Lecture 10.
Visual Bag of Words

Pattern Recognition and Computer Vision

Guanbin Li,
School of Computer Science and Engineering, Sun Yat-Sen University

扫码签到

What we will learn today?

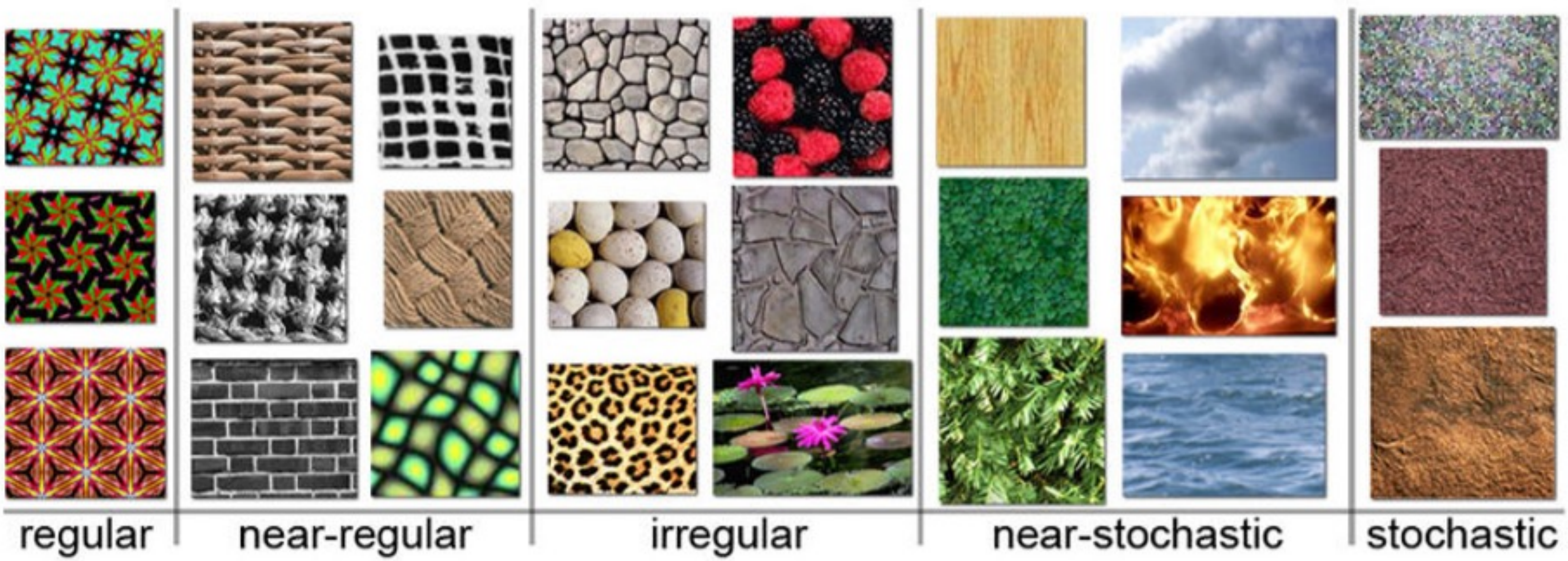
- Visual bag of words (BoW)
- Spatial Pyramid Matching
- Naive Bayes

Visual bag of words

Object → Bag of 'words'

Origin 1: Texture Recognition

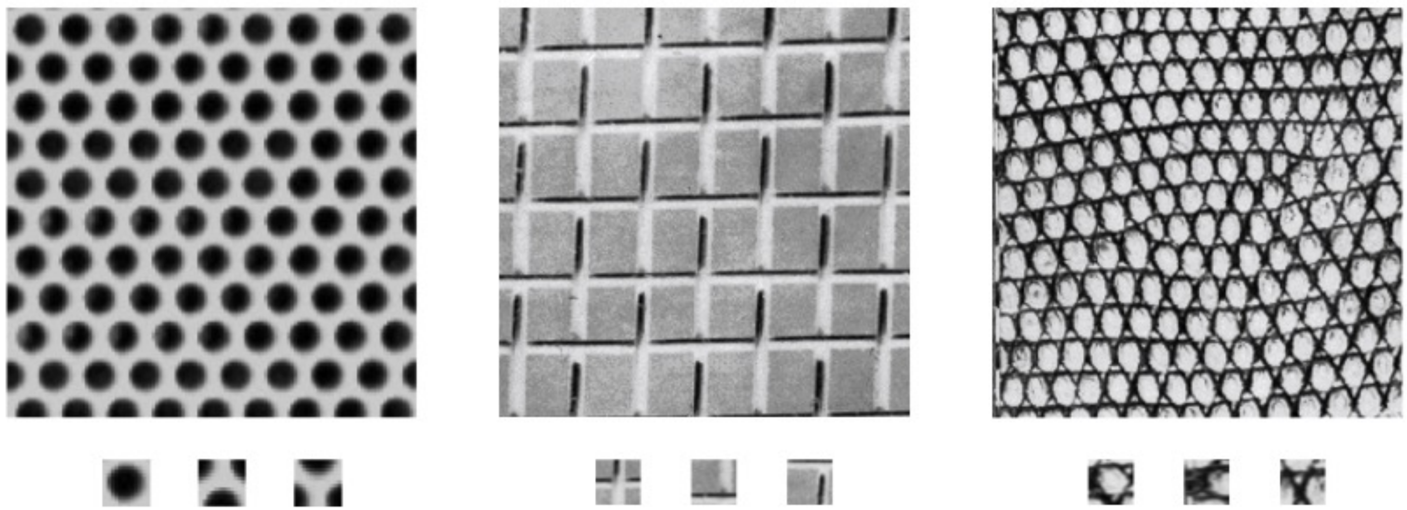
- 纹理泛指物体面上的花纹或线条，是物体上呈现的线形纹路。
- 传统意义上，在图像中纹理是特征值强度的某种局部重复模式的宏观表现，局部模式重复和平稳性是其主要特点；



Example textures (from Wikipedia)

