RECENT ADVANCEMENTS ON THE BAG OF VISUAL WORDS MODEL FOR IMAGE CLASSIFICATION AND CONCEPT DETECTION

Costantino Grana and Giuseppe Serra

DIPARTIMENTO DI INGEGNERIA Enzo Ferrari

Centro Interdipartimentale di Ricerca Softech-ICT

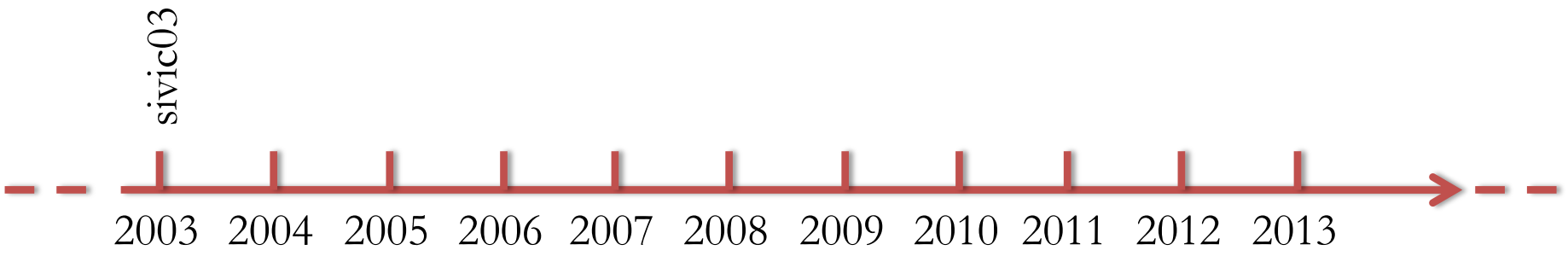
Università di Modena e Reggio Emilia, Italia

http://imagelab.ing.unimore.it

A QUICK SUMMARY OF WHAT WE WILL TOUCH IN THE TUTORIAL WITH A SPECIFIC REFERENCE TO THE PAPERS WHICH PROPOSED THE METHOD

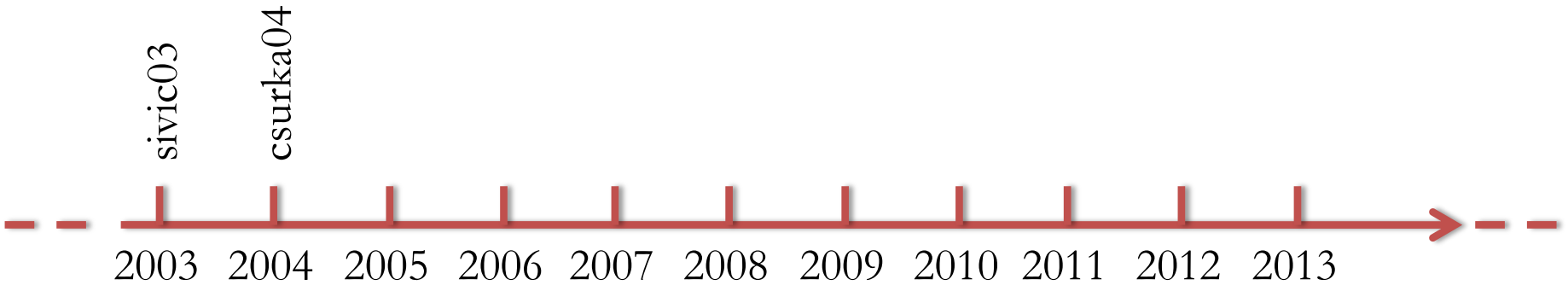
TIMELINE

TIMELINE



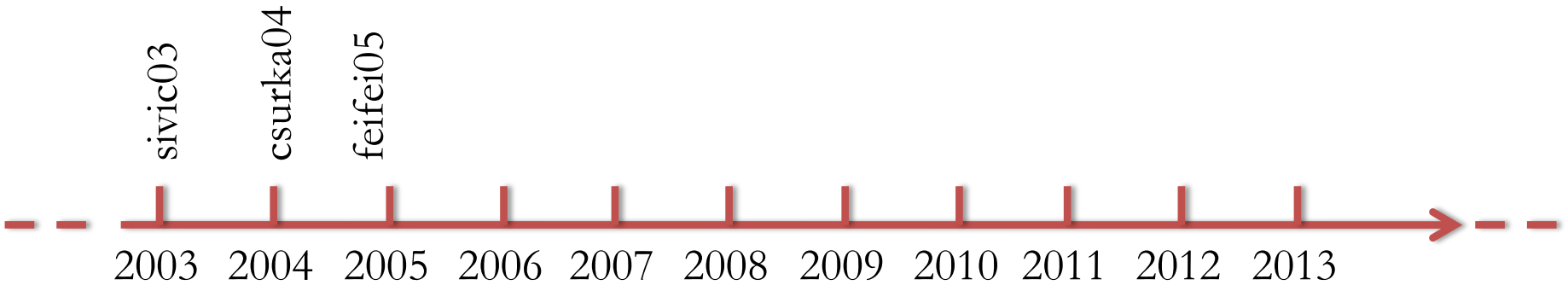
- Text retrieval techniques applied to video shot retrieval.

TIMELINE



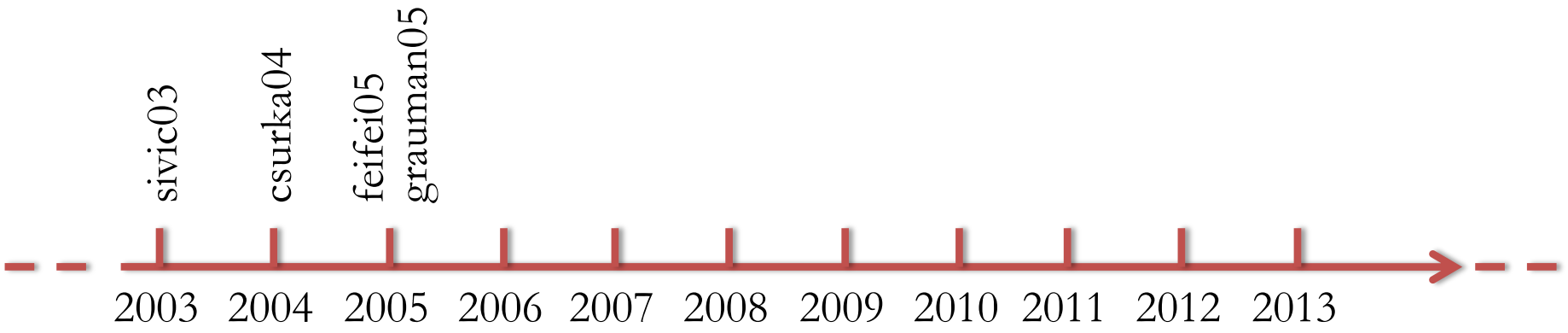
- Clear description (and name) of the bag of keypoints.

TIMELINE



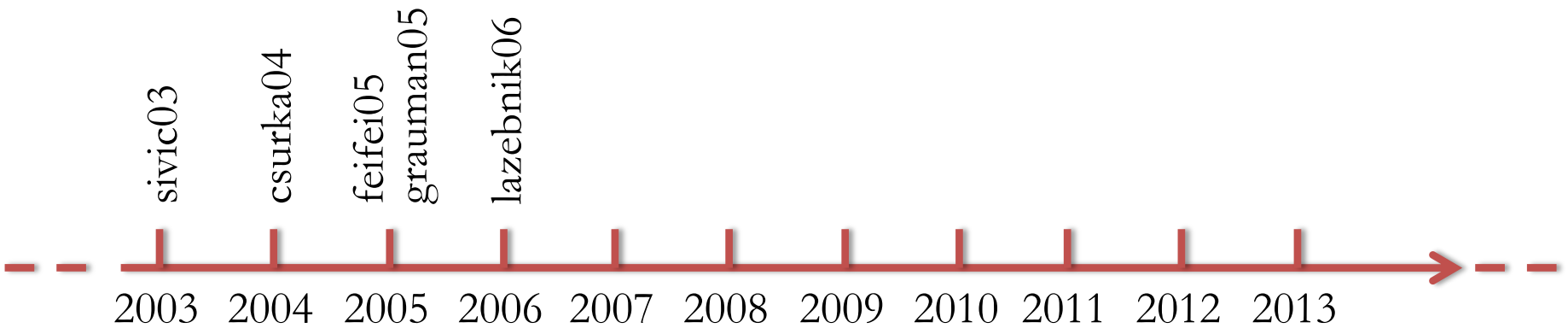
- Remove the detector using a fixed grid.

TIMELINE



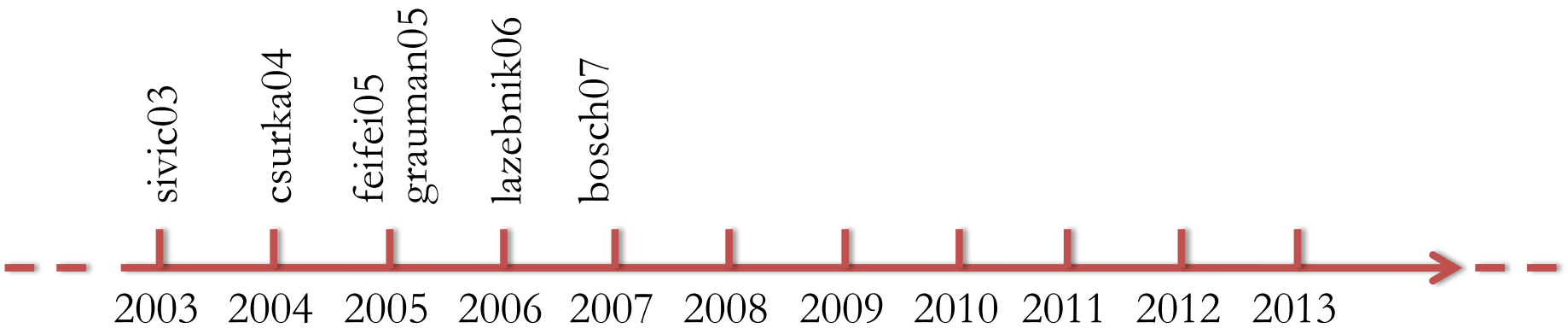
- Pyramid match kernel, histograms to approximate matched features.

TIMELINE



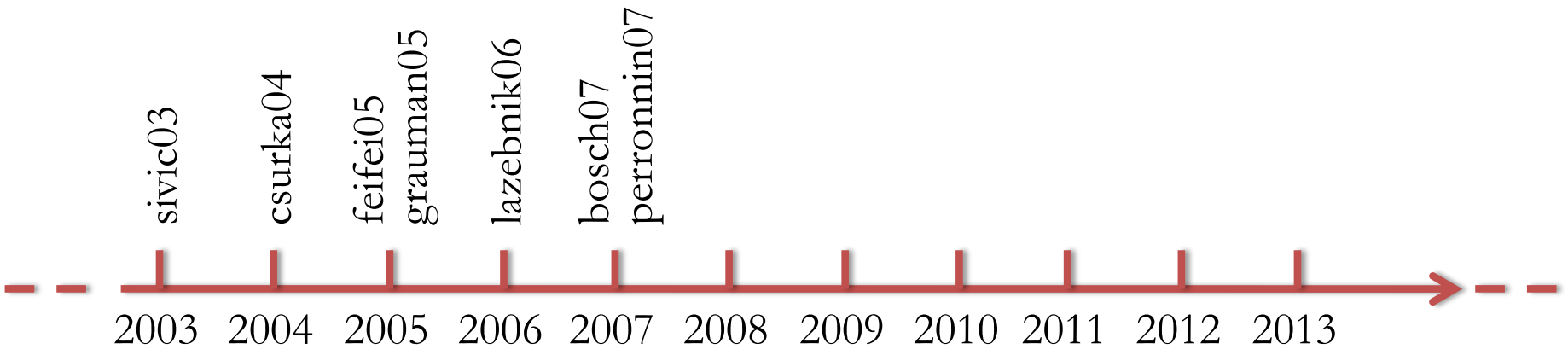
- Borrows the pyramid match kernel idea and proposes the spatial pyramid matching.

TIMELINE



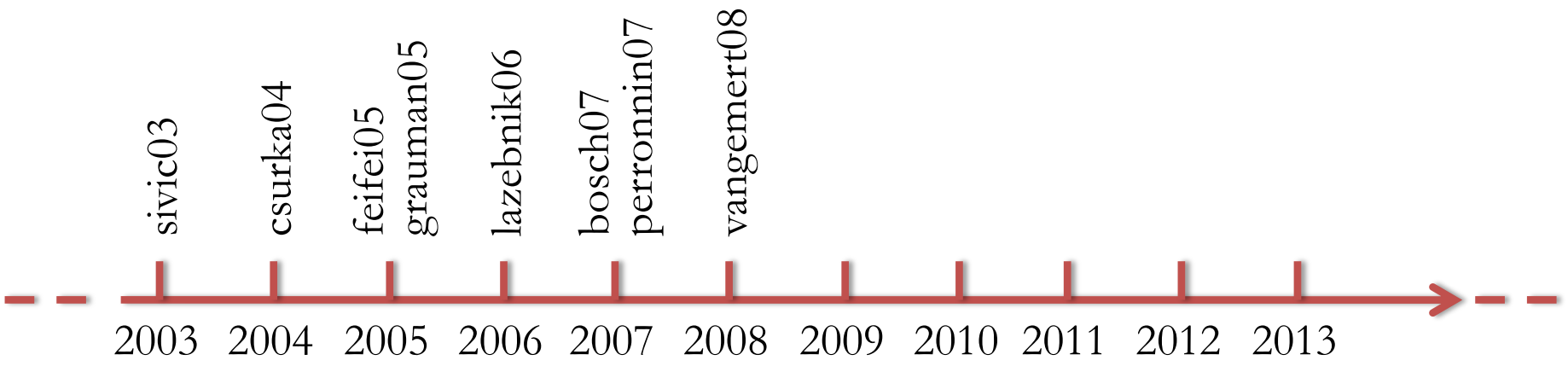
- Introduces multiple scales per point and builds a Pyramid Histogram Of visual Words (PHOW).

