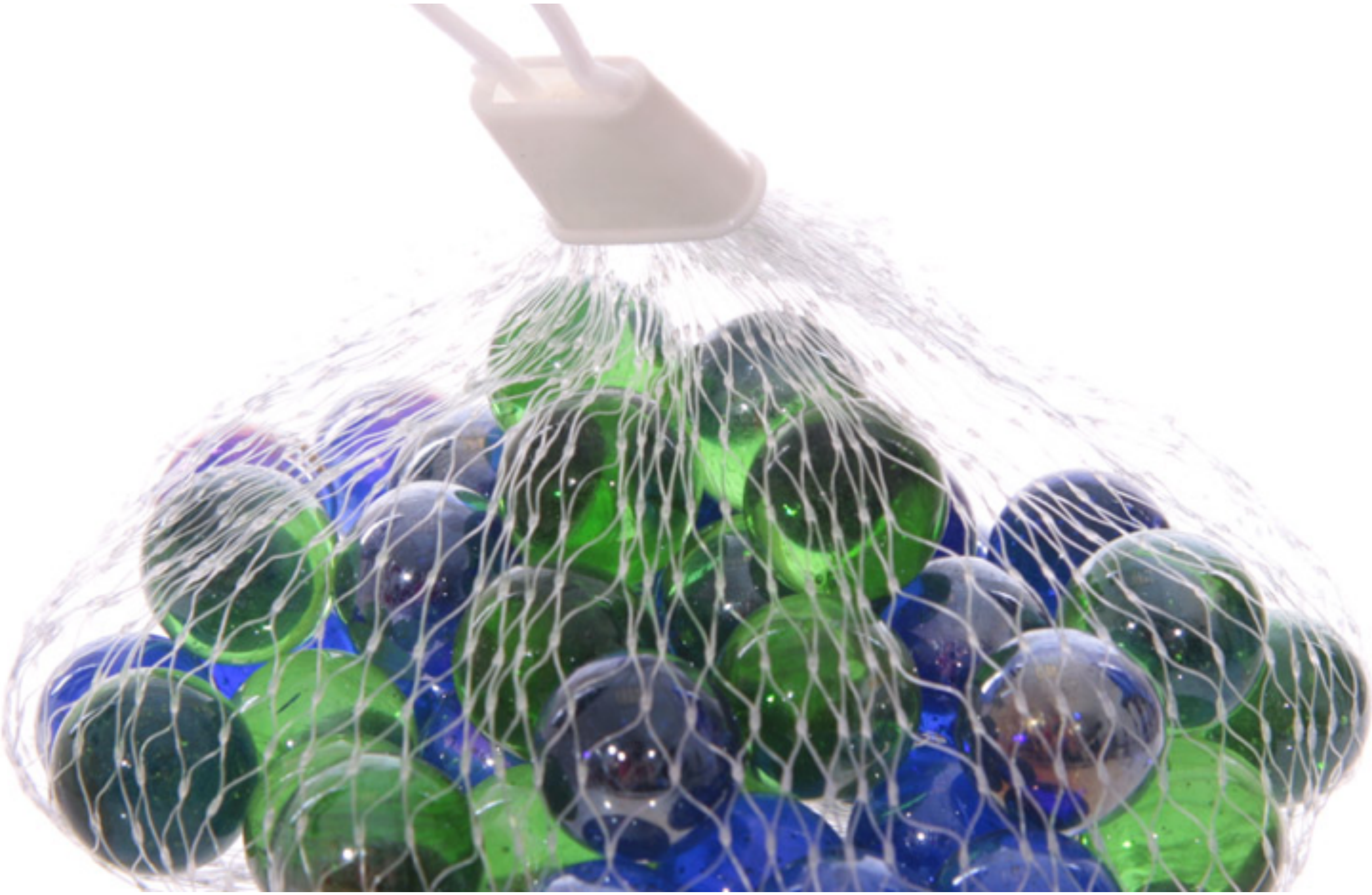
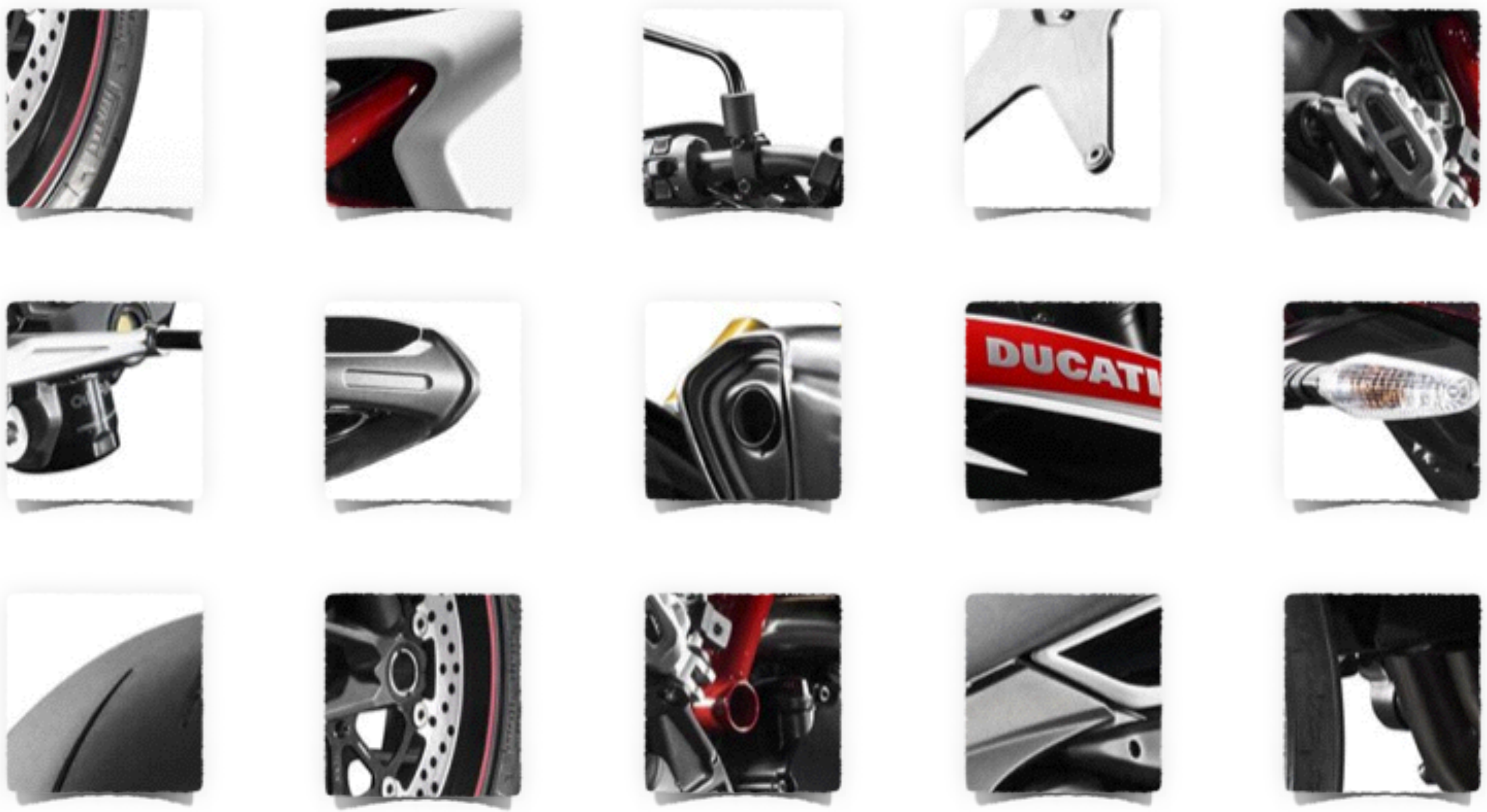


Bag-of-Visual-Words

16-385 Computer Vision

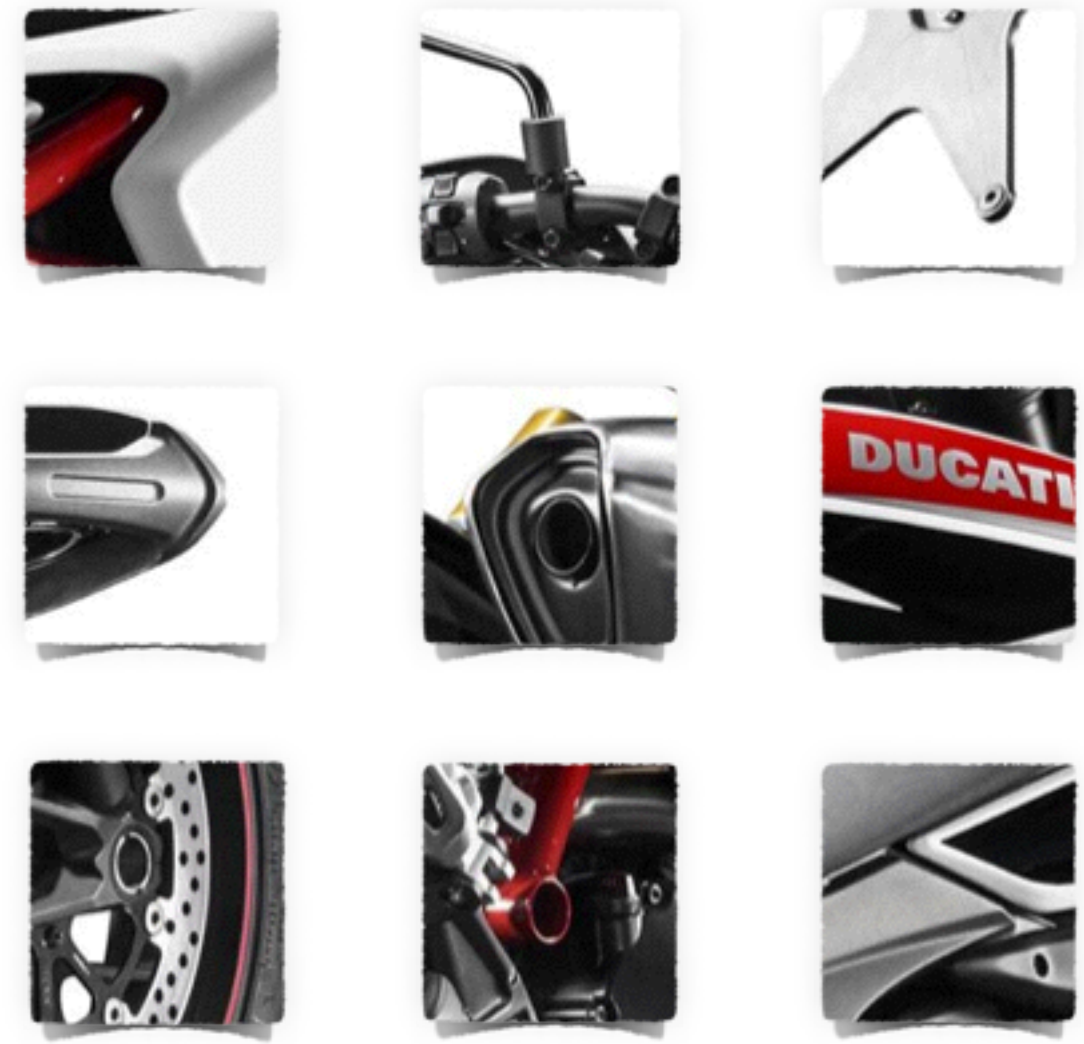
Carnegie Mellon University (Kris Kitani)

What object do these parts belong to?



