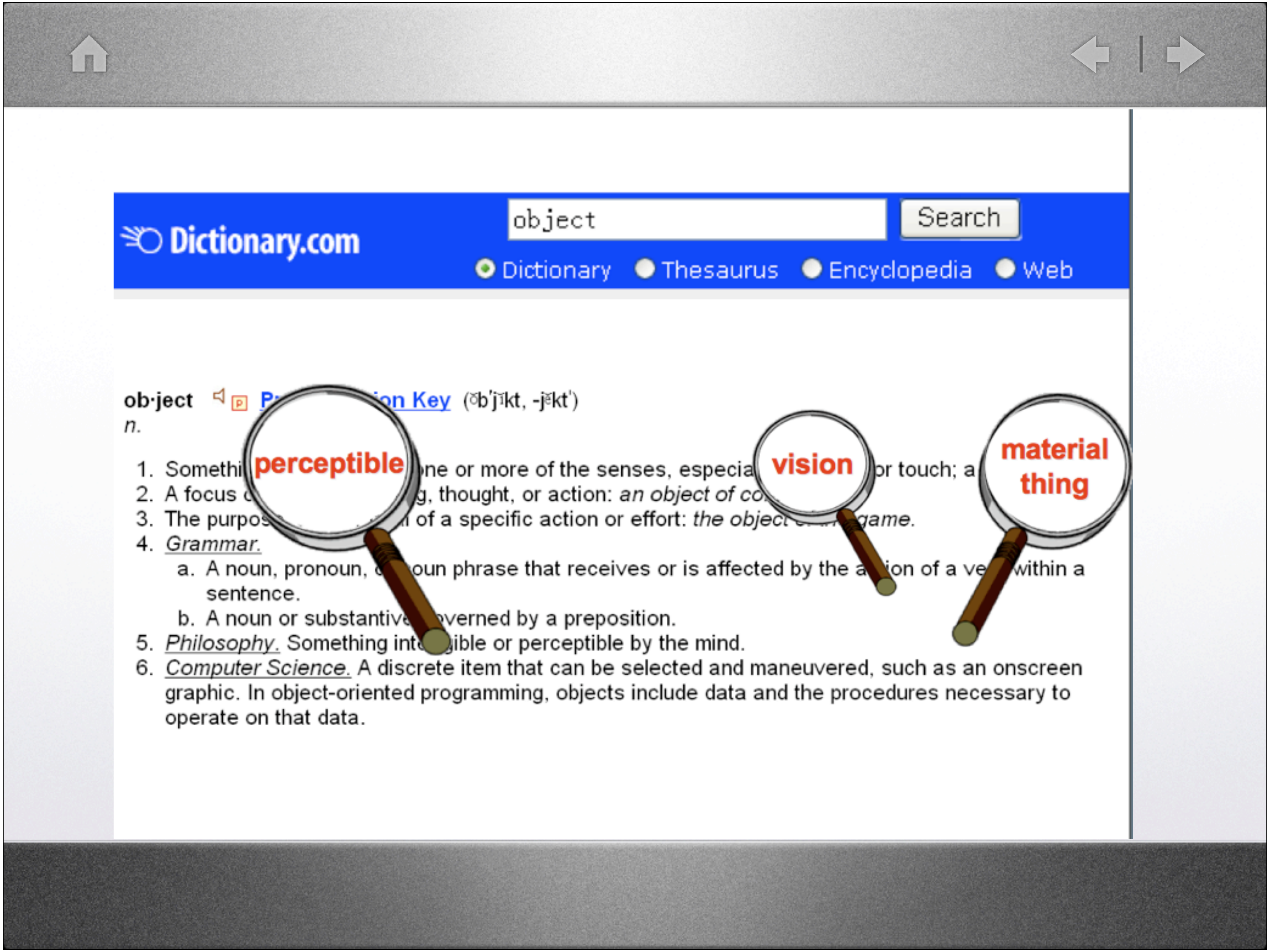
Cognitive Vision for Cognitive Systems

Barbara Caputo, Marco Fornoni
Idiap Research Institute
http://www.idiap.ch/~bcaputo
http://www.idiap.ch/~mfornoni
bcaputo@idiap.ch
mfornoni@idiap.ch

Object Recognition --the computer vision way

(slides credit: Fei Fei Li, Rob Fergus and Antonio Torralba, 2007)

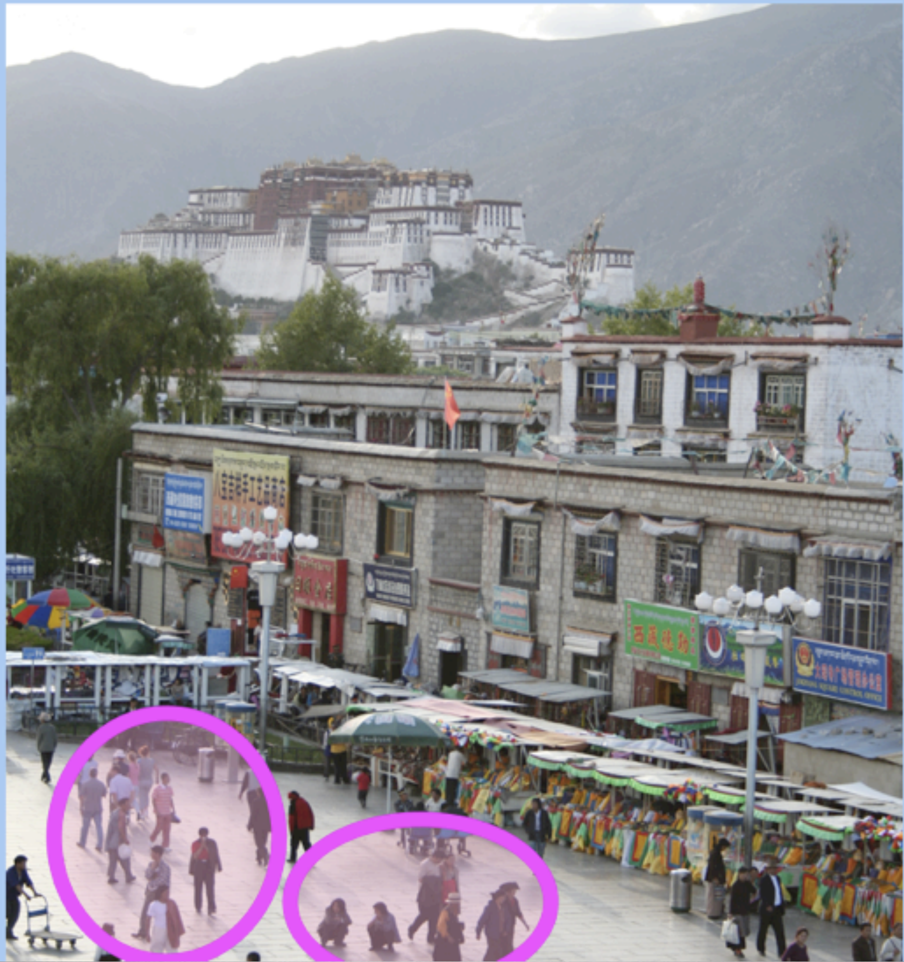

object
Search
Dictionary.com
Dictionary
Thesaurus
Encyclopedia
Web
ob·ject
n.
1. Somethi
ne or more of the senses, especia
or touch; a
2. A focus o
g, thought, or action: an object of co
3. The purpo
of a specific action or effort: the object
game.
4. Grammar.
a. A noun, pronoun,
oun phrase that receives or is affected by the a
ion of a ve
within a sentence.
b. A noun or substantive
verned by a preposition.
5. Philosophy. Something int
ible or perceptible by the mind.
6. Computer Science. A discrete item that can be selected and maneuvered, such as an onscreen graphic. In object-oriented programming, objects include data and the procedures necessary to operate on that data.
perceptible
vision
material thing

How many object categories are there?

So what does object recognition involve?

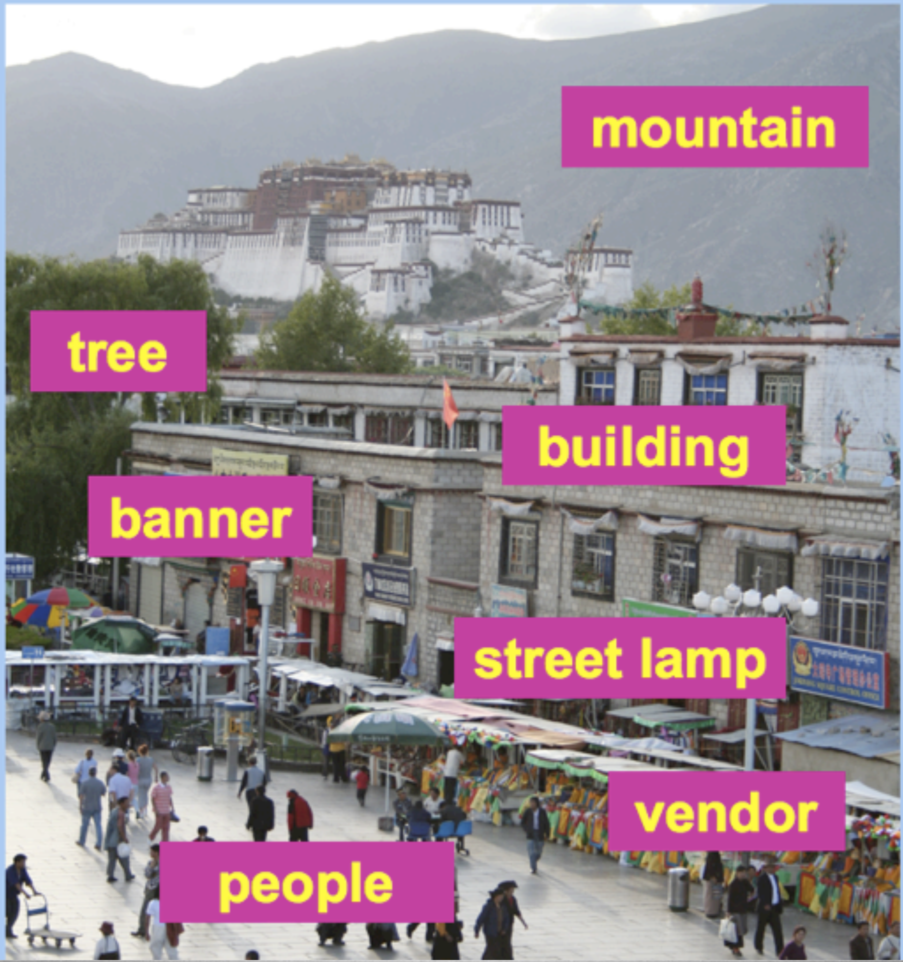
Verification: is that a lamp?

Detection: are there people?



Identification: is that Potala Palace?

Object categorization



Challenges 1: view point variation

Michelangelo 1475-1564

Challenges 2: illumination

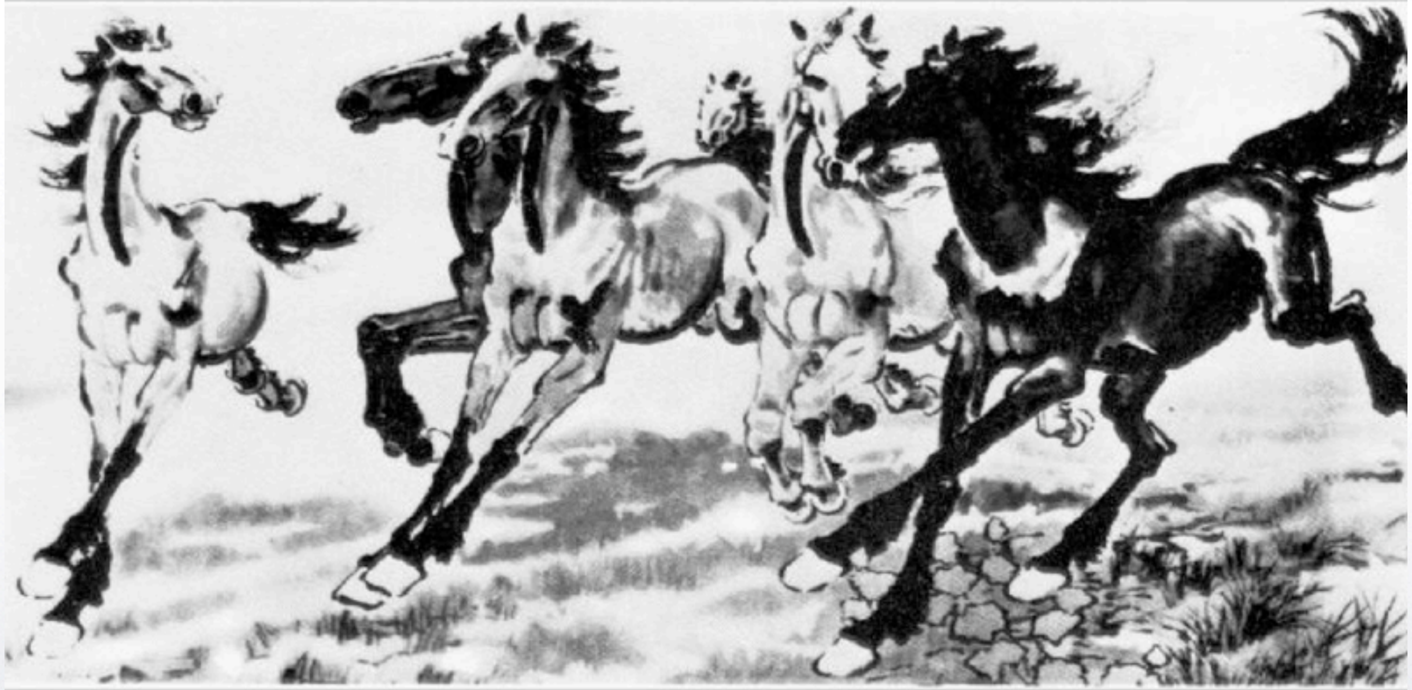
slide credit: S. Ullman

Challenges 3: occlusion

Magritte, 1957

Challenges 4: scale

Challenges 5: deformation



Xu, Beihong 1943

Challenges 6: background clutter

Klimt, 1913

Challenges 7: intra-class variation

History: single object recognition

History: single object recognition

- Lowe, et al. 1999, 2003
- Mahamud and Herbert, 2000
- Ferrari, Tuytelaars, and Van Gool, 2004
- Rothganger, Lazebnik, and Ponce, 2004
- Moreels and Perona, 2005
- …

History: early object categorization

- Turk and Pentland, 1991
- Belhumeur, Hespanha, & Kriegman, 1997
- Schneiderman & Kanade 2004
- Viola and Jones, 2000

- Amit and Geman, 1999
- LeCun et al. 1998
- Belongie and Malik, 2002

- Schneiderman & Kanade, 2004
- Argawal and Roth, 2002
- Poggio et al. 1993

Object categorization: the statistical viewpoint

$$p(zebra \mid image)$$

vs.

$$p(no\ zebra \mid image)$$

- Bayes rule:

$$\underbrace{\frac{p(zebra \mid image)}{p(no\ zebra \mid image)}}_{\text{posterior ratio}} = \underbrace{\frac{p(image \mid zebra)}{p(image \mid no\ zebra)}}_{\text{likelihood ratio}} \cdot \underbrace{\frac{p(zebra)}{p(no\ zebra)}}_{\text{prior ratio}}$$

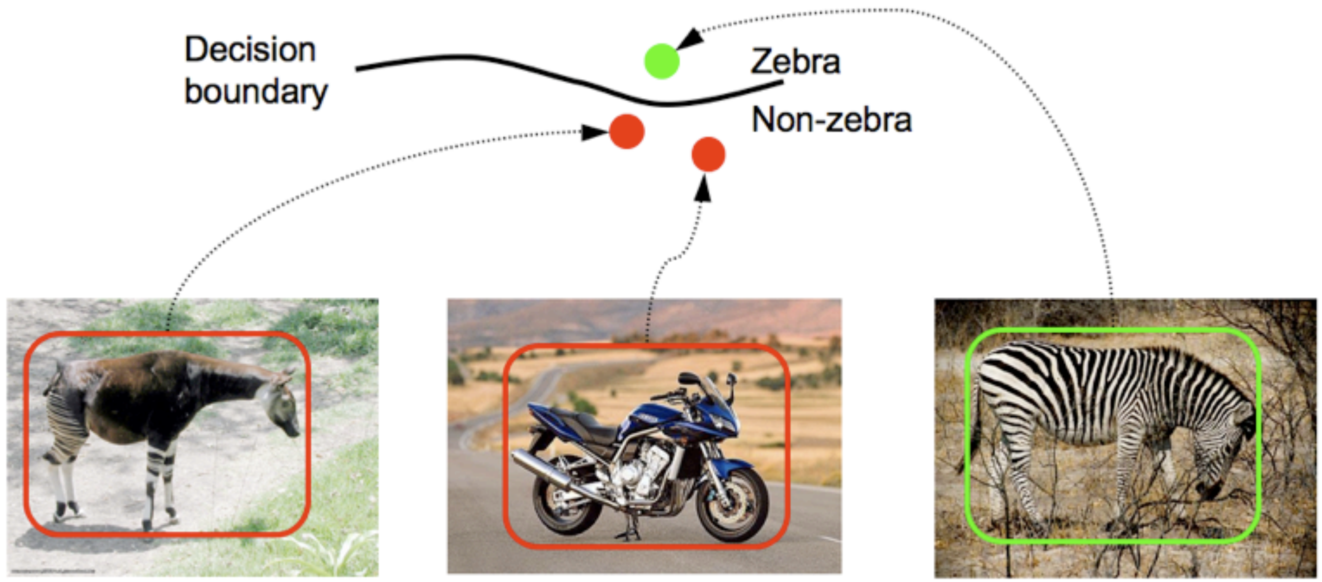
Object categorization: the statistical viewpoint

$$\underbrace{\frac{p(zebra \mid image)}{p(no\ zebra \mid image)}}_{\text{posterior ratio}} = \underbrace{\frac{p(image \mid zebra)}{p(image \mid no\ zebra)}}_{\text{likelihood ratio}} \cdot \underbrace{\frac{p(zebra)}{p(no\ zebra)}}_{\text{prior ratio}}$$