TIMELINE



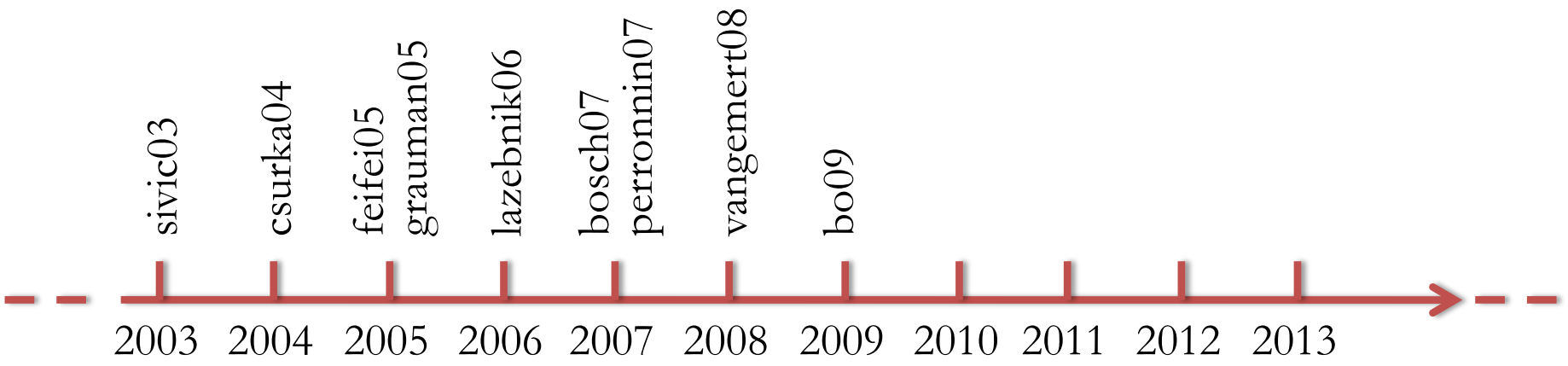
- Introduces the Fisher Kernel for image classification (but is somewhat ignored…).

TIMELINE



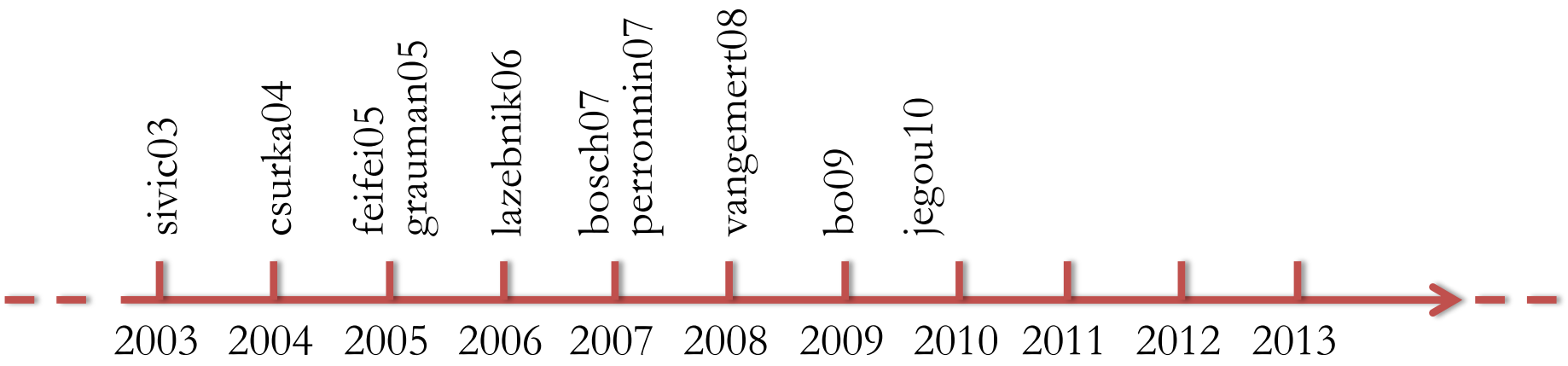
- Tackles quantization issues with soft assignment.

TIMELINE



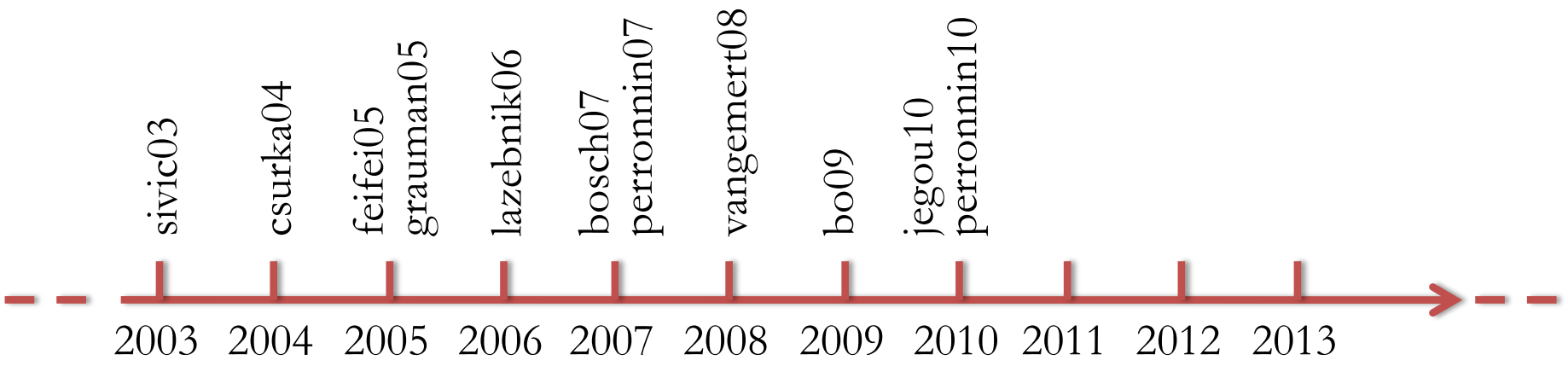
- Efficient Match Kernels to begin addressing large scale.

TIMELINE



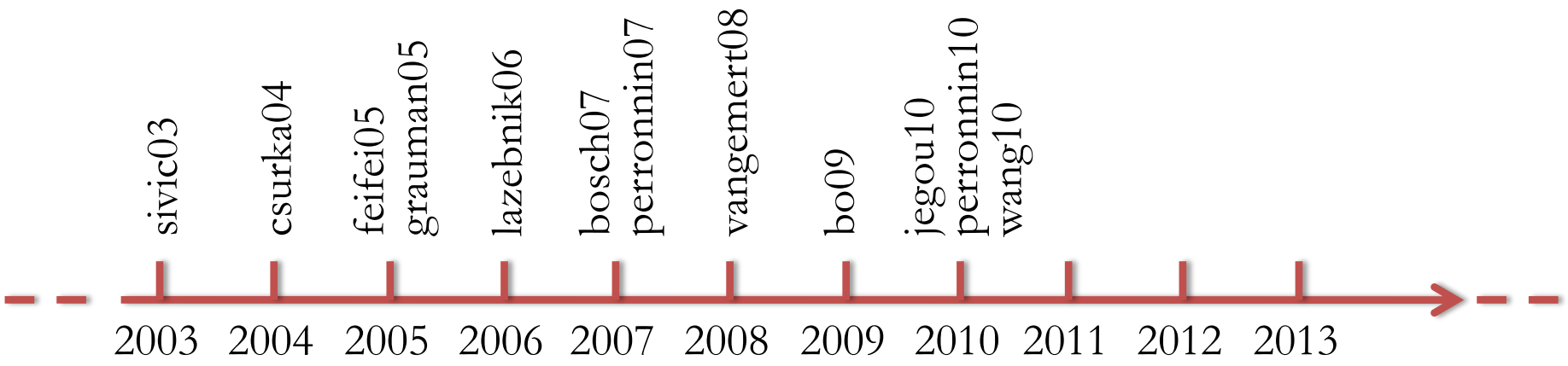
- Introduces the VLAD descriptor.

TIMELINE



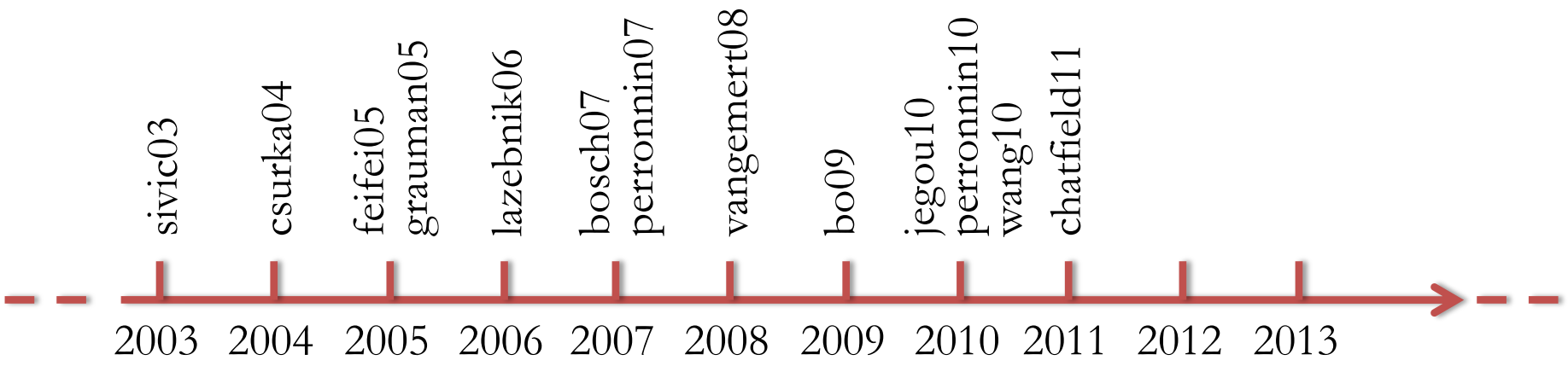
- Improves the Fisher Vectors descriptor (making it state of the art).

TIMELINE



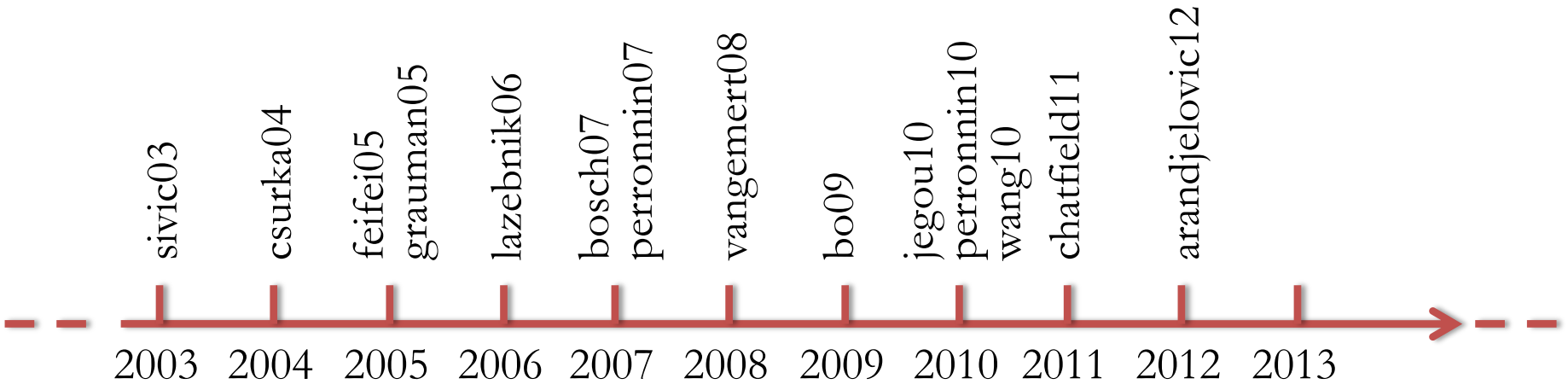
- Proposes the Locality-constrained Linear Coding (LLC) which is a fast adaptation of sparse coding theory.

