Visual words

Map high-dimensional descriptors to tokens/words by quantizing the feature space.

- Quantize via clustering; cluster centers are the visual “words”
- Assign word to each image region by finding the closest cluster center.
Why can’t we train good recognition systems?

• Training Data
  – Huge issue, but not always a variable we control.

• Representation
  – Are the local features themselves lossy?
  – What about feature quantization?
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

![Graph showing scene recognition rate versus number of training samples per class. Lines represent different feature extraction methods, with some methods significantly outperforming others.](image)
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
How many object categories are there?

~10,000 to 30,000

Biederman 1987
airplane cabin
apple orchard
bakery
car factory
construction site
food court
interior car
stream
train station
397 Well-sampled Categories

...at least 100 unique images each.
Evaluating Human Scene Classification

Accuracy

98%  90%  68%
Scene category | Most confusing categories
--- | ---
Inn (0%) | Restaurant patio (44%) | Chalet (19%)
Bayou (0%) | River (67%) | Coast (8%)
Basilica (0%) | Cathedral (29%) | Courthouse (21%)
Conclusion: humans can do it

• The SUN database is reasonably consistent and categories can be told apart by humans.

• With many very specific categories, humans get it right 2/3rds of the time from experience and from exploring the label space.

So, how do humans 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)

Texture Features
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

Textons

Filter bank → Vector of filter responses at each pixel → Kmeans over a set of vectors on a collection of images
Textons

Filter bank

K-means (100 clusters)

Malik, Belongie, Shi, Leung, 1999

Walker, Malik, 2004
Gabor filter

- Sinusoid modulated by a Gaussian kernel
Global scene descriptors: GIST

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

http://people.csail.mit.edu/torralba/code/spatialenvelope/
Gist descriptor

Oliva and Torralba, 2001

Apply oriented Gabor filters over different scales.

Average filter energy per bin.

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

8 orientations
4 scales
\( \times 16 \) bins
512 dimensions

Example visual gists

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

Oliva & Torralba (2001)
Bag of words & spatial pyramid matching


But any way to improve the quantization approach itself?
Better Bags of Visual Features

• More advanced quantization / encoding methods that are near the state-of-the-art in image classification and image retrieval.
  – Mixtures of Gaussians
  – Soft assignment (a.k.a. Kernel Codebook)
  – VLAD – Vectors of Locally-Aggregated Descriptors

• Deep learning has taken attention away from these methods...
Standard Kmeans 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**?
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  \(\times\)

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

Mixture of Gaussians (GMM)

- GMM can be thought of as “soft” kmeans.
- Each component has a mean and a standard deviation along each direction (or full covariance)
- Can easily represent non-circular distributions
Simple case: Soft Assignment

- “Kernel codebook encoding” by Chatfield et al. 2011.
- Cast a set of proportional votes (weights) to \( n \) most similar clusters, rather than a single ‘hard’ vote.
Simple case: Soft Assignment

• “Kernel codebook encoding” by Chatfield et al. 2011.
• Cast a set of proportional votes (weights) to $n$ most similar clusters, rather than a single ‘hard’ vote.

• This is fast and easy to implement, but it makes an inverted file index less sparse.
VLAD – Vectors of Locally-Aggregated Descriptors

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

- \( \text{assign: } \) \( \text{NN}(x_t) = \arg\min_{\mu_i} |x_t - \mu_i| \)
- \( \text{compute: } \) \( x_i = \sum \text{NN}(x_t) \) for cell \( i \)

VLAD – Vectors of Locally-Aggregated Descriptors

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

- **assign**: \( \text{NN}(x_t) = \arg \min_{\mu_i} ||x_t - \mu_i|| \)

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

VLAD – Vectors of Locally-Aggregated Descriptors

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: \( \text{NN}(x_t) = \arg \min_{\mu_i} ||x_t - \mu_i|| \)

2. compute: \( v_i = \sum_{x_t: \text{NN}(x_t) = \mu_i} x_t - \mu_i \)

3. concatenate \( v_i \)'s + \( \ell_2 \) normalize

A first example: the VLAD

A graphical representation of

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

Jégou, Douze, Schmid and Pérez,
“Aggregating local descriptors into a compact image representation”,
CVPR’10.
Why can’t we train good recognition systems?

• Training Data
  – Huge issue, but not always a variable we control.

• Representation
  – Are the local features themselves lossy?
  – What about feature quantization?
What about skipping quantization completely?

In Defense of Nearest-Neighbor Based Image Classification
Boiman, Shechtman, Irani

Quantization inherently averages the parts which are most discriminative !!!

Quantization error of densely computed image descriptors (SIFT) using a large codebook (size 6,000) of Caltech-101. Red = high error; Blue = low error. The most informative descriptors (eye, nose, etc.) have the highest quantization error.
What about NN image-to-image matching?

In Defense of Nearest-Neighbor Based Image Classification
Boiman, Shechtman, Irani

Image to class features NN: $KL(p_Q | p_C) = 8.35$

Image to image features NN:

- $KL(p_Q | p_1) = 17.54$
- $KL(p_Q | p_2) = 18.20$
- $KL(p_Q | p_3) = 14.56$
CalTech 101 (2004) – 100 object classes; mean images
If I do both of these, NN can be a pretty good classifier!

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

• Methods to better characterize the distribution of visual words in an image:
  – Soft assignment (a.k.a. Kernel Codebook)
  – VLAD
  – 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>Humans good</th>
<th>Comp. good</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>limousine interior</strong> (95% vs 80%)</td>
<td></td>
</tr>
<tr>
<td><strong>riding arena</strong> (100% vs 90%)</td>
<td></td>
</tr>
<tr>
<td><strong>sauna</strong> (96% vs 95%)</td>
<td></td>
</tr>
<tr>
<td><strong>skatepark</strong> (96% vs 90%)</td>
<td></td>
</tr>
<tr>
<td><strong>subway interior</strong> (96% vs 80%)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Humans bad</th>
<th>Comp. bad</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human good</td>
<td>Comp. bad</td>
</tr>
<tr>
<td>Human bad</td>
<td>Comp. good</td>
</tr>
</tbody>
</table>
How do we do better than 40%?

• Features from deep learning based on ImageNet allow us to reach 42%...

Not much better...