Image representation

370: Intro to Computer Vision

Subhransu Maji

April 22 & 24, 2025

College of
INFORMATION AND
COMPUTER SCIENCES

UMASS
AMHERST

Steps — a classical perspective

Training

Training Images

→ Image Features → Training → Learned model

Training Labels → Training

Testing

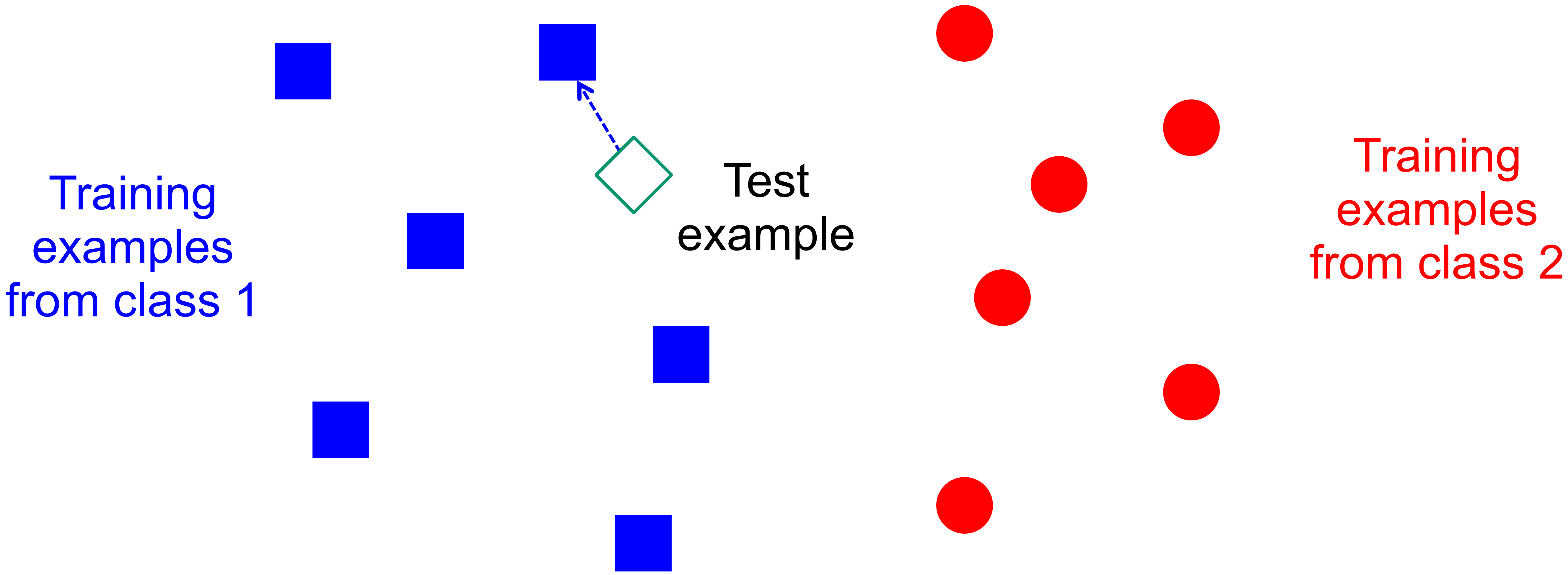
Test Image

→ Image Features → Prediction

Learned model → Prediction

Classifiers: Nearest neighbor

f(**x**) = label of the training example nearest to **x**

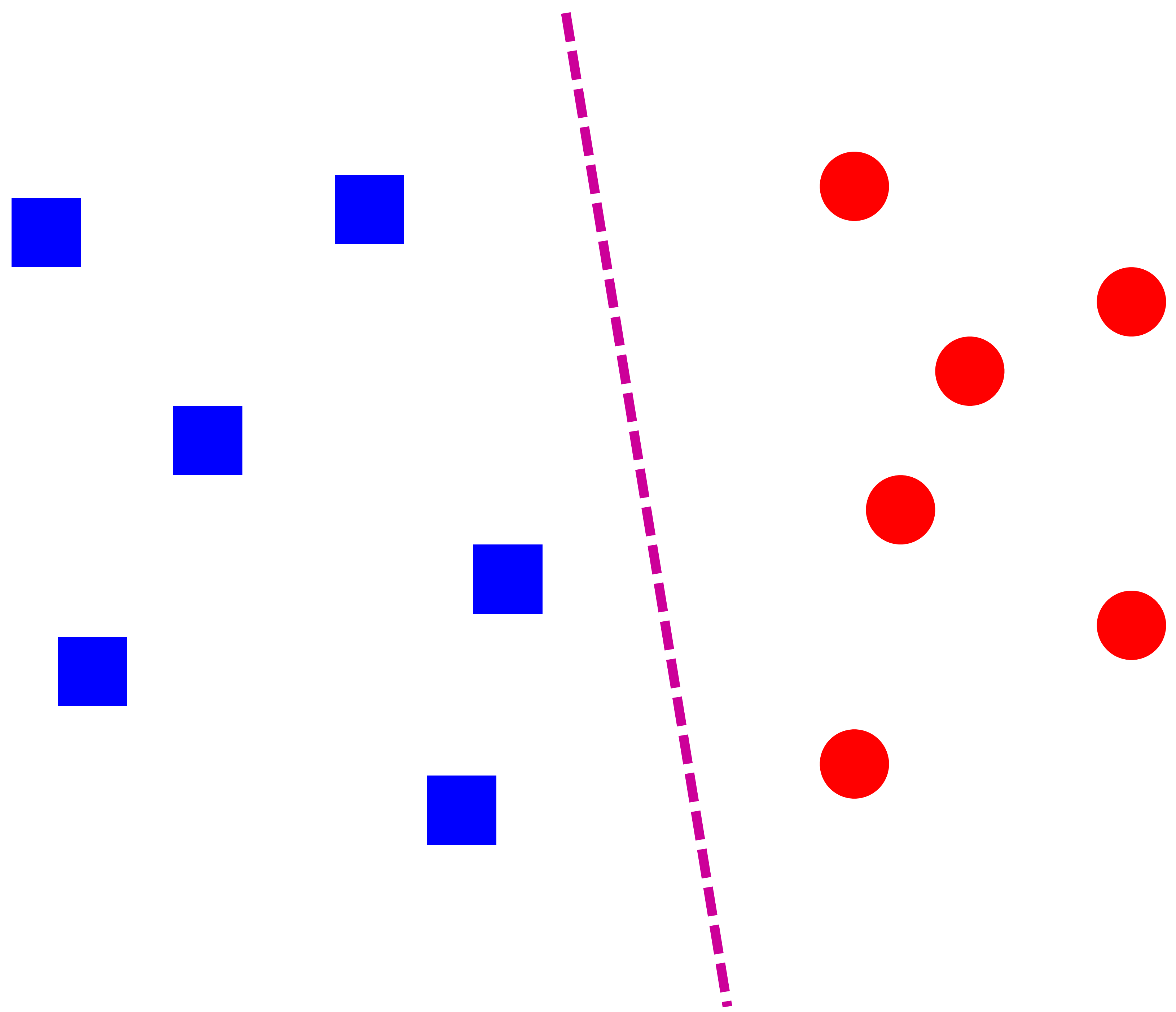


All we need is a distance function for our inputs
No training required!

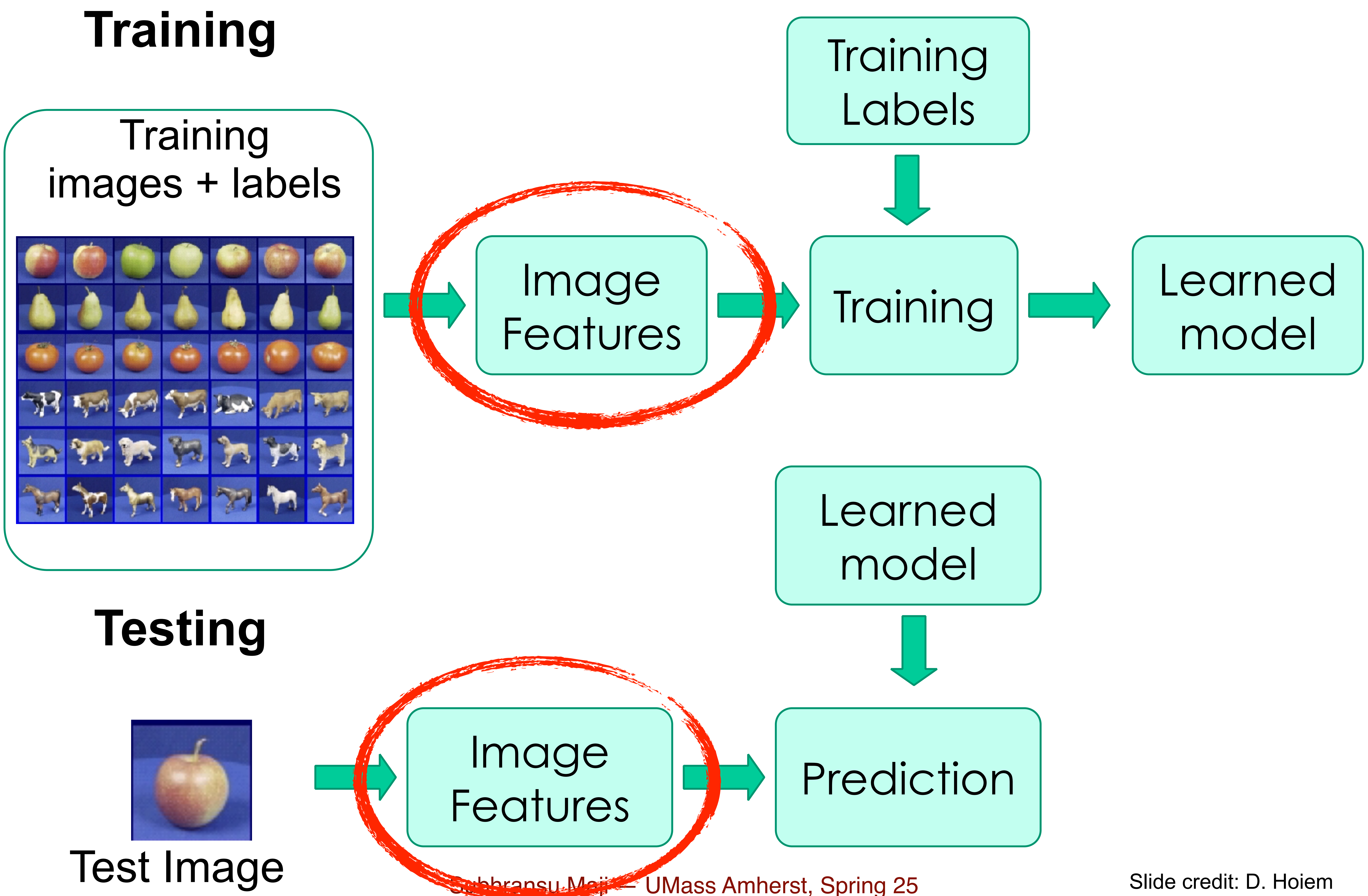
Classifiers: Linear

Find a *linear function* to separate the classes:

$$f(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$$



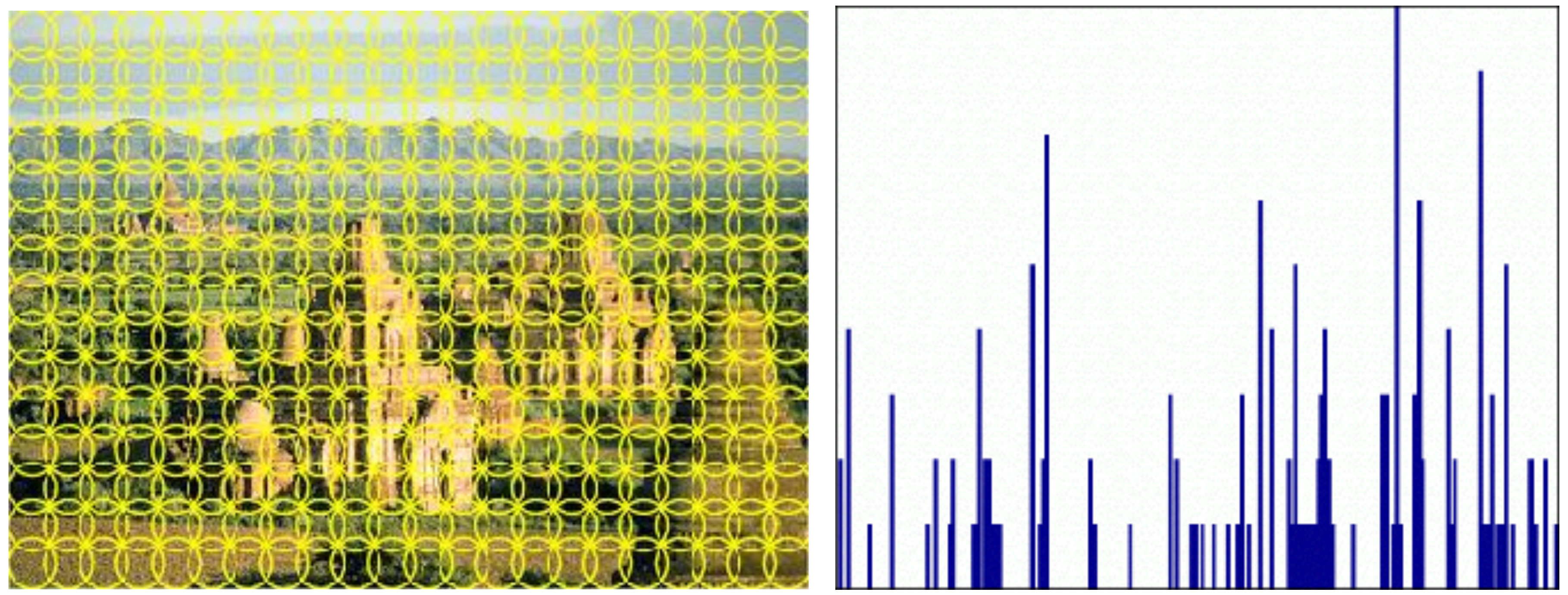
The machine learning approach: today



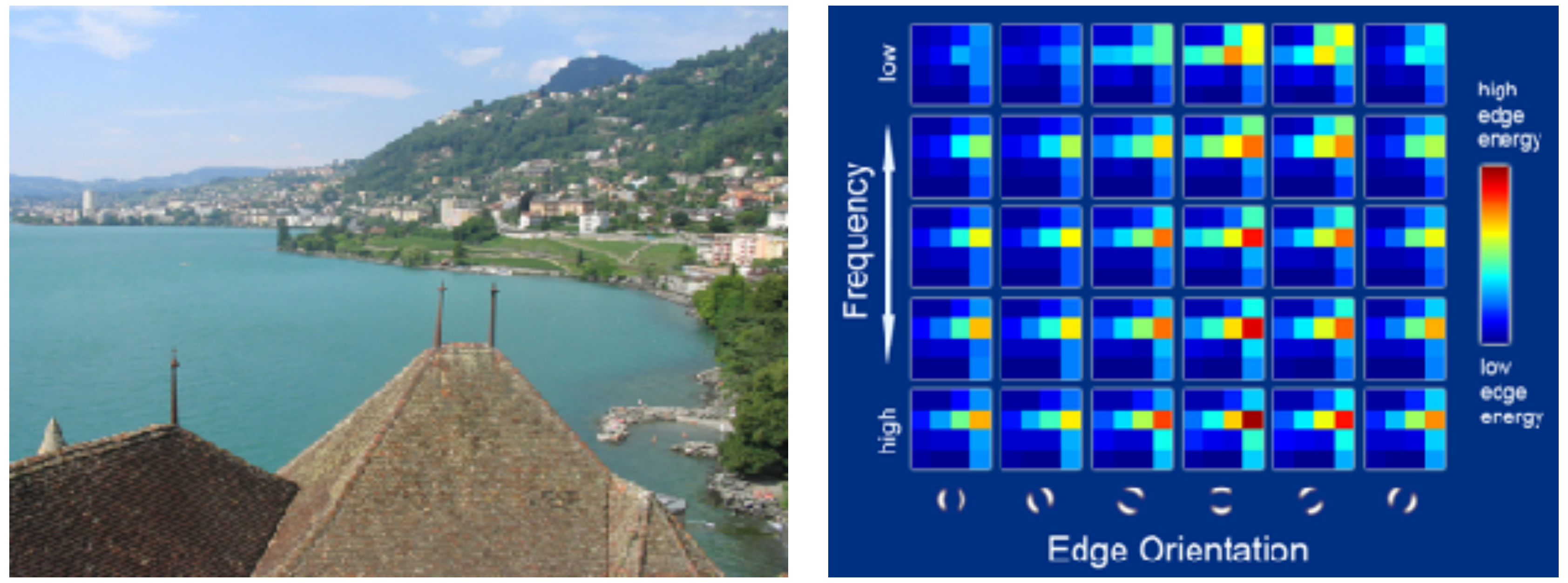
Features (examples)

Raw pixels (and simple functions of raw pixels)

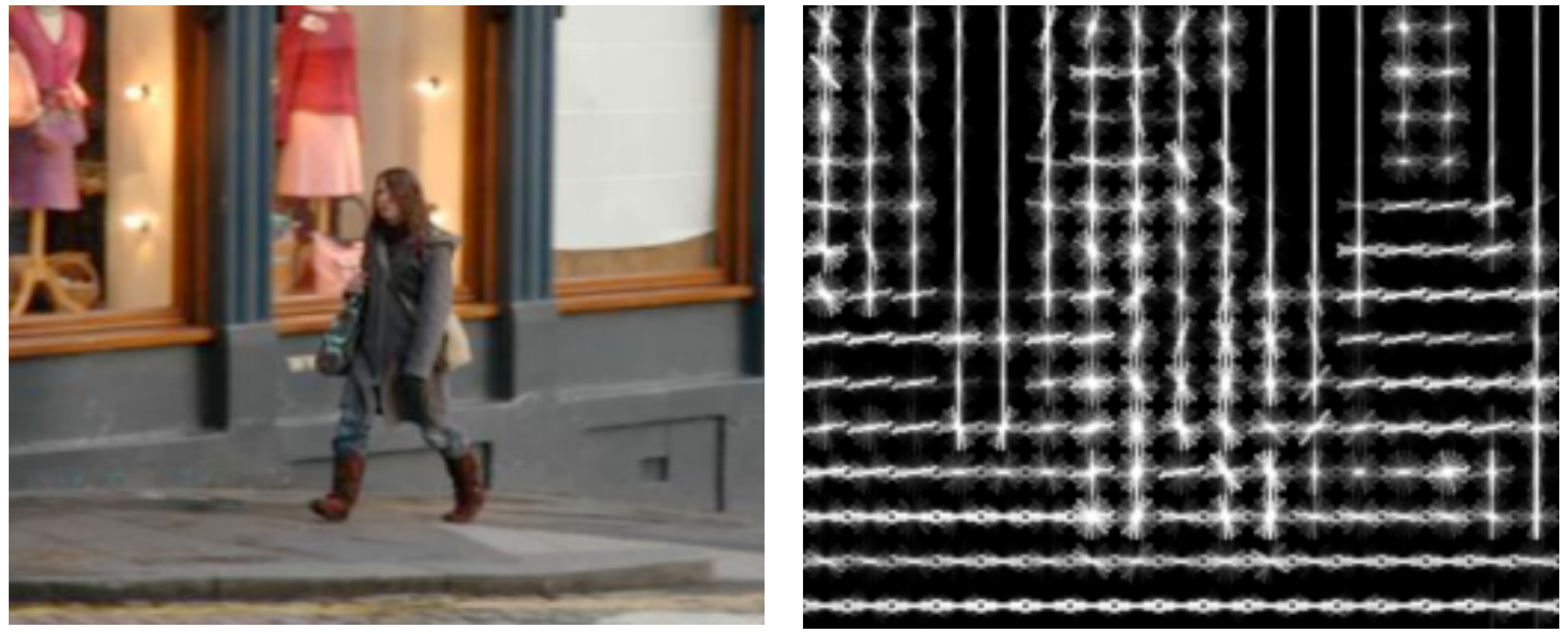
Histograms, bags of features



GIST descriptors



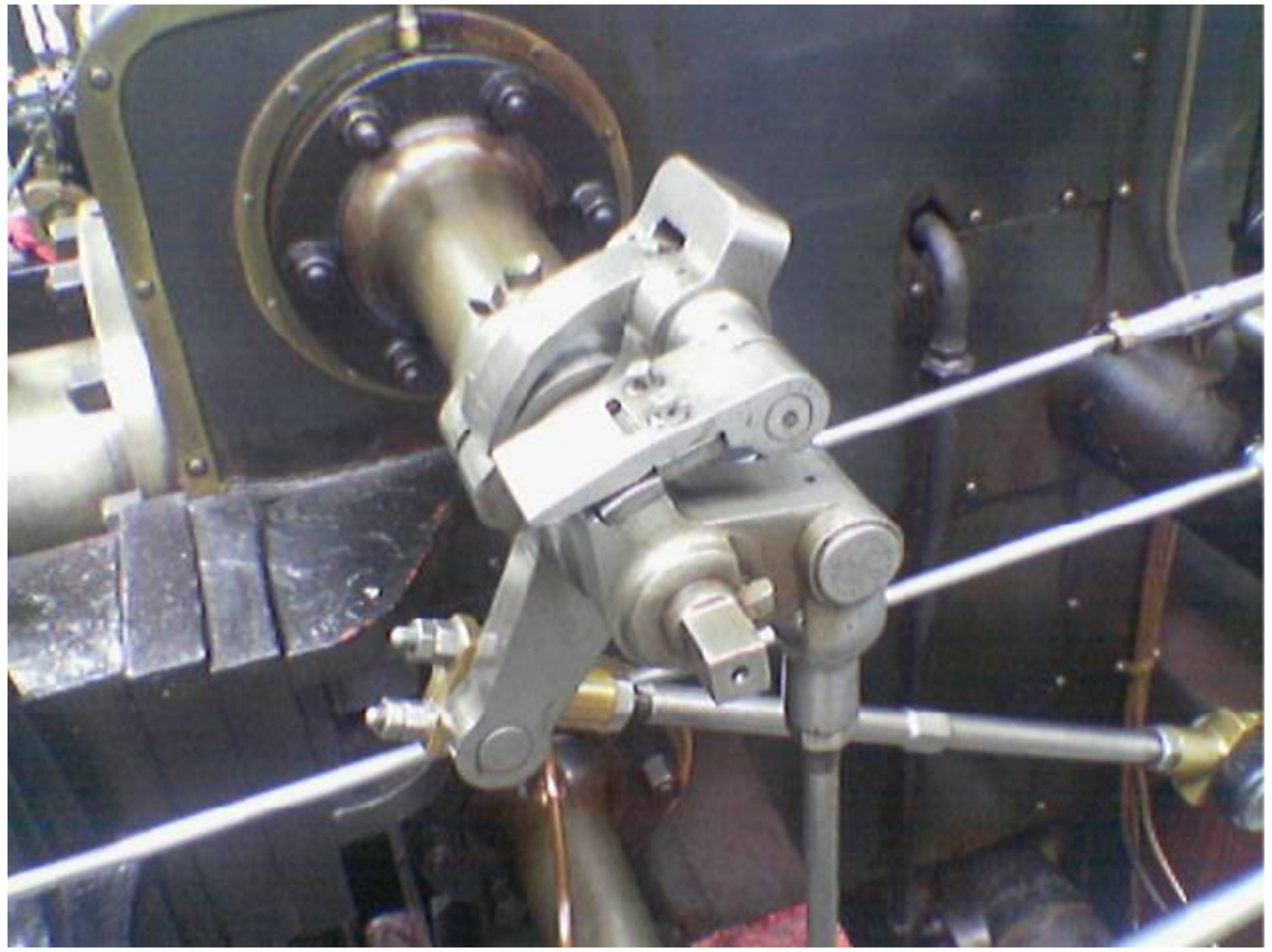
Histograms of oriented gradients(HOG)



What is a feature map?