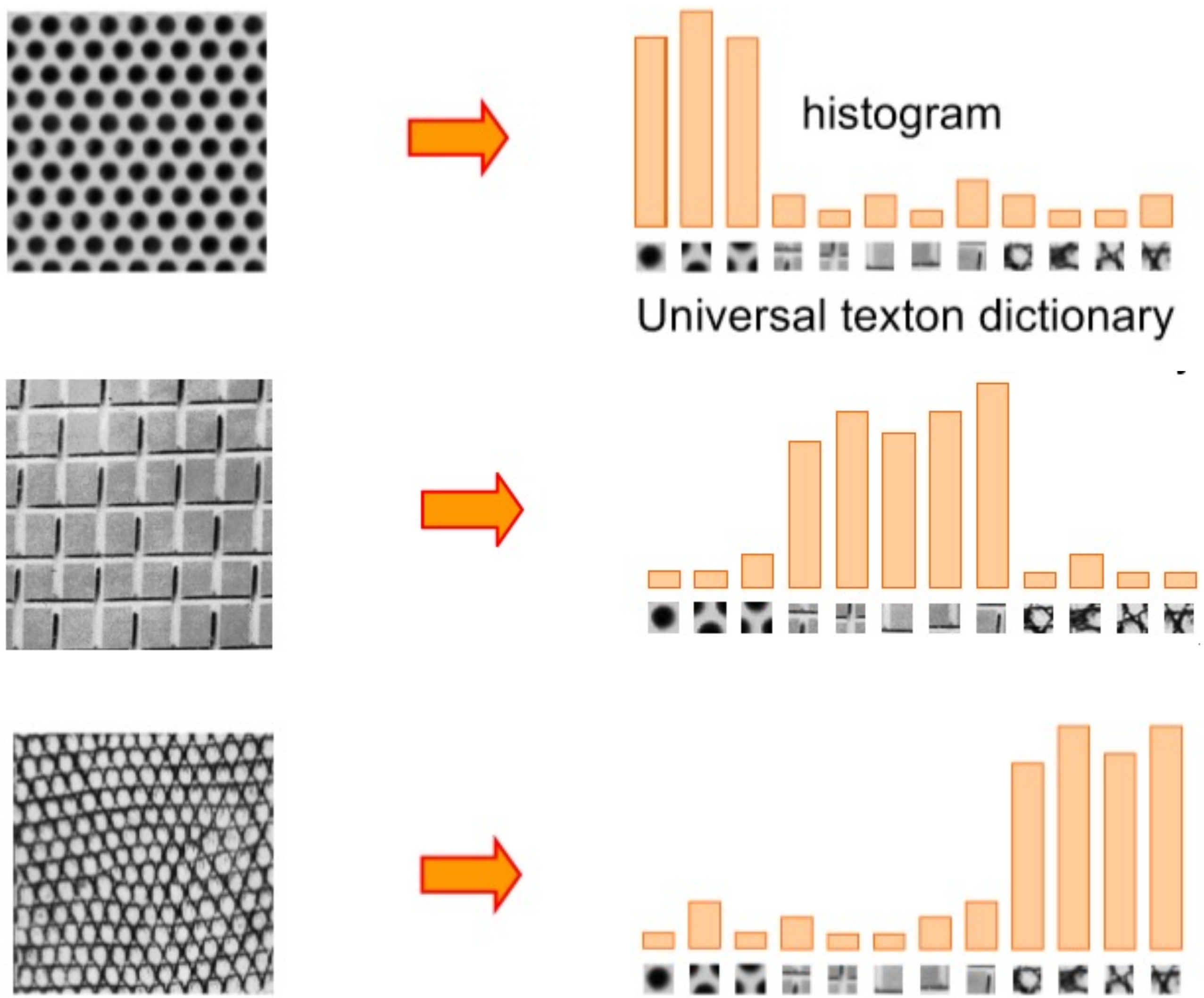
Origin 1: Texture Recognition

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



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



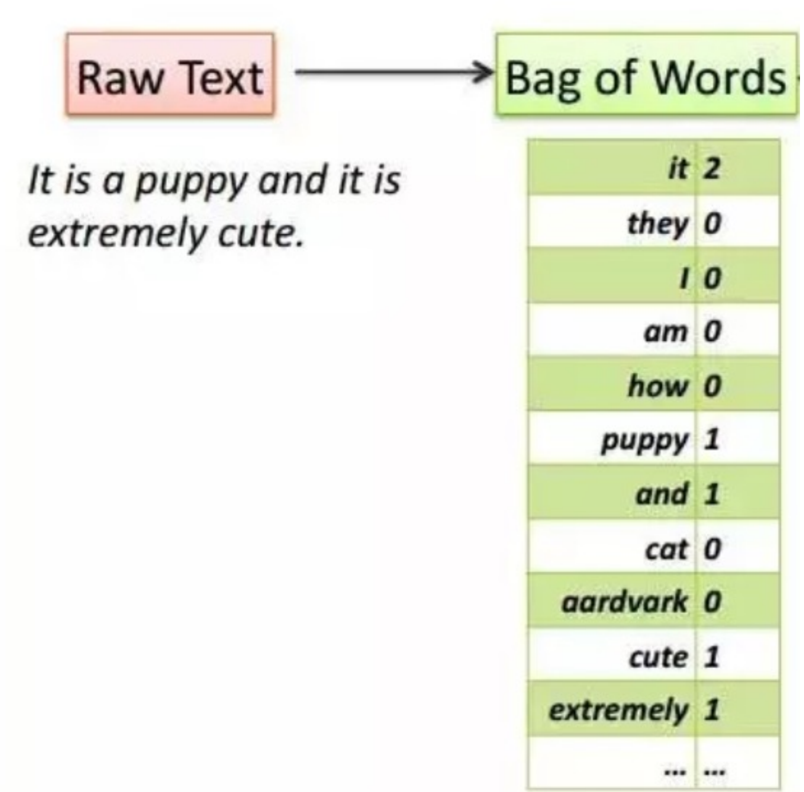
Origin 2: Bag-of-words models

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

US Presidential Speeches Tag Cloud

Origin 2: Bag-of-words models

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



Bags of features for object recognition

face, flowers, building

- • Works pretty well for image-level classification and for recognizing object instances

Bags of features for object recognition

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

Bag of features

- First, take a bunch of images, extract features, and build up a “dictionary” or “visual vocabulary” –a list of common features
- Given a new image, extract features and build a histogram – for each feature, find the closest visual word in the dictionary

Bag of features: outline

- 1. Extract features

Bag of features: outline

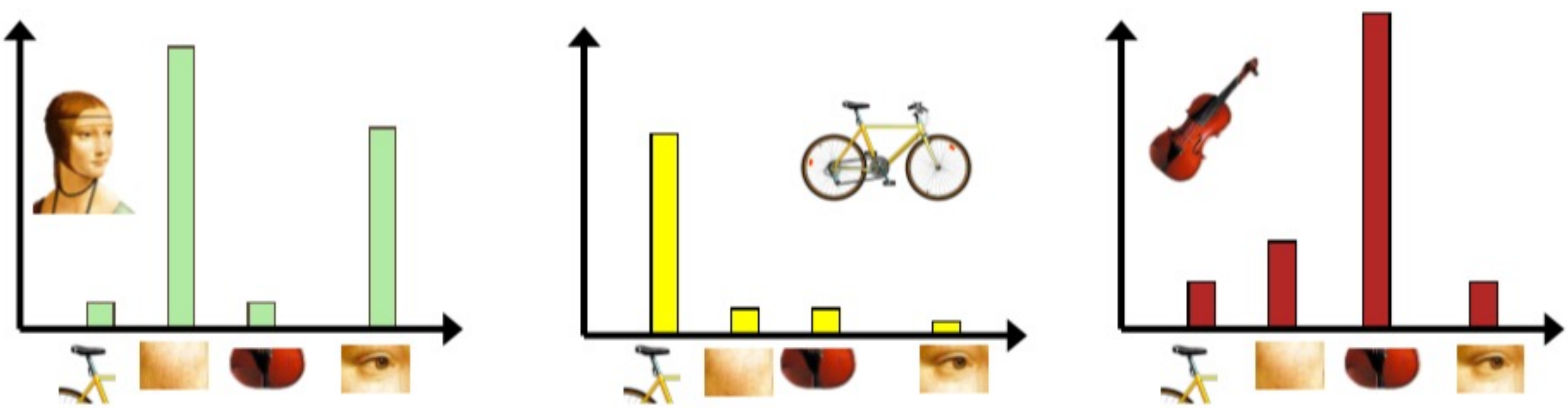
- 1. Extract features
- 2. Learn “visual vocabulary”

Bag of features: outline

- 1. Extract features
- 2. Learn “visual vocabulary”
- 3. Quantize features using visual vocabulary

