Hands on Advanced Bag-of-Words Models for Visual Recognition

Lamberto Ballan and Lorenzo Seidenari

MICC - University of Florence

- The tutorial will start at 14:30
- In the meanwhile please download the **matlab code** and **images** from: https://sites.google.com/site/iciap13handsonbow/
- We have also some USB pendrives with the material

- The starting point is the *exercises.m* file (we provide you also the *exercises_solutions.m* script)

September 9, 2013 – Villa Doria D'Angri, Napoli (Italy)

Hands on Advanced Bag-of-Words Models for Visual Recognition

Lamberto Ballan and Lorenzo Seidenari

MICC - University of Florence

Outline of this tutorial

- **Introduction**
  - Visual recognition problem definition
  - Bag of Words models (BoW)
  - Main drawbacks and solutions

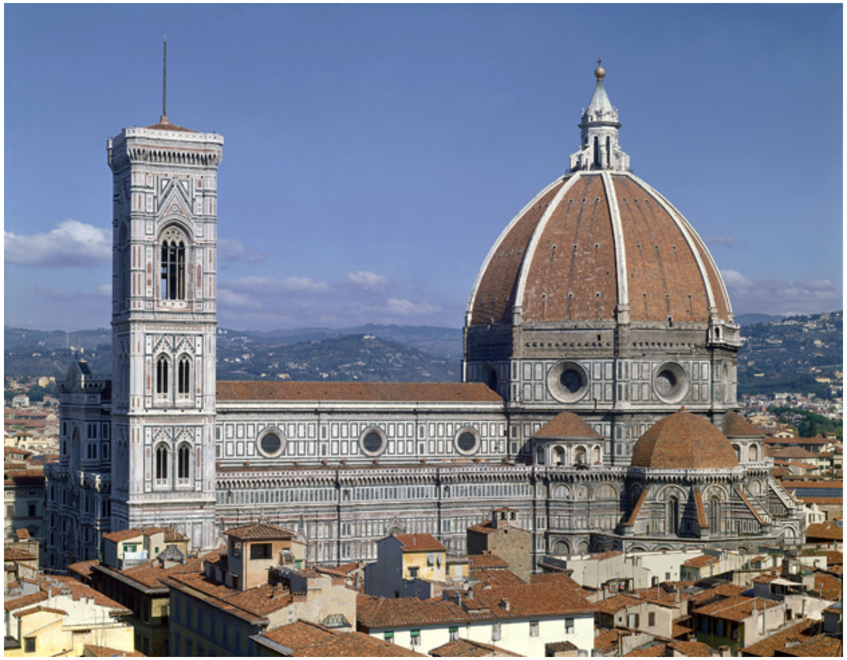
- **Session I** (practical session): Standard BoW pipeline
  - Feature sampling strategies
  - Codebook creation and feature quantization
  - Classifiers

- **Session II** (practical session): Advanced BoW models for Visual recognition
  - Feature fusion
  - Modern feature representation: reconstruction based approaches LLC
  - Spatial pooling: max pooling, spatial pyramid

Visual Recognition

Predicting the presence (absence) of an object in an image

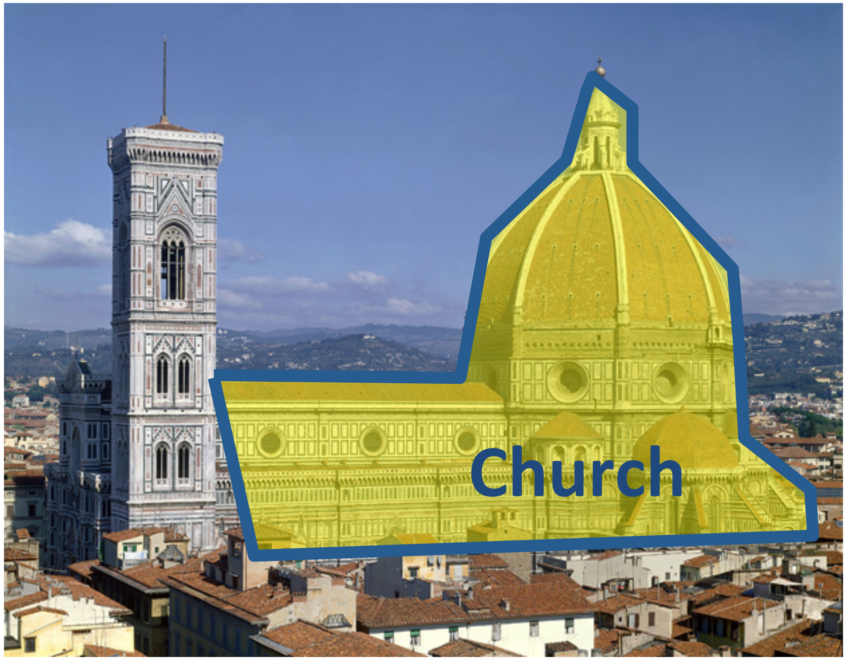
Does this image contains a **church**? [Where?]



Visual Recognition

Predicting the presence (absence) of an object in an image

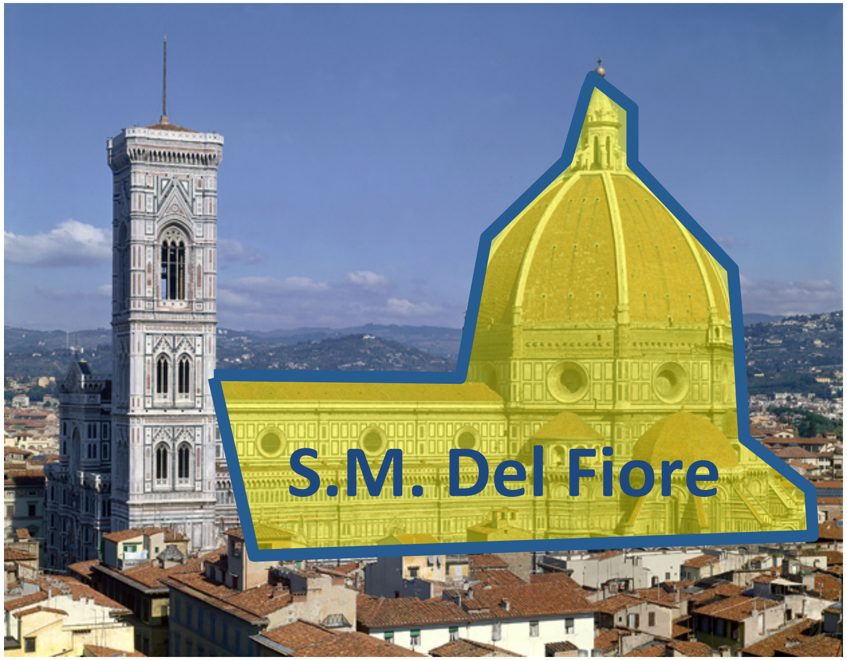
Does this image contains a **church**? [Where?]



Visual Recognition

Single instance versus category recognition

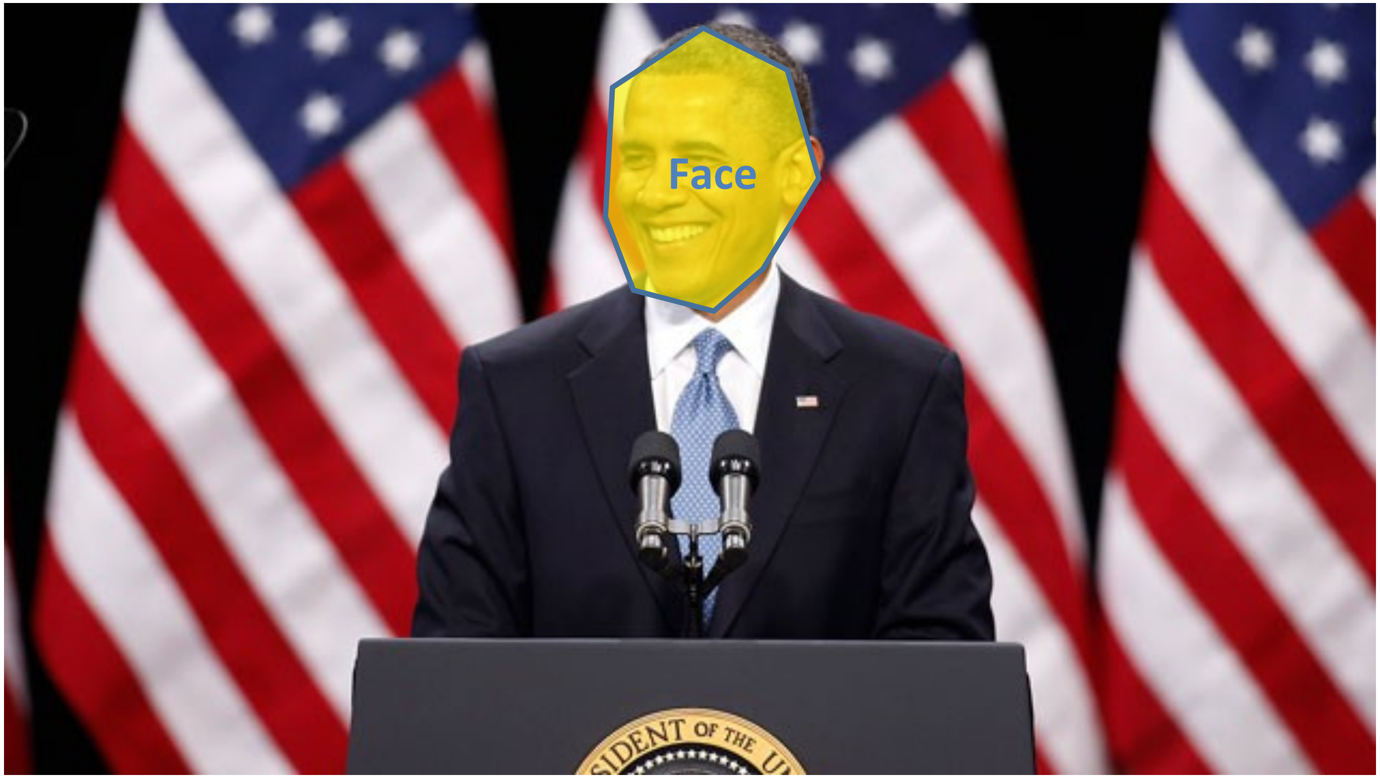
Does this image contains «Santa Maria Del Fiore Cathedral»?



Visual Recognition

Single instance versus category recognition

Does this image contain a **face?**



Visual Recognition

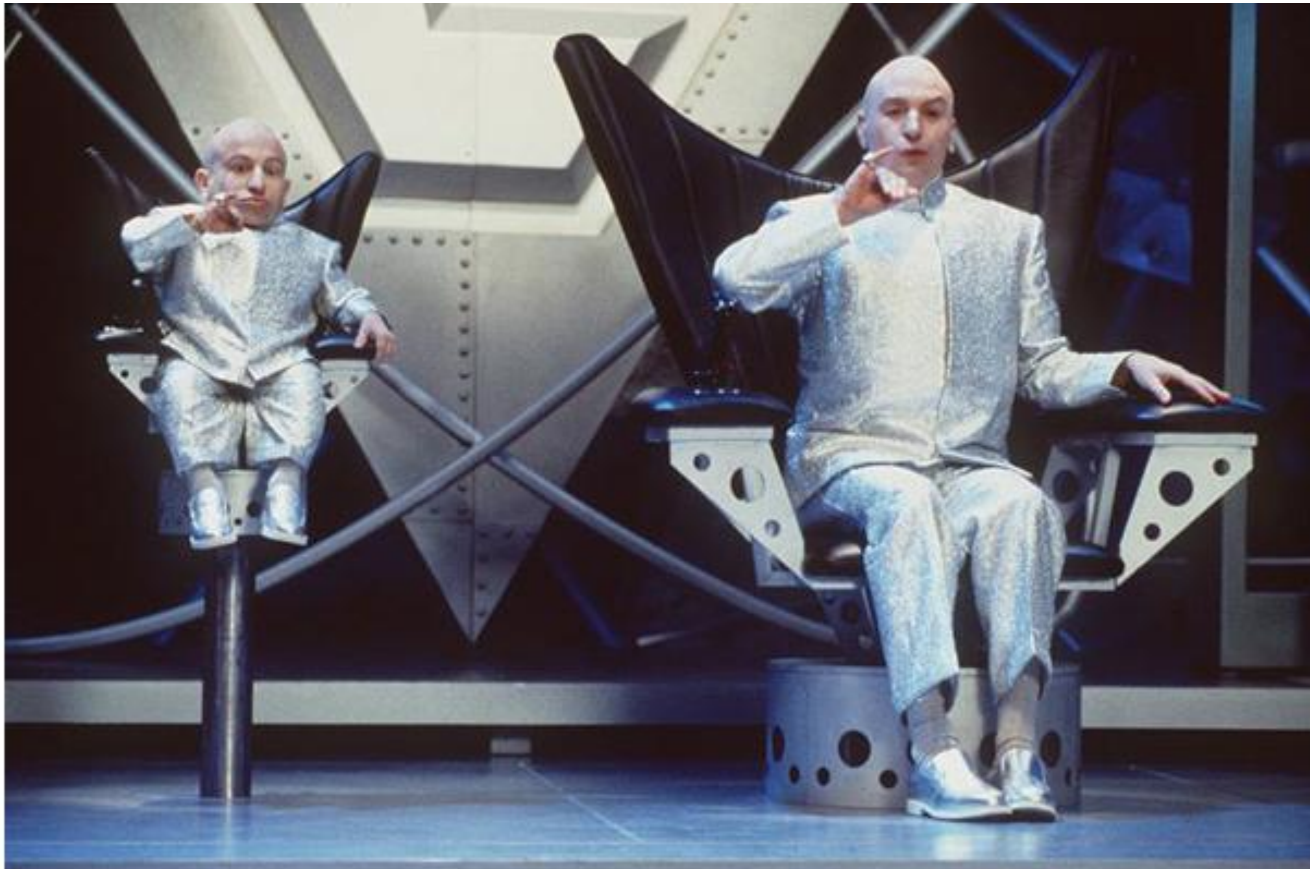
Single instance versus category recognition

Does this image contain Barak Obama?

