CMPSCI 370: Intro. to Computer Vision

Image representation

University of Massachusetts, Amherst

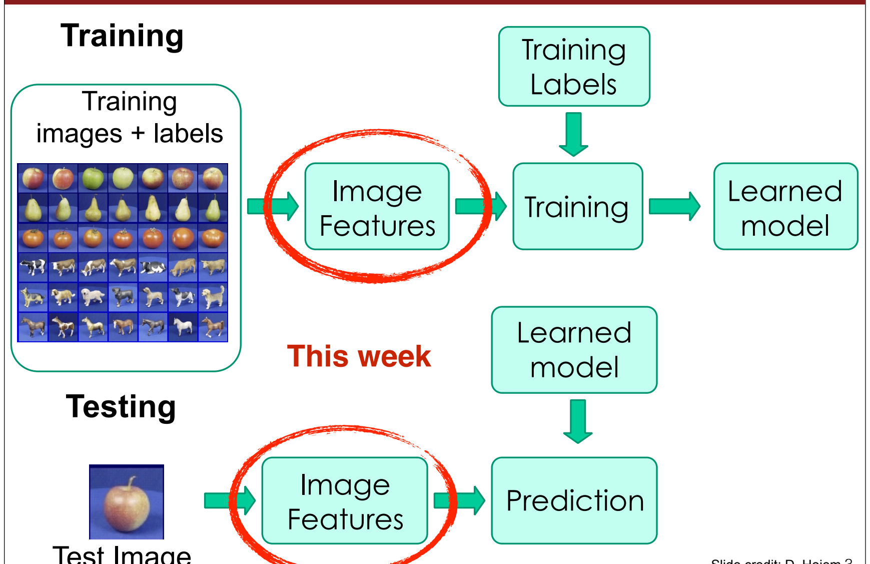
April 12/14, 2016

Instructor: Subhransu Maji

Administrivia

- Homework 5 posted
  - Due April 26, 5:00 PM (note the change in time)
  - Last day of class (don't skip class to do the homework)
- No HH section today
- In the remaining five classes
  - Image representations (this week)
  - Convolutional neural networks (next week +)
  - Some other topic (if time permits) — tracking, optical flow, computational photography, etc.

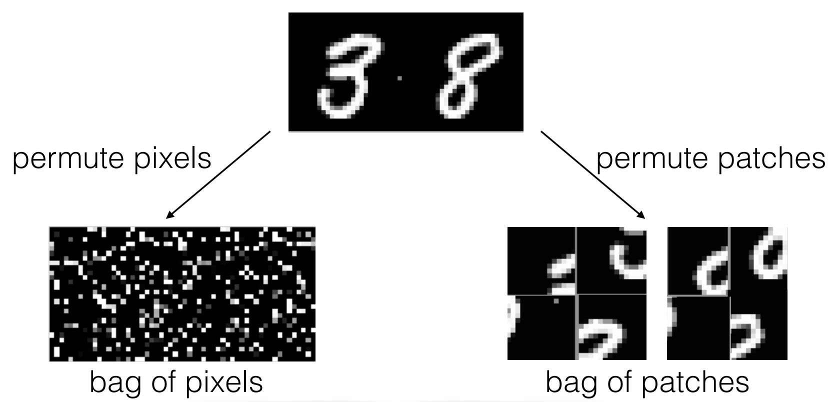
Recall the machine learning approach



Slide credit: D. Hoiem 3

The importance of good features

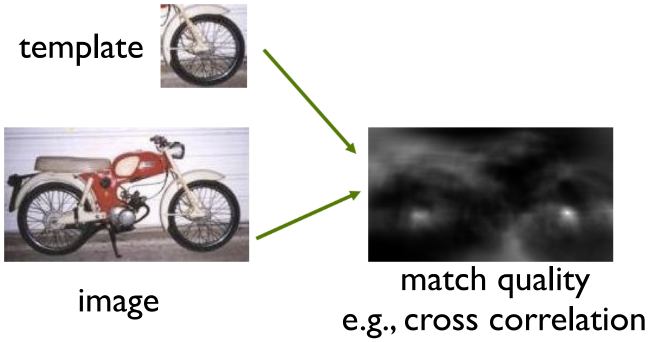
- Most learning methods are invariant to feature permutation
  - E.g., patch vs. pixel representation of images



can you recognize the digits?

The importance of good features

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



What is a feature map?

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

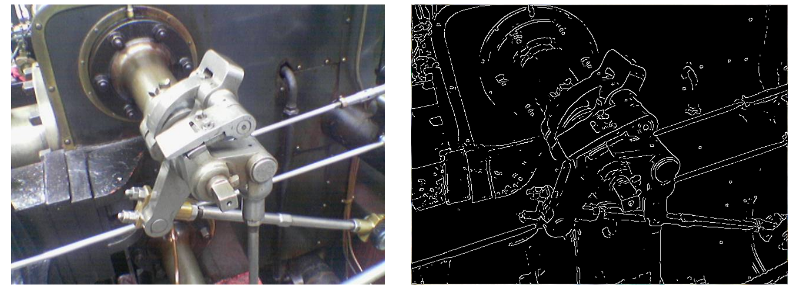


Image source: wikipedia

Feature map goals

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

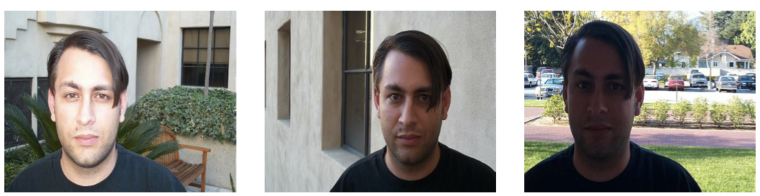


Figure 1.3: Variation in appearance due to a change in illumination

Image: [Fergus05]

