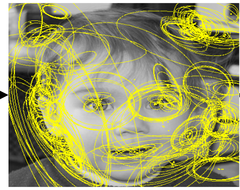
Aggregating local image descriptors for large-scale retrieval and classification

Cordelia Schmid
LEAR – INRIA Grenoble

Aggregating local descriptors

- Set of n local descriptors → 1 vector
- Popular approach: bag of features, often with SIFT features
- Recently improved aggregation schemes
  - Fisher vector [Perronnin & Dance ‘07]
  - VLAD descriptor [Jegou, Douze, Schmid, Perez ‘10]
  - Supervector [Zhou et al. ‘10]
  - Sparse coding [Wang et al. ’10, Boureau et al.’10]
- Use in very large-scale retrieval and classification

Towards large-scale image search



- Each image is represented by one vector
  - Bag-of-features, Fisher vector, GIST

- Vector compression to reduce storage requirement and search time

Aggregation of local descriptors

- Most popular approach: BoF representation [Sivic & Zisserman 03]
  - sparse vector
  - highly dimensional

→ significant dimensionality reduction introduces loss

- Vector of locally aggregated descriptors (VLAD) [Jegou et al. 10]
  - non sparse vector
  - fast to compute
  - excellent results with a small vector dimensionality

- Fisher vector [Perronnin & Dance 07]
  - probabilistic version of VLAD
  - initially used for image classification
  - comparable or improved performance over VLAD for image retrieval

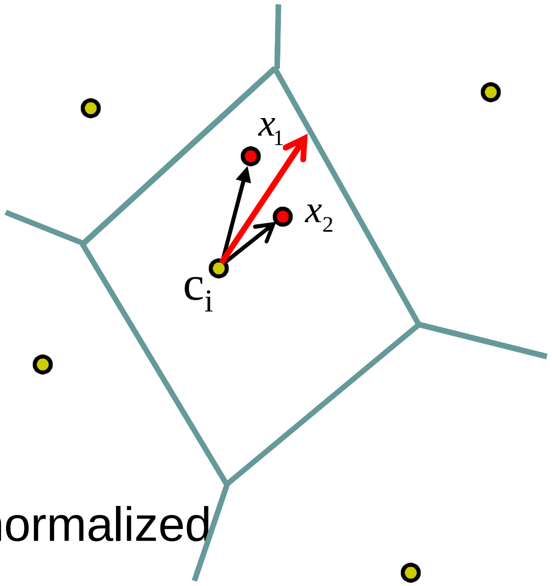
VLAD : vector of locally aggregated descriptors

- Learn a vector quantifier (*k*-means): $c_1,\dots,c_i,\dots c_k$, with c_i centroid of dim. *d*

- For a given image
  - assign each descriptor to closest center c_i
  - accumulate (sum) descriptors per cell

    $v_i := v_i + (x_j - c_i)$

    measure repartition of vectors within a cell



- VLAD of dimension $D = k \times d$

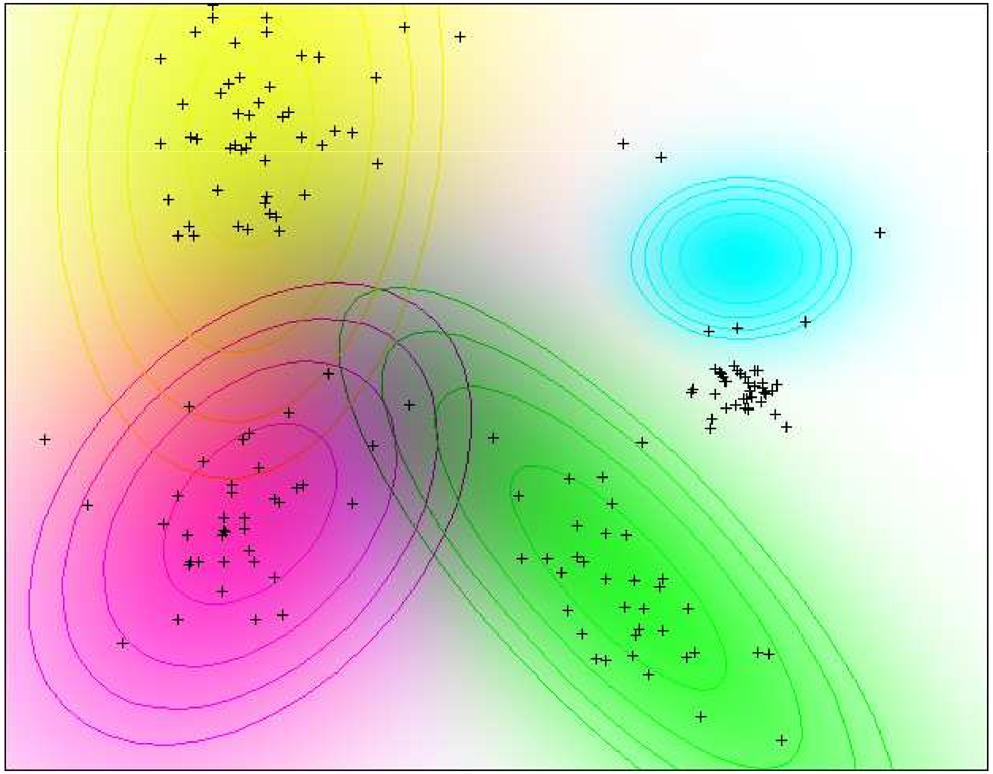
  (k typically between 16 and 256)

- The vector is square-root + L2-normalized

[Jegou, Douze, Schmid, Perez, CVPR'10]

Fisher vector

- Use a Gaussian Mixture Model as vocabulary
- Statistical measure of the descriptors of the image w.r.t the GMM
- Derivative of likelihood w.r.t. GMM parameters



GMM parameters:

w_i weight

μ_i mean

σ_i co-variance (diagonal)

Translated cluster →
large derivative on μ_i for this component

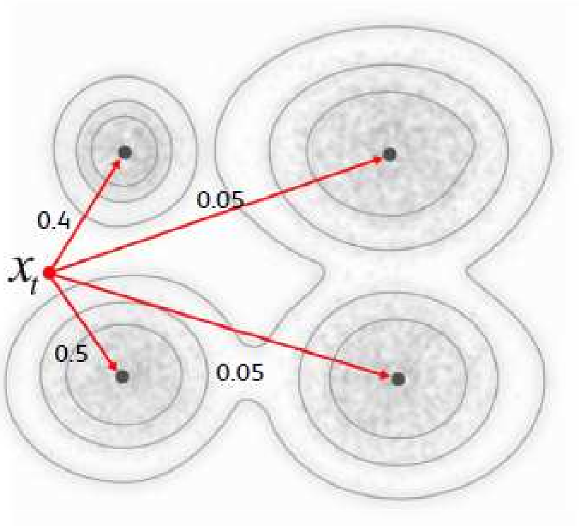
[Perronnin & Dance 07]

Fisher vector

FV formulas:

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

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



$\gamma_t(i)$ = soft-assignment of patch x_t to Gaussian i

Fisher Vector = concatenation of per-Gaussian gradient vectors

For image retrieval in our experiments:

- only deviation wrt mean, dim: K*D [K number of Gaussians, D dim of descriptor]
- variance does not improve for comparable vector length

VLAD/Fisher/BOF performance and dimensionality reduction

- We compare Fisher, VLAD and BoF on INRIA Holidays Dataset (mAP %)

- Holidays Dataset
  - 500 query images + 991 annotated true positives
  - most images are holiday photos of friends and family
  - 1 million & 10 million distractor images from Flickr
  - Vocabulary construction on a different Flickr set
  - Evaluation metric: mean average precision (in [0,1], bigger = better)

Query Database images

Query Database images

VLAD/Fisher/BOF performance and dimensionality reduction

- We compare Fisher, VLAD and BoF on INRIA Holidays Dataset (mAP %)
- Dimension is reduced to D' dimensions with PCA

Descriptor	K	D	Holidays (mAP)					
			$D'=D$	$\rightarrow D'$=2048	$\rightarrow D'$=512	$\rightarrow D'$=128	$\rightarrow D'$=64	$\rightarrow D'$=32
BOW	1 000	1 000	40.1		43.5	44.4	43.4	40.8
	20 000	20 000	43.7	41.8	44.9	45.2	44.4	41.8
Fisher (μ)	16	1 024	54.0		54.6	52.3	49.9	46.6
	64	4 096	59.5	60.7	61.0	56.5	52.0	48.0
	256	16 384	62.5	62.6	57.0	53.8	50.6	48.6
VLAD	16	1 024	52.0		52.7	52.6	50.5	47.7
	64	4 096	55.6	57.6	59.8	55.7	52.3	48.4
	256	16 384	58.7	62.1	56.7	54.2	51.3	48.1
GIST		960	36.5					

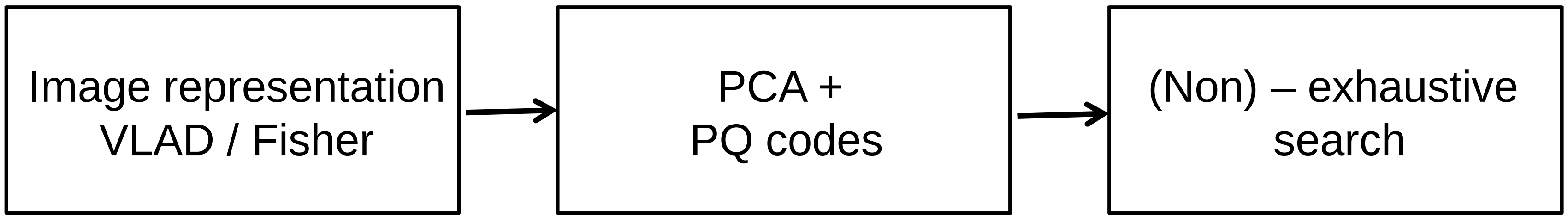
- Observations:
  - Fisher, VLAD better than BoF for a given descriptor size
  - Choose a small D if output dimension D' is small
  - Performance of GIST not competitive

[Jegou, Perronnin, Douze, Sanchez, Perez, Schmid, PAMI'12]

Compact image representation

- Aim: improving the tradeoff between
  - search speed
  - memory usage
  - search quality

- Approach: joint optimization of three stages
  - local descriptor aggregation
  - dimension reduction
  - indexing algorithm

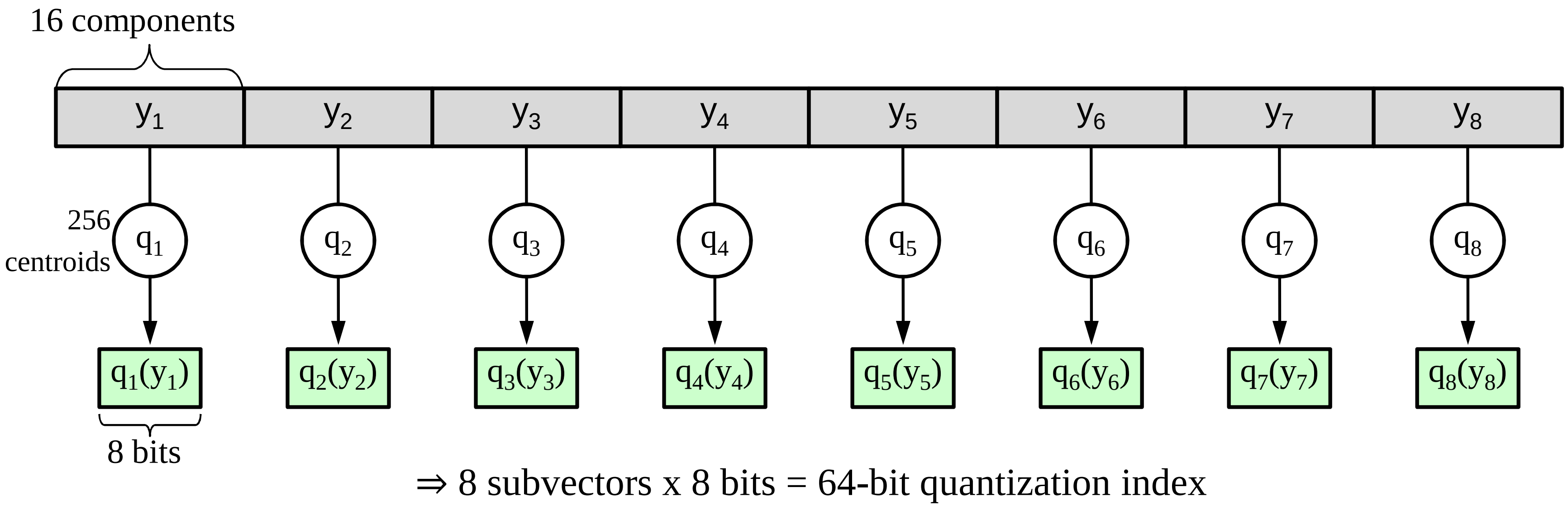


Product quantization for nearest neighbor search

- Vector split into *m* subvectors: $y \rightarrow [y_1 | \ldots | y_m]$

- Subvectors are quantized separately by quantizers $q(y) = [q_1(y_1) | \ldots | q_m(y_m)]$
  where each q_i is learned by *k*-means with a limited number of centroids

- Example: y = 128-dim vector split in 8 subvectors of dimension 16
  - each subvector is quantized with 256 centroids -> 8 bit
  - very large codebook 256^8 ~ 1.8x10^19

