Imperial College London

Visual Categorisation by Bag of Words
Clustering

Tae-Kyun (T-K) Kim
Senior Lecturer
https://labicvl.github.io/

Backgrounds:
Optimisation (EE429) - Gradient method

Notations

d : descriptor vector

D : dimension of descriptor vector

N' : number of descriptor vectors

K : number of clusters or groups.

μ_i : cluster centers where $i = 1,...,K$

r_{ni} : binary indicator variables where $i = 1,...,K$

x : feature vector for classification i.e. the *K* binned bag-of-words histogram

K : dimension of input vector **x**

N : number of training data vectors **x**

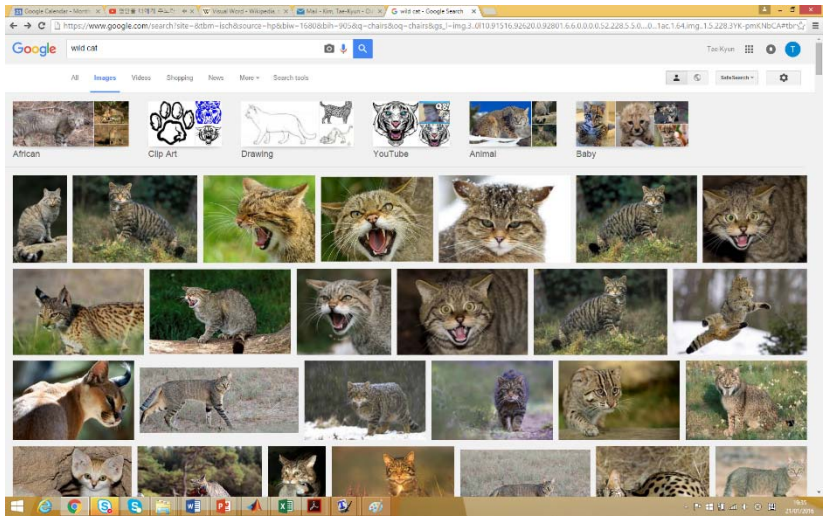
I : image

y(**x**) : SVM output (before discretization)

t_n : binary target variable of $\boldsymbol{x}_n$

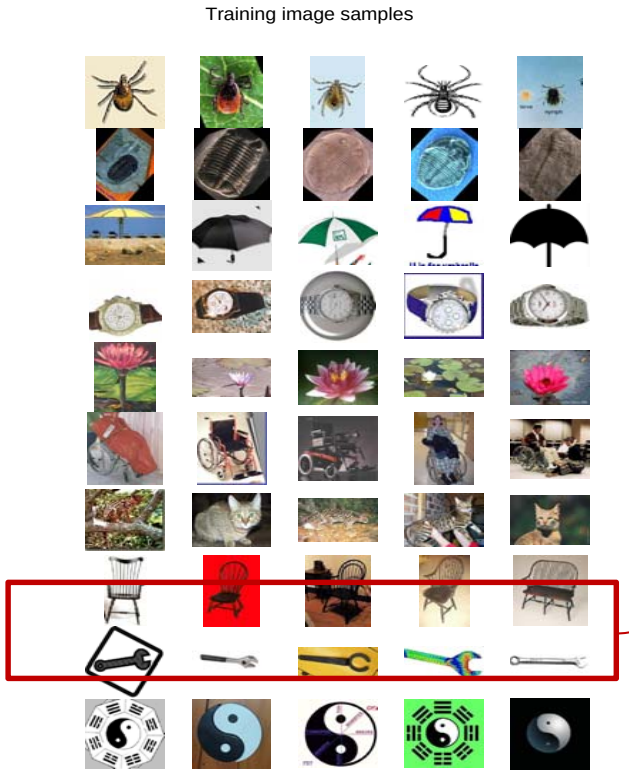
Visual object categorisation

- Widespread mobile phones and consumer-level cameras produce a fast-growing number of digital image collections.
- For managing such large-scale image data, it is essential to have access to high-level information about objects contained in images.
- An appropriate model and categorization method of image contents allows us to efficiently search, recommend, react to or reason with new image instances.
- The concerned problem is called *(generic) visual object categorization*.
- Processes need to be generic to cope with many object types.



Visual object categorisation challenges

– These processes should handle variations in view, imaging, lighting and occlusion, as well as large intra-class variations, typical of semantic classes of everyday objects.



Caltech101 data set

Examples of intra-class variation

Visual object categorization vs other problems

Object Identification:

- This concerns the identification of particular object instances.
- It would distinguish between images of e.g. two structurally distinct chairs, while categorization would place them in the same class.

Image Retrieval:

- This refers to the process of retrieving images on the basis of low-level image features, given a query image or manual description of the low-level features.

Detection:

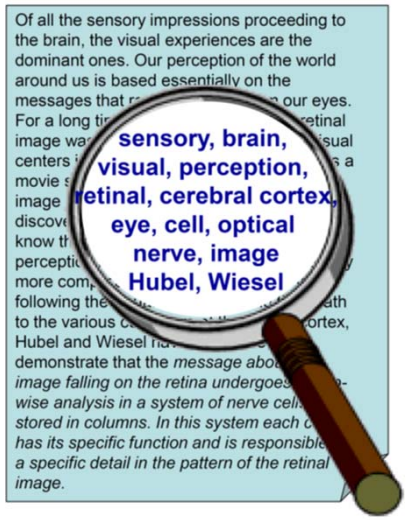
- This refers to deciding whether or not and where a member of *one visual category* is present in a given image.
- While it is possible to perform generic categorization by applying a detector for each class of interest to a given image, this approach is inefficient given a large number of classes.
- Most existing detection methods require precise manual alignment of the training images and the segregation of these images into different views, neither of which is necessary for the bag-of-words approach.

Bag of visual words

- The task-dependent and evolving nature of visual categories motivate an example based machine learning approach.
- We learn a *bag of (visual) words* approach to visual categorization.
- A bag of visual words corresponds to a histogram of the number of occurrences of particular image patterns in a given image.
- The main advantages are its simplicity, its computational efficiency and its invariance to affine transformations (translation, scaling, reflection, rotation, etc), as well as occlusion, lighting and intra-class variations to a certain degree.

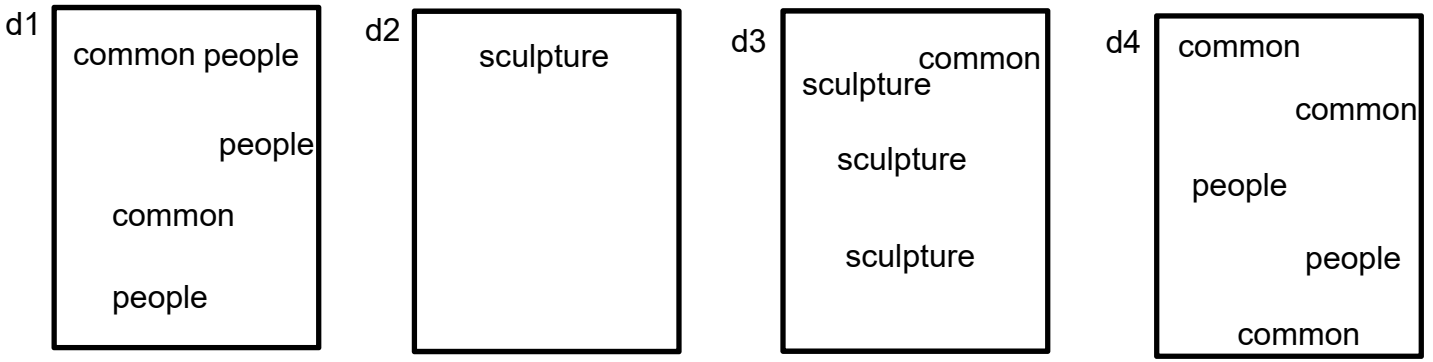
Bag of words

- *Bag of visual words* approach is motivated by an analogy to learning methods using the *bag-of-words* for natural language processing and information retrieval.
- The bag of words is a simplifying representation of a text (such as a sentence or a document) as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity.
- The bag of words is commonly used in methods of document classification, where the (frequency of) occurrence of each word is used as a feature for training a classifier.



Bag of words

- The following models a text document using bag of words.
- They have e.g. 3 distinct words.
- Using the list of words, each document is represented by a 3 entry vector, where each entry refers to count of the corresponding word in the list (this is also the histogram representation).

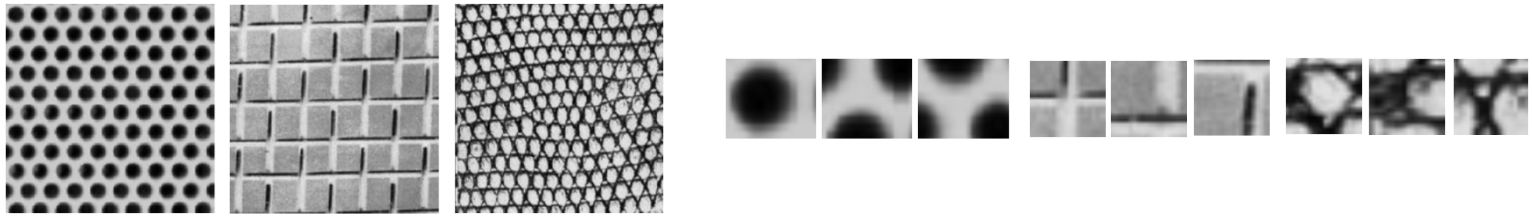


Bag of words

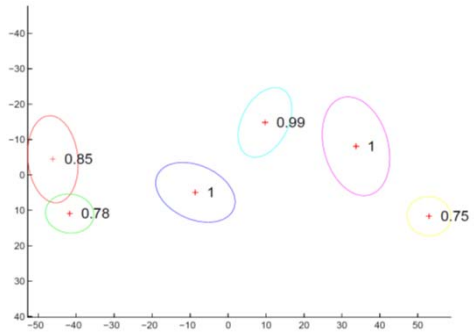
	d1	d2	d3	d4
common	2	0	1	3
people	3	0	0	2
sculpture	0	1	3	0
…	…	…	…	…

Text categorization to visual categorization

- The idea of clustering invariant descriptors of image patches has previously been applied to texture classification (e.g. Leung & Malik 2001; Schmid 2001)
- While we use clustering to obtain quite high-dimensional feature vectors, these texture recognizers use clustering to obtain relatively low-dimensional histograms (textons).



