Bag-of-features
for category classification

Cordelia Schmid

INRIA

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

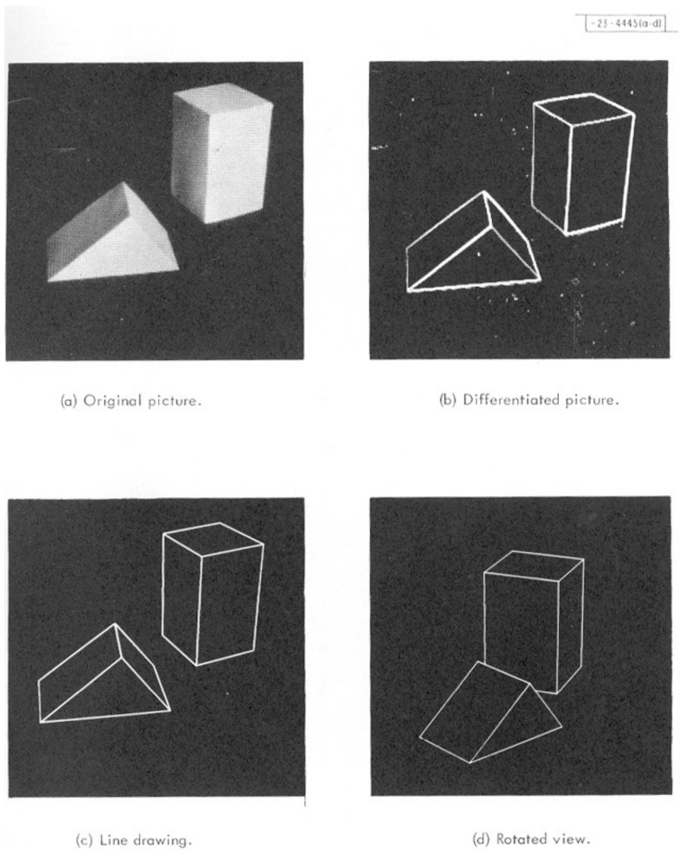
Category

Category recognition

- Robust image description
  - Appropriate descriptors for categories
- Statistical modeling and machine learning for vision
  - Use and validation of appropriate techniques

Why machine learning?

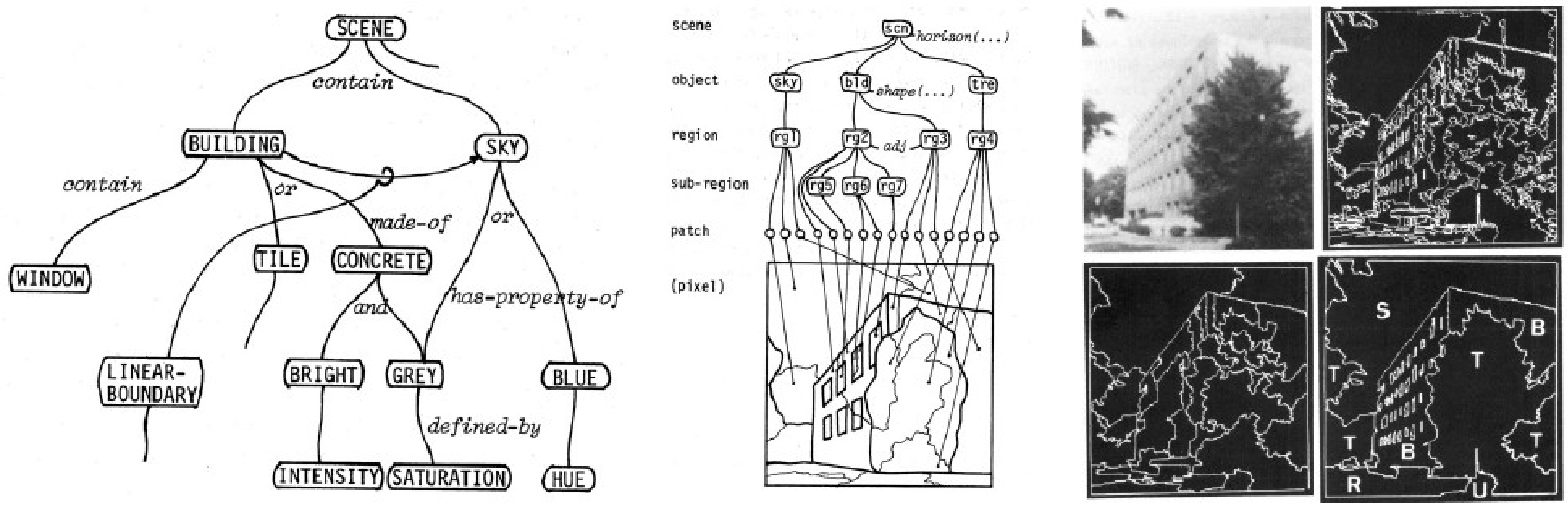
- Early approaches: simple features + handcrafted models
- Can handle only few images, simples tasks



L. G. Roberts, *Machine Perception of Three Dimensional Solids,* Ph.D. thesis, MIT Department of Electrical Engineering, 1963.

Why machine learning?

- Early approaches: manual programming of rules
- Tedious, limited and does not take into accout the data



(a) Bottom-up process (b) Top-down process (c) Result

Figure 3. A system developed in 1978 by Ohta, Kanade and Sakai [33, 32] for knowledge-based interpretation of outdoor natural scenes. The system is able to label an image (c) into semantic classes: S-sky, T-tree, R-road, B-building, U-unknown.

Y. Ohta, T. Kanade, and T. Sakai, "An Analysis System for Scenes Containing objects with Substructures," *International Joint Conference on Pattern Recognition*, 1978.

Why machine learning?

- Today lots of data, complex tasks

Internet images,
personal photo albums

Movies, news, sports

- Instead of trying to encode rules directly, learn them from examples of inputs and desired outputs

Types of learning problems

- Supervised
  - Classification
  - Regression
- Unsupervised
- Semi-supervised
- Active learning
- ….

Supervised learning

- Given training examples of inputs and corresponding outputs, produce the “correct” outputs for new inputs

- Two main scenarios:

  - **Classification:** outputs are discrete variables (category labels). Learn a decision boundary that separates one class from the other.

  - **Regression:** also known as “curve fitting” or “function approximation.” Learn a continuous input-output mapping from examples (possibly noisy).

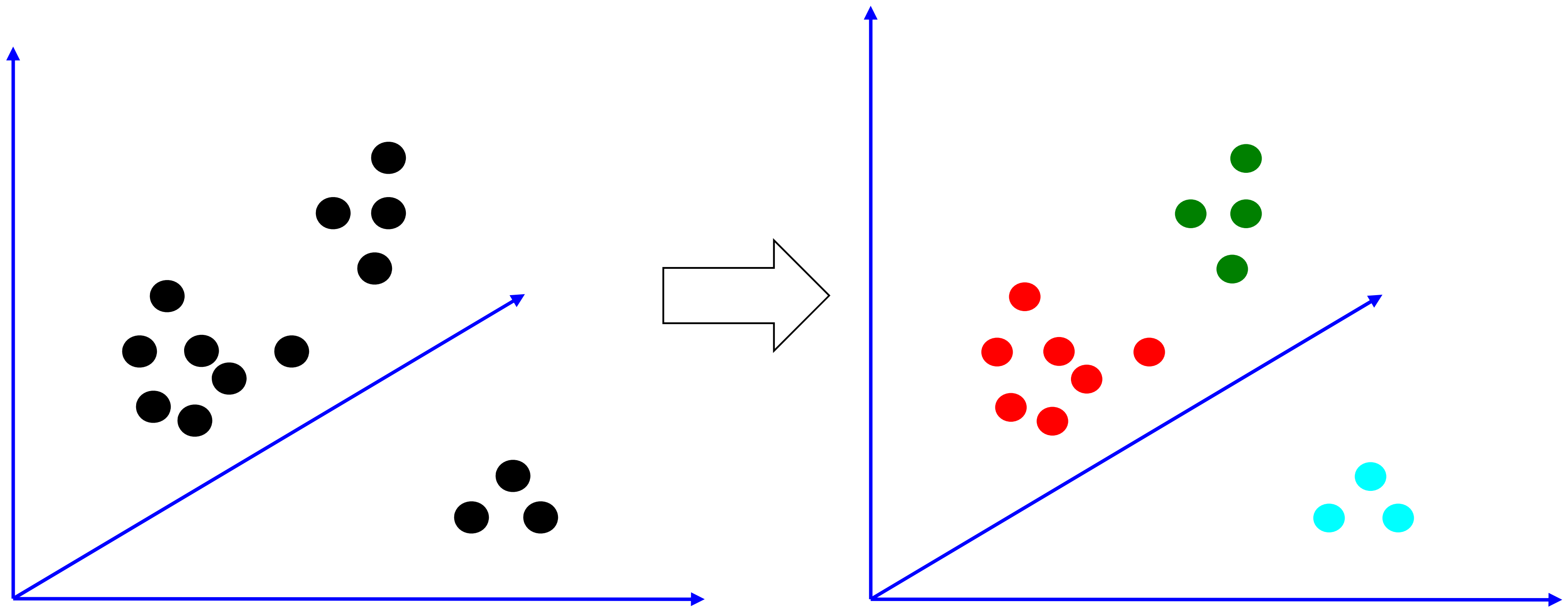
Unsupervised Learning

- Given only *unlabeled* data as input, learn some sort of structure.

- The objective is often more vague or subjective than in supervised learning. This is more an exploratory/descriptive data analysis.

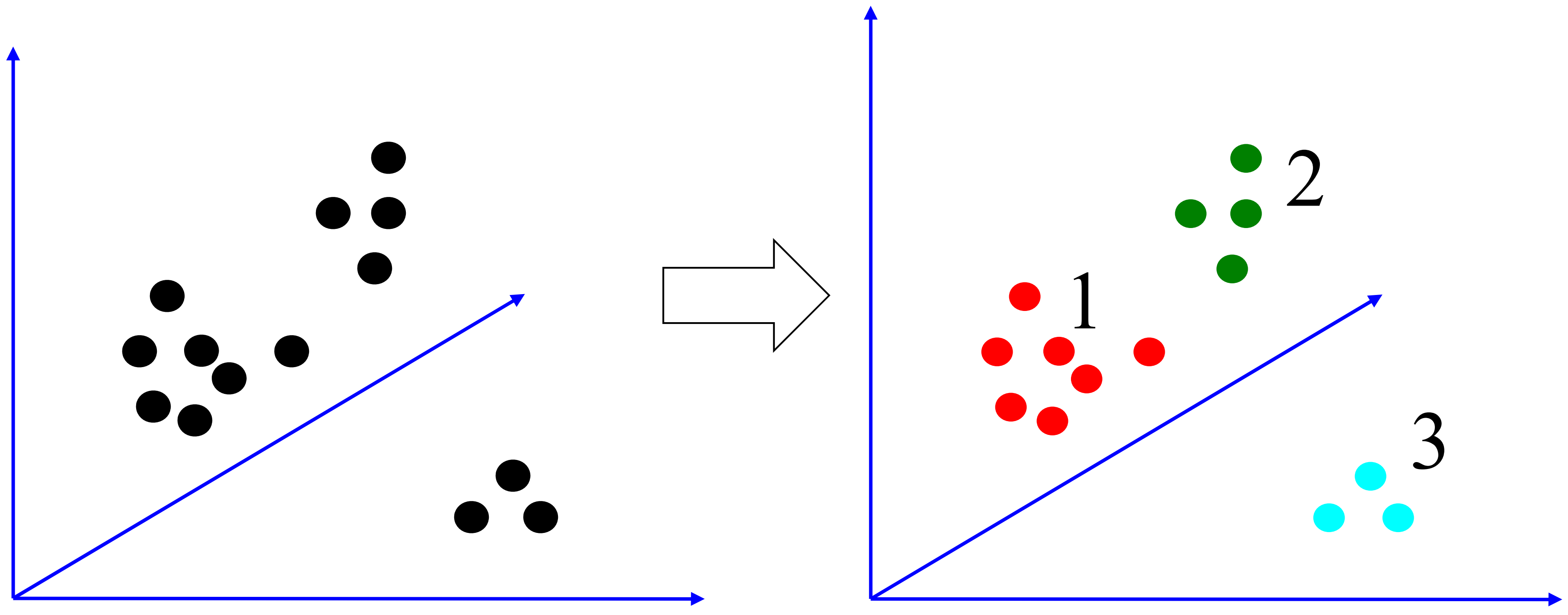
Unsupervised Learning

- **Clustering**
  - Discover groups of “similar” data points



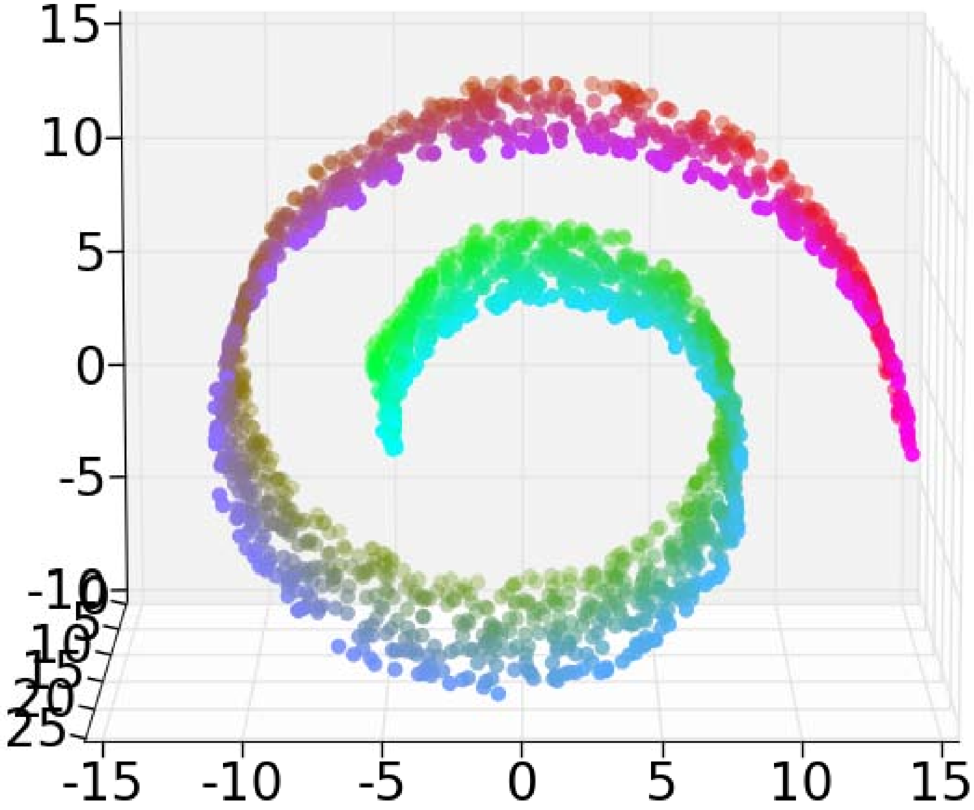
Unsupervised Learning

- **Quantization**
  - Map a continuous input to a discrete (more compact) output



Unsupervised Learning

- **Dimensionality reduction, manifold learning**
  - Discover a lower-dimensional surface on which the data lives

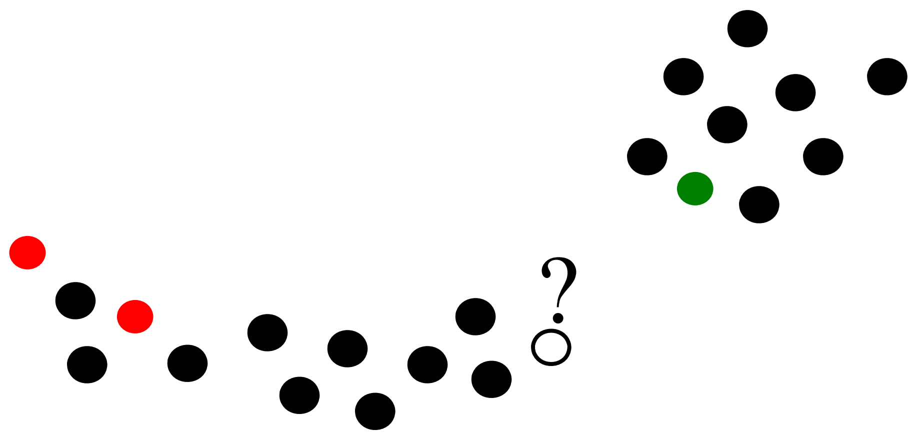


Other types of learning

- **Semi-supervised learning:** lots of data is available, but only small portion is labeled (e.g. since labeling is expensive)

