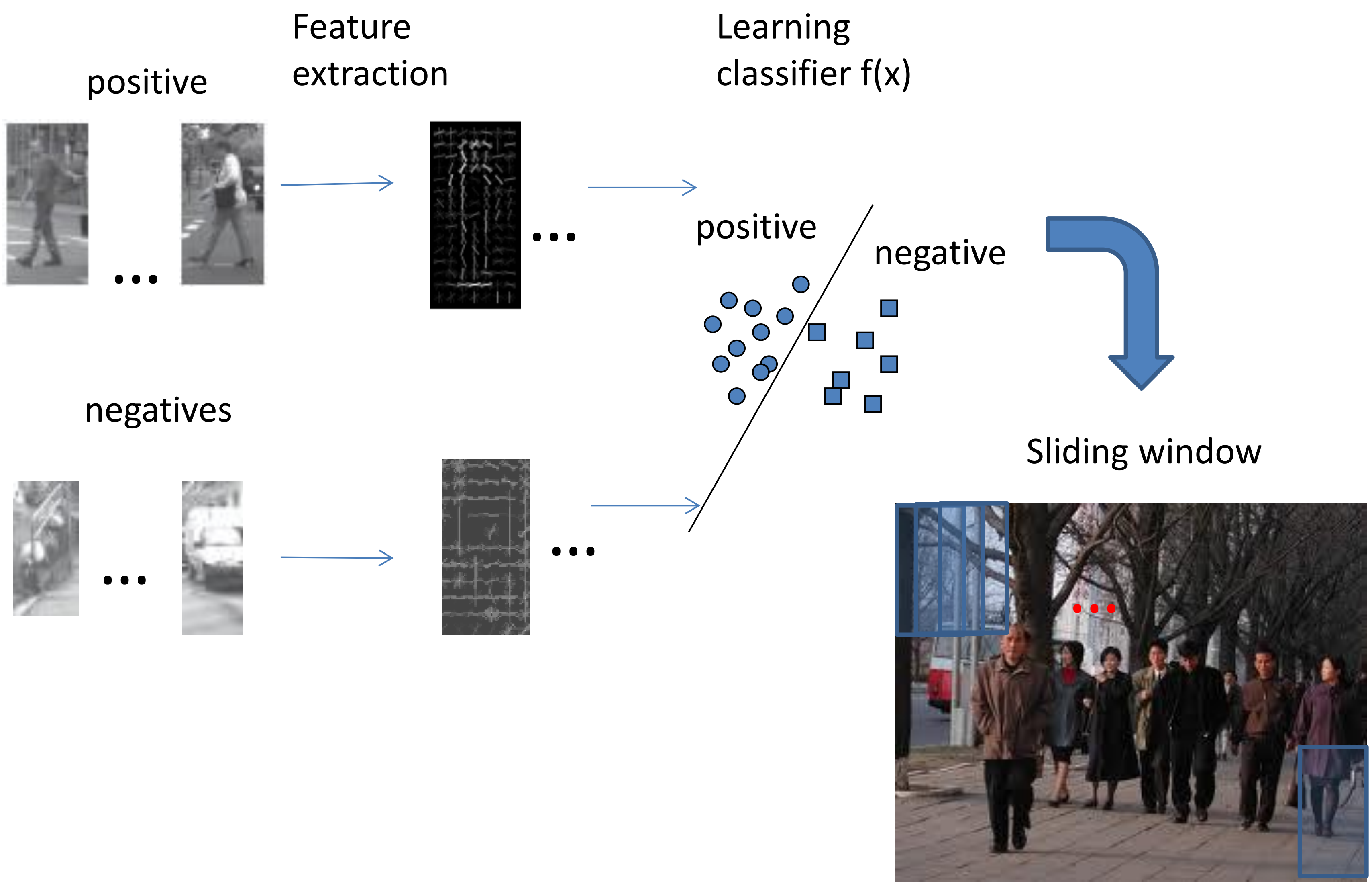
Using Context in Object Recognition

Using the Forest to See the Trees: Exploiting Context for Visual Object Detection and Localization. Torralba, Murphy, and Freeman. CACM 2009.

OBJECT LOCALIZATION

Sliding Window Approach



False Positive Problem

100
categories

…

False Positive Problem

1

100 categories

...

False Positive Problem

2

100
categories

...

False Positive Problem

100
categories

...

3

False Positive Problem

100 categories

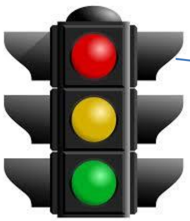
...

4

False Positive Problem

5

100
categories



...

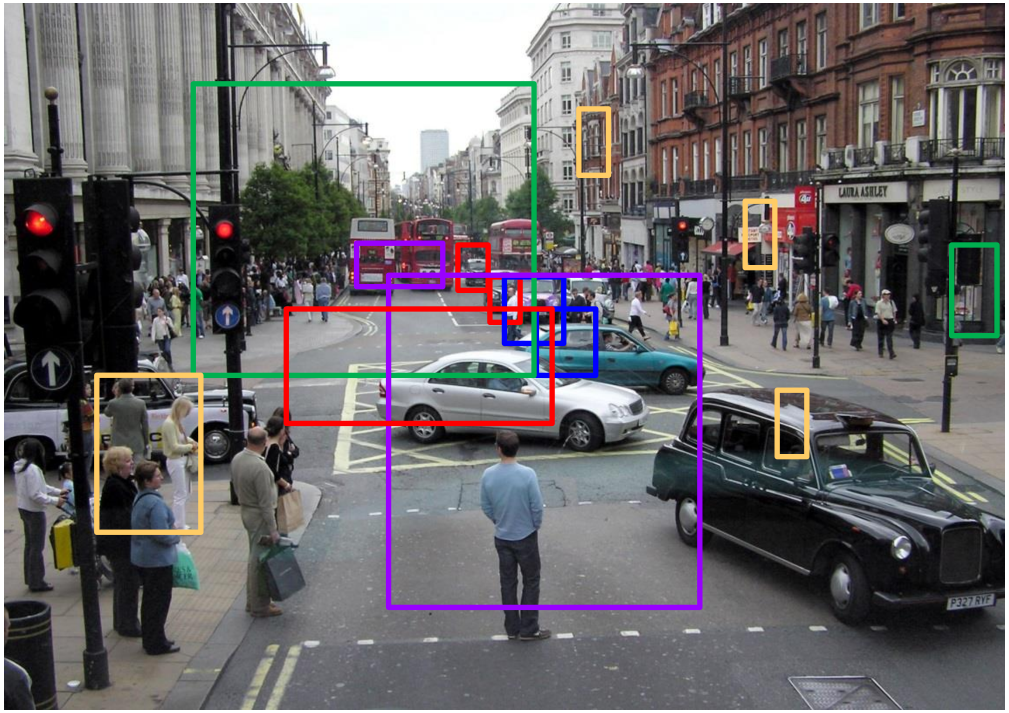
False Positive Problem

…

1000
categories

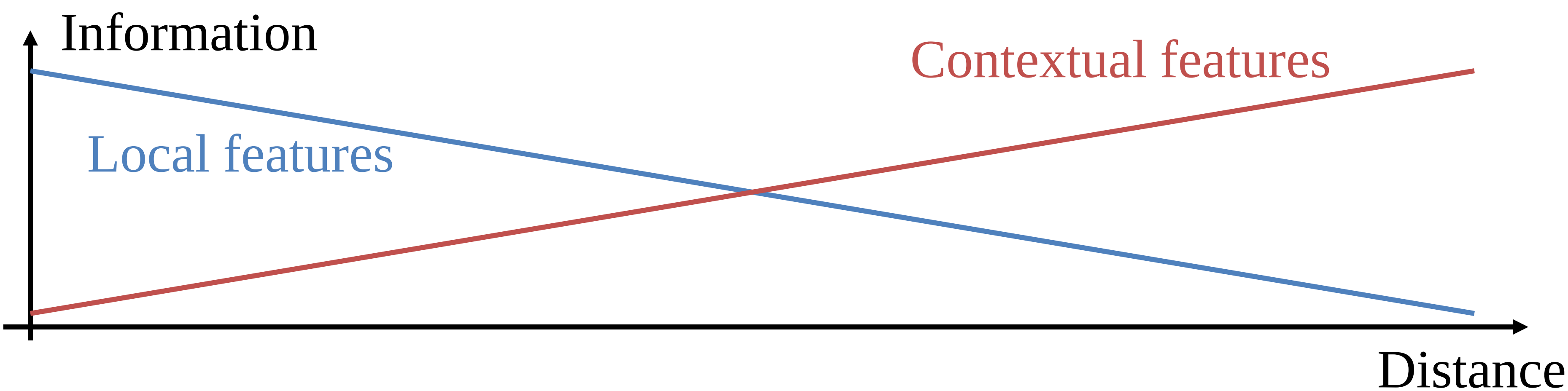
One class -> 1 f.a. every 10 images

1000 classes -> 100 f.a. every image



And it’s slow

Is local information even enough?



Slide credit: A. Torralba MIT

The system does not care about the scene, but we do…

We know there is a keyboard present in this scene even if we cannot see it clearly.

We know there is no keyboard present in this scene

… even if there is one indeed.

