1. Bag of visual words model: recognizing object categories

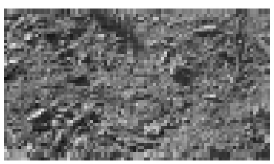
Problem: Image Classification

Given:

- positive training images containing an object class, and



- negative training images that don’t




Classify:

- a test image as to whether it contains the object class or not

?

Weakly-supervised learning

- Learn model from a set of training images containing object instances

- Know if image contains object or not
- But no segmentation of object or manual selection of features

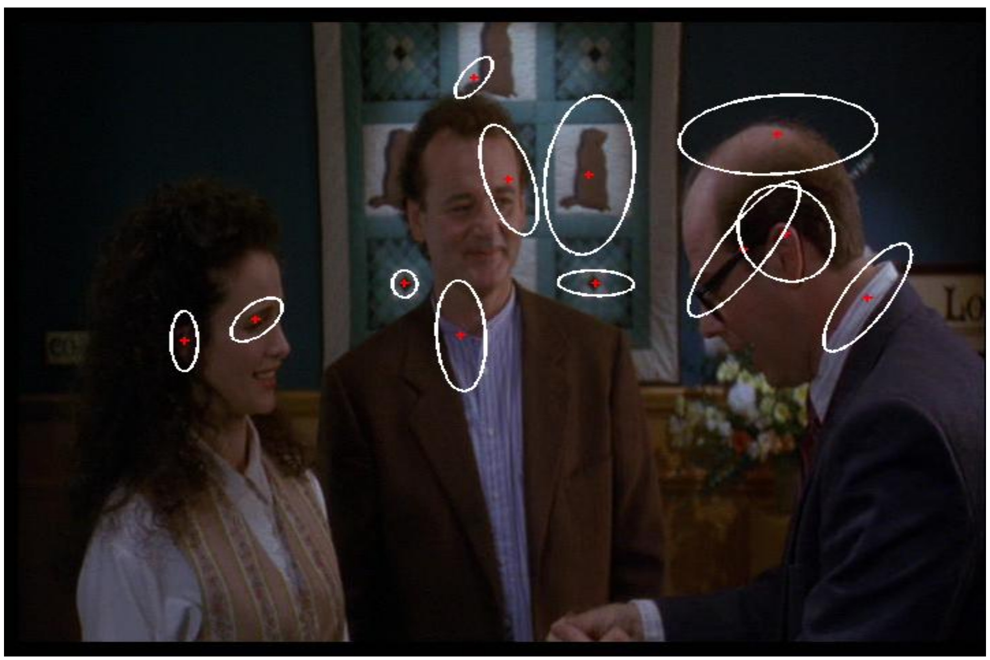
Three stages:

1. Represent each training image by a vector
   - Use a bag of visual words representation
2. Train a classify to discriminate vectors corresponding to positive and negative training images
   - Use a Support Vector Machine (SVM) classifier
3. Apply the trained classifier to the test image

Representation: Bag of visual words

Visual words are 'iconic' image patches or fragments

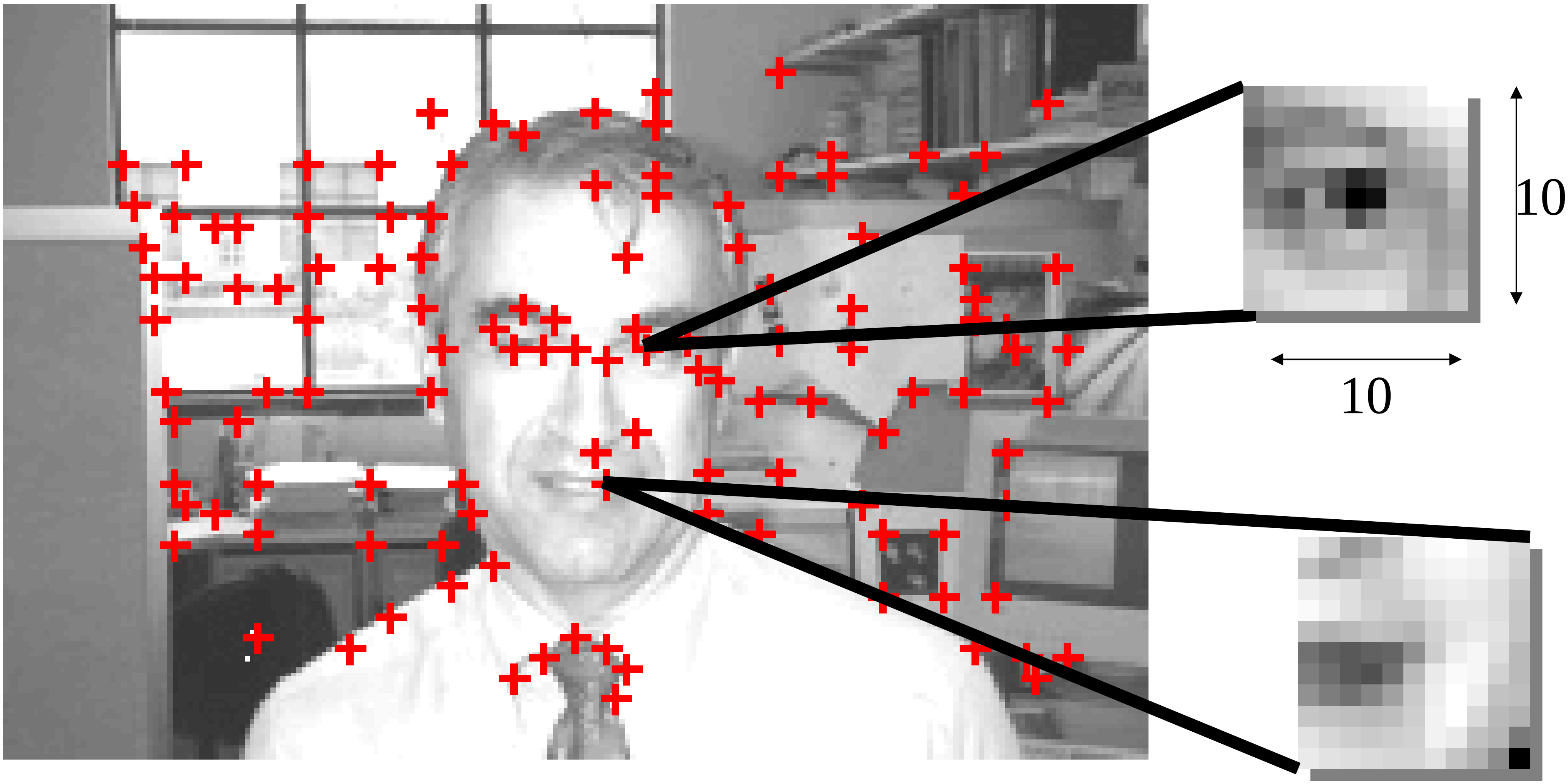
- represent the frequency of word occurrence
- but not their position



Image

Collection of visual words

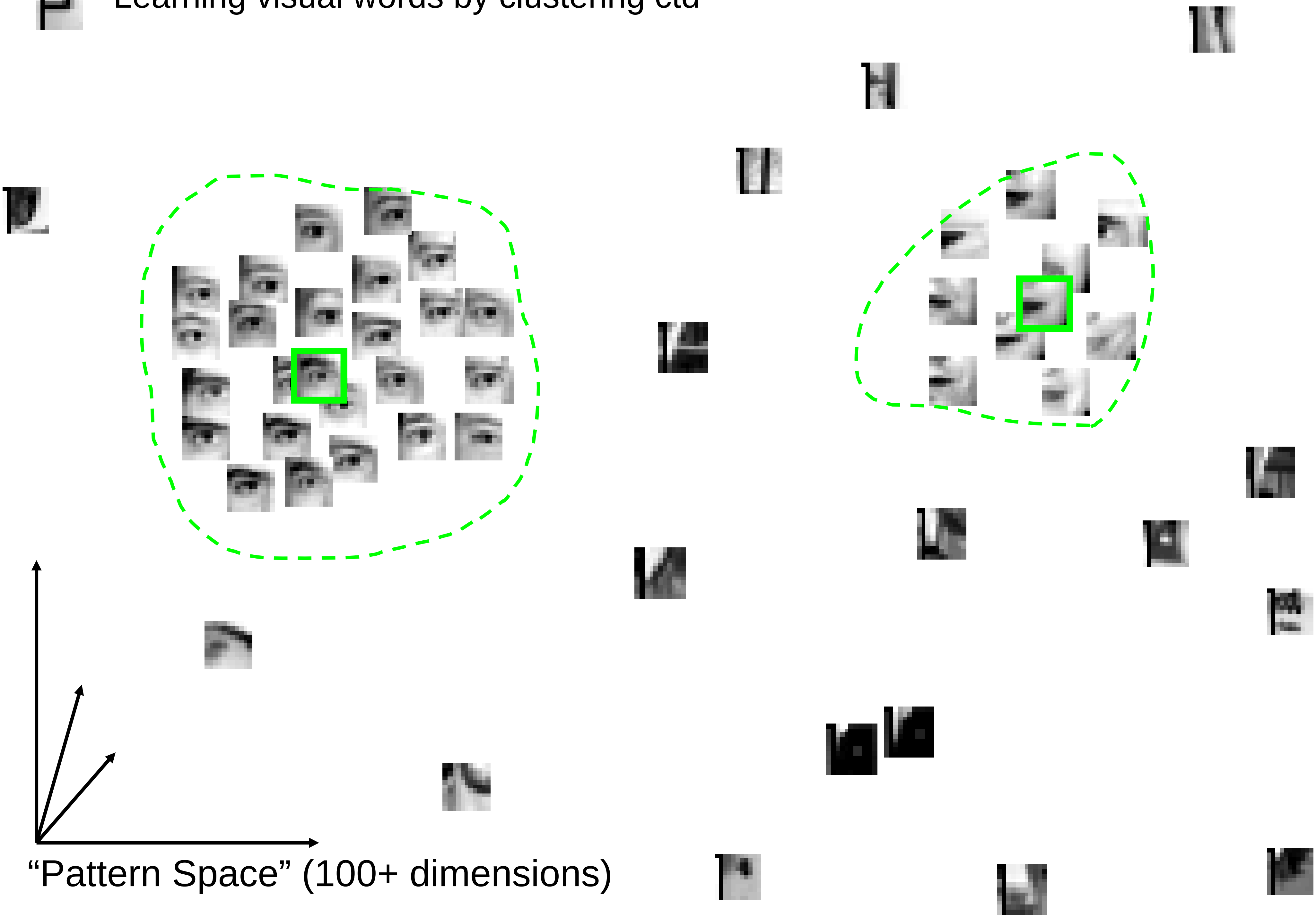
Example: Learn visual words by clustering