Some local feature are
very informative

An object as

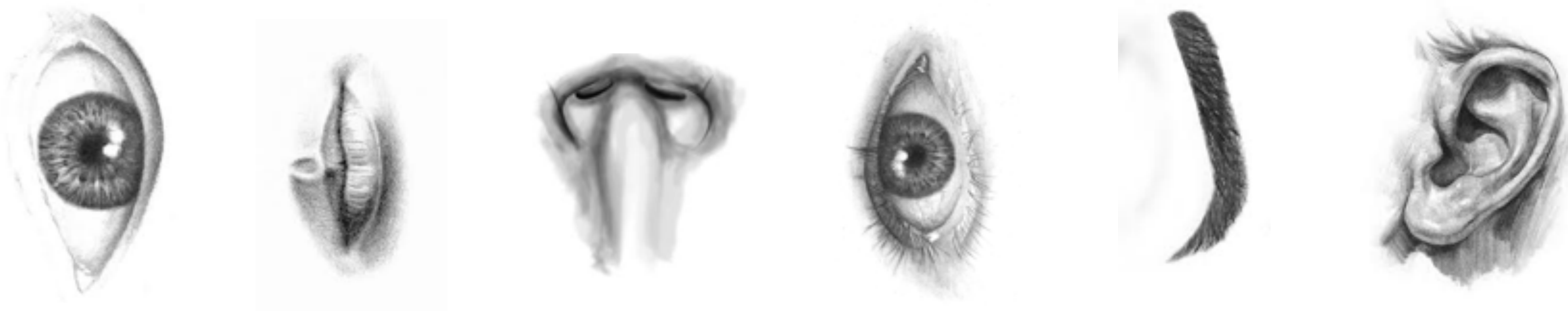


a collection of local features

(bag-of-features)

- deals well with occlusion
- scale invariant
- rotation invariant

(not so) crazy assumption



spatial information of local features
can be ignored for object recognition (i.e., verification)

CalTech6 dataset

class	bag of features Zhang et al. (2005)	bag of features Willamowski et al. (2004)	Parts-and-shape model Fergus et al. (2003)
airplanes	**98.8**	97.1	90.2
cars (rear)	98.3	**98.6**	90.3
cars (side)	**95.0**	87.3	88.5
faces	**100**	99.3	96.4
motorbikes	**98.5**	98.0	92.5
spotted cats	**97.0**	—	90.0

Works pretty well for image-level classification

Csurka et al. (2004), Willamowski et al. (2005), Grauman & Darrell (2005), Sivic et al. (2003, 2005)

Bag-of-features

represent a data item (document, texture, image)
as a histogram over features

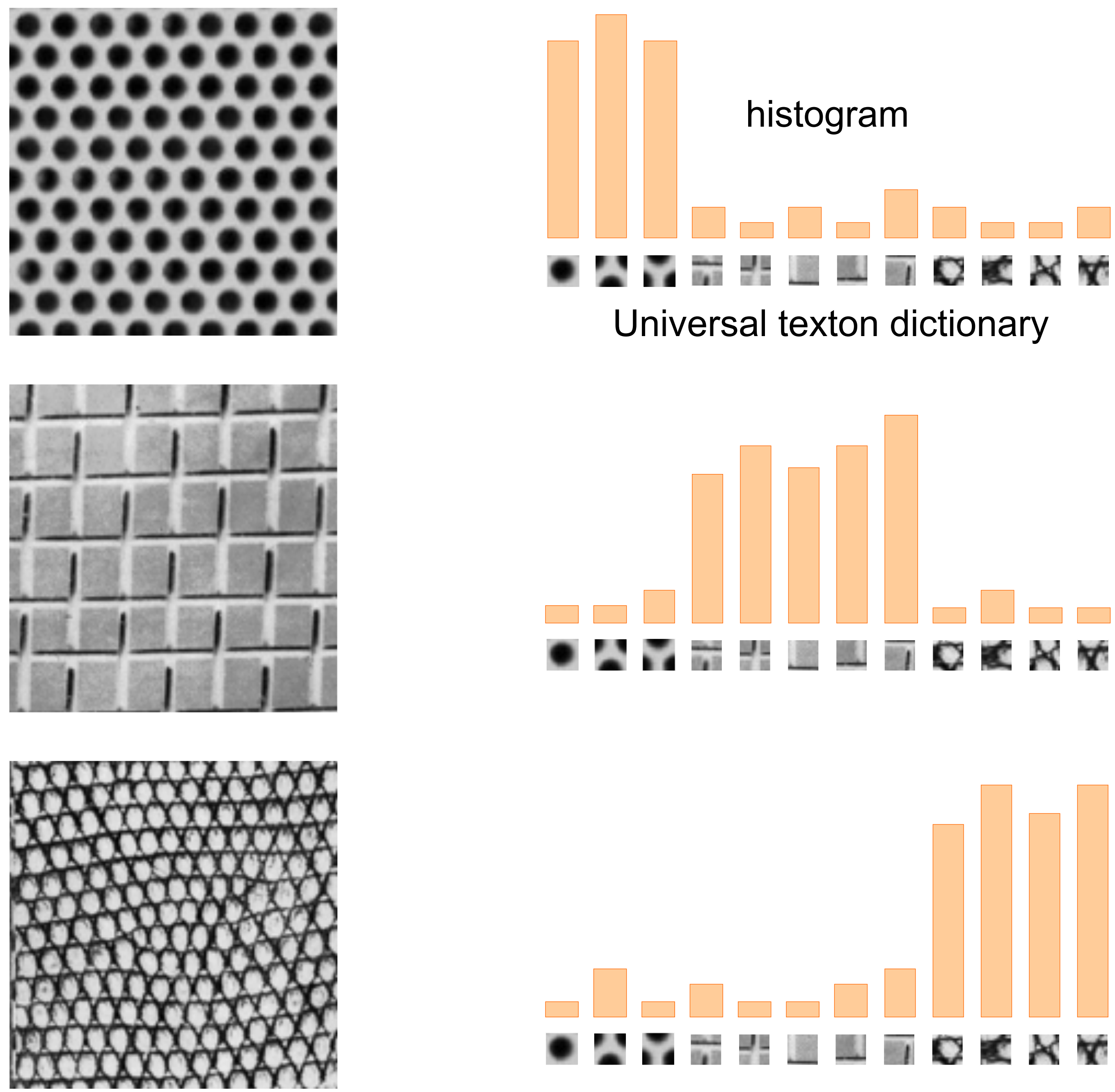
Bag-of-features

represent a data item (document, texture, image)
as a histogram over features

an old idea

(e.g., texture recognition and information retrieval)

Texture recognition

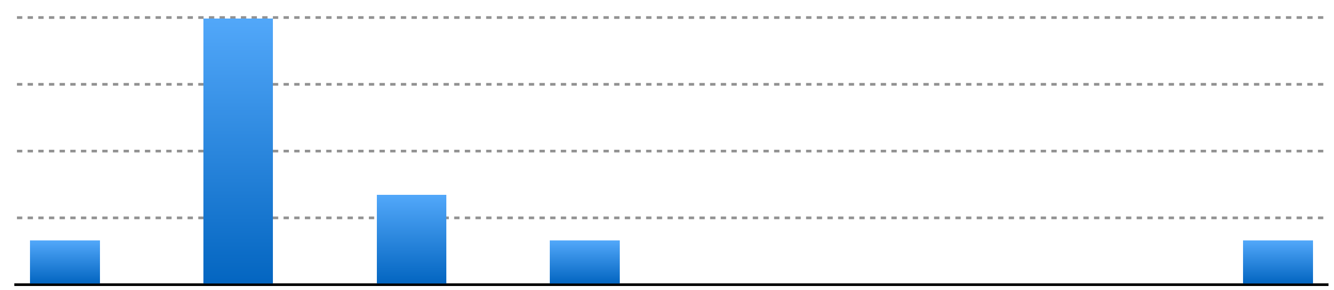


Julesz, 1981

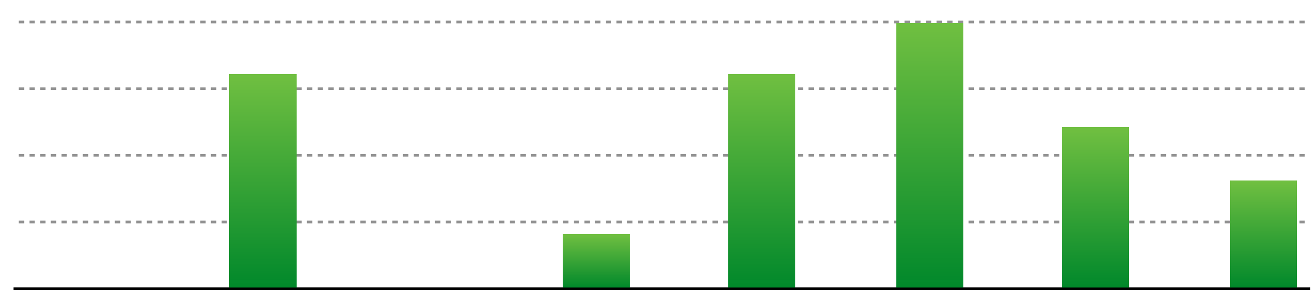
Mori, Belongie and Malik, 2001

Vector Space Model

G. Salton. ‘Mathematics and Information Retrieval’ Journal of Documentation,1979



1	6	2	1	0	0	0	1
Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor



0	4	0	1	4	5	3	2
Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor

http://www.fodey.com/generators/newspaper/snippet.asp

A document (datapoint) is a vector of counts over each word (feature)

$$\boldsymbol{v}_d = [n(w_{1,d}) \quad n(w_{2,d}) \quad \cdots \quad n(w_{T,d})]$$

just a histogram over words

$n(\cdot)$ counts the number of occurrences

What is the similarity between two documents?

The Newspa

DARPA Selects Carnegie Me

Tartan Tim

Bio-Inspired Robotic Device

A document (datapoint) is a vector of counts over each word (feature)

$$\boldsymbol{v}_d = [n(w_{1,d}) \quad n(w_{2,d}) \quad \cdots \quad n(w_{T,d})]$$

just a histogram over words

$n(\cdot)$ counts the number of occurrences

What is the similarity between two documents?

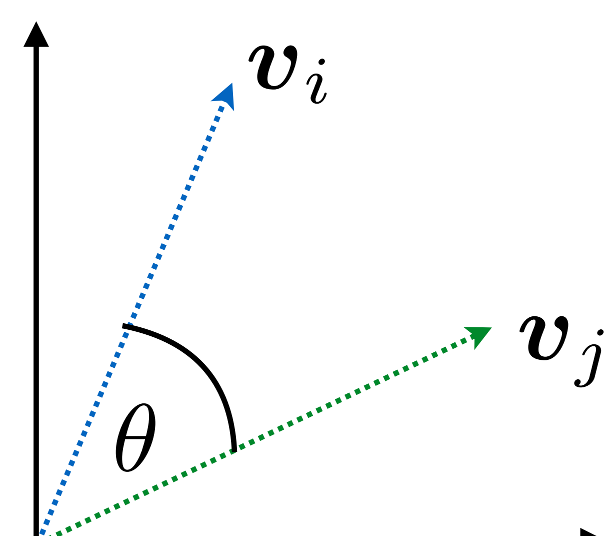

The Newspa
DARPA Selects Carnegie Me


Tartan Tim
Bio-Inspired Robotic Device

Use any distance you want but the cosine distance is fast.

$$d(\boldsymbol{v}_i, \boldsymbol{v}_j) = \cos\theta$$

$$= \frac{\boldsymbol{v}_i \cdot \boldsymbol{v}_j}{\|\boldsymbol{v}_i\| \|\boldsymbol{v}_j\|}$$



but not all words are created equal

TF-IDF

Term frequency Inverse Document Frequency

$$\boldsymbol{v}_d = \begin{bmatrix} n(w_{1,d}) & n(w_{2,d}) & \cdots & n(w_{T,d}) \end{bmatrix}$$

but not all words are created equal

$$\boldsymbol{v}_d = \begin{bmatrix} n(w_{1,d})\alpha_1 & n(w_{2,d})\alpha_2 & \cdots & n(w_{T,d})\alpha_T \end{bmatrix}$$

$$n(w_{i,d})\alpha_i = n(w_{i,d}) \log \left\{ \frac{D}{\sum_{d'} \mathbf{1}[w_i \in d']} \right\}$$

term frequency

inverse document frequency

Example of tf–idf [edit]

Suppose we have term frequency tables for a collection consisting of only two documents, as listed on the right, then calculation of tf–idf for the term "this" in document 1 is performed as follows.