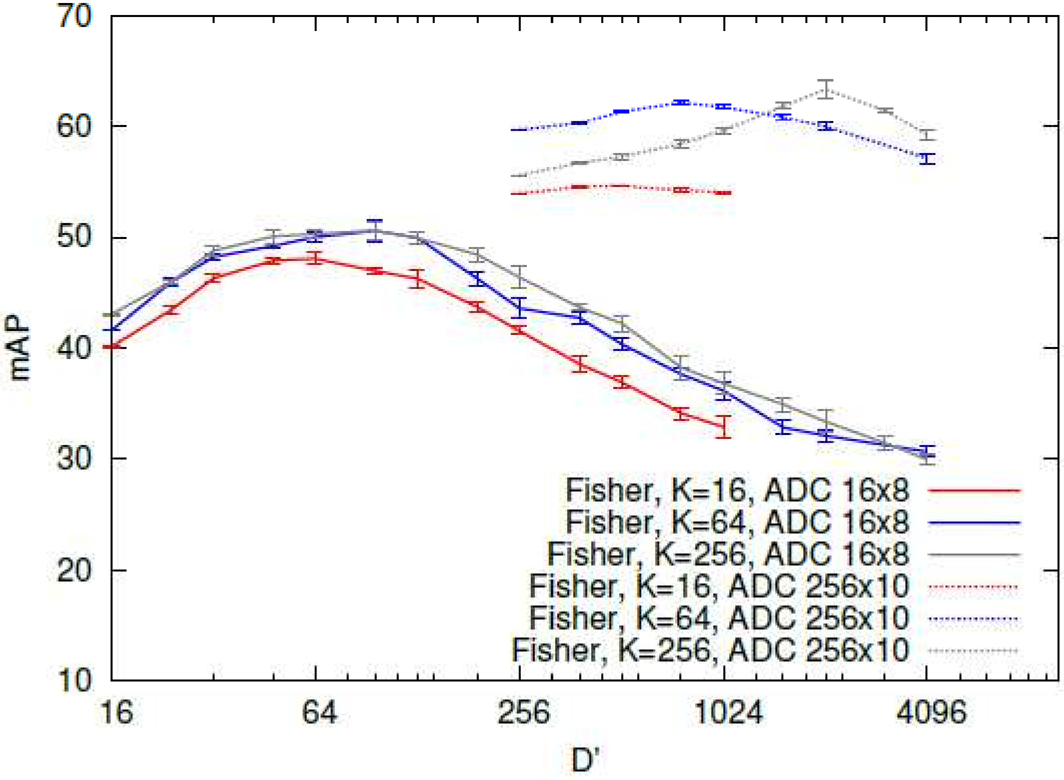


[Jegou, Douze, Schmid, PAMI'11]

Optimizing the dimension reduction and quantization together

- Fisher vectors undergoes two approximations
  - mean square error from PCA projection
  - mean square error from quantization

- Given k and bytes/image, choose D' minimizing their sum



Results on Holidays dataset:

- there exists an optimal D'
- 16 byte best results for k=64
- 320 byte best results for k=256

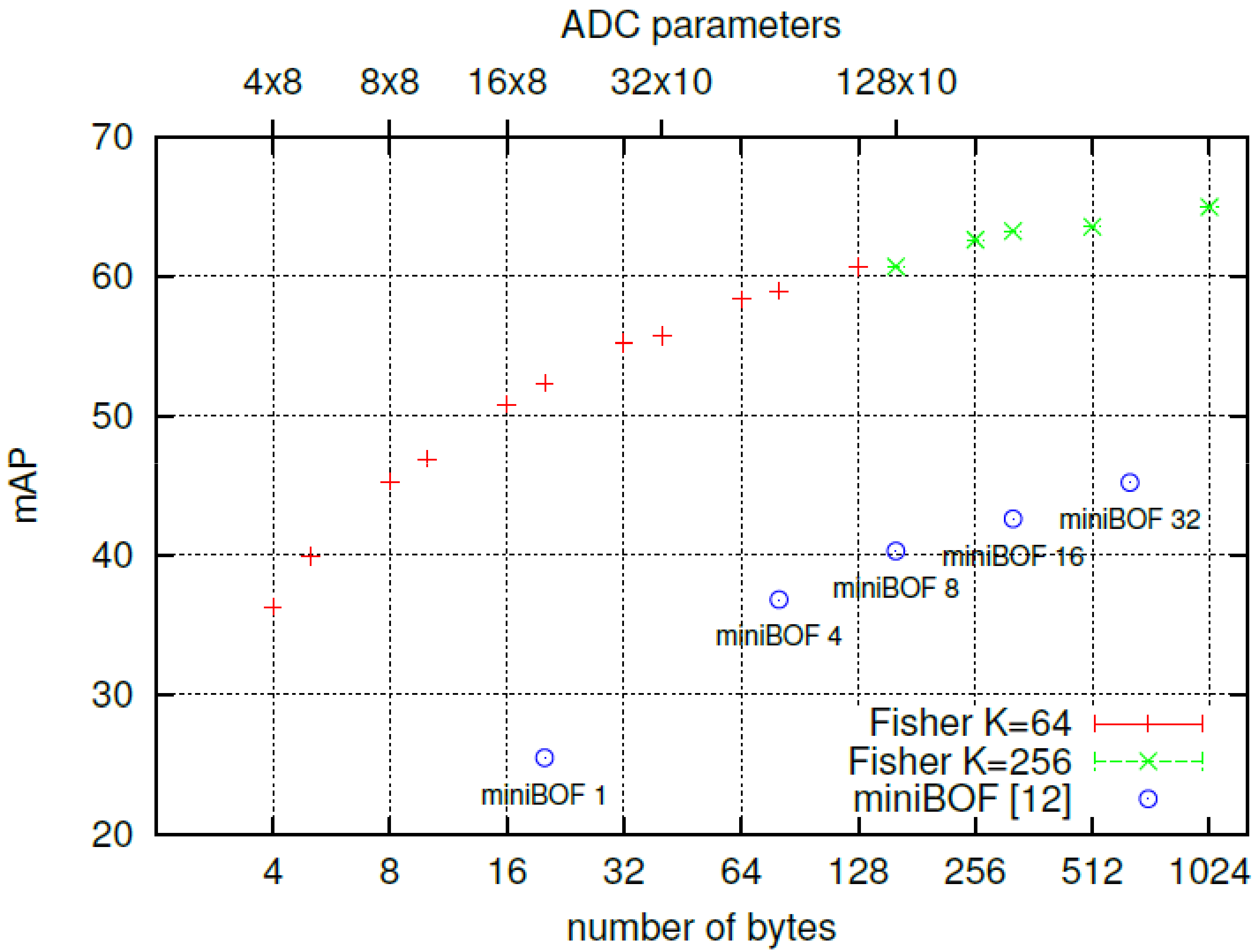
Joint optimization of Fisher and dimension reduction-indexing

- For Fisher
  - The larger *k*, the better the raw search performance
  - But large *k* produce large vectors, that are harder to index

- Optimization of the vocabulary size
  - Fixed output size (in bytes)
  - *D'* computed from *k* via the joint optimization of reduction/indexing
  - Only *k* has to be set

→ end-to-end parameter optimization

Results on the Holidays dataset with various quantization parameters



Comparison to the state of the art

Datasets:

- INRIA Holidays dataset, score: mAP (%)

- University of Kentucky benchmark (UKB)
  - 10200 images, 4 images per objects
  - score: number of relevant images retrieved in the first 4 positions, max 4

Comparison to the state of the art

Method	bytes	UKB	Holidays
BOW, K=20,000	10 364	2.87	43.7
BOW, K=200,000	12 886	2.81	54.0
miniBOF [12]	20	2.07	25.5
	80	2.72	40.3
	160	2.83	42.6
FV K=64, spectral hashing 128 bits	16	2.57	39.4
VLAD, K=16, ADC 16×8 [23]	16	2.88	46.0
VLAD, K=64, ADC 32×10 [23]	40	3.10	49.5
FV K=8, binarized [22]	65	2.79	46.0
FV K=64, binarized [22]	520	3.21	57.4
FV K=64, ADC 16×8 (D'=96)	**16**	**3.10**	**50.6**
FV K=256, ADC 256×10 (D'=2048)	**320**	**3.47**	**63.4**

[12] H. Jégou, M. Douze, and C. Schmid, "Packing bag-of-features," in *ICCV*, September 2009.

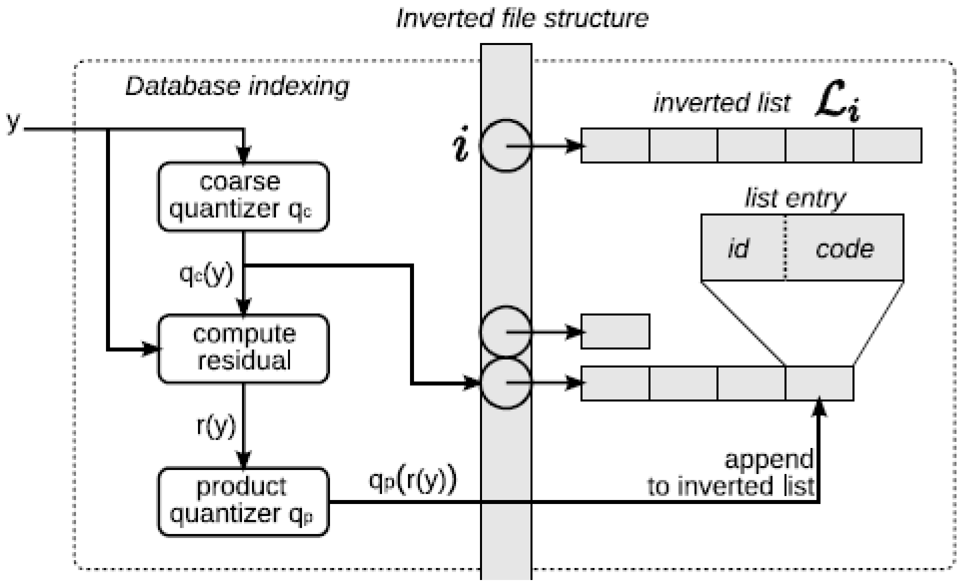
[22] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier, "Large-scale image retrieval with compressed Fisher vectors," in *CVPR*, June 2010.

[23] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *CVPR*, June 2010.

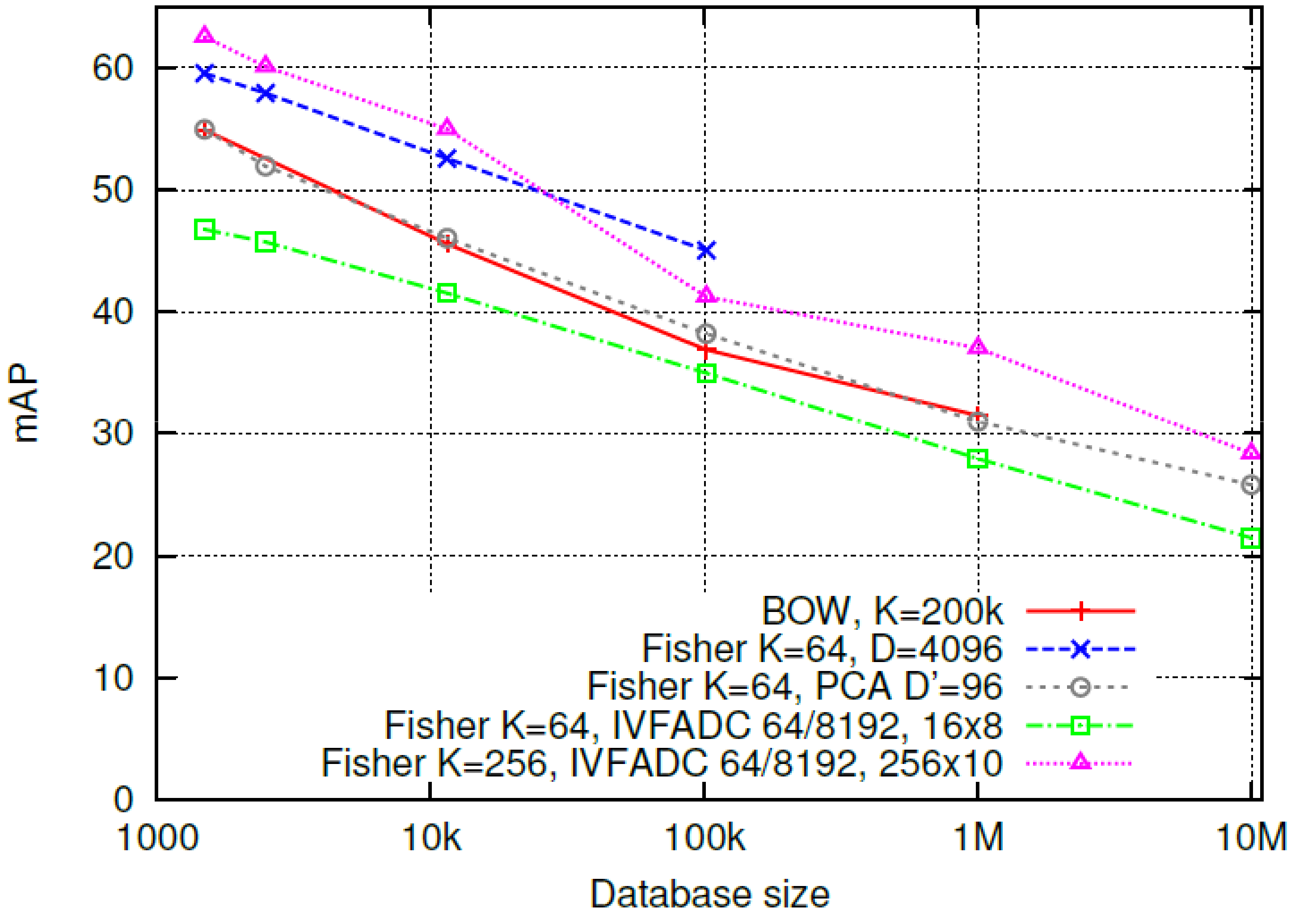
Large scale experiments (10 million images)

- With the product quantizer
  - Exhaustive search with ADC: 0.29s
  - Non-exhaustive search with IVFADC: 0.014s

IVFADC -- Combination with an inverted file



Large scale experiments (10 million images)



Timings

IVFADC: 0.02s

Conclusion

- Competitive search accuracy with a few dozen bytes per indexed image

- Tested on 220 million video frames
  - extrapolation for 1 billion images: 20GB RAM, query < 1s on 8 cores

- Code on-line available Software for Fisher computation and PQ-codes
  - http://lear.inrialpes.fr/software