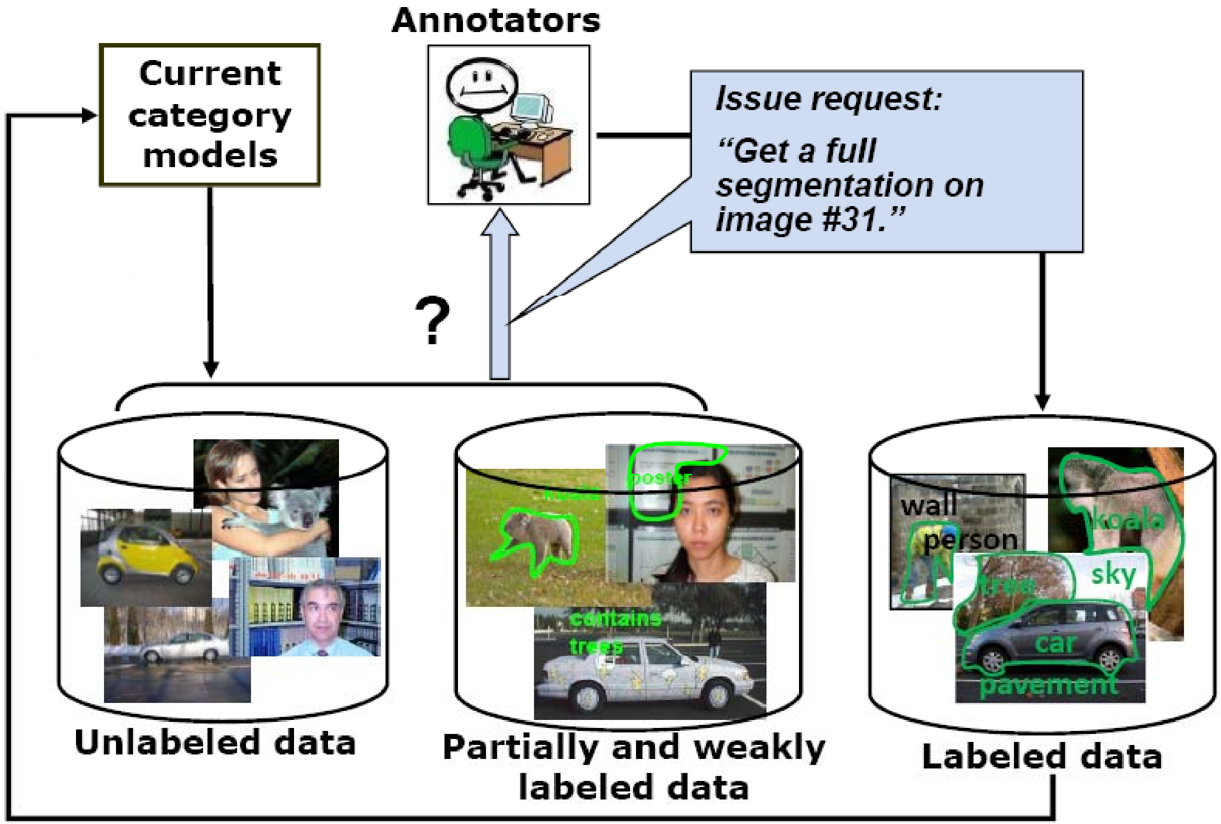
Other types of learning

- **Semi-supervised learning:** lots of data is available, but only small portion is labeled (e.g. since labeling is expensive)
  - Why is learning from labeled and unlabeled data better than learning from labeled data alone?



Other types of learning

- **Active learning:** the learning algorithm can choose its own training examples, or ask a “teacher” for an answer on selected inputs



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

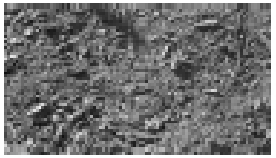
Image classification

- Given

    Positive training images containing an object class



    Negative training images that don’t



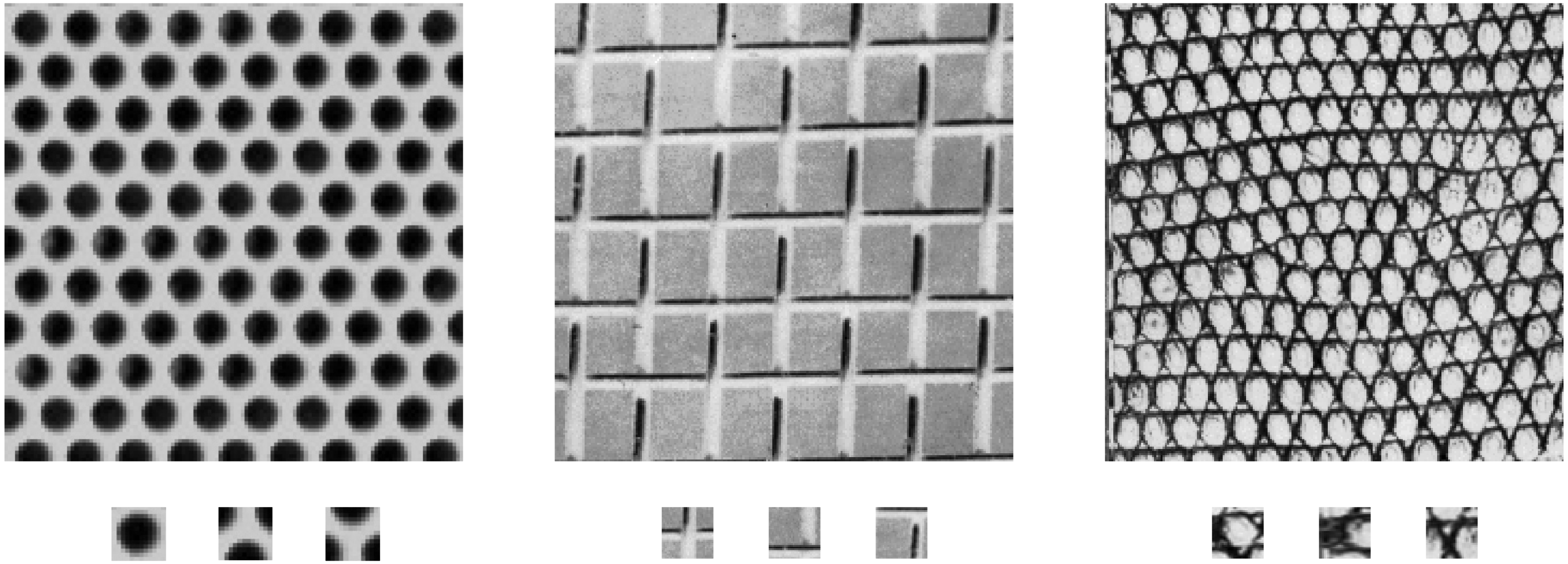
- Classify

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

?

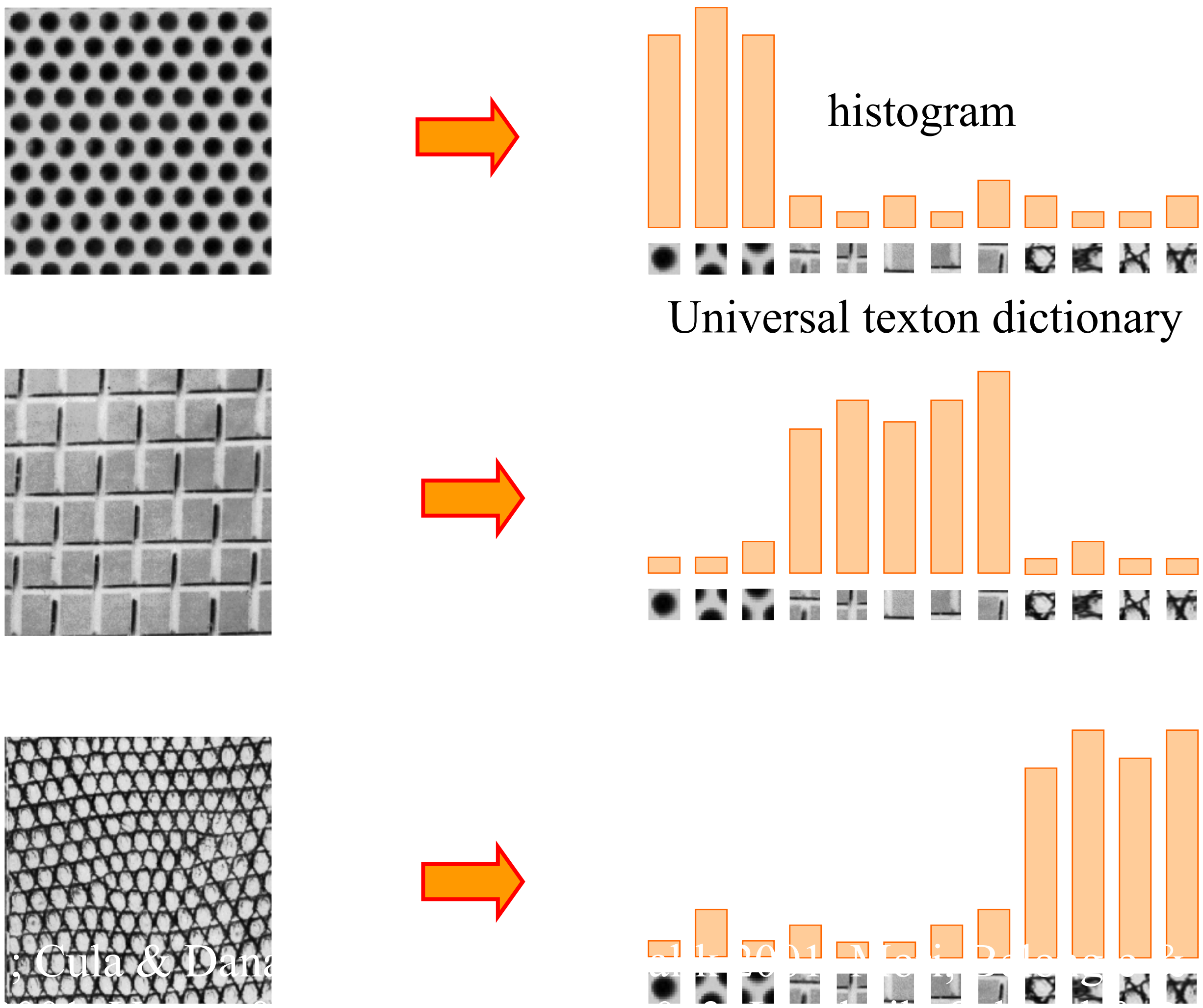
Bag-of-features for image classification

- Origin: texture recognition
  - Texture is characterized by the repetition of basic elements or *textons*



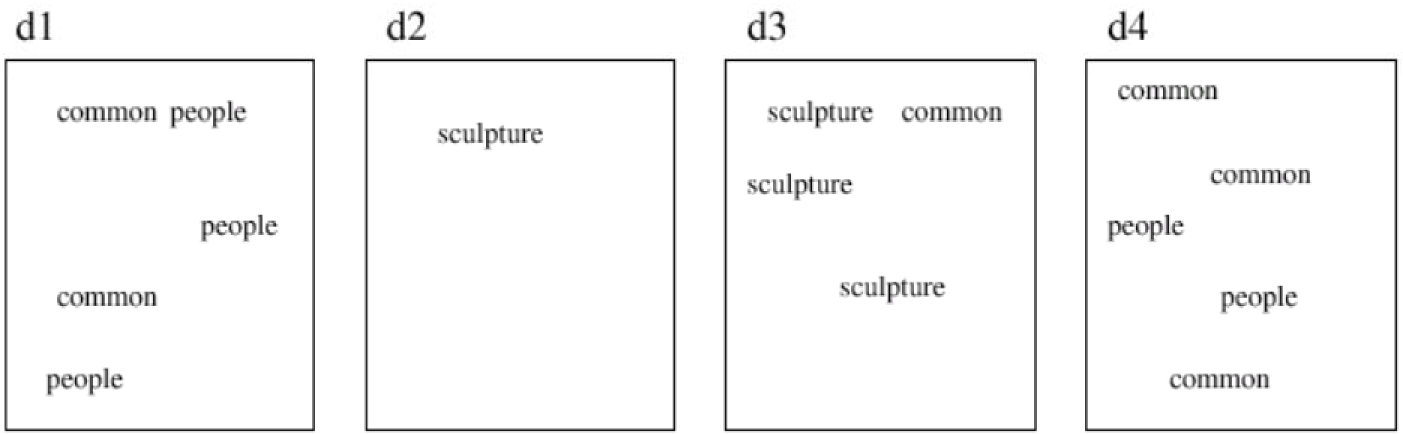
Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001
Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

Texture recognition



Bag-of-features – Origin: bag-of-words (text)

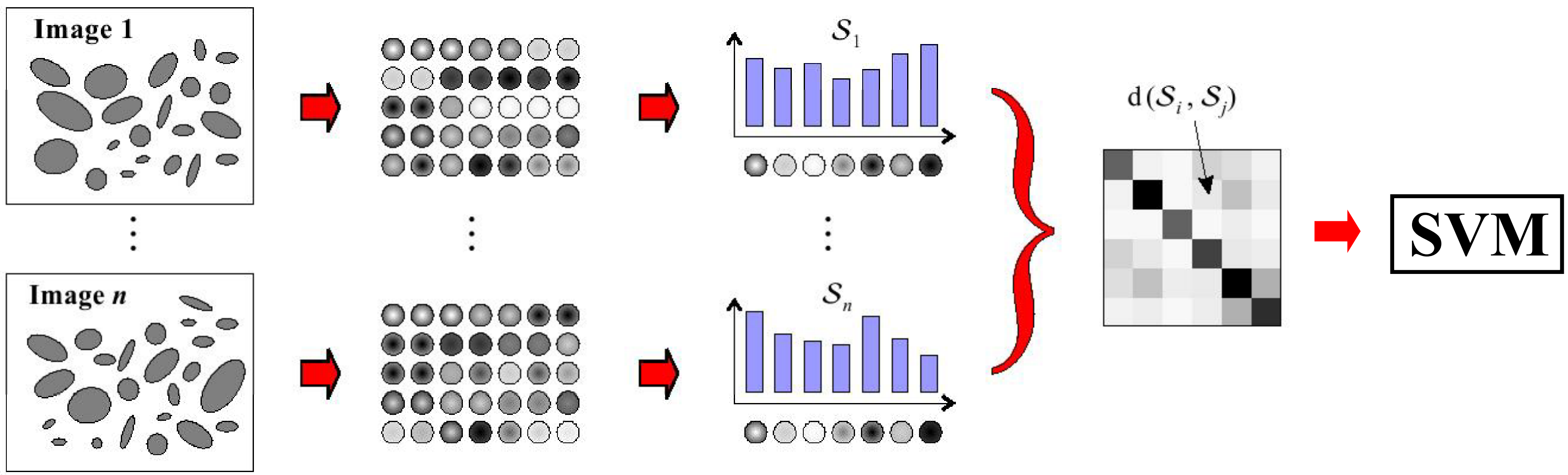
- Orderless document representation: frequencies of words from a dictionary
- Classification to determine document categories



Bag-of-words

Common	2	0	1	3
People	3	0	0	2
Sculpture	0	1	3	0
…	…	…	…	…

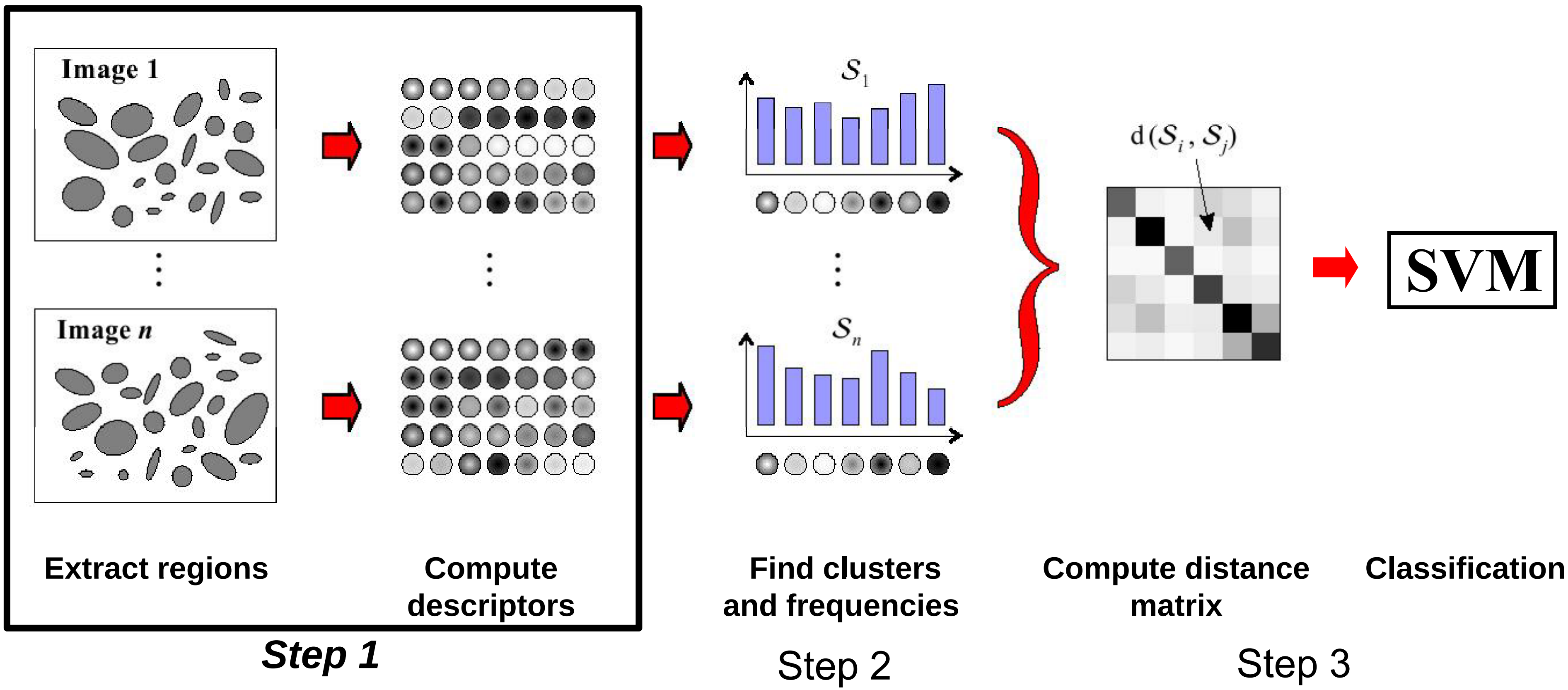
Bag-of-features for image classification