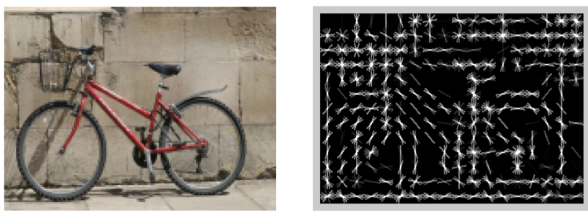
We will discuss …

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

- Applications of these features

Histogram of Oriented Gradients

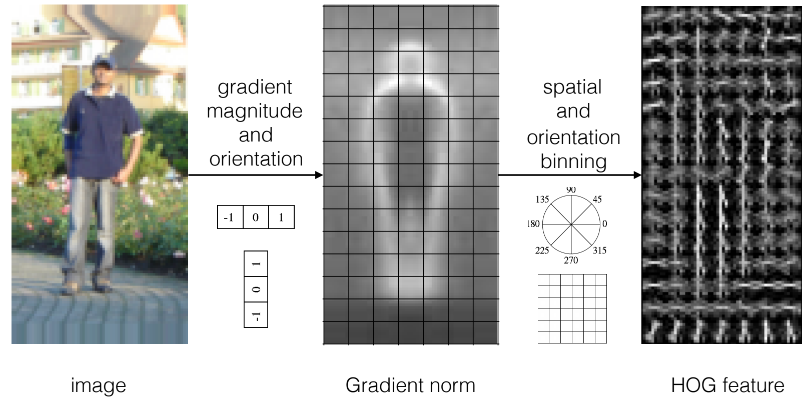
- Introduced by Dalal and Triggs (CVPR 2005)
- An extension of the SIFT feature
- HOG properties:
  - Preserves the overall structure of the image
  - Provides robustness to illumination and small deformations



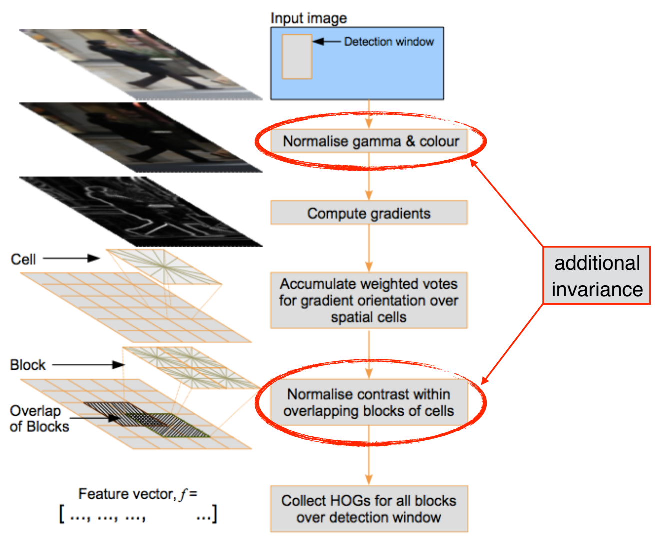
HOG feature

HOG feature: basic idea

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

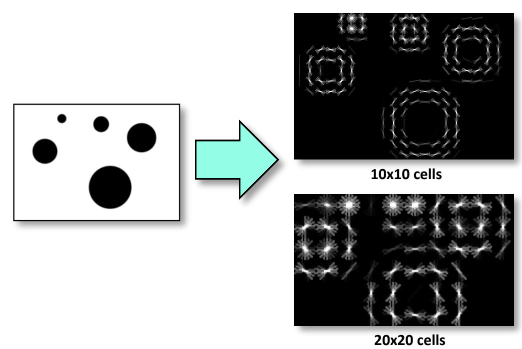


HOG feature: full pipeline

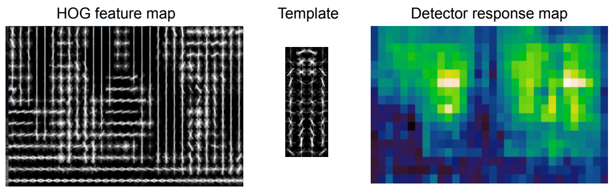


Effect of bin-size

- **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

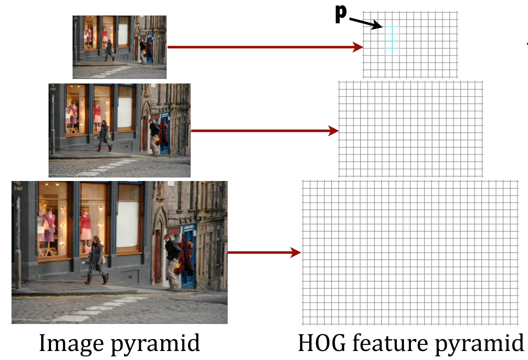


Template matching with HOG

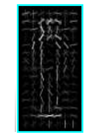


- Compute the HOG feature map for the image
- Convolve the template with the feature map to get score
- Find peaks of the response map (non-max suppression)
- What about multi-scale?