- **Discriminative methods model posterior**
- **Generative methods model likelihood and prior**

Discriminative

- Direct modeling of $$\frac{p(zebra \mid image)}{p(no\ zebra \mid image)}$$



Generative

- Model $p(image \mid zebra)$ and $p(image \mid no\ zebra)$

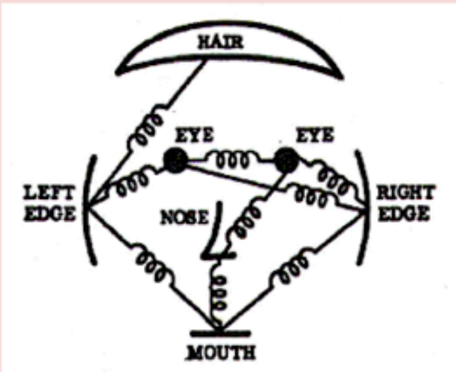
	$p(image \mid zebra)$	$p(image \mid no\ zebra)$
	Low	Middle
	High	Middle→Low

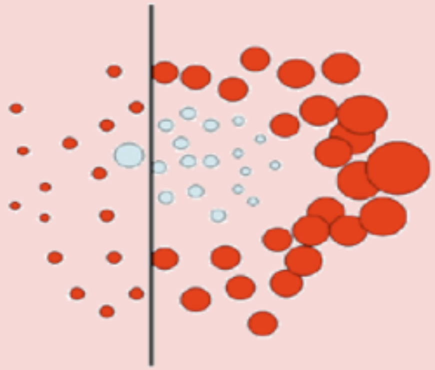
Three main issues

- Representation
  - How to represent an object category
- Learning
  - How to form the classifier, given training data
- Recognition
  - How the classifier is to be used on novel data

Representation

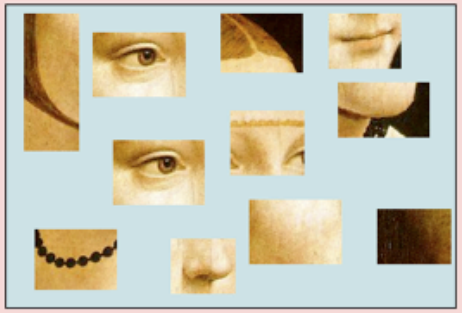
– Generative / discriminative / hybrid

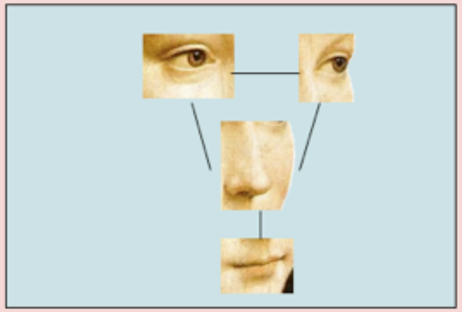




Representation

- Generative / discriminative / hybrid
- Appearance only or location and appearance



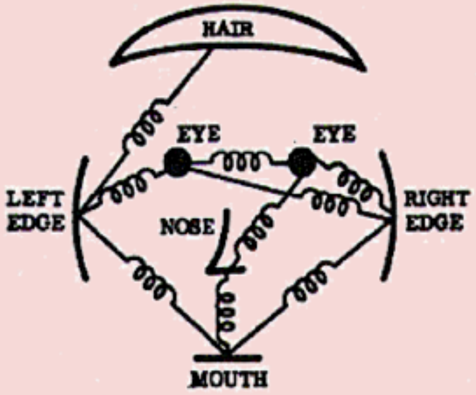


Representation

- Generative / discriminative / hybrid
- Appearance only or location and appearance
- Invariances
  - View point
  - Illumination
  - Occlusion
  - Scale
  - Deformation
  - Clutter
  - etc.

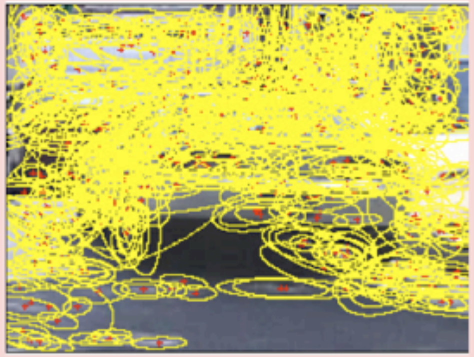
Representation

– Generative / discriminative / hybrid
– Appearance only or location and appearance
– invariances
– Part-based or global w/sub-window



Representation

- Generative / discriminative / hybrid
- Appearance only or location and appearance
- invariances
- Parts or global w/sub-window
- Use set of features or each pixel in image

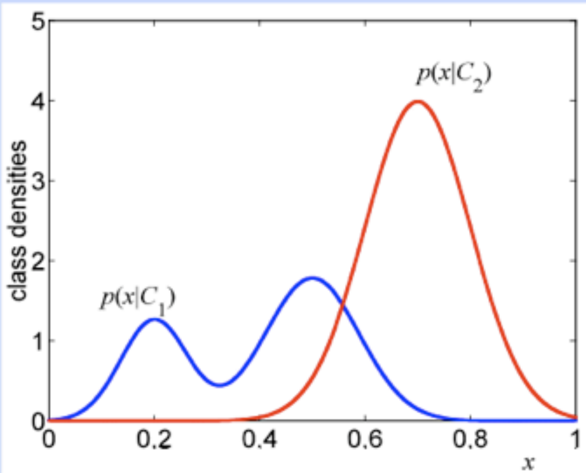


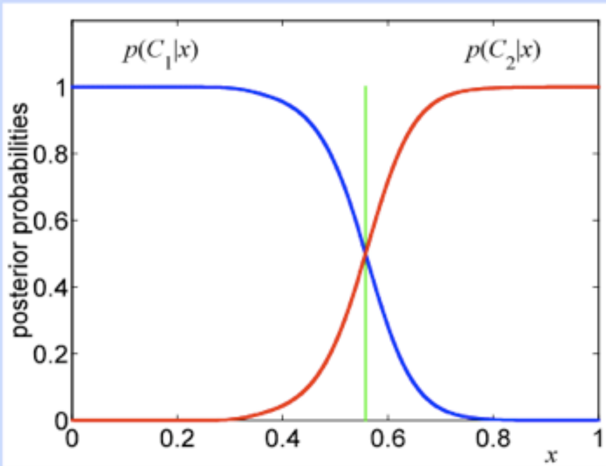
Learning

– Unclear how to model categories, so we learn what distinguishes them rather than manually specify the difference -- hence current interest in machine learning

Learning

- Unclear how to model categories, so we learn what distinguishes them rather than manually specify the difference -- hence current interest in machine learning)
- Methods of training: generative vs. discriminative

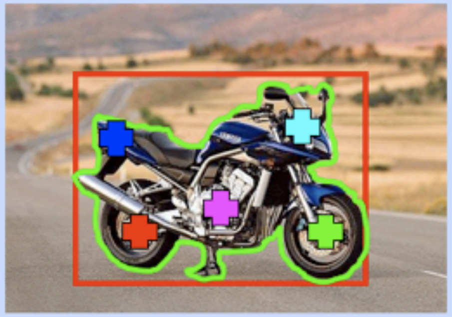




Learning

- Unclear how to model categories, so we learn what distinguishes them rather than manually specify the difference -- hence current interest in machine learning)
- What are you maximizing? Likelihood (Gen.) or performances on train/validation set (Disc.)
- Level of supervision
  - Manual segmentation; bounding box; image labels; noisy labels

Contains a motorbike



15 min break!

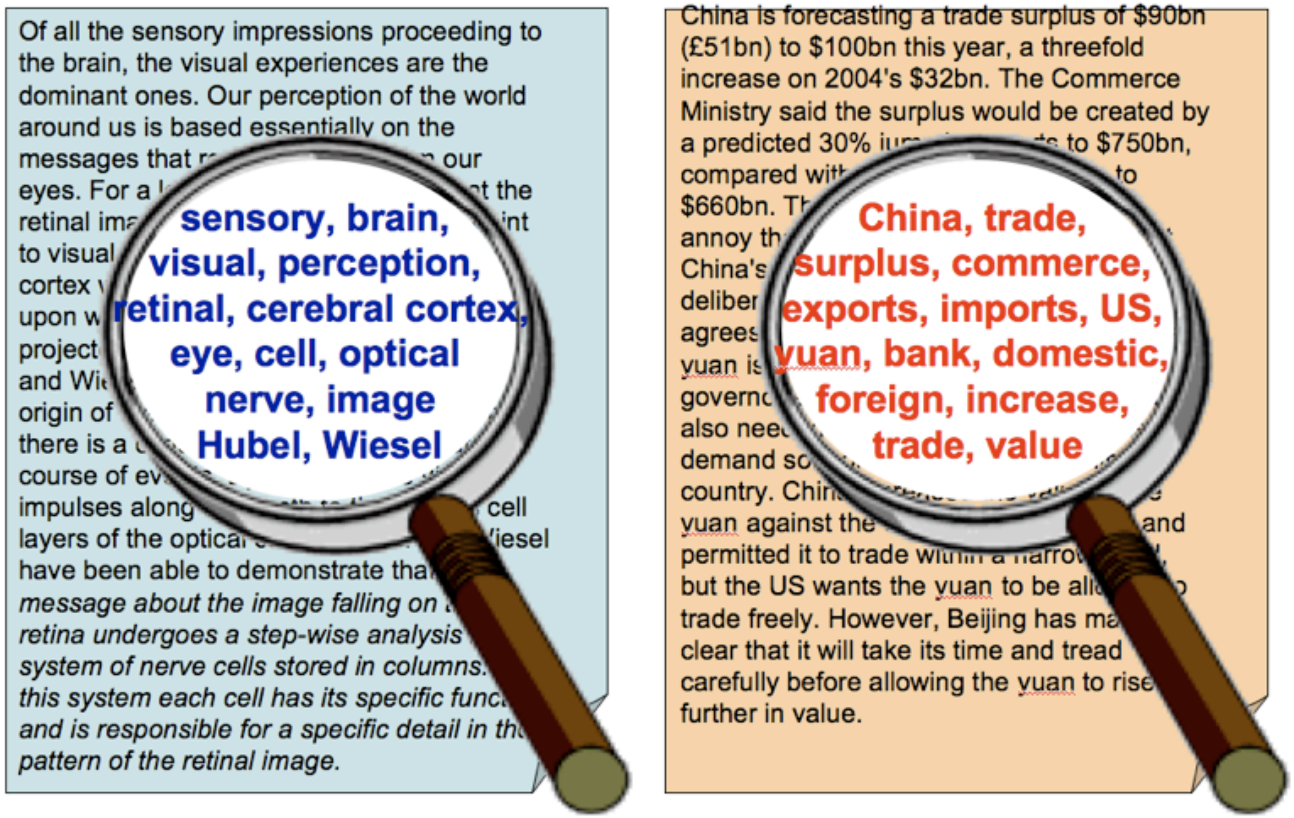
Bag-of-Words models

Related works

- Early “bag of words” models: mostly texture recognition
  - Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003;
- Hierarchical Bayesian models for documents (pLSA, LDA, etc.)
  - Hoffman 1999; Blei, Ng & Jordan, 2004; Teh, Jordan, Beal & Blei, 2004
- Object categorization
  - Csurka, Bray, Dance & Fan, 2004; Sivic, Russell, Efros, Freeman & Zisserman, 2005; Sudderth, Torralba, Freeman & Willsky, 2005;
- Natural scene categorization
  - Vogel & Schiele, 2004; Fei-Fei & Perona, 2005; Bosch, Zisserman & Munoz, 2006

Object → **Bag of ‘words’**

Analogy to documents

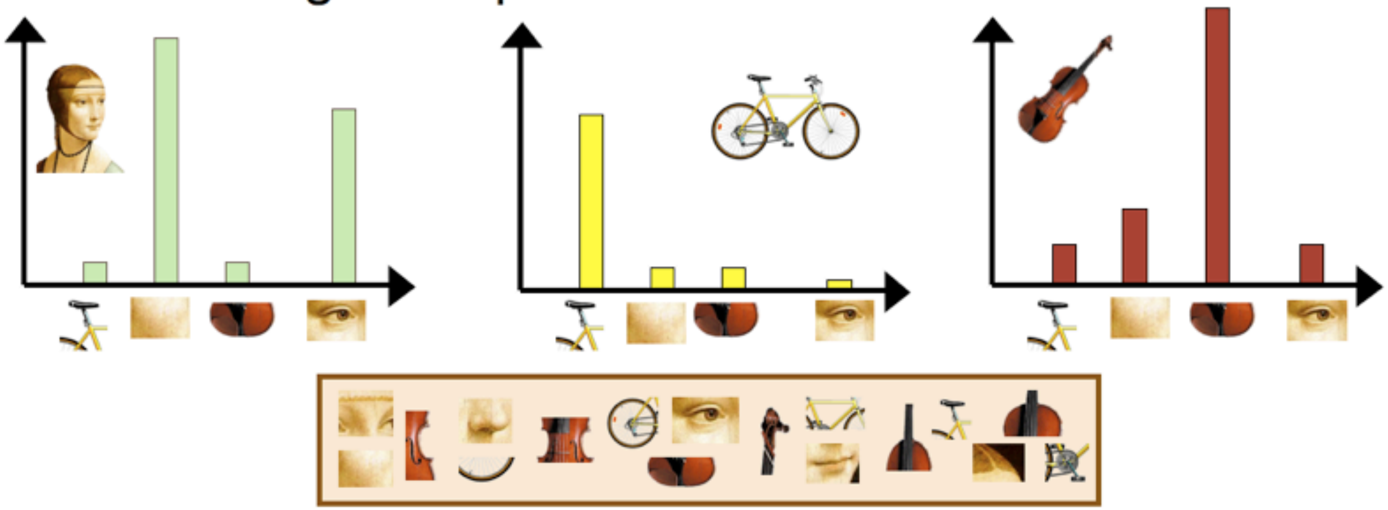


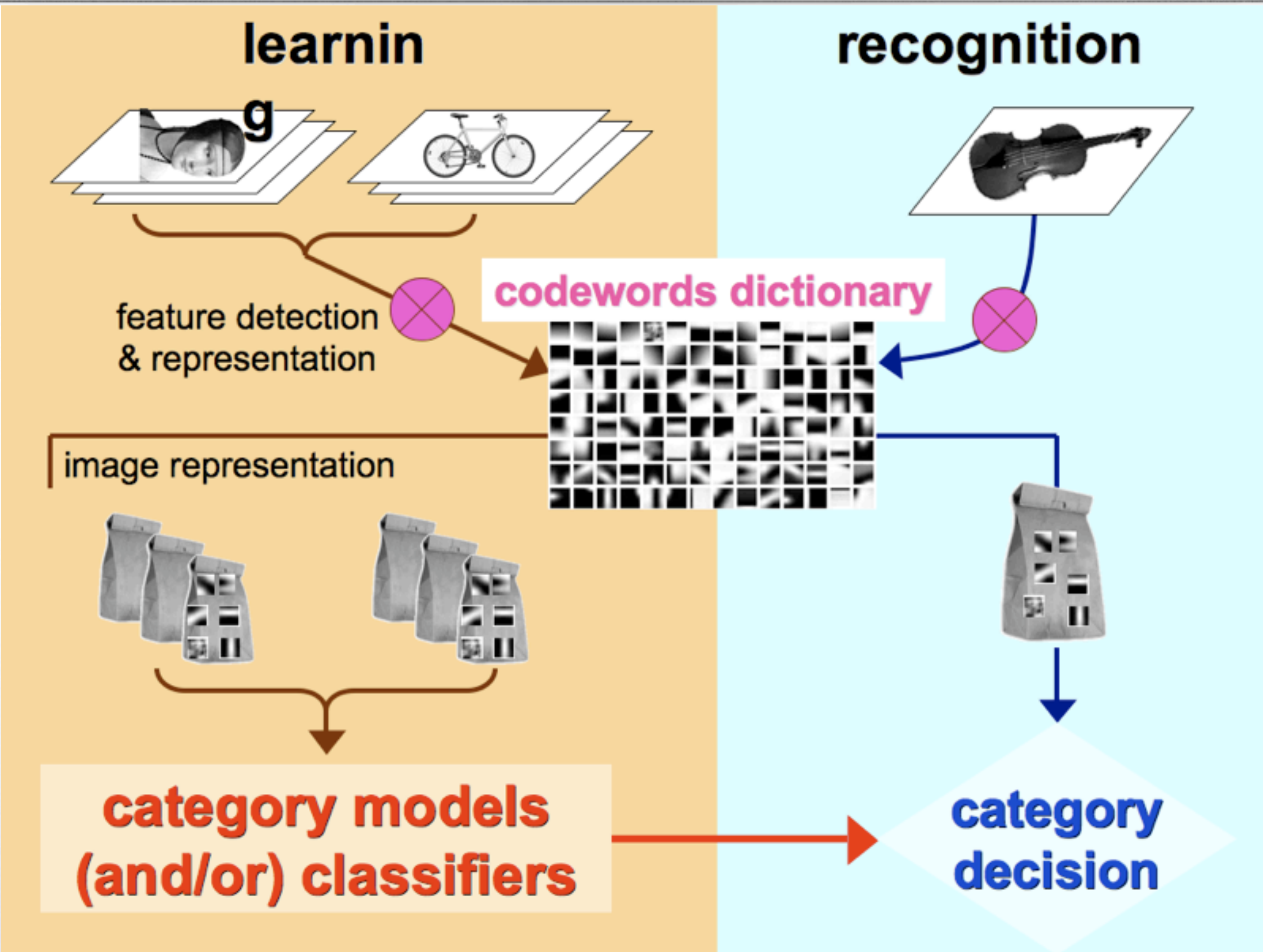
A clarification: definition of “BoW”

