Principles of Safe Autonomy
Lecture 6: Clustering and recognition

Sayan Mitra 2022

slides from Svetlana Lazebnik

Thu 10 31°/23° Partly Cloudy 8% W 13 mph

Announcements

- Demo at Highbay Thursday Feb 10, at 11 am-12:30 pm (dress warm)
  - Next to iHotel (201 st Marys's)

- Volunteer to type up notes for next week; participation points

Overview

- Recognition tasks
- A statistical learning approach
- "Classic" recognition pipeline
    - Bags of features
    - Spatial pyramids
- Classifiers: SVM

The statistical learning framework

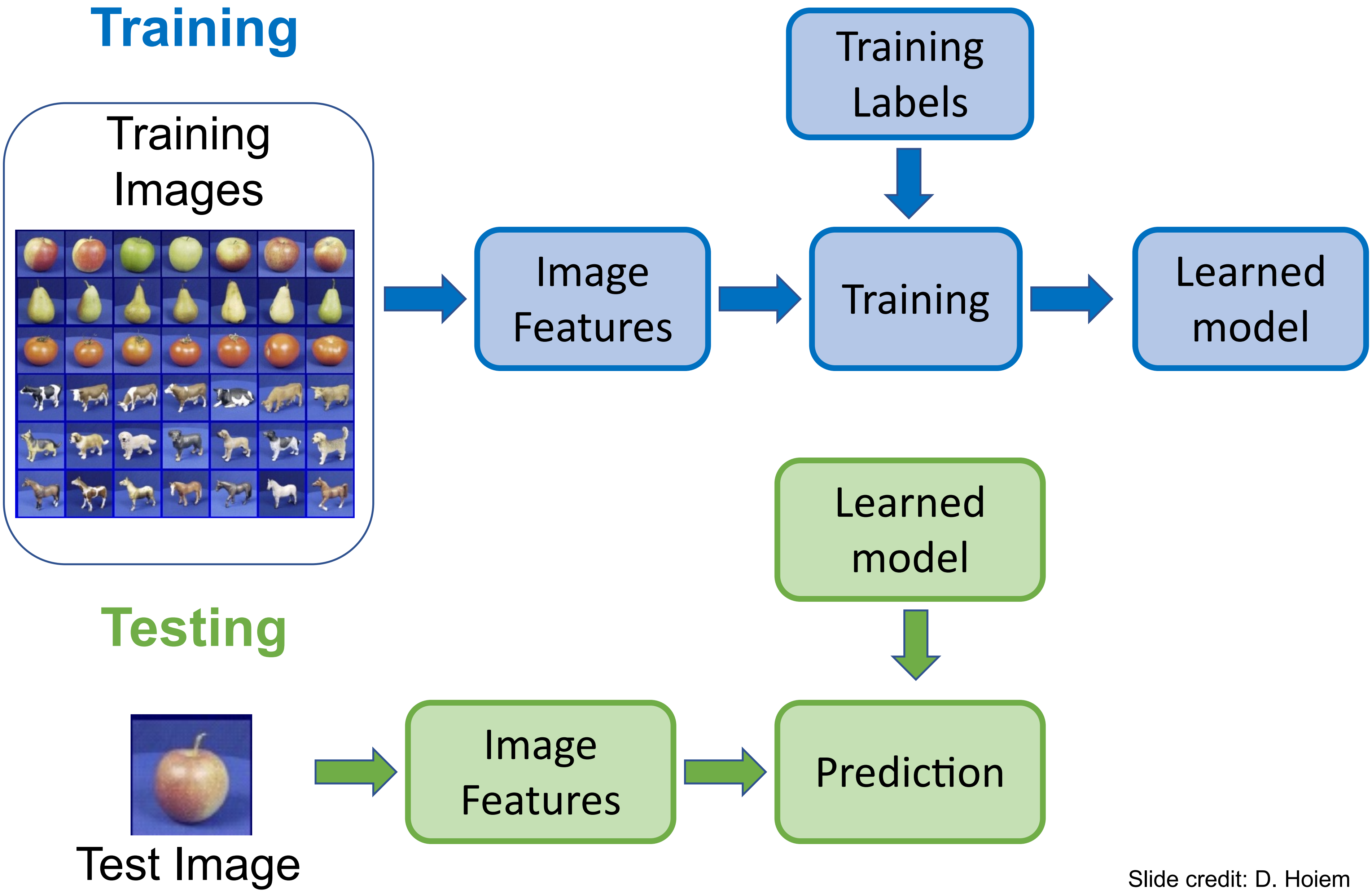
$$y = f(\mathbf{x})$$

output

prediction function

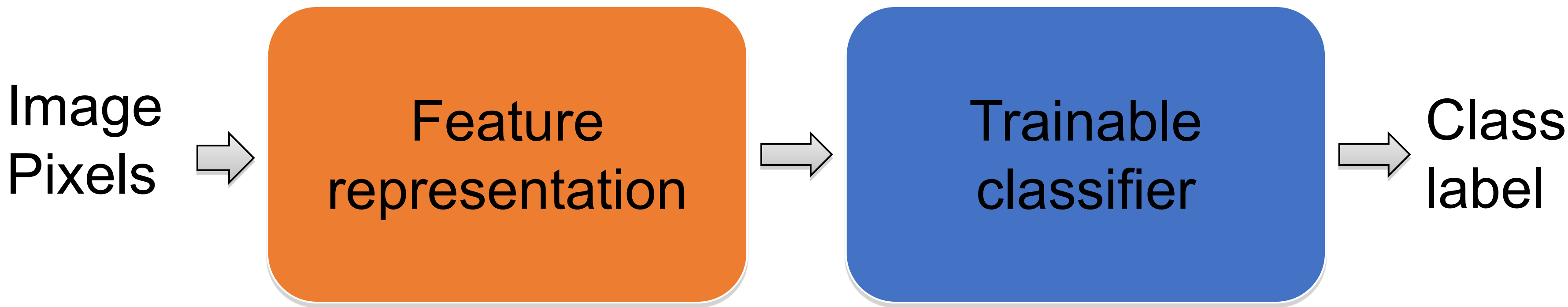
feature representation

- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing:** apply f to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$



Slide credit: D. Hoiem

Creating a “Classic” recognition pipeline



- Hand-crafted feature representation
- Off-the-shelf trainable classifier

Motivation 1: Bags of words

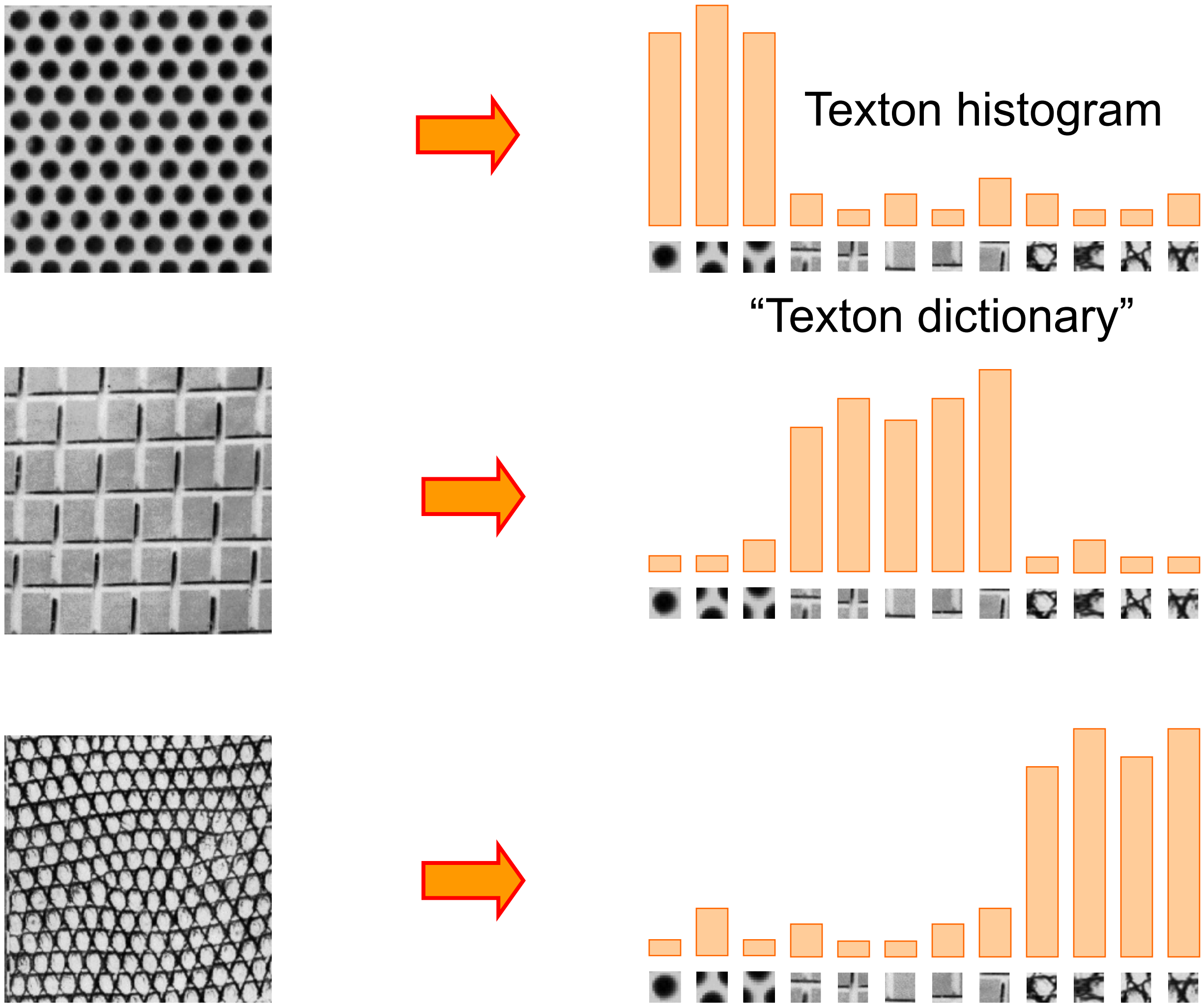
- Orderless document representation: frequencies of words from a dictionary
  Salton & McGill (1983)

US Presidential Speeches Tag Cloud
`http://chir.ag/projects/preztags/`

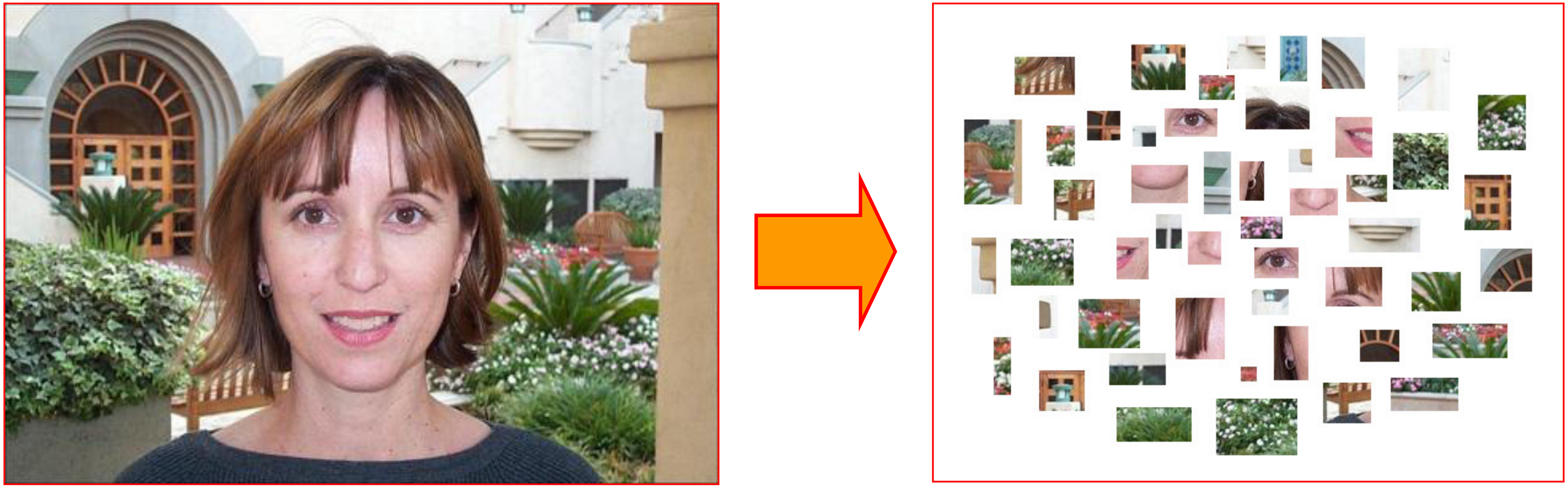
How to apply the same idea to images?

