Recall the machine learning approach

Training images + labels \[\xrightarrow{\text{Image Features}}\] Training Labels \[\xrightarrow{\text{Training}}\] Learned model

Testing

Test Image \[\xrightarrow{\text{Image Features}}\] Prediction

The importance of good features

- Most learning methods are invariant to feature permutation
  - E.g., patch vs. pixel representation of images

[Images of permuted pixels and permuted patches]

Can you recognize the digits?
The importance of good features

- Consider matching with image patches
  - What could go wrong?

  ![Template](image1.png)
  ![Image](image2.png)

  match quality
  
  e.g., cross correlation

What is a feature map?

- Any transformation of an image into a new representation
- Example: transform an image into a binary edge map

![Image source: wikipedia](image3.png)

Feature map goals

- Introduce invariance to nuisance factors
  - Illumination changes
  - Small translations, rotations, scaling, shape deformations

![Figure 1.3: Variation in appearance due to a change in illumination](image4.png)

We will discuss ...

- Two popular image features
  - Histogram of Oriented Gradients (HOG)
  - Bag of Visual Words (BoVW)

- Applications of these features
Histogram of Oriented Gradients

- Introduced by Dalal and Triggs (CVPR 2005)
- An extension of the SIFT feature
- HOG properties:
  - Preserves the overall structure of the image
  - Provides robustness to illumination and small deformations

HOG feature: basic idea

- Divide the image into blocks
- Compute histograms of gradients for each region

HOG feature: full pipeline

- Smaller bin-size: better spatial resolution
- Larger bin-size: better invariance to deformations
- Optimal value depends on the object category being modeled
  - e.g. rigid vs. deformable objects

Effect of bin-size

- 10x10 cells
- 20x20 cells
Template matching with HOG

- Compute the HOG feature map for the image
- Convolve the template with the feature map to get score
- Find peaks of the response map (non-max suppression)
- What about multi-scale?

Multi-scale template matching

\[ \text{score}(I, p) = w \cdot \phi(I, p) \]

- Compute HOG of the whole image at multiple resolutions
- Score each sub-windows of the feature pyramid

Example detections

We will discuss …

- Two popular image features
  - Histogram of Oriented Gradients (HOG)
  - Bag of Visual Words (BoVW)

Bag of visual words

- Origin and motivation of the “bag of words” model
- Algorithm pipeline
  - Extracting local features
  - Learning a dictionary — clustering using k-means
  - Encoding methods — hard vs. soft assignment
  - Spatial pooling — pyramid representations
  - Similarity functions and classifiers

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

Properties:
- Spatial structure is not preserved
- Invariance to large translations

Compare this to the HOG feature
Origin 1: Texture recognition


Origin 2: Bag-of-words models


US Presidential Speeches Tag Cloud
http://chir.ag/projects/preztags/

Origin 2: Bag-of-words models

Origin 2: Bag-of-words models


Origin and motivation of the “bag of words” model

Algorithm pipeline
- Extracting local features
- Learning a dictionary — clustering using k-means
- Encoding methods — hard vs. soft assignment
- Spatial pooling — pyramid representations
- Similarity functions and classifiers

Lecture outline

Local feature extraction

- Regular grid or interest regions

Local feature extraction

Choices of descriptor:
- SIFT
- The patch itself
- …
Local feature extraction

Extract features from many images

Lecture outline

- Origin and motivation of the “bag of words” model
- Algorithm pipeline
  - Extracting local features
  - Learning a dictionary — clustering using k-means
- Encoding methods — hard vs. soft assignment
- Spatial pooling — pyramid representations
- Similarity functions and classifiers

Learning a dictionary

- Clustering

Slide credit: Josef Sivic
**Learning a dictionary**

- Clustering

**Visual vocabulary**

- Group together similar instances

**Example:** 2D points

---

**Clustering**

- **Basic idea:** group together similar instances
- **Example:** 2D points

- What could similar mean?
  - One option: small Euclidean distance (squared)

\[
\text{dist}(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||^2_2
\]

- Clustering results are crucially dependent on the measure of similarity (or distance) between points to be clustered

---

**Clustering algorithms**

- **Simple clustering:** organize elements into k groups
  - K-means
  - Mean shift
  - Spectral clustering

- **Hierarchical clustering:** organize elements into a hierarchy
  - Bottom up - agglomerative
  - Top down - divisive
Image segmentation: break up the image into similar regions

Clustering examples

News articles

People by space and time

image credit: Berkeley segmentation benchmark

image credit: Pilho Kim
Clustering using k-means

- Given \( (x_1, x_2, \ldots, x_n) \) partition the \( n \) observations into \( k (\leq n) \) sets \( S = \{S_1, S_2, \ldots, S_k\} \) so as to minimize the within-cluster sum of squared distances.