Image classification

- Image classification: assigning a class label to the image

→

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

Image classification

- Image classification: assigning a class label to the image

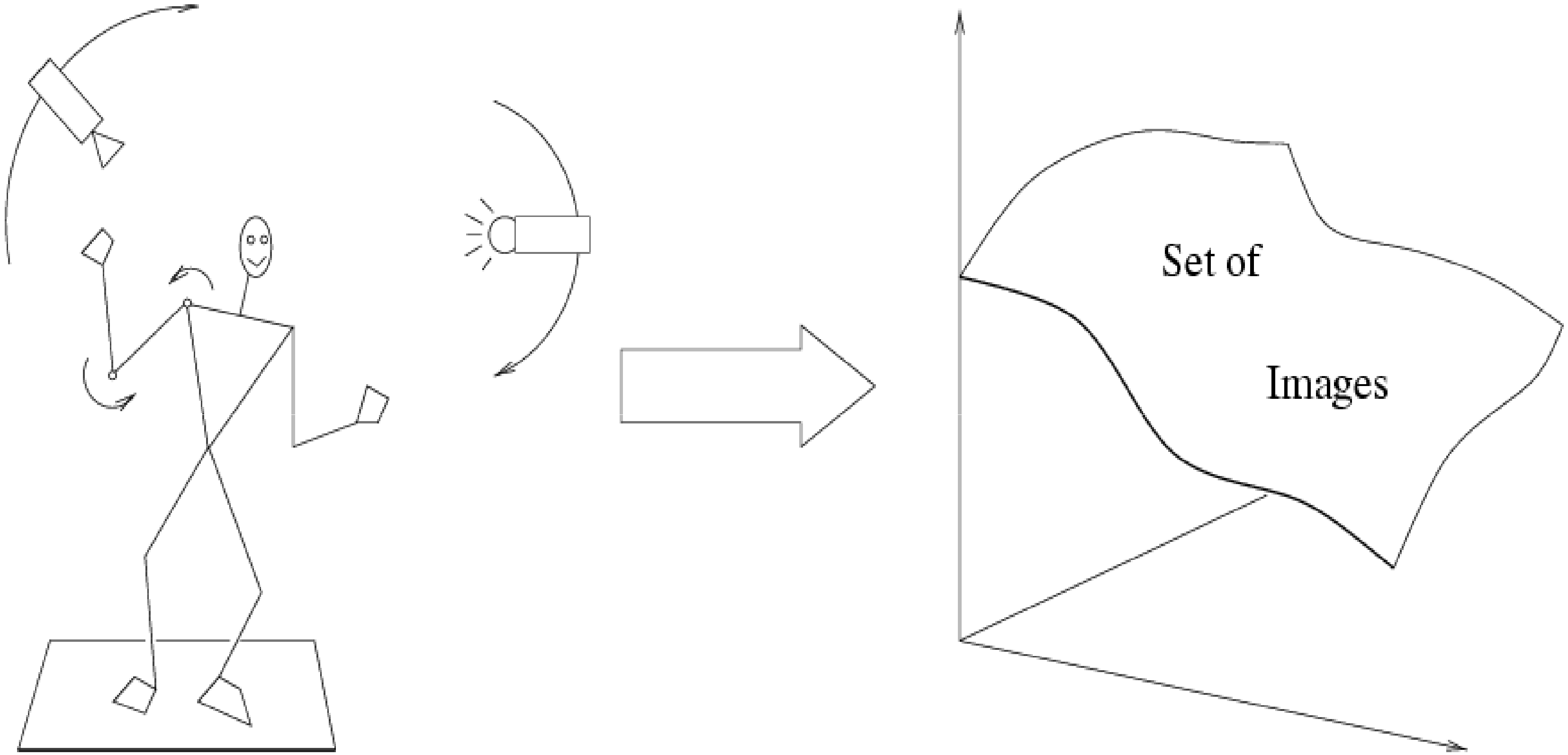
Car: present
Cow: present
Bike: not present
Horse: not present
...

- 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





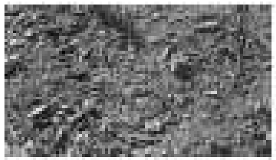

Image classification

- Given

Positive training images containing an object class



Negative training images that don't



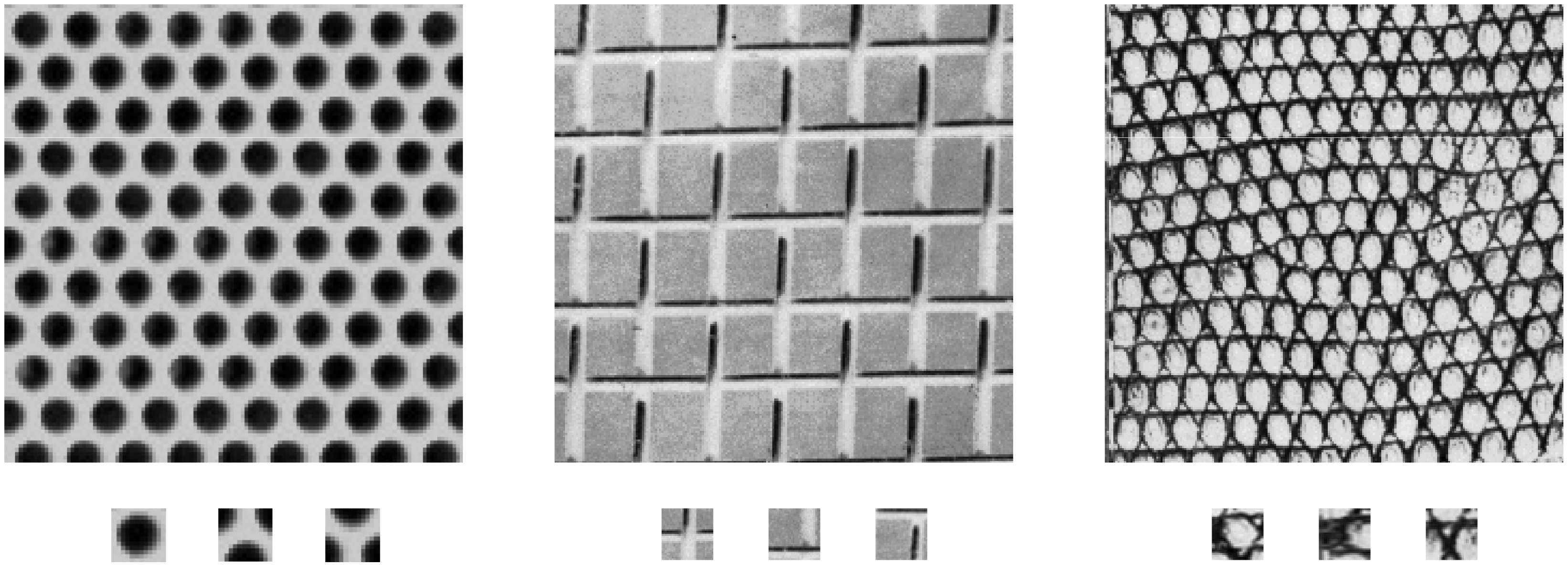
- Classify

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

?

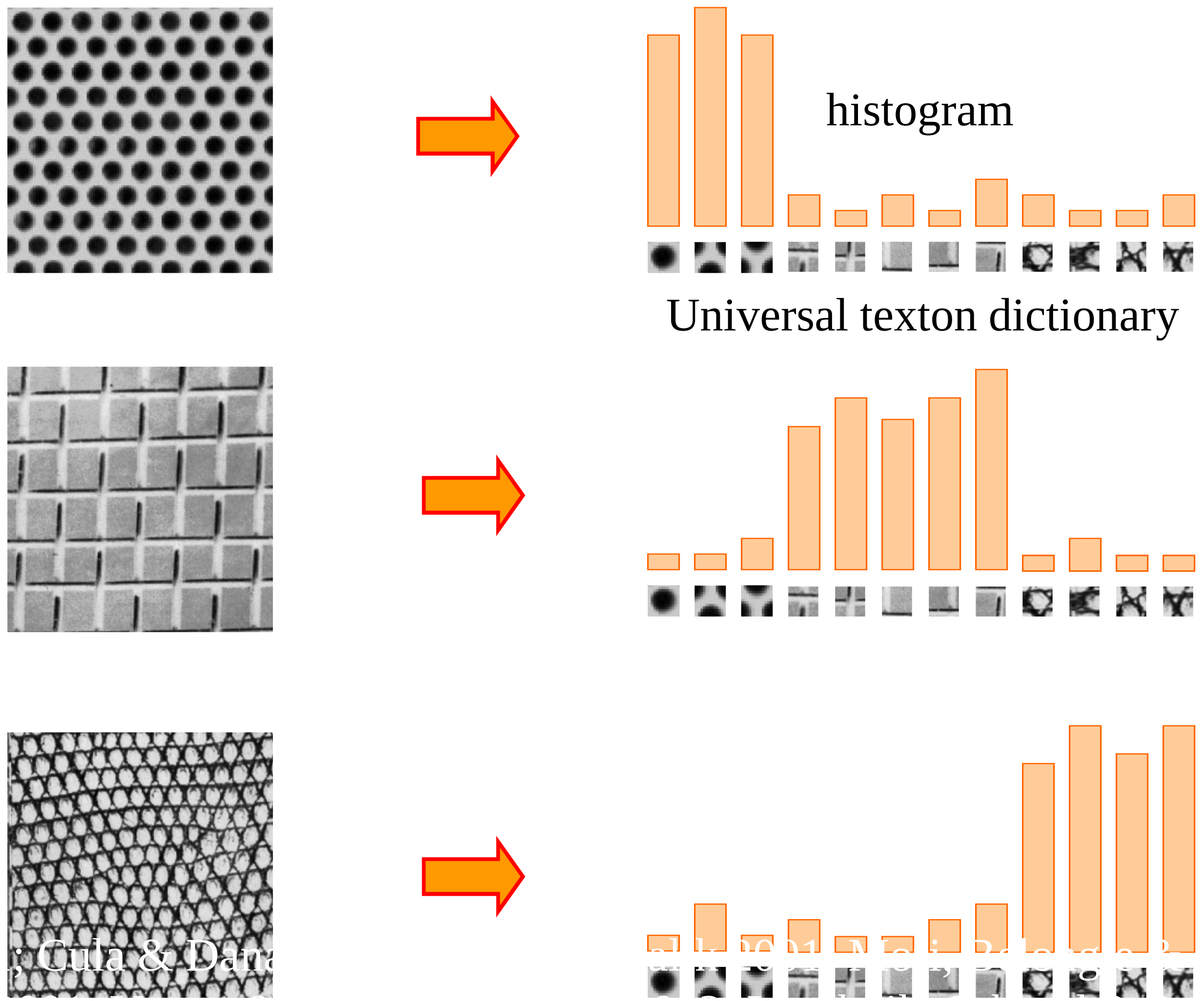
Bag-of-features – Origin: texture recognition

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



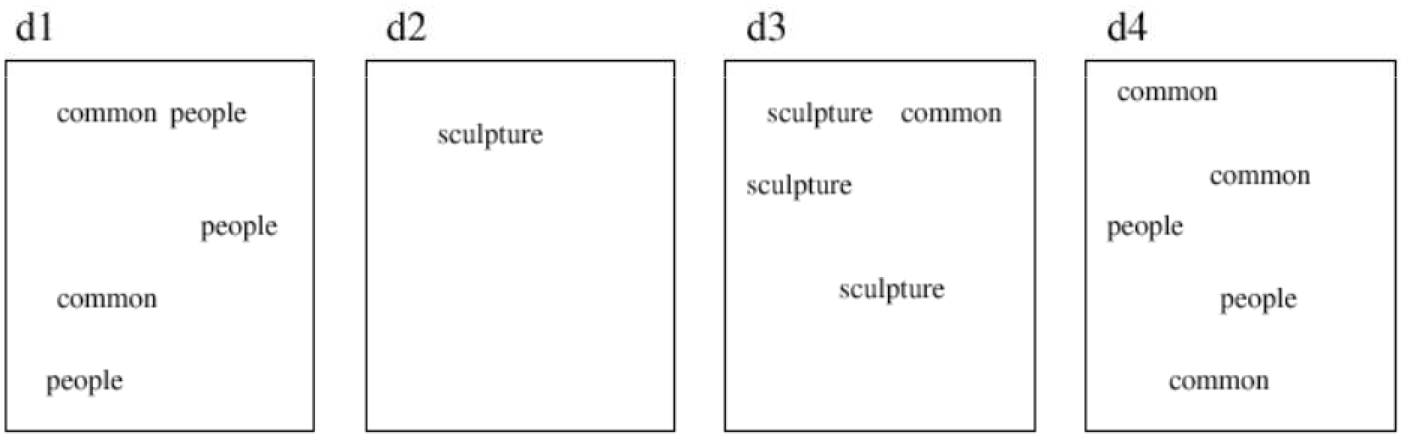
Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001 Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

Bag-of-features – Origin: texture recognition



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

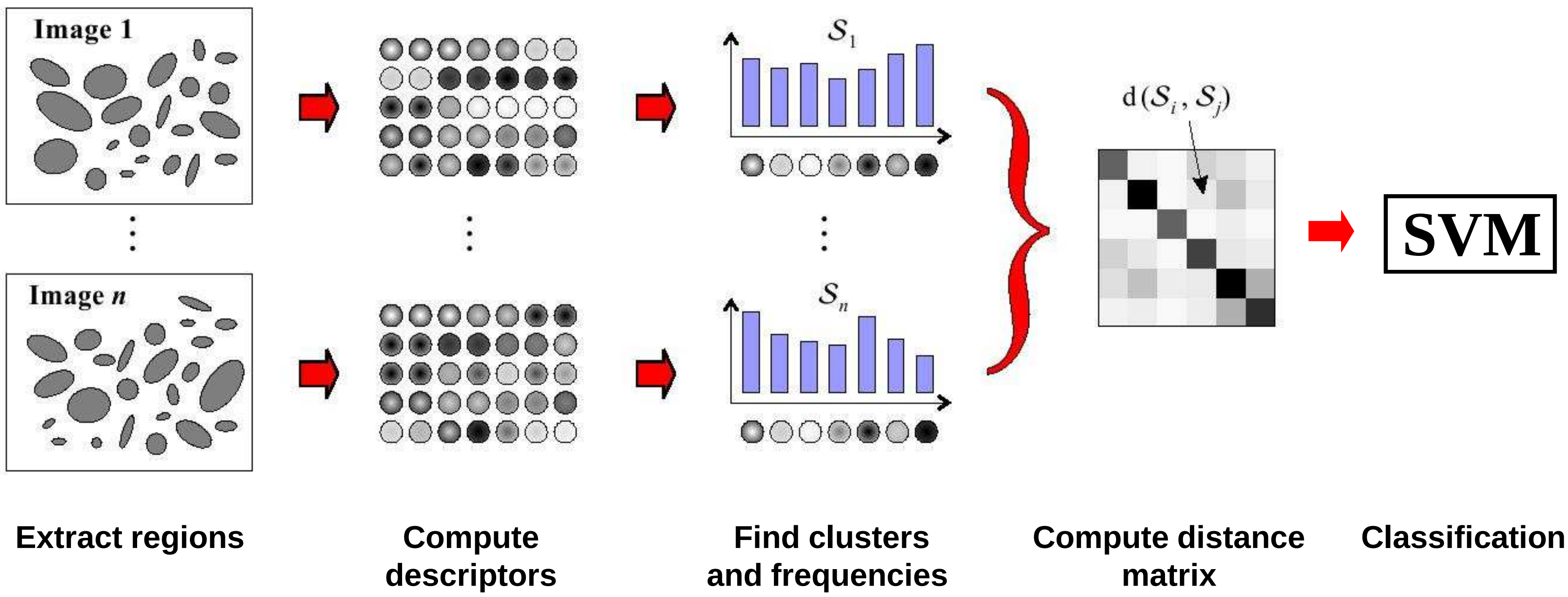
- Orderless document representation: frequencies of words from a dictionary
- Classification to determine document categories



