Bag-of-features for category classification

Cordelia Schmid

Category recognition

- Image classification: assigning a class label to the image

Car: present

Cow: present

Bike: not present

Horse: not present

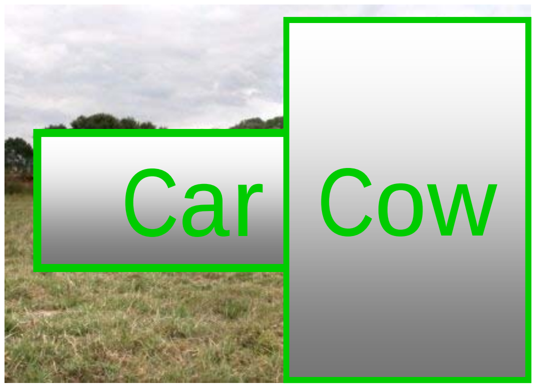
...

Category recognition

- Image classification: assigning a class label to the image

Car: present
Cow: present
Bike: not present
Horse: not present
...

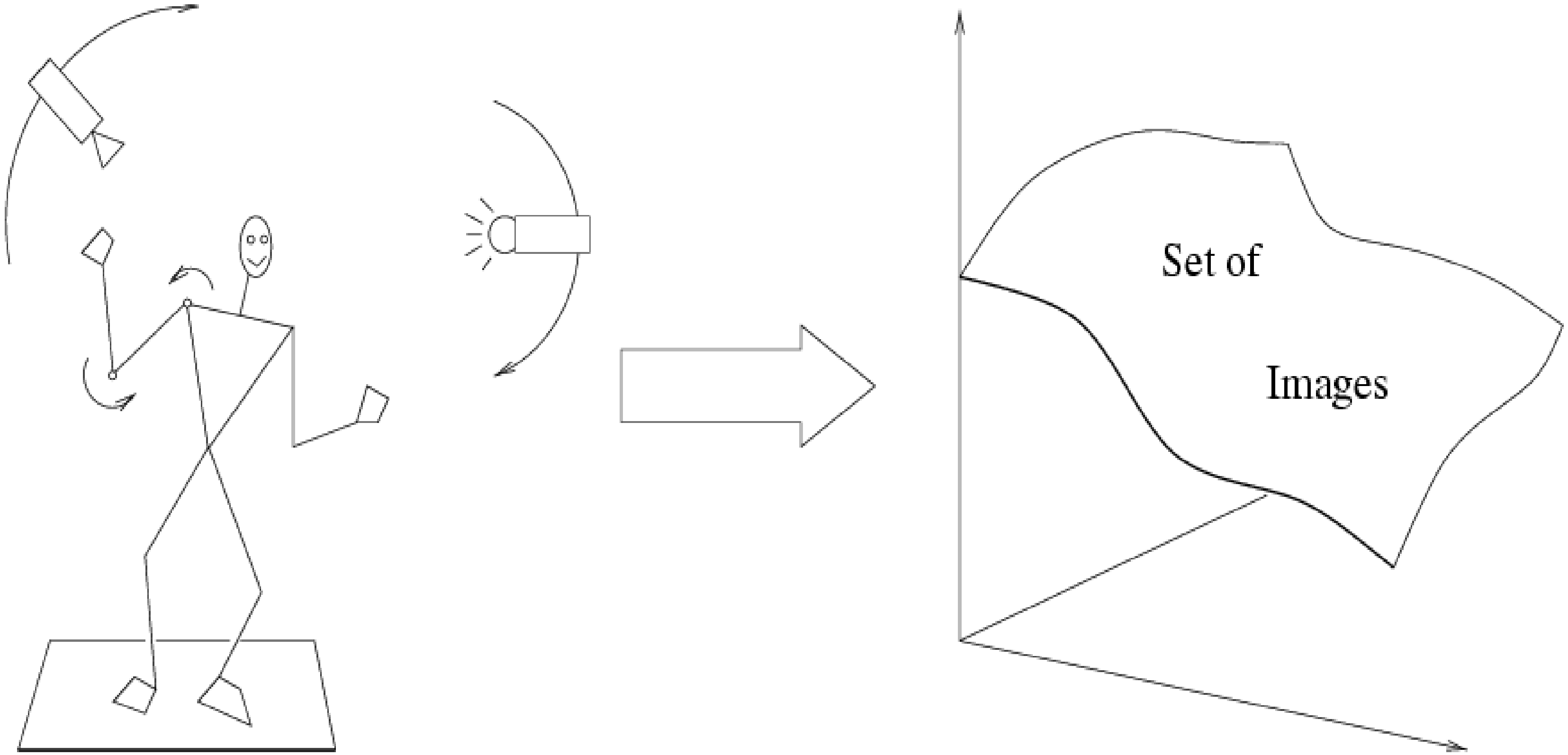
- Object localization: define the location and the category



Location

Category

Difficulties: within object variations



Variability: Camera position, Illumination, Internal parameters

Within-object variations

Difficulties: within-class variations






Category recognition

- Image classification: assigning a class label to the image

Car: present
Cow: present
Bike: not present
Horse: not present
...

- Supervised scenario: given a set of training images

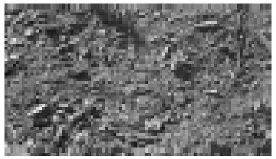
Image classification

- Given

  Positive training images containing an object class



  Negative training images that don't

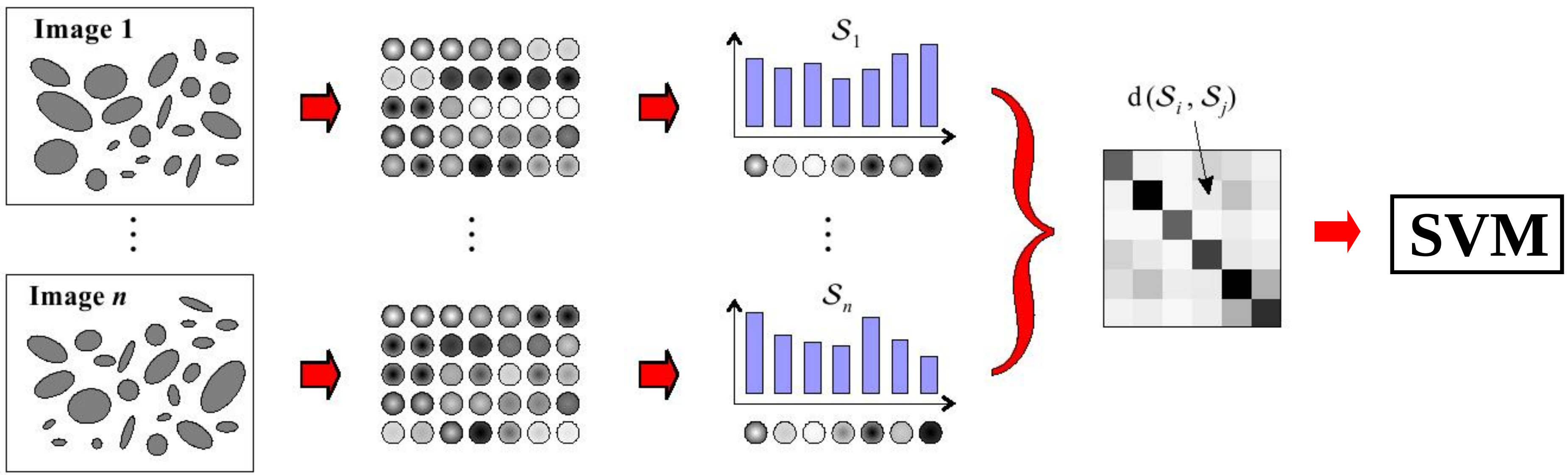


- Classify

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

?

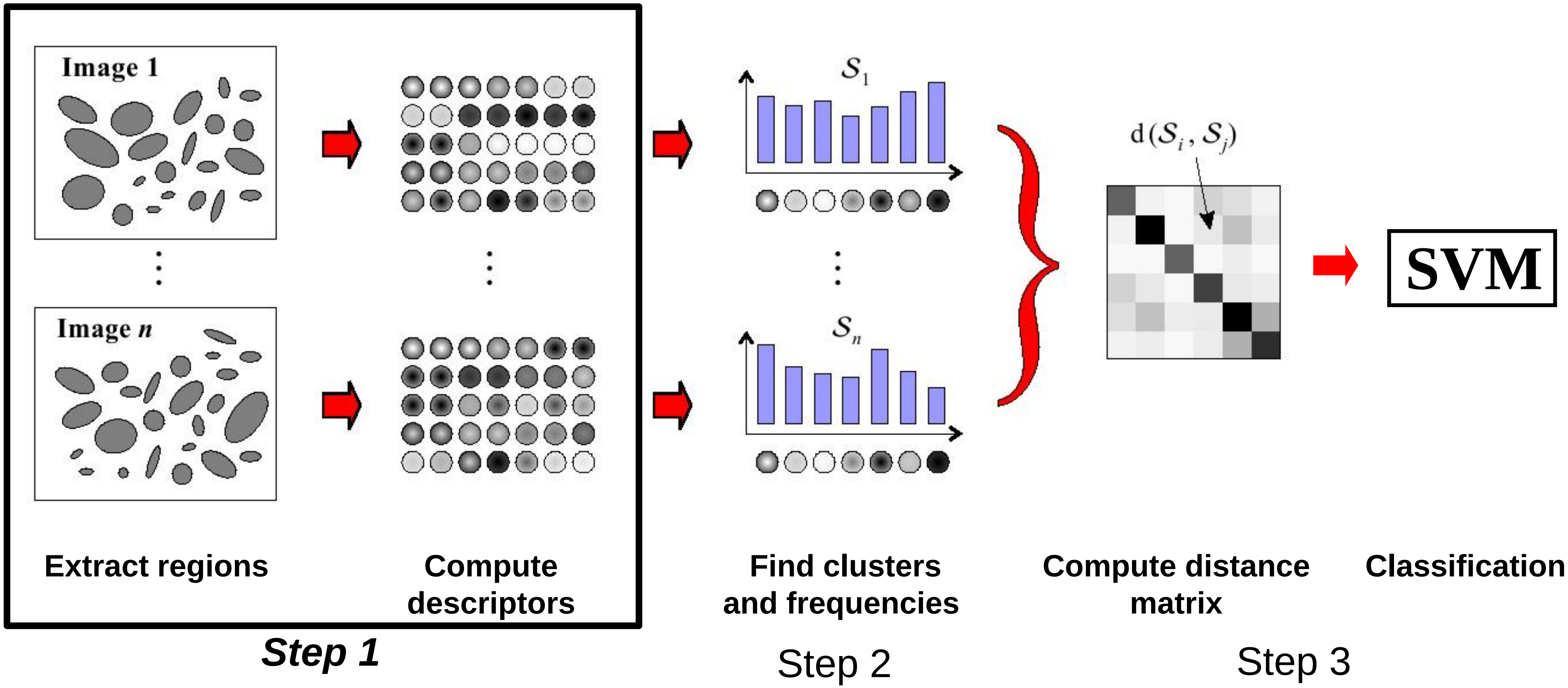
Bag-of-features for image classification



Extract regions | **Compute descriptors** | **Find clusters and frequencies** | **Compute distance matrix** | **Classification**

[Csurka et al. WS'2004], [Nowak et al. ECCV'06], [Zhang et al. IJCV'07]