Any transformation of an image into a new representation

Example: transform an image into a binary edge map



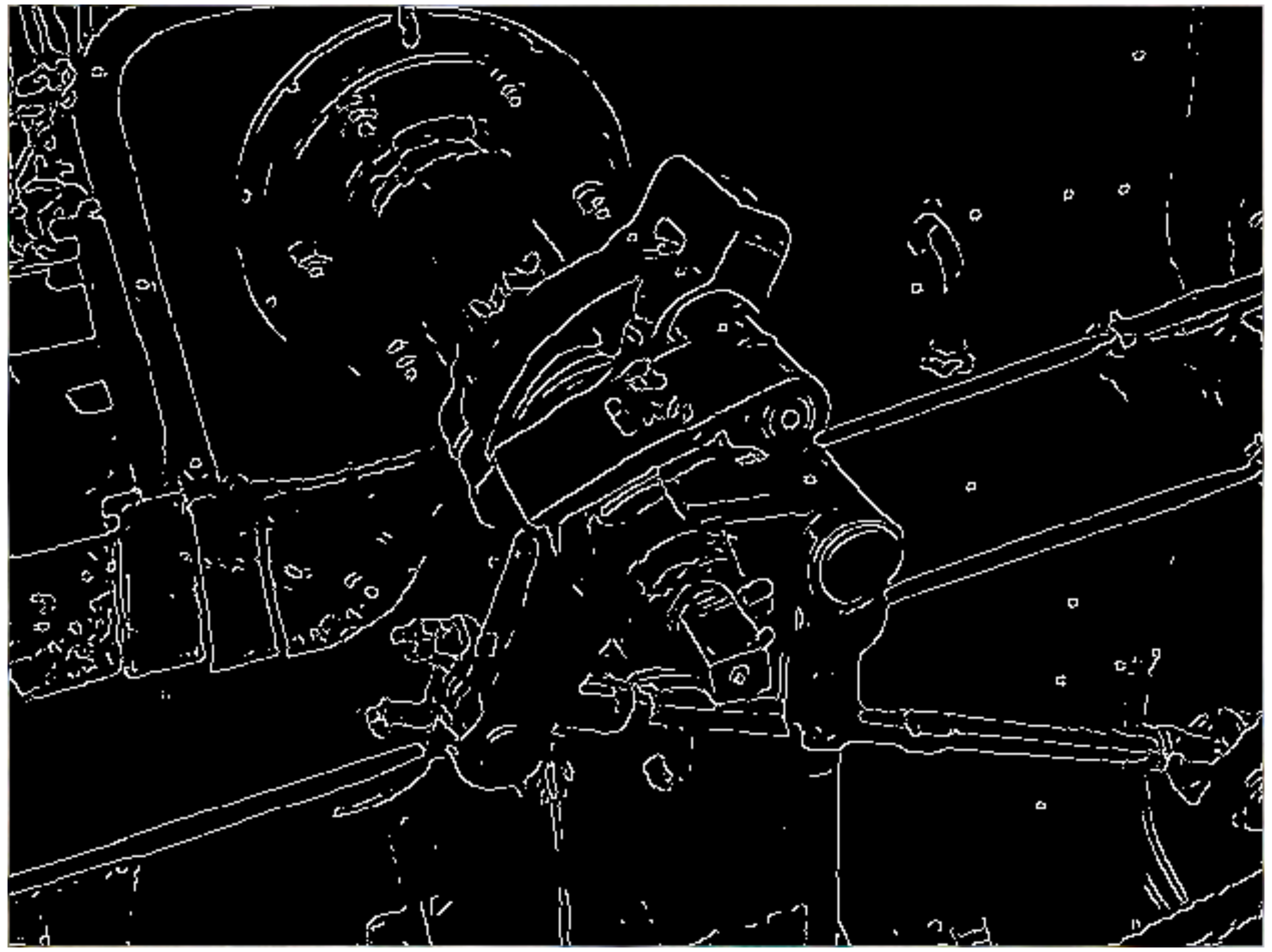


Image source: wikipedia

Feature map goals

Introduce invariance to nuisance factors

- Illumination changes
- Small translations, rotations, scaling, shape deformations

Figure 1.3: Variation in appearance due to a change in illumination

Preserve useful information: e.g., spatial structure

Image: [Fergus05]

We will discuss …

Two popular image features

- Histogram of Oriented Gradients (HOG)
- Bag of Visual Words (BoVW)

Applications of these features

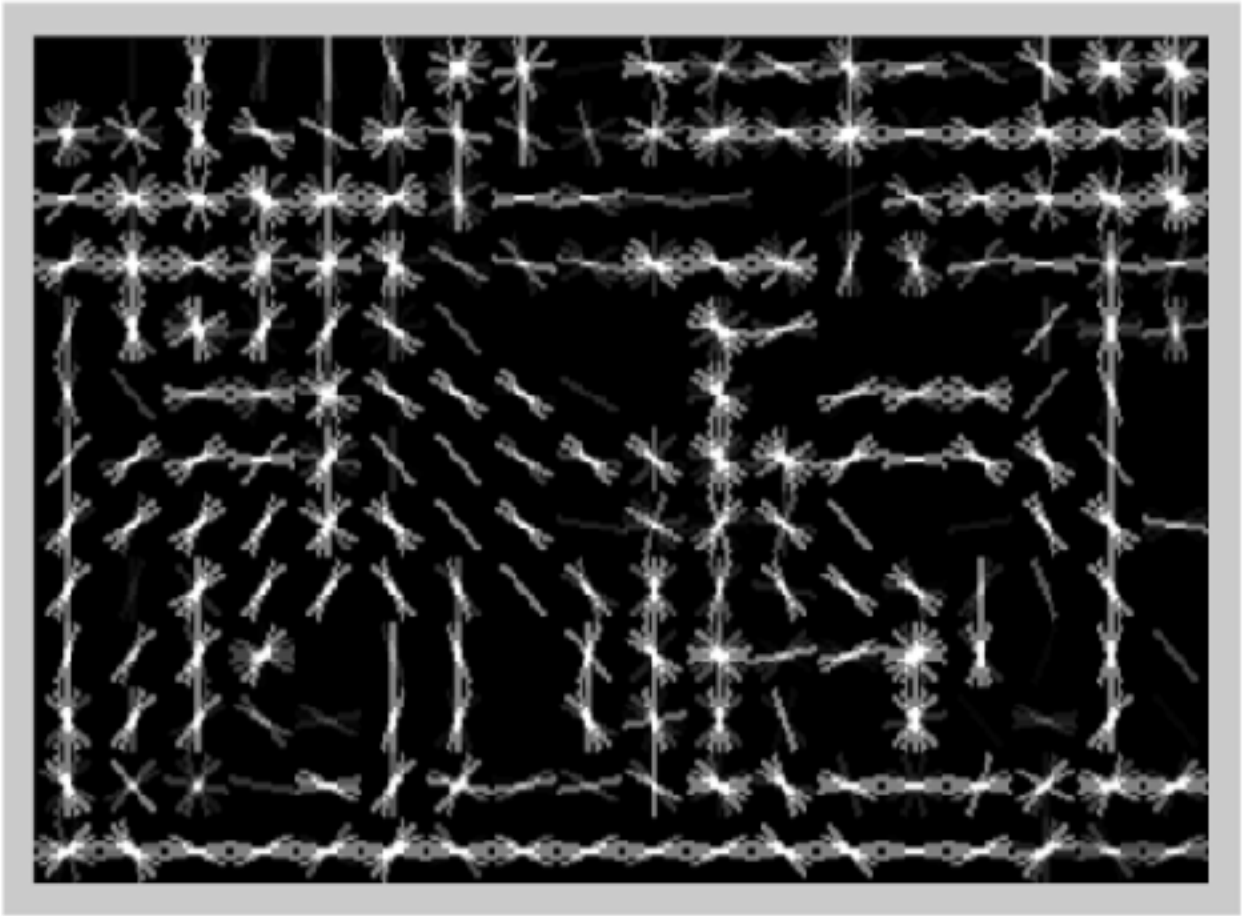
Histogram of Oriented Gradients

Introduced by Dalal and Triggs (CVPR 2005)

An extension of the SIFT feature

HOG properties:

- Preserves the overall structure of the image
- Provides robustness to illumination and small deformations

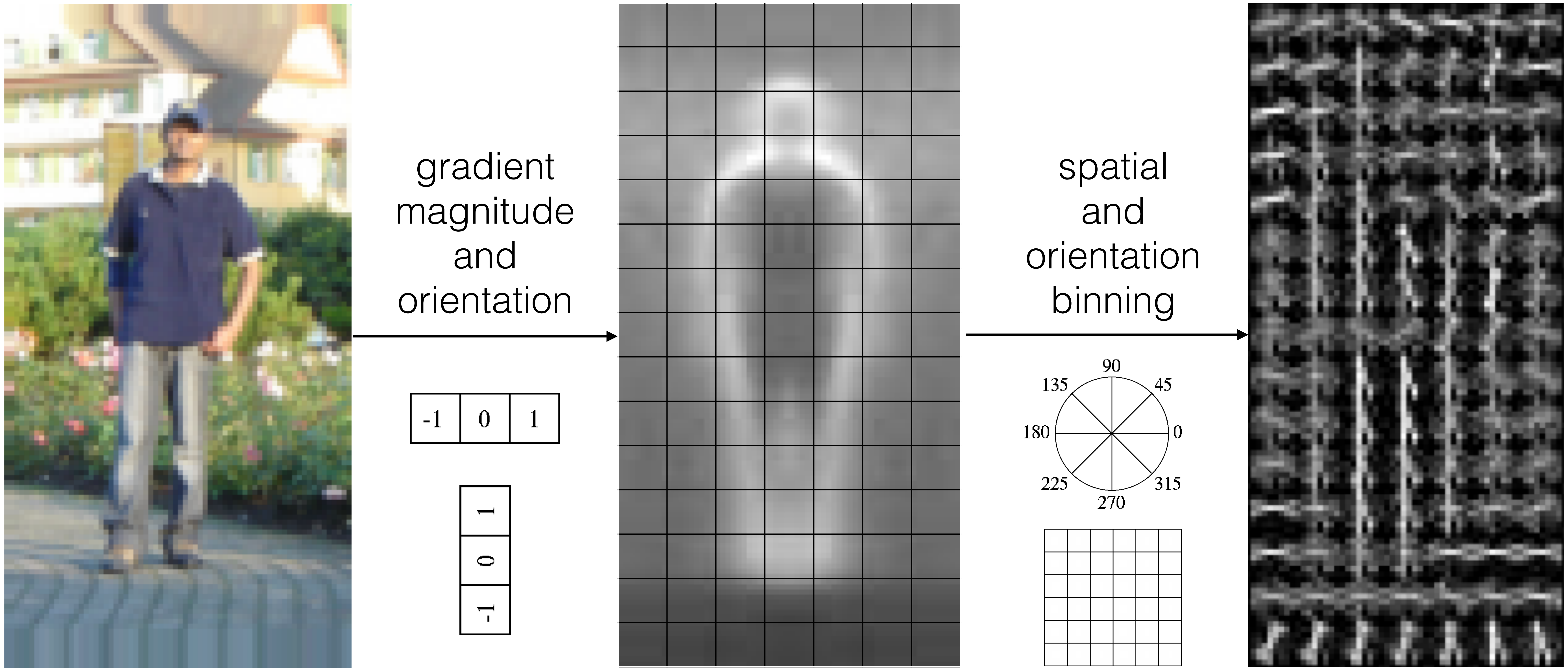


HOG feature

HOG feature: basic idea

Divide the image into blocks

Compute histograms of gradients for each regions

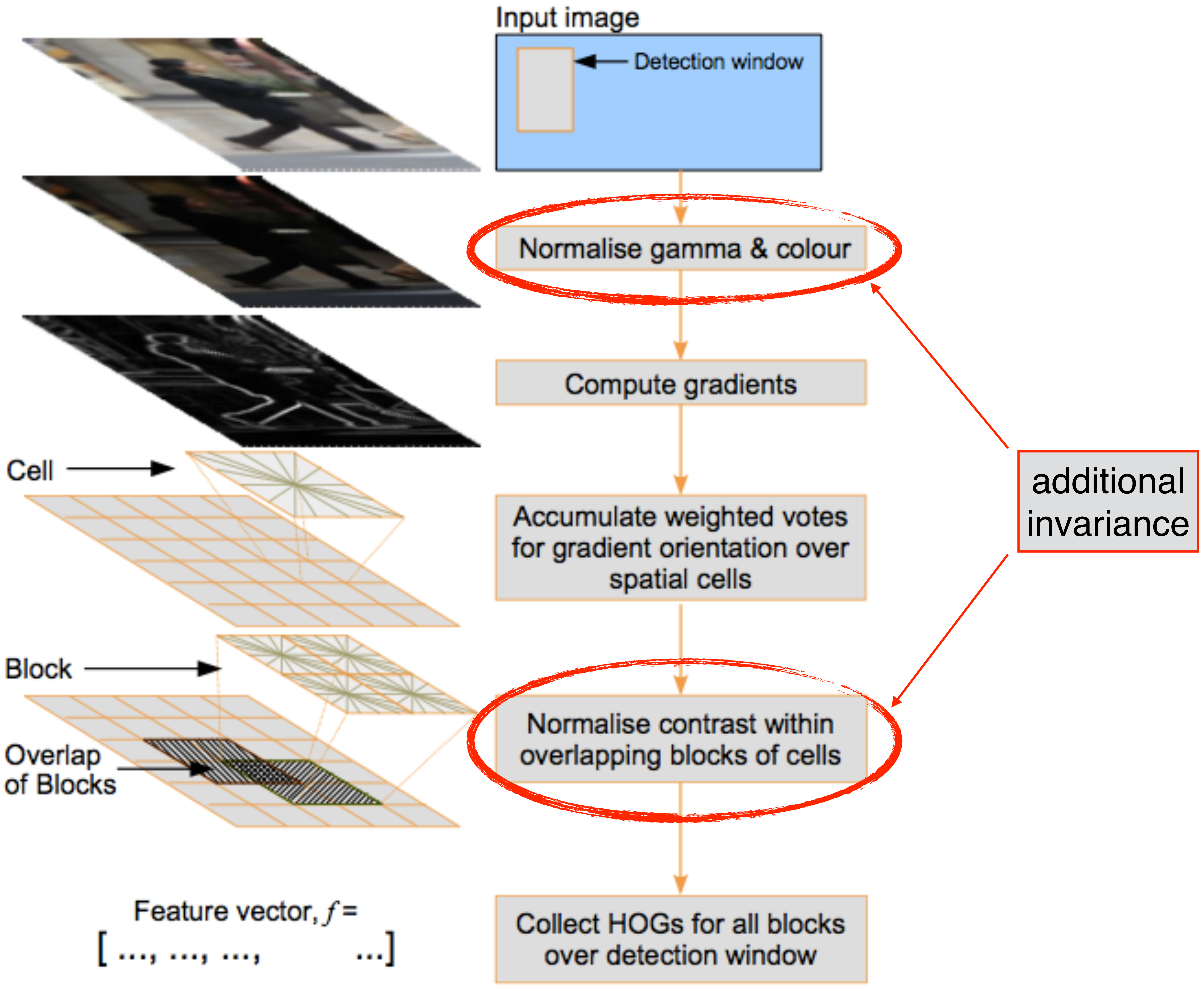


image

Gradient norm

HOG feature

HOG feature: full pipeline



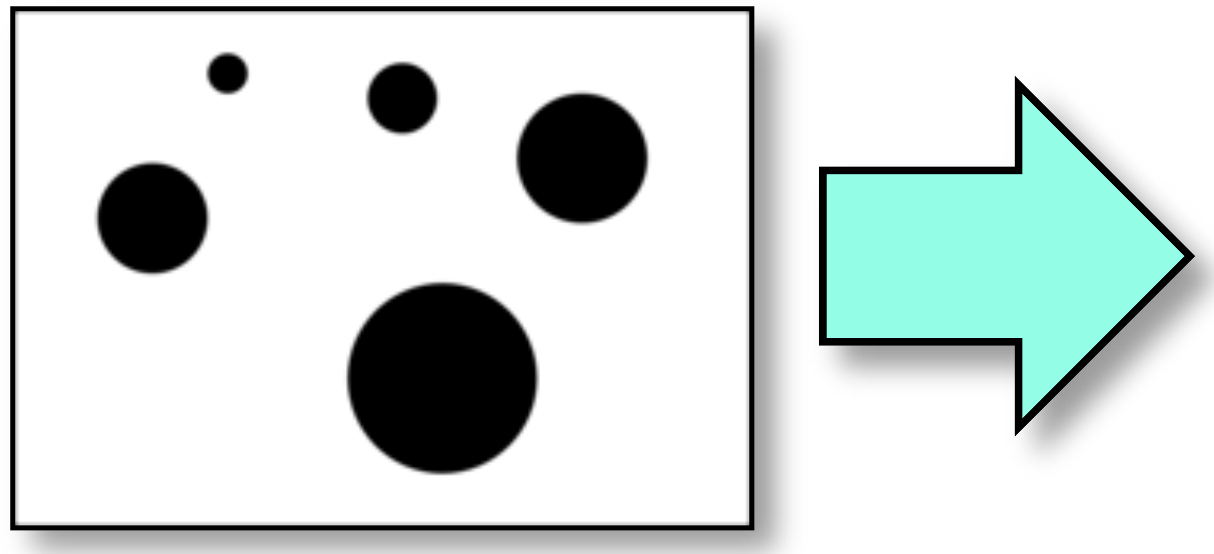
Effect of bin-size

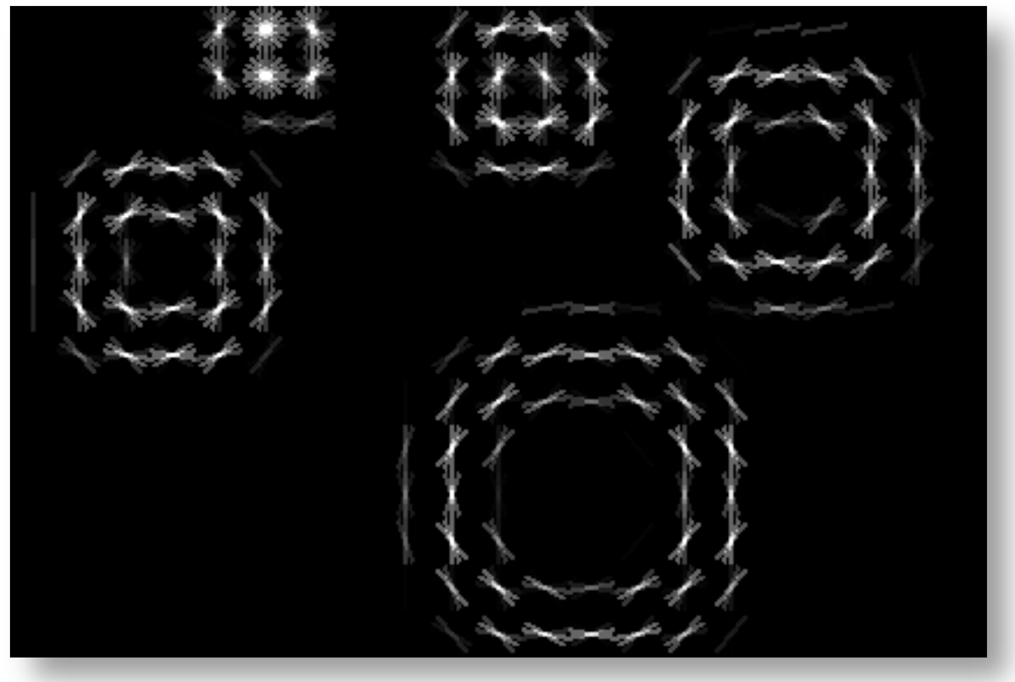
Smaller bin-size: better spatial resolution