TIMELINE



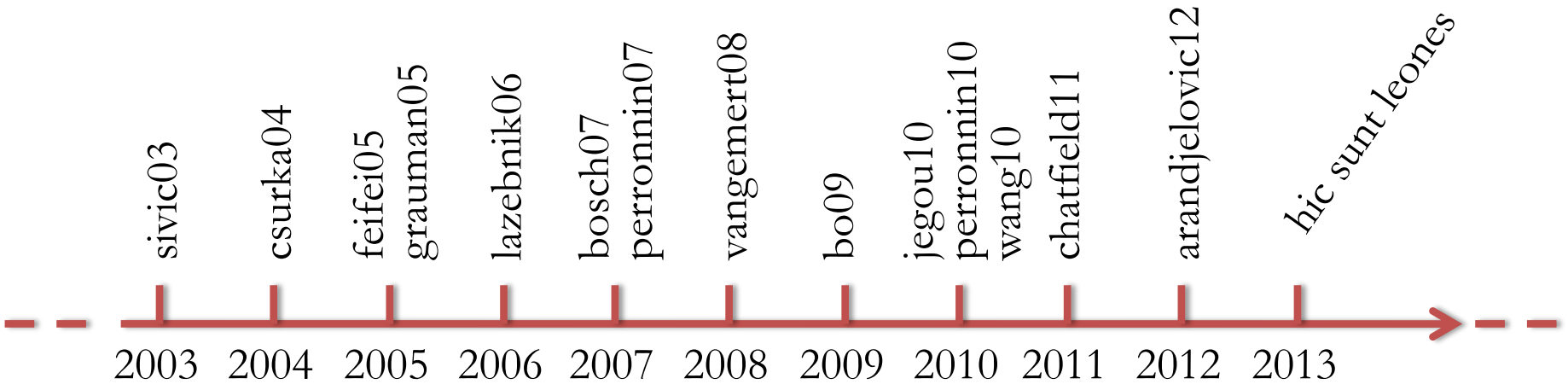
- A very important discussion on the details in implementing and comparing image classification algorithms.

TIMELINE



- Three things to be included in any image retrieval system.

TIMELINE



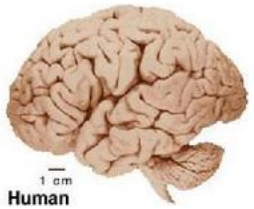
- The present and the future…

MEASURING PERFORMANCES IS ONE OF THE KEY ISSUES IN SCIENCE. REPEATABILITY IS IMPORTANT ALSO FOR OBJECT DETECTION AND CLASSIFICATION.

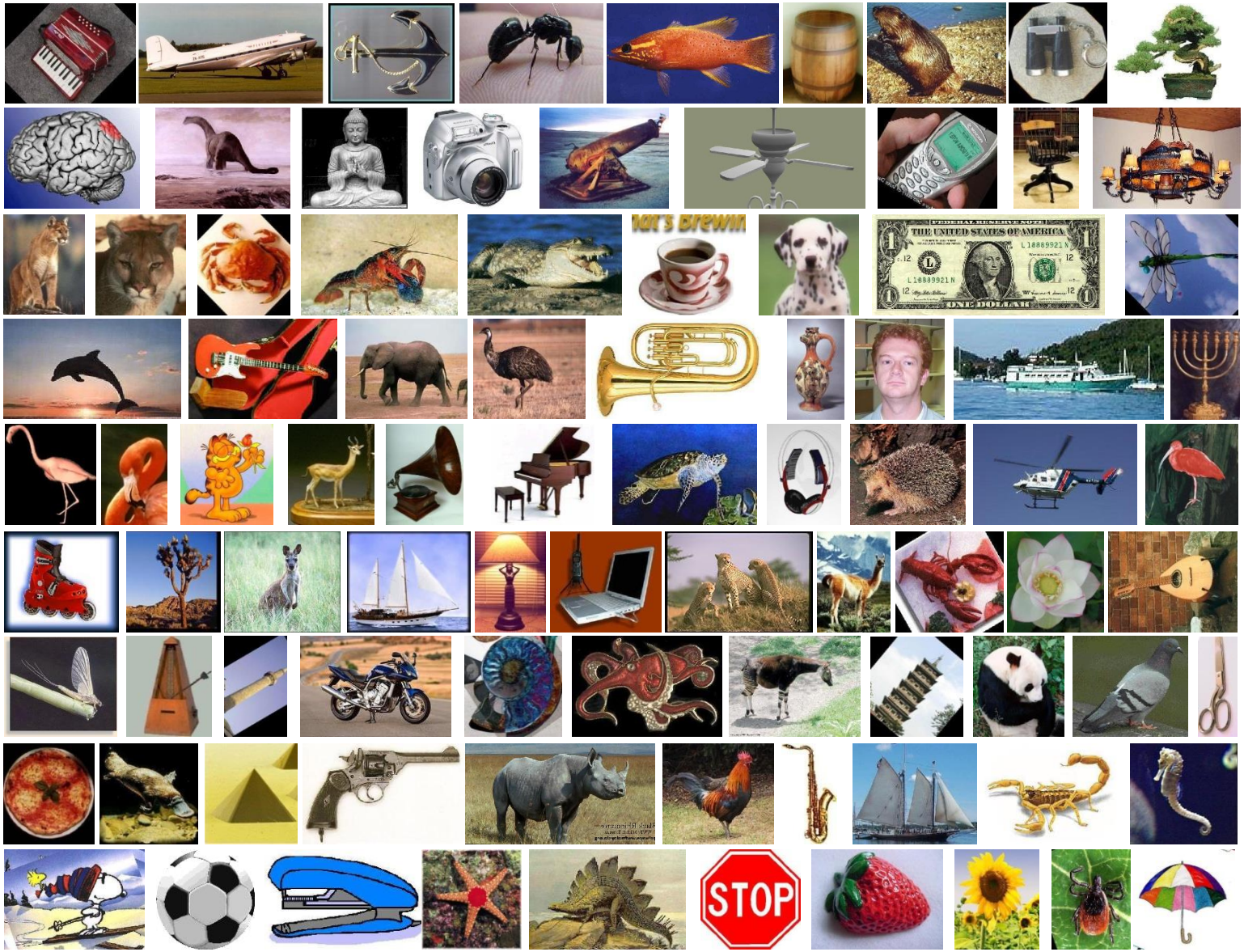
DATASETS

CALTECH-101

- Collected in **September 2003** by Fei-Fei Li et al.. It was created to allow researchers to make image classification results comparable to each other.
- It contains 9146 pictures of objects belonging to **101 categories**.
- About 40 to 800 images per category. Most categories have about 50 images. **At that time it was considered a very large dataset.**
- The size of each image is roughly 300 x 200 pixels.
- They suggest training and testing on fixed number of images and repeating the experiment with different random selections of pictures. Popular numbers of **training images** per class are **15** and **30**. For **testing** they suggest to randomly select at most **50 images** for each category
- Performance is measured by **mean recognition rate per class**.



CALTECH-101



CALTECH-101

- **Weaknesses:**
  - The **dataset is too clean**: images are very uniform in presentation, aligned from left to right, and usually not occluded.


  - **Limited number of categories**.
  - **Some categories contain few images**: certain categories are not represented as well as others, containing as few as 31 images. For example binocular (33), wild cat (34)
  - **Rotation and scaling artifacts**



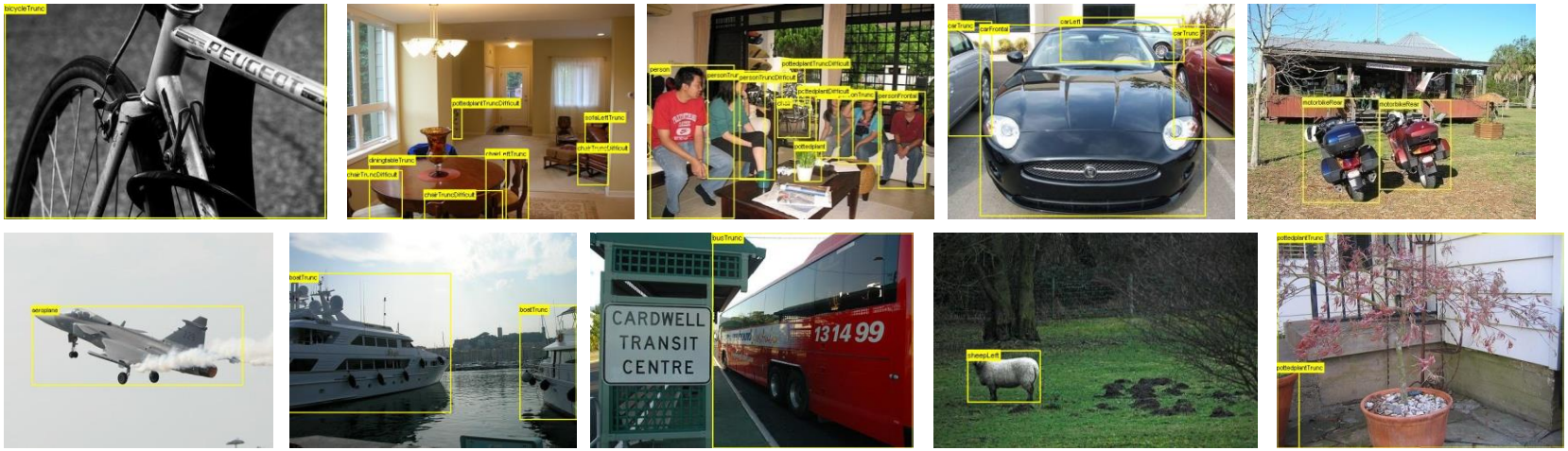
CALTECH-101 AND -256

- **Advantages:**
  - **Detailed Annotations**: overall image annotation, bounding box in which the object is located, and a detailed human-specified outline enclosing the object.
  - A **fair comparison** with a very large set of **previous methods**.
  - Now it is a small dataset, it can be used for **preliminary experiments**.

- **Caltech-256** is another image dataset created at the California Institute of technology in 2007, a successor to Caltech-101.
- It is intended to address some of the weaknesses inherent to Caltech-101.

PASCAL VISUAL OBJECT CHALLENGE

- The VOC2007 dataset consists of annotated consumer photographs collected from Flickr, not filtered for "quality"
- Complex scenes, scale, pose, lighting, occlusion, ...



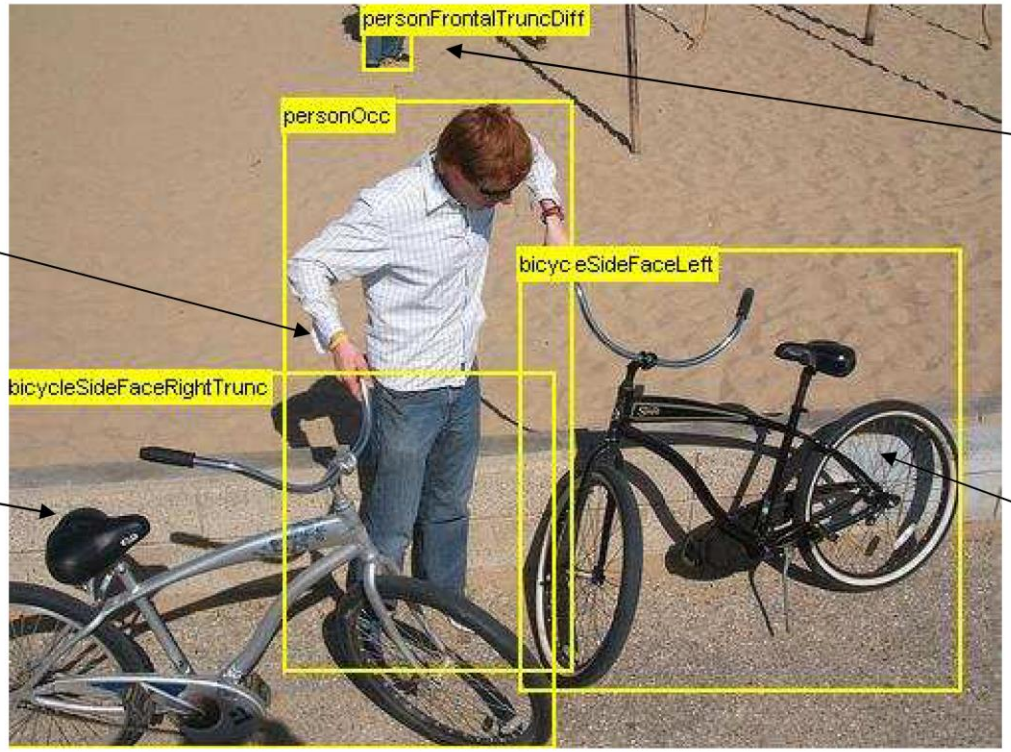
- It consists of 20 classes: aeroplane, bicycle, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, train, TV

