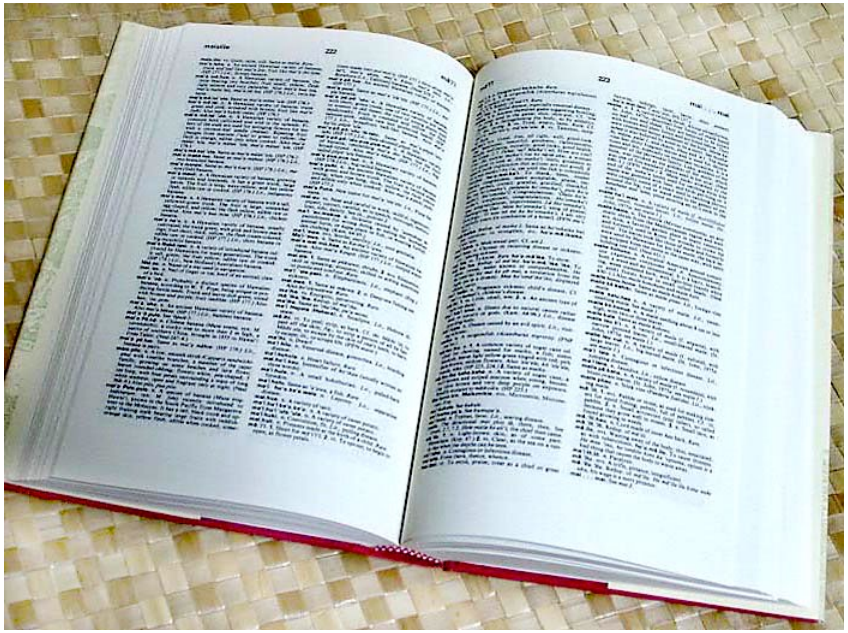
Bag of Visual Words

Thanks to Svetlana Lazebnik and Andrew Zisserman for the use of some slides

How many categories?



10,000-30,000

OBJECTS

- ANIMALS
  - .....
  - VERTEBRATE
    - MAMMALS
      - COW
      - HORSE
    - BIRDS
      - PIGEON
- PLANTS
- INANIMATE
  - NATURAL
  - MAN-MADE
    - CHAIR

History

1960s – early 1990s: geometry

1990s: appearance

Mid-1990s: sliding window

Late 1990s: local features

Early 2000s: parts-and-shape models

Mid-2000s: bags of features

Present trends: data-driven methods, context

Local features

D. Lowe (1999, 2004)

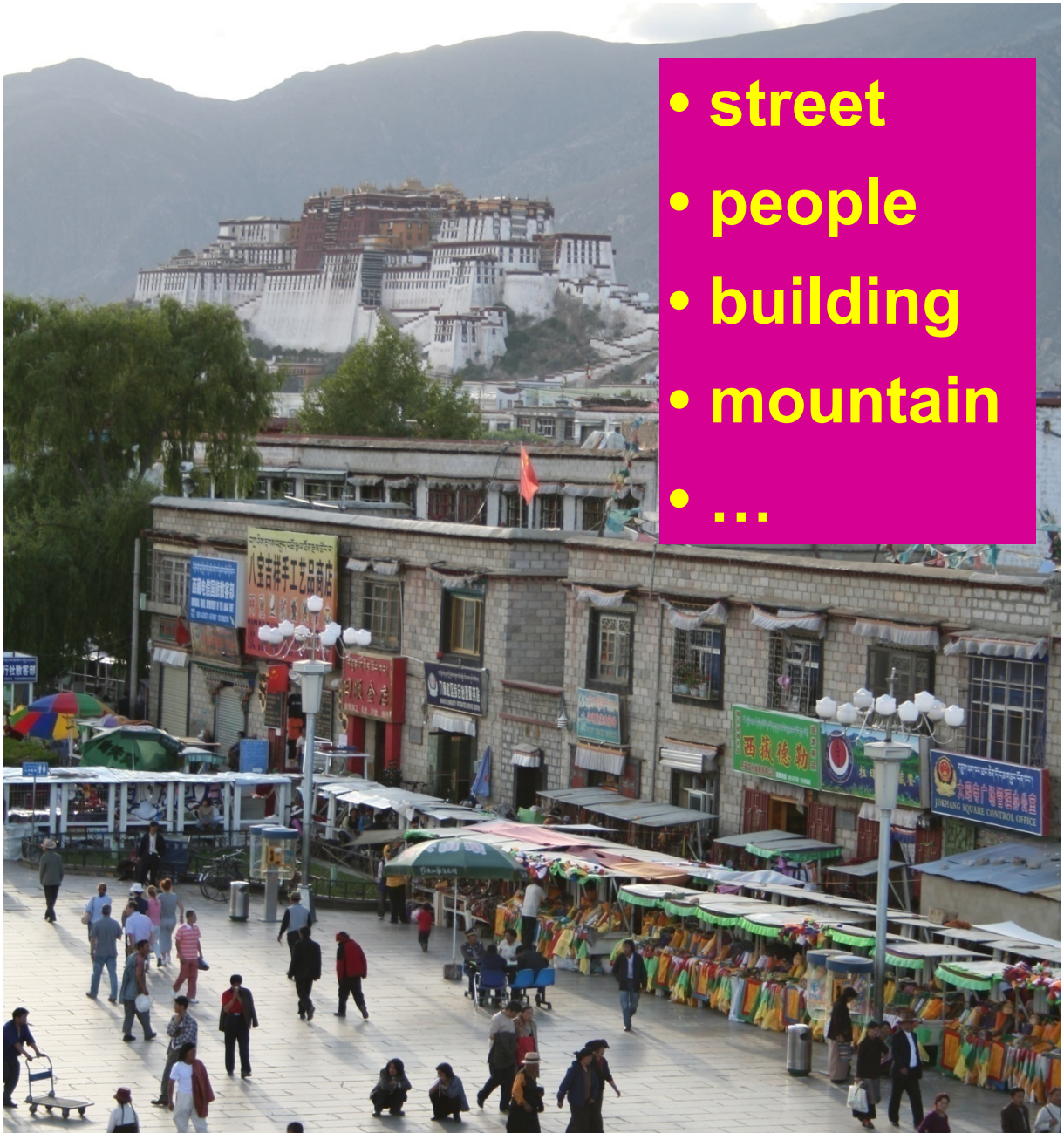
RECOGNITION AS AN APPLICATION OF MACHINE LEARNING

Common recognition tasks

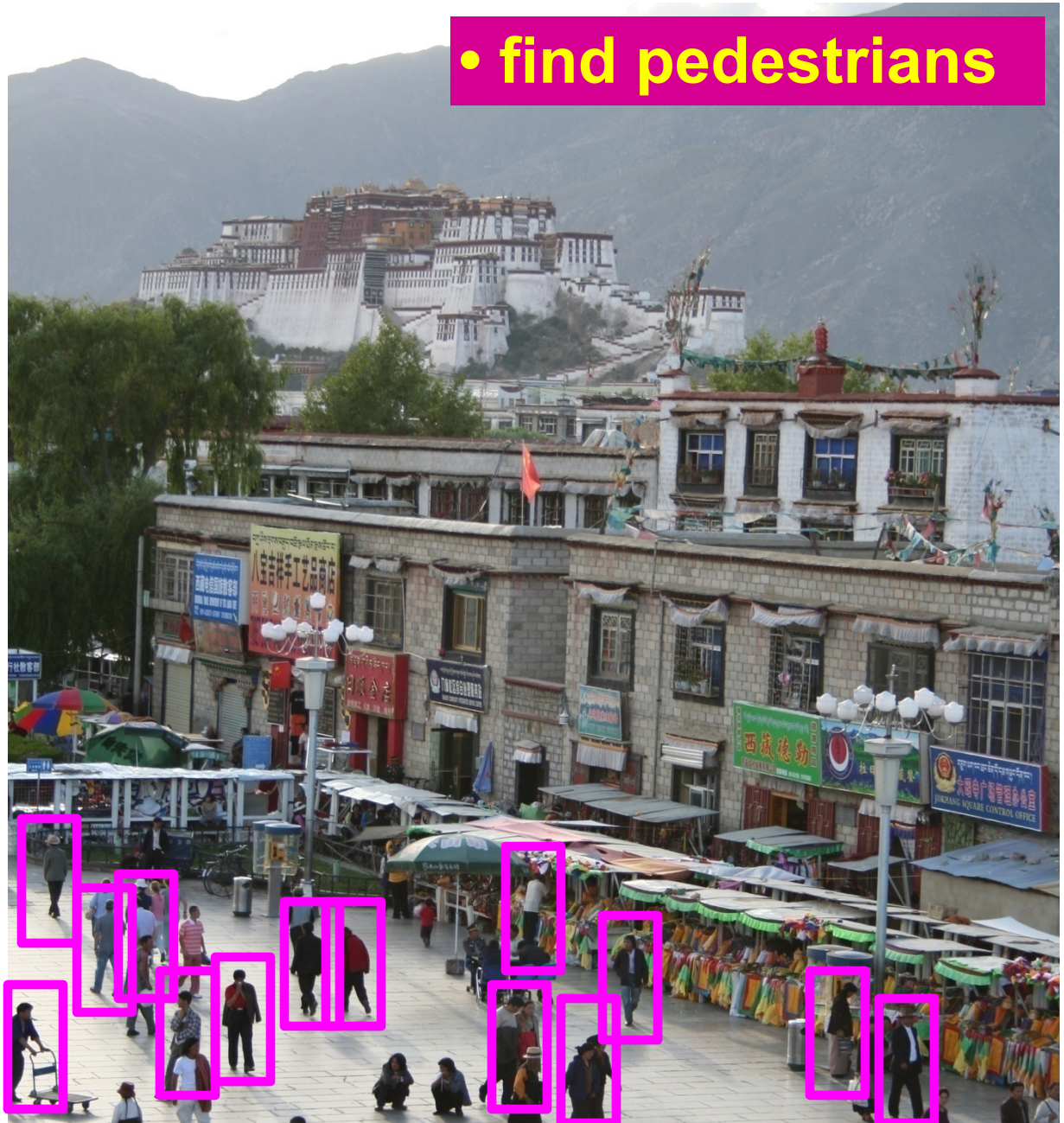
Image classification

- outdoor/indoor
- city/forest/factory/etc.

Image tagging



Object detection



Activity recognition

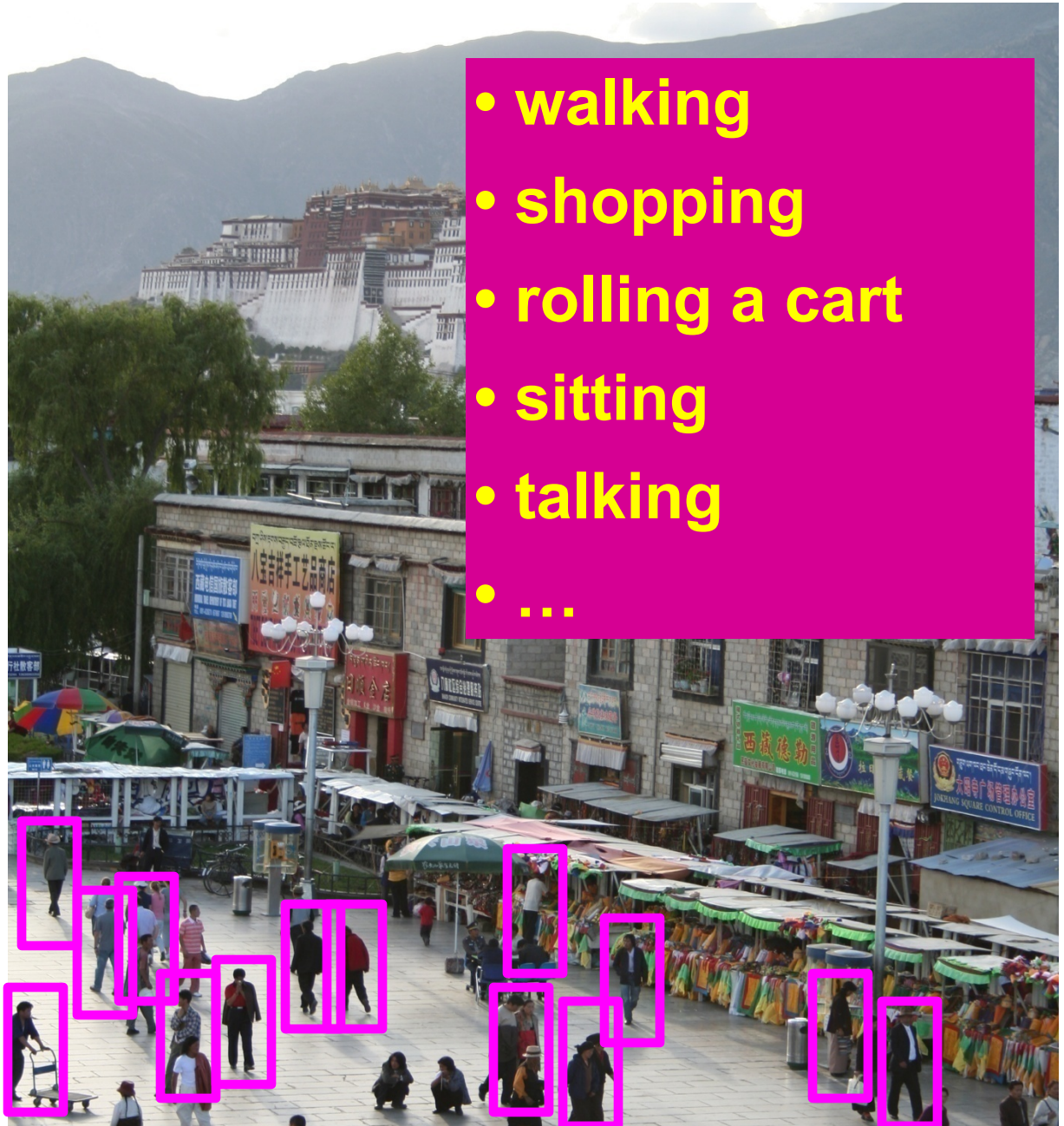
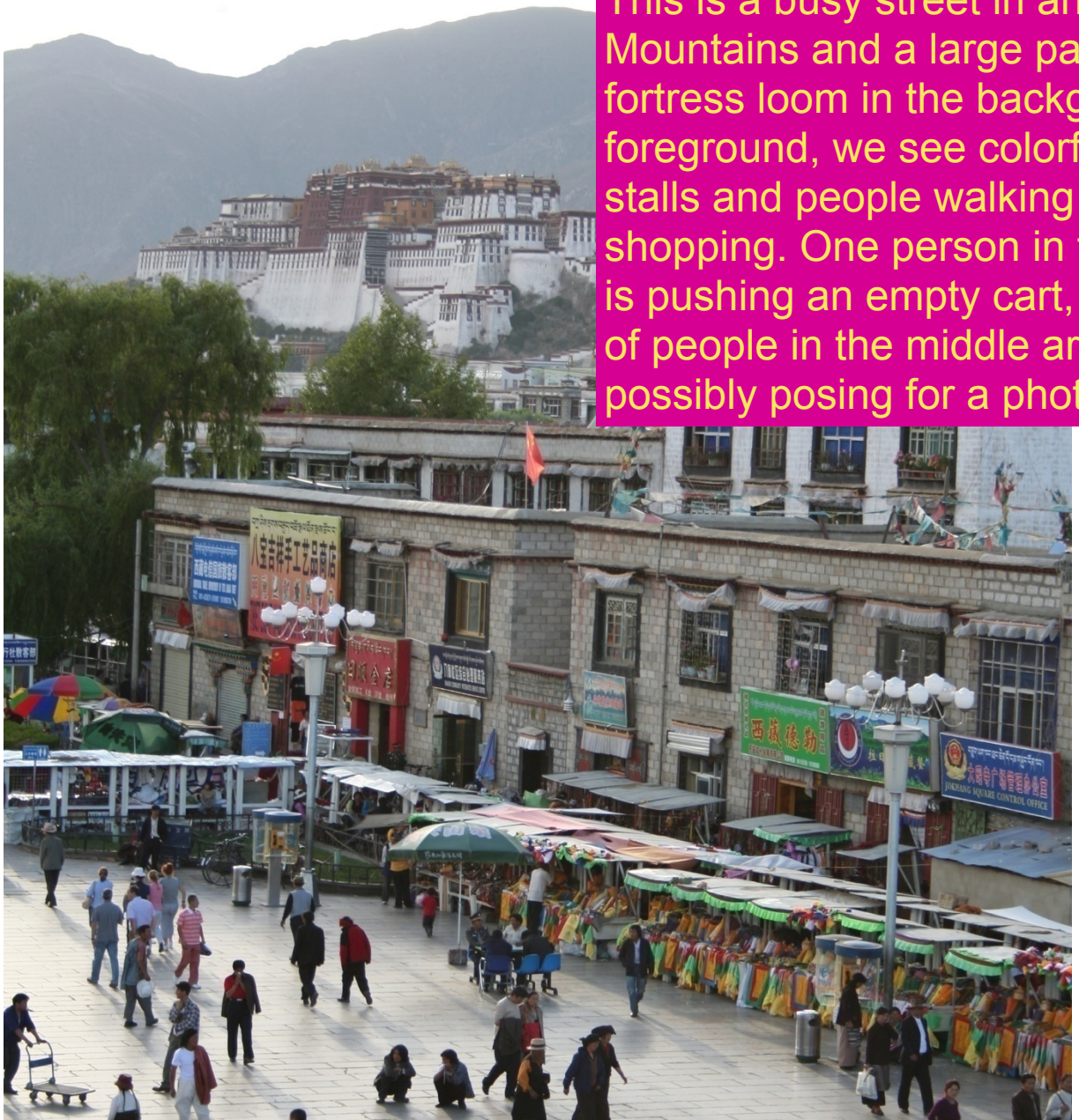