Bag of features: outline

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

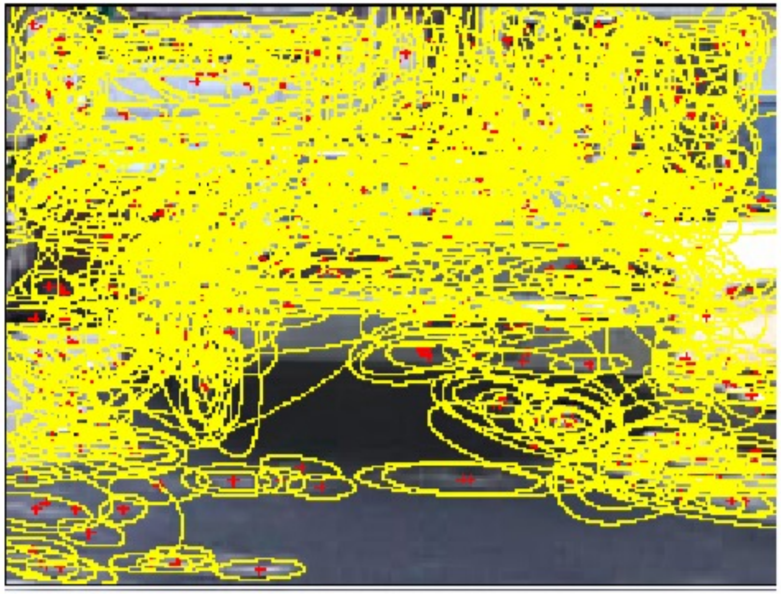


1. Feature extraction

- Regular grid
  - – Vogel & Schiele, 2003
  - – Fei-Fei & Perona, 2005

1. Feature extraction

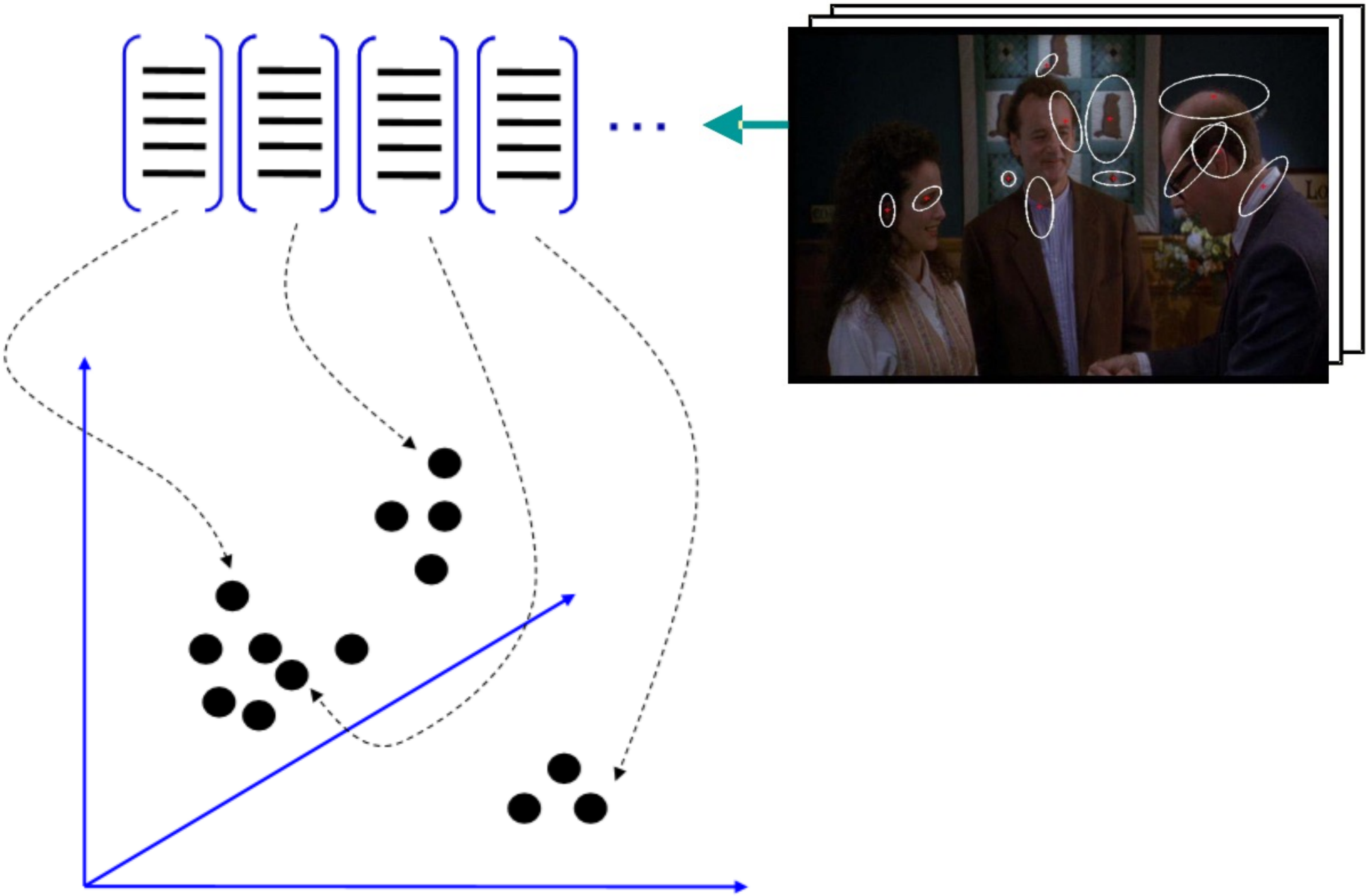
- Regular grid
  - – Vogel & Schiele, 2003
  - – Fei-Fei & Perona, 2005
- Interest point detector
  - – Csurkaet al. 2004
  - – Fei-Fei & Perona, 2005
  - – Sivicet al. 2005



