Part-based and local feature models for generic object recognition

May 28th, 2015

Yong Jae Lee

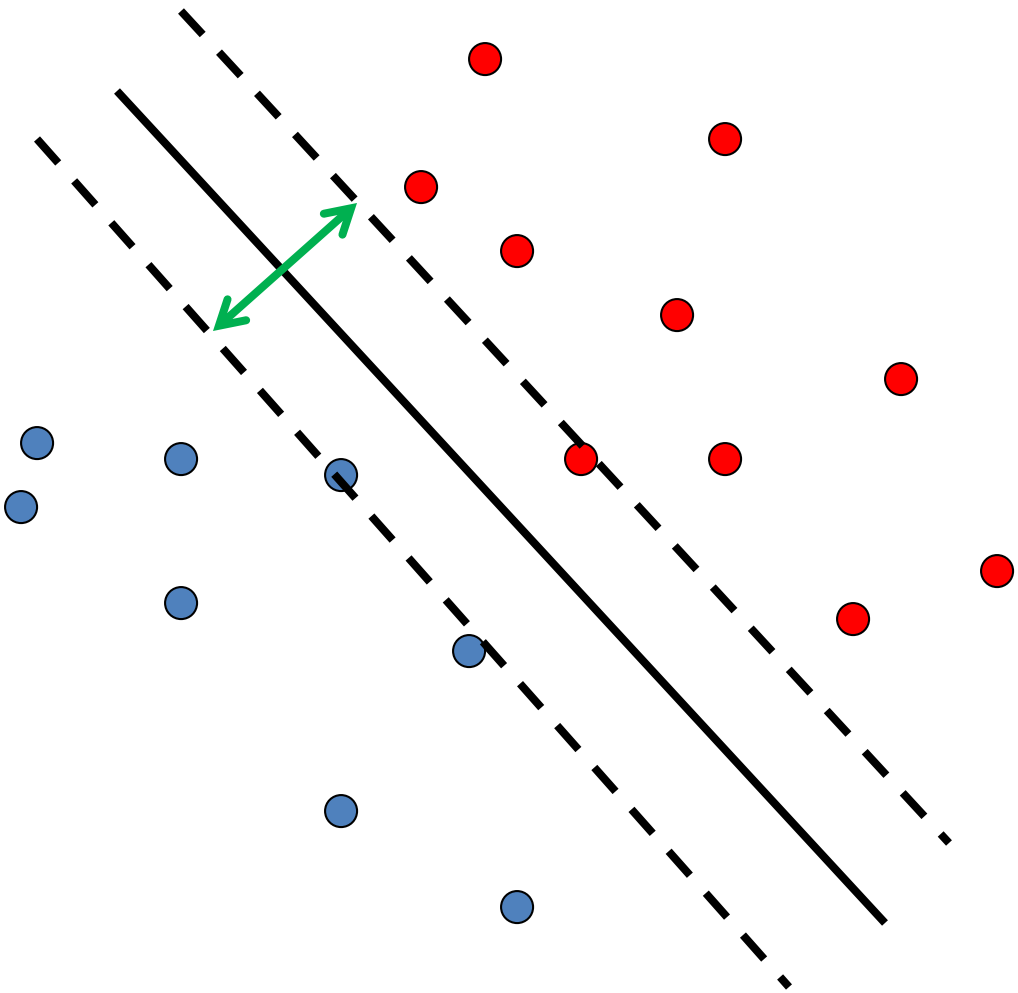
UC Davis

Announcements

- PS2 grades up on SmartSite
- PS2 stats:
  - Mean: 80.15
  - Standard Dev: 22.77

- Vote on piazza for last lecture content

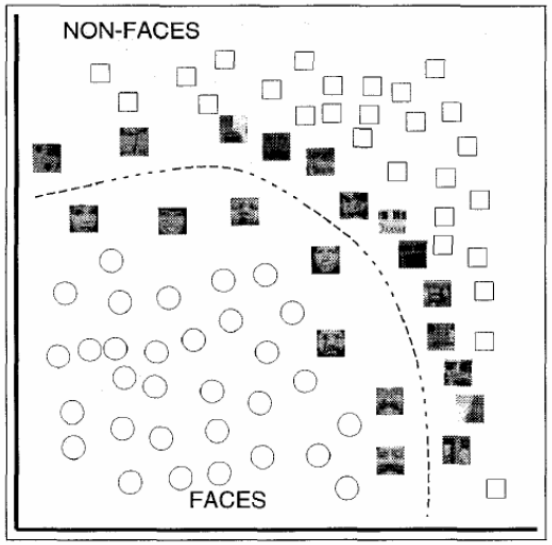
Support Vector Machines (SVMs)



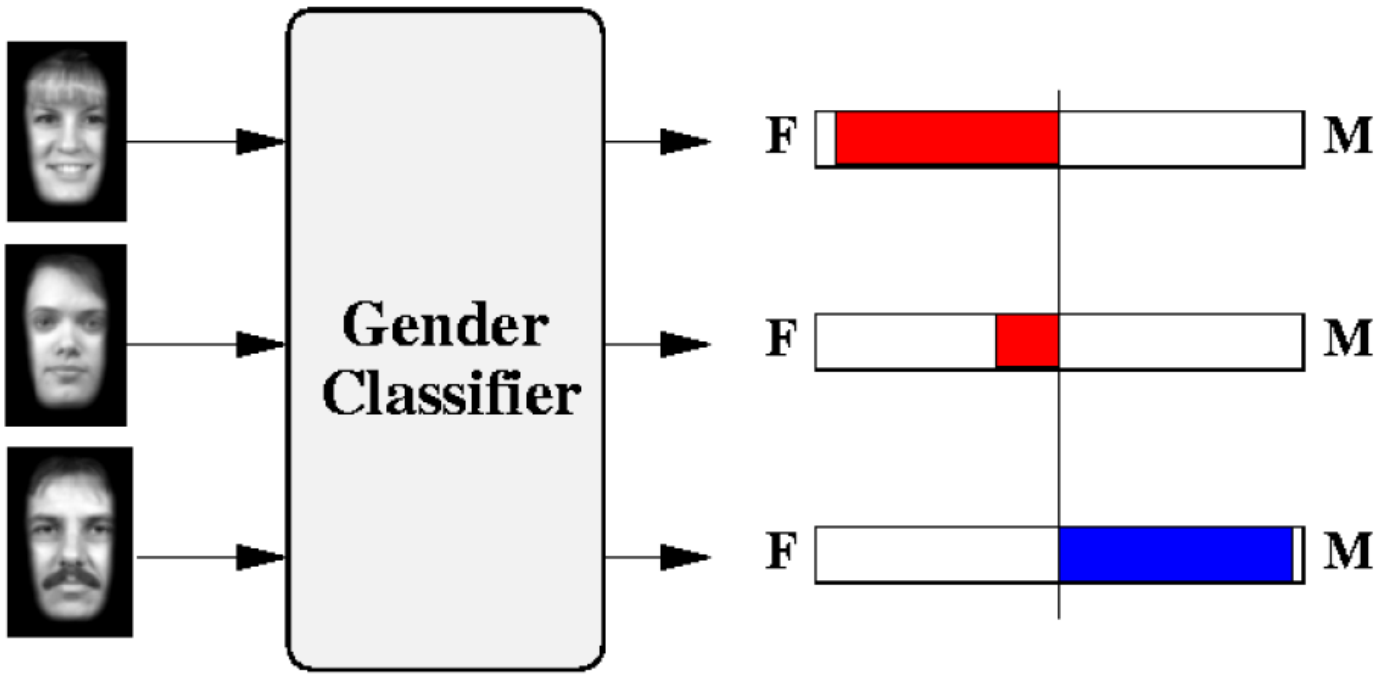
- Discriminative classifier based on *optimal separating line (for 2d case)*
- Maximize the *margin* between the positive and negative training examples

SVMs for recognition

1. Define your representation for each example.
2. Select a kernel function.
3. Compute pairwise kernel values between labeled examples
4. Use this “kernel matrix” to solve for SVM support vectors & weights.
5. To classify a new example: compute kernel values between new input and support vectors, apply weights, check sign of output.



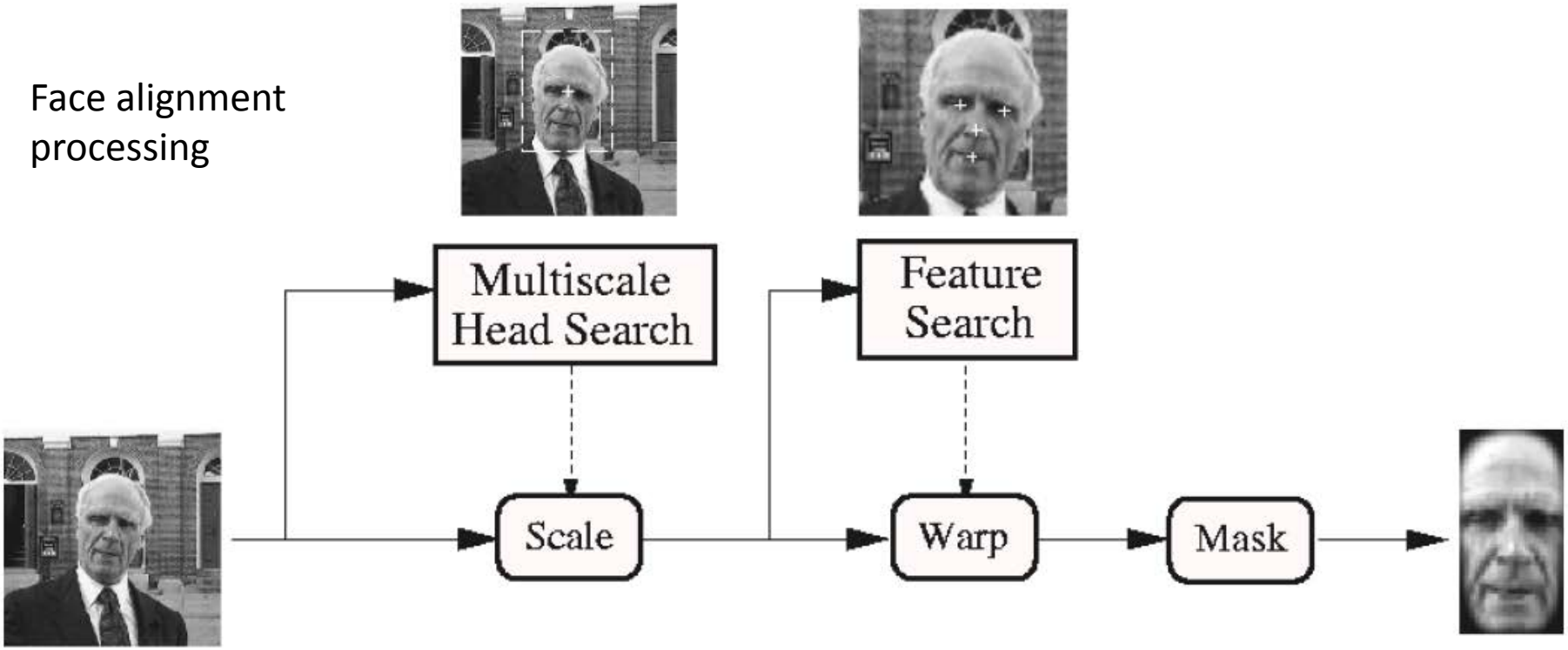
Example: learning gender with SVMs



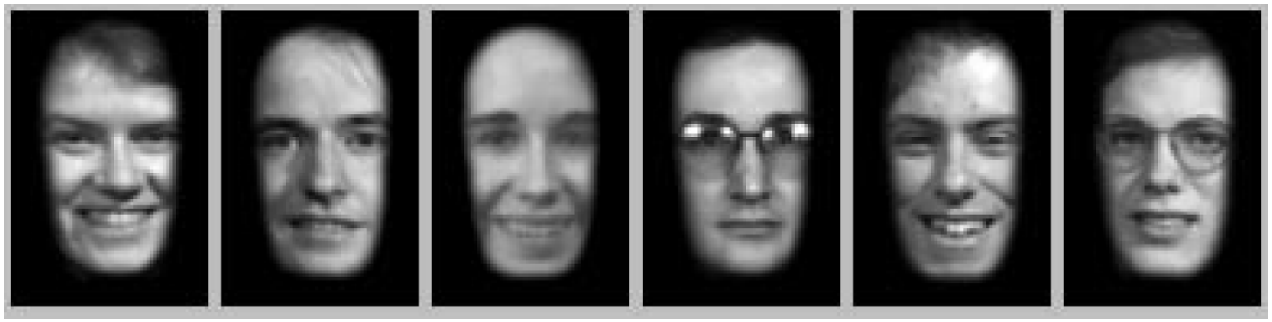
Moghaddam and Yang, Learning Gender with Support Faces, TPAMI 2002.

Moghaddam and Yang, Face & Gesture 2000.

Face alignment processing



Processed faces



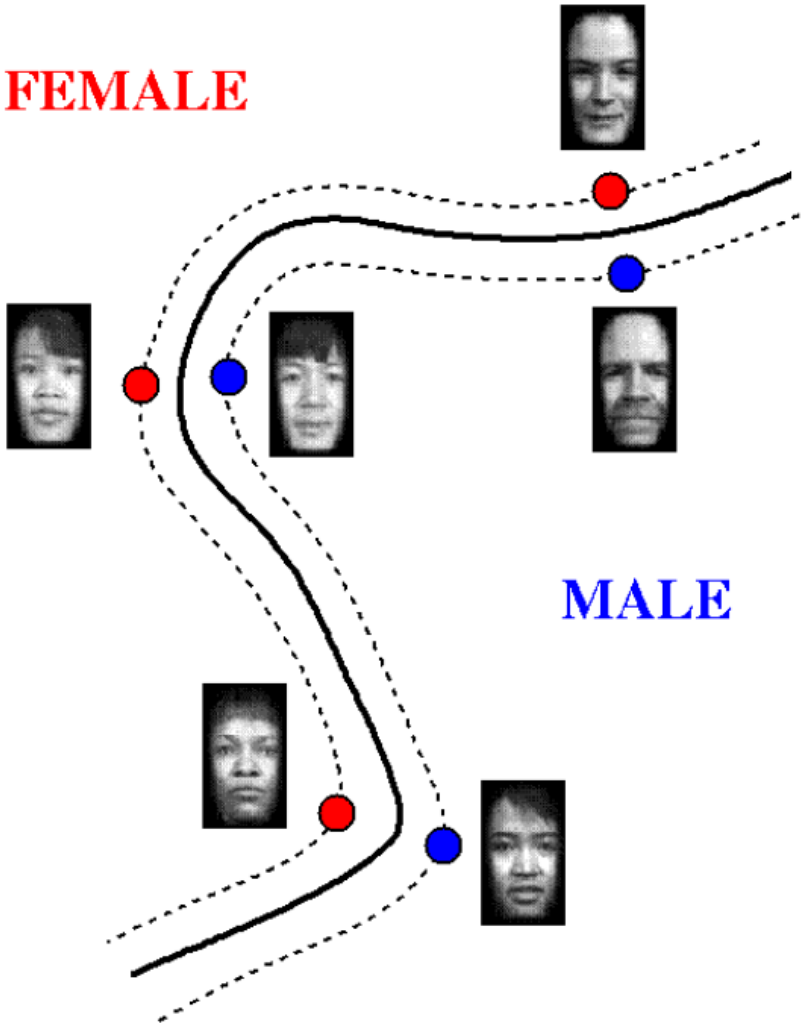
Moghaddam and Yang, Learning Gender with Support Faces, TPAMI 2002.