- A visual categorization method based on invariant descriptors of image patches was proposed (Fergus et al 2003). It exploits a probabilistic model that combines likelihoods for appearance and relative scale and position. The method is less time-efficient.



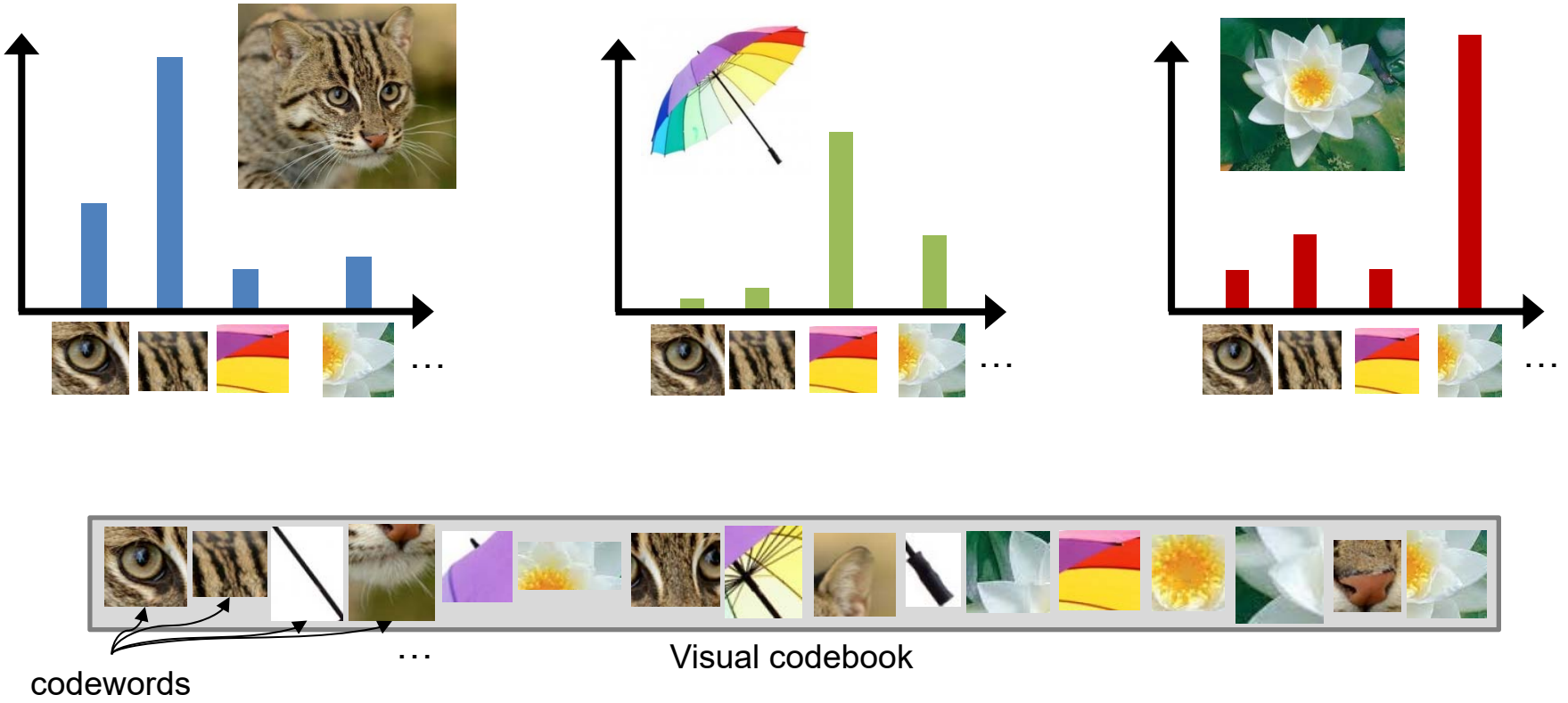
Bag of visual words model

- Visual words are base elements to describe an image.
- They are small parts (patches) of an image which carry some kind of information related to the features (such as color, shape, texture or certain descriptors).
- An object image is as a collection of visual words.

Object → **Bag of 'words'**

Bag of visual words model

- Independent features are called codewords. A codeword is a representative of similar patches.
- Then, each patch in an image is mapped to a certain codeword and the image can be represented by the histogram of the codewords.

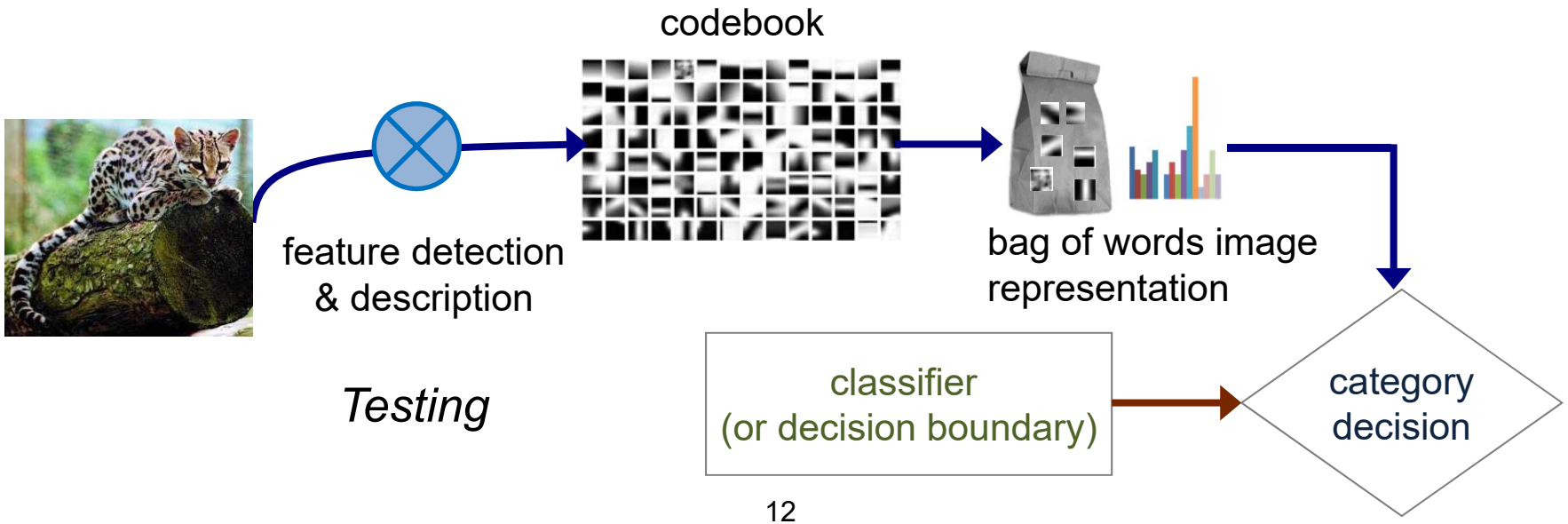


Bag of visual words

The main steps in the *testing* are:

- Step 1: Detection and description of image patches.
- Step 2: Assigning patch descriptors to a set of predetermined clusters (a visual codebook or vocabulary) by a vector quantization algorithm.

  Constructing a *bag of visual words*, which counts the number of patches assigned to each cluster.
- Step 3: Applying a multi-class classifier, treating the bag of words as the feature vector, and determine which category or class to assign to the image.



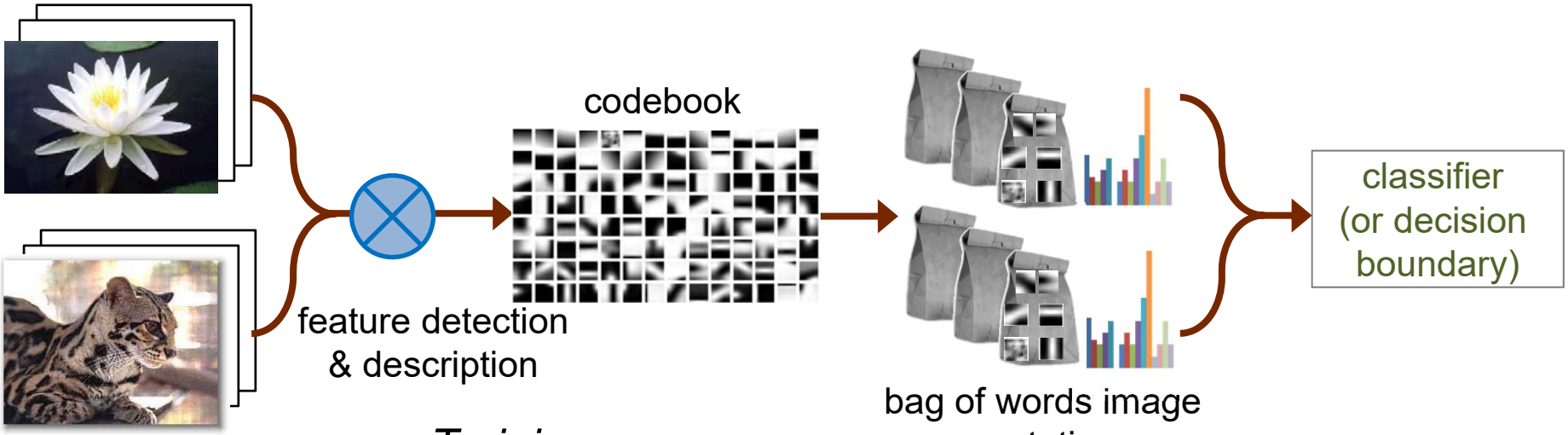
Bag of visual words

- These steps are ideally designed to maximize classification accuracy while minimizing computational effort.
- The descriptors extracted in the first step should be invariant to variations that are irrelevant to the categorization task (image transformations, lighting variations and occlusions) but rich enough to carry information to be discriminative at the category level.
- The vocabulary size used in the second step should be large enough to distinguish relevant changes in image parts, but not so large as to distinguish irrelevant variations such as noise.
- In our case “codewords” do not necessarily have an intuitive meaning such as “eyes”, or “car wheels”, nor is there an obvious best choice of vocabulary.
- Rather, our goal is to use a vocabulary that allows good categorization performance on a given dataset.