Document 1

Term	Term Count
this	1
is	1
a	2
sample	1

Document 2

Term	Term Count
this	1
is	1
another	2
example	3

Tf, in its basic form, is just the frequency that we look up in appropriate table. In this case, it's one.

Idf is a bit more involved:

$$\mathrm{idf}(\text{this}, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

The numerator of the fraction is the number of documents, which is two. The number of documents in which "this" appears is also two, giving

$$\mathrm{idf}(\text{this}, D) = \log \frac{2}{2} = 0$$

So tf–idf is zero for this term, and with the basic definition this is true of any term that occurs in all documents.

A slightly more interesting example arises from the word "example", which occurs three times but in only one document. For this document, tf–idf of "example" is:

$$\mathrm{tf}(\text{example}, d_2) = 3$$
$$\mathrm{idf}(\text{example}, D) = \log \frac{2}{1} \approx 0.6931$$
$$\mathrm{tfidf}(\text{example}, d_2) = \mathrm{tf}(\text{example}, d_2) \times \mathrm{idf}(\text{example}, D) = 3 \log 2 \approx 2.0794$$

(using the natural logarithm).

Standard BOW pipeline

(for image classification)

1. Extract features

2. Learn “visual vocabulary”

3. Quantize features using visual vocabulary

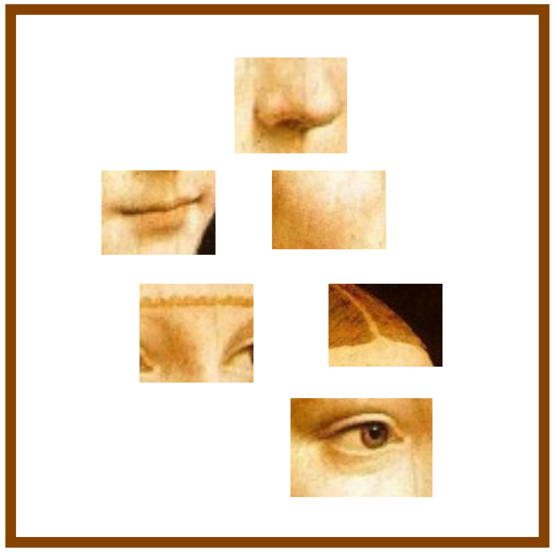
4. Represent images by frequencies of “visual words”

1. **Extract features**

2. Learn “visual vocabulary”

3. Quantize features using visual vocabulary

4. Represent images by frequencies of “visual words”



1. Extract features

2. **Learn "visual vocabulary"**

3. Quantize features using visual vocabulary

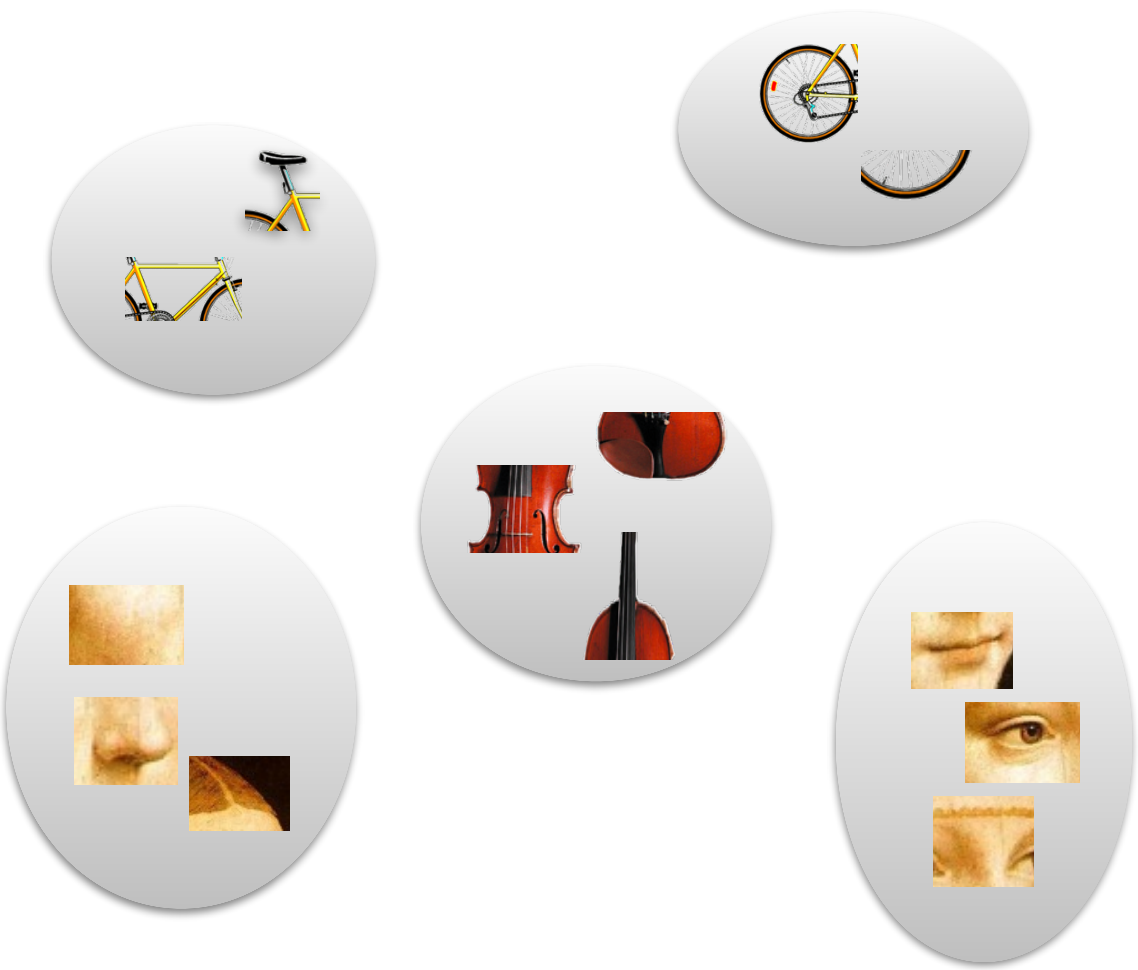
4. Represent images by frequencies of "visual words"