ANNOTATION

- Complete annotation of all objects

Occluded
Object is significantly occluded within BB

Truncated
Object extends beyond BB

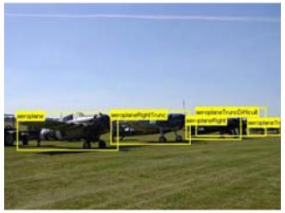


Difficult
Not scored in evaluation

Pose
Facing left

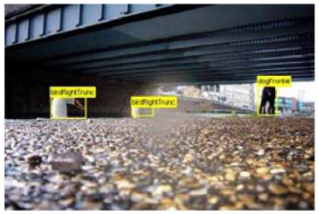
PASCAL VOC CLASSES

Aeroplane

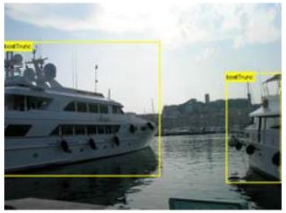


Bicycle

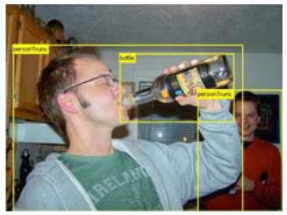
Bird

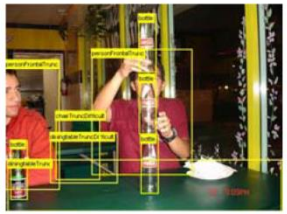


Boat



Bottle

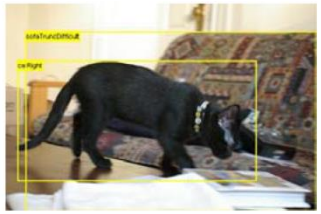




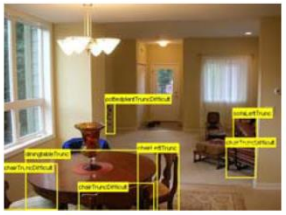
Bus

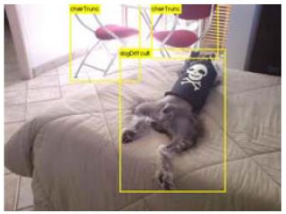
Car

Cat

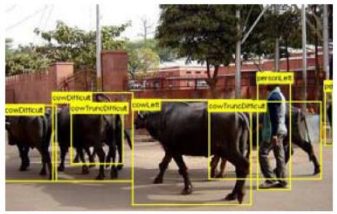


Chair



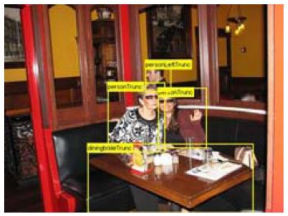


Cow

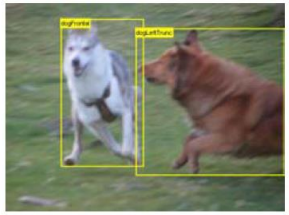


PASCAL VOC CLASSES

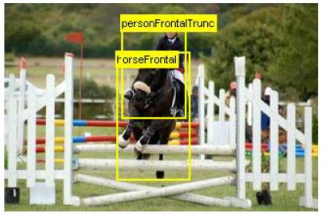
Dining Table

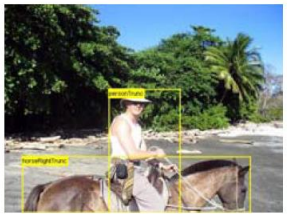


Dog

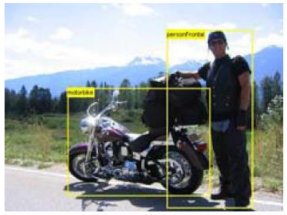


Horse

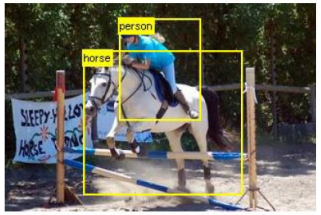




Motorbike

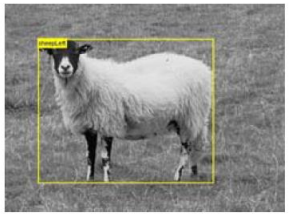


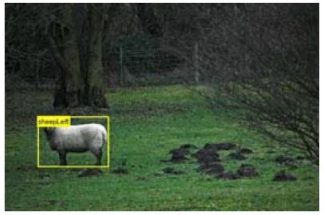
Person



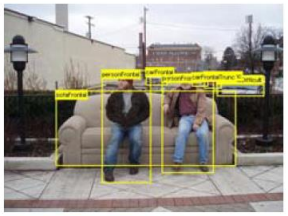
Potted Plant

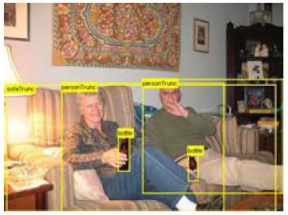
Sheep





Sofa





Train

TV/Monitor

STATISTICS OF VOC2007 DATASET

	train		val		trainval		test	
	img	**obj**	**img**	**obj**	**img**	**obj**	**img**	**obj**
Aeroplane	112	151	126	155	238	306	204	285
Bicycle	116	176	127	177	243	353	239	337
Bird	180	243	150	243	330	486	282	459
Boat	81	140	100	150	181	290	172	263
Bottle	139	253	105	252	244	505	212	469
Bus	97	115	89	114	186	229	174	213
Car	376	625	337	625	713	1,250	721	1,201
Cat	163	186	174	190	337	376	322	358
Chair	224	400	221	398	445	798	417	756
Cow	69	136	72	123	141	259	127	244
Dining table	97	103	103	112	200	215	190	206
Dog	203	253	218	257	421	510	418	489
Horse	139	182	148	180	287	362	274	348
Motorbike	120	167	125	172	245	339	222	325
Person	1,025	2,358	983	2,332	2,008	4,690	2,007	4,528
Potted plant	133	248	112	266	245	514	224	480
Sheep	48	130	48	127	96	257	97	242
Sofa	111	124	118	124	229	248	223	239
Train	127	145	134	152	261	297	259	282
Tv/monitor	128	166	128	158	256	324	229	308
Total	2,501	6,301	2,510	6,307	5,011	12,608	4,952	12,032

MEAN AVERAGE PRECISION

n	Image #	relevant
1	588	x
2	576	
3	589	x
4	342	
5	590	x
6	717	
7	984	
8	772	x
9	321	x
10	498	
11	113	
12	628	
13	772	
14	592	x

P=1/1=1

P=2/3=0.667

P=3/5=0.6

P=4/8=0.5

P=5/9=0.556

P=6/14=0.429

P= Precision@K

Average Precision: Average of the precision values at the points at which each relevant image is retrieved.
AP: (1 + 0.667 + 0.6 + 0.5 + 0.556 + 0.429)/6 = 0.625

Mean Average Precision: Average of the average precision value of the classes.

TECHNIQUES

FIRST APPROACHES

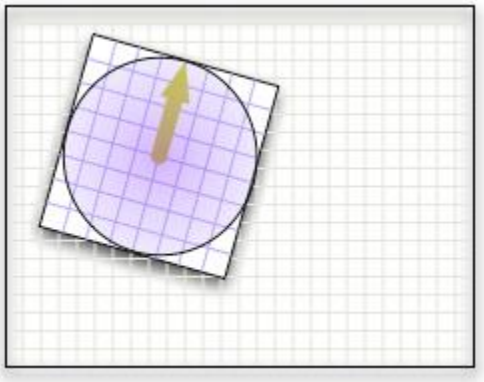
- The basic ideas for matching images used **global features**. This means that some kind of statistics were extracted from the image pixels and then used to build a fixed sized vector of numbers. 1 image → 1 vector.
- The classic example is a luminance or color histogram computed on the whole image.
- Few years before 2000 **local features** started to gain interest and proved to be a tremendous source of information for image description. They were basically built as a collection of derivatives computed in the neighborhood of an *interest point*.
- Two choices are to be made with local features:
  - how to chose interest points (*detector*)
  - how to describe the point (*descriptor*)
- With this approach 1 image → *n* vectors. It is important to note that *n* is image dependent, so comparing images requires to match *n* vectors with *m* vectors.

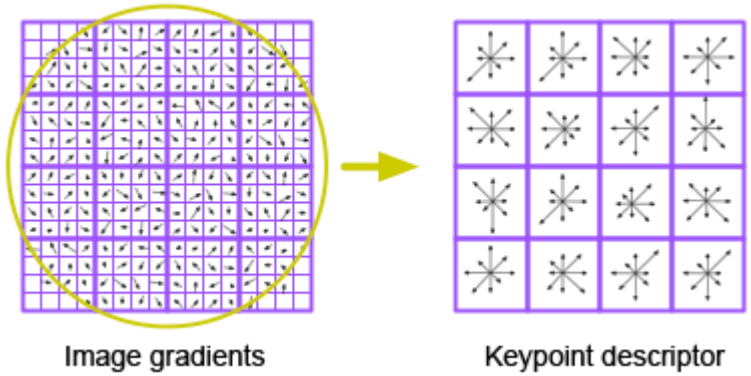
SCALE INVARIANT FEATURE TRANSFORM (SIFT)

- In 1999, D. Lowe introduced a revolutionary approach for both detection of interest points and for their description which was robust under:
  - Luminance change (based on the gradient analysis)
  - Scale change (due to scale-space theory)
  - Rotation (due to local reorientations with respect to the dominant orientation)
- A great attention was paid to the technique to be used in order to match these descriptors, because still they needed to be matched between different images, that is *n* to *m* matching.
- It is useful to consider how the descriptor is constructed in order to get an idea of what is happening around every point (also called keypoint).

SIFT DESCRIPTOR

- In order to derive a descriptor for the keypoint region we could sample intensities around the keypoint, but they are sensitive to lighting changes and to slight errors in x, y, θ. To make keypoint estimate more reliable, it is usually preferable to use a larger aggregation window.
- The SIFT descriptor is hence obtained from image gradients sampled over a 16×16 array of locations in the neighbourhood of the detected keypoint, using the level of the Gaussian pyramid at which the keypoint was detected. For each 4×4 region samples they are accumulated into a gradient orientation histogram with 8 sampled orientations.
- In total it results into a 4×4×8 = 128 dimensional feature vector





Images from: https://www.eecs.tu-berlin.de/fileadmin/fg144/Courses/06WS/scanning/Jonas/html/index.html

FROM GLOBAL TO LOCAL

- How do we simplify the *n* to *m* local feature matching?
- In 2003, Sivic and Zisserman started the trend that leveraged text retrieval approaches to efficiently treat image comparison.
- Text retrieval systems generally employ a number of standard steps:
  - The documents are first parsed into words
  - The words are represented by their stems
  - A stop list is used to reject very common words
  - The remaining words are then assigned a unique identifier
  - Each document is represented by a vector with components given by the frequency of occurrence of the words the document contains.
  - In addition the components are weighted in various ways (a common choice is "term frequency–inverse document frequency", tf-idf)
- Their approach was aimed to object matching in videos, so all the ideas were intermixed with details on how to do this efficiently and reliably in videos, making the paper quite complex.

FROM GLOBAL TO LOCAL

- In a very crude summary, they extracted from a frame two types of regions (Shape Adapted and Maximally Stable). These were represented by ellipses, providing affine invariance.
- Every region was described with a 128 dimensional SIFT descriptor.
- Visual "words" were created by clustering part of the descriptors with the K-means algorithm (with Mahalanobis distance and common covariance matrix Σ). 6k clusters were used for SA regions and 10k for MS regions.
- tf-idf was used to map every key-frame (1 per second) to a weighted vector
- The top 5% and bottom 10% frequent words were considered stop-words and discarded.
- Finally, inspired by Google approach to document ranking, spatial consistency was enforced by discarding regions which didn't have another match in their 15 nearest neighbors.

SEARCHING

The user selects an object to be searched:

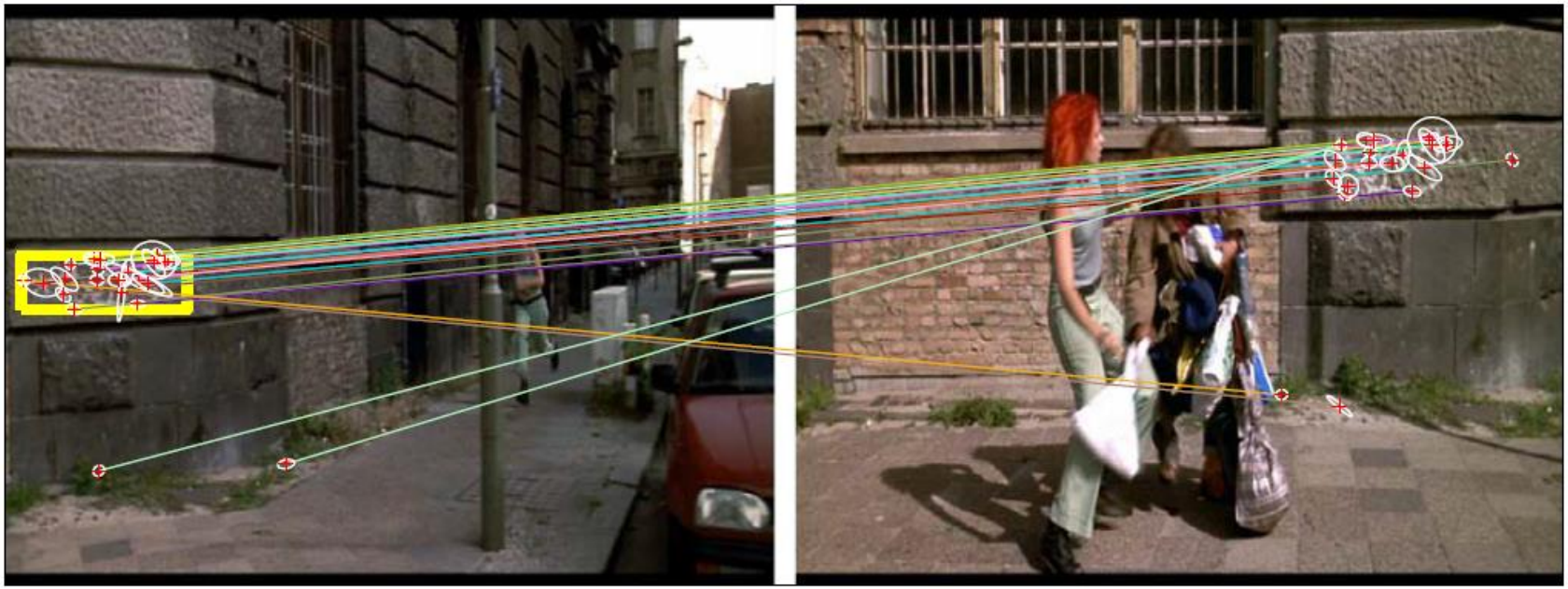
SEARCHING

Frames are sorted based on their weighted vectors similarity

Feature vectors which fall in the same cluster are a match.

