Bag-of-features models

Many slides adapted from Fei-Fei Li, Rob Fergus, and Antonio Torralba
Overview: Bag-of-features models

- Origins and motivation
- Learning visual vocabularies
  - K-means clustering
- Discriminative methods
  - Nearest-neighbor classification
  - Distance functions
  - Support vector machines
  - Kernels
- Generative methods
  - Naïve Bayes
  - Probabilistic Latent Semantic Analysis
- Extensions: incorporating spatial information
Origin 1: Texture recognition

- Texture is characterized by the repetition of basic elements or *textons*
- For stochastic textures, it is the identity of the textons, not their spatial arrangement, that matters

Origin 1: Texture recognition

Origin 2: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary  
  Salton & McGill (1983)
Origin 2: Bag-of-words models


http://chir.ag/phernalia/preztags/
Origin 2: Bag-of-words models

Origin 2: Bag-of-words models


US Presidential Speeches Tag Cloud
http://chir.ag/phernalia/preztags/
Bags of features for object recognition

- Works pretty well for image-level classification

Csurka et al. (2004), Willamowski et al. (2005), Grauman & Darrell (2005), Sivic et al. (2003, 2005)
### Bags of features for object recognition

#### Caltech6 dataset

<table>
<thead>
<tr>
<th>class</th>
<th>bag of features</th>
<th>bag of features</th>
<th>Parts-and-shape model</th>
</tr>
</thead>
<tbody>
<tr>
<td>airplanes</td>
<td>98.8</td>
<td>97.1</td>
<td>90.2</td>
</tr>
<tr>
<td>cars (rear)</td>
<td>98.3</td>
<td>98.6</td>
<td>90.3</td>
</tr>
<tr>
<td>cars (side)</td>
<td>95.0</td>
<td>87.3</td>
<td>88.5</td>
</tr>
<tr>
<td>faces</td>
<td>100</td>
<td>99.3</td>
<td>96.4</td>
</tr>
<tr>
<td>motorbikes</td>
<td>98.5</td>
<td>98.0</td>
<td>92.5</td>
</tr>
<tr>
<td>spotted cats</td>
<td>97.0</td>
<td>—</td>
<td>90.0</td>
</tr>
</tbody>
</table>
Bag of features: outline

1. Extract features
Bag of features: outline

1. Extract features
2. Learn “visual vocabulary”
Bag of features: outline

1. Extract features
2. Learn “visual vocabulary”
3. Quantize features using visual vocabulary
Bag of features: outline

1. Extract features
2. Learn “visual vocabulary”
3. Quantize features using visual vocabulary
4. Represent images by frequencies of “visual words”
1. Feature extraction

Regular grid
- Vogel & Schiele, 2003
- Fei-Fei & Perona, 2005
1. Feature extraction

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 extraction

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

Interest point detector
- Csurka et al. 2004
- Fei-Fei & Perona, 2005
- Sivic et al. 2005

Other methods
- Random sampling (Vidal-Naquet & Ullman, 2002)
- Segmentation based patches (Barnard, Duygulu, Forsyth, de Freitas, Blei, Jordan, 2003)
1. Feature extraction

- **Detect patches**
  - [Mikołajczyk and Schmid '02]
  - [Mata, Chum, Urban & Pajdla, '02]
  - [Sivic & Zisserman, '03]

- **Normalize patch**

- **Compute SIFT descriptor**
  - [Lowe'99]

Slide credit: Josef Sivic
1. Feature extraction
2. Learning the visual vocabulary
2. Learning the visual vocabulary

Slide credit: Josef Sivic
2. Learning the visual vocabulary
K-means clustering

- Want to minimize sum of squared Euclidean distances between points $x_i$ and their nearest cluster centers $m_k$

$$D(X, M) = \sum_{k} \sum_{\text{point } i \text{ in cluster } k} (x_i - m_k)^2$$

Algorithm:
- Randomly initialize K cluster centers
- Iterate until convergence:
  - Assign each data point to the nearest center
  - Recompute each cluster center as the mean of all points assigned to it
From clustering to vector quantization

- Clustering is a common method for learning a visual vocabulary or codebook
  - Unsupervised learning process
  - Each cluster center produced by k-means becomes a codevector
  - Codebook can be learned on separate training set
  - Provided the training set is sufficiently representative, the codebook will be “universal”

- The codebook is used for quantizing features
  - A vector quantizer takes a feature vector and maps it to the index of the nearest codevector in a codebook
  - Codebook = visual vocabulary
  - Codevector = visual word
Example visual vocabulary

Fei-Fei et al. 2005
Image patch examples of visual words

Sivic et al. 2005
Visual vocabularies: Issues

- How to choose vocabulary size?
  - Too small: visual words not representative of all patches
  - Too large: quantization artifacts, overfitting

- Generative or discriminative learning?

- Computational efficiency
  - Vocabulary trees
    (Nister & Stewenius, 2006)
3. Image representation

![Image representation diagram with bars and codewords]
Image classification

• Given the bag-of-features representations of images from different classes, how do we learn a model for distinguishing them?