- Interest point features: textured neighborhoods are selected
- produces 100-1000 regions per image

Weber, Welling & Perona 2000

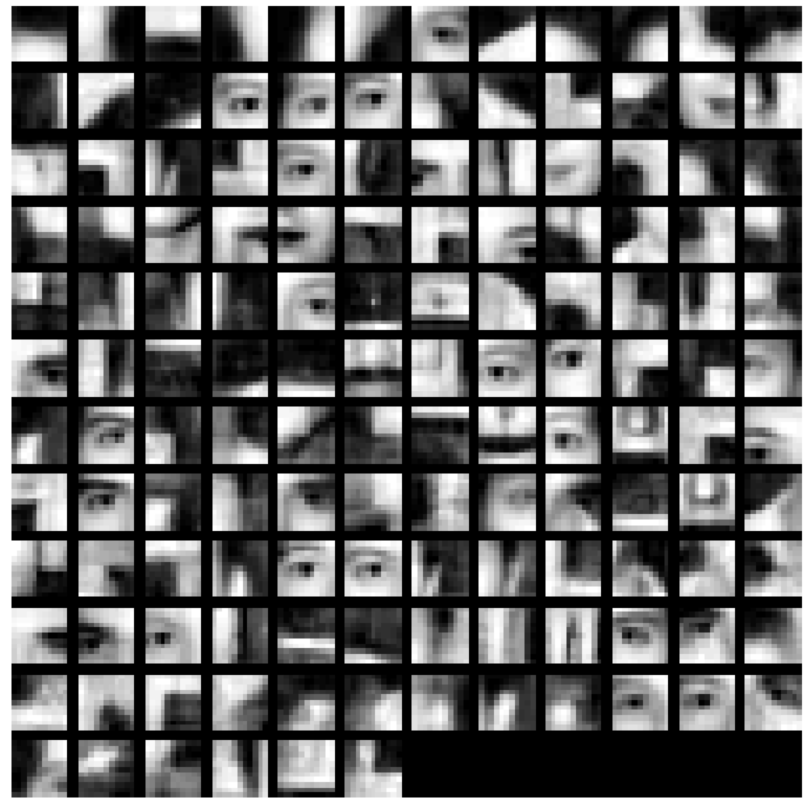
Learning visual words by clustering ctd



“Pattern Space” (100+ dimensions)

Example of visual words learnt by clustering faces

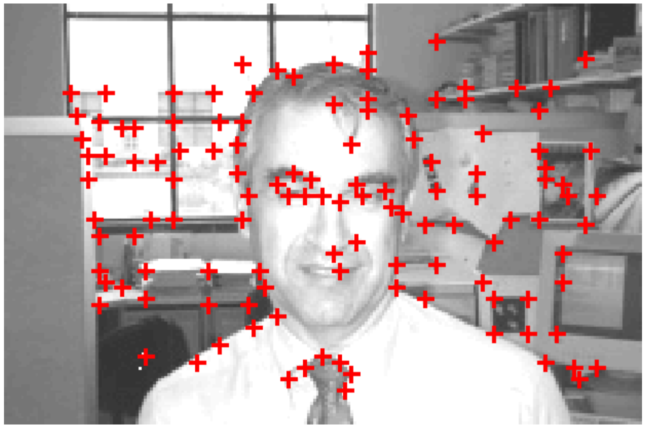
100-1000 images

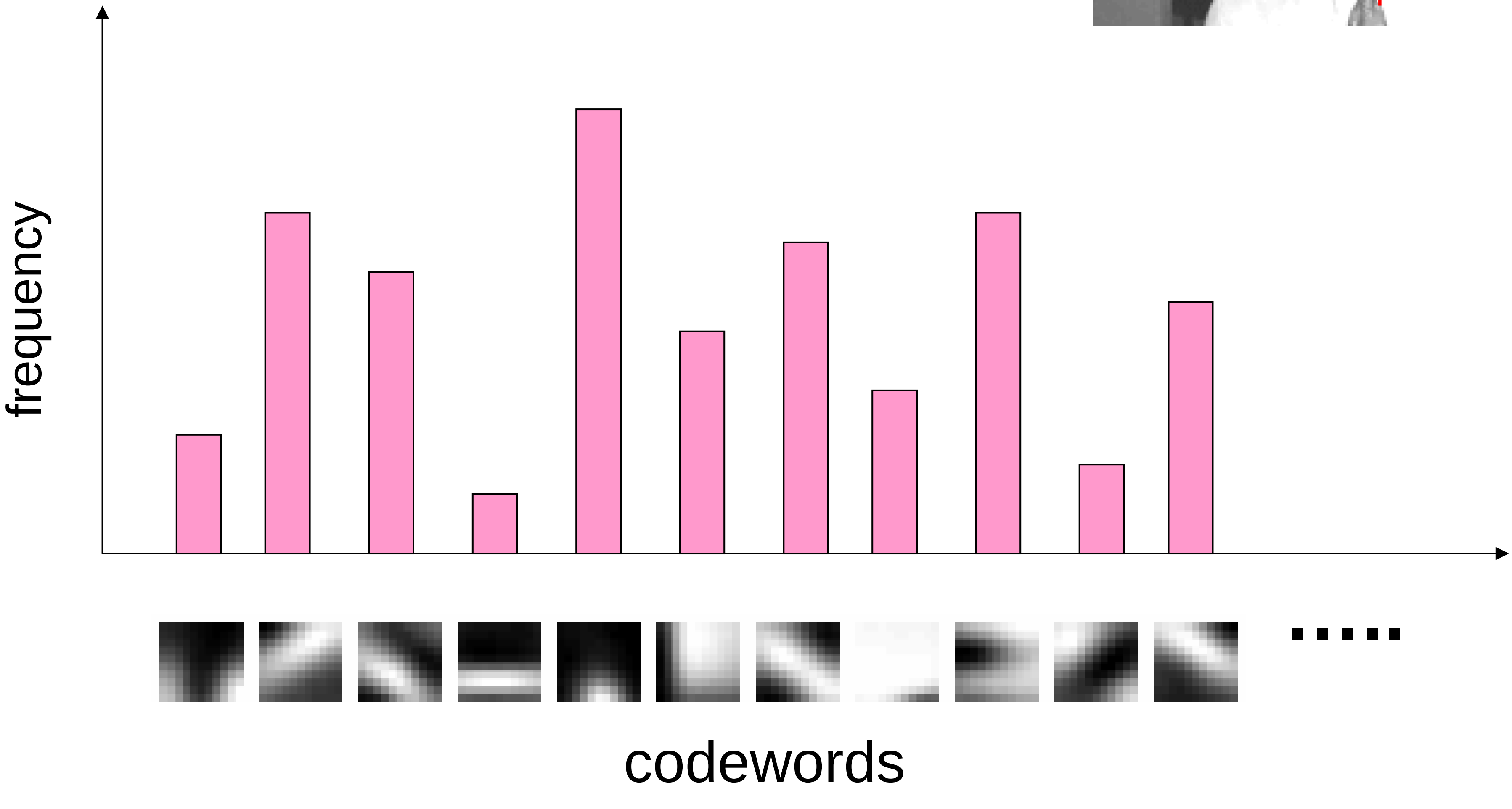


~100 visual words

Image representation – normalized histogram

- detect interest point features
- find closest visual word to region around detected points
- record number of occurrences, but not position





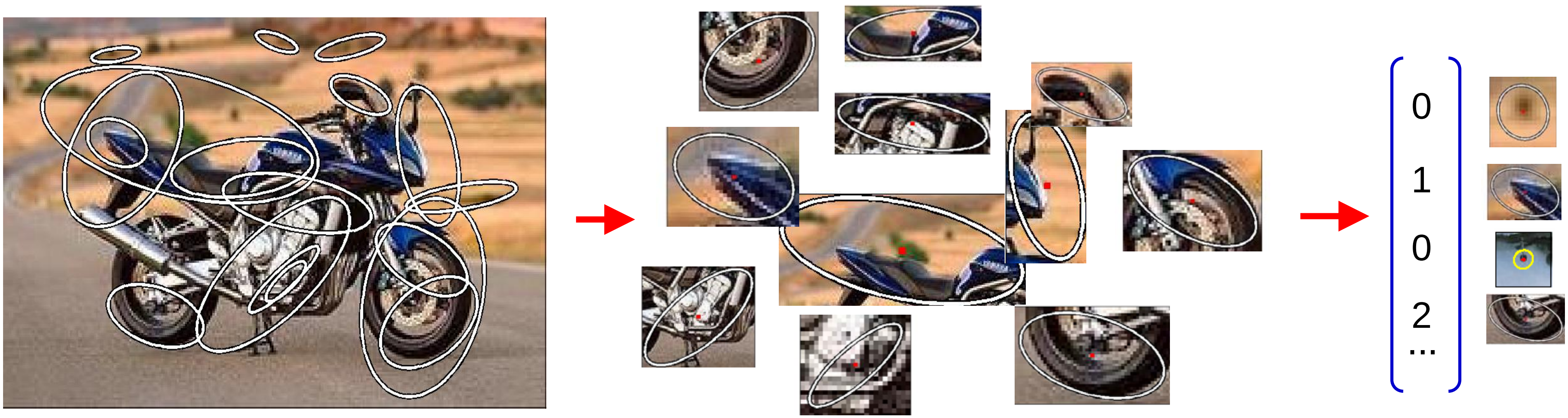
Example Image collection: four object classes + background