Larger bin-size: better invariance to deformations

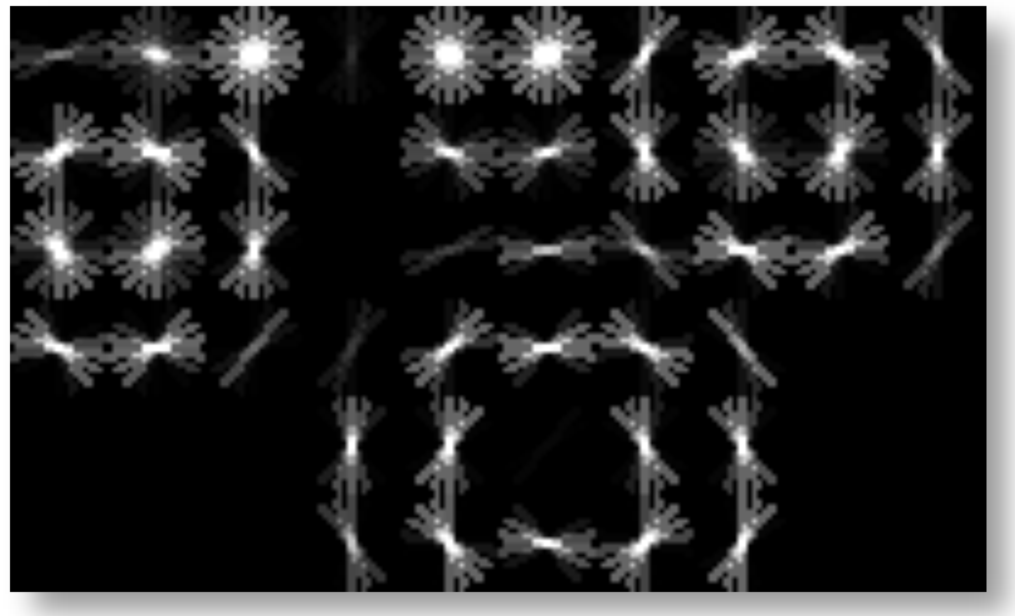
Optimal value depends on the object category being modeled

- e.g. rigid vs. deformable objects





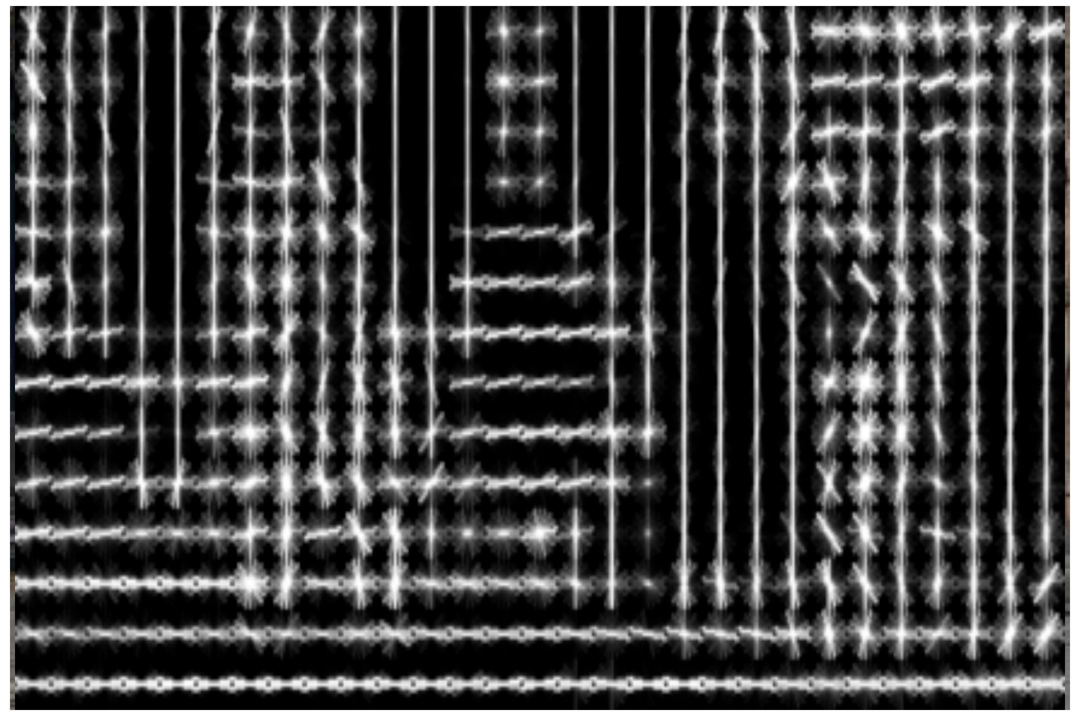
10x10 cells



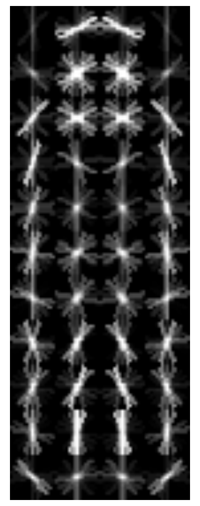
20x20 cells

Template matching with HOG

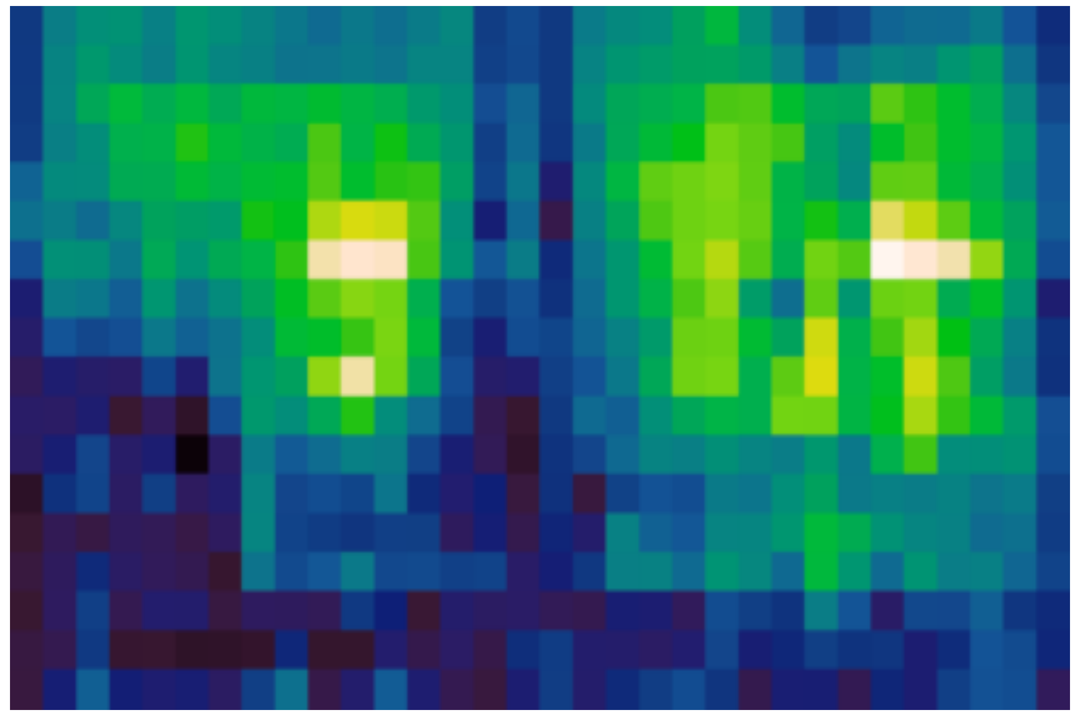
HOG feature map



Template



Detector response map



Compute the HOG feature map for the image

Convolve the template with the feature map to get score

Find peaks of the response map (non-max suppression)

Apply it at multiple scales

Multi-scale detection

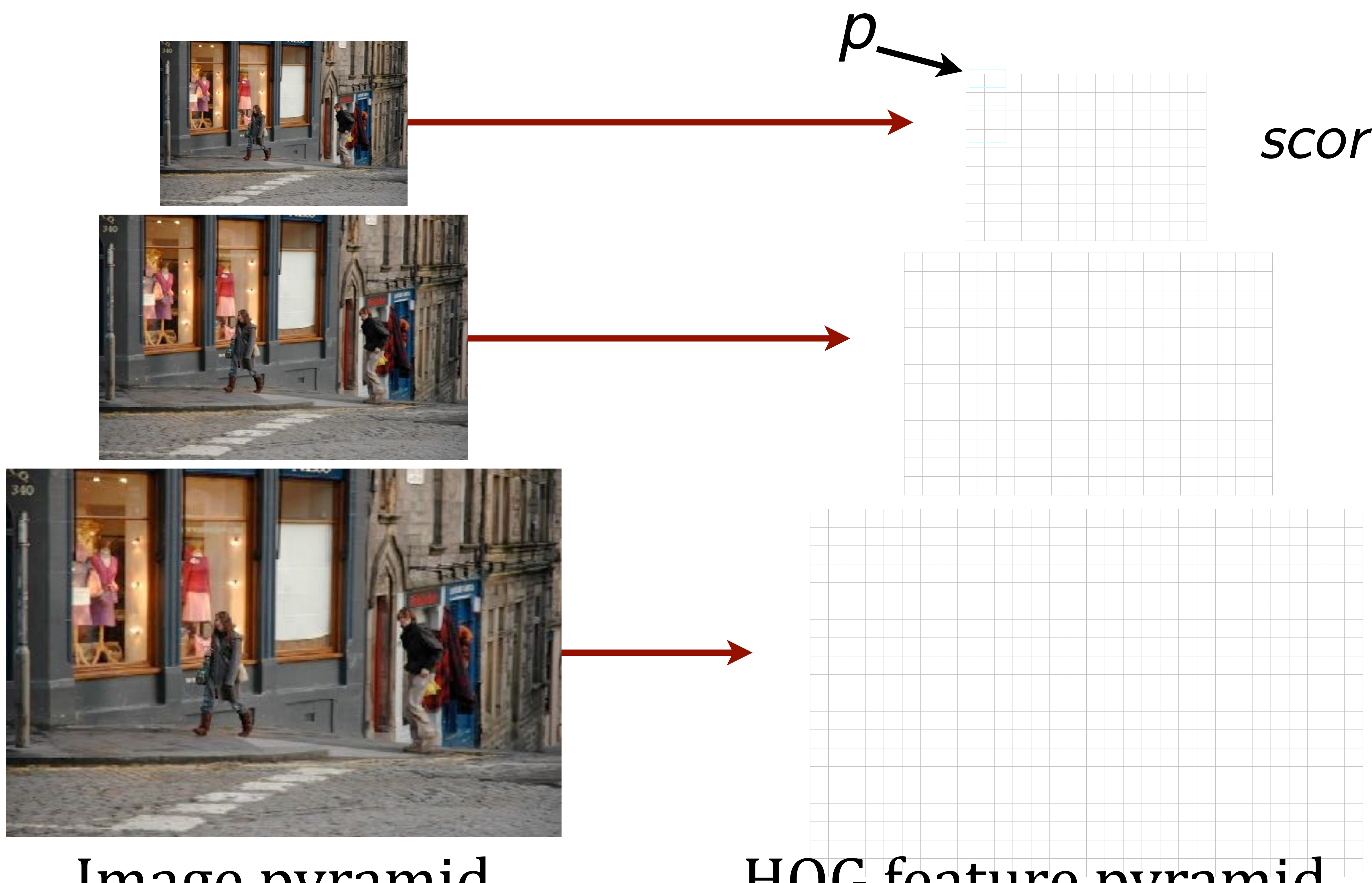


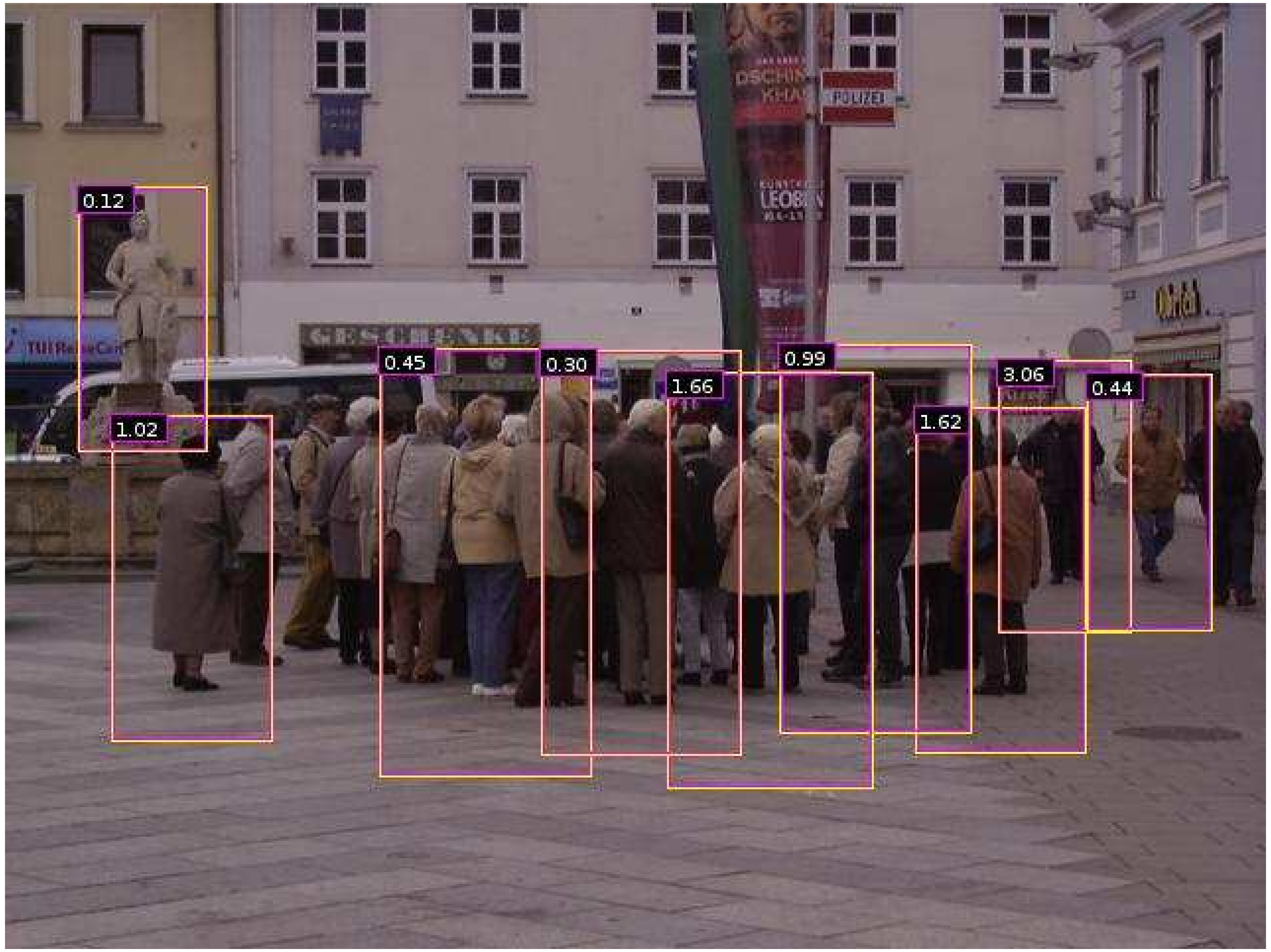
Image pyramid HOG feature pyramid

$$score(I, p) = \mathbf{w} \cdot \phi(I, p)$$

Compute HOG of the whole image at multiple resolutions

Score each sub-windows of the feature pyramid

Example detections



N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

Example detections

N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

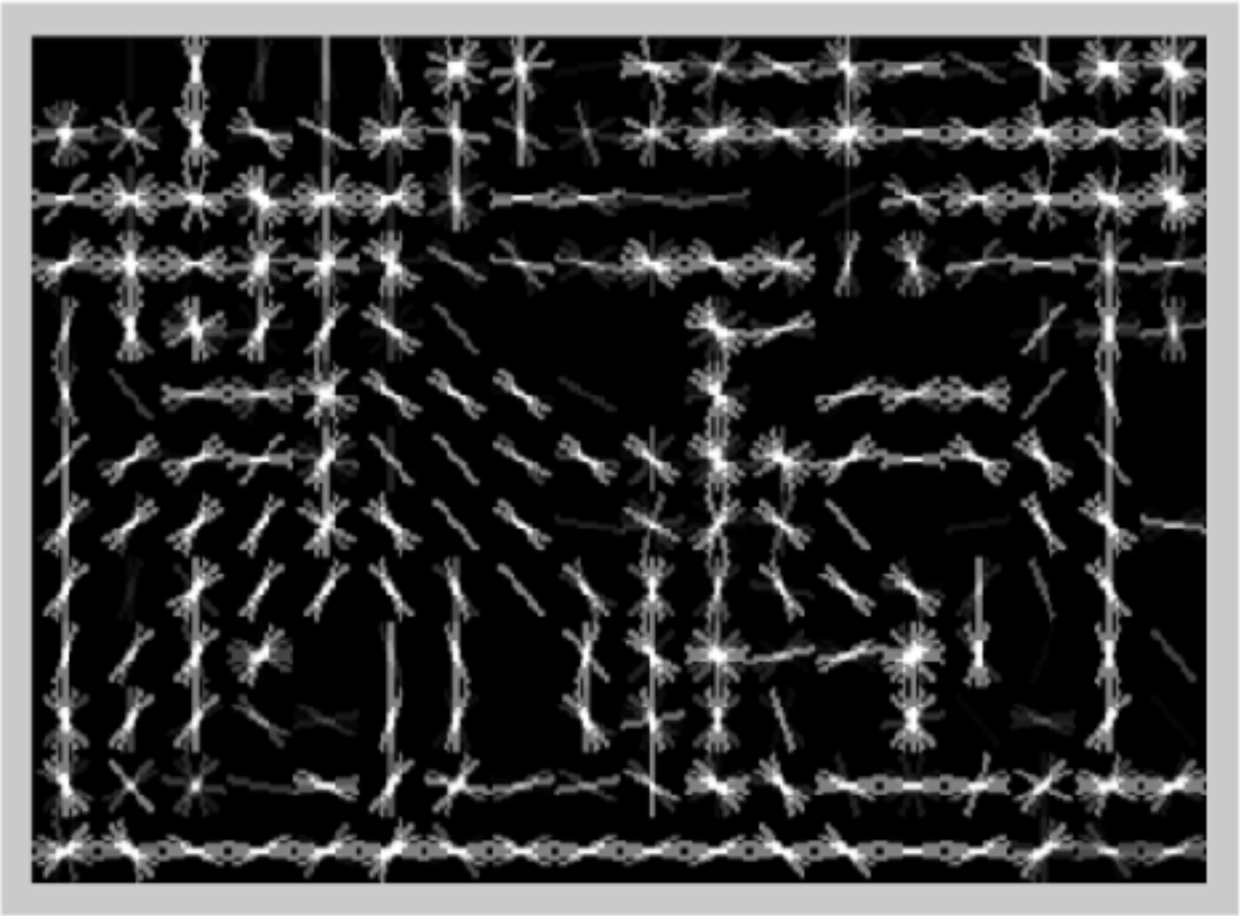
Summary: Histogram of Oriented Gradients

Introduced by Dalal and Triggs (CVPR 2005)

An extension of the SIFT feature

HOG properties:

- Preserves the overall structure of the image
- Provides robustness to illumination and small deformations



HOG feature

We will discuss …

Two popular image features

- Histogram of Oriented Gradients (HOG)
- $\boxed{\text{Bag of Visual Words (BoVW)}}$

Bag of visual words

Origin and motivation of the “bag of words” model

Algorithm pipeline

- Extracting local features
- Learning a dictionary — clustering using k-means
- Encoding methods — hard vs. soft assignment
- Spatial pooling — pyramid representations
- Similarity functions and classifiers