Learning gender with SVMs

- Training examples:
  - 1044 males
  - 713 females
- Experiment with various kernels, select Gaussian RBF

$$K(\mathbf{x_i}, \mathbf{x_j}) = \exp\left(-\frac{\left\|\mathbf{x_i} - \mathbf{x_j}\right\|^2}{2\sigma^2}\right)$$

Support Faces



Moghaddam and Yang, Learning Gender with Support Faces, TPAMI 2002.

Classifier Performance

Classifier	Error Rate		
	Overall	Male	Female
SVM with RBF kernel	**3.38%**	**2.05%**	**4.79%**
SVM with cubic polynomial kernel	**4.88%**	**4.21%**	**5.59%**
Large Ensemble of RBF	5.54%	4.59%	6.55%
Classical RBF	7.79%	6.89%	8.75%
Quadratic classifier	10.63%	9.44%	11.88%
Fisher linear discriminant	13.03%	12.31%	13.78%
Nearest neighbor	27.16%	26.53%	28.04%
Linear classifier	58.95%	58.47%	59.45%

Moghaddam and Yang, Learning Gender with Support Faces, TPAMI 2002.

Gender perception experiment: How well can humans do?

- Subjects:
  - 30 people (22 male, 8 female)
  - Ages mid-20's to mid-40's
- Test data:
  - 254 face images
  - Low res (6 males, 4 females)
  - High res versions
- Task:
  - Classify as male or female, forced choice
  - No time limit

Moghaddam and Yang, Face & Gesture 2000.

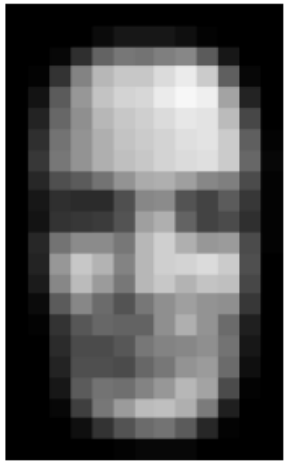
Gender perception experiment: How well can humans do?

Stimuli →

84 x 48

N = 4032

21 x 12



N = 252

Results →

High-Res	Low-Res
6.54%	30.7%
Error	Error

$\sigma = 3.7\%$

Moghaddam and Yang, Face & Gesture 2000.

Human vs. Machine

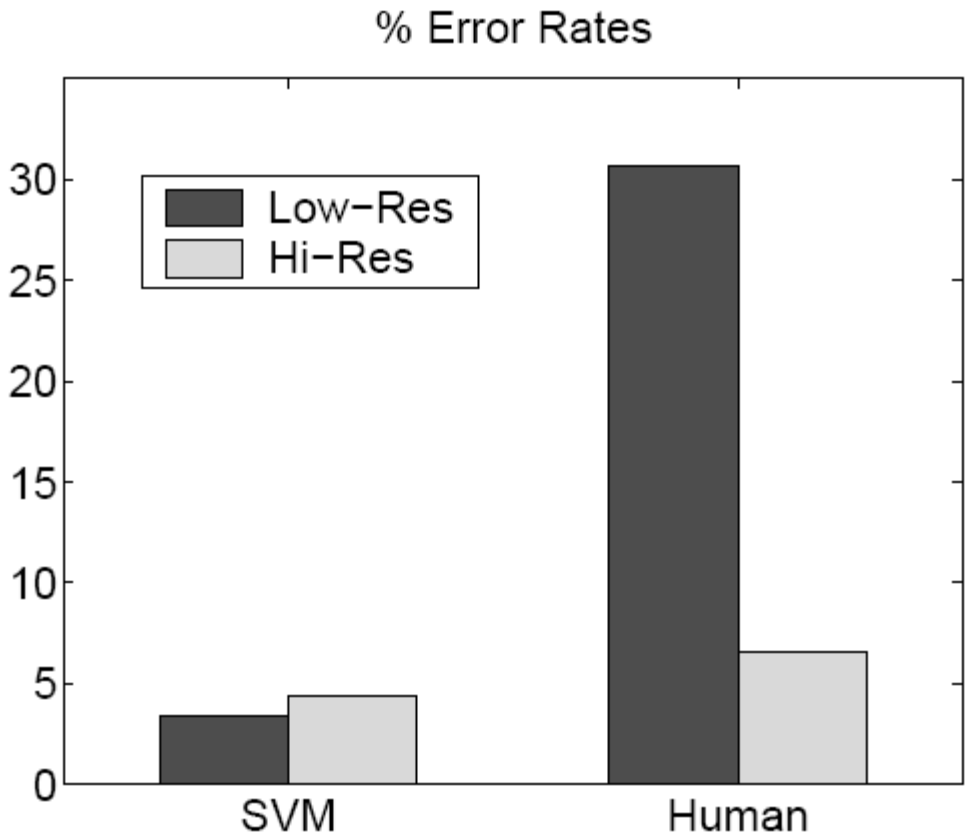
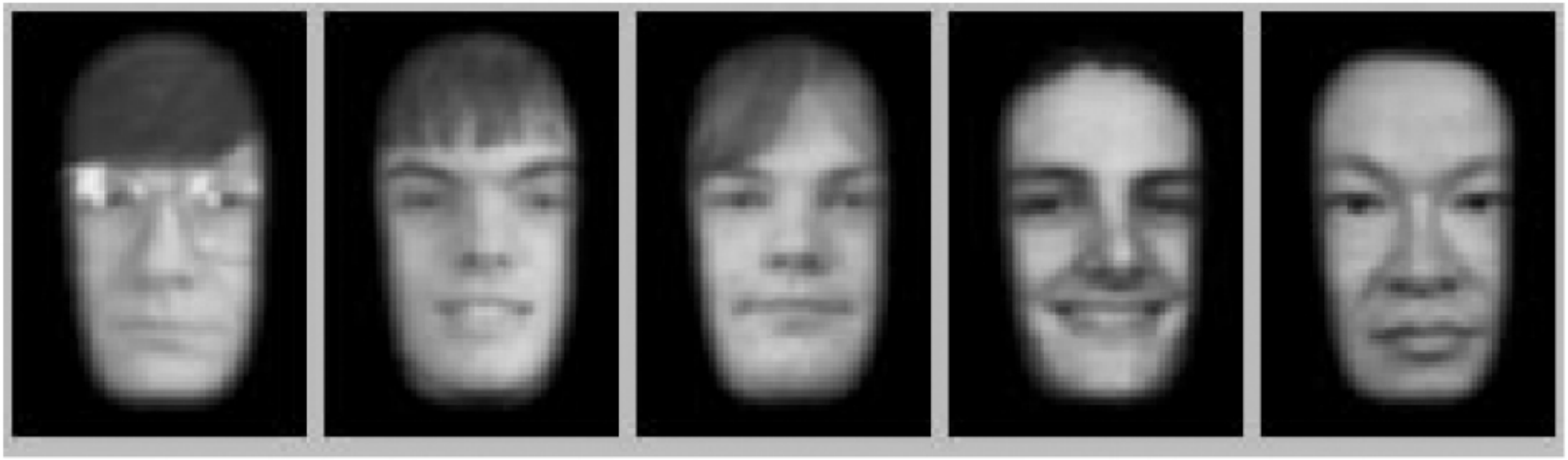


Figure 6. SVM vs. Human performance

- SVMs performed better than any single human test subject, at either resolution

Hardest examples for humans



Top five human misclassifications

Moghaddam and Yang, Face & Gesture 2000.

Questions

- What if the features are not 2d?
- What if the data is not linearly separable?
- **What if we have more than just two categories?**

Multi-class SVMs

- Achieve multi-class classifier by combining a number of binary classifiers

- **<u>One vs. all</u>**
  - Training: learn an SVM for each class vs. the rest
  - Testing: apply each SVM to test example and assign to it the class of the SVM that returns the highest decision value

- **<u>One vs. one</u>**
  - 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
  - http://www.csie.ntu.edu.tw/~cjlin/libsvm/
  - Kernel-based framework is very powerful, flexible
  - Often a sparse set of support vectors – compact at test time
  - Work very well in practice, even with very small training sample sizes
- Cons
  - No "direct" multi-class SVM, must combine two-class SVMs
  - Can be tricky to select best kernel function for a problem
  - 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

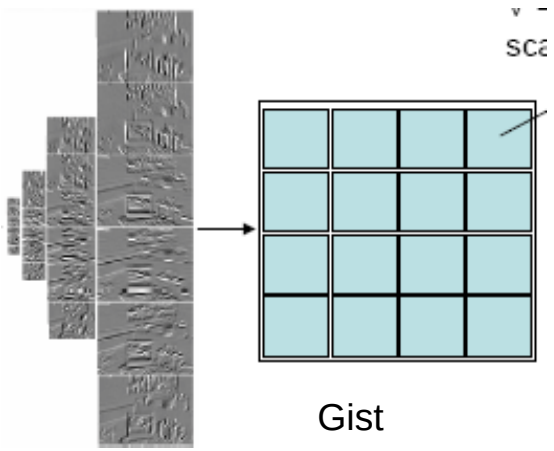
Adapted from Lana Lazebnik

Previously

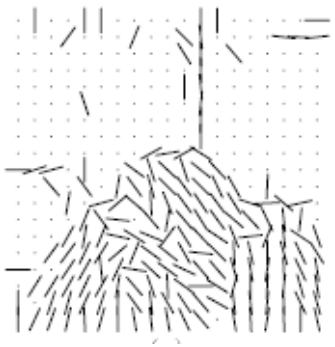
- Discriminative classifiers
  - Boosting
  - Nearest neighbors
  - Support vector machines

- Useful for object recognition when combined with “window-based” or holistic appearance descriptors

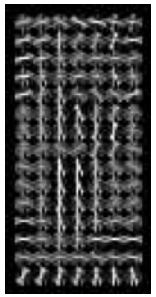
Global window-based appearance representations



Gist



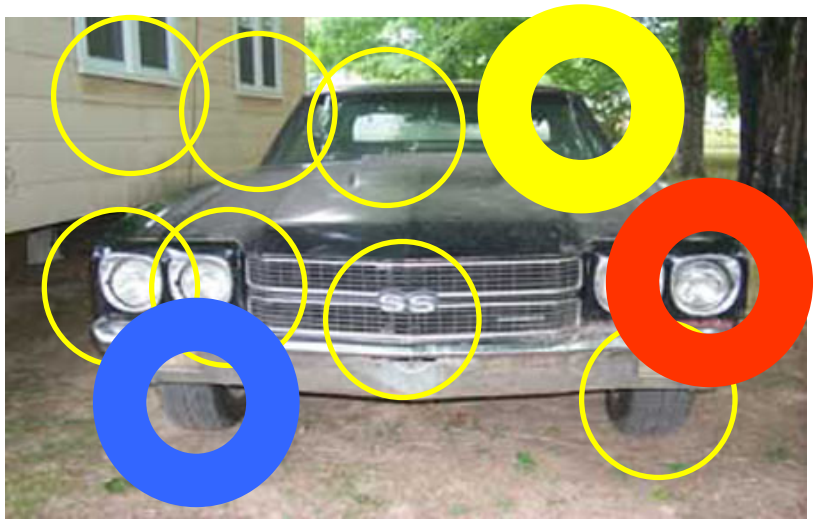
Map of local orientations



Histograms of oriented gradients

- These examples are truly global; each pixel in the window contributes to the representation.
- Classifier can account for relative relevance…
- *When might this not be ideal?*

Part-based and local feature models for recognition

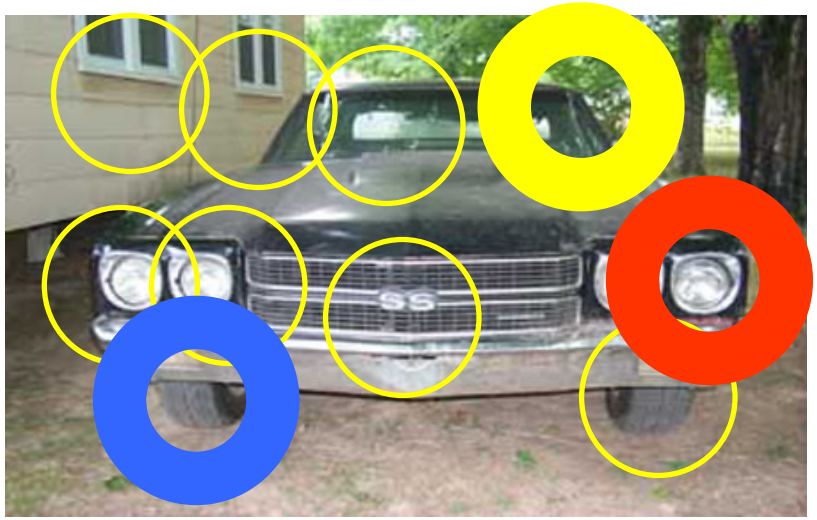


Main idea:

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

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

Part-based and local feature models for recognition

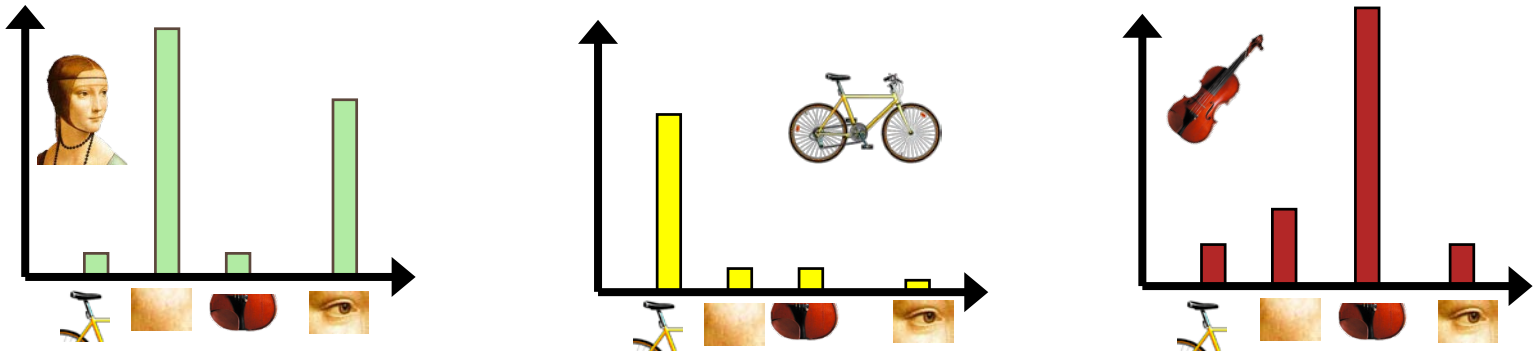


We'll look at three forms:

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

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.



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?