Faces	435
Motorbikes	800
Airplanes	800
Cars (rear)	1155
Background	900
Total:	**4090**

The “Caltech 5”

Represent an image as a histogram of visual words

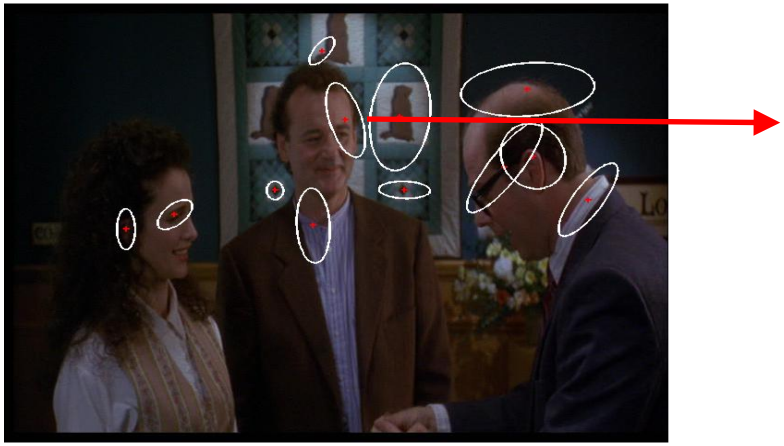
iconic image fragments



Bag of words model

- Detect affine covariant regions
- Represent each region by a SIFT descriptor
- Build visual vocabulary by k-means clustering (K~1,000)
- Assign each region to the nearest cluster centre

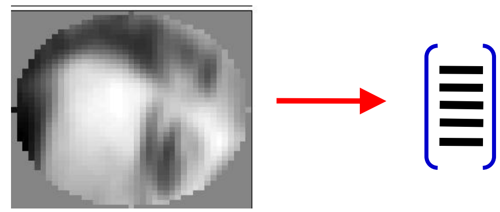
Visual vocabulary for affine covariant patches



Detect patches

[Mikolajczyk and Schmid ’02]

[Matas et al. ’02]

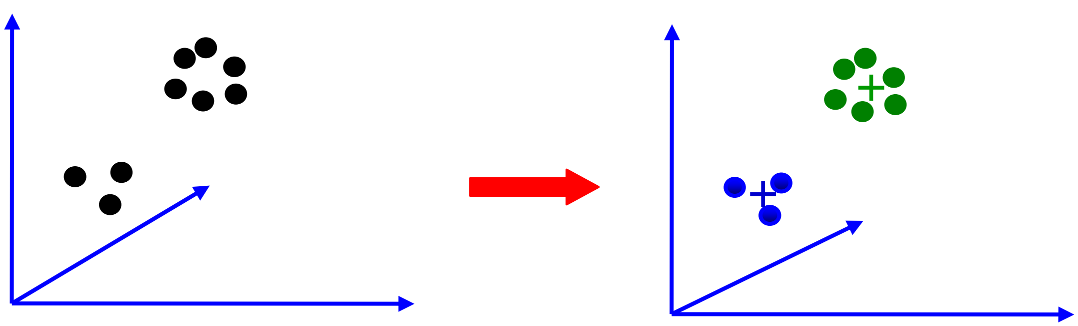


Normalize patch

Compute SIFT descriptor

[Lowe’99]

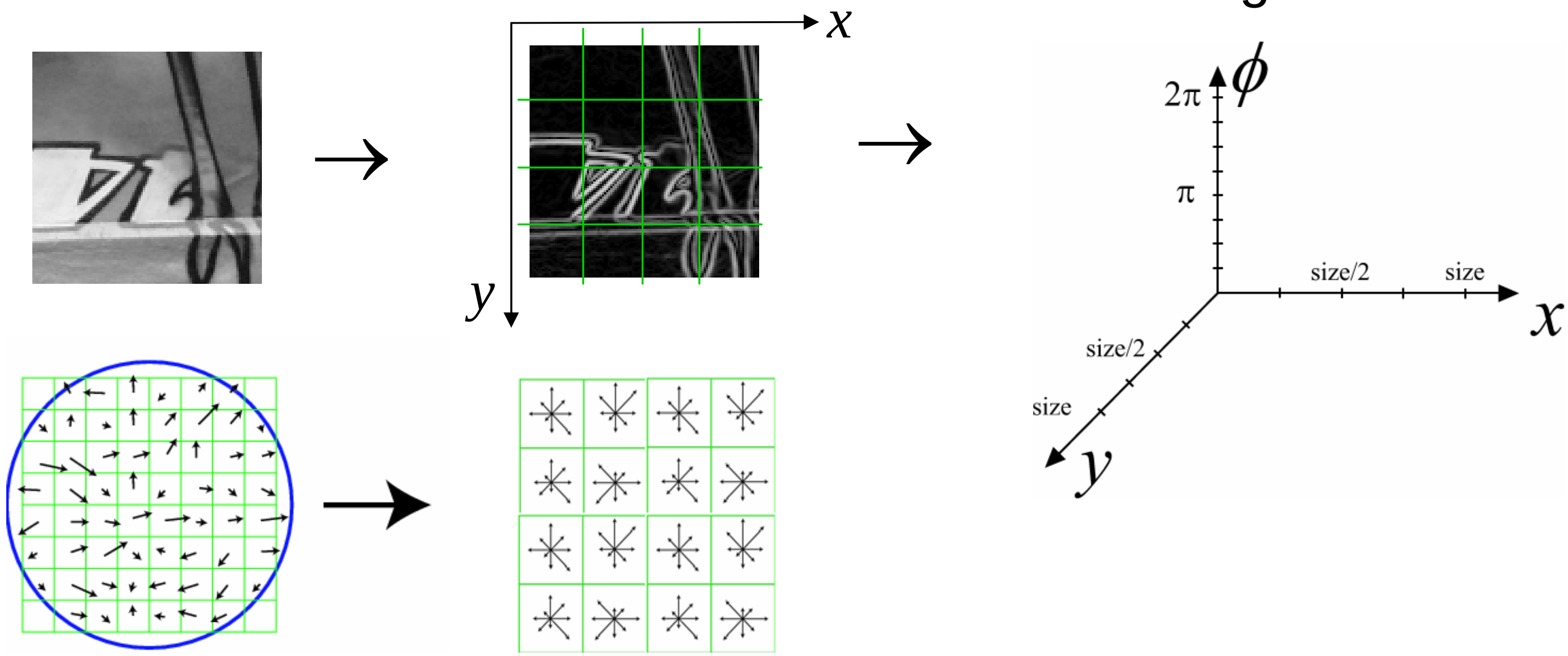
Vector quantize descriptors from a set of training images using k-means



Descriptors – SIFT [Lowe'99]