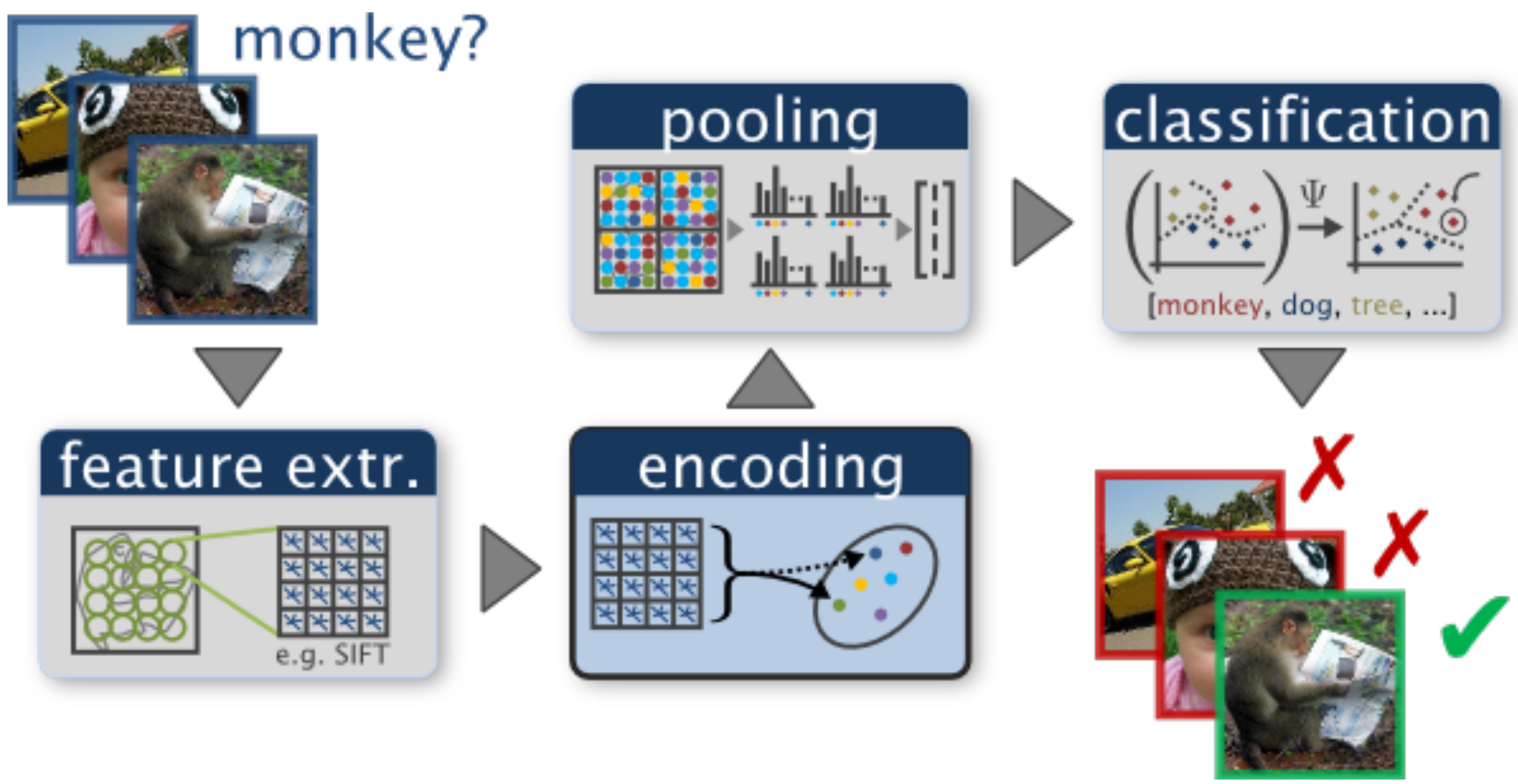


Figure from *Chatfield et al.,2011*

Bag of features

Properties:

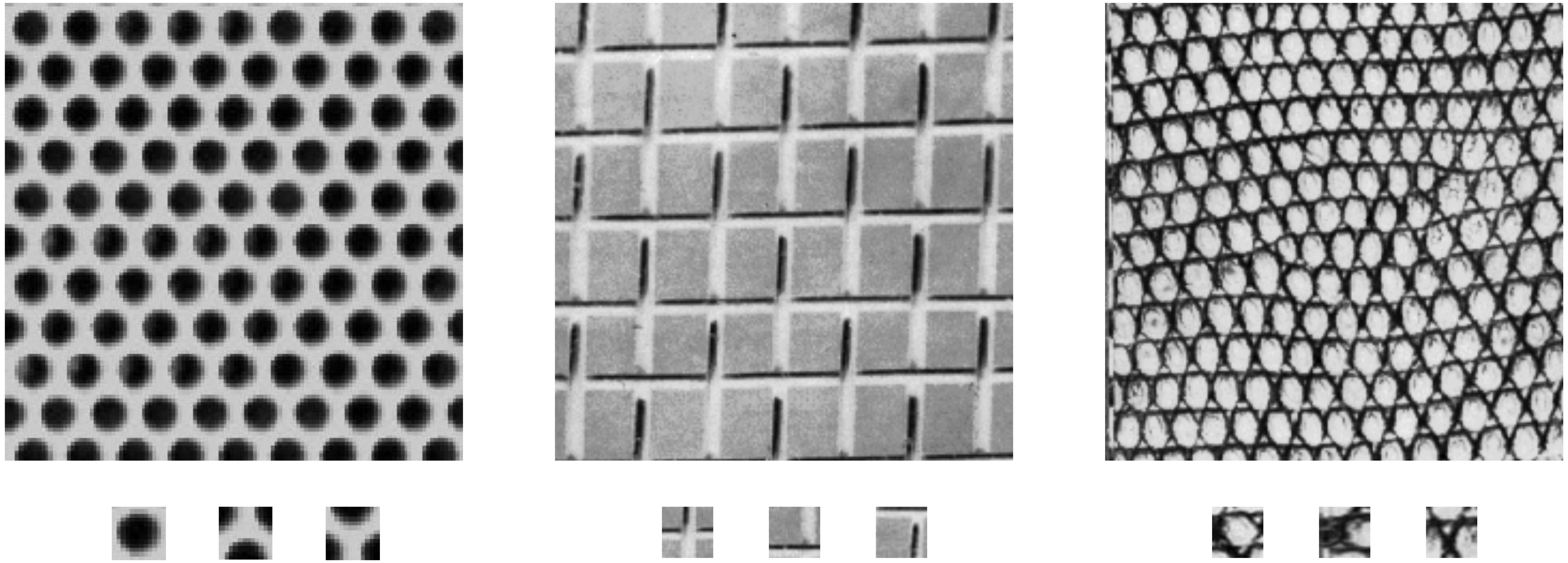
- Spatial structure is not preserved
- Invariance to large translations

Compare this to the HOG feature

Origin 1: Texture recognition

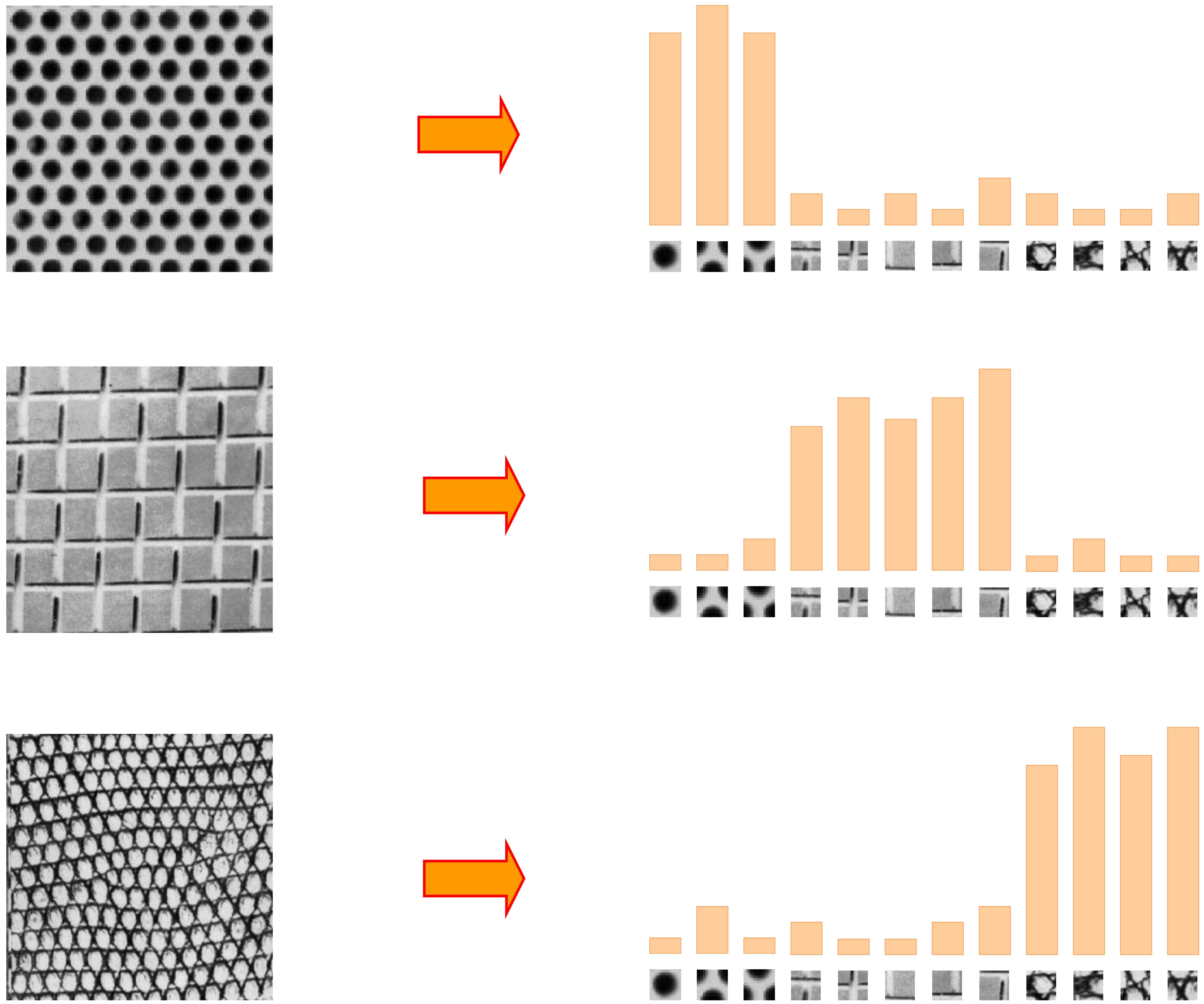
Texture is characterized by the repetition of basic elements or *textons*

For stochastic textures, it is the identity of the textons, not their spatial arrangement, that matters



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

Origin 1: Texture recognition



Origin 2: Bag-of-words models

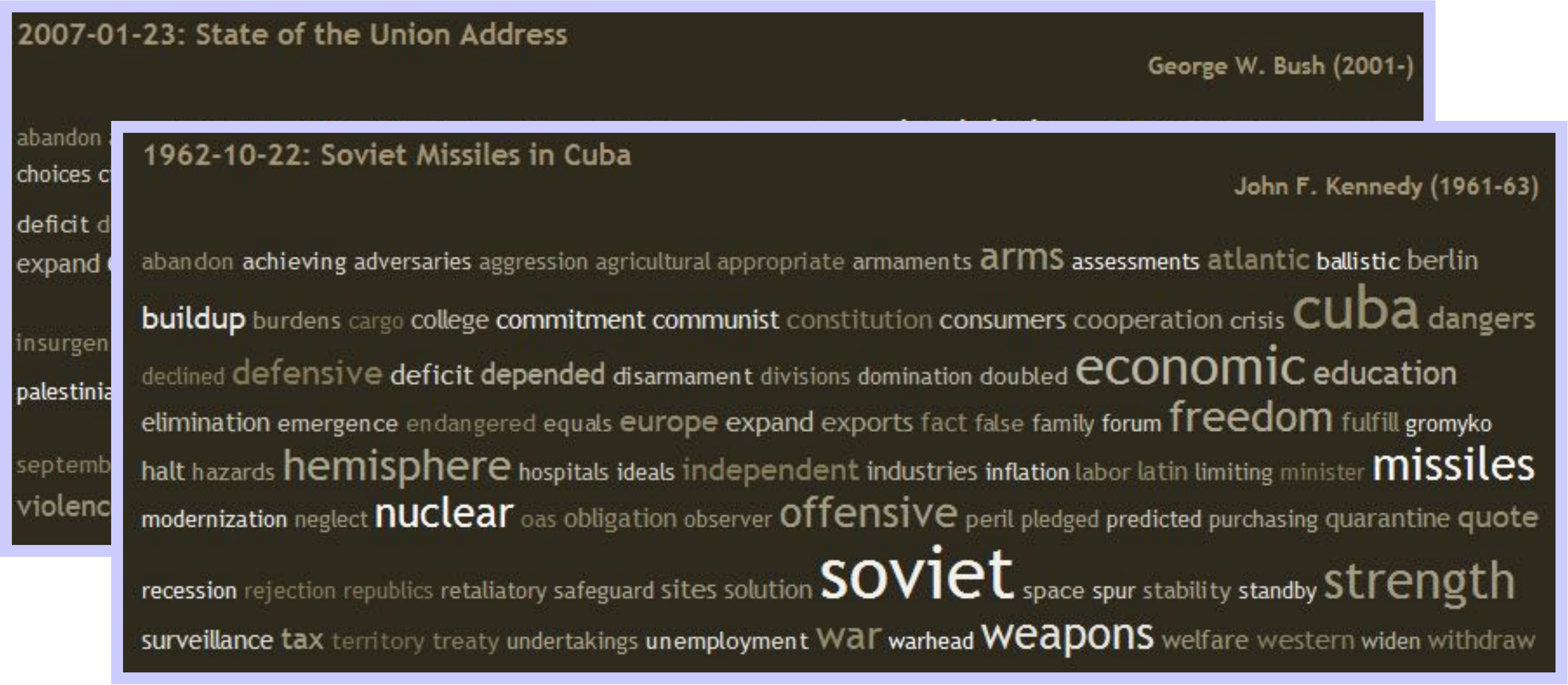
Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)

Origin 2: Bag-of-words models

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

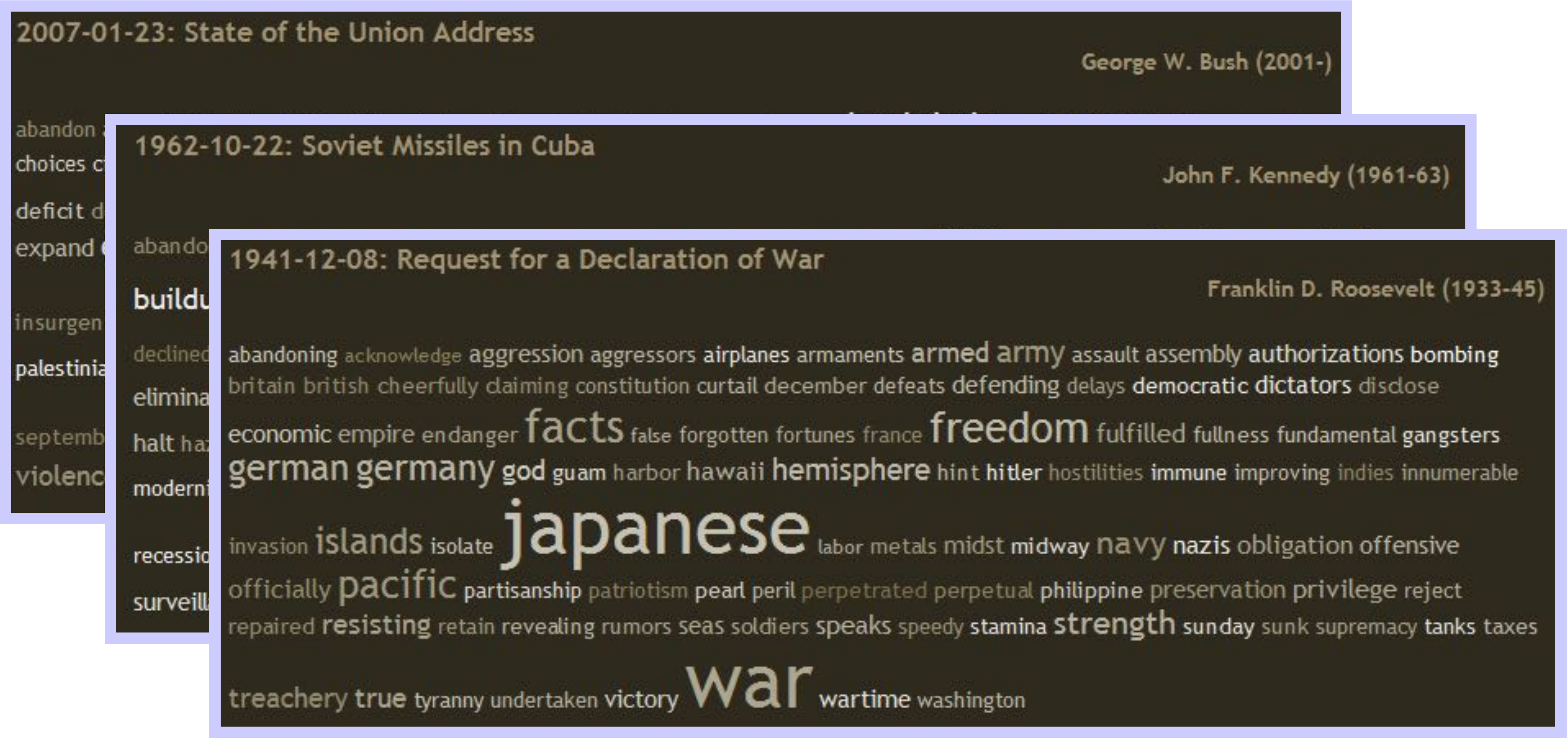
Origin 2: Bag-of-words models

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



Origin 2: Bag-of-words models

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



Lecture outline

Origin and motivation of the “bag of words” model

Algorithm pipeline

- Extracting local features
- Learning a dictionary — clustering using k-means
- Encoding methods — hard vs. soft assignment
- Spatial pooling — pyramid representations
- Similarity functions and classifiers