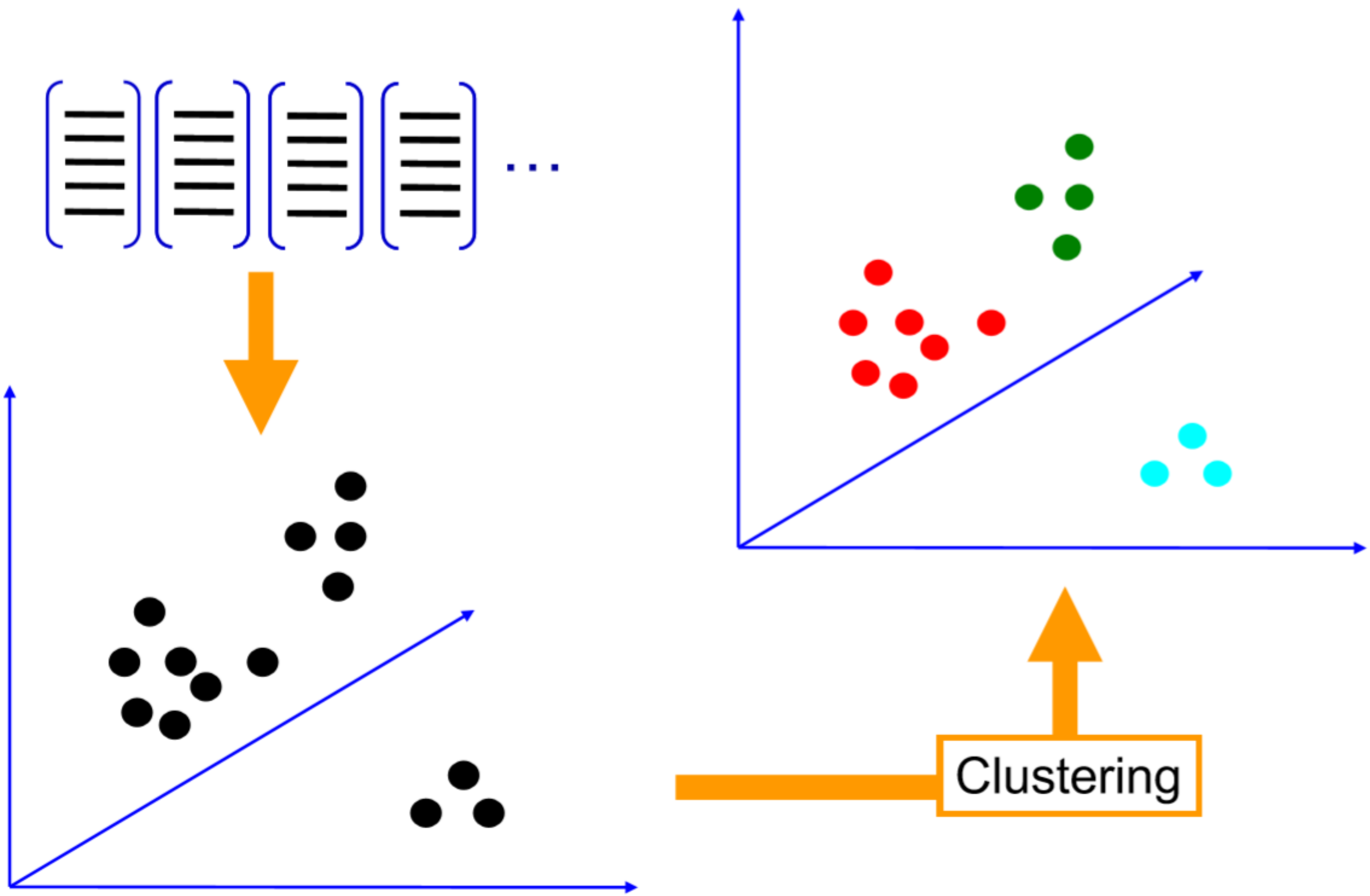
1. Feature extraction

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

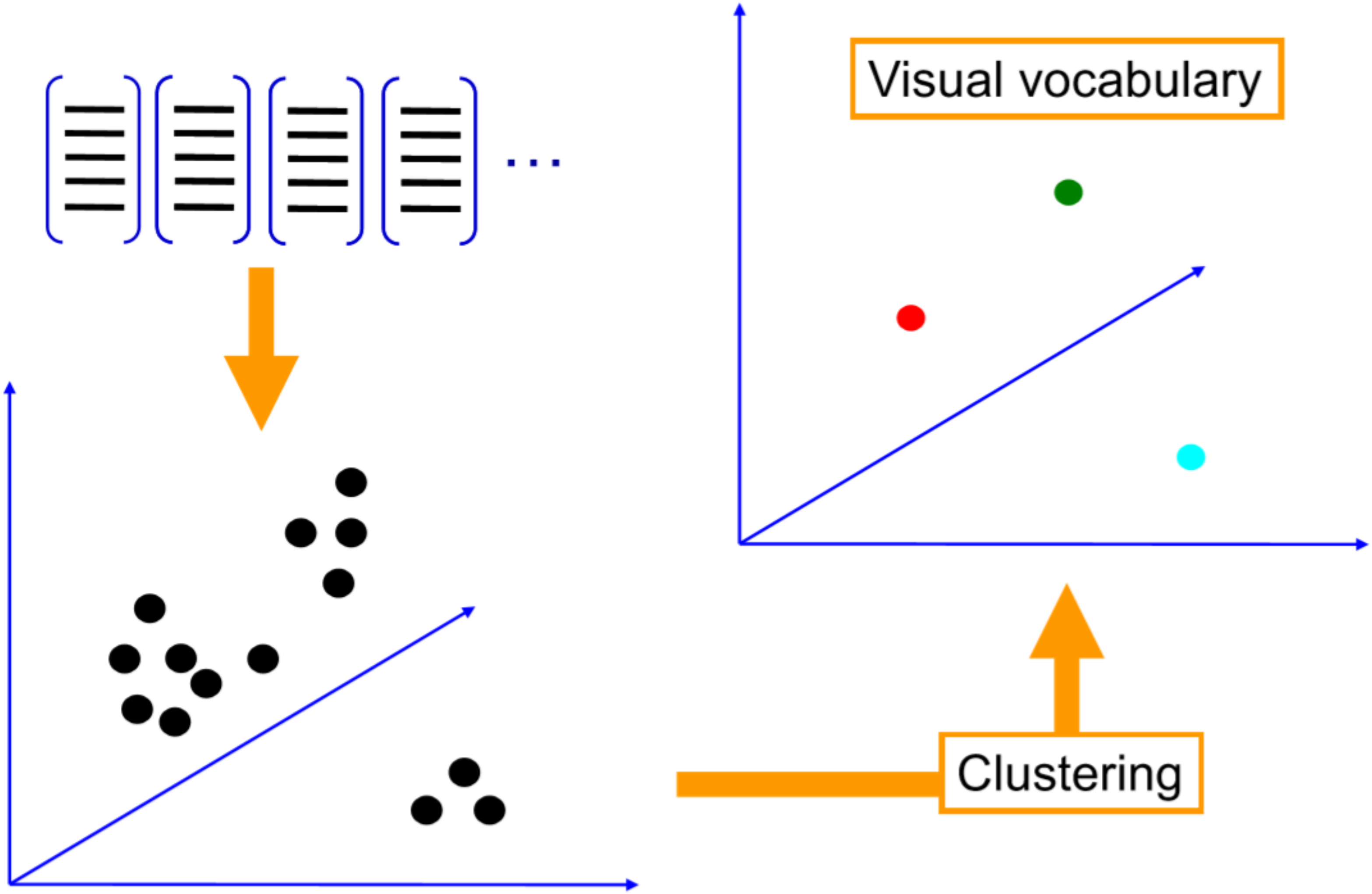
2. Learning the visual vocabulary



2. Learning the visual vocabulary



2. Learning the visual vocabulary



K-means clustering recap

- Want to minimize sum of squared Euclidean distances between points x_{i} and their nearest cluster centers m_k。

$$D(X,M)=\sum_{\text{cluster } k}\ \sum_{\substack{\text{point } i \text{ in}\\ \text{cluster } k}}(x_i-m_k)^2$$

- •Algorithm:
- •Randomly initialize K cluster centers
- Iterate until convergence:
  - – Assign each data point to the nearest center
  - – Recompute each cluster center as the mean of all points assigned to it

From clustering to 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

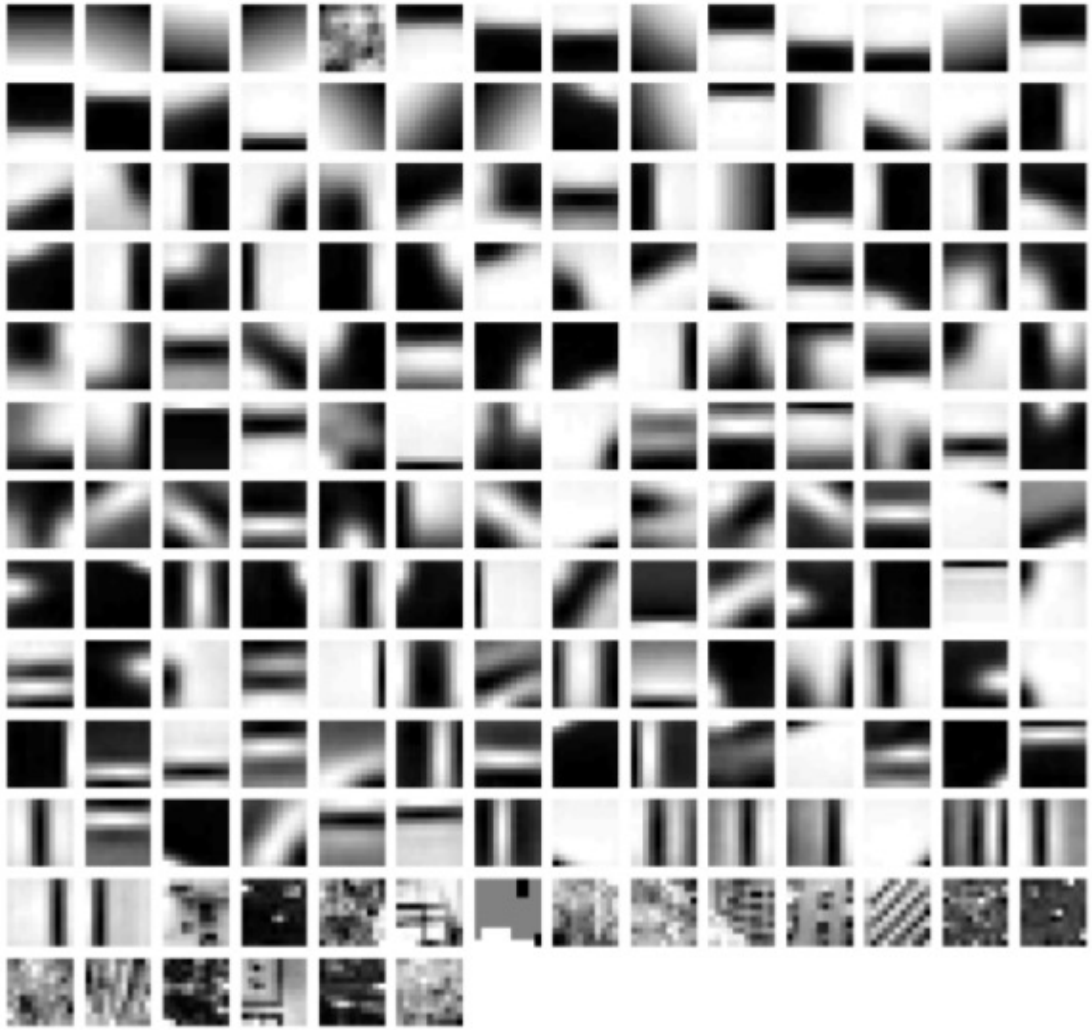
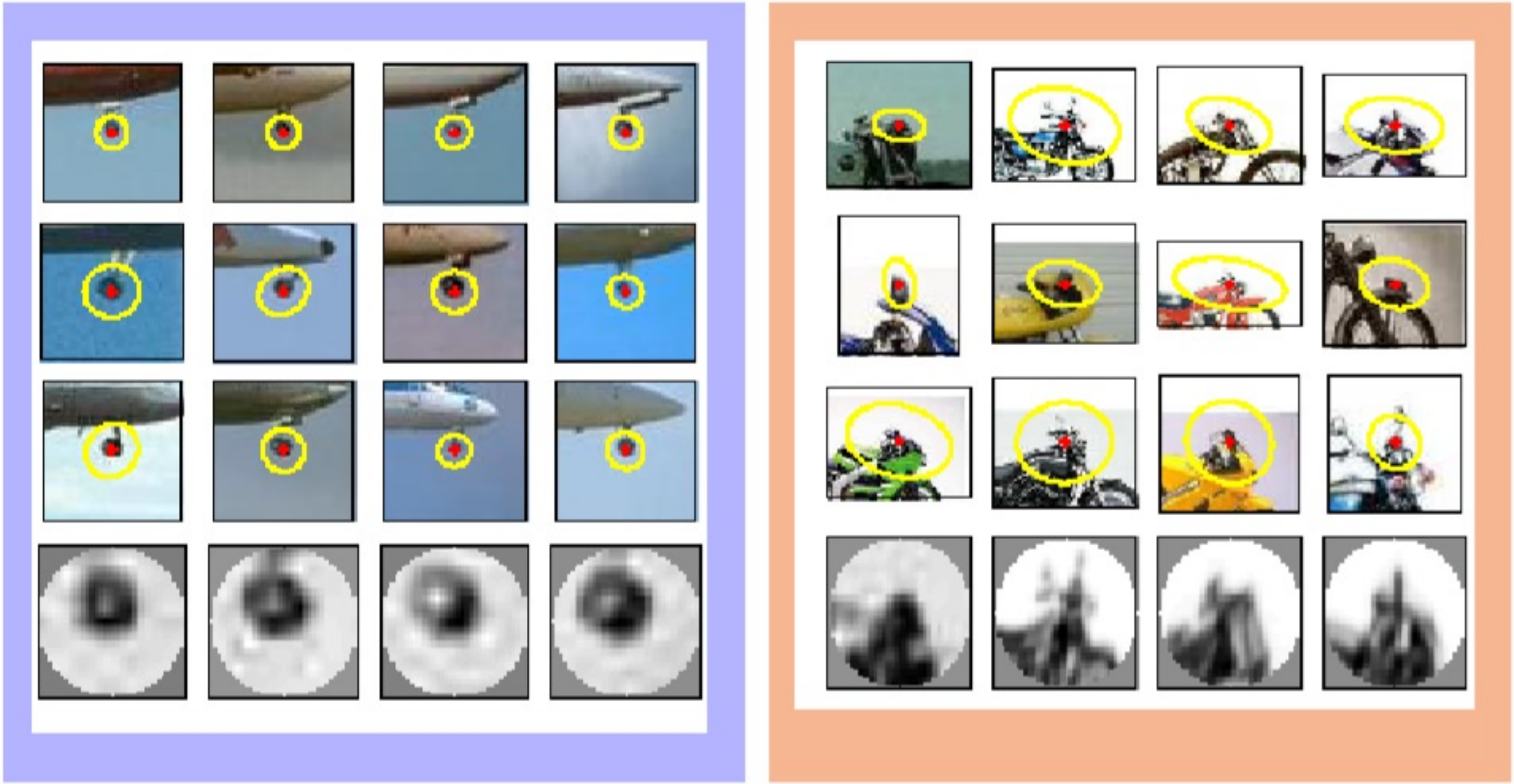
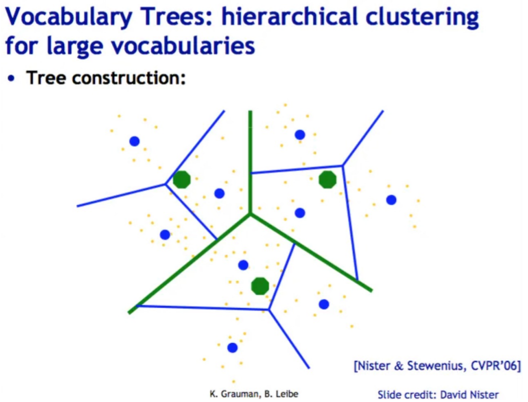