Visual Recognition Challenges

Scale

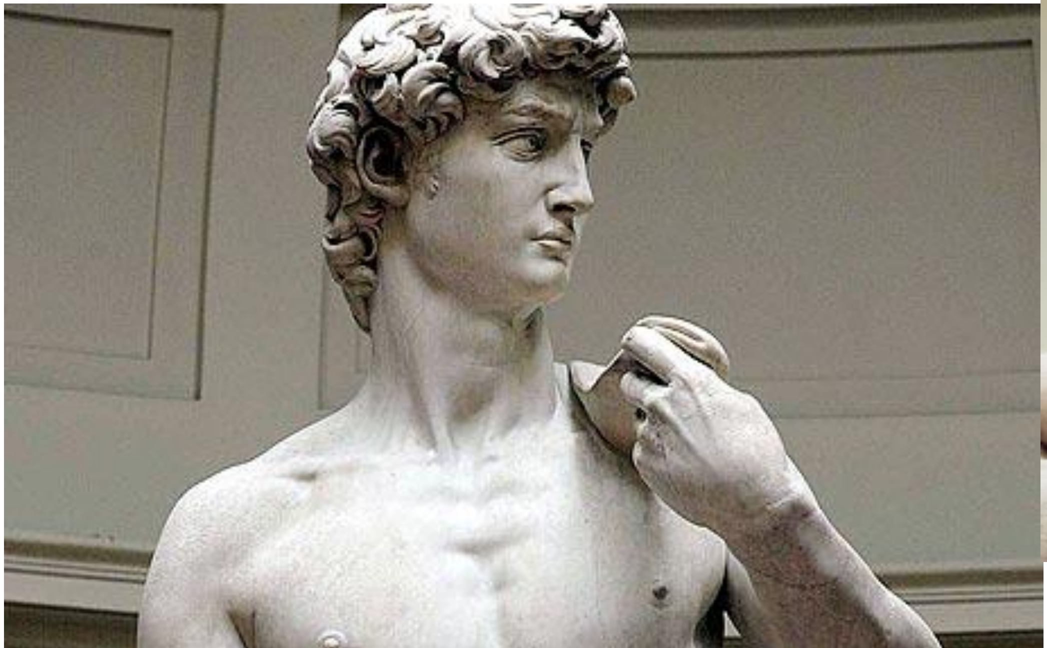
- Objects of different size
- Perspective

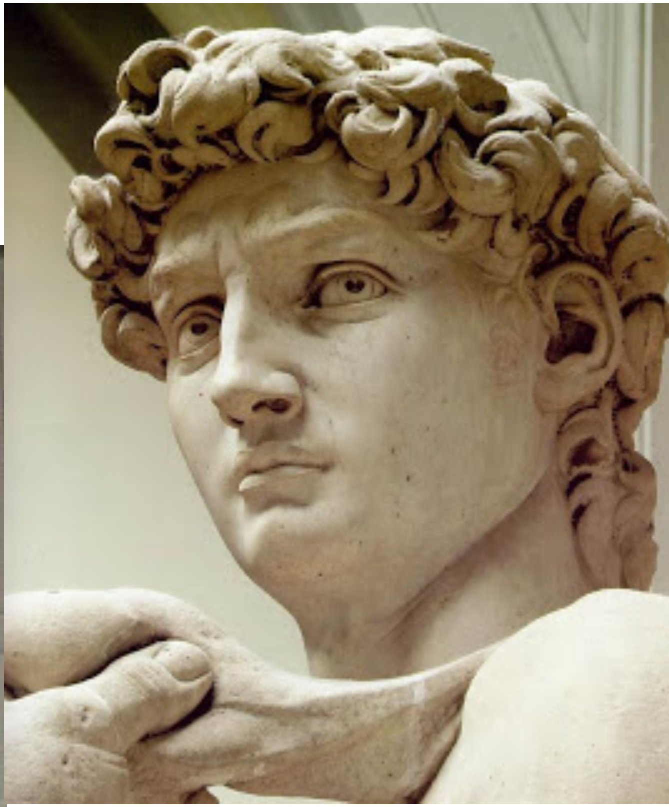


Visual Recognition Challenges

Viewpoint

- Object pose



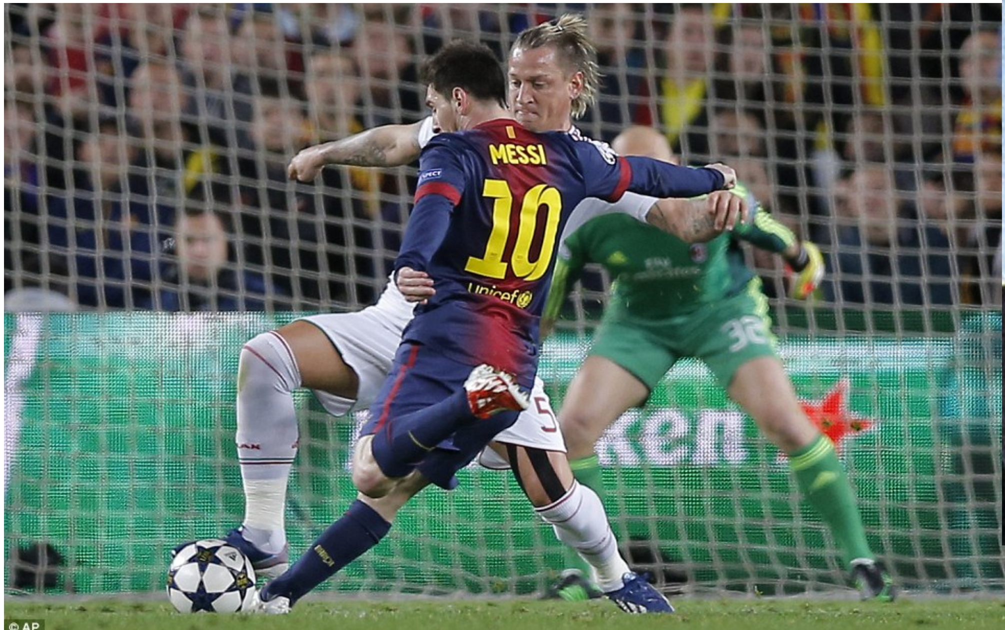


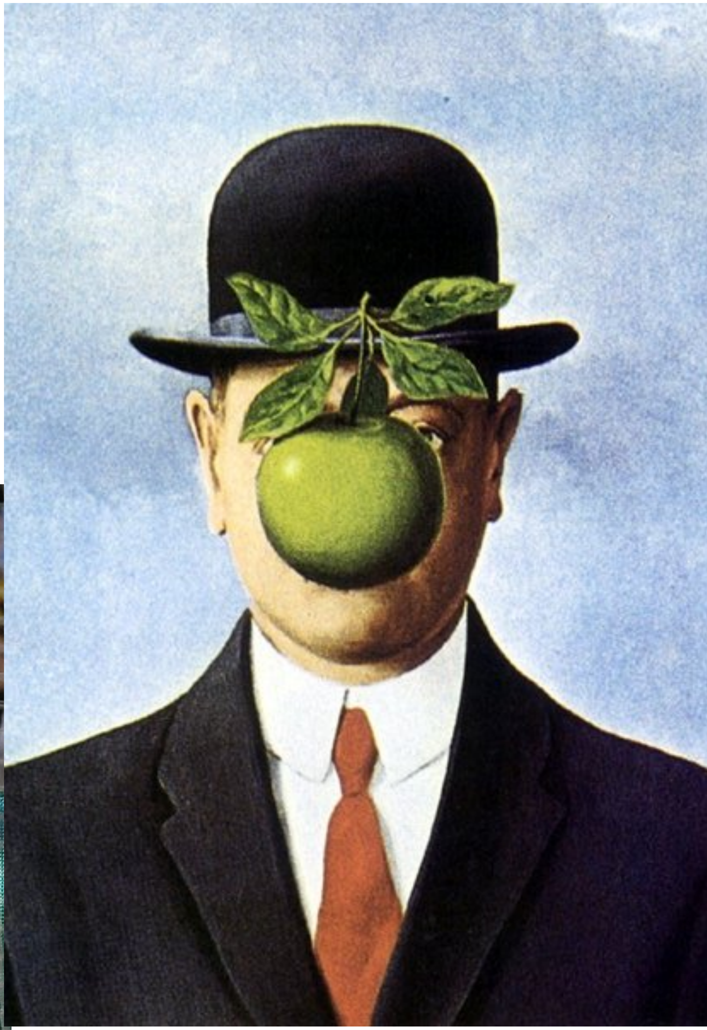
Michelangelo’s David

Visual Recognition Challenges

Occlusion

- 3D scene layout
- Articulated entities

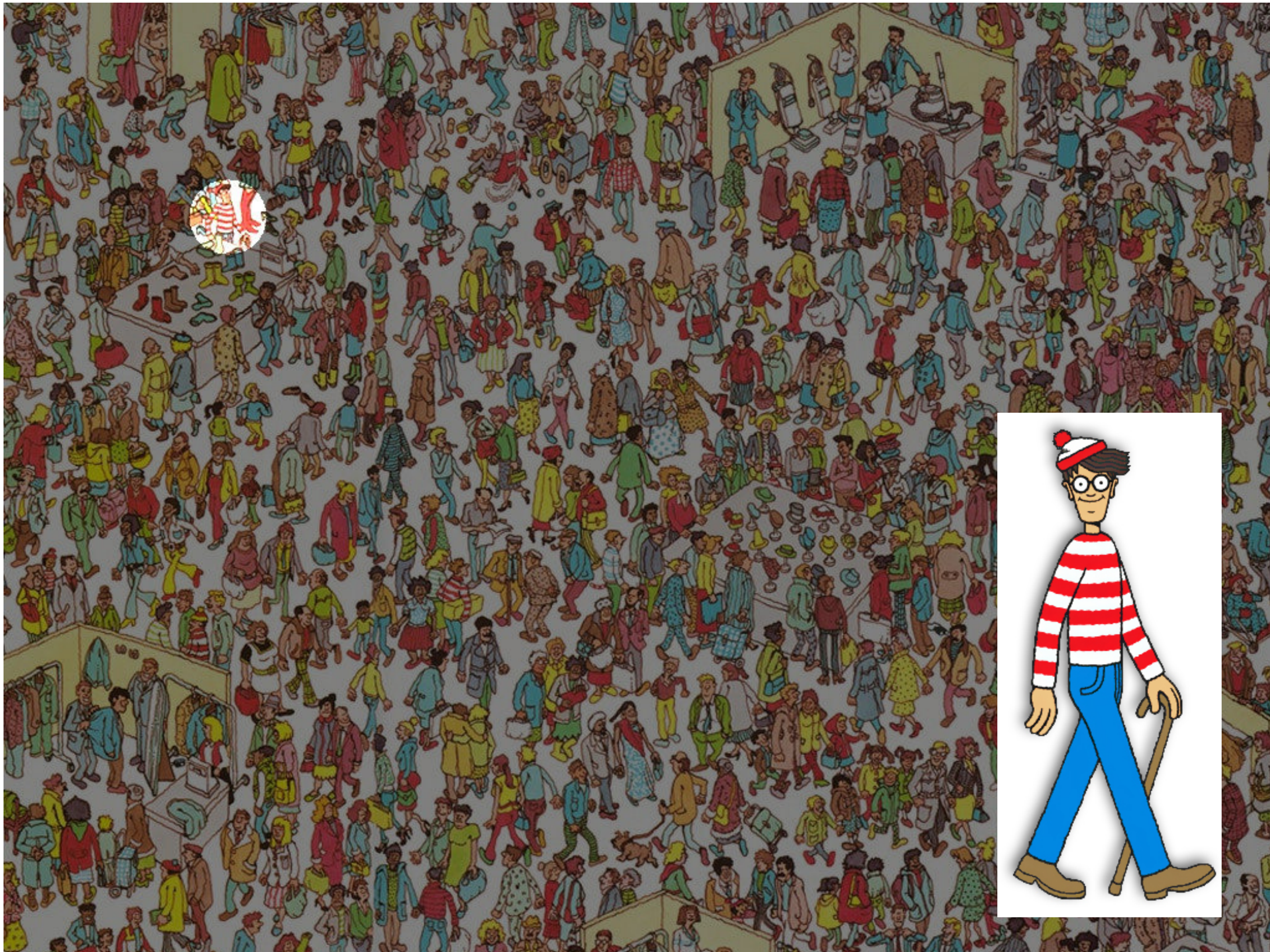




Magritte's "The Son of Man"