distribution of the gradient over an image patch

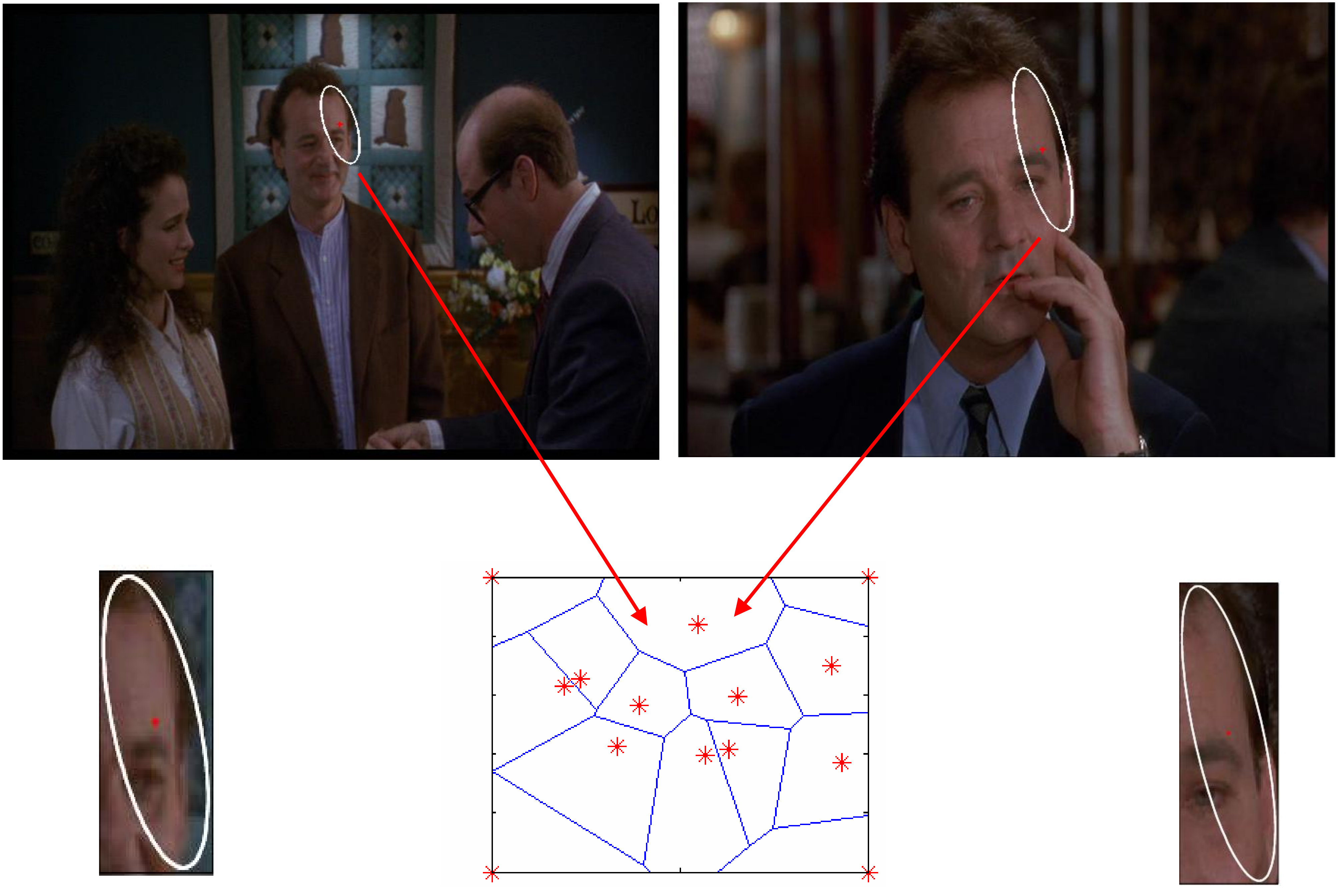
image patch gradient 3D histogram



4x4 location grid and 8 orientations (128 dimensions)

very good performance in image matching [Mikolaczyk and Schmid'03]

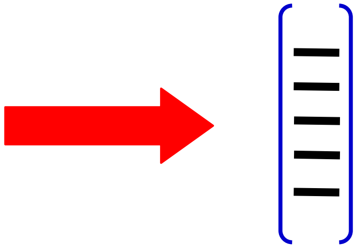
Vector quantize the descriptor space (SIFT)



The same visual word

Each image: assign all detections to their visual words

- gives bag of visual word representation
- normalized histogram of word frequencies
- also called ‘bag of key points’



Visual words from affine covariant patches

Vector quantize SIFT descriptors to a vocabulary of iconic "visual words".

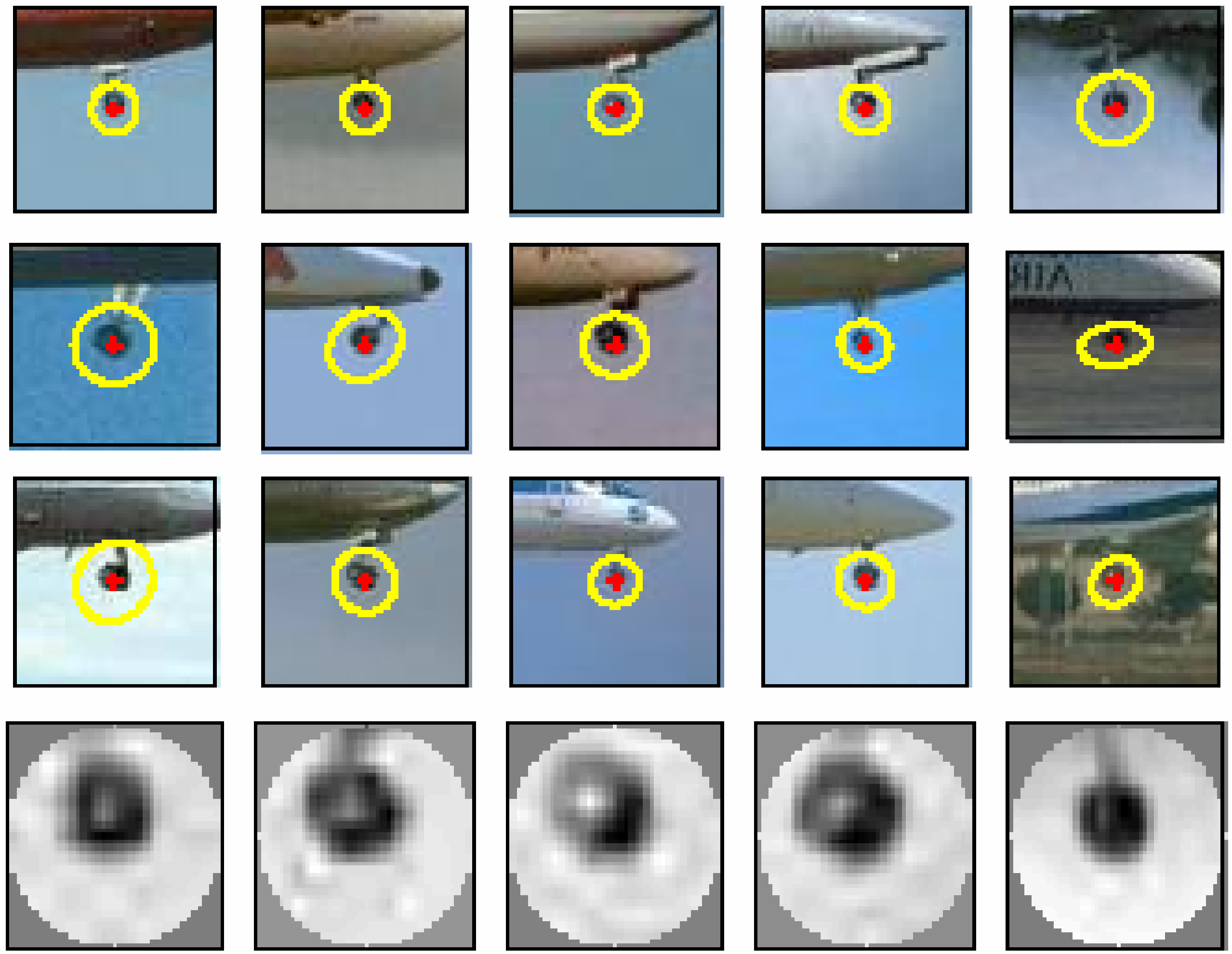
Design of descriptors makes these words invariant to:

- illumination
- affine transformations (viewpoint)

Size (granularity) of vocabulary is an important parameter

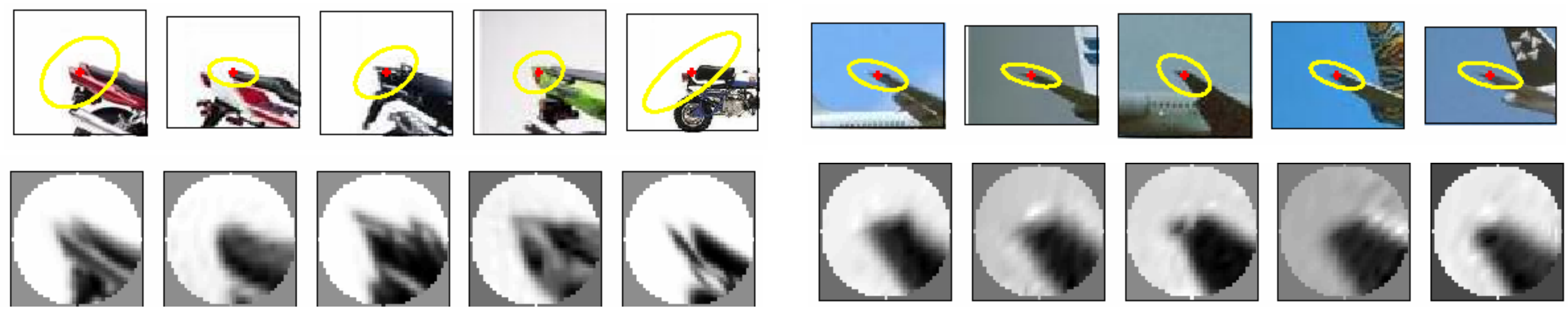
- fine grained – represent model instances
- coarse grained – represent object categories

Examples of visual words

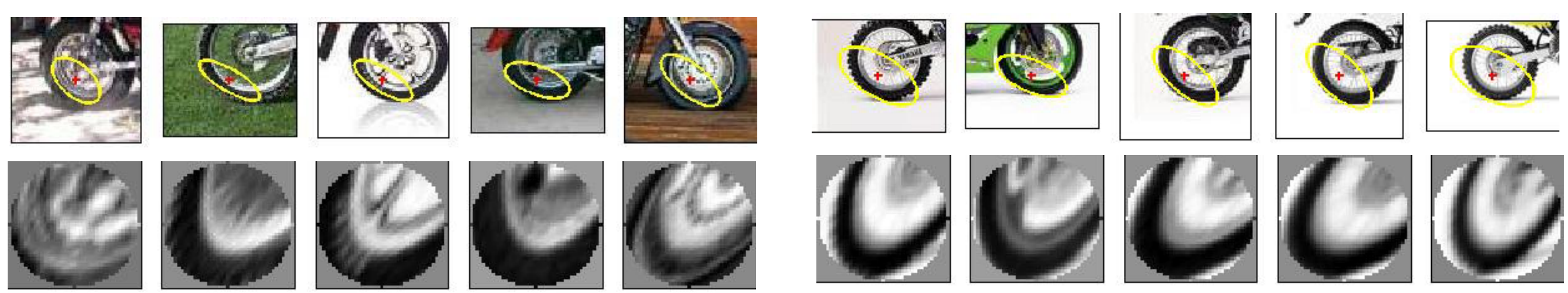


More visual words

Visual synonyms and polysemy



Visual Polysemy: Single visual word occurring on different (but locally similar) parts on different object categories.