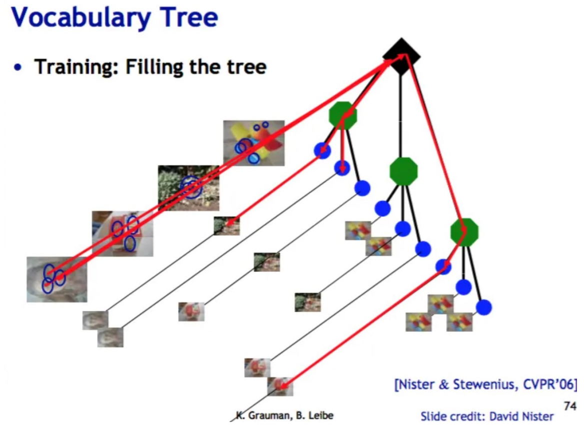
Image patch examples of visual words

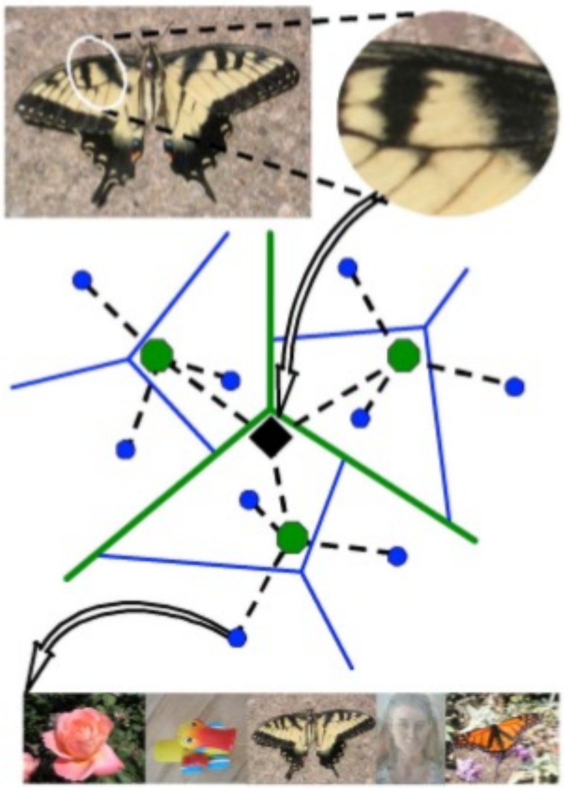


Visual vocabularies: Issues

- • 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)







https://zhuanlan.zhihu.com/p/20554144

3. Image representation

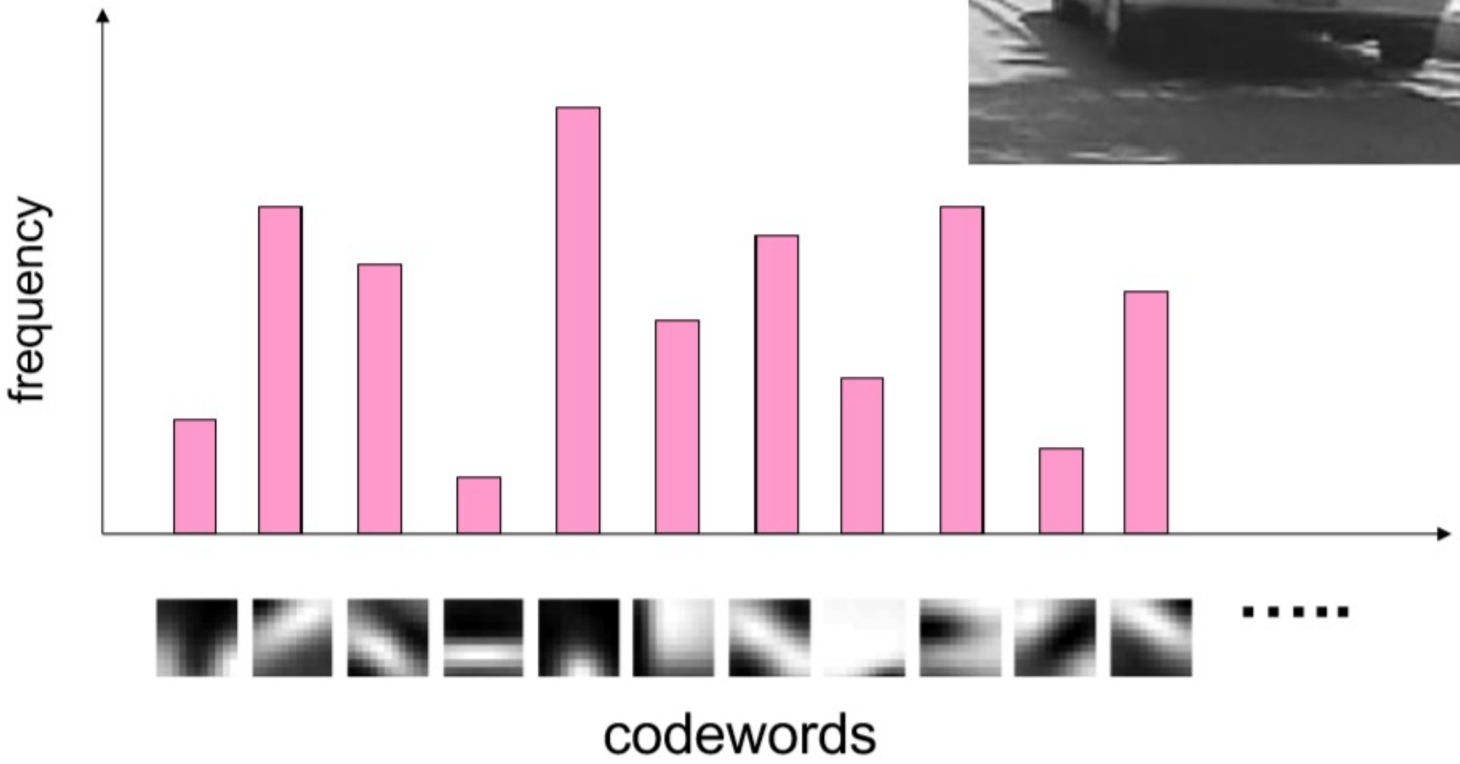
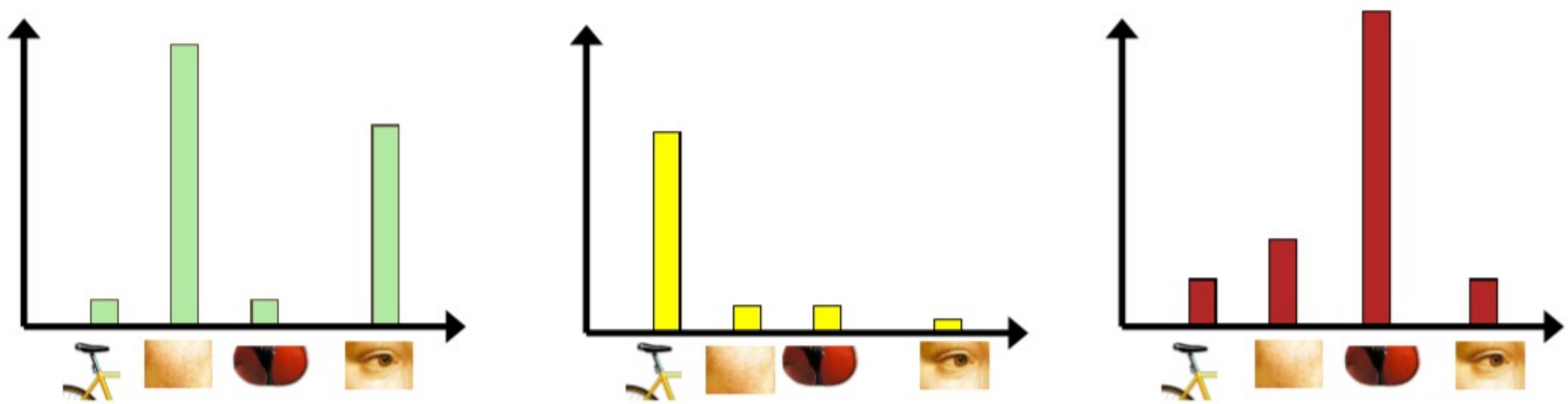


Image classification

- Given the bag-of-features representations of images from different classes, how do we learn a model for distinguishing them?