Visual Recognition Challenges

Clutter



Visual Recognition Challenges

Intra-class variation

- All these are chairs

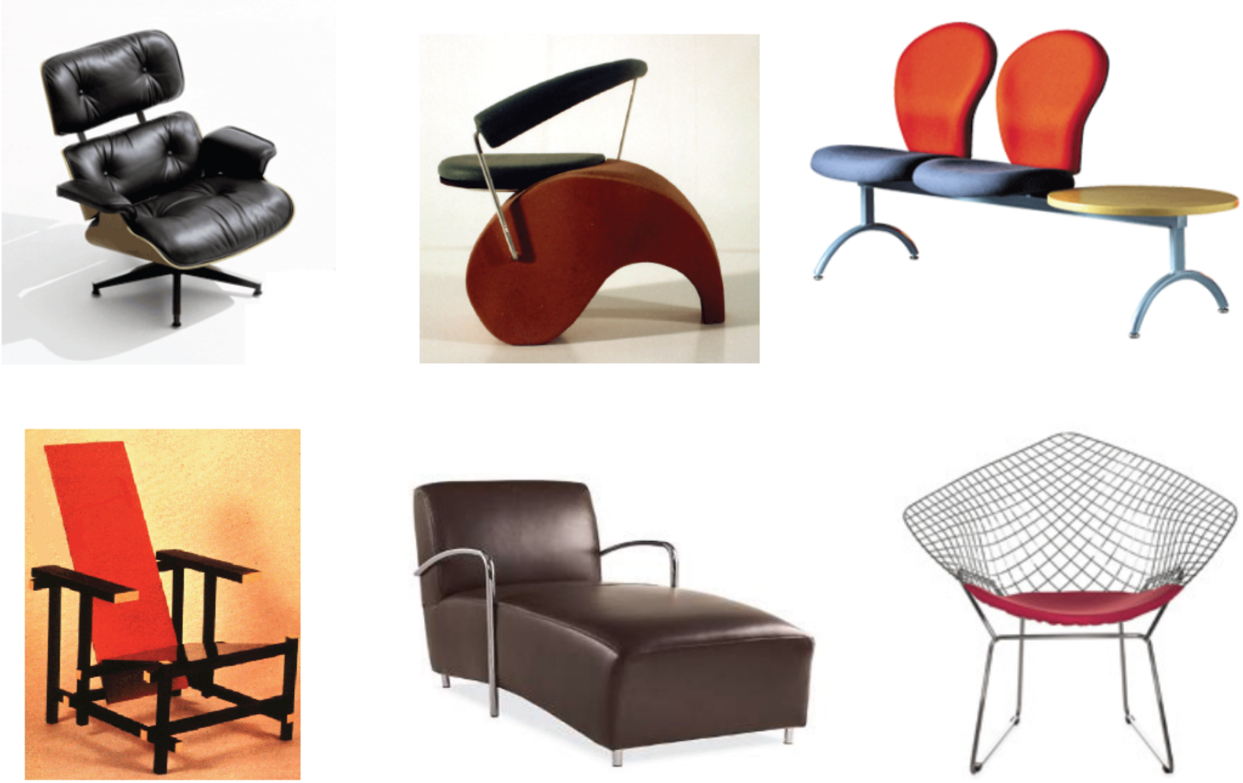


Image credit: L. Fei-Fei

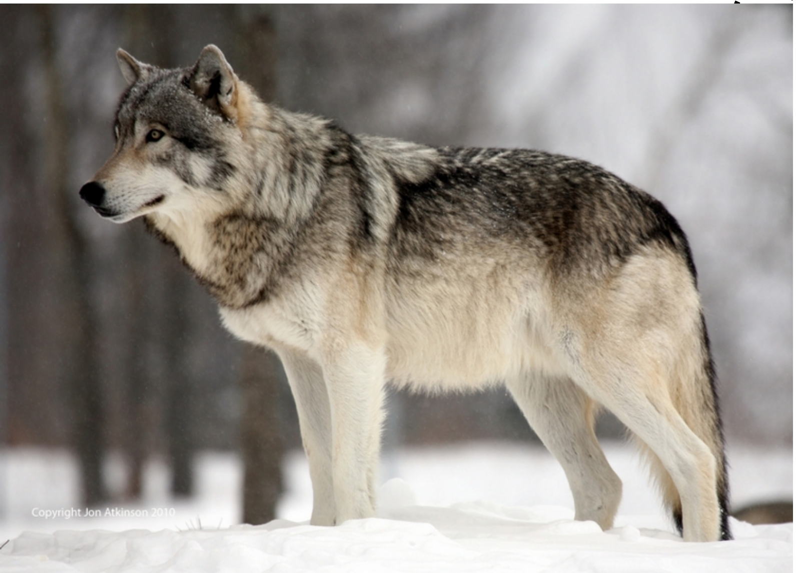
Visual Recognition Challenges

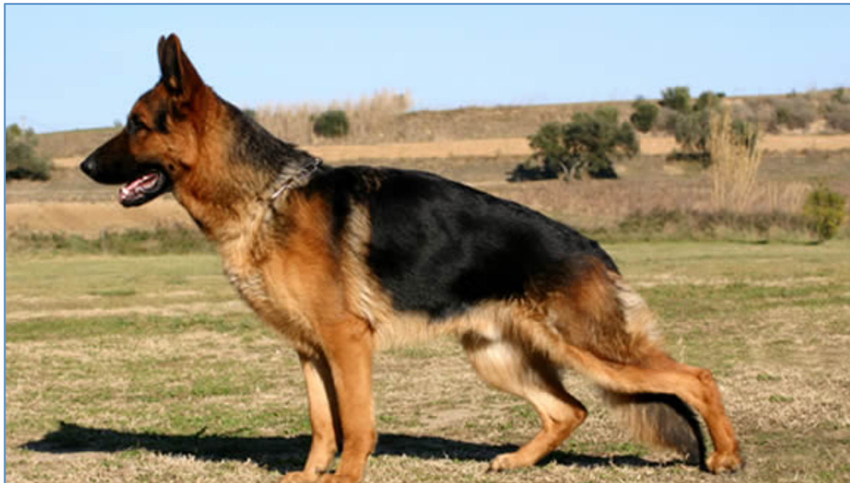
Inter-class similarity

- A dog can be very similar to a wolf

Dog

Wolf

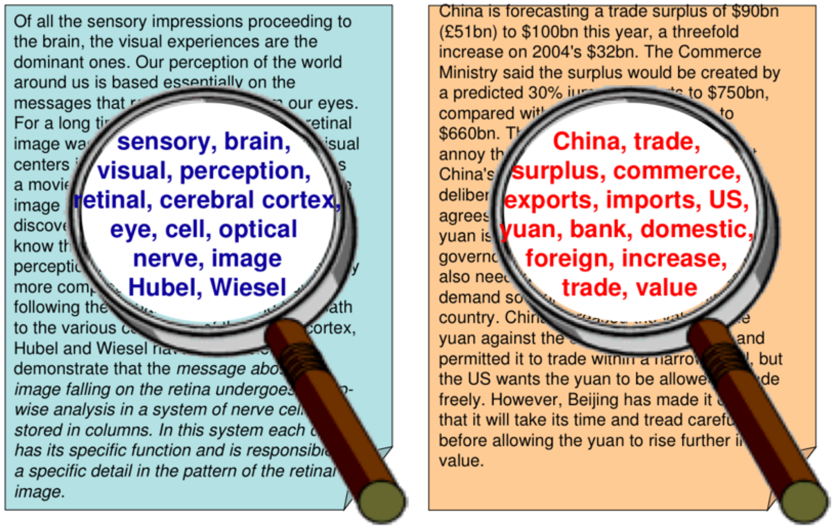




Bag-of-Words models

- **Text categorization:** the task is to assign a textual document to one or more categories based on its content

is it something about medicine/biology ? →



← *is it a document about business ?*

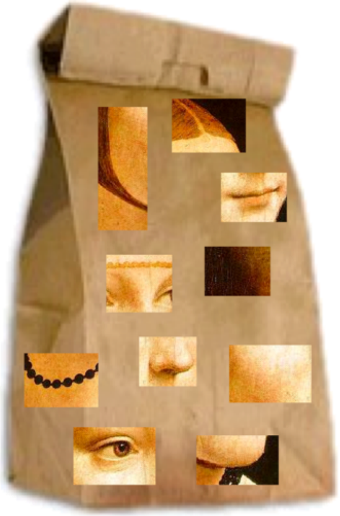
- "Bag of Words" (BoW) model, combined with advanced classification techniques, reaches state-of-the-art results
- The approach:
  - A text document is represented as an **unordered** collection of words, disregarding grammar and word order;
  - Method *ingredients* are: vocabulary, word histograms, a classifier

Image credit: L. Fei-Fei

Same approach usable with visual data

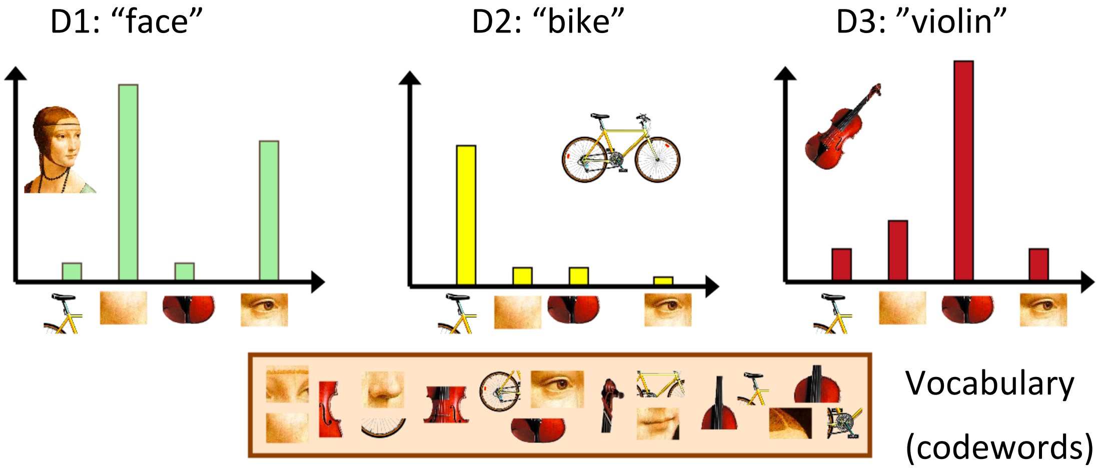
- An image can be treated as a document, and features extracted from the image are considered as the "visual words"...

image of an “object” category



bag of visual words

Bag of (visual) Words: an image is represented as an unordered collection of visual words



Vocabulary

(codewords)