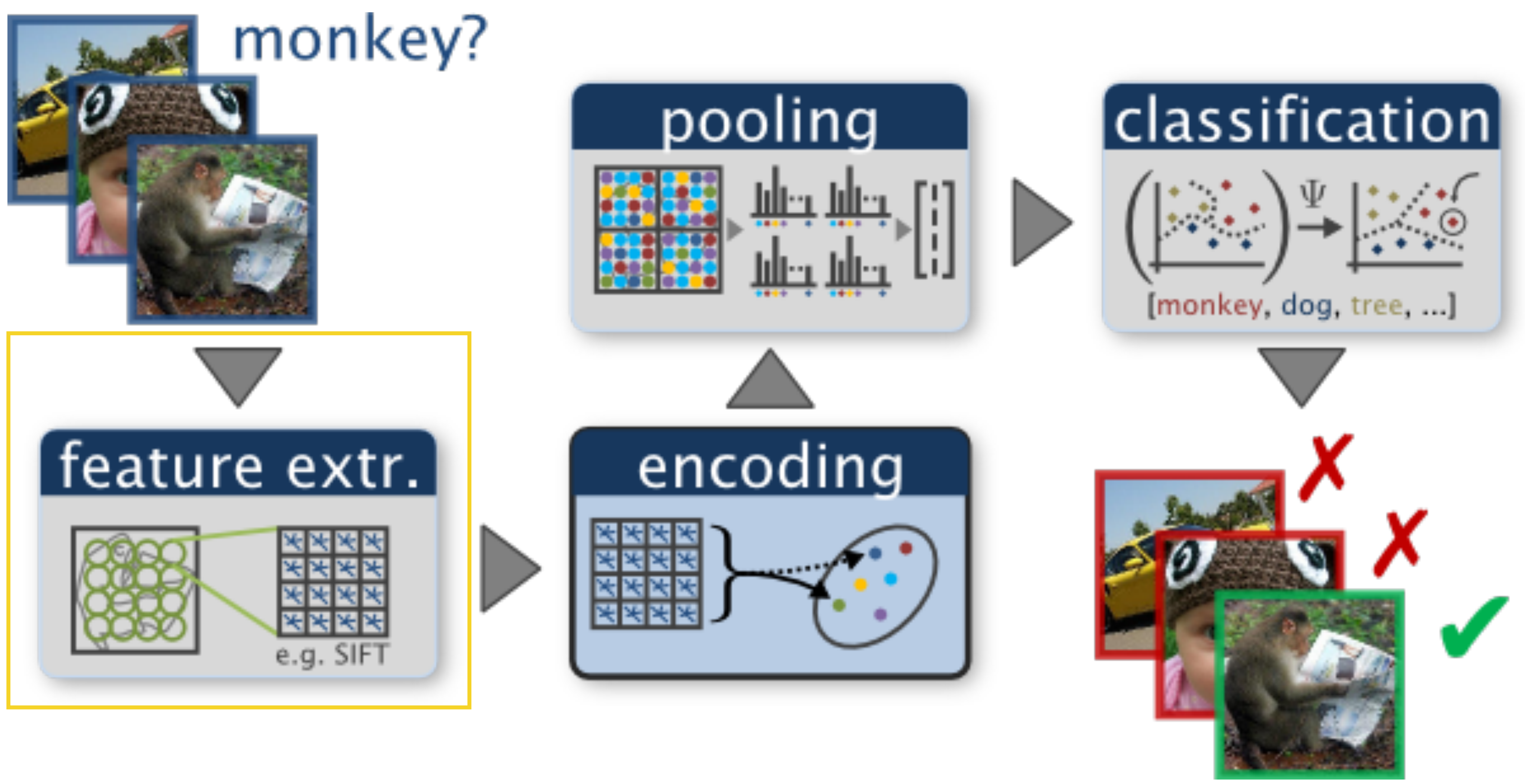


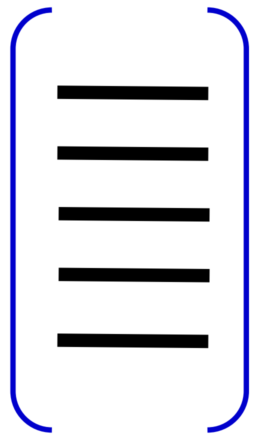
Figure from *Chatfield et al.,2011*

Local feature extraction

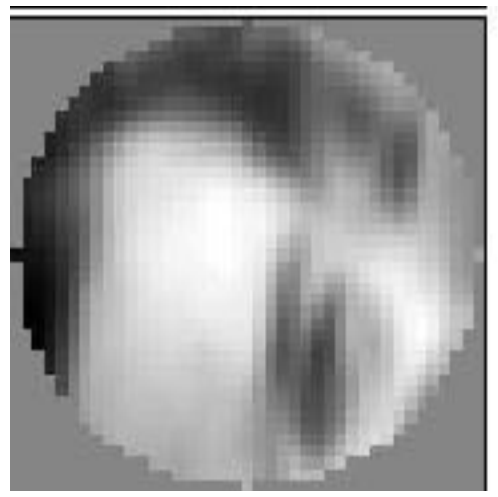
Regular grid or interest regions

corner detector

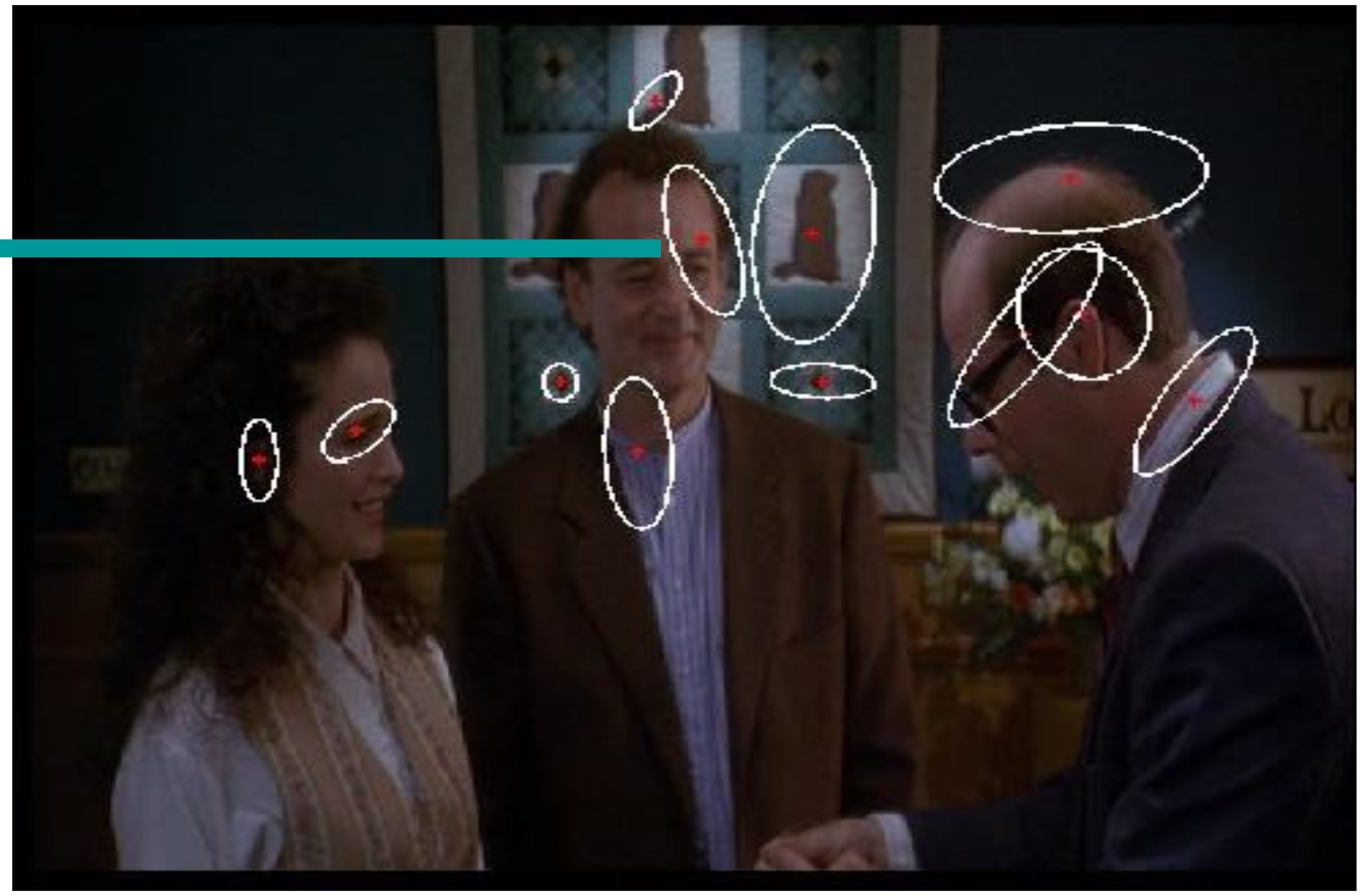
Local feature extraction



Compute descriptor



Normalize patch



Detect patches

Choices of descriptor:

- Pixels
- SIFT
- Shape context
- Geometric blur
- …

Slide credit: Josef Sivic

Lecture outline

Origin and motivation of the “bag of words” model

Algorithm pipeline

- Extracting local features
- Learning a dictionary — clustering using k-means
- Encoding methods — hard vs. soft assignment
- Spatial pooling — pyramid representations
- Similarity functions and classifiers

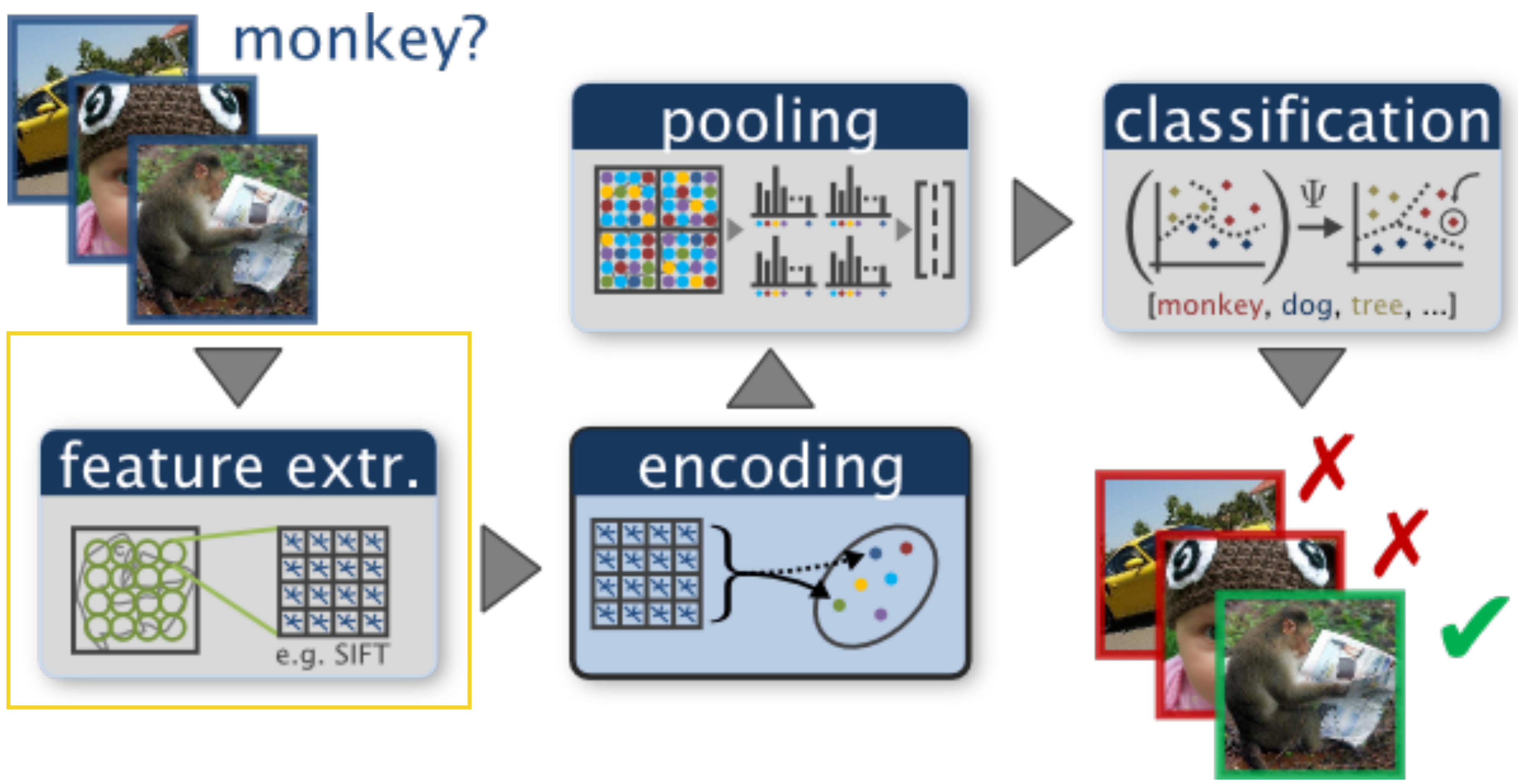
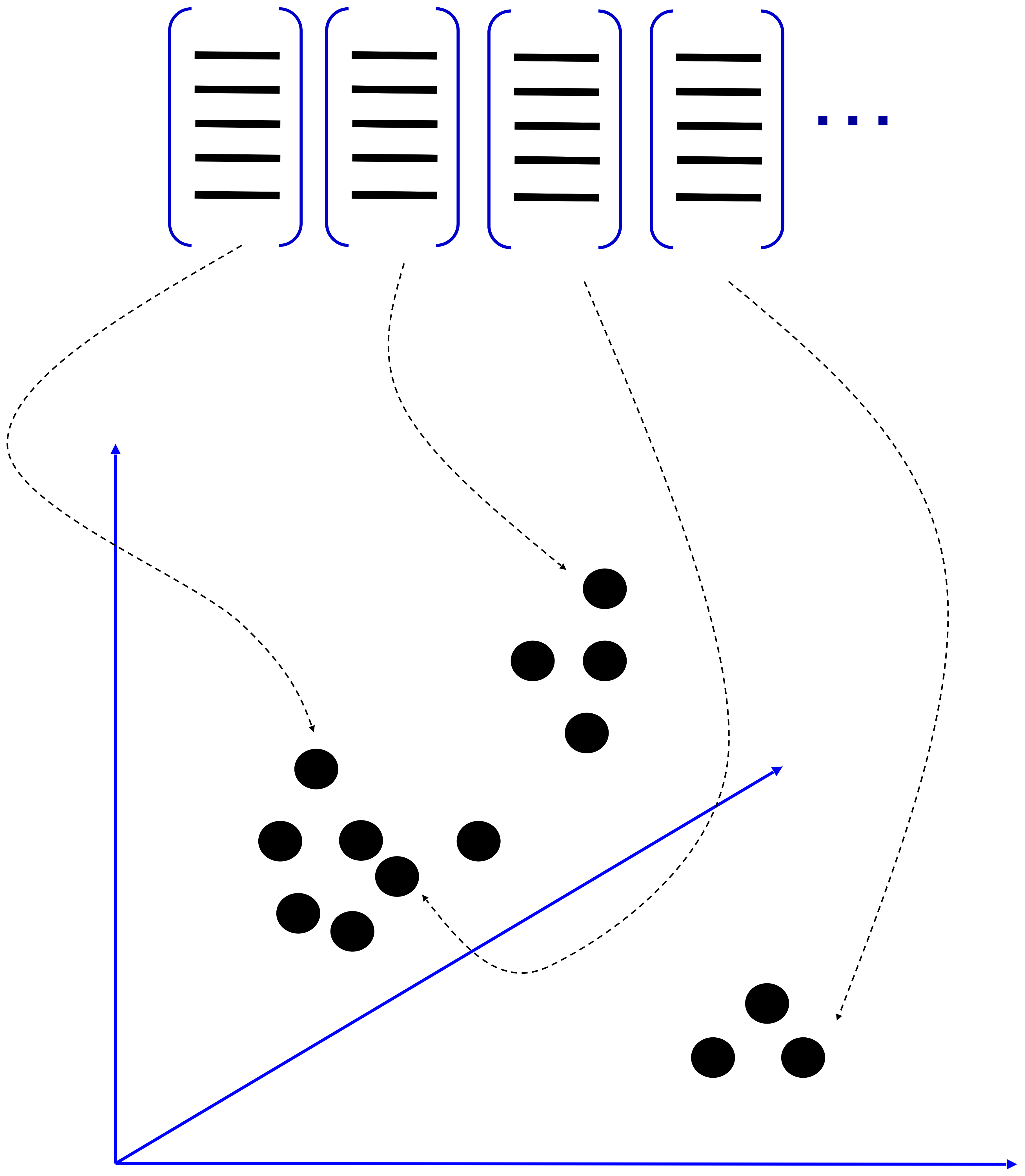


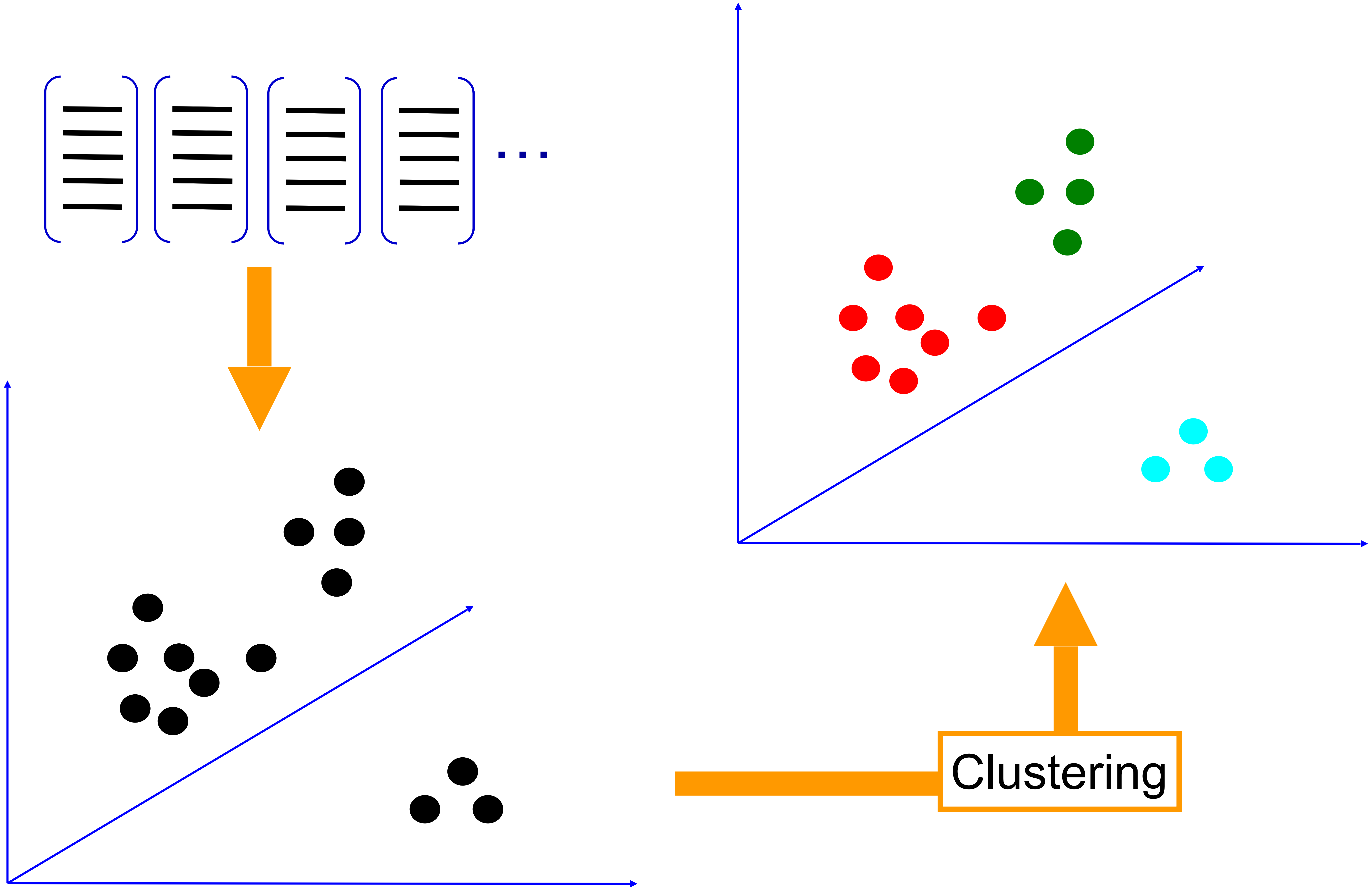
Figure from *Chatfield et al.,2011*

Learning a dictionary



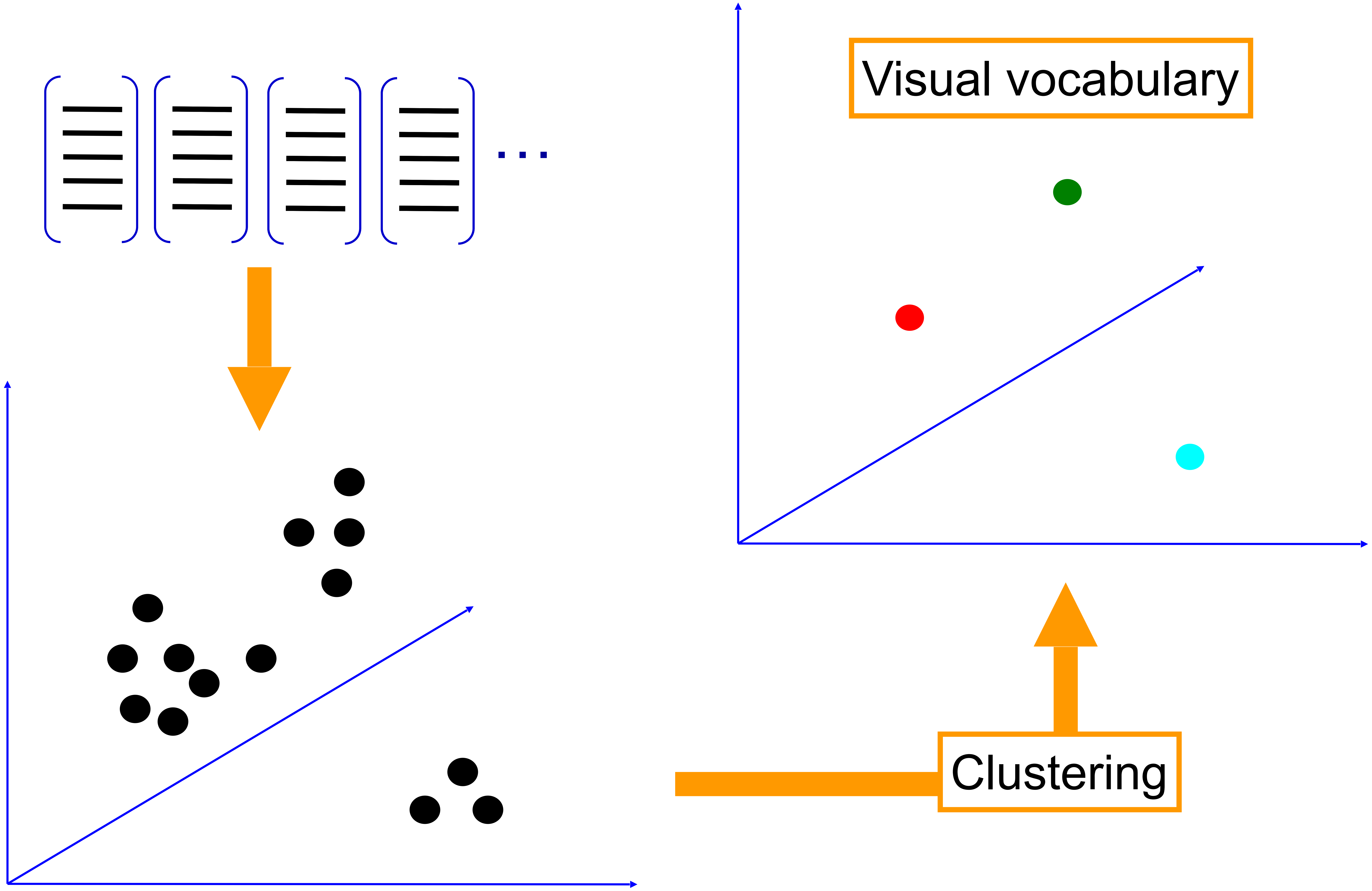
Slide credit: Josef Sivic

Learning a dictionary



Slide credit: Josef Sivic

Learning a dictionary



Slide credit: Josef Sivic

Lloyd's algorithm for k-means

Initialize k centers by picking k points randomly among all the points

Repeat till convergence (or max iterations)

- Assign each point to the nearest center (assignment step)

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \underline{\sum_{\mathbf{x}\in S_i}} ||\mathbf{x} - \mu_i||^2$$

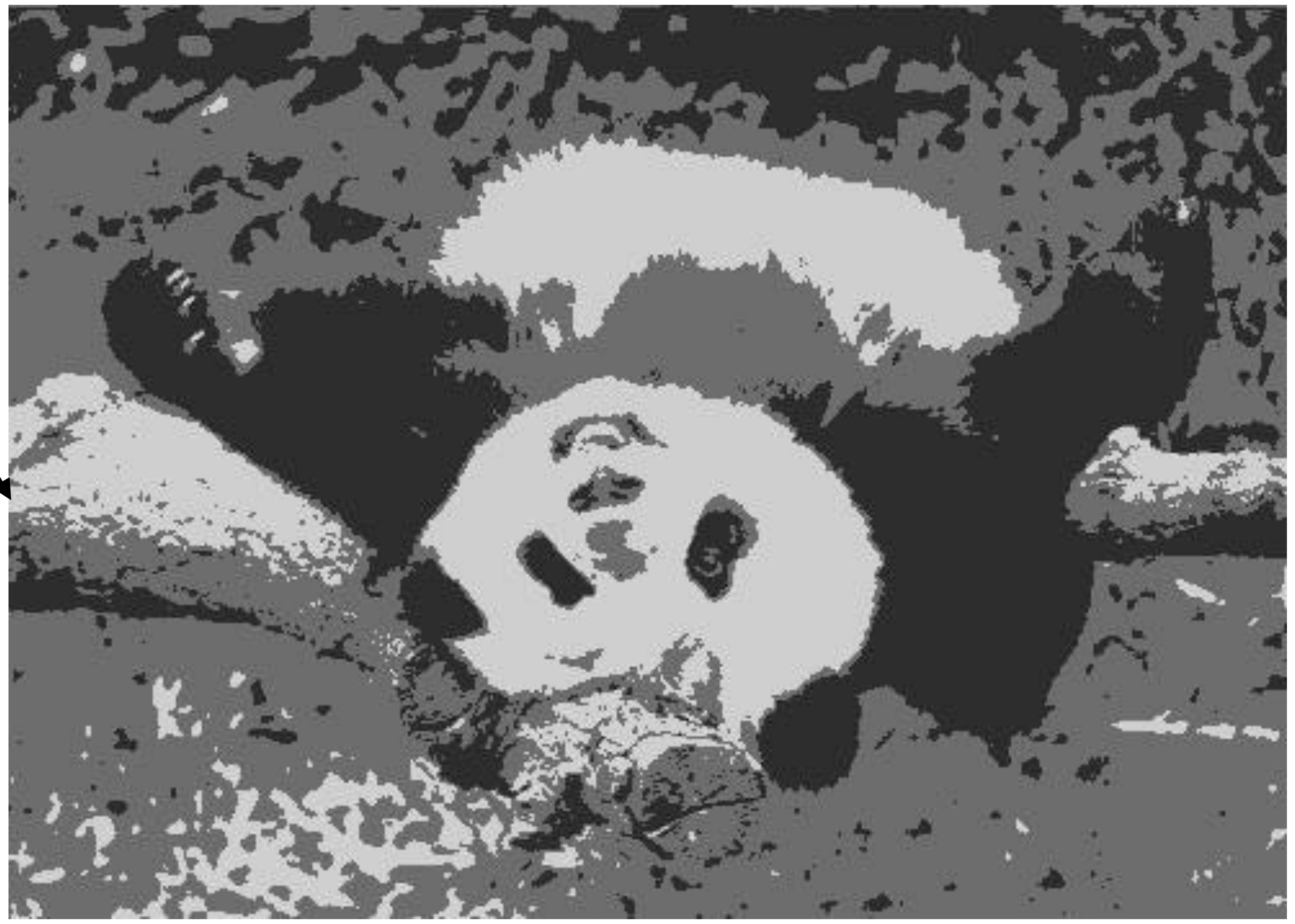
- Estimate the mean of each group (update step)