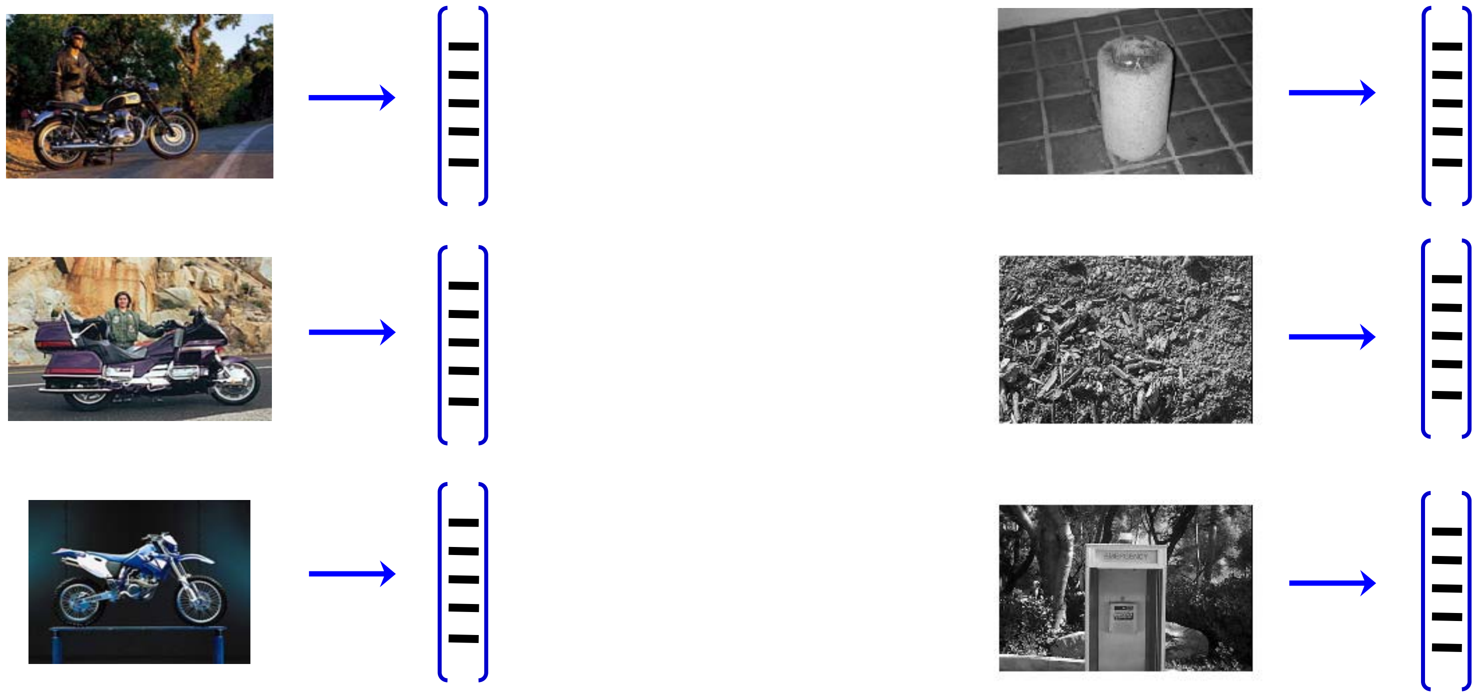


Visual Synonyms: Two different visual words representing a similar part of an object (wheel of a motorbike).