Image credit: L. Fei-Fei

Pipeline

1. Feature detection (sampling) and description

2. Codebook formation and image representation

3. Learning and recognition

Pipeline

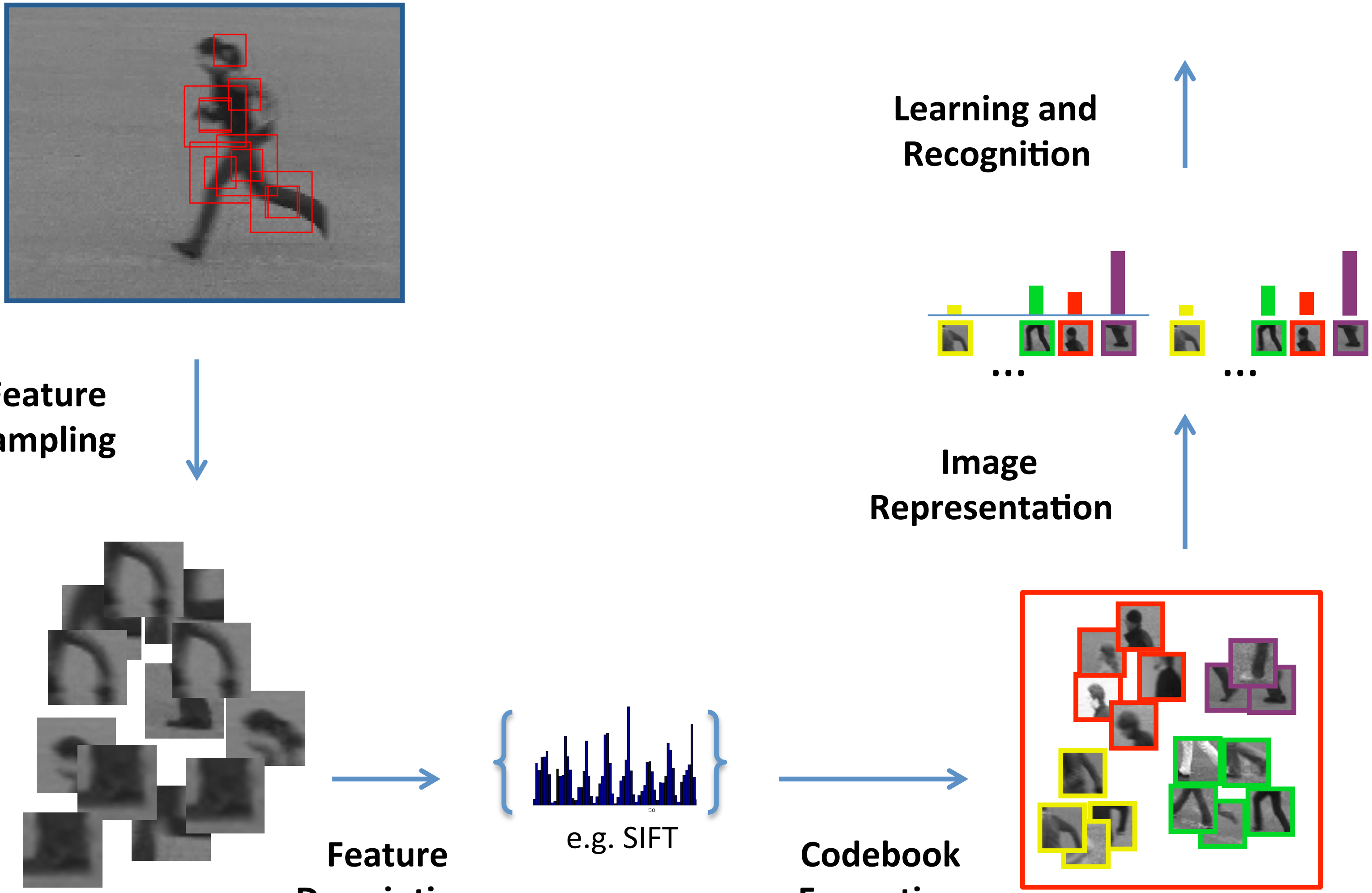
1. Feature detection (sampling) and description

2. Codebook formation and image representation

3. Learning and recognition

The focus of this tutorial

Pipeline



Category label

Learning and Recognition

Image Representation

Feature Sampling

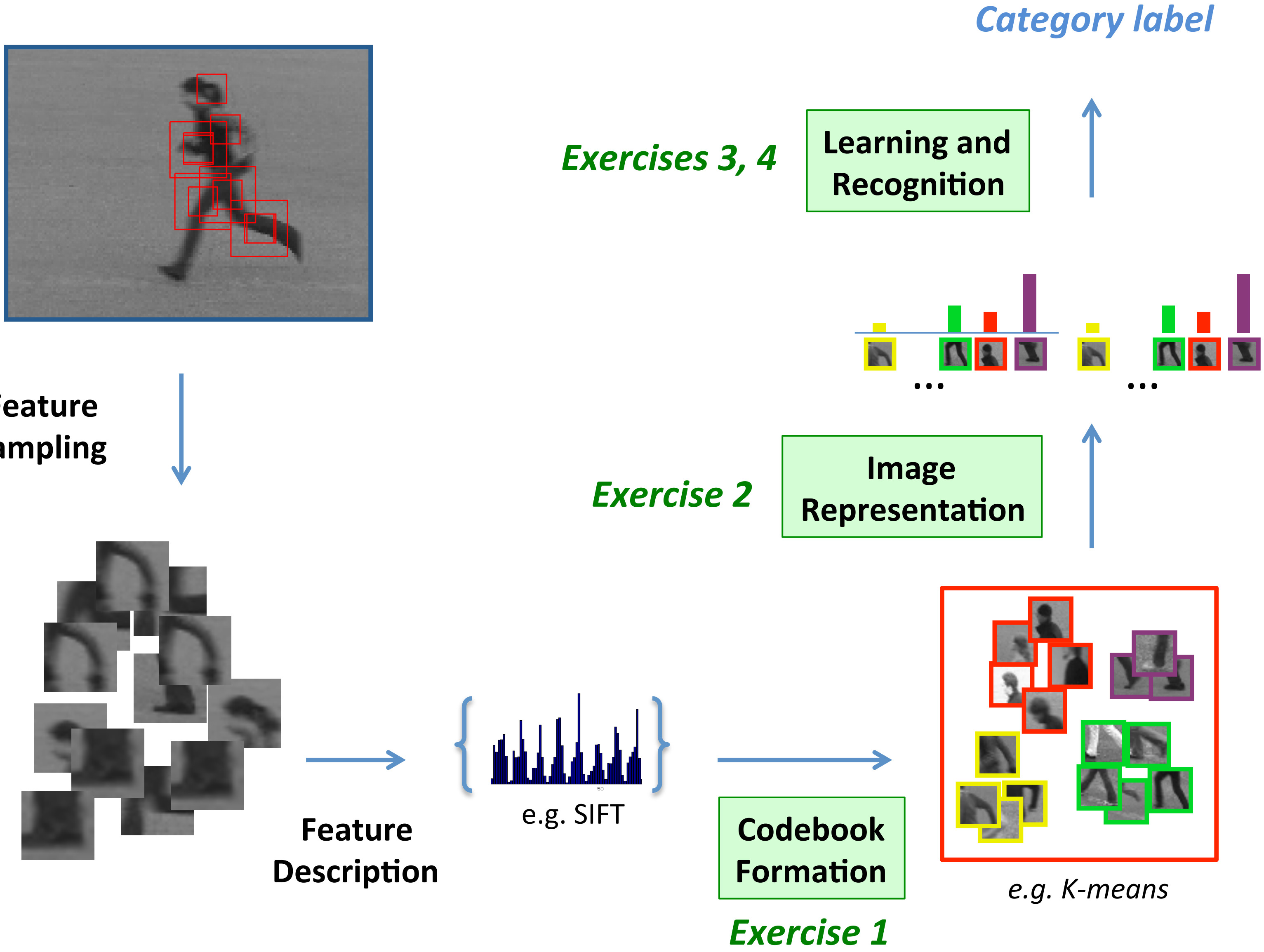
Feature Description

e.g. SIFT

Codebook Formation

e.g. K-means

Pipeline



Feature Sampling

Feature Description

e.g. SIFT

Codebook Formation

Exercise 1

e.g. K-means

Image Representation

Exercise 2

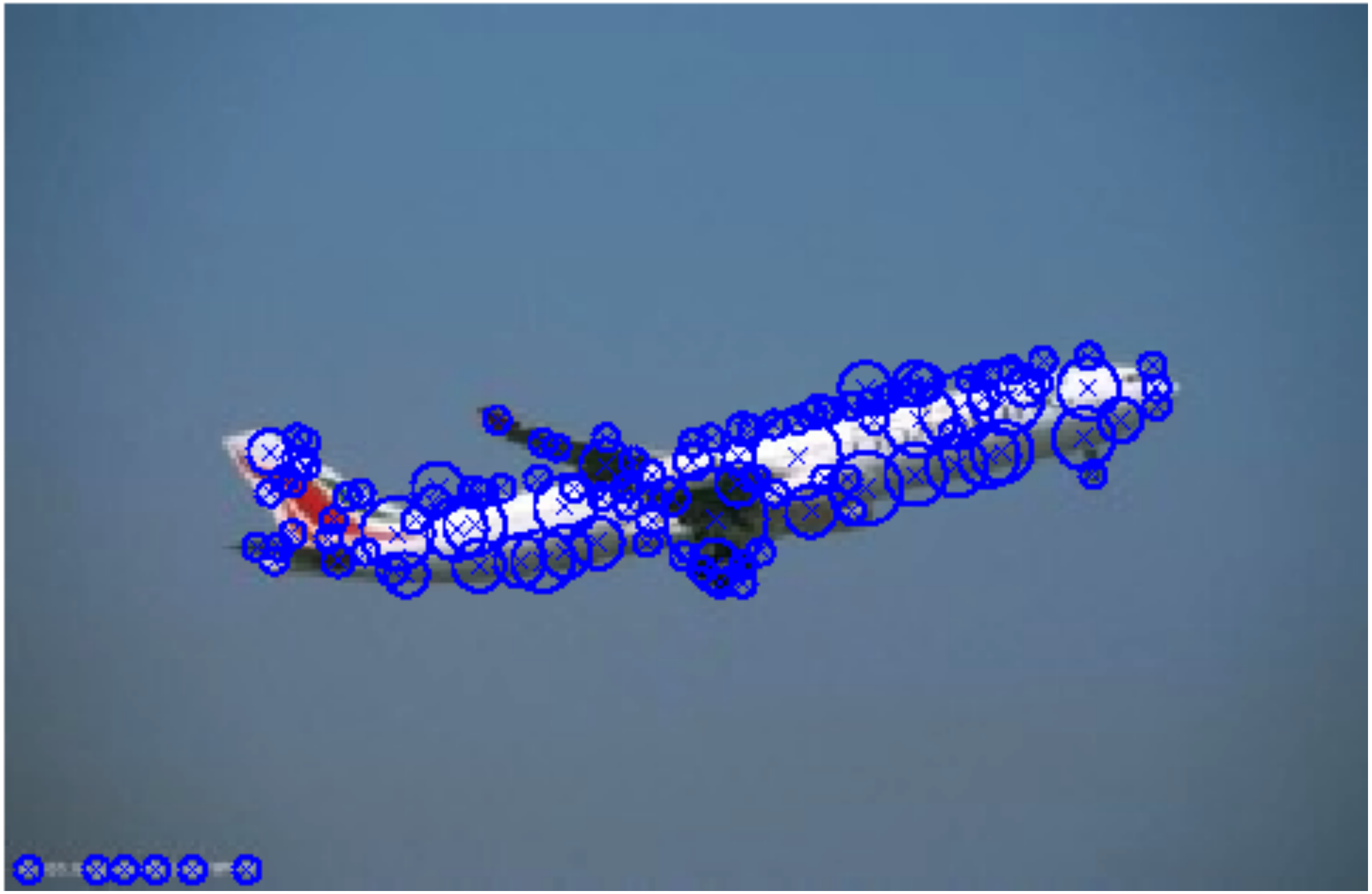
Learning and Recognition

Exercises 3, 4

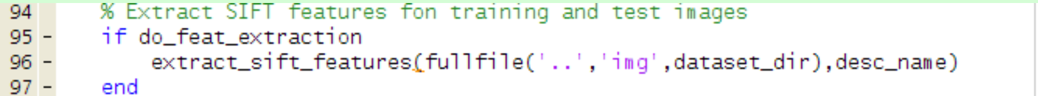
Category label

Feature Sampling

Interest operators (e.g. DoG, Harris, MSER …)

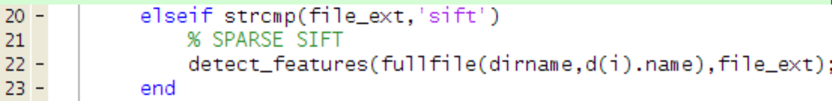


exercises.m



```
% Extract SIFT features fon training and test images
if do_feat_extraction
    extract_sift_features(fullfile('..','img',dataset_dir),desc_name)
end
```