Multi-scale template matching

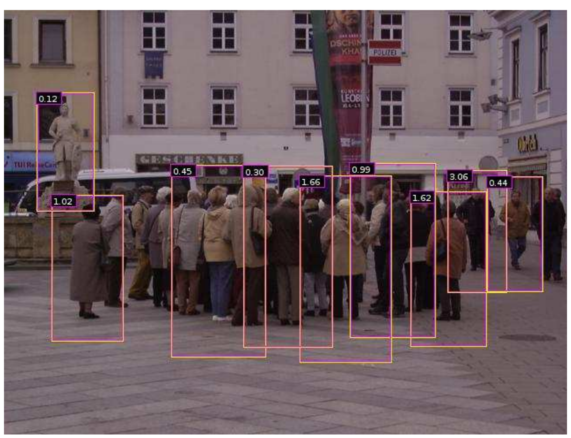


$$score(I, p) = \mathbf{w} \cdot \phi(I, p)$$



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

Example detections



N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

Example detections

N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

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

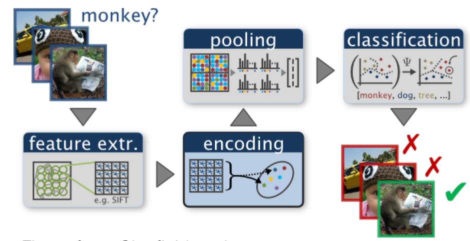
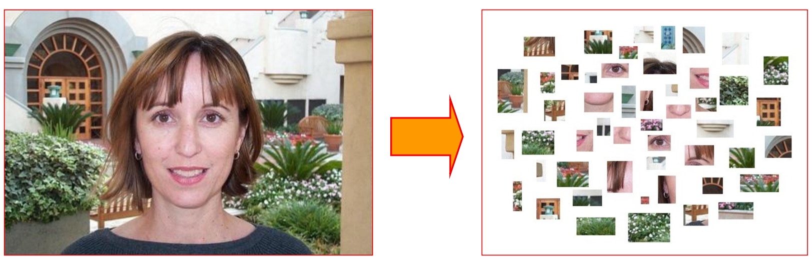


Figure from *Chatfield et al.,2011*

Bag of features

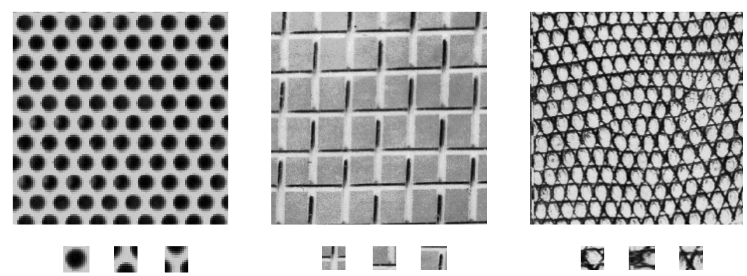


Properties:
- Spatial structure is not preserved
- Invariance to large translations

Compare this to the HOG feature

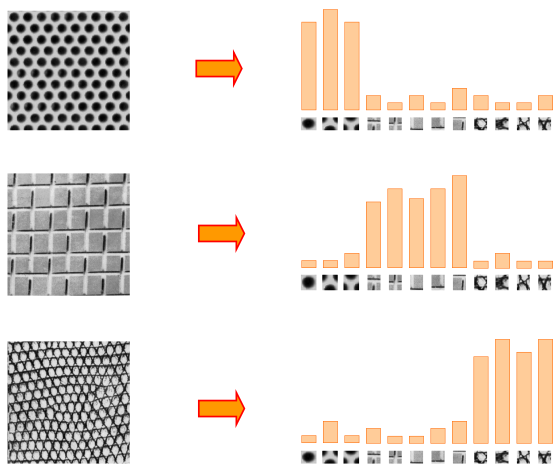
Origin 1: Texture recognition

- 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



Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

Origin 1: Texture recognition



Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

Origin 2: Bag-of-words models

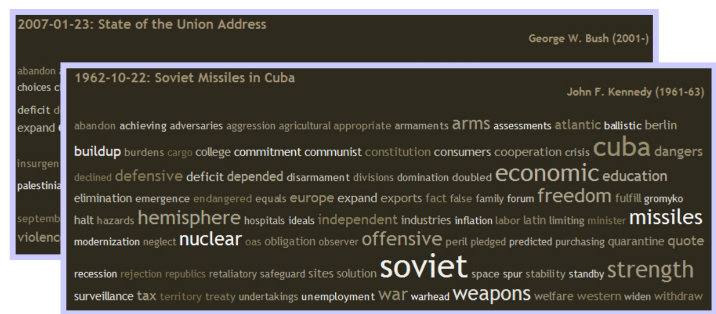
- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)

Origin 2: Bag-of-words models

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

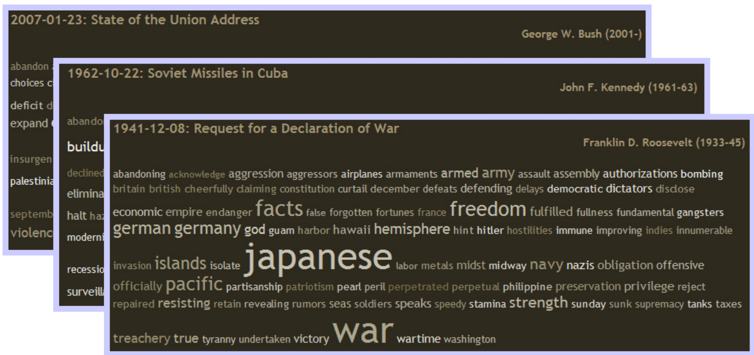
Origin 2: Bag-of-words models

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



Origin 2: Bag-of-words models

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



25

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

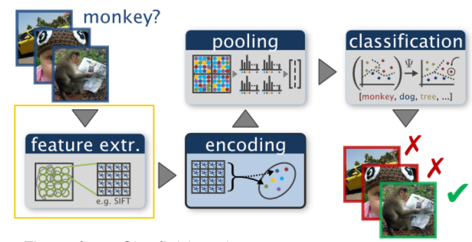
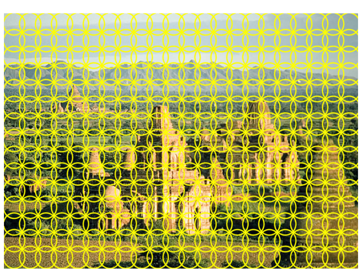