Bag-of-words

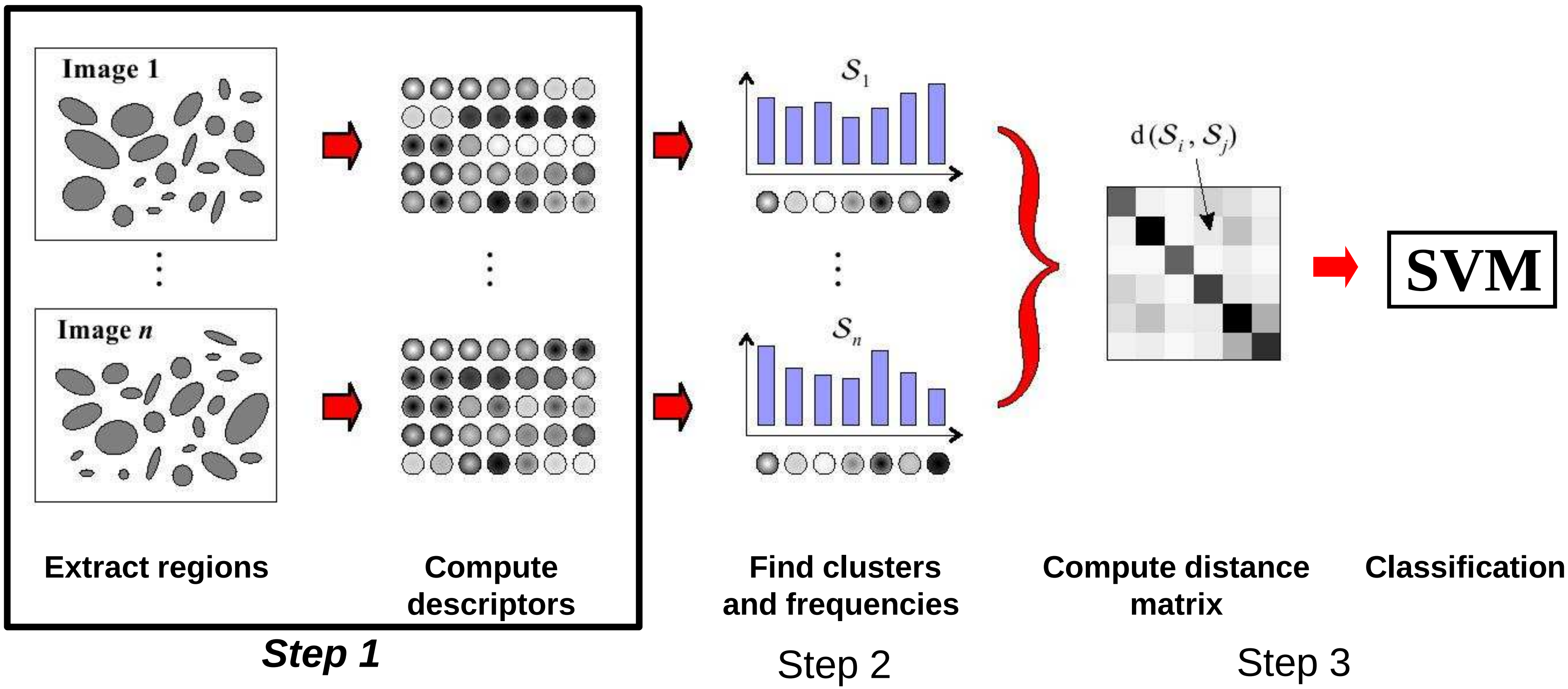
Common	2	0	1	3
People	3	0	0	2
Sculpture	0	1	3	0
…	…	…	…	…

Bag-of-features for image classification



[Csurka et al., ECCV Workshop'04], [Nowak,Jurie&Triggs,ECCV'06], [Zhang,Marszalek,Lazebnik&Schmid,IJCV'07]

Bag-of-features for image classification



Step 1: feature extraction

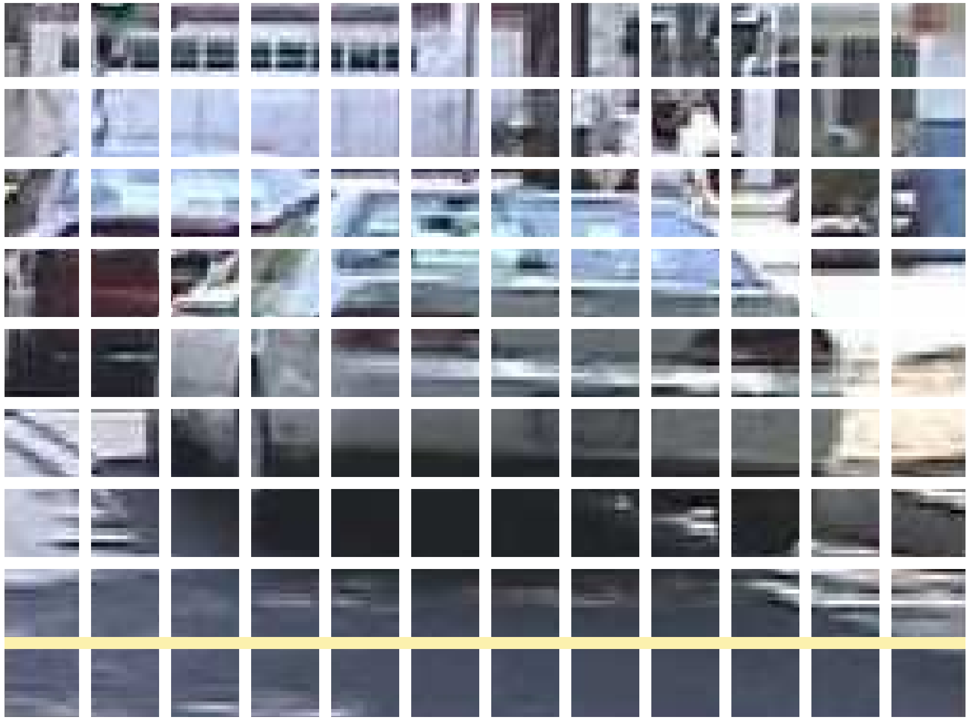
- Scale-invariant image regions + SIFT (see previous lecture)
  - Affine invariant regions give “too” much invariance
  - Rotation invariance for many realistic collections “too” much invariance

- 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

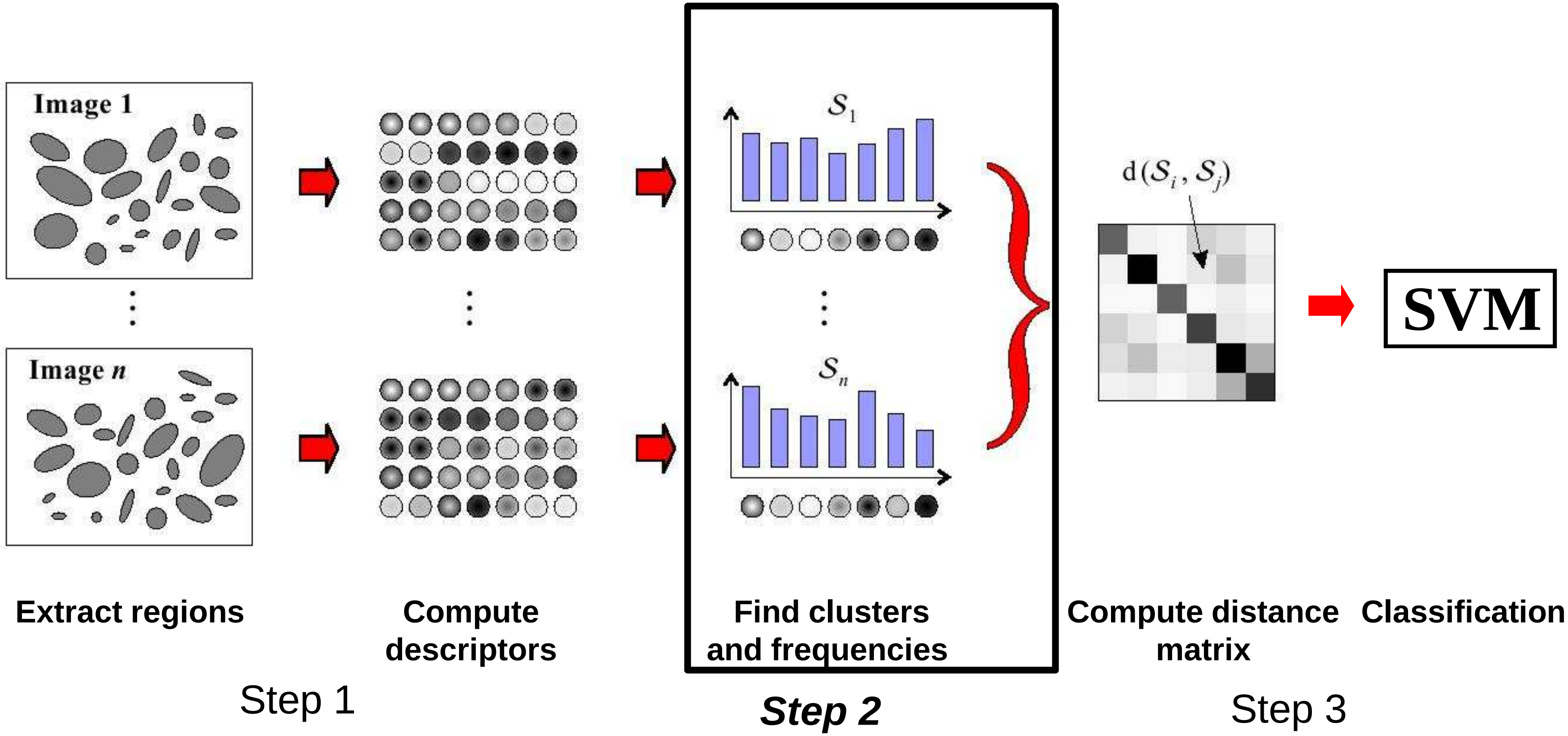


- Multi-scale dense grid: extraction of small overlapping patches at multiple scales

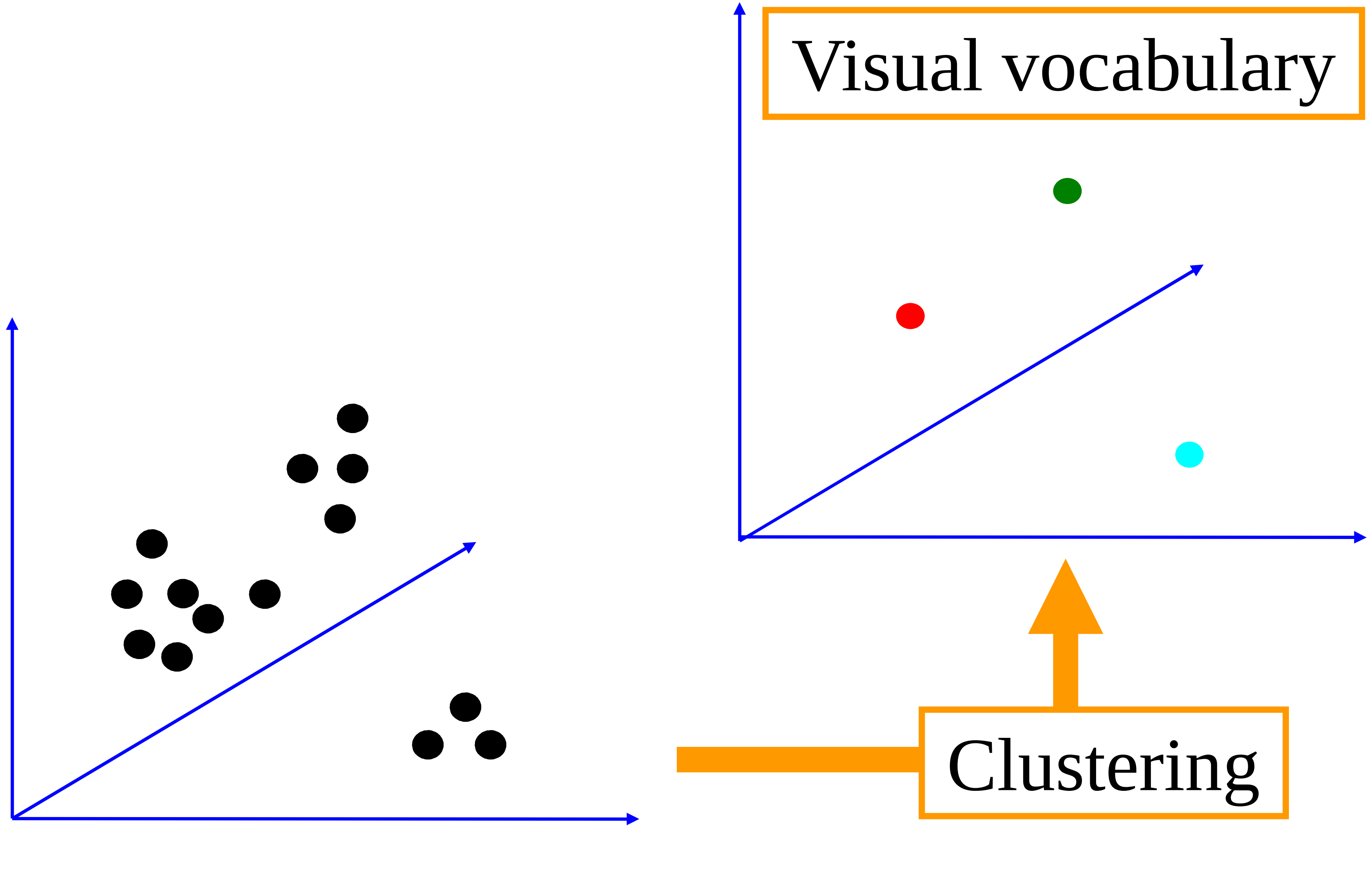
-Computation of the SIFT descriptor for each grid cells