1. Extract features

2. Learn “visual vocabulary”

3. **Quantize features using visual vocabulary**

4. Represent images by frequencies of “visual words”

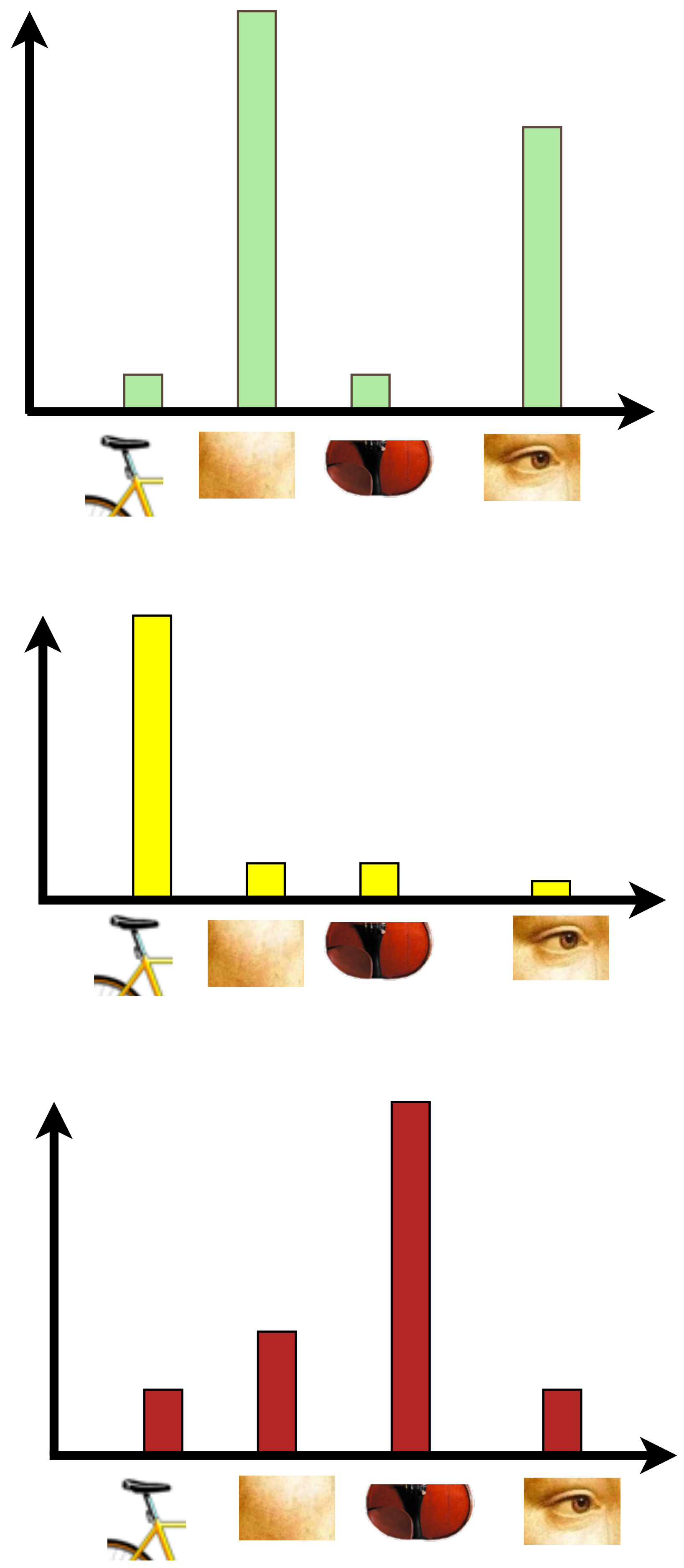


1. Extract features

2. Learn “visual vocabulary”

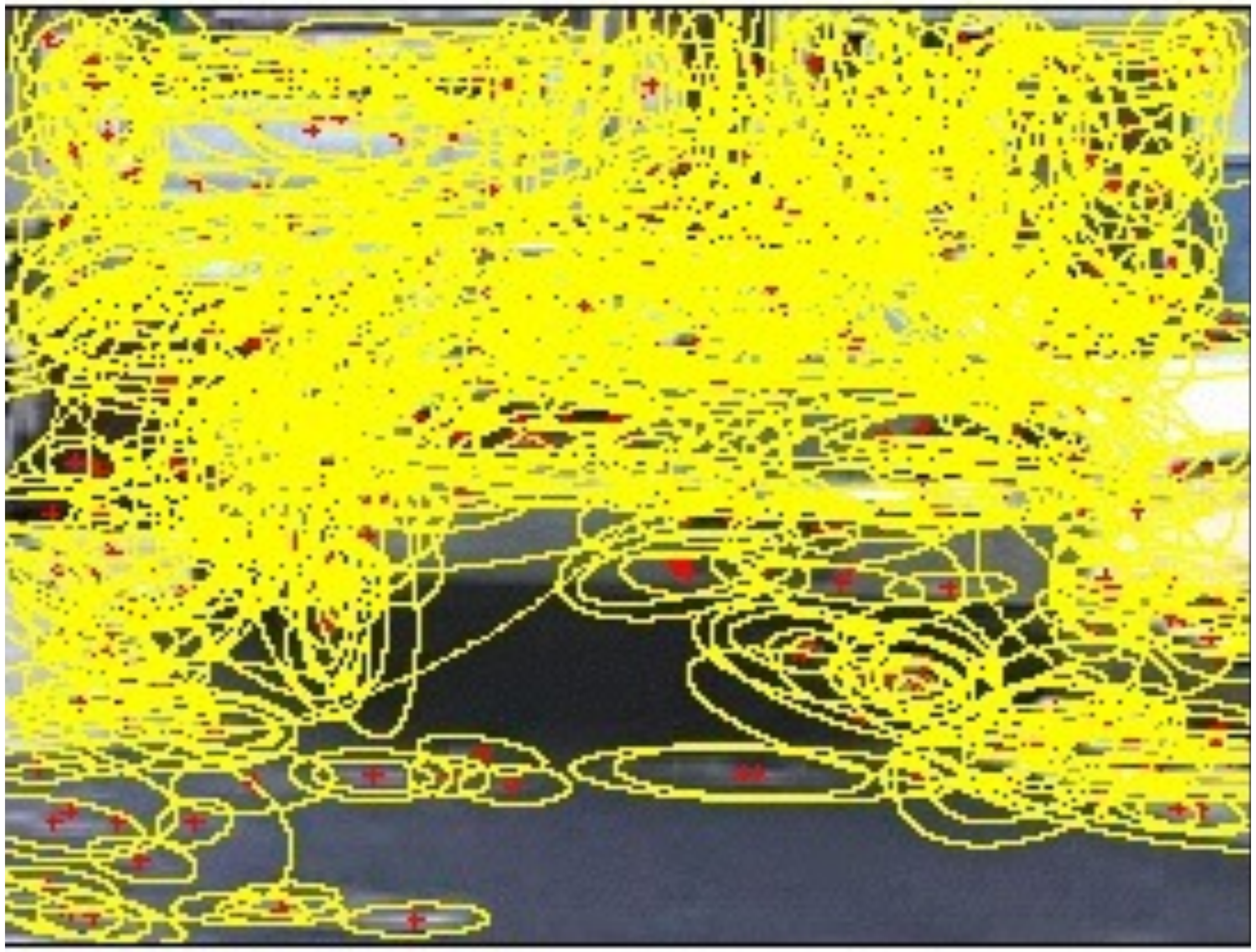
3. Quantize features using visual vocabulary

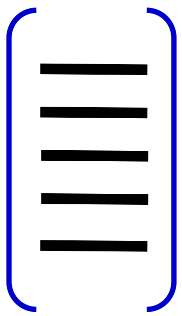
4. **Represent images by frequencies of “visual words”**



Feature Extraction

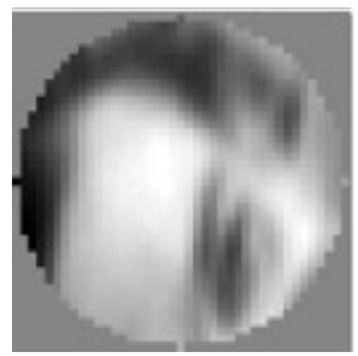
- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005
- Interest point detector
  - Csurka et al. 2004
  - Fei-Fei & Perona, 2005
  - Sivic et al. 2005
- Other methods
  - Random sampling (Vidal-Naquet & Ullman, 2002)
  - Segmentation-based patches (Barnard et al. 2003)



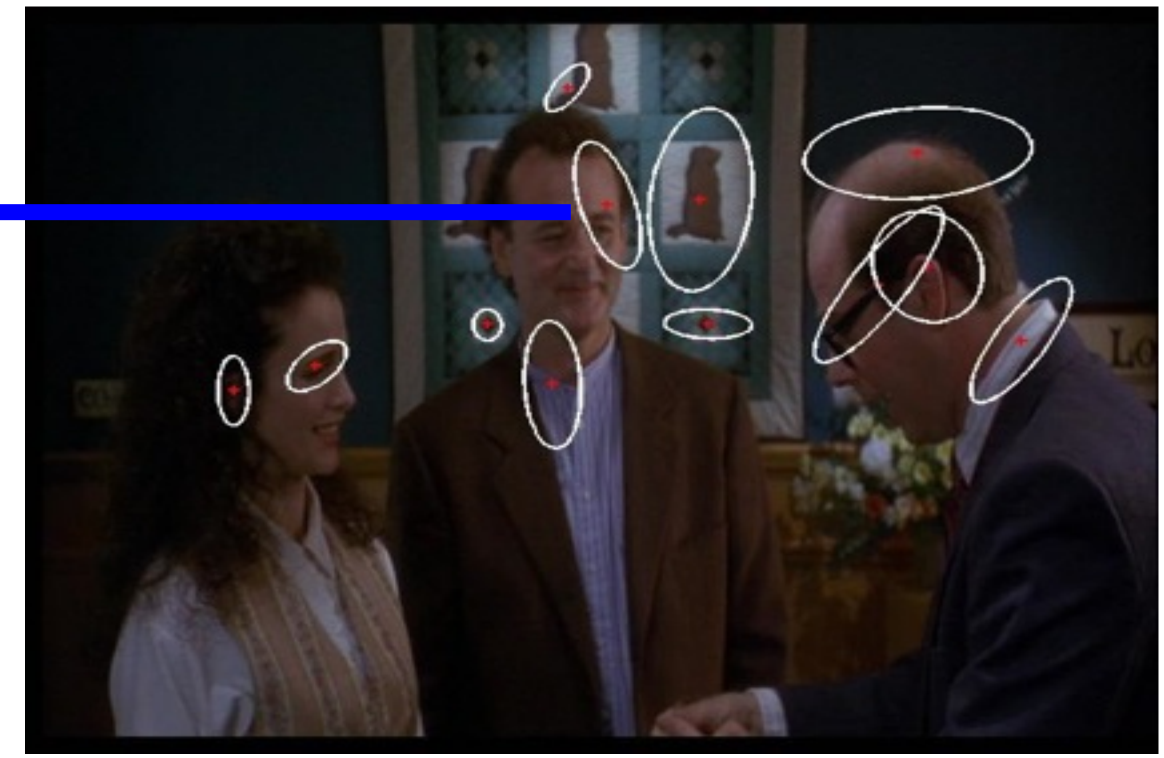


Compute SIFT descriptor

[Lowe'99]



Normalize patch

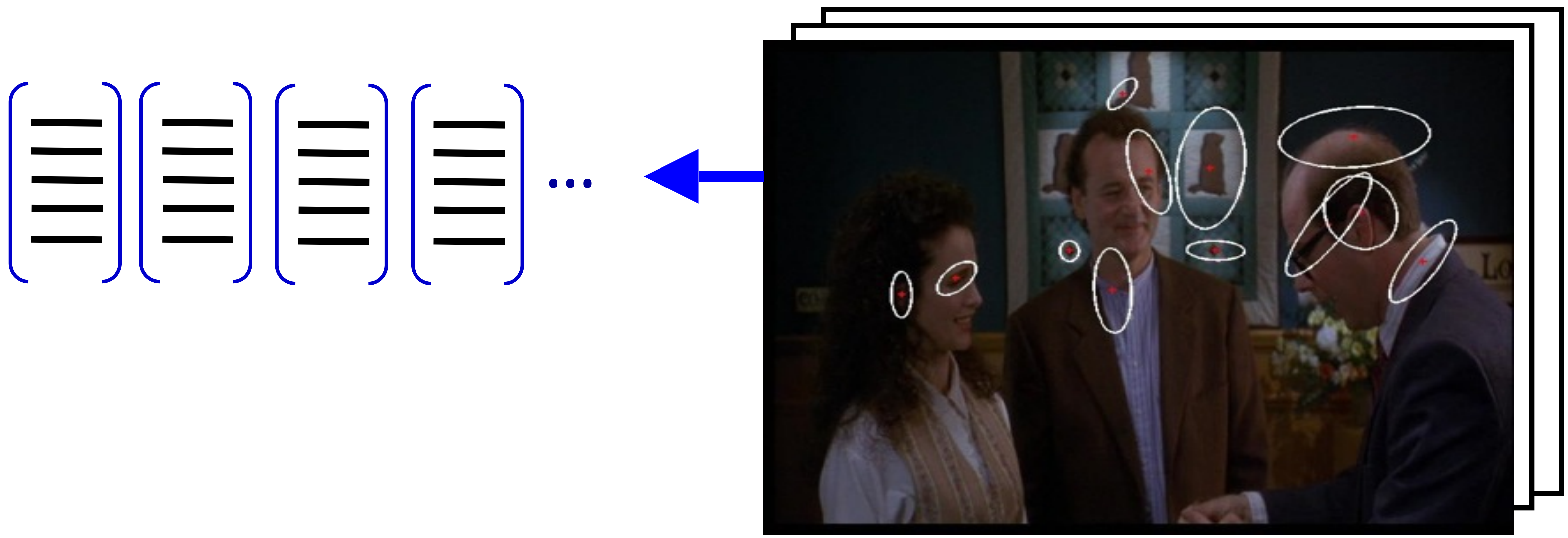


Detect patches

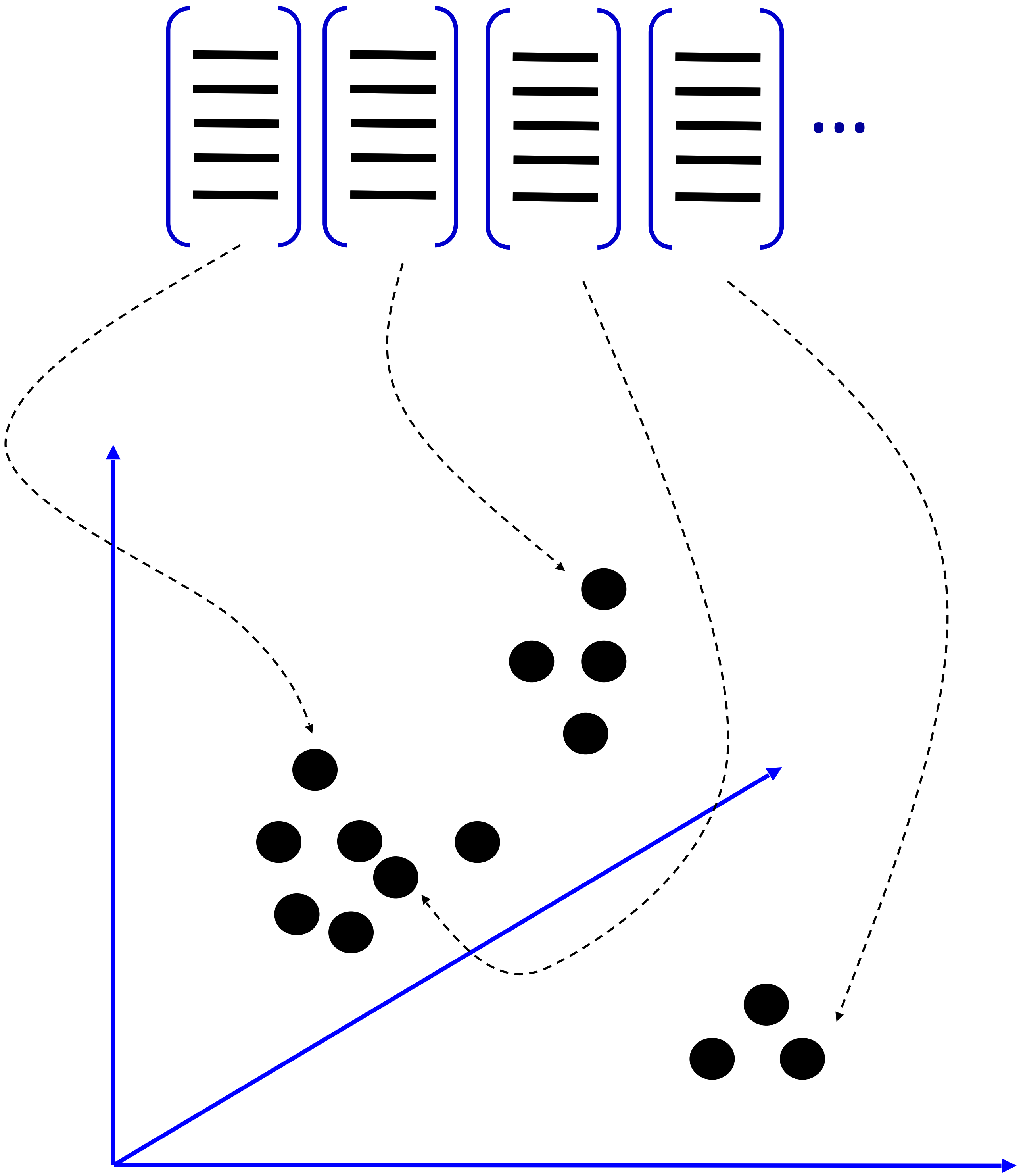
[Mikojaczyk and Schmid '02]