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]

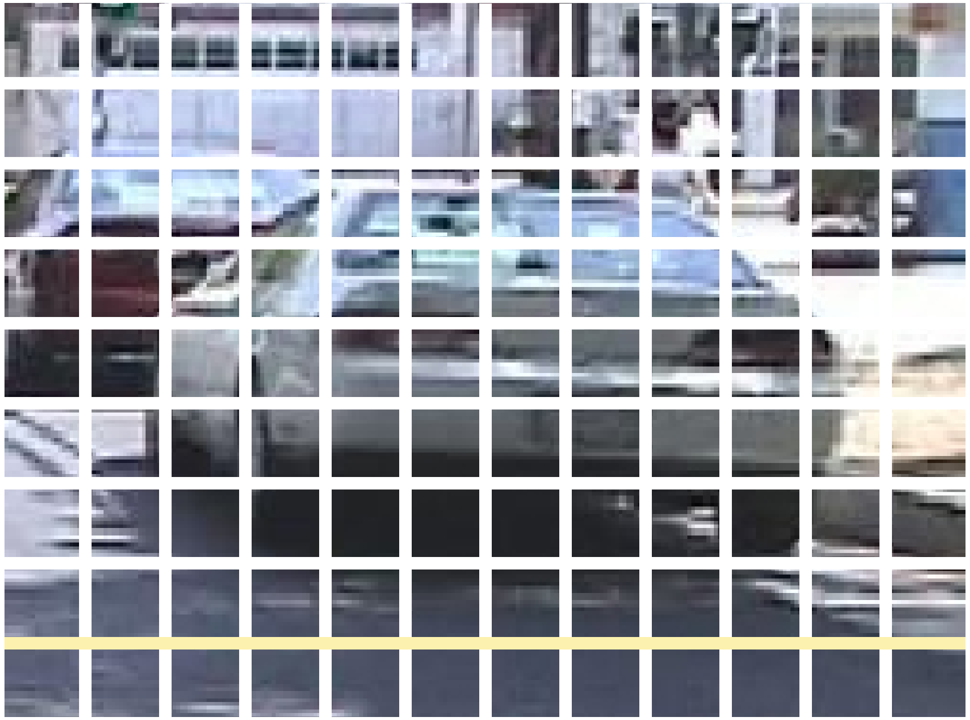
Bag-of-features for image classification



Step 1: feature extraction

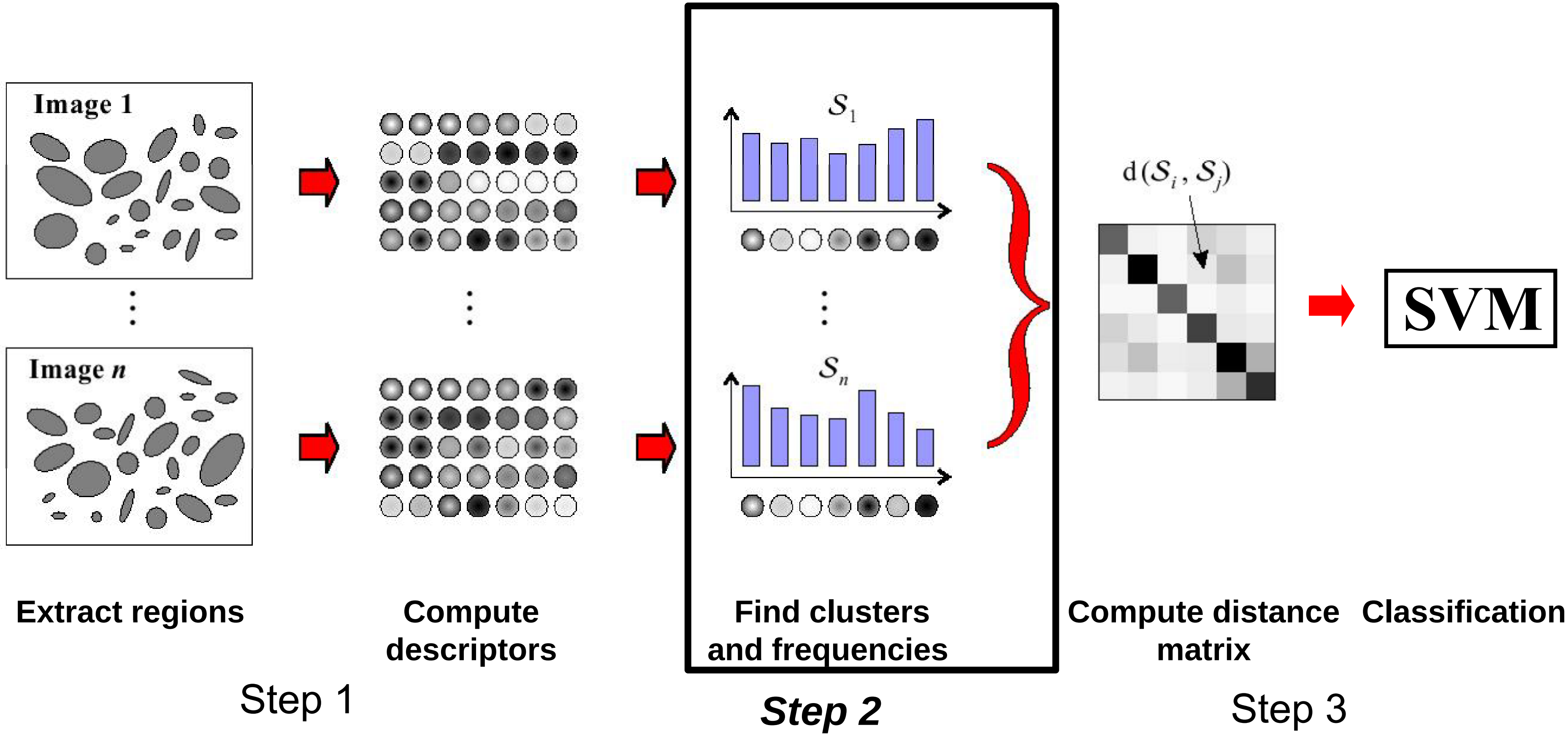
- Scale-invariant image regions + SIFT (see lecture 2)
  - Affine invariant regions give “too” much invariance
  - Rotation invariance for many realistic collections “too” much invariance
- Dense descriptors
  - Improve results in the context of categories (for most categories)
  - Interest points do not necessarily capture “all” features
- Color-based descriptors
- Shape-based descriptors

Dense features

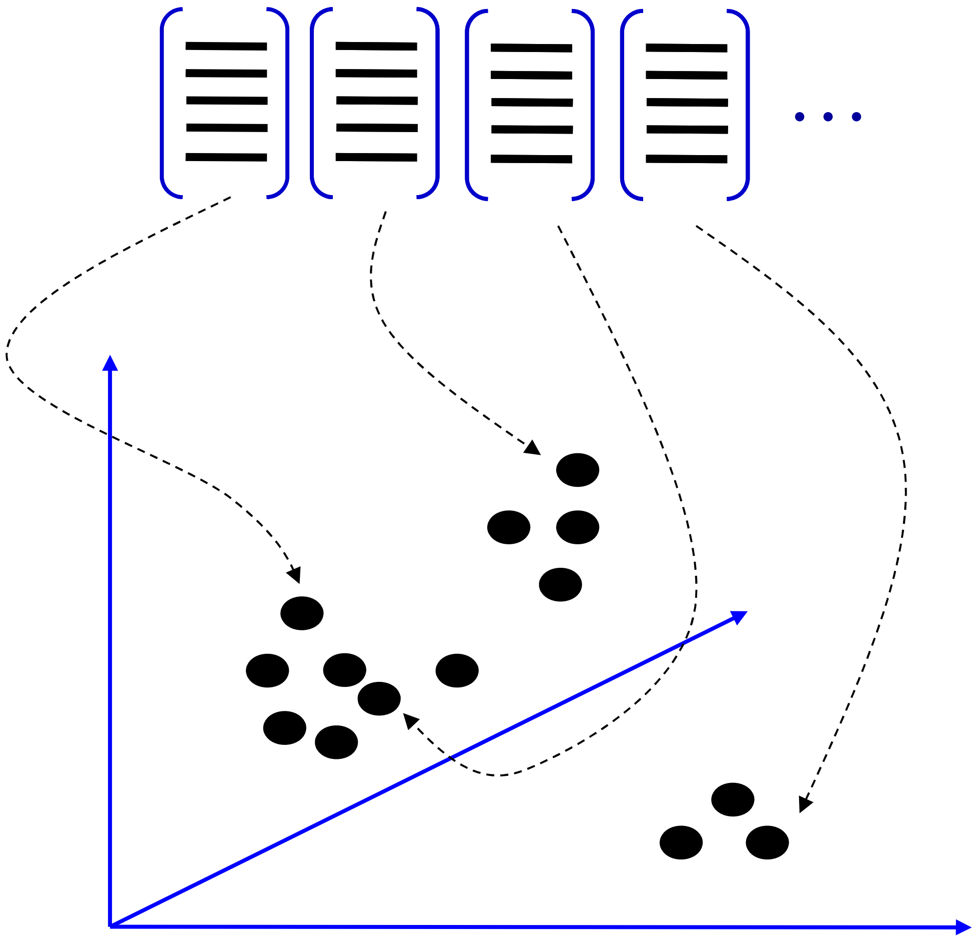


- Multi-scale dense grid: extraction of small overlapping patches at multiple scales
- Computation of the SIFT descriptor for each grid cells
- Exp.: Horizontal/vertical step size 3-6 pixel, scaling factor of 1.2 per level

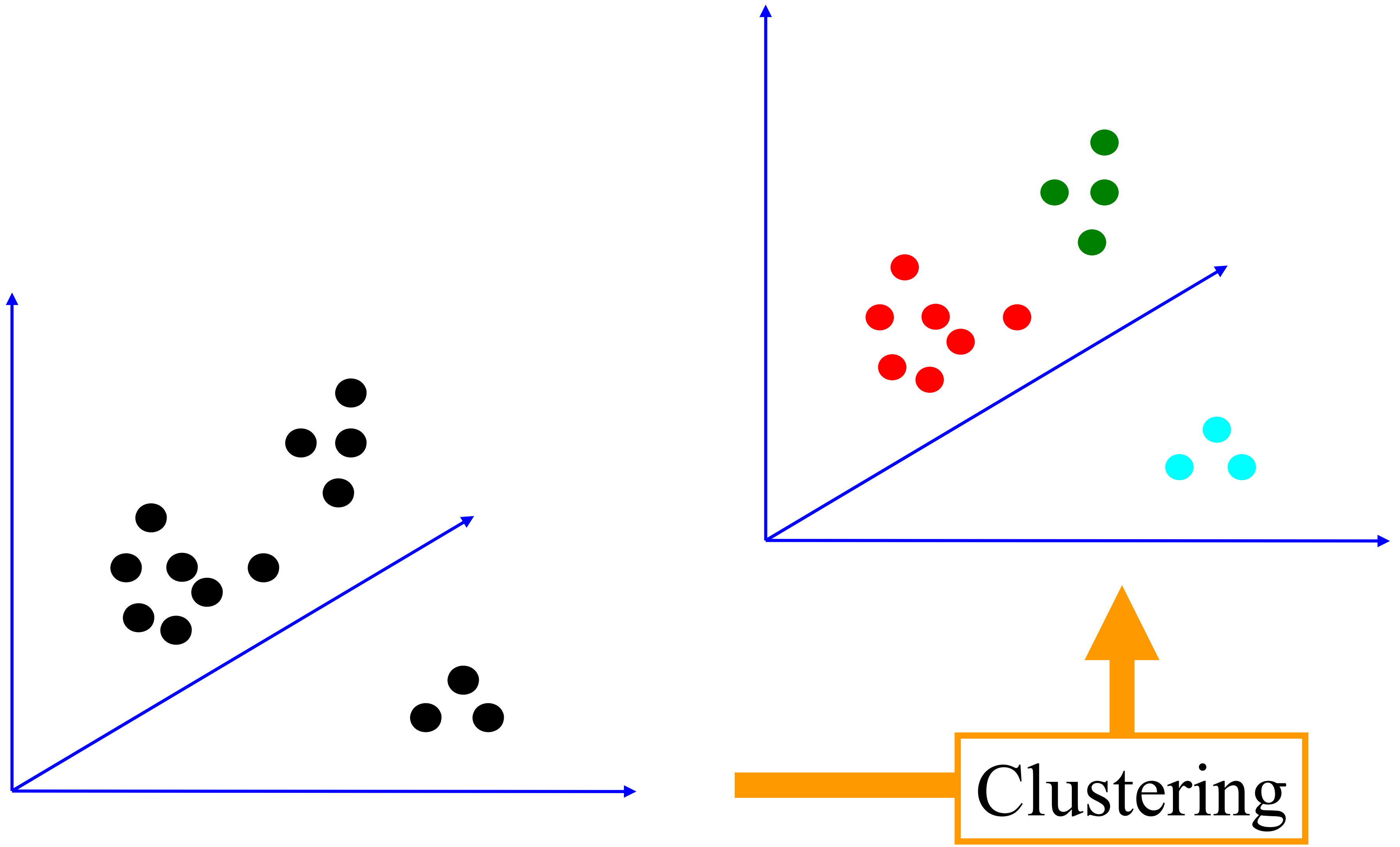
Bag-of-features for image classification