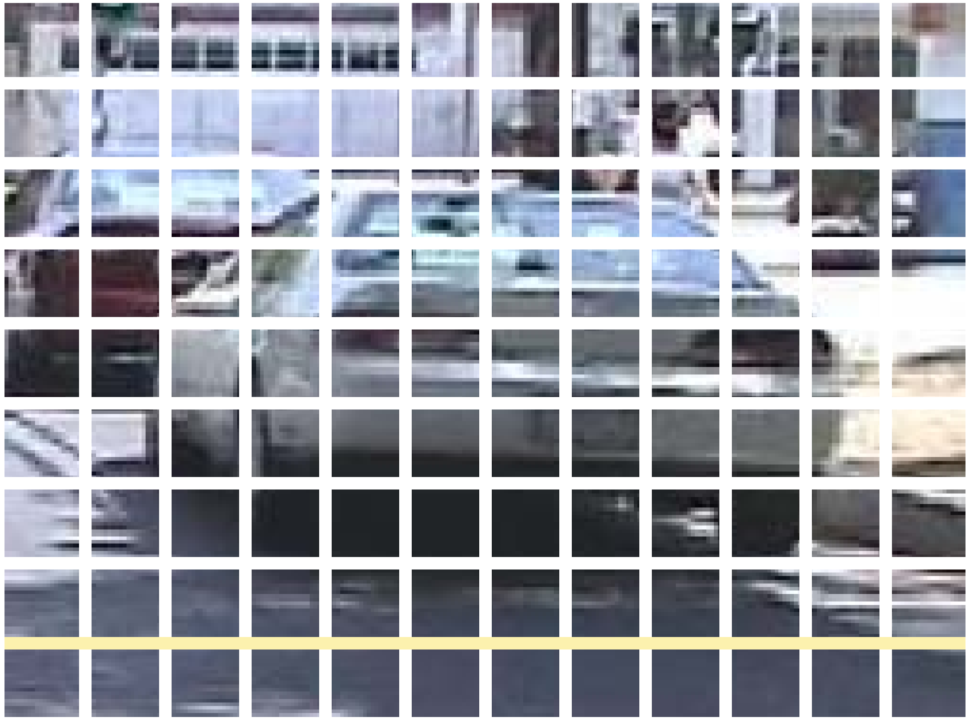
Bag-of-features for image classification



Step 1: feature extraction

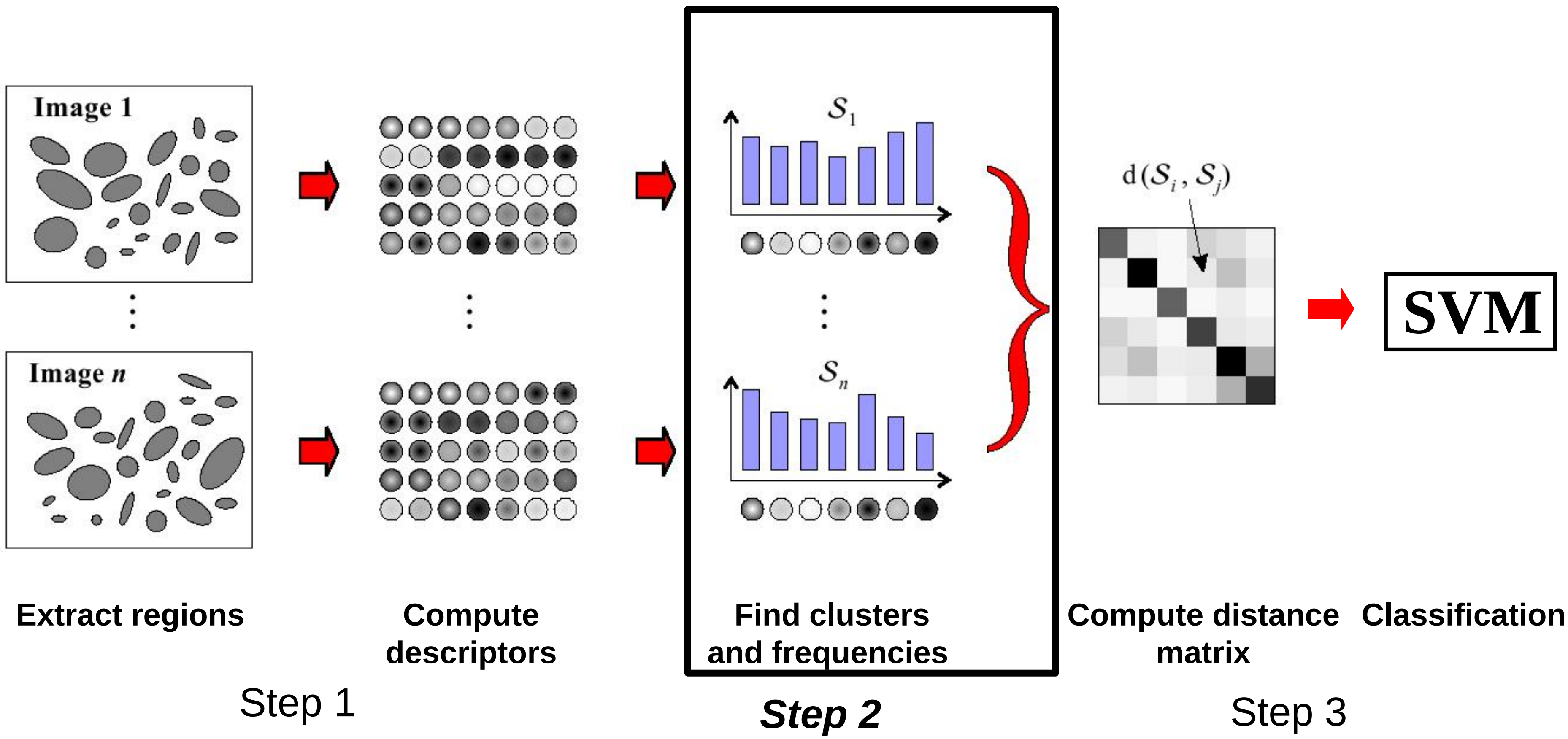
- Scale-invariant image regions + SIFT
  - Affine invariant regions give “too” much invariance
  - Rotation invariance for many realistic collections “too” much invariance
- Dense descriptors
  - Improve results in the context of categories (for most categories)
  - Interest points do not necessarily capture “all” features
- Color-based descriptors

Dense features

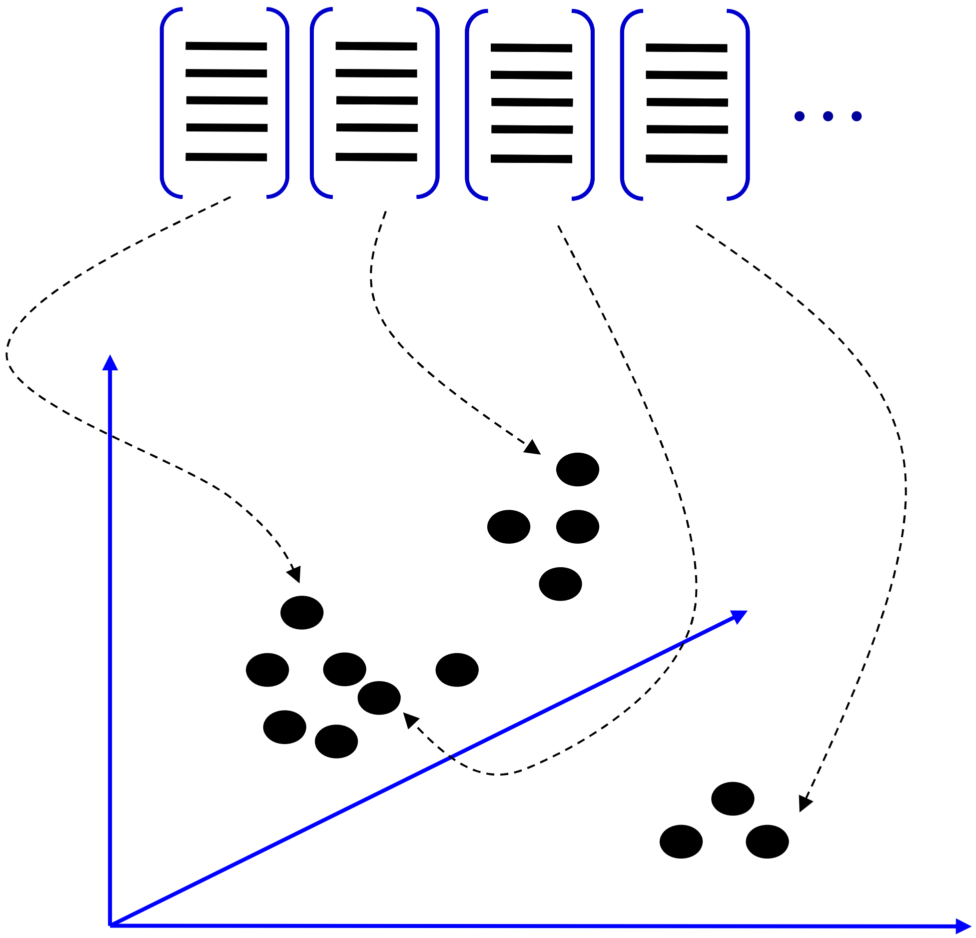


- Multi-scale dense grid: extraction of small overlapping patches at multiple scales
- Computation of the SIFT descriptor for each grid cells
- Exp.: Horizontal/vertical step size 3-6 pixel, scaling factor of 1.2 per level

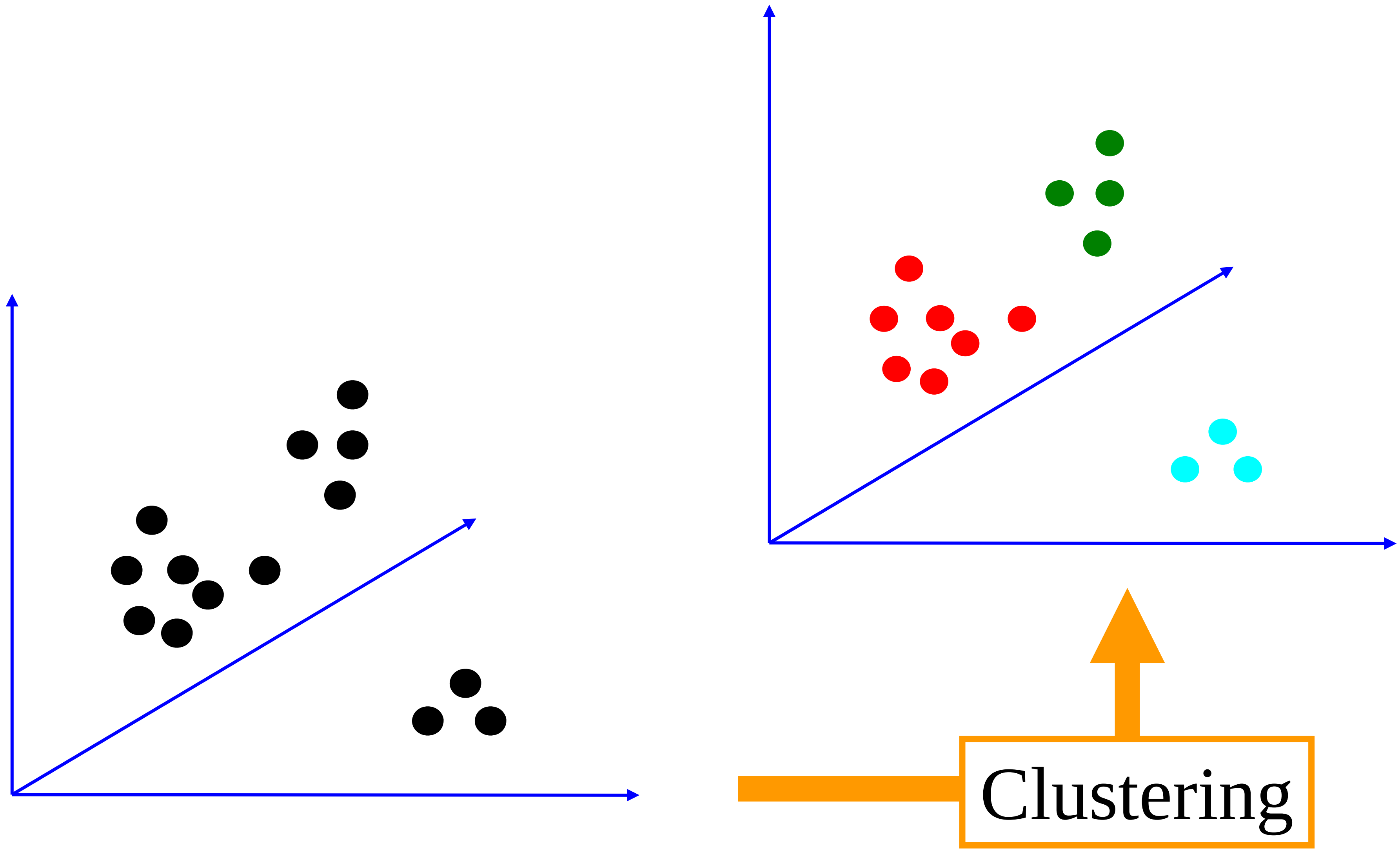
Bag-of-features for image classification