Image parsing

Image description



This is a busy street in an Asian city. Mountains and a large palace or fortress loom in the background. In the foreground, we see colorful souvenir stalls and people walking around and shopping. One person in the lower left is pushing an empty cart, and a couple of people in the middle are sitting, possibly posing for a photograph.

Image classification

Steps

Training

Training Images

Training Labels

Image Features

Training

Learned model

Testing

Test Image

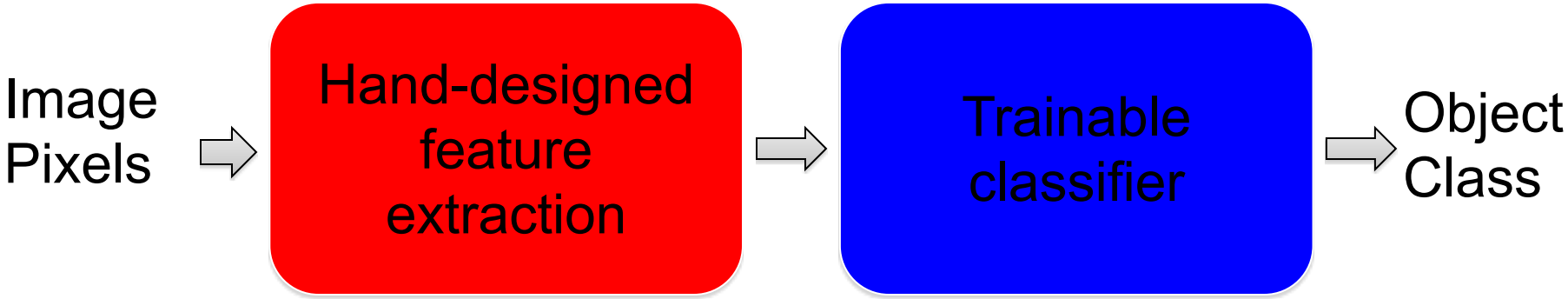
Image Features

Learned model

Prediction

Slide credit: D. Hoiem

Traditional recognition pipeline



- Features are not learned
- Trainable classifier is often generic (e.g. SVM)

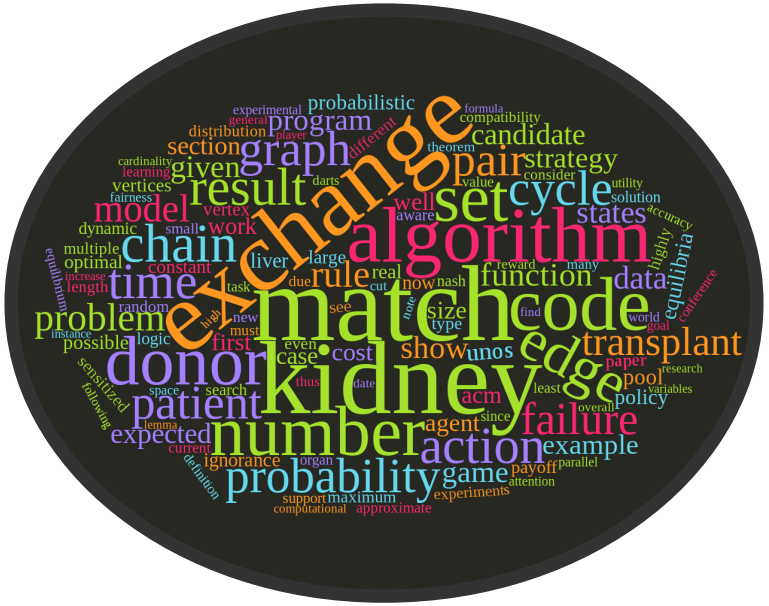
Bags of features

Bags of features: Motivation

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

Representing a document “in math”

Simplest method: bag of words



Represent each document as a vector of word frequencies

- Order of words does not matter, just #occurrences

Bag of words Example

the quick brown fox jumps over the lazy dog

I am he as you are he as you are me

he said the CMSC320 is 189 more CMSCs than the CMSC131

	the	CMSC320	you	he	I	quick	dog	me	CMSCs	…	than
Document 1	2	0	0	0	0	1	1	0	0		0
Document 2	0	0	2	2	1	0	0	1	0	…	0
Document 3	2	1	0	1	0	0	0	0	1		1

Term Frequency

Term frequency: the number of times a term appears in a specific document

- tf_{ij}: frequency of word *j* in document *i*

This can be the raw count (like in the BOW in the last slide):

- $tf_{ij} \in \{0,1\}$ if word *j* appears or doesn't appear in doc *i*
- $\log(1 + tf_{ij})$ – reduce the effect of outliers
- $tf_{ij} / \max_j tf_{ij}$ – normalize by document i's most frequent word

What can we do with this?

- Use as features to learn a classifier *w* → *y* …!

Defining features From Term Frequency

Suppose we are classifying if a document was written by The Beatles or not (i.e., binary classification):

- Two classes $y \in Y = \{ 0, 1 \} = \{$ not_beatles, beatles $\}$

Let's use $tf_{ij} \in \{0,1\}$, which gives:

	the	CMS C320	you	he	I	quick	dog	me	CMS Cs	...	than
$x_1^T =$	1	0	0	0	0	1	1	0	0		0
$x_2^T =$	0	0	1	1	1	0	0	1	0	…	0
$x_3^T =$	1	1	0	1	0	0	0	0	1		1

$y_1 = 0$

$y_2 = 1$

$y_3 = 0$

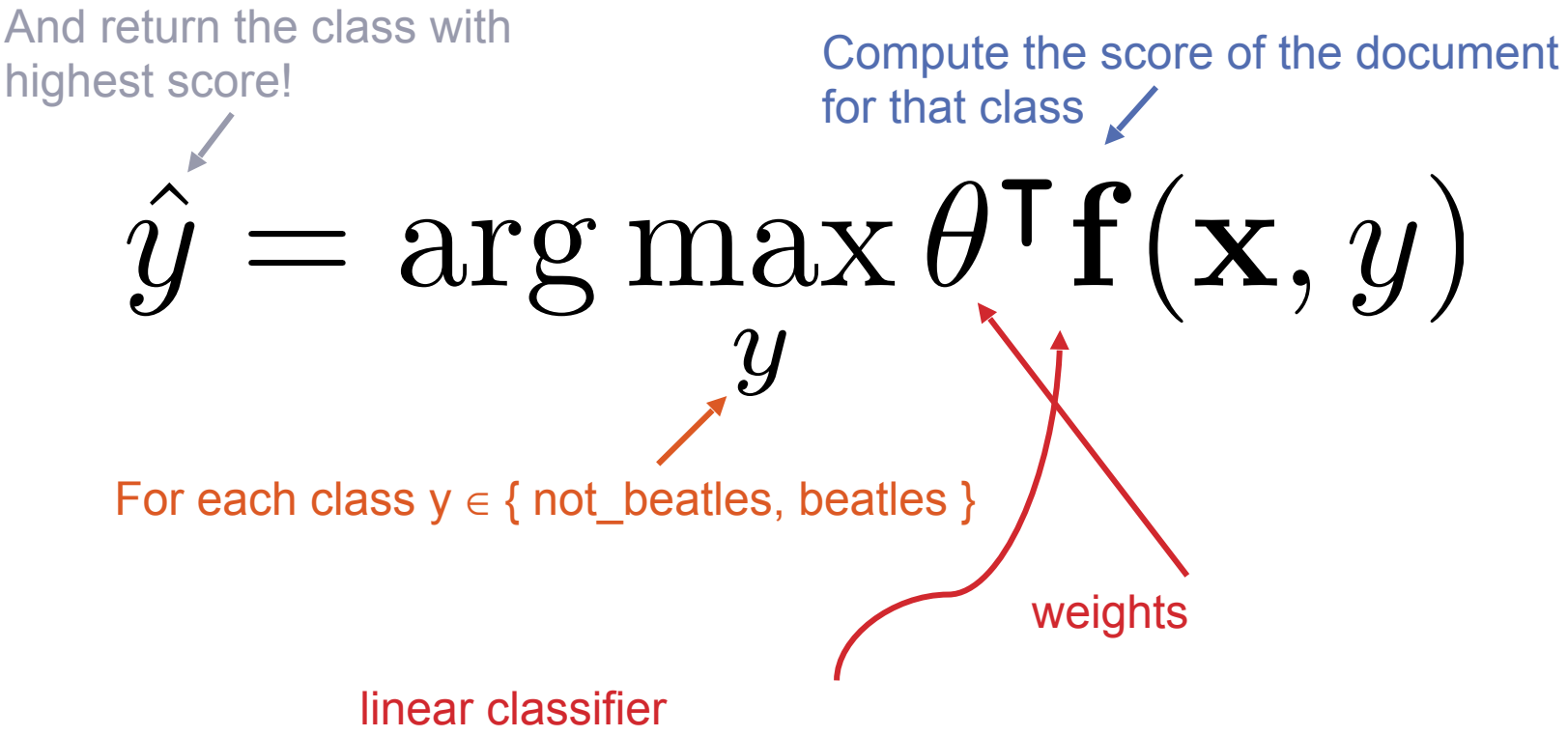
Then represent documents with a feature function:

f(x, y = not_beatles = 0) = $[x^T, 0^T, 1]^T$

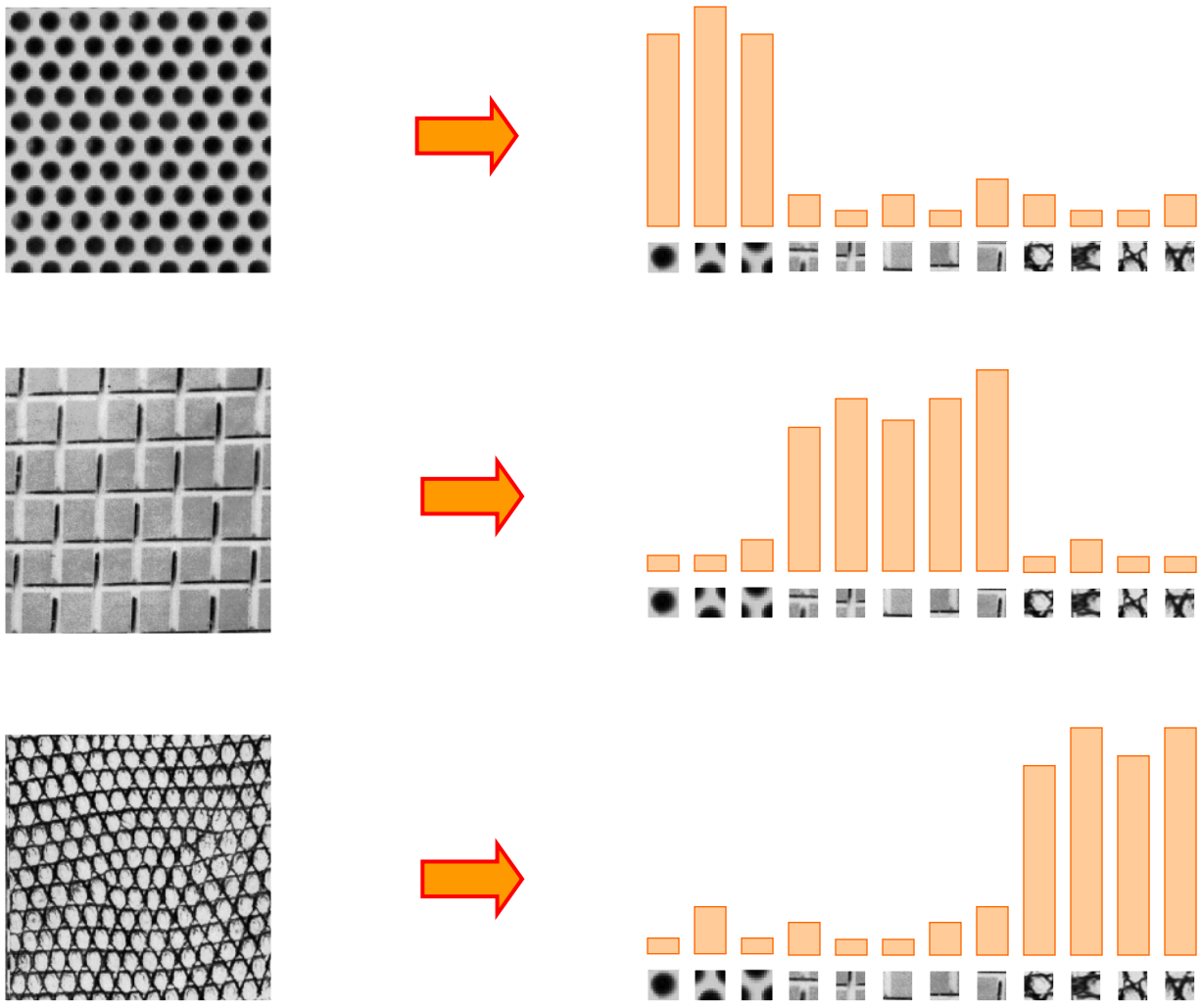
f(x, y = beatles = 1) = $[0^T, x^T, 1]^T$