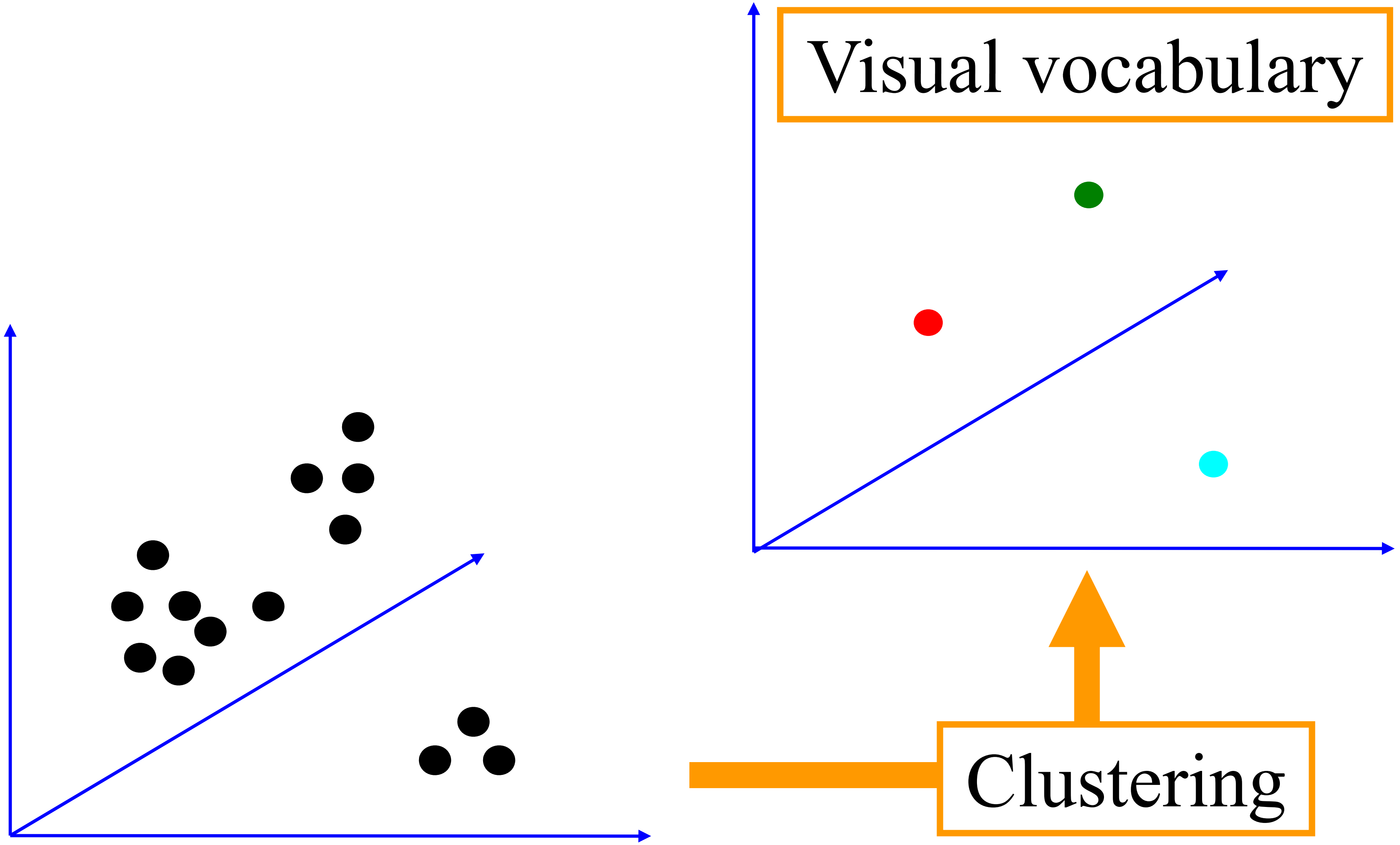
Step 2: Quantization



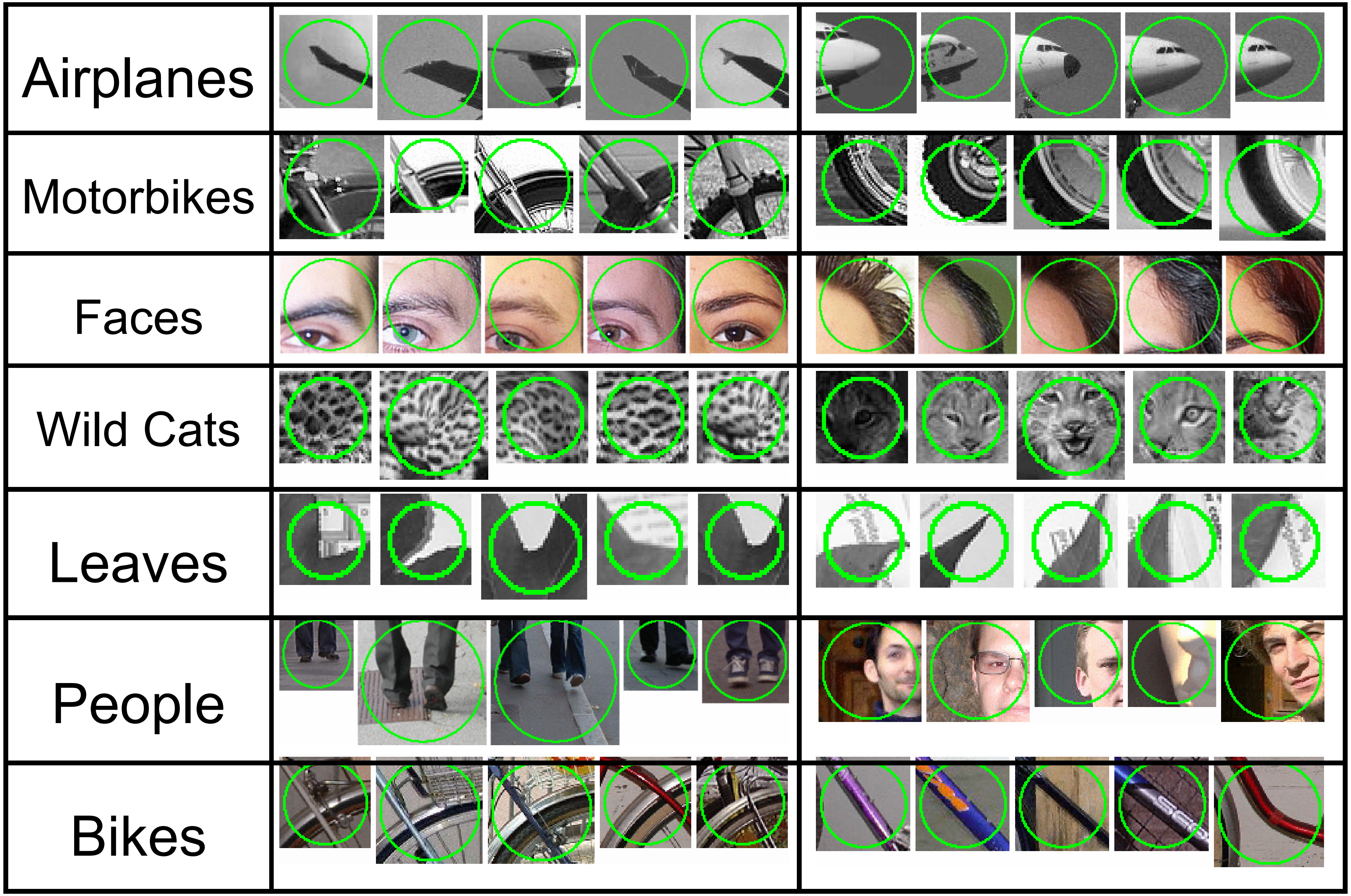
Step 2:Quantization



Step 2: Quantization



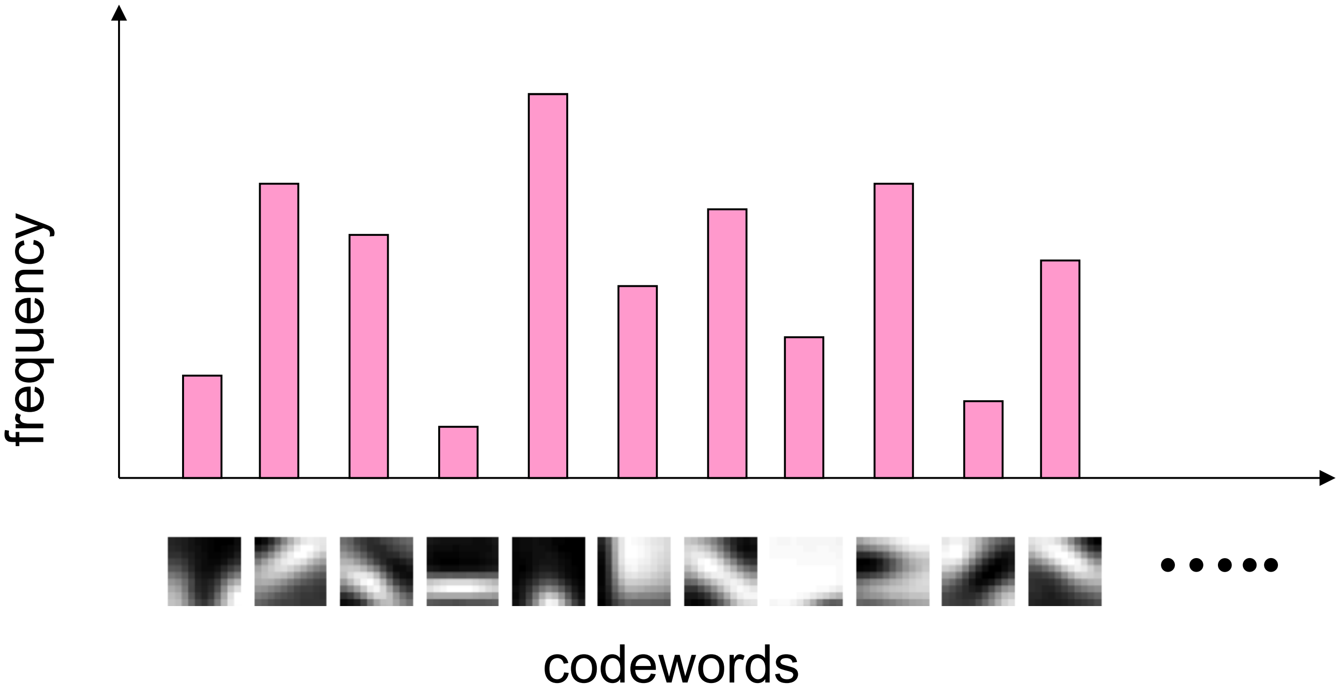
Examples for visual words



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



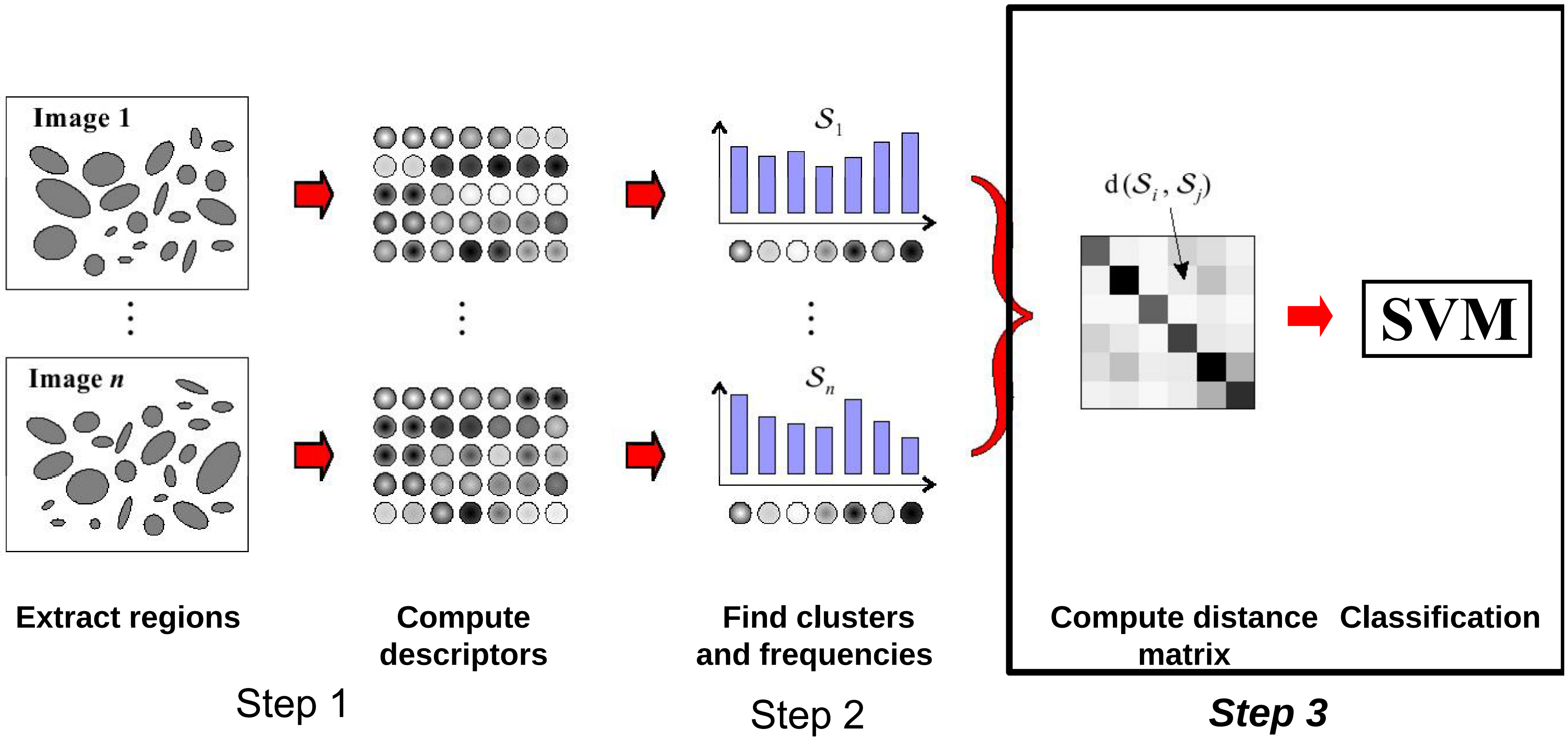
• each image is represented by a vector

• typically 1000-4000 dimension

• fine grained – represent model instances

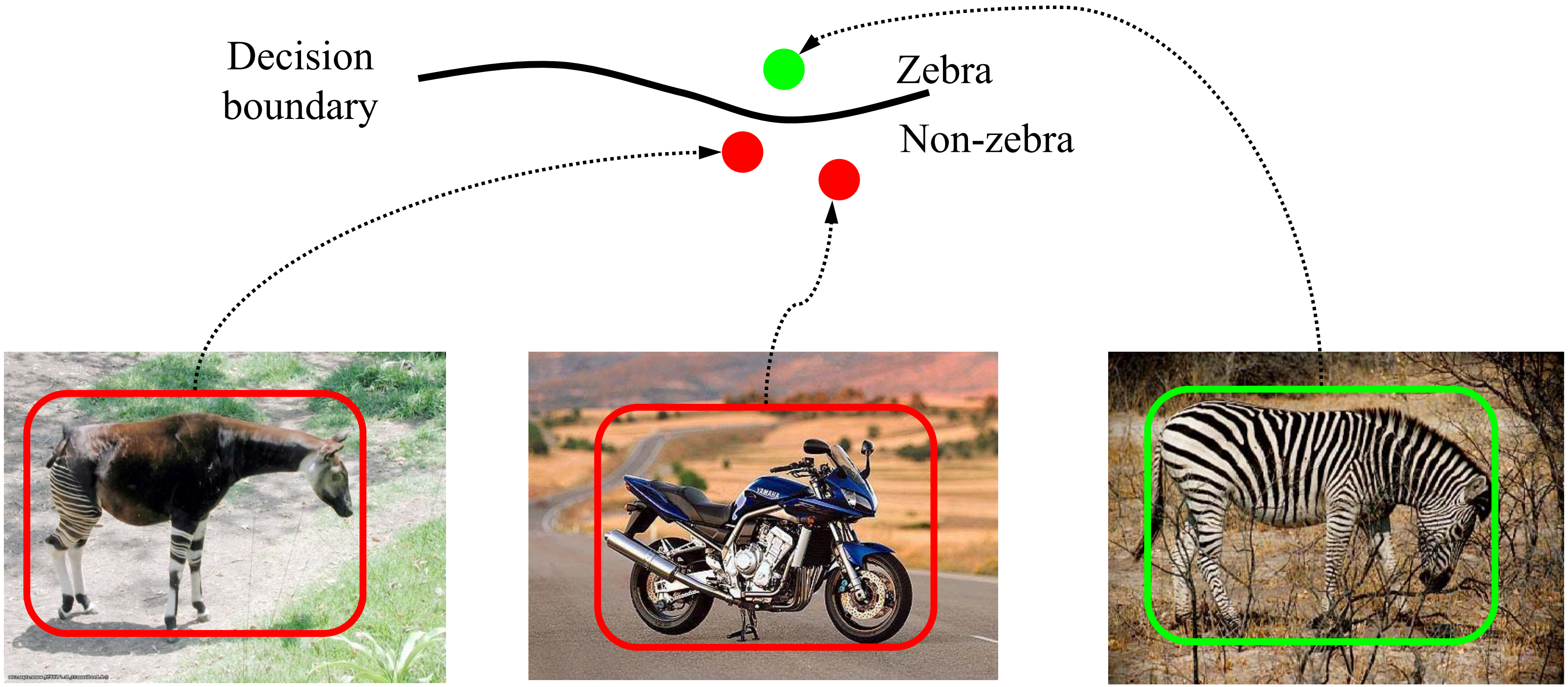
• coarse grained – represent object categories