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}\in S_i} \underline{||\mathbf{x} - \mu_i||^2}$$

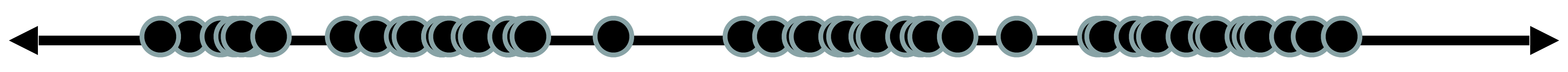
k-means for image segmentation

K=2

K=3

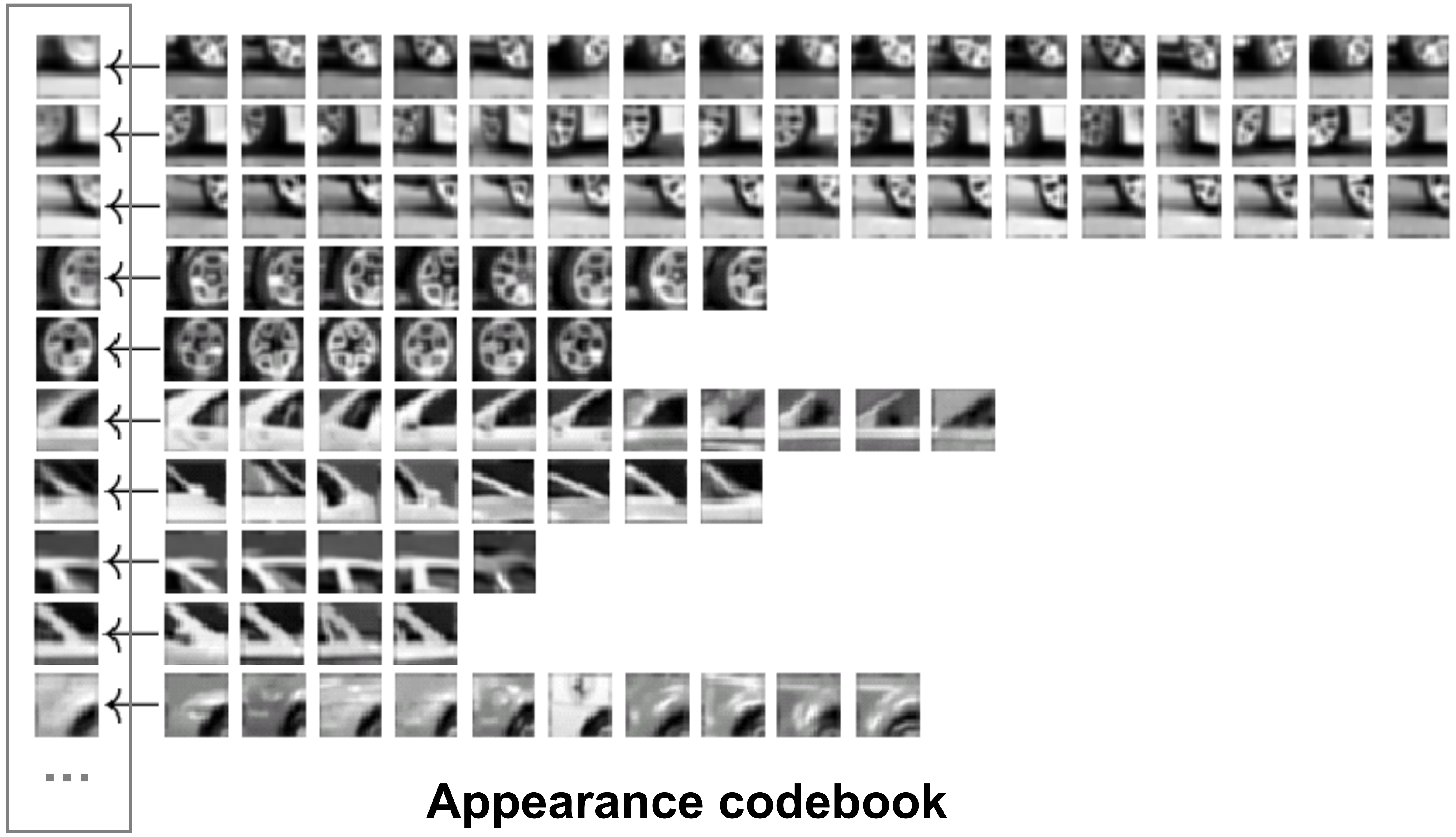


Grouping pixels based on **intensity** similarity



feature space: intensity value (1D)

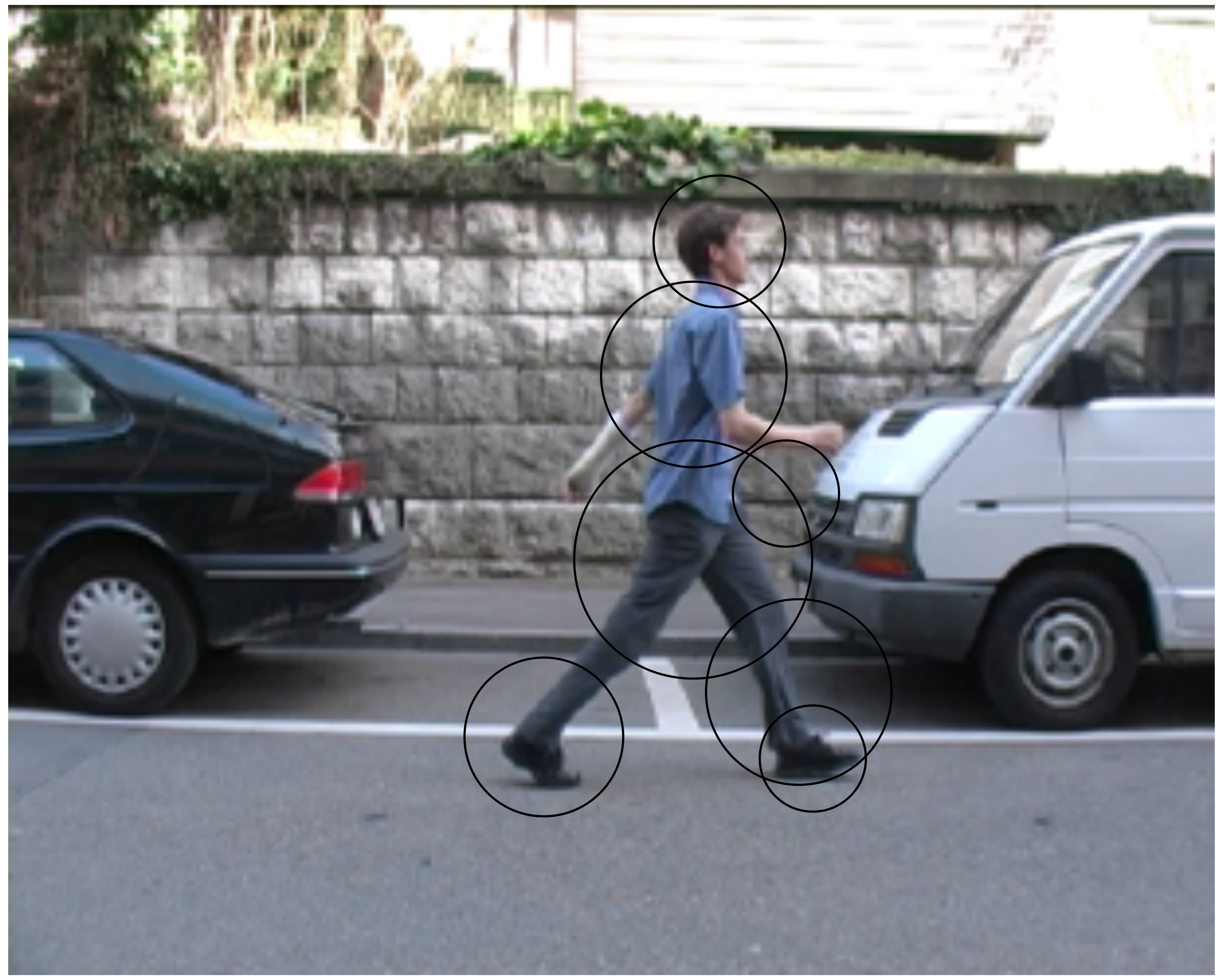
Example codebook

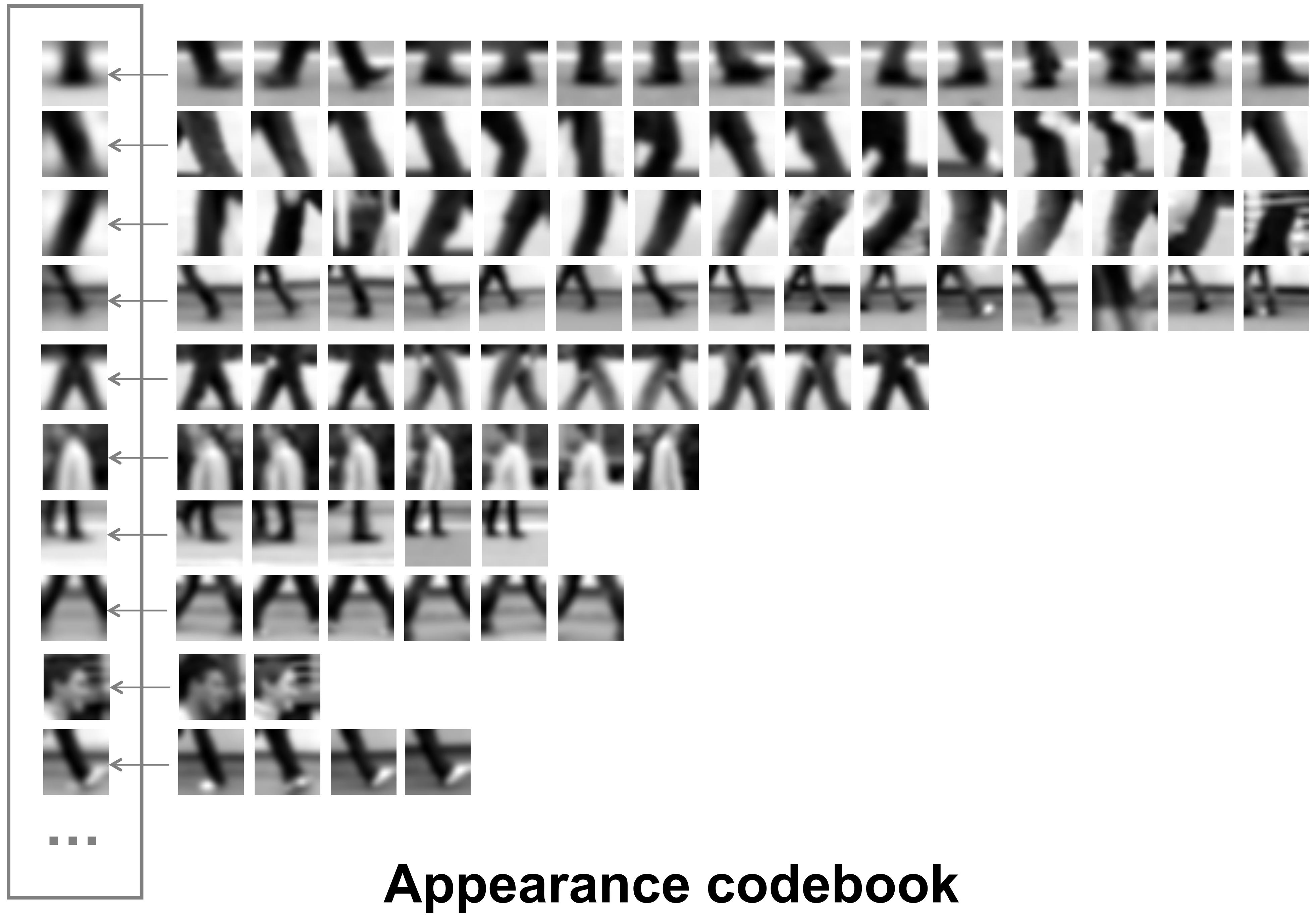


Appearance codebook

Source: B. Leibe

Another codebook





Appearance codebook

Source: B. Leibe

Lecture outline

Origin and motivation of the "bag of words" model

Algorithm pipeline

- Extracting local features
- Learning a dictionary — clustering using k-means
- Encoding methods — hard vs. soft assignment
- Spatial pooling — pyramid representations
- Similarity functions and classifiers

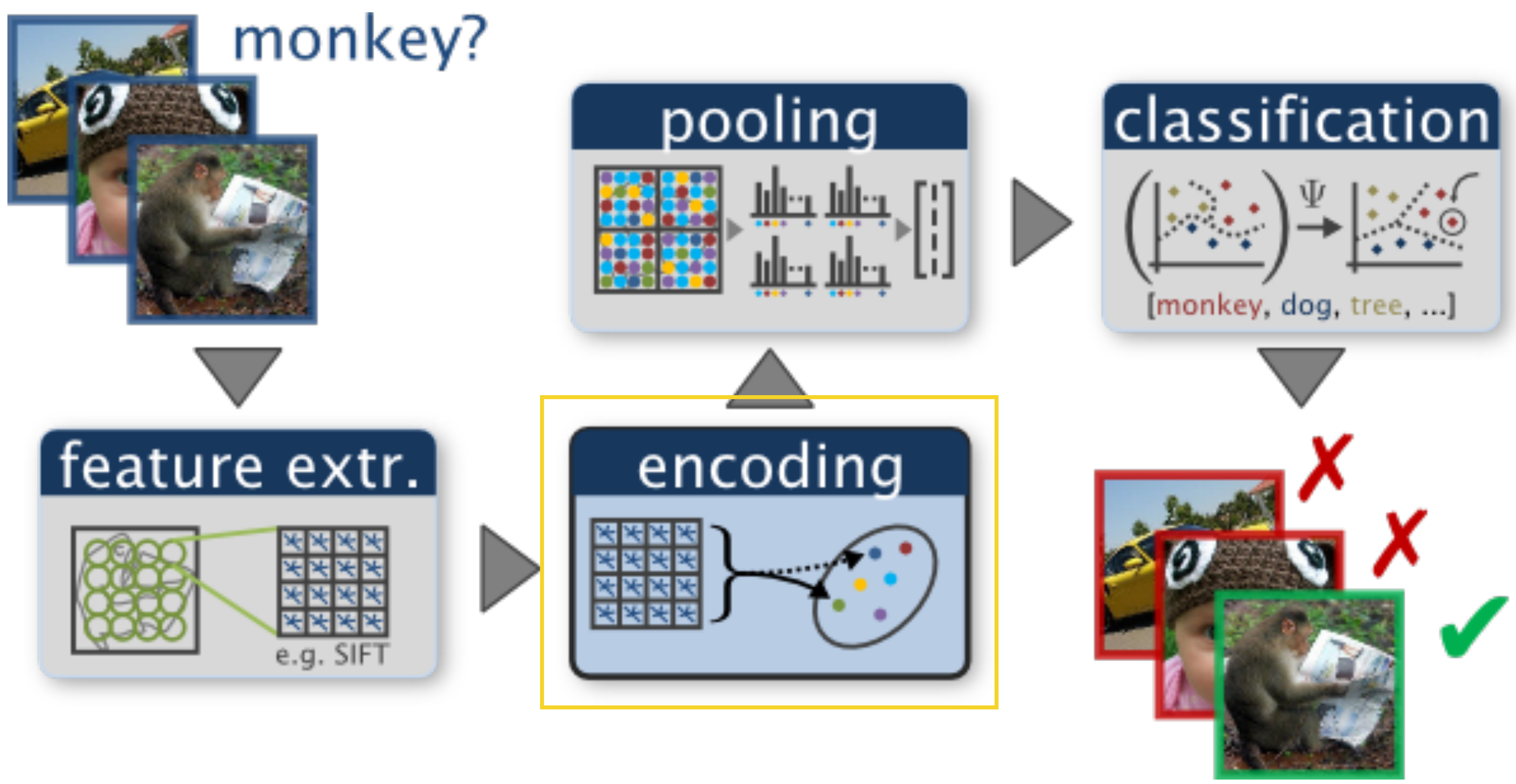
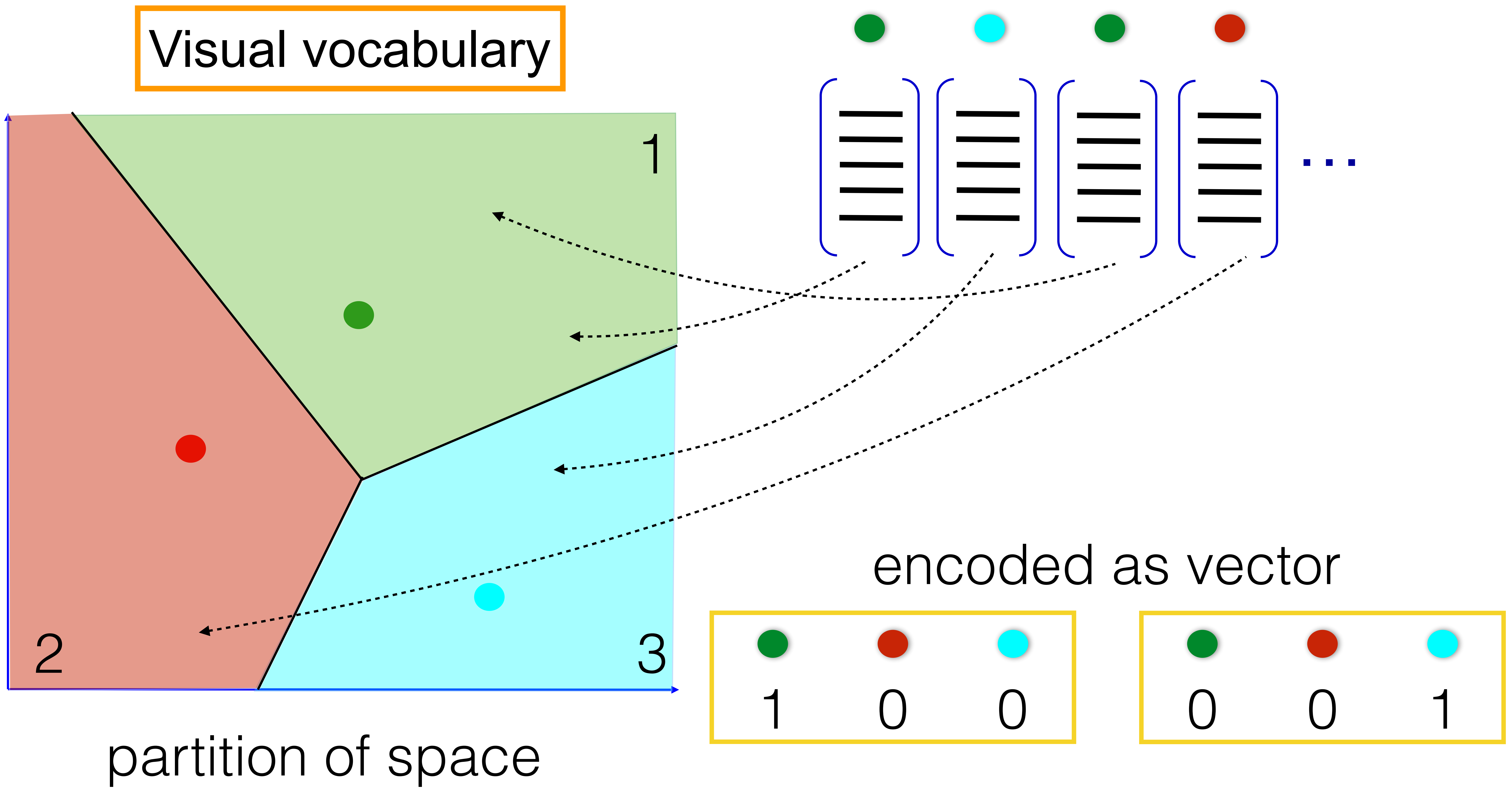