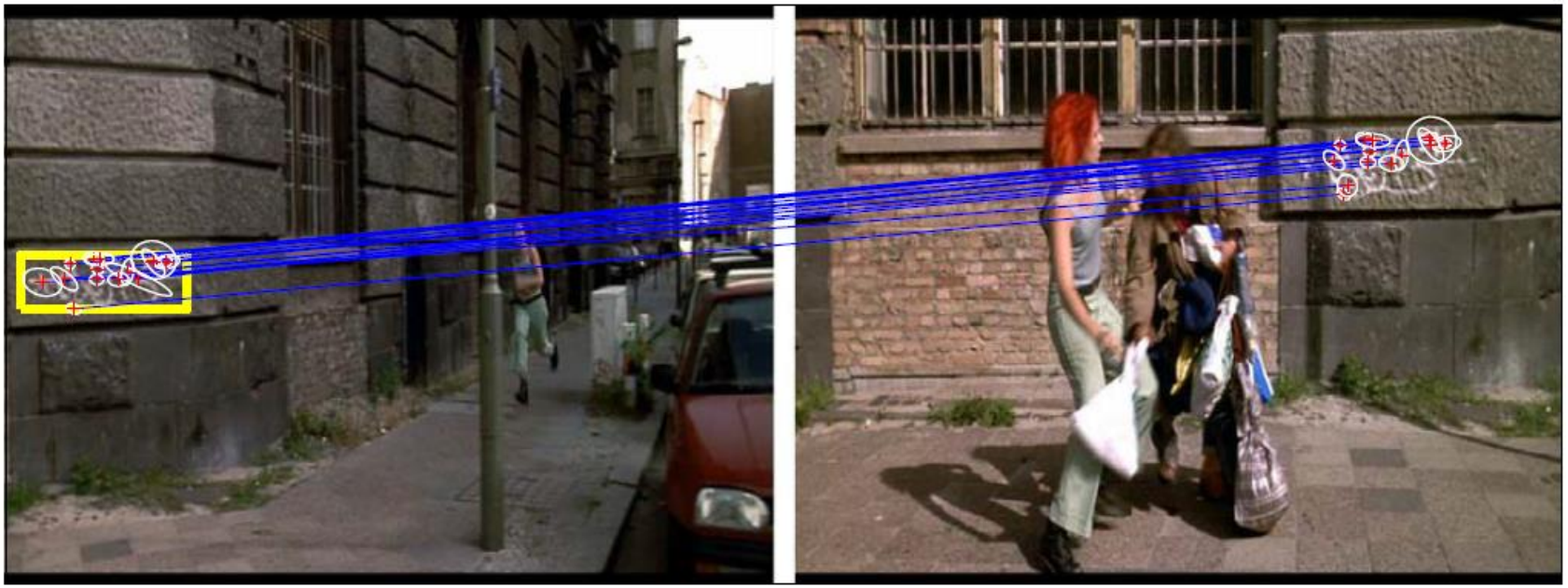
SEARCHING

Stop words are removed



SEARCHING

Spatial consistency is enforced.



MORE ON THIS

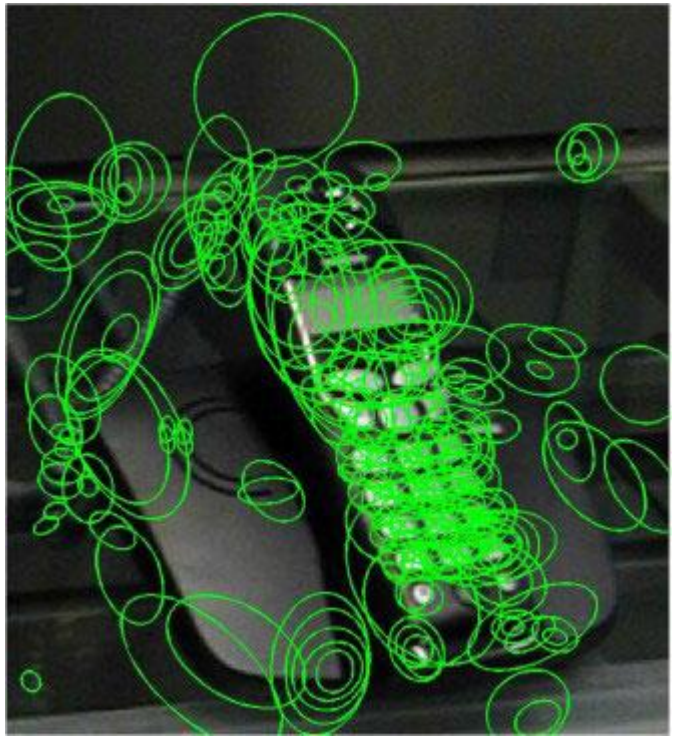
- Easy? Well, not really. The application to video, the use of two kind of regions with separate clustering, the use of stop words and spatial consistency, made it a complex article.
- In 2004, Csurka *et al.* basically applied the exact same idea to image categorization, but managed to present it clearly and only employed the basics. Moreover they picked a nice name, which helped the paper diffusion!
- They presented the *bag-of-keypoints* approach, which "corresponds to a histogram of the number of occurrences of particular image patterns in a given image". That's it.
- Their version of the text bag-of-words uses *Harris affine detector* for selecting interest points, then SIFT descriptors are computed on these. *k*-means is applied on a set of training images to form a visual vocabulary.
- An image can now be described by the *bag-of-keypoints*, which counts the number of patches assigned to each cluster.
- They use a naive Bayes classifier and a multiclass linear SVM to classify images into different visual categories.

BAG OF KEYPOINTS

Take some pictures from a training set

BAG OF KEYPOINTS

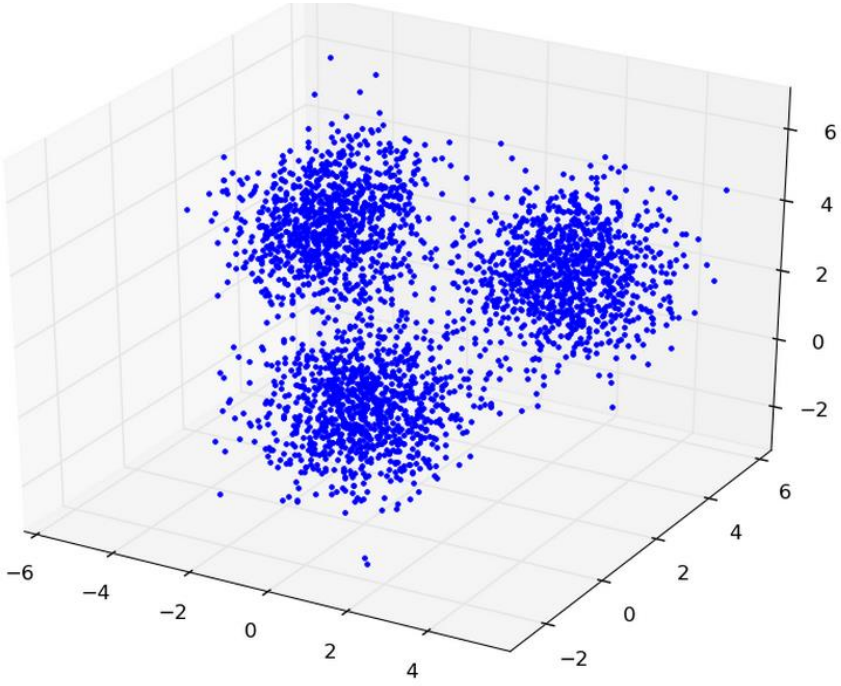
For every picture find some interest points and describe them with SIFT descriptors.

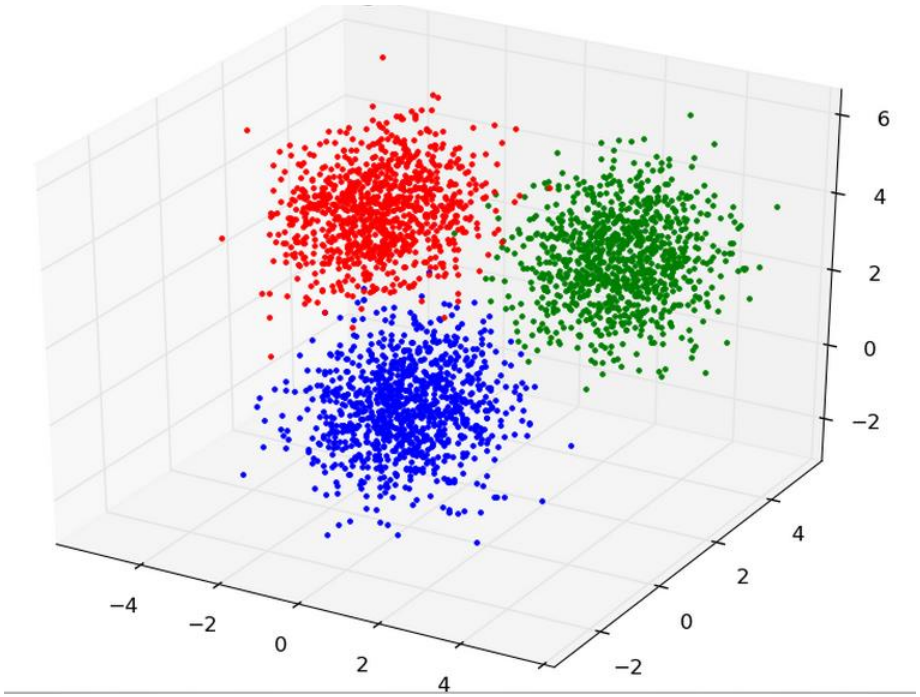


Forget **where** the SIFT descriptors are extracted and just keep their values in an **unordered** collection.

BAG OF KEYPOINTS

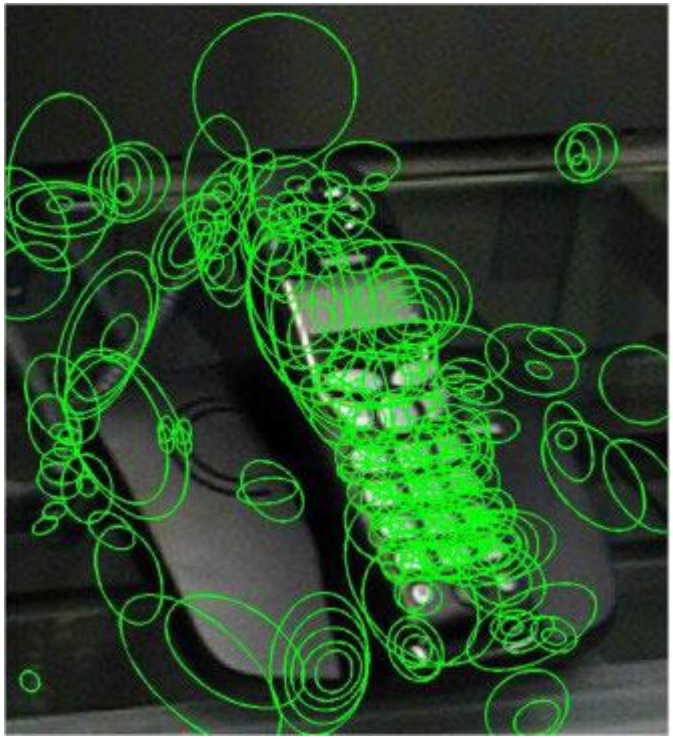
Cluster all SIFT descriptors, coming from all images, in k(=1000) clusters.