-Exp.: Horizontal/vertical step size 3-6 pixel, scaling factor of 1.2 per level

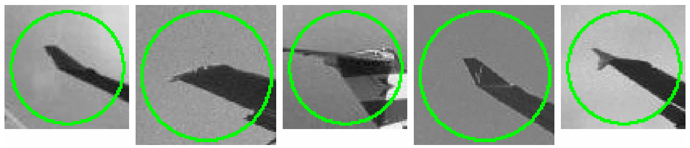
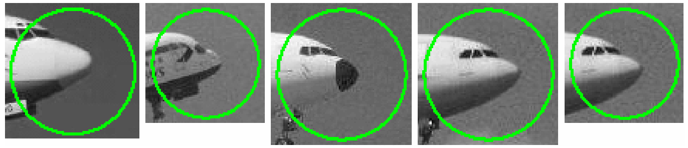
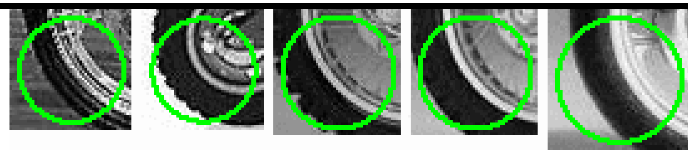
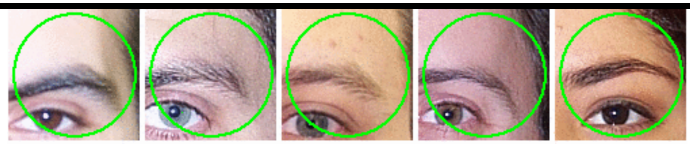
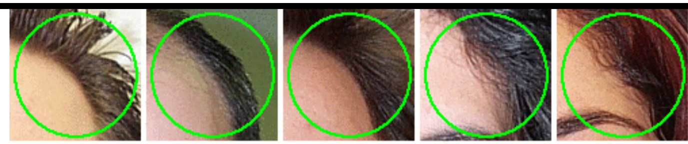
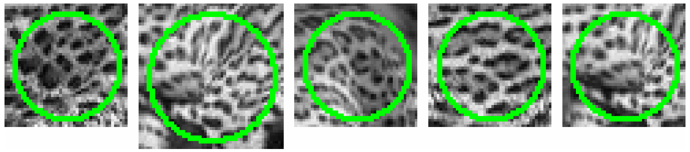
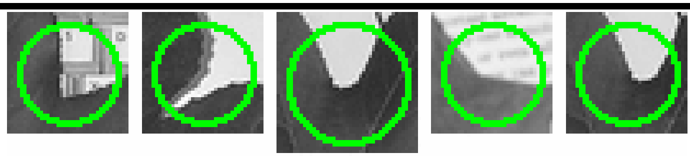
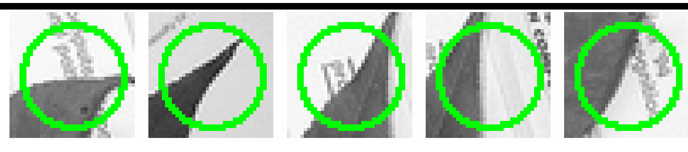
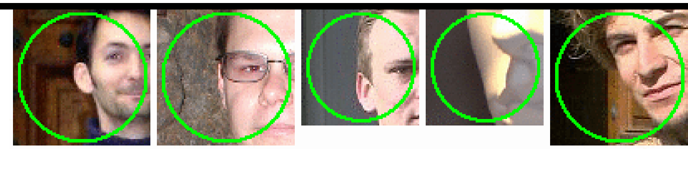
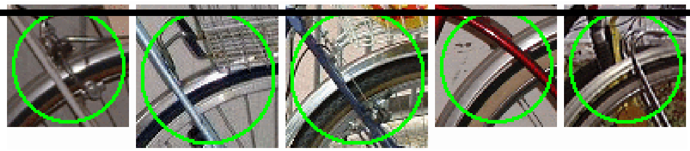
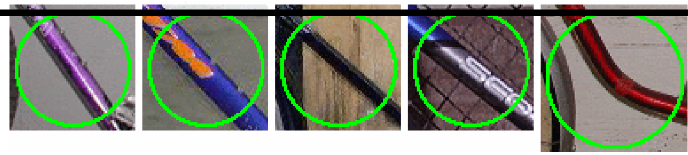
Bag-of-features for image classification



Step 2: Quantization



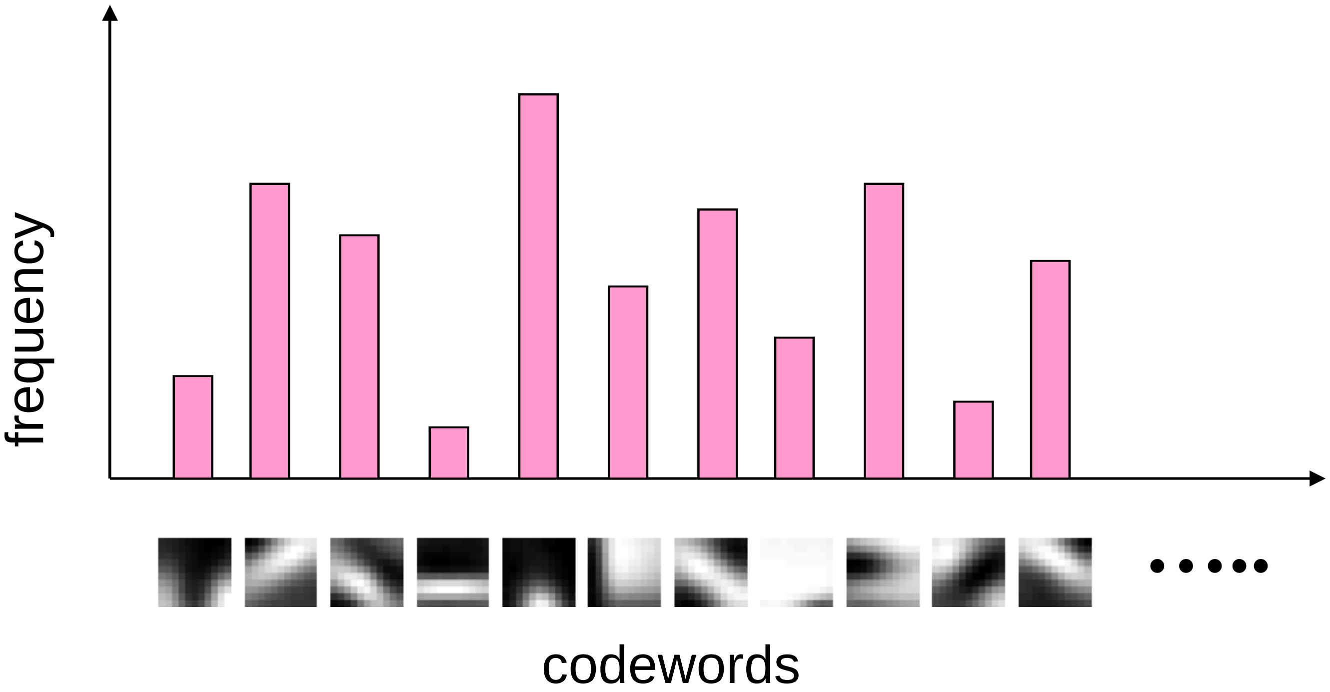
Examples for visual words

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

Image representation

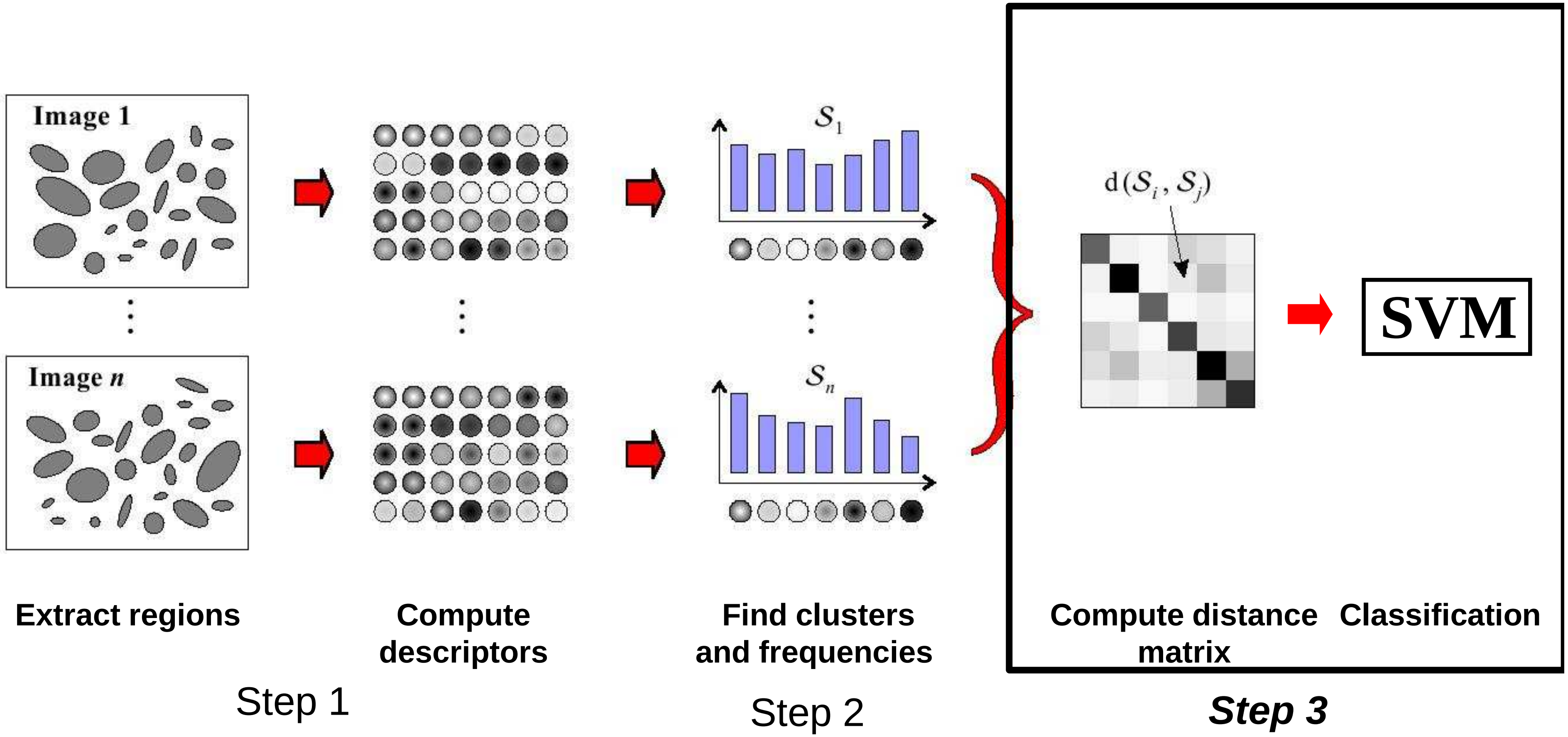


• each image is represented by a vector, typically 1000-4000 dimension, normalization with L1/L2 norm

• fine grained – represent model instances

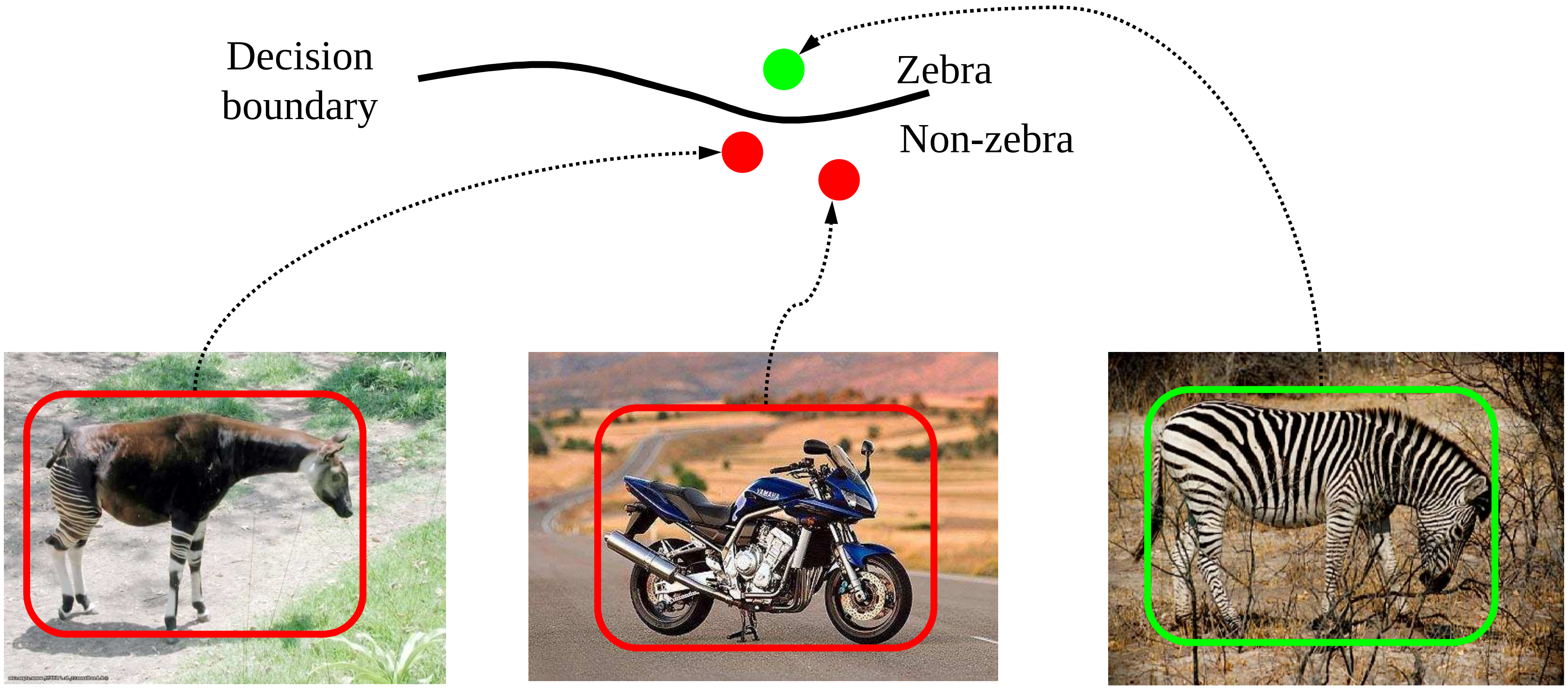
• coarse grained – represent object categories