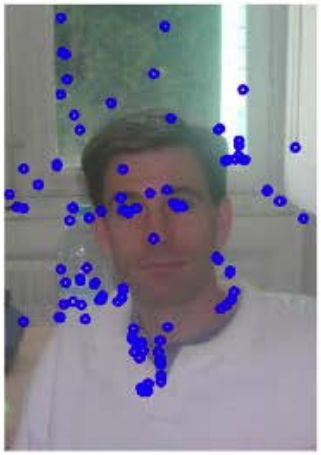
Confusion matrix

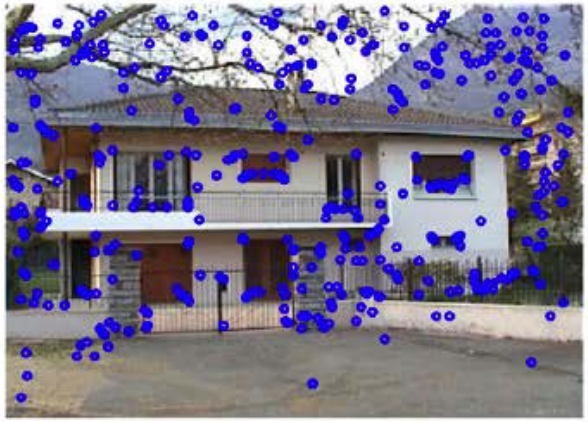
True classes →	*faces*	*buildings*	*trees*	*cars*	*phones*	*bikes*	*books*
faces	**76**	4	2	3	4	4	13
buildings	2	**44**	5	0	5	1	3
trees	3	2	**80**	0	0	5	0
cars	4	1	0	**75**	3	1	4
phones	9	15	1	16	**70**	14	11
bikes	2	15	12	0	8	**73**	0
books	4	19	0	6	7	2	**69**

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

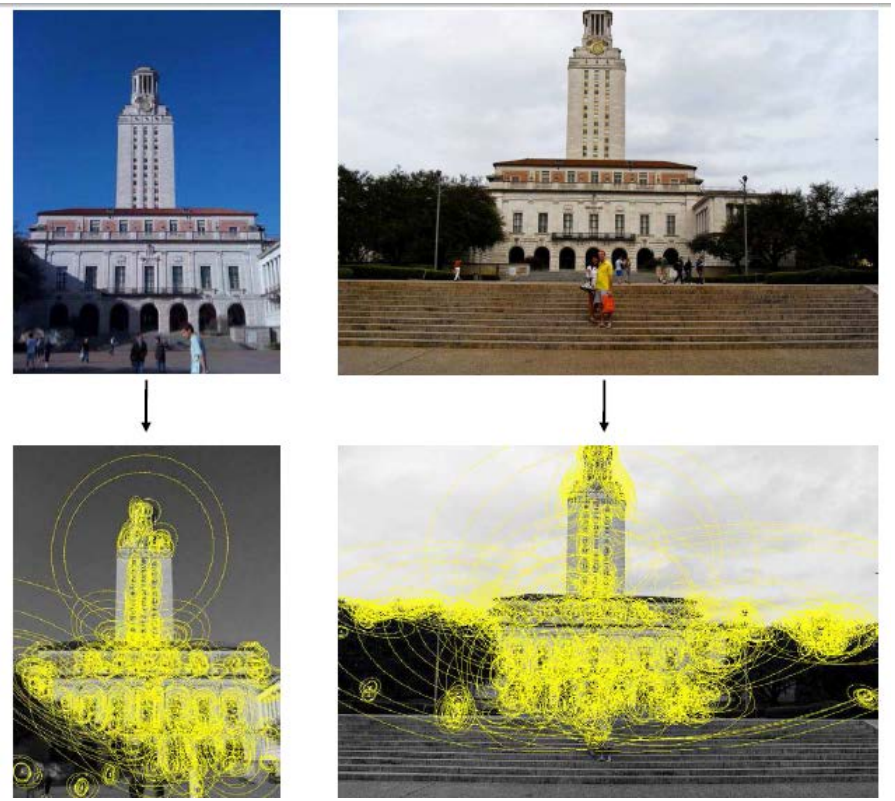
Csurka et al. 2004

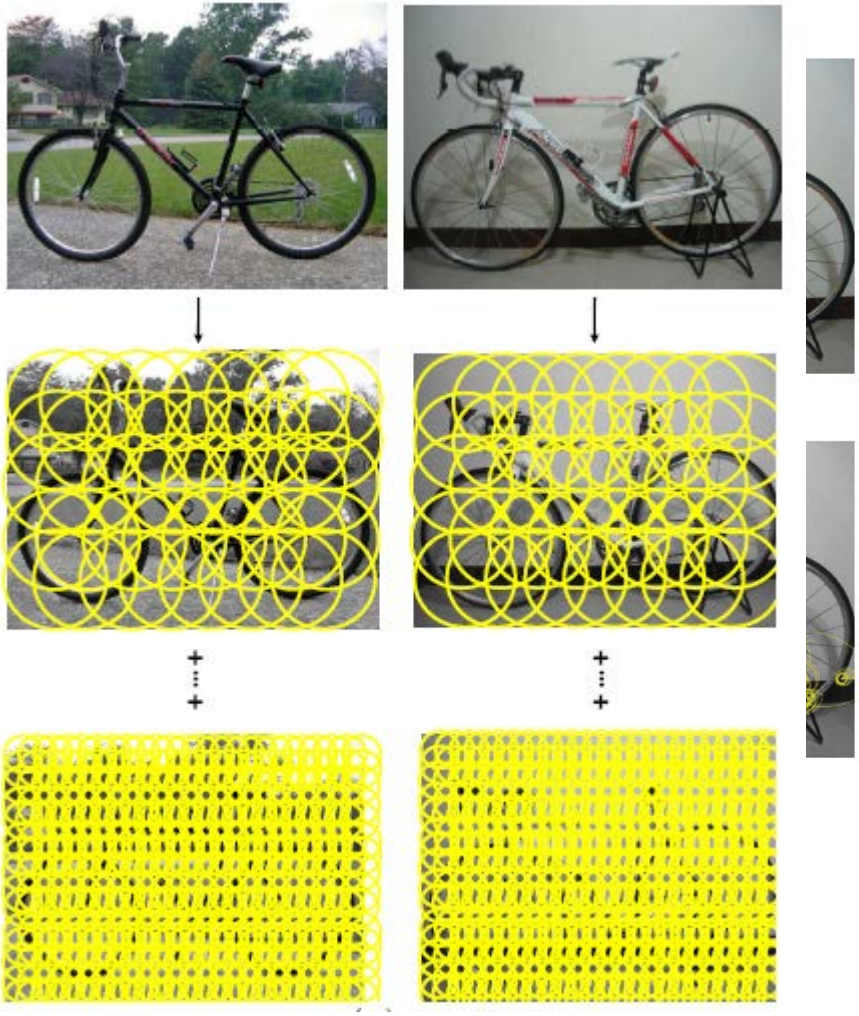
Clutter…or context?





Sampling strategies

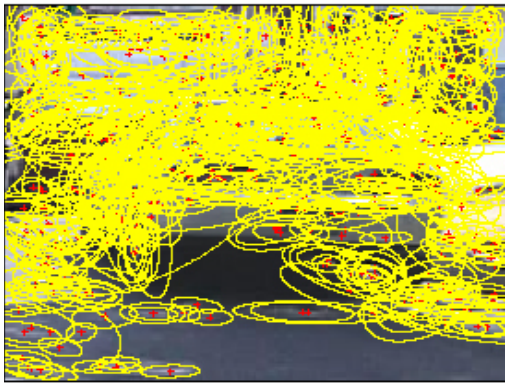




Specific object

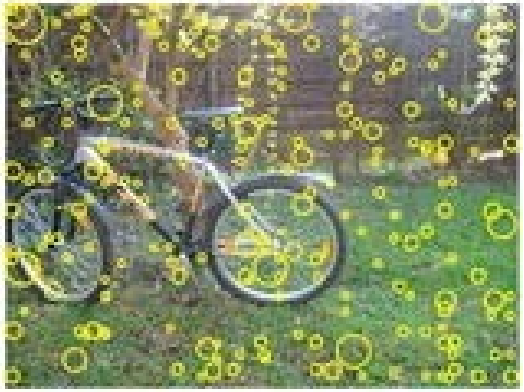
Category

Sampling strategies

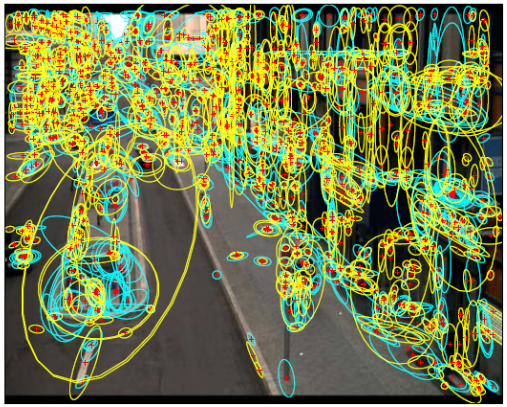


Sparse, at interest points

Dense, uniformly



Randomly



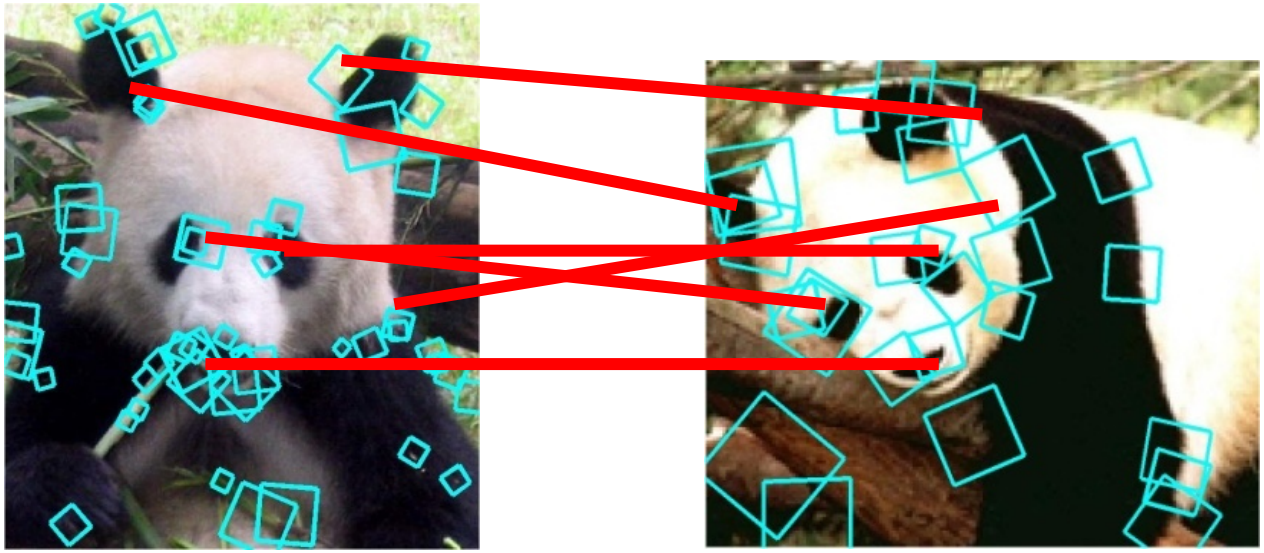
Multiple interest operators

- 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]

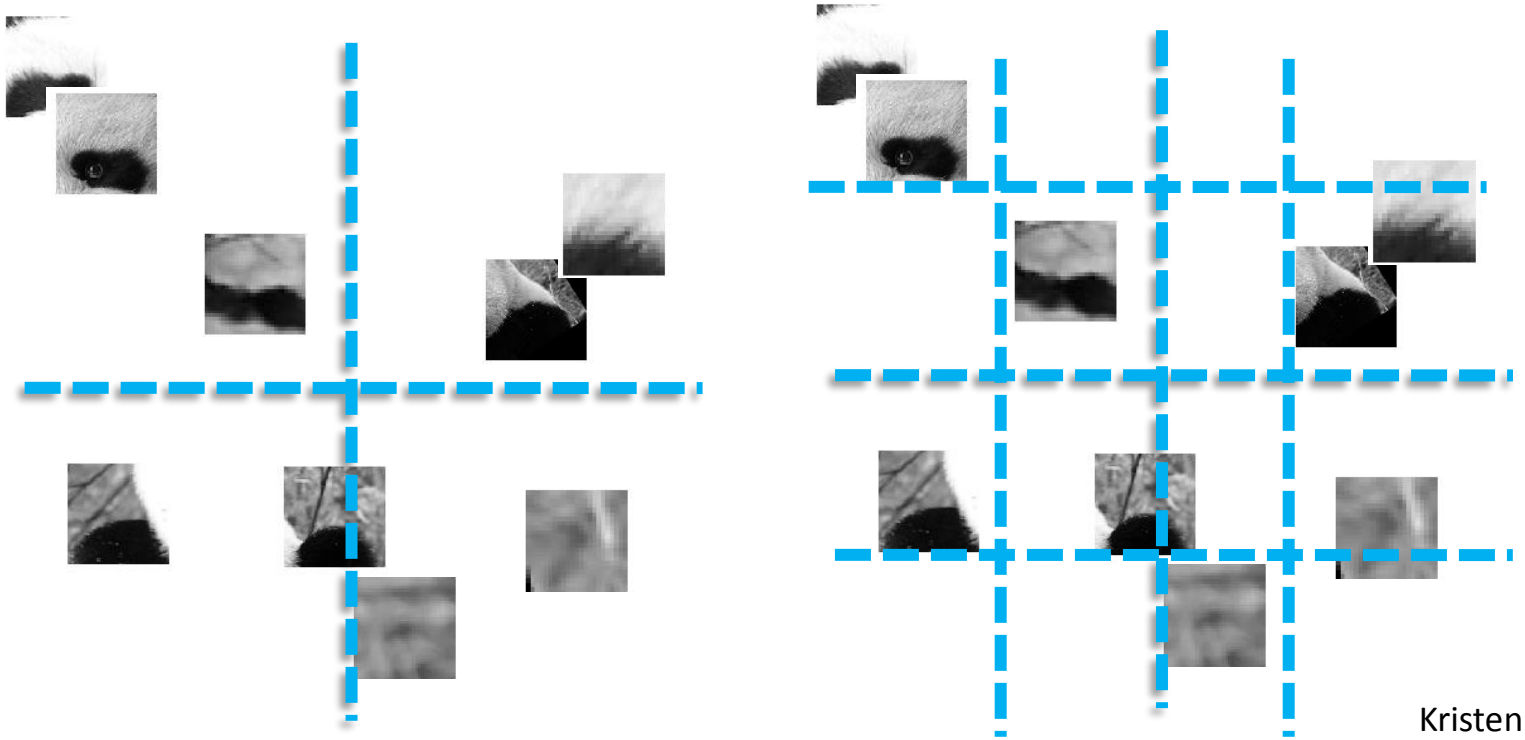
Local feature correspondence for generic object categories



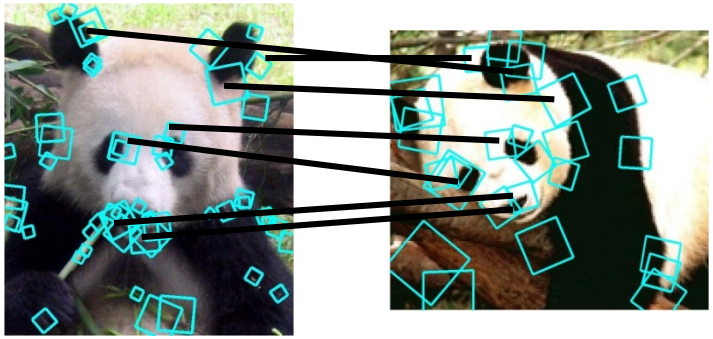
$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$ $\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$

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…



Partially matching sets of features



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

Optimal match: O(m^3)
Greedy match: O(m^2 log m)
Pyramid match: O(m)

