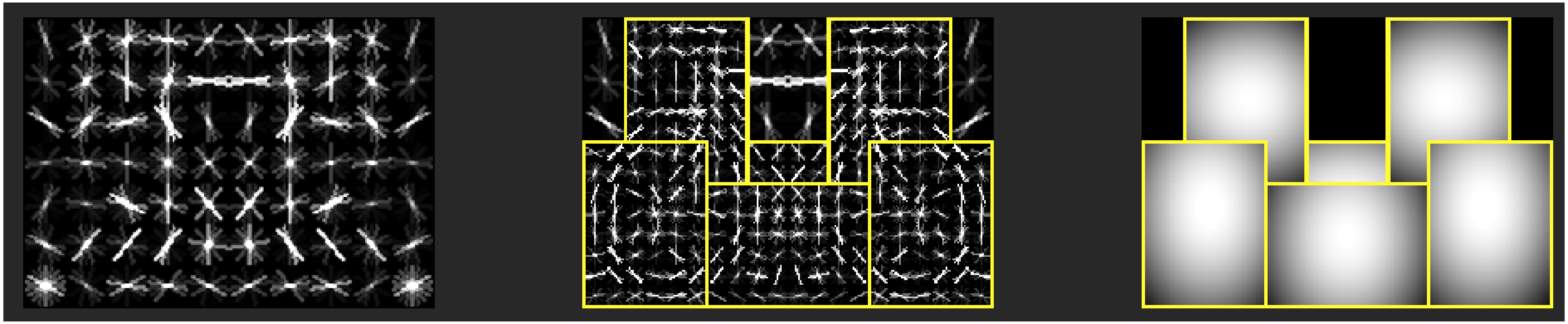
Example models

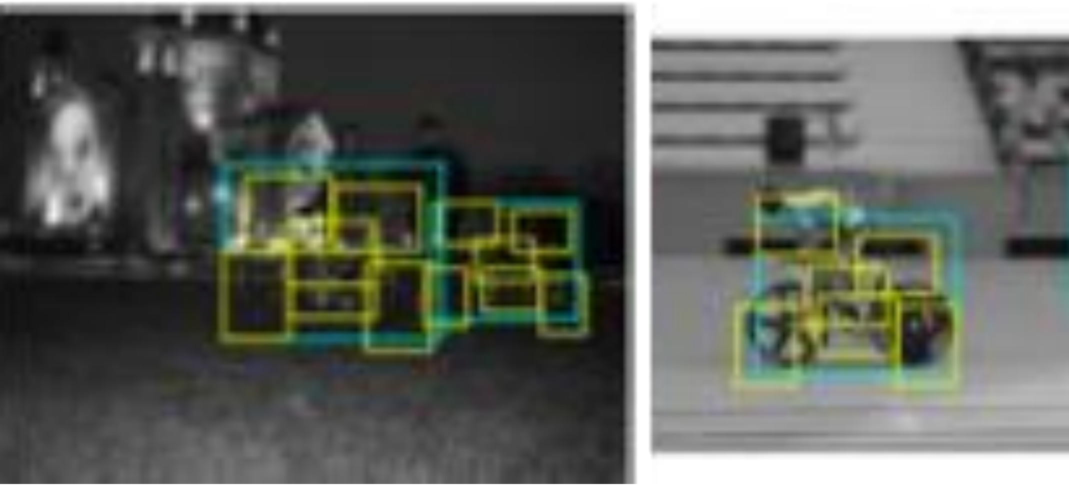




Source: Deva Ramanan

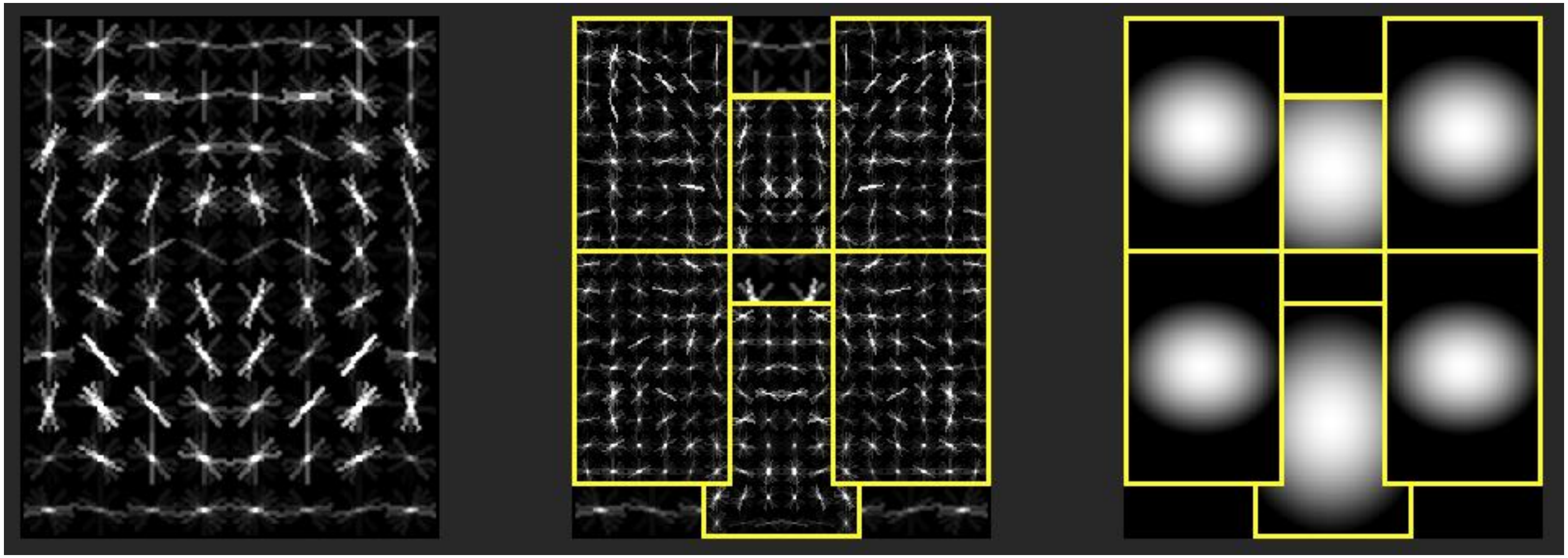
Example models