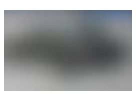
Slide credit: A. Torralba MIT

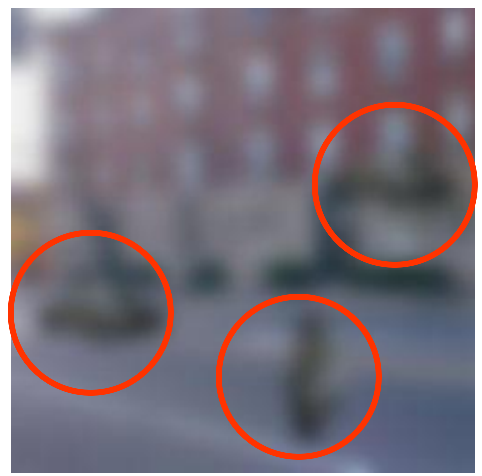
The multiple personalities of a blob




Slide credit: A. Torralba MIT

The multiple personalities of a blob





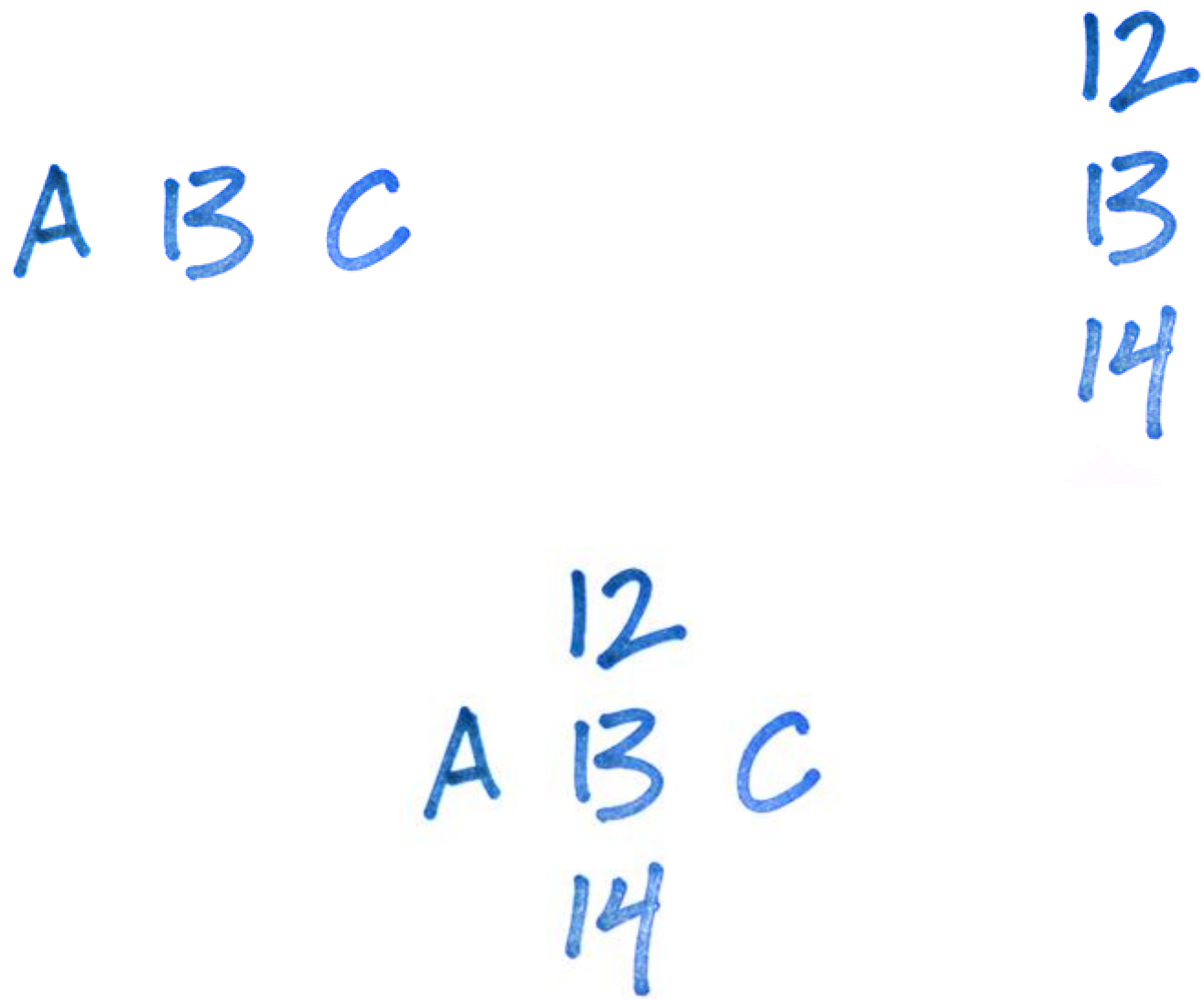
Slide credit: A. Torralba MIT

A B C

Slide credit: A. Torralba MIT

12
13
14

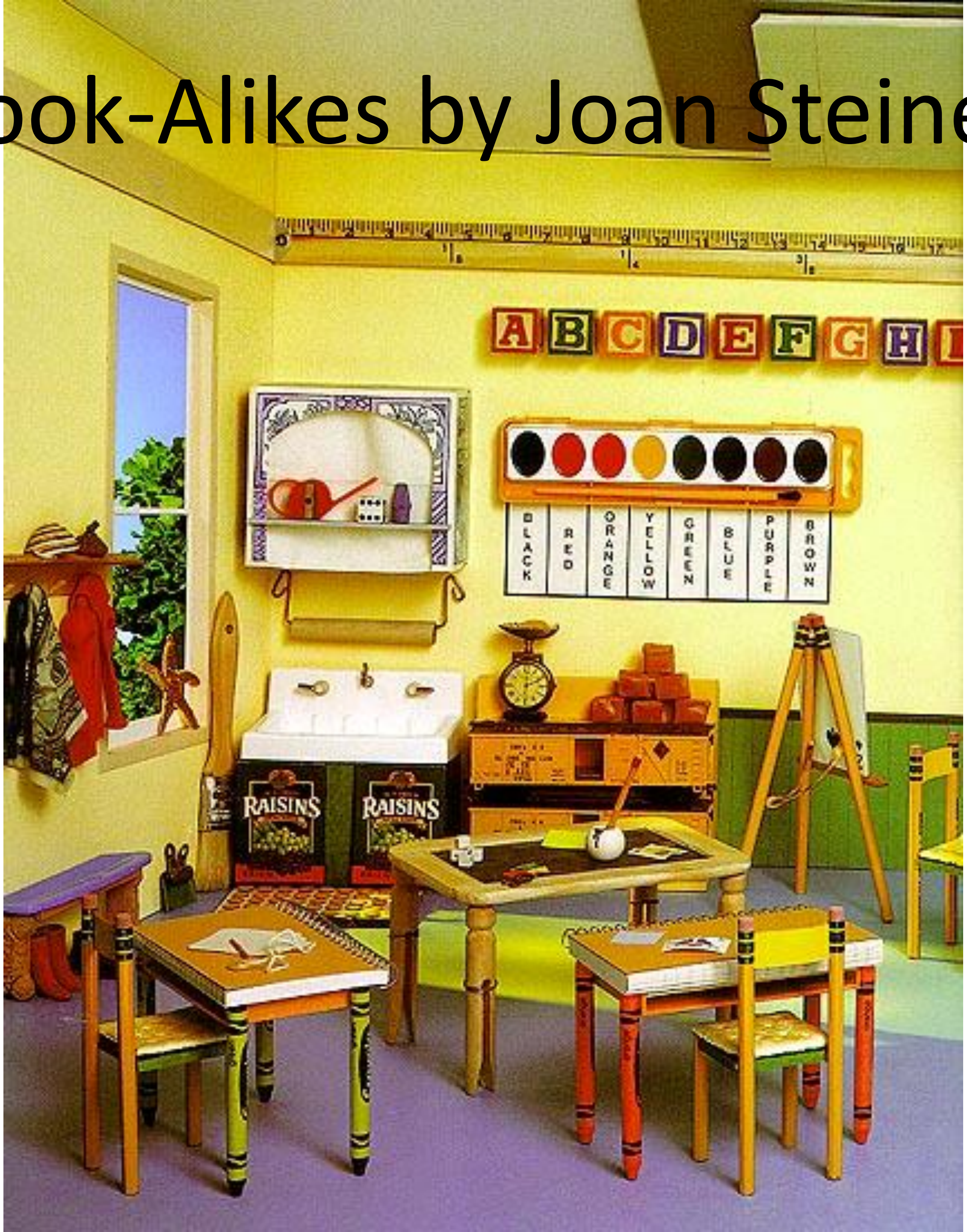
Slide credit: A. Torralba MIT



Slide credit: A. Torralba MIT

Look-Alikes by Joan Steiner

Look-Alikes by Joan Steiner



Look-Alikes by Joan Steiner

The context challenge

How far can you go without using an object detector?

Slide credit: A. Torralba MIT

What are the hidden objects?

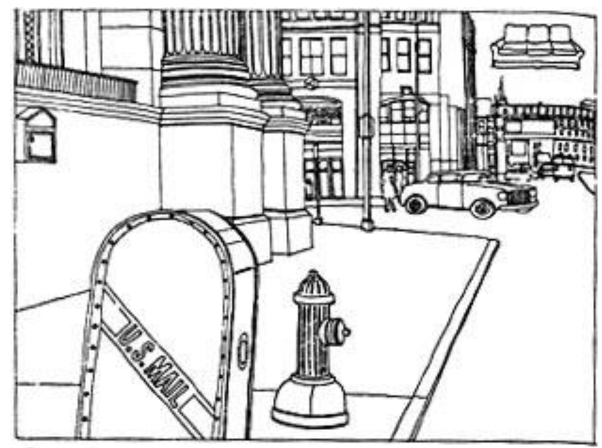
Slide credit: A. Torralba MIT

What are the hidden objects?

Slide credit: A. Torralba MIT

Biederman 1982

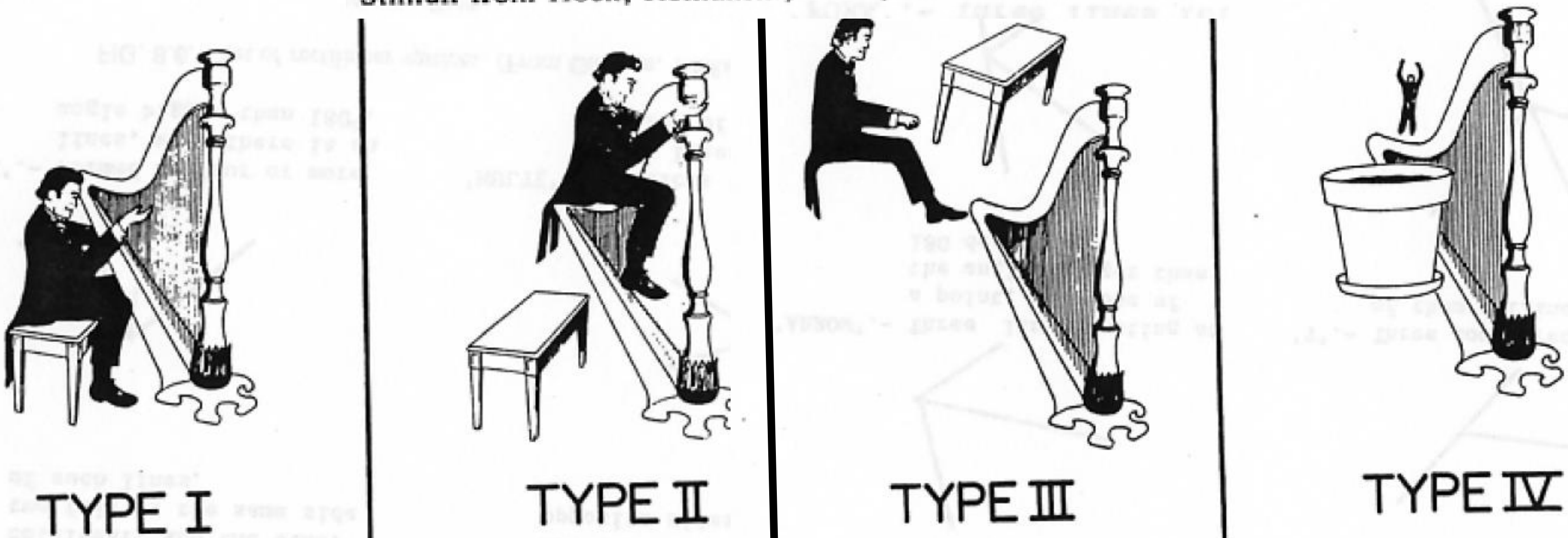
- Pictures shown for 150 ms.
- Objects in appropriate context were detected more accurately than objects in an inappropriate context.
- Scene consistency affects object detection.



