Bag-of-features for category classification

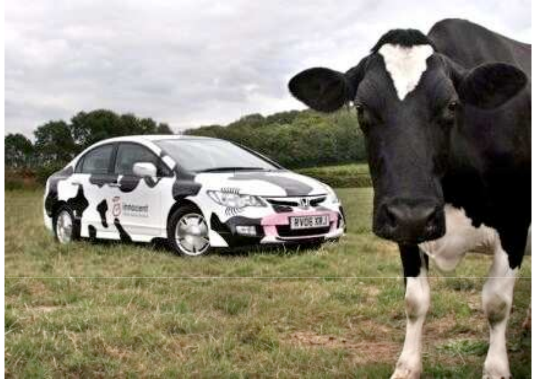
Cordelia Schmid

INRIA

LEAR

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

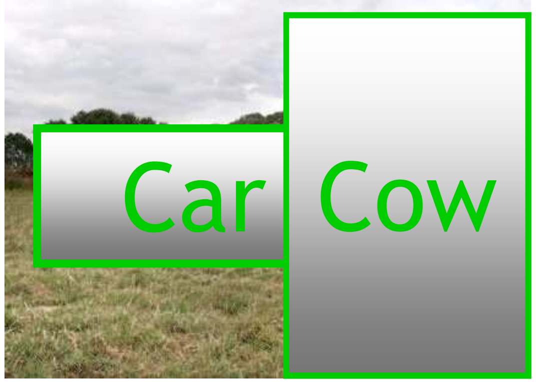
Cow: present

Bike: not present

Horse: not present

…

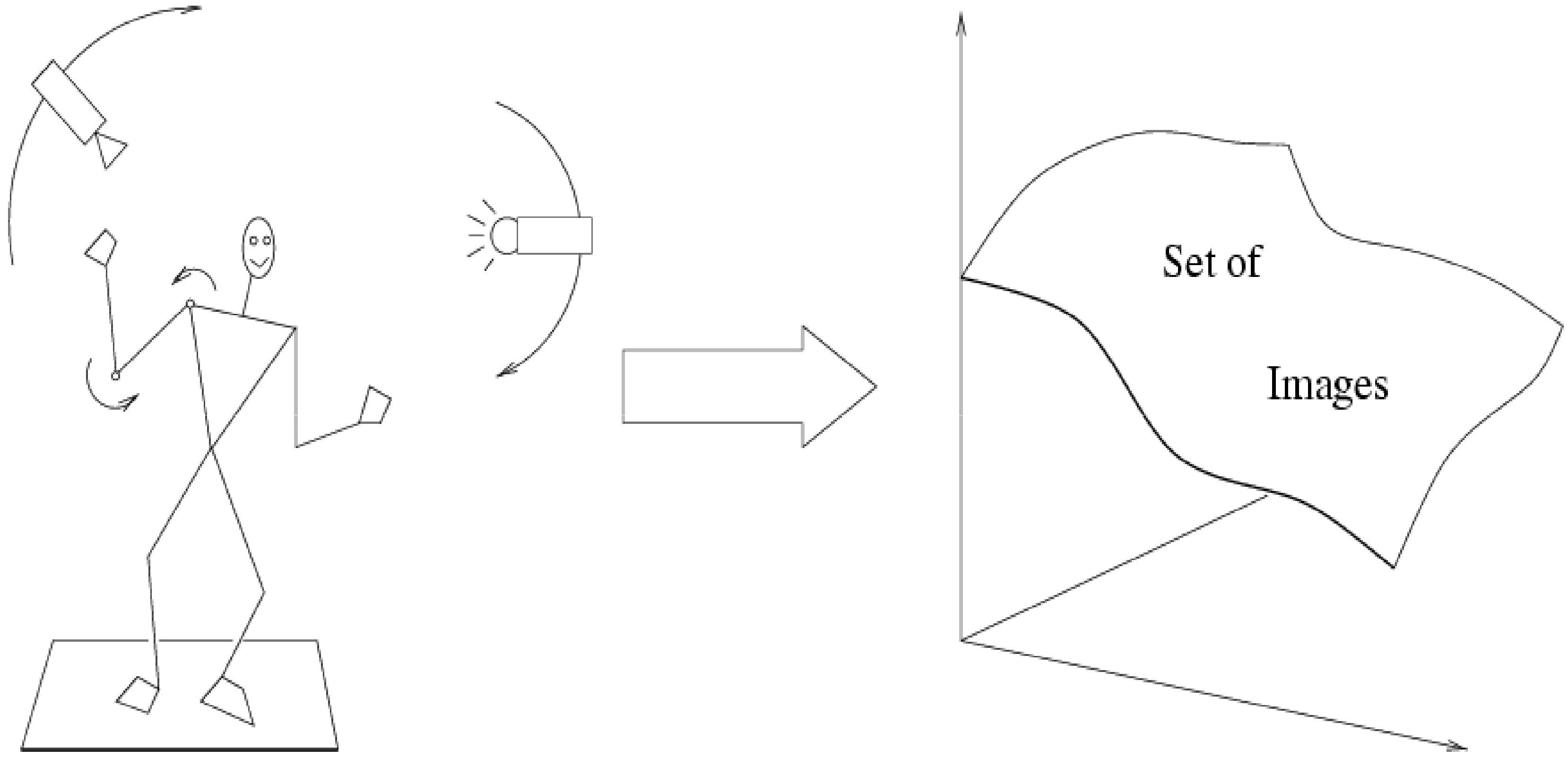
- Object localization: define the location and the category



Location

Category

Difficulties: within object variations



Variability: Camera position, Illumination,Internal parameters

⇒ Within-object variations

Difficulties: within-class variations

Category recognition

- Robust image description
  - Appropriate descriptors for categories

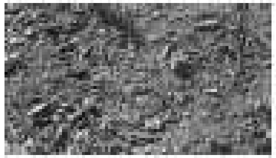
- Statistical modeling and machine learning for vision
  - Use and validation of appropriate techniques

Image classification

- Given

  Positive training images containing an object class

  Negative training images that don’t



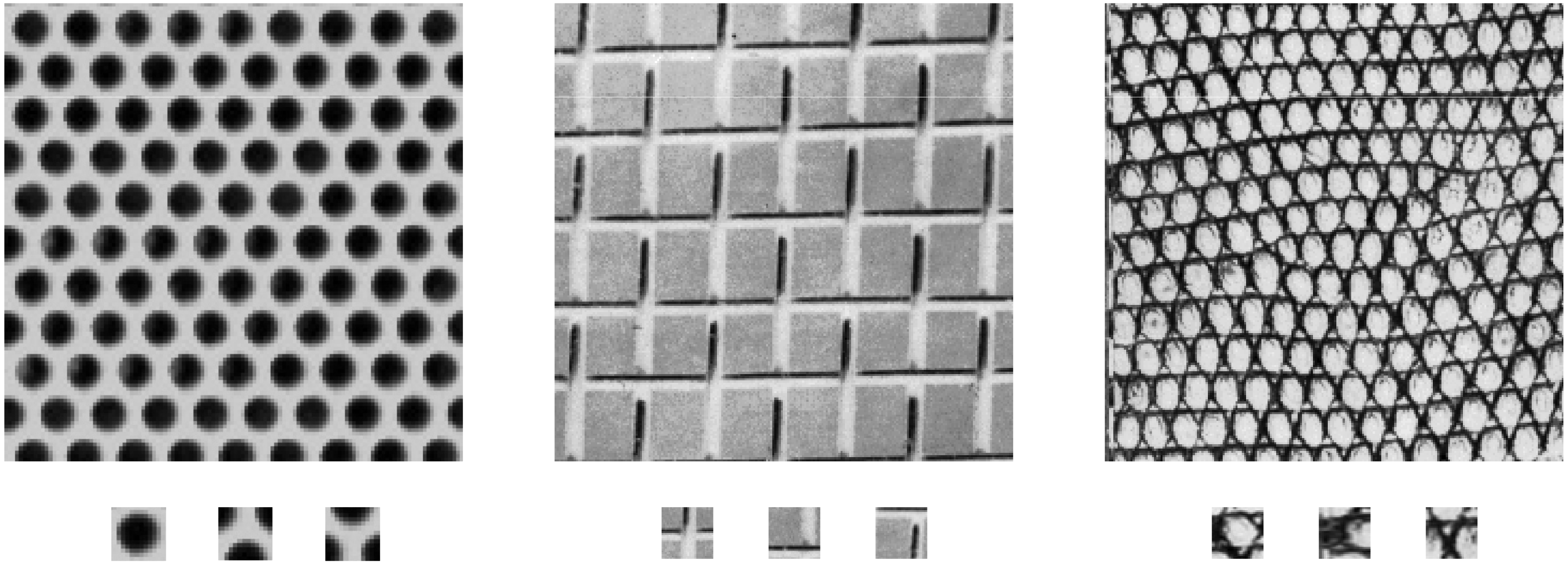
- Classify

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

?