Uses of BoW representation

- Treat as feature vector for standard classifier
  - –e.g k-nearest neighbors, support vector machine
- Cluster BoW vectors over image collection
  - –Discover visual themes

Large-scale image matching

11,400 images of game covers
(Caltech games dataset)

- Bag-of-words models have been useful in matching an image to a large database of object instances

how do I find this image in the database?

Large-scale image search

- Build the database:
  - –Extract features from the database images
  - –Learn a vocabulary using k-means (typical k: 100,000)
  - –Compute weights for each word
  - –Create an inverted file mapping words to images

Weighting the words

- Just as with text, some visual words are more discriminative than others

  the, and, or vs. ***cow, AT&T, Cher***

- the bigger fraction of the documents a word appears in, the less useful it is for matching
  - –e.g., a word that appears in all documents is not helping us

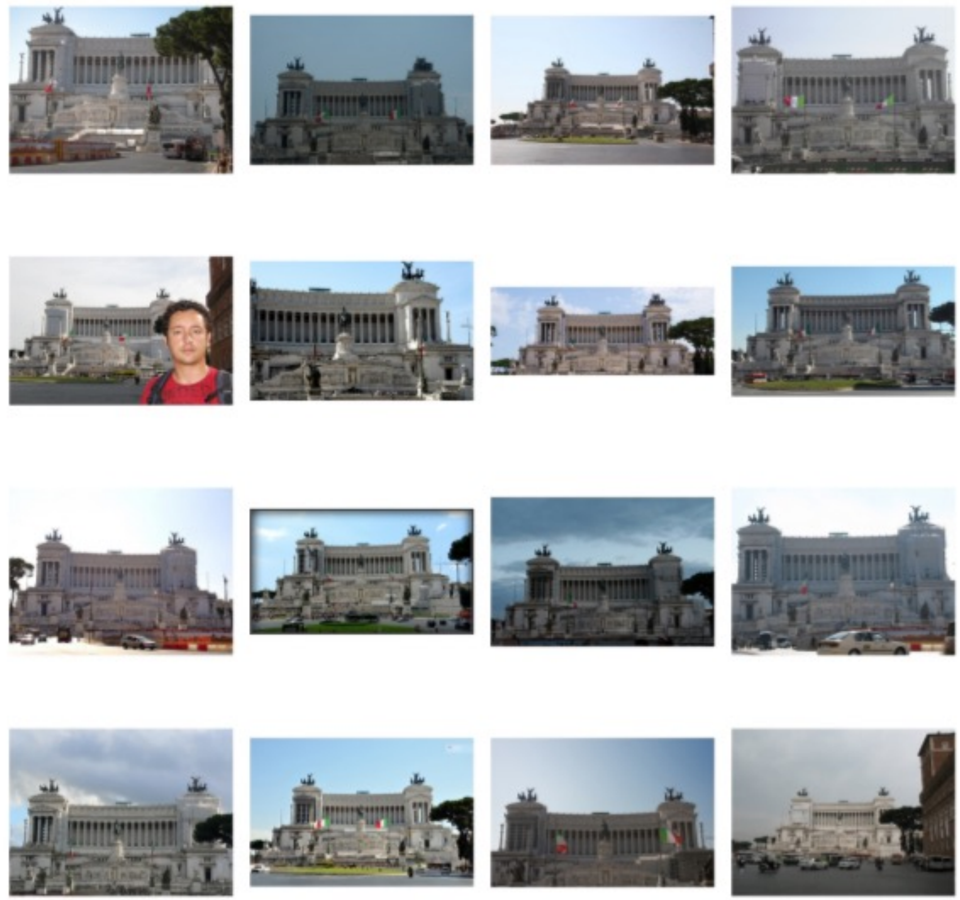
Large-scale image search

- • Pros:
  - –Works well for CD covers, movie posters
  - –Real-time performance possible