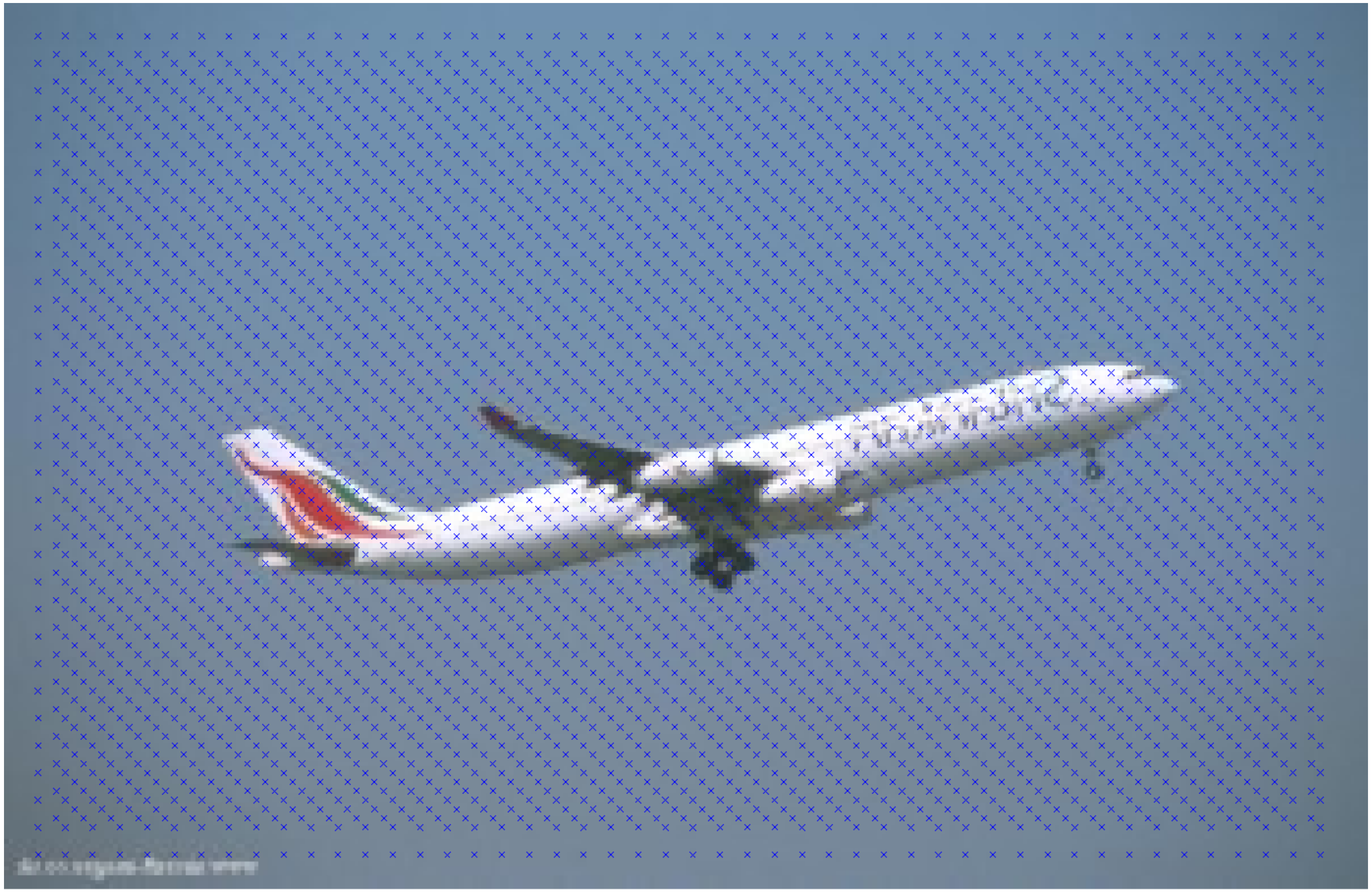
extract_sift_features.m



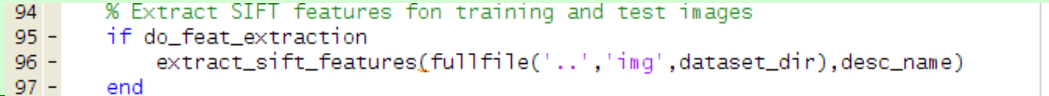
```
elseif strcmp(file_ext,'sift')
    % SPARSE SIFT
    detect_features(fullfile(dirname,d(i).name),file_ext);
end
```

Feature Sampling

Dense sampling (regular grid)

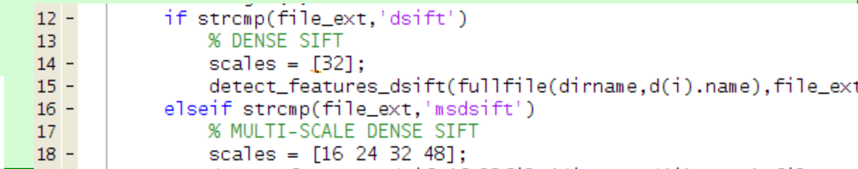


exercises.m



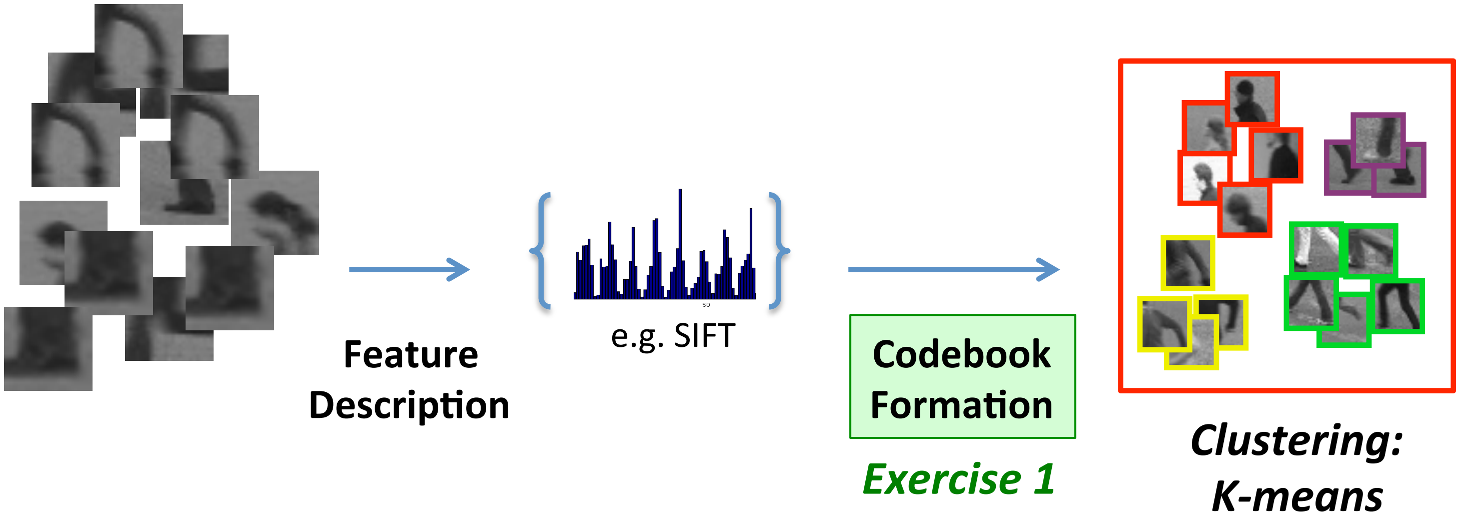
```
% Extract SIFT features fon training and test images
if do_feat_extraction
    extract_sift_features(fullfile('..','img',dataset_dir),desc_name)
end
```

extract_sift_features.m



```
if strcmp(file_ext,'dsift')
    % DENSE SIFT
    scales = [32];
    detect_features_dsift(fullfile(dirname,d(i).name),file_ext
elseif strcmp(file_ext,'msdsift')
    % MULTI-SCALE DENSE SIFT
    scales = [16 24 32 48];
```

Codebook Formation



Feature Description

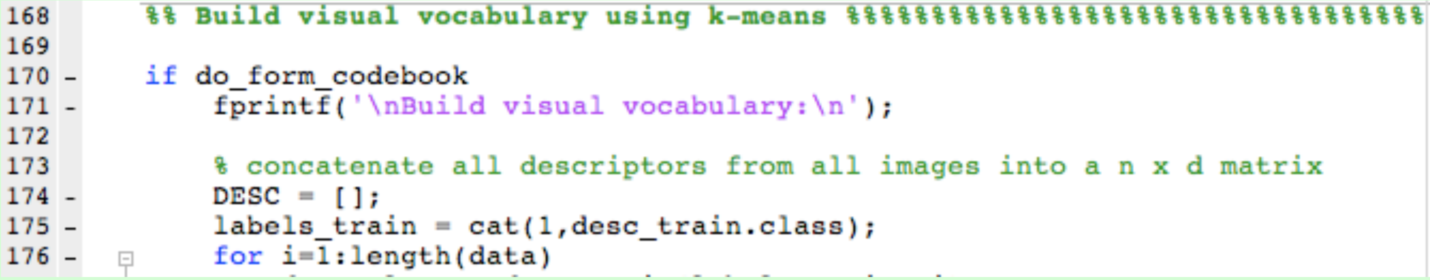
e.g. SIFT

Codebook Formation

Exercise 1

Clustering: K-means

exercises.m

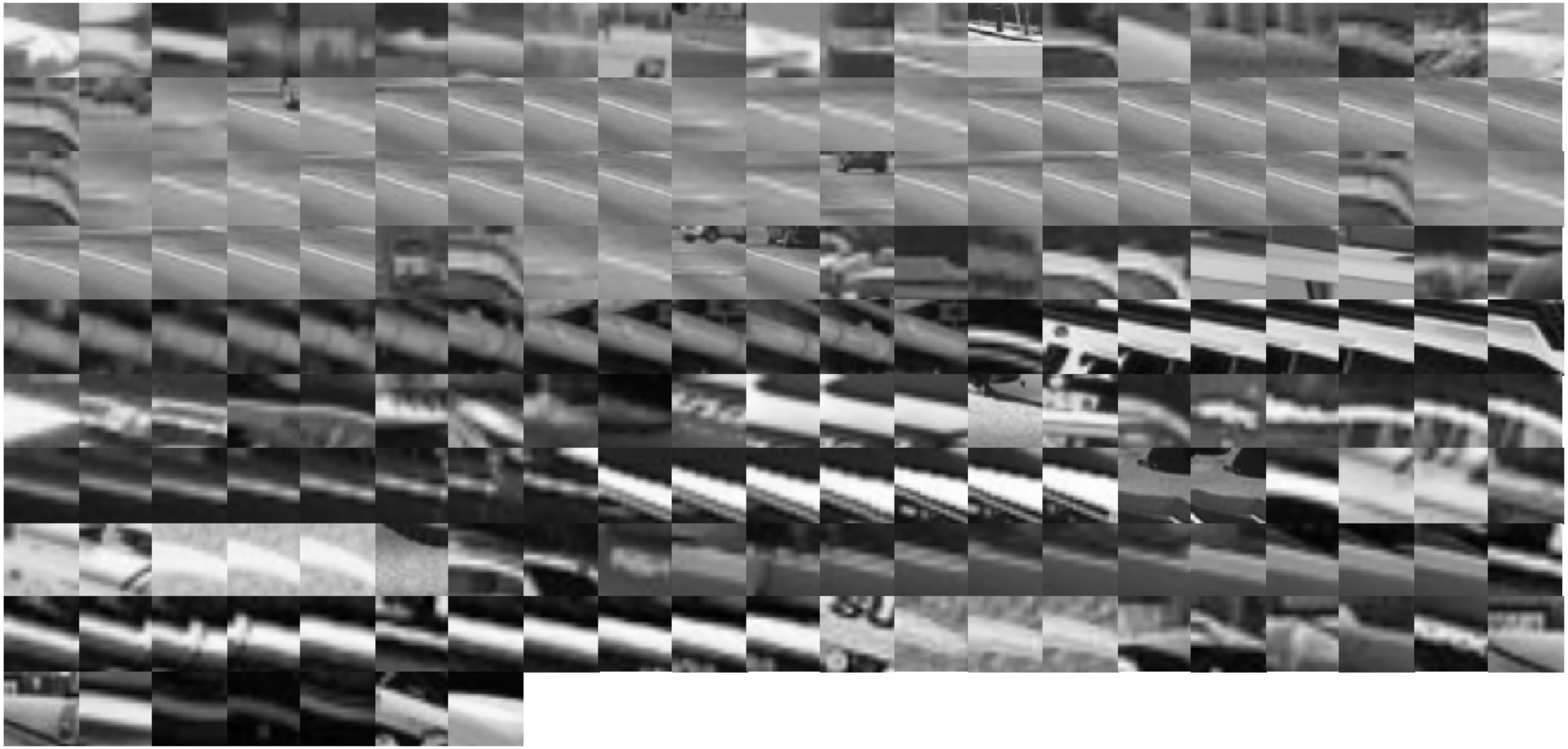


```
%% Build visual vocabulary using k-means %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if do_form_codebook
    fprintf('\nBuild visual vocabulary:\n');

    % concatenate all descriptors from all images into a n x d matrix
    DESC = [];
    labels_train = cat(1,desc_train.class);
    for i=1:length(data)
```