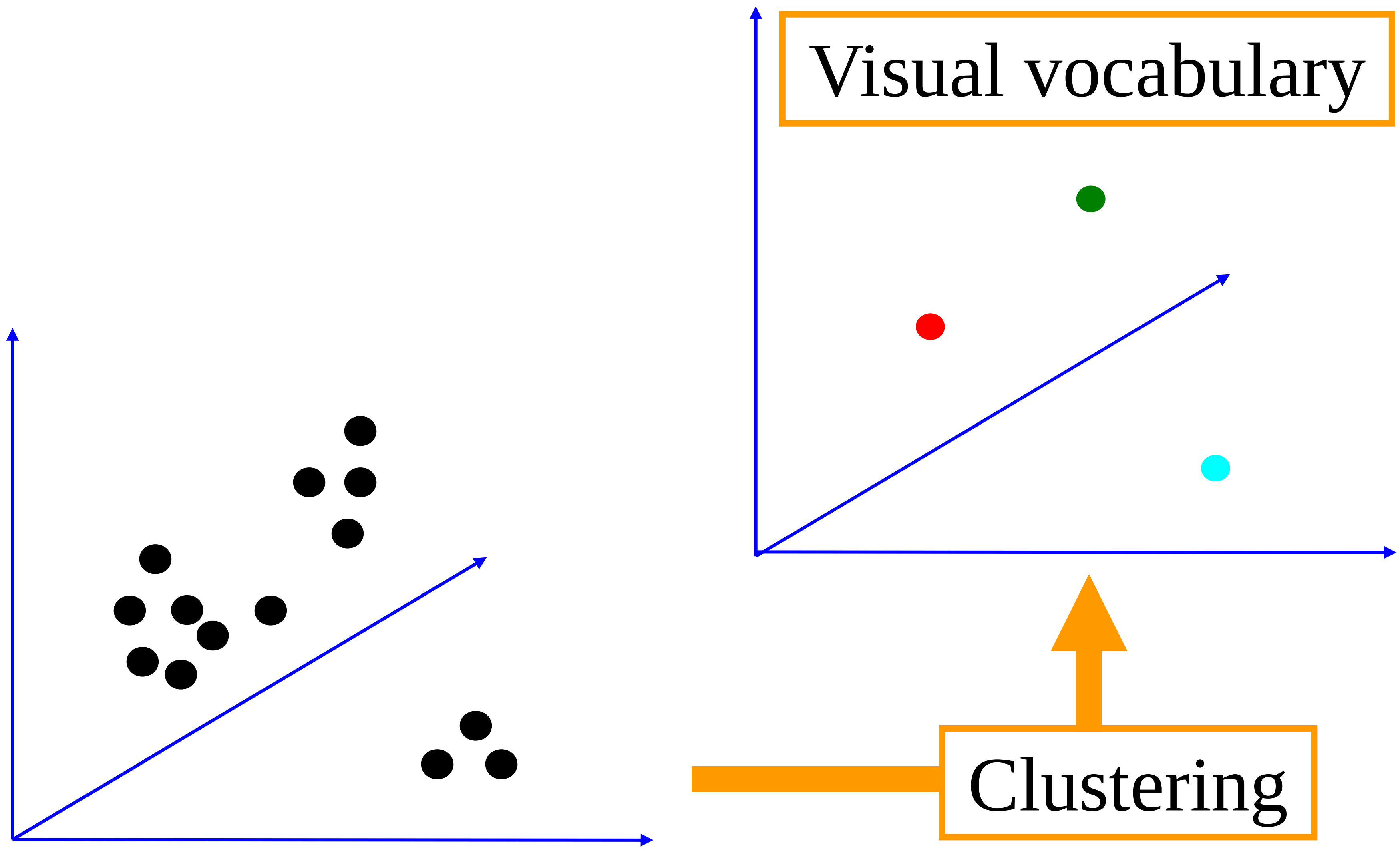
Step 2: Quantization



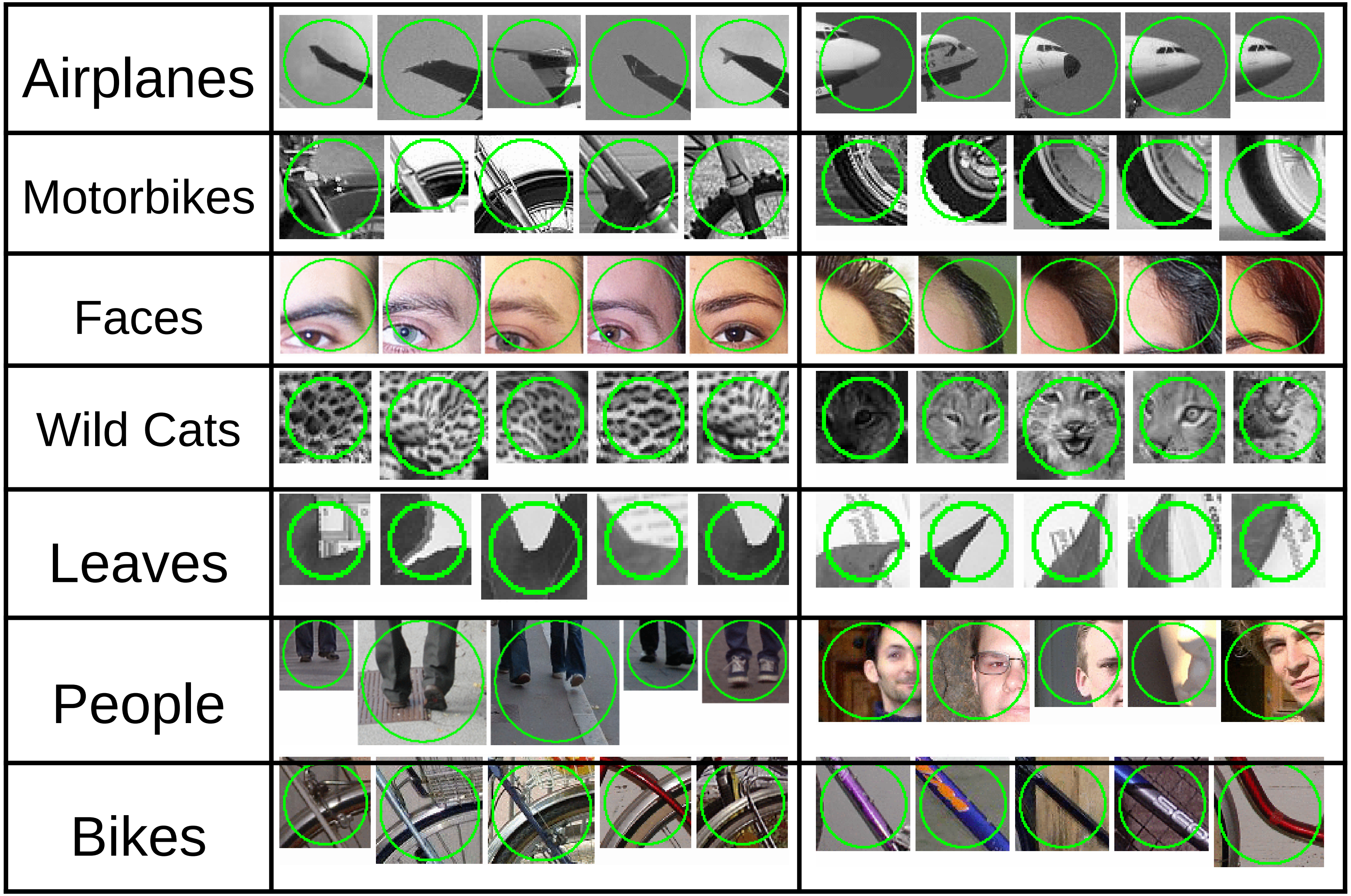
Step 2:Quantization



Step 2: Quantization



Examples for visual words



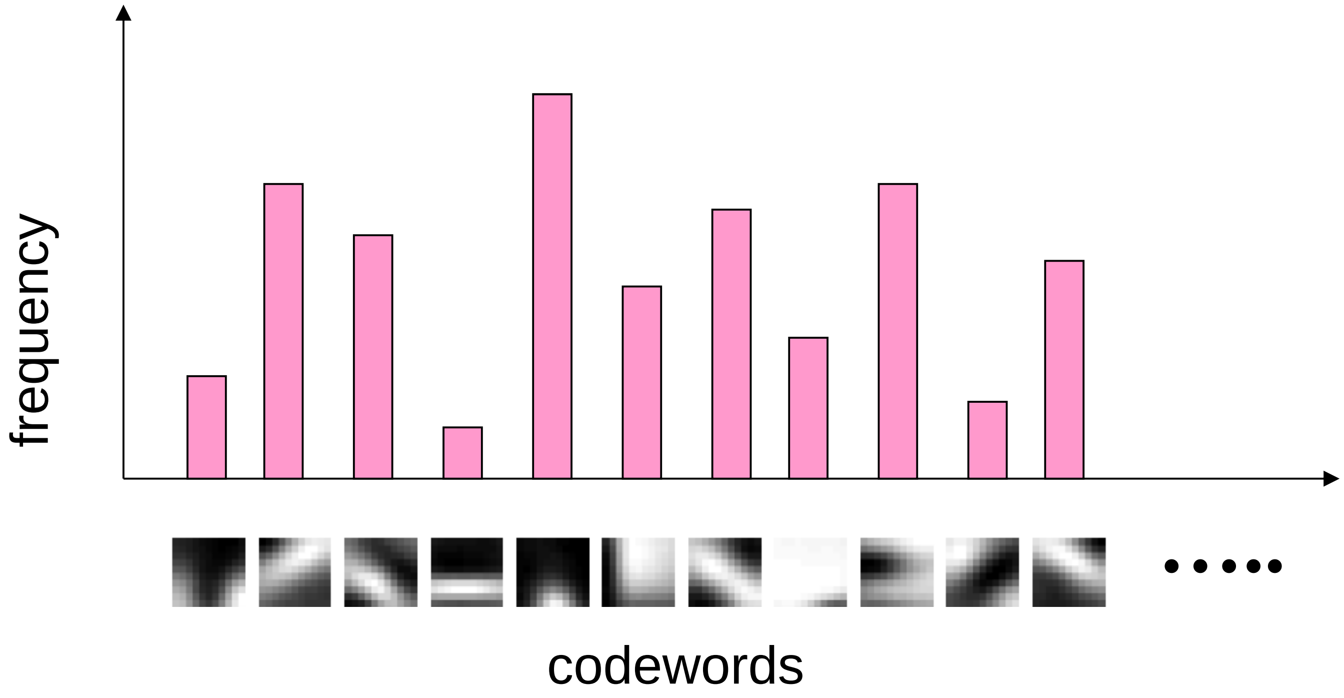
Step 2: Quantization

- Cluster descriptors
  - K-means
  - Gaussian mixture model
- Assign each visual word to a cluster
  - Hard or soft assignment
- Build frequency histogram