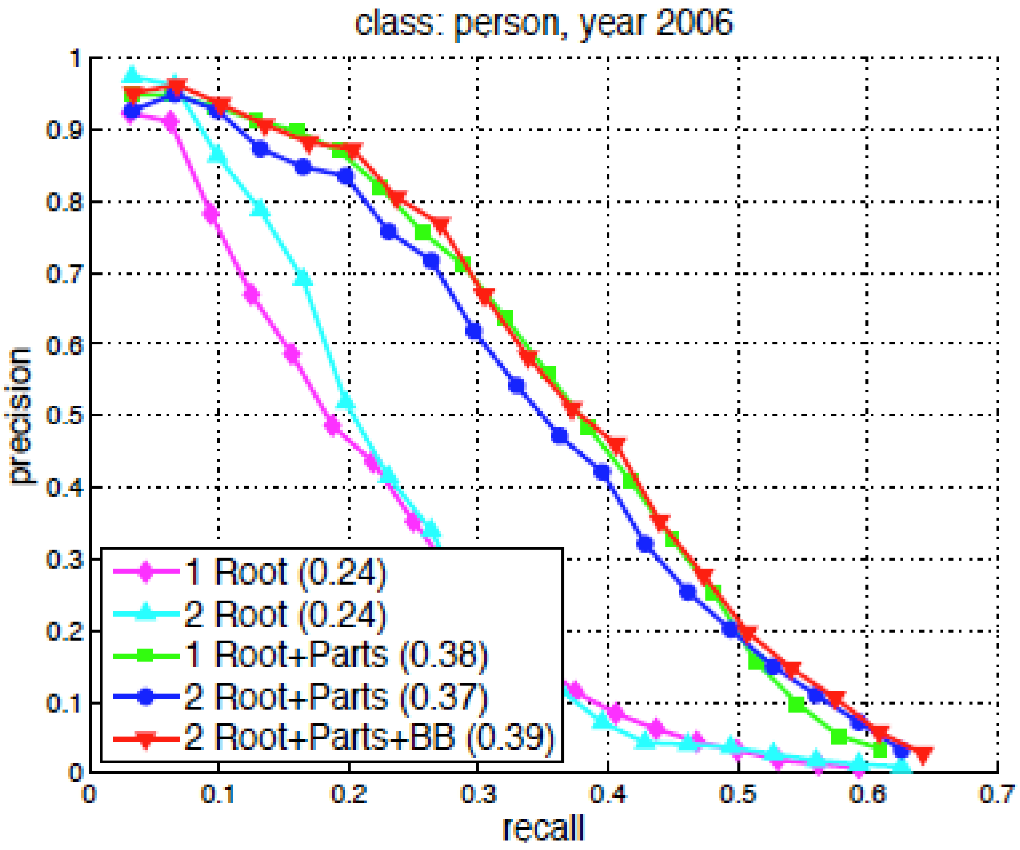
Source: Deva Ramanan

Example models



False positive due to imprecise bounding box

Source: Deva Ramanan

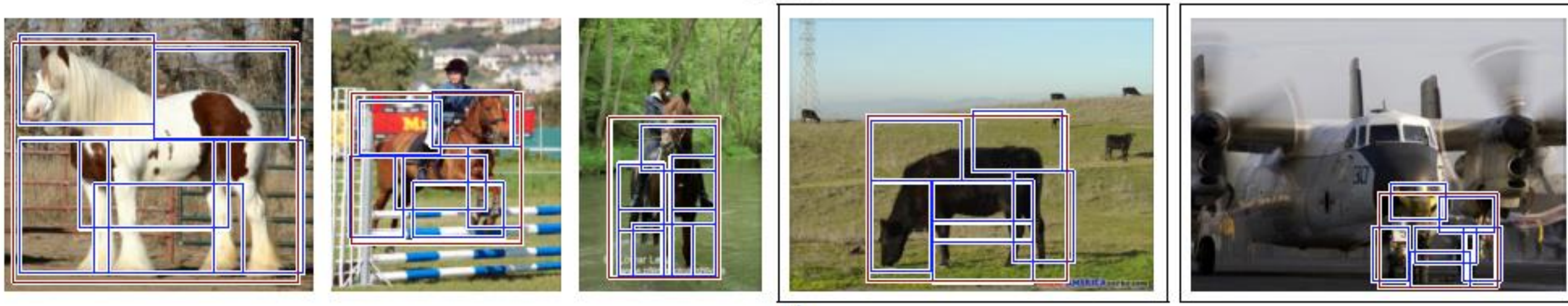


Other tricks:

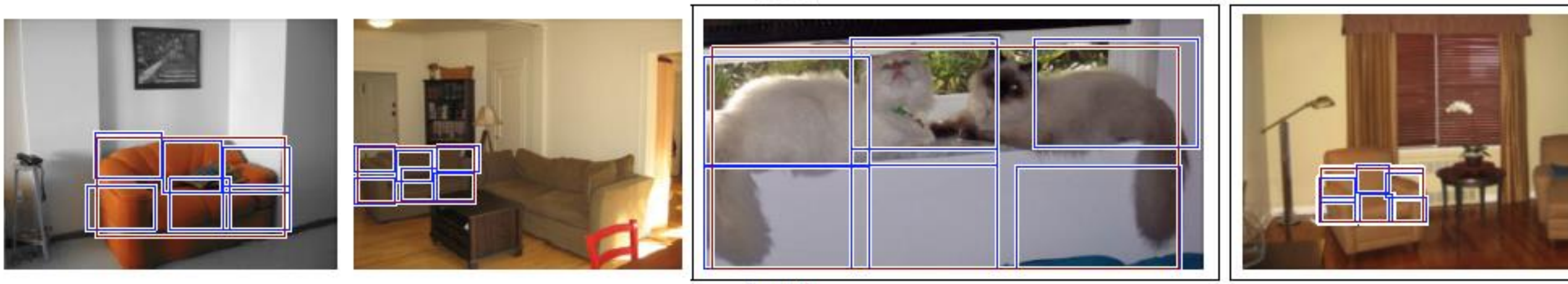
•Mining hard negative examples

•Noisy annotations

horse

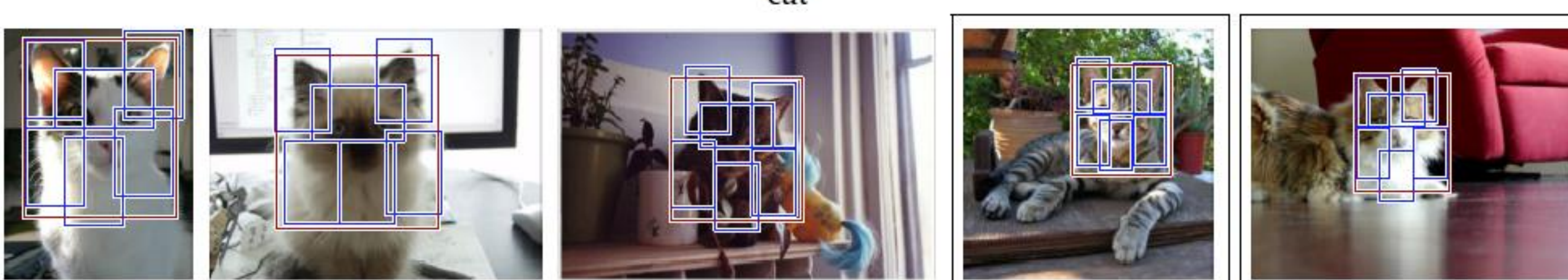


sofa



bottle

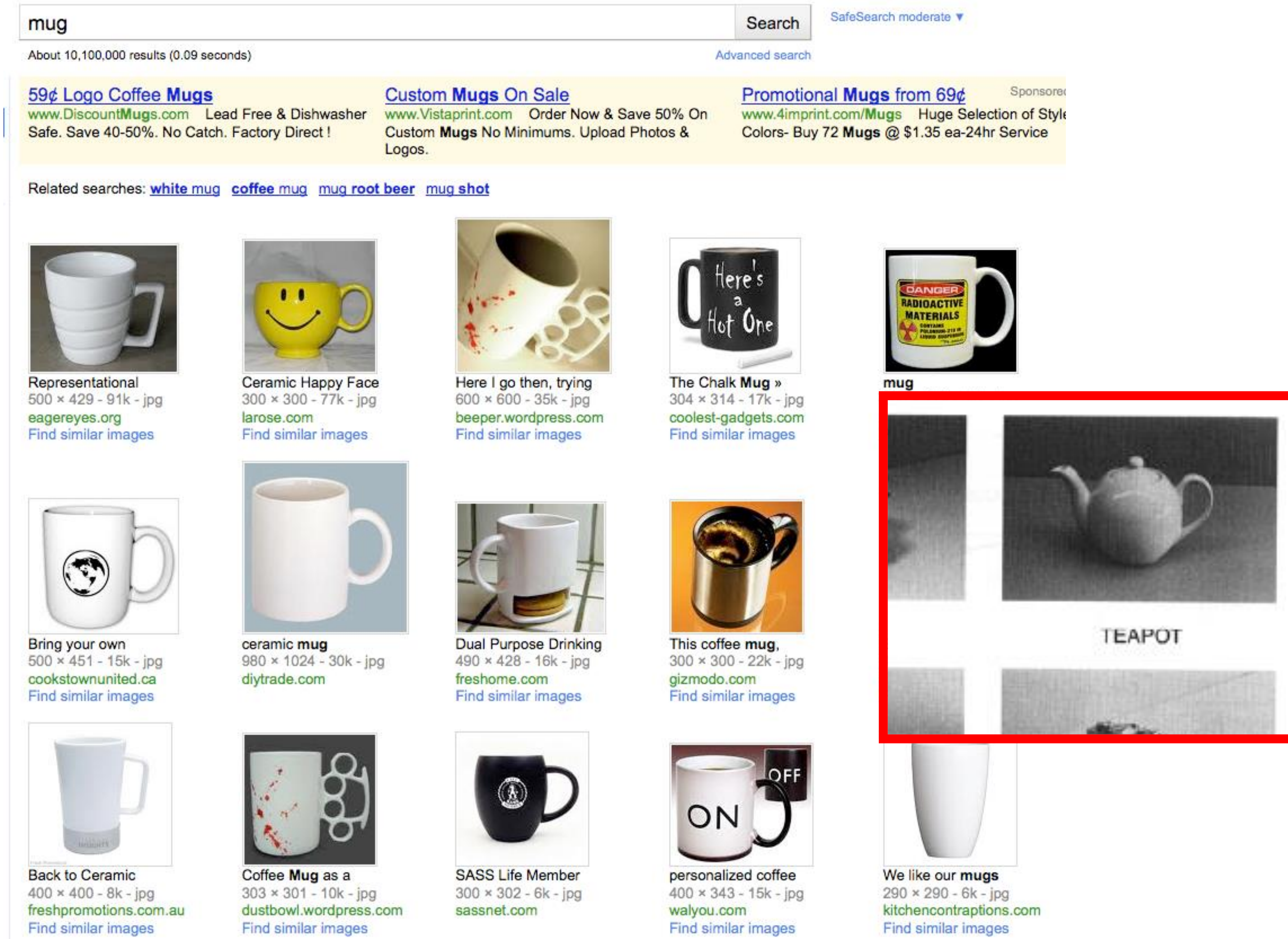