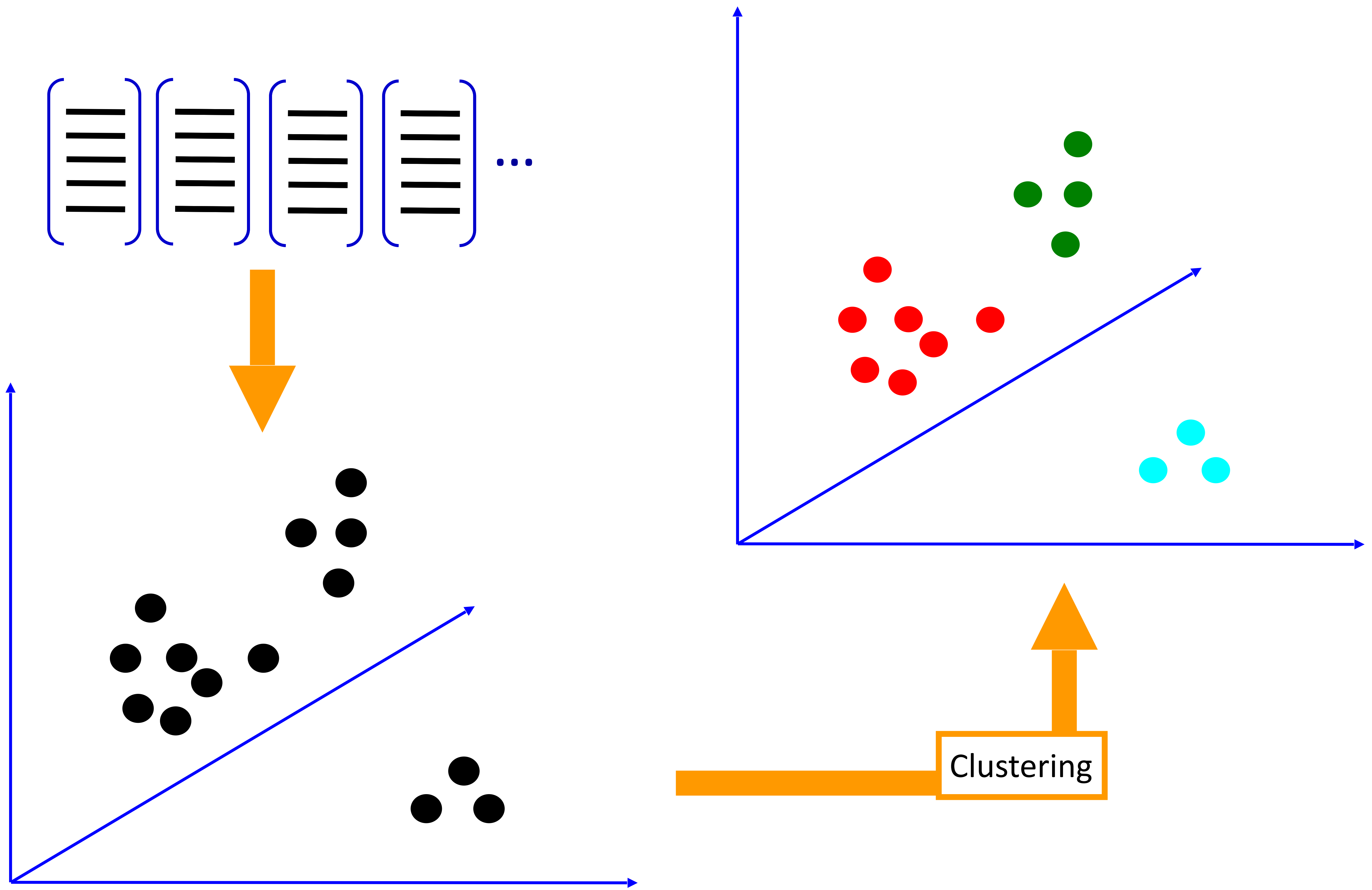
[Mata, Chum, Urban & Pajdla, '02]

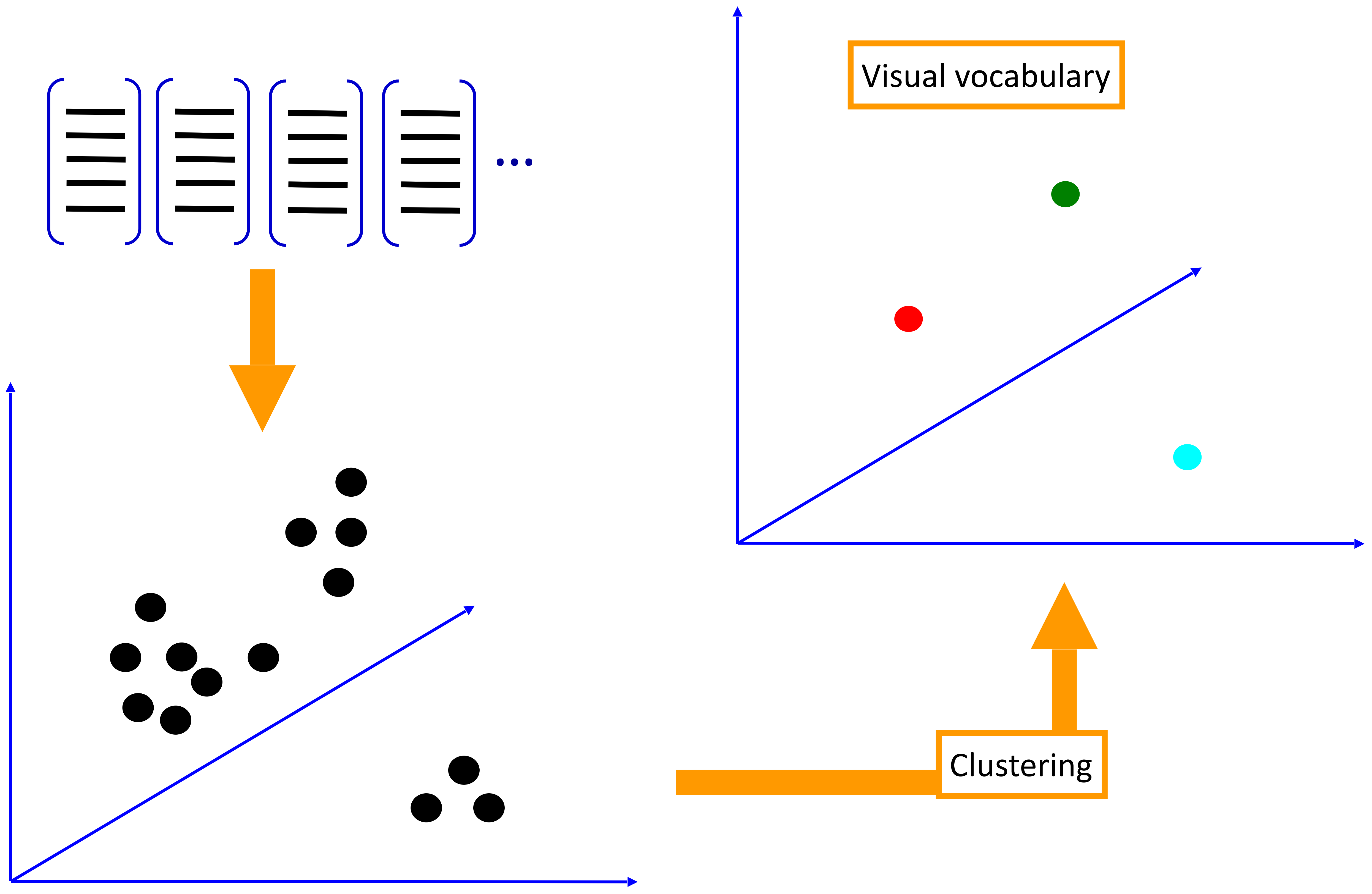
[Sivic & Zisserman, '03]



Visual Vocabulary




Clustering



Visual vocabulary
Clustering

K-means Clustering

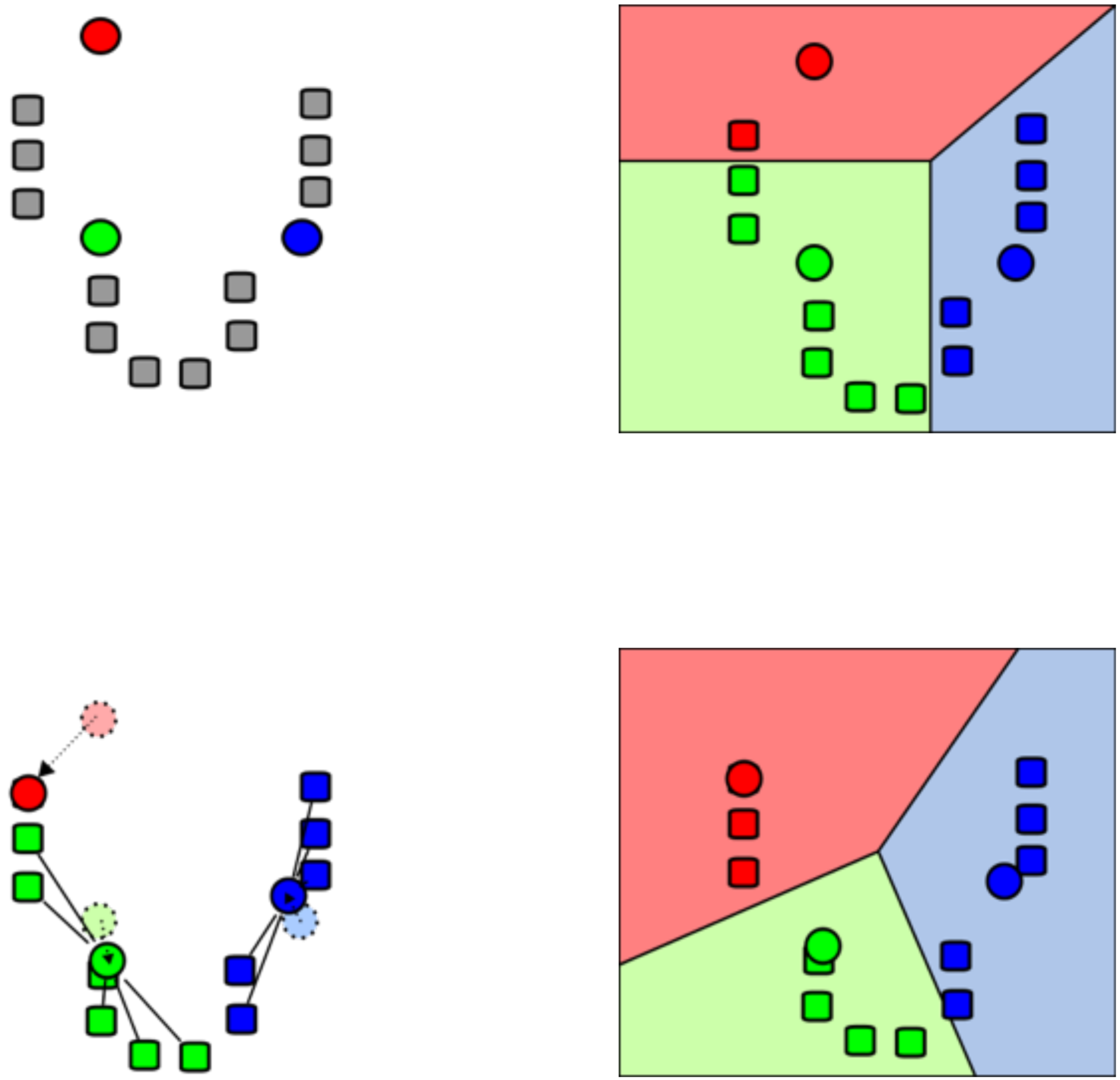
Given k:

1. Select initial centroids at random.

2. Assign each object to the cluster with the nearest centroid.

3. Compute each centroid as the mean of the objects assigned to it.

4. Repeat previous 2 steps until no change.

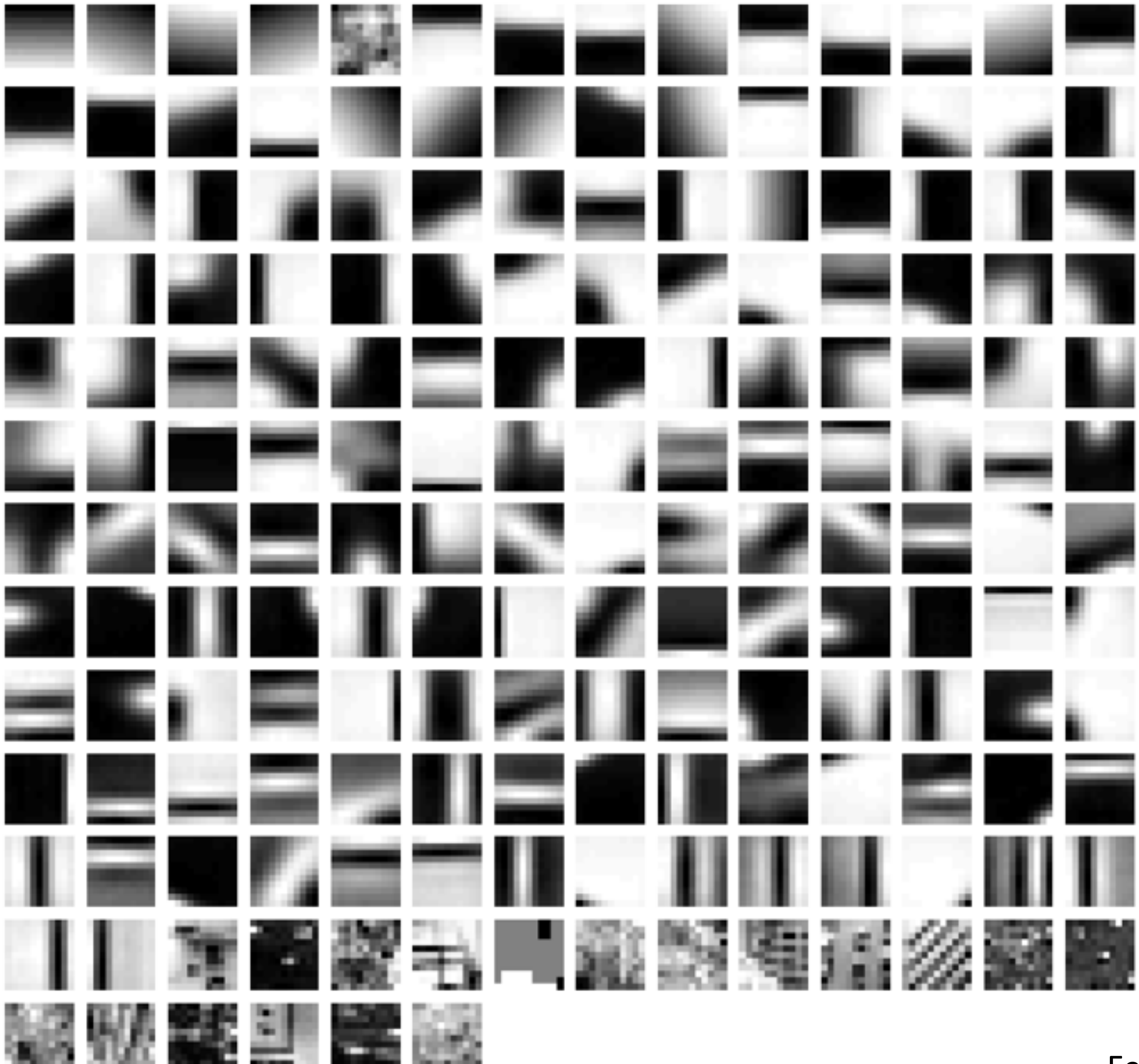


Clustering and vector quantization

- Clustering is a common method for learning a visual vocabulary or codebook
  - Unsupervised learning process
  - Each cluster center produced by k-means becomes a codevector
  - Codebook can be learned on separate training set
  - Provided the training set is sufficiently representative, the codebook will be “universal”

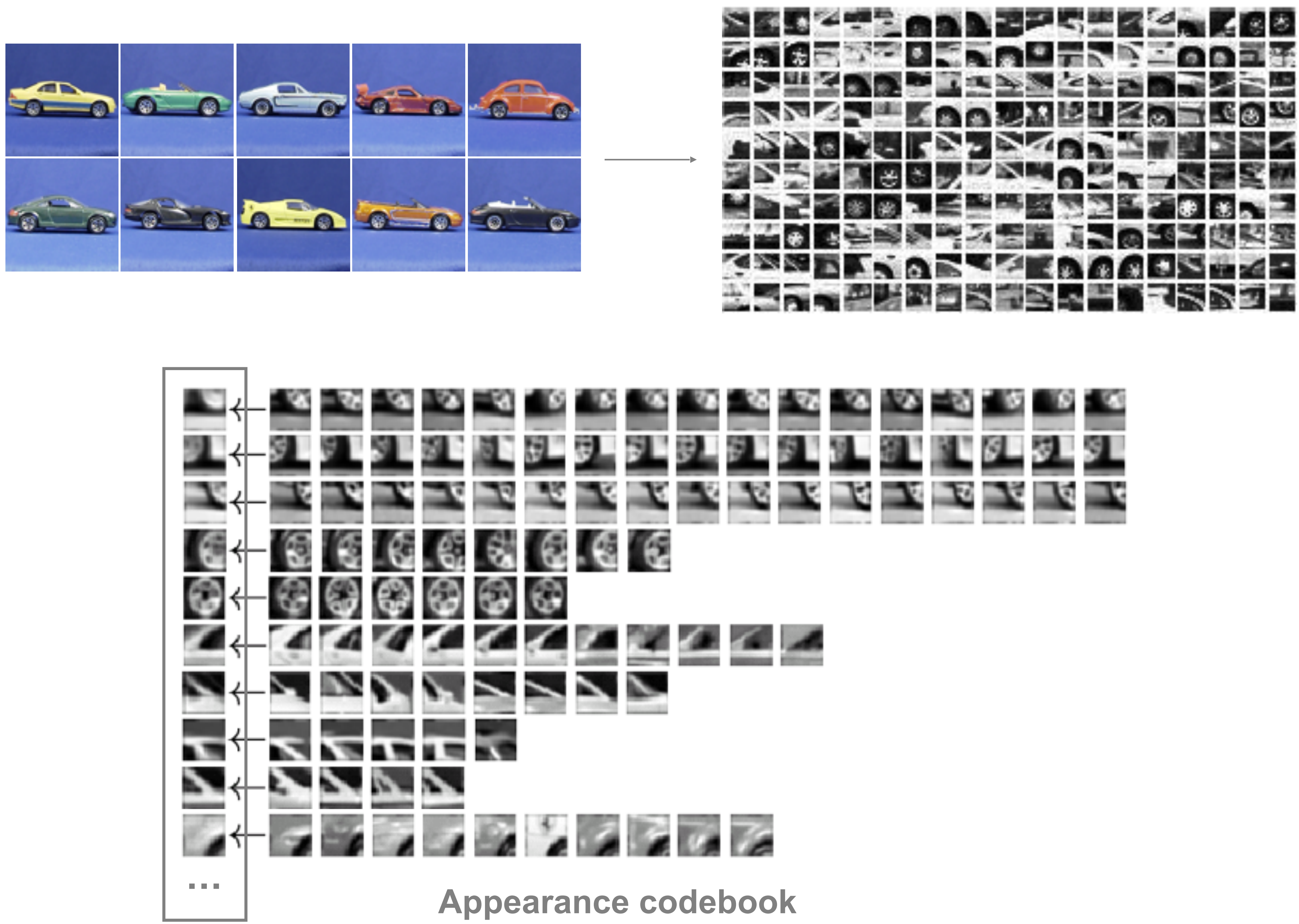
- The codebook is used for quantizing features
  - A *vector quantizer* takes a feature vector and maps it to the index of the nearest codevector in a codebook
  - Codebook = visual vocabulary
  - Codevector = visual word