real-time retrieval from a database of 40,000 CD covers

Nister & Stewenius, **Scalable Recognition with a Vocabulary Tree**

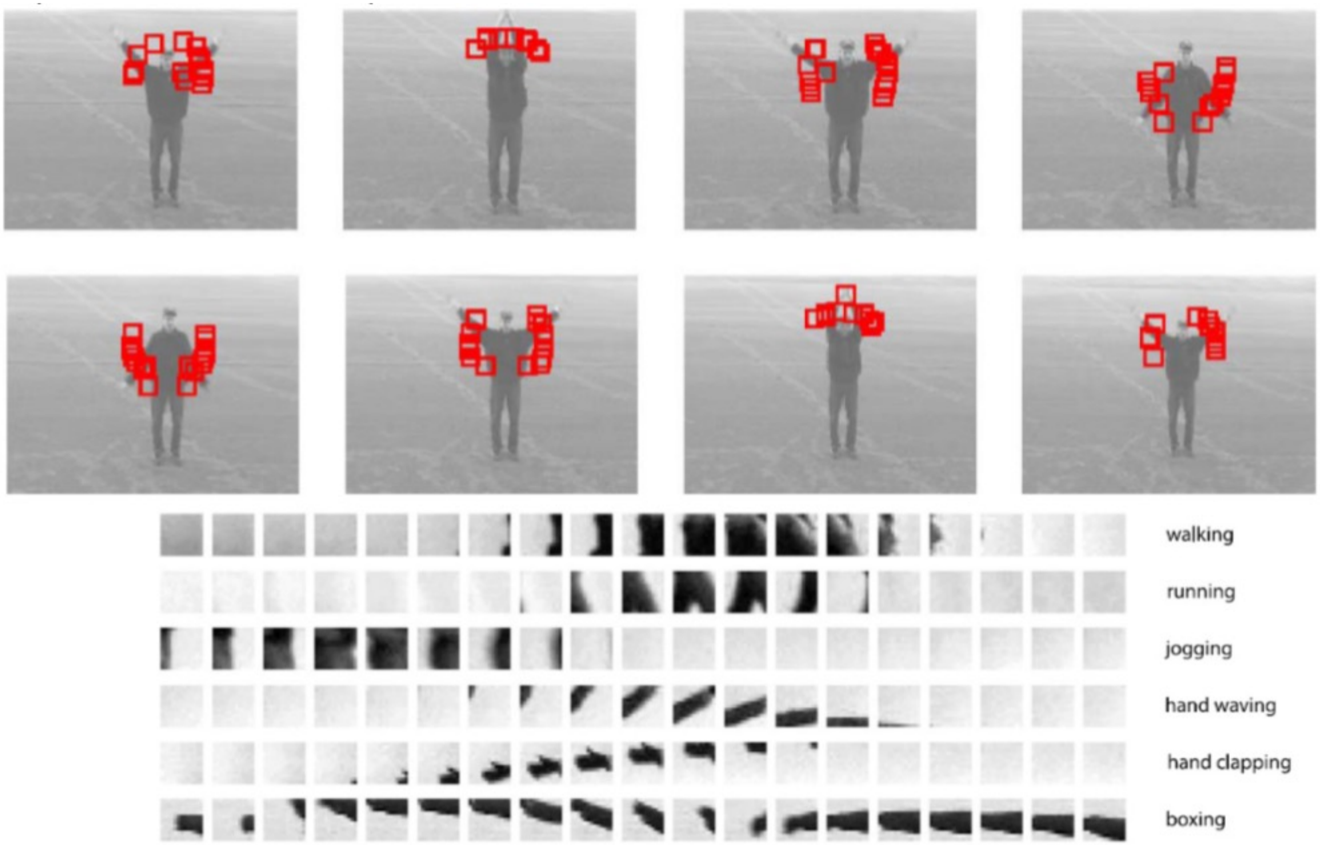
Example bag-of-words matches



Example bag-of-words matches

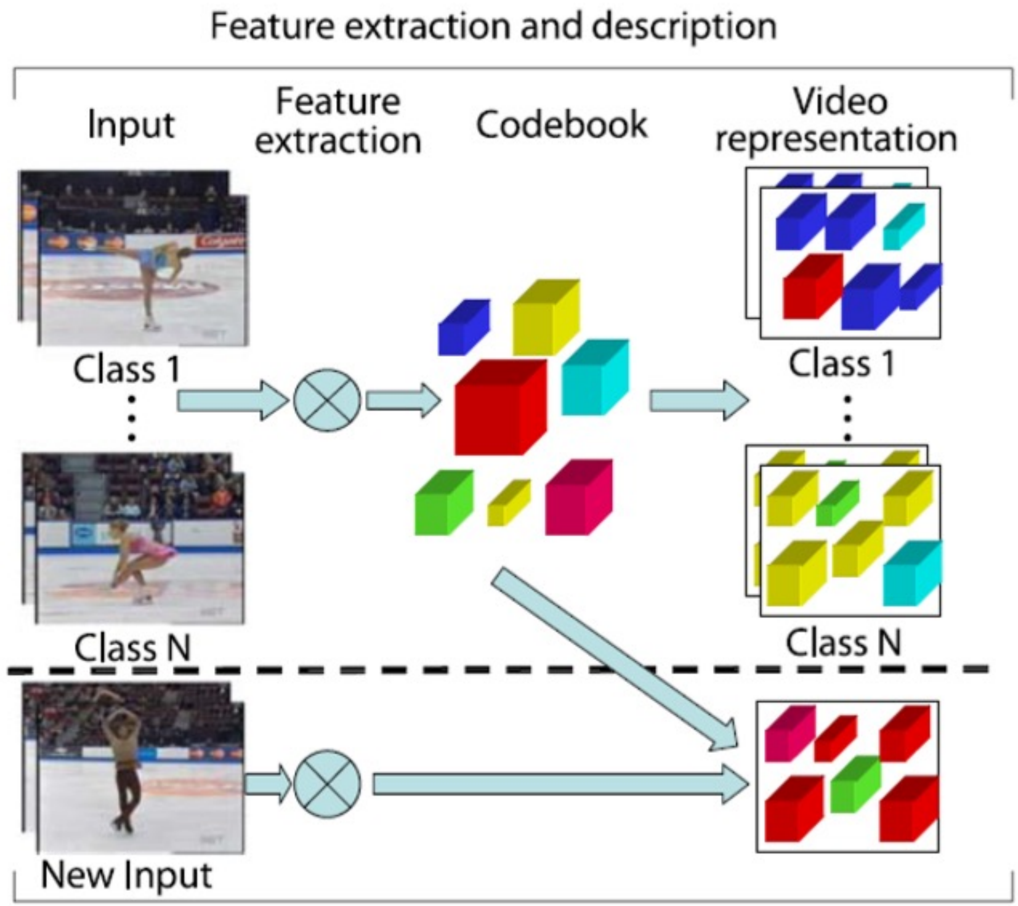
Bags of features for action recognition

- Space-time interest points



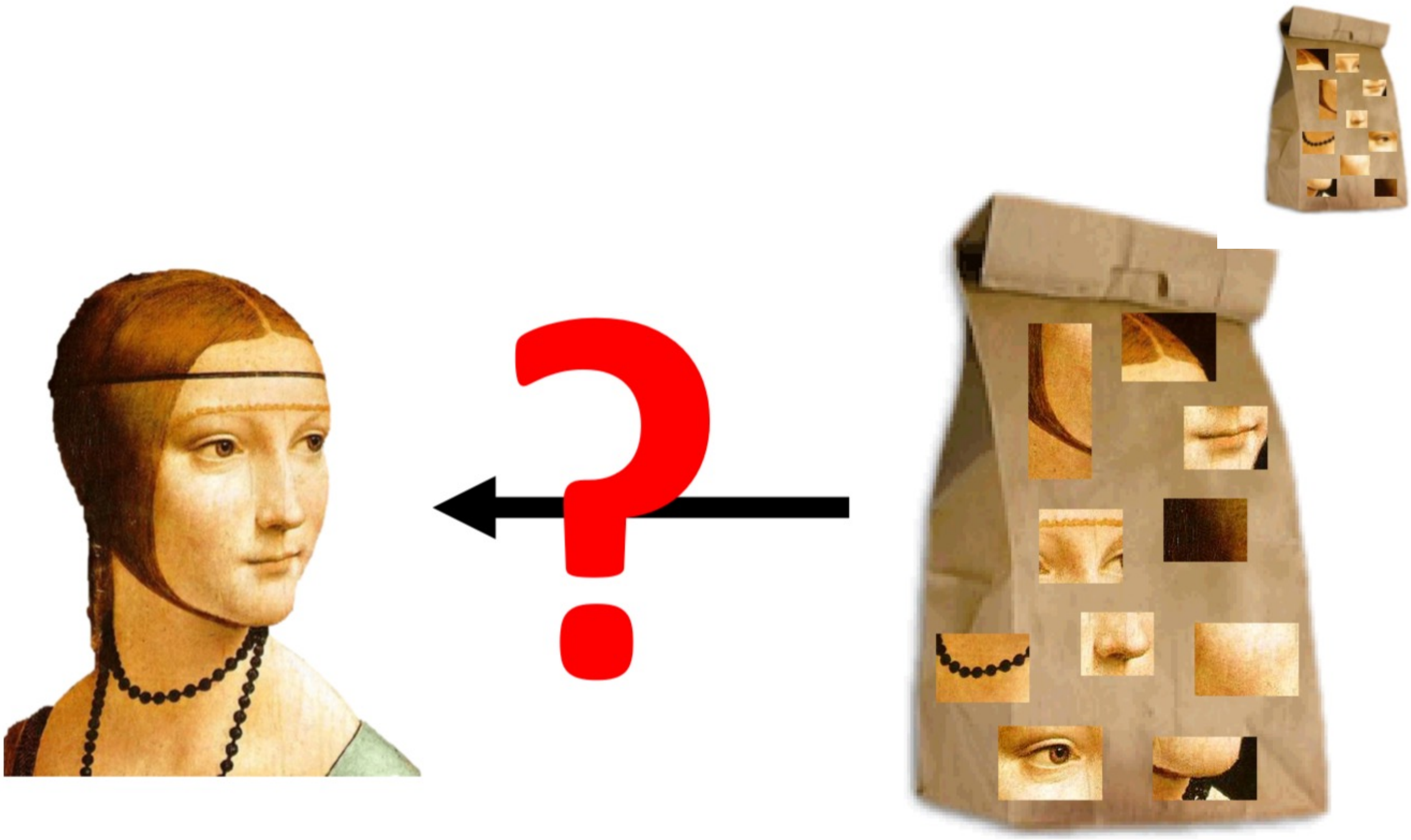
Juan Carlos Niebles, Hongcheng Wang and Li Fei-Fei, [Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words](), IJCV 2008.

Bags of features for action recognition



Juan Carlos Niebles, Hongcheng Wang and Li Fei-Fei, [Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words](), IJCV 2008.

What about spatial info?



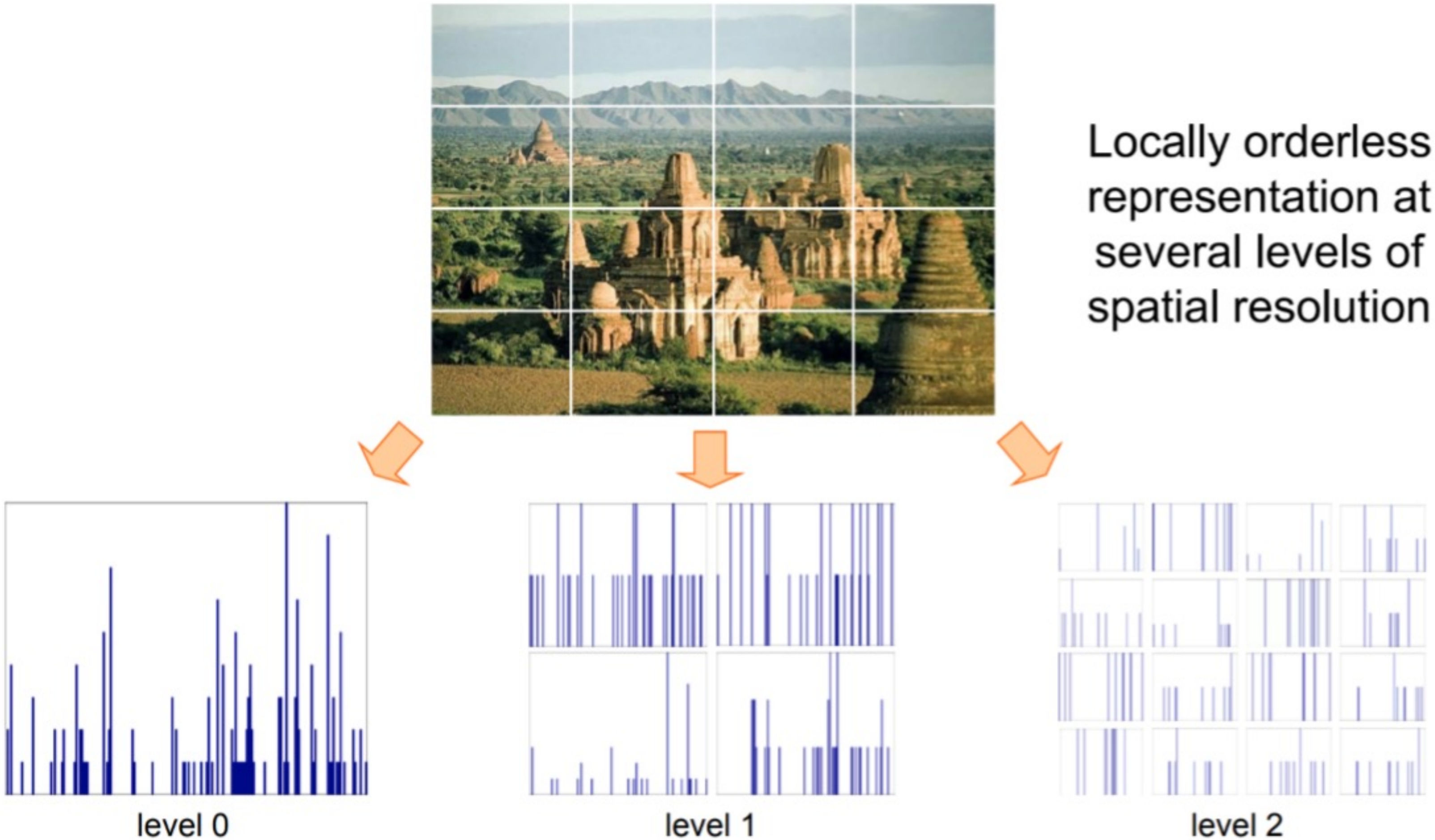
What we will learn today?