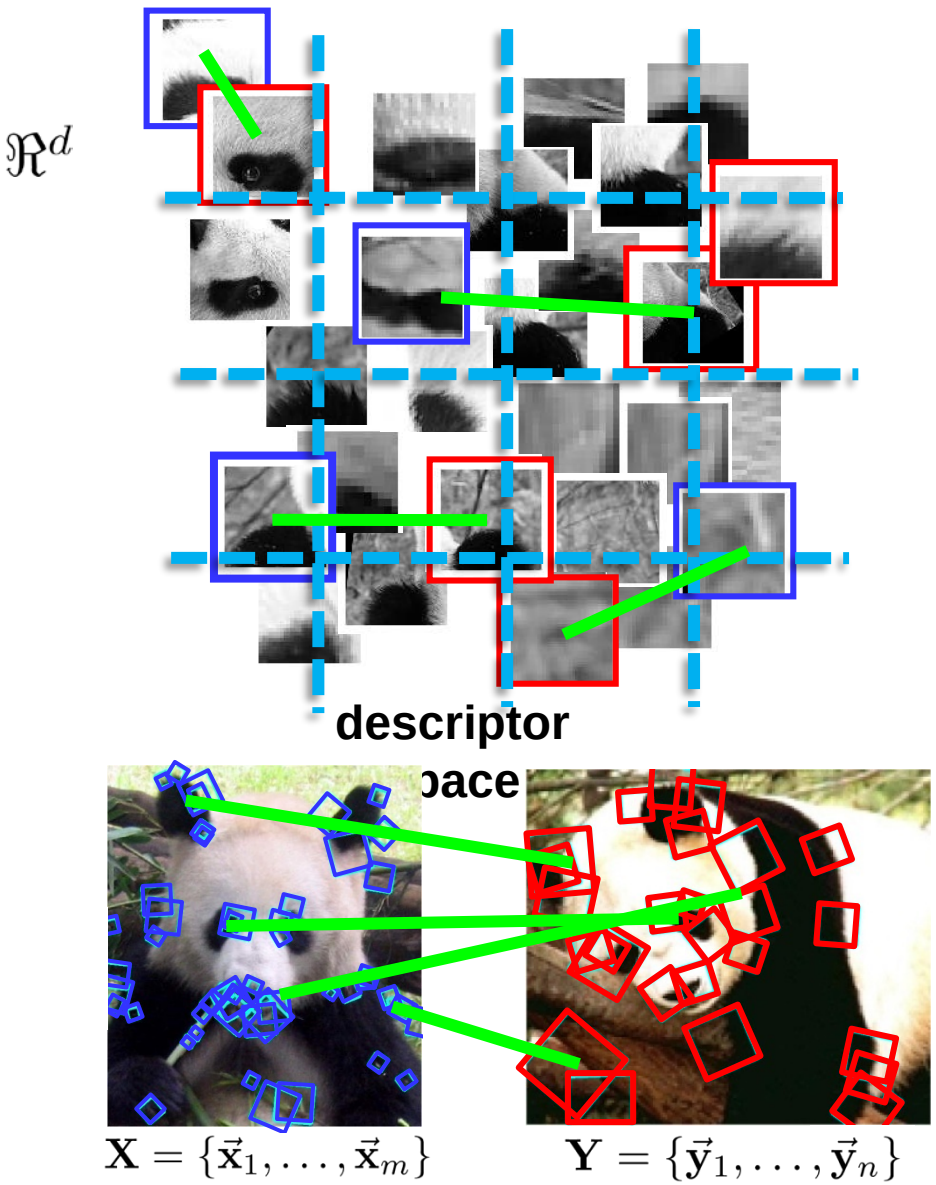
(**m**=num pts)

$$\min_{\pi:\mathbf{X}\rightarrow\mathbf{Y}} \sum_{\mathbf{x}_i \in \mathbf{X}} ||\mathbf{x}_i - \pi(\mathbf{x}_i)||$$

\nate matching kernel that makes it practical to compare large sets of features based on their partial correspondences.

[Previous work: Indyk & Thaper, Bartal, Charikar, Agarwal & Varadarajan, …]

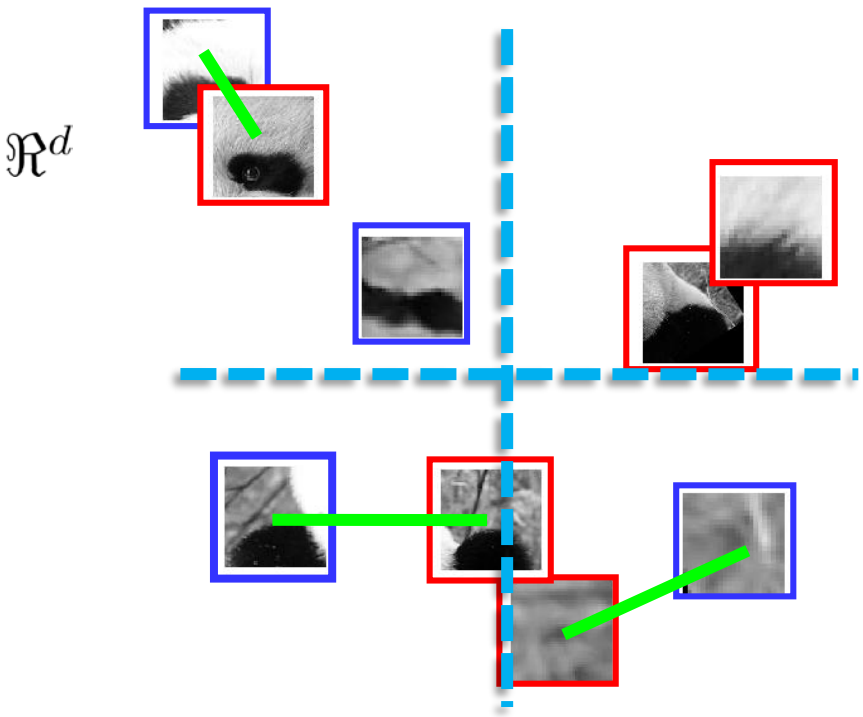
Pyramid match: main idea

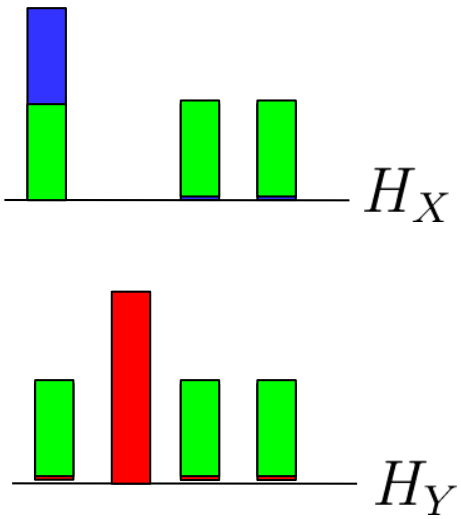


Feature space partitions serve to “match” the local descriptors within successively wider regions.

[Grauman & Darrell, ICCV 2005]

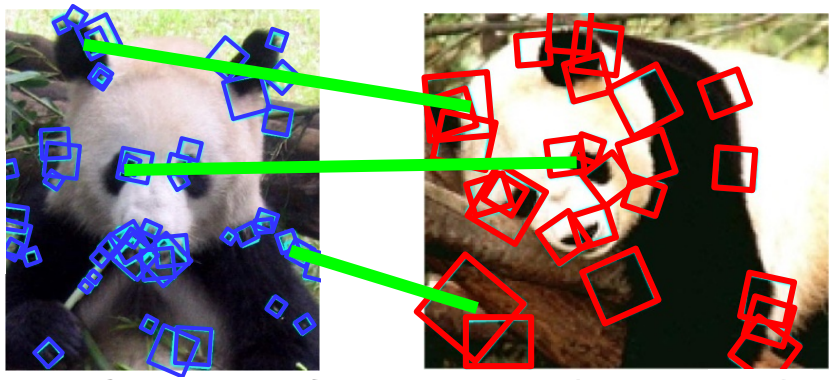
Pyramid match: main idea





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

$$= 3$$



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

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

[Grauman & Darrell, ICCV 2005]

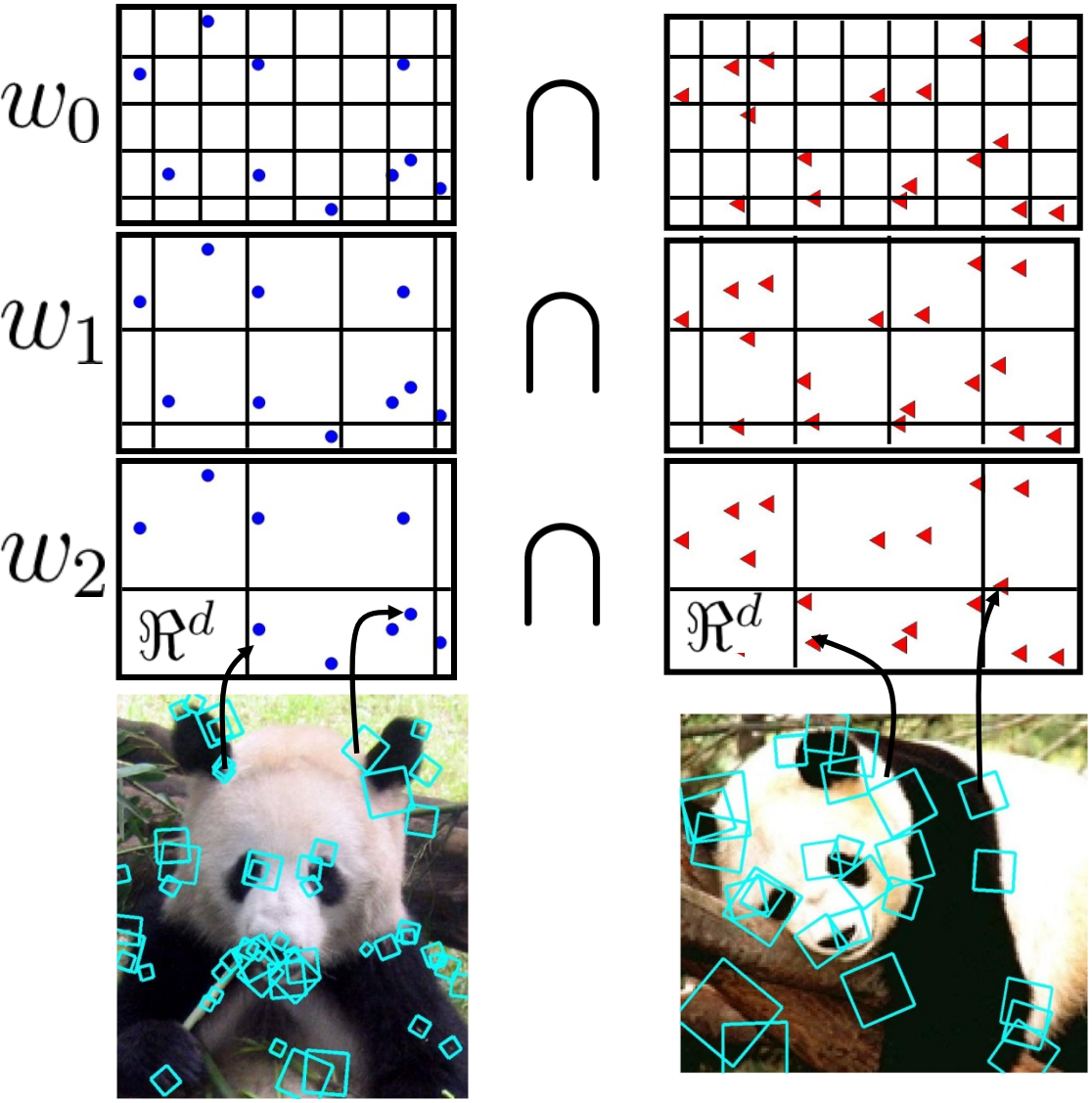
Pyramid match kernel

$$K_\Delta(X,Y) = \sum_{i=0}^{L} \underbrace{2^{-i}}_{\substack{\text{measures}\\ \text{difficulty of a}\\ \text{match at level } i}} \underbrace{\mathcal{I}\left(H_X^{(i)}, H_Y^{(i)}\right) - \mathcal{I}\left(H_X^{(i-1)}, H_Y^{(i-1)}\right)}_{\substack{\text{number of newly matched}\\ \text{pairs at level } i}}$$

- For similarity, weights inversely proportional to bin size (or may be learned)
- Normalize these kernel values to avoid favoring large sets

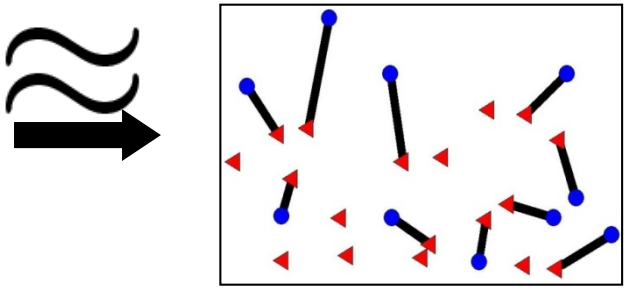
[Grauman & Darrell, ICCV 2005]

Pyramid match kernel



Optimal match: O(m³)

Pyramid match: O(mL)



optimal partial matching

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

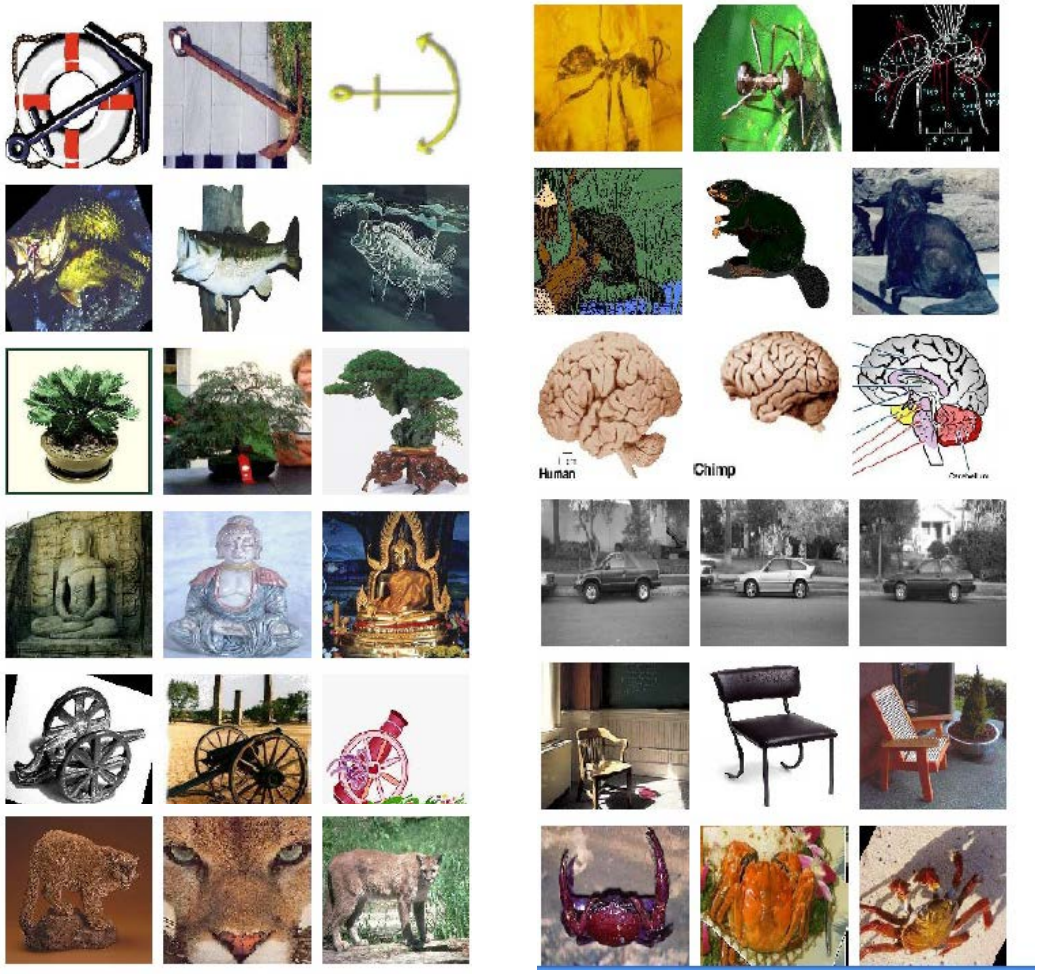
[Grauman & Darrell, ICCV 2005]