Figure from *Chatfield et al.,2011*

Encoding methods

Assigning words to features

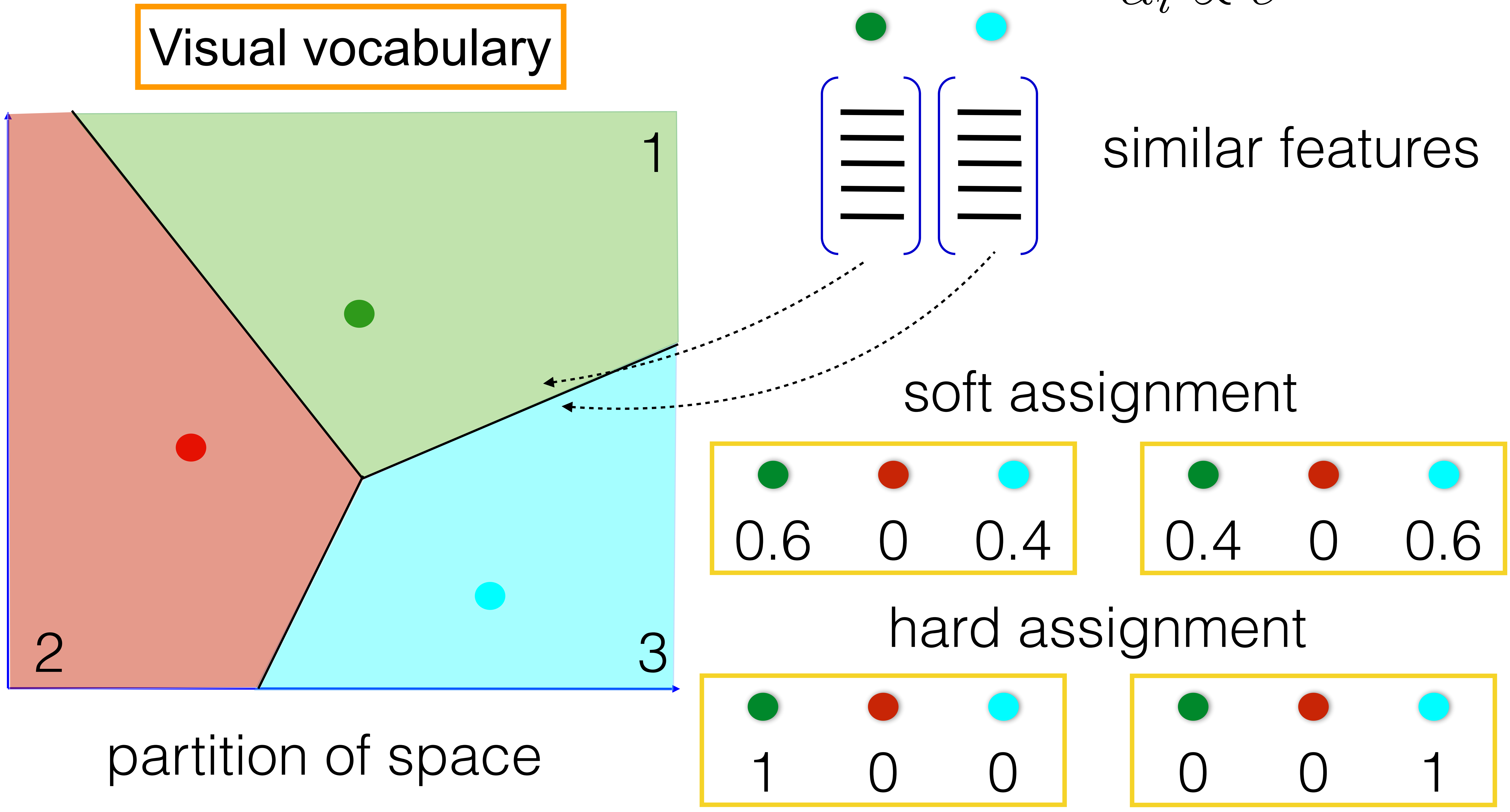


Encoding methods

Assigning words to features

soft assignment

$$\alpha_i \propto e^{-f(d(\mathbf{x},\mathbf{c_i}))}$$



Encoding considerations

What should be the size of the dictionary?

- Too small: don't capture the variability of the dataset
- Too large: have too few points per cluster

Speed of embedding

- Exact nearest neighbor is slow if the dictionary is large
- Approximate nearest neighbor techniques
  - Search trees — organize data in a tree
  - Hashing — create buckets in the feature space

Lecture outline

Origin and motivation of the "bag of words" model

Algorithm pipeline

- Extracting local features
- Learning a dictionary — clustering using k-means
- Encoding methods — hard vs. soft assignment
- Spatial pooling — pyramid representations
- Similarity functions and classifiers

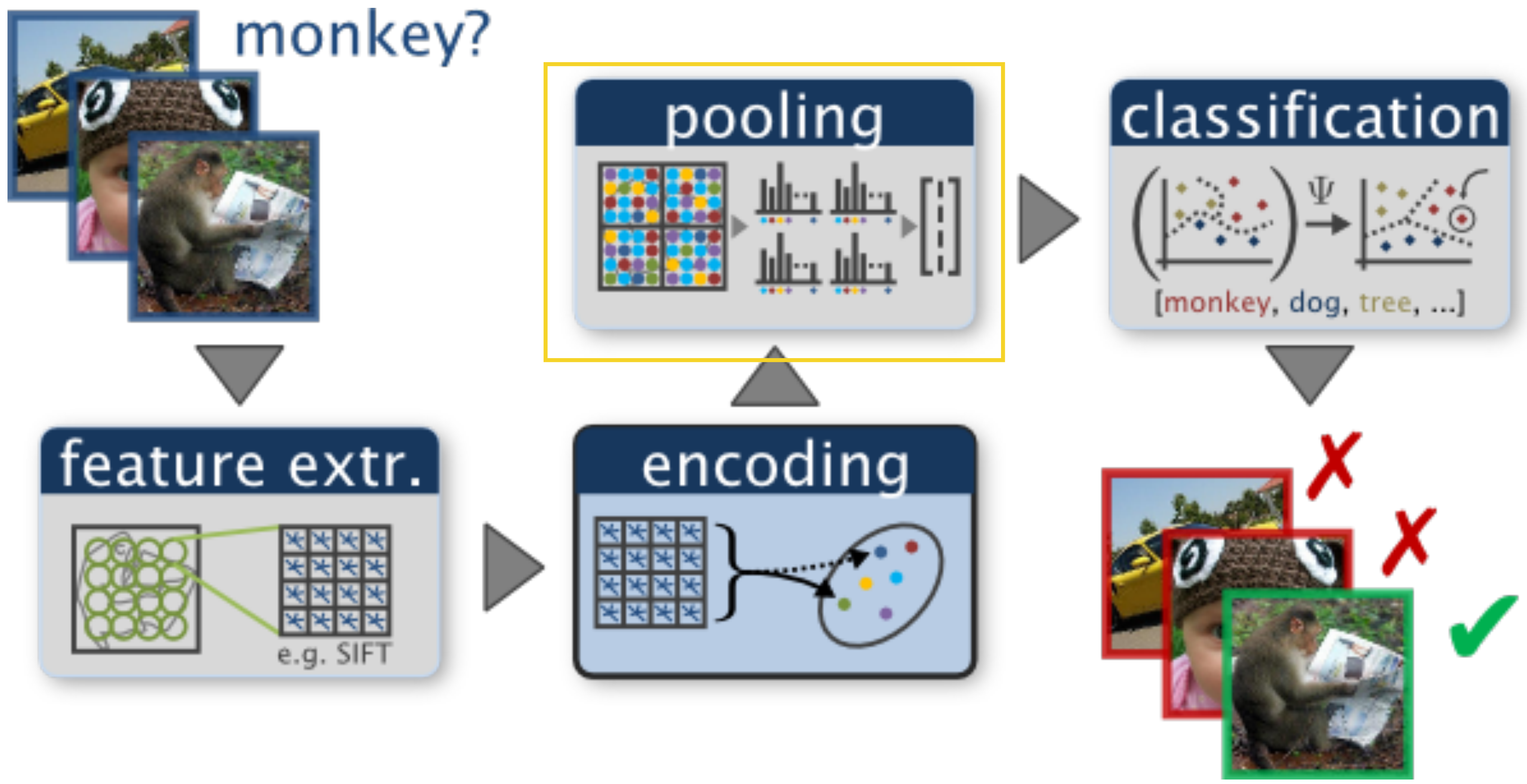
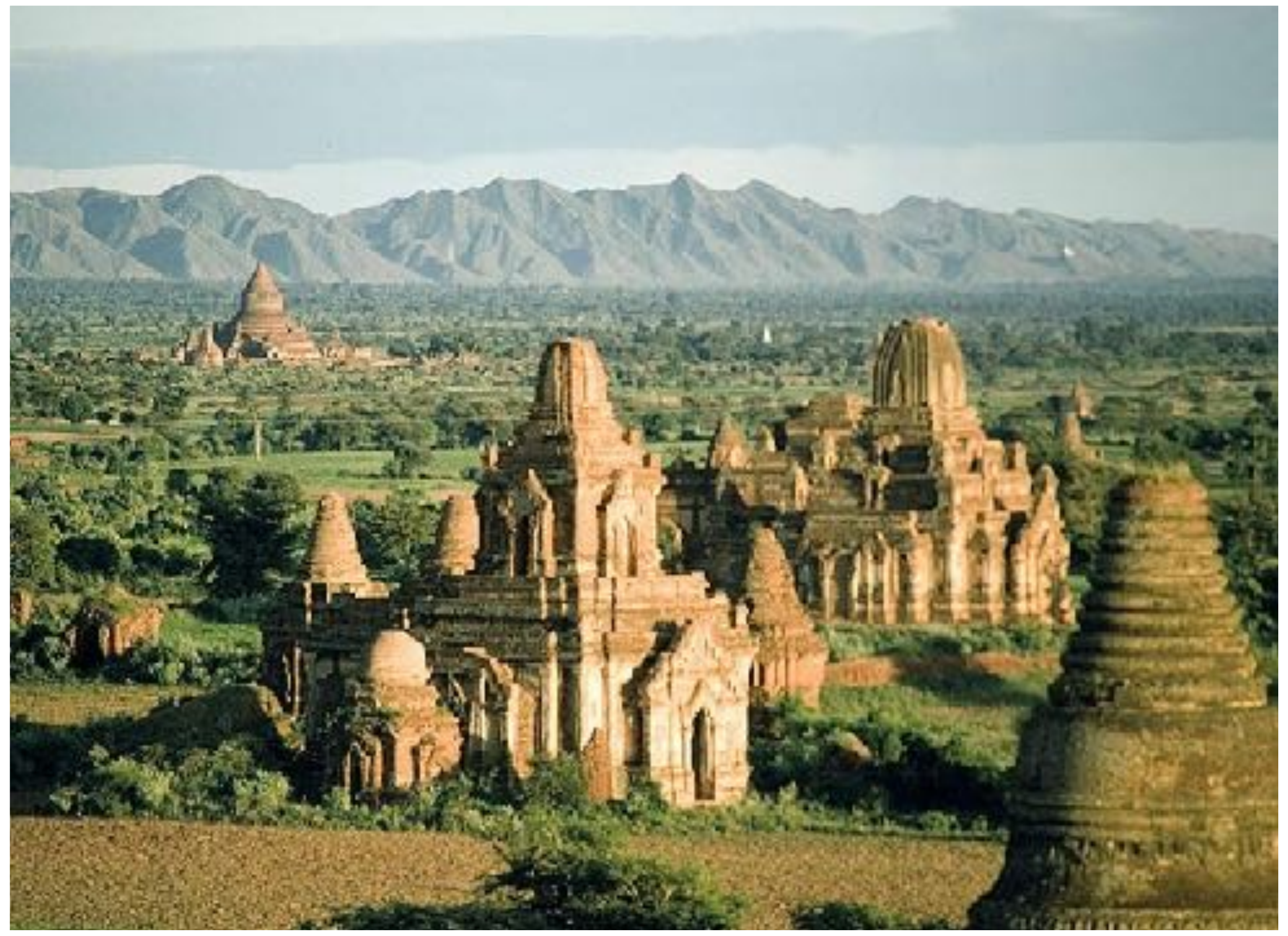
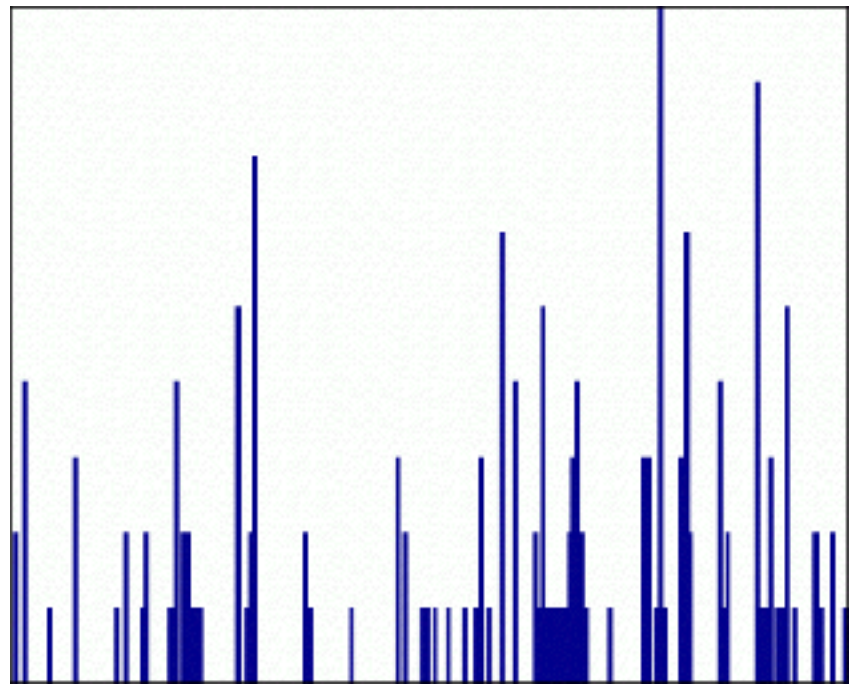


Figure from Chatfield et al, 2011

Spatial pyramids

pooling: aggregate features within a region

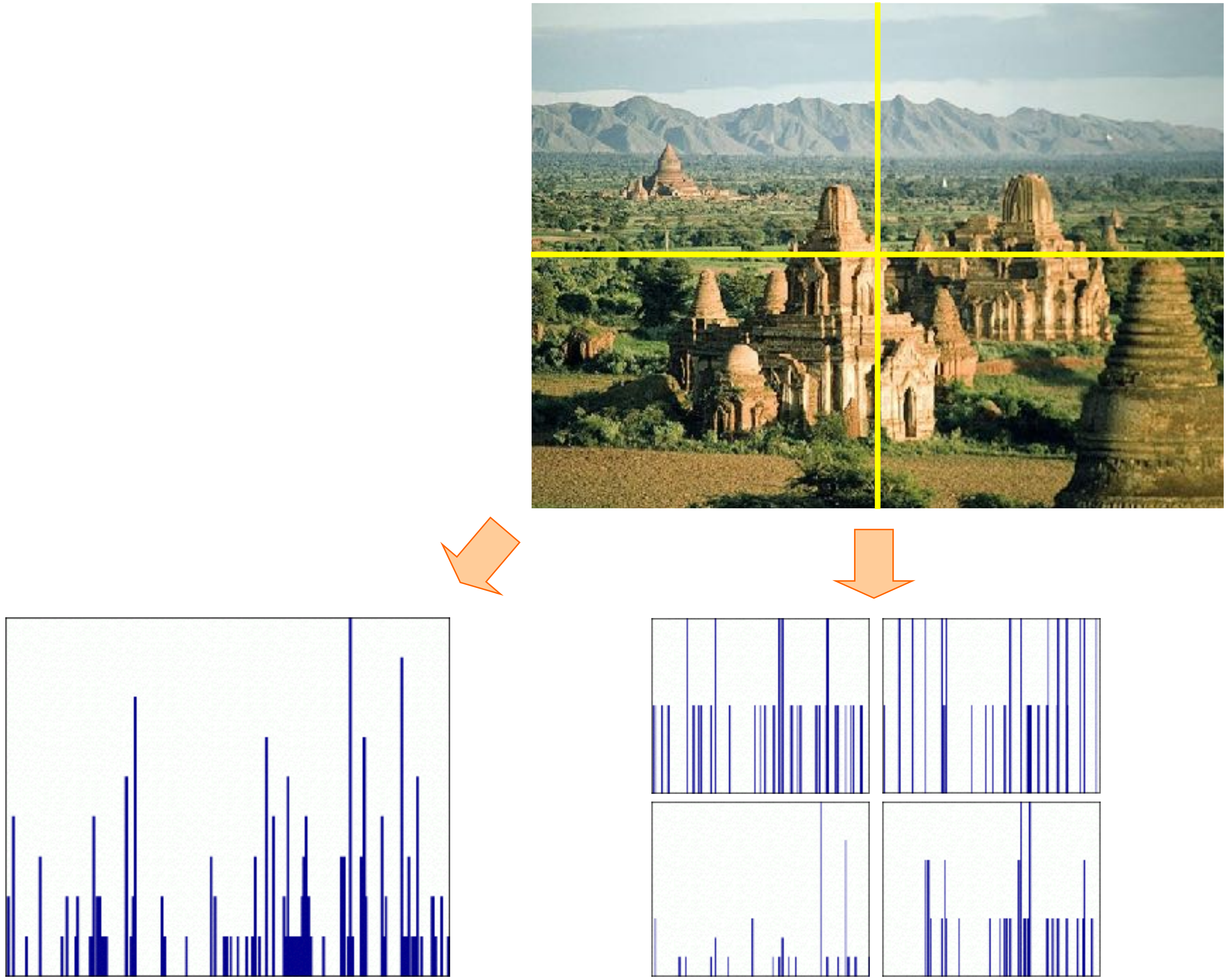




Lazebnik, Schmid & Ponce (CVPR 2006)

Spatial pyramids

pooling: aggregate features within a region



Same motivation as **SIFT** — keep coarse layout information

Lazebnik, Schmid & Ponce (CVPR 2006)

Spatial pyramids

pooling: aggregate features within a region

Same motivation as **SIFT** — keep coarse layout information

Lazebnik, Schmid & Ponce (CVPR 2006)

Lecture outline

Origin and motivation of the “bag of words” model

Algorithm pipeline

- Extracting local features
- Learning a dictionary — clustering using k-means
- Encoding methods — hard vs. soft assignment
- Spatial pooling — pyramid representations
- Similarity functions and classifiers

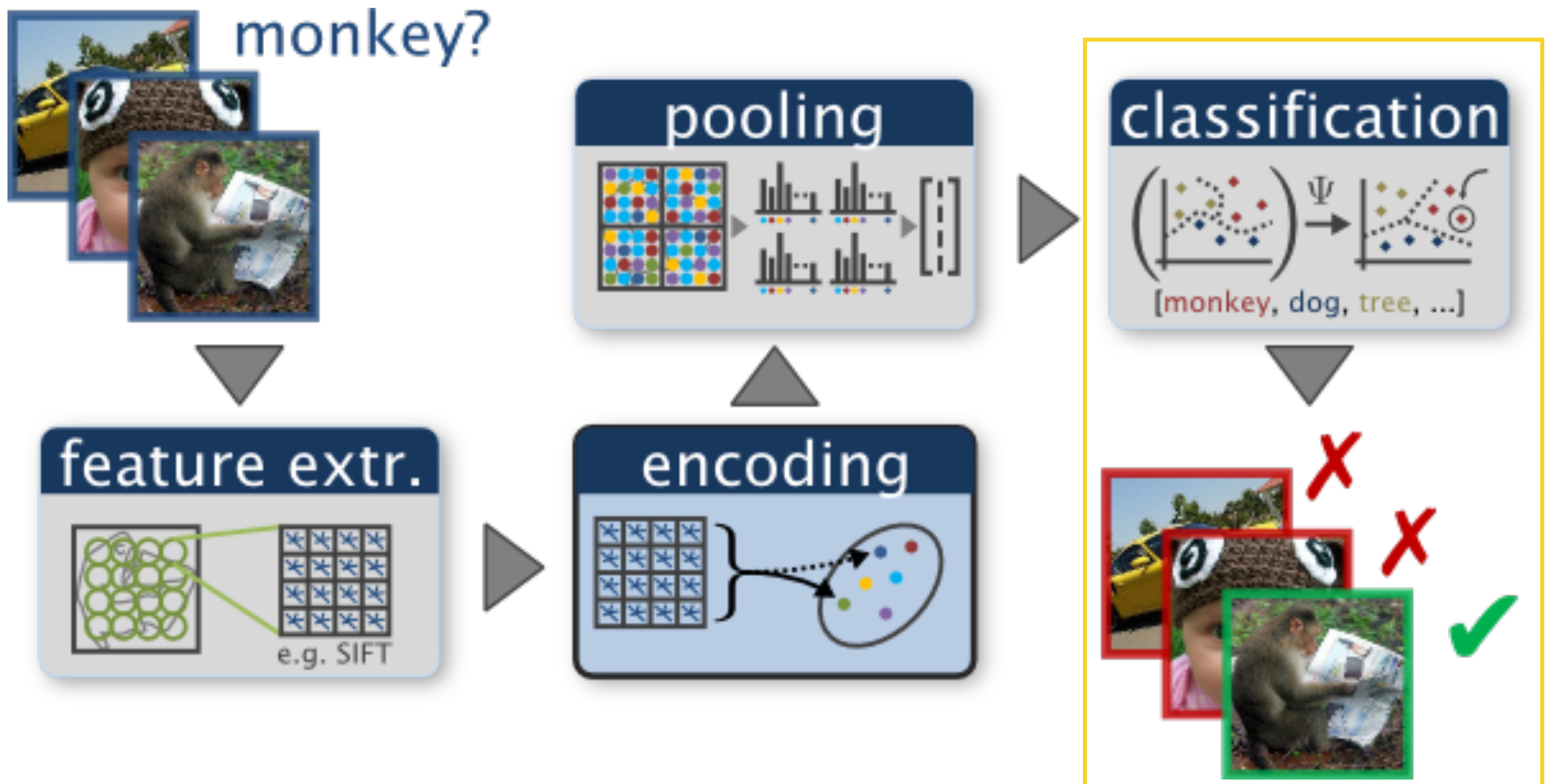


Figure from *Chatfield et al.,2011*

Bags of features representation

I

$\mathbf{h} = \Phi(I)$

image similarity = feature similarity

Similarity functions and classifiers

Euclidean distance:

$$D(\mathbf{h}_1, \mathbf{h}_2) = \sqrt{\sum_{i=1}^{N} (\mathbf{h}_1(i) - \mathbf{h}_2(i))^2}$$