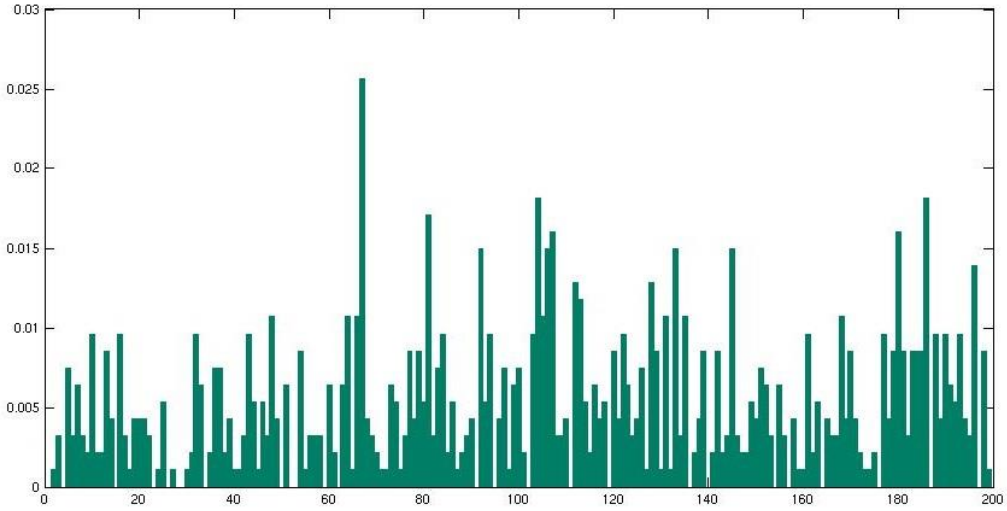




BAG OF KEYPOINTS

For every picture (during training or testing) substitute every SIFT descriptor with the nearest centroid. **Count** how many you get of every one.

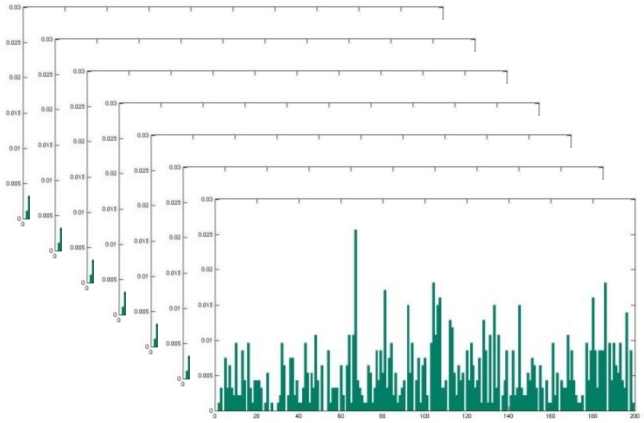


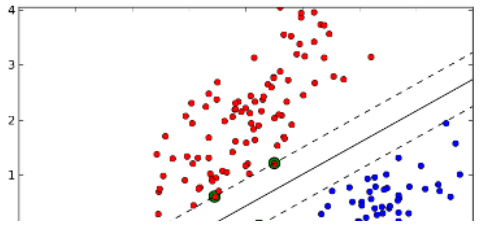


There is no relation between centroid 20 and 21 or 167, so x-axis is not ordered in any way. It just needs to be consistent between images.

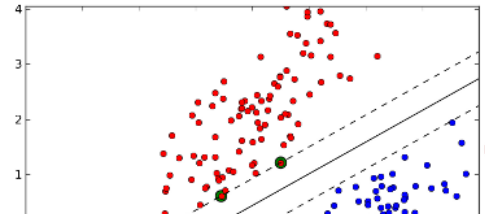
BAG OF KEYPOINTS

Learn a multiclass classifier from all training bag of keypoints histograms.

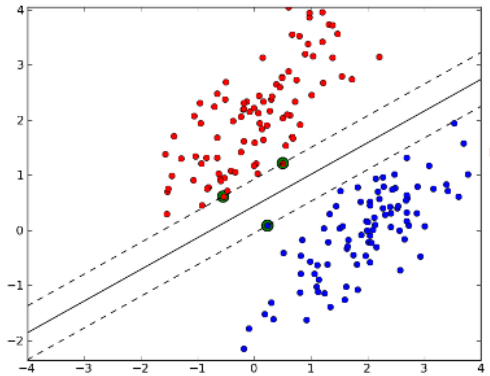




face



car

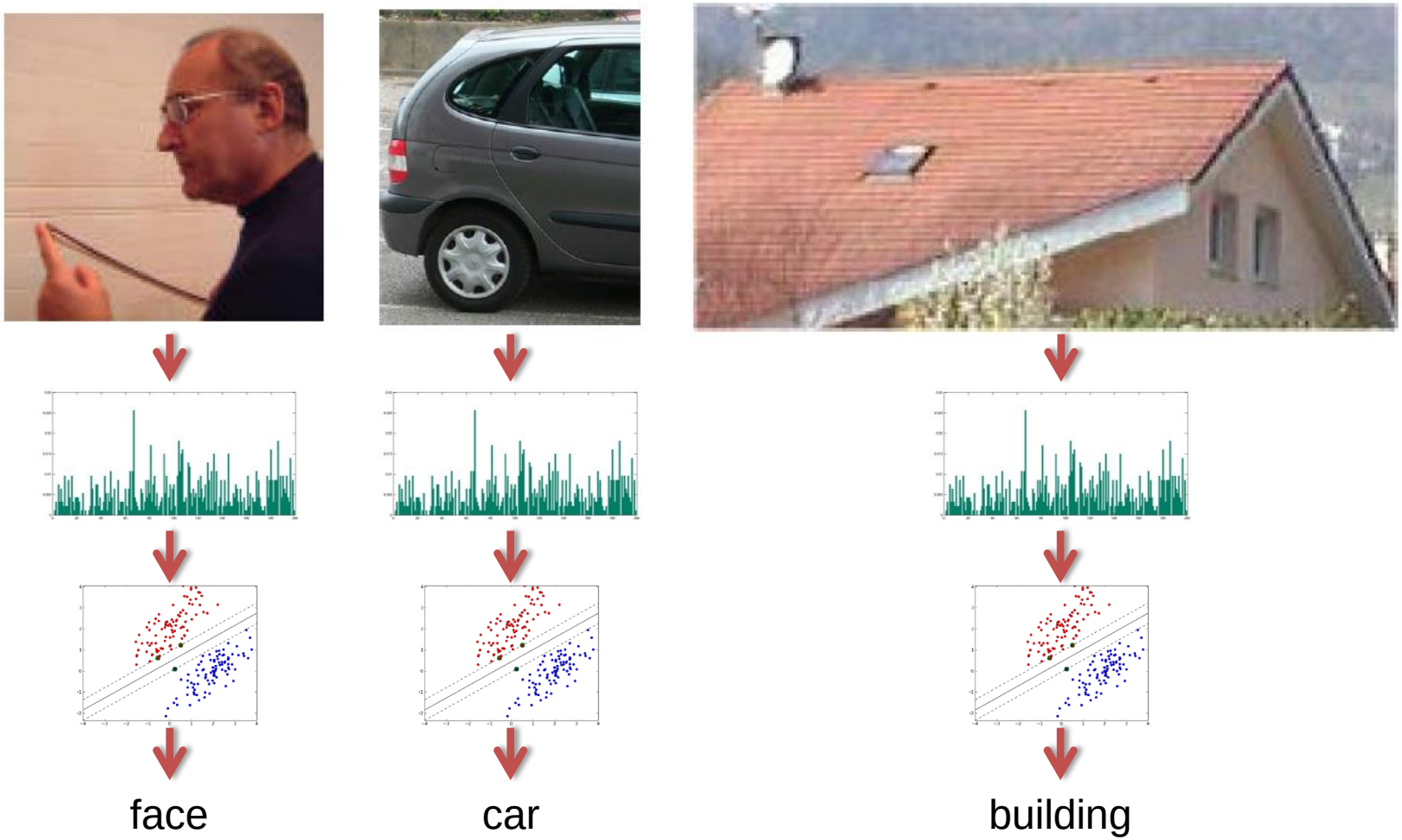


building

...

BAG OF KEYPOINTS

For every test image compute its bag of keypoints, feed it to the classifier and get the classification.



FURTHER ANALYSIS

- In 2005, Fei-Fei and Perona proposed a solution to avoid pre classification of images, using a theme discovering technique similar to Latent Dirichlet Allocation.
- This is not really related to our tutorial, but the paper is still very important for another point. They compare four different techniques for selecting interest points and describe each interest point with SIFT and with an embarrassing simple "normalized 11×11 pixel gray values".
- Interest points are extracted with:
  - Evenly Sampled Grid (10×10 and scale is randomly between 10 and 30)
  - Random Sampling (500 points with scale randomly between 10 and 30)
  - Kadir & Brady Saliency Detector (scale provided by detector)
  - SIFT Detector (scale provided by detector)
- This is an important step which really started to move the bag of visual words model away from a stripped down SIFT matching.

RESULTS

This is their observation:

Descriptor	Grid	Random	Saliency [4]	DoG [7]
11×11 Pixel	64.0%	47.5%	45.5%	N/A
128-dim Sift	65.2%	60.7%	53.1%	52.5%

Table 1. Performance comparison given different feature detectors and representations. The performance is quoted from the mean of the confusion table similar to that of Fig.7. SIFT representation seems to be in general more robust than the pixel grayvalue representation. The sliding grid, which yields the most number of patches, out performs all other detectors.

So: SIFT are better than raw pixels, but not absurdly so, and **the more patches, the better the results**! No fancy interest points, just a regular scanning over the whole image.

NOT BAG OF (VISUAL) WORDS

- At the same time (2005), Grauman and Darrel proposed a different approach to compute nearest neighbor matching in discriminative settings (SVMs): *pyramid match kernels*.
- We already mentioned that *n* to *m* matching is costly and everyone was trying to avoid it.
- Their proposal is exactly aimed to that purpose: define a kernel which provides a fixed length representation, but still allows to compute the nearest neighbors between local features.
- Each feature set is mapped to a multiresolution histogram that preserves the individual features' distinctness at the finest level.
- The histogram pyramids are then compared using a weighted histogram intersection computation, which is shown to define an implicit correspondence based on the finest resolution histogram cell where a matched pair first appears.

PYRAMID MATCH KERNELS

- Consider an input space X of sets of d-dimensional feature vectors:

$$X = \left\{x | x = \left\{\left[f_1^1, \dots, f_d^1\right], \dots, \left[f_1^{m_x}, \dots, f_d^{m_x}\right]\right\}\right\}$$

- The feature extraction function Ψ is defined as:

$$\Psi(x) = [H_0(x), \dots, H_L(x)]$$

- where $L = \lceil \log_2 D \rceil$, and $H_i(x)$ is a histogram vector formed over the data x using d-dimensional bins of side length 2^i. $H_i(x)$ has dimension $r_i = \left(\frac{D}{2^i}\right)^d$.
- In practice an unordered set of local features is mapped to a concatenation of histograms with bins that double in size at every level.
- The trick is now how to count matches from this sequence of histograms.

PYRAMID MATCH KERNELS

- How do we compare these histograms? With simple histogram intersection:

$$\mathcal{I}(\mathbf{A}, \mathbf{B}) = \sum_{j=1}^{r} \min\left(\mathbf{A}^{(j)}, \mathbf{B}^{(j)}\right)$$

- All matches at one level are included in those of the following one, so

$$N_i = \mathcal{I}(H_i(\boldsymbol{y}), H_i(\boldsymbol{z})) - \mathcal{I}(H_{i-1}(\boldsymbol{y}), H_{i-1}(\boldsymbol{z}))$$

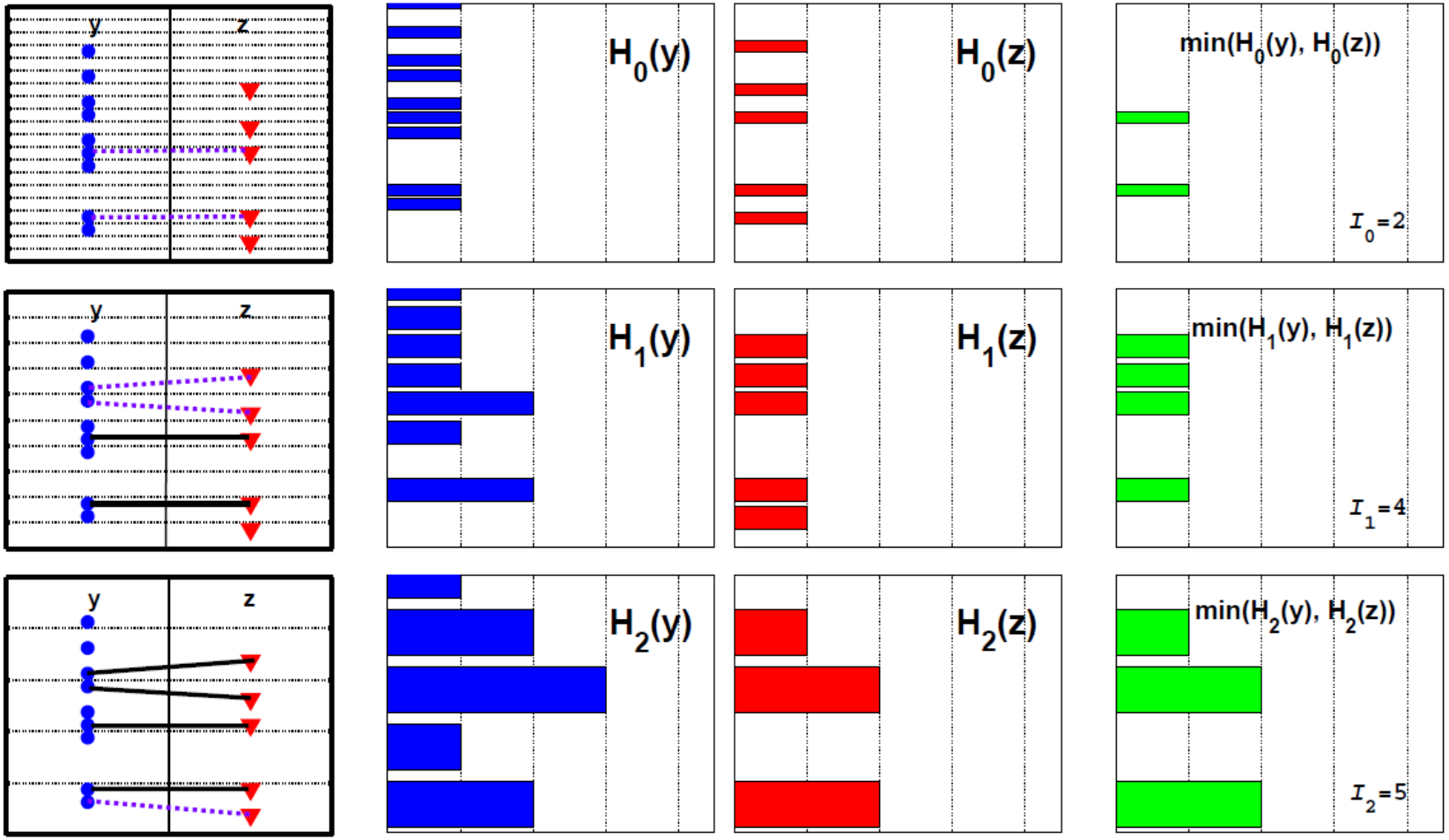
  provides the number of new matches found at level *i*.