Bag-of-features for image classification



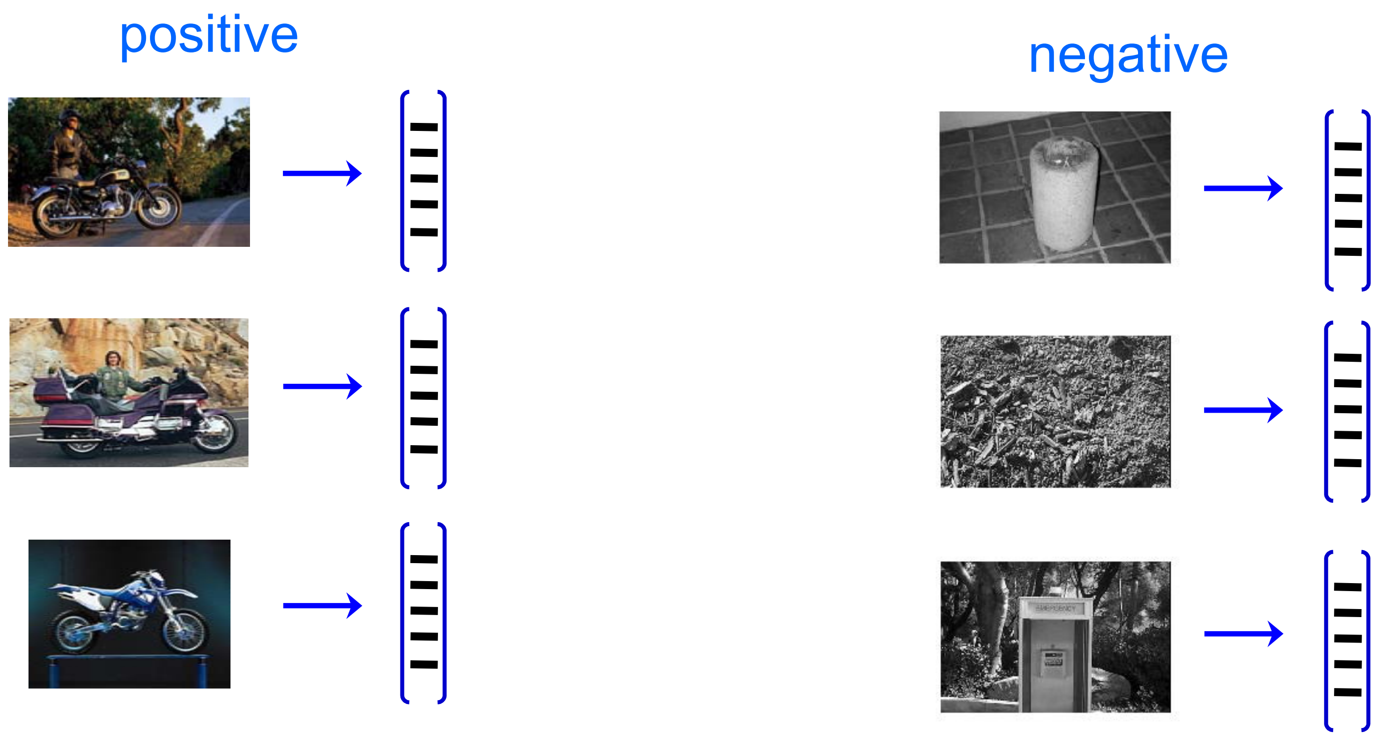
Step 3: Classification

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



Training data

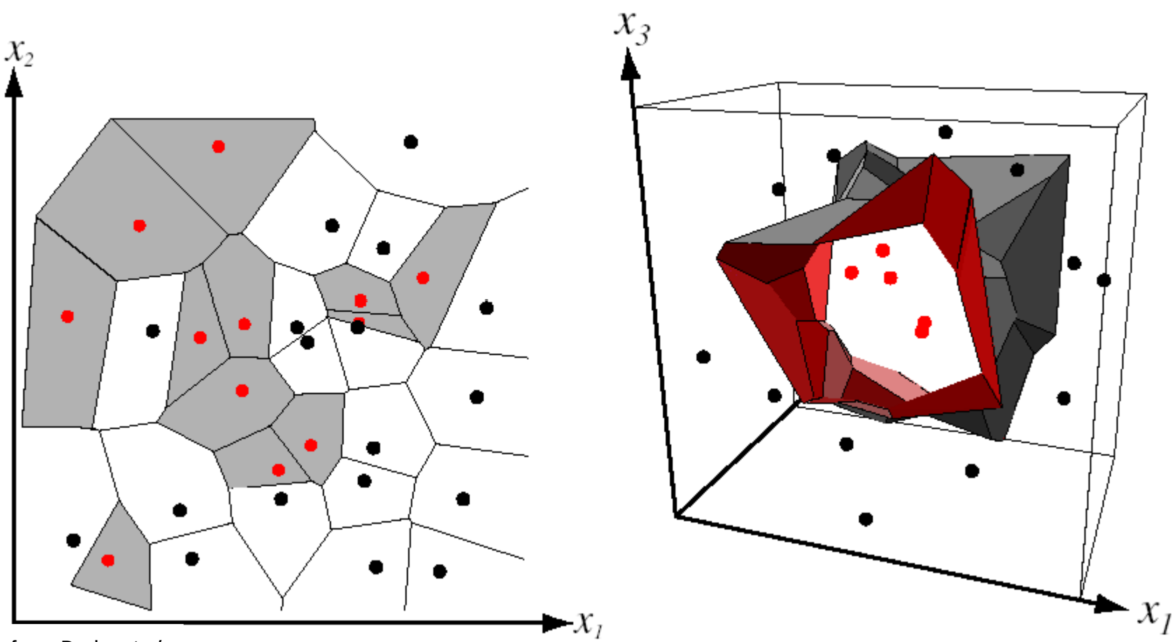
Vectors are histograms, one from each training image



Train classifier,e.g.SVM

Nearest Neighbor Classifier

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



from Duda *et al.*

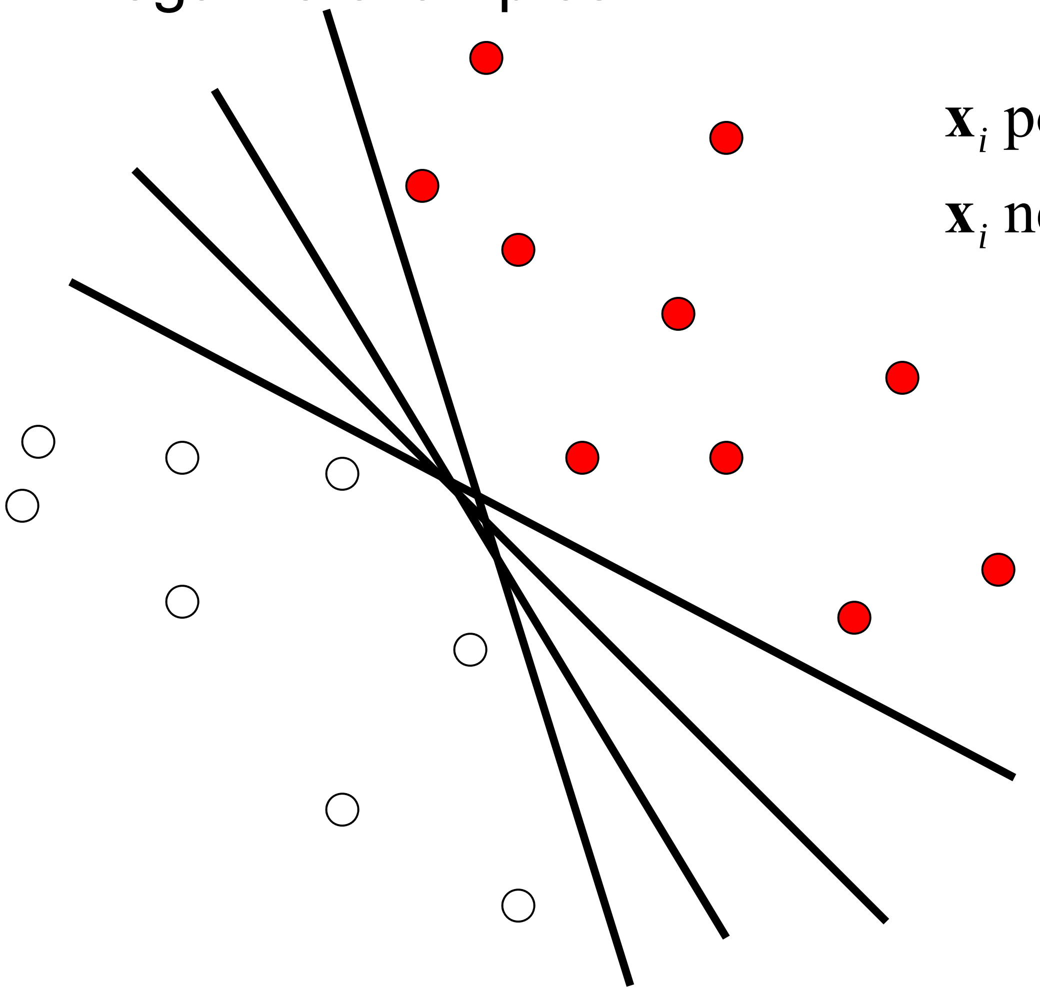
Voronoi partitioning of feature space
for 2-category 2-D and 3-D data

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



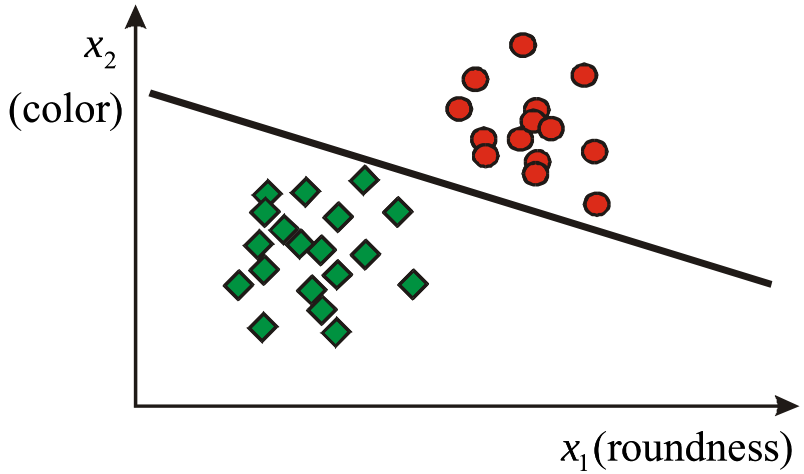
$$\mathbf{x}_i \text{ positive}: \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0$$

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

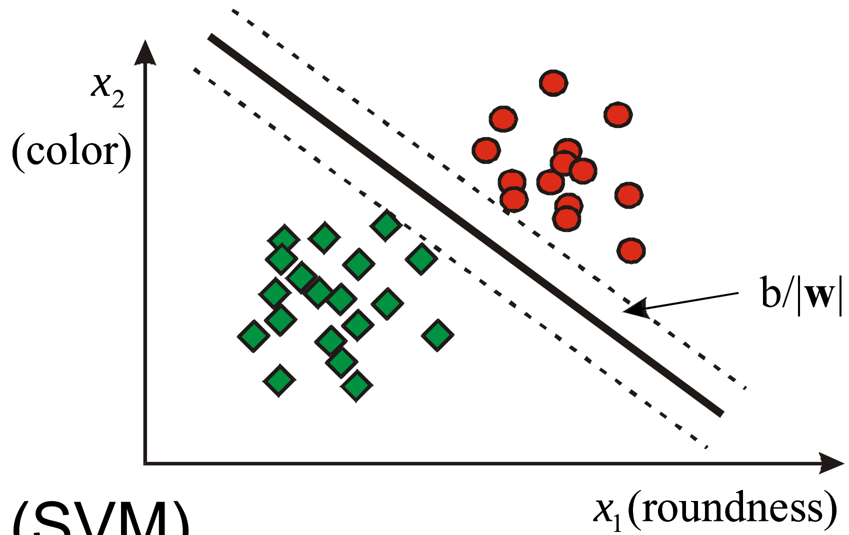
Which hyperplane
is best?

Linear classifiers - margin

- Generalization is not good in this case:



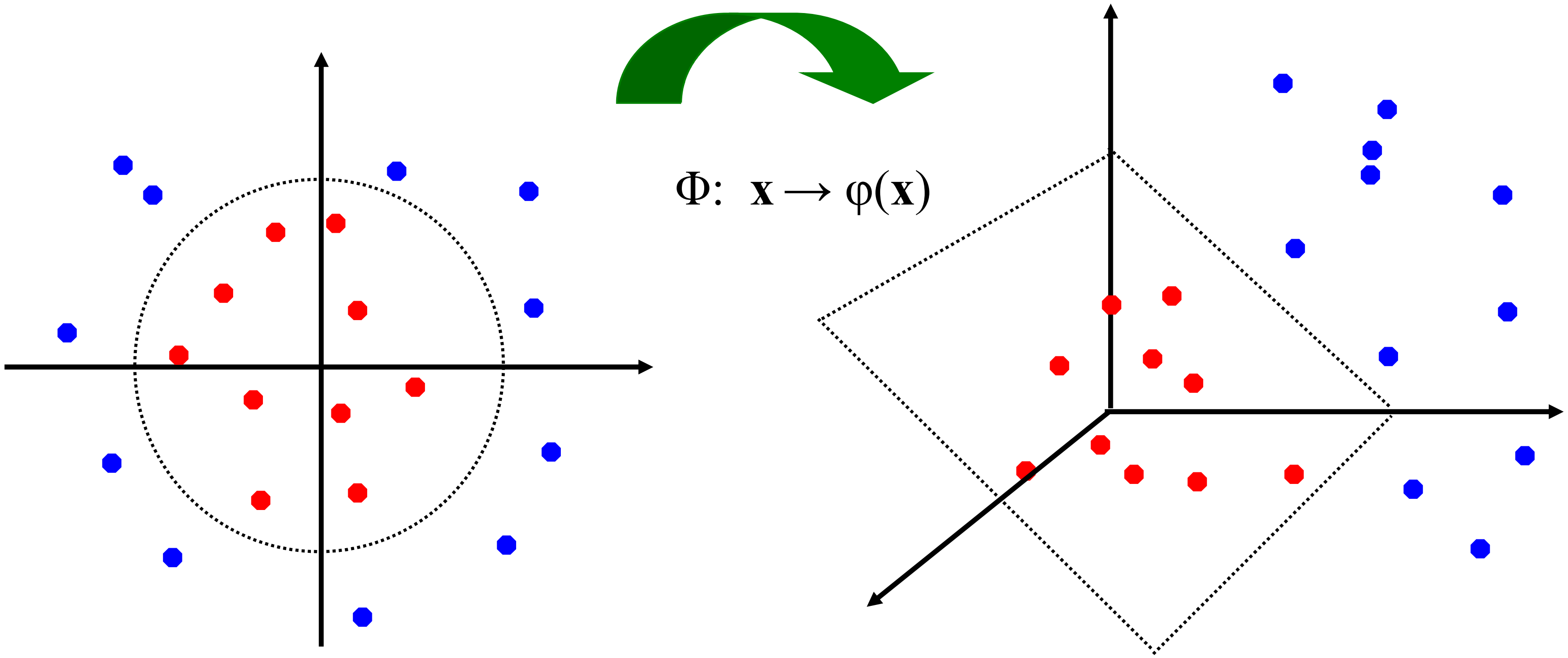
- Better if a margin is introduced:



=> Support vector machines (SVM)

Nonlinear SVMs

- General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable:



Nonlinear SVMs

- *The kernel trick*: instead of explicitly computing the lifting transformation $\varphi(\mathbf{x})$, define a kernel function K such that

$$K(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\varphi}(\mathbf{x}_i) \cdot \boldsymbol{\varphi}(\mathbf{x}_j)$$

- This gives a nonlinear decision boundary in the original feature space:

$$\sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

Kernels for bags of features

- Hellinger kernel $K(h_1, h_2) = \sum_{i=1}^{N} \sqrt{h_1(i) h_2(i)}$
- Histogram intersection kernel $I(h_1, h_2) = \sum_{i=1}^{N} \min(h_1(i), h_2(i))$
- Generalized Gaussian kernel $K(h_1, h_2) = \exp\left(-\frac{1}{A} D(h_1, h_2)^2 \right)$
- *D* can be Euclidean distance, χ^2 distance etc.

$$D_{\chi^2}(h_1, h_2) = \sum_{i=1}^{N} \frac{\left(h_1(i) - h_2(i)\right)^2}{h_1(i) + h_2(i)}$$

Combining features

- SVM with multi-channel chi-square kernel

$$K(H_i, H_j) = \exp\left(-\sum_{c \in \mathcal{C}} \frac{1}{A_c} D_c(H_i, H_j) \right)$$

- Channel *c* is a combination of detector, descriptor
- $D_c(H_i, H_j)$ is the chi-square distance between histograms

$$D_c(H_1, H_2) = \frac{1}{2} \sum_{i=1}^{m} [(h_{1i} - h_{2i})^2 / (h_{1i} + h_{2i})]$$

- A_c is the mean value of the distances between all training sample
- Extension: learning of the weights, for example with Multiple Kernel Learning (MKL)

J. Zhang, M. Marszalek, S. Lazebnik and C. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study, IJCV 2007.

Combining features

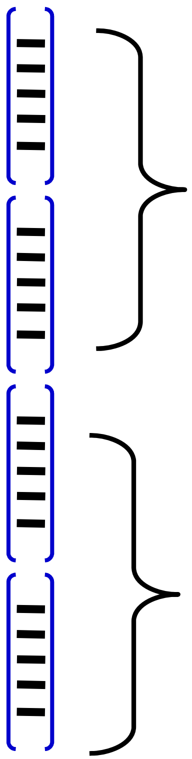
- For linear SVMs
  - Early fusion: concatenation the descriptors
  - Late fusion: learning weights to combine the classification scores
- Theoretically no clear winner
- In practice late fusion give better results
  - In particular if different modalities are combined

Multi-class SVMs

- Various direct formulations exist, but they are not widely used in practice. It is more common to obtain multi-class SVMs by combining two-class SVMs in various ways.
- One versus all:
  - Training: learn an SVM for each class versus the others
  - Testing: apply each SVM to test example and assign to it the class of the SVM that returns the highest decision value
- One versus one:
  - Training: learn an SVM for each pair of classes
  - Testing: each learned SVM “votes” for a class to assign to the test example

Why does SVM learning work?