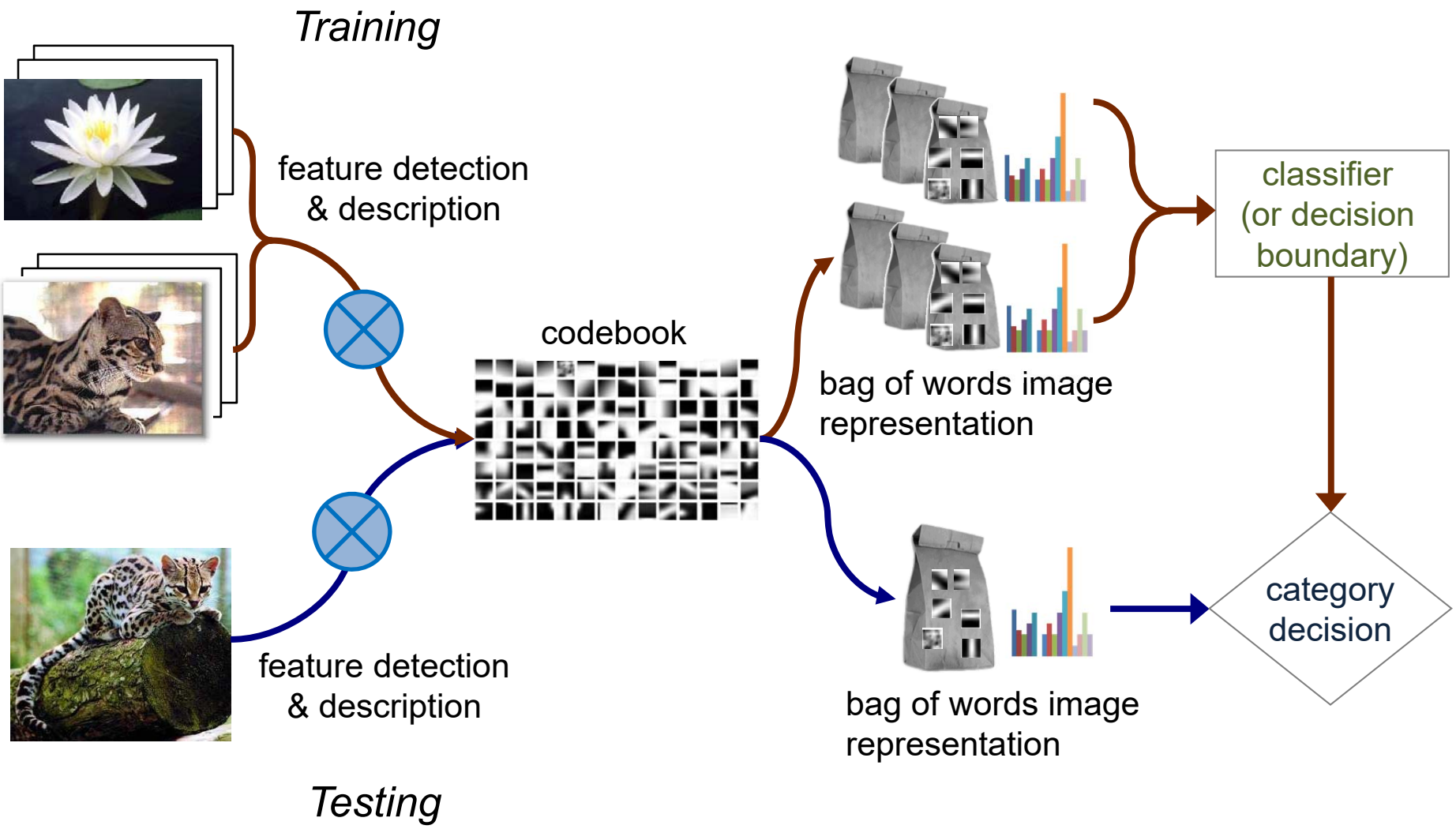
Bag of visual words

The steps in the *training* are

- Step 1: Detection and description of image patches for a set of labelled training images.
- Step 2: Constructing a visual codebook (or vocabulary): which is a set of cluster centres.
- Step 3: Extracting bags of visual words using this vocabulary.
- Step 4: Training multi-class classifiers using the bags of words as feature vectors.
- Step 5: Selecting the vocabulary size and classifier parameters giving the best overall classification accuracy.

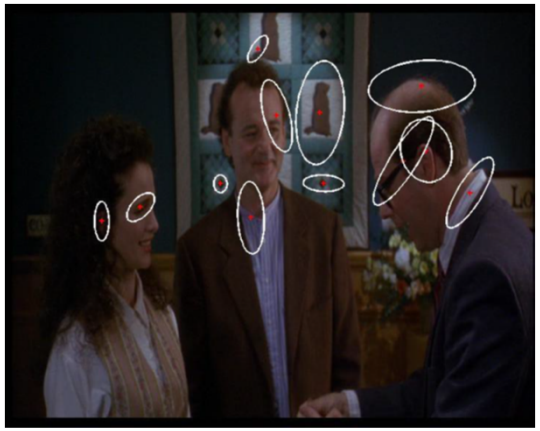


Training


Training
feature detection & description
codebook
bag of words image representation
classifier (or decision boundary)
feature detection & description
bag of words image representation
category decision
Testing

Feature detection & description

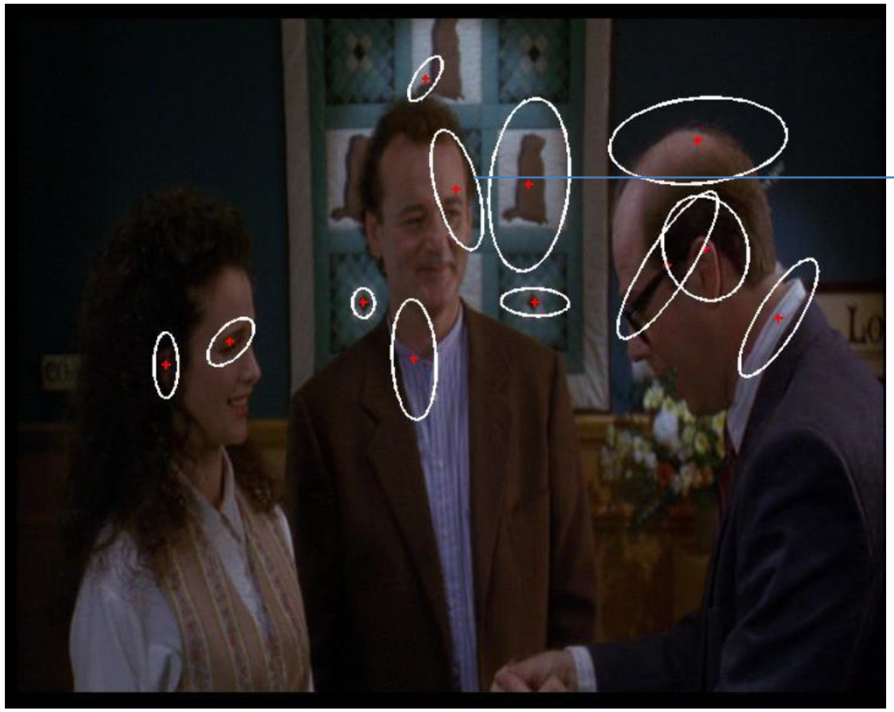
- In computer vision, local descriptors (i.e. features computed over limited spatial region) have proved useful for matching and recognition tasks, as they are robust to partial visibility and clutter.
- Such tasks require descriptors that are repeatable. Even if there is a transformation between two instances of an object, corresponding points are detected and (ideally) identical descriptors are obtained around each.
- This has motivated the development of different scale and affine (translation, reflection, rotation, etc) invariant point detectors, as well as descriptors that are resistant to geometric and illumination variations.
- Scale Invariant Feature Transform (SIFT) (Lowe 99) is one of the most popular descriptors.