What are “visual words” ?

Motivation 2: Texture models

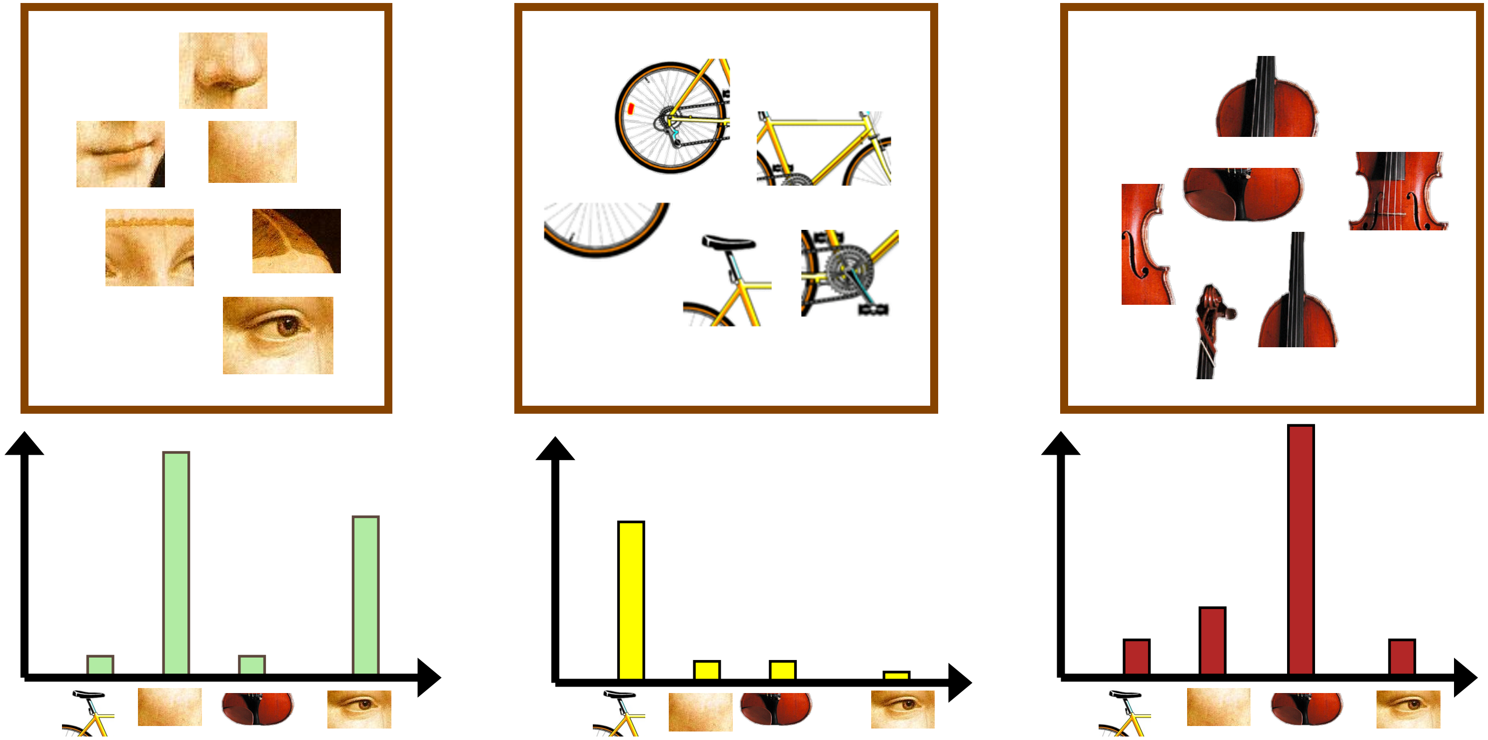


“Classic” representation: Bag of features

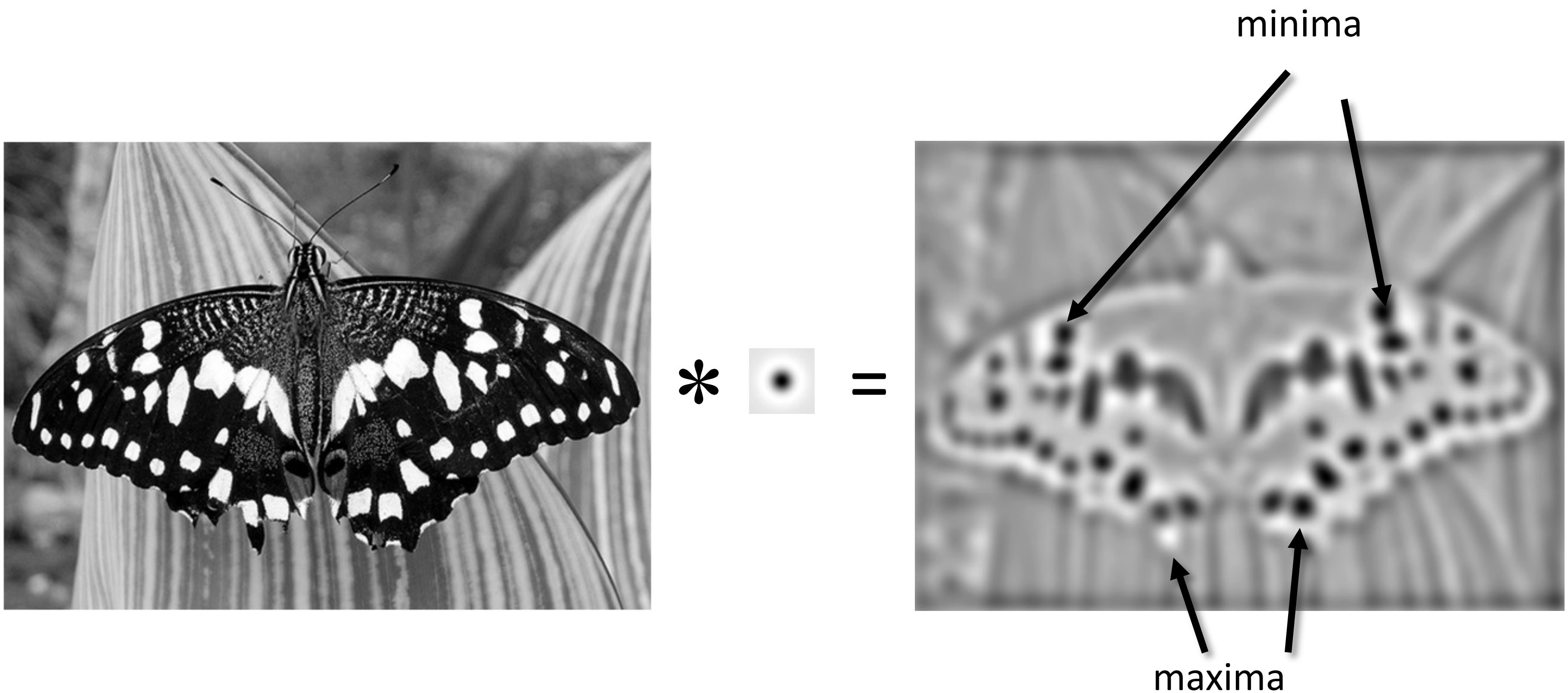


Bag of features: Outline

1. Extract local features
2. Learn "visual vocabulary"
3. Quantize local features using visual vocabulary
4. Represent images by frequencies of "visual words"