Training data: vectors are histograms, one from each training image

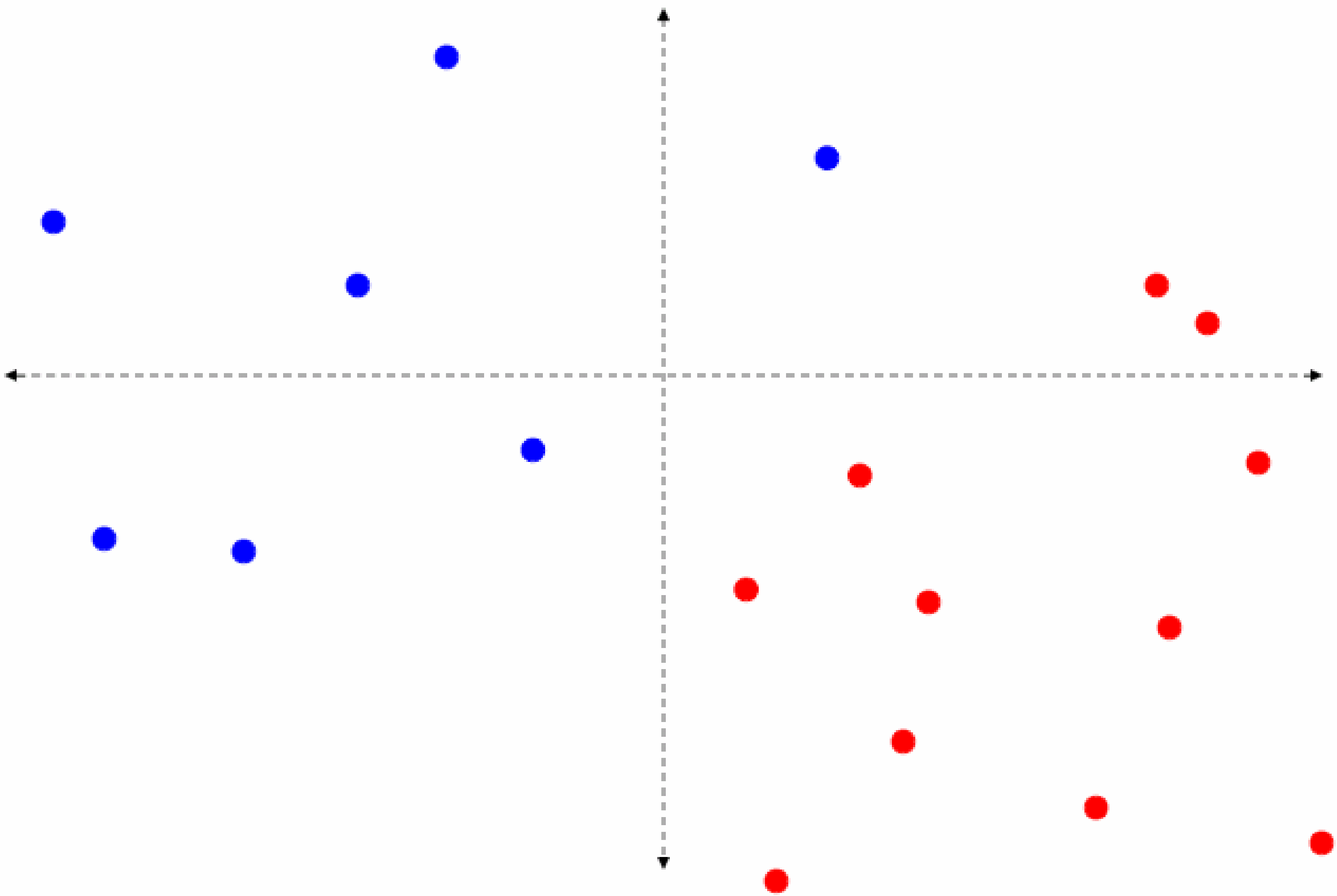
positive

negative

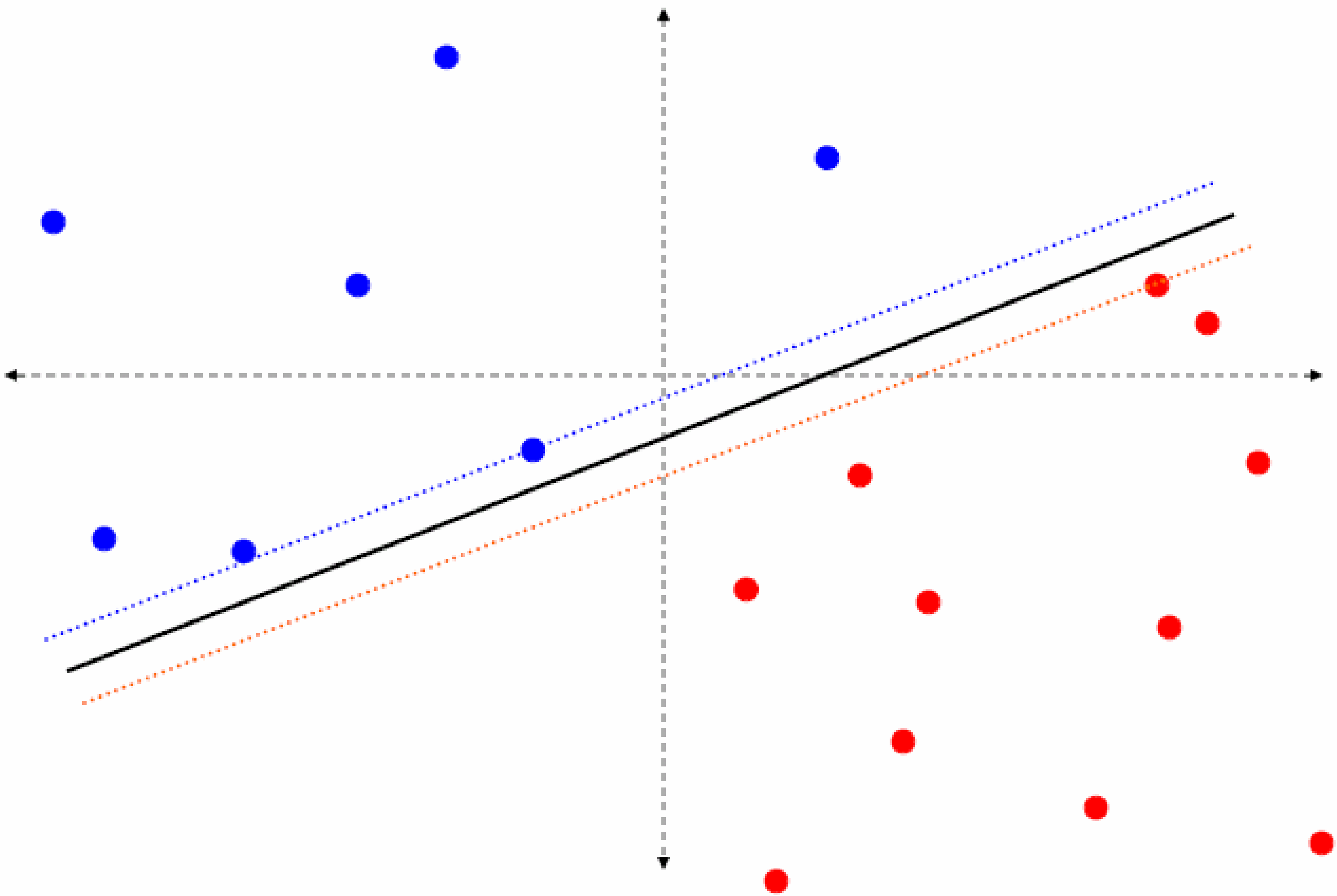


Train classifier,e.g.SVM

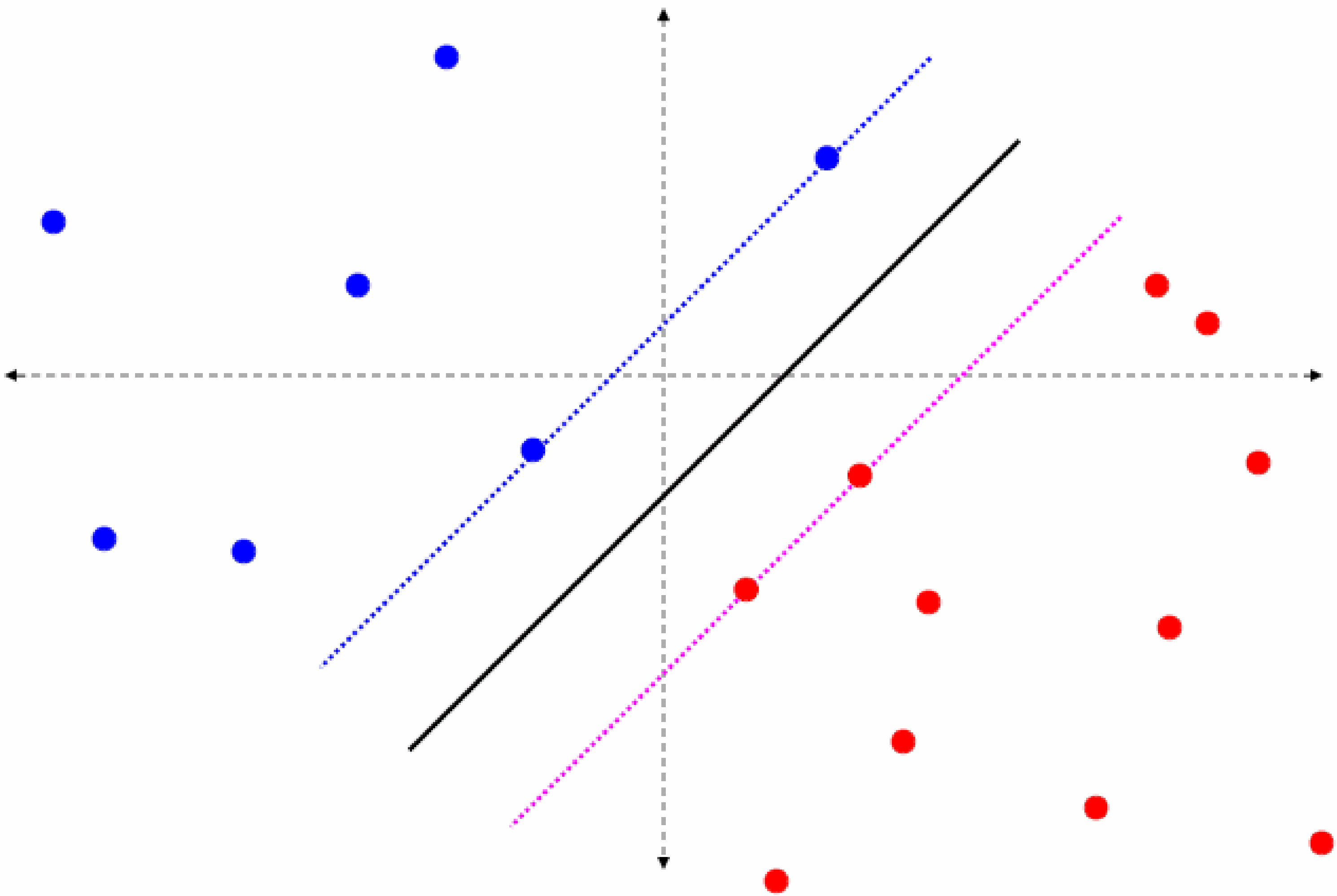
The Binary Classification Problem



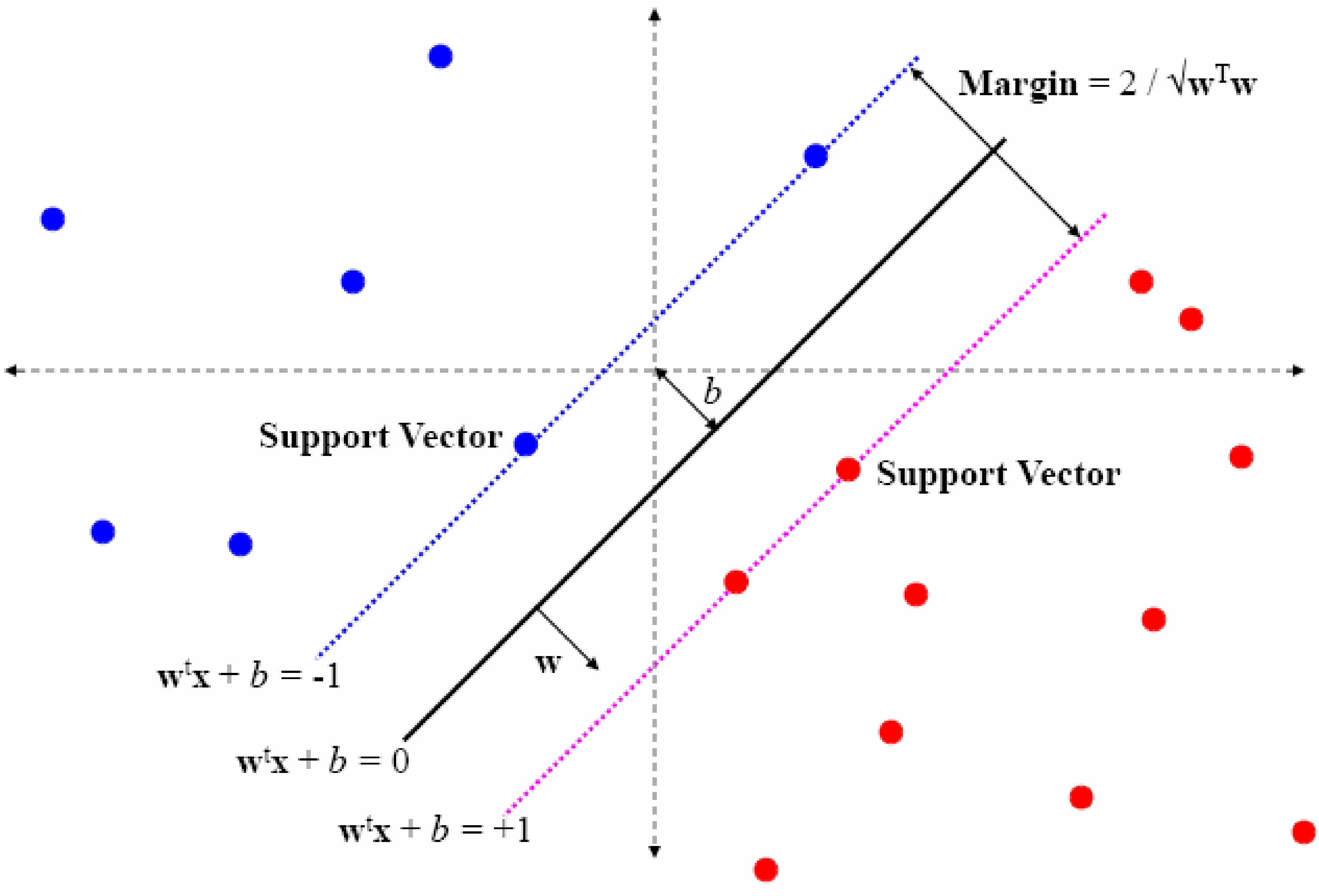
A Separating Hyperplane



Maximal Margin Hyperplane



SVM Terminology



SVM classifier with kernels

N = size of training data

$$f(\mathbf{x}) = \sum_i^N \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b$$