TODO: Exercise 1 to complete the codebook formation

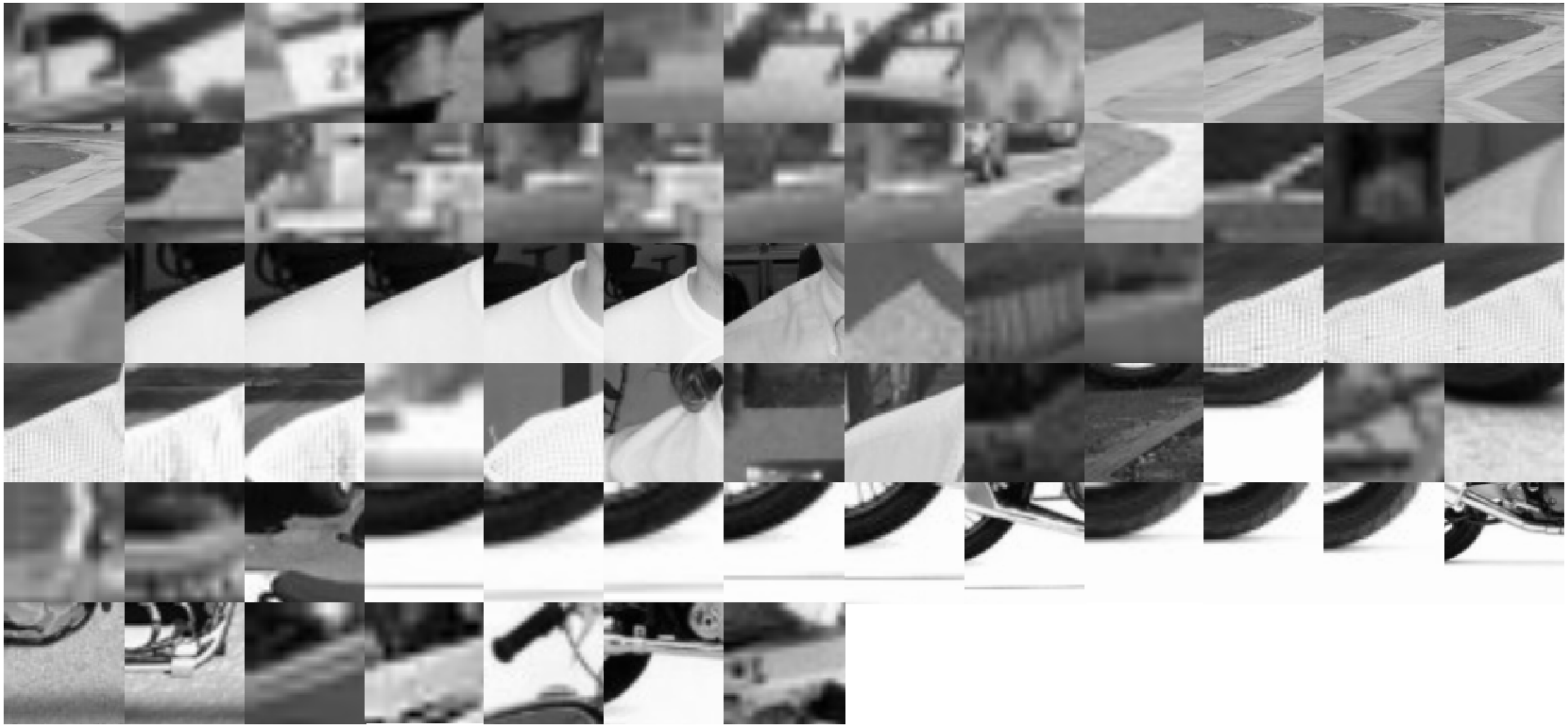
Codebook Formation

Visual Word example 1: what's inside a cluster



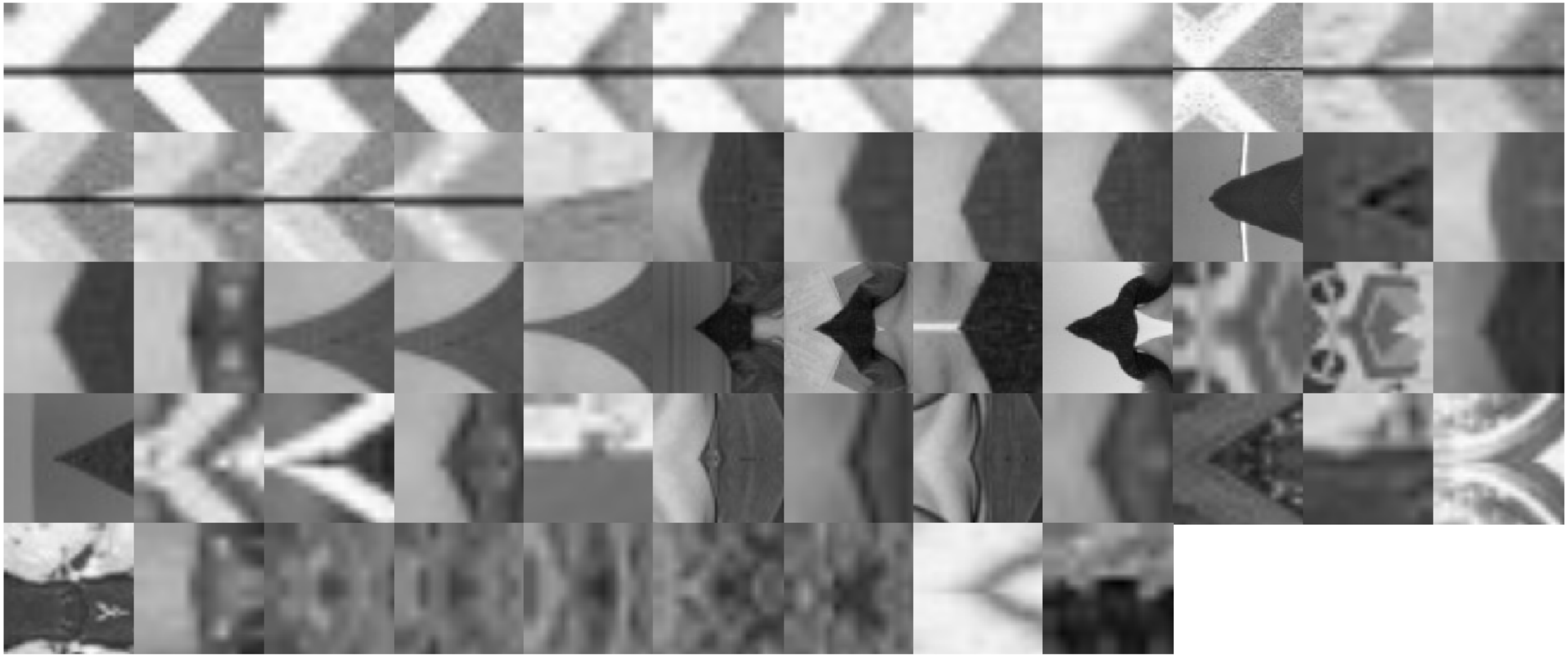
Codebook Formation

Visual Word example 2: what’s inside a cluster



Codebook Formation

Visual Word example 3: what's inside a cluster



Codebook Issues

How to choose vocabulary size?

- Too small: visual words not representative of all patches
- Too large: quantization artifacts, overfitting

Computational efficiency

- Vocabulary trees (Nister'06)

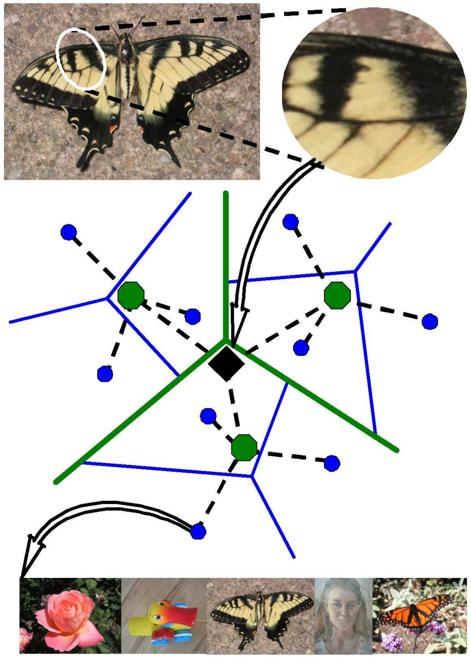
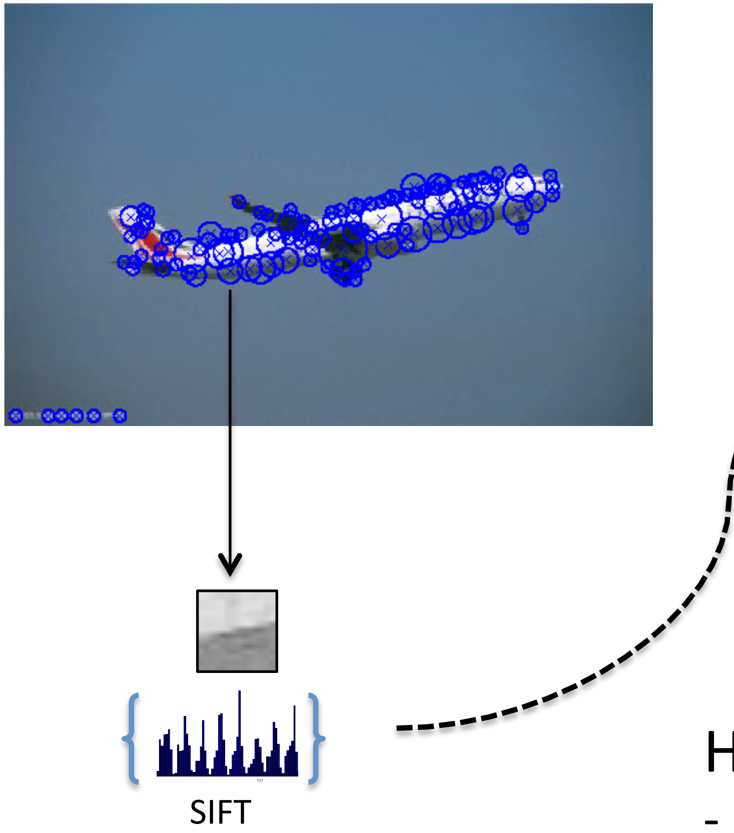
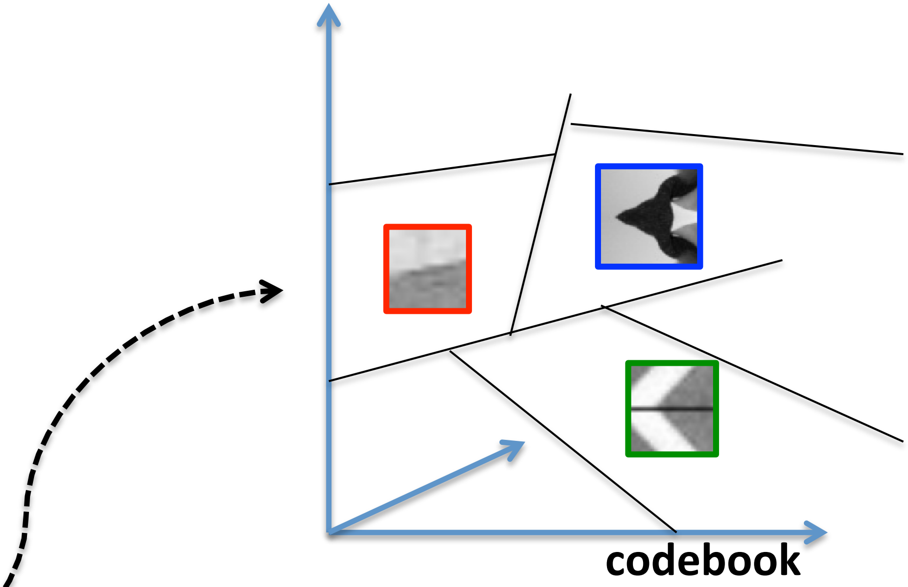