Figure from *Chatfield et al.,2011*

26

Local feature extraction

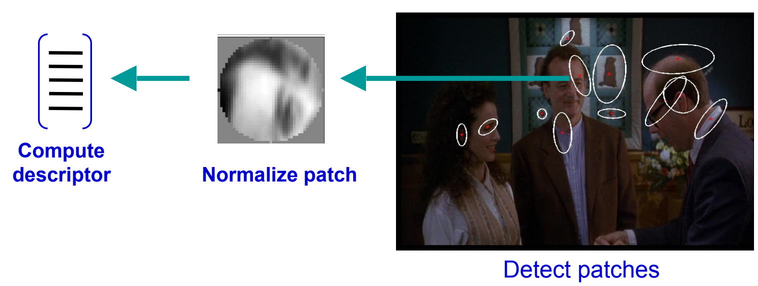
- Regular grid or interest regions



corner detector

27

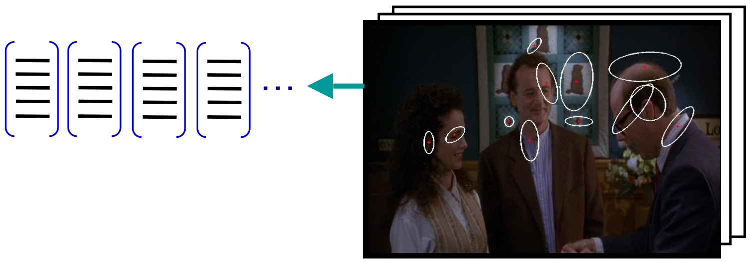
Local feature extraction



Choices of descriptor:
- SIFT
- The patch itself
- …

Slide credit: Josef Sivic 28

Local feature extraction



Extract features from many images

Slide credit: Josef Sivic

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

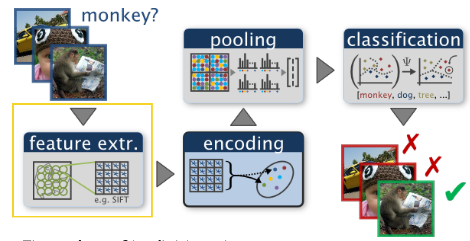
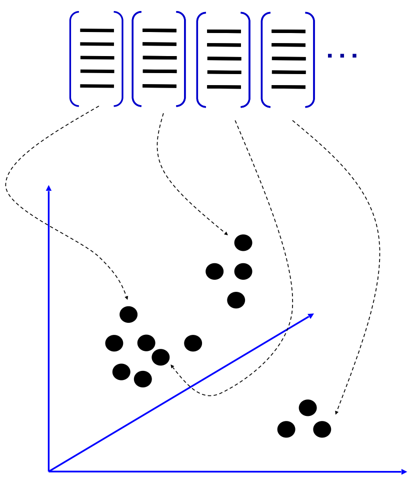