- Looser definition
  - Independent features

A clarification: definition of “BoW”

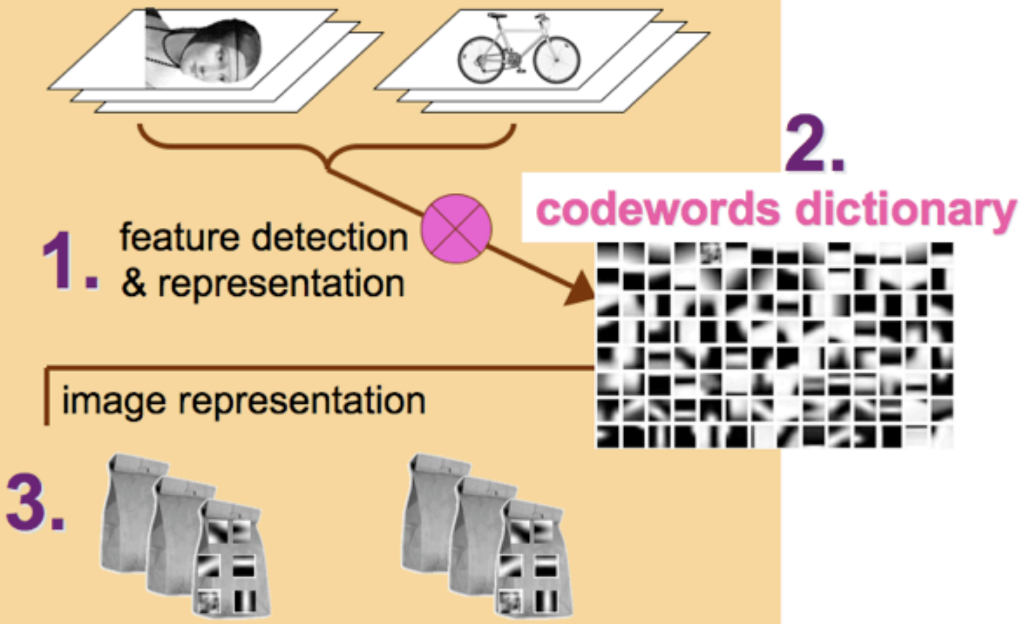
- Looser definition
  - Independent features
- Stricter definition
  - Independent features
  - histogram representation





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

Representation



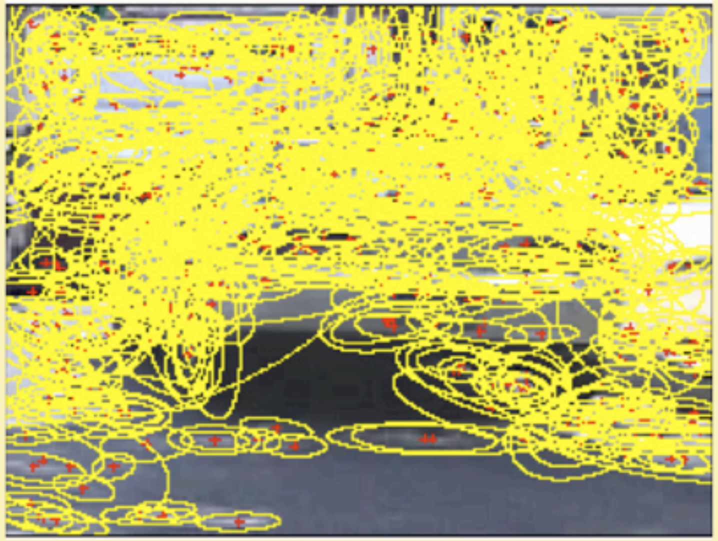
1.Feature detection and representation

1.Feature detection and representation

- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005

1.Feature detection and representation

- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005
- Interest point detector
  - Csurka, et al. 2004
  - Fei-Fei & Perona, 2005
  - Sivic, et al. 2005



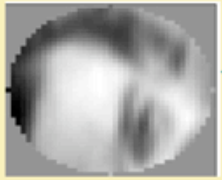
1.Feature detection and representation

- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005
- Interest point detector
  - Csurka, Bray, Dance & Fan, 2004
  - Fei-Fei & Perona, 2005
  - Sivic, Russell, Efros, Freeman & Zisserman, 2005
- Other methods
  - Random sampling (Vidal-Naquet & Ullman, 2002)
  - Segmentation based patches (Barnard, Duygulu, Forsyth, de Freitas, Blei, Jordan, 2003)

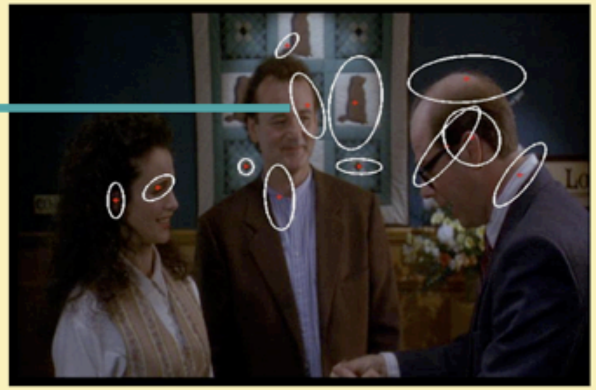
1.Feature detection and representation

Compute SIFT descriptor

[Lowe'99]



Normalize patch



Detect patches

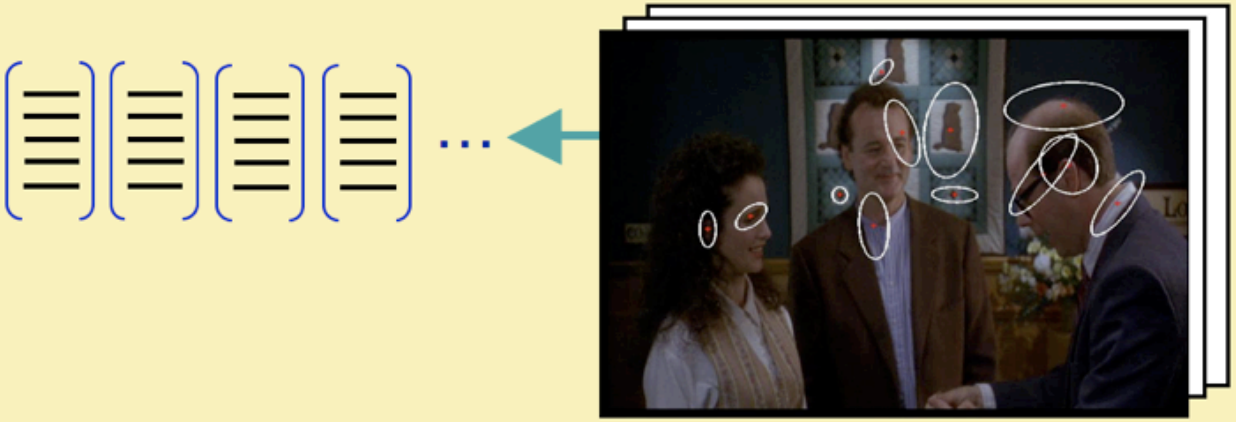
[Mikojaczyk and Schmid '02]

[Mata, Chum, Urban & Pajdla, '02]

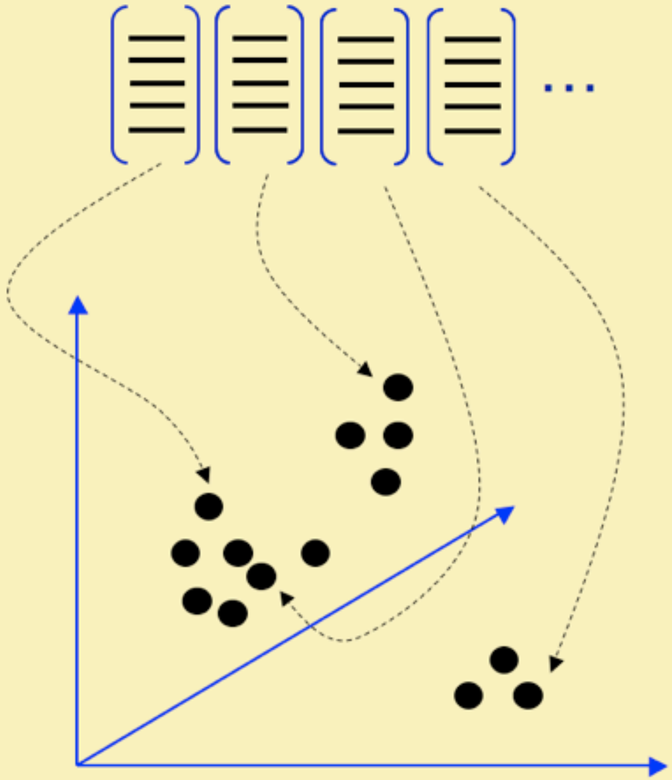
[Sivic & Zisserman, '03]