cat



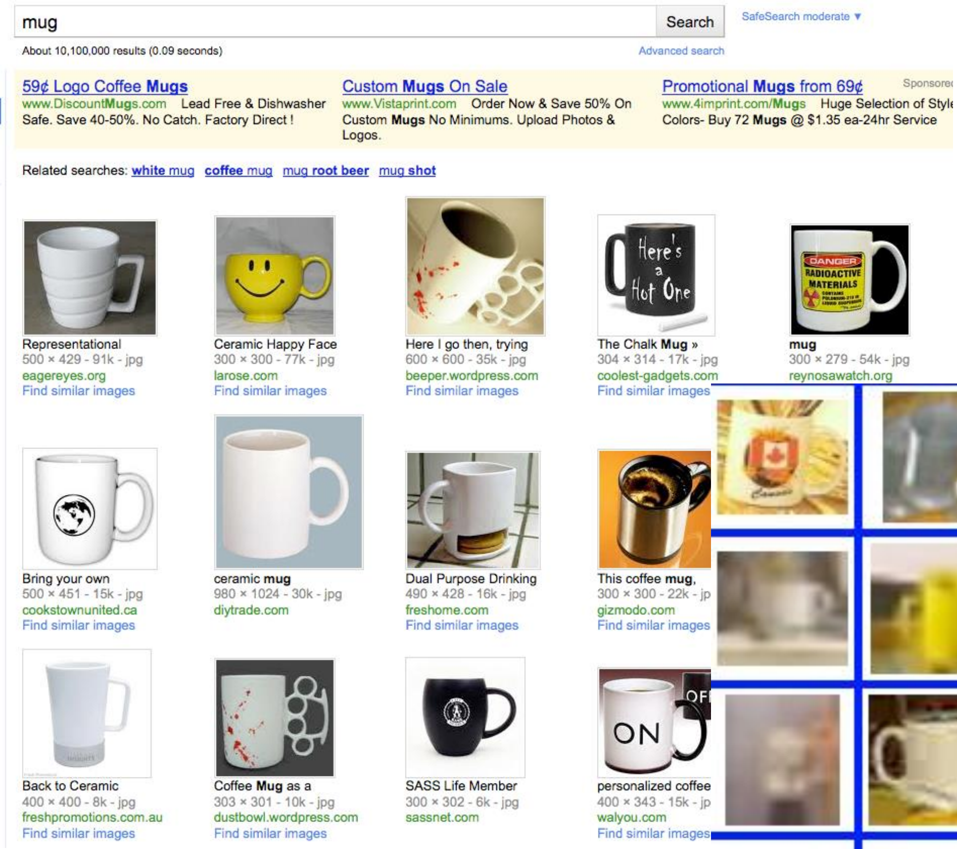
Outline

- Image matching and oriented gradients: SIFT, HOG
- Object detection
- Dataset and generalization issues