- The final kernel is defined as:

$$K_\Delta\left(\Psi(\boldsymbol{y}), \Psi(\boldsymbol{z})\right) = \sum_{i=0}^{L} \frac{N_i}{2^i}$$

- The following figure provides an example.

PYRAMID MATCH KERNELS



(a) Point sets (b) Histogram pyramids (c) Intersections

PYRAMID MATCH KERNELS

- Perfect! We can now exactly compare images' descriptors, so no more bag of keypoints!
- Wait. Is it too good to be true. Let's look back at the histograms sizes:

$$r_i = \left(\frac{D}{2^i}\right)^d$$

- So at first level, $r_0 = D^d$. SIFT descriptors have $d = 128$, which immediately sounds bad. Suppose that we quantize every descriptor component to 8 bits, this gives a size of $(2^8)^{128}$.
- Of course it is not necessary to have a dense histogram representation, since this can be implemented sparsely with indexes and possibly hash tables.
- Still, all reported results limited the SIFT dimensions by applying PCA to it and drastically reducing its size (10 dimensions).
- Why is this paper important?

SPATIAL PYRAMID MATCHING

- The difficulties in correctly implementing the previous approach have probably limited its diffusion.
- Nevertheless, the idea of using multiple histograms and leveraging the fact that their intersection can be used to compute new matches by difference with the previous level, inspired, in 2006, Lazebnik *et al.* to explore an "orthogonal" approach: **perform pyramid matching in the two-dimensional image space**, and use traditional clustering techniques in feature space.
- Another interesting thing in the paper is that a different notation is used and a very nice observation is drawn.
- Let's say that level i is a grid with 2^i cells along each dimension, so that level 0 contains in a single cell every possible feature point, while level L is the least quantized grid. The number of matches at level i is given by histogram intersection $\mathcal{I}(H_i(\boldsymbol{x}), H_i(\boldsymbol{y}))$, which will be indicated as $\mathcal{I}^i$.
- All matches found at level i+1 are also present at level i.

SPATIAL PYRAMID MATCHING

- As before we can write the *pyramid match kernel*:

$$K_\Delta(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{I}^L + \sum_{i=0}^{L-1} \frac{1}{2^{L-i}} \left(\mathcal{I}^i - \mathcal{I}^{i+1}\right)$$

- A few simple steps allow us to change this to a different writing:

$$= \mathcal{I}^L + \sum_{i=0}^{L-1} \frac{1}{2^{L-i}} \mathcal{I}^i - \sum_{i=0}^{L-1} \frac{1}{2^{L-i}} \mathcal{I}^{i+1} =$$

$$= \sum_{i=0}^{L} \frac{1}{2^{L-i}} \mathcal{I}^i - \sum_{j=1}^{L} \frac{1}{2^{L-(j-1)}} \mathcal{I}^j =$$

$$= \frac{1}{2^L} \mathcal{I}^0 + \sum_{i=1}^{L} \frac{1}{2^{L-i}} \mathcal{I}^i - \frac{1}{2} \sum_{i=1}^{L} \frac{1}{2^{L-i}} \mathcal{I}^i =$$

$$= \frac{1}{2^L} \mathcal{I}^0 + \sum_{i=1}^{L} \frac{1}{2^{L-i+1}} \mathcal{I}^i$$

SPATIAL PYRAMID MATCHING

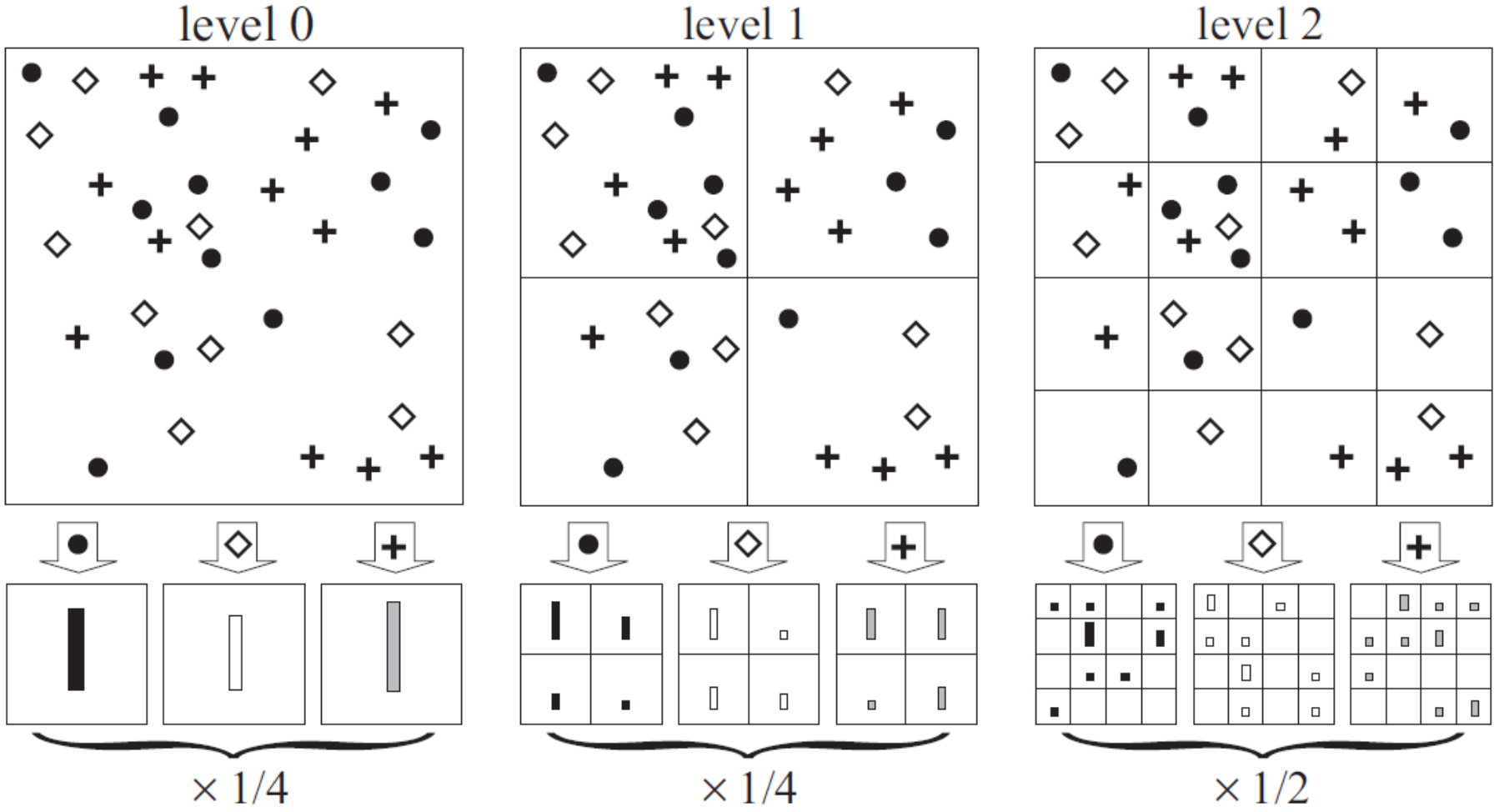
- What did we get?

$$K_\Delta(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2^L}\mathcal{I}^0 + \sum_{i=1}^{L} \frac{1}{2^{L-i+1}}\mathcal{I}^i$$

- With a simple change, it is clear that every histogram intersection is weighted differently based on the level only, so it is possible to weight the histograms beforehand and compute **a single long histogram intersection** on the concatenation of all histograms.
- What is the result? The *spatial pyramid matching* is simply a bag-of-keypoints performed on separate spatial regions.
- The regions BOW vectors are then weighted and concatenated, effectively making a longer single image descriptor which can be compared with others with histogram intersection.

SPATIAL PYRAMID MATCHING

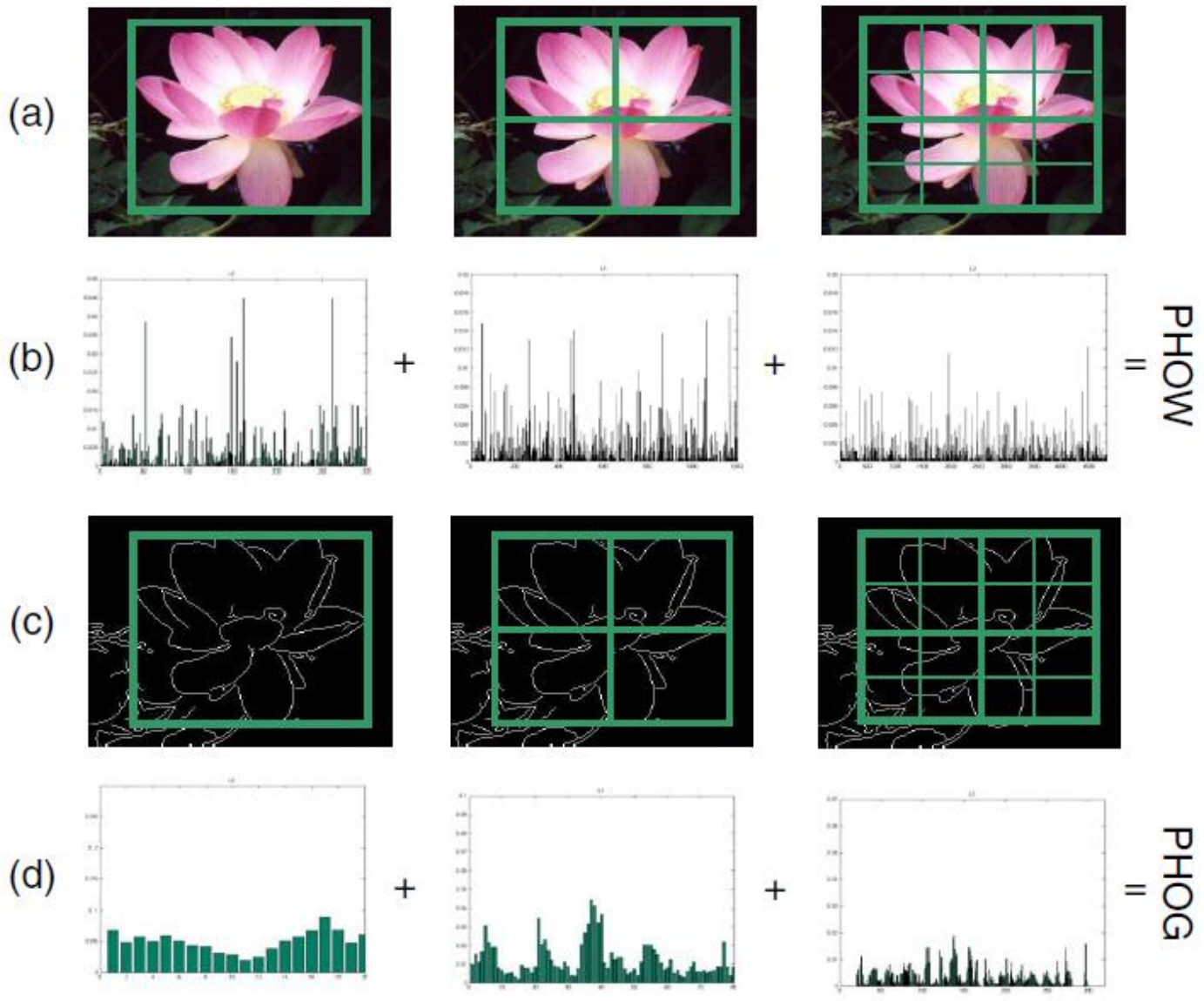
- This is a very simple way of reintroducing spatial information in the BOW representation.