Slide credit: Josef Sivic

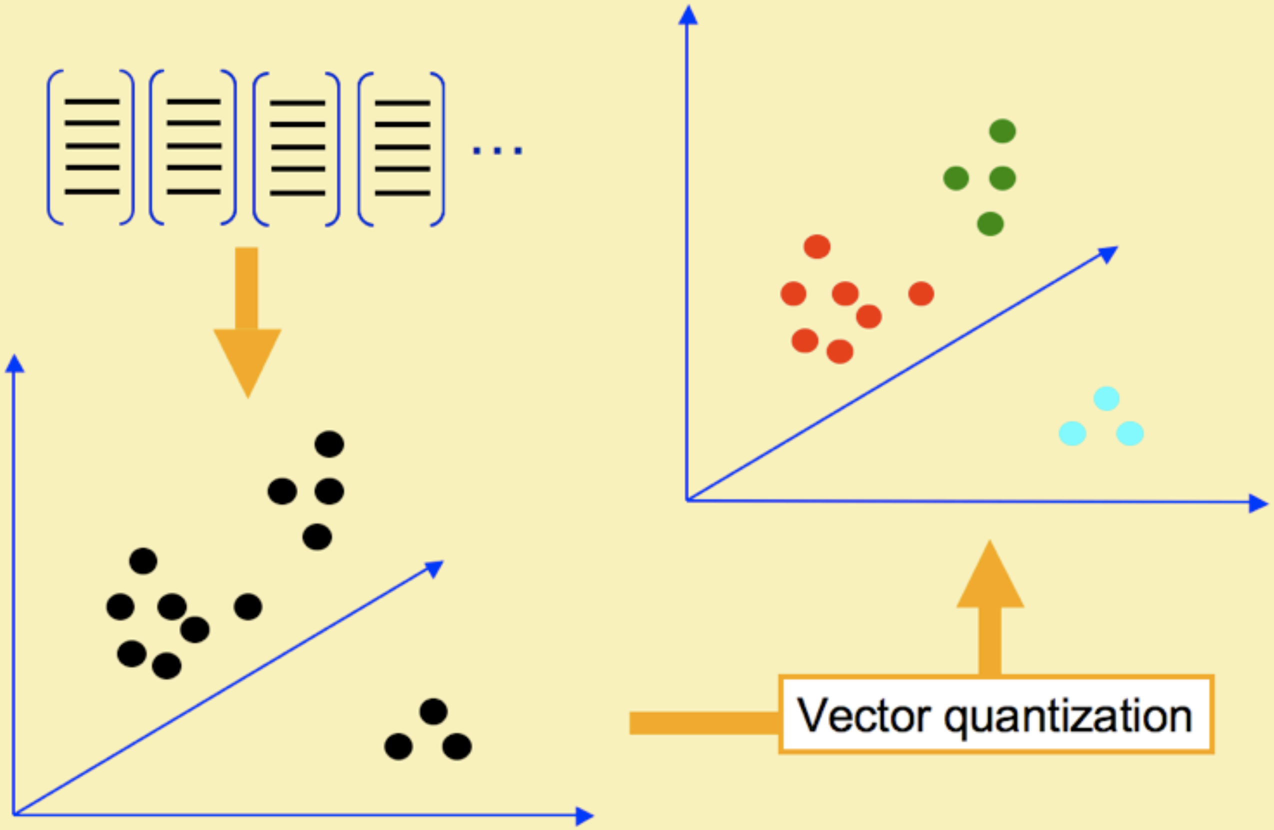
1.Feature detection and representation



2. Codewords dictionary formation

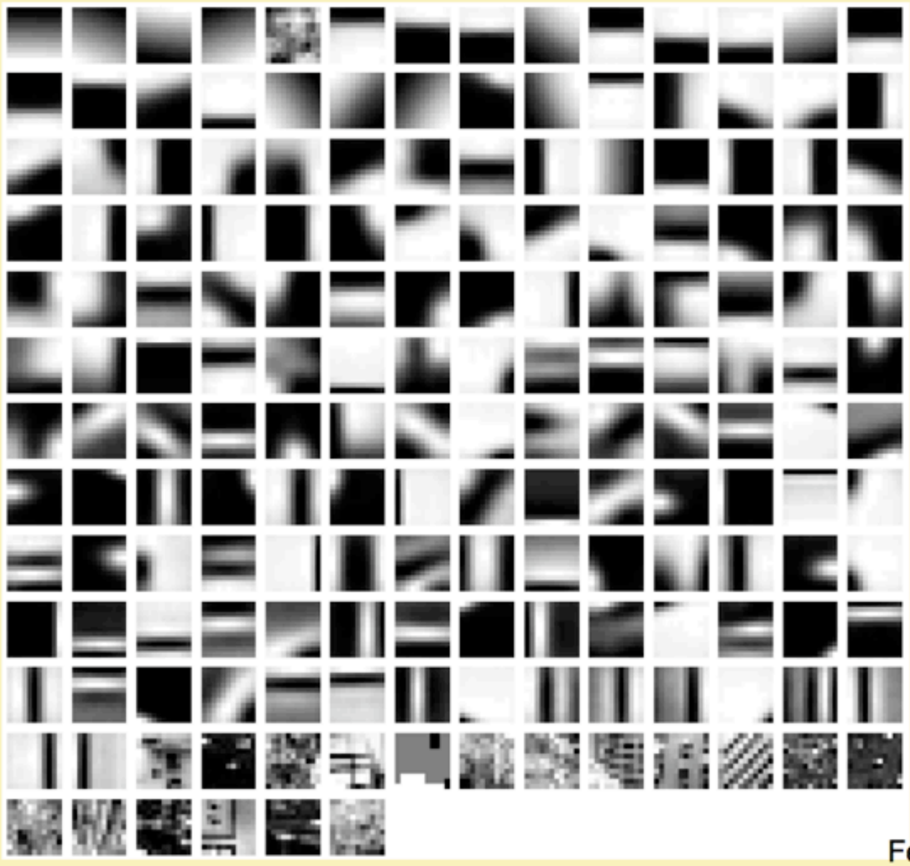


2. Codewords dictionary formation



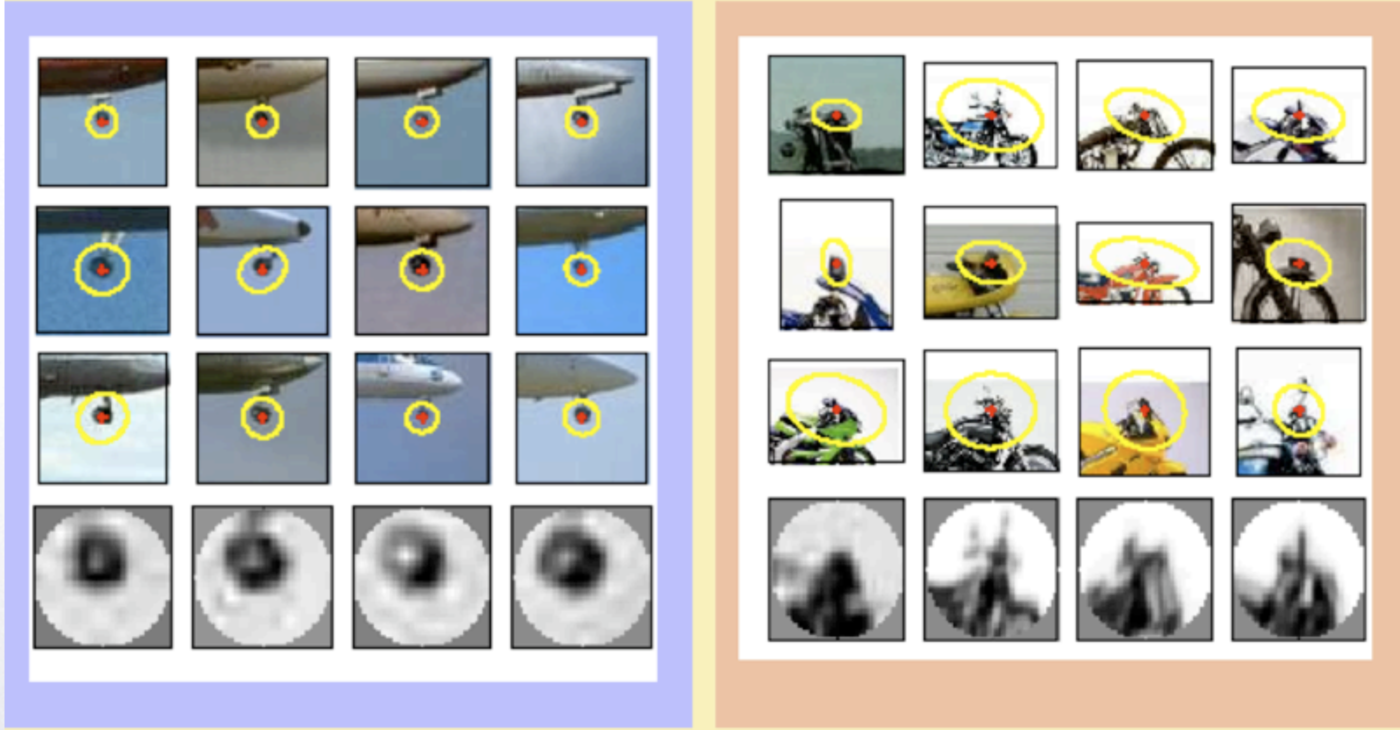
Slide credit: Josef Sivic

2. Codewords dictionary formation



Fei-Fei et al. 2005

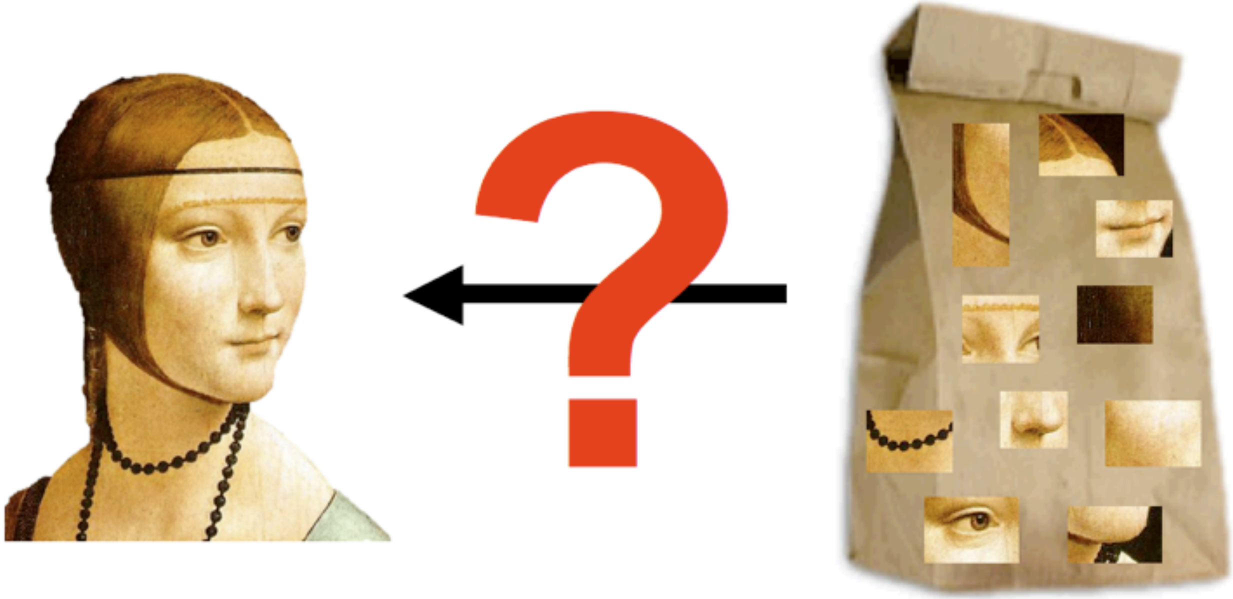
Image patch examples of codewords



Sivic et al. 2005

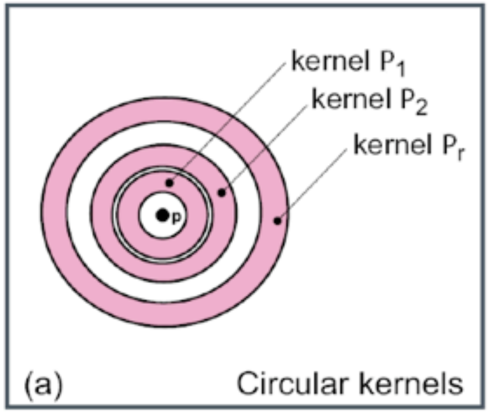
3. Image representation

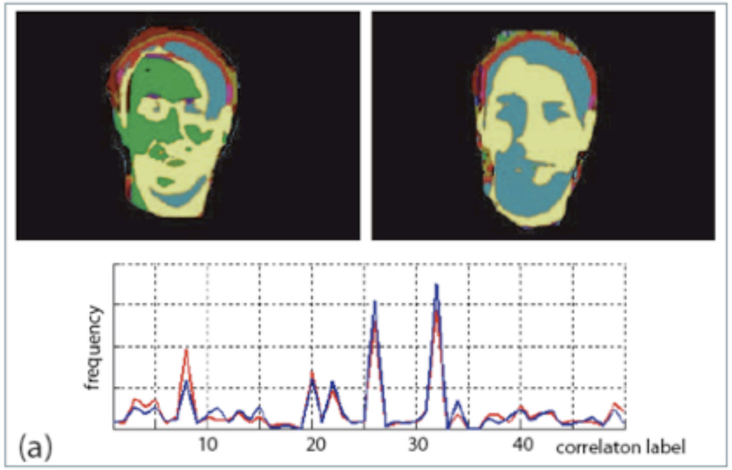
What about spatial info?



What about spatial info?

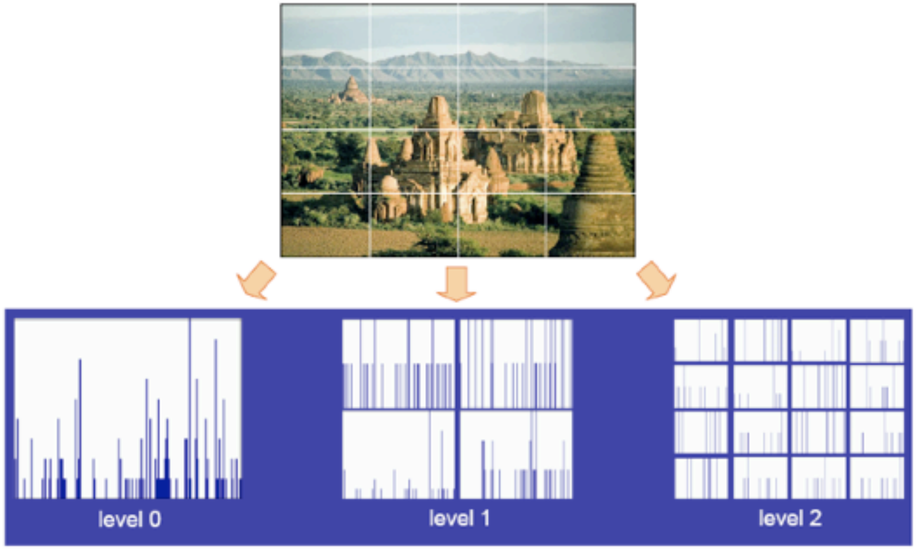
- Feature level
  - Spatial influence through correlogram features: Savarese, Winn and Criminisi, CVPR 2006





What about spatial info?

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



Invariance issues

- Scale and rotation
  - Implicit
  - Detectors and descriptors

Kadir and Brady. 2003

Invariance issues

- Scale and rotation
- Occlusion
  - Implicit in the models
  - Codeword distribution: small variations
  - (In theory) Theme (z) distribution: different occlusion patterns

Invariance issues

- Scale and rotation
- Occlusion
- Translation
  - Encode (relative) location information
    - Sudderth, Torralba, Freeman & Willsky, 2005, 2006
    - Niebles & Fei-Fei, 2007

Invariance issues

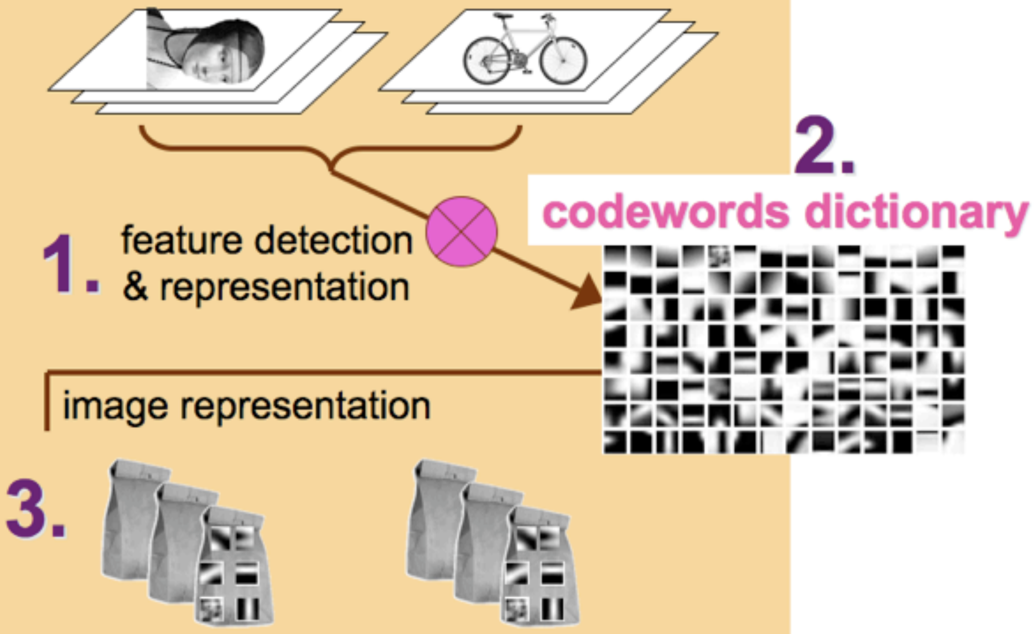
- Scale and rotation
- Occlusion
- Translation
- View point (in theory)
  - Codewords: detector and descriptor
  - Theme distributions: different view points

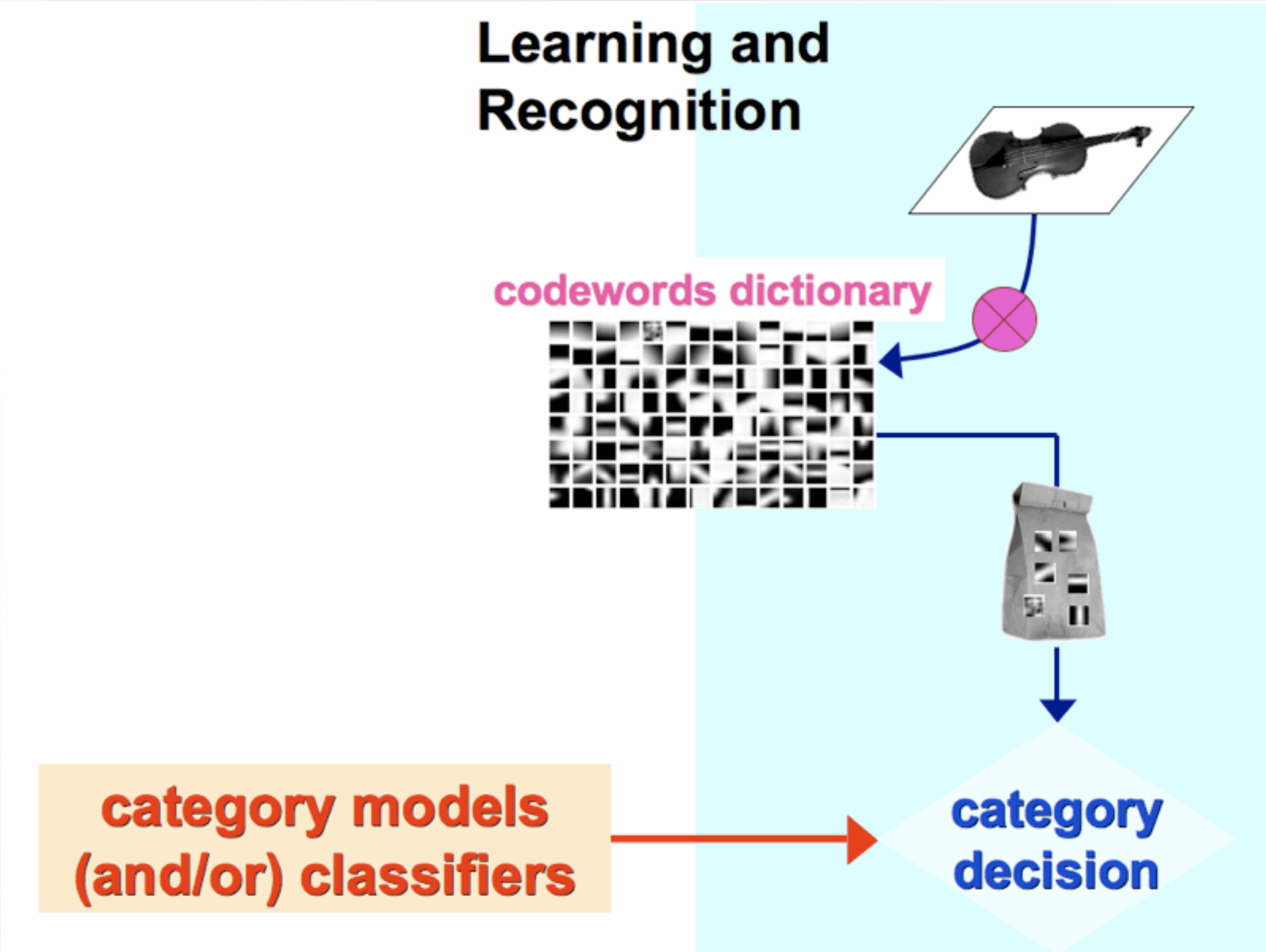
Fergus, Fei-Fei, Perona & Zisserman, 2005

Weakness of the model

- No rigorous geometric information of the object components
- It's intuitive to most of us that objects are made of parts – no such information
- Not extensively tested yet for
  - View point invariance
  - Scale invariance
- Segmentation and localization unclear

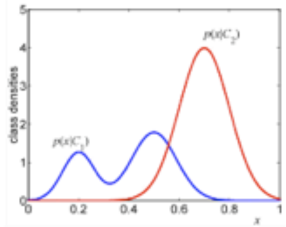
Representation



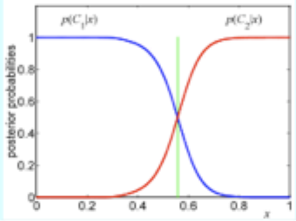

Learning and
Recognition
codewords dictionary
category models
(and/or) classifiers
category
decision