Highlights of the pyramid match

- Linear time complexity
- Formal bounds on expected error
- Mercer kernel
- Data-driven partitions allow accurate matches even in high-dim. feature spaces
- Strong performance on benchmark object recognition datasets

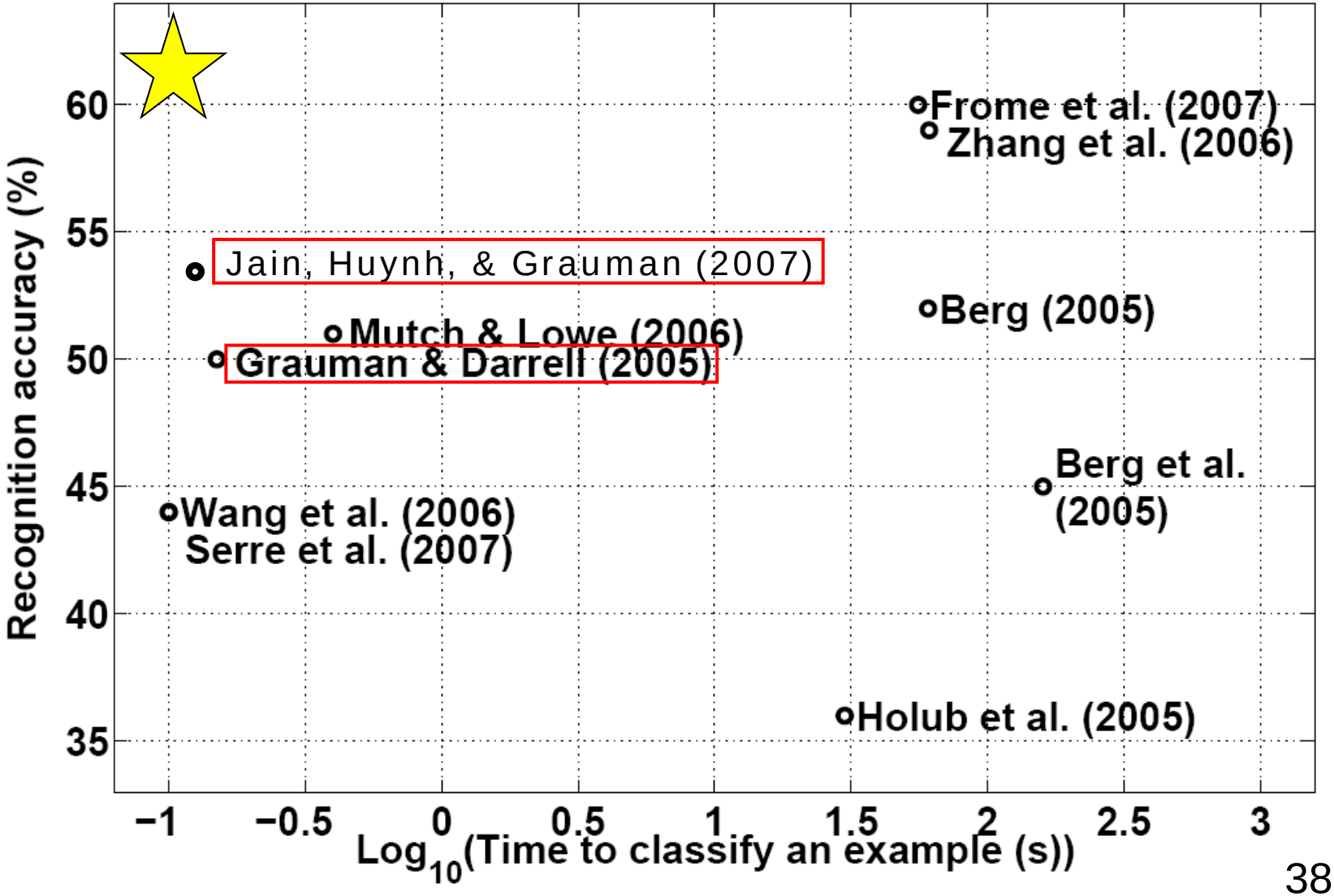
Example recognition results:
Caltech-101 dataset

- 101 categories
  40-800 images per class



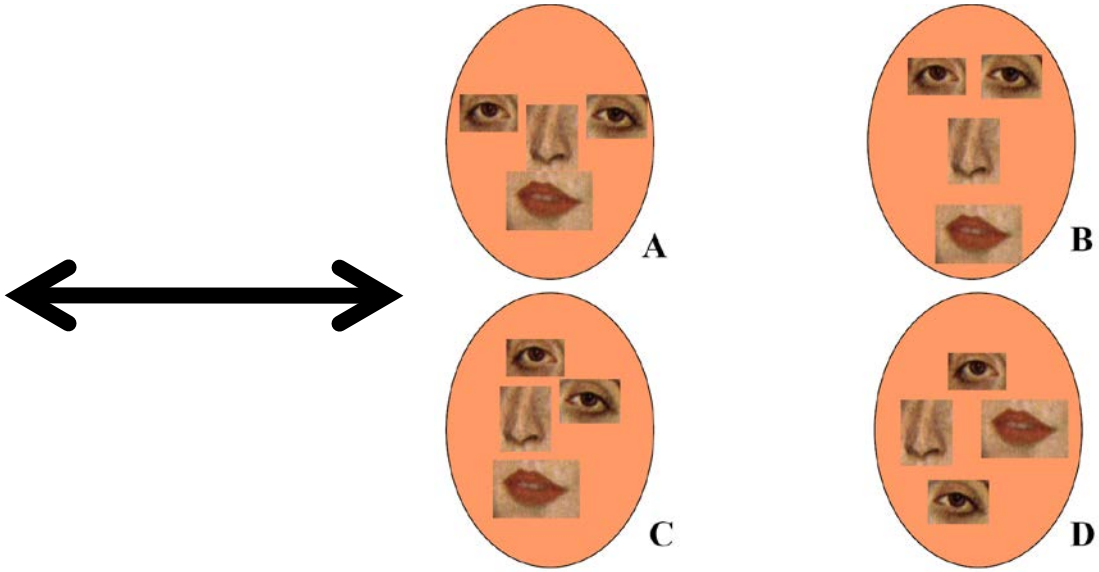
Data provided by Fei-Fei, Fergus, and Perona

Recognition results:
Caltech-101 dataset



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

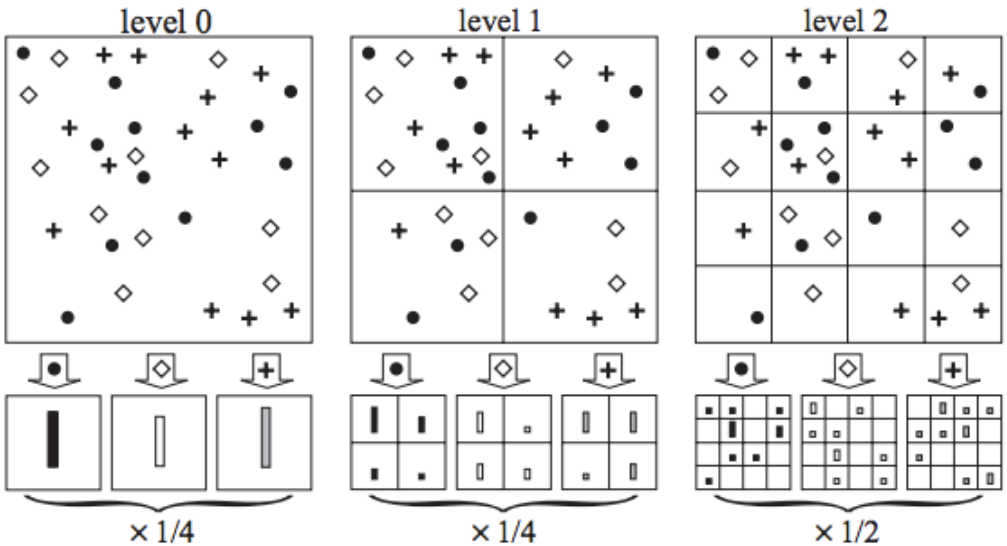
Too much?



Too little?

Spatial pyramid match

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



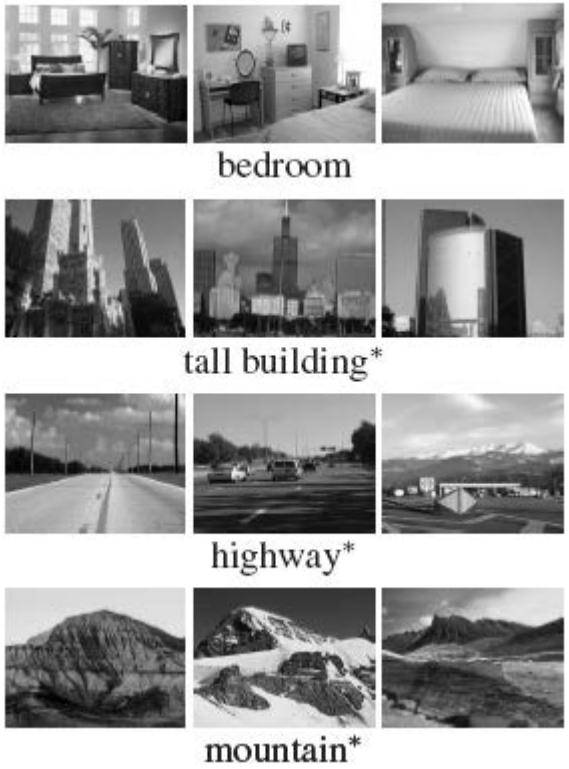
$$K^L(X,Y) = \sum_{m=1}^{M} \kappa^L(X_m, Y_m)$$

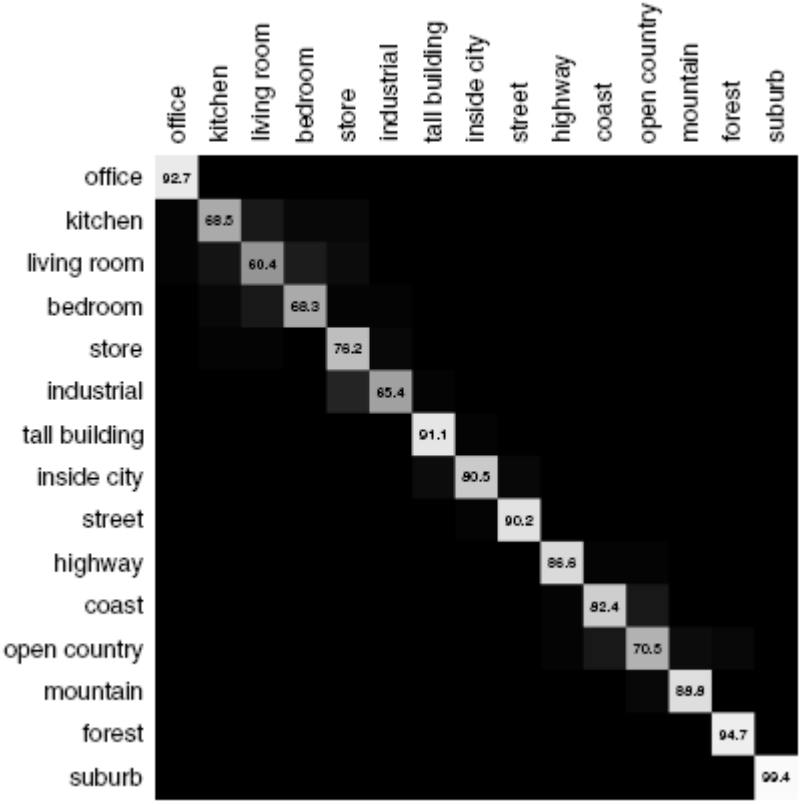
Sum over PMKs computed in *image coordinate* space, one per word.

[Lazebnik, Schmid & Ponce, CVPR 2006]

Spatial pyramid match

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



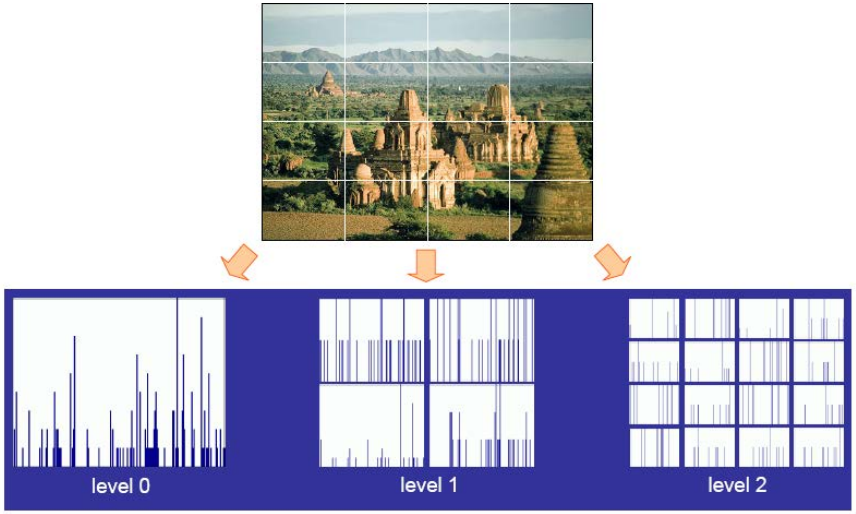


Confusion matrix

Kristen Grauman

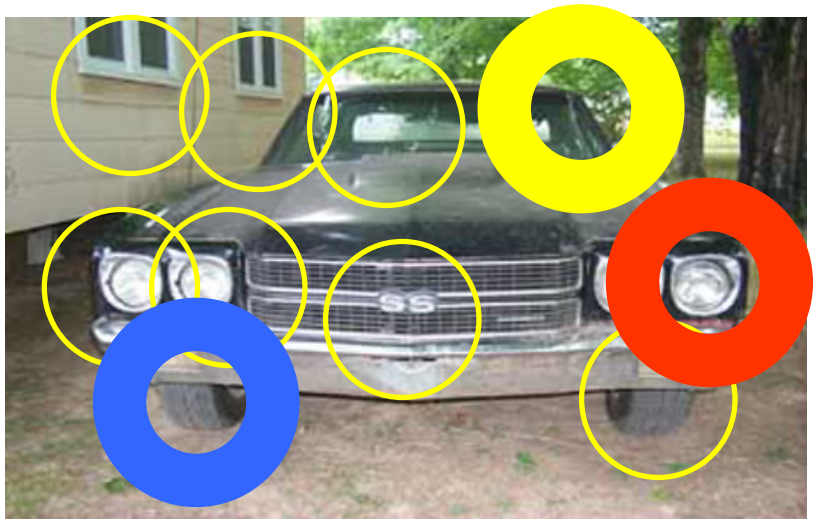
Spatial pyramid match

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

Part-based and local feature models for recognition

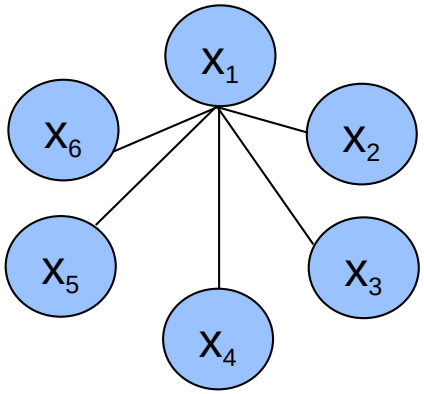


We'll look at three forms:

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

Shape representation in part-based models

“Star” shape model

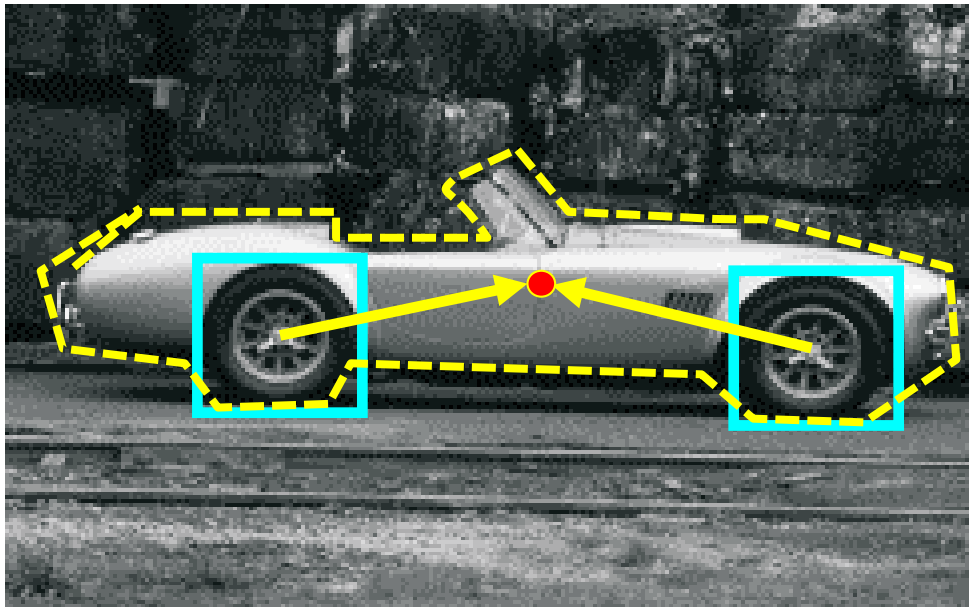


- **e.g. implicit shape model**
- **Parts mutually independent**

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

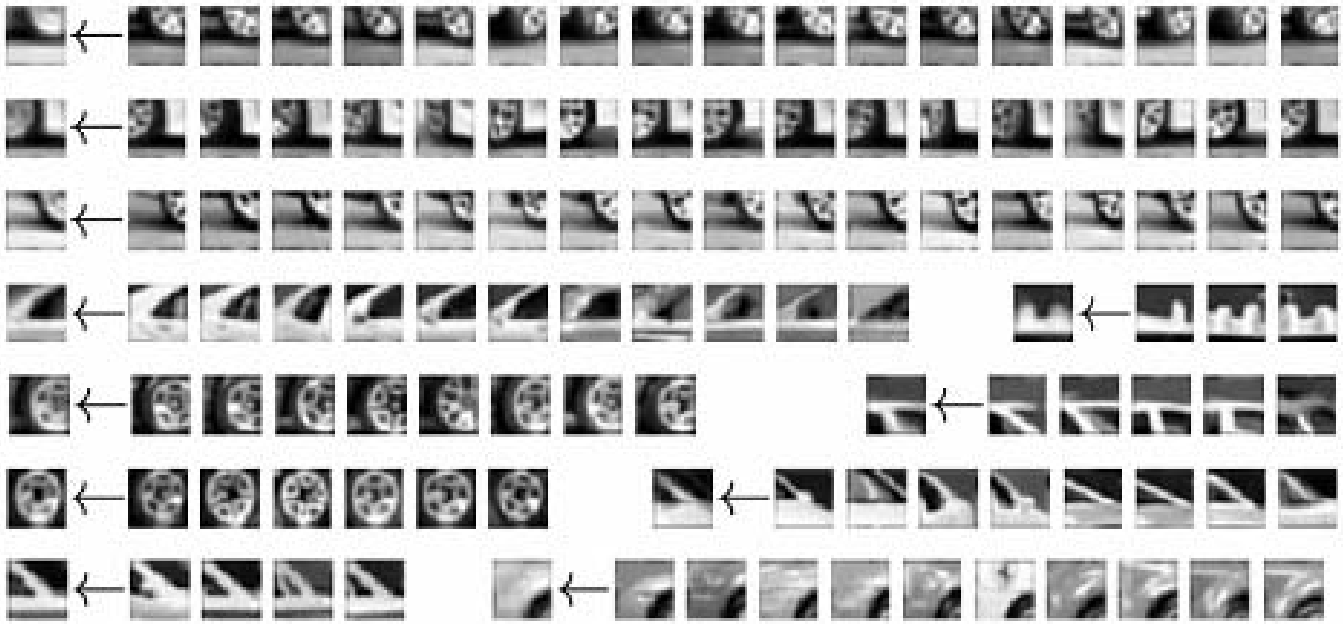
- Visual vocabulary is used to index votes for object position [a visual word = “part”]

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

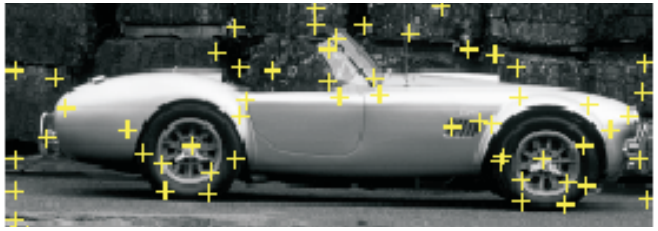
Implicit shape models: Training

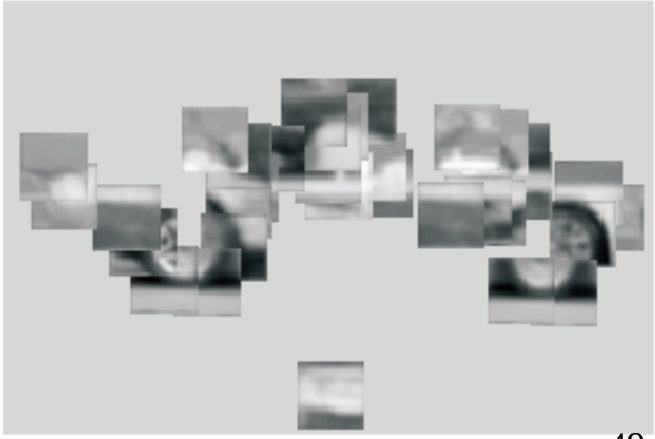
1. Build vocabulary of patches around extracted interest points using clustering



Implicit shape models: Training

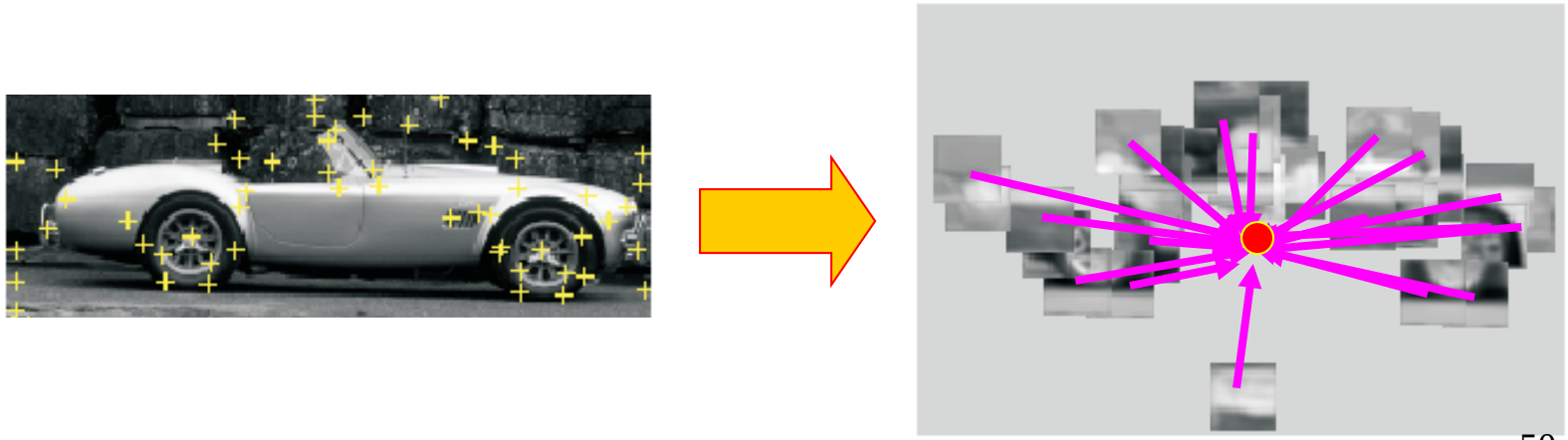
1. Build vocabulary of patches around extracted interest points using clustering
2. Map the patch around each interest point to closest word





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)

What is the dimension of the Hough space?