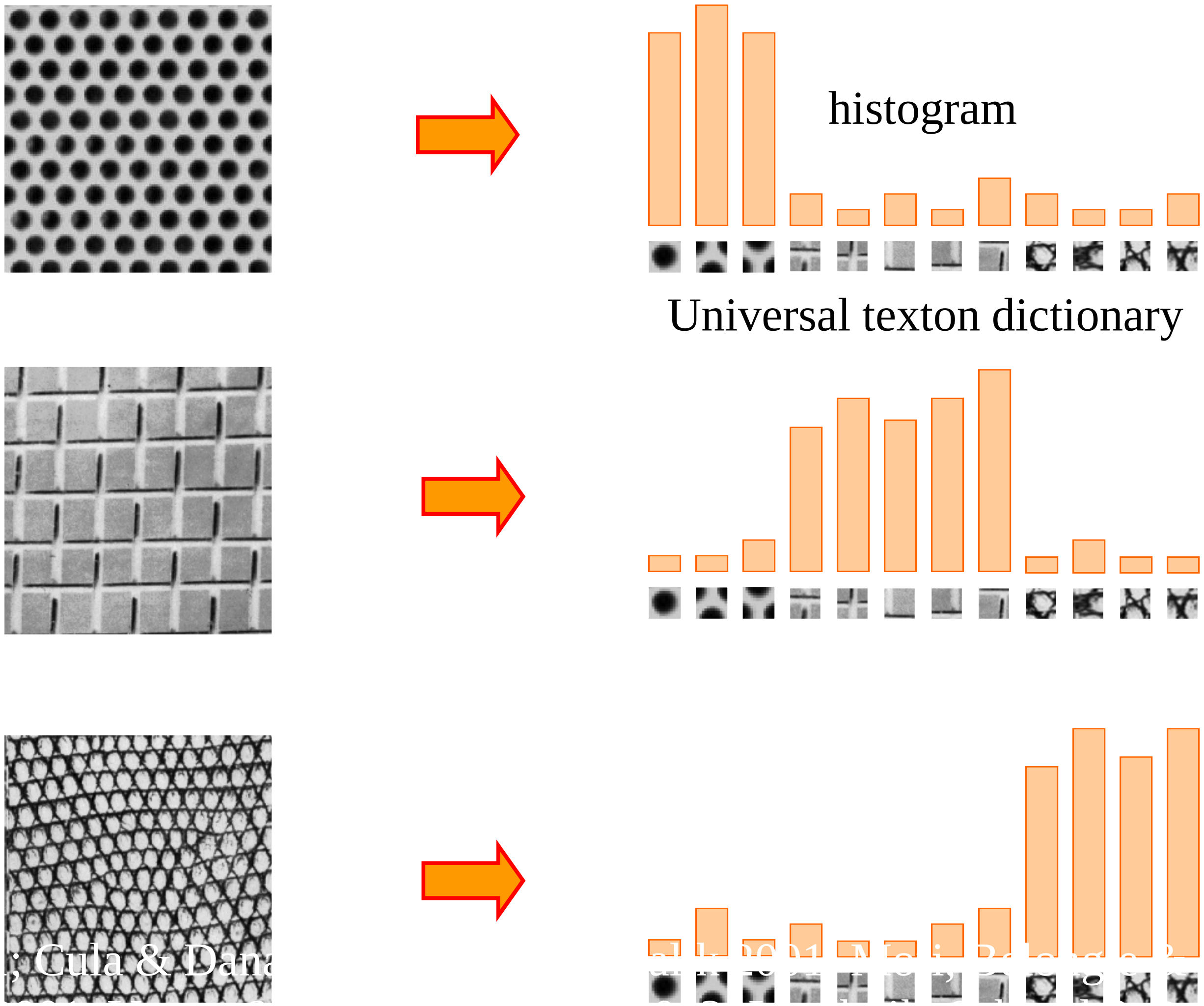
Bag-of-features for image classification

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



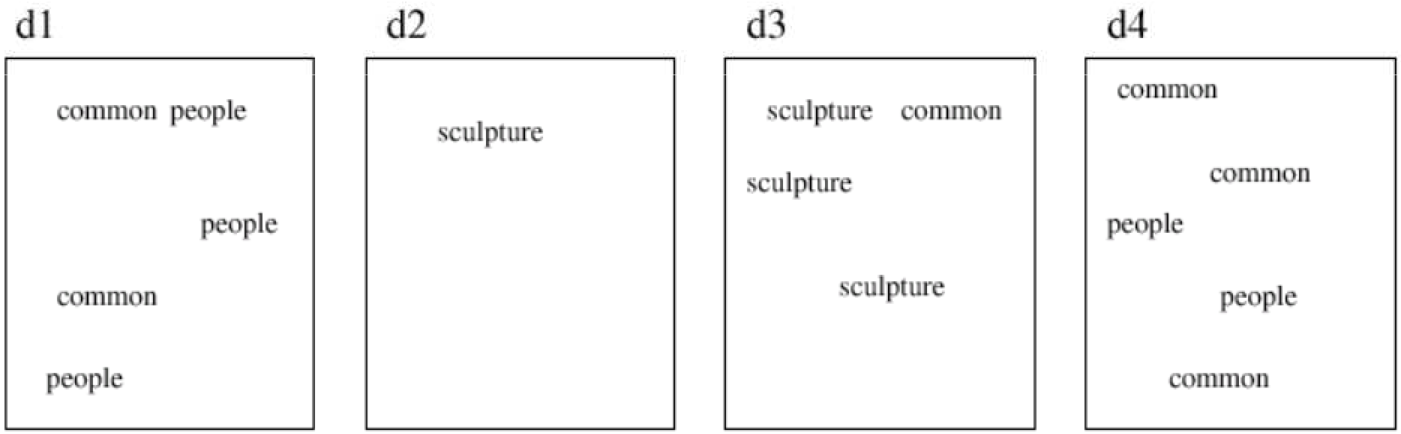
Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001 Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

Texture recognition



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

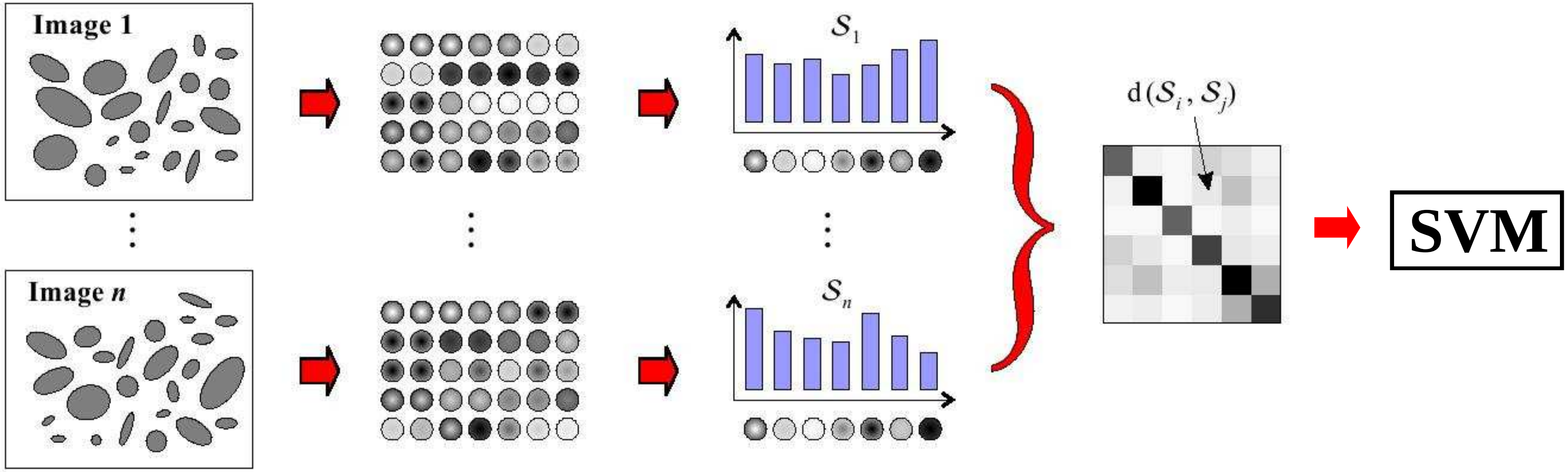
- Orderless document representation: frequencies of words from a dictionary
- Classification to determine document categories



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

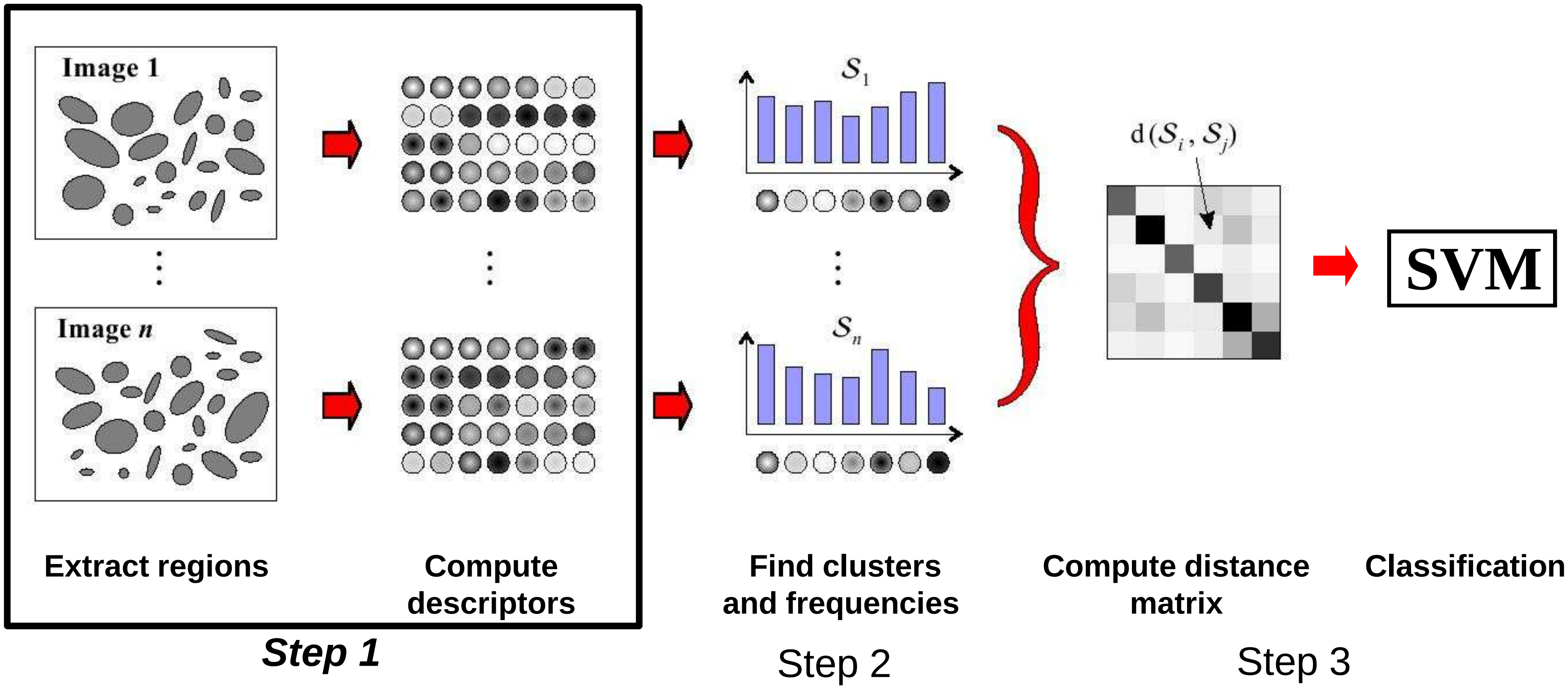
Find clusters and frequencies

Compute distance matrix

Classification

[Nowak,Jurie&Triggs,ECCV’06], [Zhang,Marszalek,Lazebnik&Schmid,IJCV’07]