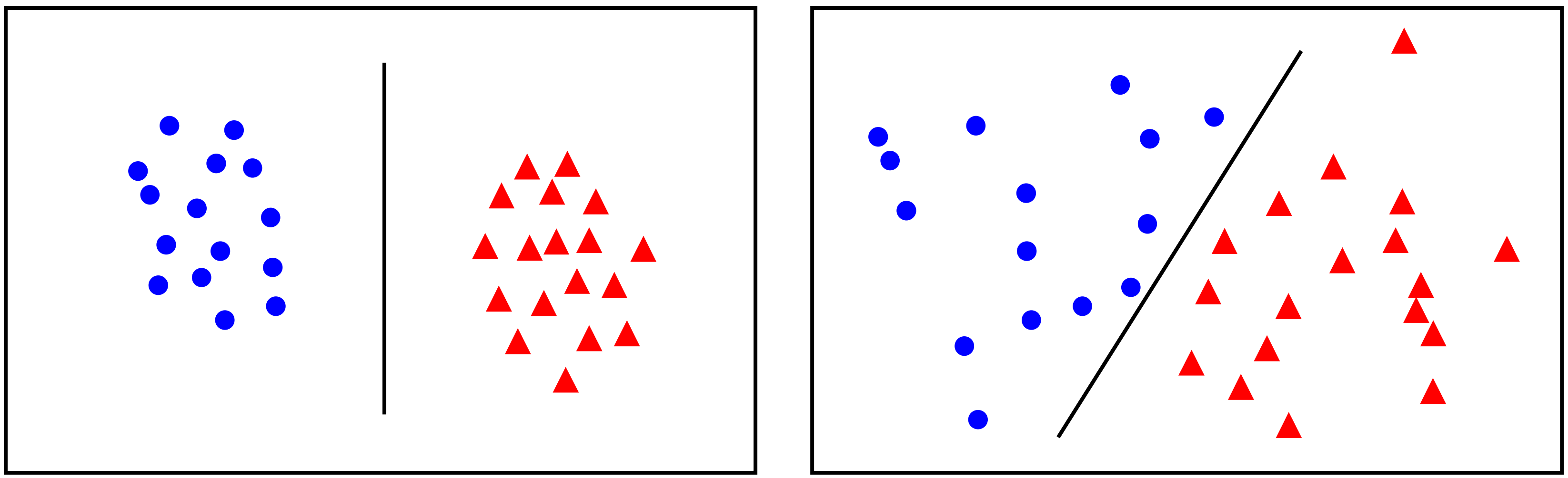
weight (may be zero)

support vector

$$f(\mathbf{x}) \begin{cases} \geq 0 & \text{positive class} \\ < 0 & \text{negative class} \end{cases}$$

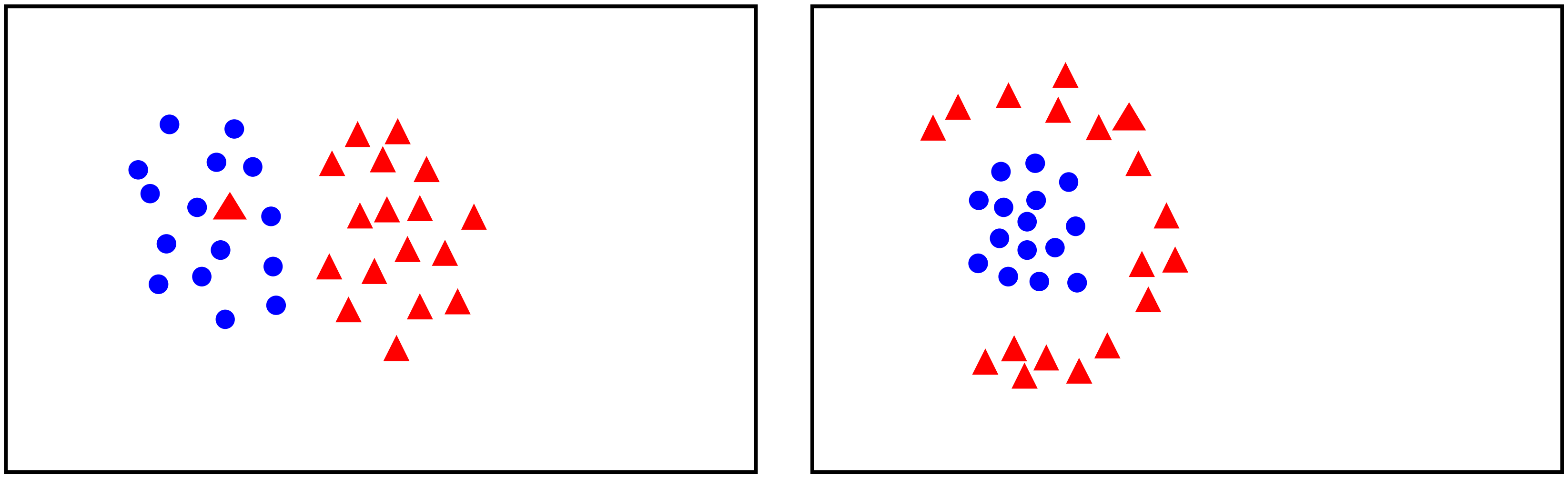
Linear separability

linearly
separable



linear kernel sufficient

not
linearly
separable



use non-linear kernel

Some popular kernels

- Linear: $K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y}$

- Polynomial: $K(\mathbf{x}, \mathbf{y}) = \left(\mathbf{x}^\top \mathbf{y} + c\right)^n$

- Radial basis function: $K(\mathbf{x}, \mathbf{y}) = e^{-\gamma ||\mathbf{x}-\mathbf{y}||^2}$

- Chi-squared: $K(\mathbf{x}, \mathbf{y}) = e^{-\gamma \chi^2(\mathbf{x},\mathbf{y})}$

  where $\chi^2(\mathbf{x}, \mathbf{y}) = \sum_j \frac{(x_j - y_j)^2}{x_j + y_j}$

Advantage of linear kernels – at test time

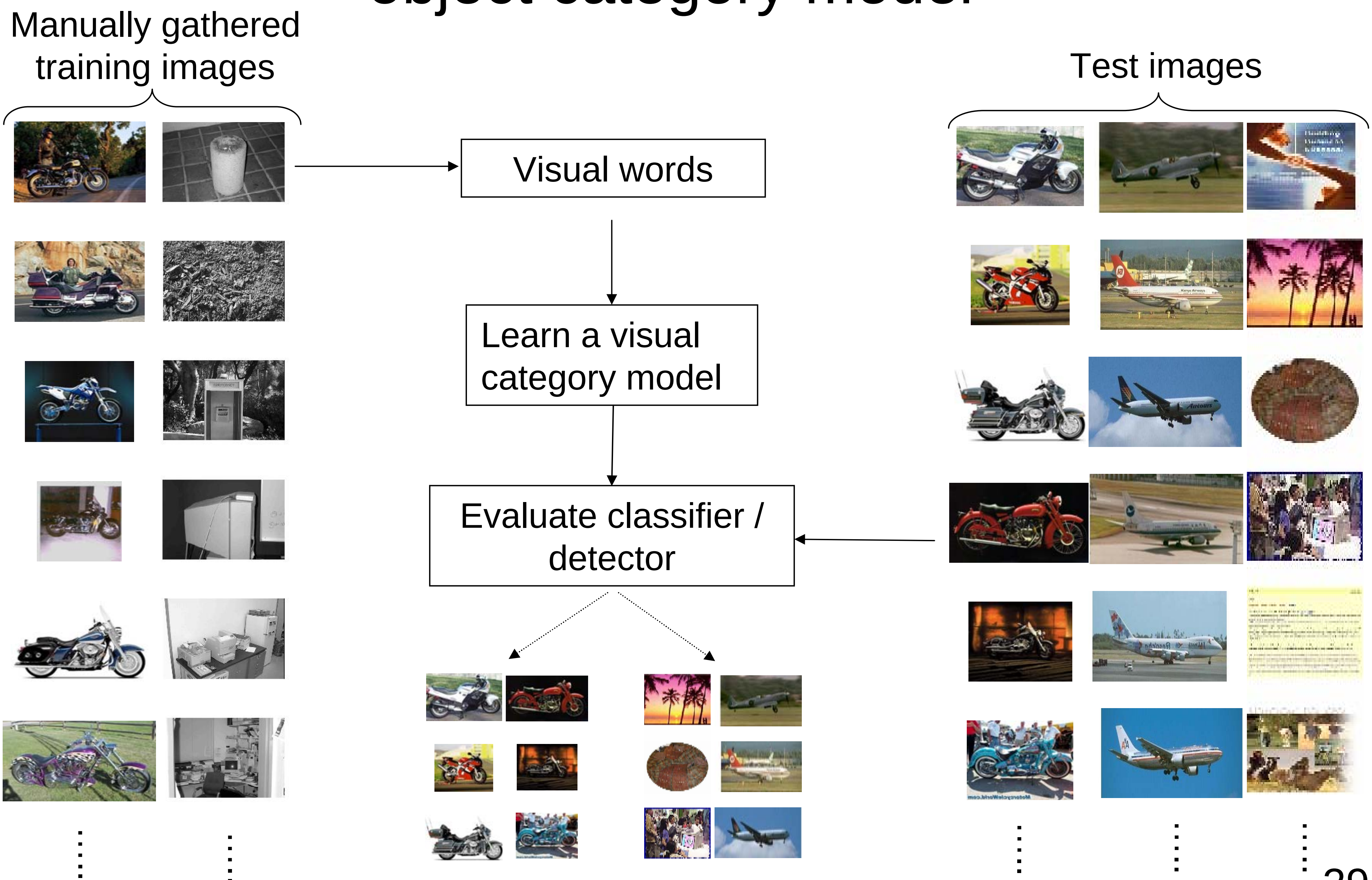
N = size of training data

$$f(\mathbf{x}) = \sum_i^N \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b$$

$$\begin{aligned} f(\mathbf{x}) &= \sum_i^N \alpha_i \mathbf{x}_i{}^\top \mathbf{x} + b \\ &= \mathbf{w}^\top \mathbf{x} + b \end{aligned}$$