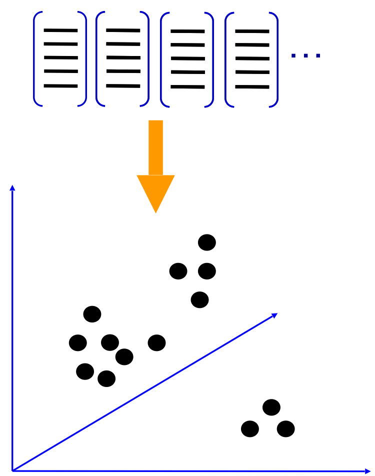
Figure from *Chatfield et al.,2011*

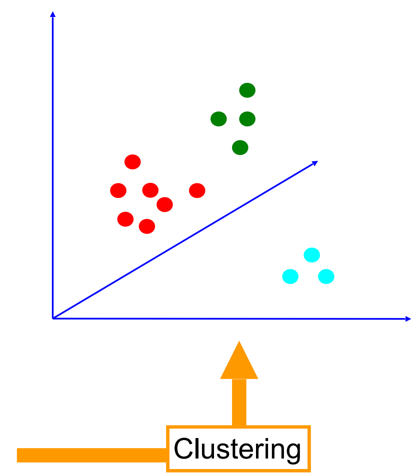
Learning a dictionary



Slide credit: Josef Sivic

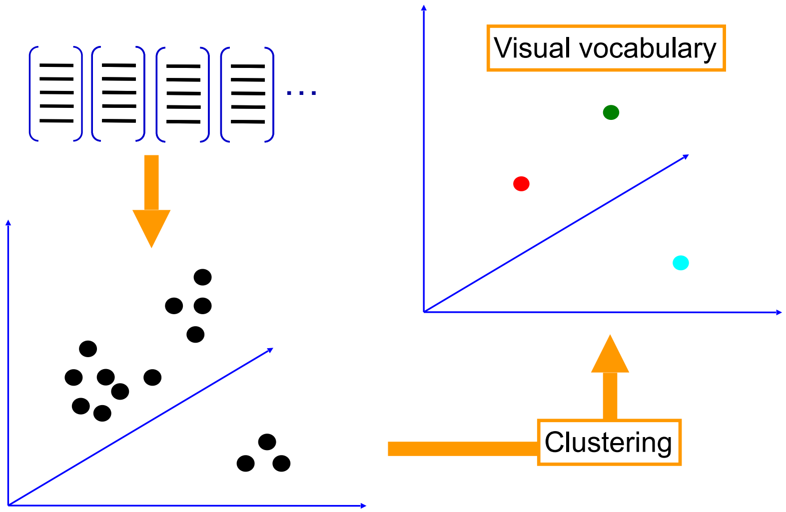
Learning a dictionary





Slide credit: Josef Sivic

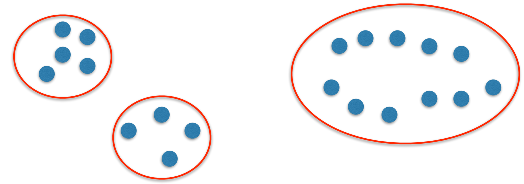
Learning a dictionary



Slide credit: Josef Sivic

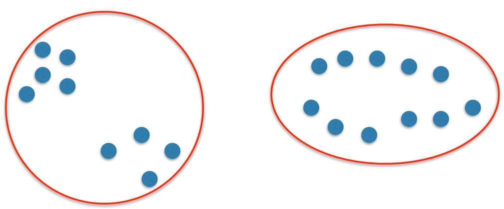
Clustering

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



Clustering

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



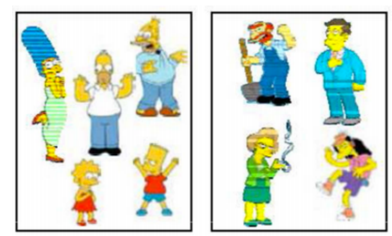
- What could similar mean?
  - One option: small Euclidean distance (squared)

$$\mathrm{dist}(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_2^2$$

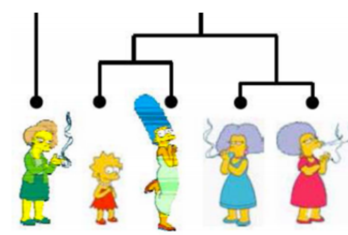
  - Clustering results are crucially dependent on the measure of similarity (or distance) between points to be clustered

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



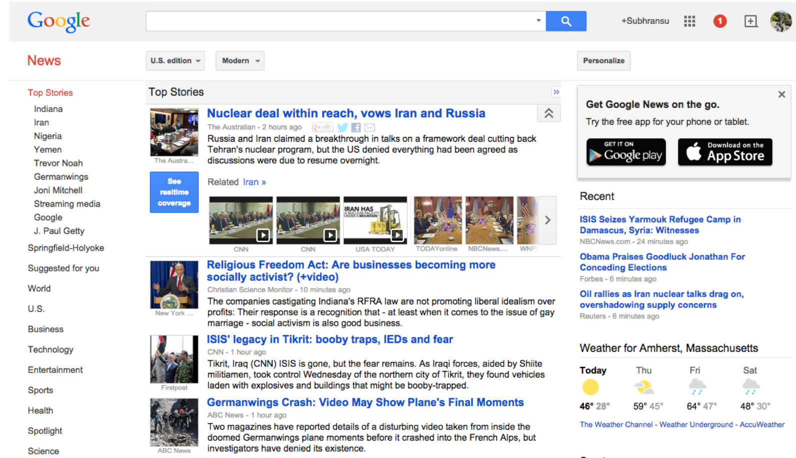
Clustering examples

- Image segmentation: break up the image into similar regions

image credit: Berkeley segmentation benchmark

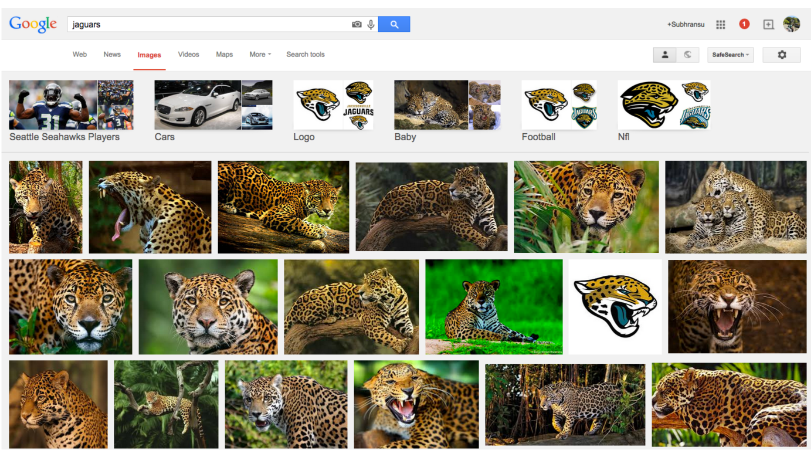
Clustering examples

- Clustering news articles



Clustering examples

- Clustering queries



Clustering examples

- Clustering people by space and time

image credit: Pilho Kim

Clustering using k-means

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

- The objective is to minimize:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} ||\mathbf{x} - \mu_i||^2$$

cluster center

Lloyd's algorithm for k-means

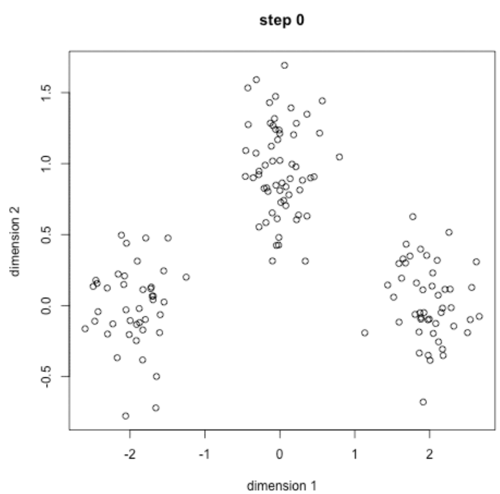
- Initialize k centers by picking k points randomly among all the points
- Repeat till convergence (or max iterations)
  - Assign each point to the nearest center (assignment step)