- Learns foreground and background visual words

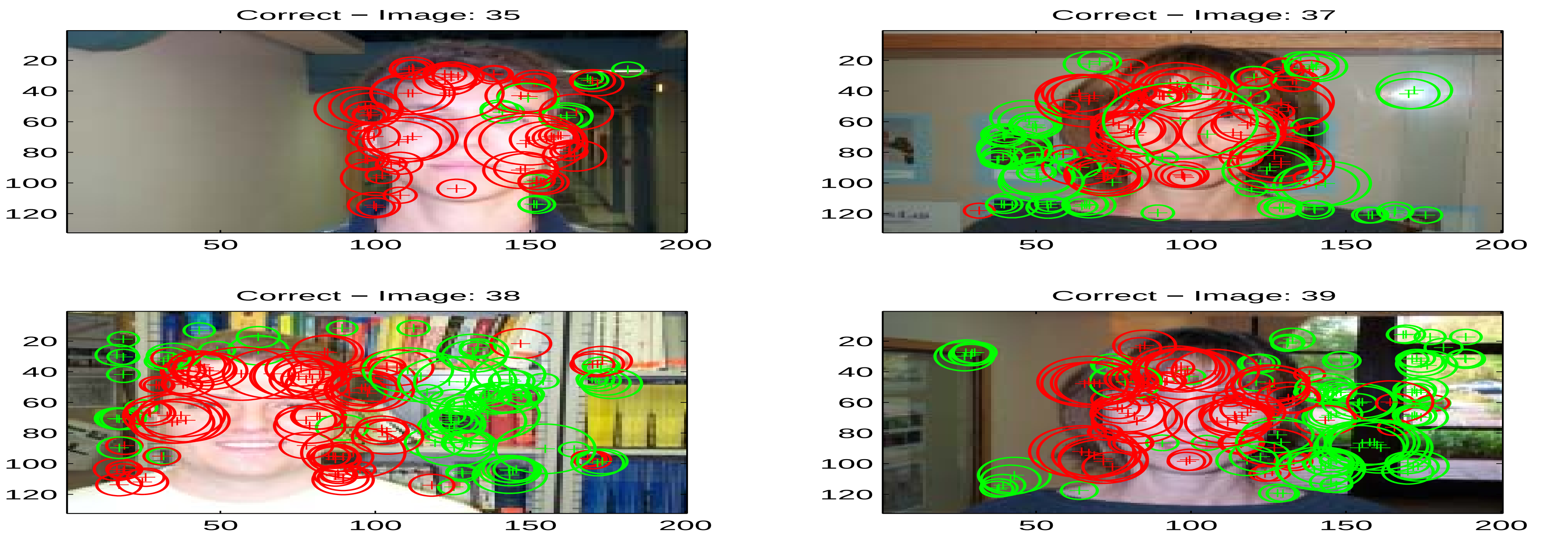


foreground words – high weight

background words – low weight

Illustration

Localization according to visual word probability



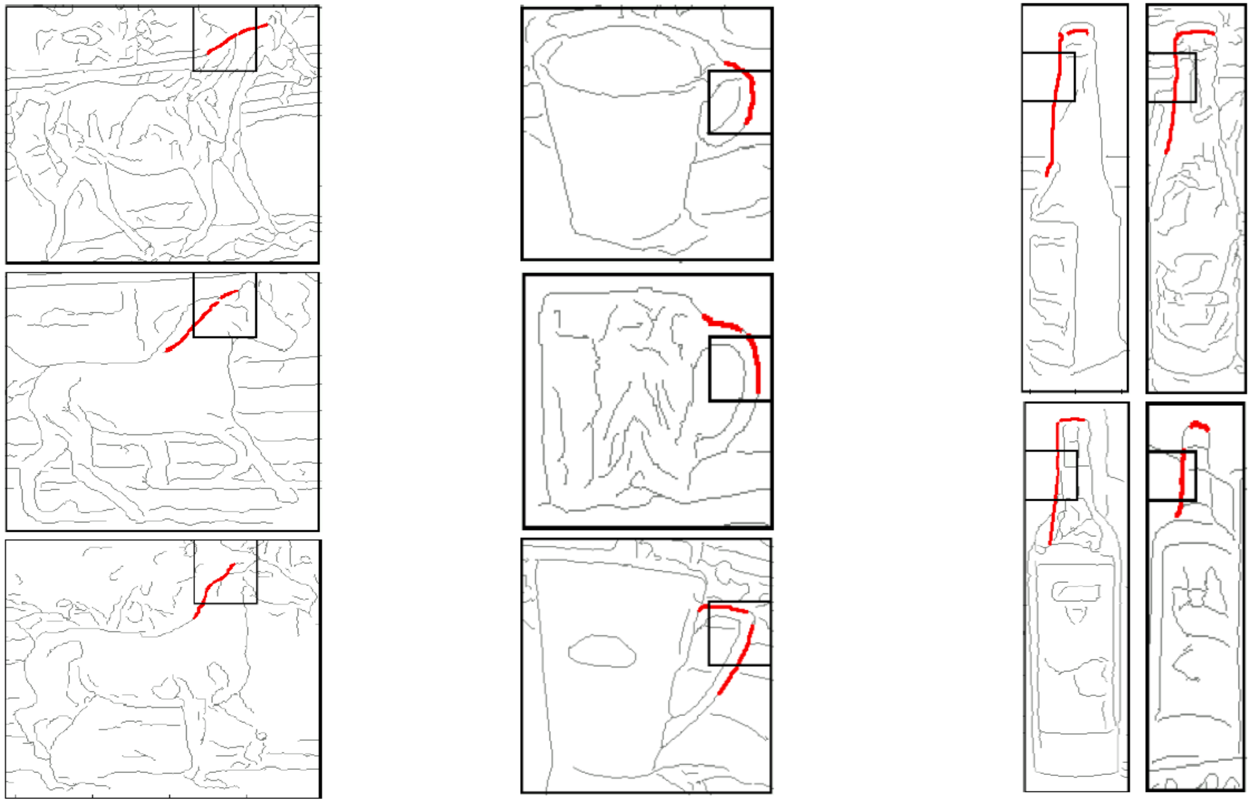
foreground word more probable

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 [05-10] 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
  - Class labels per image + bounding boxes
  - 5011 training images, 4952 test images

- Evaluation measure: average precision

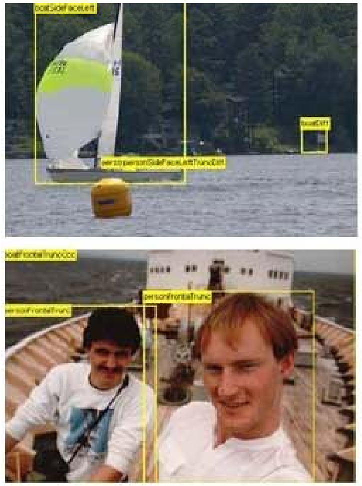
PASCAL 2007 dataset

Aeroplane

Bicycle

Bird

Boat



Bottle

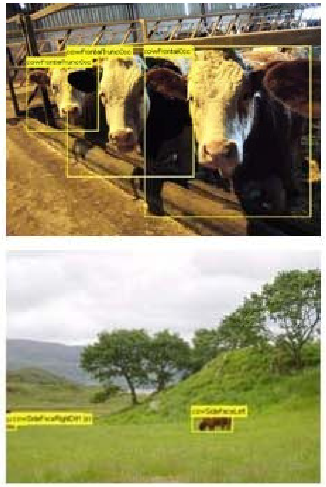
Bus

Car

Cat

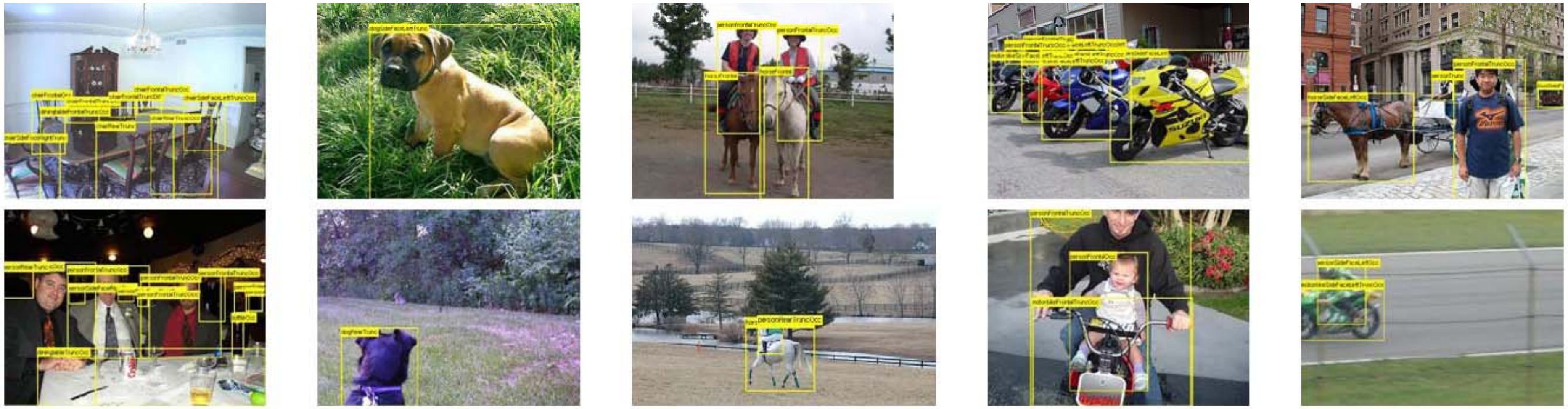
Chair

Cow

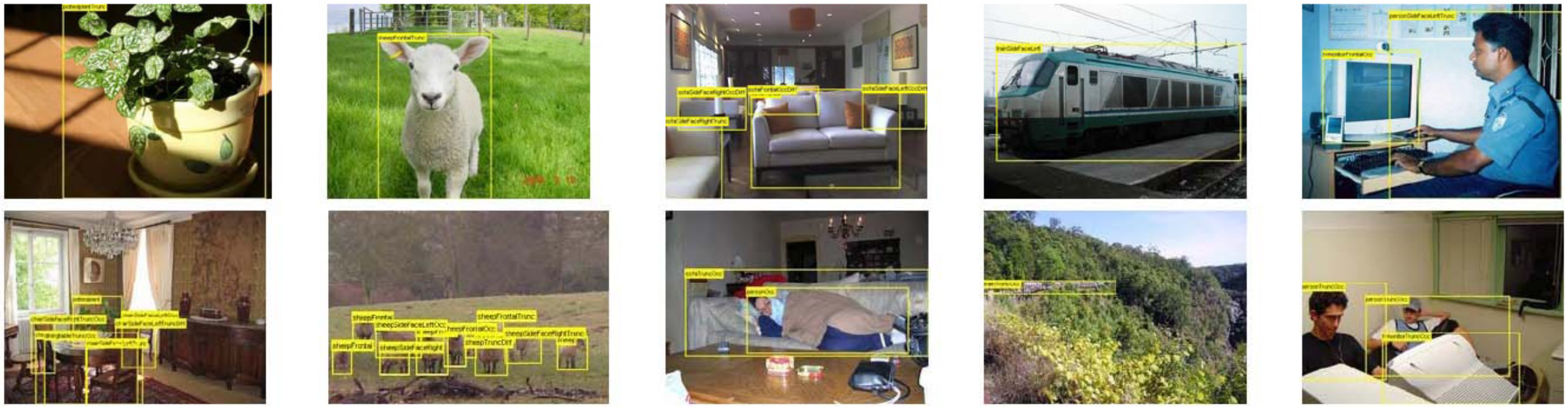


PASCAL 2007 dataset

Dining Table | Dog | Horse | Motorbike | Person

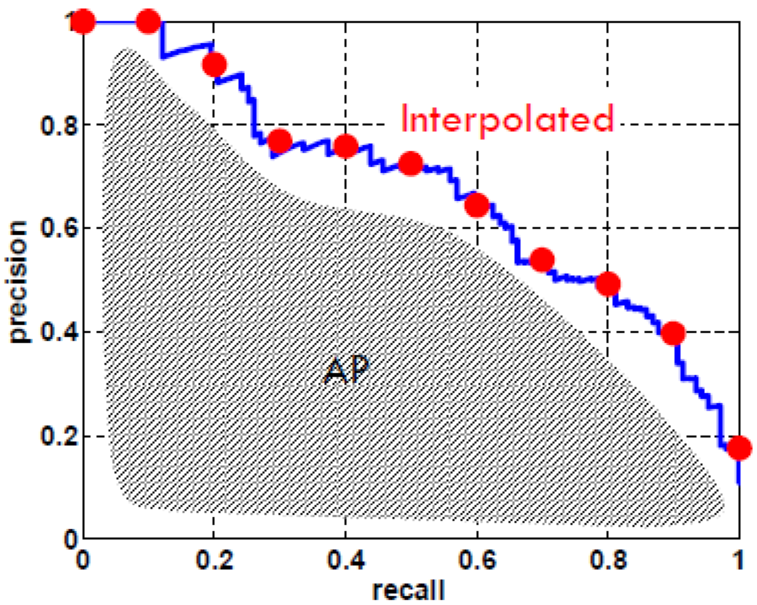


Potted Plant | Sheep | Sofa | Train | TV/Monitor



Evaluation

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

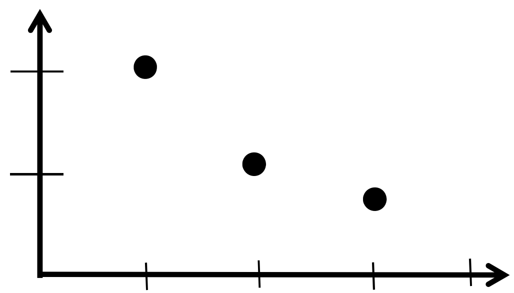


- A good score requires both high recall and high precision
- Application-independent
- Penalizes methods giving high precision but low recall

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]

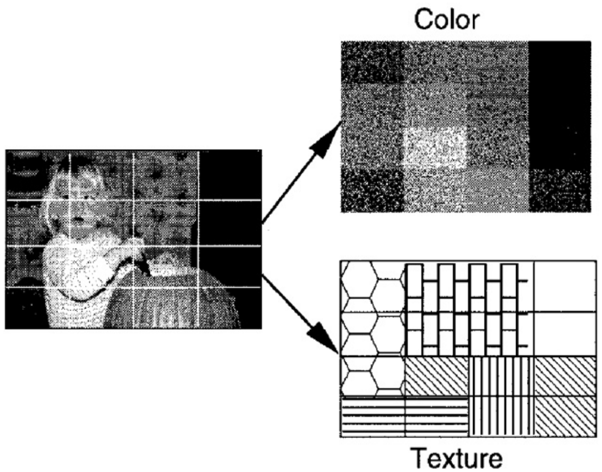
Related work

Similar approaches:

Subblock description [Szummer & Picard, 1997]

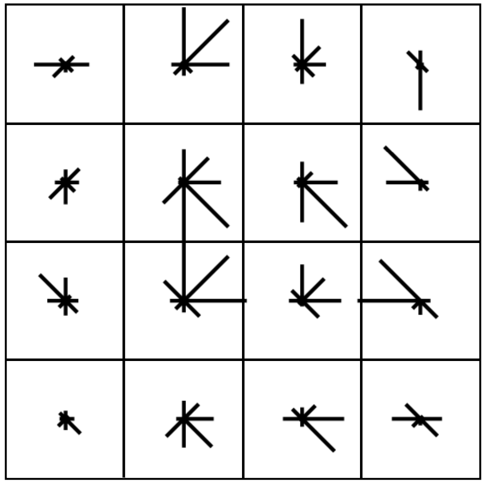
SIFT [Lowe, 1999]

GIST [Torralba et al., 2003]



Szummer & Picard (1997)

SIFT



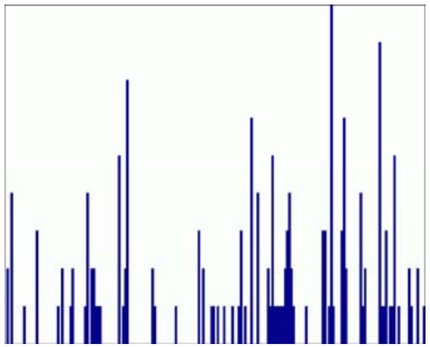
Lowe (1999, 2004)

Gist

Torralba et al. (2003)