Bag-of-features for image classification



Extract regions

Compute descriptors

Find clusters and frequencies

Compute distance matrix

Classification

Step 1

Step 2

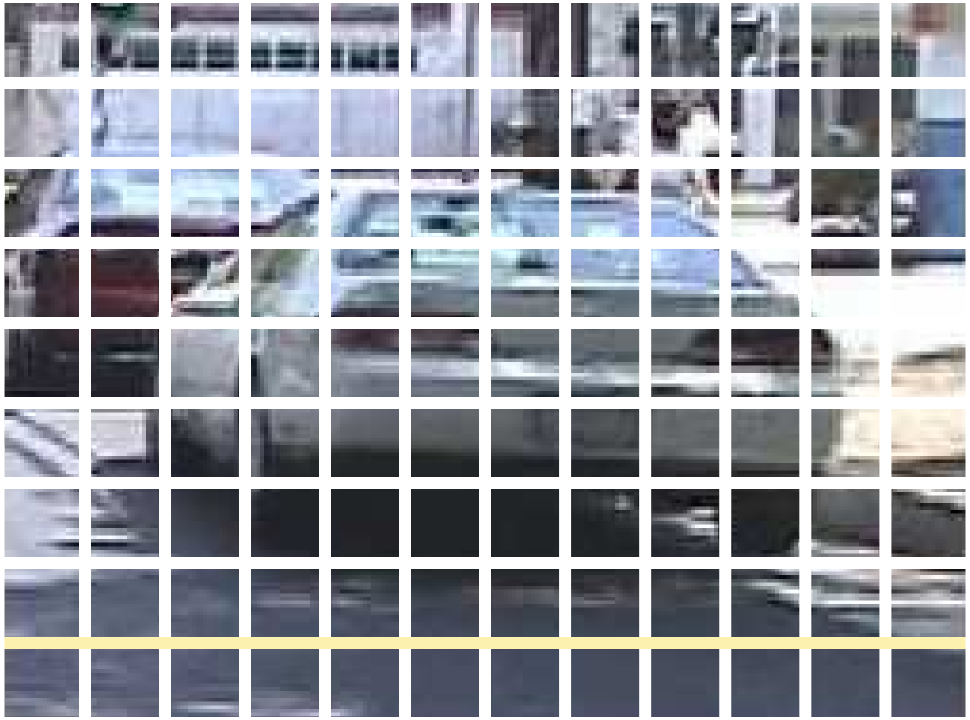
Step 3

[Nowak,Jurie&Triggs,ECCV'06], [Zhang,Marszalek,Lazebnik&Schmid,IJCV'07]

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

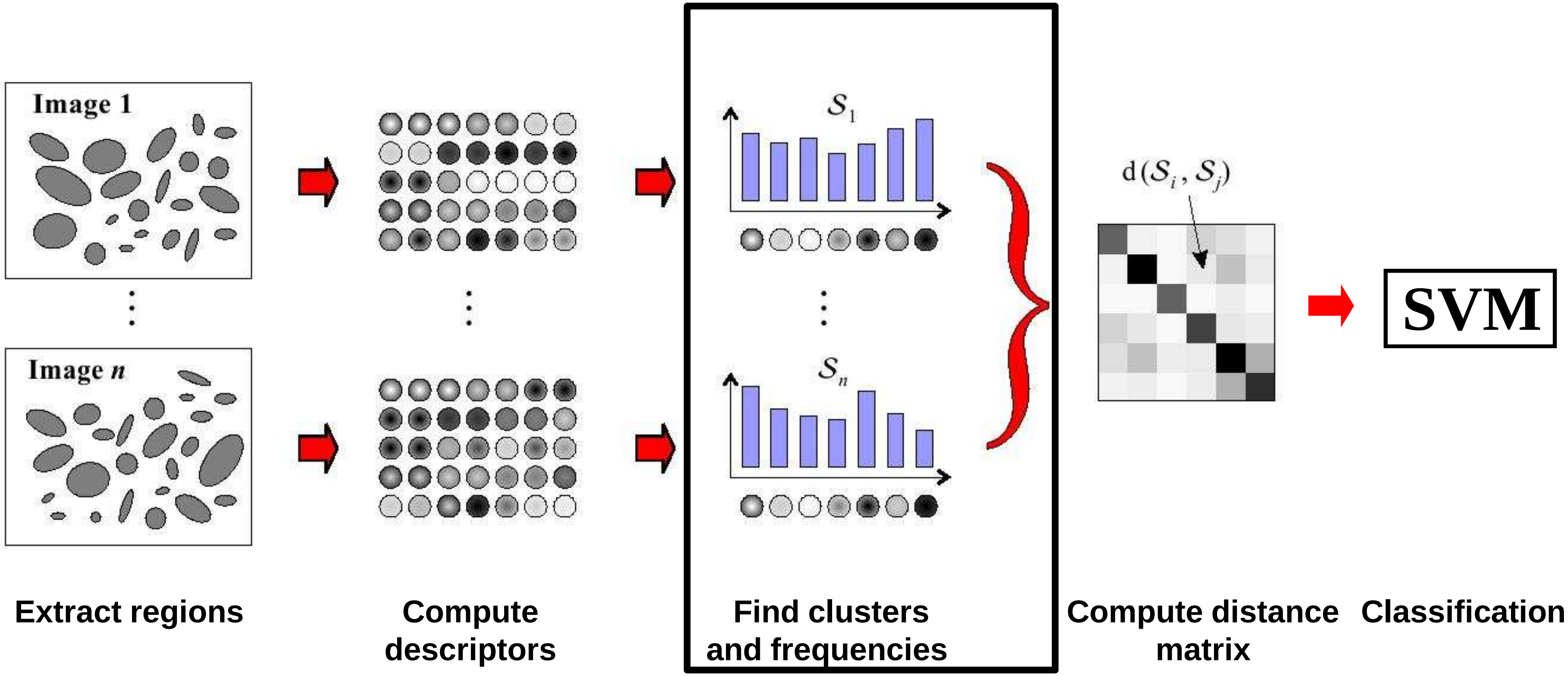


- Multi-scale dense grid: extraction of small overlapping patches at multiple scales