Linear classification

We have a feature function f(x, *y*) and a score $\psi_{xy} = \theta^T$ f(x, *y*)



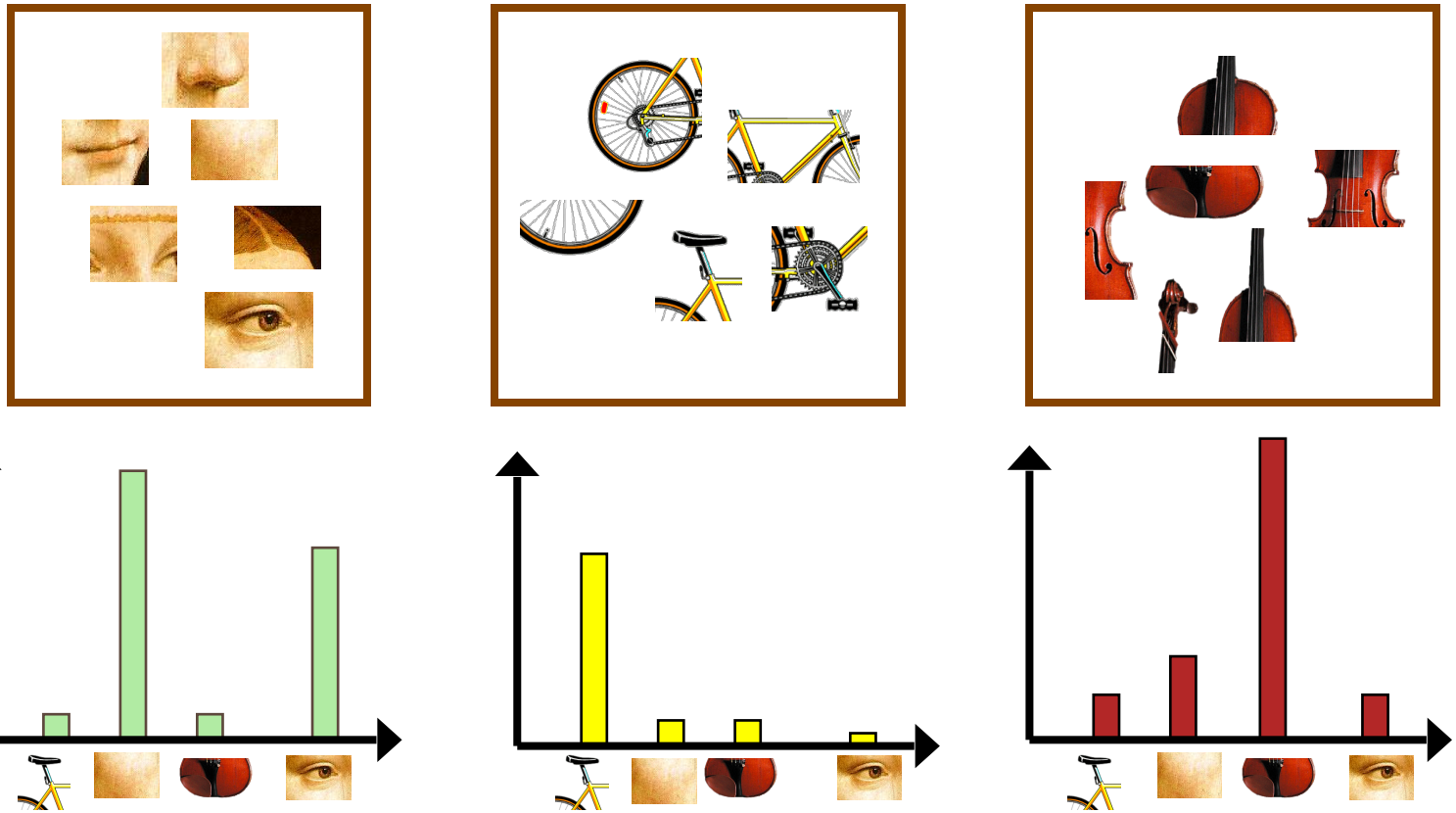
Texture recognition



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

Traditional features: Bags-of-features

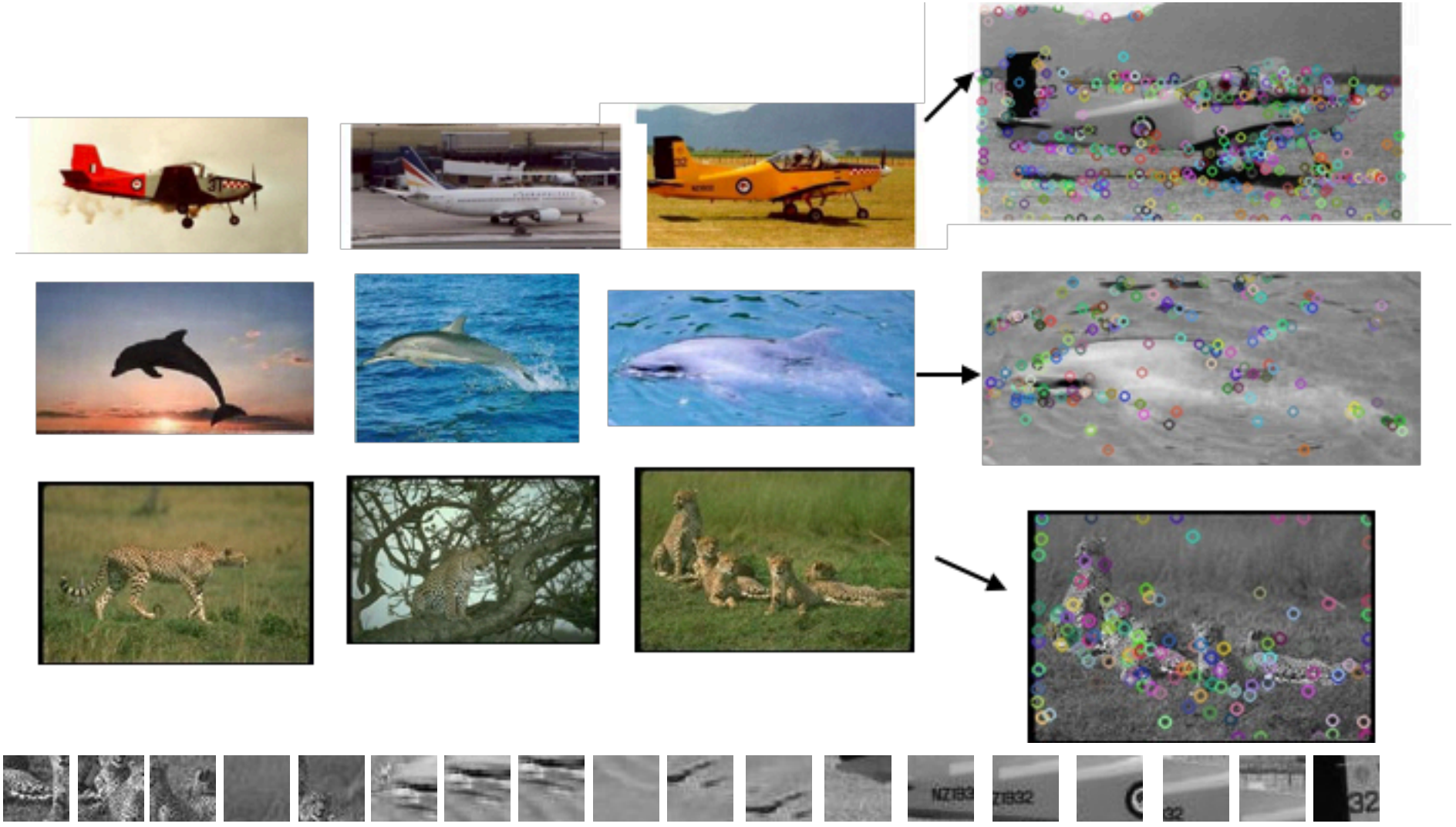
1. Extract local features
2. Learn “visual vocabulary”
3. Quantize local features using visual vocabulary
4. Represent images by frequencies of “visual words”



1. Local feature extraction

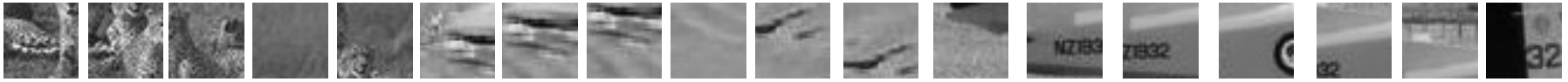
- Sample patches and extract descriptors

