Large-scale category recognition and Advanced feature encoding

Computer Vision
James Hays
Why do good recognition systems go bad?

• E.g. Why isn’t our Bag of Words classifier at 90% instead of 70%?

• Training Data
  – Huge issue, but not necessarily a variable you can manipulate.

• Representation
  – Are the local features themselves lossy?
  – What about feature quantization? That’s VERY lossy.

• Learning method
  – Probably not such a big issue, unless you’re learning the representation (e.g. deep learning).
CalTech 101 - 2004
The SUN Attribute Database: Beyond Categories for Deeper Scene Understanding.

Genevieve Patterson, Chen Xu, Hang Su, and James Hays.
SUN Database: Large-scale Scene Categorization and Detection

Jianxiong Xiao, James Hays†, Krista A. Ehinger, Aude Oliva, Antonio Torralba
Massachusetts Institute of Technology
† Brown University
Scene Categorization

Oliva and Torralba, 2001

Coast  Forest  Highway  Inside City  Mountain  Open Country  Street  Tall Building

Fei Fei and Perona, 2005

Bedroom  Kitchen  Living Room  Office  Suburb

Lazebnik, Schmid, and Ponce, 2006

Industrial  Store

15 Scene Database
15 Scene Recognition Rate
How many object categories are there?

~10,000 to 30,000

Biederman 1987
abbey
airplane cabin
apple orchard
assembly hall
bakery
construction site
food court
interior car
stadium
stream
train station
397 Well-sampled Categories
Evaluating Human Scene Classification

“Good worker”
Accuracy

98%  90%  68%
<table>
<thead>
<tr>
<th>Scene category</th>
<th>Most confusing categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inn (0%)</td>
<td>Restaurant patio (44%)</td>
</tr>
<tr>
<td>Bayou (0%)</td>
<td>River (67%)</td>
</tr>
<tr>
<td>Basilica (0%)</td>
<td>Cathedral (29%)</td>
</tr>
<tr>
<td></td>
<td>Chalet (19%)</td>
</tr>
<tr>
<td></td>
<td>Coast (8%)</td>
</tr>
<tr>
<td></td>
<td>Courthouse (21%)</td>
</tr>
</tbody>
</table>
Conclusion: humans can do it

• The SUN database is reasonably consistent and differentiable -- even with a huge number of very specific categories, humans get it right 2/3rds of the time with no training.

• We also have a good benchmark for computational methods.

How do we classify scenes?
How do we classify scenes?

Different objects, different spatial layout
Which are the important elements?

Similar objects, and similar spatial layout

Different lighting, different materials, different “stuff”
Scene emergent features

“Recognition via features that are not those of individual objects but “emerge” as objects are brought into relation to each other to form a scene.” – Biederman 81
Global Image Descriptors

- Tiny images  (Torralba et al, 2008)
- Color histograms
- Self-similarity  (Shechtman and Irani, 2007)
- Geometric class layout  (Hoiem et al, 2005)
- Geometry-specific histograms  (Lalonde et al, 2007)
- Dense and Sparse SIFT histograms
- Berkeley texton histograms  (Martin et al, 2001)
- HoG 2x2 spatial pyramids
- Gist scene descriptor  (Oliva and Torralba, 2008)
Global Texture Descriptors

Bag of words

Sivic et. al., ICCV 2005
Fei-Fei and Perona, CVPR 2005

Non localized textons

Walker, Malik. Vision Research 2004

Spatially organized textures

M. Gorkani, R. Picard, ICPR 1994
A. Oliva, A. Torralba, IJCV 2001

S. Lazebnik, et al, CVPR 2006

Gist descriptor

Oliva and Torralba, 2001

- Apply oriented Gabor filters over different scales
- Average filter energy in each bin

Similar to SIFT (Lowe 1999) applied to the entire image

Global scene descriptors

- The “gist” of a scene: Oliva & Torralba (2001)

http://people.csail.mit.edu/torralba/code/spatialenvelope/
Example visual gists

Global features (I) ~ global features (I’)

Oliva & Torralba (2001)
Textons

Vector of filter responses at each pixel

Kmeans over a set of vectors on a collection of images

Filter bank

Malik, Belongie, Shi, Leung, 1999
Textons

Filter bank

K-means (100 clusters)

Malik, Belongie, Shi, Leung, 1999

Walker, Malik, 2004

Malik, Belongie, Shi, Leung, 1999

Walker, Malik, 2004
Bag of words

Bag of words model

Spatially organized textures
Bag of words & spatial pyramid matching

Better Bags of Visual Features

• More advanced quantization / encoding methods that are near the state-of-the-art in image classification and image retrieval.
  – Soft assignment (a.k.a. Kernel Codebook)
  – VLAD
  – Fisher Vector

• Deep learning has taken attention away from these methods.
Standard K-means Bag of Words

Motivation

*Bag of Visual Words* is only about **counting** the number of local descriptors assigned to each Voronoi region.