Hard or soft assignment

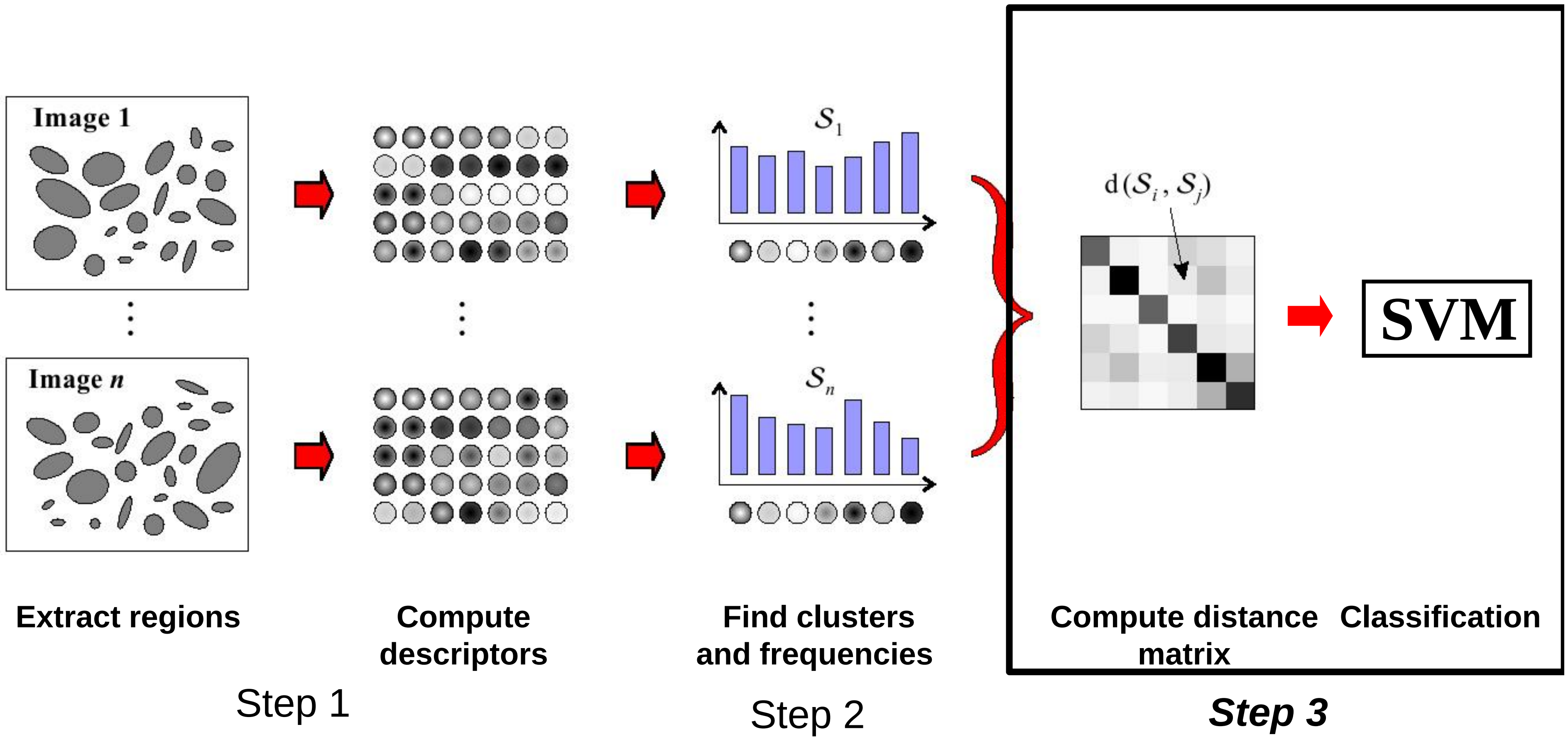
- K-means → hard assignment
  - Assign to the closest cluster center
  - Count number of descriptors assigned to a center
- Gaussian mixture model → soft assignment
  - Estimate distance to all centers
  - Sum over number of descriptors
- Represent image by a frequency histogram

Image representation



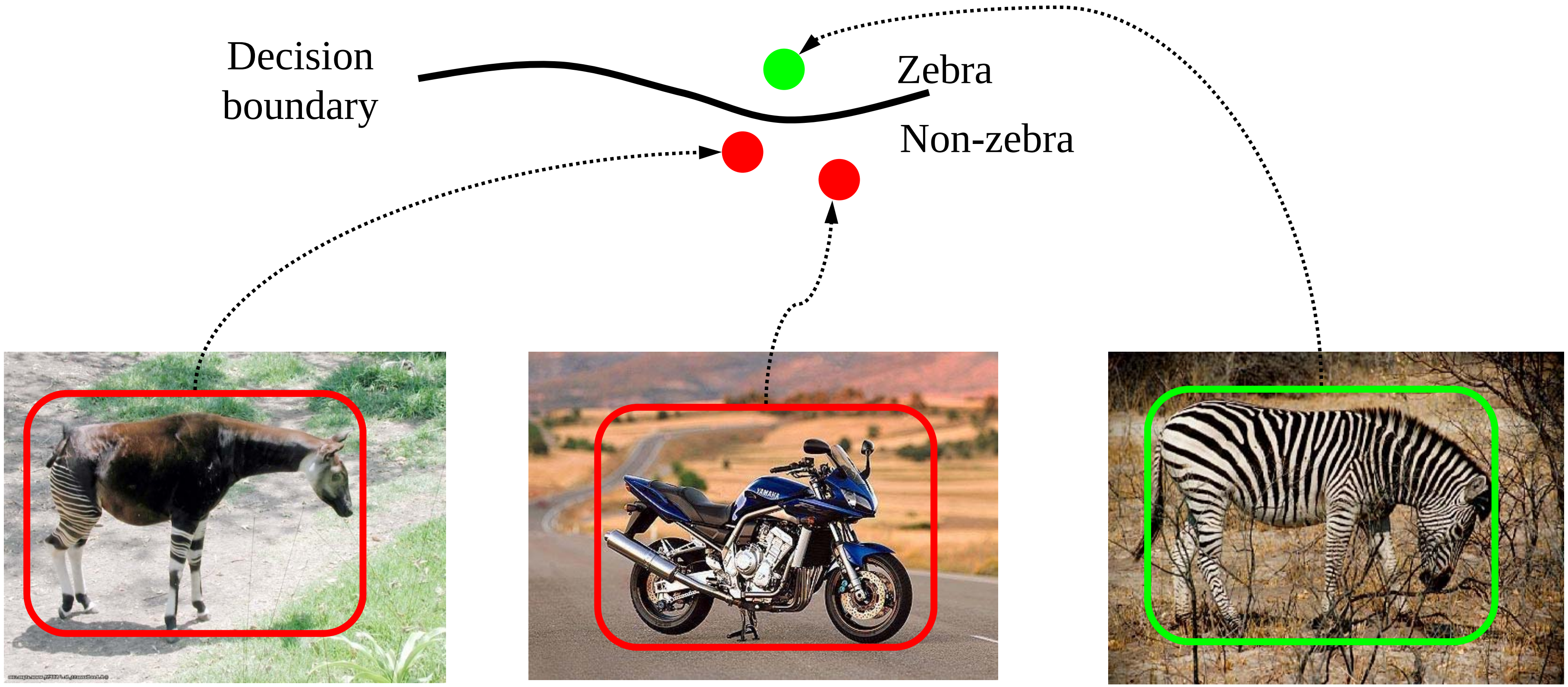
- each image is represented by a vector, typically 1000-4000 dimension, normalization with L2 norm
- fine grained – represent model instances
- coarse grained – represent object categories

Bag-of-features for image classification



Step 3: Classification

- Learn a decision rule (classifier) assigning bag-of-features representations of images to different classes

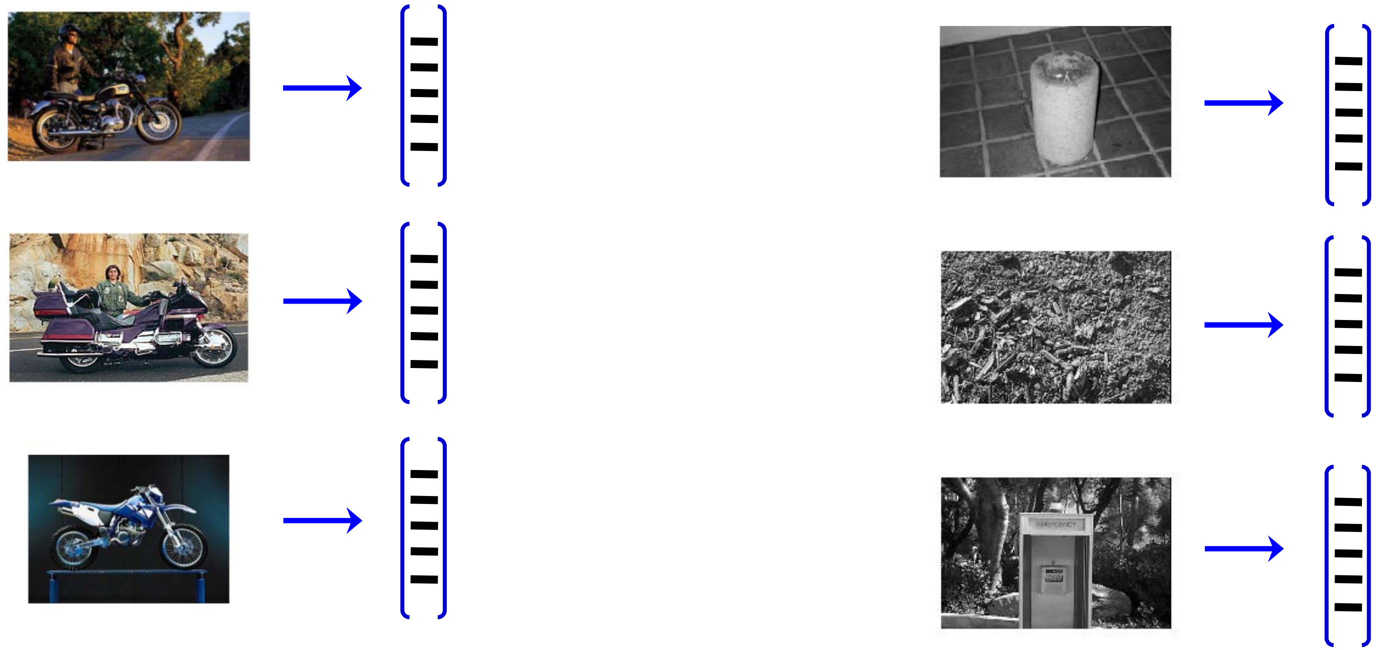


Training data

Vectors are histograms, one from each training image

positive

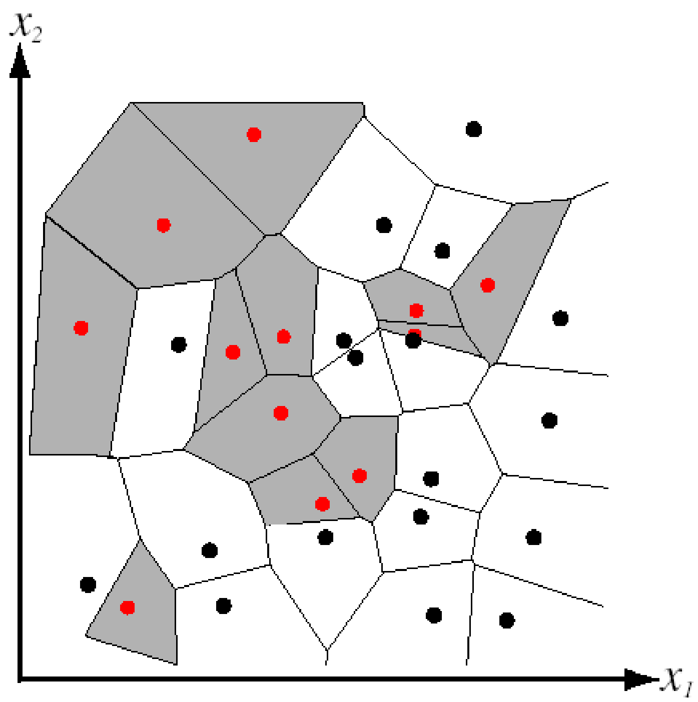
negative



Train classifier,e.g.SVM

Nearest Neighbor Classifier

- Assign label of nearest training data point to each test data point

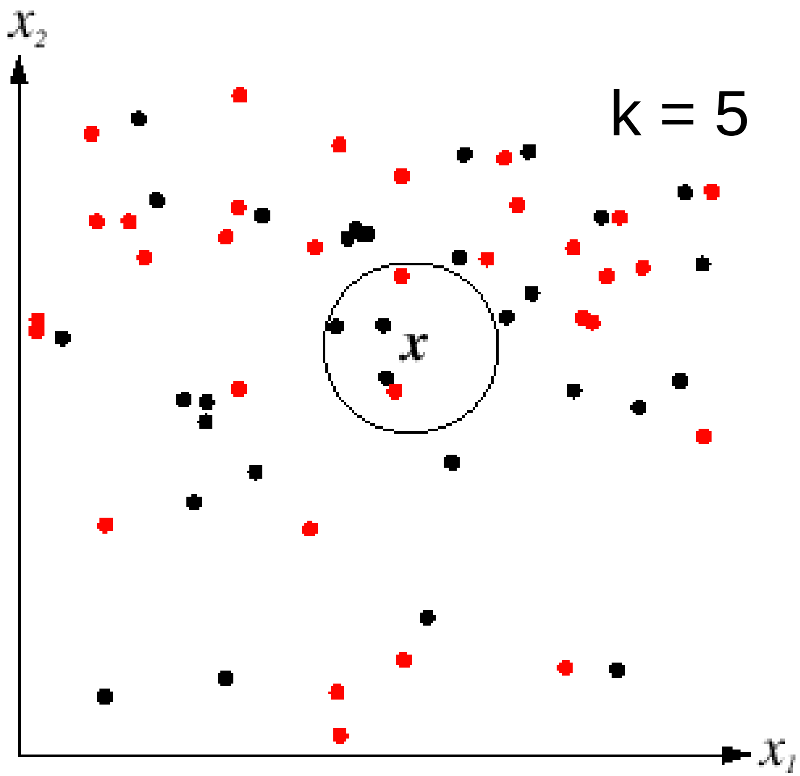


from Duda *et al.*

Voronoi partitioning of feature space
for 2-categories and 2-D data

k-Nearest Neighbors

- For a new point, find the k closest points from the training data
- Labels of the k points “vote” to classify