Learning and Recognition

1. Generative method:
   - graphical models

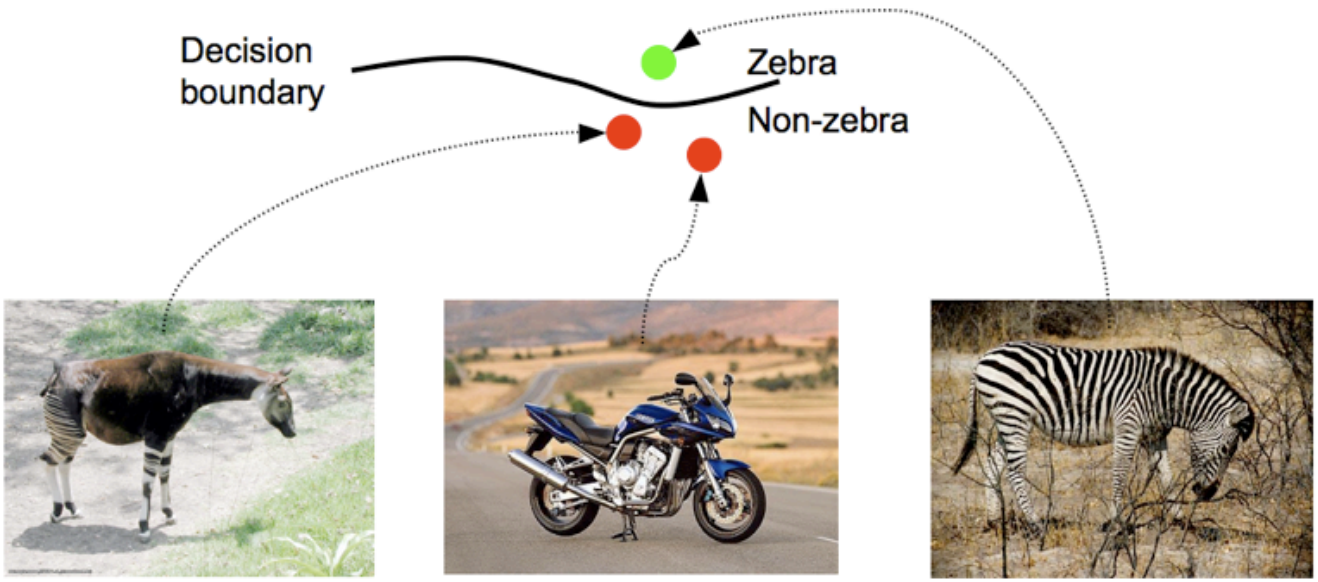


2. Discriminative method:
   - SVM



category models (and/or) classifiers

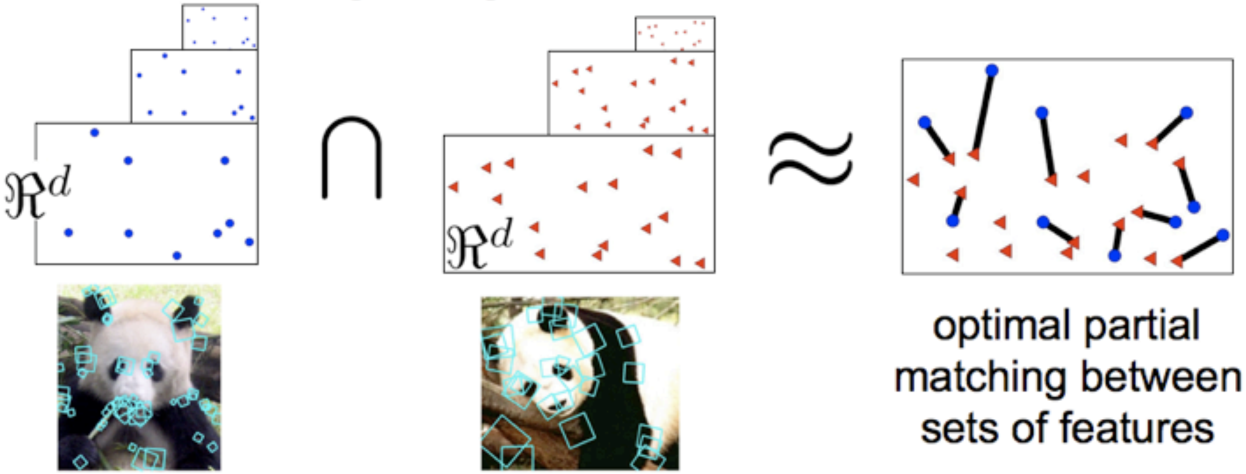
Discriminative methods based on ‘bag of words’ representation



Discriminative methods based on 'bag of words' representation

- Grauman & Darrell, 2005, 2006:
  - SVM w/ Pyramid Match kernels
- Others
  - Csurka, Bray, Dance & Fan, 2004
  - Serre & Poggio, 2005

Summary: Pyramid match kernel



optimal partial matching between sets of features

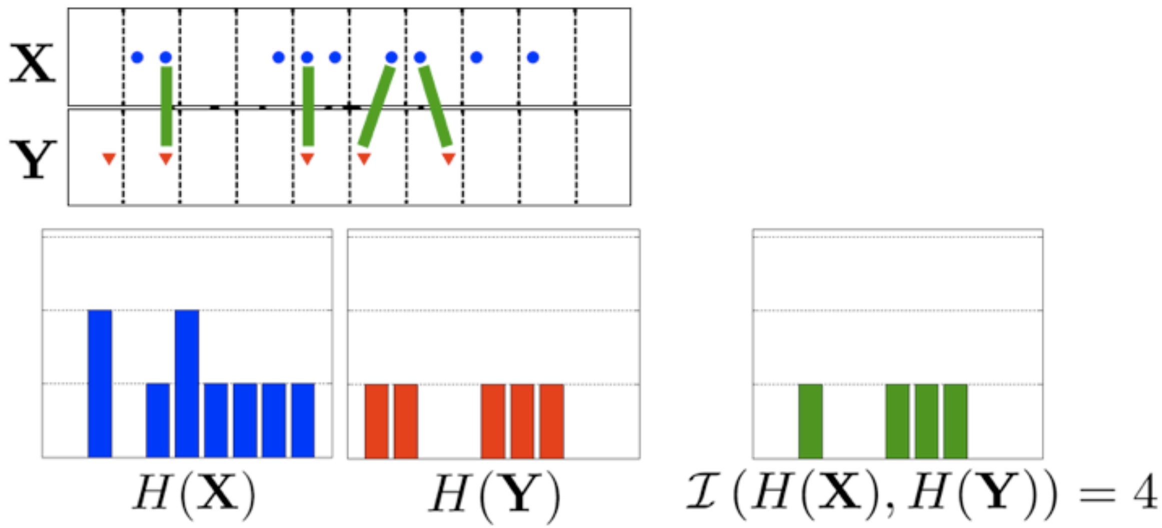
$$K_\Delta\left(\Psi(\mathbf{X}), \Psi(\mathbf{Y})\right)$$

Grauman & Darrell, 2005, Slide credit: Kristen Grauman

Pyramid Match (Grauman & Darrell 2005)

Histogram intersection

$$\mathcal{I}\left(H(\mathbf{X}), H(\mathbf{Y})\right) = \sum_{j=1}^{r} \min\left(H(\mathbf{X})_j, H(\mathbf{Y})_j\right)$$



Slide credit: Kristen Grauman

Pyramid Match (Grauman & Darrell 2005)

Histogram intersection
$$\mathcal{I}\left(H(\mathbf{X}), H(\mathbf{Y})\right) = \sum_{j=1}^{r} \min\left(H(\mathbf{X})_j, H(\mathbf{Y})_j\right)$$

$$\boxed{N_i = \overbrace{\mathcal{I}\left(H_i(\mathbf{X}), H_i(\mathbf{Y})\right)}^{\text{matches at this level}} - \overbrace{\mathcal{I}\left(H_{i-1}(\mathbf{X}), H_{i-1}(\mathbf{Y})\right)}^{\text{matches at previous level}}}$$

Difference in histogram intersections across levels counts *number of new pairs* matched

Slide credit: Kristen Grauman

Pyramid match kernel

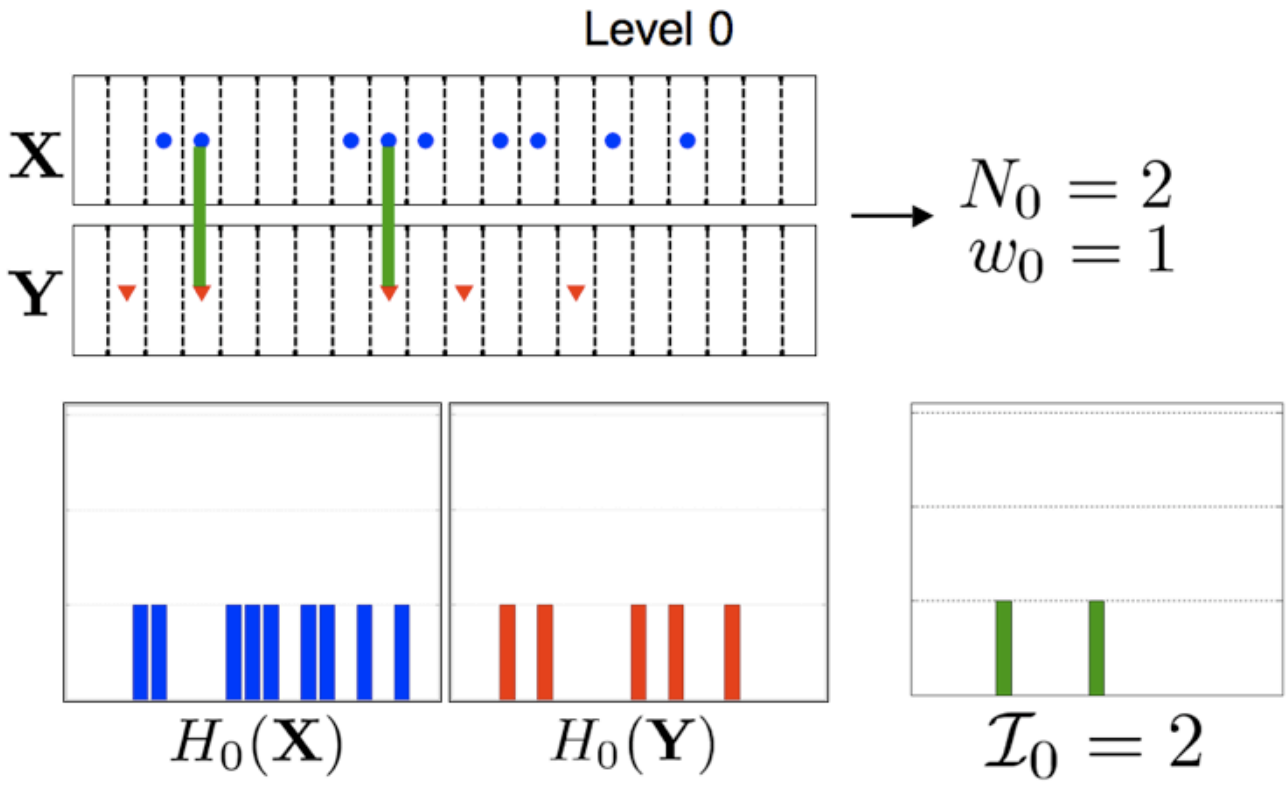
$$K_\Delta \left(\overbrace{\Psi(\mathbf{X}), \Psi(\mathbf{Y})}^{\text{histogram pyramids}} \right) =$$

$$\sum_{i=0}^{L} \underbrace{\frac{1}{2^i}}_{\substack{\text{measure of difficulty of} \\ \text{a match at level } i}} \left(\underbrace{\mathcal{I}\left(H_i(\mathbf{X}), H_i(\mathbf{Y})\right) - \mathcal{I}\left(H_{i-1}(\mathbf{X}), H_{i-1}(\mathbf{Y})\right)}_{\text{number of newly matched pairs at level } i} \right)$$

- Weights inversely proportional to bin size
- Normalize kernel values to avoid favoring large sets

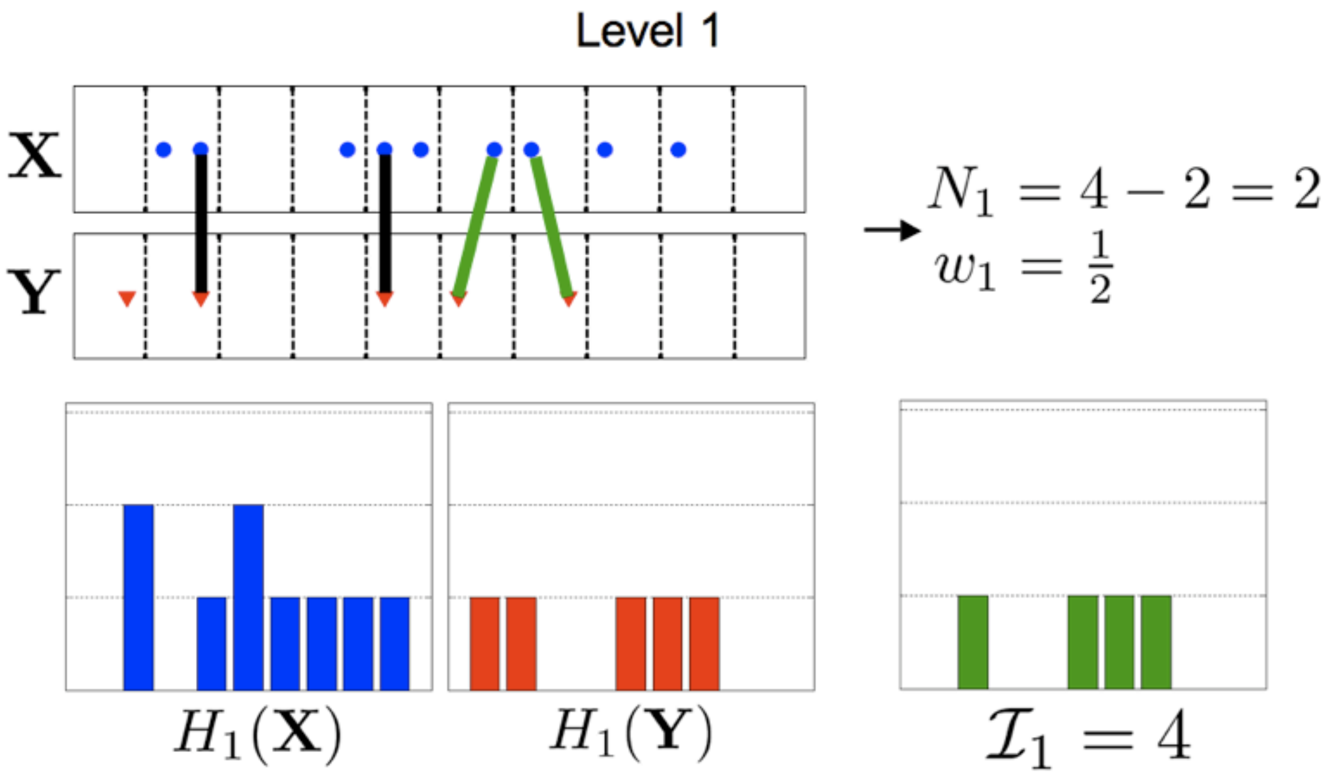
Slide credit: Kristen Grauman

Example pyramid match



Slide credit: Kristen Grauman

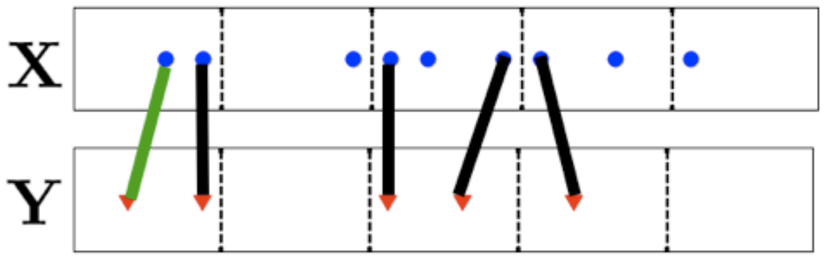
Example pyramid match



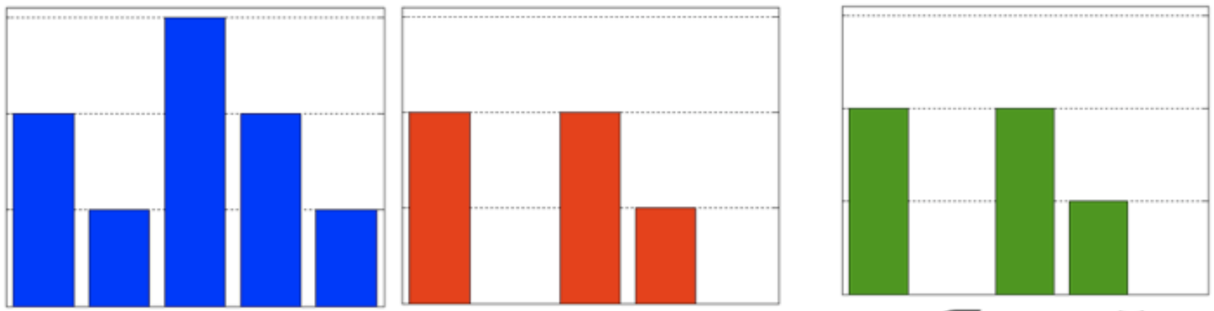
Slide credit: Kristen Grauman

Example pyramid match

Level 2



$$N_2 = 5 - 4 = 1$$
$$w_2 = \frac{1}{4}$$

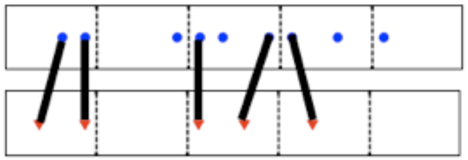


$H_2(\mathbf{X})$ $\quad$ $H_2(\mathbf{Y})$ $\quad$ $\mathcal{I}_2 = 5$

Slide credit: Kristen Grauman

Example pyramid match

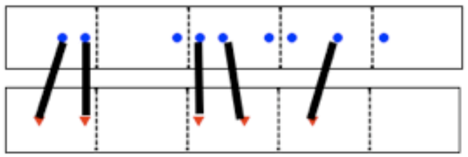
pyramid match



$$K_\Delta = \sum_{i=0}^{L} w_i N_i$$

$$= 1(2) + \tfrac{1}{2}(2) + \tfrac{1}{4}(1) = 3.25$$

optimal match

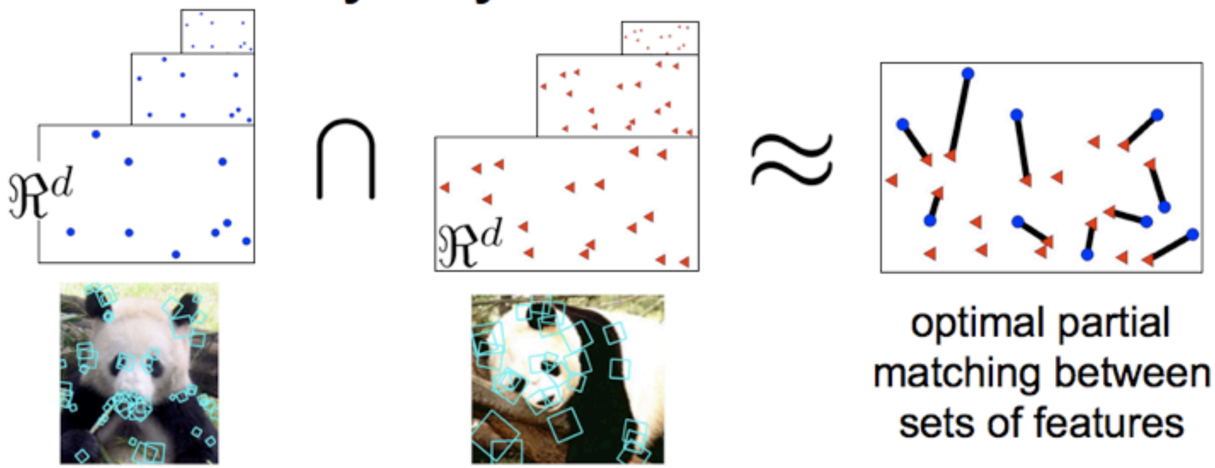


$$K = \max_{\pi:\mathbf{X}\to\mathbf{Y}} \sum_{\mathbf{x}_i \in \mathbf{X}} \mathcal{S}(\mathbf{x}_i, \pi(\mathbf{x}_i))$$