Why not including **other statistics**?
We already looked at the Spatial Pyramid

But today we’re not talking about ways to preserve *spatial* information.
Motivation

*Bag of Visual Words* is only about **counting** the number of local descriptors assigned to each Voronoi region

Why not including **other statistics**? For instance:
- mean of local descriptors  

---

Motivation

*Bag of Visual Words* is only about **counting** the number of local descriptors assigned to each Voronoi region.

Why not including **other statistics**? For instance:

- mean of local descriptors
- (co)variance of local descriptors

Simple case: Soft Assignment

• Called “Kernel codebook encoding” by Chatfield et al. 2011. Cast a weighted vote into the most similar clusters.
Simple case: Soft Assignment

- Called “Kernel codebook encoding” by Chatfield et al. 2011. Cast a weighted vote into the most similar clusters.
- This is fast and easy to implement (try it for Project 4!) but it does have some downsides for image retrieval – the inverted file index becomes less sparse.
VLAD

Given a codebook $\{\mu_i, i = 1 \ldots N\}$, e.g. learned with K-means, and a set of local descriptors $X = \{x_t, t = 1 \ldots T\}$

- ① assign $\text{NN}(x_t) = \arg \min_{\mu_i} ||x_t - \mu_i||$

- ②③ compute: $v_i = \sum_{x_t: \text{NN}(x_t) = \mu_i} x_t - \mu_i$

- concatenate $v_i$'s $+ \ell_2$ normalize

A first example: the VLAD

A graphical representation of $v_i = \sum_{x_t: \text{NN}(x_t) = \mu_i} x_t - \mu_i$.

The Fisher vector

Score function

Given a likelihood function \( u_\lambda \) with parameters \( \lambda \), the score function of a given sample \( X \) is given by:

\[
G^X_\lambda = \nabla_\lambda \log u_\lambda(X)
\]

→ Fixed-length vector whose dimensionality depends only on # parameters.

Intuition: direction in which the parameters \( \lambda \) of the model should we modified to better fit the data.
Aside: Mixture of Gaussians (GMM)

- For Fisher Vector image representations, \( u \lambda \) is a GMM.
- GMM can be thought of as “soft” k-means.

- Each component has a mean and a standard deviation along each direction (or full covariance)
What about skipping quantization / summarization completely?

In Defense of Nearest-Neighbor Based Image Classification
Boiman, Shechtman, Irani. CVPR 2008
Summary

• We’ve looked at methods to better characterize the distribution of visual words in an image:
  – Soft assignment (a.k.a. Kernel Codebook)
  – VLAD
  – Fisher Vector
  – No quantization
Learning Scene Categorization

Forest path Vs. all

Living-room Vs. all
Classifier: 1-vs-all SVM with histogram intersection, chi squared, or RBF kernel.
A look into the results

Airplane cabin (64%)

Art gallery (38%)

All the results available on the web
<table>
<thead>
<tr>
<th>Practical Field</th>
<th>Performance Ratings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limousine interior</td>
<td>95% vs 80%</td>
</tr>
<tr>
<td>Riding arena</td>
<td>100% vs 90%</td>
</tr>
<tr>
<td>Sauna</td>
<td>96% vs 95%</td>
</tr>
<tr>
<td>Skatepark</td>
<td>96% vs 90%</td>
</tr>
<tr>
<td>Subway interior</td>
<td>96% vs 80%</td>
</tr>
</tbody>
</table>

**Humans good**

**Comp. good**

**Humans bad**

**Comp. bad**

**Human good**

**Comp. bad**

**Human bad**

**Comp. good**
Database and source code available at http://groups.csail.mit.edu/vision/SUN/

Additional details available:

How do we do better than 40%?

- Features from deep learning on ImageNet get 42%
- Fisher vector encoding gets up to 47.2%