-Computation of the SIFT descriptor for each grid cells

-Exp.: Horizontal/vertical step size 6 pixel, scaling factor of 1.2 per level

Bag-of-features for image classification



Extract regions

Compute descriptors

Find clusters and frequencies

Compute distance matrix

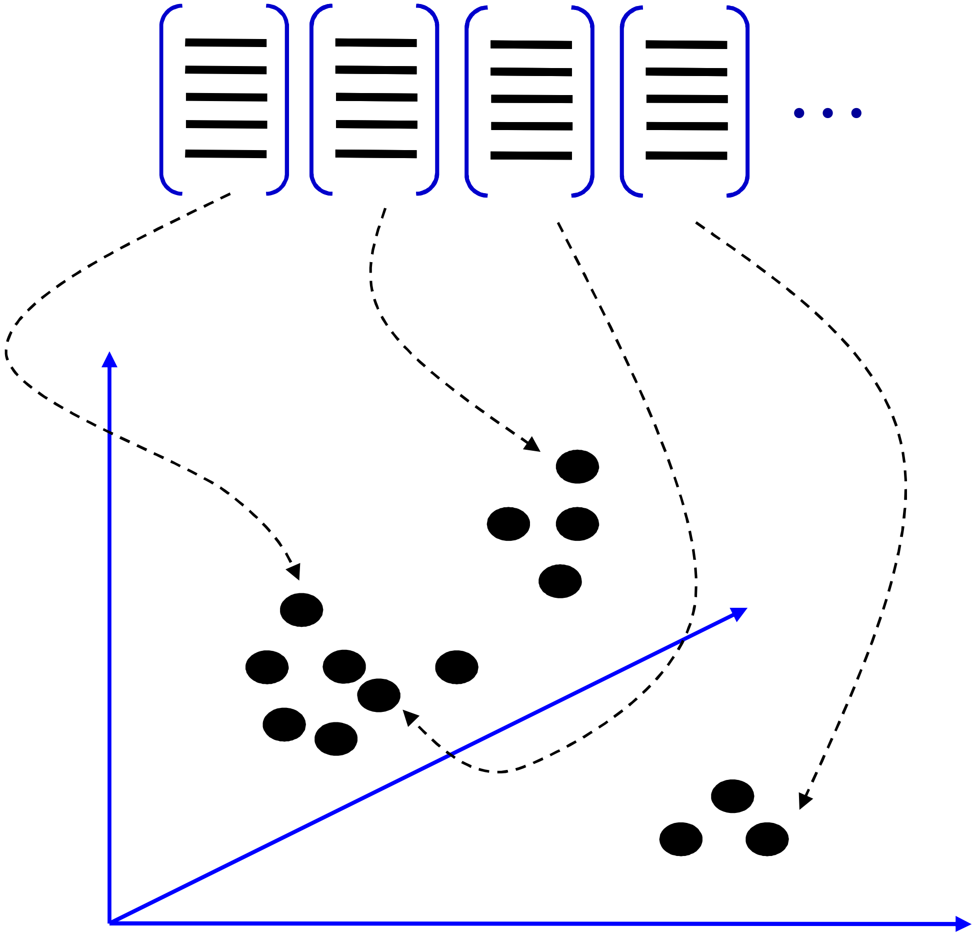
Classification

Step 1

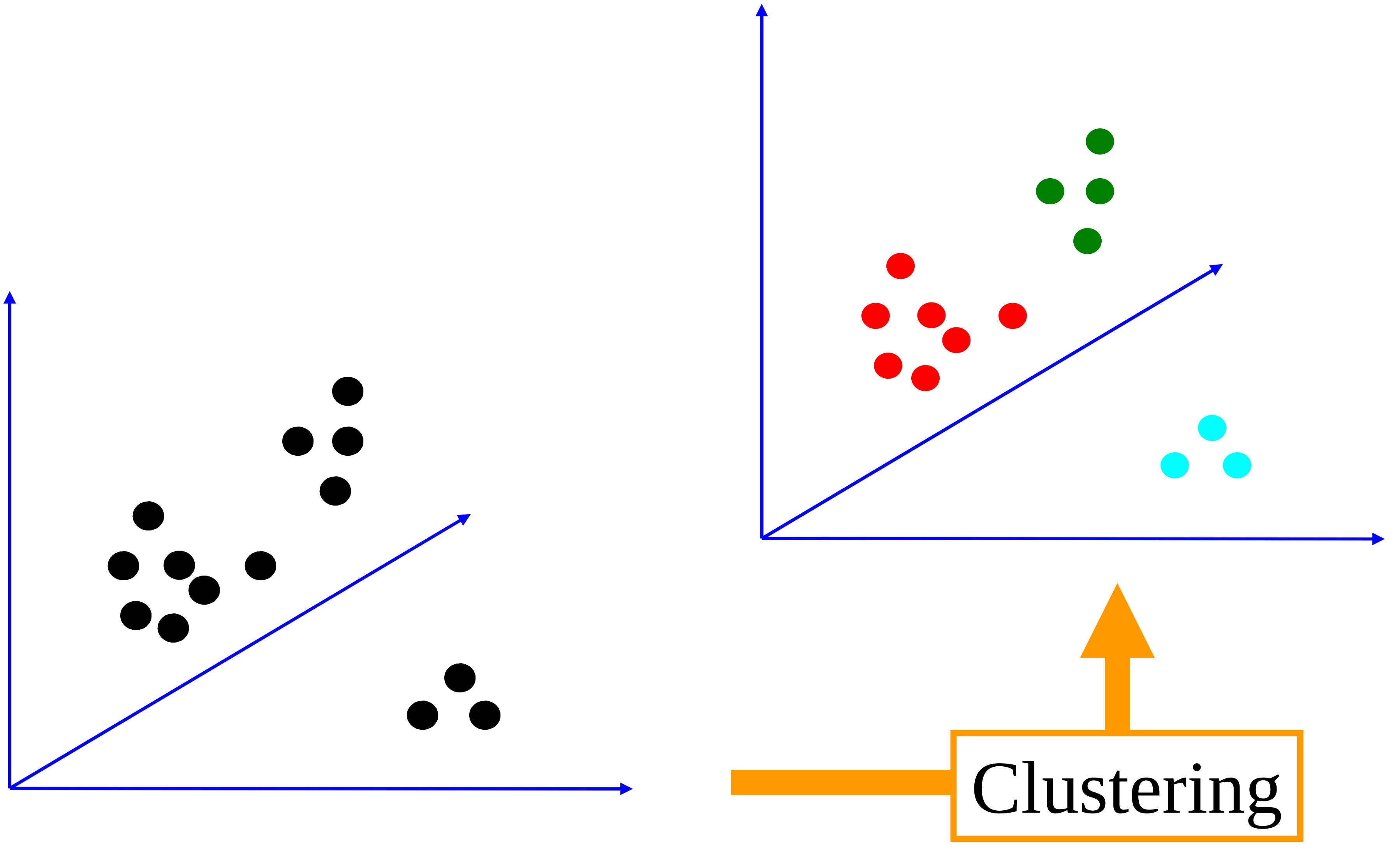
Step 2

Step 3

Step 2: Quantization

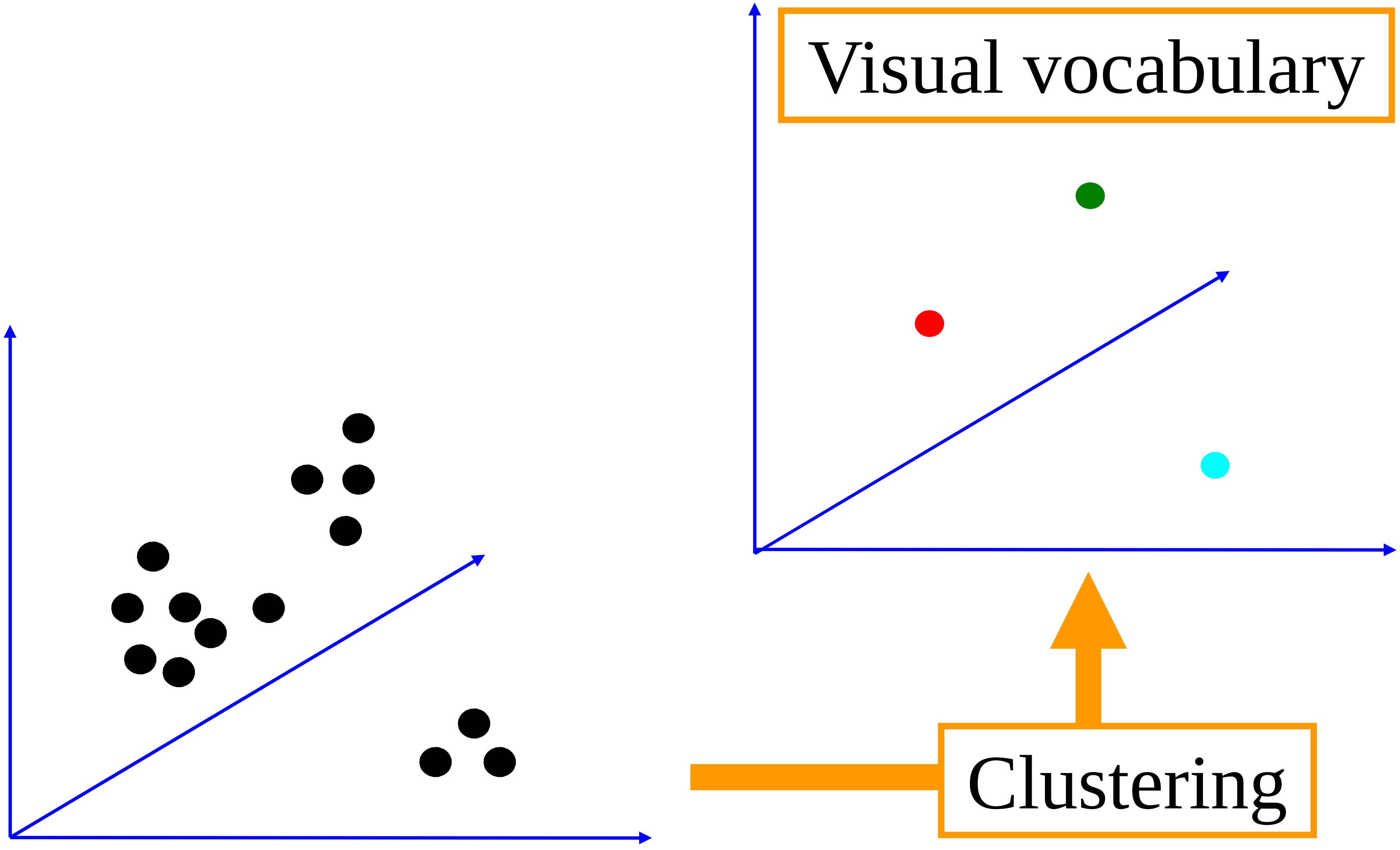


Step 2:Quantization

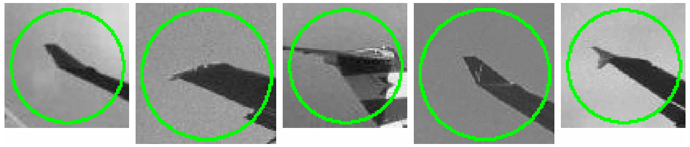
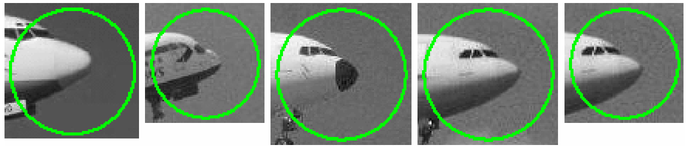
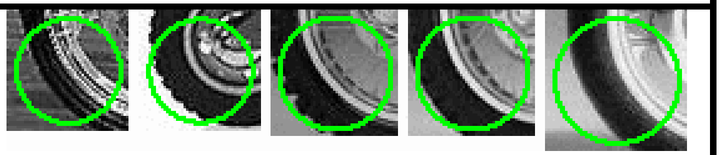
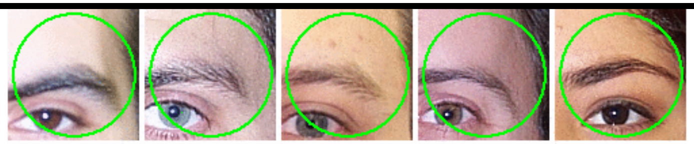
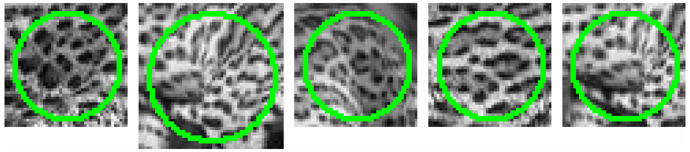
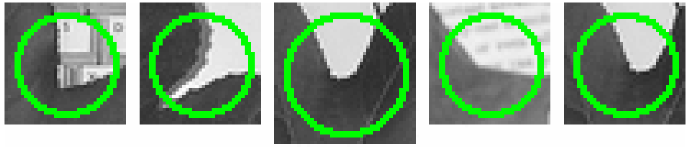
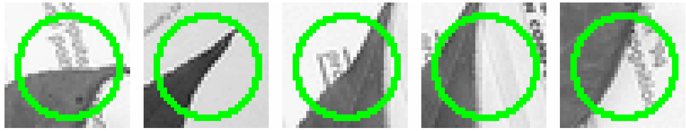
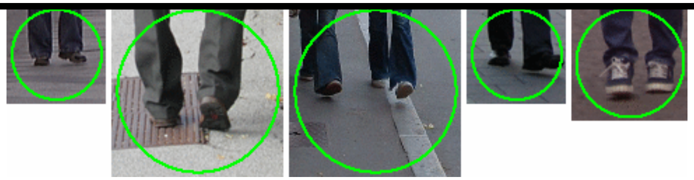
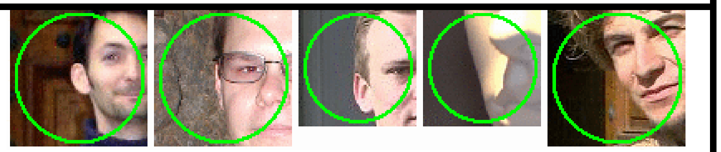