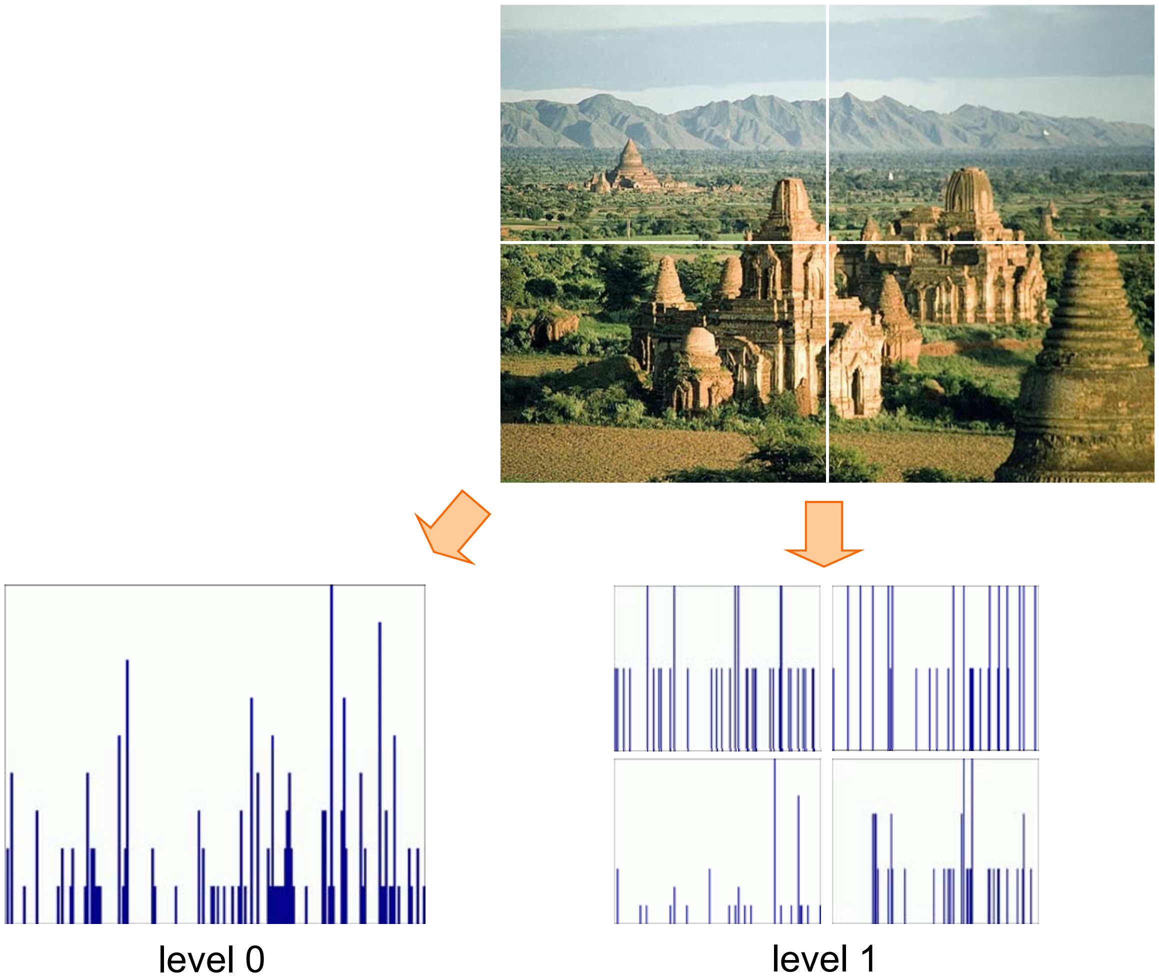
Spatial pyramid representation

Locally orderless representation at several levels of spatial resolution



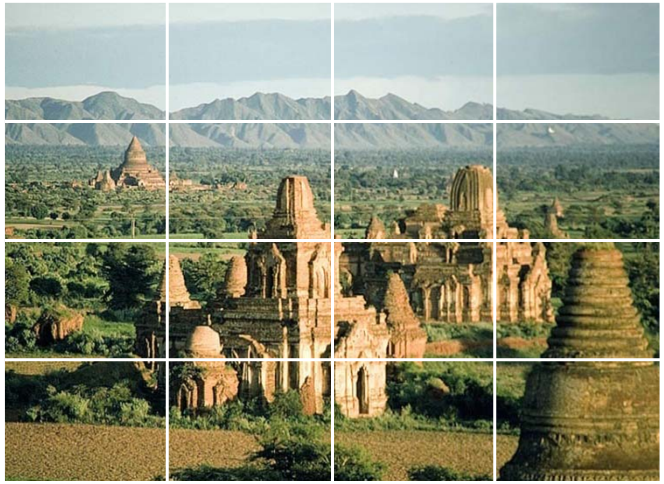
level 0

Spatial pyramid representation

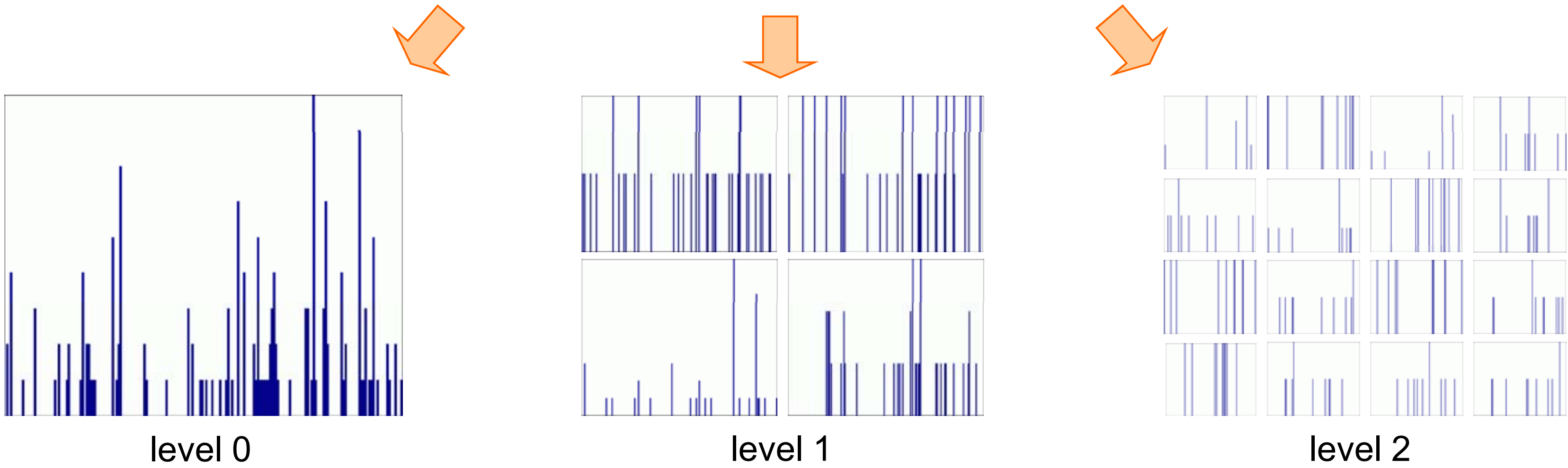


Locally orderless representation at several levels of spatial resolution

Spatial pyramid representation



Locally orderless representation at several levels of spatial resolution



Student presentation 12/12/2015

Recent extensions

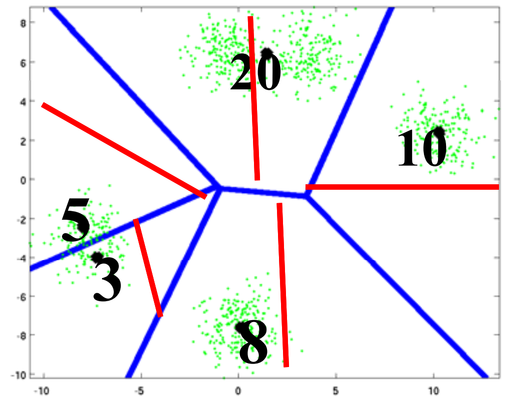
- Linear Spatial Pyramid Matching Using Sparse Coding for Image Classification. J. Yang et al., CVPR’09.
  - Local coordinate coding, linear SVM, excellent results in 2009 PASCAL challenge
- Learning Mid-level features for recognition, Y. Boureau et al., CVPR’10.
  - Use of sparse coding techniques and max pooling

Recent extensions

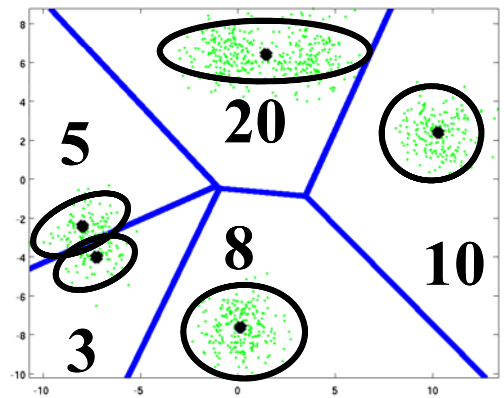
- Efficient Additive Kernels via Explicit Feature Maps, A. Vedaldi and Zisserman, CVPR'10.
  - approximation by linear kernels
- Improving the Fisher Kernel for Large-Scale Image Classification, Perronnin et al., ECCV'10
  - More discriminative descriptor, power normalization, linear SVM
- Excellent results of the Fisher vector in a recent evaluation, Chatfield et al. BMVC 2011

Fisher vector image representation

- Mixture of Gaussian/ k-means stores nr of points per cell



- Fisher vector adds 1st & 2nd order moments
  - More precise description of regions assigned to cluster
  - Fewer clusters needed for same accuracy
  - Per cluster store: mean and variance of data in cell
  - Representation 2D times larger, at same computational cost
  - High dimensional, robust representation



Relation to BOF

FV formulas:

$$\mathcal{G}^X_{\mu,i} = \frac{1}{T\sqrt{w_i}} \sum_{t=1}^{T} \gamma_t(i) \left(\frac{x_t - \mu_i}{\sigma_i}\right)$$

$$\mathcal{G}^X_{\sigma,i} = \frac{1}{T\sqrt{2w_i}} \sum_{t=1}^{T} \gamma_t(i) \left[\frac{(x_t - \mu_i)^2}{\sigma_i^2} - 1\right]$$

Soft BOV formula:

$$\frac{1}{T}\sum_{t=1}^{T} \gamma_t(i)$$

The FV extends the BOV and includes higher-order statistics (up to 2^{nd} order)

Results on VOC 2007: BOV = 43.6 % $\rightarrow$ FV = 57.7 % $\rightarrow$ √FV = 62.1 %

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

Large-scale image classification

- Classification approach
  - Linear classifier
  - One-versus-rest classifiers
  - Stochastic gradient descent (SGD)
  - At each step choose a sample at random and update the parameters using a sample-wise estimate of the regularized risk

- Data reweighting
  - When some classes are significantly more populated than others, rebalancing positive and negative examples
  - Empirical risk with reweighting

$$\frac{\rho}{N_+} \sum_{i \in I_+} L_{\mathrm{OVR}}(\mathbf{x}_i, y_i; \mathbf{w}) + \frac{1-\rho}{N_-} \sum_{i \in I_-} L_{\mathrm{OVR}}(\mathbf{x}_i, y_i; \mathbf{w})$$

$\rho = 1/2$ Natural rebalancing, same weight to positive and negatives

Experimental results

- 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%)

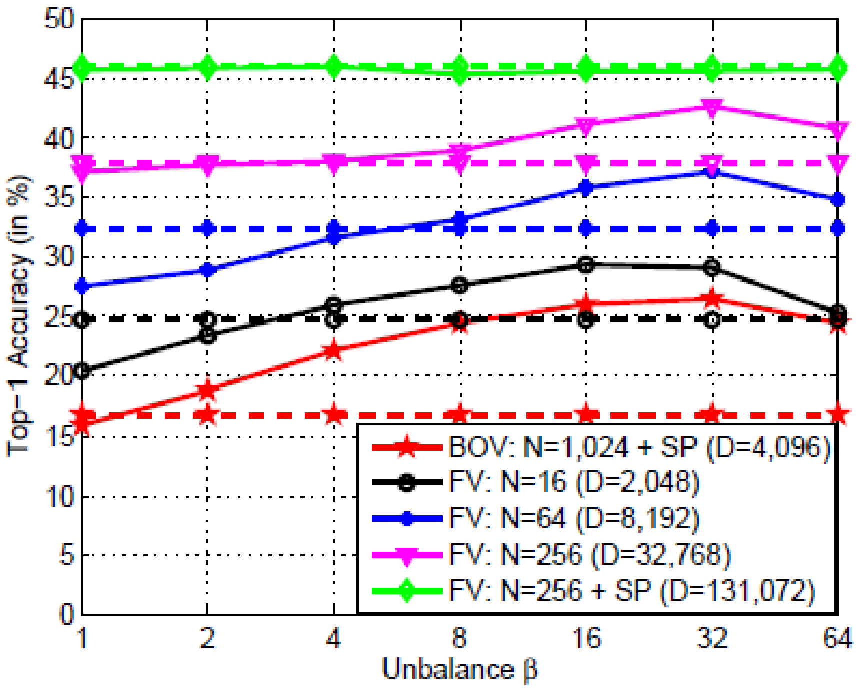
Experimental results

- Features: dense SIFT, reduced to 64 dim with PCA

- Fisher vectors
  - 256 Gaussians, using mean and variance
  - Spatial pyramid with 4 regions
  - Approx. 130K dimensions (4x [2x64x256])
  - Normalization: square-rooting and L2 norm

- BOF: dim 1024 + R=4
  - 4960 dimensions
  - Normalization: square-rooting and L2 norm

Importance of re-weighting



- Plain lines correspond to w-OVR, dashed one to u-OVR
- ß is number of negatives samples for each positive, β=1 natural rebalancing
- Results for ILSVRC 2010

- Significant impact on accuracy
- For very high dimensions little impact

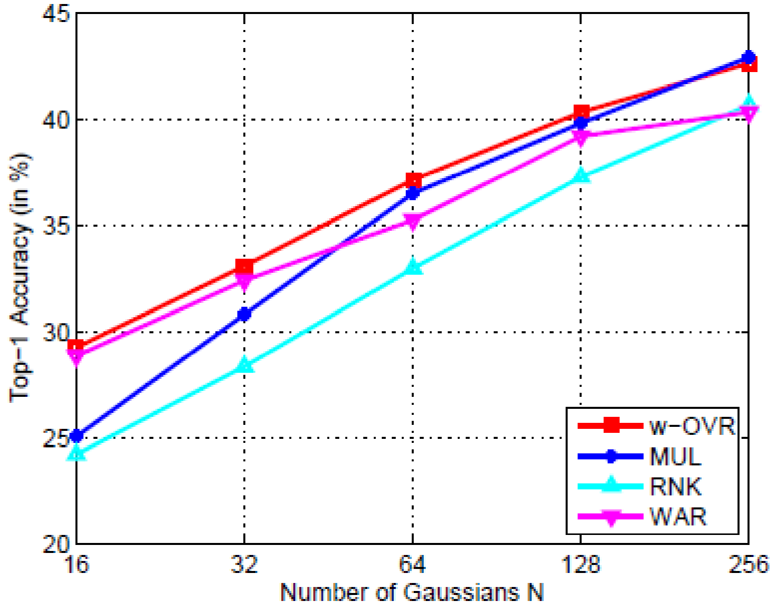
One-versus-rest works

- 256 Gaussian Fisher vector + SP with R=4 (dim 130k)
- BOF dim=1024 + SP with R=4 (dim 4000)
- Results for ILSVRC 2010
- FV >> BOF

		w-OVR
Top-1	BOV	26.4
	FV	45.7

Impact of the image signature size

- Fisher vector (no SP) for varying number of Gaussians + different classification methods, ILSVRC 2010



- Performance improves for higher dimensional vectors

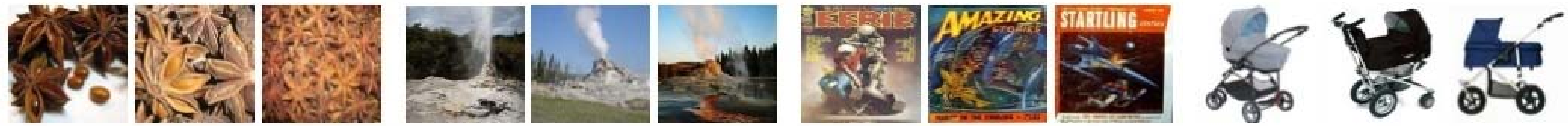
Large-scale experiment on ImageNet10k

	u-OVR	w-OVR
BOV 4K-dim	3.8	7.5
FV 130K-dim	16.7	19.1

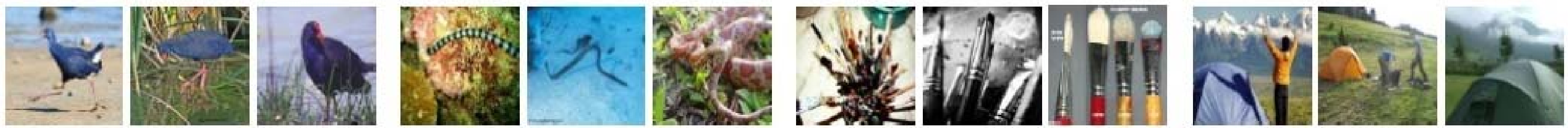
- Significant gain by data re-weighting, even for high-dimensional Fisher vectors
- w-OVR > u-OVR