Implicit shape models: Testing

Original Image

Interest Points

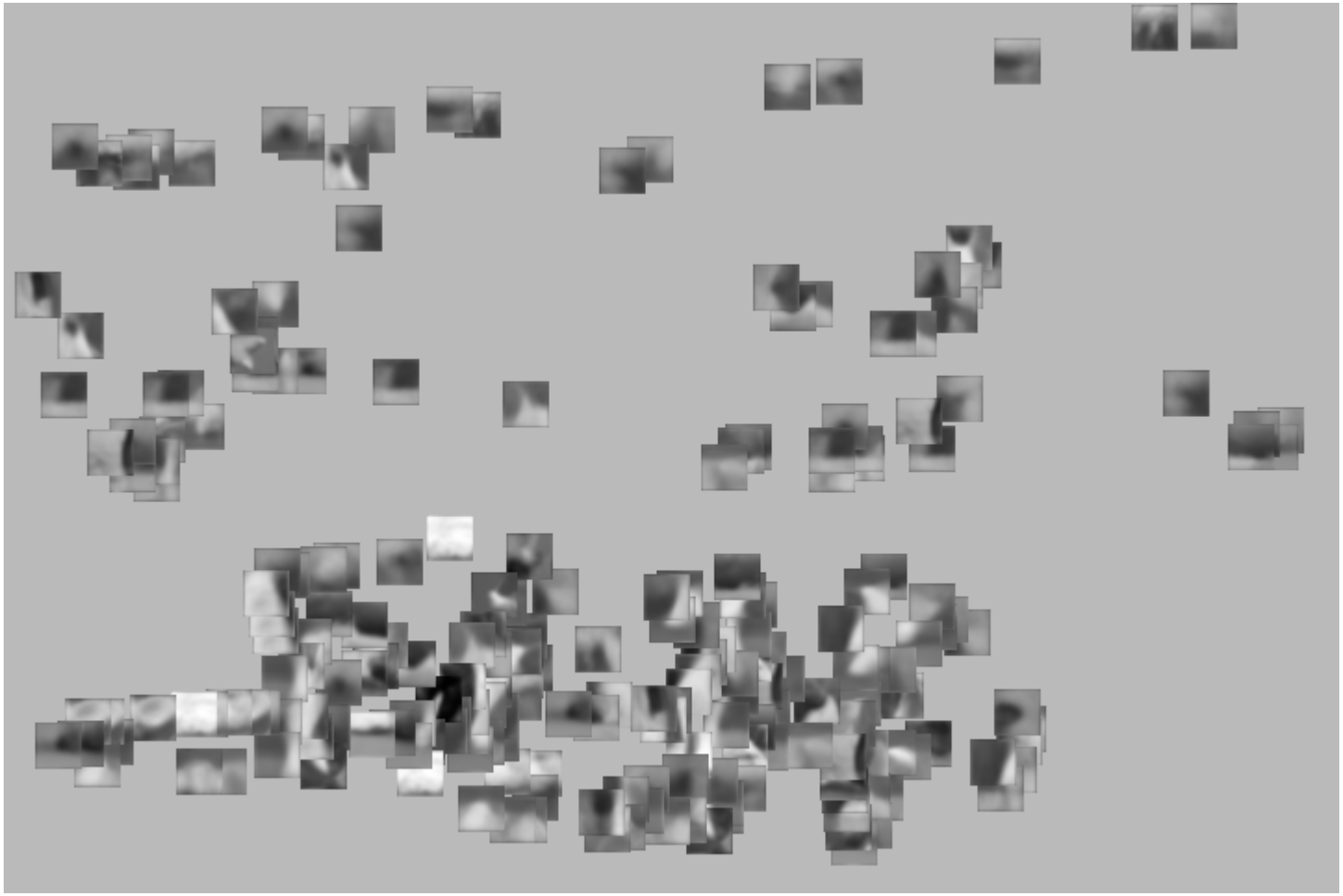
Example: Results on Cows

Original image

Example: Results on Cows

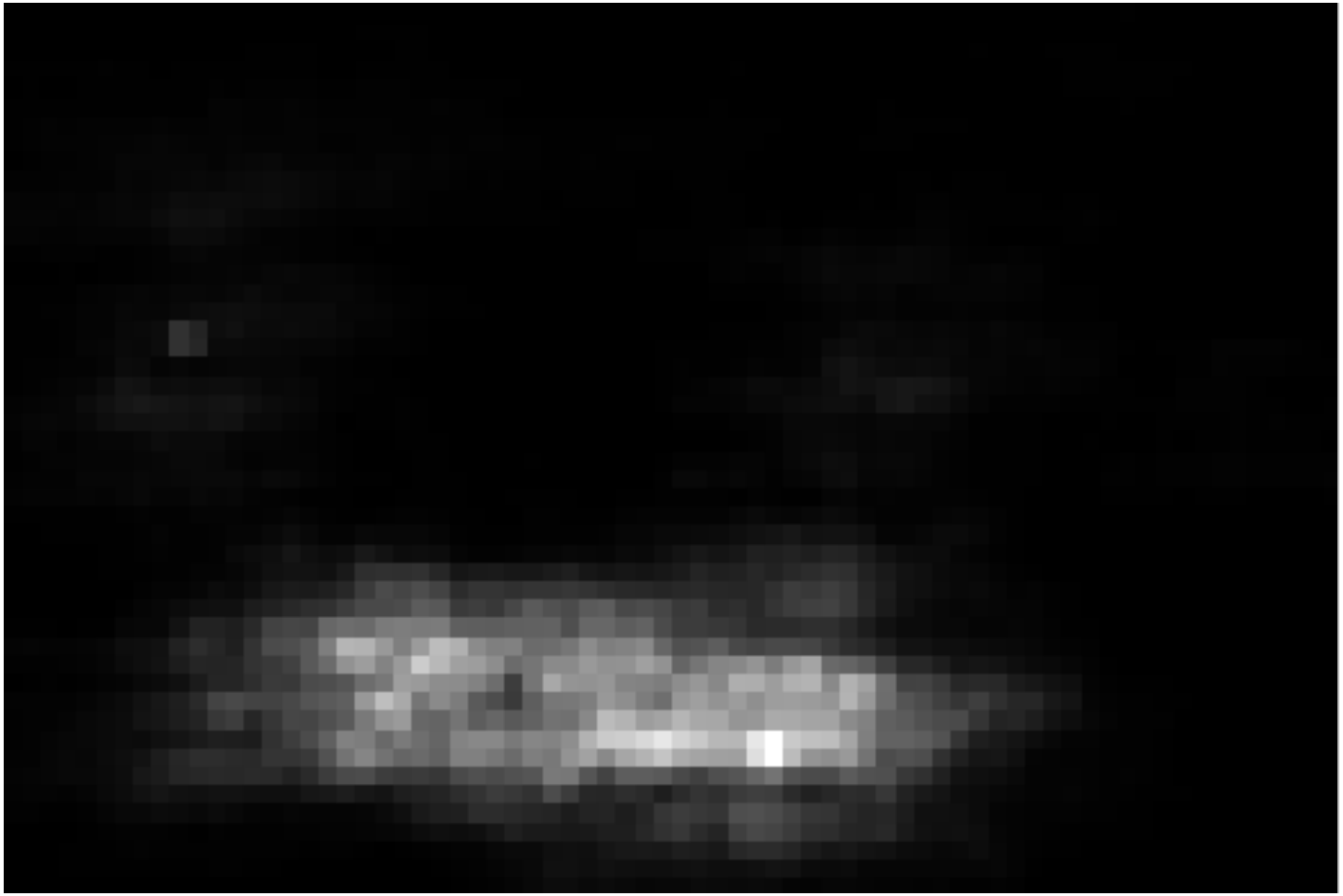
Interest points

Example: Results on Cows



Matched patches

Example: Results on Cows



Votes

Example: Results on Cows

1st hypothesis

Example: Results on Cows

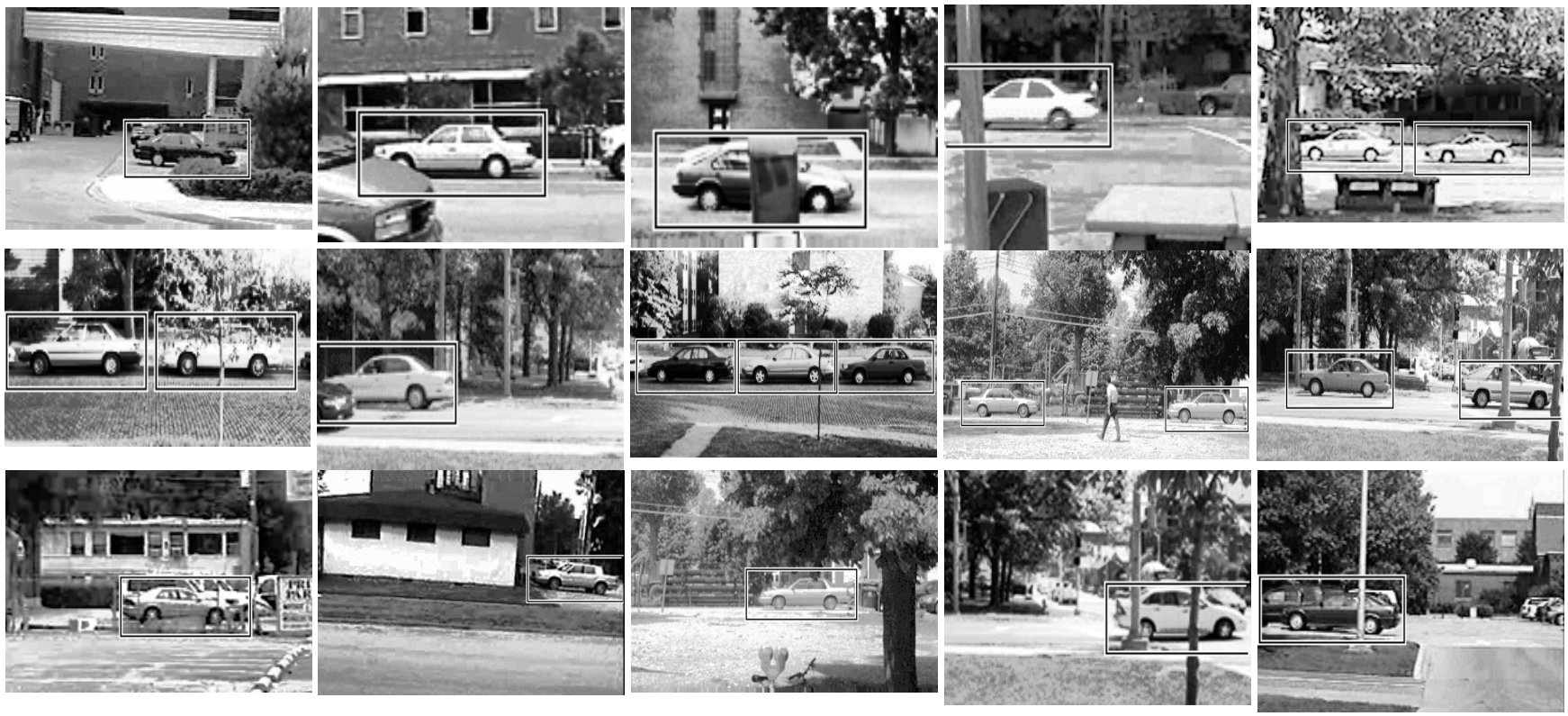
2nd hypothesis

Example: Results on Cows

3rd hypothesis

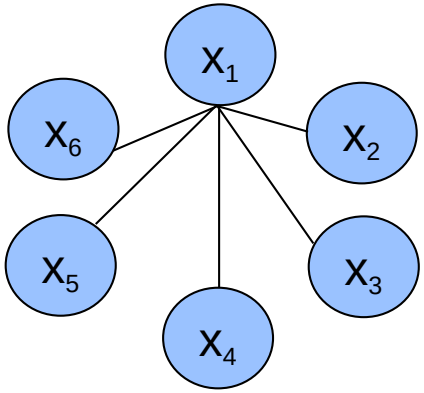
Detection Results

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



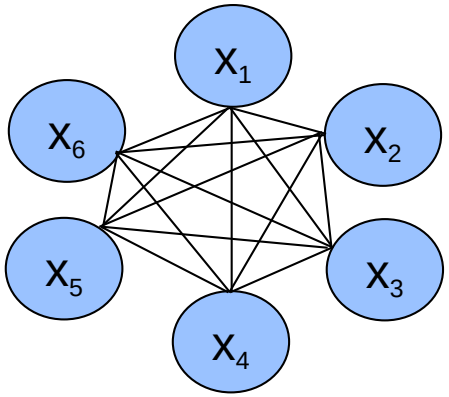
Shape representation in part-based models

"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

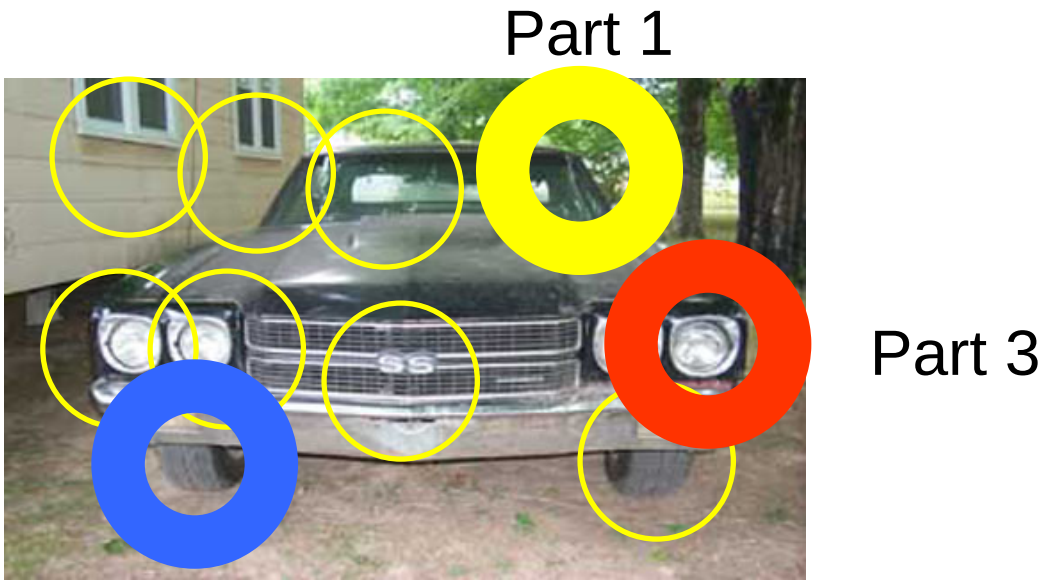
Part locations

Candidate parts

Source: Lana Lazebnik

Probabilistic constellation model

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



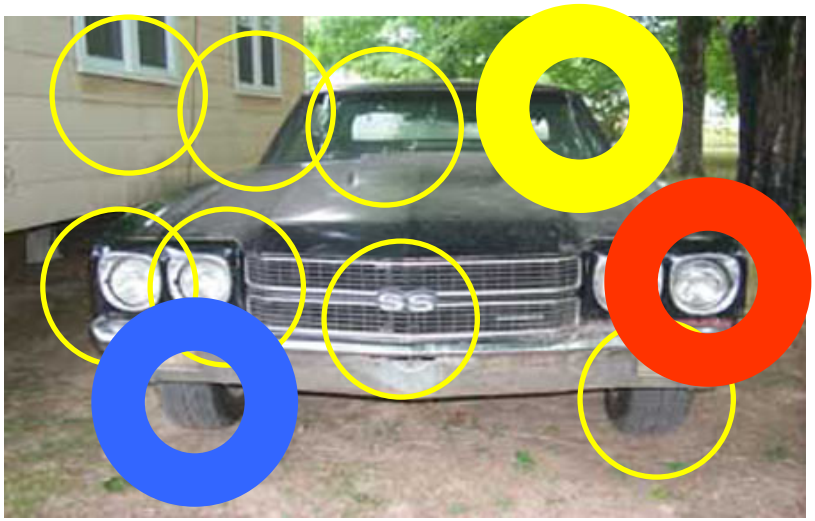
Source: Lana Lazebnik

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

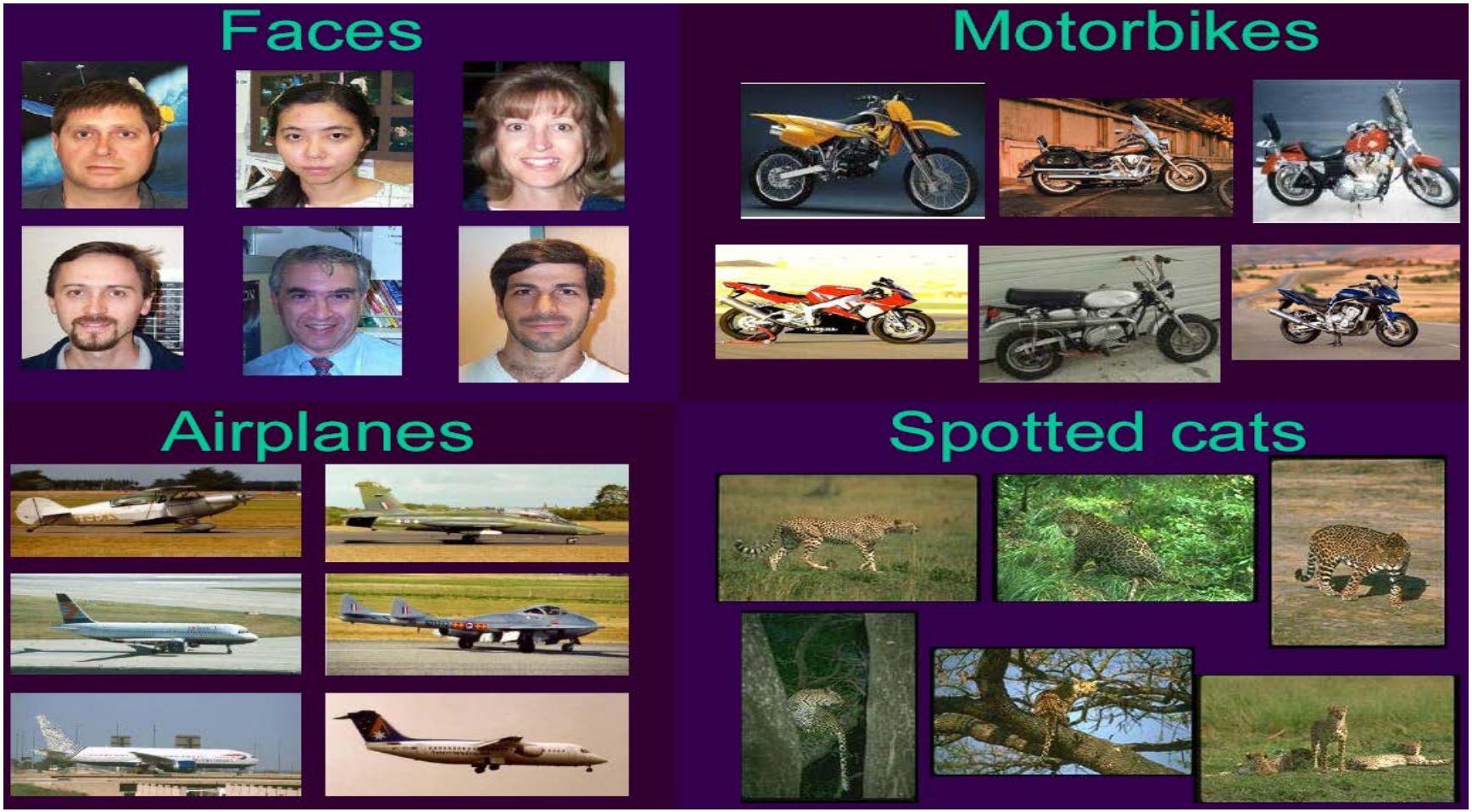


Part 3

Part 2

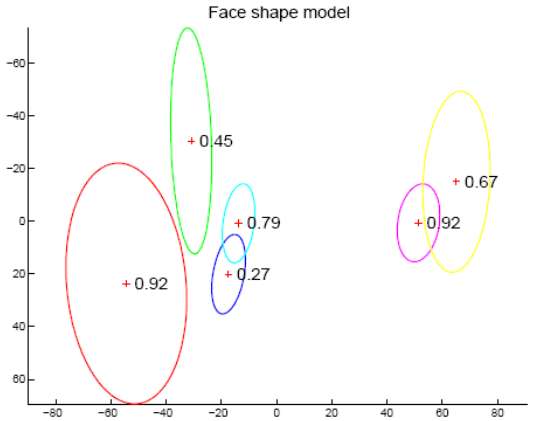
Source: Lana Lazebnik

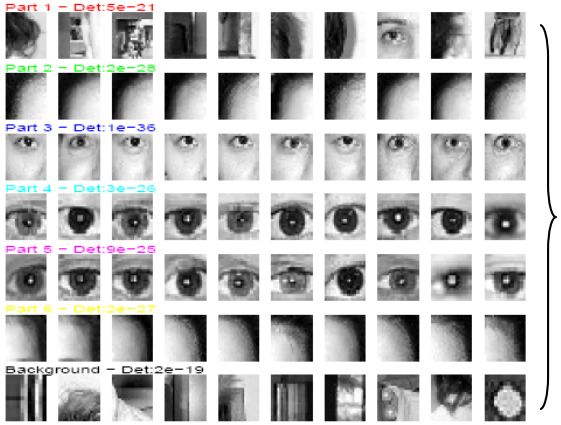
Example results from constellation model: data from four categories