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \underline{\sum_{\mathbf{x} \in S_i}} ||\mathbf{x} - \mu_i||^2$$

  - Estimate the mean of each group (update step)

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \underline{||\mathbf{x} - \mu_i||^2}$$

k-means in action

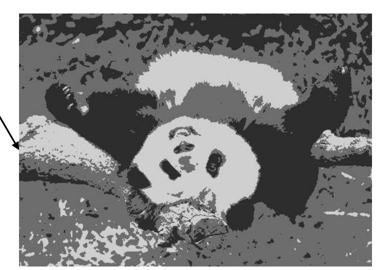


http://simplystatistics.org/2014/02/18/k-means-clustering-in-a-gif/

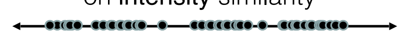
k-means for image segmentation

K=2

K=3

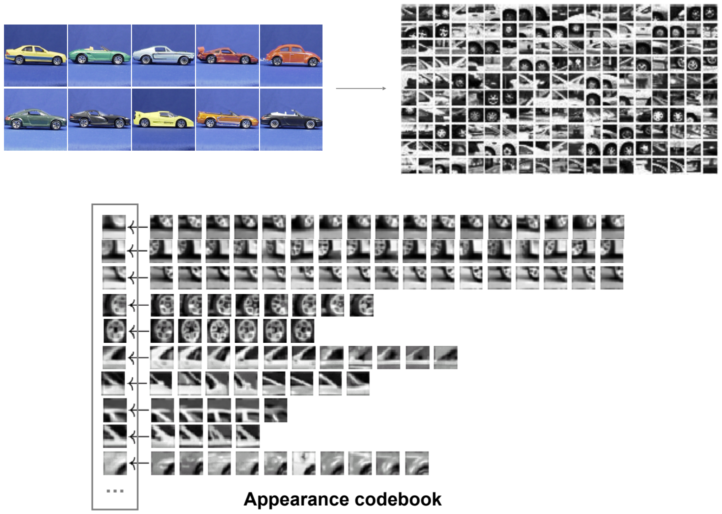


Grouping pixels based on **intensity** similarity



feature space: intensity value (1D)

Example codebook



Appearance codebook

Source: B. Leibe 45

Another codebook

Appearance codebook

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

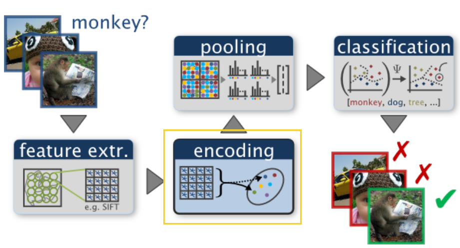
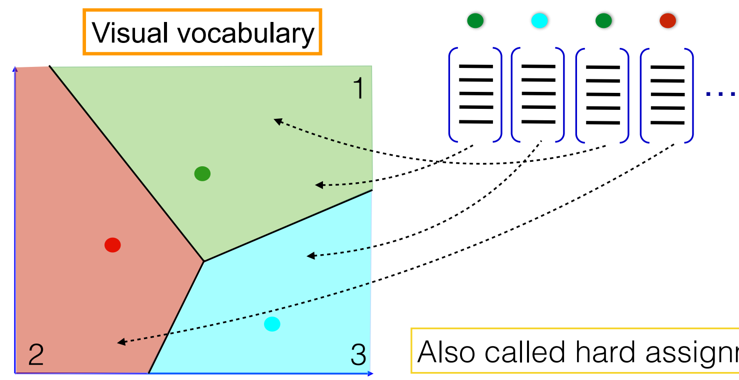