Keypoints

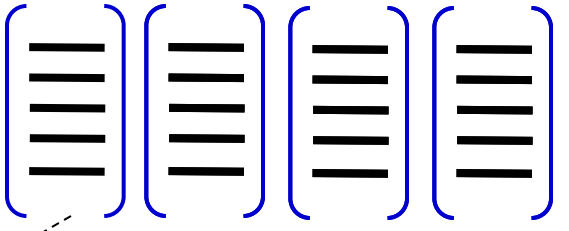


patches surrounding keypoints

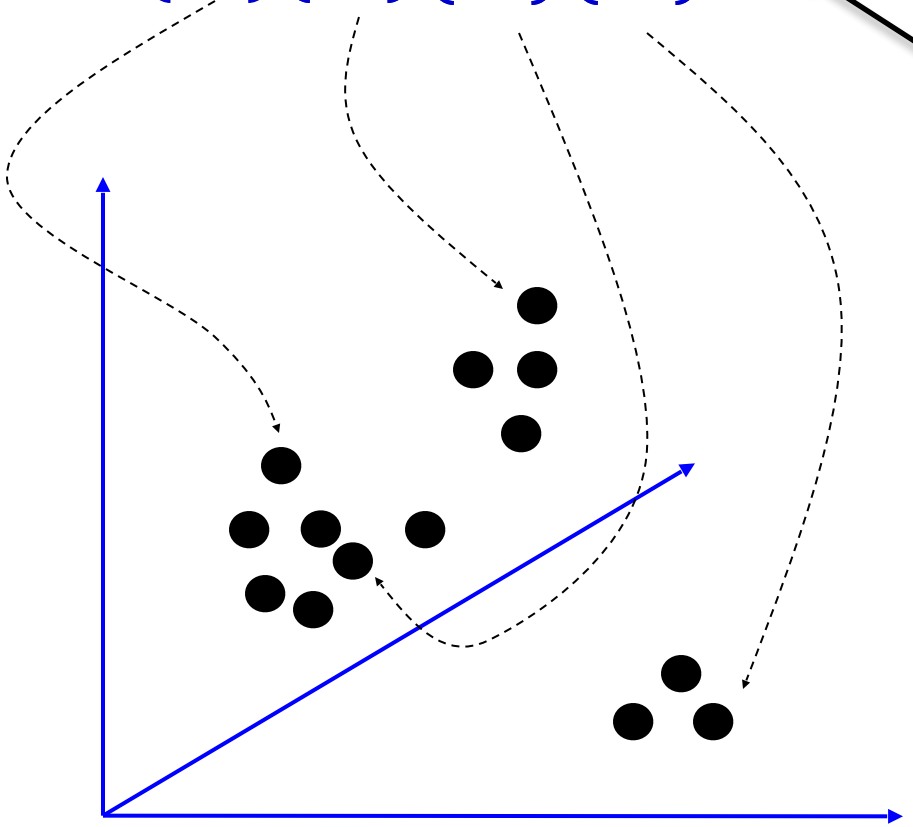
2. Learning the visual vocabulary



patches surrounding keypoints

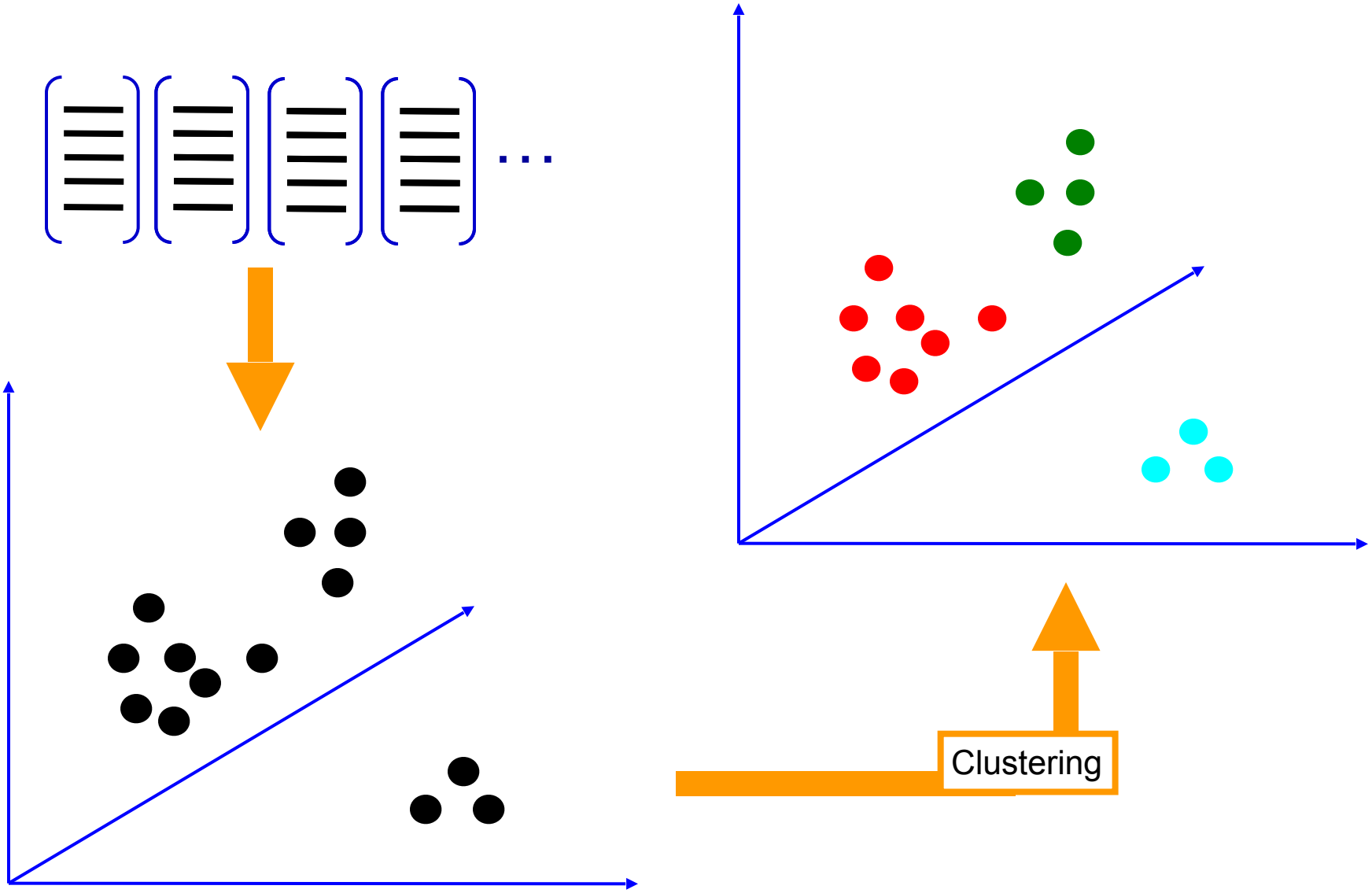


Extracted descriptors from the training set



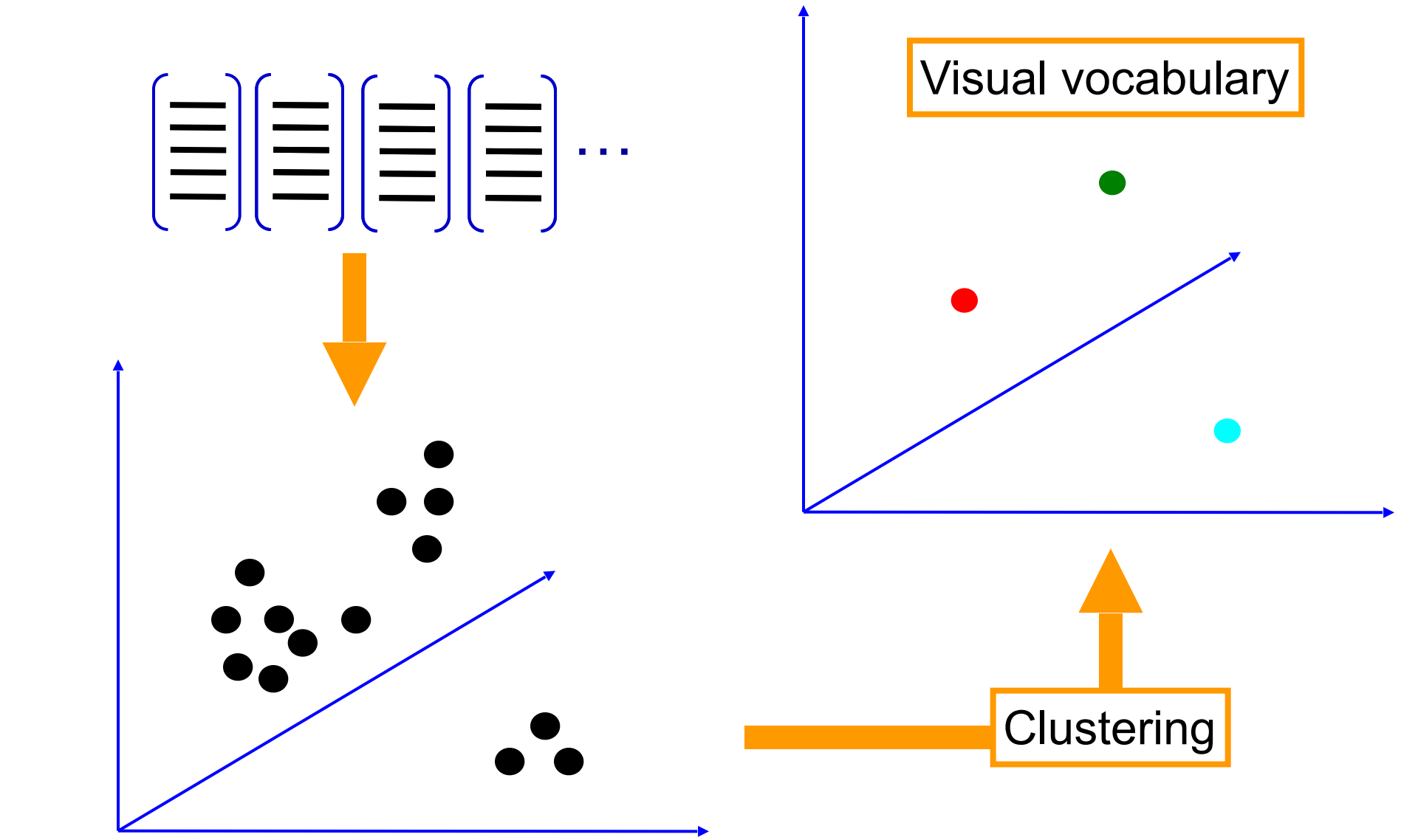
Slide credit: Josef Sivic except for the image patches

2. Learning the visual vocabulary



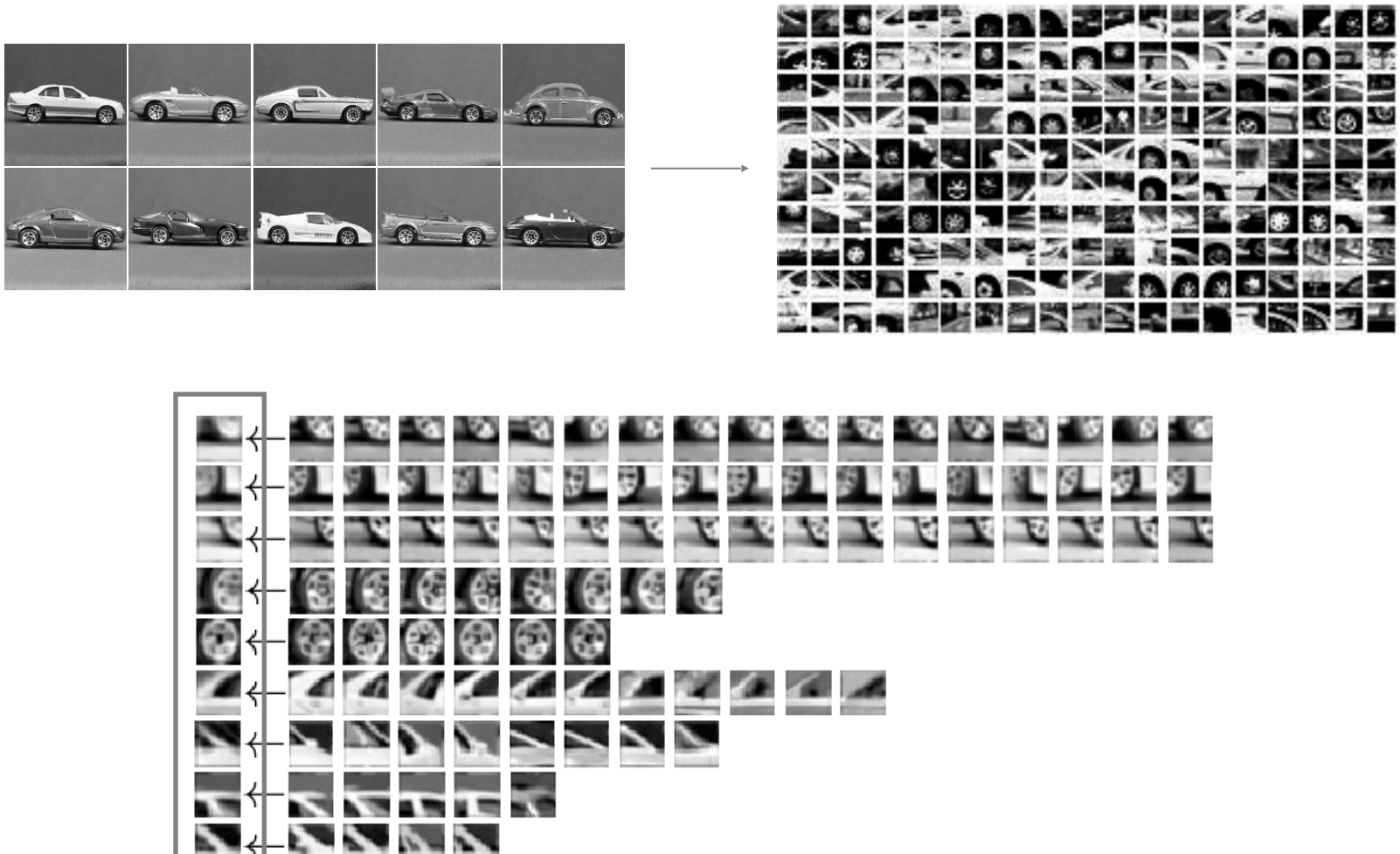
Slide credit: Josef Sivic

2. Learning the visual vocabulary



Slide credit: Josef Sivic

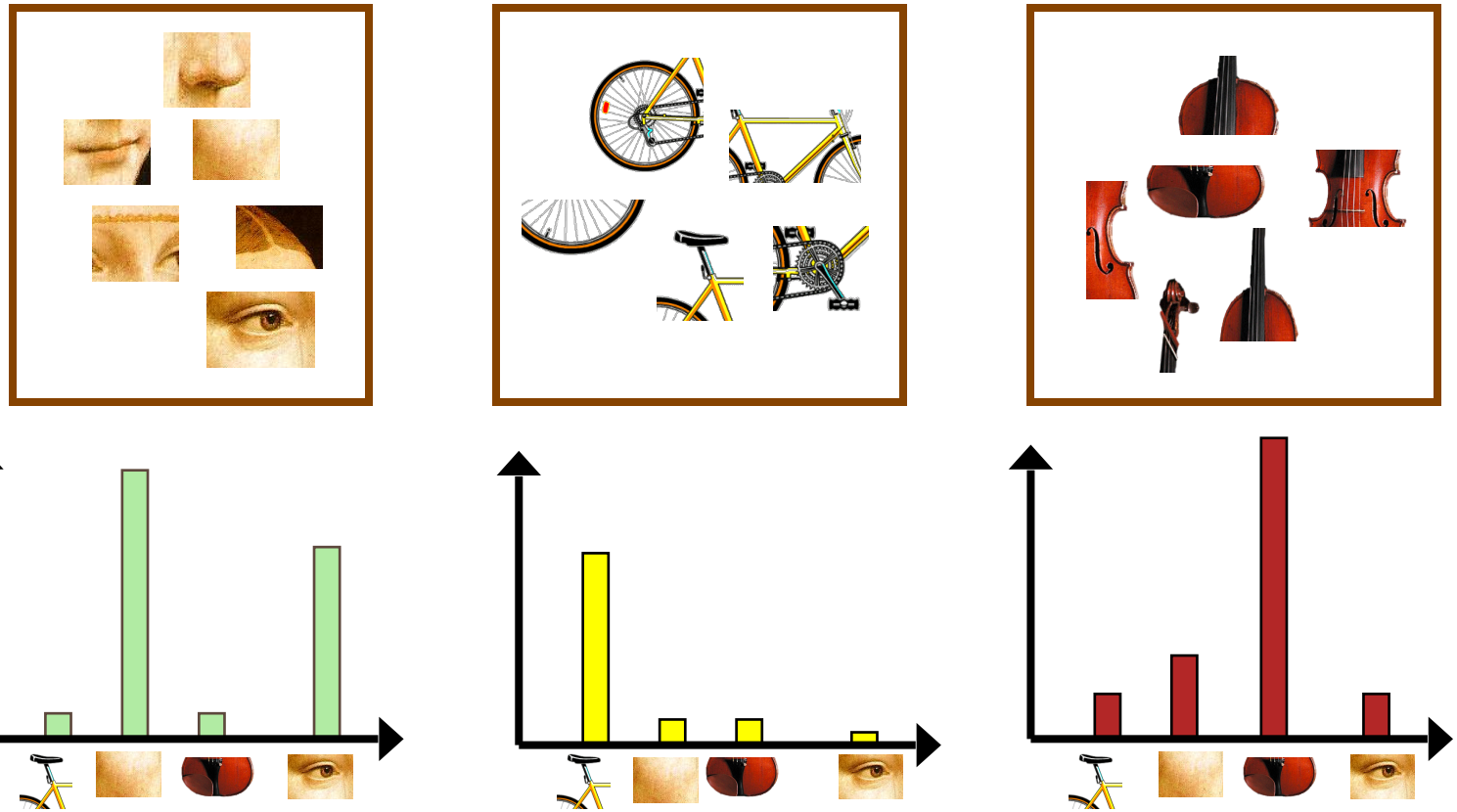
Example visual vocabulary



Source: B. Leibe

Bag-of-features steps

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

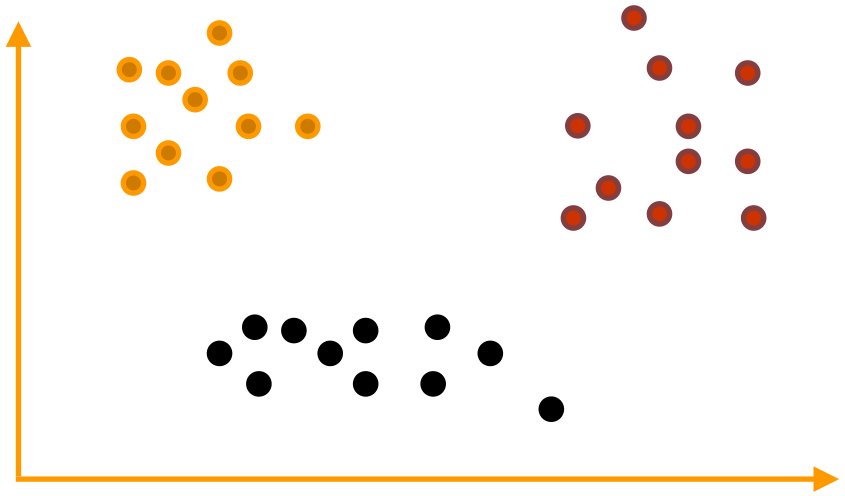


CLUSTERING

- Group a collection of points into clusters
- We have seen “supervised methods”, where the outcome (or response) is based on various predictors.
- In clustering, we want to extract patterns on variables without analyzing a specific response variable.
- This is a form of “unsupervised learning”

