Using Context in Object Recognition

OBJECT LOCALIZATION
Sliding Window Approach

Positive

Feature extraction

Learning classifier f(x)

Positive

Negative

Sliding window
False Positive Problem

100 categories

...
False Positive Problem

100 categories

1
False Positive Problem

100 categories
False Positive Problem

100 categories

...
False Positive Problem

100 categories

...
False Positive Problem

100 categories

...
False Positive Problem

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

1000 classes -> 100 f.a. every image

And it’s slow
Is local information even enough?

- **Local features**
- **Contextual features**

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
Slide credit: A. Torralba MIT
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
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

Knowing scene

presence of object

location priming

Low-level features

Candidate window
GIST

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

Spatial Envelope → Polar Form → Vectorize
GIST

Masks: divide the image in a grid of $4 \times 4$ non-overlapping windows

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

Luminance channel of the image

Gabor filters (6 orientations and 4 scales)

Pixel wise multiplication

convolution

$$g = \begin{pmatrix} g_1 \\ g_2 \\ g_m \end{pmatrix}$$
RGB space -> L*a*b*

Lab

Luminance

a (red - green)

b (yellow - blue)
<table>
<thead>
<tr>
<th>input</th>
<th>Output of filter bank</th>
<th>Synthetic image, producing the same GIST as the input</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Input Image" /></td>
<td><img src="image2.png" alt="Output Image" /></td>
<td><img src="image3.png" alt="Synthetic Image" /></td>
</tr>
<tr>
<td><img src="image4.png" alt="Input Image" /></td>
<td><img src="image5.png" alt="Output Image" /></td>
<td><img src="image6.png" alt="Synthetic Image" /></td>
</tr>
<tr>
<td><img src="image7.png" alt="Input Image" /></td>
<td><img src="image8.png" alt="Output Image" /></td>
<td><img src="image9.png" alt="Synthetic Image" /></td>
</tr>
<tr>
<td><img src="image10.png" alt="Input Image" /></td>
<td><img src="image11.png" alt="Output Image" /></td>
<td><img src="image12.png" alt="Synthetic Image" /></td>
</tr>
<tr>
<td><img src="image13.png" alt="Input Image" /></td>
<td><img src="image14.png" alt="Output Image" /></td>
<td><img src="image15.png" alt="Synthetic Image" /></td>
</tr>
</tbody>
</table>
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 ($i=1:N$, $N\approx 100$, 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)
Car Detection (low-level features)
Presence using high level features

- Step 1: Classify the scene from gist
- Step 2: Use scene label to predict the number of objects present

\[ p(N^t = n | g) = \sum_s p(N^t = n | S = s)p(S = s | g) \]

Number of instances

scene

GIST

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

\[
p(O_{1:D}^t | c_{1:D}^t, g) \propto p(O_{1:D}^t | g) \prod_{i=1}^{D} p(c_i^t | O_i^t)
\]

Presence of object given gist

\[
p(O_{1:D}^t | g) \propto \sum_{n=0}^{D} p(O_{1:D}^t | n)p(N^t = n | g)
\]

Confidence scores, given presence of object

=1 only if \(O_{1:D}\) has precisely \(n\) bits on
Adding Location Information

• Let $l^t_i$ indicate the location of the top $ith$ ($i=1..D$) detection of class $t$.

• Combine expected location and presence as follows:

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

$$p(\ell^t_i | O^t_i, g) = \int p(\ell^t_i | O^t_i, Y_t) p(Y_t | g) dY_t$$

$Y_t$ 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

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

- 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
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
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

Objective Image

Object Detector Responses

Spatial Pyramid

Object Bank Representation

Max Response (OB)

Response

Objects

Sailboat

Water

Sky

Bear
• 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

\[
min_{\beta \in \mathbb{R}^j} \lambda R(\beta) + \frac{1}{m} \sum_{i=1}^{m} L(\beta; x_i, y_i)
\]

- regularizer
- Loss function
Loss function

Logistic loss:

\[ L = \log \left( \frac{1}{P(y_i | x_i, \beta)} \right) \]

\[ P(y | x, \beta) = \frac{1}{Z} \exp \left( \frac{1}{2} y(x \cdot \beta) \right) \]
Regularizer

- **L2**  \( R(\beta) \triangleq \|\beta\|_2 \)  
- **L1**  \( R(\beta) \triangleq \|\beta\|_1 \)  
- **L1/2 (group regularizer)**  
  \[ R(\beta) \triangleq \|\beta\|_{1,2} = \sum_{j=1}^{J} \|\beta^j\|_2, \]
  where \( \beta^j \) 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**  
  \[ R(\beta) \triangleq \lambda_1\|\beta\|_{1,2} + \lambda_2\|\beta\|_1. \]
## Results

<table>
<thead>
<tr>
<th></th>
<th>15-Scene</th>
<th>UIUC-Sports</th>
<th>MIT-Indoor</th>
</tr>
</thead>
<tbody>
<tr>
<td>state-of-the-art</td>
<td>72.2% [19]</td>
<td>66.0% [32]</td>
<td>26% [27]</td>
</tr>
<tr>
<td>OB</td>
<td>81.1% [19]</td>
<td>73.4% [22]</td>
<td>37.6%</td>
</tr>
</tbody>
</table>

**Degree of clutter**
Results

GIST, BOW and SPM are trained with SVM
Results

Classification on MIT Indoor

Classification on UIUC-Sports
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