Bag-of-features for image classification



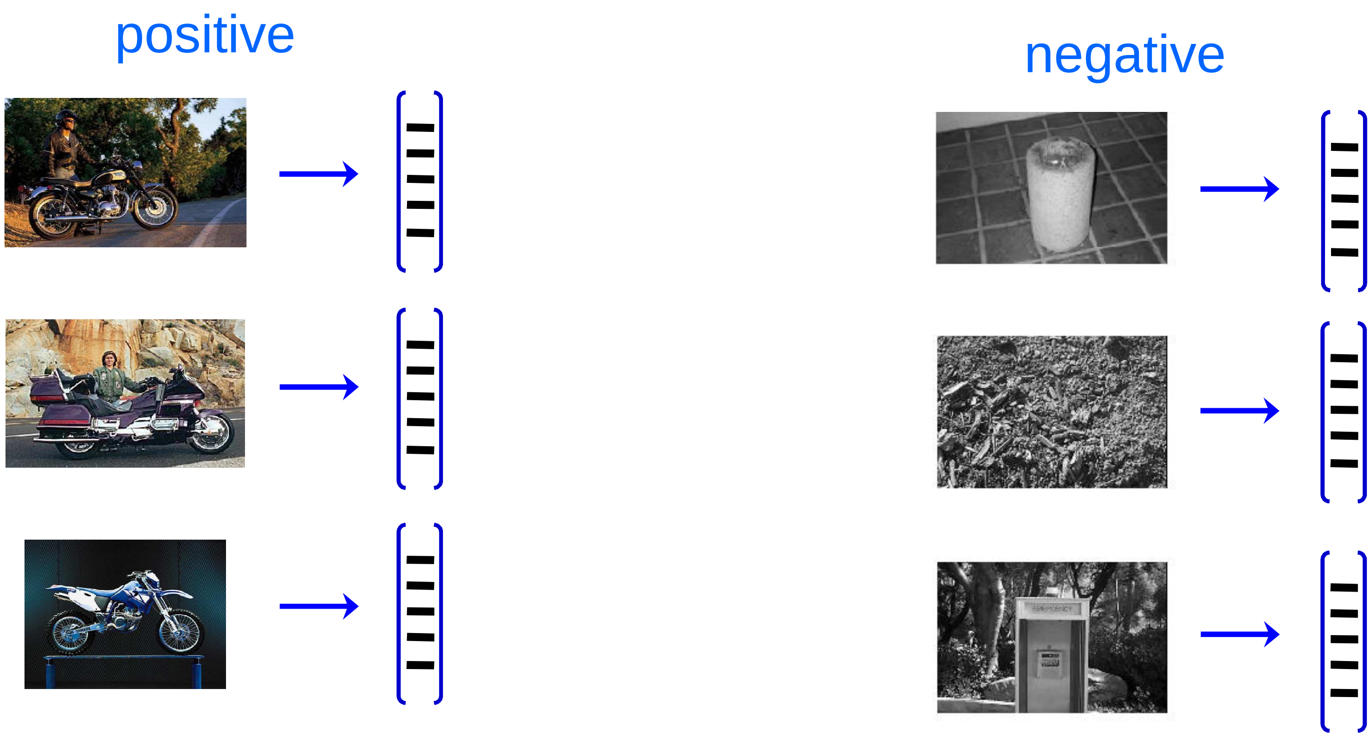
Step 3: Classification

- 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



Train classifier,e.g.SVM

Kernels for bags of features

- 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 → RBF kernel
- D can be χ^2 distance $D(h_1, h_2) = \sum_{i=1}^{N} \frac{\left(h_1(i) - h_2(i)\right)^2}{h_1(i) + h_2(i)}$
- Earth mover's distance

Combining features

- SVM with multi-channel chi-square kernel

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

- 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.

Multi-class SVMs

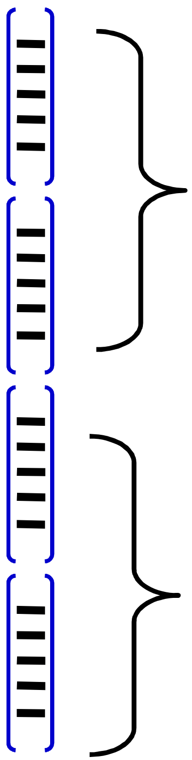
- Various direct formulations exist, but they are not widely used in practice. It is more common to obtain multi-class SVMs by combining two-class SVMs in various ways.

- 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

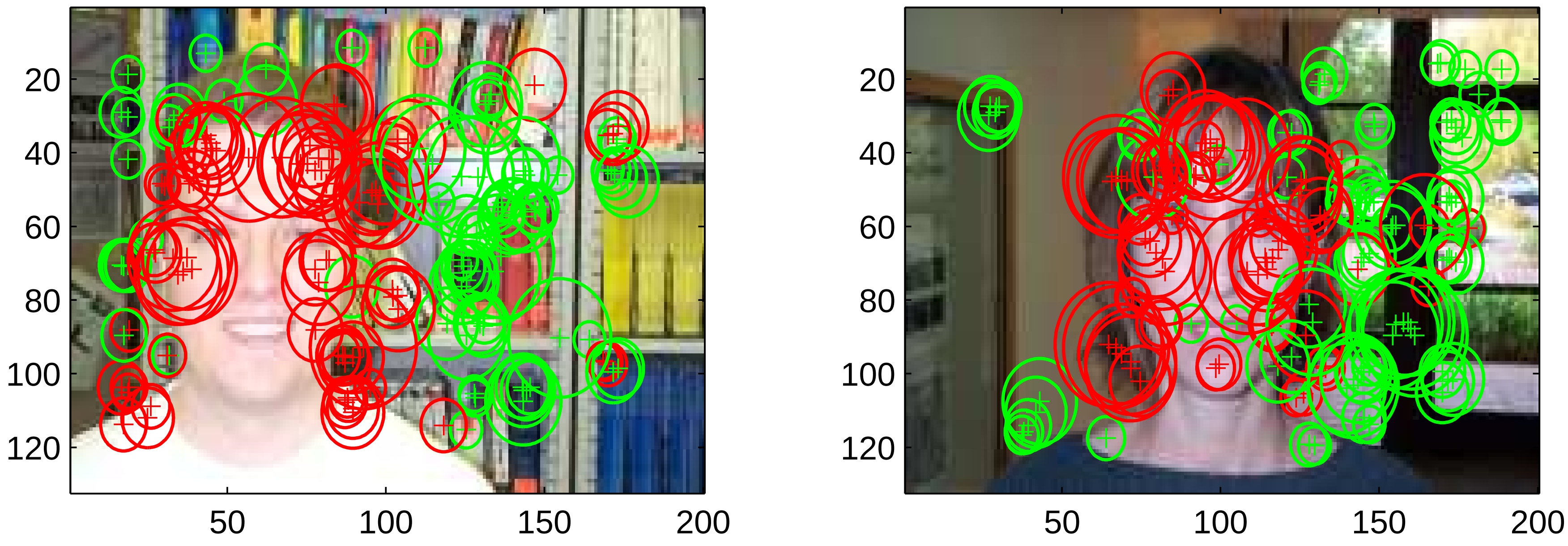


foreground words – high weight

background words – low weight

Illustration

Localization according to visual word probability



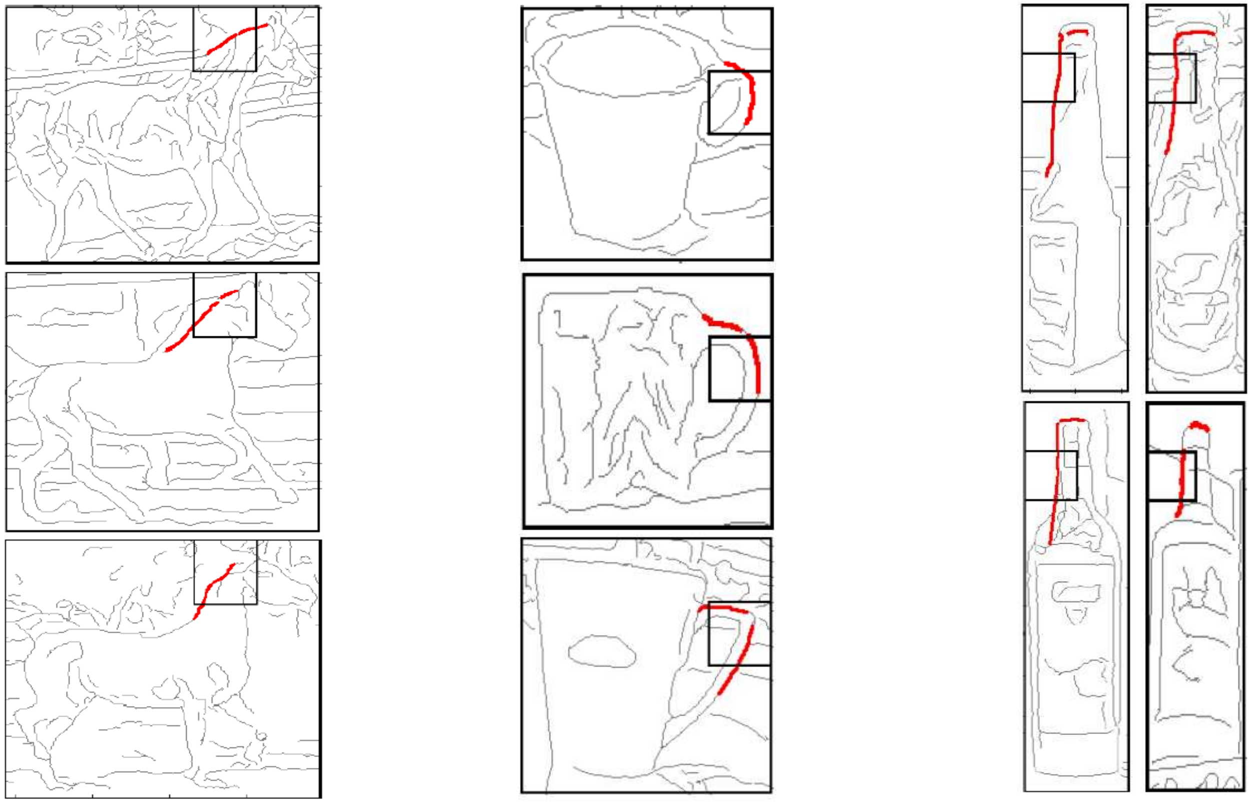
foreground word more probable

background word more probable

Illustration

A linear SVM trained from positive and negative window descriptors

A few of the highest weighted 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

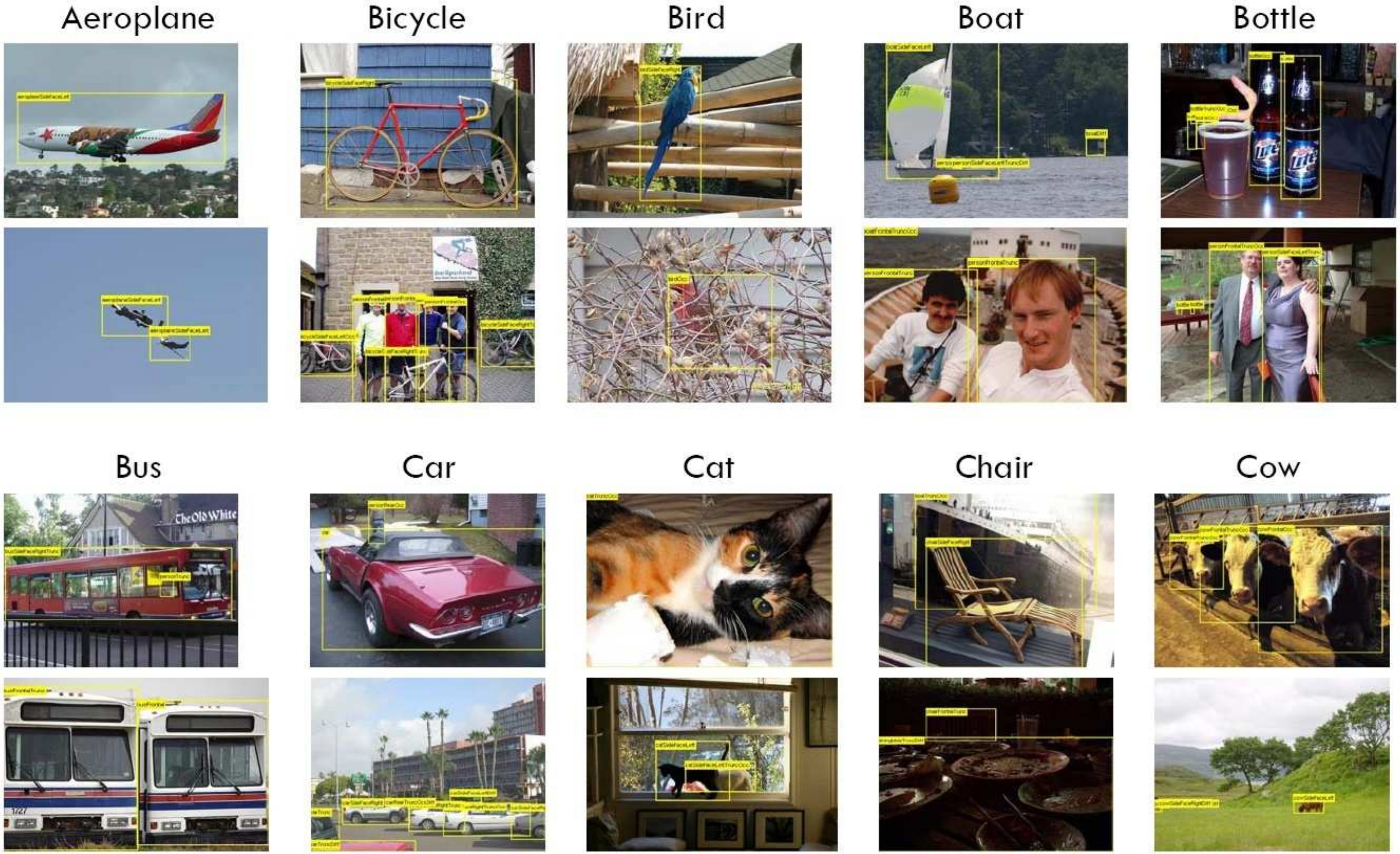
Evaluation of image classification

- 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

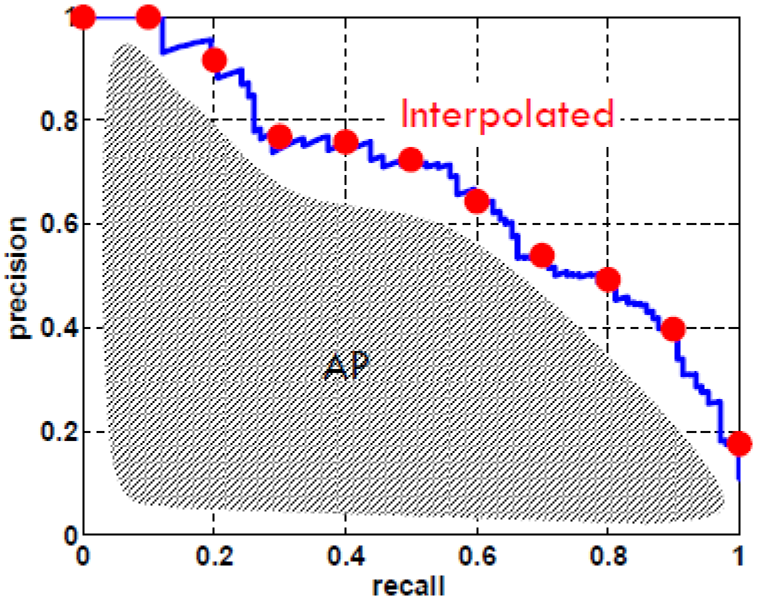
PASCAL 2007 dataset



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
- …..

