Bag of Words

- Bag-of-words
- (Spatial) pyramid matching
- Matlab demo
Political observers say that the government of Zorgia does not control the political situation. The government will not hold elections …

The ZH-20 unit is a 200Gigahertz processor with 2Gigabyte memory. Its strength is its bus and high-speed memory……

How to compare the two articles?
Bag-of-words

Compare histograms

Analogy:
Text fragment ↔ Image region
Word ↔ Texton
Representing textures as bags-of-visual words


Source: Lana Lazebnik
Given a large set of vectorized image patches: \( x \in \mathbb{R}^{M \times M} \Rightarrow x \in \mathbb{R}^{M^2} \)
and a bank of vectorized filters \( F = [f_1, f_2, \ldots, f_b] \)

1. Project each patch into *basis* spanned by \( F \): \( y = F^T x, \quad y \in \mathbb{R}^b \)
   (does this basis span \( \mathbb{R}^{M^2} \)? Is it orthonormal?)

2. Cluster patches in this projected space
Use pseudoinverse of filter bank to visualize cluster means in original space

Given a $M \times M$ image patch ‘$x$’ (reshaped into a $M^2$ vector) and a filter bank of $B$ filters, filter bank responses can be seen as a change of basis

$$y = F^T x, \quad x \in \mathbb{R}^{M^2}, y \in \mathbb{R}^B$$

$$x \approx (F^T)^+ y$$

$$Vis(d_j) \approx (F^T)^+ d_j$$
Recognition with bag-of-words

- Summarize entire image based on its distribution (histogram) of word occurrences.
- Compare to stored library of images (or class-specific *models*)

Image credit: Fei-Fei Li
Digression: alternative to quantization

Krauman & Darrell

\[ \mathbf{X} = \{ \tilde{x}_1, \ldots, \tilde{x}_m \}, \quad \tilde{x}_i \in \mathbb{R}^d \]

\[ \mathbf{Y} = \{ \tilde{y}_1, \ldots, \tilde{y}_n \}, \quad \tilde{y}_i \in \mathbb{R}^d \]

Optimal partial matching

\[
\max_{\pi: \mathbf{X} \rightarrow \mathbf{Y}} \sum_{x_i \in \mathbf{X}} S(x_i, \pi(x_i))
\]
Pyramid match kernel

Approximate partial match similarity

\[ K_{\Delta} = \sum_{i=0}^{L} w_i N_i \]

Number of newly matched pairs at level \( i \)

Measure of difficulty of a match at level \( i \)
Feature extraction

\[ X = \{ \vec{x}_1, \ldots, \vec{x}_m \}, \quad \vec{x}_i \in \mathbb{R}^d \]

Histogram pyramid: level \( i \) has bins of size \( 2^i \)

\[ \Psi(X) = [H_0(X), \ldots, H_L(X)] \]
Counting matches

Histogram intersection

$$\mathcal{I}(H(X), H(Y)) = \sum_{j=1}^{r} \min(H(X)_j, H(Y)_j)$$

$X$

$Y$

$H(X)$

$H(Y)$

$\mathcal{I}(H(X), H(Y)) = 4$
Counting new matches

Histogram intersection

\[ I(H(X), H(Y)) = \sum_{j=1}^{r} \min(H(X)_j, H(Y)_j) \]

matches at this level

\[ N_i = I(H_i(X), H_i(Y)) - I(H_{i-1}(X), H_{i-1}(Y)) \]

matches at previous level

Difference in histogram intersections across levels counts \textit{number of new pairs} matched
Pyramid match kernel

\[
K_\Delta (\Psi(X), \Psi(Y)) = \sum_{i=0}^{L} \frac{1}{2^i} \left( \mathcal{I}(H_i(X), H_i(Y)) - \mathcal{I}(H_{i-1}(X), H_{i-1}(Y)) \right)
\]

• Weights inversely proportional to bin size
• Normalize kernel values to avoid favoring large sets


**Spatial Pyramid Matching**

Quantize features into words, but build pyramid in space

Multiresolution representations are powerful!

Find maximum-weight matching (weight is inversely proportional to distance)

Indyk & Thaper (2003), Grauman & Darrell (2005)

Total weight (value of pyramid match kernel): $I_3 + \frac{1}{2}(I_2 - I_3) + \frac{1}{4}(I_1 - I_2) + \frac{1}{8}(I_0 - I_1)$
Outline

• Efficiency (pyramids, separability, steerability)

• Linear algebra

• Bag-of-words

• Frequency analysis (don’t expect to get to)