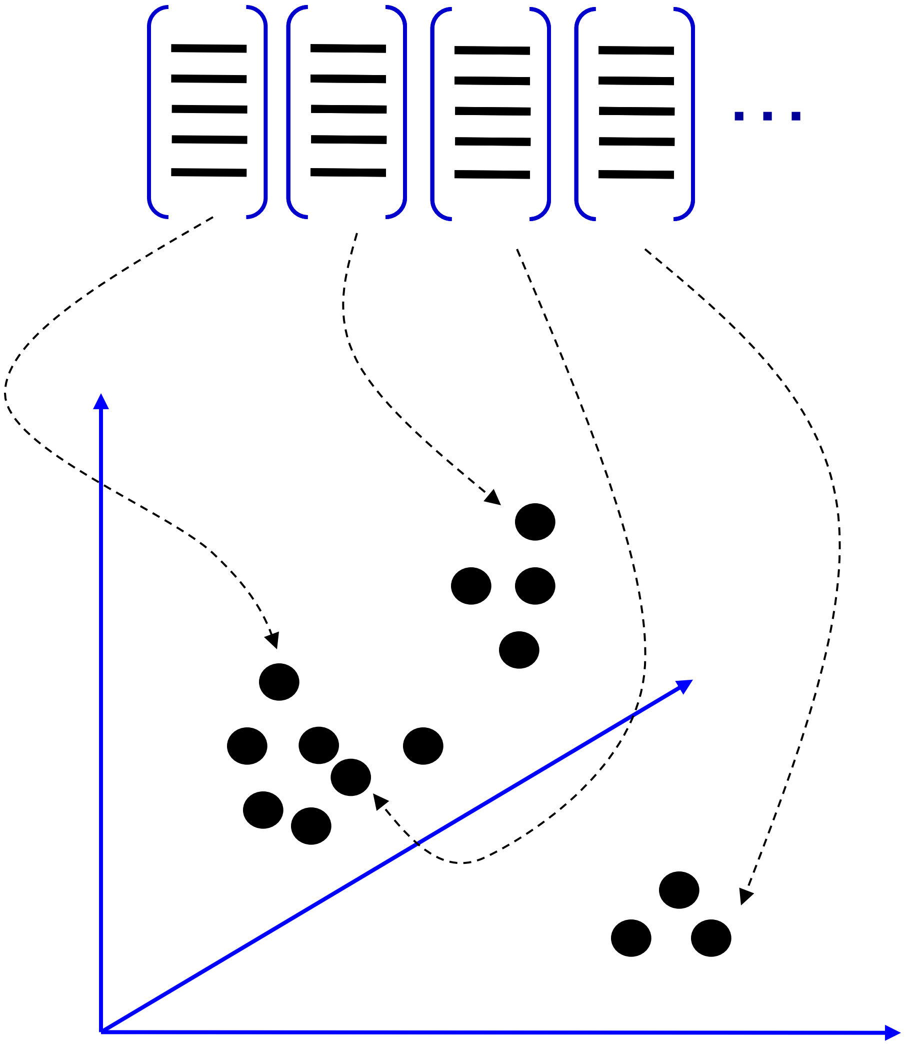


Blob detection



Find maxima *and minima* of blob filter response in space *and scale*

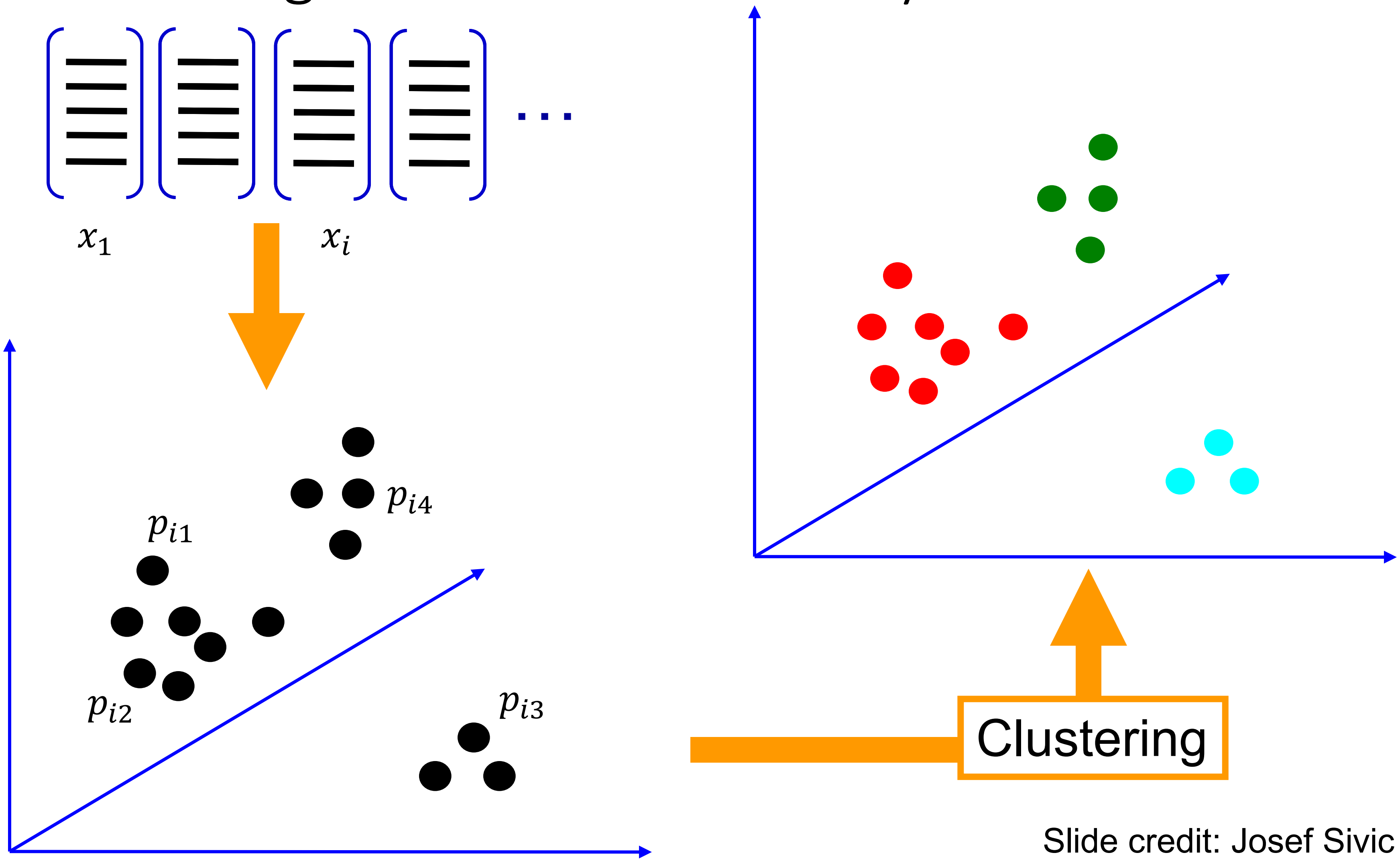
2. Learning the visual vocabulary



training set of images

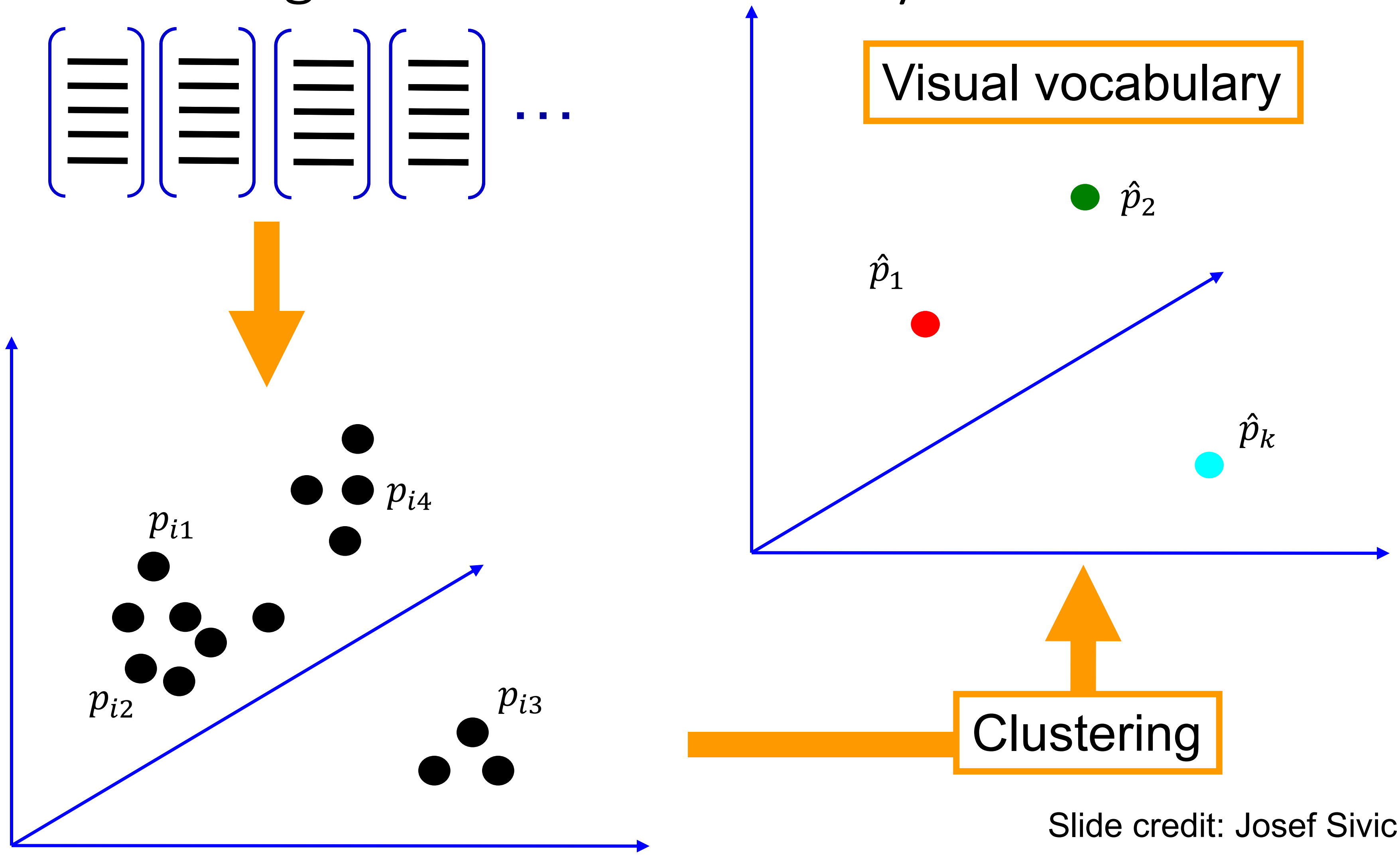
From each image [ ] patch [–] in the training set, extract descriptors (e.g., HOG, SIFT). This gives a vector in R^n which is used for clustering

2. Learning the visual vocabulary



Slide credit: Josef Sivic

2. Learning the visual vocabulary



Slide credit: Josef Sivic

Clustering

Given N vectors $x_1, \dots, x_N \in \mathbb{R}^n$, the goal is to partition them into k groups so that the vectors in the same group are close to one another

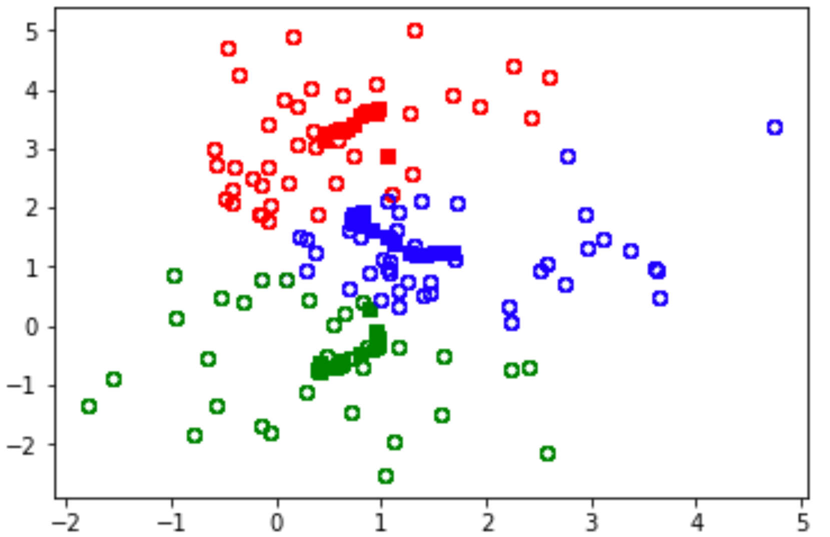
Examples: image compression (vectors are pixel values); patient clustering (patient attributes, tests);

$c_i \in \{1, \dots, k\}$ is the group x_i belongs to

$G_{c_i} \subseteq \{x_1, \dots, x_n\}$ group

z_{c_i} group representative

Clustering objective minimize $J_{clust} = \frac{1}{N}\sum_{i=1}^{N}|x_i - z_{c_i}|^2$ by choosing the groups $\{c_i\}$ and the representatives



from Boyd & Vandenb

Algorithm: Step 1

- Suppose the representatives $z_1, \dots, z_k$ are given, how do we assign the vectors $x_1, \dots, x_N$ to the k groups?
- Assign x_i to the nearest representative z_j
- Recall $\mathrm{J}_{clust} = \frac{1}{n}\sum_{i=1}^{N} \left| x_i - z_{c_i} \right|^2$
- $\min_j \frac{1}{N}\sum_{i=1}^{N} \left| x_i - z_j \right|^2 = \quad \frac{1}{N}\sum_{i=1}^{N} \min_j \left| x_i - z_j \right|^2$

Algorithm: Step 2

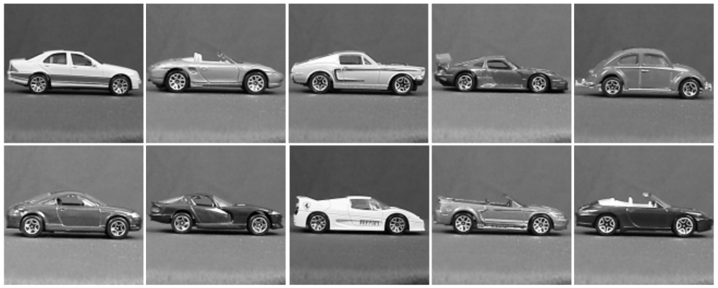
- Given the partition $G_1, \ldots, G_k$, how to choose the representatives $z_1, \ldots, z_k$?
- Choose z_j to minimize J_j, that is $z_j = \frac{1}{|G_j|}\sum_{i \in G_j} x_i$ the mean (centroid)
- $J_{clust} = J_1 + \cdots + J_k = \sum_{j=1}^{k} \frac{1}{|G_j|} \sum_{i \in G_j} \left| x_i - z_j \right|^2$