Some bias comes from the way the data is collected

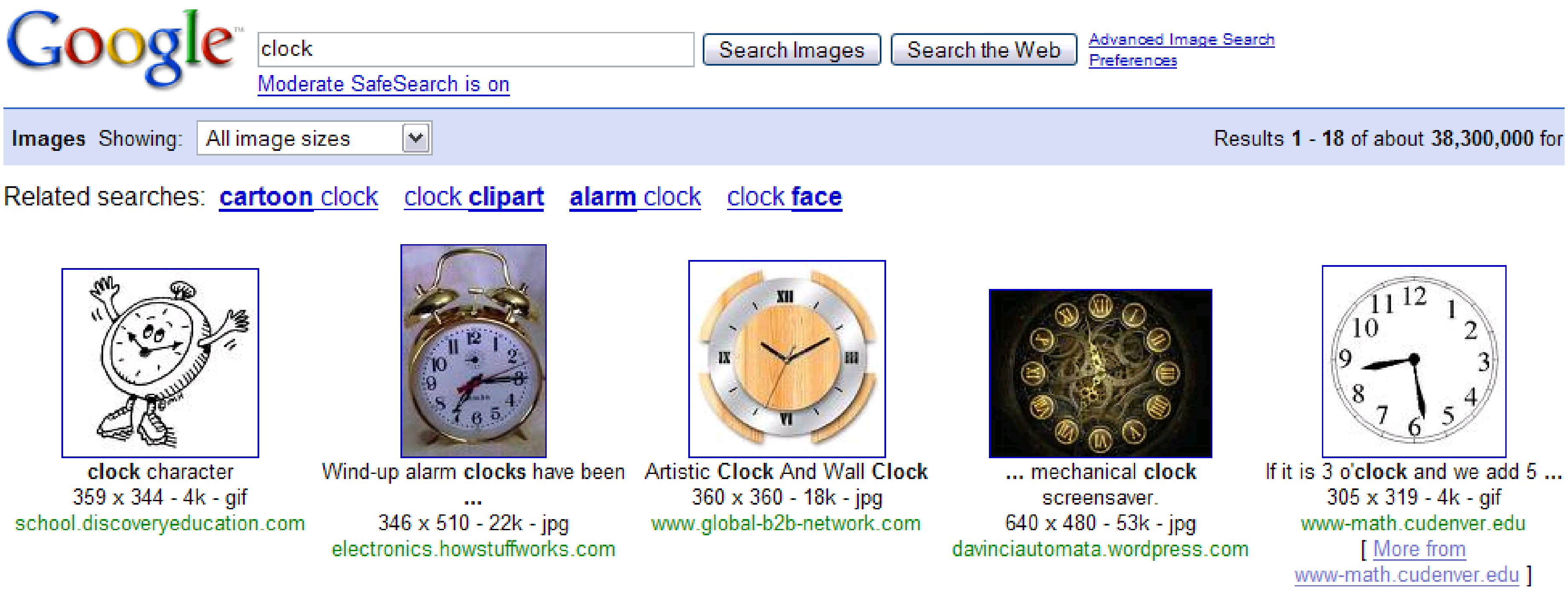


Google mugs

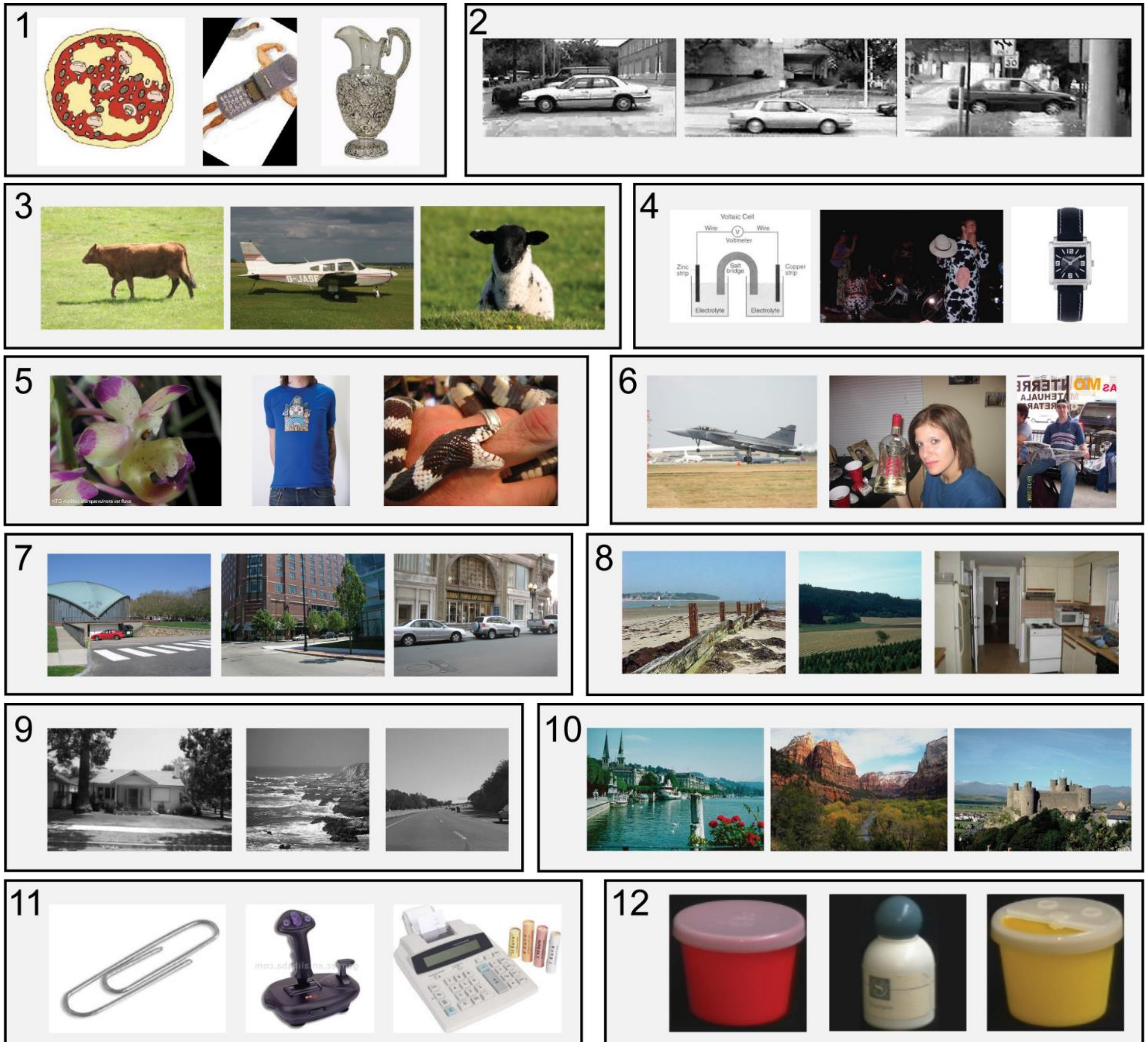


Mugs from LabelMe

CLOCK


Google
clock
Search Images
Search the Web
Advanced Image Search
Preferences
Moderate SafeSearch is on
Images Showing: All image sizes
Results 1 - 18 of about 38,300,000 for
Related searches: cartoon clock clock clipart alarm clock clock face
clock character
359 x 344 - 4k - gif
school.discoveryeducation.com
Wind-up alarm clocks have been ...
346 x 510 - 22k - jpg
electronics.howstuffworks.com
Artistic Clock And Wall Clock
360 x 360 - 18k - jpg
www.global-b2b-network.com
... mechanical clock screensaver.
640 x 480 - 53k - jpg
davinciautomata.wordpress.com
If it is 3 o'clock and we add 5 ...
305 x 319 - 4k - gif
www-math.cudenver.edu
[More from www-math.cudenver.edu]