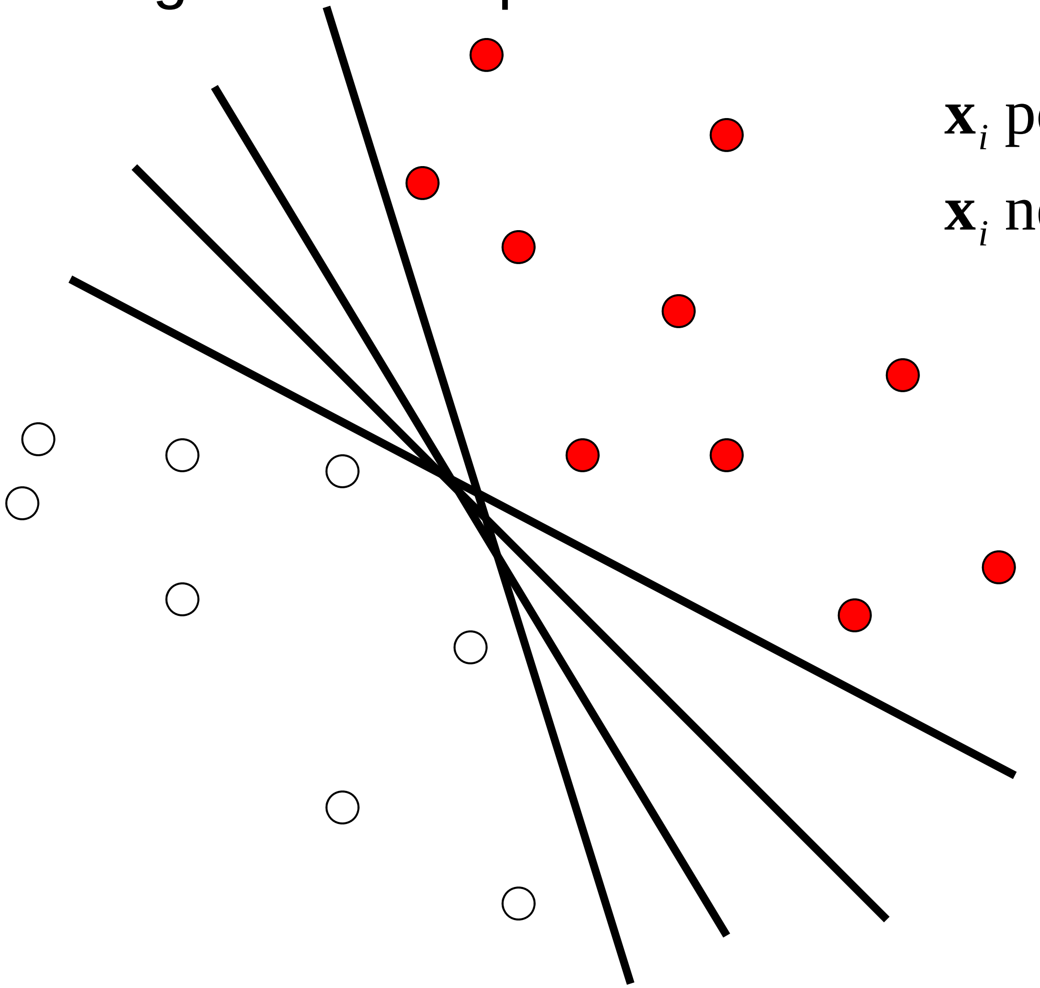


Nearest Neighbor Classifier

- For each test data point : assign label of nearest training data point
- K-nearest neighbors: labels of the k nearest points, vote to classify
- Works well provided there is lots of data and the distance function is good

Linear classifiers

- Find linear function (*hyperplane*) to separate positive and negative examples



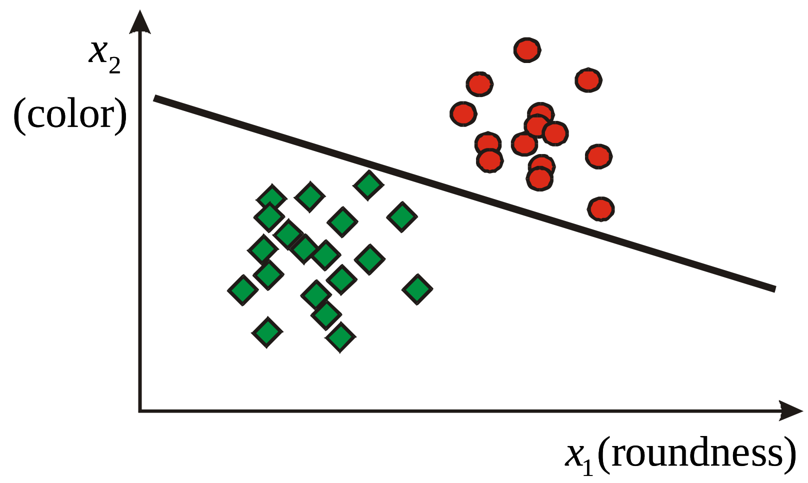
$$\mathbf{x}_i \text{ positive}: \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0$$

$$\mathbf{x}_i \text{ negative}: \quad \mathbf{x}_i \cdot \mathbf{w} + b < 0$$

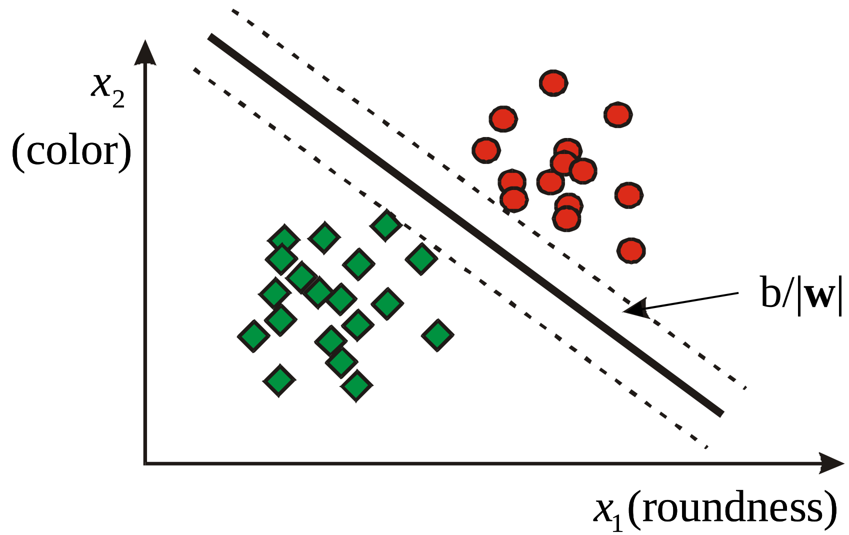
Which hyperplane is best?

Linear classifiers - margin

- Generalization is not good in this case:

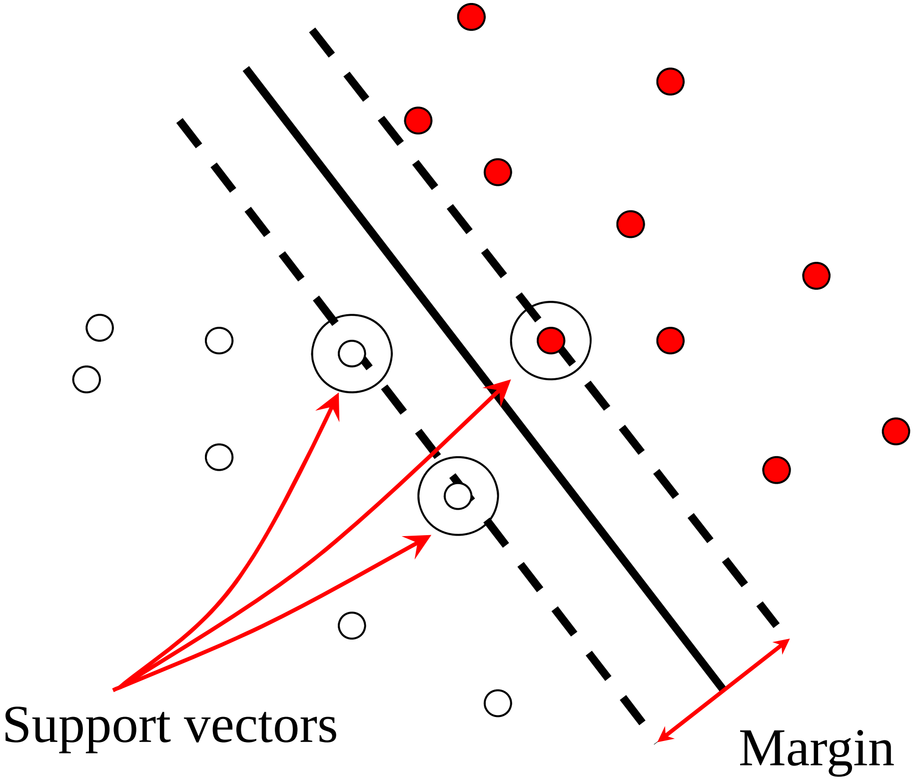


- Better if a margin is introduced:



Support vector machines

- Find hyperplane that maximizes the *margin* between the positive and negative examples



$$\mathbf{x}_i \text{ positive } (y_i = 1): \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$$

$$\mathbf{x}_i \text{ negative } (y_i = -1): \quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$$

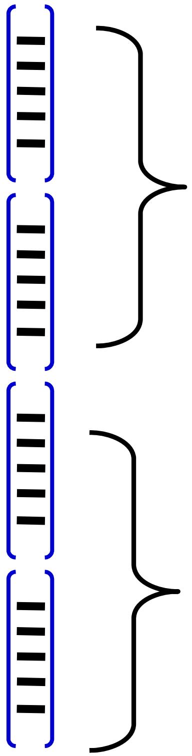
For support vectors: $\mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

Data not perfectly separable, introduction of slack variable

$$y^{(i)}(w^T x^{(i)} + b) \quad \geq \quad 1 - \xi_i$$

Why does SVM learning work?