Slide credit: A. Torralba MIT

Objects and Scenes

Stimuli from Hock, Romanski, Galie, and Williams (1978).



Biederman's violations (1981):

1. *Support* (e.g., a floating fire hydrant). The object does not appear to be resting on a surface.
2. *Interposition* (e.g., the background appearing through the hydrant). The objects undergoing this violation appear to be transparent or passing through another object.
3. *Probability* (e.g., the hydrant in a kitchen). The object is unlikely to appear in the scene.
4. *Position* (e.g., the fire hydrant on top of a mailbox in a street scene). The object is likely to occur in that scene, but it is unlikely to be in that particular position.
5. *Size* (e.g., the fire hydrant appearing larger than a building). The object appears to be too large or too small relative to the other objects in the scene.

Object priming

Increasing contextual information

Torralba, Sinha, Oliva, VSS 2001

The layered structure of scenes

Assuming a human observer standing on the ground

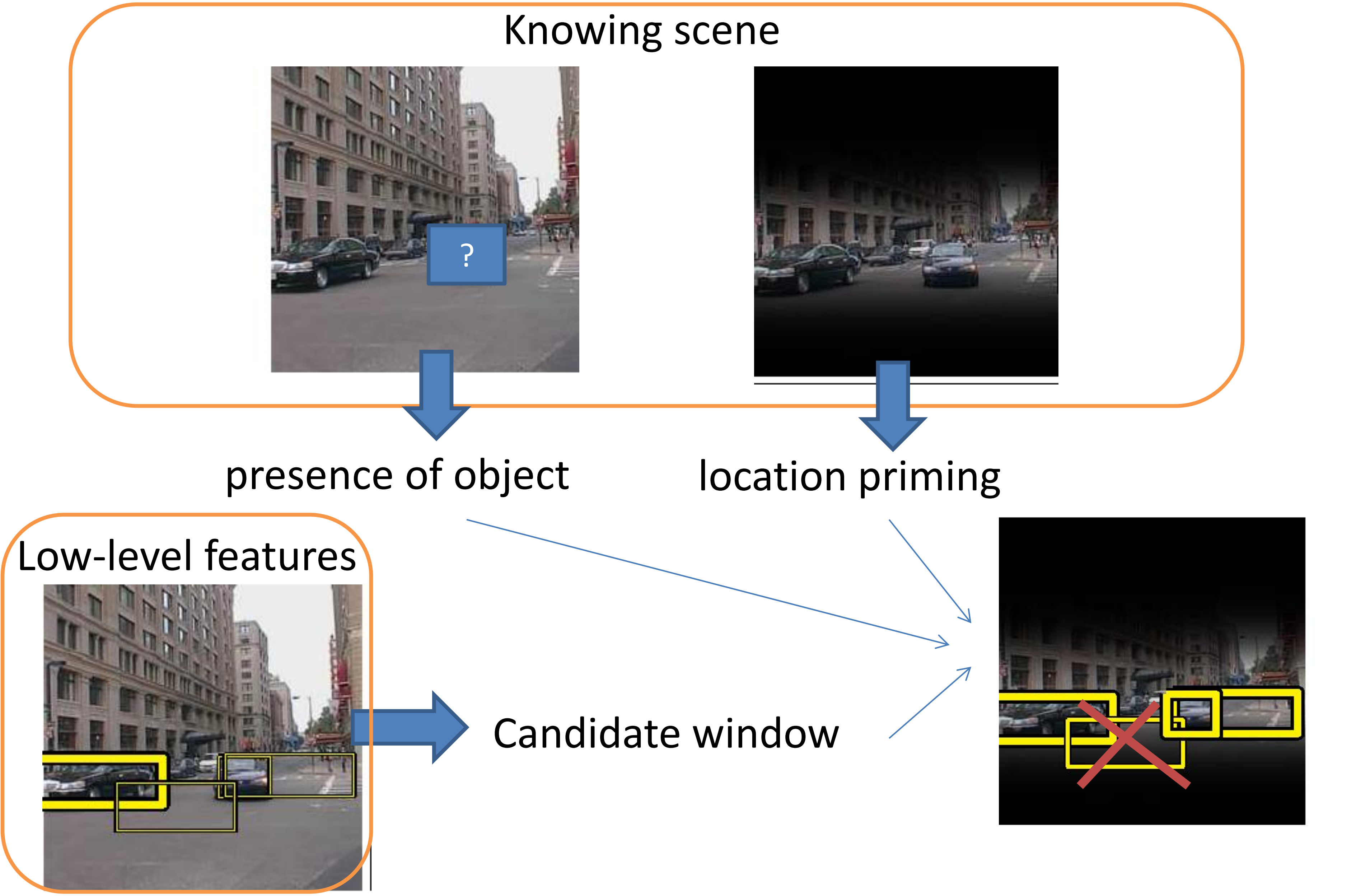
In a display with multiple targets present, the location of one target constraints the ‘y’ coordinate of the remaining targets, but not the ‘x’ coordinate.

Slide credit: A. Torralba MIT

Detecting faces without a face detector

Torralba & Sinha, 01; Torralba, 03

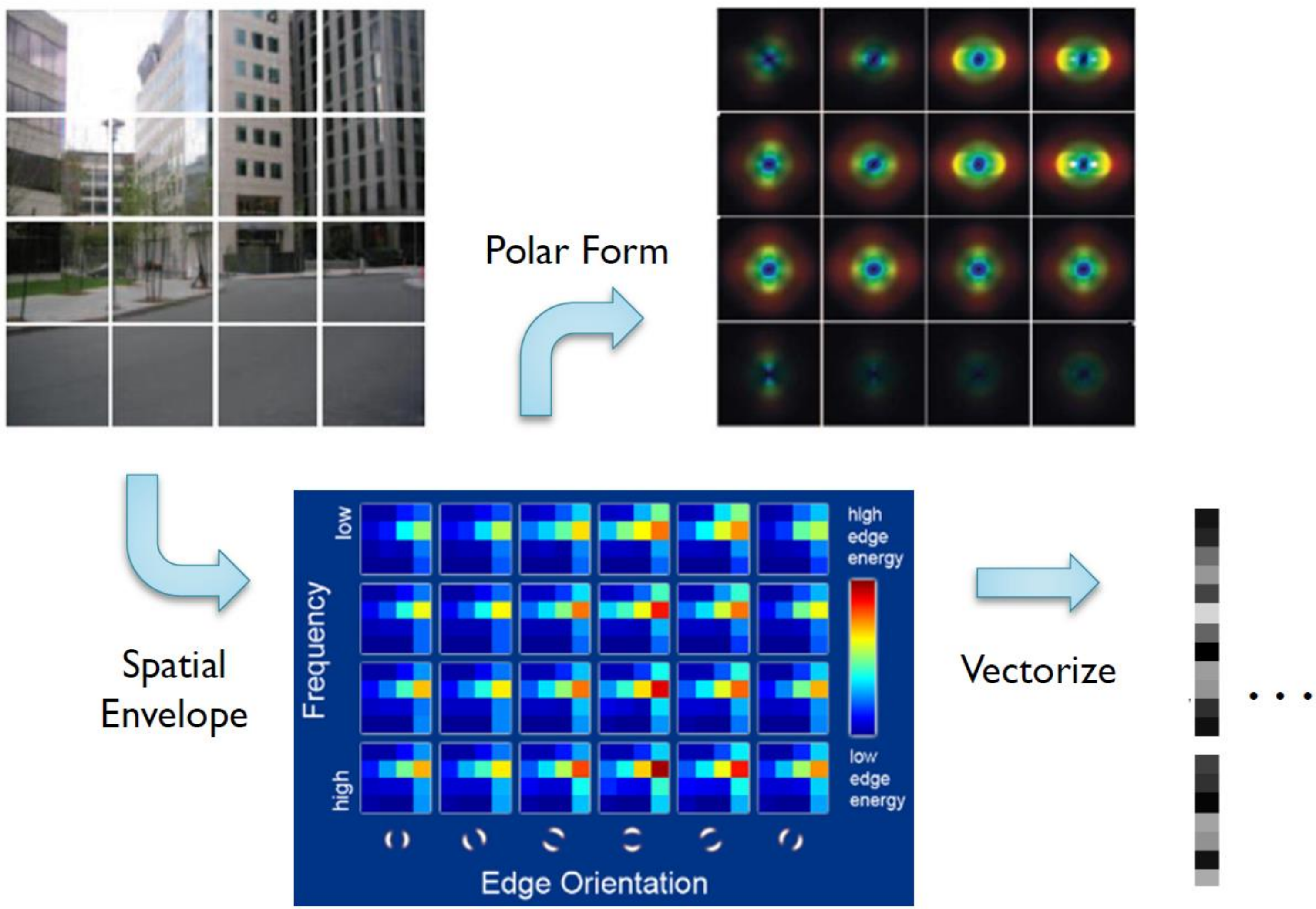
General Approach



GIST

- Global descriptor for scenes
- Based on statistics of low-level features over fixed image patches

GIST


Polar Form
Spatial Envelope
Frequency
low
high
high edge energy
low edge energy
Edge Orientation
Vectorize
. . .