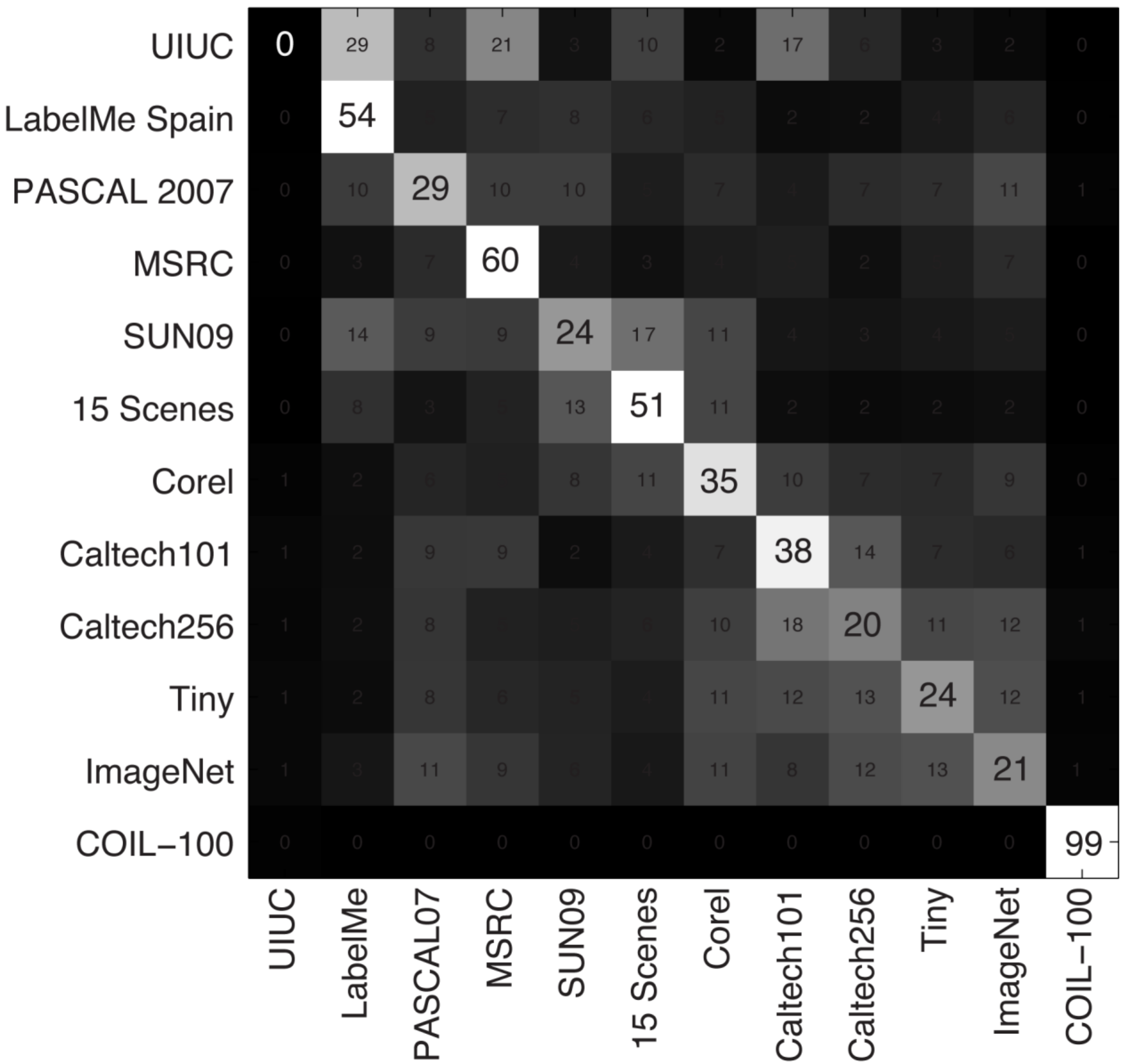
"Name That Dataset!" game



__ Caltech 101
__ Caltech 256
__ MSRC
__ UIUC cars
__ Tiny Images
__ Corel
__ PASCAL 2007
__ LabelMe
__ COIL-100
__ ImageNet
__ 15 Scenes
__ SUN'09

SVM plays *“Name that dataset!”*

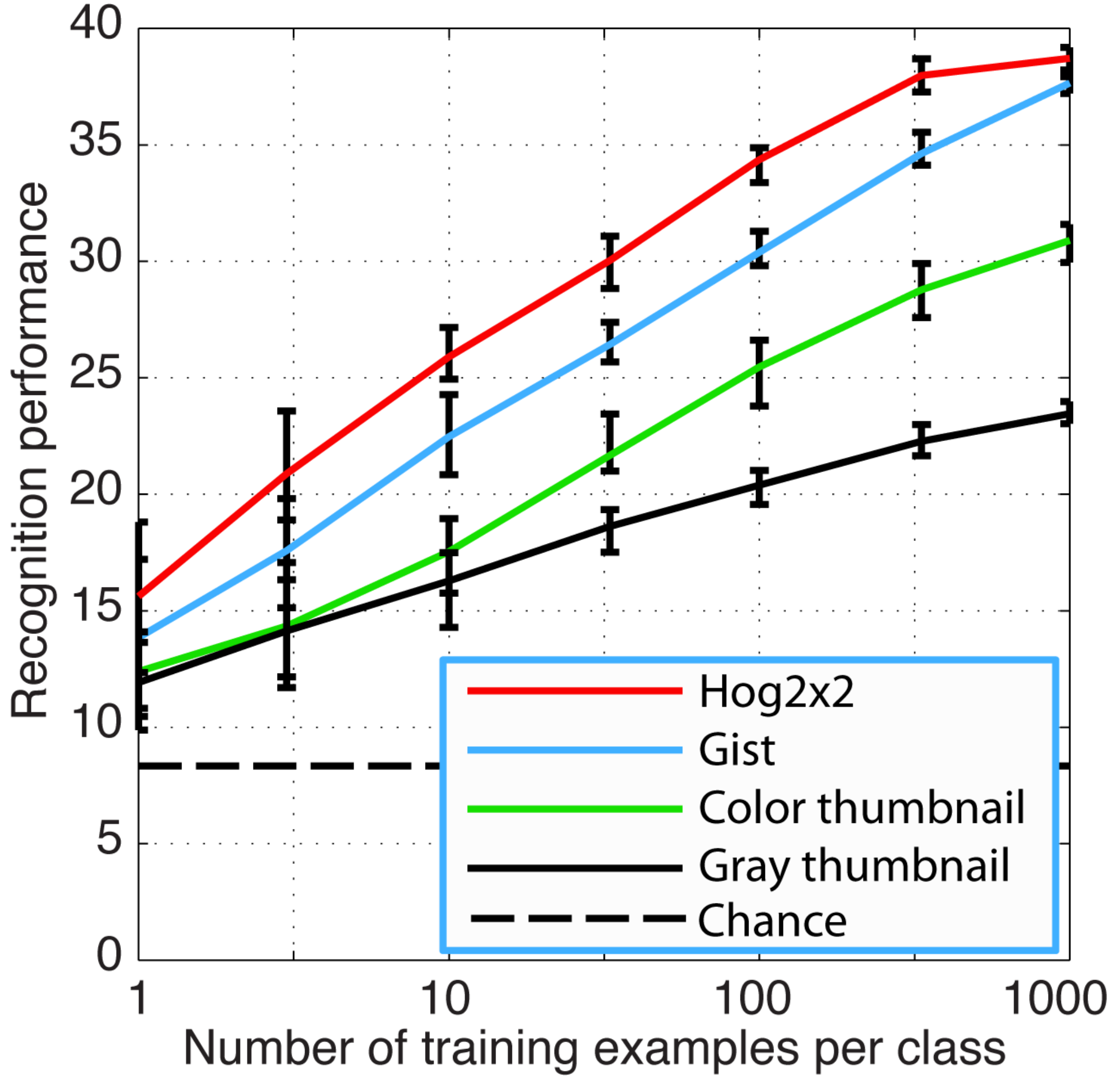
SVM plays *"Name that dataset!"*



- 12 1-vs-all classifiers
- Standard full-image features

- 39% performance (chance is 8%)