Image credit: D. Nister

Bag-of-Words Representation

Quantization: assign each feature to the most representative visual word

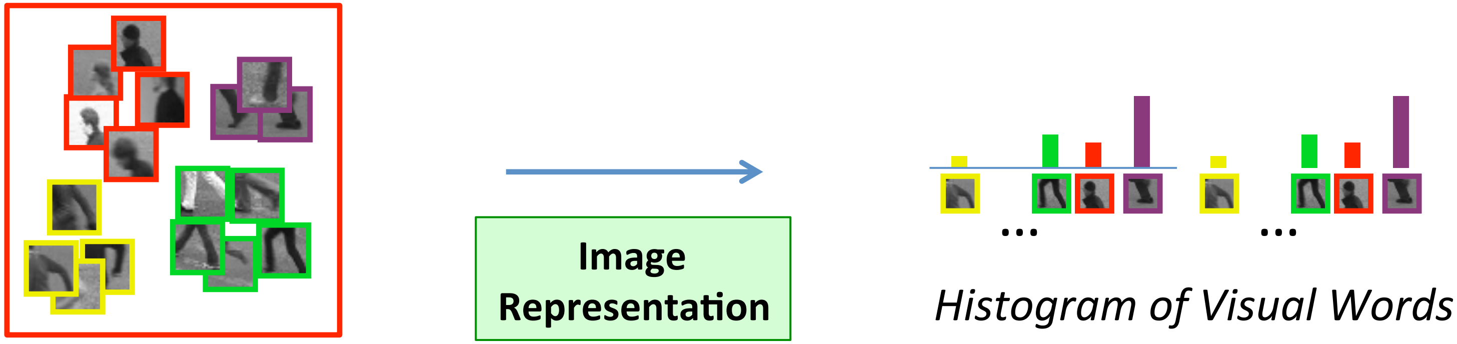




Hard assignment

- Nearest-neighbors assignment
- K-D tree search strategy

Image Representation



Visual Codebook

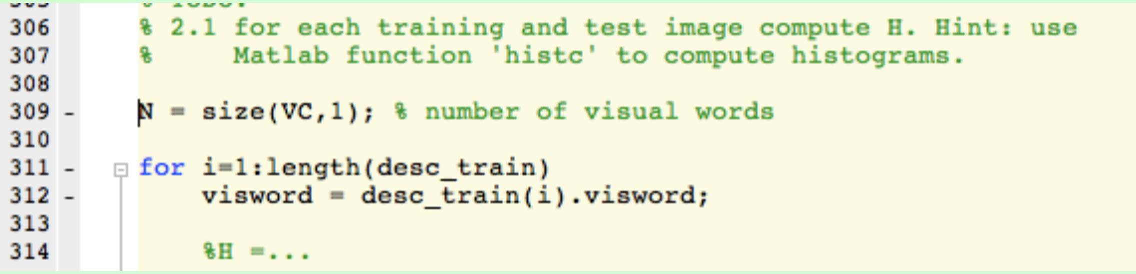
Image Representation

Histogram of Visual Words

Exercise 2

Once each feature is assigned to a visual word we can compute our image representation

exercises.m

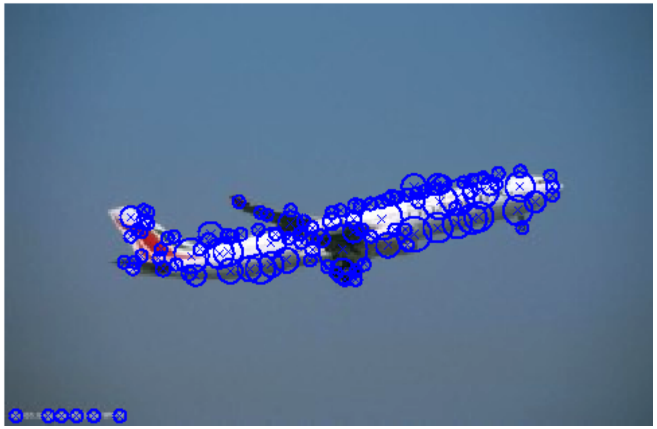


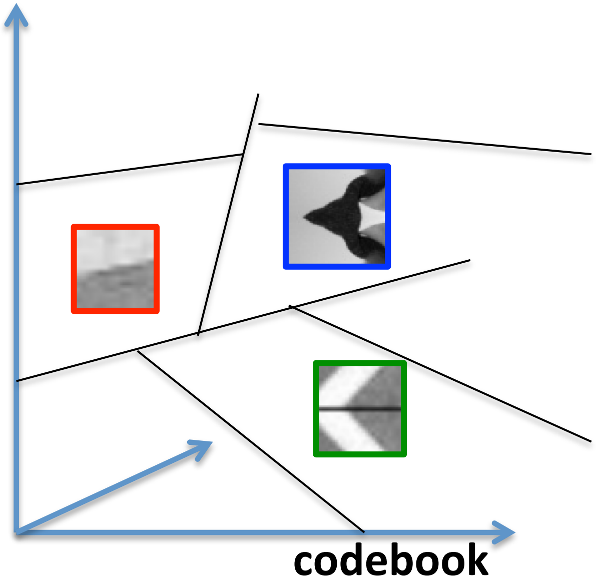
```
% TODO:
% 2.1 for each training and test image compute H. Hint: use
%     Matlab function 'histc' to compute histograms.

N = size(VC,1); % number of visual words

for i=1:length(desc_train)
    visword = desc_train(i).visword;

    %H =...
```

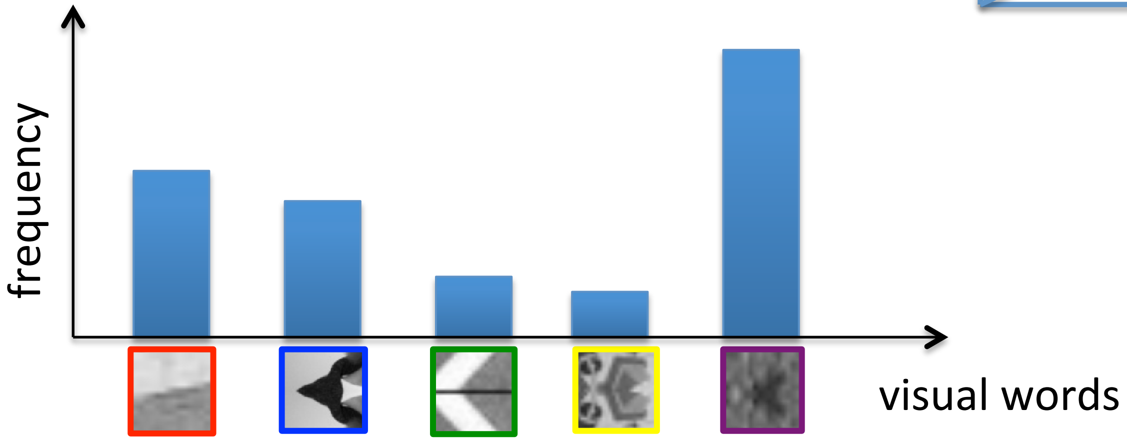
TODO: Exercise 2 to represent images as BoW histograms

Image Representation

Compute histograms of visual word frequencies







Learning and Recognition

Learn category models/classifiers from a training set

Discriminative methods (covered by this tutorial)

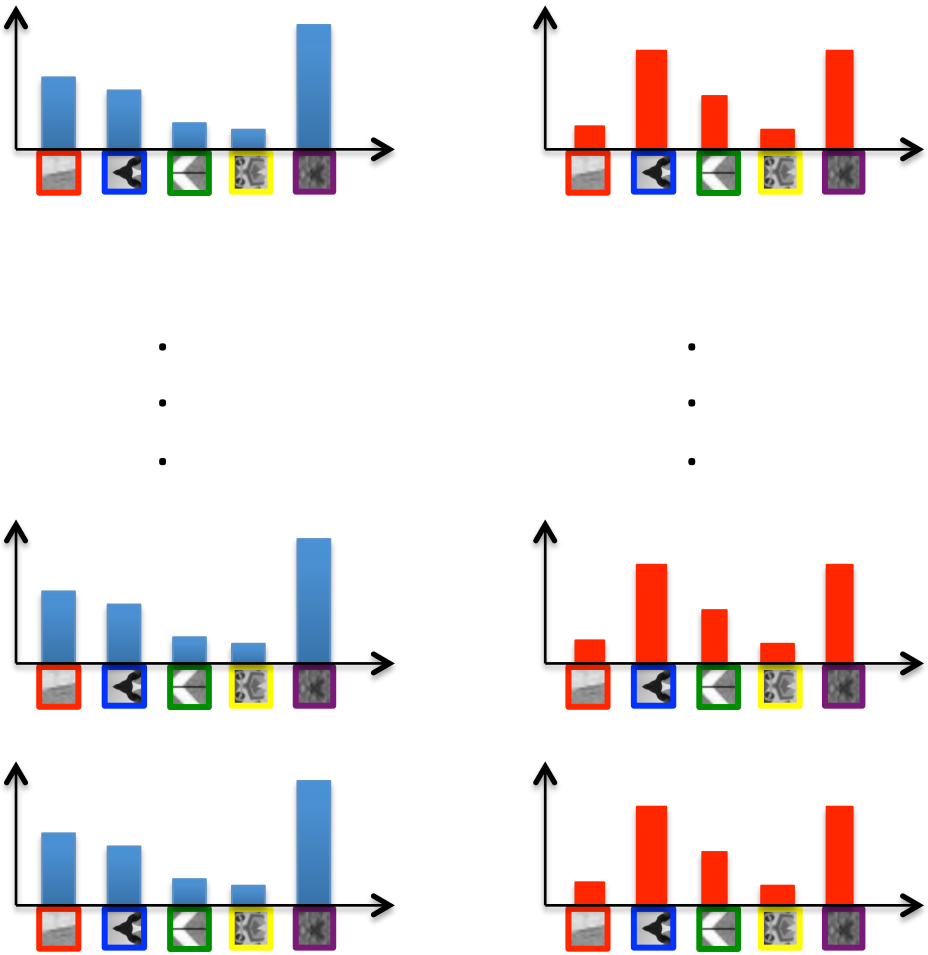
- k-NN
- SVM: linear and non-linear kernels (RBF, Intersection, …)

Generative methods

- graphical models (pLSA, LDA, …)

Discriminative Classifiers

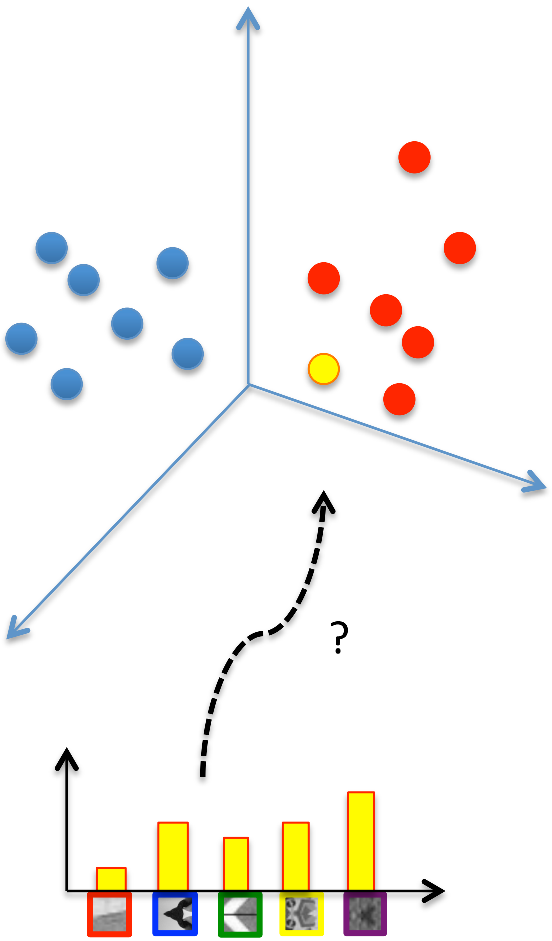
Category models



Class 1

Class 2

Model space



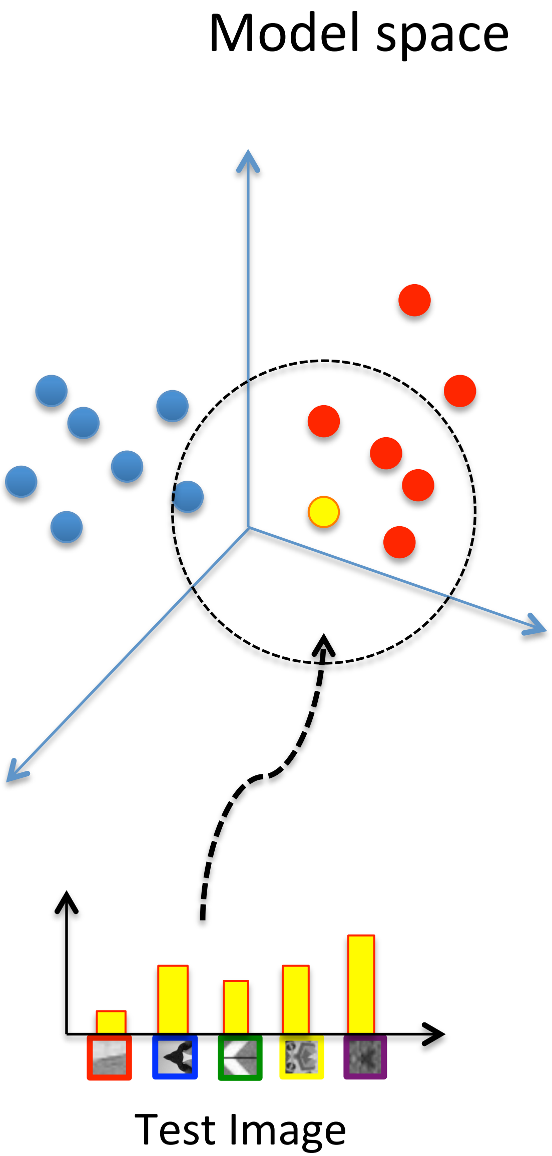
Test Image

k-Nearest Neighbors Classifier

- For a test image find the k closest points from training data
- Labels of the k points vote to classify