Spatial pyramid matching

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

[Lazebnik, Schmid & Ponce, CVPR 2006]

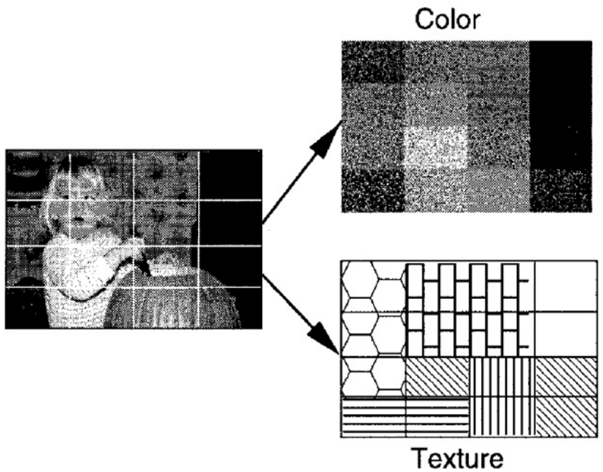
Related work

Similar approaches:

Subblock description [Szummer & Picard, 1997]

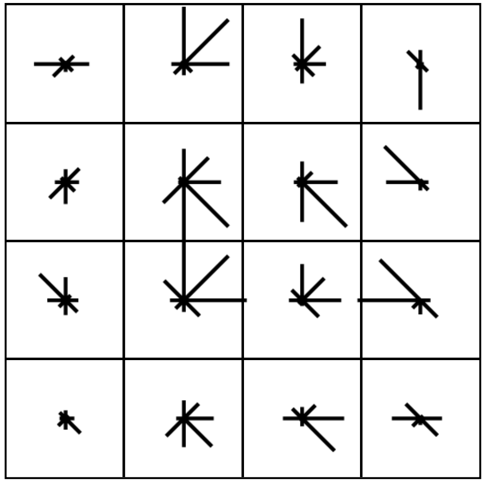
SIFT [Lowe, 1999]

GIST [Torralba et al., 2003]



Szummer & Picard (1997)

SIFT

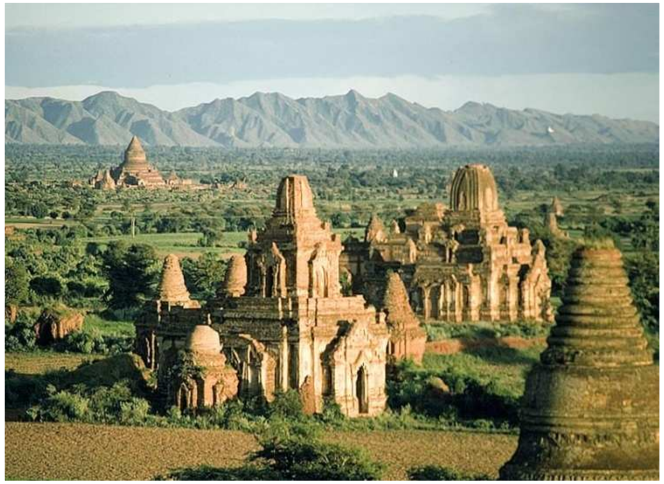


Lowe (1999, 2004)

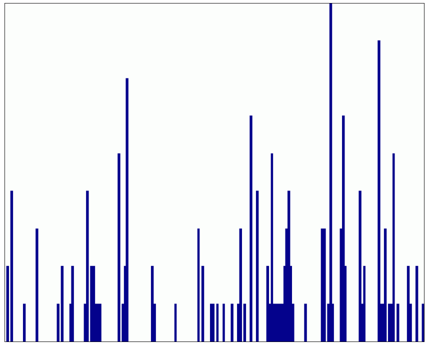
Gist

Torralba et al. (2003)