SVM plays *"Name that dataset!"*

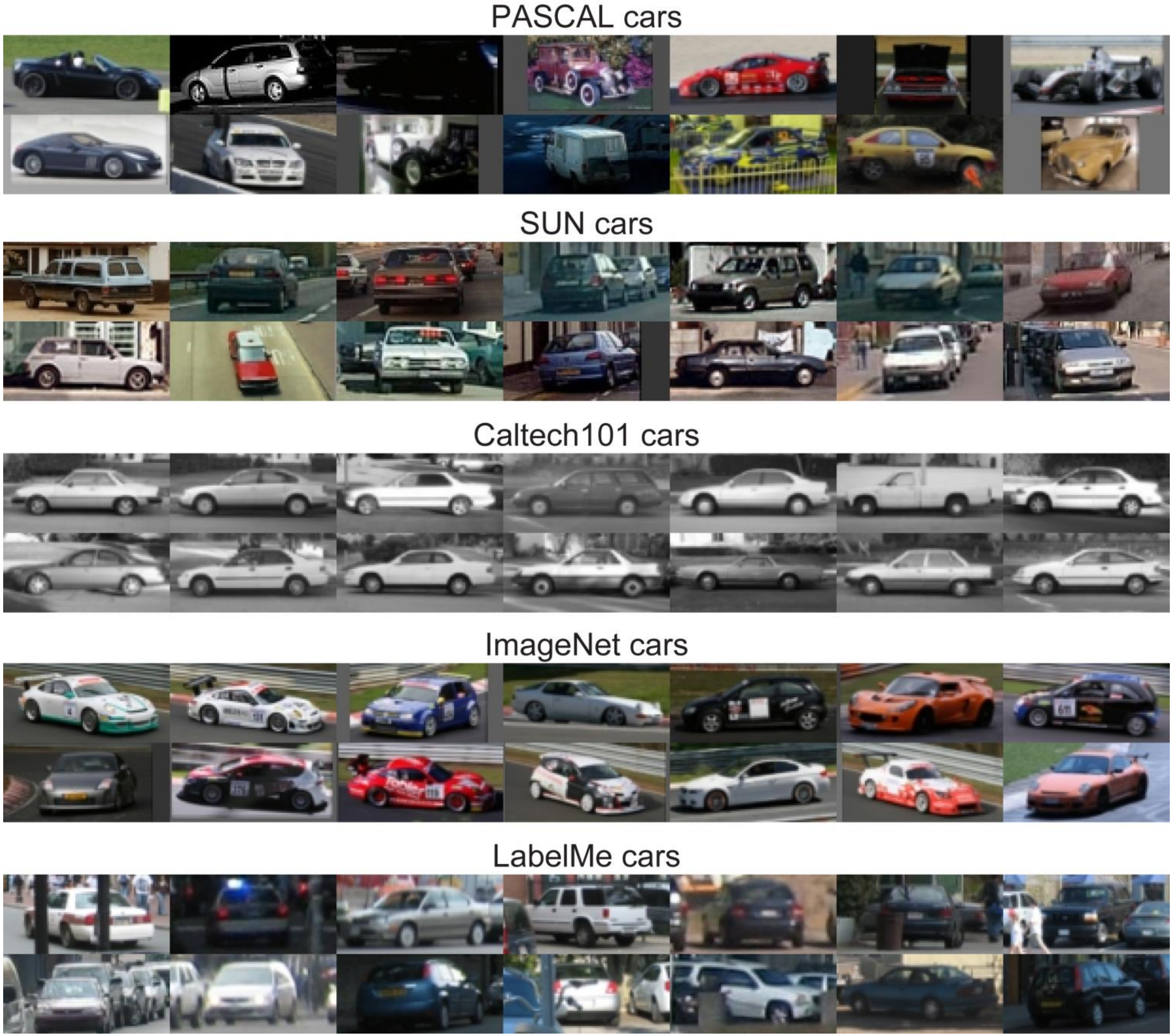


Datasets have different goals…

- Some are object-centric (e.g. Caltech, ImageNet)
- Otherwise are scene-centric (e.g. LabelMe, SUN’09)

- What about playing *“name that dataset”* on bounding boxes?

Similar results



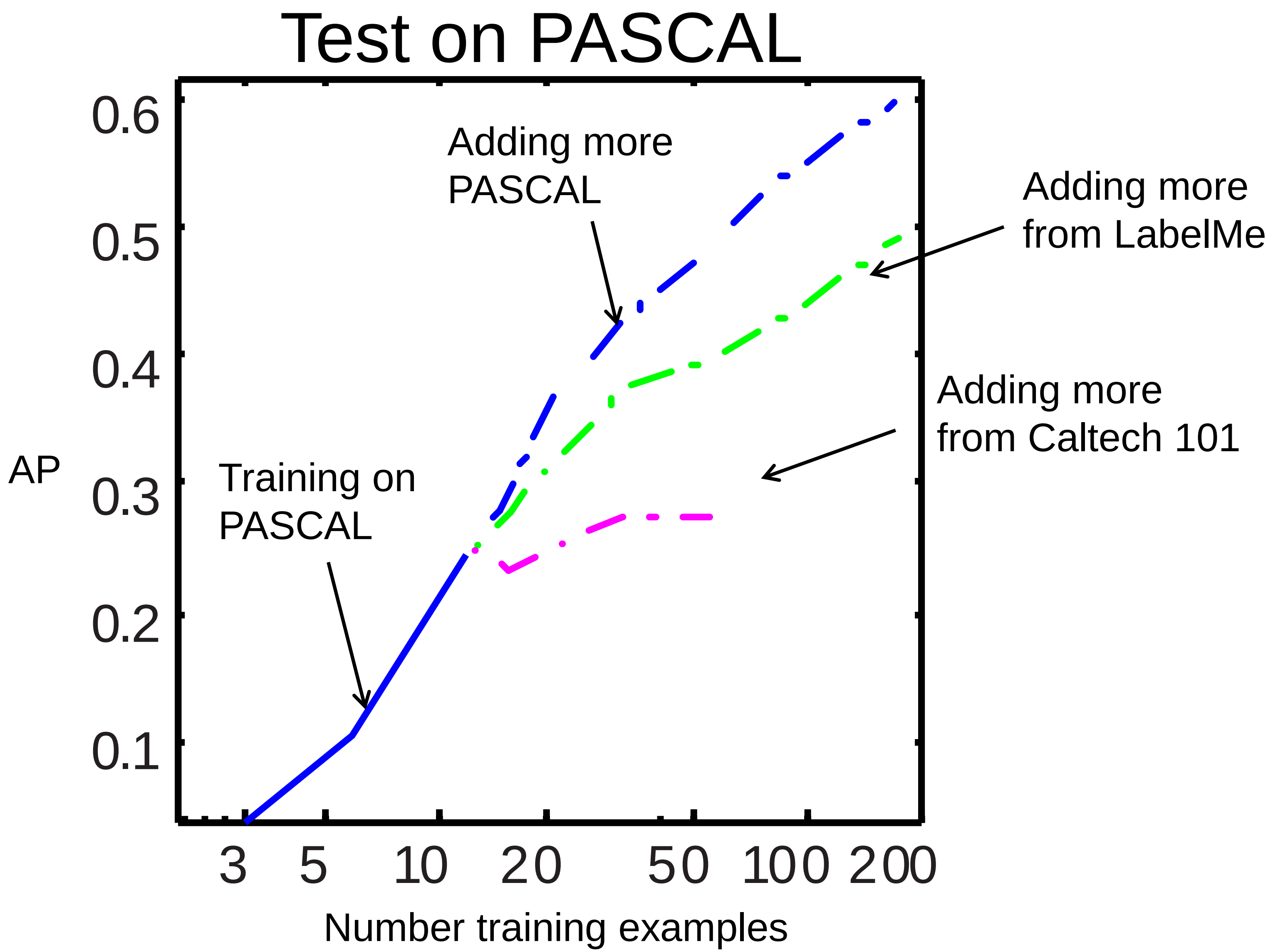
Performance: 61%
(chance: 20%)