Figure from *Chatfield et al.,2011*

Encoding methods

- Assigning words to features

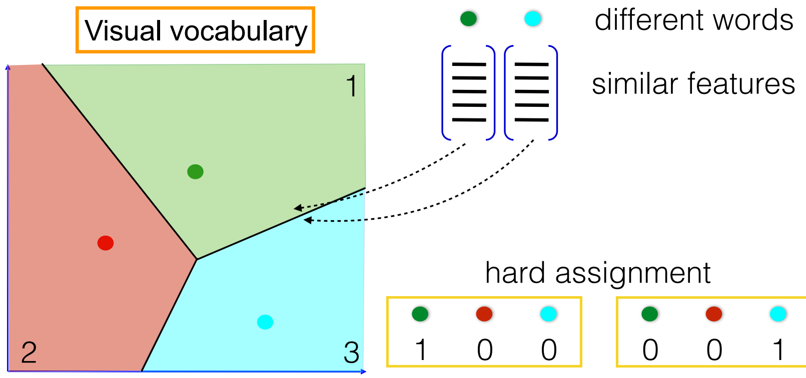


Also called hard assignment

partition of space

Encoding methods

- Assigning words to features



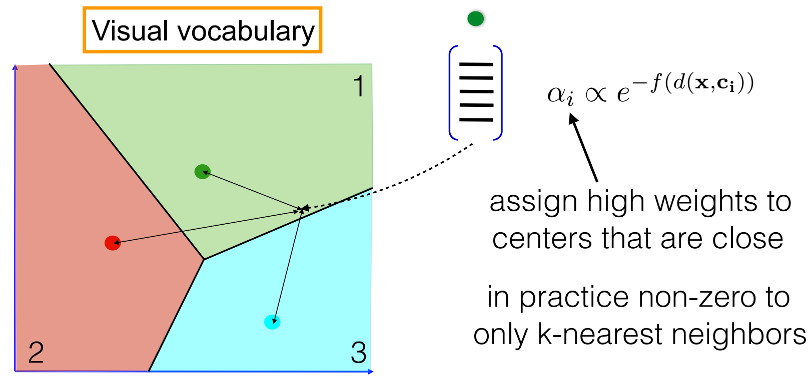
partition of space

large quantization error

Encoding methods

- Assigning words to features

soft assignment



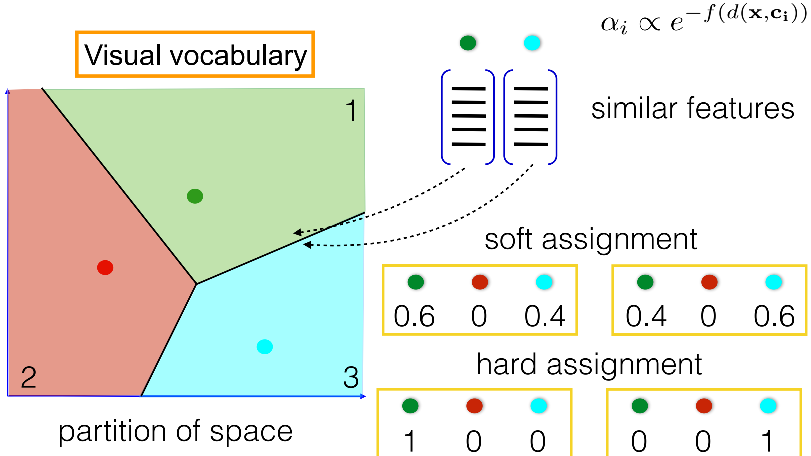
partition of space

Encoding methods

- Assigning words to features

soft assignment

$\alpha_i \propto e^{-f(d(\mathbf{x},\mathbf{c_i}))}$



partition of space

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

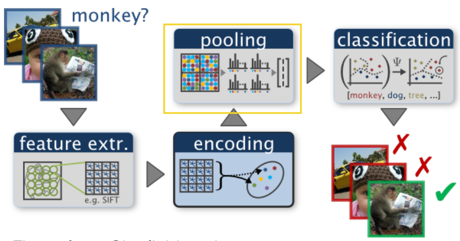
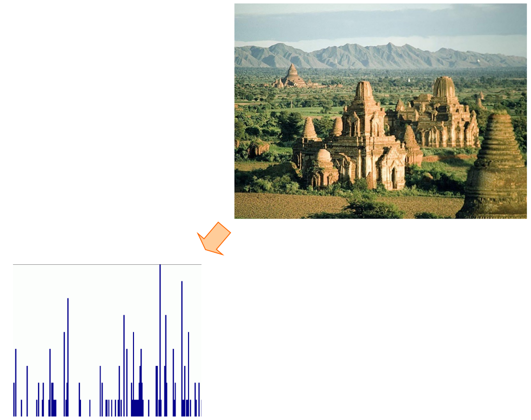


Figure from *Chatfield et al.,2011*

Spatial pyramids

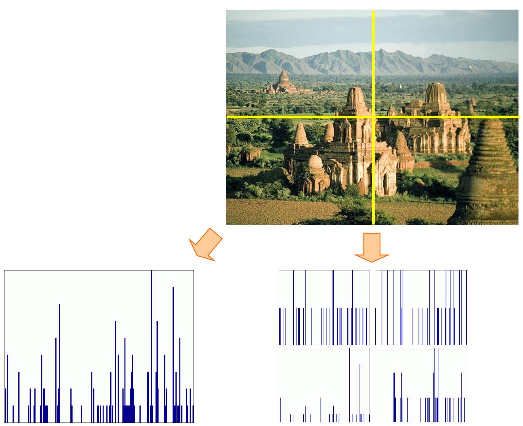
pooling: sum embeddings of local features within a region



Lazebnik, Schmid & Ponce (CVPR 2006)

Spatial pyramids

pooling: sum embeddings of local features within a region

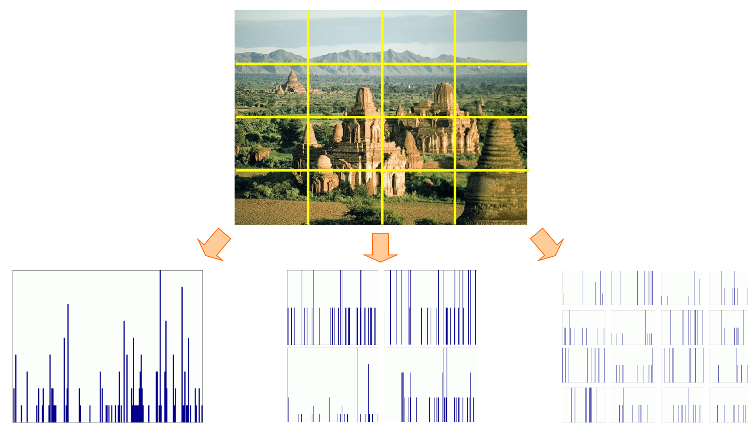


Same motivation as **SIFT** — keep coarse layout information

Lazebnik, Schmid & Ponce (CVPR 2006)