- Learns foreground and background visual words

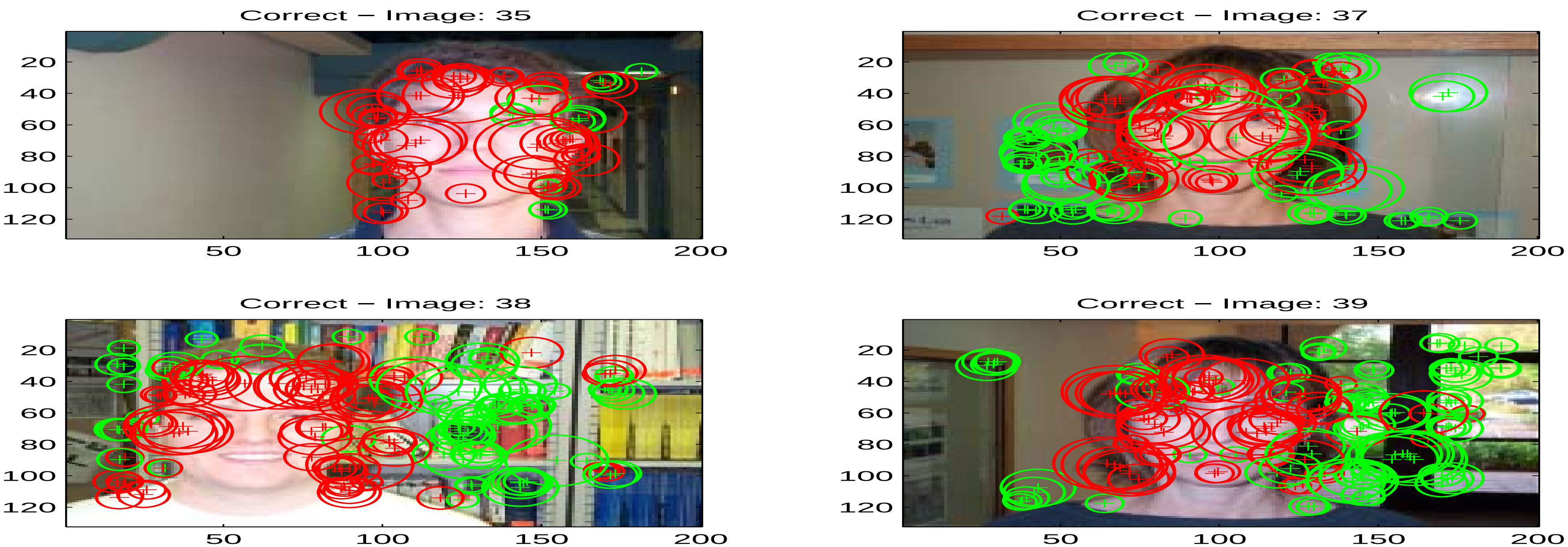


foreground words – high weight

background words – low weight

Illustration

Localization according to visual word probability



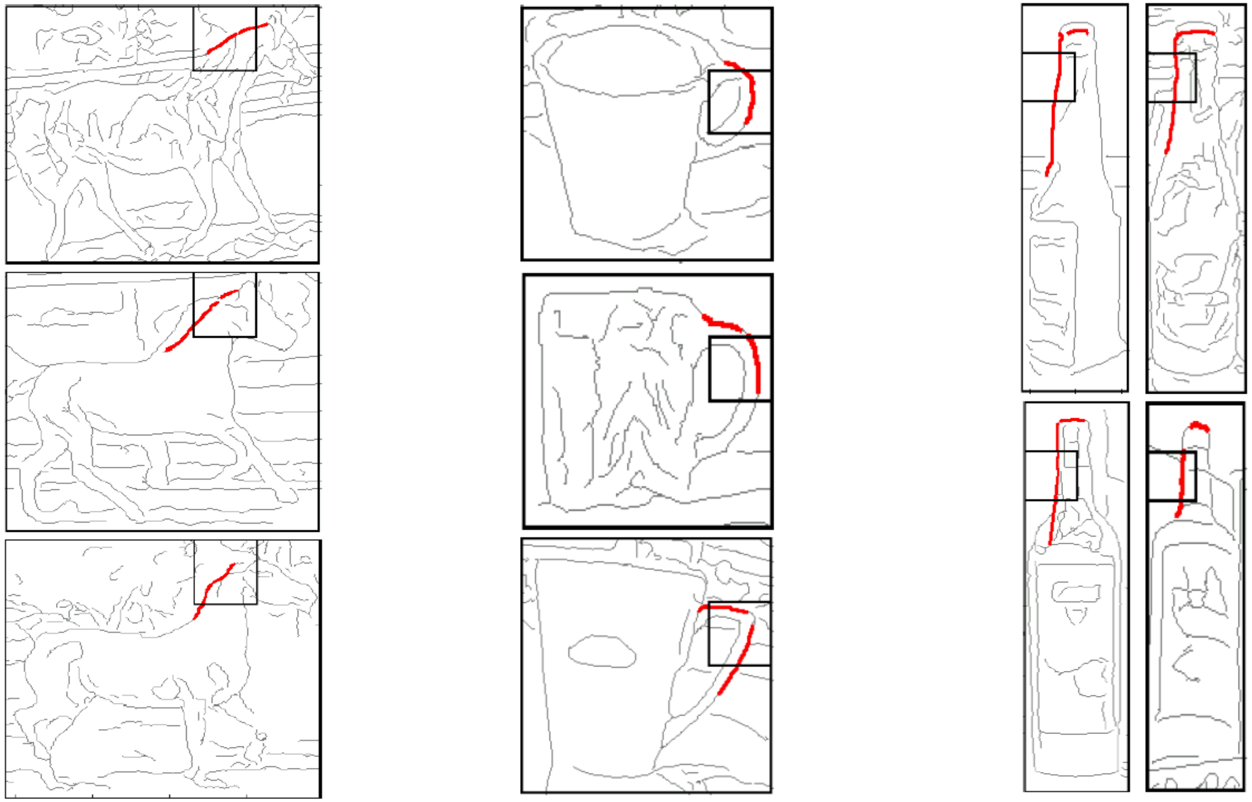
foreground word more probable

background word more probable

Illustration

A linear SVM trained from positive and negative window descriptors

A few of the highest weighed descriptor vector dimensions (= 'PAS + tile')



+ lie on object boundary (= local shape structures common to many training exemplars)

Bag-of-features for image classification

- Excellent results in the presence of background clutter

bikes books building cars people phones trees

Examples for misclassified images


Books- misclassified into faces, faces, buildings



Buildings- misclassified into faces, trees, trees



Cars- misclassified into buildings, phones, phones

Bag of visual words summary

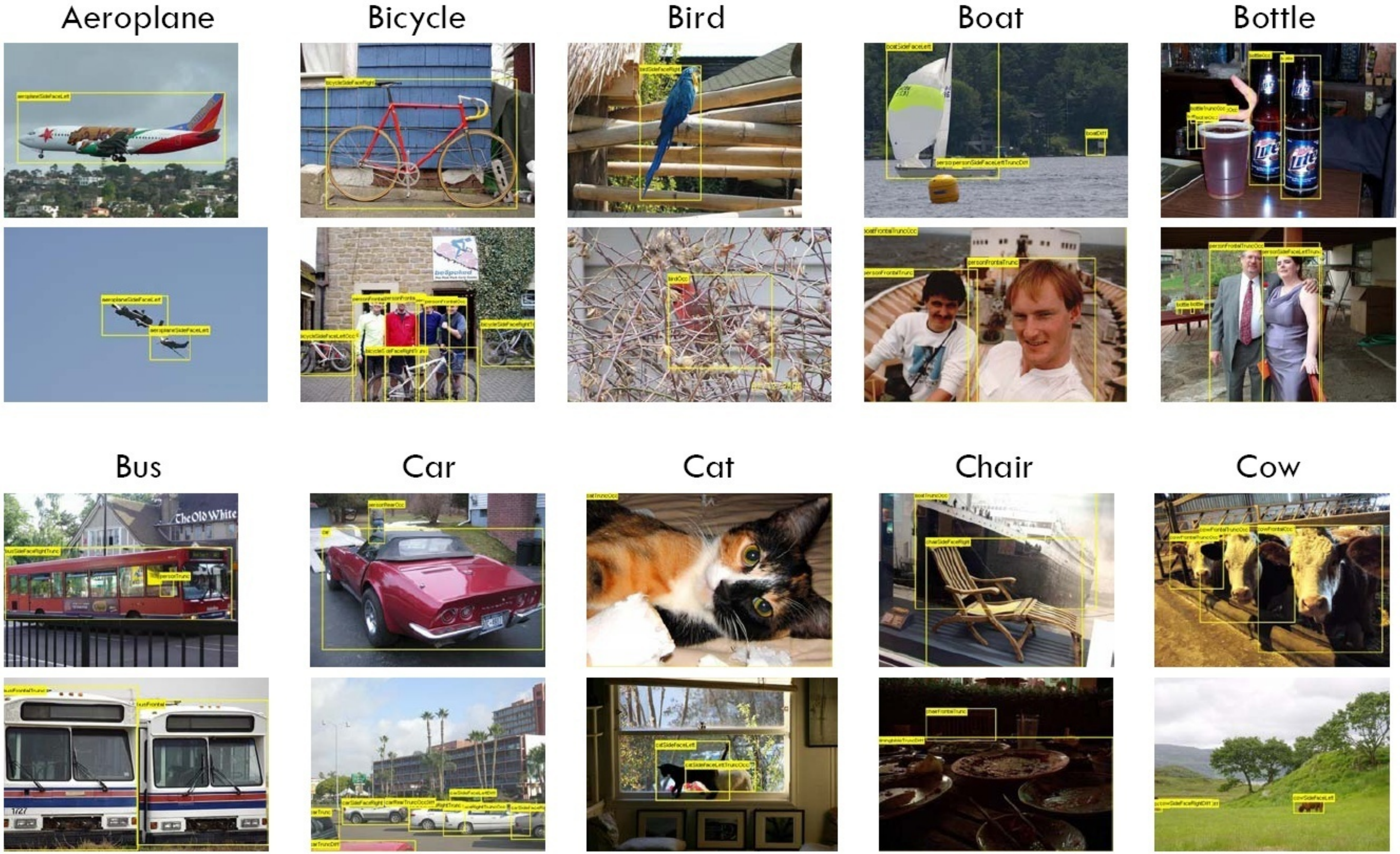
- Advantages:
  - largely unaffected by position and orientation of object in image
  - fixed length vector irrespective of number of detections
  - very successful in classifying images according to the objects they contain

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

Evaluation of image classification

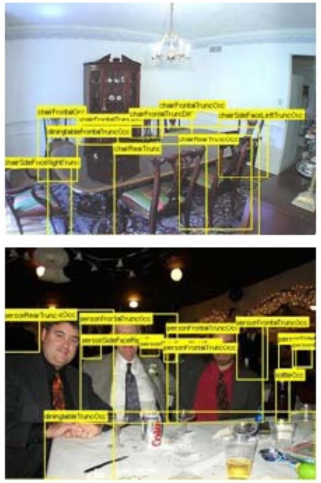
- PASCAL VOC [05-12] datasets
- PASCAL VOC 2007
  - Training *and* test dataset available
  - Used to report state-of-the-art results
  - Collected January 2007 from Flickr
  - 500 000 images downloaded and random subset selected
  - 20 classes manually annotated
  - Class labels per image + bounding boxes
  - 5011 training images, 4952 test images
- Evaluation measure: average precision