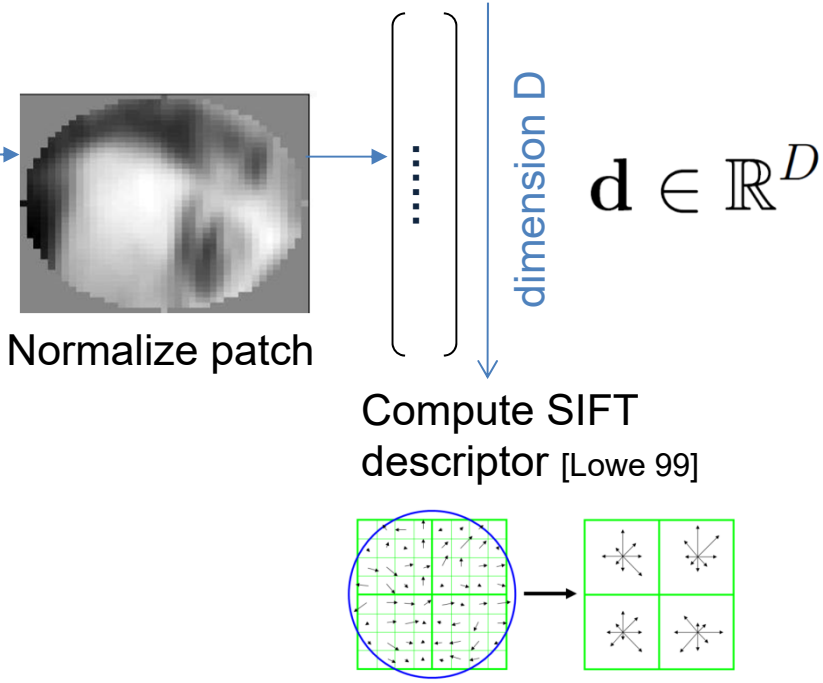
Detecting patches

Feature detection & description

– SIFT descriptor is computed on an image neighborhood. They are Gaussian derivatives computed at 8 orientation planes over a 4x4 grid of spatial locations, giving a 128-dimension vector.

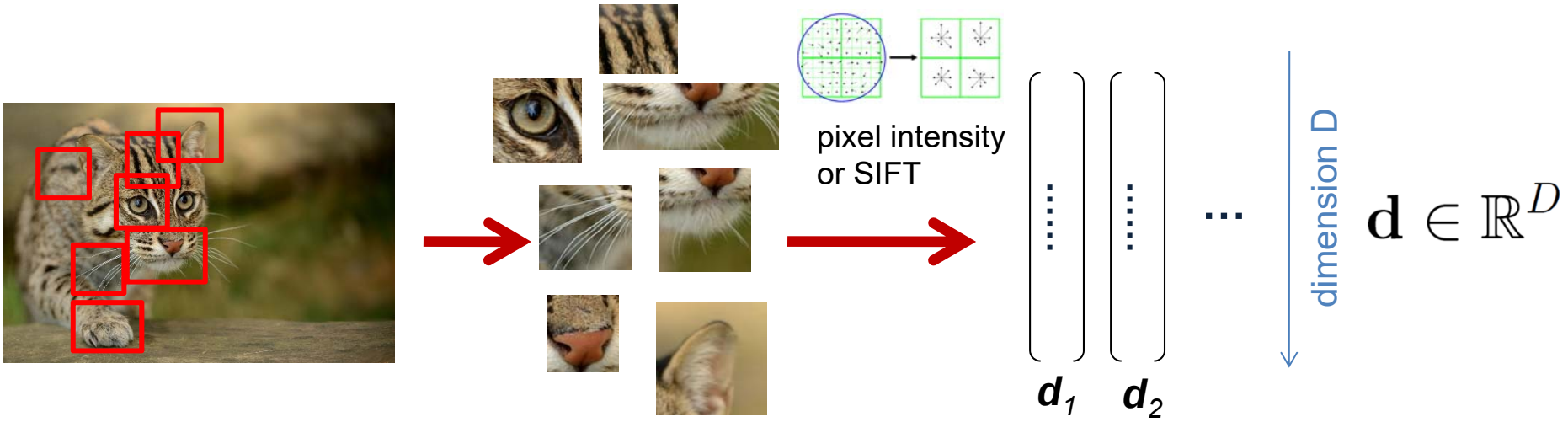


Detect patches [Mikojaczyk and Schmid 02] [Mata, Chum, Urban & Pajdla 02] [Sivic & Zisserman 03]



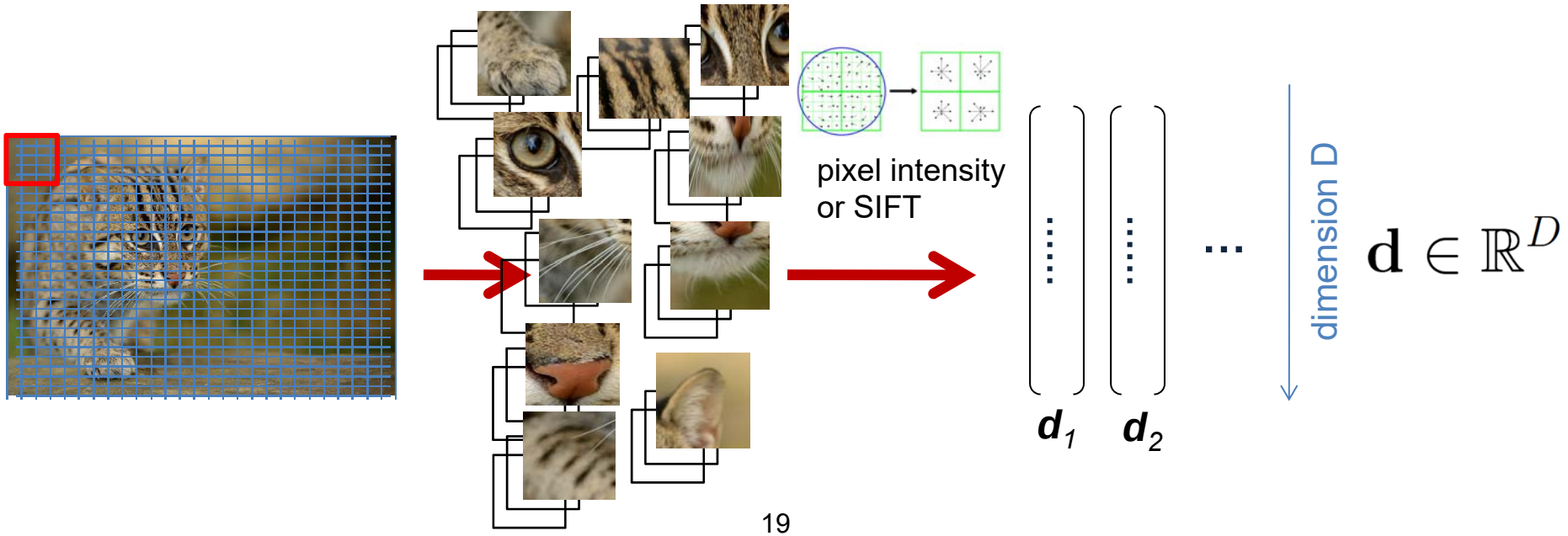
Visual vocabulary construction

- Interest points are detected from an image.
  - Corners, Blob detector, or SIFT (Scale-Invariant Feature Transform) detector
- Image patches are collected around the points and represented by descriptors.
  - Pixel intensity values or SIFT



Visual vocabulary construction

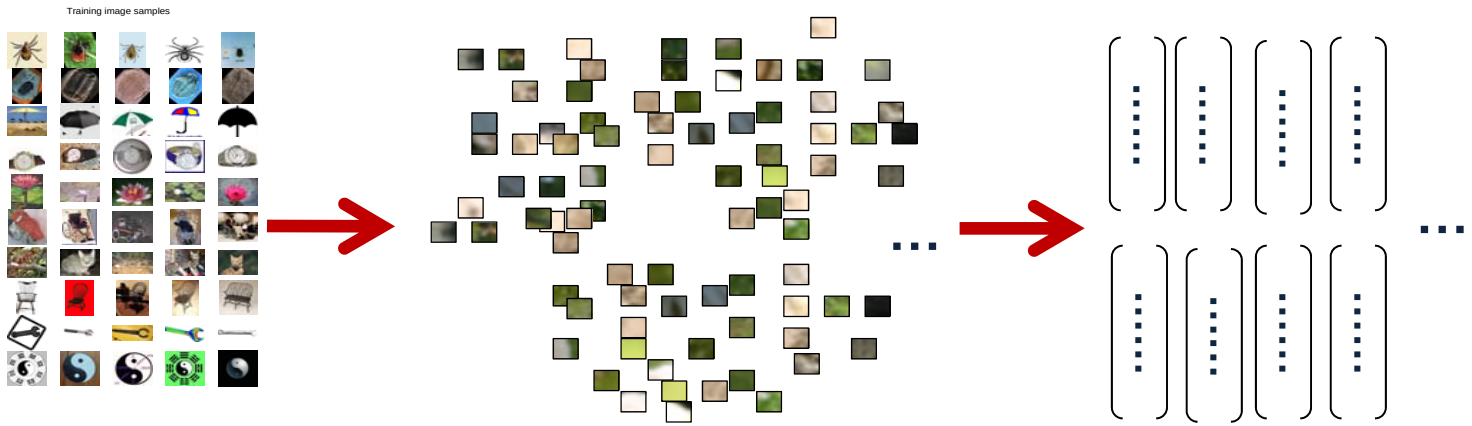
- Interest points are detected from an image.
  - Corners, Blob detector, or SIFT (Scale-Invariant Feature Transform) detector
- Image patches are collected around the points and represented by descriptors.
  - Pixel intensity values or SIFT
- Instead of a sparse interest point detector, we can use a dense grid. Image patches are collected around all points on the grid.