Works well if there is lots of data and the distance function is good

TODO: Exercise 3, NN image classification using Chi-2 distance

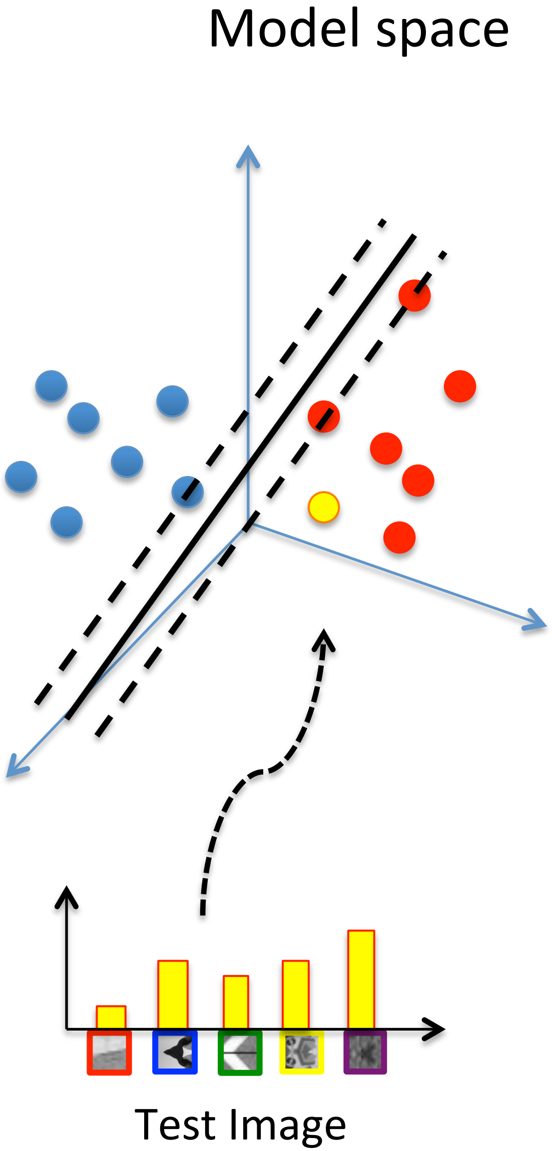


SVM Classifier

Find hyperplane that maximizes the *margin* between the positive and negative examples

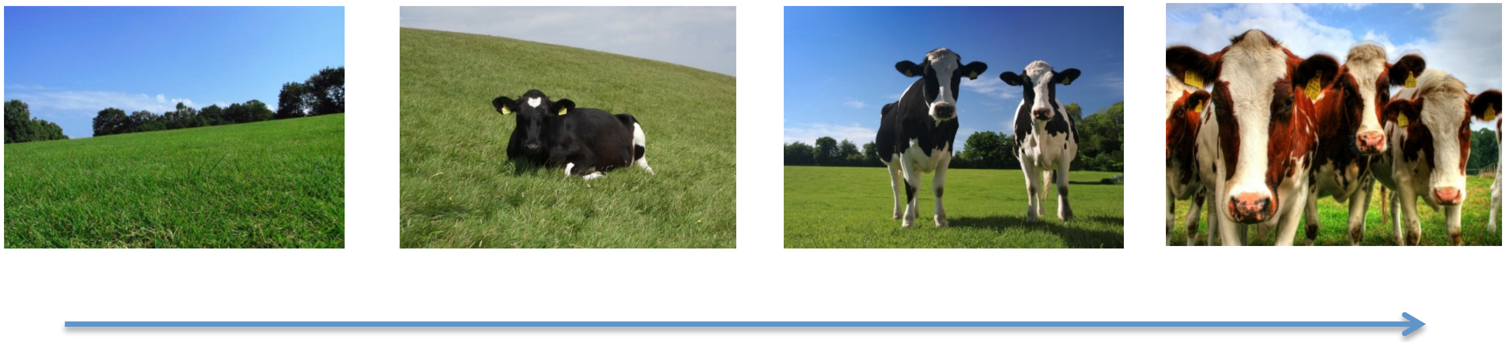
- Datasets that are linearly separable work out great
- But what if the dataset is not linearly separable? We can map it to a higher dimensional space (*lifting*)

TODO: Exercise 4, SVM image classification using different pre-computed kernels

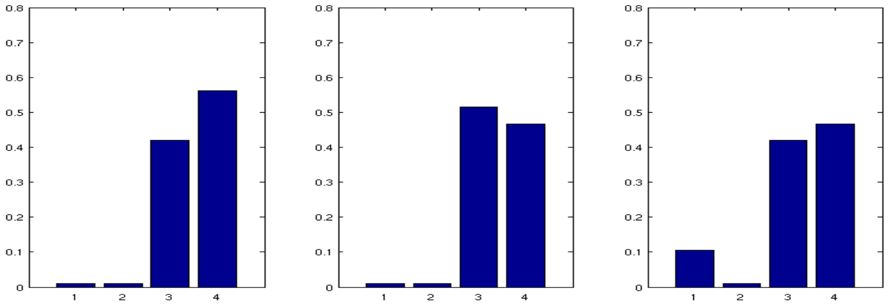


Kernels for histograms

- Linear classification with BoW histograms:
  - Each occurrence of a visual word index leads to same score increment
  - Classification score proportional to object size!



- We should **discount** small changes in large feature values



Slide credit: J. Verbeek

- Hellinger and Chi-2 distances apply a discount to large values changes

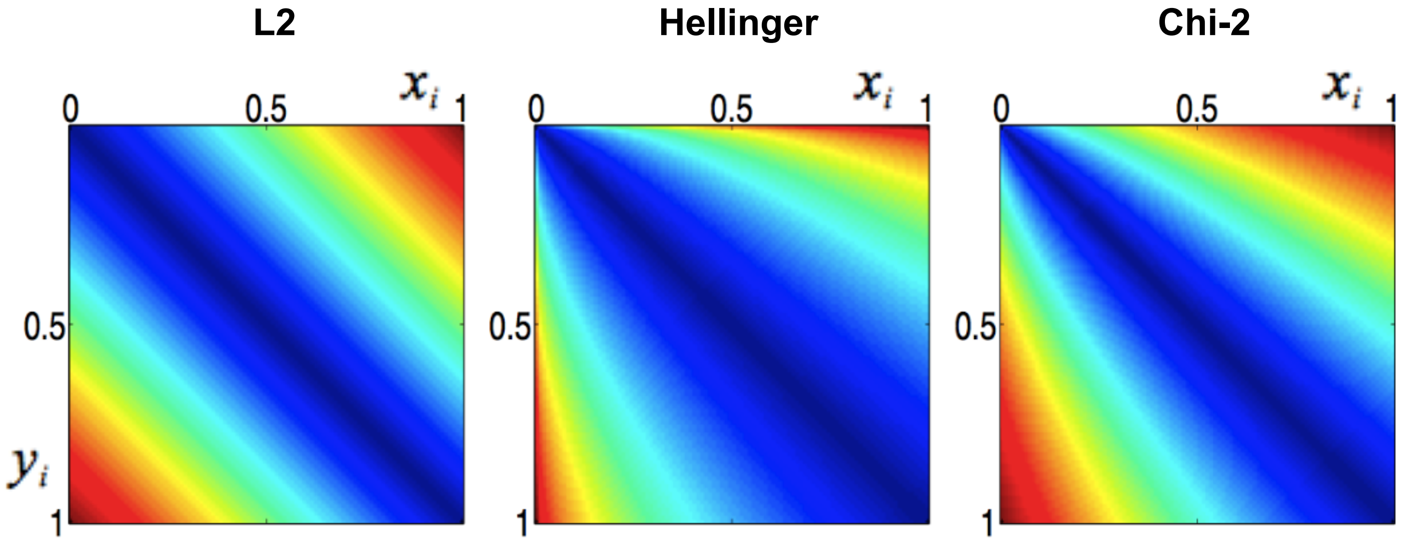
- Hellinger distance: element-wise square rooting

$$d(x,y)=(\sqrt{x}-\sqrt{y})^2 \qquad K(x,y)=\sum_i(\sqrt{x_i}-\sqrt{y_i})^2$$

- Chi-2 distance between vectors

$$d(x,y)=\frac{1}{2}\frac{(x-y)^2}{x+y} \qquad K(x,y)=\exp(-\gamma\sum_i\frac{(x_i-y_i)^2}{x_i+y_i})$$

- Discounting effect of distances:



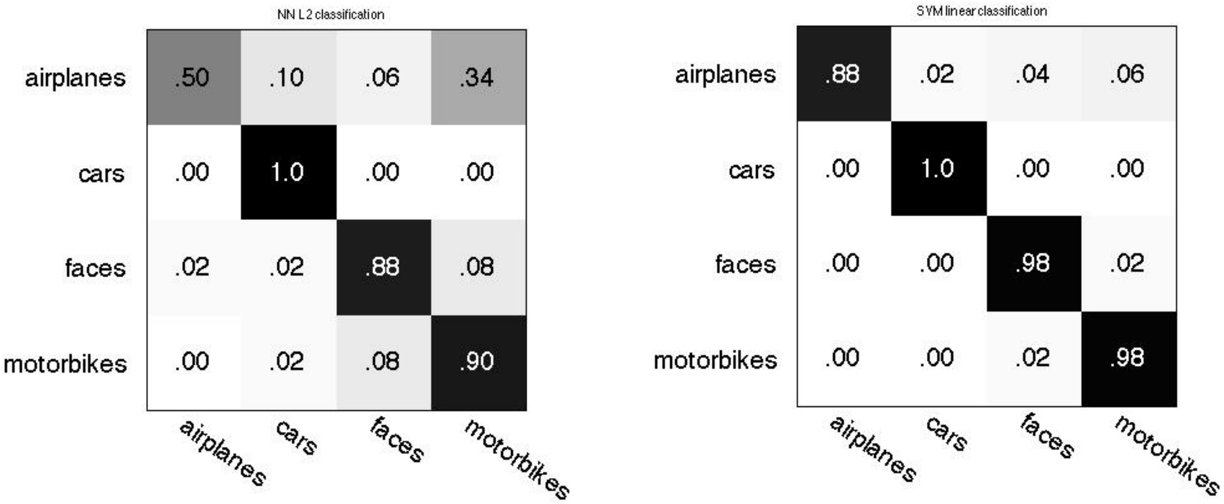
Slide credit: J. Verbeek

Experiments

- Two different datasets (both are subsets of Caltech-101)
  - **4 Object Categories:** *faces, airplanes, cars, motorbikes*
  - **15 Object Categories:** *bonsai, butterfly, crab, elephant, euphonium, faces, grandpiano, joshuatree, leopards, lotus, motorbikes, schooner, stopsign, sunflower, watch*

- Experimental protocol
  - for each class 30 images are selected for train and (up to) 50 for test
  - results are reported by measuring *accuracy*

*Confusion matrices obtained on the **4 Objects** dataset (using NN and linear SVM)*



NN L2 classification

	airplanes	cars	faces	motorbikes
airplanes	.50	.10	.06	.34
cars	.00	1.0	.00	.00
faces	.02	.02	.88	.08
motorbikes	.00	.02	.08	.90

SVM linear classification

	airplanes	cars	faces	motorbikes
airplanes	.88	.02	.04	.06
cars	.00	1.0	.00	.00
faces	.00	.00	.98	.02
motorbikes	.00	.00	.02	.98