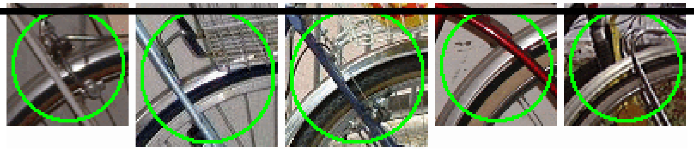


Clustering

Step 2: Quantization



Examples for visual words

Airplanes		
Motorbikes		
Faces		
Wild Cats		
Leaves		
People		
Bikes		

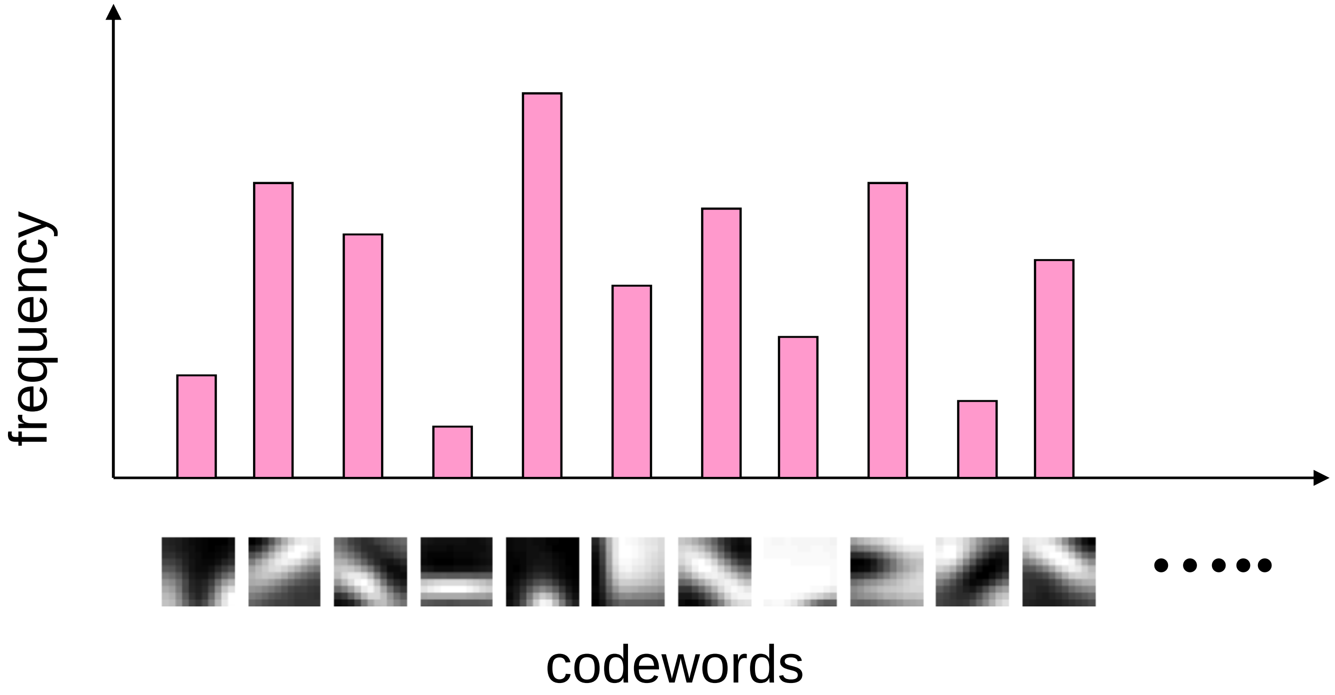
Step 2: Quantization

- 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

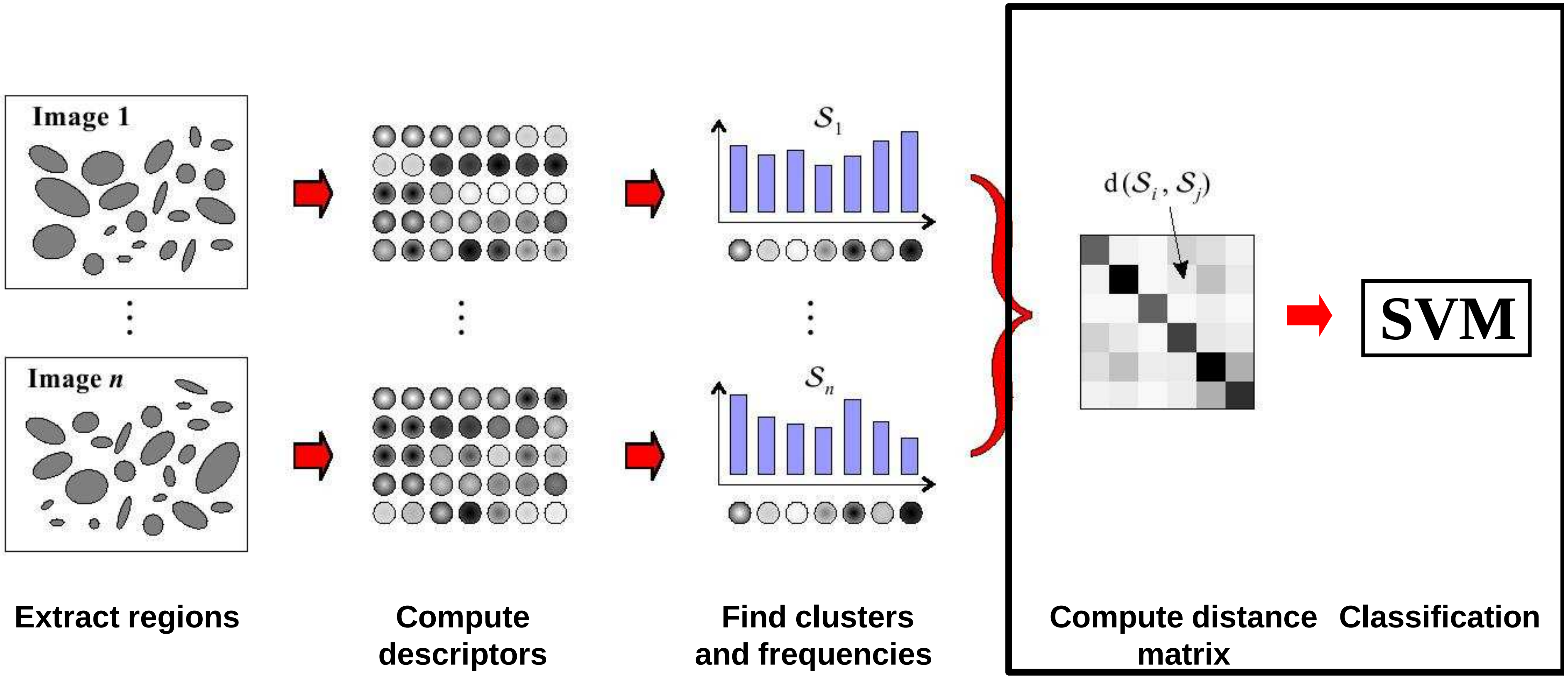
- K-means → hard assignment
  - Assign to the closest cluster center
  - Count number of descriptors assigned to a center
- Gaussian mixture model → soft assignment
  - Estimate distance to all centers
  - Sum over number of descriptors
- Represent image by a frequency histogram

Image representation



- Each image is represented by a vector, typically 1000-4000 dimension, normalization with L1 norm
- fine grained – represent model instances
- coarse grained – represent object categories