GIST

Masks: divide the image in a grid of 4×4 non-overlapping windows

Luminance channel of the image

Gabor filters (6 orientations and 4 scales)

$$g_k = \sum_{x,y} w_k(x,y) \times |I(x,y) \otimes h_k(x,y)|^2$$

Pixel wise multiplication

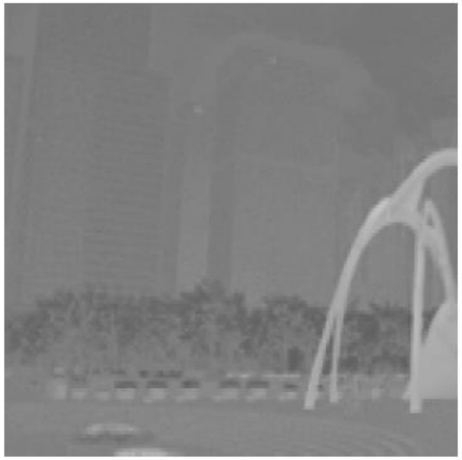
convolution

$$g = \begin{pmatrix} g_1 \\ g_2 \\ \\ g_m \end{pmatrix}$$

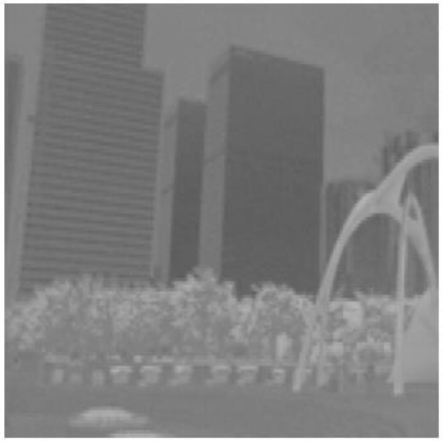
R G B space -> L*a*b*

Lab

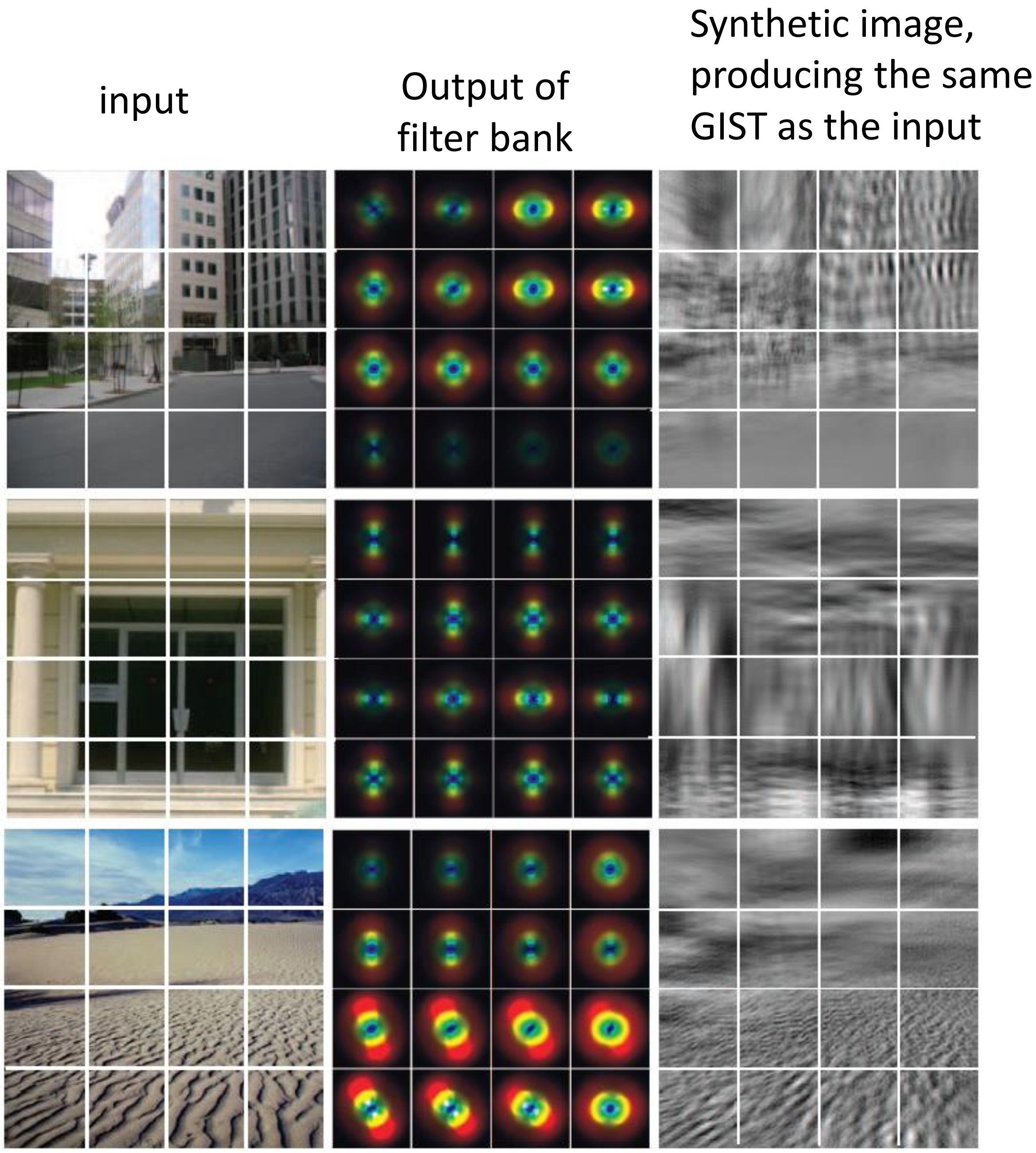
Luminance



a (red - green)



b (yellow - blue)


input
Output of filter bank
Synthetic image, producing the same GIST as the input

Target Problems

- Object presence/absence

  (generalization: the number of object instances in the scene)

- Object localization
  - specify the location and size of each of the object instances

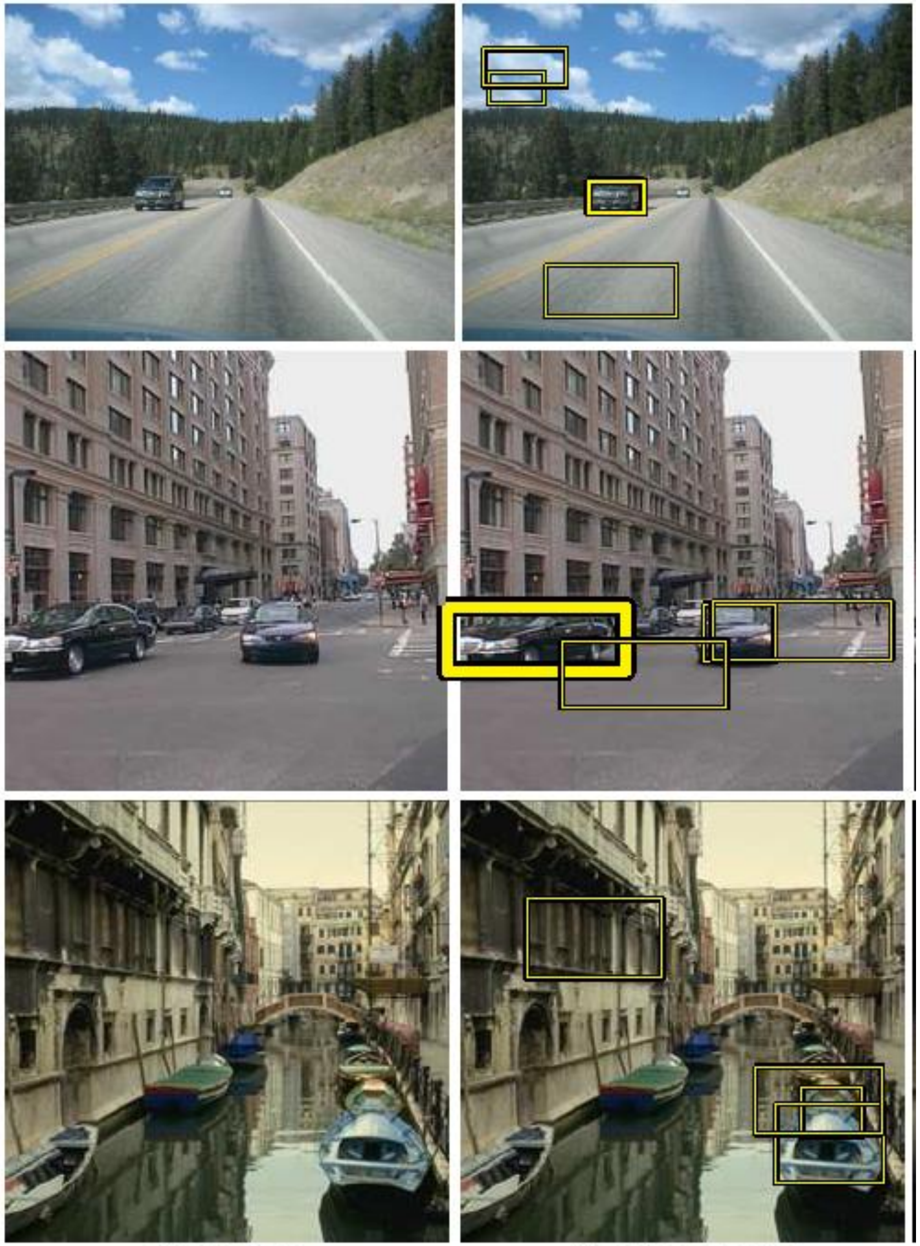
Localization using low-level features

binary random variable representing whether image patch *i* contains an object of type *t* or not(i=1:N, N≈100, number of patches)

Local image features, extracted from patch i of an image from class t

$$c_i^t = \log p(f_i^t | O_i^t = 1) / p(f_i^t | O_i^t = 0),$$

Compute this for each location i and object type t and output patches that score best (10 best scores)



a) input image b) car detector output

Car Detection (low-level features)

Presence using high level features

- Step1: Classify the scene from gist
- Step 2: Use scene label to predict the number of objects present

Number of instances

scene

GIST

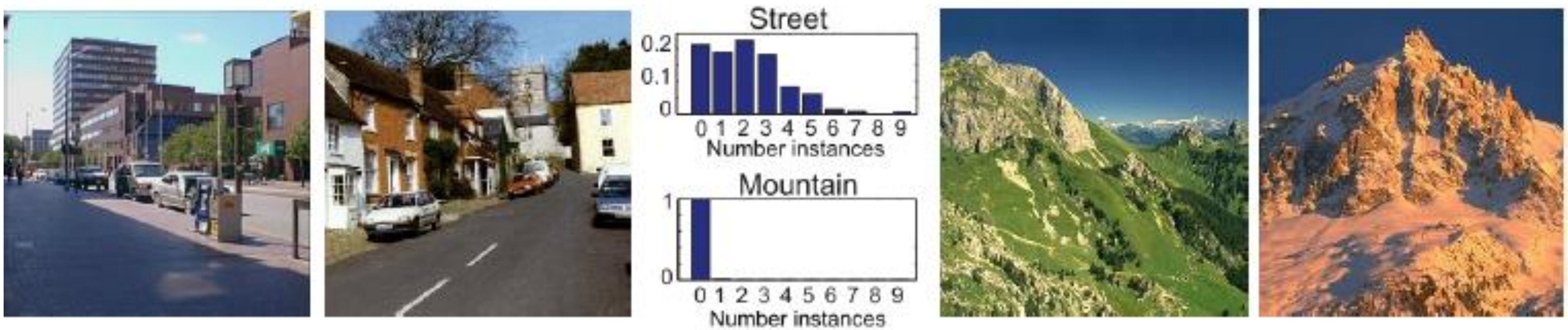
$$p(N^t = n|g) = \sum_s \boxed{p(N^t = n|S = s)}\, p(S = s|g)$$

Done in step one

Estimated by simple counting

Probability of n objects, given scene

Number of cars in a scene



Location Priming



- Learn the mapping from gist to vertical location (mixture of expert model)
- Predict most likely vertical location
- Mask out unlikely regions for class category

Integrated Model

- Combine global and local cues:
  - scores from the localization using local features
  - probability of n objects, given scene
  - location priming

Presence

- Find the number of objects present using gist
- Show that many confidence scores

$$p(O^t_{1:D} | c^t_{1:D}, g) \propto \underbrace{\boxed{p(O^t_{1:D} | g)}}_{\text{Presence of object given gist}} \prod_{i=1}^{D} \underbrace{\boxed{p(c^t_i | O^t_i)}}_{\text{Confidence scores, given presence of object}}$$

$$p(O^t_{1:D} | g) \propto \sum_{n=0}^{D} \underbrace{p(O^t_{1:D} | n)}_{=1 \text{ only if } O_{1:D} \text{ has precisely } n \text{ bits on}} p(N^t = n | g)$$

Adding Location Information

- Let l_i^t indicate the location of the top i*th* (i=1..D) detection of class t.
- Combine expected location and presence as follows:

$$p(O_{1:D}^t | c_{1:D}^t, \ell_{1:D}^t, g) \propto p(O_{1:D}^t | g) \prod_{i=1}^{D} p(c_i^t | O_i^t) \boxed{p(\ell_i^t | O_i^t, g)}$$

$$p(\ell_i^t | O_i^t, g) = \int \boxed{p(\ell_i^t | O_i^t, Y_t)} \boxed{p(Y_t | g)} dY_t$$

Y_t is an expected location of class t

Likelihood of the location of patch i for class t, given it's expected location.