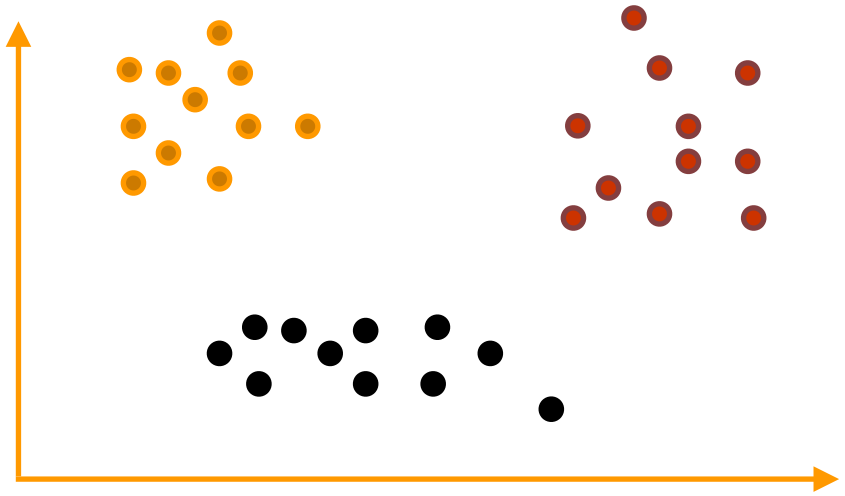
CLUSTERING

- The points in each cluster are closer to one another and far from the points in other clusters.



Data Points

- Each of the data points belong to some n-dimensional space.



Clusters

- The clusters are grouped based on some distance /similarity measurement.

- Two-main types of clustering techniques are:
  - point-assignment or k-means
  - hierarchical or agglomerative

- Sometimes we work with dissimilarity (or distance) measurements.

- First let us discuss the (dis)similarity measurements.

dissimilarity measurements

Given measurements x_{ij} for $i = 1,\ldots,M$ observations over $j = 1,\ldots,n$ predictors.

Define dissimilarity, $d_j(x_{ij}, x_{i'j})$

- We can define dissimilarity between objects as

$$d(x_i, x_{i'}) = \sum_{j=1}^{N} d_j(x_{ij}, x_{i'j})$$

- The most common distance measure is squared distance

$$d_j(x_{ij}, x_{i'j}) = (x_{ij} - x_{i'j})^2$$

dissimilarity measurements

- Absolute difference $d_j(x_{ij}, x_{i'j}) = |x_{ij} - x_{i'j}|$

- For categorical variables, we could set

$$d_j(x_{ij}, x_{i'j}) = 0 \text{ if } x_{ij} = x_{i'j}$$

$$1 \text{ otherwise}$$

Clustering techniques

- Two-main types of clustering techniques are:
    - point-assignment or k-means
    - hierarchical or agglomerative

K-Means Clustering

- A commonly used algorithm to perform clustering
- Assumptions:
  - Euclidean distance,

$$d_{ii'} = d(x_i, x_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = ||x_i - x_{i'}||^2$$

- K-means partitions observations into K clusters, with K provided as a parameter.

K-Means Objective function

- We are given an integer *k* and a set of *n* data points $X \subset \mathbb{R}^p$
  we wish to choose *k* centers *C* so as to minimize within-cluster dissimilarity.

$$W = \sum_{i=1}^{N} \min_{c \in C} ||x_i - c||^2$$

- The criteria to minimize is the total distance given by each observation to the mean(centroid) of the cluster to which the observation is assigned.

K-Means - Iterative algorithm

1. Arbitrarily choose an initial *k* centers

$$C = \{c_1, c_2, \ldots, c_k\}$$

2. For each $i \in \{1,\ldots,k\}$ set the cluster C_i to be the set of points *X* that are closer to c_i than they are to c_j for all $j \neq i$

3. For each $i \in \{1,\ldots,k\}$, set c_i to be the center of mass of all points in C_i : $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$

4. Repeat Steps 2 and 3 until *C* no longer changes

K-means - picking K

- Minimization, W, It is not a convex criteria, we may not obtain global optima.

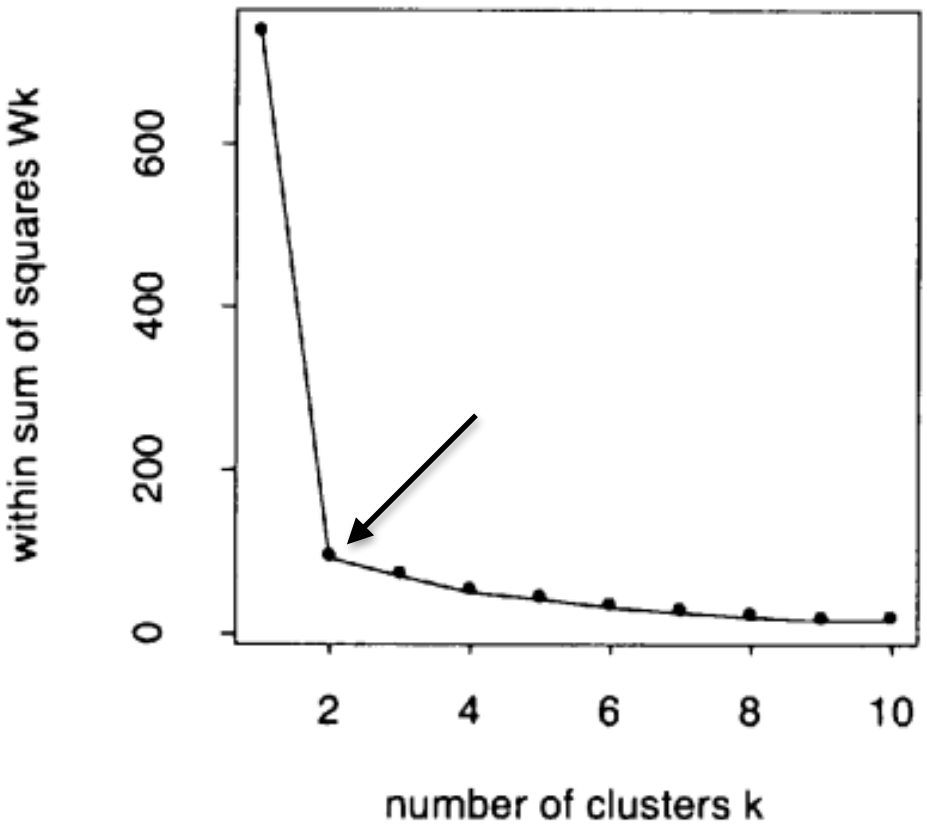
- In practice the algorithm is run with multiple initializations (step 1) and the best clustering is used.

Choosing the number of clusters

1. The number of parameters must be determined before running k-means algorithm.

2. There is no clean direct method for choosing the number of clusters to use in the K-means algorithm

Choosing the number of clusters - Elbow method



Picking k - Gap statistic (Hastie et al.)

https://statweb.stanford.edu/~gwalther/gap

- Suppose data is clustered into k clusters, $C_1, C_2, \ldots, C_k$

  with C_r denoting the observations in cluster r, and $n_r = |C_r|$
- The sum of pairwise distances for all points in cluster r is given by $D_r = \sum_{i,i' \in C_r} d_{ii'}$

  $$d_{ii'} = d(x_i, x_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = ||x_i - x_{i'}||^2$$

  and $W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r$
- If d is the squared Euclidean distance then W_k is the pooled within-cluster sum of squares around the cluster means.

Gap Statistic - Procedure (Tibshirani et al.)

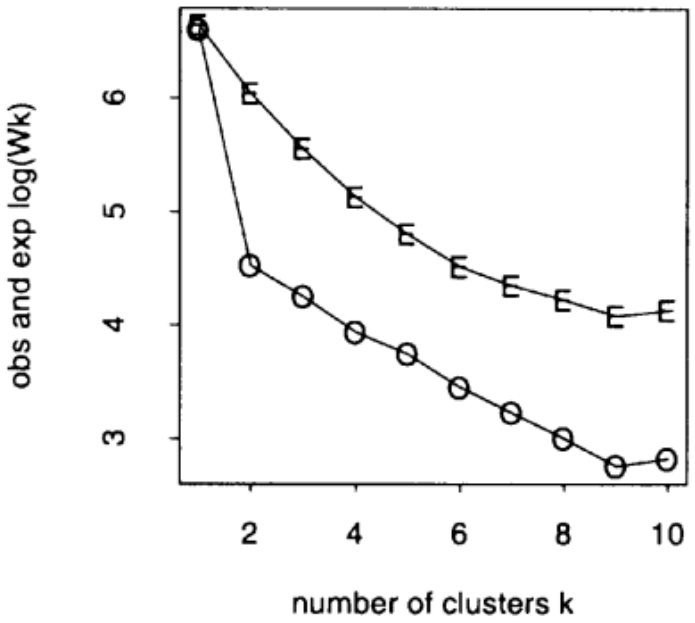
1. Cluster the observed data, varying the total number of clusters from $k = 1,2,\ldots,K$. Obtain W_k
2. Generate B reference sets, using uniform distribution or SVD followed by uniform distribution. W^*_{kb}, with $b = 1,2,\ldots,B, k = 1,2,\ldots,K$
3. Compute the estimated Gap statistic
   $Gap(k) = (1/B)\sum_b log(W^*_{kb}) - log(W_k)$
4. Let $\bar{l} = (1/B)\sum_b log(W^*_{kb})$, compute the standard deviation

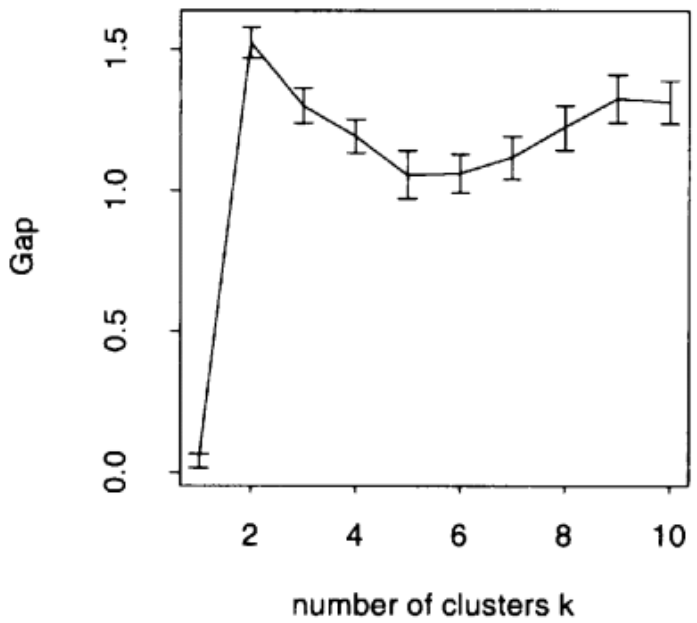
   $sd_k = [(1/B)\sum_b \{log(W^*_{kb}) - \bar{l}\}^2]^{1/2}$
5. Define $s_k = sd_k\sqrt{1 + 1/B}$.
6. Choose the number of clusters via
   $\hat{k}$ = smallest k such that $Gap(k) \geq Gap(k+1) - s_{k+1}$

Choosing the number of clusters - Gap statistic

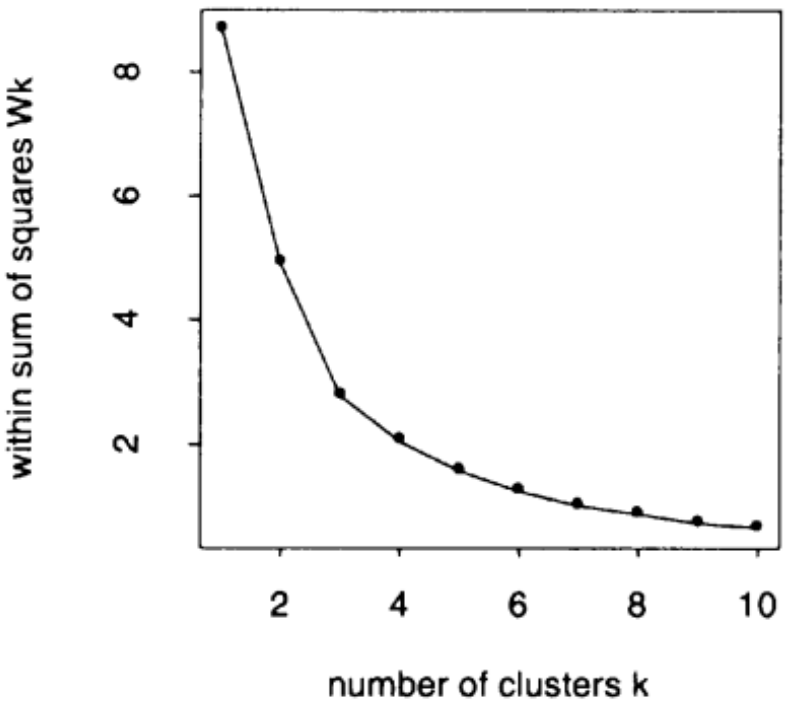
$$Gap(k) = E[log(W^*_{kb})] - log(W_k)$$