Bag-of-features for image classification



Extract regions

Compute descriptors

Step 1

Find clusters and frequencies

Step 2

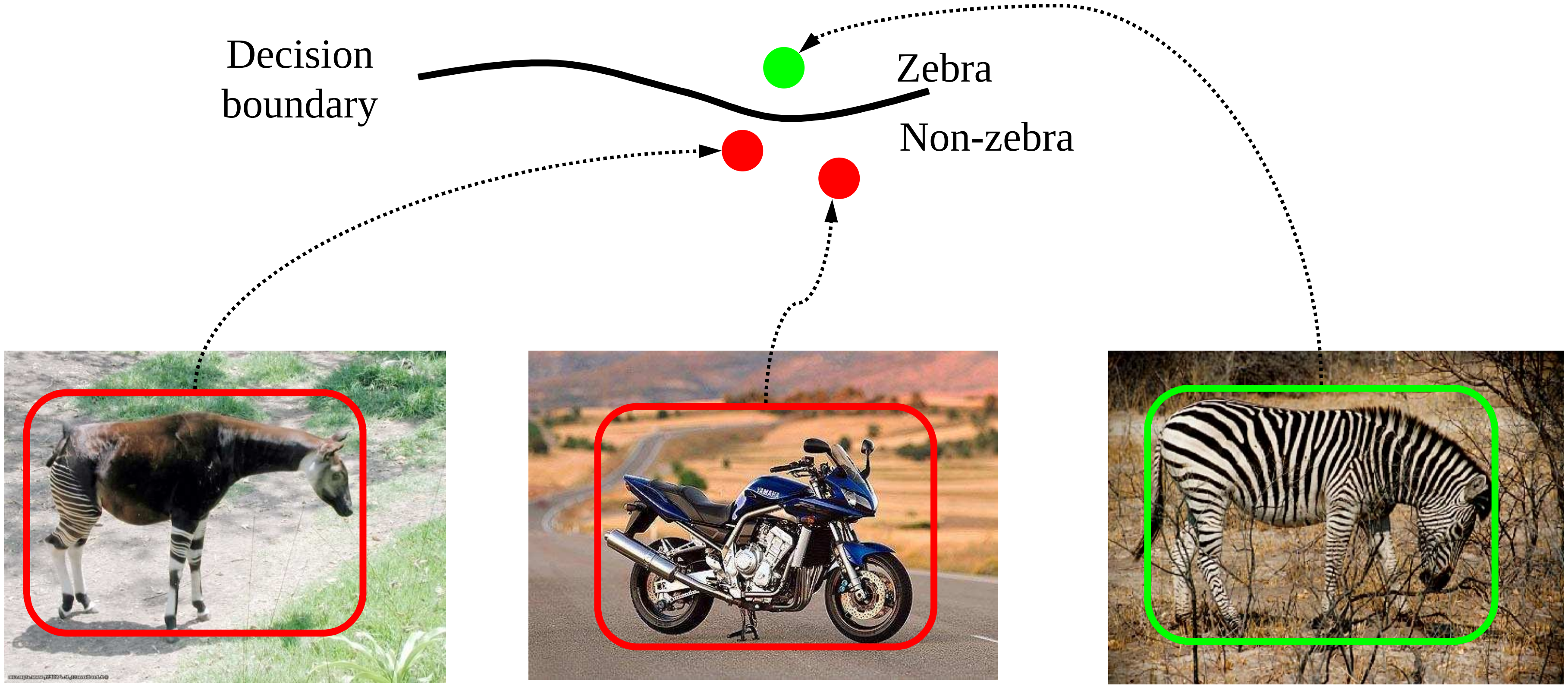
Compute distance matrix

Classification

Step 3

Step 3: Classification

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



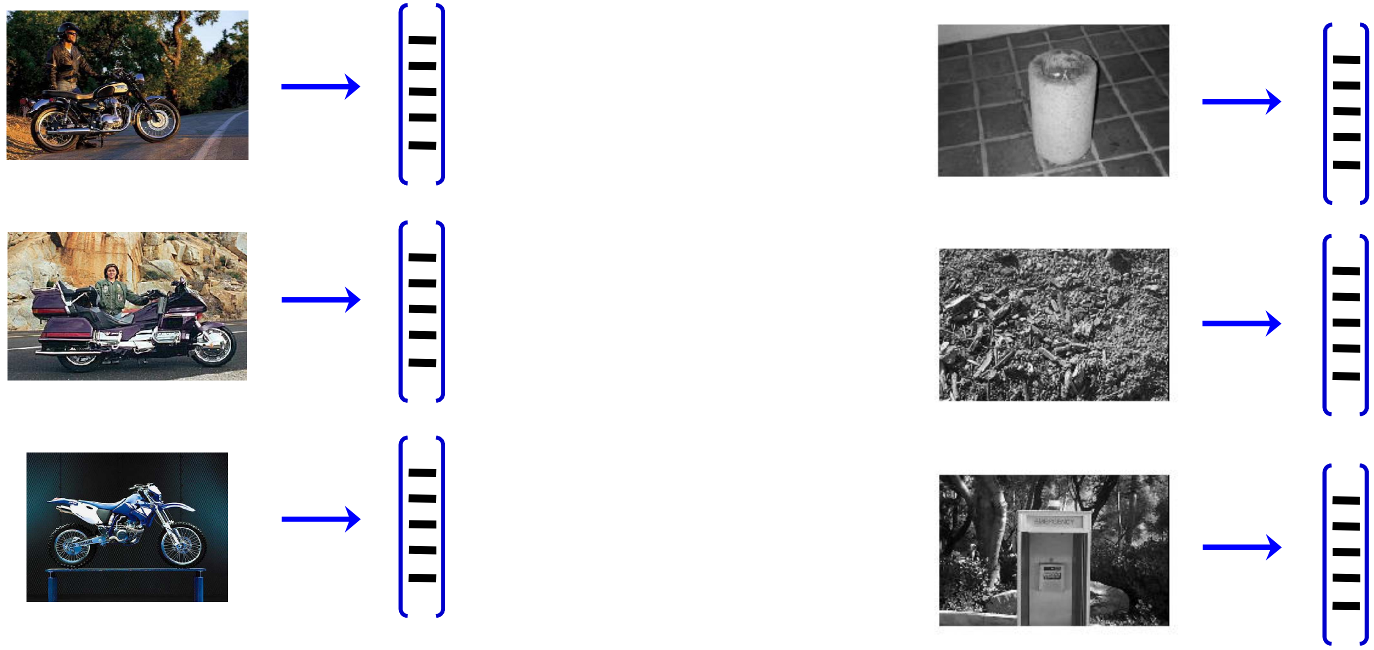
Decision boundary

Zebra

Non-zebra

Training data

Vectors are histograms, one from each training image



positive

negative

Train classifier,e.g.SVM

Classifiers

- K-nearest neighbor classifier

- Linear classifier
  - Support Vector Machine

- Non-linear classifier
  - Kernel trick
  - Explicit lifting

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

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

- 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.

Multi-class SVMs

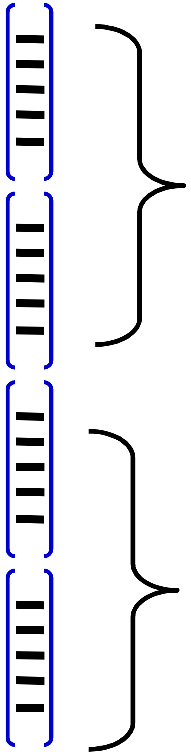
- Various direct formulations exist, but they are not widely used in practice. It is more common to obtain multi-class SVMs by combining two-class SVMs in various ways.