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

Face model

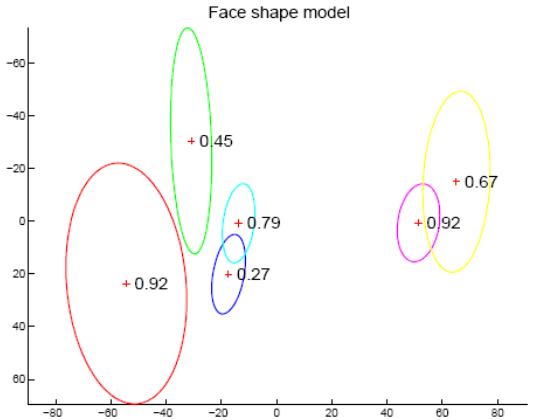


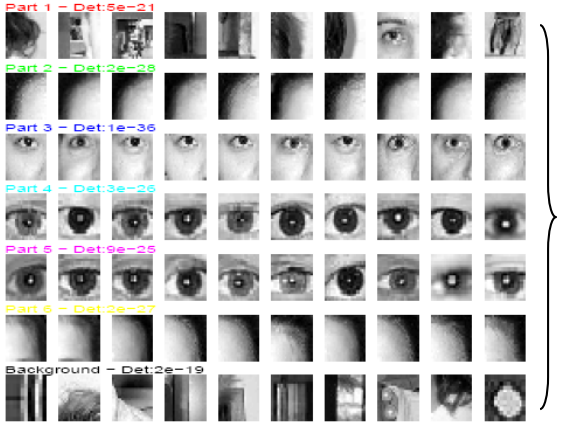


Appearance: 10 patches closest to mean for each part

Fergus et al. CVPR 2003

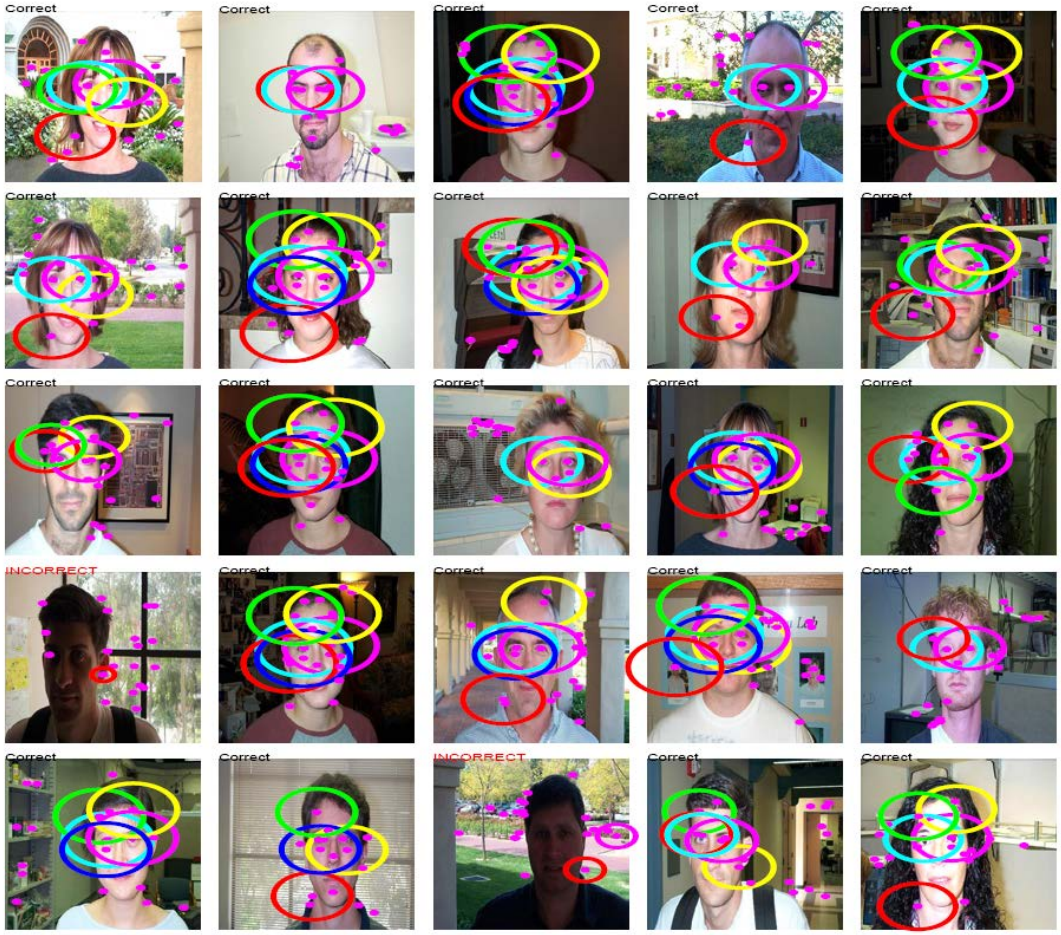
Face model





Appearance: 10 patches closest to mean for each part

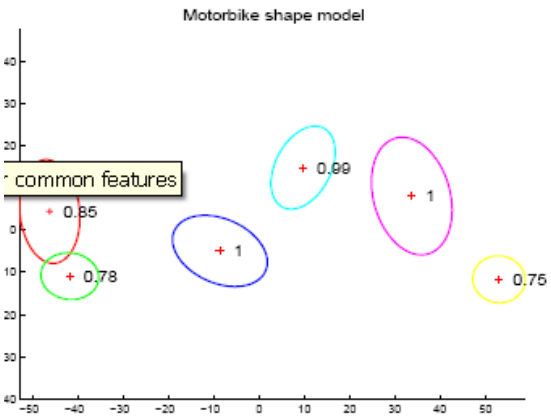
Recognition results

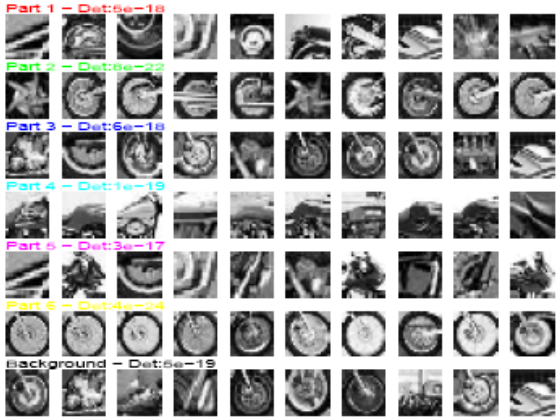


Test images: size of circles indicates score of hypothesis

Fergus et al. CVPR 2003

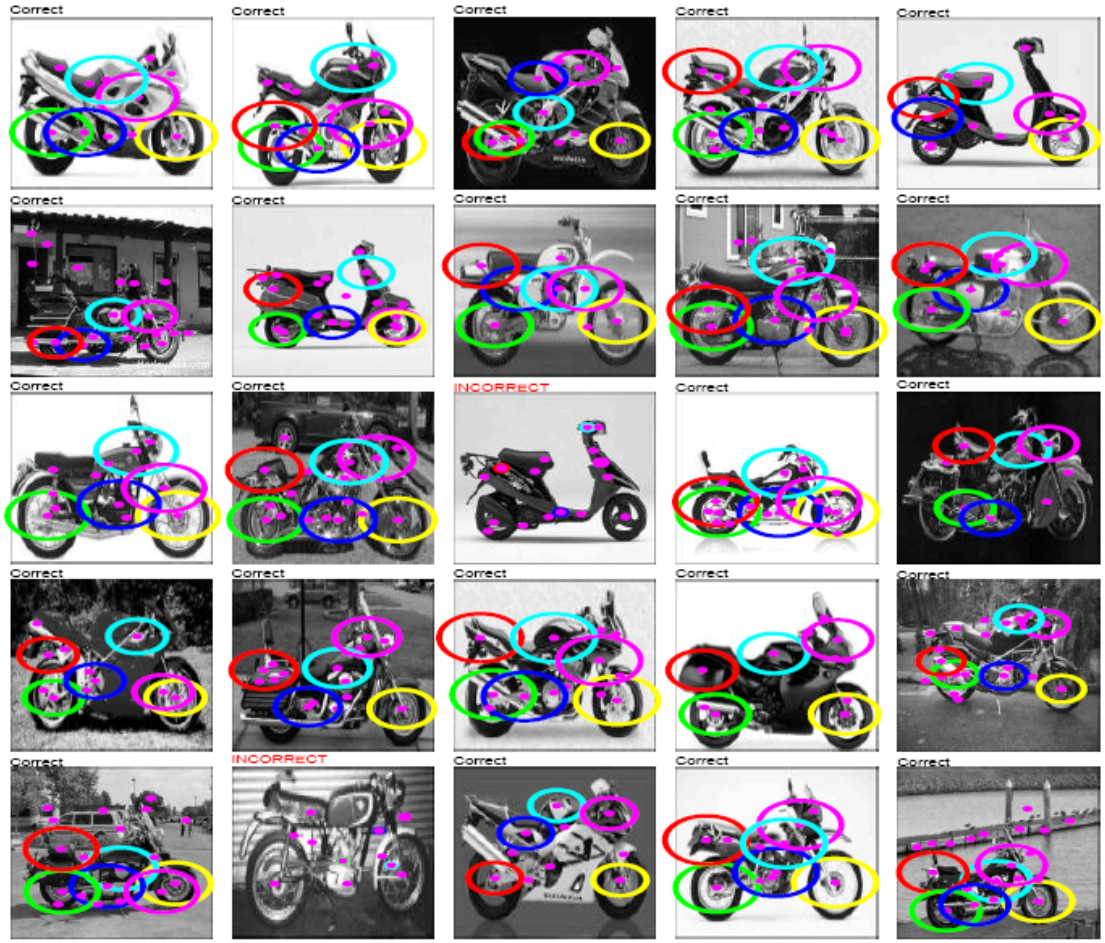
Motorbike model





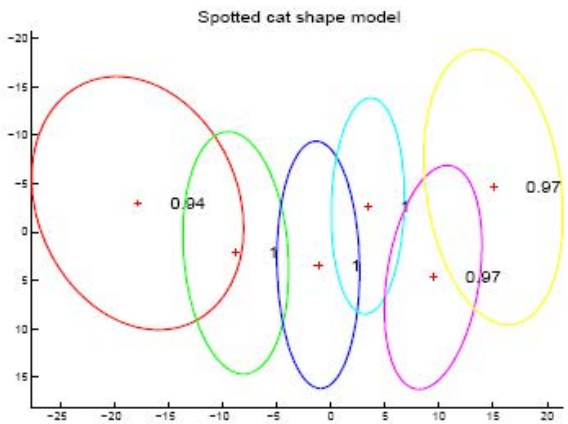
Appearance: 10 patches closest to mean for each part

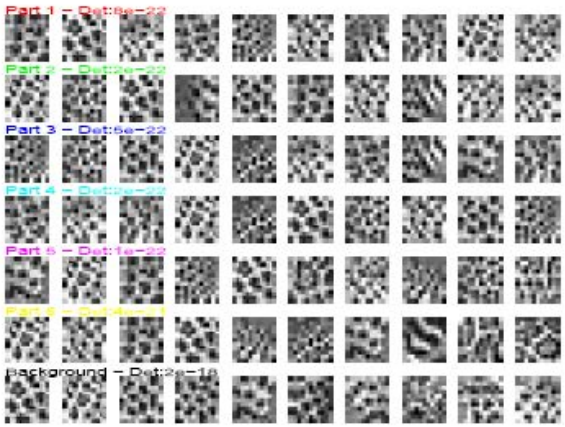
Recognition results



Fergus et al. CVPR 2003

Spotted cat model





Appearance: 10 patches closest to mean for each part

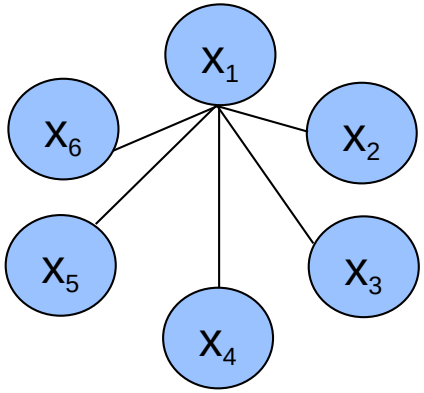
Recognition results

Kristen Grauman

Fergus et al. CVPR 2003

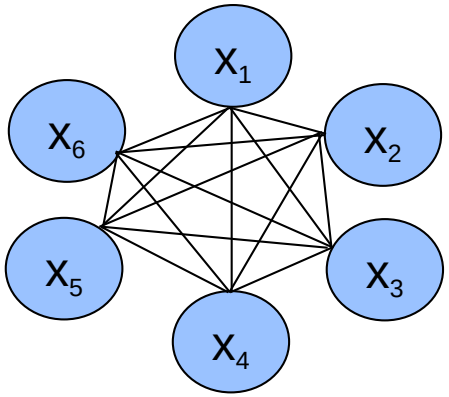
Shape representation in part-based models

"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

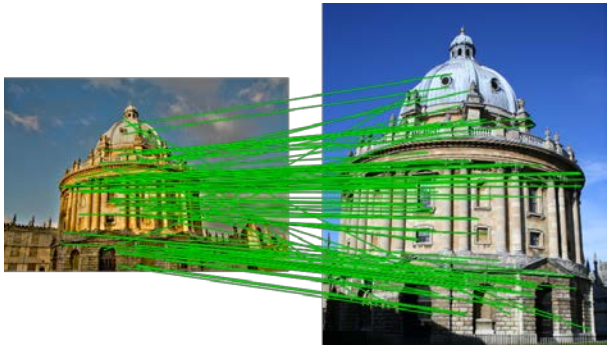
Slide credit: Rob Fergus

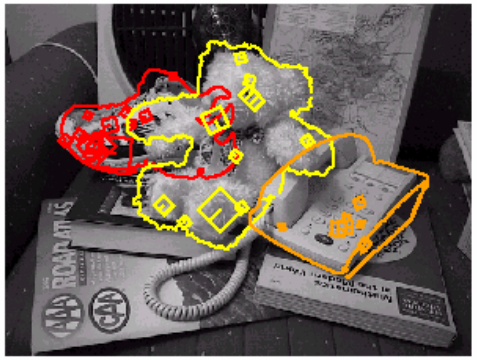
Summary:
part-based and local feature models for generic object recognition

- **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

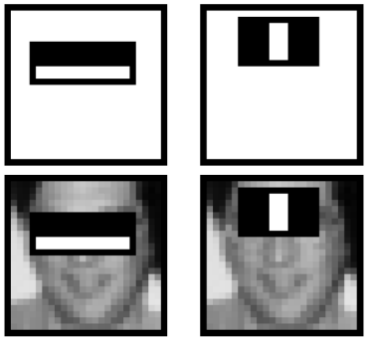
- **Part-based models** encode category's part appearance together with 2d layout and allow detection within cluttered image
  - **"implicit shape model"**: shape based on layout of all parts relative to a reference part; Generalized Hough for detection
  - **"constellation model"**: explicitly model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

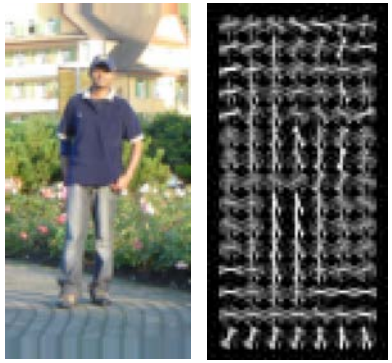
Recognition models



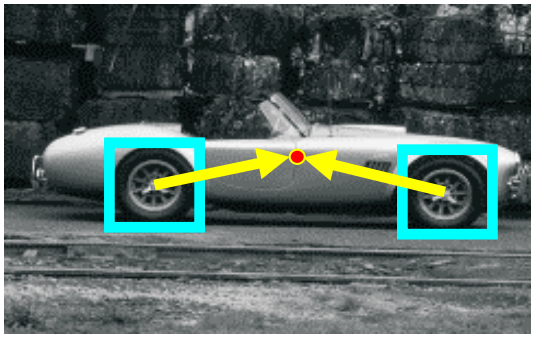


Instances: recognition by alignment





Categories: Holistic appearance models (and sliding window detection)



Categories: Local feature and part-based models

Kristen Grauman

Questions?

See you Tuesday!