$$E[log(W^*_{kb})] = \bar{l} = (1/B)\sum_b log(W^*_{kb})$$

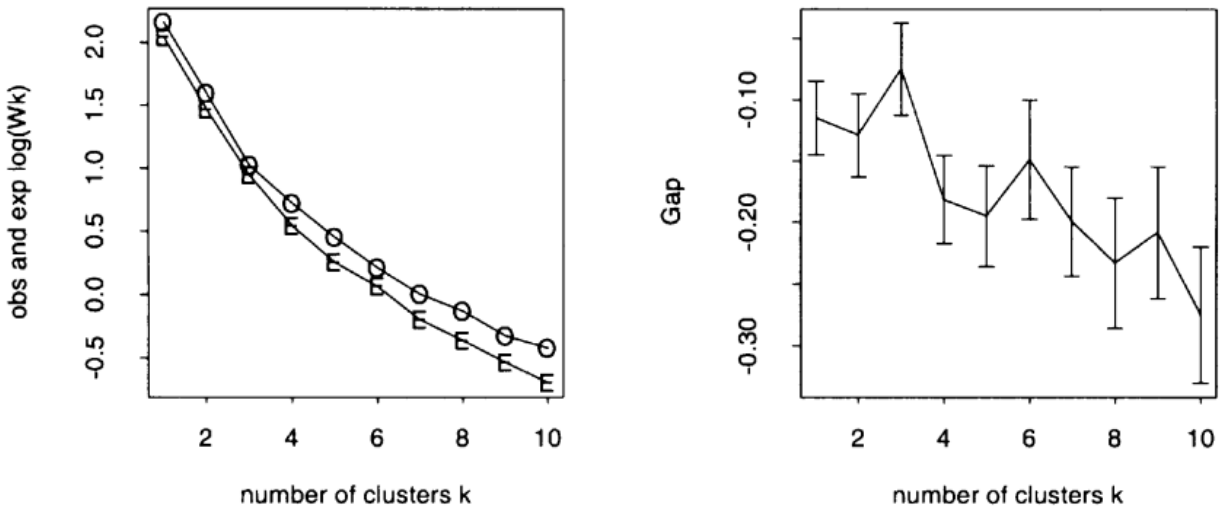




Choosing the number of clusters - Elbow method

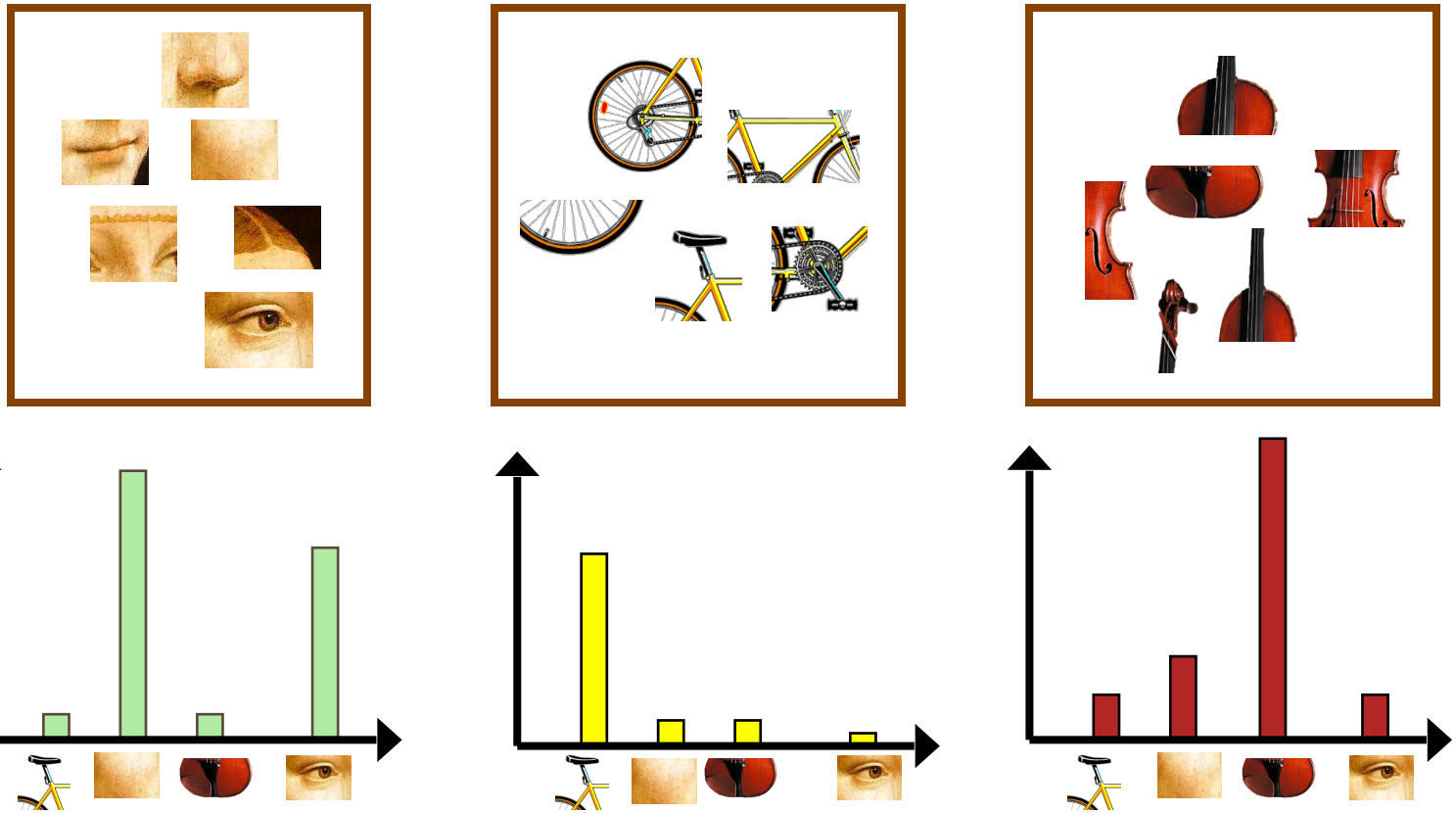


Choosing the number of clusters - Gap statistic



Traditional features: Bags-of-features

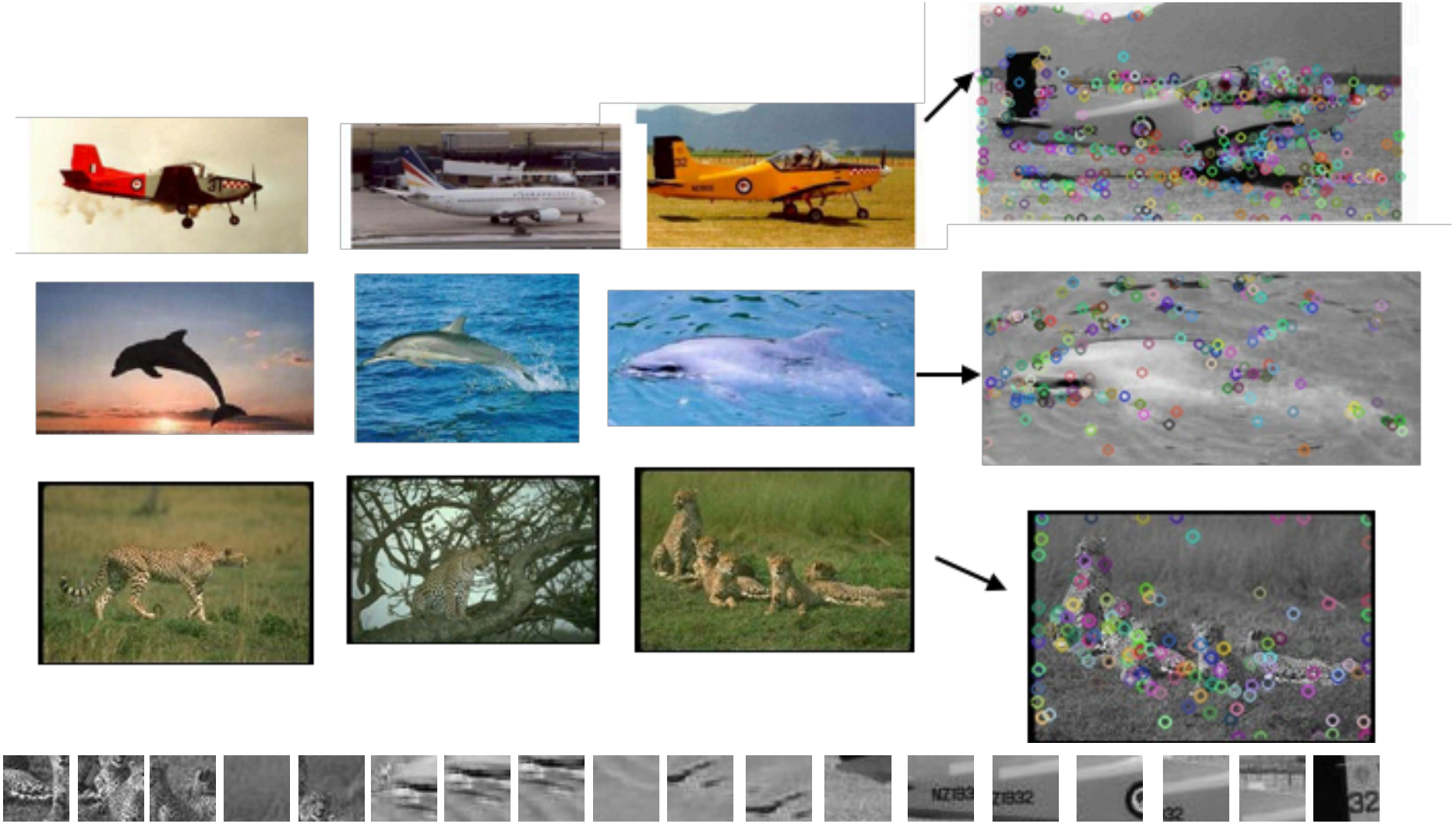
1. Extract local features
2. Learn “visual vocabulary”
3. Quantize local features using visual vocabulary
4. Represent images by frequencies of “visual words”



1. Local feature extraction

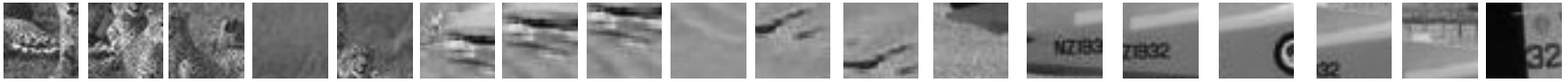
- Sample patches and extract descriptors