$$= 1(2) + \tfrac{1}{2}(3) = 3.5$$

Slide credit: Kristen Grauman

Summary: Pyramid match kernel



optimal partial matching between sets of features

$$K_\Delta\left(\Psi(\mathbf{X}), \Psi(\mathbf{Y})\right) =$$

$$\sum_{i=0}^{L} \underbrace{\frac{1}{2^i}}_{\text{difficulty of a match at level i}} \left(\underbrace{\mathcal{I}\left(H_i(\mathbf{X}), H_i(\mathbf{Y})\right) - \mathcal{I}\left(H_{i-1}(\mathbf{X}), H_{i-1}(\mathbf{Y})\right)}_{\text{number of new matches at level i}} \right)$$

Slide credit: Kristen Grauman

Object recognition results

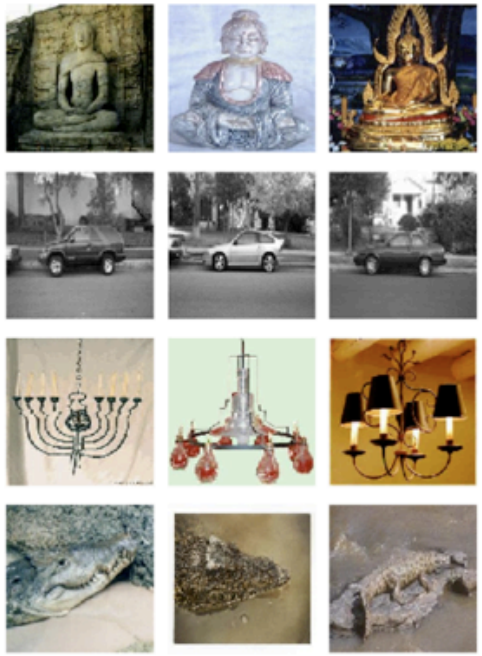
- ETH-80 database
  8 object classes
  (*Eichhorn and Chapelle 2004*)
- Features:
  - Harris detector
  - PCA-SIFT descriptor, *d*=10

Kernel	Complexity	Recognition rate
Match *[Wallraven et al.]*	$O(dm^2)$	84%
Bhattacharyya affinity *[Kondor & Jebara]*	$O(dm^3)$	85%
Pyramid match	$O(dmL)$	84%

Slide credit: Kristen Grauman

Object recognition results

- Caltech objects database
  101 object classes
- Features:
  - SIFT detector
  - PCA-SIFT descriptor, d=10
- 30 training images / class
- 43% recognition rate
  (1% chance performance)
- 0.002 seconds per match



Slide credit: Kristen Grauman

15 min break!

Object Recognition --the importance of multiple cues

Which features should I use?

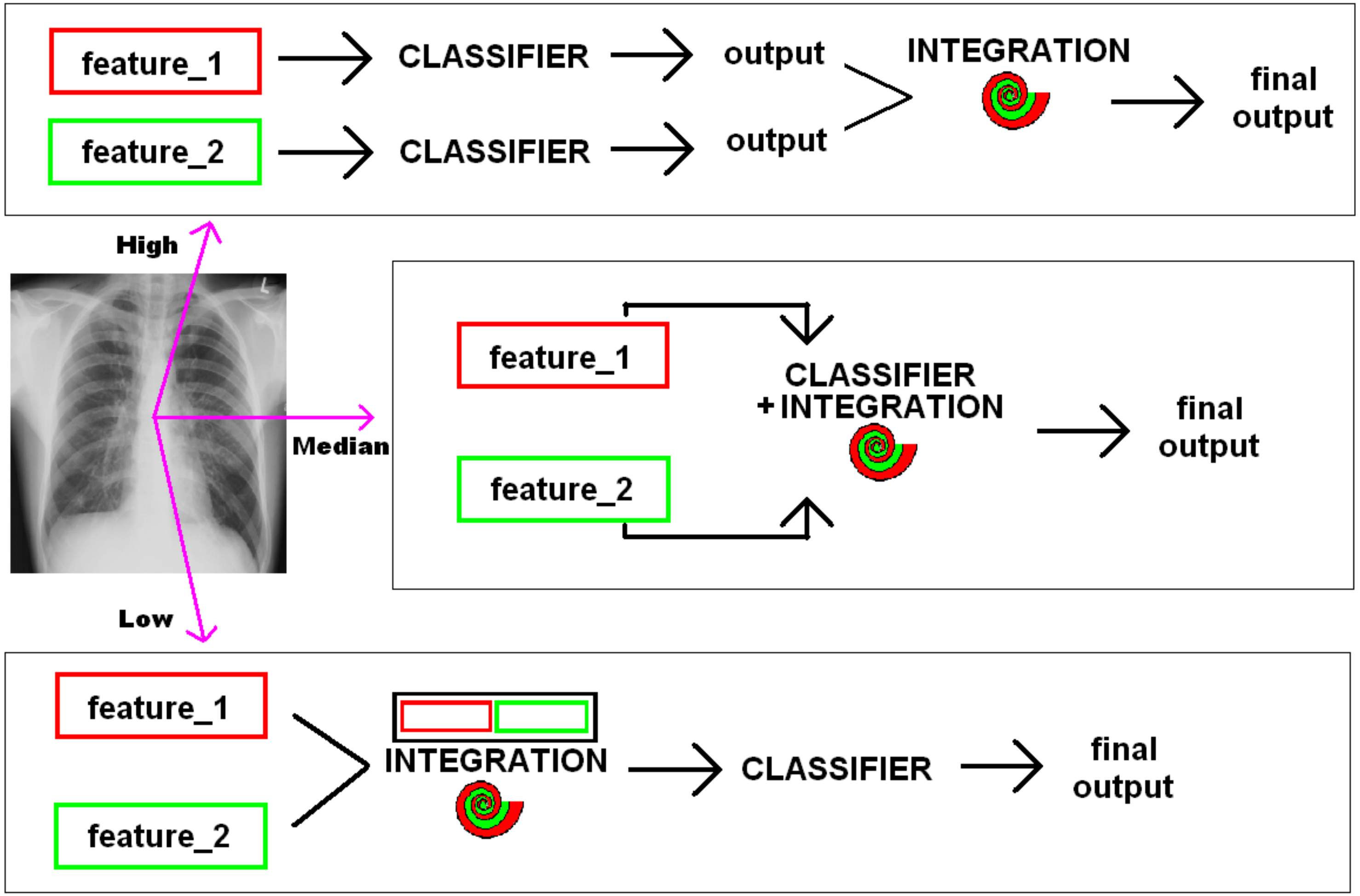
Which features should I use?

Which features should I use?

Which features should I use?



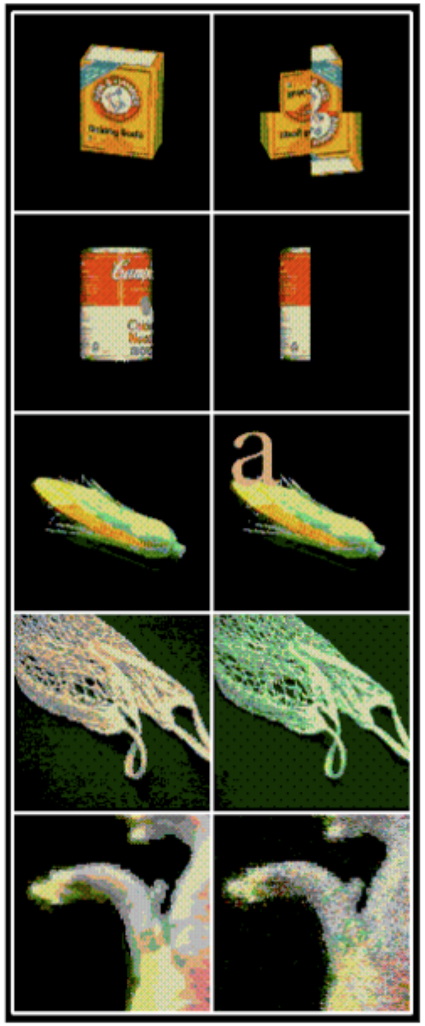
- One should try to chose the features depending on the problem at hand
- When dealing with a multi-class categorization problem, not obvious what to chose --combine many!


feature_1
feature_2
CLASSIFIER
CLASSIFIER
output
output
INTEGRATION
final output
High
Median
Low
feature_1
feature_2
CLASSIFIER + INTEGRATION
final output
feature_1
feature_2
INTEGRATION
CLASSIFIER
final output

B. W. Mel. *SEEMORE: combining color, shape and texture histogramming in a neurally inspired approach to visual object recognition.* Neural Computation, 9, 777-804 (1997)

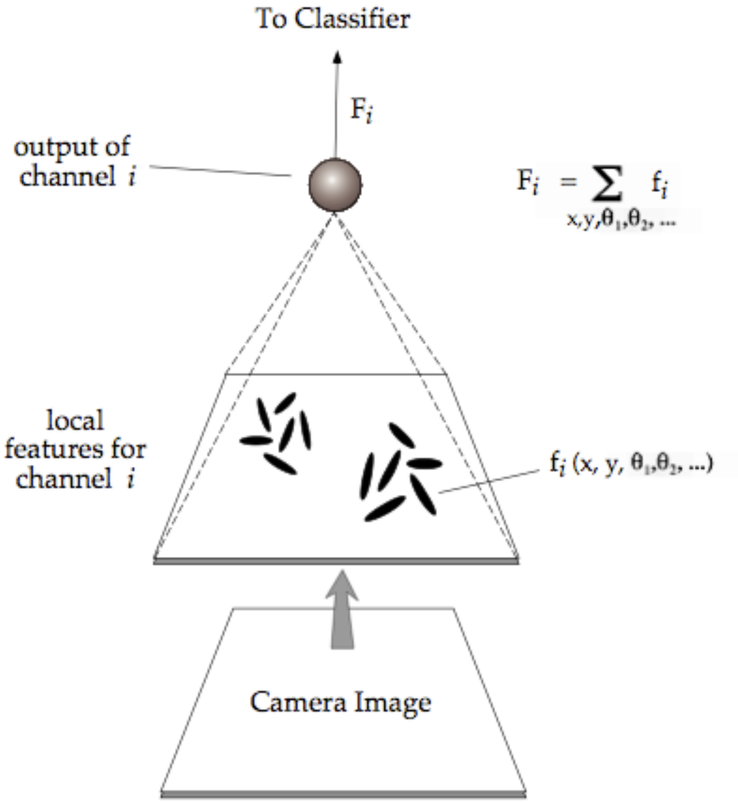
- **Contribution I**:first example of multi cue object recognition system
- **Contribution II:** biologically motivated low-level integration scheme

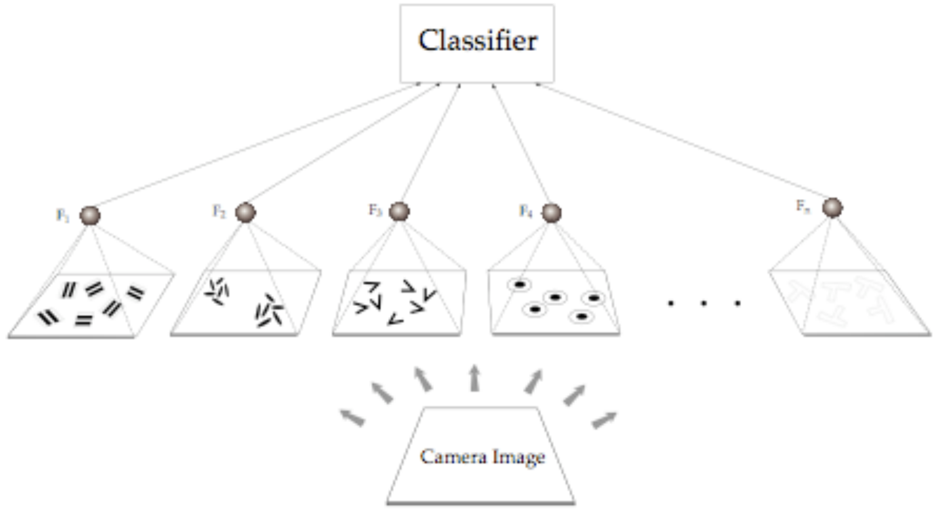
- 100 objects
- training: 12 to 36 views at different viewpoints and scales

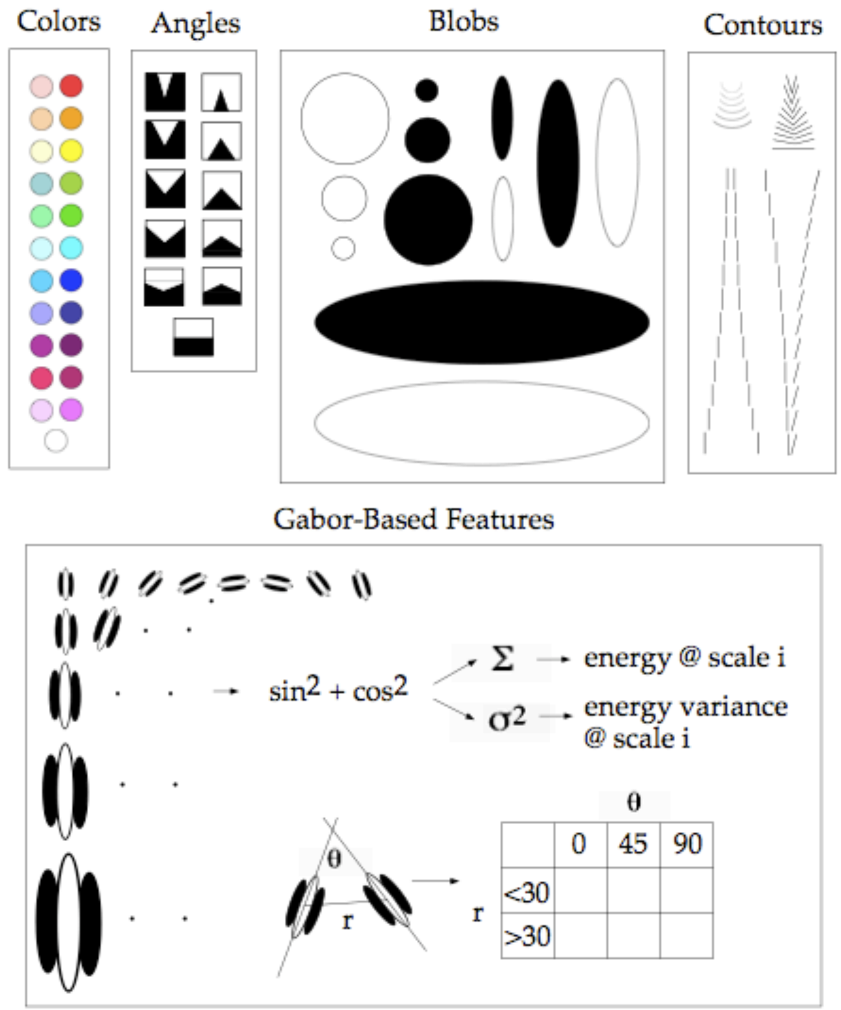


- test to various forms of degradations: scrambling, occlusion, coloring

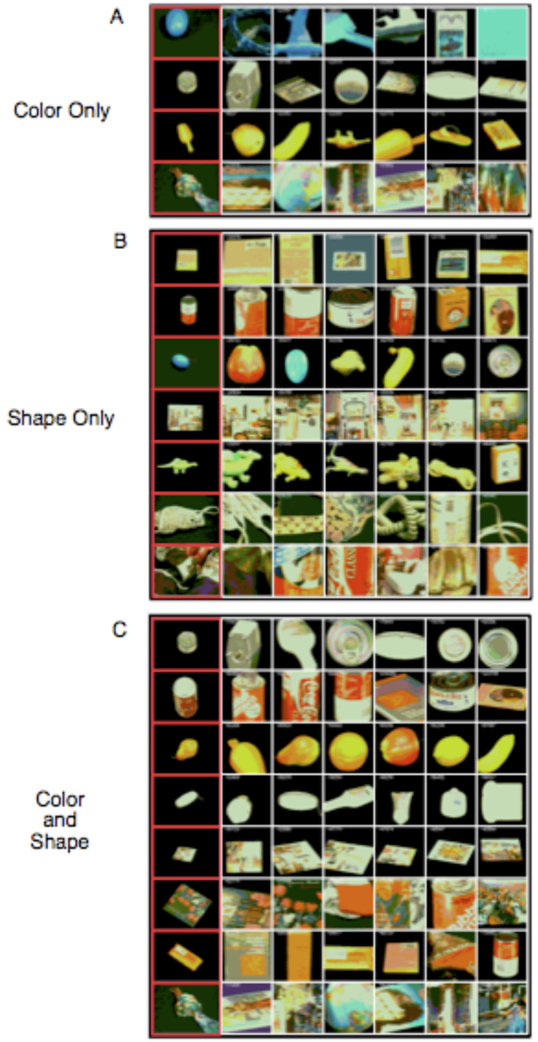
features


To Classifier
F_i
output of channel i
$F_i = \sum_{x,y,\theta_1,\theta_2,...} f_i$
local features for channel i
f_i (x, y, θ_1,θ_2, ...)
Camera Image