Visual vocabulary construction

- A large number of image patches are collected from a training set of images.
- Image patches are represented as descriptor vectors.
- Visual words (real-valued descriptor vectors) can be compared using Euclidean distance:

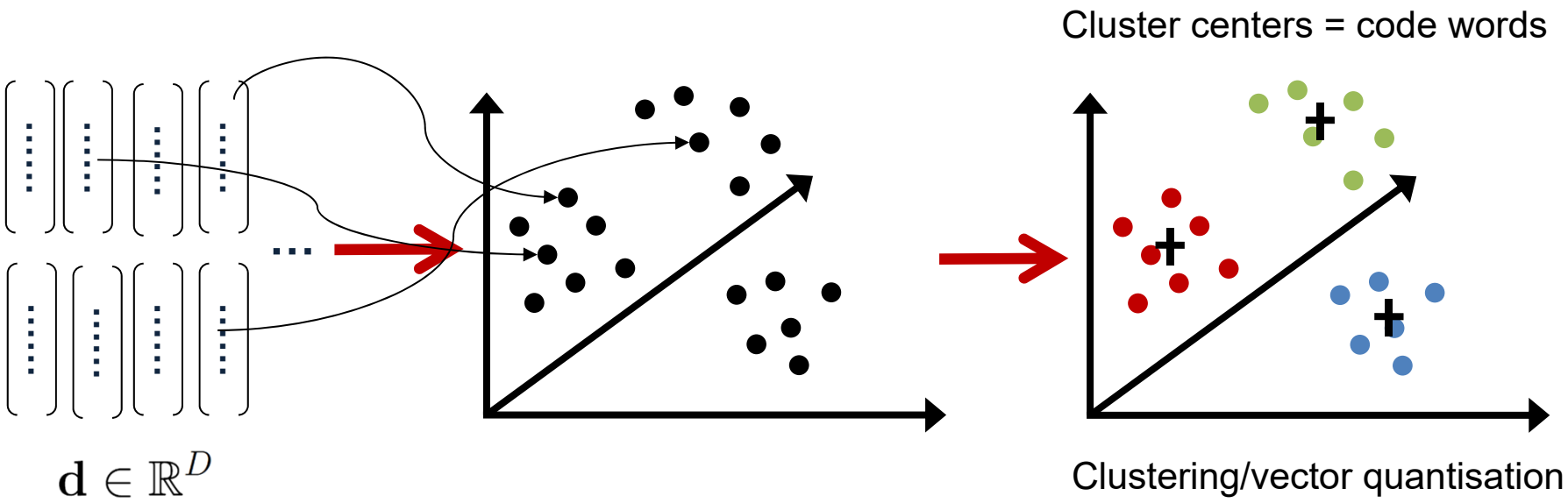
$$E(\mathbf{d}_1, \mathbf{d}_2) = ||\mathbf{d}_1 - \mathbf{d}_2|| = \sqrt{\sum_{i=1}^{D} (\mathbf{d}_1^i - \mathbf{d}_2^i)^2}$$



A training set of images

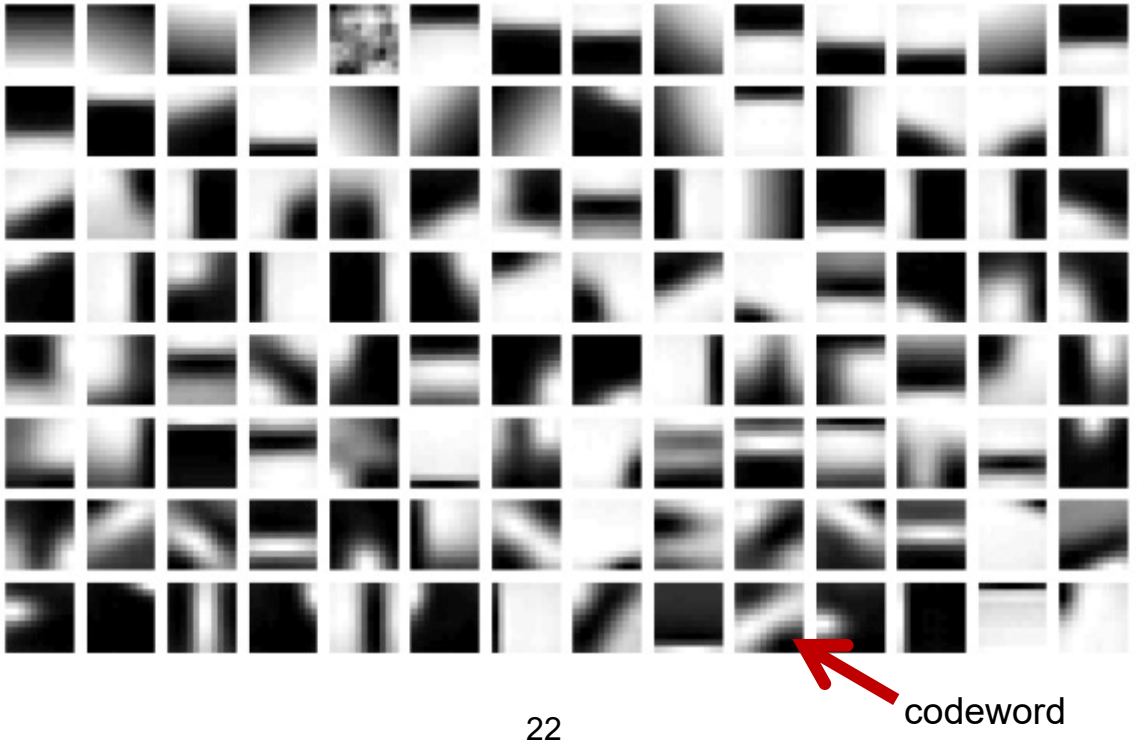
K-means clustering

- These vectors are divided into *K* groups which are similar, essentially clustered together, to form a codebook, by K-means.
- The two steps repeat until no change in membership (randomly initialize the means):
  - Step1: Assign each data point to the cluster whose center is nearest
  - Step2: Compute the center of each cluster as the data mean of the respective cluster
- The *K* cluster centers (mean vectors) form a visual codebook.



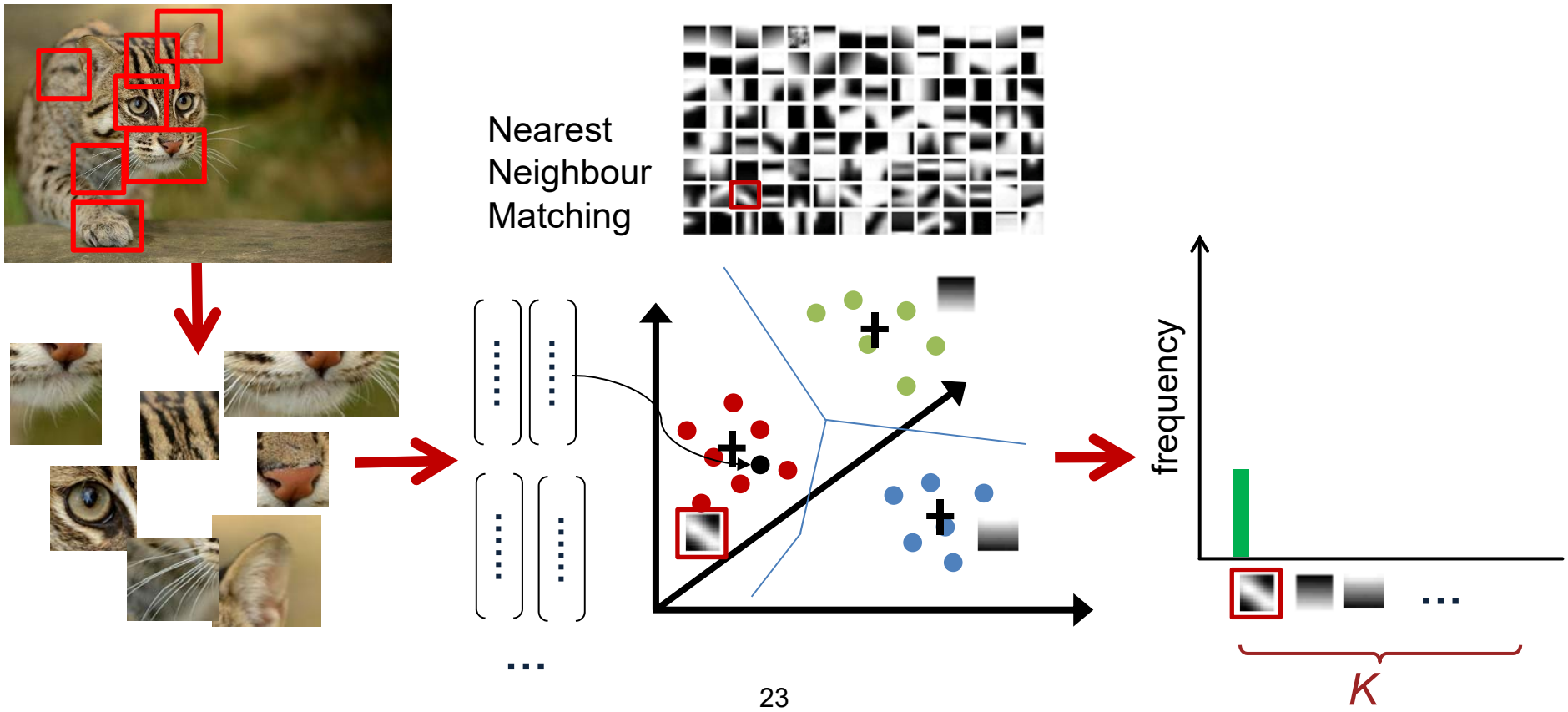
Visual vocabulary construction

- Visualisation of the codebook:
  - Descriptor vectors are linked to respective image patches.
  - We use the cluster membership found in K-means, and take the mean of image patches in each cluster.
- The codebook has *K* codewords (the vocabulary size is *K*).