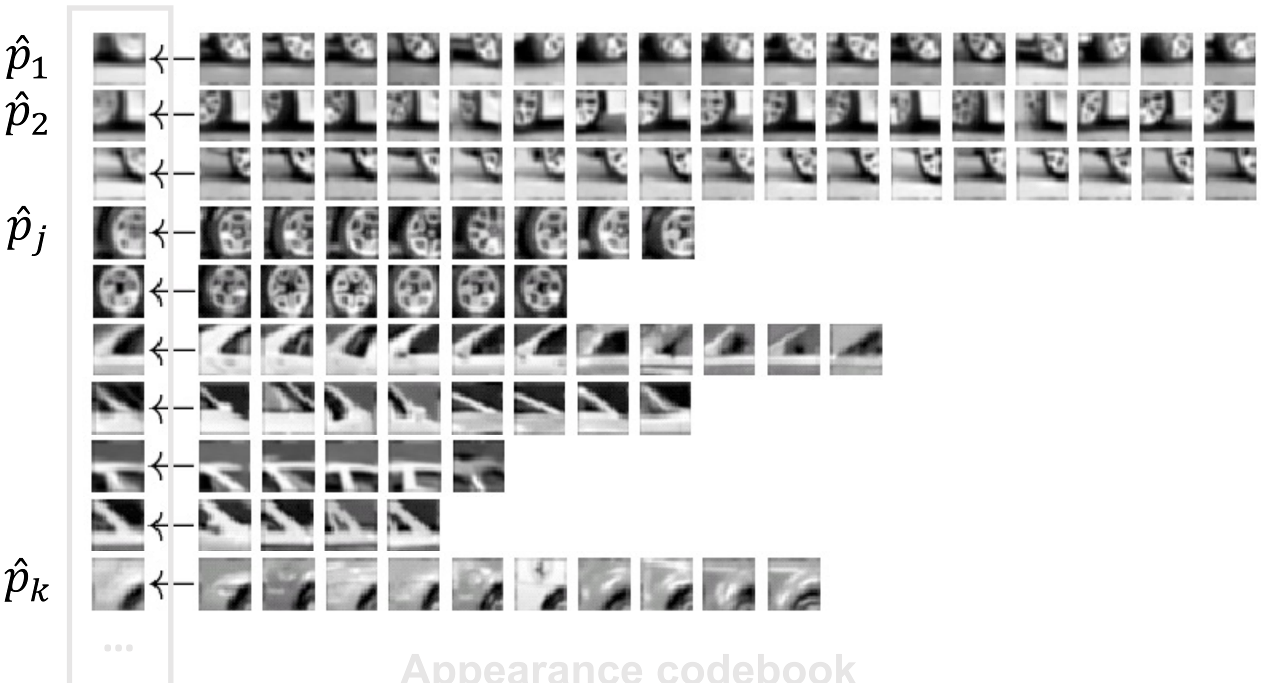
Algorithm: Combined

- alternate between updating the partition, then the representatives
- a famous algorithm called *k-means clustering*
- objective J_{clust} decreases in each step
- It terminates (Why?) but not necessarily to the global minimum (Why?)

given $x_1, \ldots, x_N \in \mathbb{R}^n$ and $z_1, \ldots, z_k \in \mathbb{R}^n$
repeat
    update partition: assign i to G_j, $j = \mathrm{argmin}_j \left|x_i - z_j\right|_2^2$
    update centroids: $z_j = \frac{1}{|P_j|} \sum_{i \in P_j} x_i$
until $z_1, \ldots, z_k$ stop changing

Recall: Visual vocabularies from cluster centers

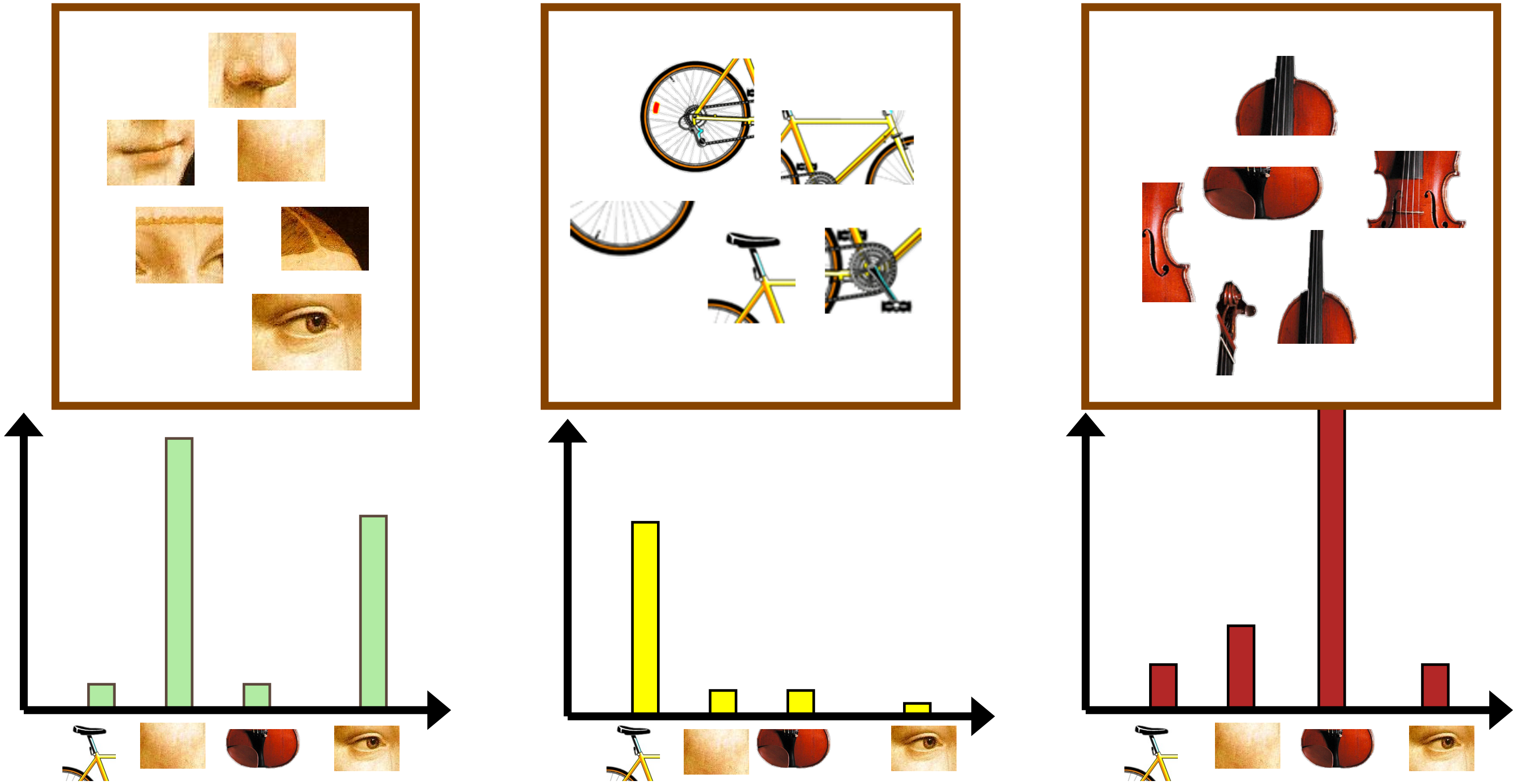




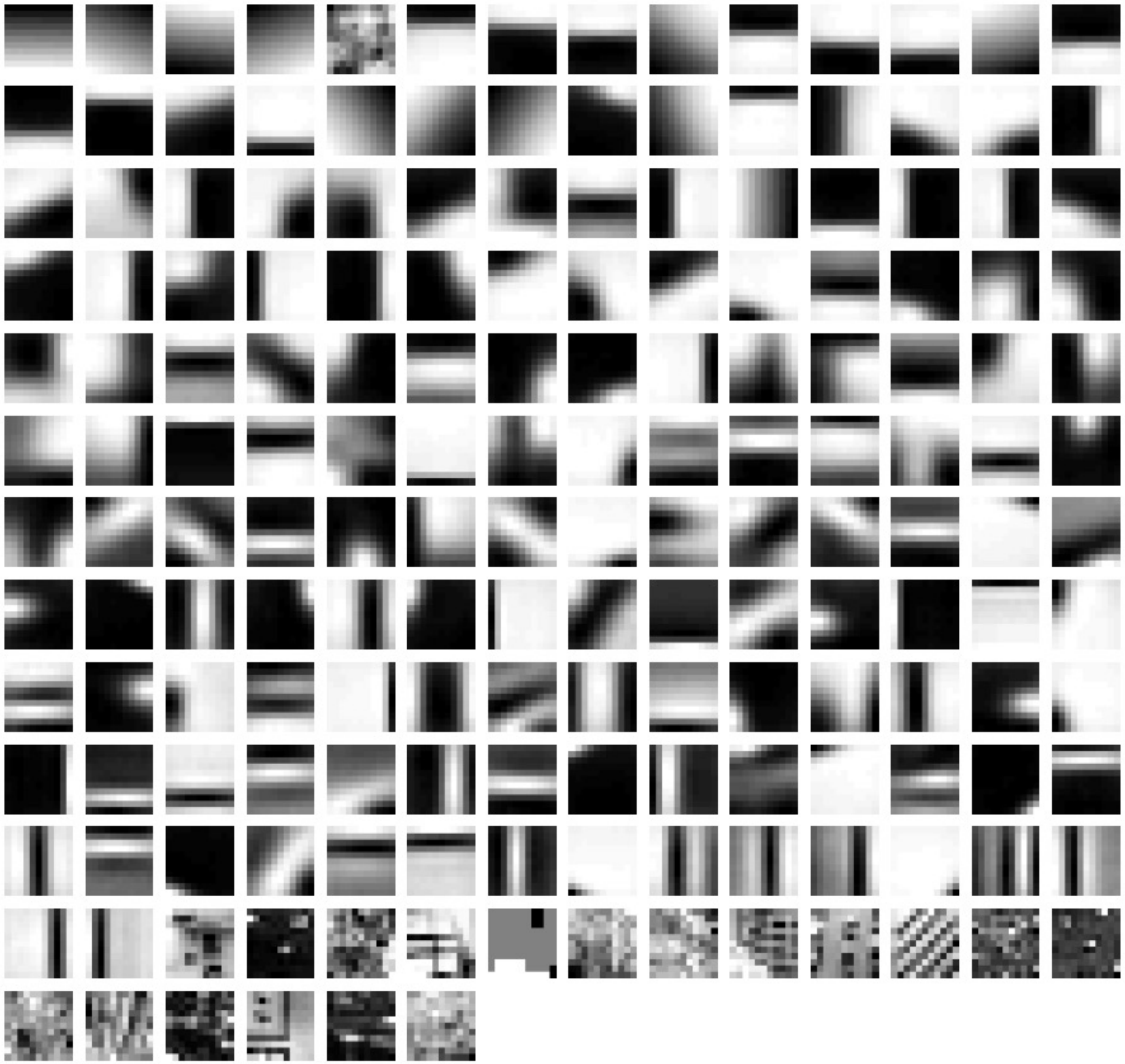
Source: B. Leibe

Bag of features: Outline

1. Extract local features
2. Learn "visual vocabulary"
3. **Quantize local features using visual vocabulary**
4. **Represent images by frequencies of "visual words"**