Classifier
. . .
Camera Image



five different groups of features, combined together in a single feature representation



results

	Intact	Nonrigid	Scrambled	Occluded	Cluttered	Colorized	Noisy
Shape only	79.7	76.7	62.2	38.2	57.3	43.5	35.8
Color only	87.3	94.4	86.5	72.2	61.2	6.8	47.2
Color and shape	96.7	97.8	93.7	79.0	79.0	19.8	58.3

M. E. Nilsback, B. Caputo. *Cue Integration through discriminative accumulation.* Proc CVPR 2004.

- **Contribution I**: cast the cue integration problem within a discriminative framework
- **Contribution II:** one of the first examples of high-level integration applied to the object recognition problem
- **Contribution III**: one of the first examples of high-level integration using SVM

Object Categorization

Three possible scenarios:

1. Learning and Recognition in Controlled Settings

Object Categorization

Three possible scenarios:

1. Learning and Recognition in Controlled Settings
2. Learning in Controlled Settings, Recognition Unconstrained

Object Categorization

Three possible scenarions:

1. Learning and Recognition in Controlled Settings
2. Learning in Controlled Settings, Recognition Unconstrained
3. Learning and Recognition Unconstrained




Cue Integration via Accumulation

- **Step 1: Single-cue SVMs** From the original training set $\{\boldsymbol{I}_i^j\}_{i=1}^{N_j}$, for each object j, with $j = 1, \ldots, M$ define P new training sets $\{T_p(\boldsymbol{I}_i^j)\}_{i=1}^{N_j}, j = 1, \ldots, M, p = 1 \ldots, P$, each relative to a single cue. For each new training set we train an SVM. Then, given a test image $\widehat{\boldsymbol{I}}$, for each single-cue SVM we compute the margin:

$$D_j(p) = \sum_{i=1}^{m_j^p} \alpha_{ij}^p y_{ij} K_p(T_p(\boldsymbol{I}_i^j), T_p(\widehat{\boldsymbol{I}})) + b_j^p.$$

The index p on $(m_j^p, \alpha_{ij}^p, K_p(\cdot,\cdot), b_j^p)$ indicates that in general these quantities have different values for different cues.

Cue Integration via Accumulation

- Step 2: Discriminative Accumulation After we collect all the margins $\{D_j(p)\}_{p=1}^{P}$, for all the j objects $j = 1, \ldots, M$ and the p cues $p = 1, \ldots, P$, we classify the image $\widehat{I}$ using their linear combination:

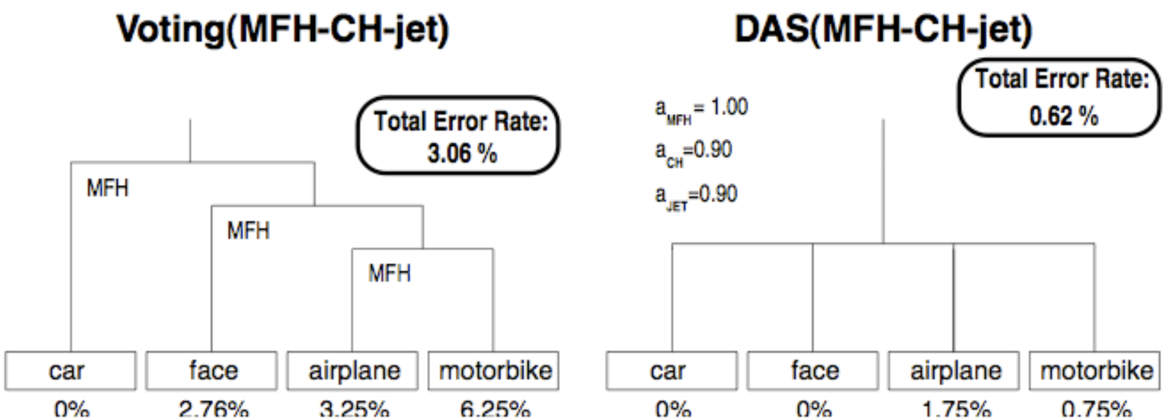
$$j^* = \operatorname*{argmax}_{j=1}^{M} \left\{ \sum_{p=1}^{P} a_p D_j(p) \right\}, a_p \in \Re^+.$$

$\{a_p\}_{p=1}^{P}$ are evaluated via model selection during the training step.

This means that the relevance of each cue, for a specific task, is evaluated during the training step from the training data

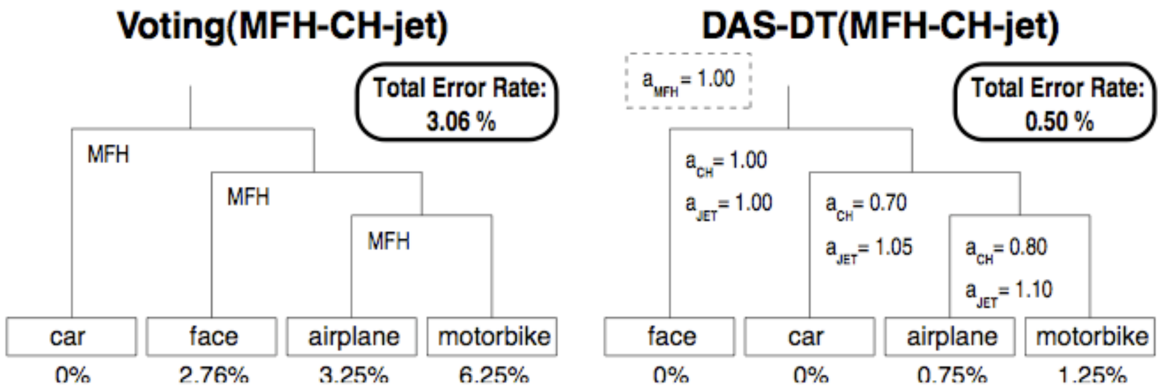
Object Categorization

Results on Caltech Database



Object Categorization

Results on Caltech Database



B. Caputo, Categorization using a Discriminative Approach

P. Gehler, S. Nowozin. *On feature combination for multiclass object classification*. Proc ICCV 2009.

- **Contribution I**: cast the cue integration problem within the Multi Kernel Learning (MKL) framework
- **Contribution II:** thorough evaluation of MKL algorithms, definition of competitive baselines
- **Contribution III**: boosting-based high-level cue integration scheme

Formal definition:

Definition 1 (Feature Combination Problem) *Given a training set* $\{(x_i, y_i)\}_{i=1,\dots,N}$ *of* N *instances consisting of an image* $x_i \in \mathcal{X}$ *and a class label* $y_i \in \{1, \dots, C\}$, *and given a set of* F *image features* $f_m : \mathcal{X} \to \mathbb{R}^{d_m}$, $m = 1, \dots, F$ *where* d_m *denotes the dimensionality of the* m*'th feature, the problem of learning a classification function* $y : \mathcal{X} \to \{1, \dots, C\}$ *from the features and training set is called* feature combination problem.

Kernel function between image features:

$$k_m(x, x') = k(f_m(x), f_m(x'))$$

Kernel response of the m-th feature:

$$K_m(x) = [k_m(x, x_1), k_m(x, x_2), \ldots, k_m(x, x_N)]^T$$

Kernel selection = feature selection

Baseline Method I: Averaging Kernels

$$k^*(x,x') = \tfrac{1}{F}\sum_{m=1}^{F} k_m(x,x')$$

Baseline Method II: Product Kernels

$$k^*(x,x') = \left(\prod_{m=1}^{F} k_m(x,x')\right)^{1/F}$$

Multiple Kernel Learning: joint optimization over a linear combination of kernels and SVM parameters

$$k^*(x, x') = \sum_{m=1}^{F} \beta_m k_m(x, x')$$

$$\beta_m \geq 0 \qquad \sum_{m=1}^{F} \beta_m = 1$$

The final decision function of MKL is

$$F_{\mathrm{MKL}}(x) = \mathbf{sign}\left(\sum_{m=1}^{F} \beta_m (K_m(x)^T \alpha + b) \right)$$

Boosting approaches: LPBoost

If one considers $f_m(x) = K_m(x)^T\alpha + b$

Then MKL can be seen as

$$F(x) = \text{sign} \sum_{m=1}^{F} \beta_m f_m(x)$$

first train the individual SVMs, then do boosting to find the betas!

Two possible variations:
a single beta for all classes, or each class a specific beta

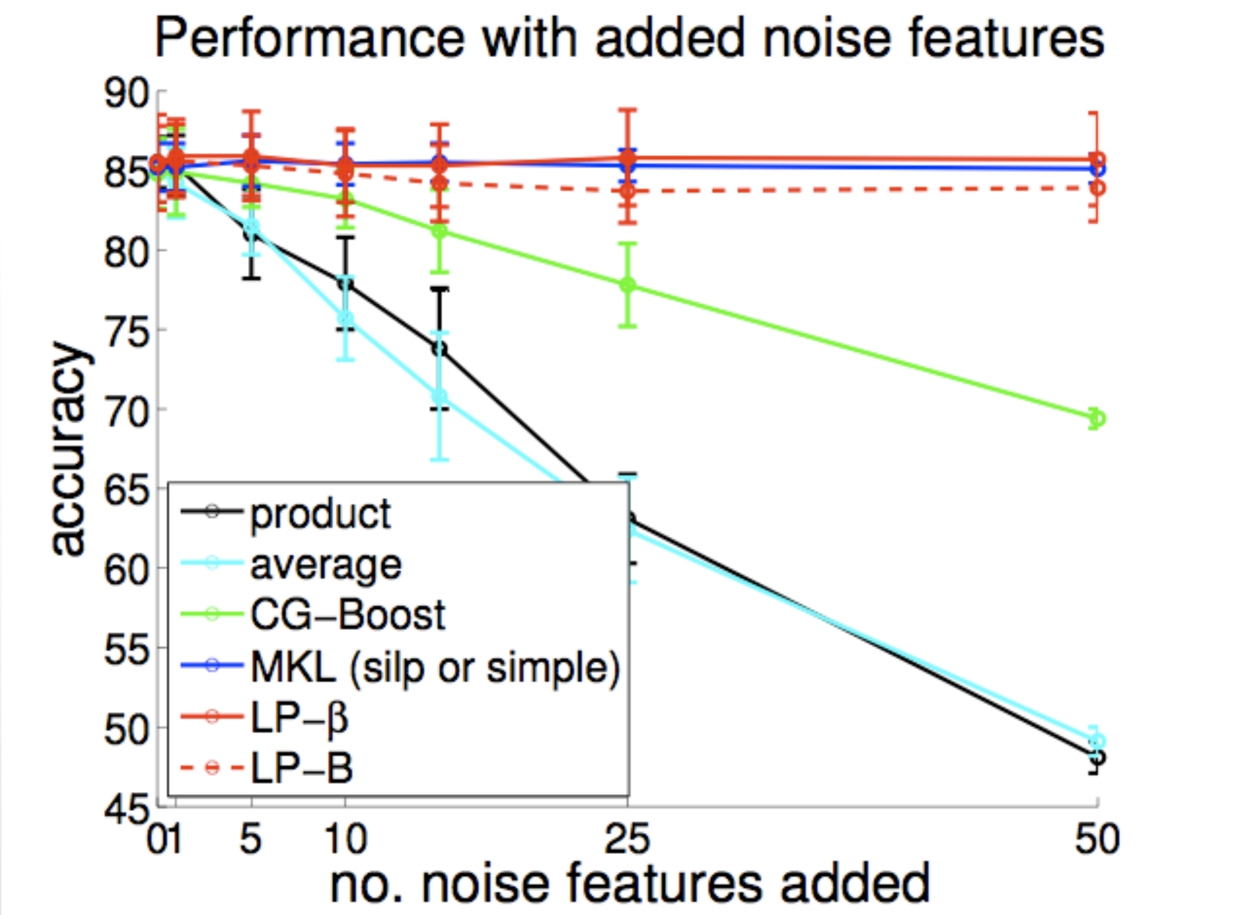
Summary of algorithms

Name	Test-time function	Coefficients	Training	Parameters	References
Averaging	$y(x) = \underset{c=1,\ldots,C}{\text{argmax}}\left[\left(\frac{1}{F}\sum_{m=1}^{F} K_m(x)\right)^T \alpha_c + b_c\right]$	$\alpha \in \mathbb{R}^{C\times N}$ $b \in \mathbb{R}^{C}$	$(\alpha, b)_c$, ind.	C_c	
Product	$y(x) = \underset{c=1,\ldots,C}{\text{argmax}}\left[\left(\left(\prod_{m=1}^{F} K_m(x)\right)^{1/F}\right)^T \alpha_c + b_c\right]$	$\alpha \in \mathbb{R}^{C\times N}$ $b \in \mathbb{R}^{C}$	$(\alpha, b)_c$, ind.	C_c	
MKL	$y(x) = \underset{c=1,\ldots,C}{\text{argmax}} \sum_{m=1}^{F} \beta_m^c \left(K_m(x)^T \alpha_c + b_c\right)$	$\beta \in \mathbb{R}^{C\times F}$ $\alpha \in \mathbb{R}^{C\times N}$ $b \in \mathbb{R}^{C}$	$(\alpha_c, b_c, \beta^c)_c$ ind.	C_c	[20, 18, 1]
CG-Boost	$y(x) = \underset{c=1,\ldots,C}{\text{argmax}}\left[\sum_{m=1}^{F} K_m(x)^T \alpha_{c,m} + b_c\right]$	$\alpha \in \mathbb{R}^{C\times F\times N}$ $b \in \mathbb{R}^{C}$	$(\alpha, b)_c$, ind.	C_c	[2]
LP-β	$y(x) = \underset{c=1,\ldots,C}{\text{argmax}} \sum_{m=1}^{F} \beta_m \left(K_m(x)^T \alpha_{c,m} + b_{c,m}\right)$	$\beta \in \mathbb{R}^{F}$ $\alpha \in \mathbb{R}^{C\times F\times N}$ $b \in \mathbb{R}^{C\times F}$	1. $(\alpha, b)_c$, ind 2. β, jointly	1. C_m 2. $\nu \in (0, 1)$	[4]
LP-B	$y(x) = \underset{c=1,\ldots,C}{\text{argmax}} \sum_{m=1}^{F} B_m^c \left(K_m(x)^T \alpha_{c,m} + b_{c,m}\right)$	$B \in \mathbb{R}^{F\times C}$ $\alpha \in \mathbb{R}^{C\times F\times N}$ $b \in \mathbb{R}^{C\times F}$	1. $(\alpha, b)_c$, ind 2. B, jointly	1. C_m, 2. $\nu \in (0, 1)$	[4]