Spatial pyramid representation

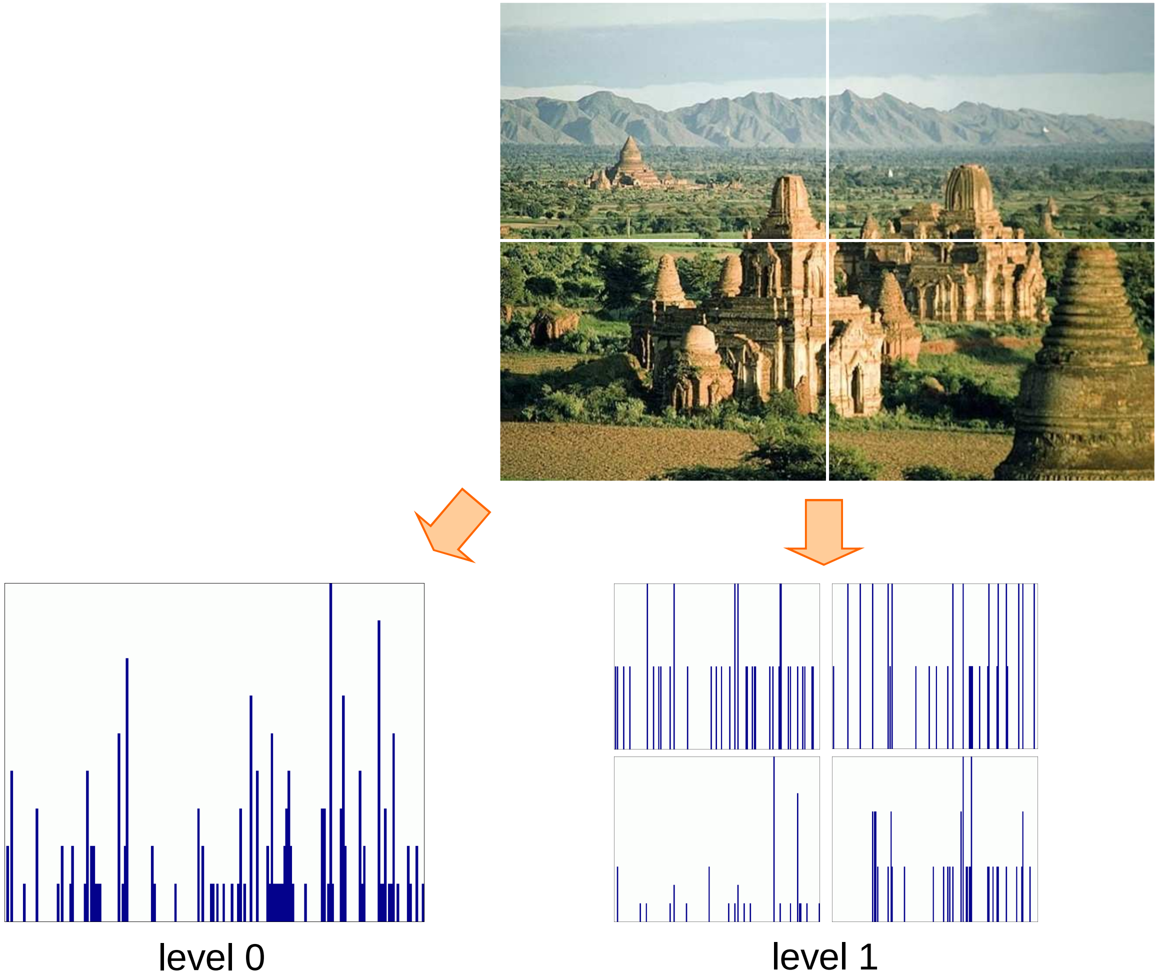


Locally orderless representation at several levels of spatial resolution



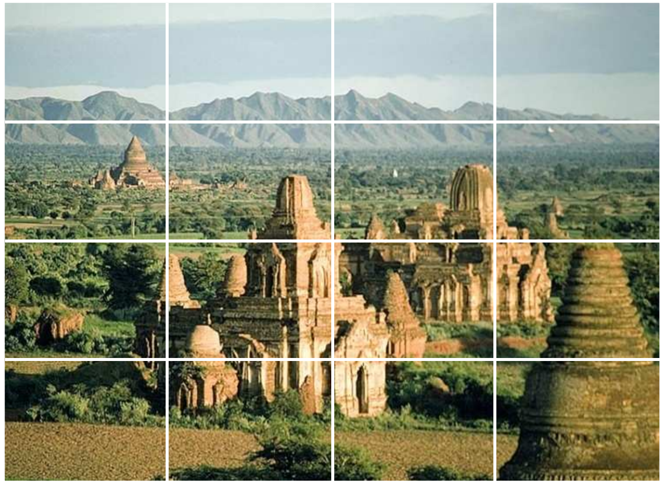
level 0

Spatial pyramid representation

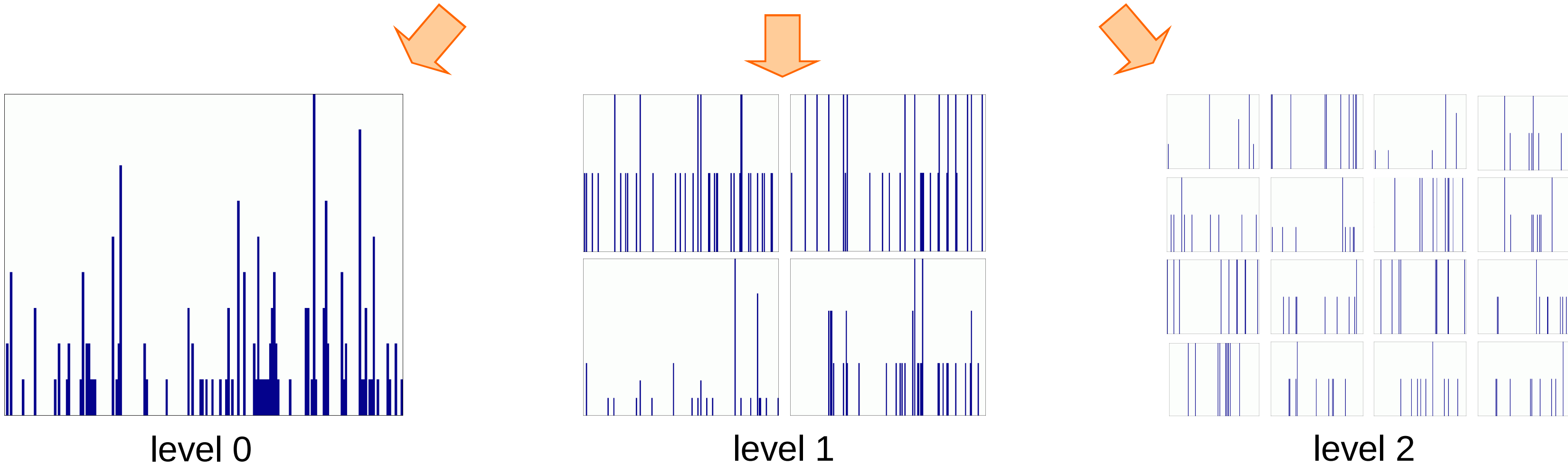


Locally orderless representation at several levels of spatial resolution

Spatial pyramid representation

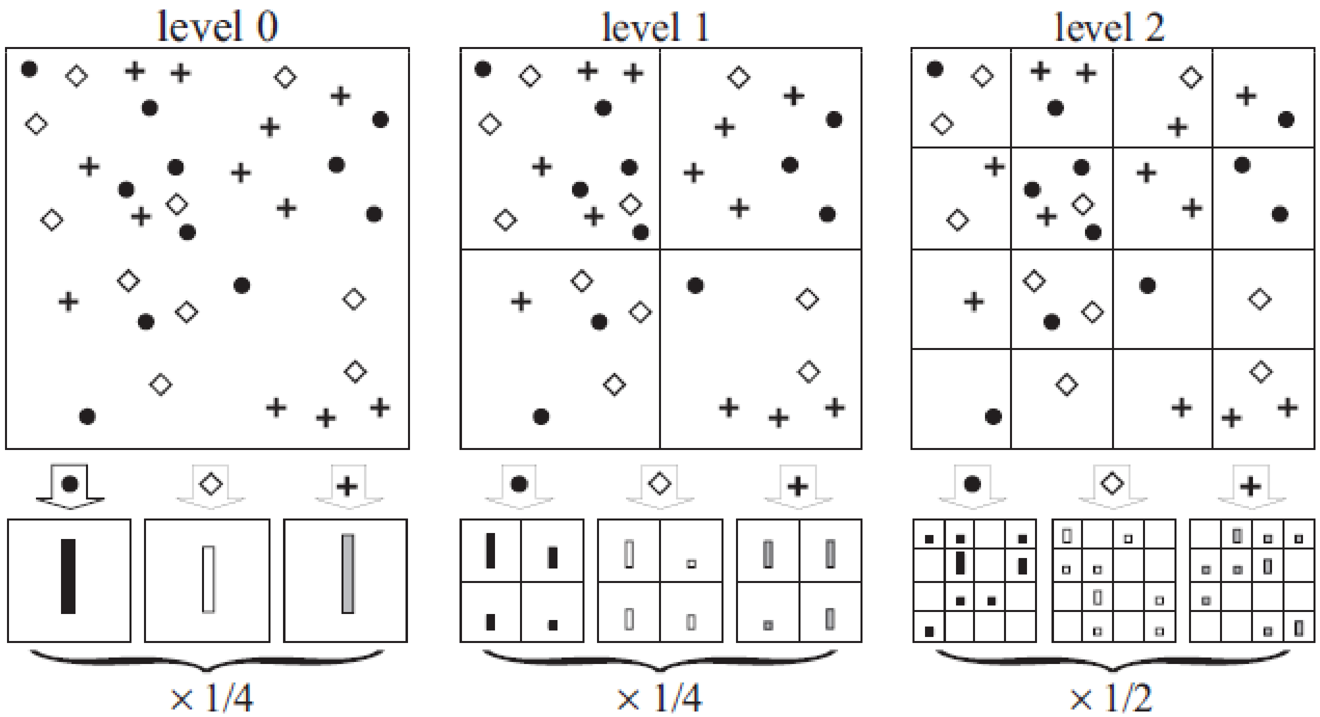


Locally orderless representation at several levels of spatial resolution

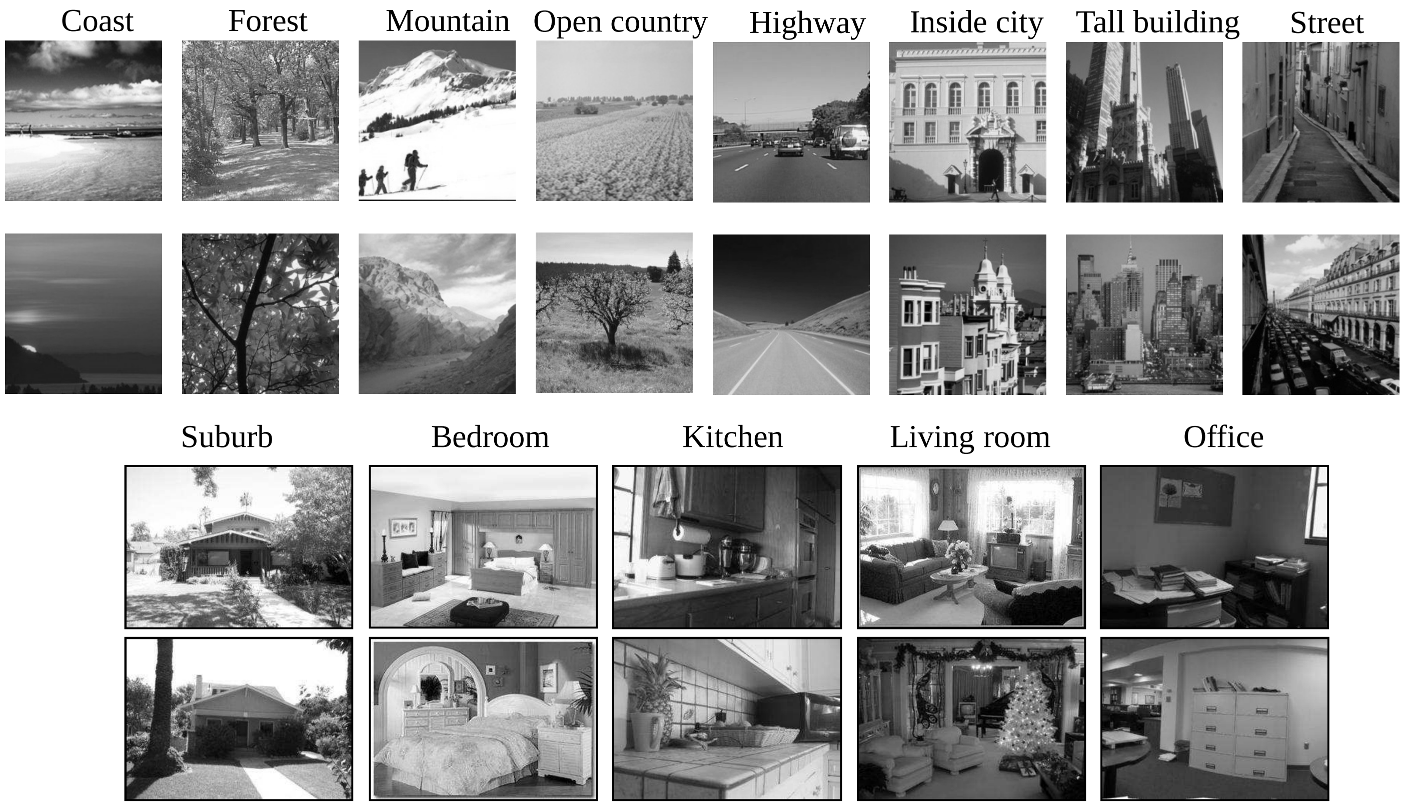


Spatial pyramid matching

- Combination of spatial levels with pyramid match kernel
  [Grauman & Darell'05]
- Intersect histograms, more weight to finer grids



Scene dataset [Labzenik et al.'06]



4385 images

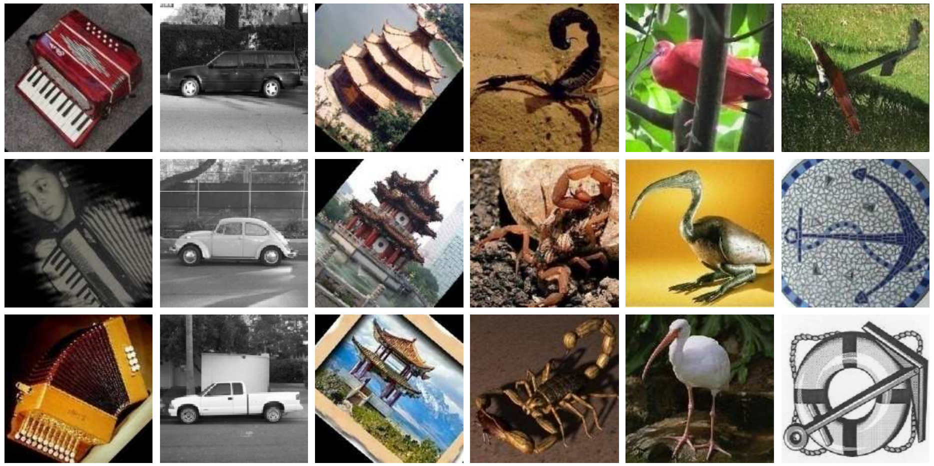
15 categories

Scene classification

L	Single-level	Pyramid
0(1x1)	72.2±0.6	
1(2x2)	77.9±0.6	79.0 ±0.5
2(4x4)	79.4±0.3	*81.1 ±0.3*
3(8x8)	77.2±0.4	80.7 ±0.3

Retrieval examples

Category classification – CalTech101



L	Single-level	Pyramid
0(1x1)	41.2±1.2	
1(2x2)	55.9±0.9	57.0 ±0.8
2(4x4)	63.6±0.9	*64.6 ±0.8*
3(8x8)	60.3±0.9	64.6 ±0.7

Discussion

- Summary
  - Spatial pyramid representation: appearance of local image patches + coarse global position information
  - Substantial improvement over bag of features
  - Depends on the similarity of image layout
- Extensions
  - Flexible, object-centered grid

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

- Image descriptors
  - Fisher vector (high dimensional)
  - Normalization: square-rooting or latent MOG+ L2 normalization

    [Image categorization using Fisher kernels of non-iid image models, Cinbis, Verbeek, Schmid, CVPR'12] [Perronnin'10]

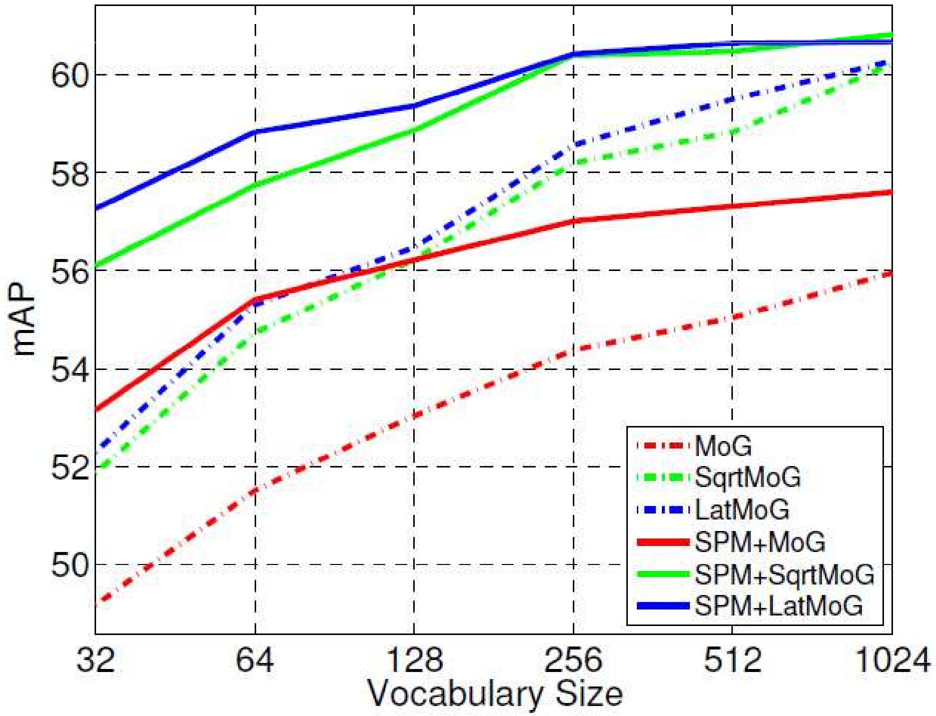
- Classification approach
  - Linear classifiers
  - One versus rest classifier
  - Stochastic gradient descent optimization

    [Towards good practice in large-scale learning for image classification, Perronnin, Akata, Harchaoui, Schmid, CVPR'12]

Evaluation image description

- Comparing on PASCAL VOC'07 linear classifiers with
  - Fisher vector
  - Sqrt transformation of Fisher vector
  - Latent GMM of Fisher vector

- Sqrt transform + latent MOG models lead to improvement
- State-of-the-art performance obtained with linear classifier



Evaluation image description

Fisher versus BOF vector + linear classifier on Pascal Voc'07

SPM	Method	64	128	256	512	1024
No	BoW	20.1	29.0	36.2	40.7	44.1
No	SqrtBoW	21.0	29.5	37.4	**41.3**	**46.1**
No	LatBoW	**22.9**	**30.1**	**38.9**	41.2	44.5
Yes	BoW	37.1	40.1	42.4	46.4	48.9
Yes	SqrtBoW	37.8	41.2	44.6	47.8	51.6
Yes	LatBoW	**39.3**	**41.7**	**45.3**	**48.7**	**52.2**

SPM	Method	32	64	128	256	512	1024
No	MoG	49.2	51.5	53.0	54.4	55.0	55.9
No	SqrtMoG	51.9	54.7	56.2	58.2	58.8	60.2
No	LatMoG	**52.3**	**55.3**	**56.5**	**58.6**	**59.5**	**60.3**
Yes	MoG	53.2	55.4	56.2	57.0	57.3	57.6
Yes	SqrtMoG	56.1	57.7	58.9	**60.4**	60.5	**60.8**
Yes	LatMoG	**57.3**	**58.8**	**59.4**	**60.4**	**60.6**	60.7

- Fisher improves over BOF
- Fisher comparable to BOF + non-linear classifier
- Limited gain due to SPM on PASCAL
- Sqrt helps for Fisher and BOF
- [Chatfield et al. 2011]

Large-scale image classification

- Classification approach
  - 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%)

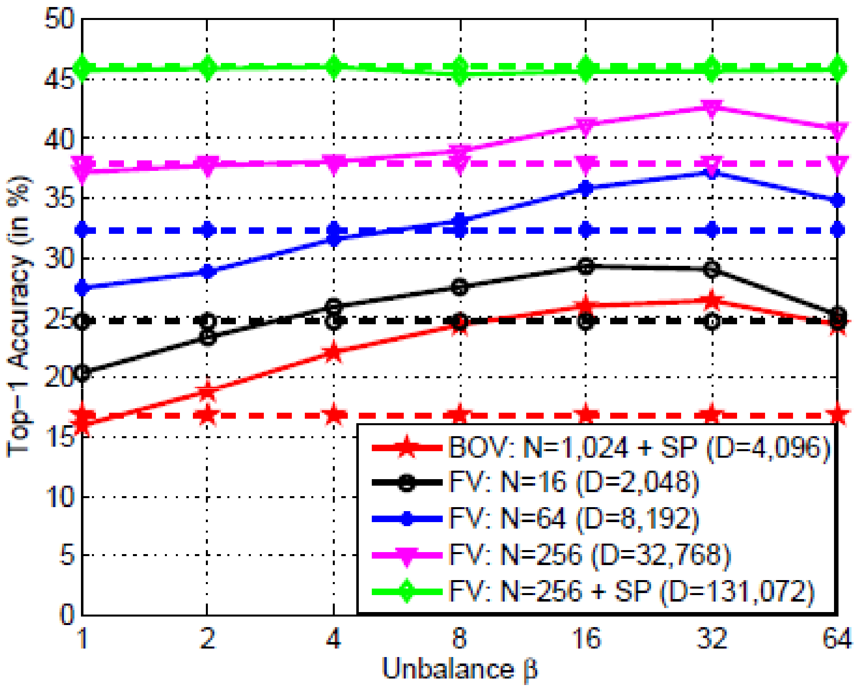
Experimental results

- 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

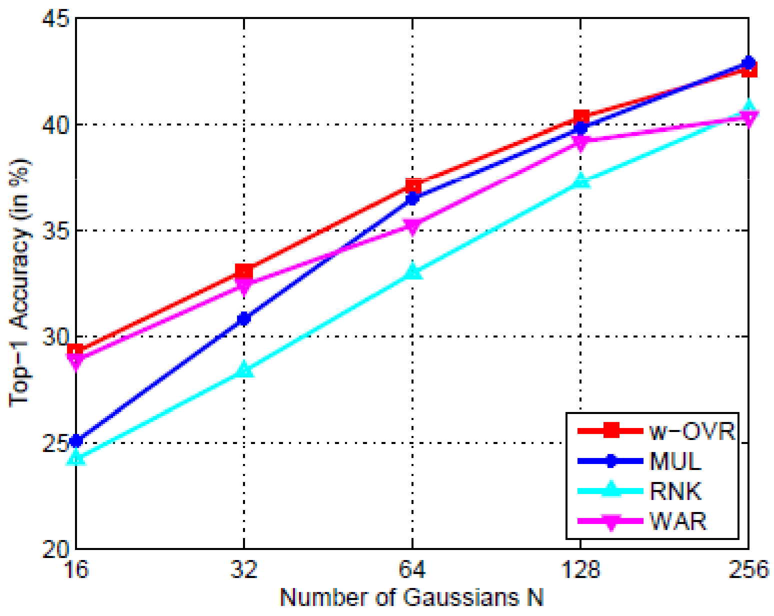
One-versus-rest works

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

		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
- Improves over state of the art: 6.4% [Deng et. al] and WAR [Weston et al.]

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%)

Conclusion

- *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

Conclusion

- State-of-the-art performance for large-scale image classification
- Code on-line available at http://lear.inrialpes.fr/software
- Future work
  - Beyond a single representation of the entire image
  - Take into account the hierarchical structure