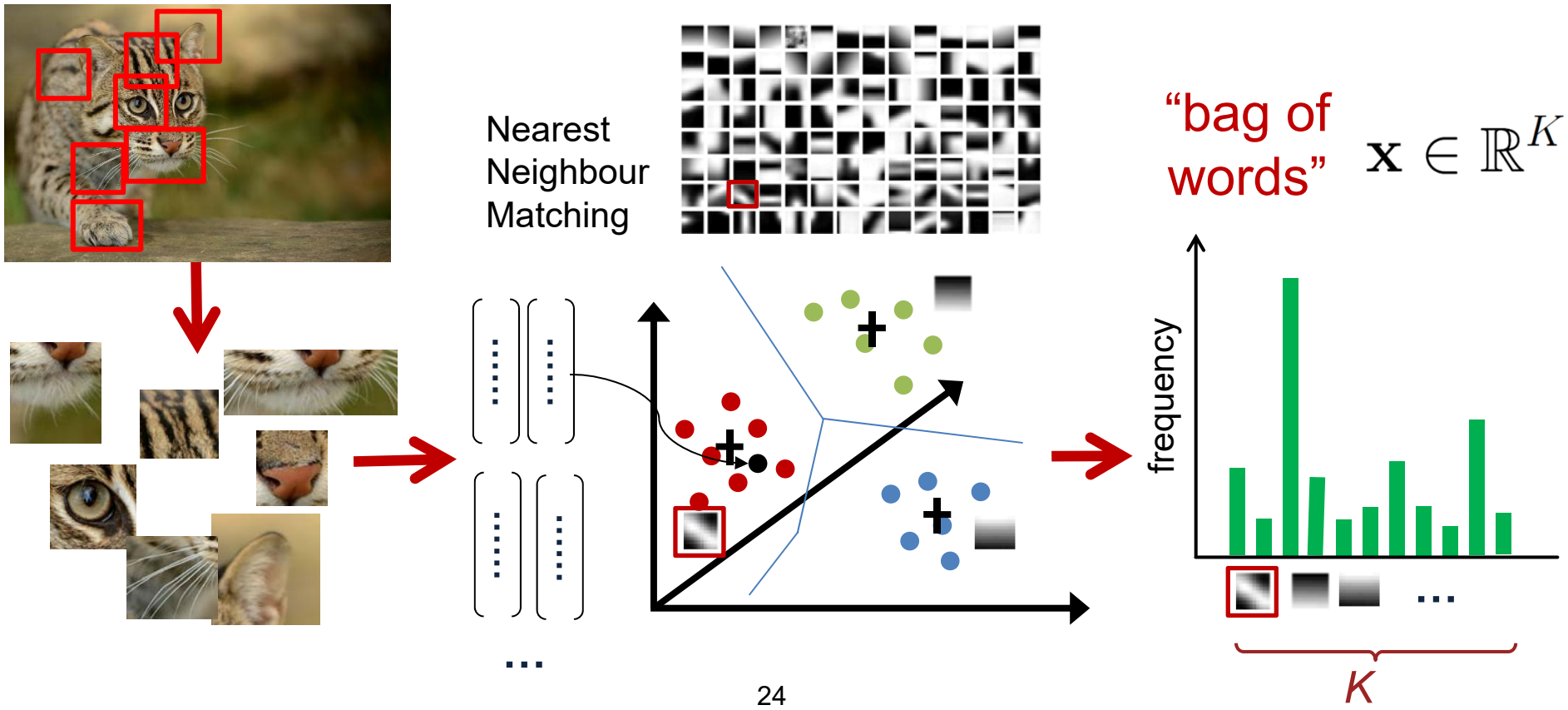
Histogram of visual words (vector quantisation)

- Every visual word is compared with codewords and assigned to the nearest codeword (Nearest Neighbour matching).
- Histogram bins are codewords and each bin counts the number of words assigned to each codeword.



Histogram of visual words (vector quantisation)

- Every visual word is compared with codewords and assigned to the nearest codeword (Nearest Neighbour matching).
- Histogram bins are codewords and each bin counts the number of words assigned to each codeword.



Summary: visual vocabularies

- The vocabulary (or codebook) is a way of constructing a feature vector for classification, which relates “new” descriptors in query images to descriptors previously seen in training.
- One extreme of this approach is to compare each query descriptor to all training descriptors: this is impractical given the huge number of training descriptors (# of images x # of descriptors per image).
- In practice, we find the best trade offs of accuracy and computational efficiency, which is obtained by intermediate sizes of clustering.

- Most clustering algorithms are based on iterative square error partitioning (e.g. K-means) or on hierarchical techniques.
  - Hierarchical techniques organize data in a nested sequence of groups which can be displayed in the form of a tree.

Summary: visual vocabularies by K-means

- The histogram is *K*-dimenional, a highly compact and robust representation of images.
- The histogram representation greatly facilitates modelling images and categorising them.
- *K*-means is an unsupervised learning method.

Issues:

- Codeword assignment (vector quantisation process) by Nearest Neighbor matching is time-demanding.
  - Computational efficient method by vocabulary trees (e.g. Nister & Stewenius 2006)
- *K*-means algorithm converges only to local optima of the squared distortion.
  - It depends on initialisation.
  - Randomly initialize means, run several times, and select the best clusters in terms of the squared distortion.
  - PCA on the descriptor vectors, when the descriptor dimension is high, can be considered.

Summary: visual vocabularies by K-means

Issues:

- *K*-means algorithm does not determine the parameter *K*. How to choose the vocabulary size?
  - Too small: codewords not representative of all patches
  - Too large: quantization artifacts, overfitting
  - There exist methods allowing to automatically estimating the number of clusters based on Bayesian Information Criterion (e.g. Pelleg & Moore, 2000).
  - We run *K*-means with different number of desired clusters (*K*) and different initialisation. We select the final clustering giving the lowest risk (or error) in categorization.

K-means Clustering

Backgrounds:

Vector calculus - Matrix and vector derivatives

Optimisation (EE429) - Gradient method

Further reading:

Appendix A: Mathematical Foundations, R.Duda, P.Hart, D.Stork, Pattern Classifcation (Second Edition), JOHN WILEY & SONS, Inc. 2001.

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.320.4607&rep=rep1&type=pdf

http://cns-classes.bu.edu/cn550/Readings/duda-etal-00.pdf

K-means clustering

- Given a data set $\{\boldsymbol{d}_{1,\ldots,} \boldsymbol{d}_{N'}\}$ of N' observations in a D-dimensional space, our goal is to partition the data set into K clusters or groups.

- The vectors μ_i, where $i = 1,\ldots,K$, represent i-th cluster, e.g. the centers of the clusters.

- Binary indicator variables are defined for each data point $\boldsymbol{d}_n$, s.t. $r_{ni} \in \{0, 1\}$, where $i = 1,\ldots,K$.

- 1-of-K coding scheme: $\boldsymbol{d}_n$ is assigned to cluster i then $r_{ni} = 1$, and $r_{nj} = 0$ for $j \neq i$.

K-means clustering

– The objective function that measures distortion is

$$J = \sum_{n=1}^{N'} \sum_{i=1}^{K} r_{ni} ||\mathbf{d}_n - \mu_i||^2$$

– The square-error partitioning algorithm attempts to obtain the partition which minimizes the within-cluster scatter (alternatively, maximizes the between-cluster scatter).

– We ought to find $\{r_{ni}\}$ and $\{\mu_i\}$ that minimise J.

Iterative solution for K-means

First we choose some initial values for μ_i, $i = 1,...,K$, We iterate the two steps till convergence.

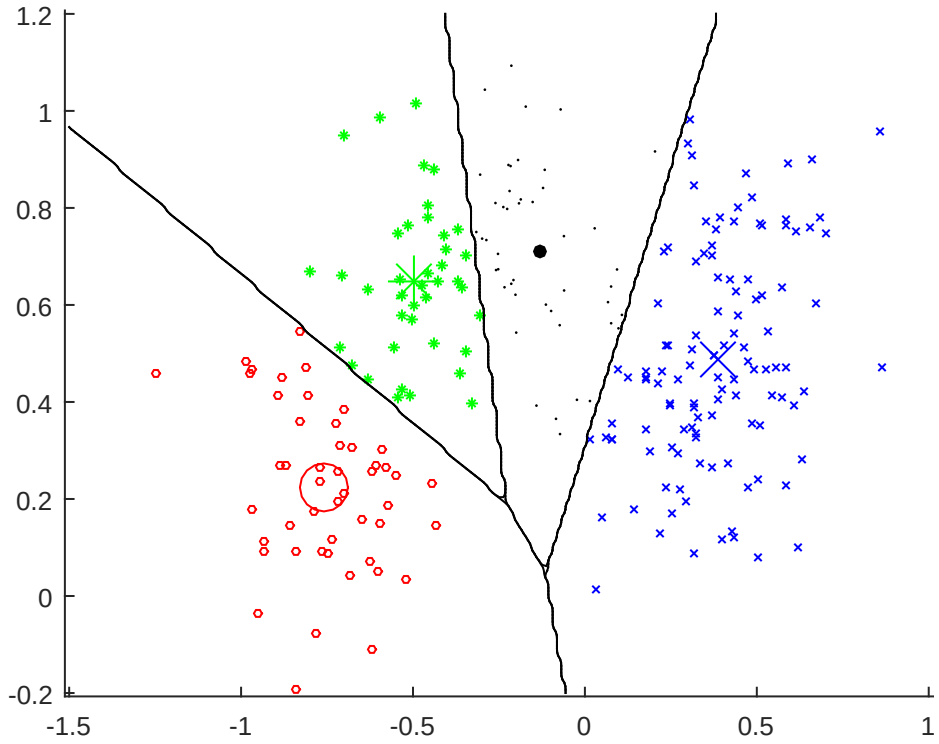
- Step 1: We minimise J with respect to r_{ni}, keeping μ_i fixed. J is a linear function of r_{ni}, we have a closed form solution

$$r_{ni} = \begin{cases} 1, & \text{if } i = \arg\min_j ||\mathbf{d}_n - \mu_j||^2 \\ 0, & \text{otherwise} \end{cases}$$