Location priming

Adding Location Information

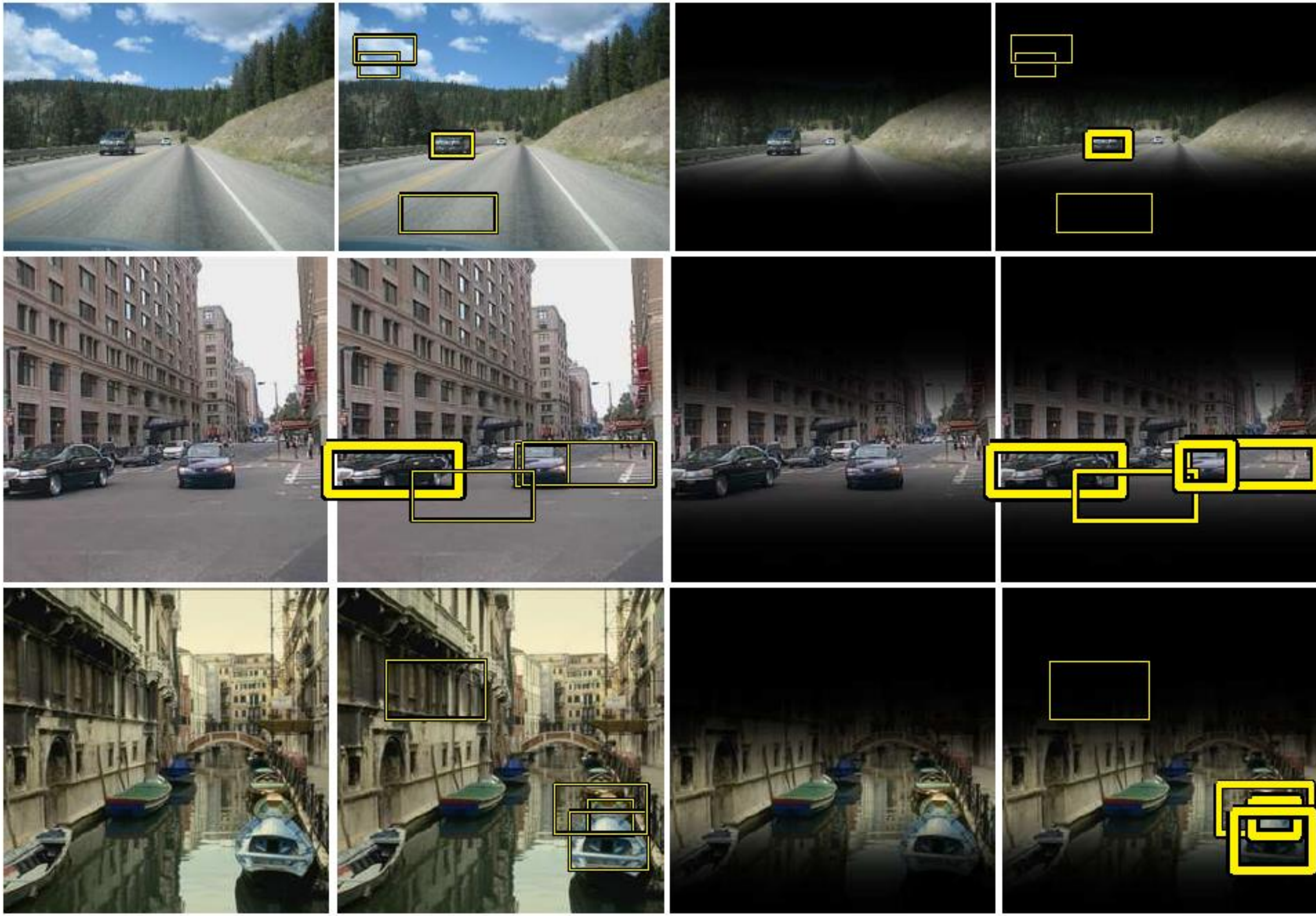
$$p(O^t_{1:D} | c^t_{1:D}, \ell^t_{1:D}, g) \propto \underline{p(O^t_{1:D} | g) \prod_{i=1}^{D} p(c^t_i | O^t_i)}\, \underline{p(\ell^t_i | O^t_i, g)}$$

- confident detections in improbable locations are suppressed;
- unconfident detections in likely locations are boosted.

Results

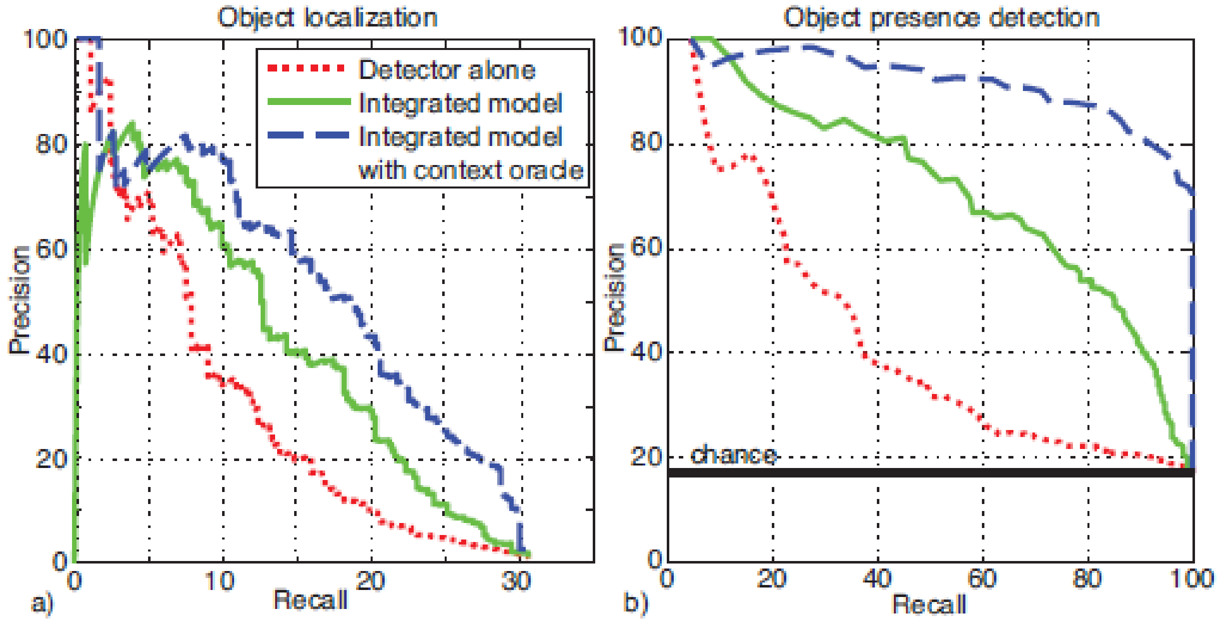
- 2688 images with 8 scenes
  - half for training, half for testing
- Focused solely on car identification
- Integrated model is better than local features

Slide credit: Aron Yu



a) input image b) car detector output c) location priming c) integrated model output

Results



- Improves precision but not recall
  - If the detector misses an instance (due to occlusion or noise), context doesn't help
  - Reduces the number of false positives, removes cars in scenes where cars are not expected

Evaluation - Strength

- Probabilistic information fusion
- Boost confidence of probable regions suppress confidence of non-probable regions
- Location priming makes intuitive sense
- Better performance than with only local features

Slide credit: Aron Yu

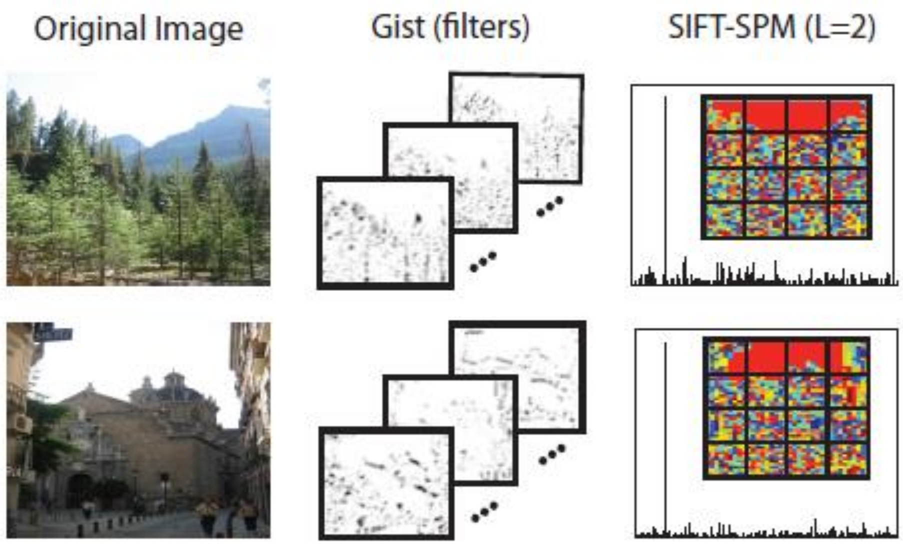
Evaluation - Weakness

- Tested on a single object (cars)
- Boosts false positives within probable regions
- Relies heavily on object detector accuracy
- Suppresses true positives within non-probable regions

Object Bank: A High-Level Image Representation for Scene Classification & Semantic Feature Sparsification