Independent of size of training data

Current Paradigm for learning an object category model



Example: weak supervision

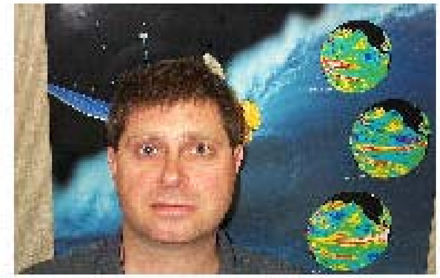
Training

- 50% images
- No identifcation of object within image

Motorbikes

Airplanes

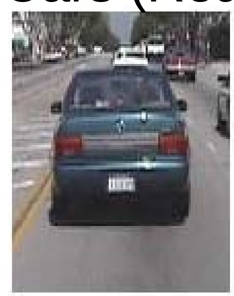
Frontal Faces



Testing

- 50% images
- Simple object present/absent test

Cars (Rear)



Background

Learning

- SVM classifier
- Gaussian kernel using χ^2 as distance between histograms

Result

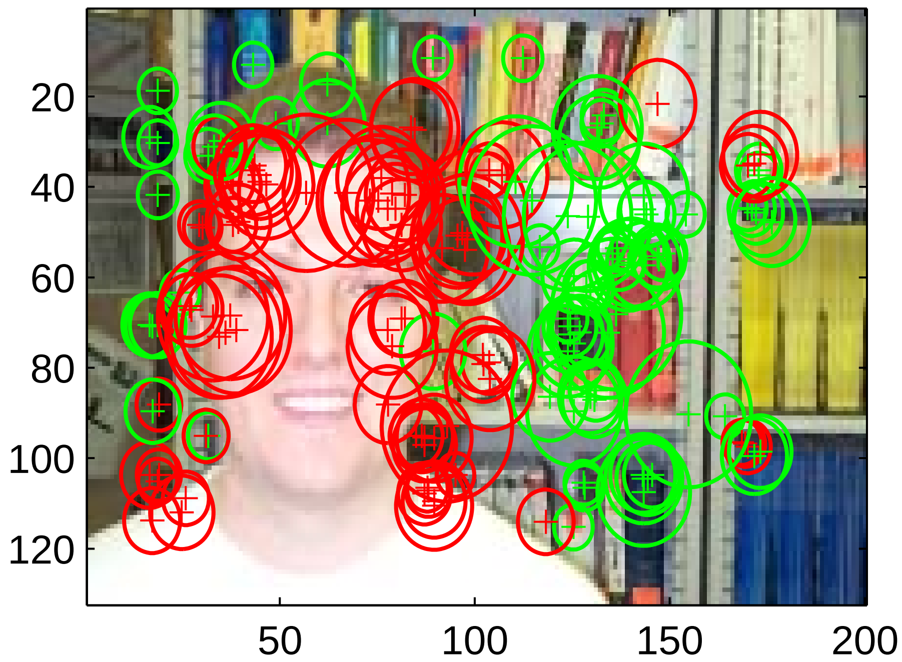
- Between 98.3 – 100% correct, depending on class

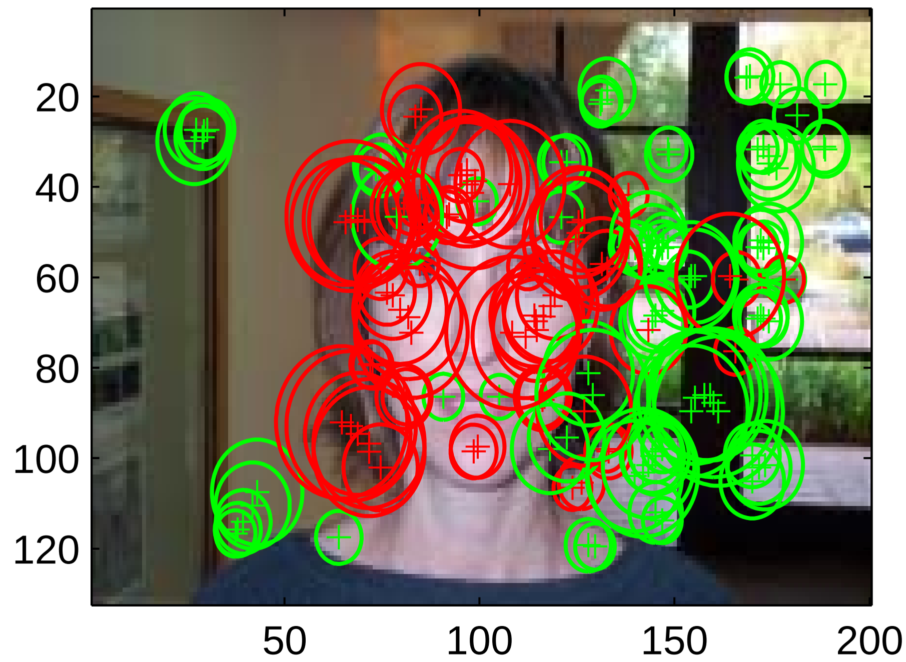
Csurka et al 2004

Zhang et al 2005

Localization according to visual word probability

sparse segmentation



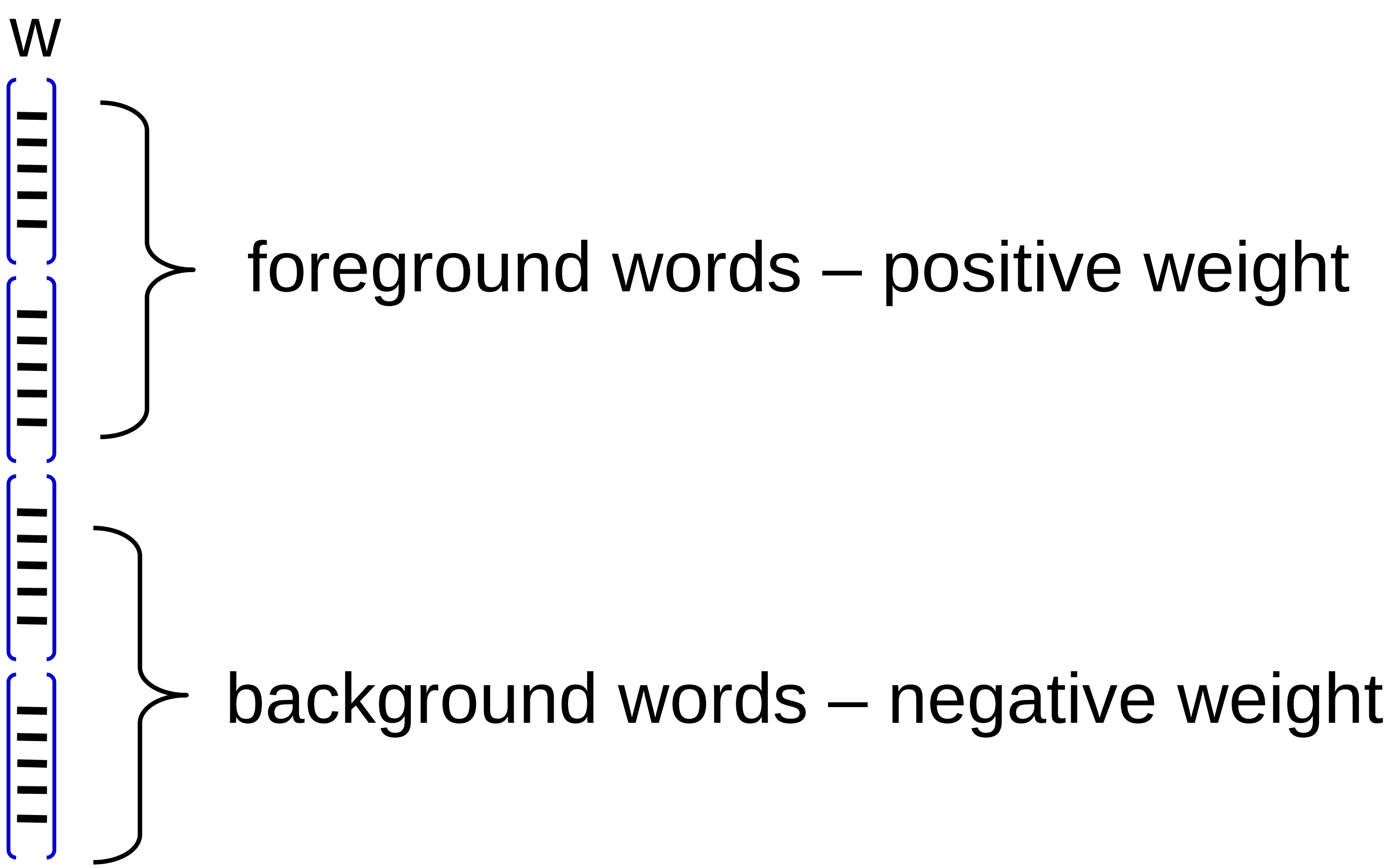


foreground word more probable

background word more probable

Why does SVM learning work?

- Learns foreground and background visual words



Bag of visual words summary

- Advantages:
  - largely unaffected by position and orientation of object in image
  - fixed length vector irrespective of number of detections
  - Very successful in classifying images according to the objects they contain
  - Still requires further testing for large changes in scale and viewpoint

- Disadvantages:
  - No explicit use of configuration of visual word positions
  - Poor at localizing objects within an image