- Step 2: We minimise J with respect to μ_i keeping r_{ni} fixed. J is a quadratic of μ_i. We set its derivative with respect to μ_i to zero,

$$2\sum_{n=1}^{N'} r_{ni}(\mathbf{d}_n - \mu_i) = 0 \quad \longrightarrow \quad \mu_i = \frac{\sum_n r_{ni}\mathbf{d}_n}{\sum_n r_{ni}}$$

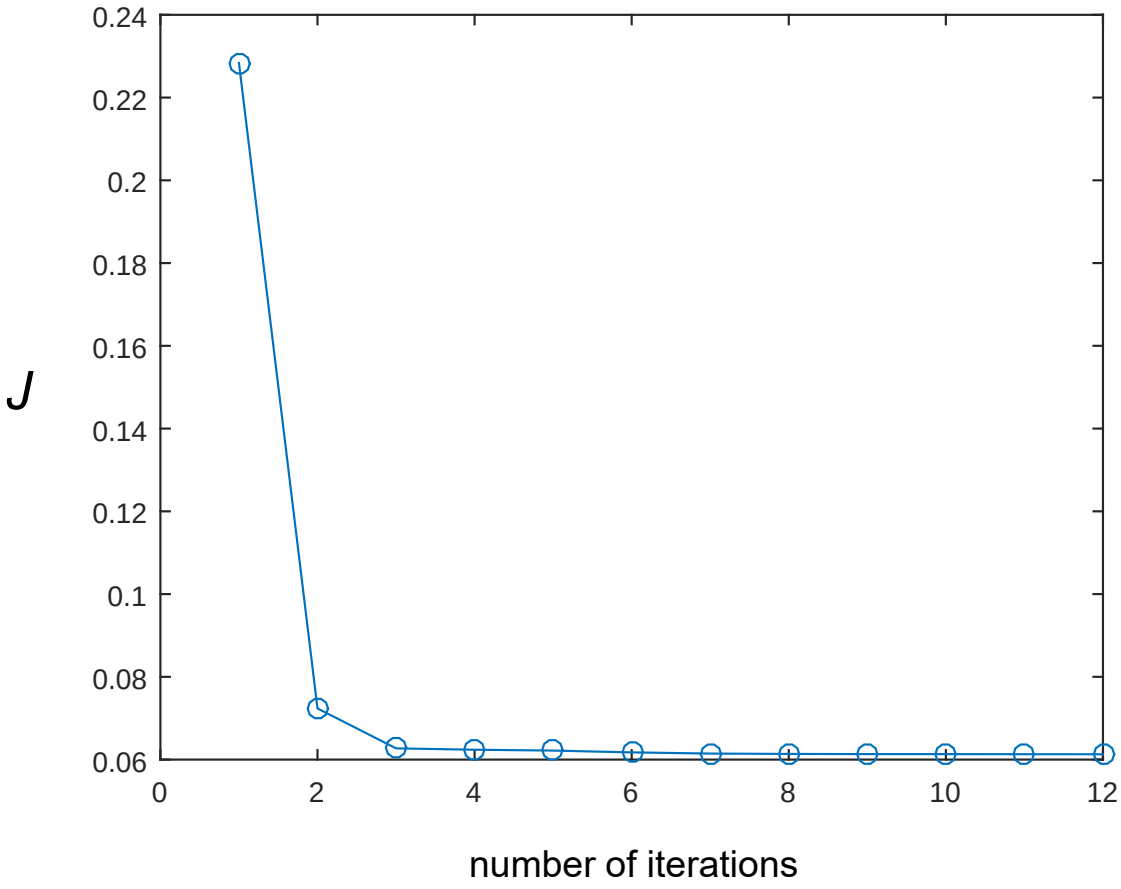
K-means clustering



K=4

K-means clustering

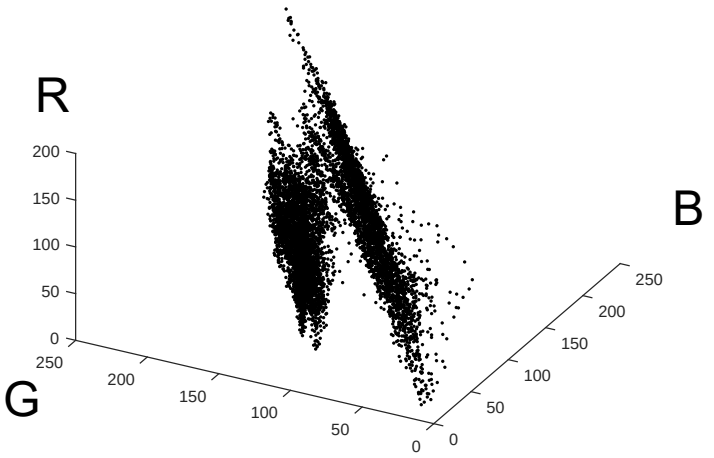
- It converges to a local minima.
- Its result depends on initial values of μ_i.



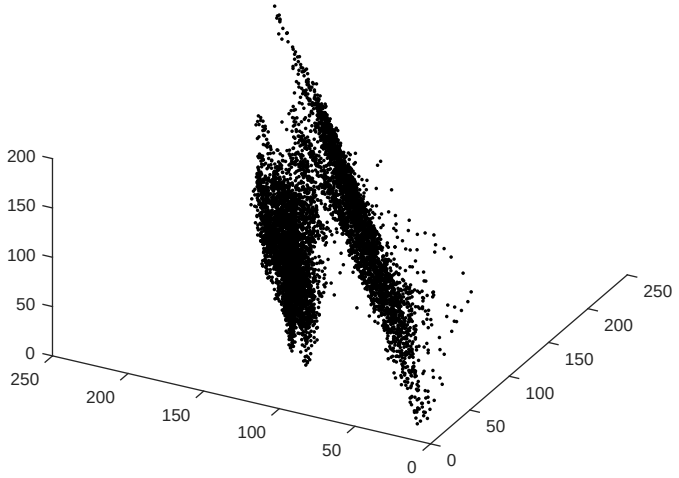
Pixel Clustering (Image Quantisation)

- Image pixels are represented by 3-dimensional vectors of R,G,B values.
- The vectors are grouped to *K* clusters, and represented by the mean values of the respective clusters.

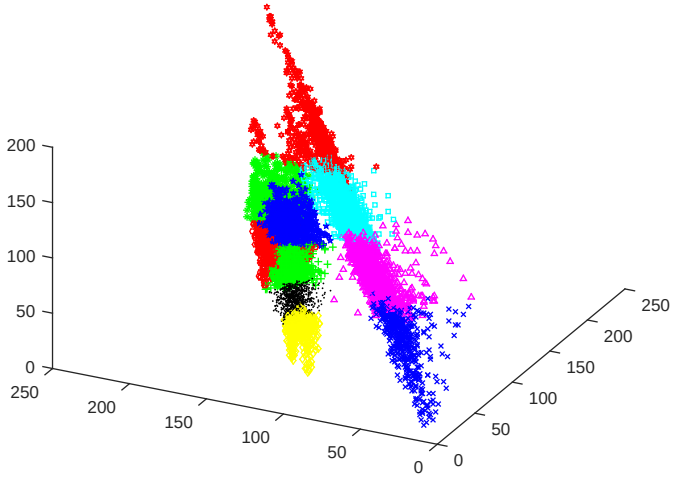
$$\mathbf{d}=\begin{pmatrix}R\\G\\B\end{pmatrix}\in\mathbb{R}^3$$



Pixel clustering (image quantisation)

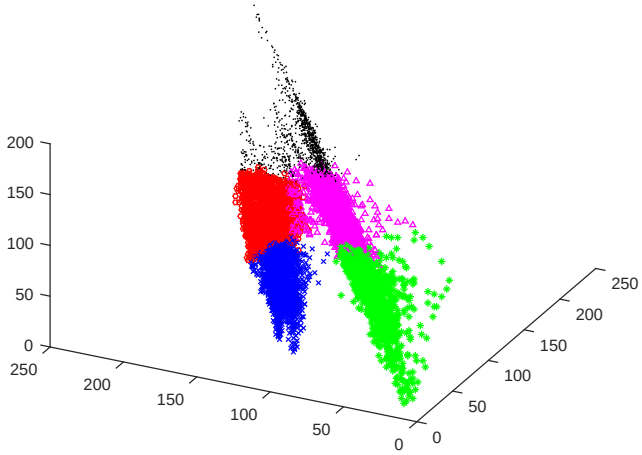


Input vector distribution (left) and image (right)

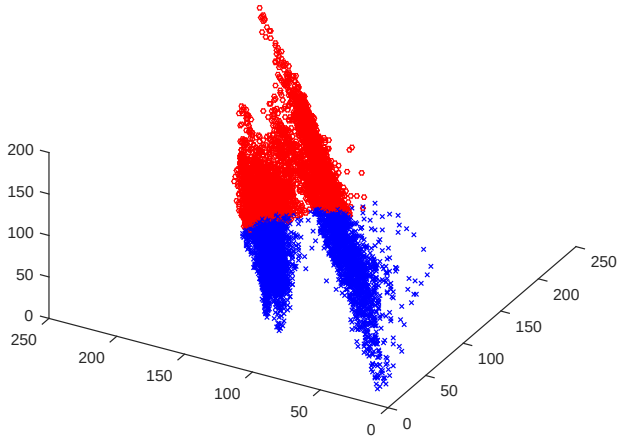


Vectors grouped to *K* clusters (left) and quantised image (right) , *K*=10

Pixel clustering (image quantisation)