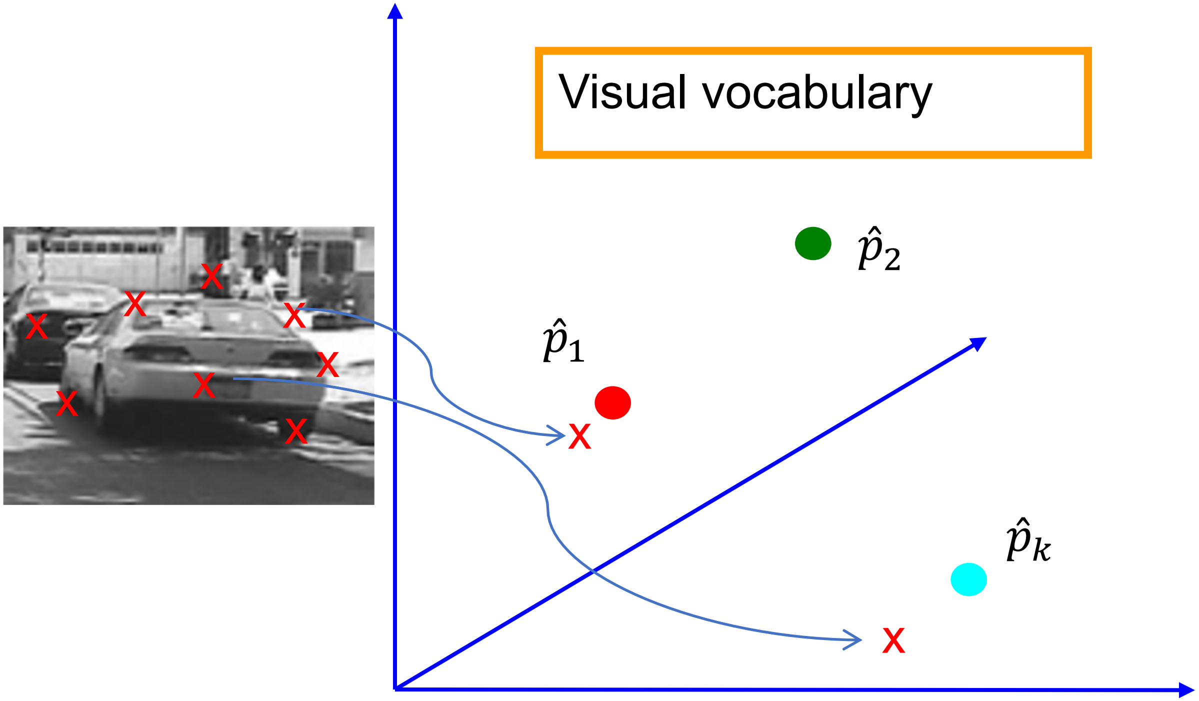
Example visual vocabulary



Fei-Fei et al. 2005

Image Representation

- For a query image

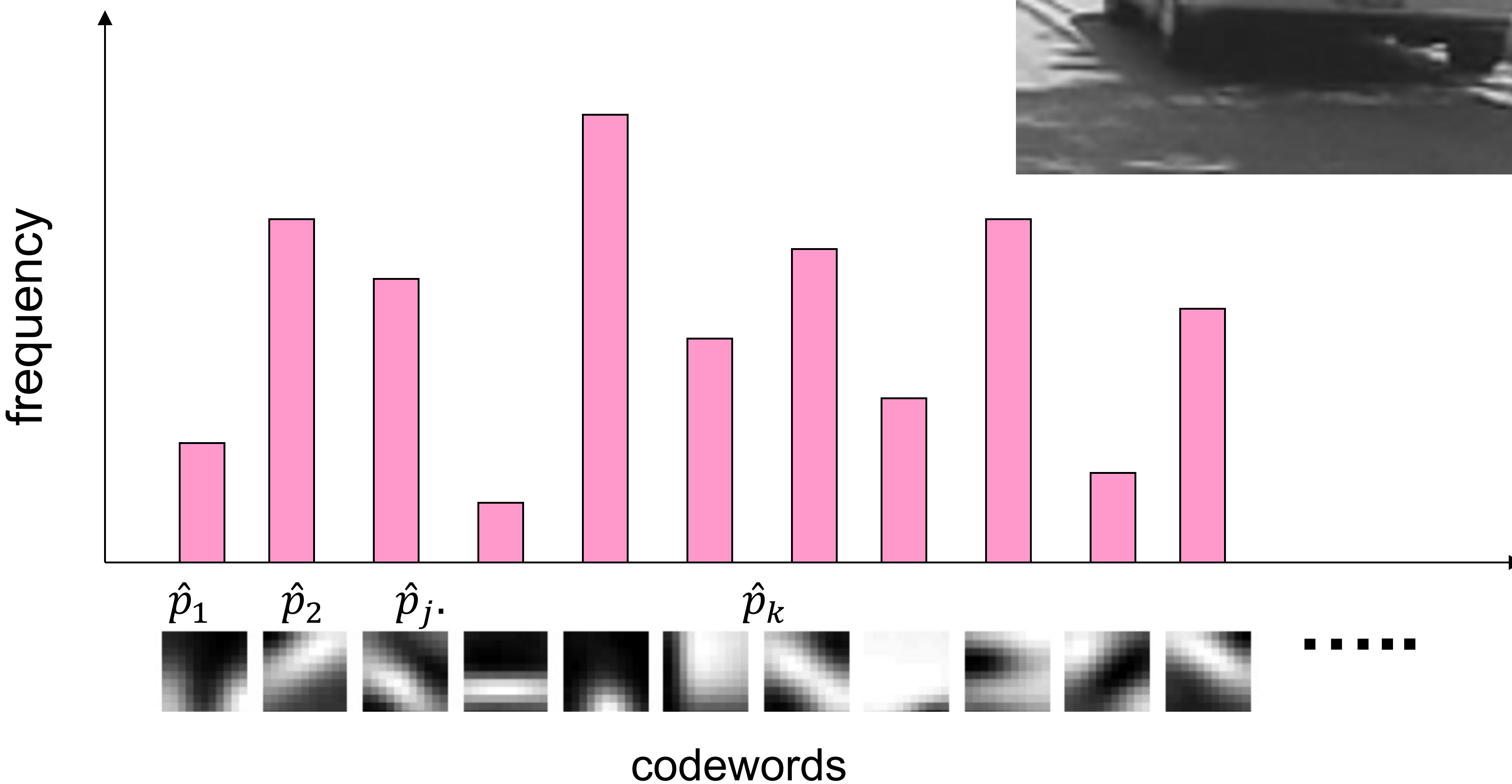


Extract features

Associate each feature with the nearest cluster center (visual word)

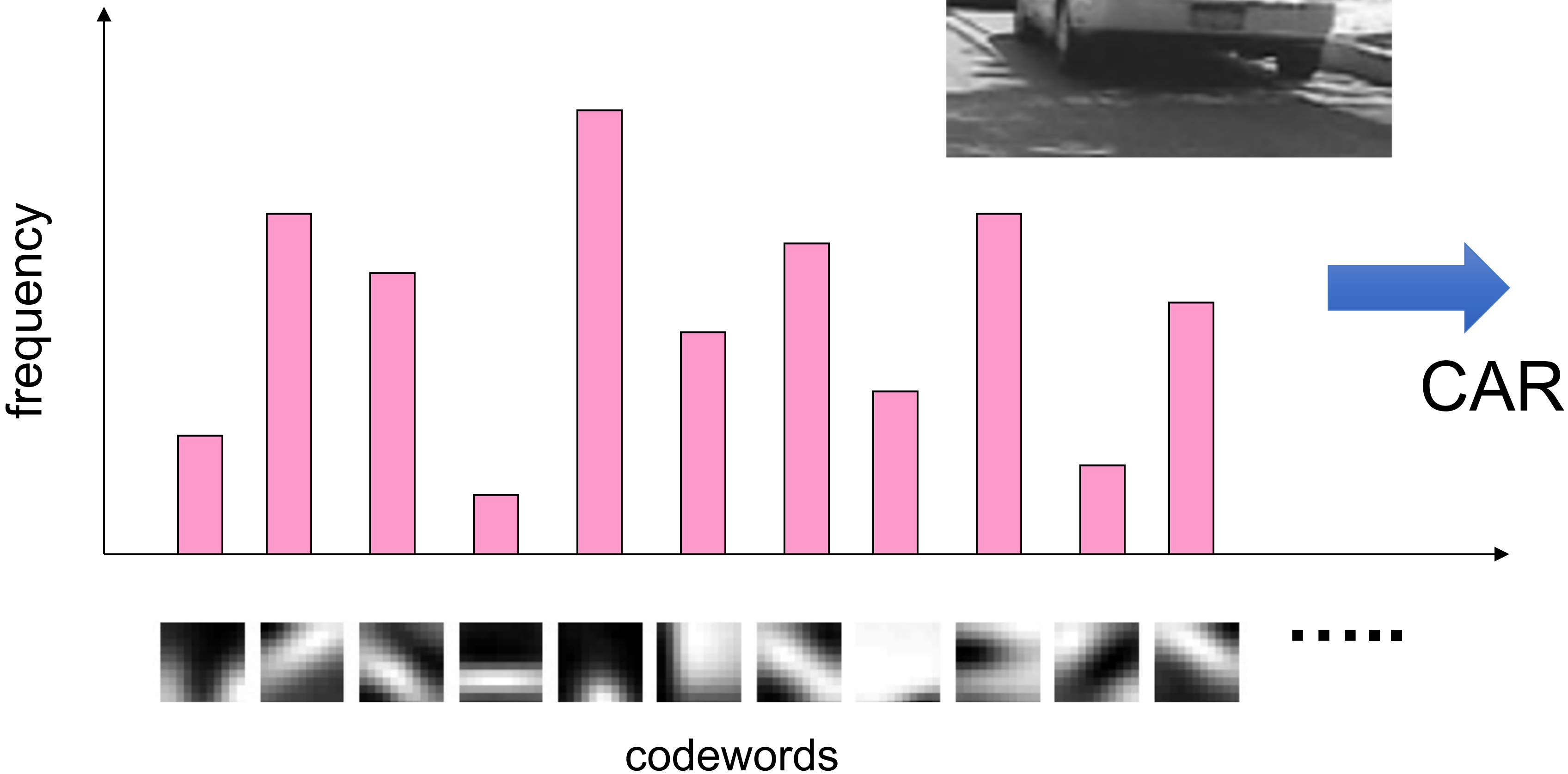
Accumulate visual word frequencies over the image

3. Image representation



source: Svetlana Lazebnik

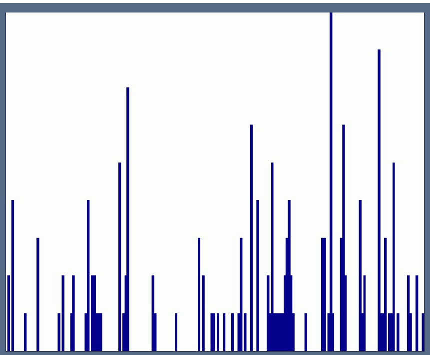
4. Image classification



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

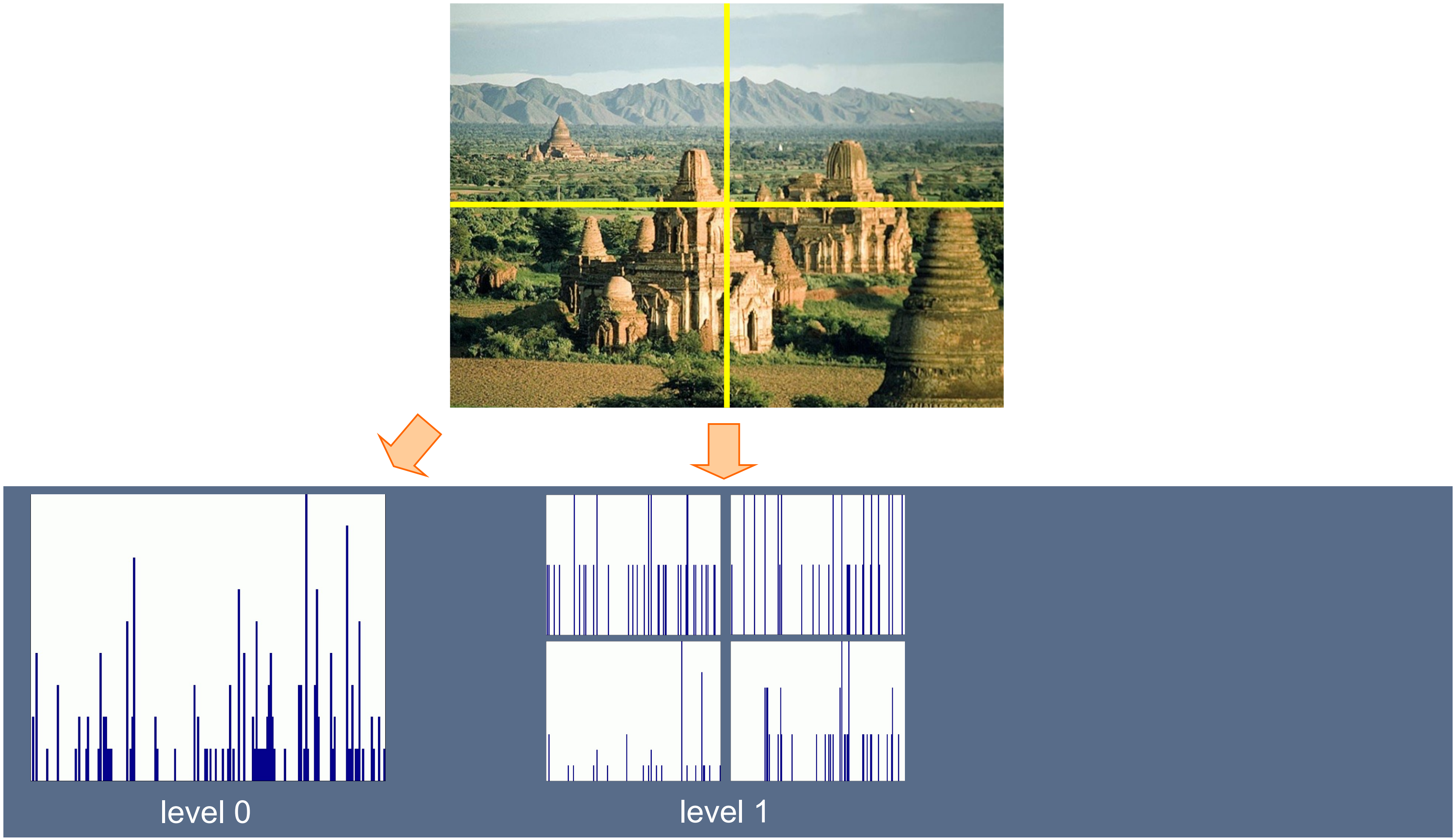
source: Svetlana Lazebnik

Spatial pyramids (orderless -> locally orderless)