- Visual bag of words (BoW)
- Spatial Pyramid Matching
- Naive Bayes

Pyramids

- Very useful for representing images.
- Pyramid is built by using multiple copies of image.
- Each level in the pyramid is 1/4 of the size of previous level.
- The lowest level is of the highest resolution.
- The highest level is of the lowest resolution.

Bag of words + pyramids



Locally orderless representation at several levels of spatial resolution

Scene category dataset

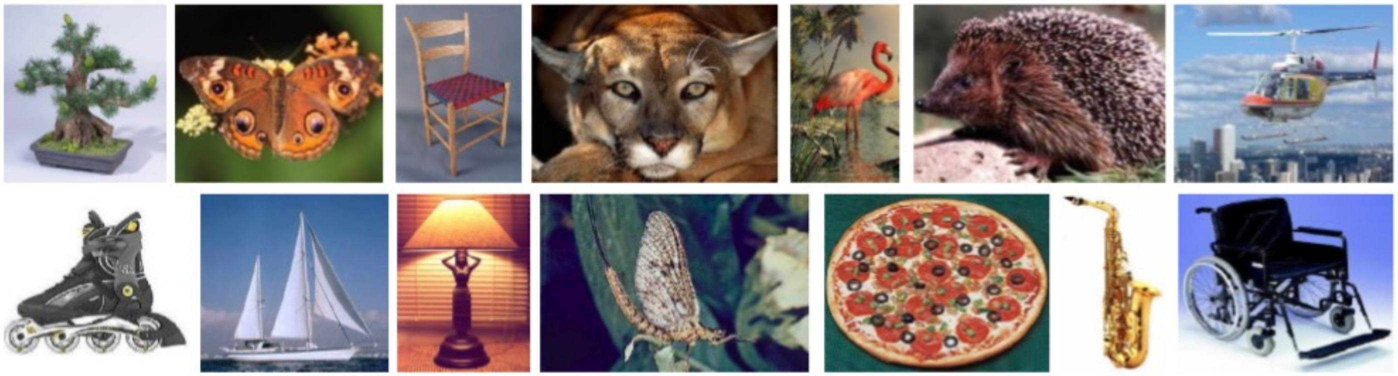
Multi-class classification results
(100 training images per class)

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

Bag of words + pyramids

Caltech101 dataset

http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html



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

Slide credit: Svetlana Lazebnik

What we will learn today?

- Visual bag of words (BoW)
- Spatial Pyramid Matching
- Naive Bayes

Naïve Bayes

- Classify image using histograms of occurrences on visual words:

- • where:
  - $-x^i$ is the event of visual word v^i appearing in the image,
  - $-N(i)$ the number of times word v^i occurs in the image,
  - $-m$ is the number of words in our vocabulary.

Naïve Bayes -classification

- Our goal is to classify that the image represented by $\boldsymbol{x}$ is belongs class that has the highest posterior probability:

$$c^* = arg \max_{c} P(c \mid \boldsymbol{x})$$

Naïve Bayes –conditional independence

- Naïve Bayes classifier assumes that visual words are conditionally independent given object class.
- Therefore, we can multiply the probability of each visual word to obtain the joint probability.
- Model for image x under object class c:

$$P(x \mid c) = \prod_{i=1}^{m} P(x_i \mid c)$$

- How do we compute $P(x^i|c)$

Naïve Bayes –prior

- Class priors $P(c)$ encode how likely we are to see one class versus others.
- Note that:

$$\sum_{i=1}^{m} P(c) = 1$$

Naïve Bayes -posterior

- equations from the previous slides, we can now calculate the probability that an image represented by $\boldsymbol{x}$ belongs to class category c.

$$P(c \mid \boldsymbol{x}) = \frac{P(c)\, P(\boldsymbol{x} \mid c)}{\sum_{c'} P(c')\, P(\boldsymbol{x} \mid c')}$$

Bayes Theorem

Naïve Bayes -posterior

- equations from the previous slides, we can now calculate the probability that an image represented by $\boldsymbol{x}$ belongs to class category c.

$$P(c \mid \boldsymbol{x}) = \frac{P(c)\, P(\boldsymbol{x} \mid c)}{\sum_{c'} P(c')\, P(\boldsymbol{x} \mid c')}$$

$$P(c \mid \boldsymbol{x}) = \frac{P(c) \prod_{i=1}^{m} P(x_i \mid c)}{\sum_{c'} P(c') \prod_{i=1}^{m} P(x_i \mid c')}$$

Naïve Bayes -classification

- We can now classify that the image represented by $\boldsymbol{x}$ is belongs class that has the highest probability:

$$c^* = arg\max_c P(c \mid \boldsymbol{x})$$

$$c^* = arg\max_c \log P(c \mid \boldsymbol{x})$$

Let's break down the posterior

The probability that $\boldsymbol{x}$ belongs to class c_1:

$$P(c_1 \mid \boldsymbol{x}) = \frac{P(c_1) \prod_{i=1}^{m} P(x_i \mid c_1)}{\sum_{c'} P(c') \prod_{i=1}^{m} P(x_i \mid c')}$$

And the probability that $\boldsymbol{x}$ belongs to class c_2:

$$P(c_2 \mid \boldsymbol{x}) = \frac{P(c_2) \prod_{i=1}^{m} P(x_i \mid c_2)}{\sum_{c'} P(c') \prod_{i=1}^{m} P(x_i \mid c')}$$

Both their denominators are the same

The probability that $\boldsymbol{x}$ belongs to class c_1:

$$P(c_1 \mid \boldsymbol{x}) = \frac{P(c_1)\prod_{i=1}^{m} P(x_i \mid c_1)}{\sum_{c'} P(c')\prod_{i=1}^{m} P(x_i \mid c')}$$

And the probability that $\boldsymbol{x}$ belongs to class c_2:

$$P(c_2 \mid \boldsymbol{x}) = \frac{P(c_2)\prod_{i=1}^{m} P(x_i \mid c_2)}{\sum_{c'} P(c')\prod_{i=1}^{m} P(x_i \mid c')}$$

Both their denominators are the same

- Since we only want the max, we can ignore the denominator:

$$P(c_1 \mid \boldsymbol{x}) \propto P(c_1) \prod_{i=1}^{m} P(x_i \mid c_1)$$

$$P(c_2 \mid \boldsymbol{x}) \propto P(c_2) \prod_{i=1}^{m} P(x_i \mid c_2)$$

For the general class c,

$$P(c \mid \boldsymbol{x}) \propto P(c) \prod_{i=1}^{m} P(x_i \mid c)$$

We can take the log:

$$\log P(c \mid \boldsymbol{x}) \propto logP(c) + \sum_{i=1}^{m} logP(x_i \mid c)$$

Naïve Bayes -classification

- So, the following classification becomes:

$$c^* = arg\max_c P(c \mid \boldsymbol{x})$$
$$c^* = arg\max_c \log P(c \mid \boldsymbol{x})$$

$$c^* = arg\max_c logP(c) + \sum_{i=1}^{m} logP(x_i \mid c)$$

What we have learned today

- Visual bag of words (BoW)
- Spatial Pyramid Matching
- Naive Bayes