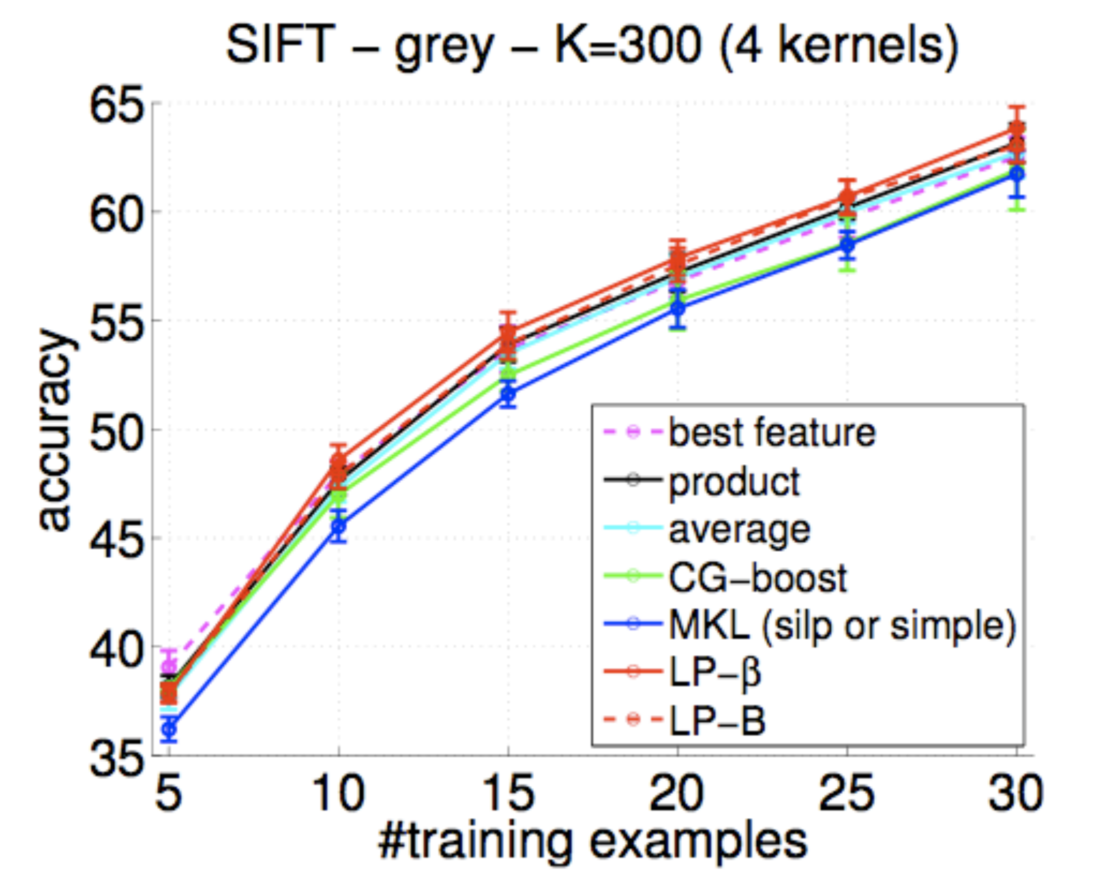
Results: Oxford Flowers Database

Single features			Combination methods		
Method	Accuracy	Time	Method	Accuracy	Time
Colour	60.9 ± 2.1	3	product	85.5 ± 1.2	2
Shape	70.2 ± 1.3	4	averaging	84.9 ± 1.9	10
Texture	63.7 ± 2.7	3	CG-Boost	84.8 ± 2.2	1225
HOG	58.5 ± 4.5	4	MKL (SILP)	85.2 ± 1.5	97
HSV	61.3 ± 0.7	3	MKL (Simple)	85.2 ± 1.5	152
siftint	70.6 ± 1.6	4	LP-β	85.5 ± 3.0	80
siftbdy	59.4 ± 3.3	5	LP-B	85.4 ± 2.4	98

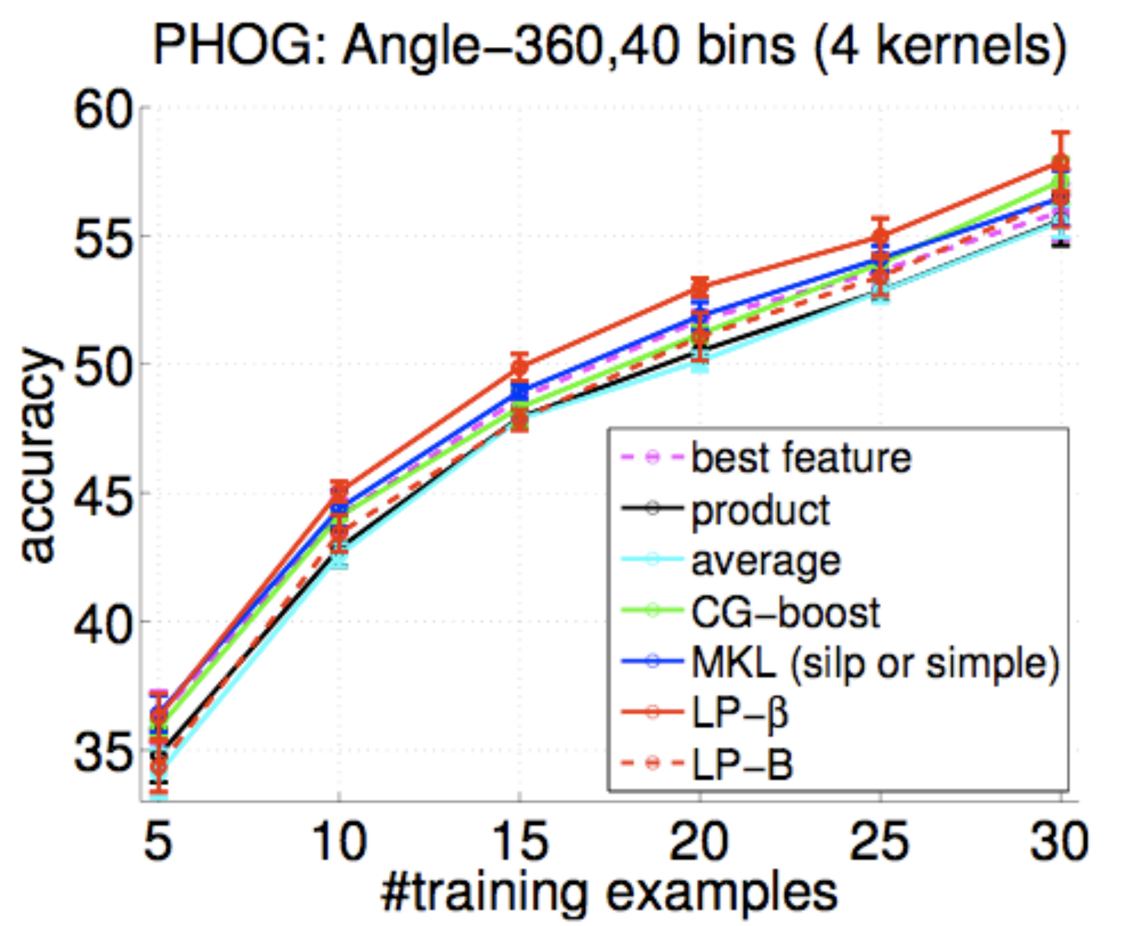
Results: Oxford Flowers Database



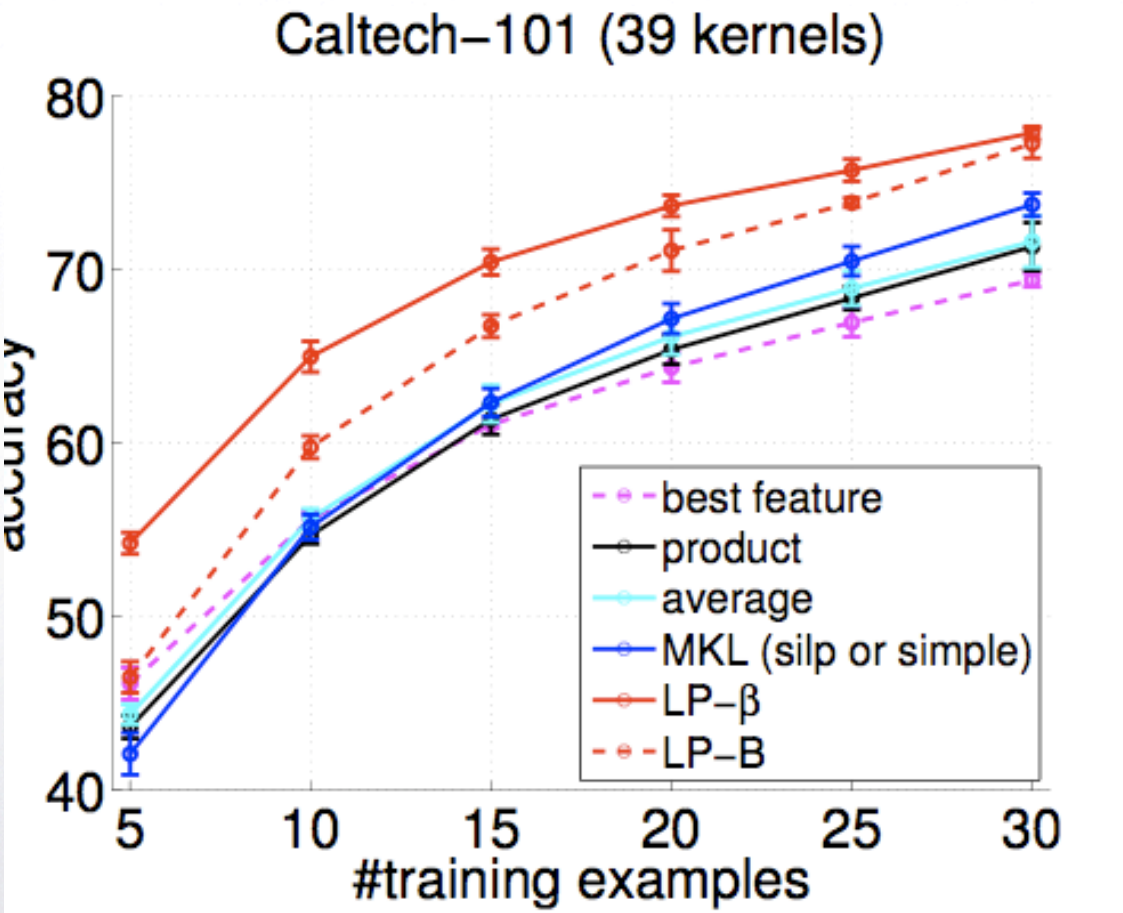
Results: Caltech 101 Database



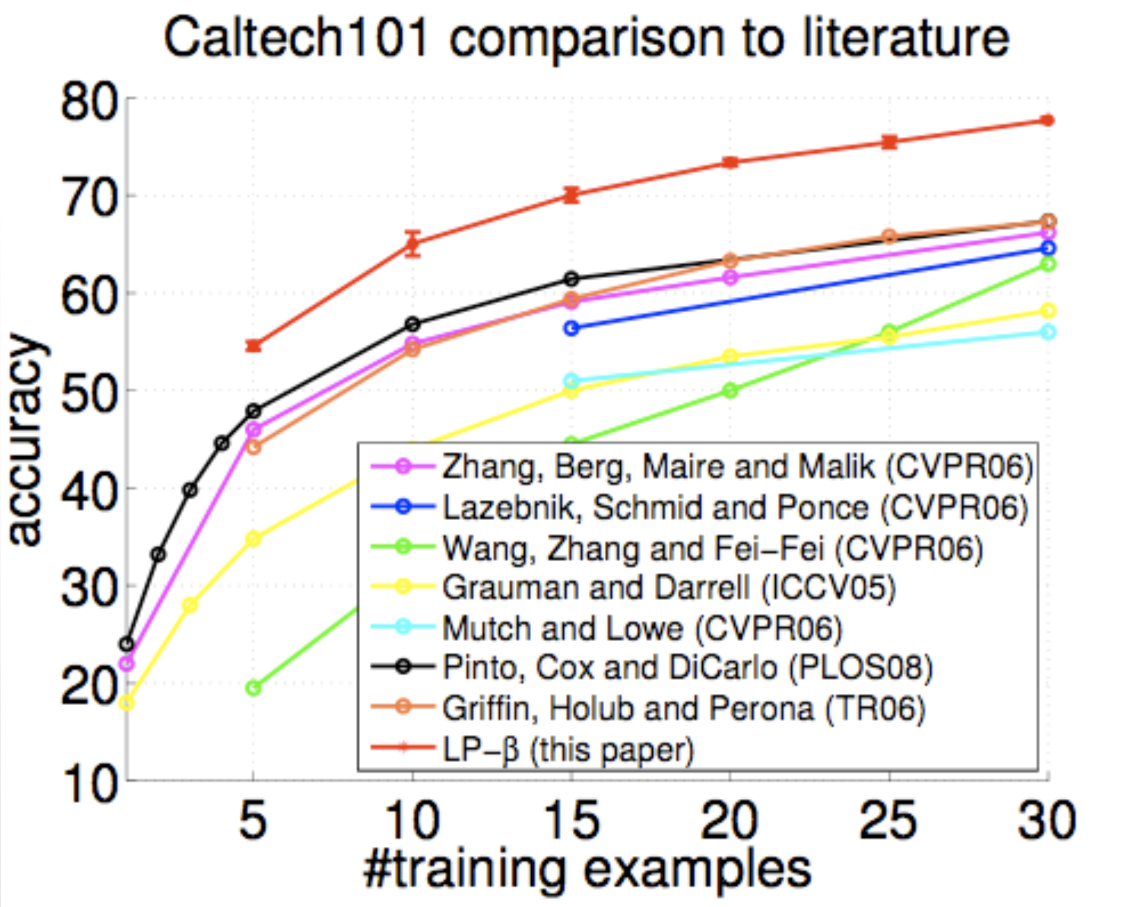
Results: Caltech 101 Database



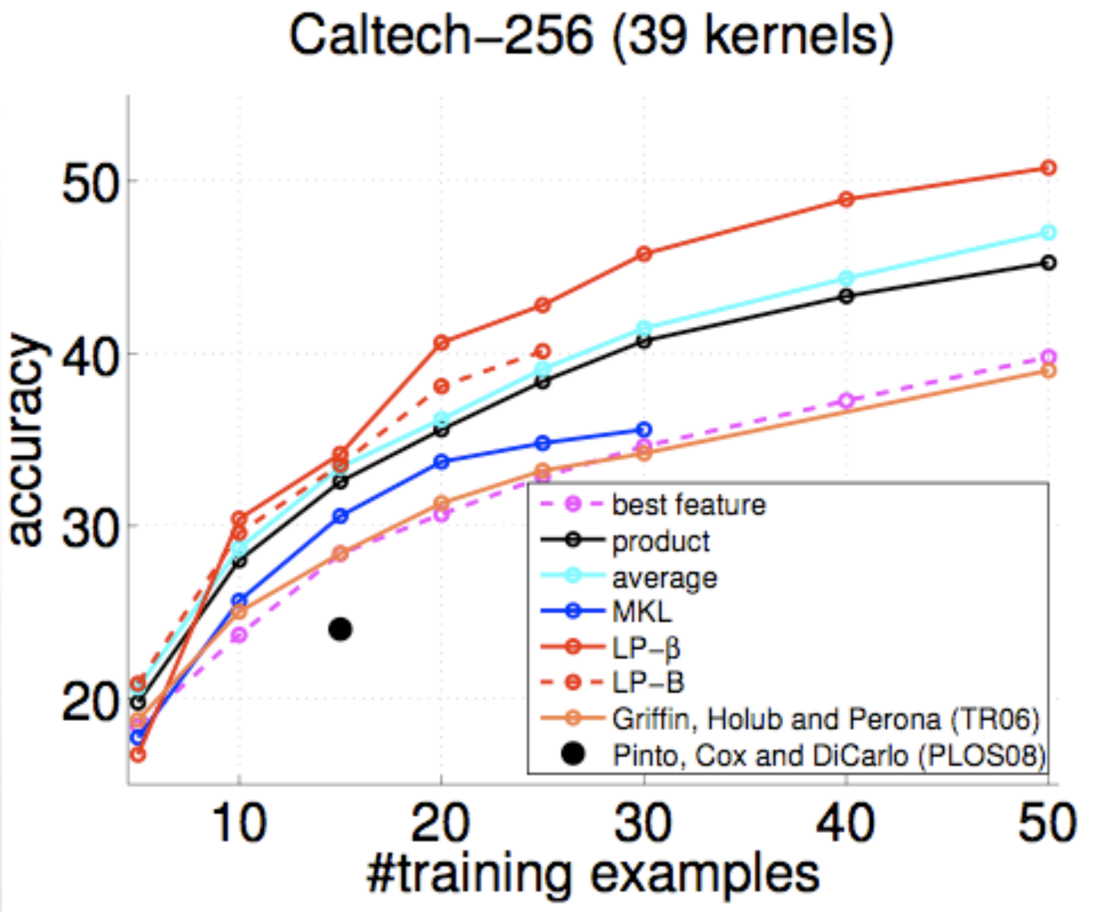
Results: Caltech 101 Database



Results: Caltech 101 Database



Results: Caltech 101 Database



Wrapping up

- Always Always Always use multiple cues!
- No-brainer cue integration method: kernel averaging
- More sophisticated things: high-level schemes most probably give better results, but the computational cost considerably higher --is it worth it?