- 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

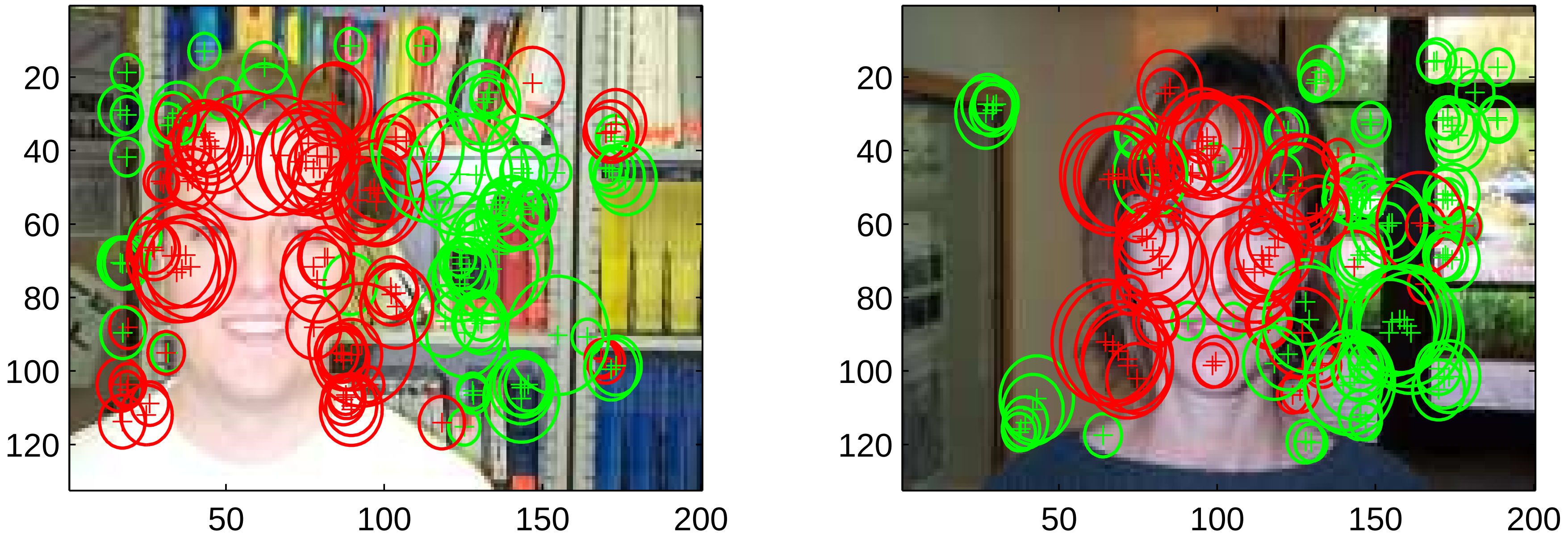


foreground words – high weight

background words – low weight

Illustration

Localization according to visual word probability



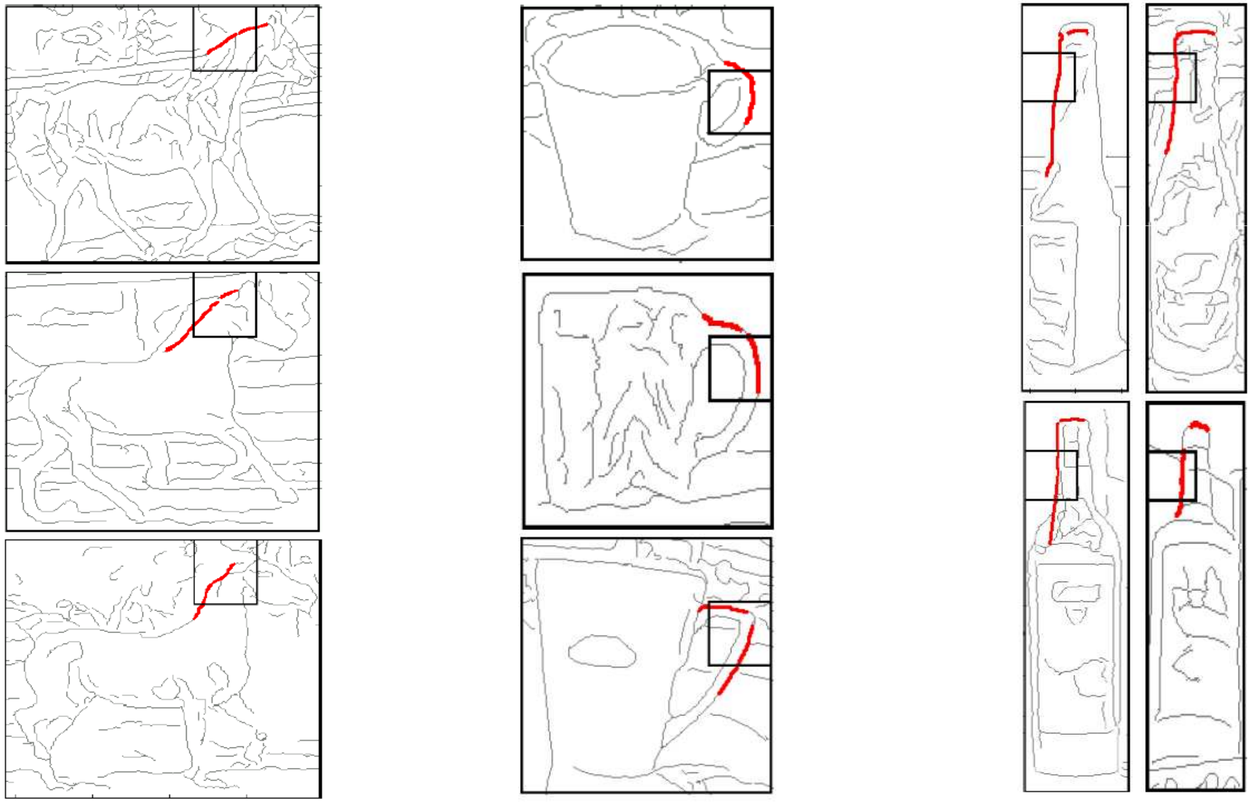
⭕ foreground word more probable

○ 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')

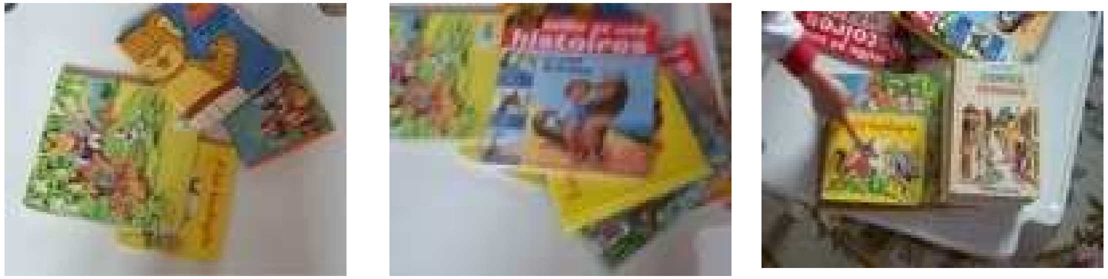


+ lie on object boundary (= local shape structures common to many training exemplars)

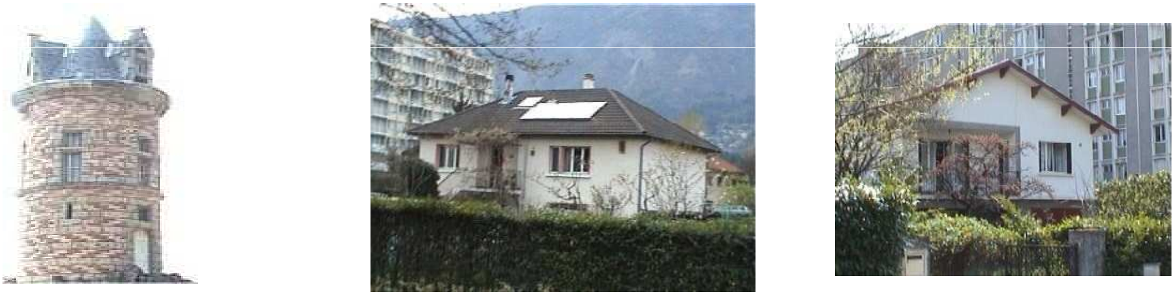
Bag-of-features for image classification