Cross-Dataset Generalization

MSRC

Classifier trained on MSRC cars

Mixing datasets



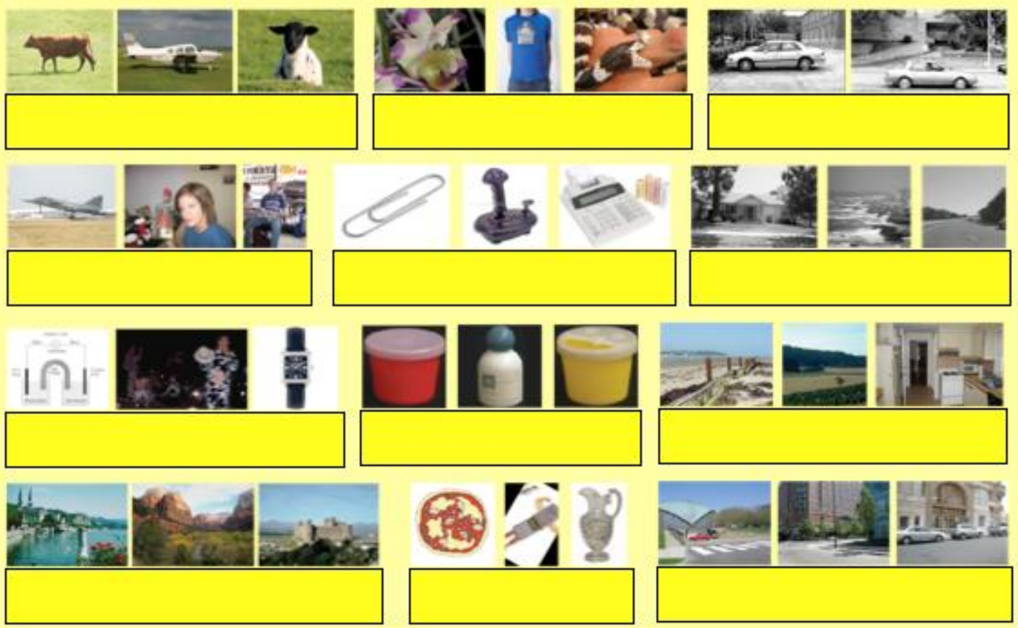
Unbiased Look at Dataset Bias

Antonio Torralba
MIT

Alyosha Efros
CMU

Let's play Name That Dataset!!!

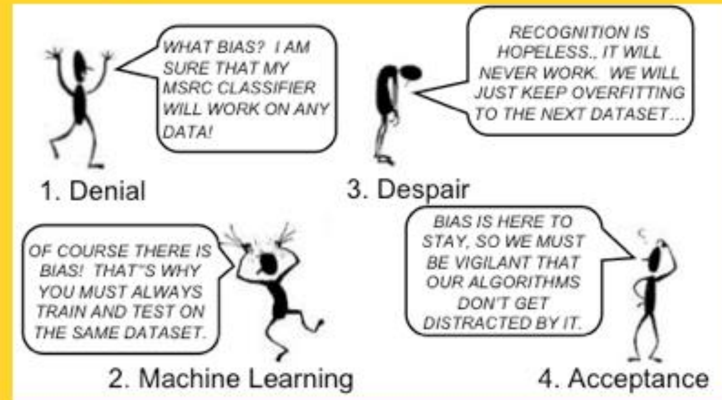
Given some images from twelve popular object recognition datasets, can you match the images with the dataset? Drag the dataset names into the yellow boxes bellow each set of images. **The score will appear once you have placed the 12 dataset names.**



Drag and drop each dataset name on the yellow boxes

Caltech 101	Caltech 256	MSRC	UIUC cars
Tiny Images	Corel	PASCAL 2007	LabelMe
COIL-100	ImageNet	15 Scenes	SUN'09

Four Stages of Dataset Grief



Download the paper

Acknowledgments

The authors would like to thank the Eyjafjallajokull volcano as well as the wonderful kirs at the Buvette in Jardin du Luxembourg for the motivation (former) and the inspiration (later) to write this paper. This work is part of a larger effort, joint with David Forsyth and Jay Yagnik, on understanding the benefits and pitfalls of using large data in vision. The paper was co-sponsored by ONR MURIs N000141010933 and N000141010934. No graduate students were harmed in the production of this paper. Authors are listed in order of increasing procrastination ability.