PASCAL 2007 dataset



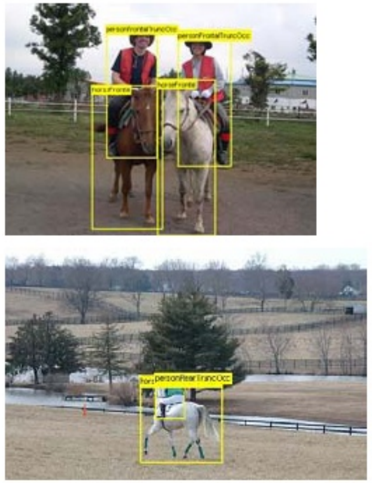
PASCAL 2007 dataset

Dining Table



Dog

Horse

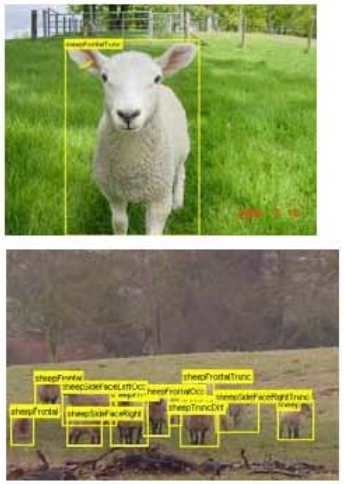


Motorbike

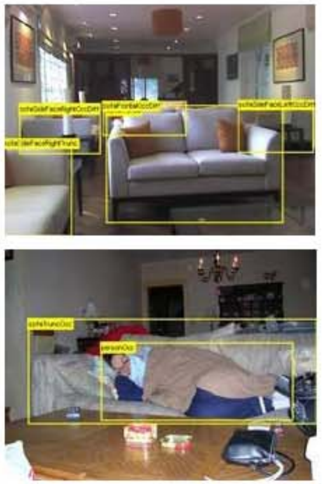
Person

Potted Plant

Sheep

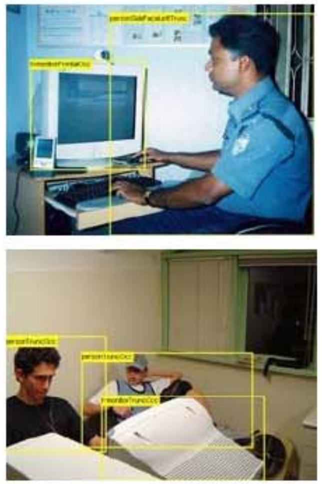


Sofa



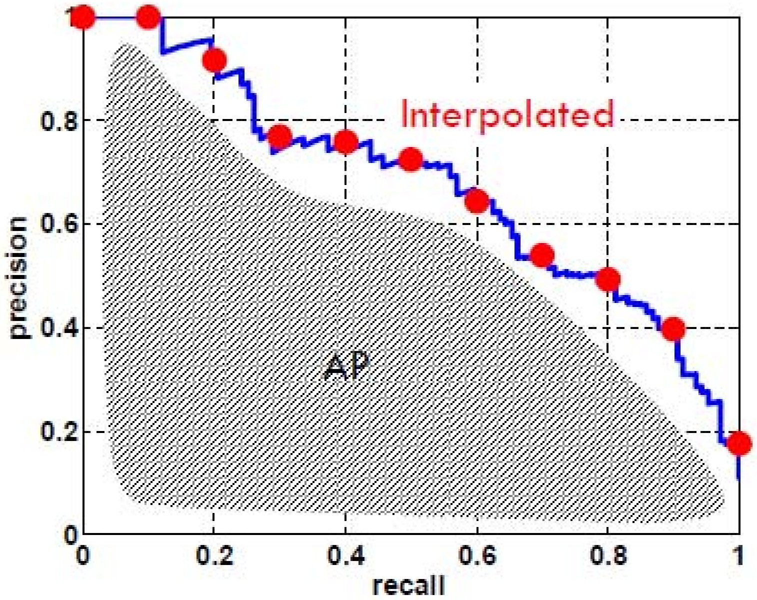
Train

TV/Monitor



Evaluation

- Average Precision [TREC] averages precision over the entire range of recall
  - Curve interpolated to reduce influence of “outliers”

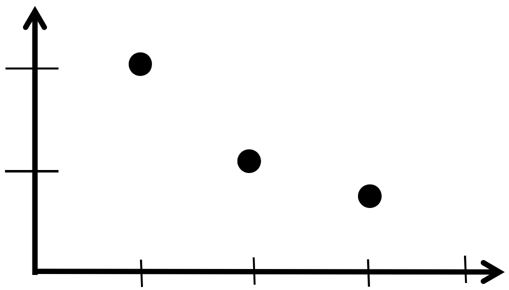


- A good score requires both high recall and high precision
- Application-independent
- Penalizes methods giving high precision but low recall

Precision/Recall

- Ranked list for category A :

A, C, B, A, B, C, C, A ; in total four images with category A



Results for PASCAL 2007

- Winner of PASCAL 2007 [Marszalek et al.] : mAP 59.4
  - Combining several channels with non-linear SVM and Gaussian kernel
- Multiple kernel learning [Yang et al. 2009] : mAP 62.2
  - Combination of several features, Group-based MKL approach
- Object localization & classification [Harzallah et al.'09] : mAP 63.5
  - Use detection results to improve classification
- Adding objectness boxes [Sanchez at al.'12] : mAP 66.3
- Convolutional Neural Networks [Oquab et al.'14] : mAP 77.7

Spatial pyramid matching

- Add spatial information to the bag-of-features
- Perform matching in 2D image space

[Lazebnik, Schmid & Ponce, CVPR 2006]

Extensions to BOF

- Efficient Additive Kernels via Explicit Feature Maps,
  A. Vedaldi and Zisserman, CVPR'10.
  - approximation by linear kernels

- Improved aggregation schemes, such as the Fisher vector, Perronnin et al., ECCV'10
  - More discriminative descriptor, power normalization, linear SVM

- Excellent results of the Fisher vector in a recent evaluation, Chatfield et al. BMVC 2011

Large-scale image classification

- Image classification: assigning a class label to the image

Car: present
Cow: present
Bike: not present
Horse: not present
...

- What makes it large-scale?
  - number of images
  - number of classes
  - dimensionality of descriptor

IMAGENET has 14M images from 22k classes

ImageNet

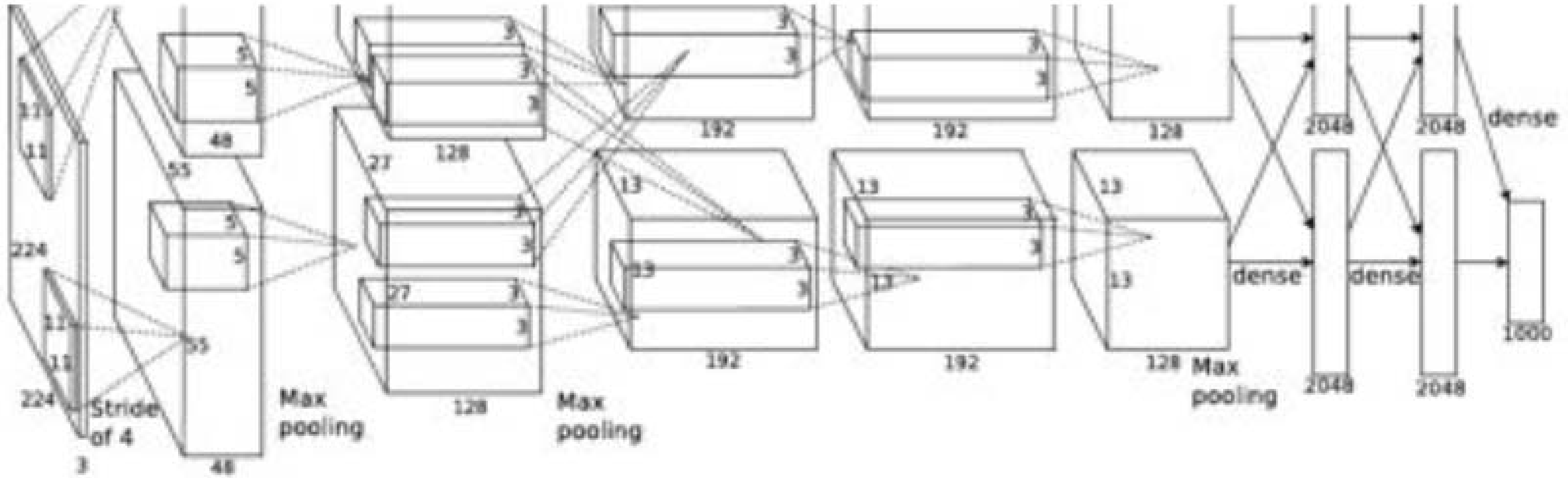
- Datasets
  - ImageNet Large Scale Visual Recognition Challenge 2010 (ILSVRC)
    - 1000 classes and 1.4M images
  - ImageNet10K dataset
    - 10184 classes and ~ 9 M images

(a) Star Anise (92.45%) (b) Geyser (85.45%) (c) Pulp Magazine (83.01%) (d) Carrycot (81.48%)

(e) European gallinule (15.00%) (f) Sea Snake (10.00 %) (g) Paintbrush (4.68 %) (h) Mountain Tent (0.00%)

Large-scale image classification

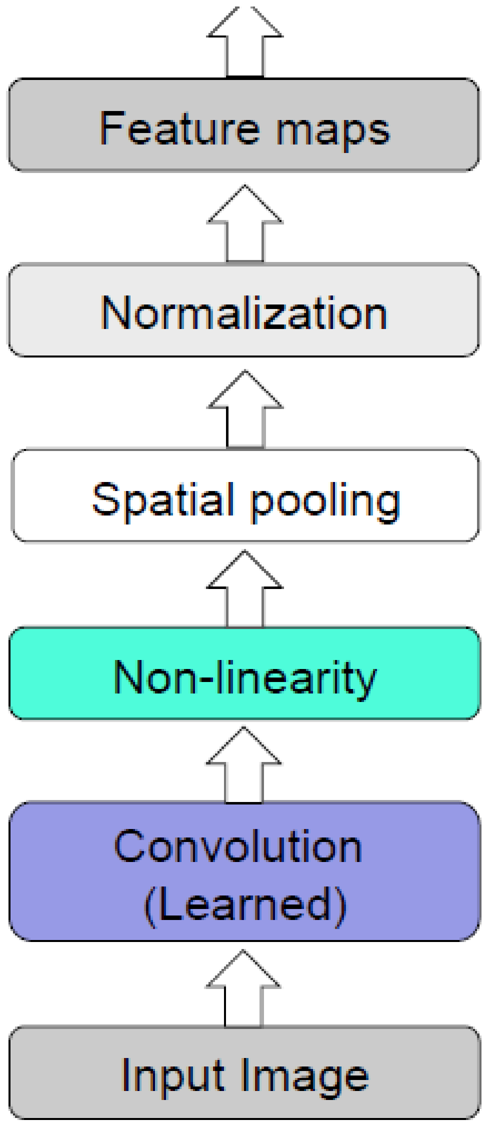
- Convolutional neural networks (CNN)
- Large model (7 hidden layers, 650k unit, 60M parameters)
- Requires large training set (ImageNet)
- GPU implementation (50x speed up over CPU)