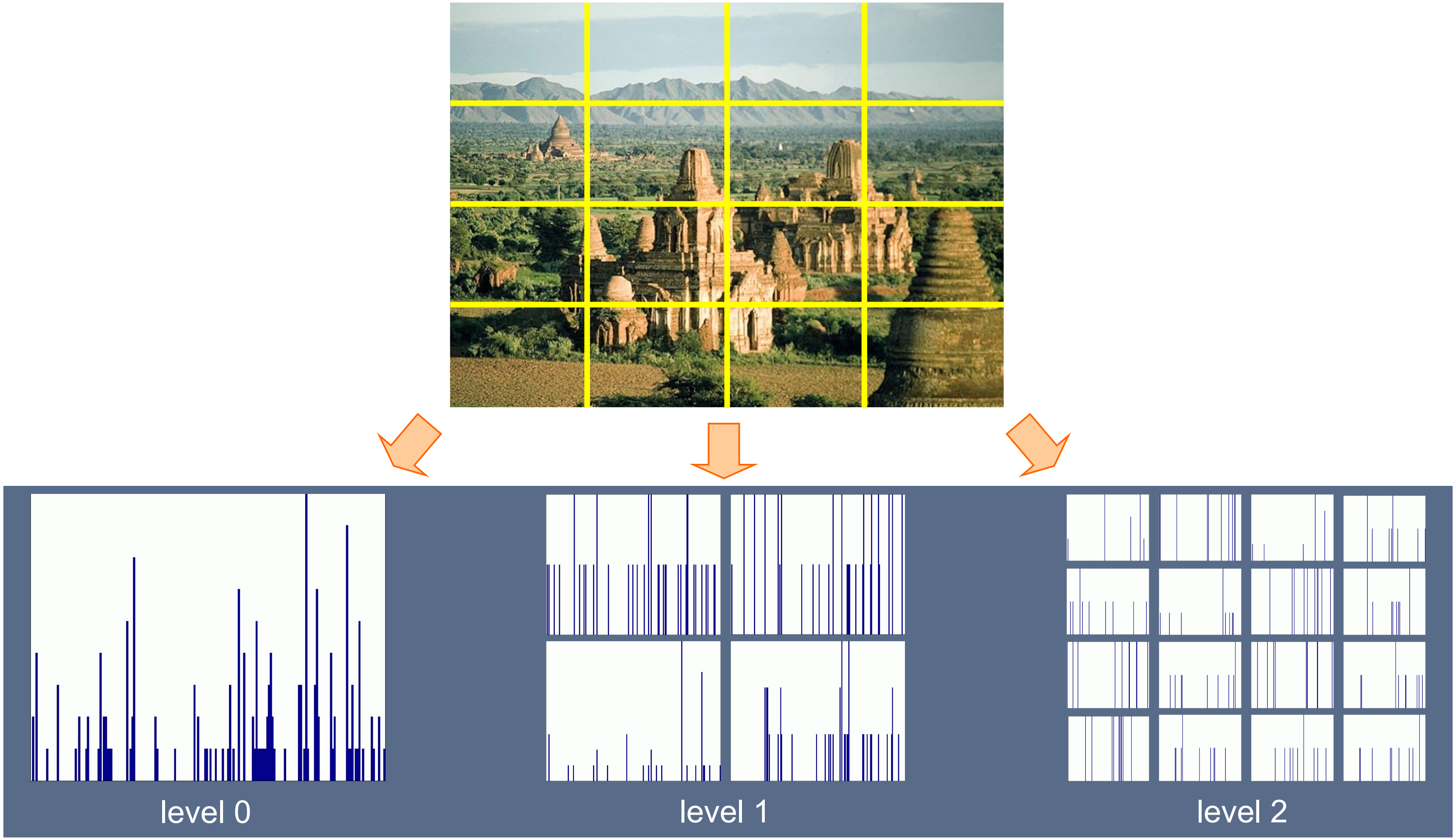
level 0

Spatial pyramids



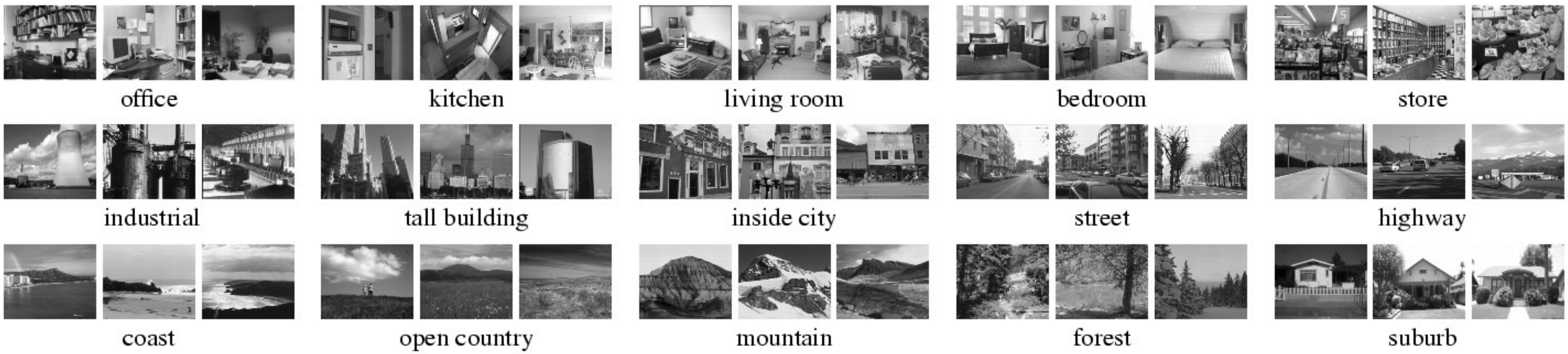
Lazebnik, Schmid & Ponce (CVPR 2006)

Spatial pyramids



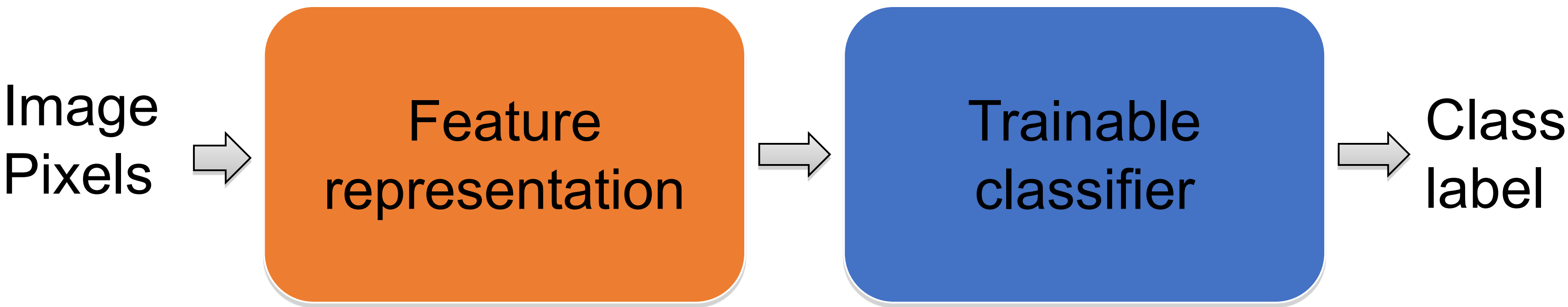
Lazebnik, Schmid & Ponce (CVPR 2006)

Spatial pyramids



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

"Classic" recognition pipeline

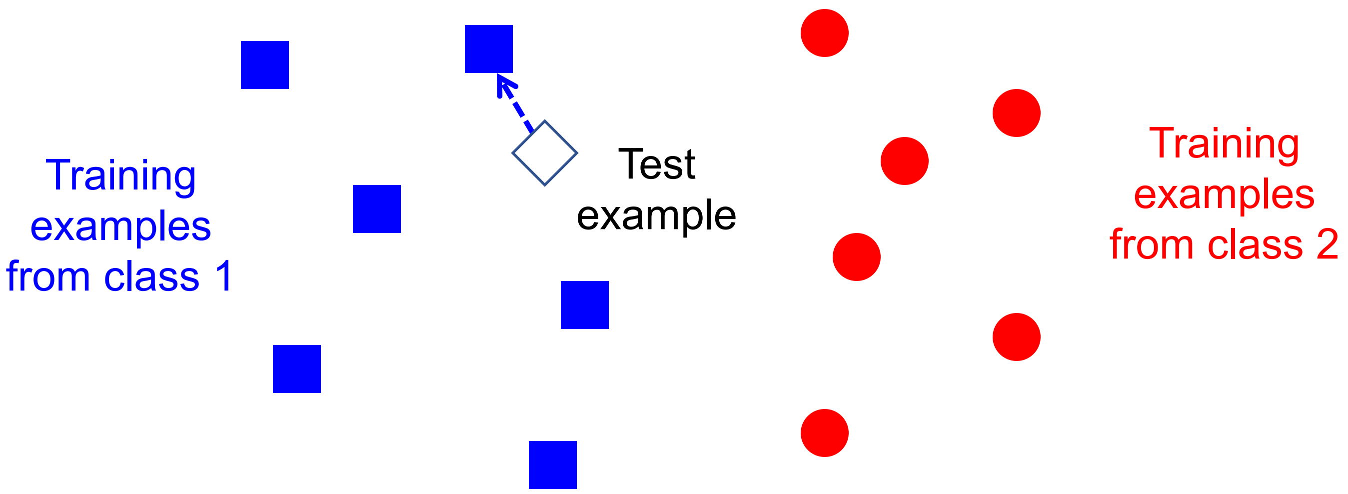


- Hand-crafted feature representation
- Off-the-shelf trainable classifier

Classification

Given a new image, and the vector of trained visual word histograms, how to classify the new image?