Keypoints

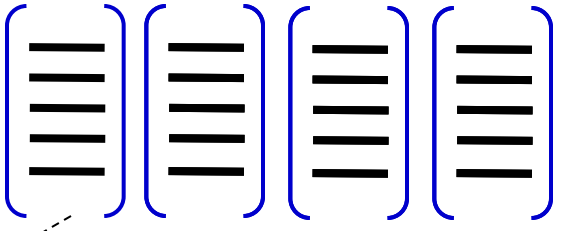


patches surrounding keypoints

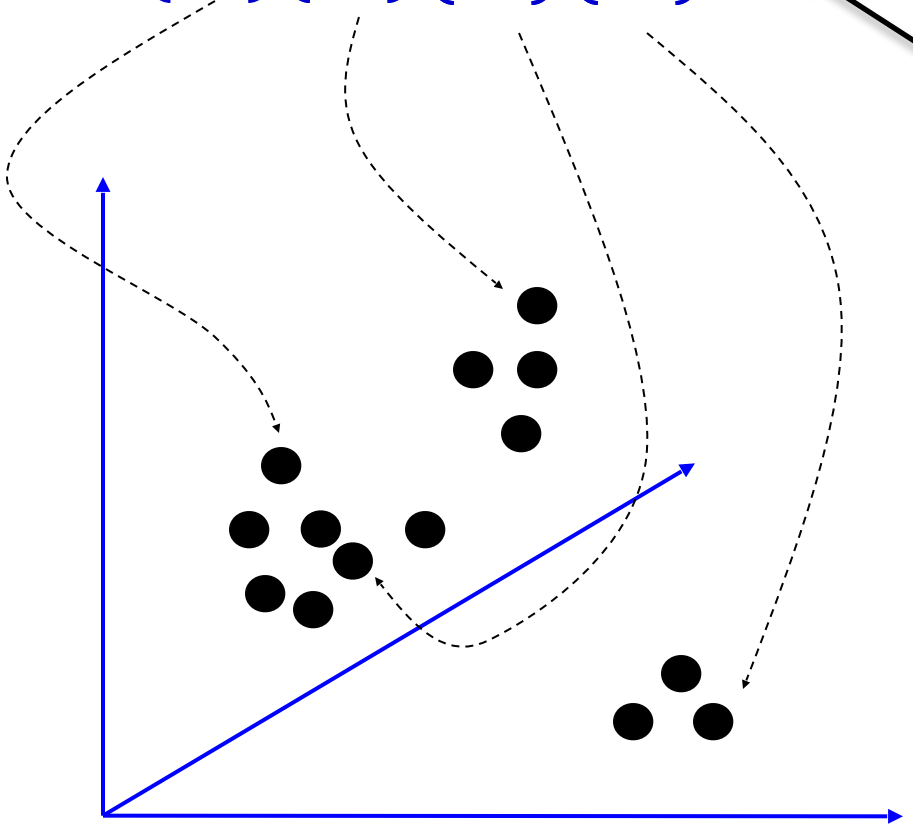
2. Learning the visual vocabulary



patches surrounding keypoints

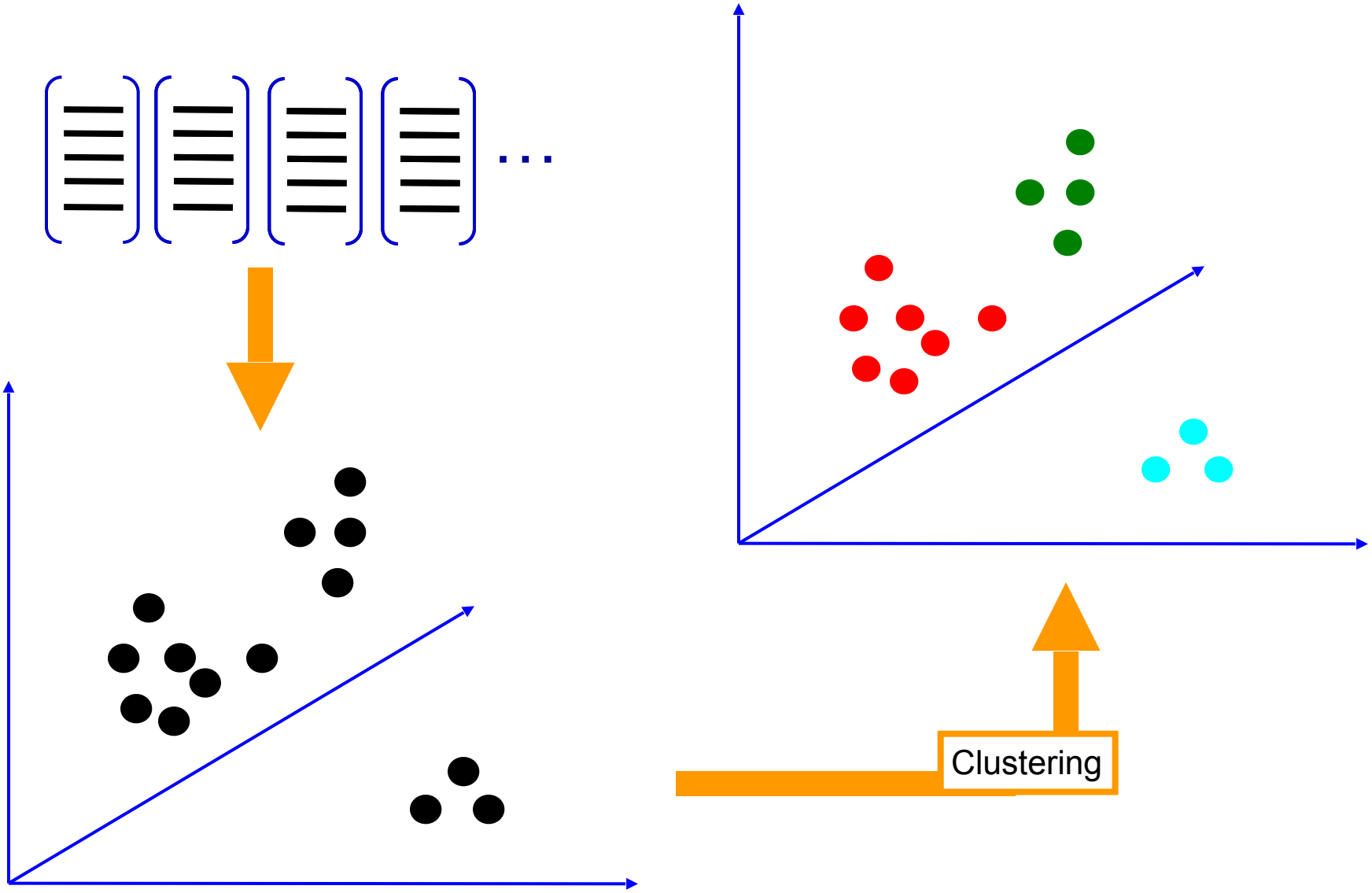


Extracted descriptors from the training set



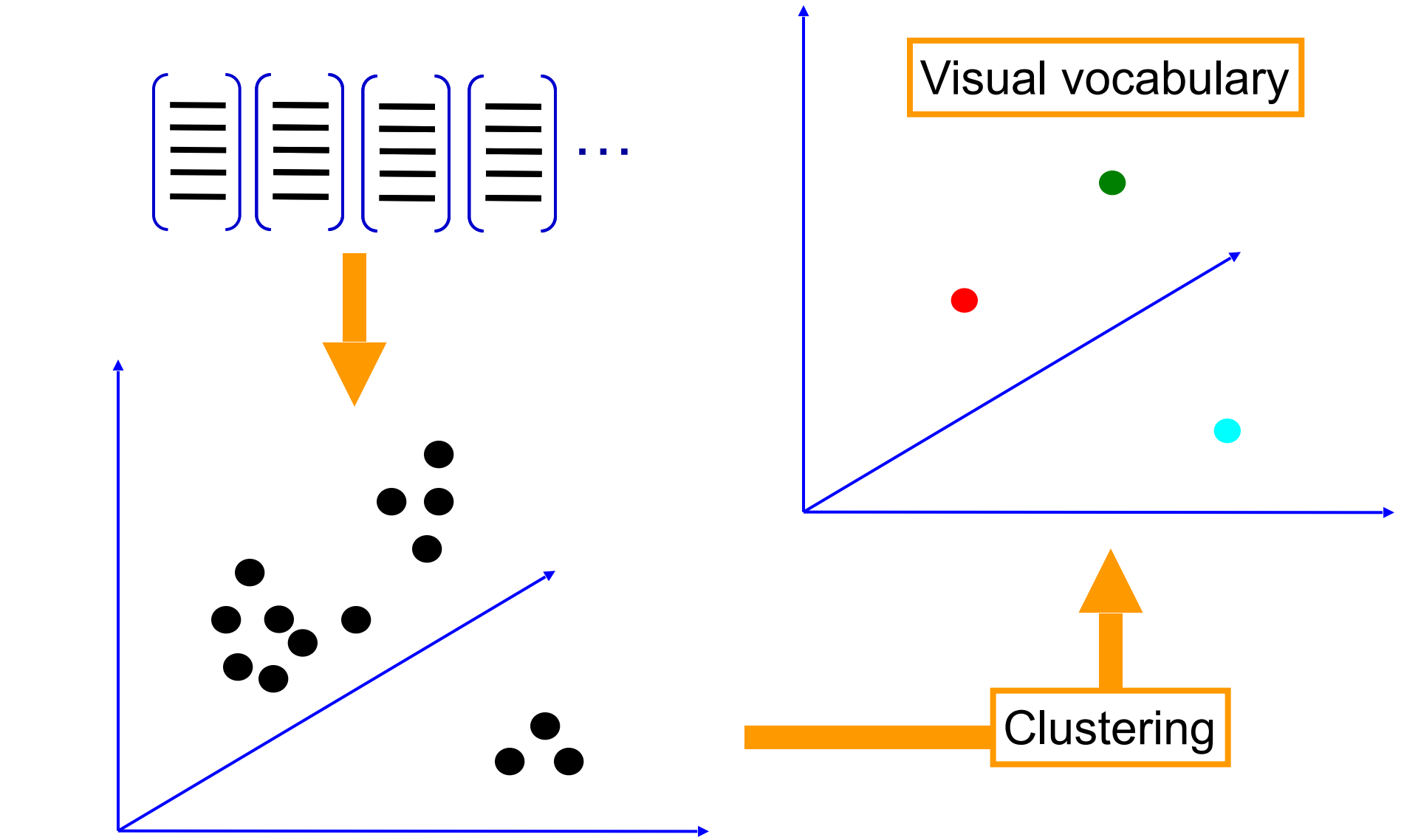
Slide credit: Josef Sivic except for the image patches