A. Krizhevsky, I. Sutskever, and G. Hinton,
ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012

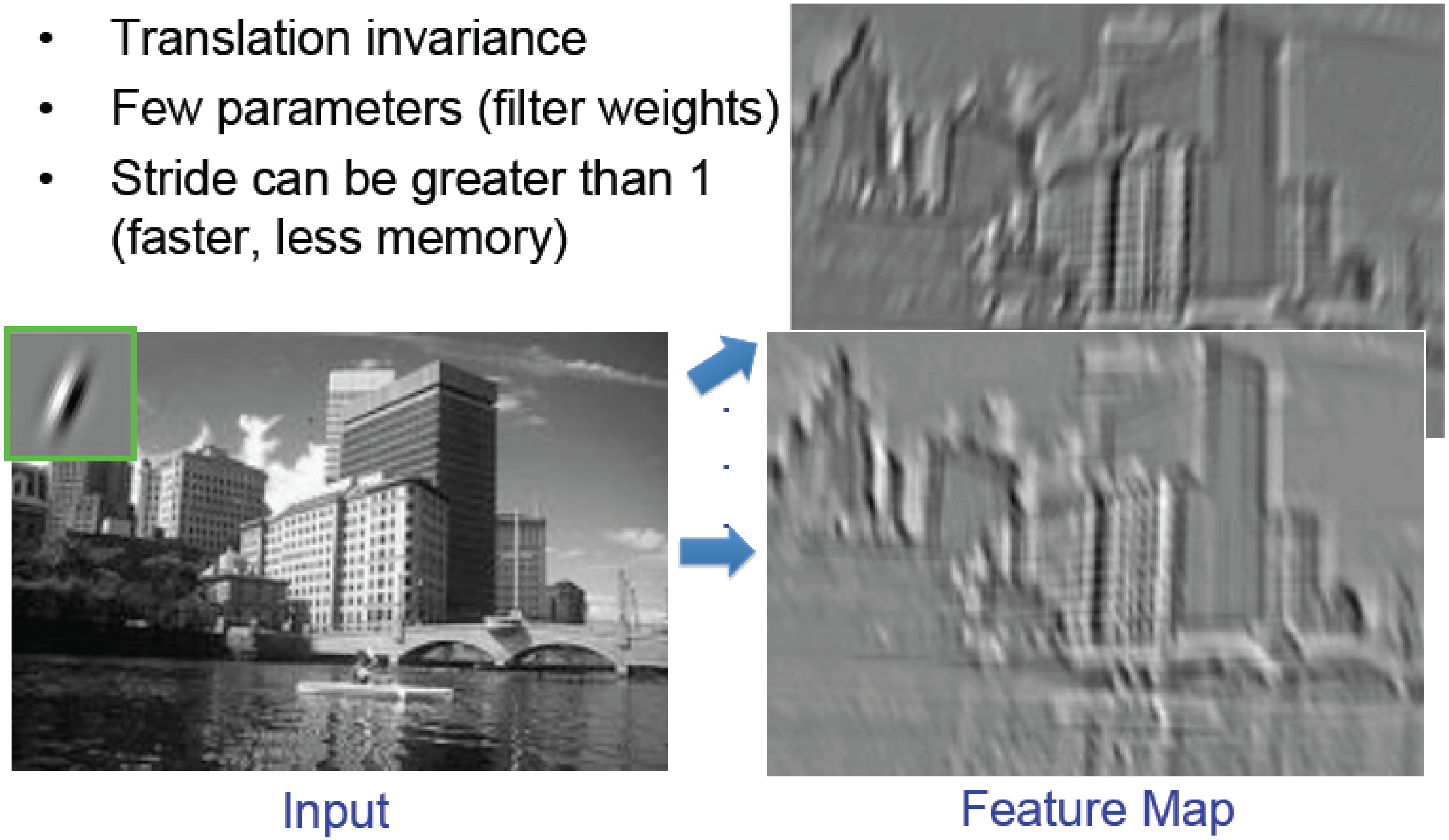
Convolutional neural networks

- Feed-forward feature extraction:
  1. Convolve input with learned filters
  2. Non-linearity
  3. Spatial pooling
  4. Normalization
- Supervised training of convolutional filters by back-propagating classification error



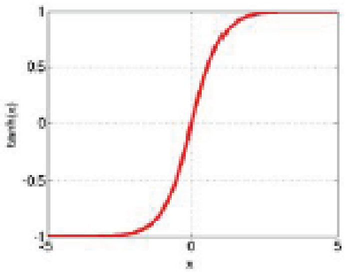
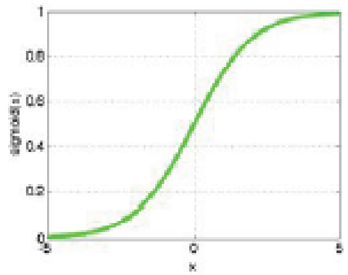
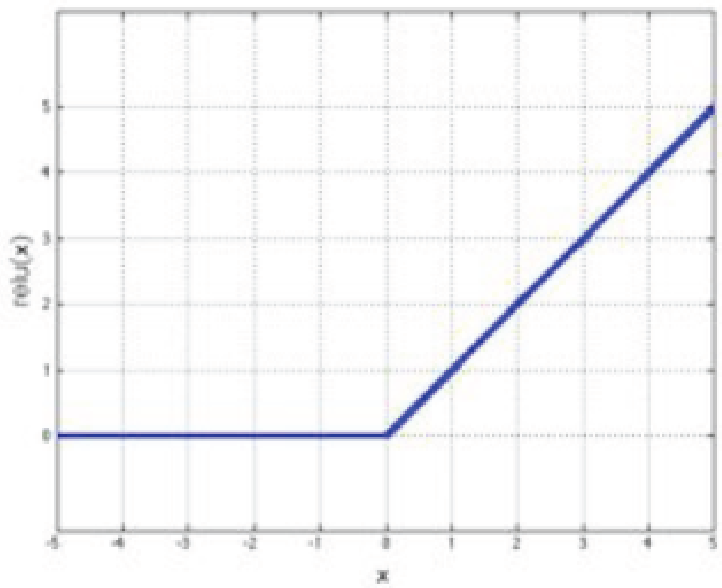
1. Convolution

- Dependencies are local
- Translation invariance
- Few parameters (filter weights)
- Stride can be greater than 1 (faster, less memory)



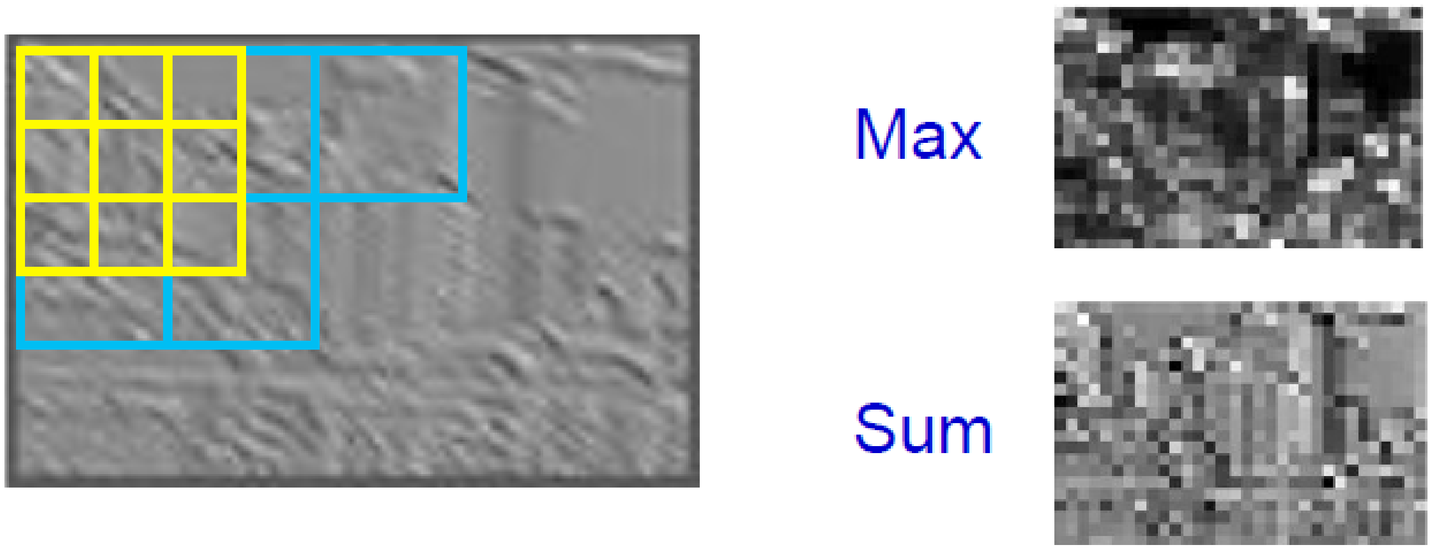
2. Non-linearity

- Per-element (independent)
- Options:
  - Tanh
  - Sigmoid: 1/(1+exp(-x))
  - Rectified linear unit (ReLU)
    - Simplifies backpropagation
    - Makes learning faster
    - Avoids saturation issues
      → Preferred option

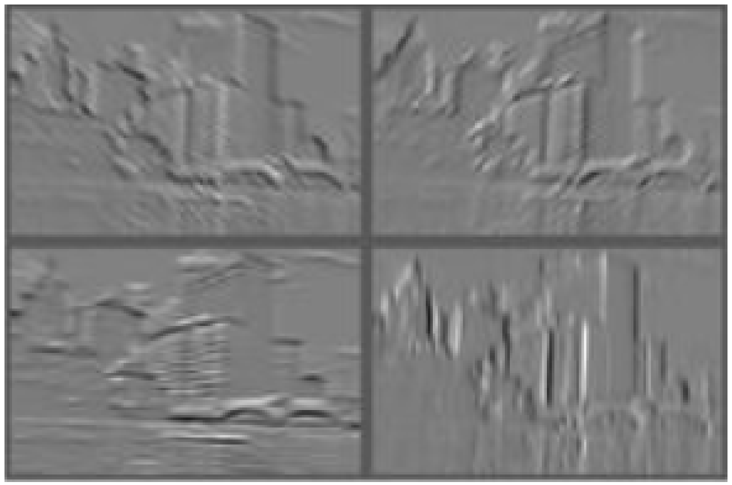
3. Spatial pooling

- Sum or max
- Non-overlapping / overlapping regions
- Role of pooling:
  - Invariance to small transformations
  - Larger receptive fields (see more of input)

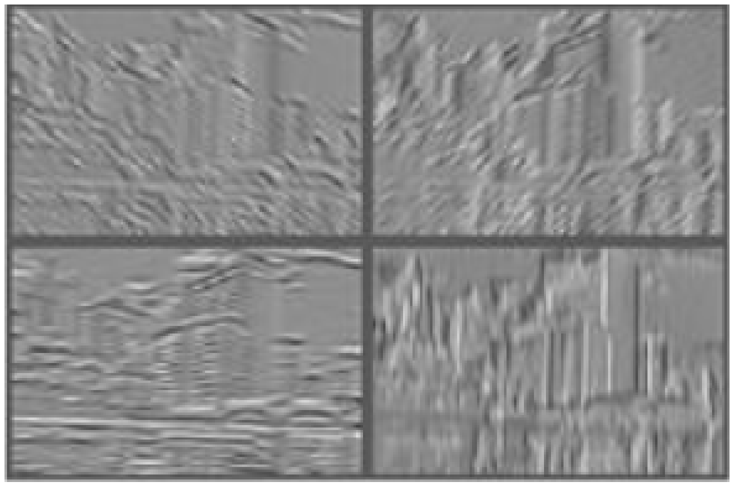


4. Normlization

- Within or across feature maps
- Before or after spatial pooling



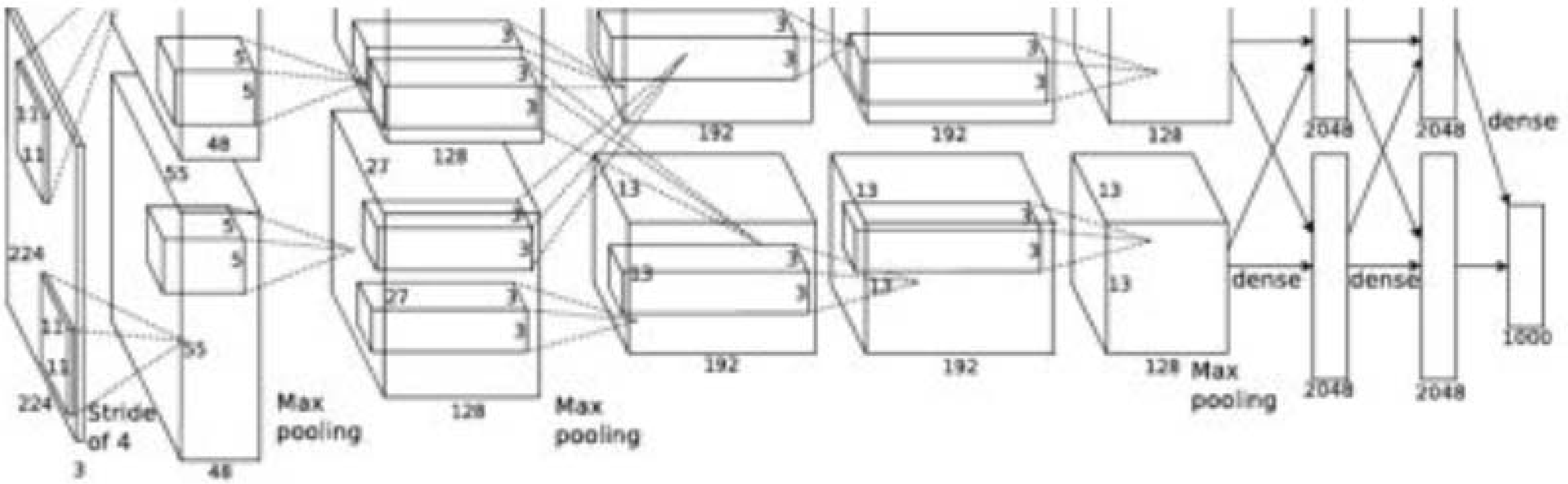
Feature Maps



Feature Maps
After Contrast Normalization

Large-scale image classification

- State-of-the-art performance on ImageNet



A. Krizhevsky, I. Sutskever, and G. Hinton,
ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012