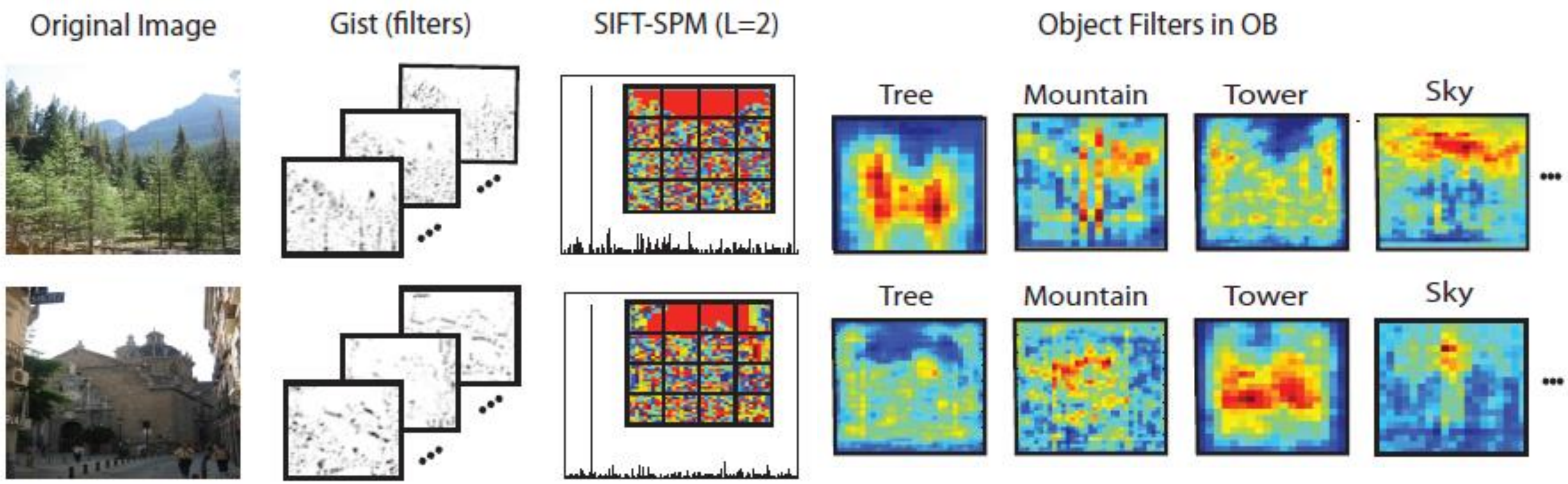
CATEGORIZATION

Motivation



Different images have similar statistics of two most popular features

Motivation



Different images have similar statistics of two most popular features

Responses of object detectors are more specific

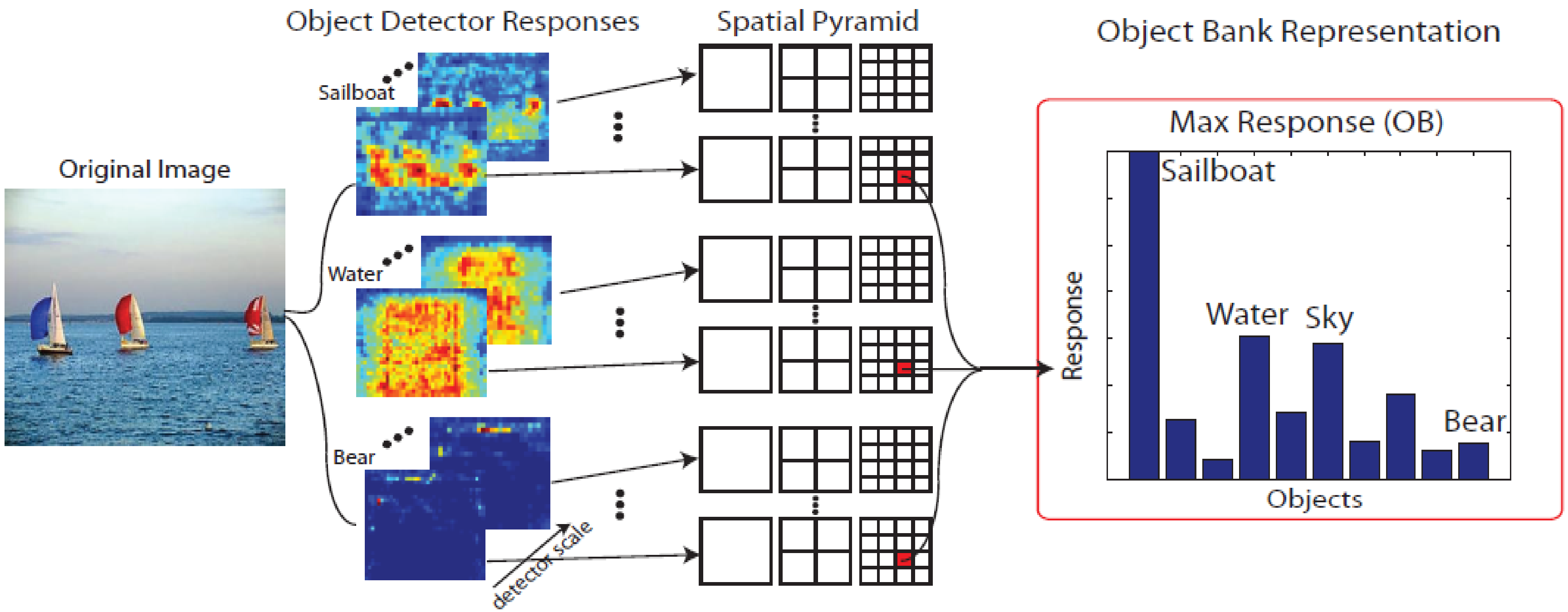
Motivation

- Before: quantized local features as words
- New: use objects as words.
- Simple Motivation: scene consists of objects

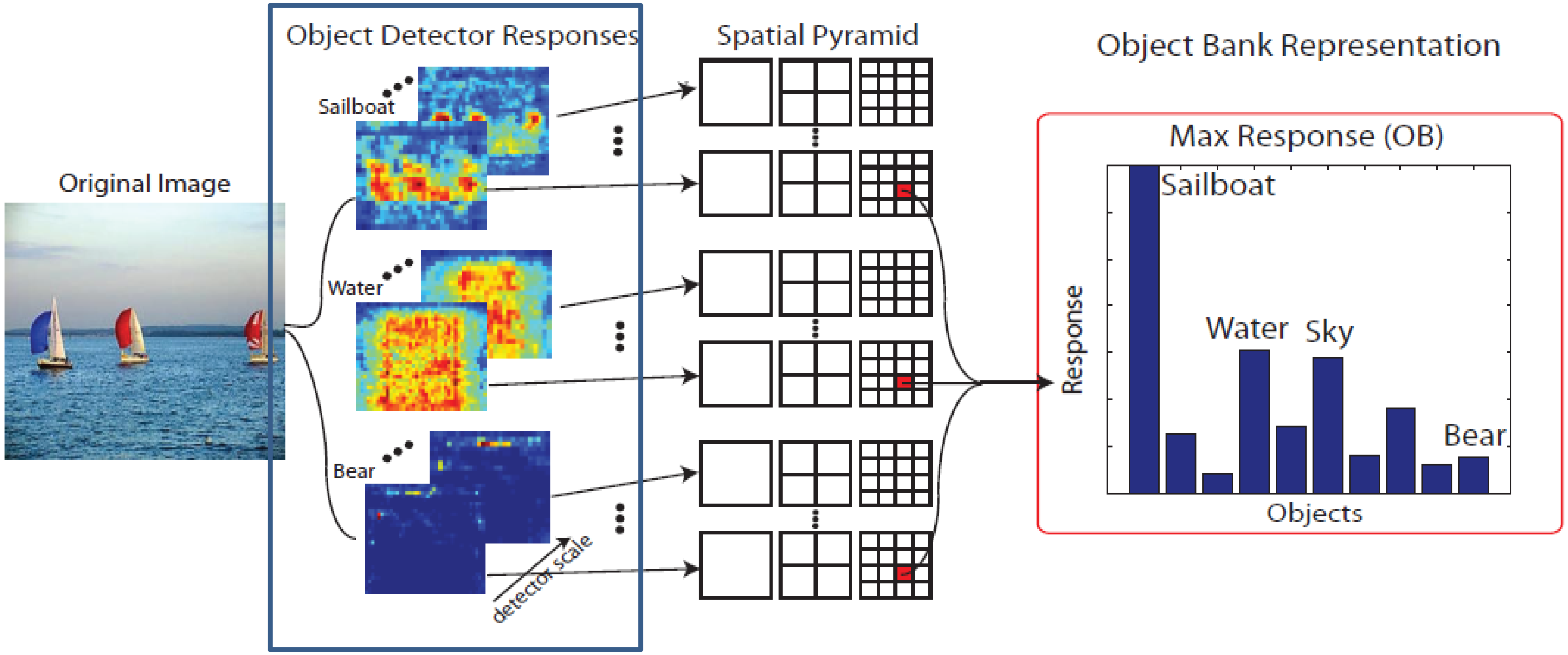
Object Bank

- representation of natural images
- based on objects
  - a collection of object sensing filters built on a generic collection of labeled objects

Object Bank


Object Detector Responses
Spatial Pyramid
Object Bank Representation
Original Image
Sailboat
Water
Bear
detector scale
Max Response (OB)
Sailboat
Water
Sky
Bear
Response
Objects

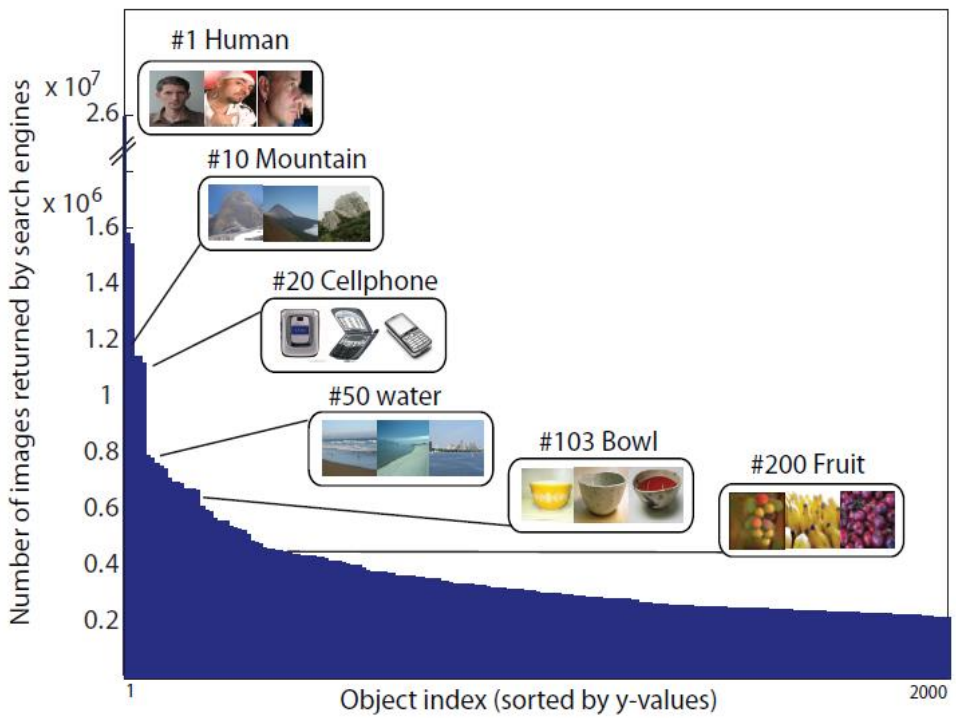
Object Bank



- Use latent SVM detector for blobby objects: tables, cars, humans, etc.
- Use texture classifier by Hoiem for more texture- and material-based objects: sky, road, sand, etc.
- 200 object detectors at 12 detection scales

How many objects to use?