2. Learning the visual vocabulary



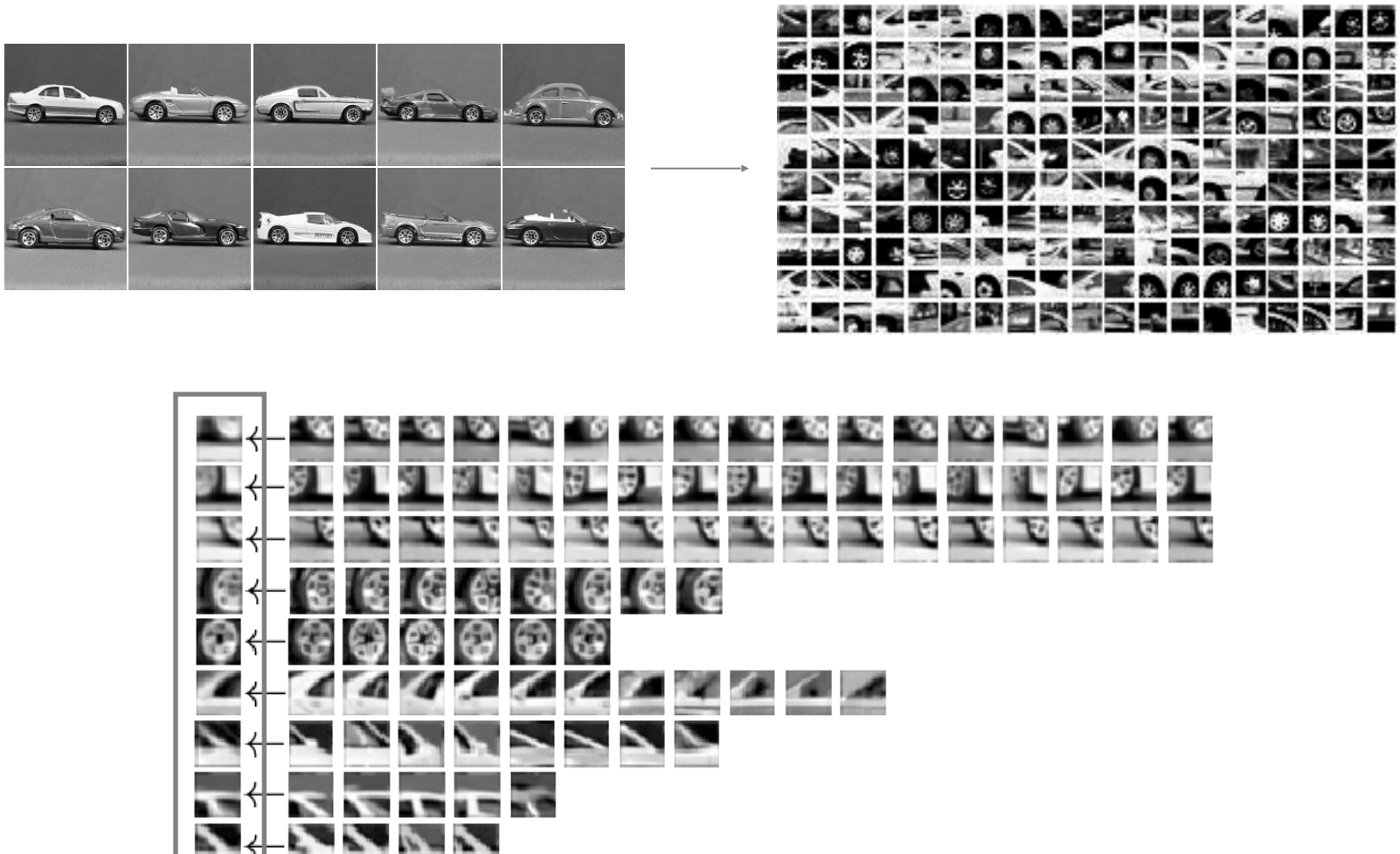
Slide credit: Josef Sivic

2. Learning the visual vocabulary



Slide credit: Josef Sivic

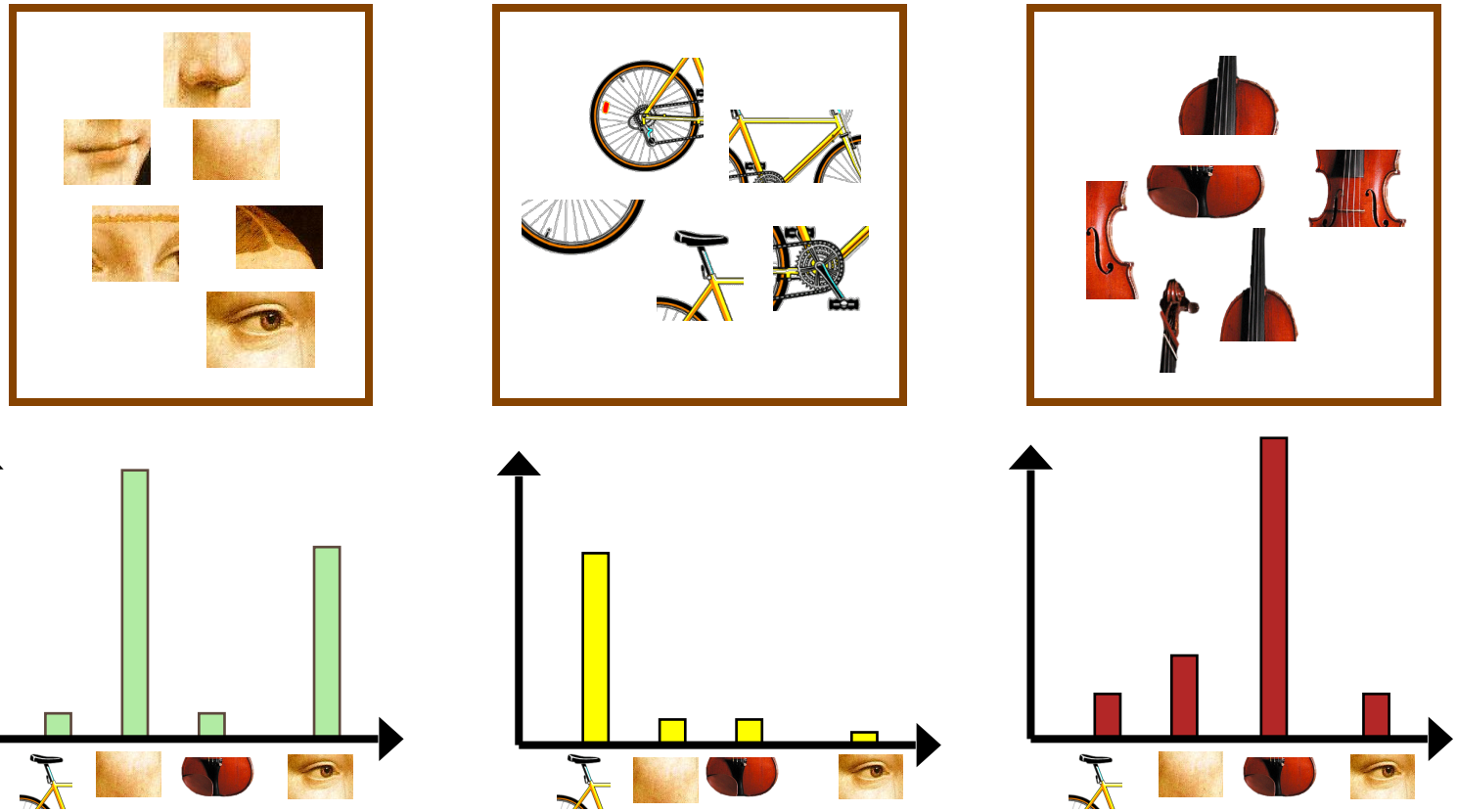
Example visual vocabulary



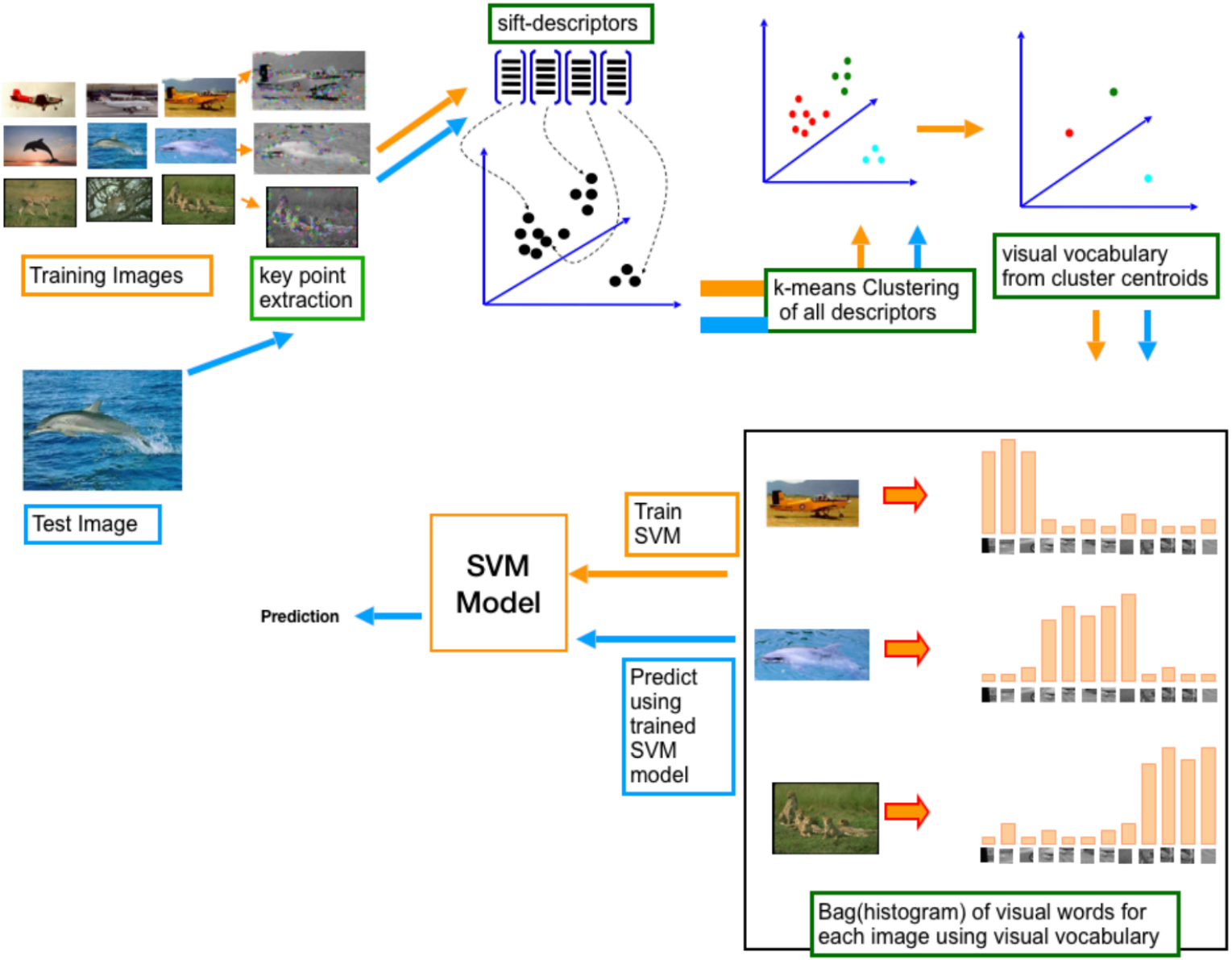
Source: B. Leibe

Bag-of-features steps

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



Bag-of-features (Visual Words) steps



Generalization

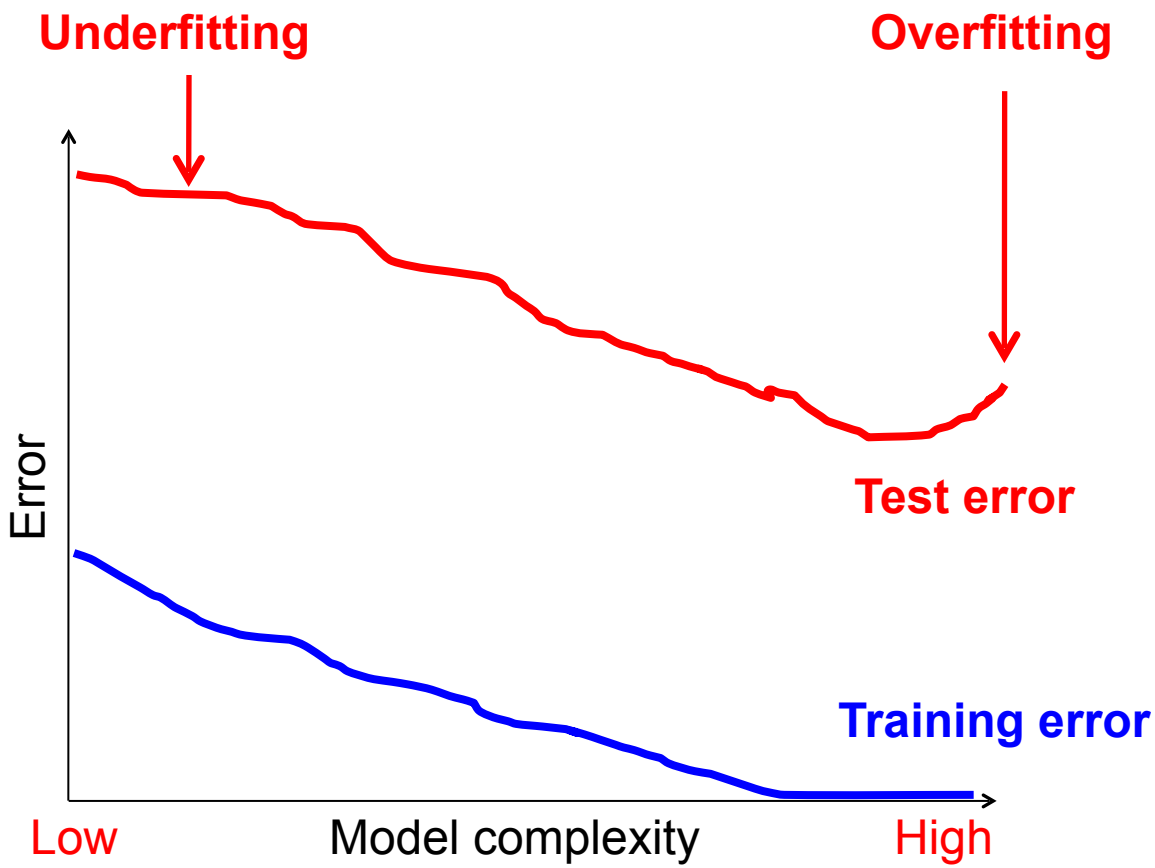
- Generalization refers to the ability to correctly classify never before seen examples
- Can be controlled by turning “knobs” that affect the complexity of the model

Training set (labels known)

Test set (labels unknown)

Diagnosing generalization ability

- **Training error:** how does the model perform on the data on which it was trained?
- **Test error:** how does it perform on never before seen data?



Source: D. Hoiem

Underfitting and overfitting

- **Underfitting:** training and test error are both *high*
  - Model does an equally poor job on the training and the test set
  - Either the training procedure is ineffective or the model is too "simple" to represent the data
- **Overfitting:** Training error is *low* but test error is *high*
  - Model fits irrelevant characteristics (noise) in the training data
  - Model is too complex or amount of training data is insufficient

Underfitting | Good generalization | Overfitting

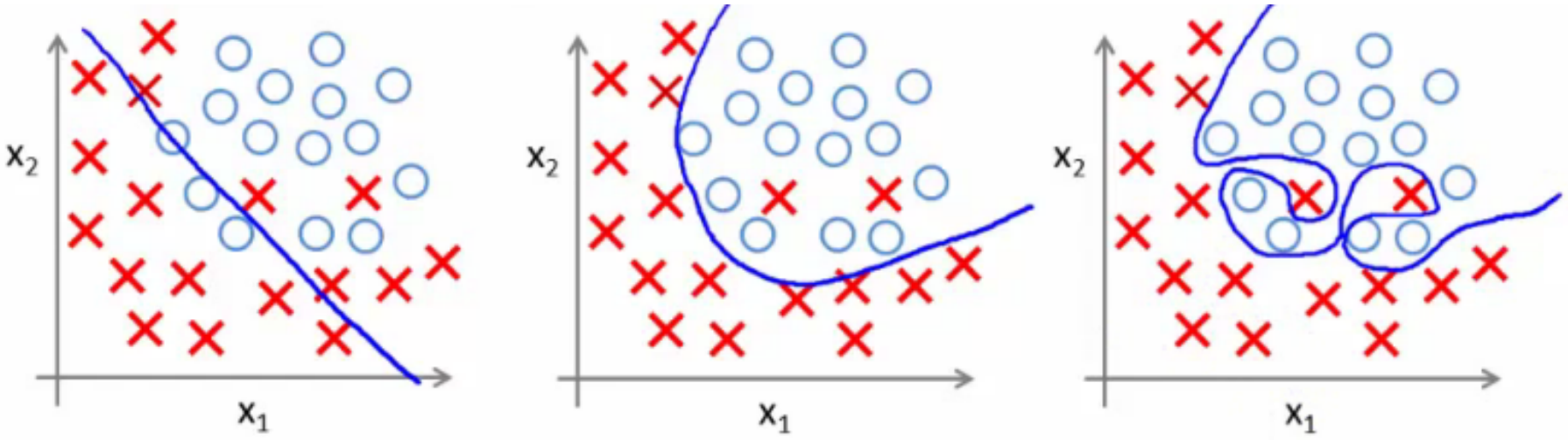
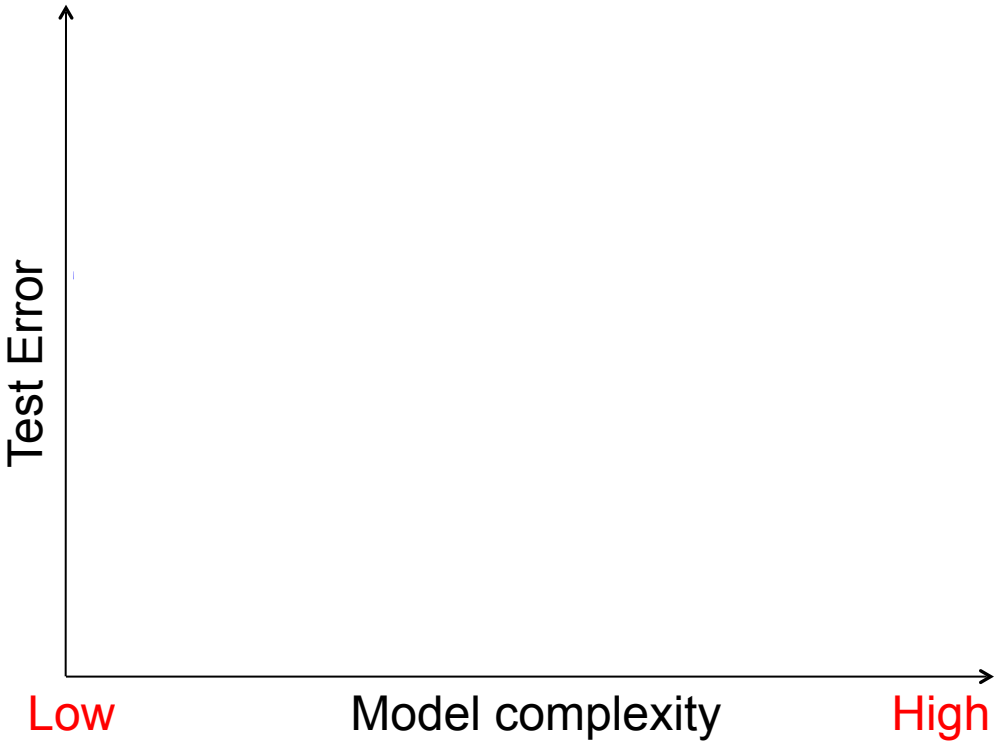


Figure source

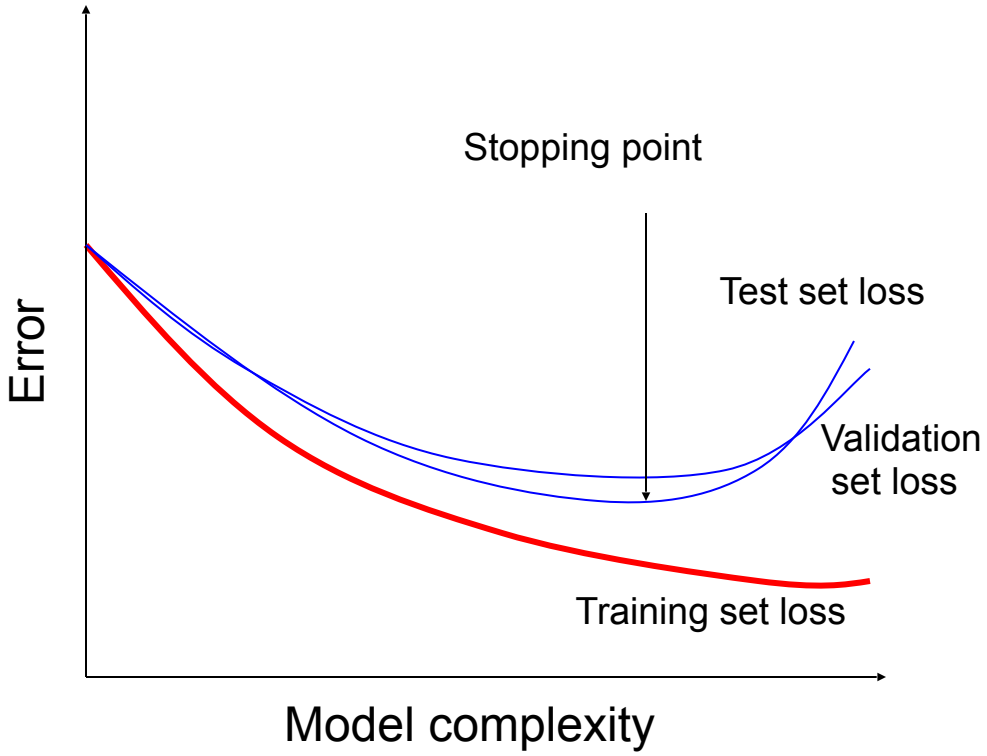
Effect of training set size



Source: D. Hoiem

Validation

- Split the data into **training**, **validation**, and **test** subsets
- Use training set to **optimize model parameters**
- Use validation test to **choose the best model**
- Use test set only to **evaluate performance**



Summary

The different steps