Large-scale experiment on ImageNet10k

- Illustration of results obtained with w-OVR and 130K-dim Fisher vectors, ImageNet10K top-1 accuracy



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



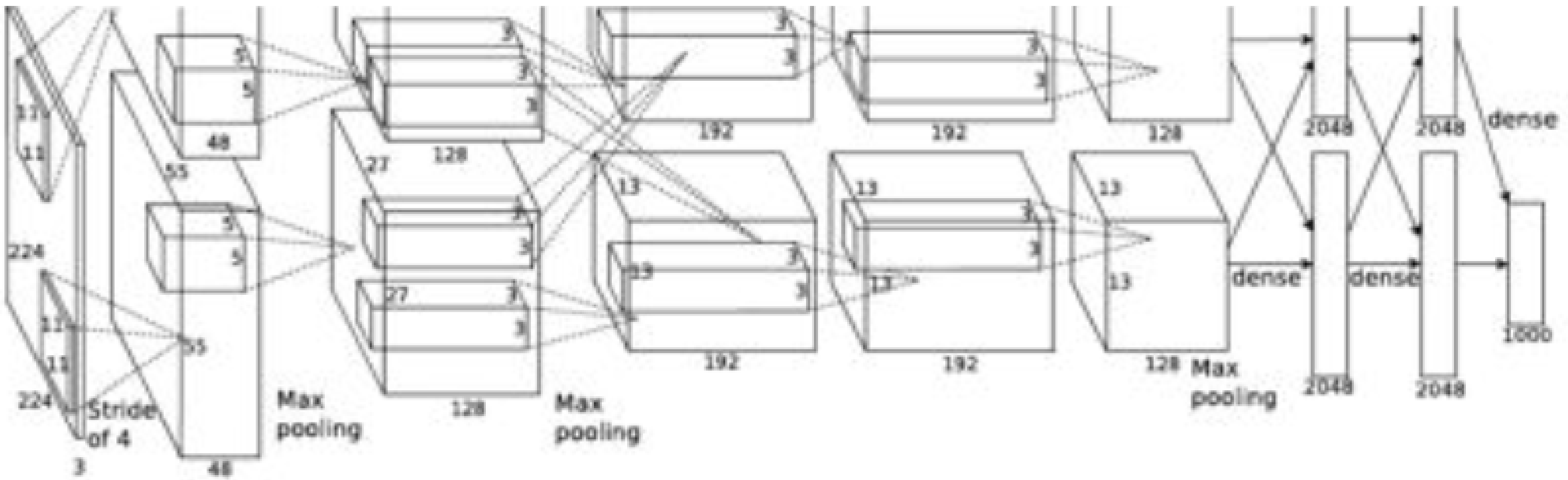
(e) European gallinule (15.00%) (f) Sea Snake (10.00 %) (g) Paintbrush (4.68 %) (h) Mountain Tent (0.00%)

Large-scale classification

- *Stochastic training:* learning with SGD is well-suited for large-scale datasets
- *One-versus-rest*: a flexible option for large-scale image classification
- *Class imbalance:* optimize the imbalance parameter in one-versus-rest strategy is a must for competitive performance

Large-scale image classification

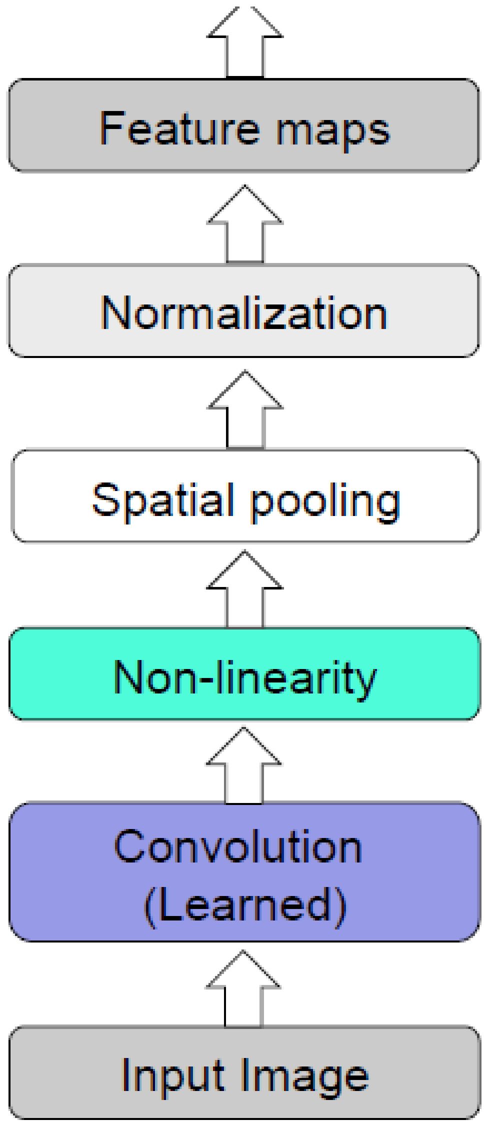
- Convolutional neural networks (CNN)
- Large model (7 hidden layers, 650k unit, 60M parameters)
- Requires large training set (ImageNet)
- GPU implementation (50x speed up over CPU)



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

Convolutional neural networks

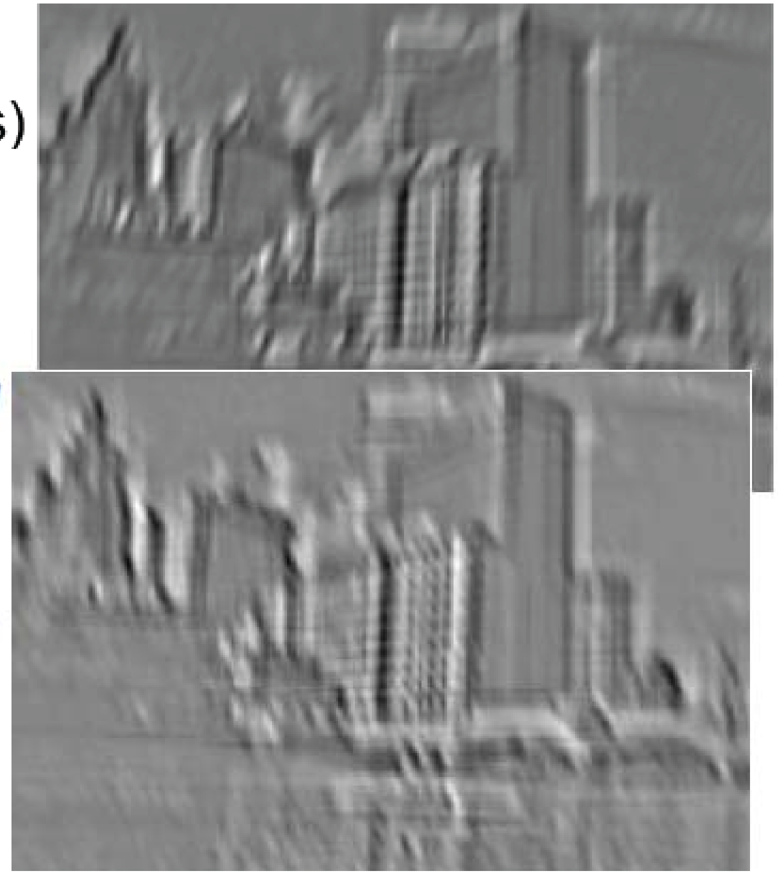
- Feed-forward feature extraction:
    1. Convolve input with learned filters
    2. Non-linearity
    3. Spatial pooling
    4. Normalization
- Supervised training of convolutional filters by back-propagating classification error



1. Convolution

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

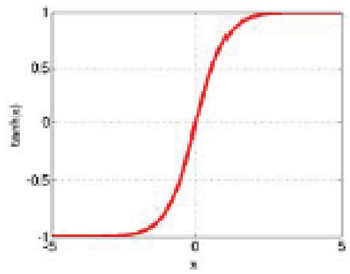
Input

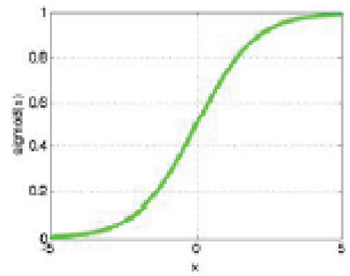


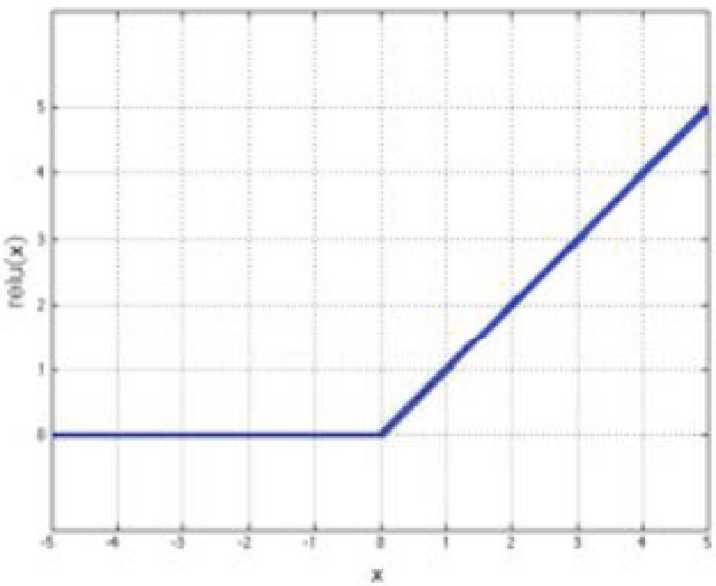
Feature Map

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

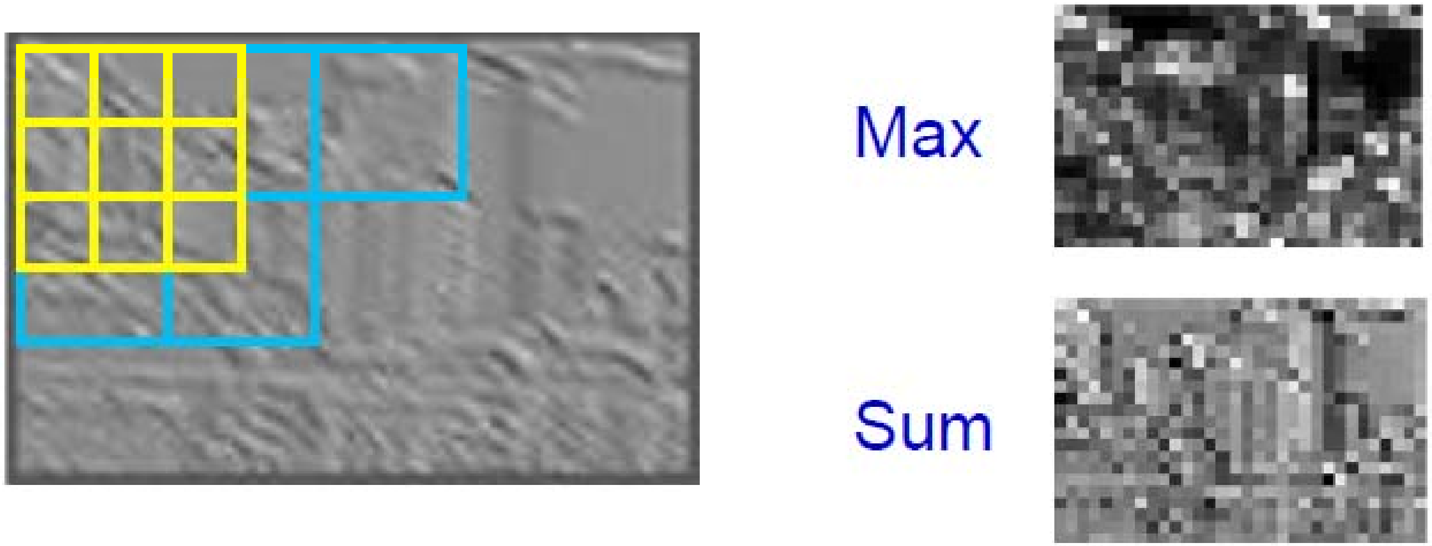






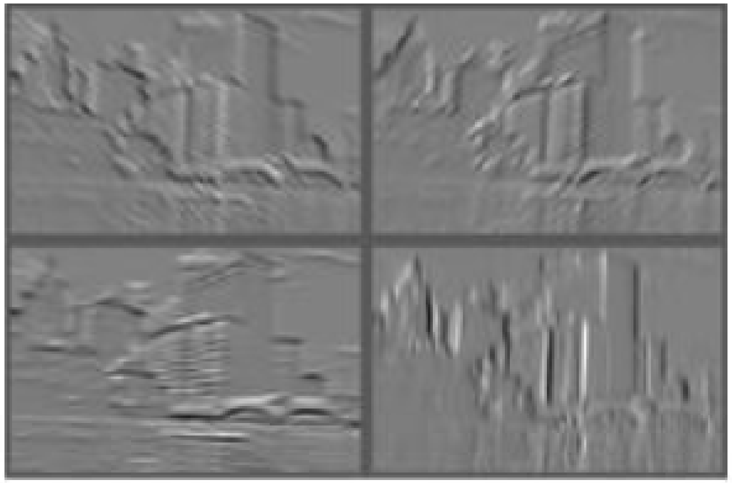
3. Spatial pooling

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

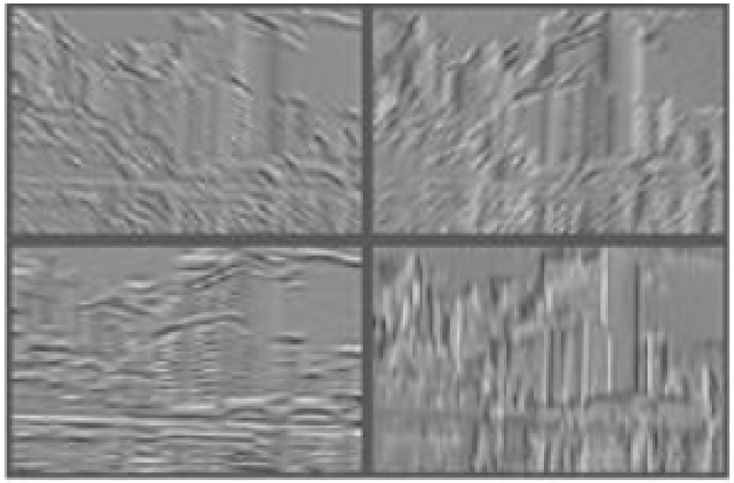


4. Normlization

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



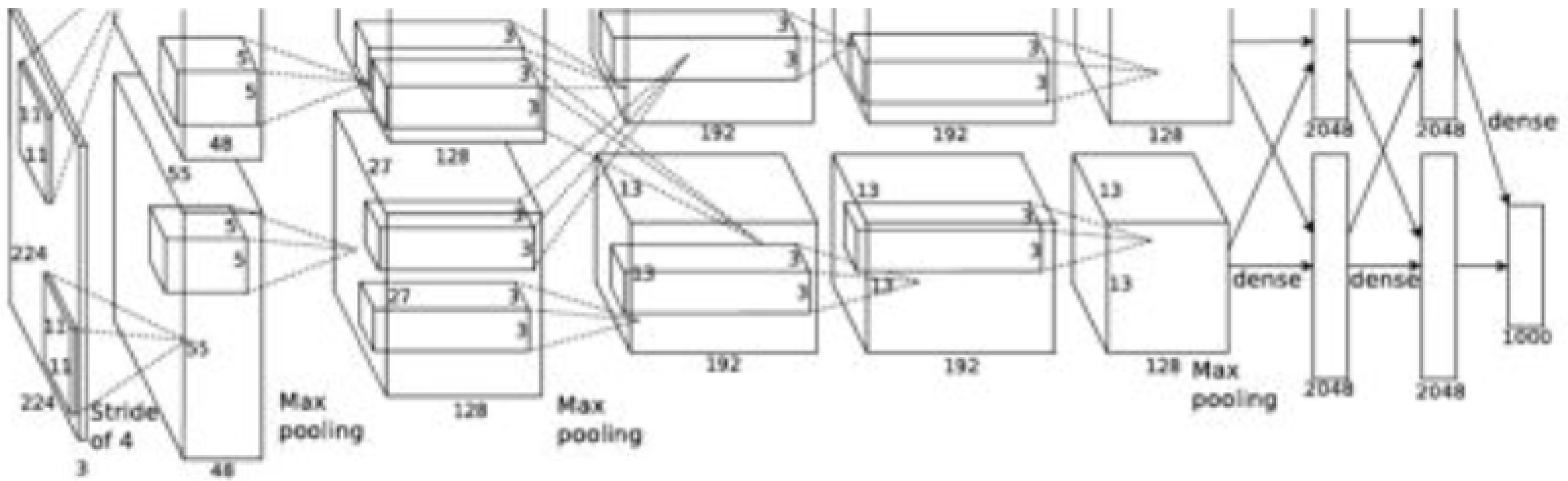
Feature Maps



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