- All (tens of thousands of generic objects)
  - Computationally infeasible
  - Some object are more important than others:



Choose few hundred most popular objects in images.

How to choose objects for OB?

- Enough training images for each object detectors
- Dataset: ESP , LabelMe, ImageNet , and the Flickr.
- Take the intersection set of the most frequent 1000 objects, resulting in 200 objects
- Training and validation are done on different sets.

Object Bank Representation

- For each object at each scale, a three-level spatial pyramid representation of the resulting object filter map is used
  - No. of objects x No. of scales x(1x4x16) grids
- The maximum response for each object in each grid is then computed
  - No. of objects representation for each grid
- A concatenation of features in all grids leads to an OB descriptor for the image.

Learning Scene Classifier

- Stacking filter outputs of all object detectors => very large dimension => overfitting
- Use strong regularizers

$$min_{\beta \in \mathbb{R}^J} \underbrace{\lambda R(\beta)}_{\text{regularizer}} + \frac{1}{m} \sum_{i=1}^{m} \underbrace{L(\beta; x_i, y_i)}_{\text{Loss function}}$$

Loss function

Logistic loss:

$$L = \log(1/P(y_i|\mathbf{x}_i, \beta))$$

$$P(y|\mathbf{x}, \beta)) = \frac{1}{Z}\exp(\frac{1}{2}y(\mathbf{x} \cdot \beta))$$

Regularizer

- L2 $R(\beta) \triangleq ||\beta||_2$ $\boxed{\text{LR}}$
- L1 $R(\beta) \triangleq ||\beta||_1$ Feature sparsity, many of βi=0 $\boxed{\text{LR1}}$
- L1/2 (group regularizer) $\boxed{\text{LRG}}$

  $$R(\beta) \triangleq ||\beta||_{1,2} = \sum_{j=1}^{J} ||\beta^j||_2,$$

  where β^j is the j-th group (i.e., features grouped by an object j)

  Object level sparsity – all features of object go to zero

- L1/L2+L1 joint object feature sparsity $\boxed{\text{LRG1}}$

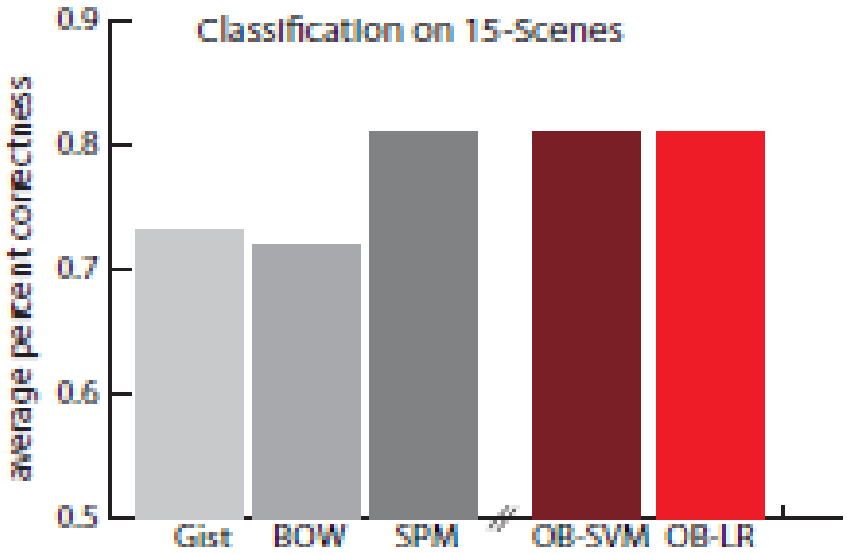
  $$R(\beta) \triangleq \lambda_1 ||\beta||_{1,2} + \lambda_2 ||\beta||_1$$

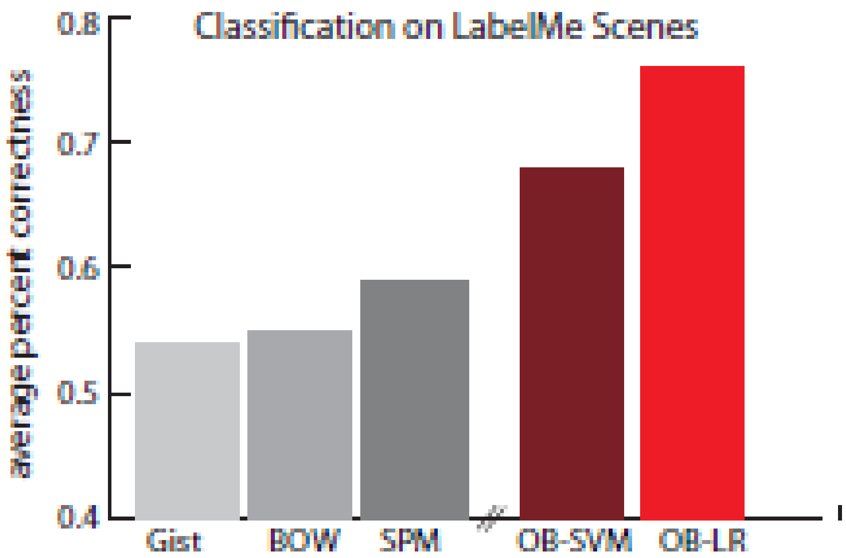
Results

	15-Scene	UIUC-Sports	MIT-Indoor
state-of-the-art	72.2%[19]	66.0% [32]	26% [27]
	81.1%[19]	73.4% [22]	
OB	80.9%	76.3%	37.6%

Degree of clutter

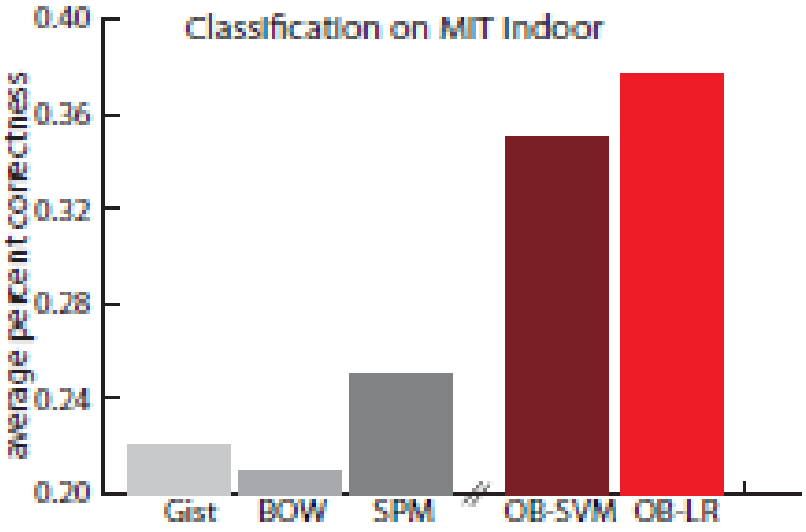
Results

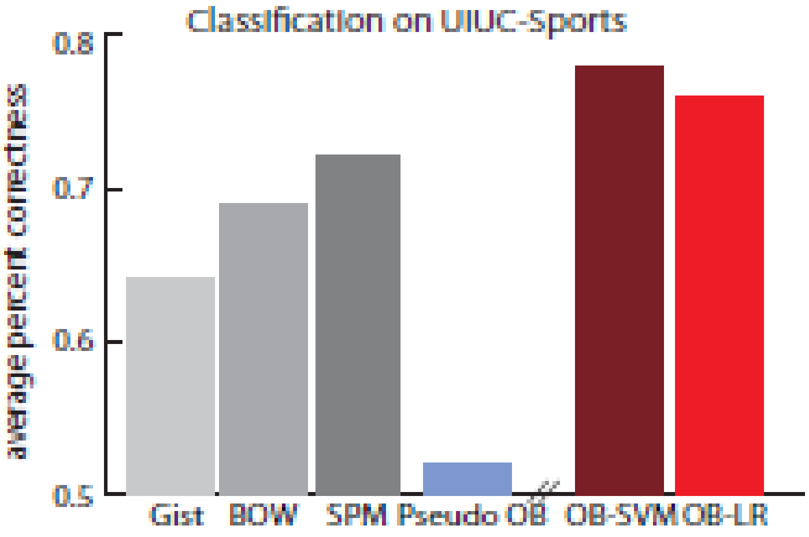




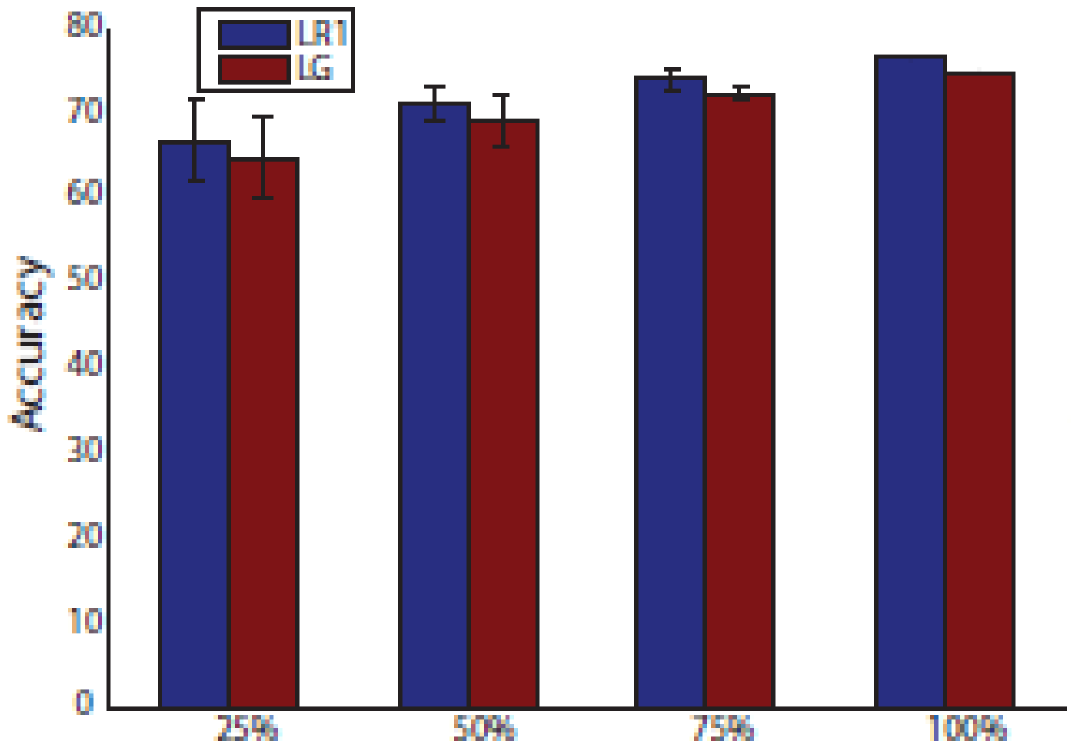
GIST, BOW and SPM are trained with SVM

Results


Classification on MIT Indoor
average percent correctness
0.40
0.36
0.32
0.28
0.24
0.20
Gist
BOW
SPM
OB-SVM
OB-LR