Example visual vocabulary



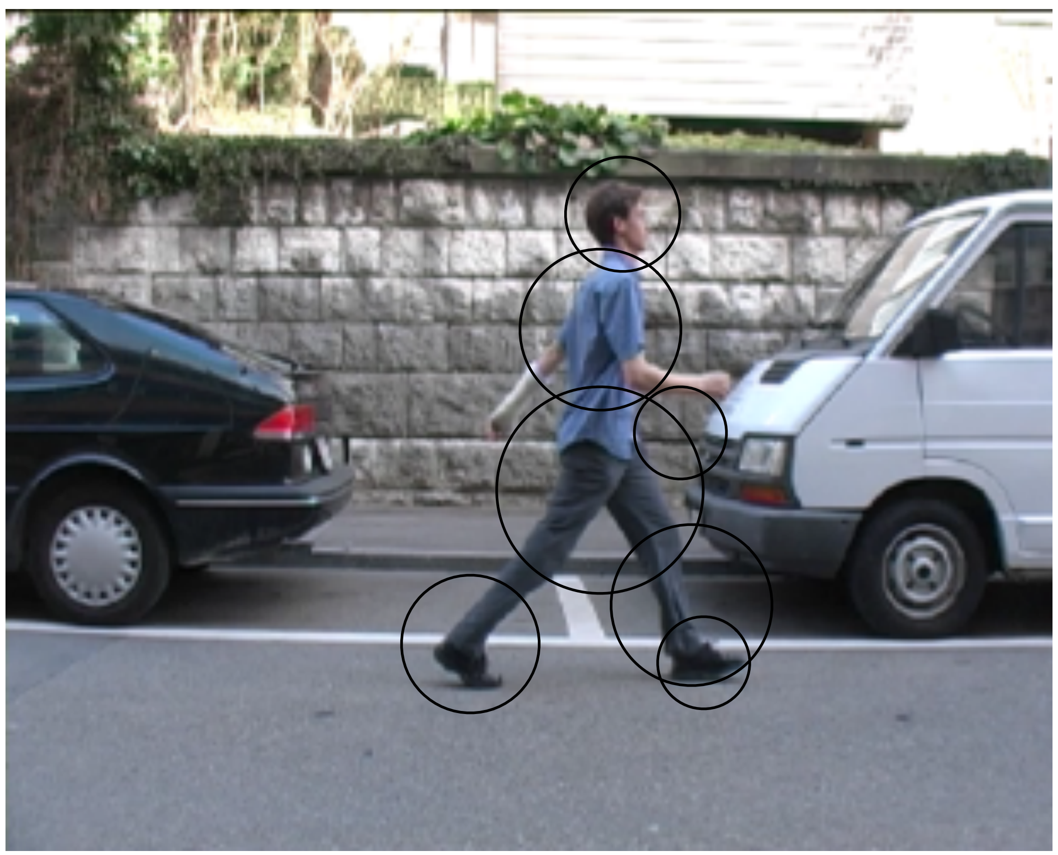
Fei-Fei et al. 2005

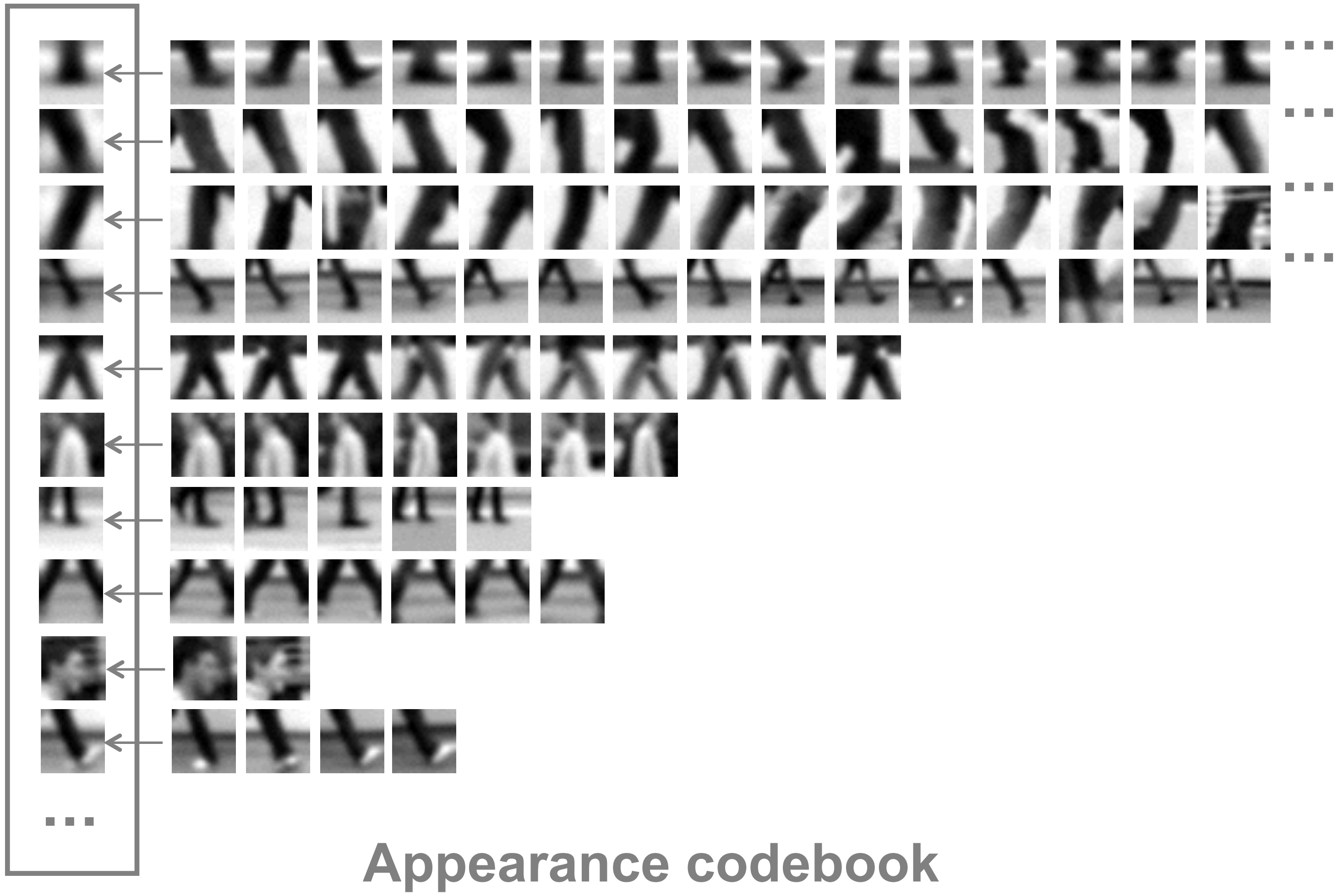
Example codebook



Source: B. Leibe

Another codebook



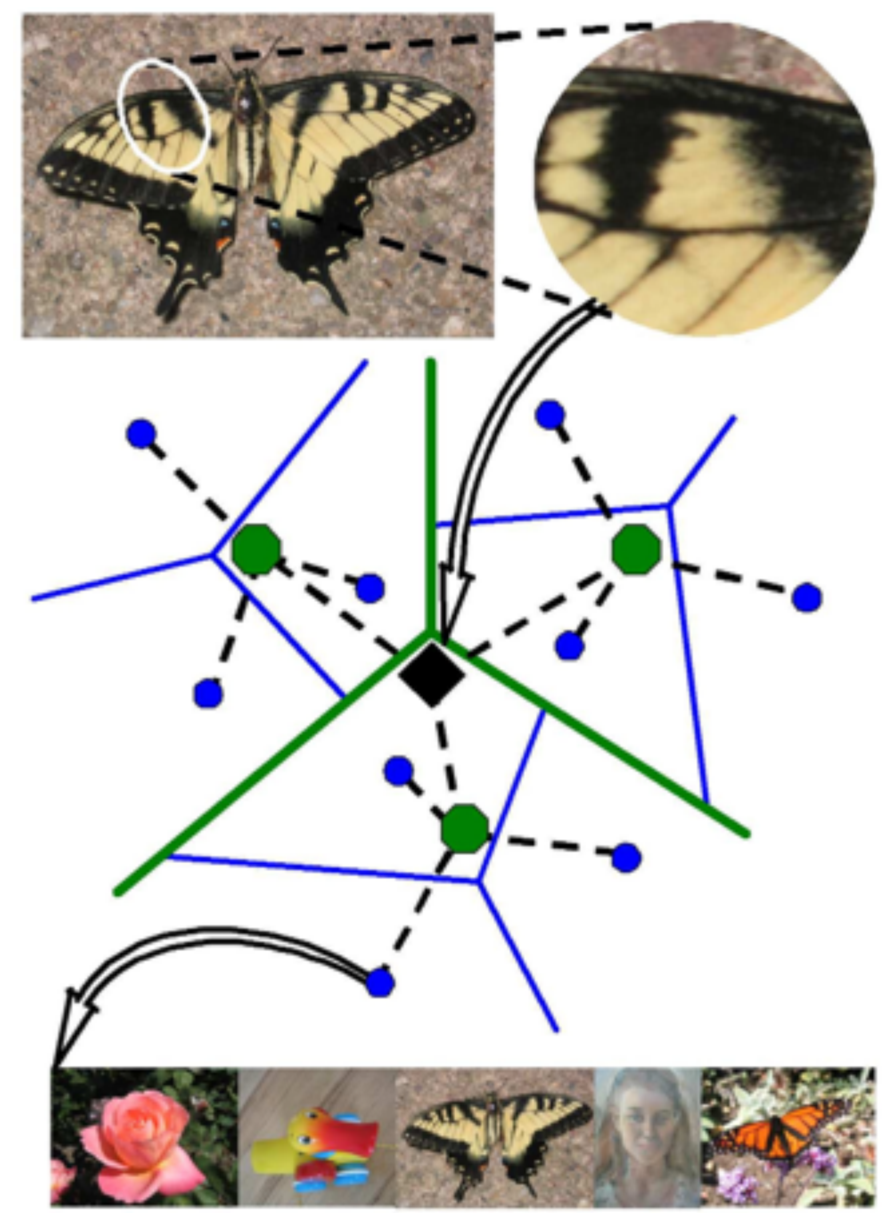


Appearance codebook

Source: B. Leibe

Visual vocabularies: 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 & Stewenius, 2006)



3. Image representation

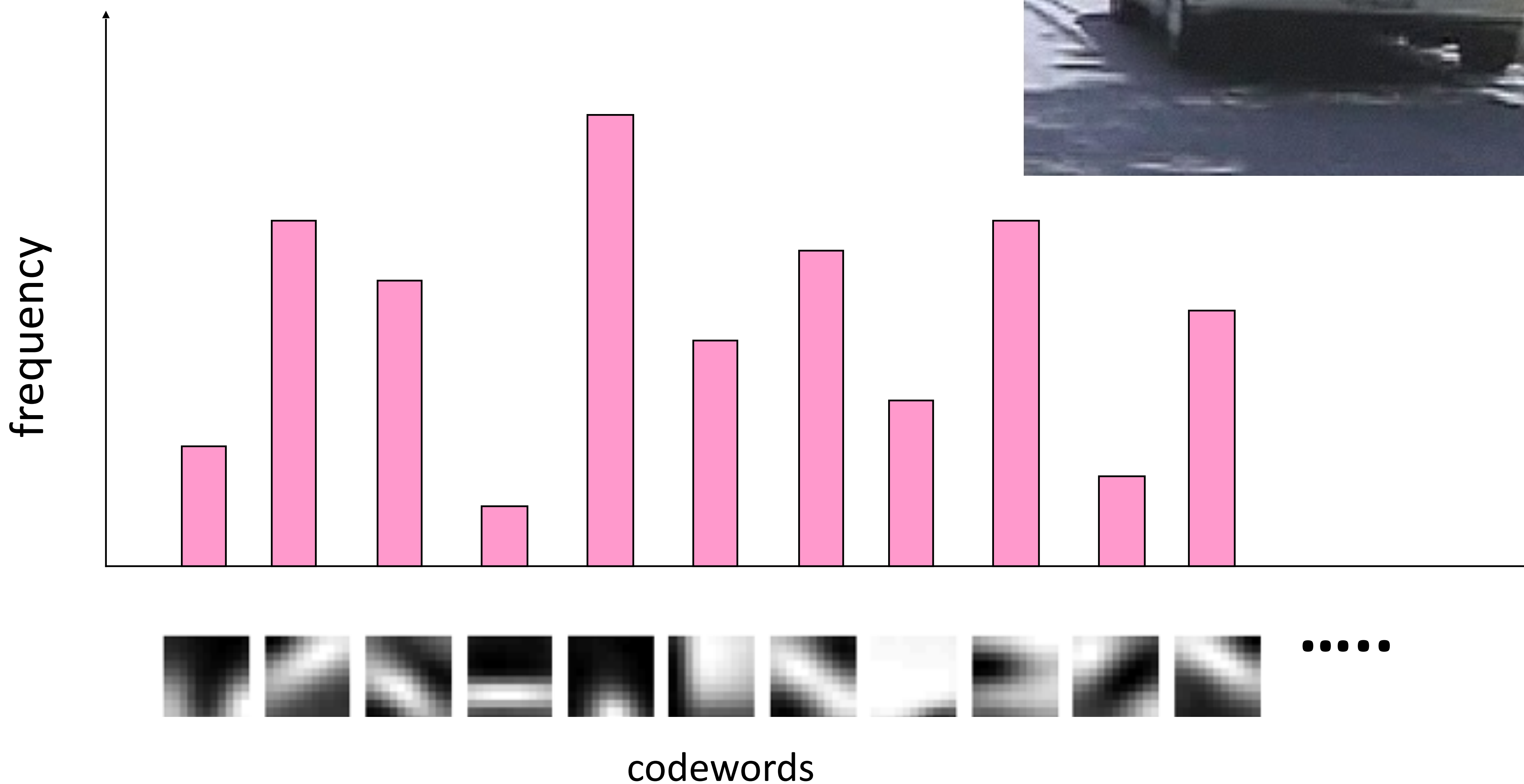
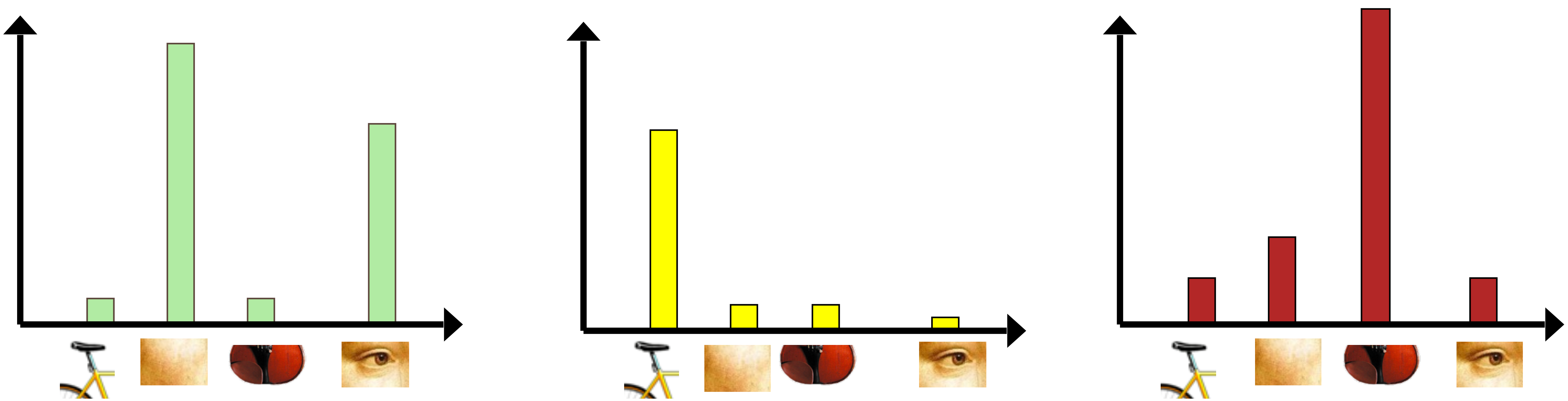