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

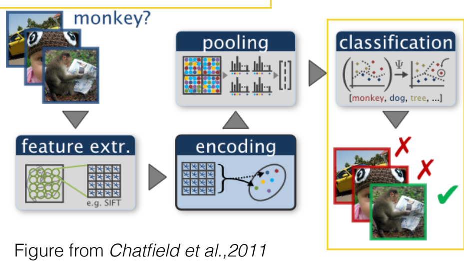


Figure from *Chatfield et al.,2011*

Bags of features representation

I $\qquad\qquad$ $\mathbf{h} = \Phi(I)$

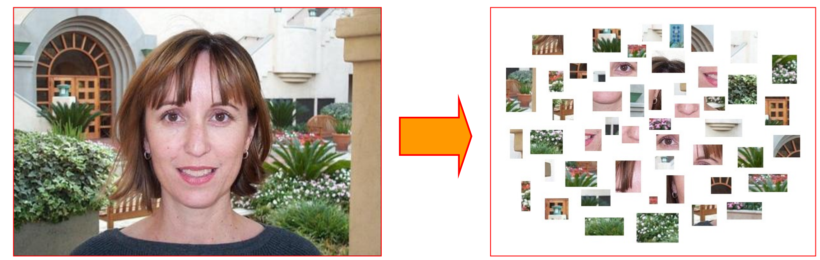


image similarity = feature similarity

Comparing features

- Euclidean distance:

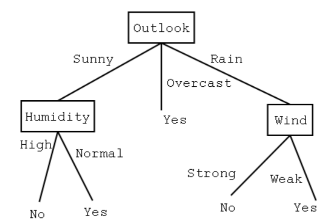
$$D(\mathbf{h}_1, \mathbf{h}_2) = \sqrt{\sum_{i=1}^{N} (\mathbf{h}_1(i) - \mathbf{h}_2(i))^2}$$

- L1 distance:

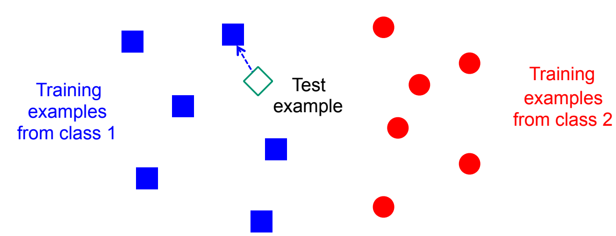
$$D(\mathbf{h}_1, \mathbf{h}_2) = \sum_{i=1}^{N} |\,\mathbf{h}_1(i) - \mathbf{h}_2(i)\,|$$

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

Putting it all together

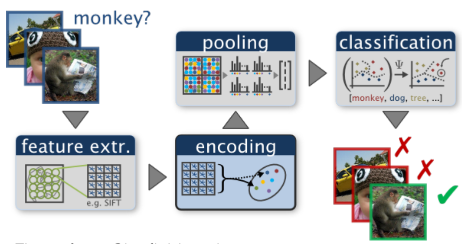
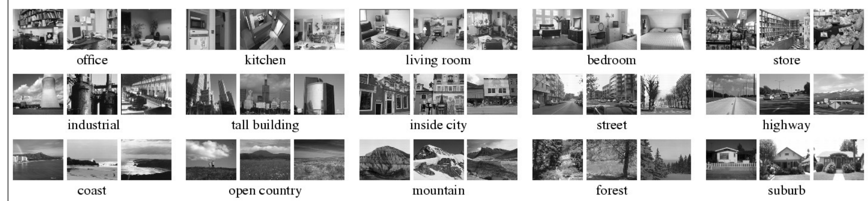


Figure from *Chatfield et al.,2011*

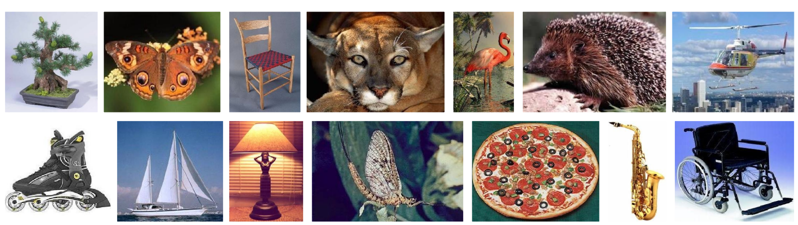
Results: scene category dataset



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

	Weak features (vocabulary size: 16)		Strong features (vocabulary size: 200)	
Level	Single-level	Pyramid	Single-level	Pyramid
0 (1×1)	45.3 ±0.5		72.2 ±0.6	
1 (2×2)	53.6 ±0.3	56.2 ±0.6	77.9 ±0.6	79.0 ±0.5
2 (4×4)	61.7 ±0.6	64.7 ±0.7	79.4 ±0.3	**81.1** ±0.3
3 (8×8)	63.3 ±0.8	**66.8** ±0.6	77.2 ±0.4	80.7 ±0.3

Results: Caltech-101 dataset



Multi-class classification results (30 training images per class)

	Weak features (16)		Strong features (200)	
Level	Single-level	Pyramid	Single-level	Pyramid
0	15.5 ±0.9		41.2 ±1.2	
1	31.4 ±1.2	32.8 ±1.3	55.9 ±0.9	57.0 ±0.8
2	47.2 ±1.1	49.3 ±1.4	63.6 ±0.9	**64.6** ±0.8
3	52.2 ±0.8	**54.0** ±1.1	60.3 ±0.9	64.6 ±0.7

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)