- Excellent results in the presence of background clutter

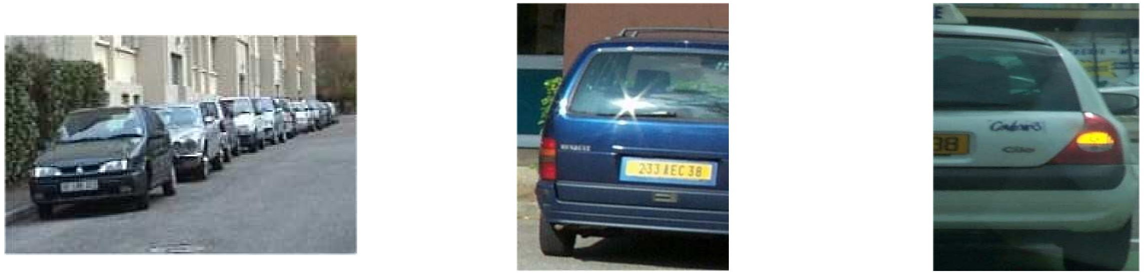
Examples for misclassified images



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

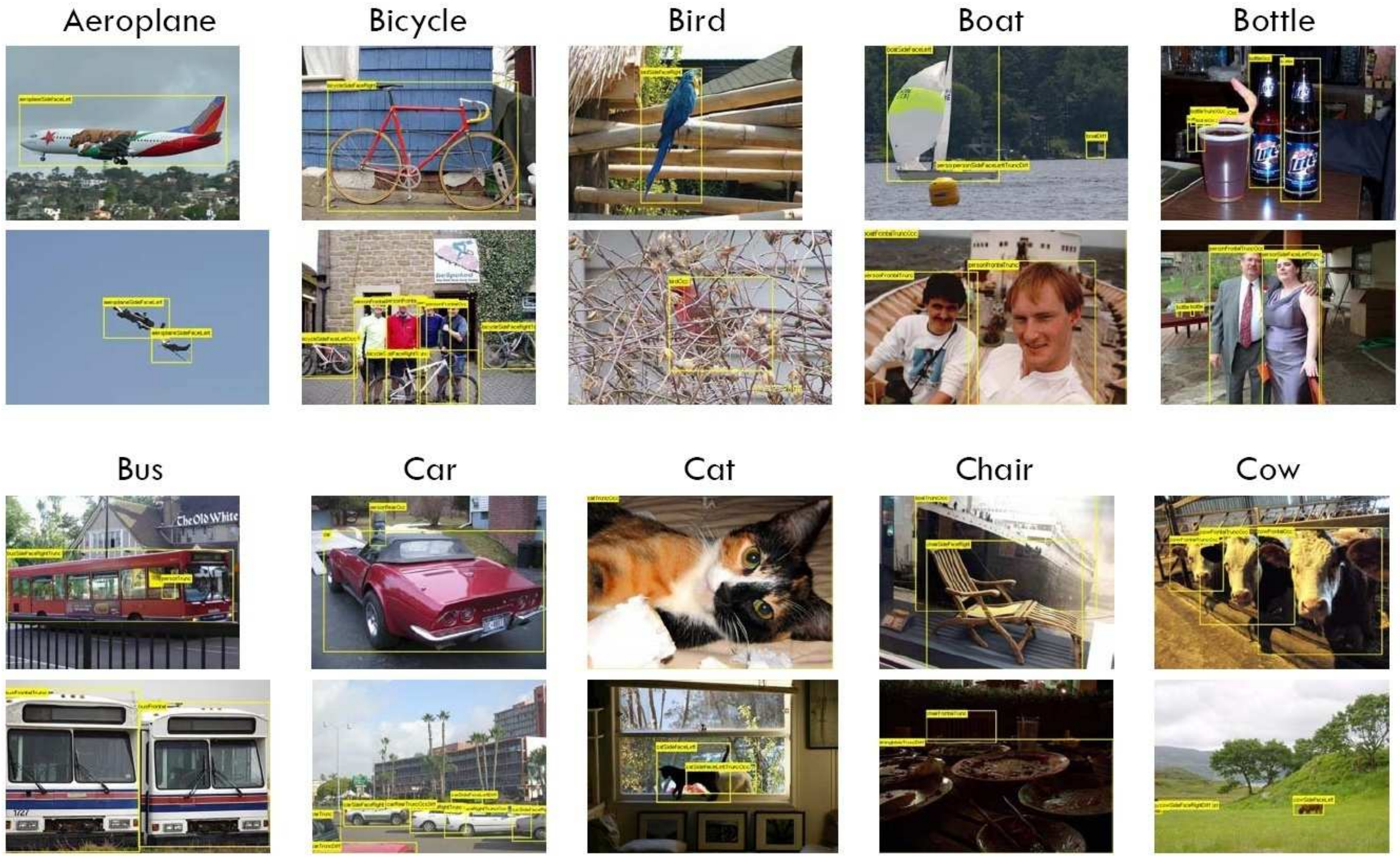
Evaluation of image classification

- PASCAL VOC [05-10] datasets

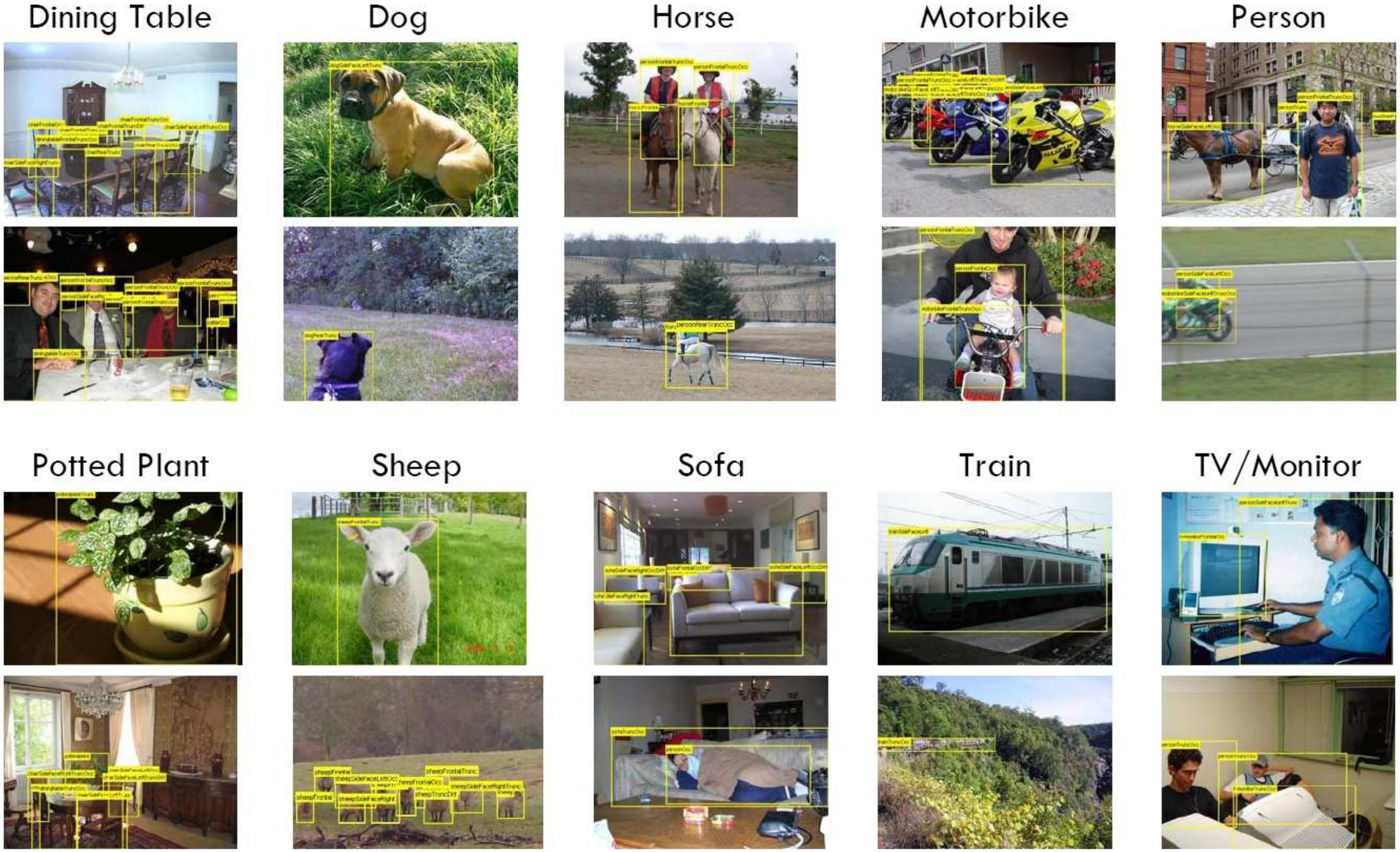
- PASCAL VOC 2007
  - Training *and* test dataset available
  - Used to report state-of-the-art results
  - Collected January 2007 from Flickr
  - 500 000 images downloaded and random subset selected
  - 20 classes
  - Class labels per image + bounding boxes
  - 5011 training images, 4952 test images

- Evaluation measure: average precision

PASCAL 2007 dataset

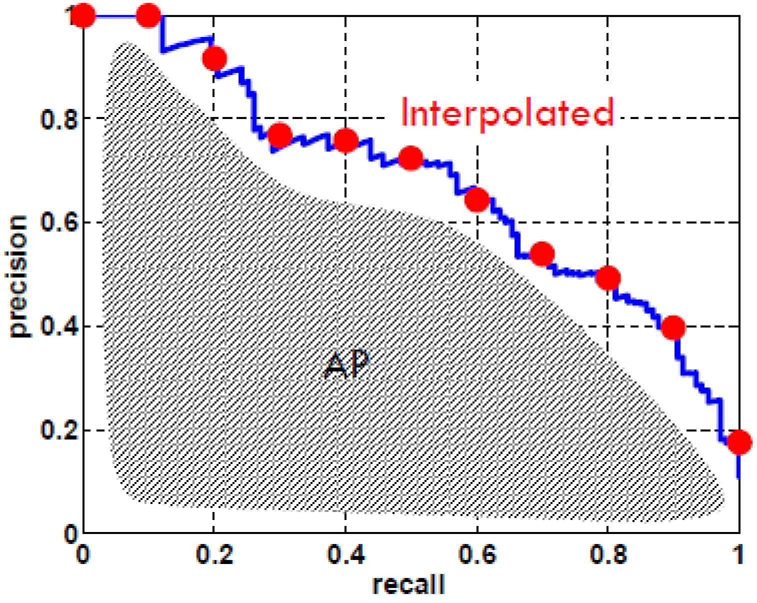


PASCAL 2007 dataset



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

Results for PASCAL 2007

- Winner of PASCAL 2007 [Marszalek et al.] : mAP 59.4
  - Combination of several different channels (dense + interest points, SIFT + color descriptors, spatial grids)
  - Non-linear SVM with Gaussian kernel

- Multiple kernel learning [Yang et al. 2009] : mAP 62.2
  - Combination of several features
  - Group-based MKL approach

- Combining object localization and classification [Harzallah et al.'09] : mAP 63.5
  - Use detection results to improve classification

Comparison interest point - dense

Image classification results on PASCAL'07 train/val set

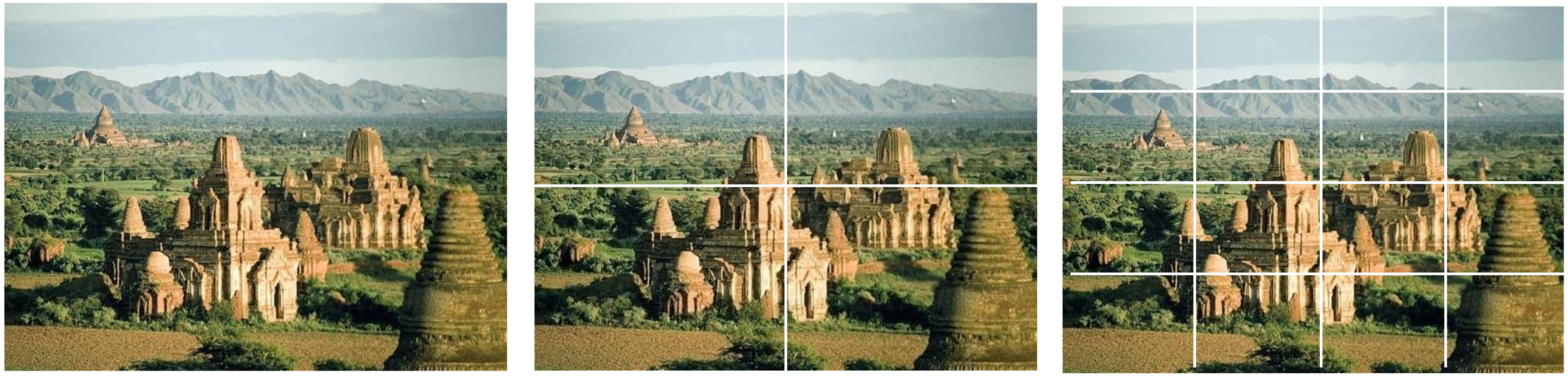
	AP
(SHarris + Lap) x SIFT	0.452
MSDense x SIFT	0.489
(SHarris + Lap + MSDense) x SIFT	0.515

Method: bag-of-features + SVM classifier

- Dense is on average a bit better
- IP and dense are complementary, combination improves results

Spatial pyramid matching

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



[Lazebnik, Schmid & Ponce, CVPR 2006]

Evaluation spatial pyramid

Image classification results on PASCAL’07 train/val set

(SH, Lap, MSD) x (SIFT,SIFTC) spatial layout	AP
1	0.53
2x2	0.52
3x1	0.52
1,2x2,3x1	0.54

Spatial layout not dominant for PASCAL’07 dataset

Combination improves average results, i.e., it is appropriate for some classes

Evaluation spatial pyramid

Image classification results on PASCAL’07 train/val set for individual categories

	1	3x1
Sheep	**0.339**	0.256
Bird	**0.539**	0.484
DiningTable	0.455	**0.502**
Train	0.724	**0.745**

Results are category dependent!

➔ Combination helps somewhat

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