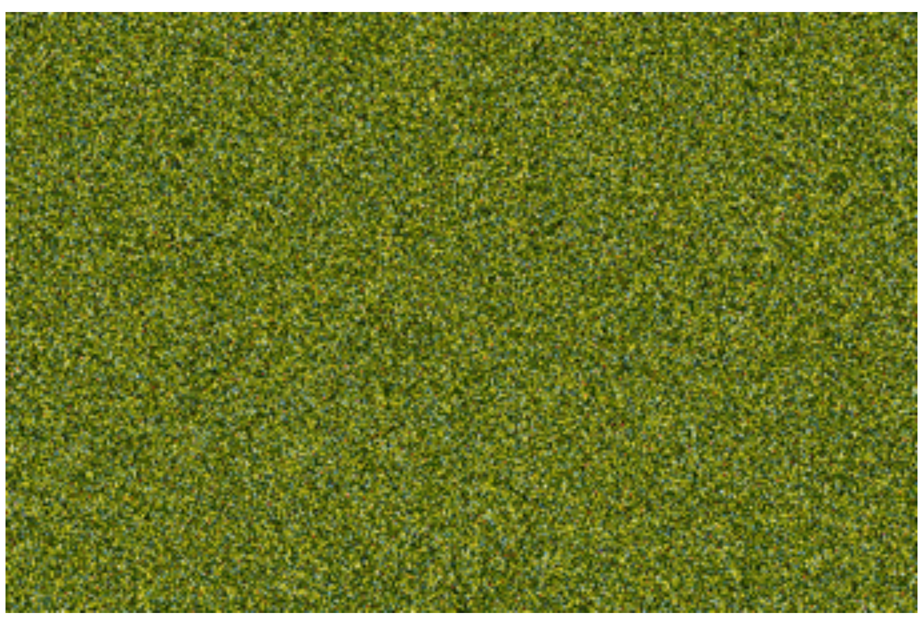
Image classification

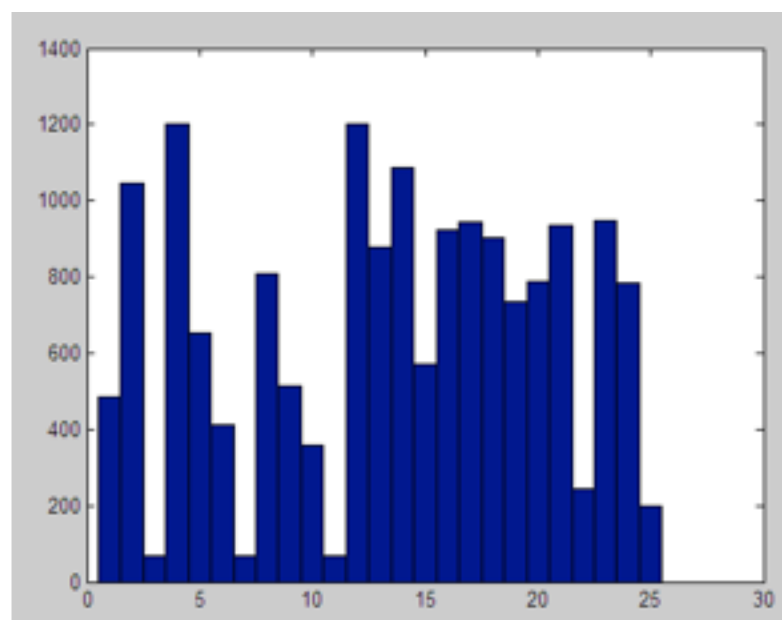
- Given the bag-of-features representations of images from different classes, learn a classifier using machine learning

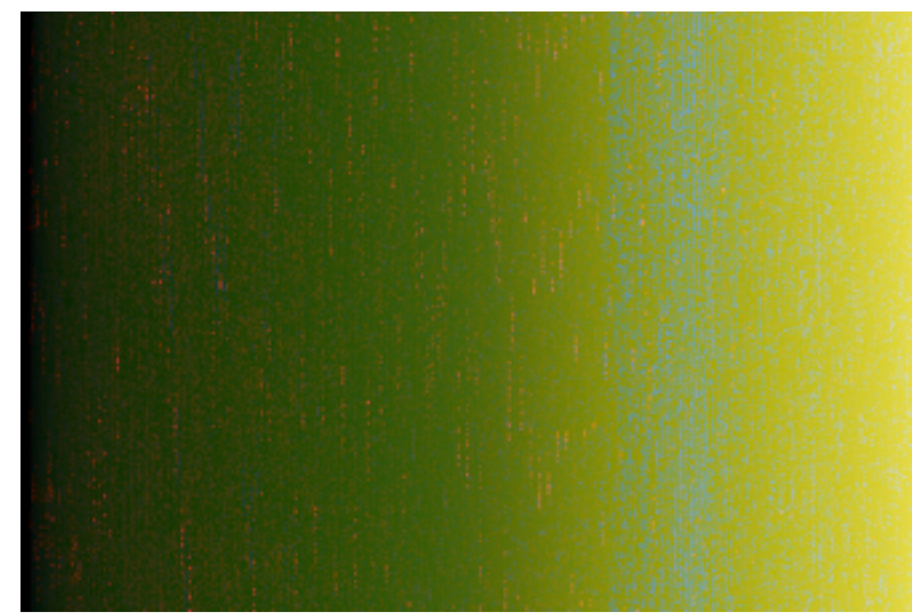


Extension to bag-of-words models

But what about layout?



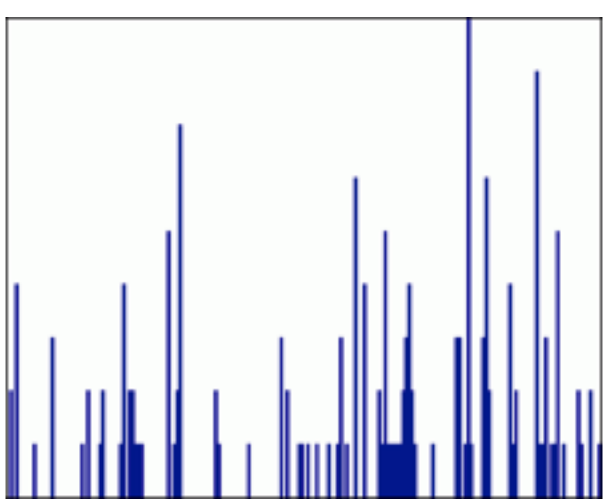




All of these images have the same color histogram

Spatial pyramid representation

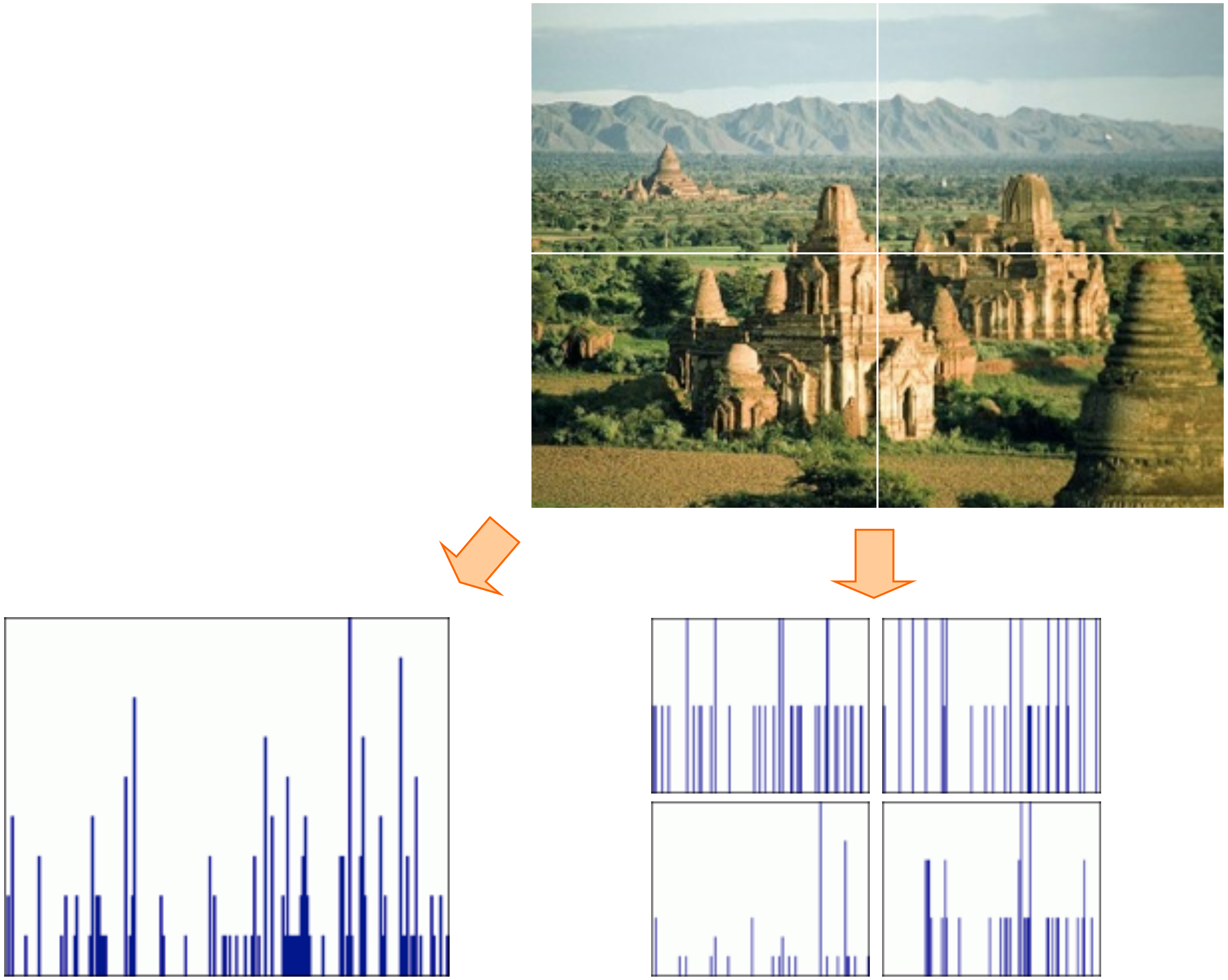
- Extension of a bag of features
- Locally orderless representation at several levels of resolution



Lazebnik, Schmid & Ponce (CVPR 2006)

Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution



Lazebnik, Schmid & Ponce (CVPR 2006)

Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution

Lazebnik, Schmid & Ponce (CVPR 2006)