Vectors grouped to *K* clusters (left) and quantised image (right) , *K*=5



Vectors grouped to *K* clusters (left) and quantised image (right) , *K*=2

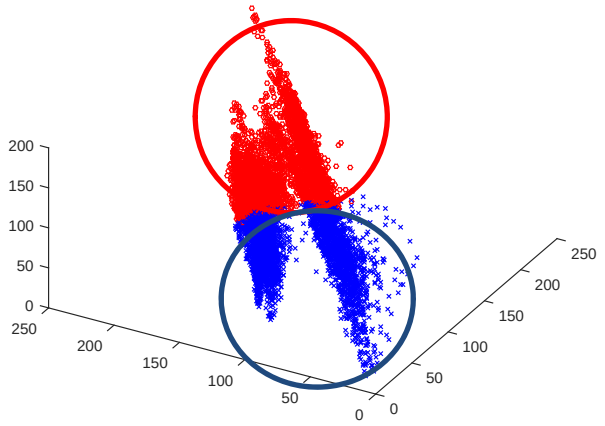
K-means pros and cons

Pros

- Simple and fast
- Guaranteed to converge in iterations

Cons

- Needs to pick K
- Sensitive to initialization (i.e. local minimum of the error function)
- Only finds spherical clusters
- Sensitive to outliers



Example of spherical clusters, *K*=2

Categorization by SVM

- Once descriptors have been assigned to clusters to form feature vectors, we reduce the problem of generic visual categorization to that of multi-class classification, with as many classes as defined visual categories.
- The categorizer performs two separate steps in order to predict the classes of images: training and testing.
- During training, (class) labeled data is sent to the classifier and used to learn a statistical decision procedure for distinguishing categories.
- Among many available classifiers, including NN (Nearest Neighbor) classifier, we can adopt the SVM (Support Vector Machine).
- It is often known to produce state-of-the-art results in high-dimensional problems.

Categorization by SVM

– The SVM classifier finds a hyperplane (or decision boundary) which separates two-class data.

– For given observations **x**, and corresponding labels t which takes values ±1, it finds a classification function:

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$$

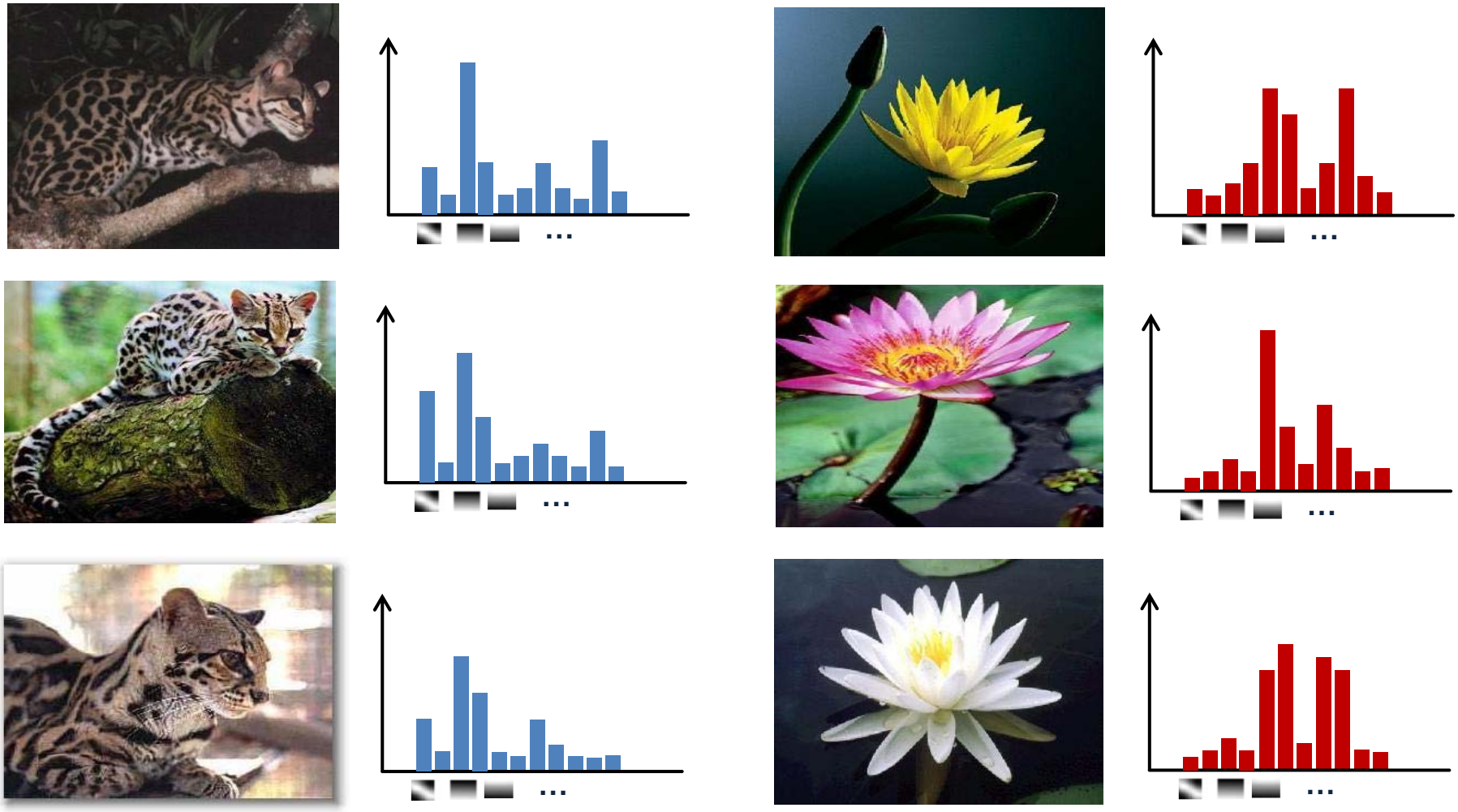
where **w**, *b* represent the parameters of the hyperplane.

– Data points **x** are classified by the sign of y(**x**).

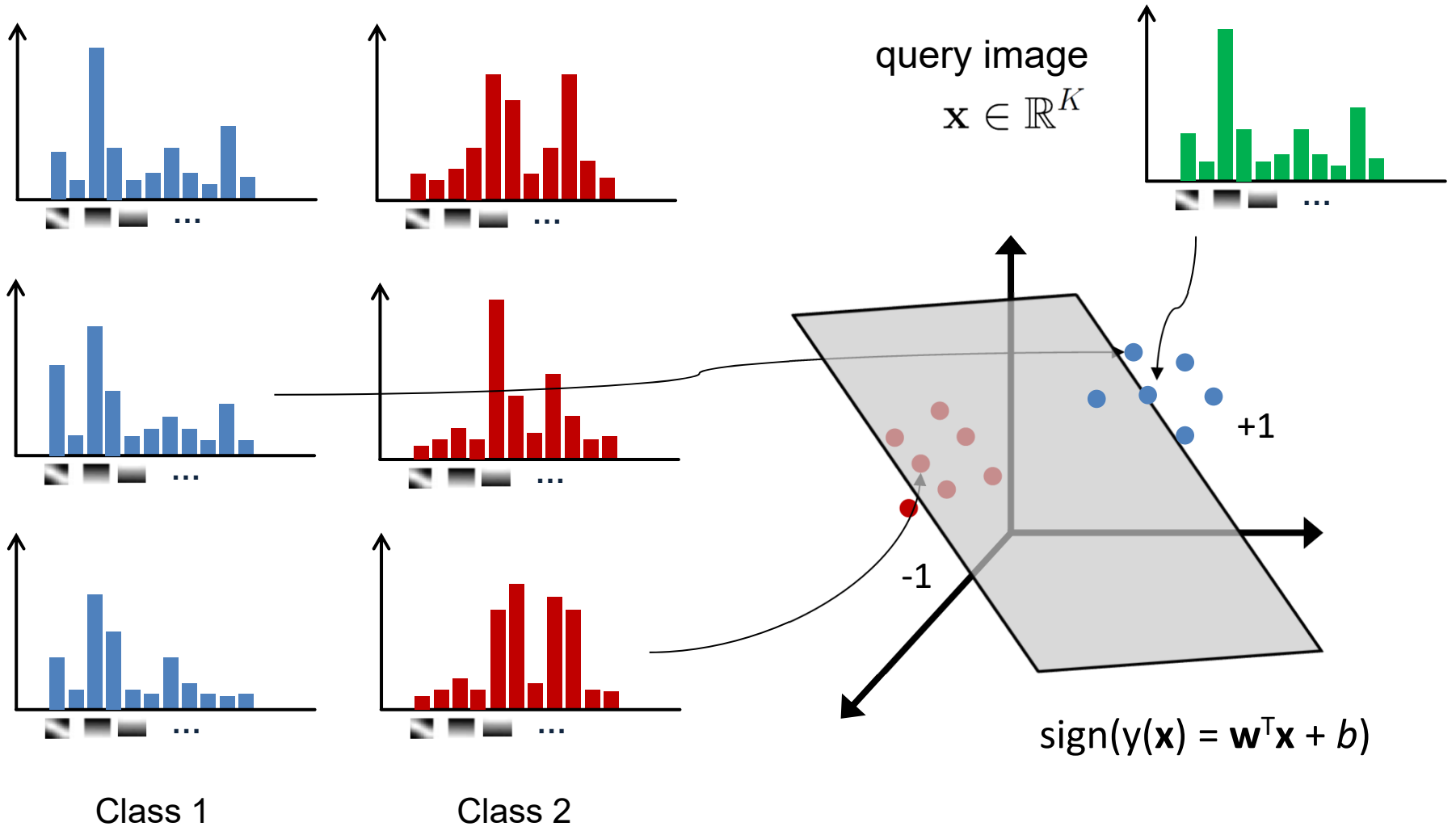
– In our case, the input feature vector **x** is the *K* binned histogram formed by the number of occurrences of each codeword in the image **I**.

– In order to apply SVM to multi-class problems, we take the multi-class extensions of SVM.

Category model



Discriminative classifier (linear classifier)



Video-based object recognition