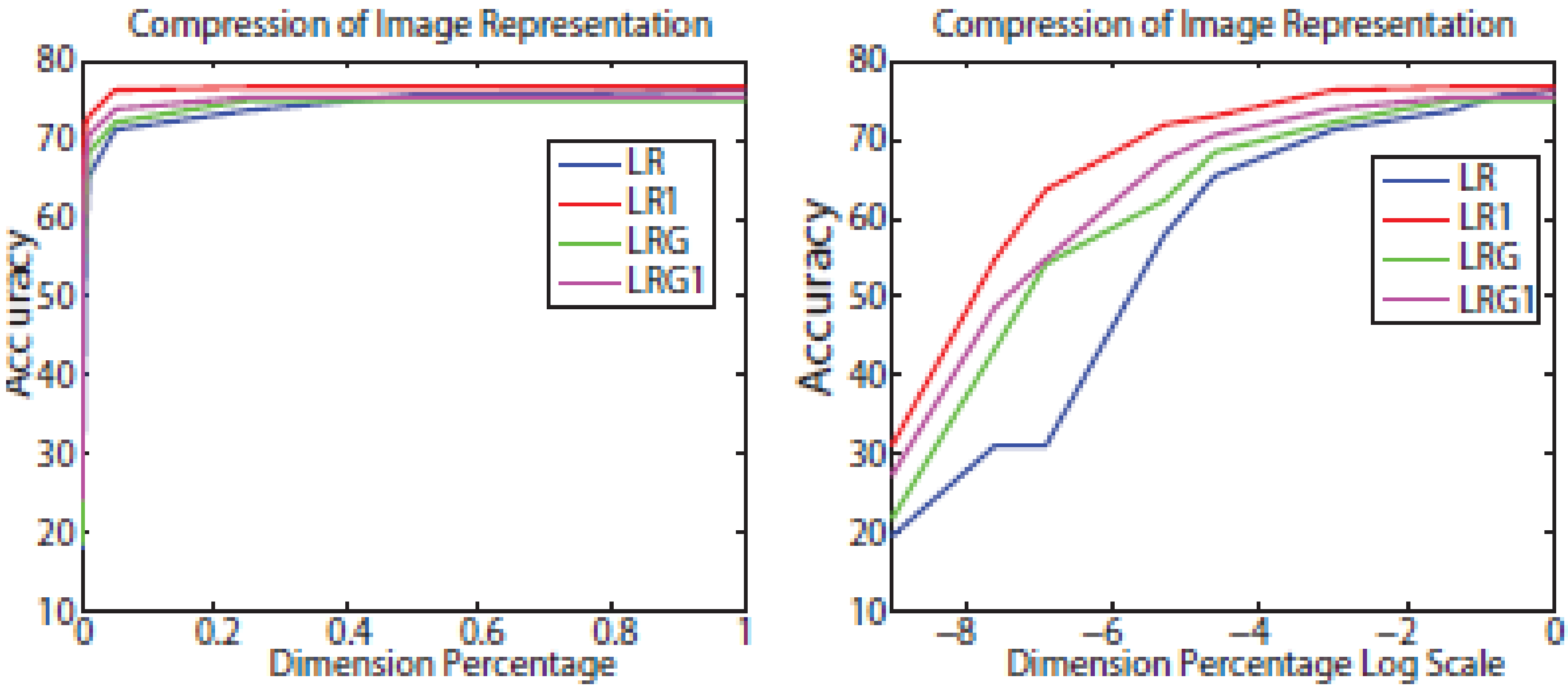

Classification on UIUC-Sports
average percent correctness
0.8
0.7
0.6
0.5
Gist
BOW
SPM
Pseudo OB
OB-SVM
OB-LR

Accuracy vs. number of examples



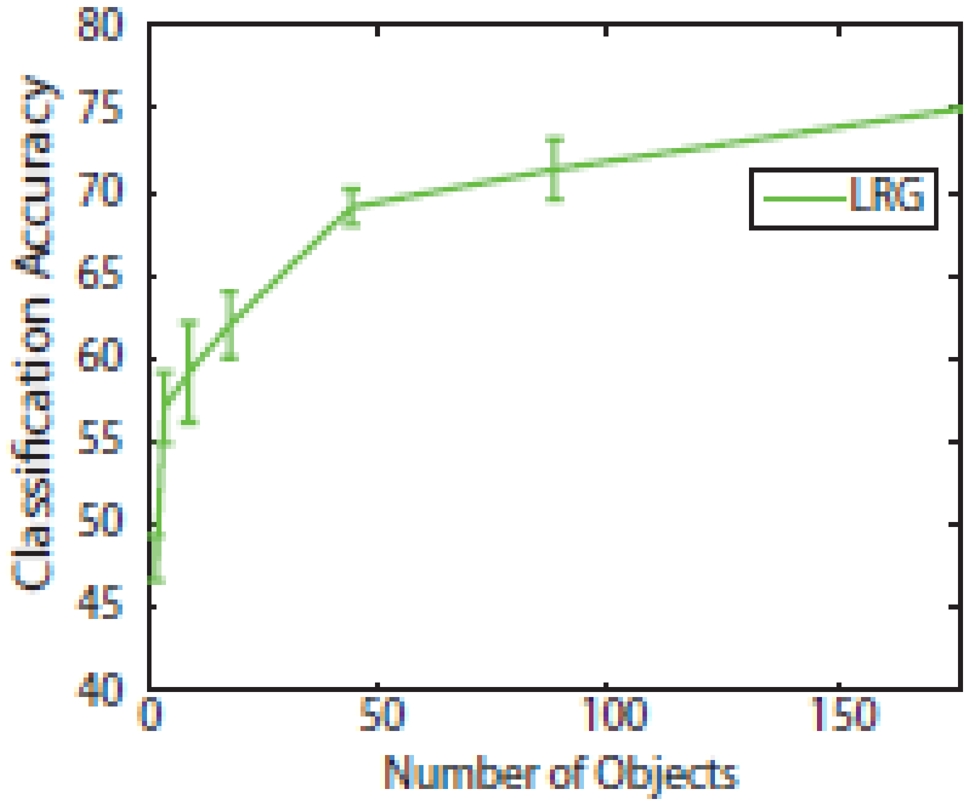
Conclusion: OB representation requires less training examples per scene.

Accuracy vs. percentage of features used



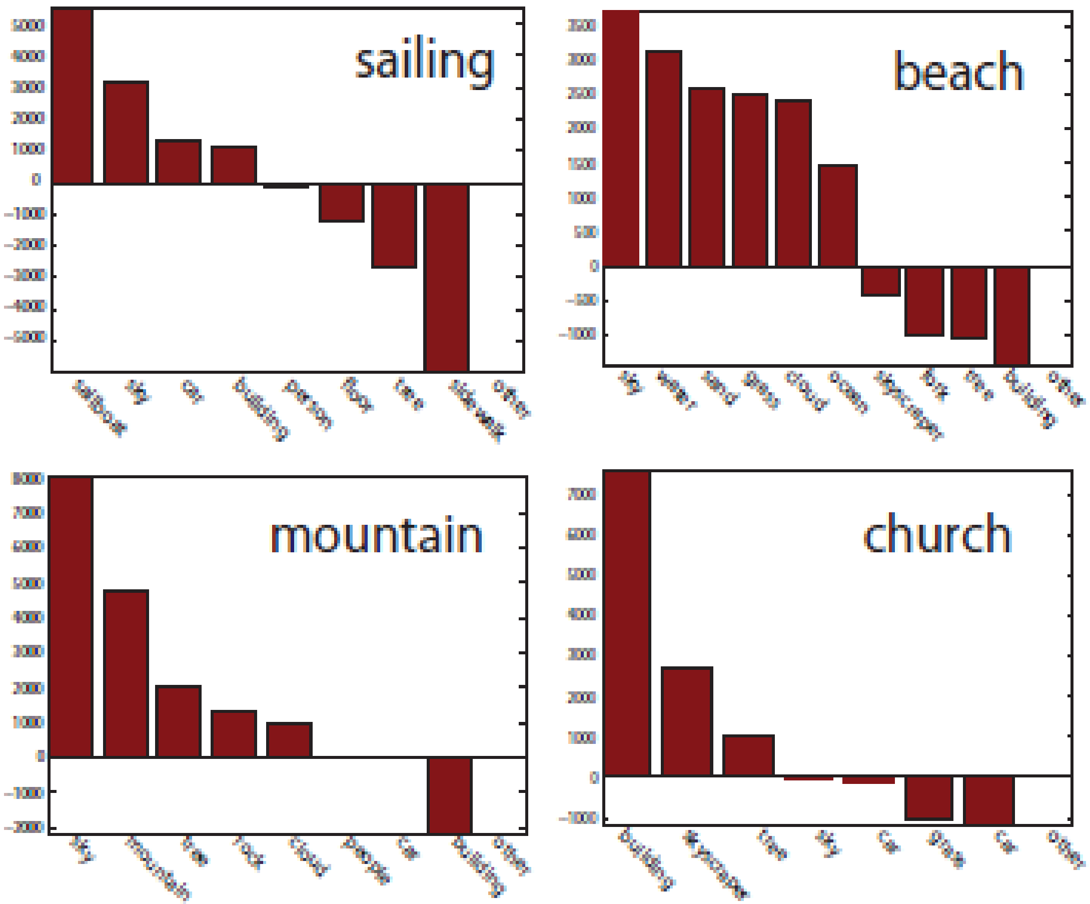
Conclusion: OB representation allows sparser representations. OB is over-complete representation

Accuracy as a function of number of objects



Conclusion: OB representation improves when the number of objects increases.

Interpretation of the compressed representation



Object-wise coefficients given scene class. Selected objects correspond to non-zero values learned by LRG.

Conclusions

- Object Bank representation is powerful on scene classification tasks
  - it carries rich semantic level image information
  - Allows to achieve nearly lossless semantic-preserving compression