- The objective is to minimize:

\[
\arg \min_S \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2
\]

Lloyd’s algorithm for k-means

- Initialize \( k \) centers by picking \( k \) points randomly among all the points.
- Repeat till convergence (or max iterations):
  - Assign each point to the nearest center (assignment step):

\[
\arg \min_S \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2
\]

  - Estimate the mean of each group (update step):

\[
\arg \min_S \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2
\]

k-means in action

- Grouping pixels based on intensity similarity.

k-means for image segmentation

- Feature space: intensity value (1D).
Example codebook

Another codebook

Source: B. Leibe 45
Source: B. Leibe 46

Lecture outline

- Origin and motivation of the "bag of words" model
- Algorithm pipeline
  - Extracting local features
  - Learning a dictionary — clustering using k-means
  - Encoding methods — hard vs. soft assignment
  - Spatial pooling — pyramid representations
  - Similarity functions and classifiers

Encoding methods

Assigning words to features

Visual vocabulary

partition of space

Figure from Chatfield et al., 2011

Also called hard assignment
Assigning words to features

- **Visual vocabulary**
- different words
- similar features
- hard assignment
- large quantization error

• Assigning words to features

- **Soft assignment**

\[ \alpha_i \propto e^{-f(d(x, c_i))} \]

assign high weights to centers that are close in practice non-zero to only k-nearest neighbors

= 52

**Encoding considerations**

- What should be the size of the dictionary?
  - Too small: don’t capture the variability of the dataset
  - Too large: have too few points per cluster

- Speed of embedding
  - Exact nearest neighbor is slow if the dictionary is large
  - Approximate nearest neighbor techniques
    - Search trees — organize data in a tree
    - Hashing — create buckets in the feature space
Lecture outline

- Origin and motivation of the “bag of words” model
- Algorithm pipeline
  - Extracting local features
  - Learning a dictionary — clustering using k-means
  - Encoding methods — hard vs. soft assignment
  - **Spatial pooling — pyramid representations**
- Similarity functions and classifiers

Figure from Chatfield et al., 2011

Spatial pyramids

**pooling**: sum embeddings of local features within a region

Same motivation as SIFT — keep coarse layout information

Lazebnik, Schmid & Ponce (CVPR 2006)
Lecture outline

- Origin and motivation of the “bag of words” model
- Algorithm pipeline
  - Extracting local features
  - Learning a dictionary — clustering using k-means
  - Encoding methods — hard vs. soft assignment
  - Spatial pooling — pyramid representations
- Similarity functions and classifiers

Bags of features representation

\[ I \]

\[ h = \Phi(I) \]

image similarity = feature similarity

Comparing features

- Euclidean distance:
  \[ D(h_1, h_2) = \sqrt{\sum_{i=1}^{N} (h_1(i) - h_2(i))^2} \]

- L1 distance:
  \[ D(h_1, h_2) = \sum_{i=1}^{N} |h_1(i) - h_2(i)| \]

Classifiers

- Decision trees

- Nearest neighbor classifiers
Lecture outline

- Origin and motivation of the “bag of words” model
- Algorithm pipeline
  - Extracting local features
  - Learning a dictionary — clustering using k-means
  - Encoding methods — hard vs. soft assignment
  - Spatial pooling — pyramid representations
- Similarity functions and classifiers

Results: scene category dataset

Putting it all together

Results: Caltech-101 dataset

Multi-class classification results (100 training images per class)

<table>
<thead>
<tr>
<th>Level</th>
<th>Weak features (vocabulary size: 16)</th>
<th>Strong features (vocabulary size: 200)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Single-level</td>
<td>Pyramid</td>
</tr>
<tr>
<td>0 (1 × 1)</td>
<td>45.3 ± 0.5</td>
<td>72.2 ± 0.6</td>
</tr>
<tr>
<td>1 (2 × 2)</td>
<td>53.6 ± 0.3</td>
<td>56.2 ± 0.6</td>
</tr>
<tr>
<td>2 (4 × 4)</td>
<td>61.7 ± 0.6</td>
<td>64.7 ± 0.7</td>
</tr>
<tr>
<td>3 (8 × 8)</td>
<td>63.3 ± 0.8</td>
<td>66.8 ± 0.6</td>
</tr>
</tbody>
</table>

Further thoughts and readings ...

- All about embeddings (detailed experiments and code)
- K. Chatfield et al., The devil is in the details: an evaluation of recent feature encoding methods, BMVC 2011
- http://www.robots.ox.ac.uk/~vgg/research/encoding_eval/
- Includes discussion of advanced embeddings such as Fisher vector representations and locally linear coding (LLC)