L1 distance:

$$D(\mathbf{h}_1, \mathbf{h}_2) = \sum_{i=1}^{N} |\, \mathbf{h}_1(i) - \mathbf{h}_2(i) \,|$$

Use k-NN classifiers with these distances, or linear classifiers

Lecture outline

Origin and motivation of the “bag of words” model

Algorithm pipeline

- Extracting local features
- Learning a dictionary — clustering using k-means
- Encoding methods — hard vs. soft assignment
- Spatial pooling — pyramid representations
- Similarity functions and classifiers

Putting it all together

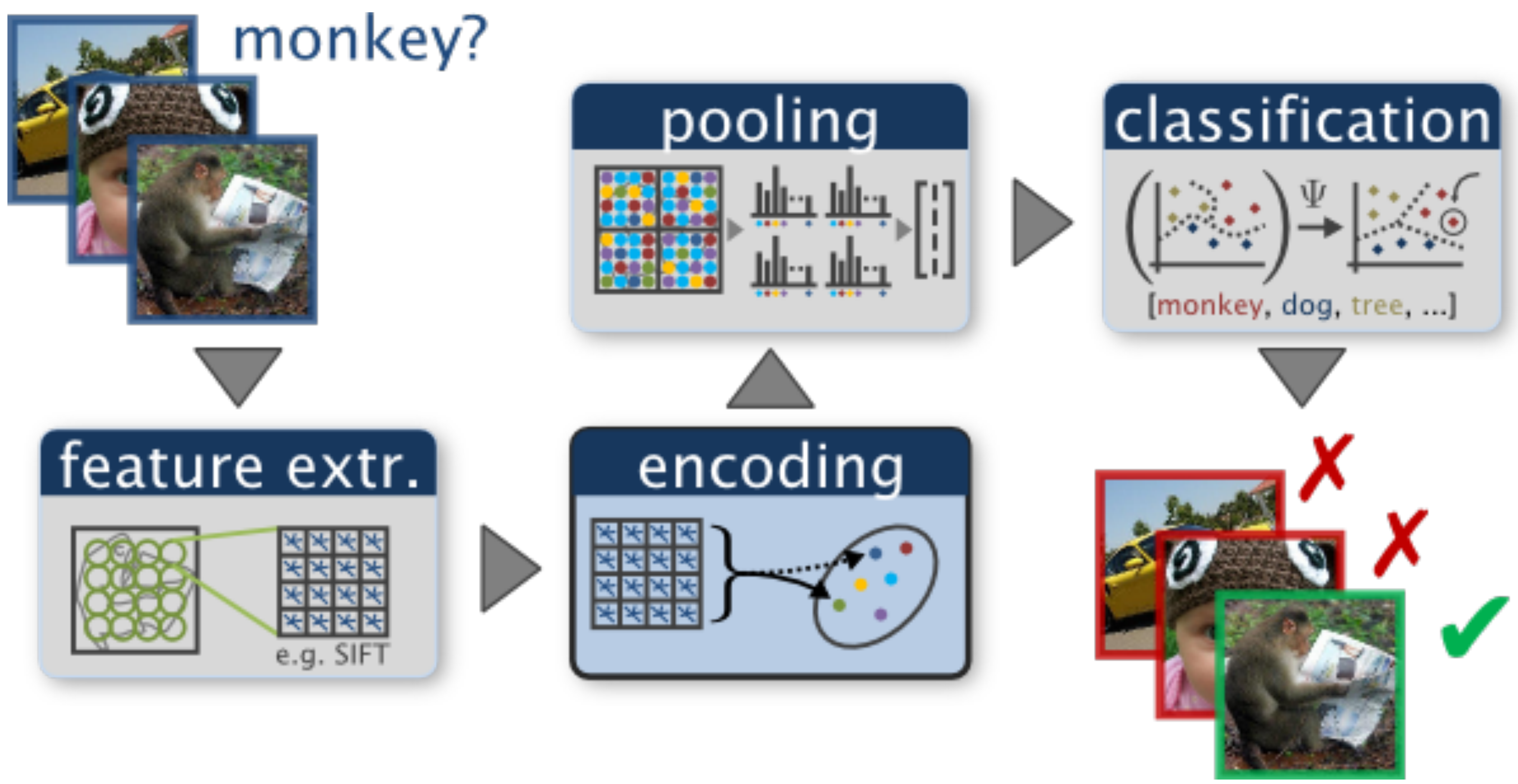
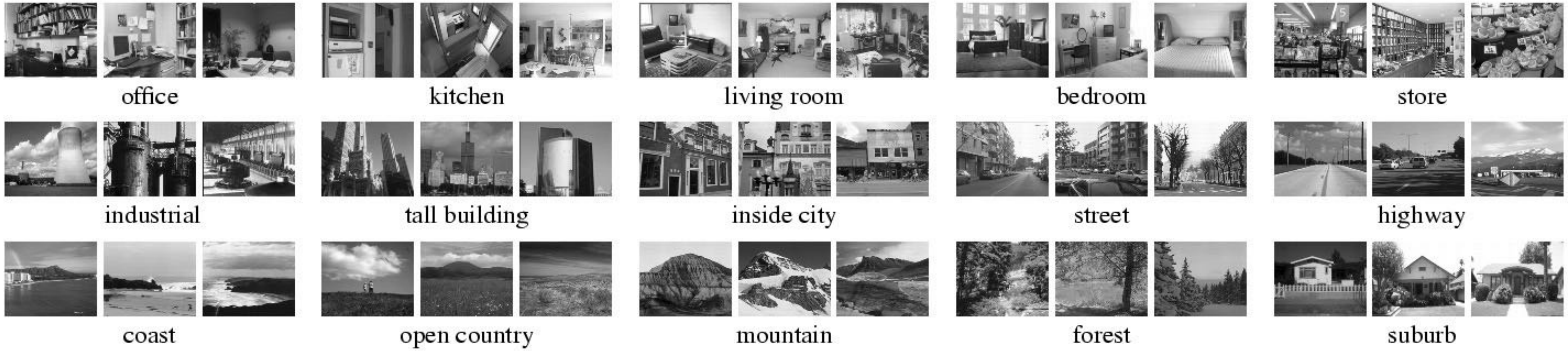


Figure from Chatfield et al. 2011

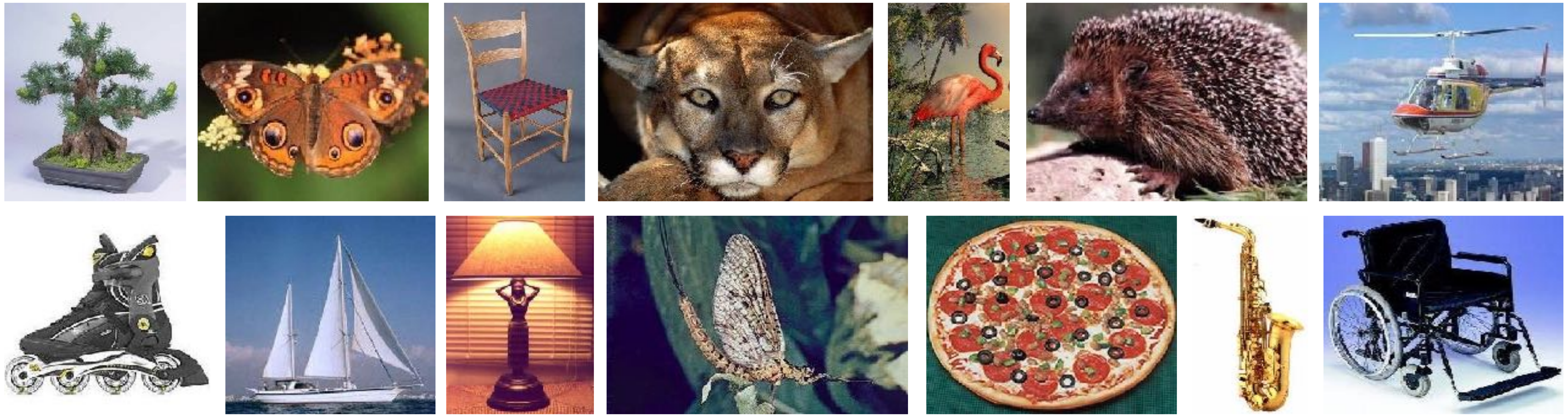
Results: scene category dataset



Multi-class classification results
(100 training images per class)

	Weak features (vocabulary size: 16)		Strong features (vocabulary size: 200)	
Level	Single-level	Pyramid	Single-level	Pyramid
0 (1×1)	45.3 $\pm$0.5		72.2 $\pm$0.6	
1 (2×2)	53.6 $\pm$0.3	56.2 $\pm$0.6	77.9 $\pm$0.6	79.0 $\pm$0.5
2 (4×4)	61.7 $\pm$0.6	64.7 $\pm$0.7	79.4 $\pm$0.3	**81.1** $\pm$0.3
3 (8×8)	63.3 $\pm$0.8	**66.8** $\pm$0.6	77.2 $\pm$0.4	80.7 $\pm$0.3

Results: Caltech-101 dataset

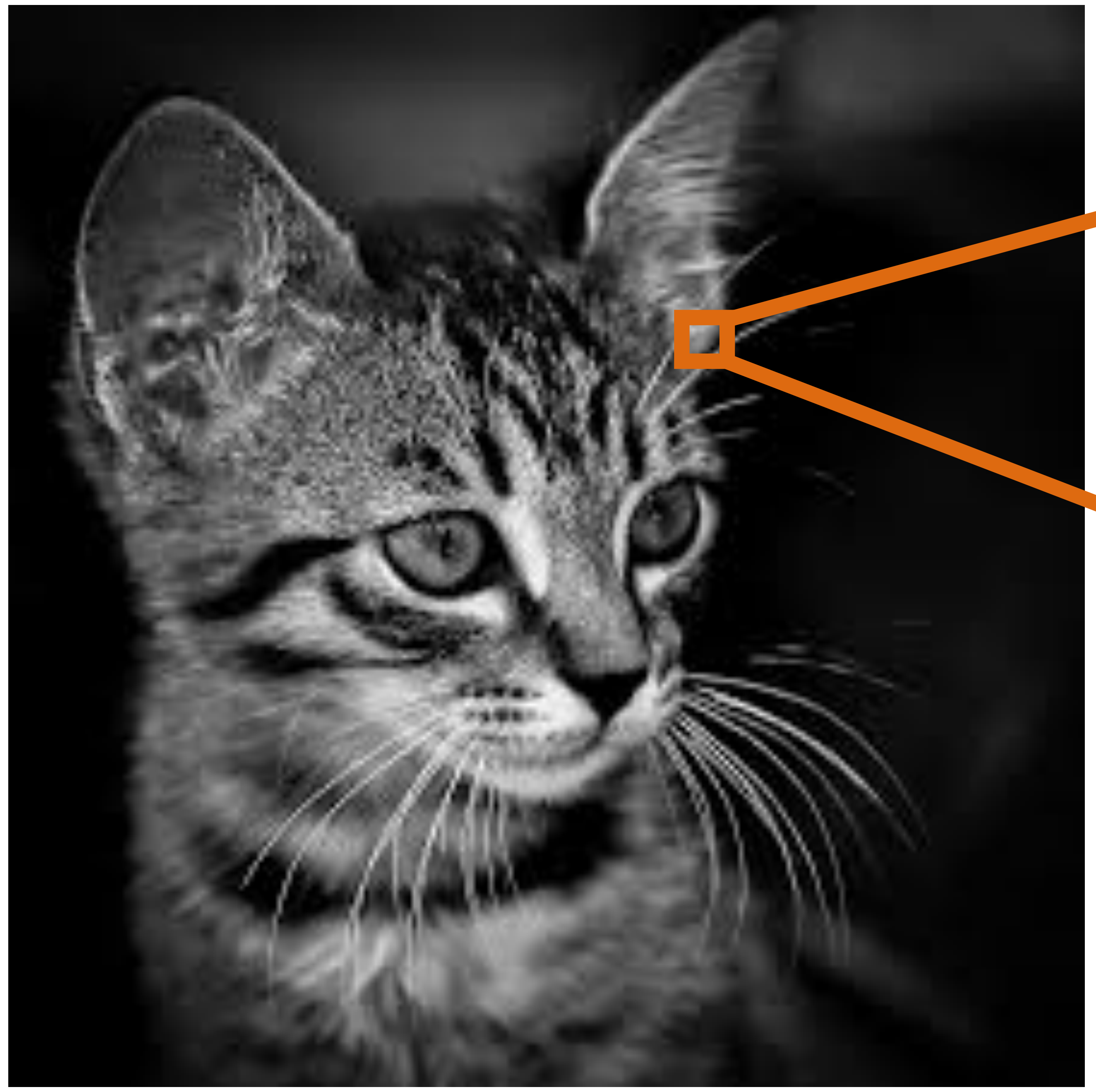


Multi-class classification results (30 training images per class)

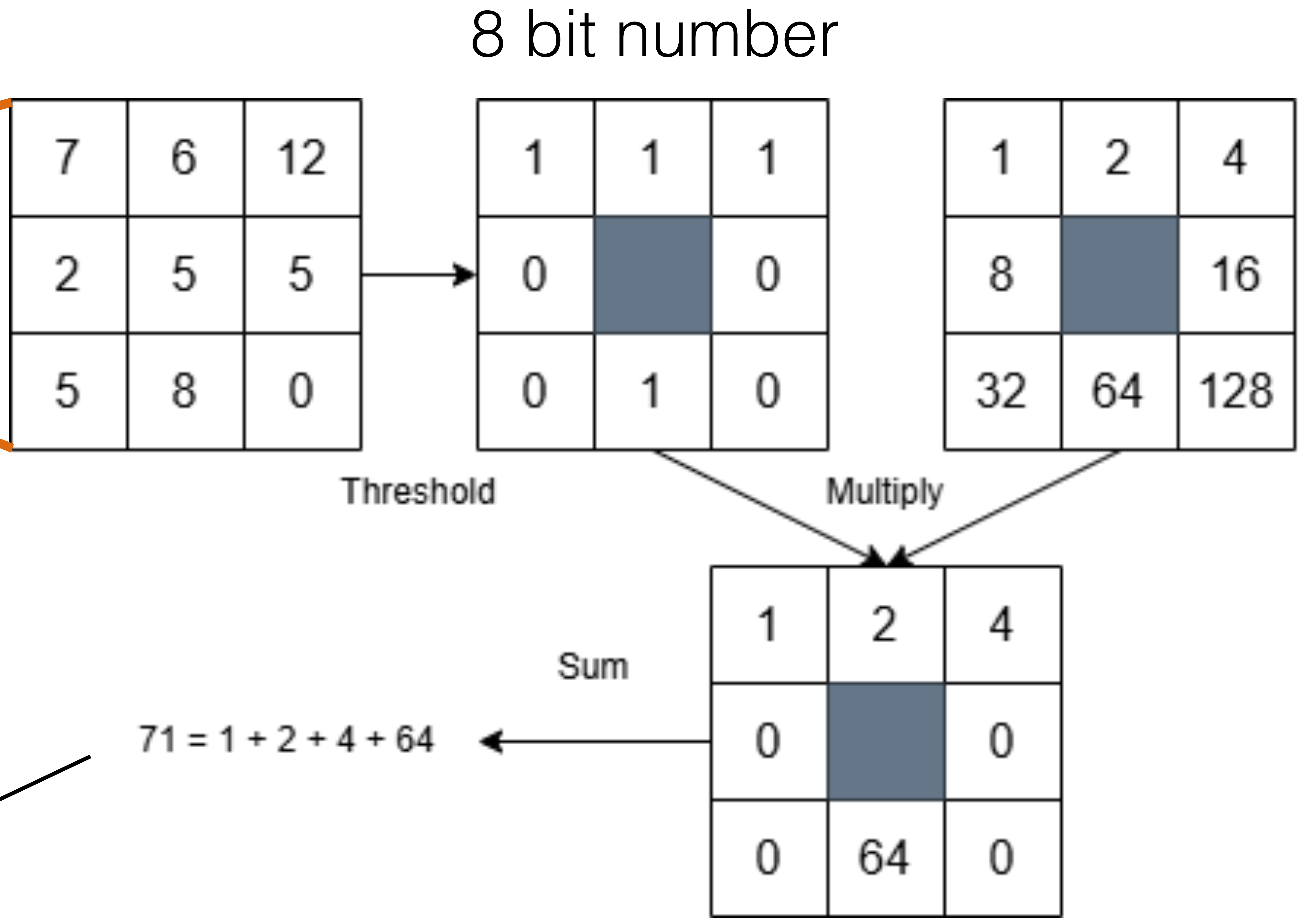
	Weak features (16)		Strong features (200)	
Level	Single-level	Pyramid	Single-level	Pyramid
0	15.5 ±0.9		41.2 ±1.2	
1	31.4 ±1.2	32.8 ±1.3	55.9 ±0.9	57.0 ±0.8
2	47.2 ±1.1	49.3 ±1.4	63.6 ±0.9	**64.6** ±0.8
3	52.2 ±0.8	**54.0** ±1.1	60.3 ±0.9	64.6 ±0.7

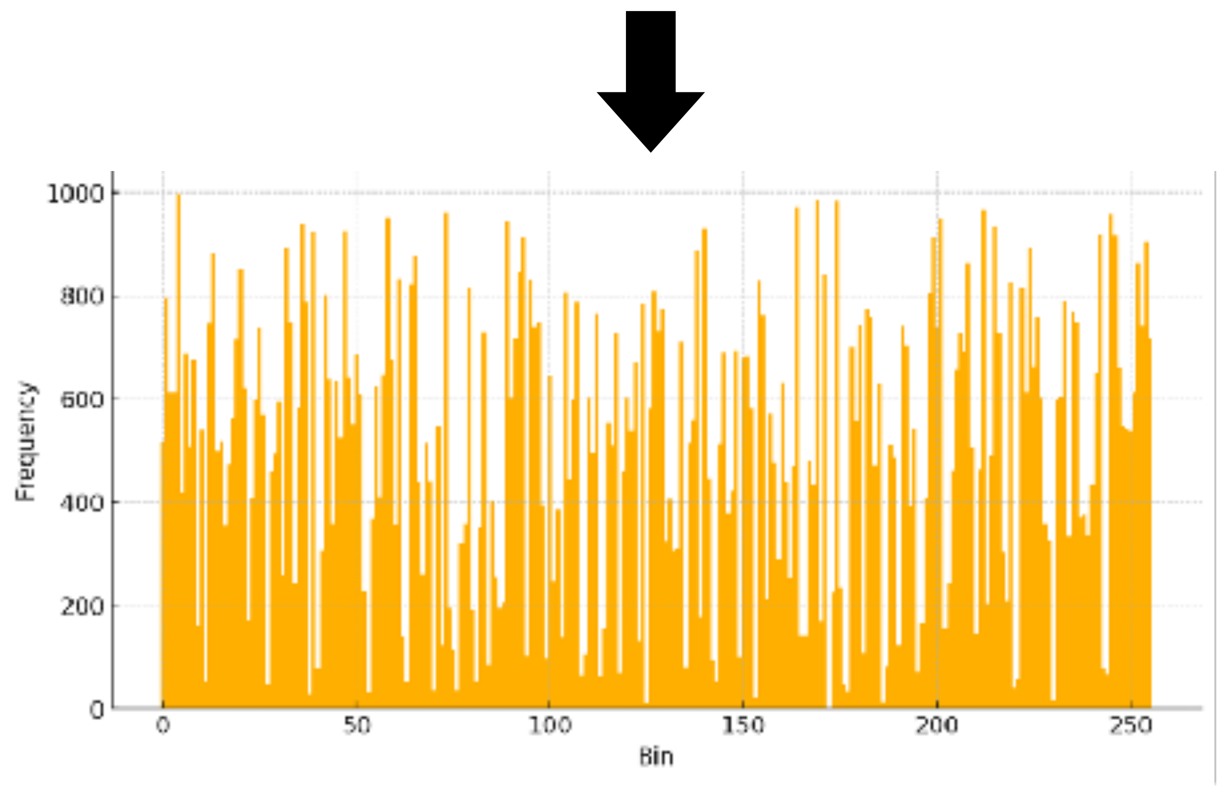
Local binary pattern — homework

T. Ojala, **M. Pietikäinen**, and D. Harwood “Performance evaluation of texture measures with classification based on Kullback discrimination of distributions”, ICPR’94



https://www.pickpik.com/kitten-black-white-cat-cute-domestic-136197





Further thoughts and readings …

All about embeddings (detailed experiments and code)

- K. Chatfield et al., The devil is in the details: an evaluation of recent feature encoding methods, BMVC 2011
- **http://www.robots.ox.ac.uk/~vgg/research/encoding_eval/**
- Includes discussion of advanced embeddings such as Fisher vector representations and locally linear coding (LLC)