Classifiers: Nearest neighbor



f($\mathbf{x}$) = label of the training example nearest to $\mathbf{x}$

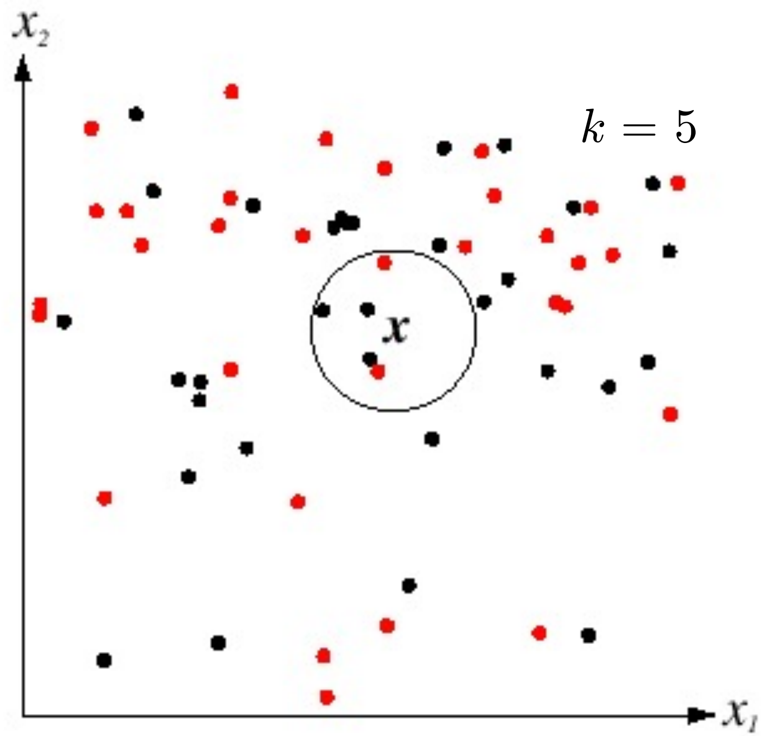
- All we need is a distance or similarity function for our inputs
- No training required!

Functions for comparing histograms

- L1 distance: $$D(h_1,h_2)=\sum_{i=1}^{N}|\,h_1(i)-h_2(i)\,|$$
- χ^2 distance: $$D(h_1,h_2)=\sum_{i=1}^{N}\frac{\left(h_1(i)-h_2(i)\right)^2}{h_1(i)+h_2(i)}$$
- Quadratic distance (*cross-bin distance*): $$D(h_1,h_2)=\sum_{i,j}A_{ij}(h_1(i)-h_2(j))^2$$
- Histogram intersection (similarity function): $$I(h_1,h_2)=\sum_{i=1}^{N}\min(h_1(i),h_2(i))$$

K-nearest neighbor classifier

- For a new point, find the k closest points from training data
- Vote for class label with labels of the k points



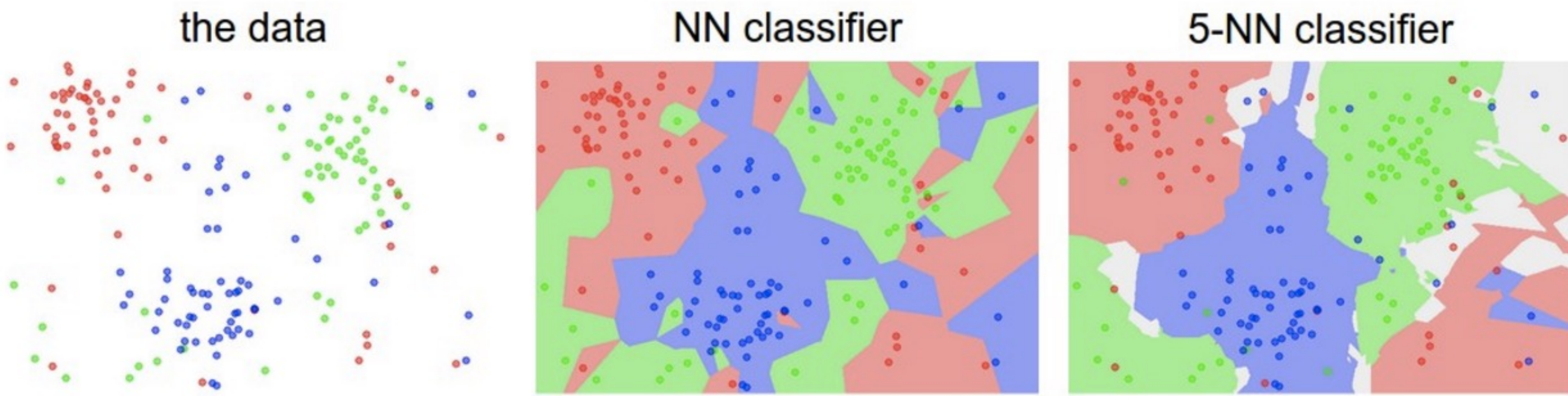
Effects of scaling

- Some components of the vector with large values may influence the classification more than others

- Normalize vectors
  - x_{ij} value for the i^{th} sample and j^{th} feature
  - μ_j mean of all x_{ij} for feature j
  - σ_j standard deviation of all x_{ij} over all input samples

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

K-nearest neighbor classifier

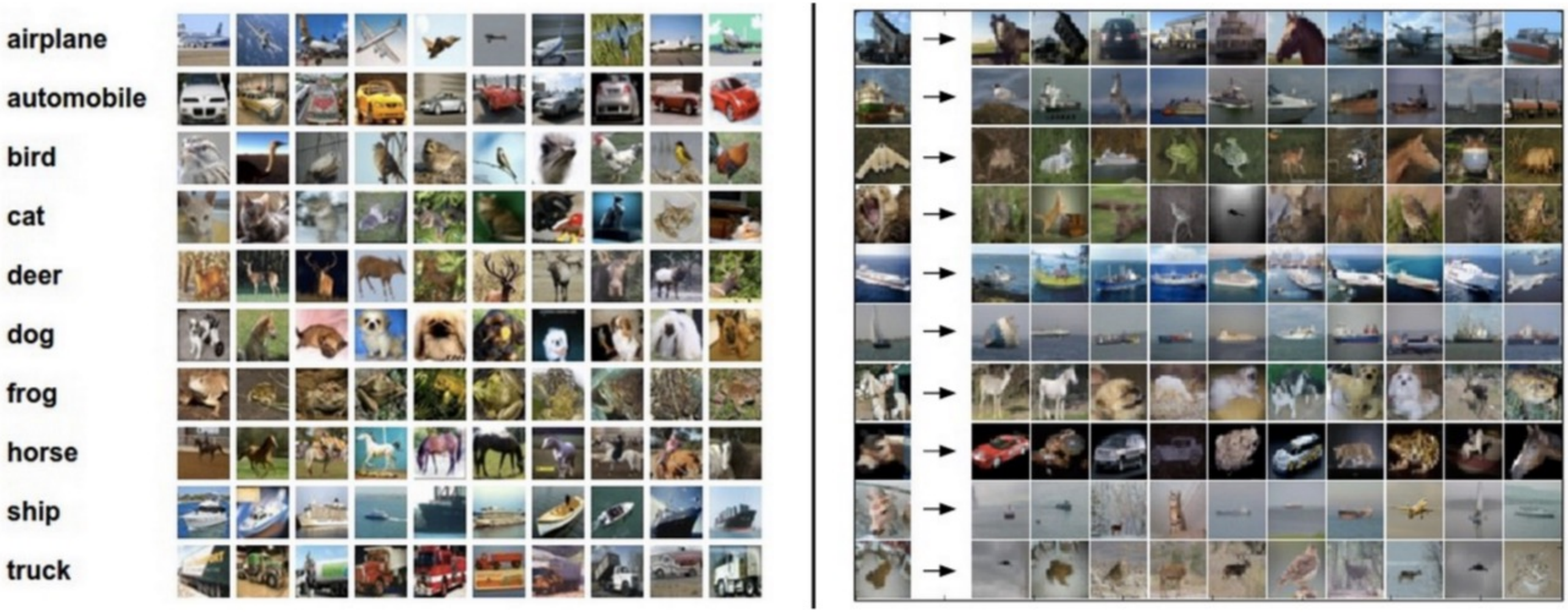


- 2d points, and 3 classes. White regions are “ambiguous”
- Which classifier is more robust to *outliers*?

Credit: Andrej Karpathy, http://cs231n.github.io/classification/

Hyperparameters for K-NN

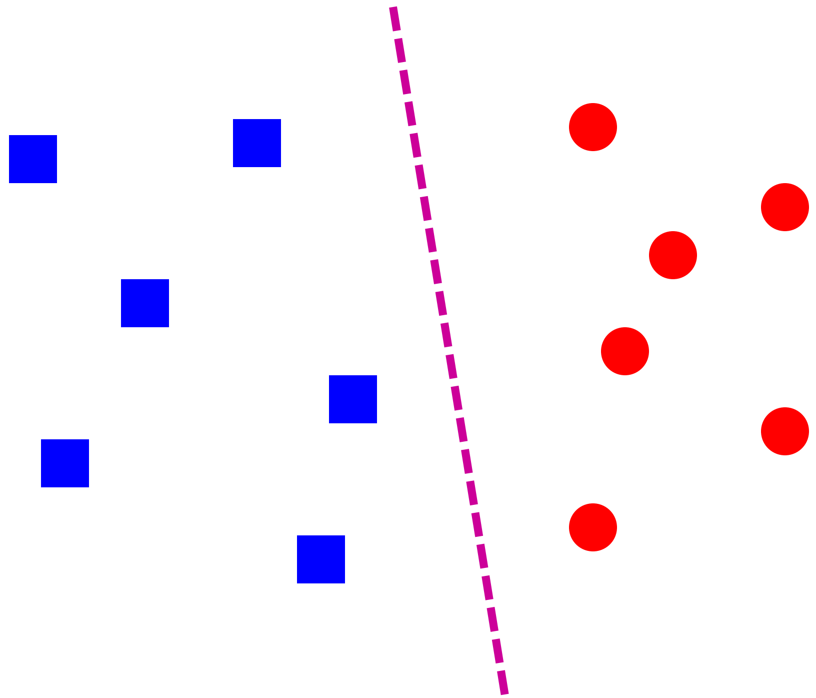
K-nearest neighbor classifier



Left: Example images from the CIFAR-10 dataset. Right: first column shows a few test images and next to each we show the top 10 nearest neighbors in the training set according to pixel-wise difference.

Credit: Andrej Karpathy, http://cs231n.github.io/classification/

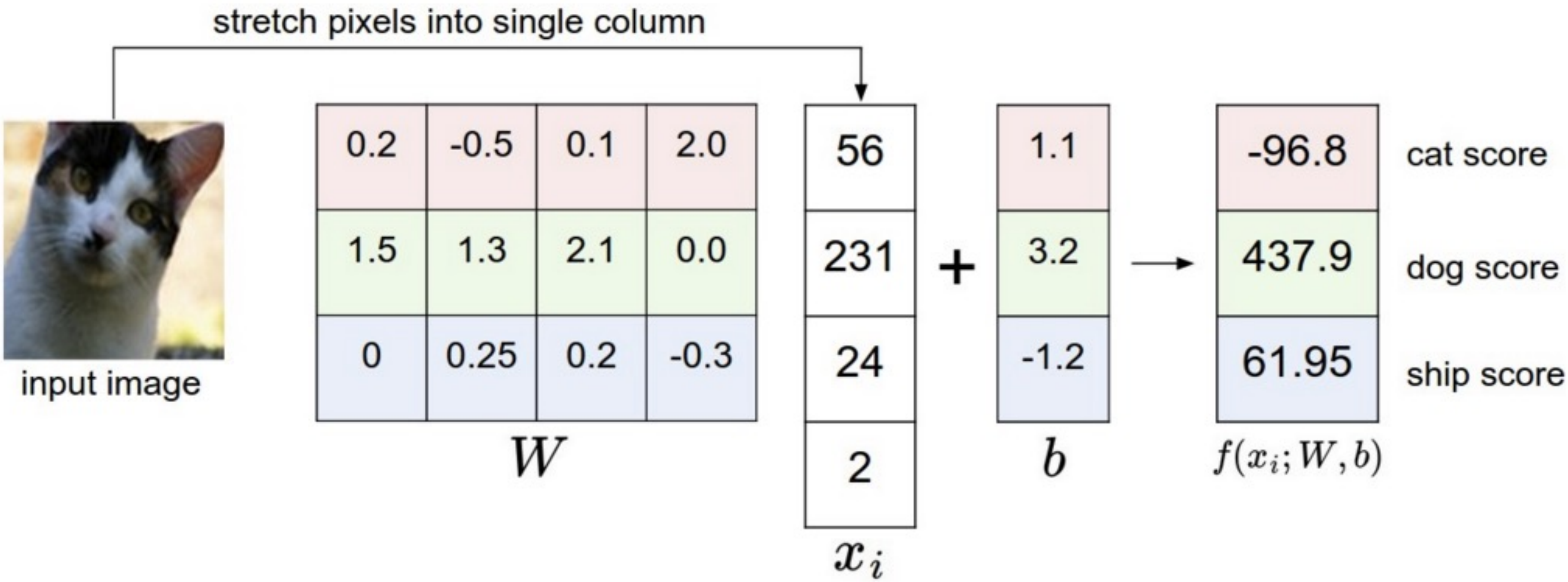
Linear classifiers



- Find a *linear function* to separate the classes:

$$f(\mathbf{x}) = \mathrm{sgn}(\mathbf{w} \cdot \mathbf{x} + b)$$

Visualizing linear classifiers



Source: Andrej Karpathy, http://cs231n.github.io/linear-classify/

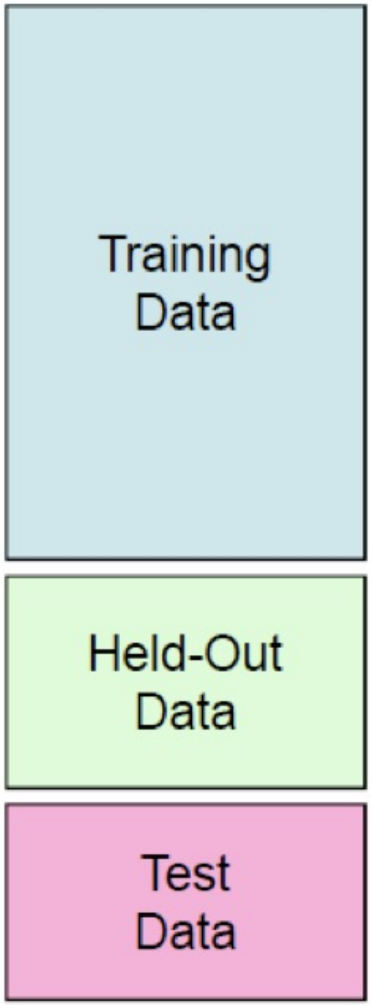
Nearest neighbor vs. linear classifiers

- NN pros:
  - Simple to implement
  - Decision boundaries not necessarily linear
  - Works for any number of classes
  - *Nonparametric* method
- NN cons:
  - Need good distance function
  - Slow at test time
- Linear pros:
  - Low-dimensional *parametric* representation
  - Very fast at test time
- Linear cons:
  - Works for two classes
  - How to train the linear function?
  - What if data is not linearly separable?

Best practices for training classifiers

- Goal: obtain a classifier with **good generalization** or performance on never before seen data

1. Learn *parameters* on the *training set*
2. Tune *hyperparameters* (implementation choices) on the *held out validation set*
3. Evaluate performance on the *test set*
   - Crucial: do not peek at the test set when iterating steps 1 and 2!



Kahoot!

- https://play.kahoot.it/v2/?quizId=4a8862f8-402c-42f9-bb61-2d63336a1171

What's the big deal?

Baidu admits cheating in international supercomputer competition

ZDNet

Baidu recently apologised for violating the rules of an international supercomputer test in May, when the Chinese search engine giant claimed to beat both Google and Microsoft on the ImageNet image-recognition test.

By Cyrus Lee | June 10, 2015 -- 00:15 GMT (17:15 PDT) | Topic: China

TECHNOLOGY

The New York Times

Computer Scientists Are Astir After Baidu Team Is Barred From A.I. Competition

By JOHN MARKOFF JUNE 3, 2015

engadget

Baidu caught gaming recent supercomputer performance test

by Andrew Tarantola | @terrortola | June 3rd 2015 At 11:09pm

IMAGENET Large Scale Visual Recognition Challenge (ILSVRC)

Date: June 2, 2015

Dear ILSVRC community,

This is a follow up to the announcement on May 19, 2015 with some more details and the status of the test server.

During the period of November 28th, 2014 to May 13th, 2015, there were at least 30 accounts used by a team from Baidu to submit to the test server at least 200 times, far exceeding the specified limit of two submissions per week. This includes short periods of very high usage, for example with more than 40 submissions over 5 days from March 15th, 2015 to March 19th, 2015. Figure A below shows submissions from ImageNet accounts known to be associated with the team in question. Figure B shows a comparison to the activity from all other accounts.

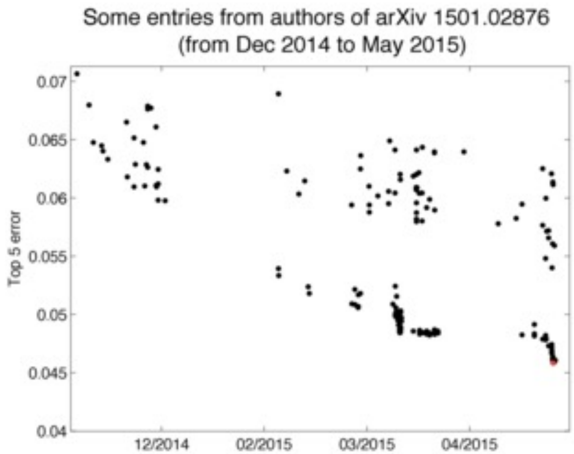


Figure A

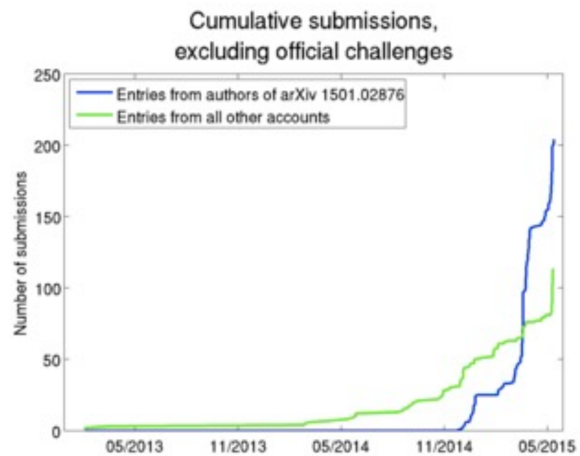


Figure B

The results obtained during this period are reported in a recent arXiv paper. Because of the violation of the regulations of the test server, these results may not be directly comparable to results obtained and reported by other teams. To make this clear, by exploiting the ability to test many slightly different solutions on the test server it is possible to 1) select the best out of a set of very similar solutions based on test performance and achieve a small but potentially significant advantage and 2) choose methods for further research and development based directly on the test data instead of using only the training and validation data for such choices.

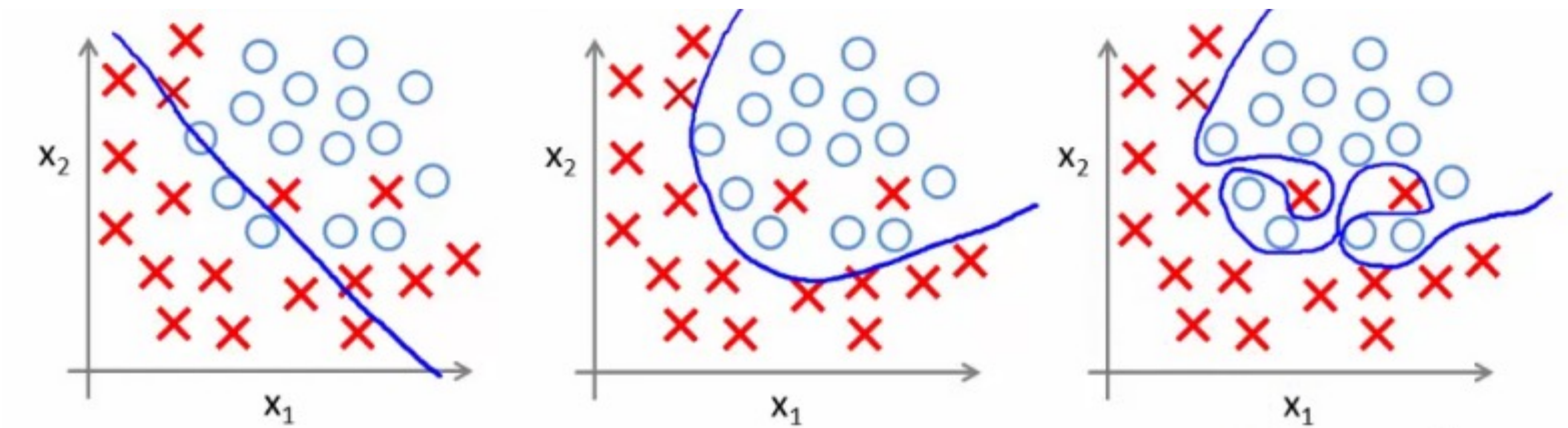
http://www.image-net.org/challenges/LSVRC/announcement-June-2-2015

Bias-variance tradeoff

- Prediction error of learning algorithms has two main components:
  - **Bias:** error due to simplifying model assumptions
  - **Variance:** error due to randomness of training set
- **Bias-variance tradeoff** can be controlled by turning "knobs" that determine model complexity

High bias, low variance

Low bias, high variance



Underfitting and overfitting

- **Underfitting:** training and test error are both *high*
  - Model does an equally poor job on the training and the test set
  - The model is too "simple" to represent the data or the model is not trained well
- **Overfitting:** Training error is *low* but test error is *high*
  - Model fits irrelevant characteristics (noise) in the training data
  - Model is too complex or amount of training data is insufficient

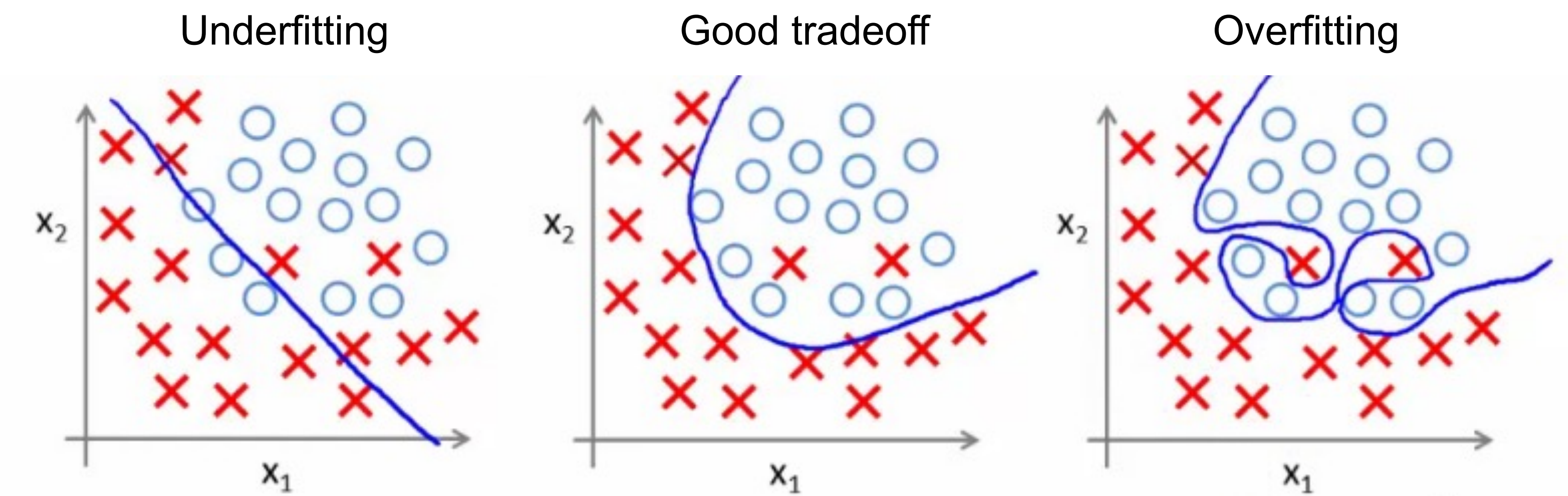


Figure source