PACKING MORE INFORMATION IN THE PYRAMID

- The spatial pyramid proved to be a strong source of information and most of the subsequent papers started to leverage on it.
- In their ICCV 2007 paper, Bosch *et al.* packed together most of the observation we touched up to now with their ECCV 2006 paper in two descriptors, which enabled further improvement of the results.
- Their descriptors are called *Pyramid Histogram Of visual Words* (PHOW), and *Pyramid Histogram of Oriented Gradients* (PHOG).
- PHOW: SIFT descriptors are computed at points on a regular grid with spacing *M* pixels. At each grid point the descriptors are computed over **four** circular support patches with different radii, consequently each point is represented by four SIFT descriptors. The descriptors are clustered (without any regard to scale) with K-means.
- PHOG: Each bin in the histogram represents the number of edges that have orientations within a certain angular range. This representation can be compared to the traditional "bag of (visual) words", where here each visual word is a quantization on edge orientations.

PHOW AND PHOG

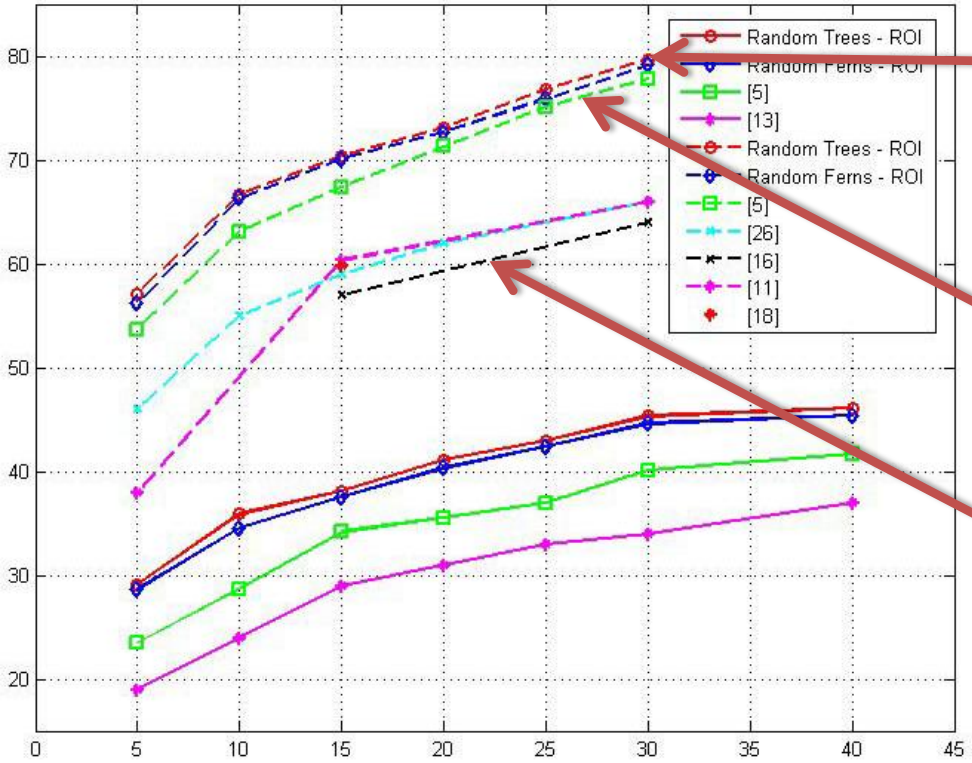


SOME DETAILS

- In their experiments the authors used both grey level and color cues. SIFT descriptors are computed at points on a regular grid with spacing $M = 10$.
- At each grid point the descriptors are computed over circular support patches with radii $r = 4, 8, 12$ and 16 pixels.
- For color the SIFT descriptors are computed separately for each HSV component.
- The K-means clustering is performed over 5 training images per category selected at random. A vocabulary of $V = 300$ words is used.
- Edge contours are extracted using the Canny edge detector. The orientation gradients are then computed using a 3×3 Sobel mask.
- Two shape descriptors are used: one with orientations in the range [0, 180] and the other with range [0, 360] using all orientation. The histogram descriptor is discretized into K = 20 and K = 40 bins respectively.

ANALYZING RESULTS

- It's not simple to look at numbers and draw a conclusion on the method.
- By looking at a result graph on Caltech-101 and Caltech-256 in the mentioned paper, it looks as if the original work of Lazebnik *et al.* was totally behind:



Bosch 2007 (ICCV)

Bosch 2007 (CIVR) - no ROI, regular SVM

Lazebnik 2006 - single scale, no color, no texture, SVM.

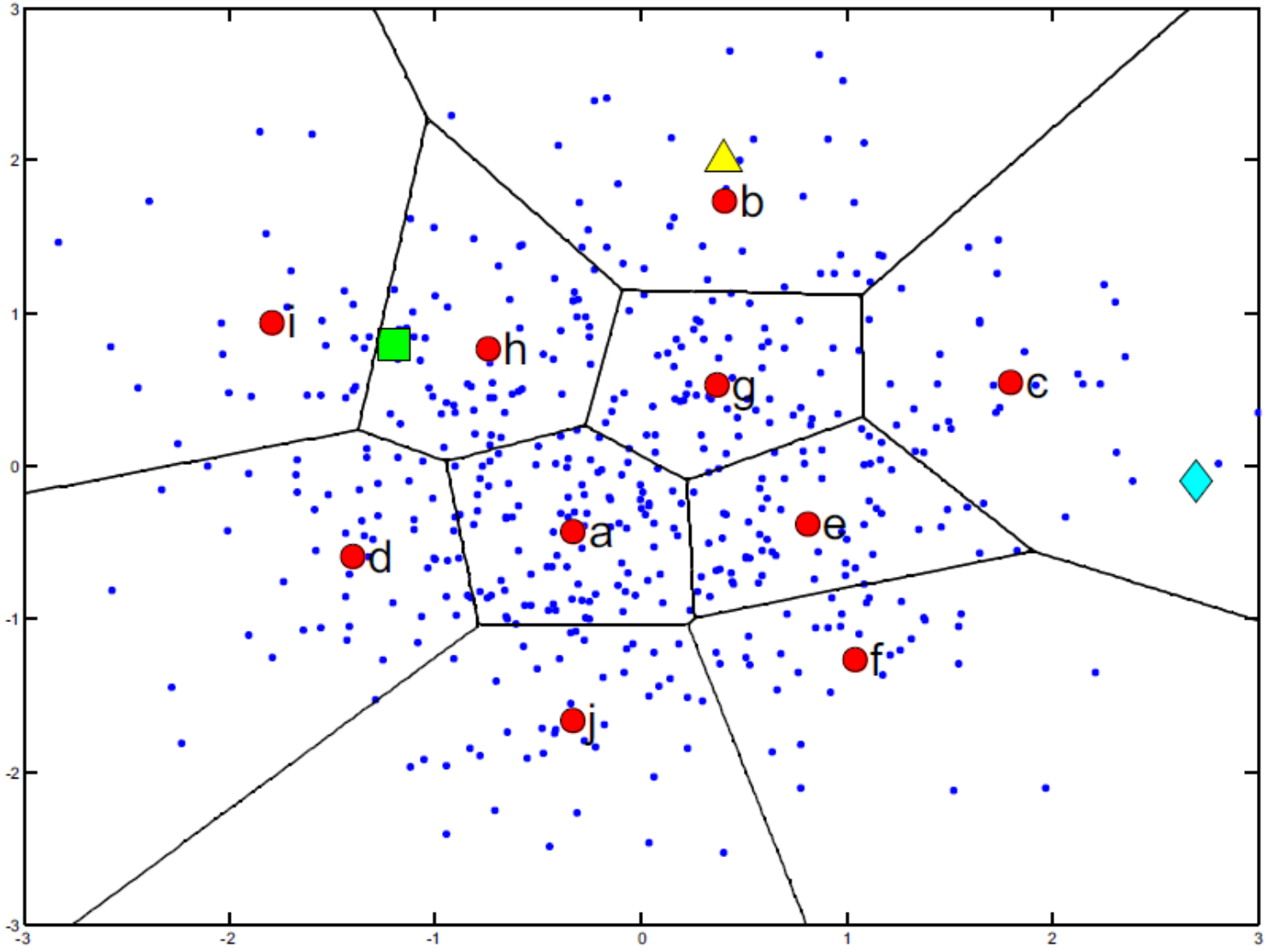
ANALYZING RESULTS

- There is a number of elements which must be considered:
  - how many features?
  - which features?
  - which preprocessing?
  - which classifier(s)?
- There is a number of conclusions which can be drawn here:
  - **Using multiple scales for every pixel in a grid is really beneficial**
  - Using color SIFTs or histograms of gradients makes sense, but this could have been done also with the regular spatial pyramid (so we should factor out it when measuring improvements)
  - Another element is the introduction of a ROI detector, which works great on Caltech datasets, but is not easy to extend to other testing environments.

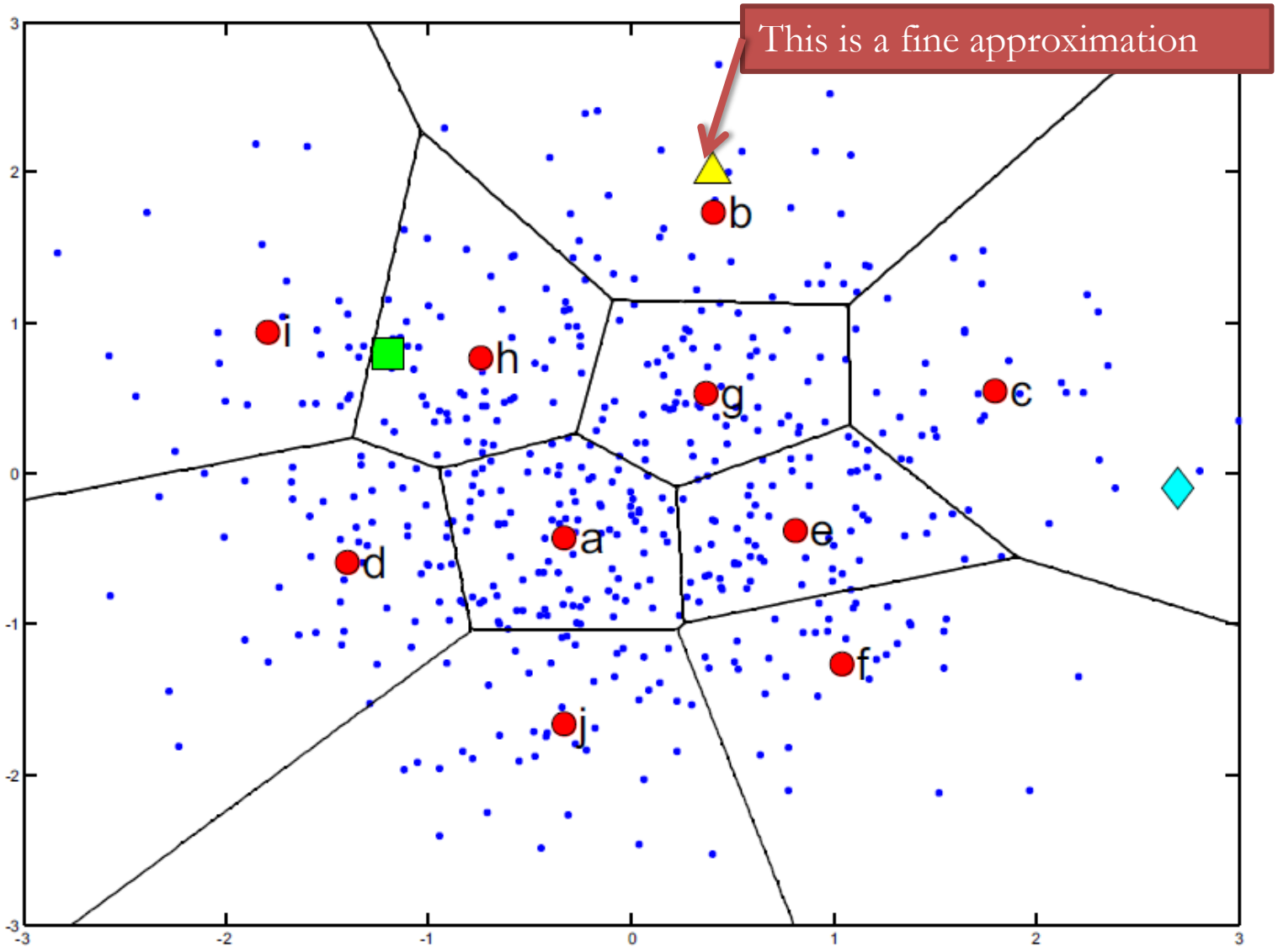
LOOK IN ANOTHER DIRECTION

- Up to this point we concentrated on improving features and with PHOW and PHOG we are given a very powerful tool.
- Look now at the basic model we are using: take a point in feature space and record a (partially) wrong representation of it.
- In fact we are (hard) quantizing features in bins, totally forgetting were they are originally and disregarding what their real position was.
- Many papers proposed solution for this but we believe that a very clear and nice interpretation of the problem was provided by van Gemert *et al.* at ECCV 2008.
- Just consider a 2D toy problem and apply *k*-means to that.

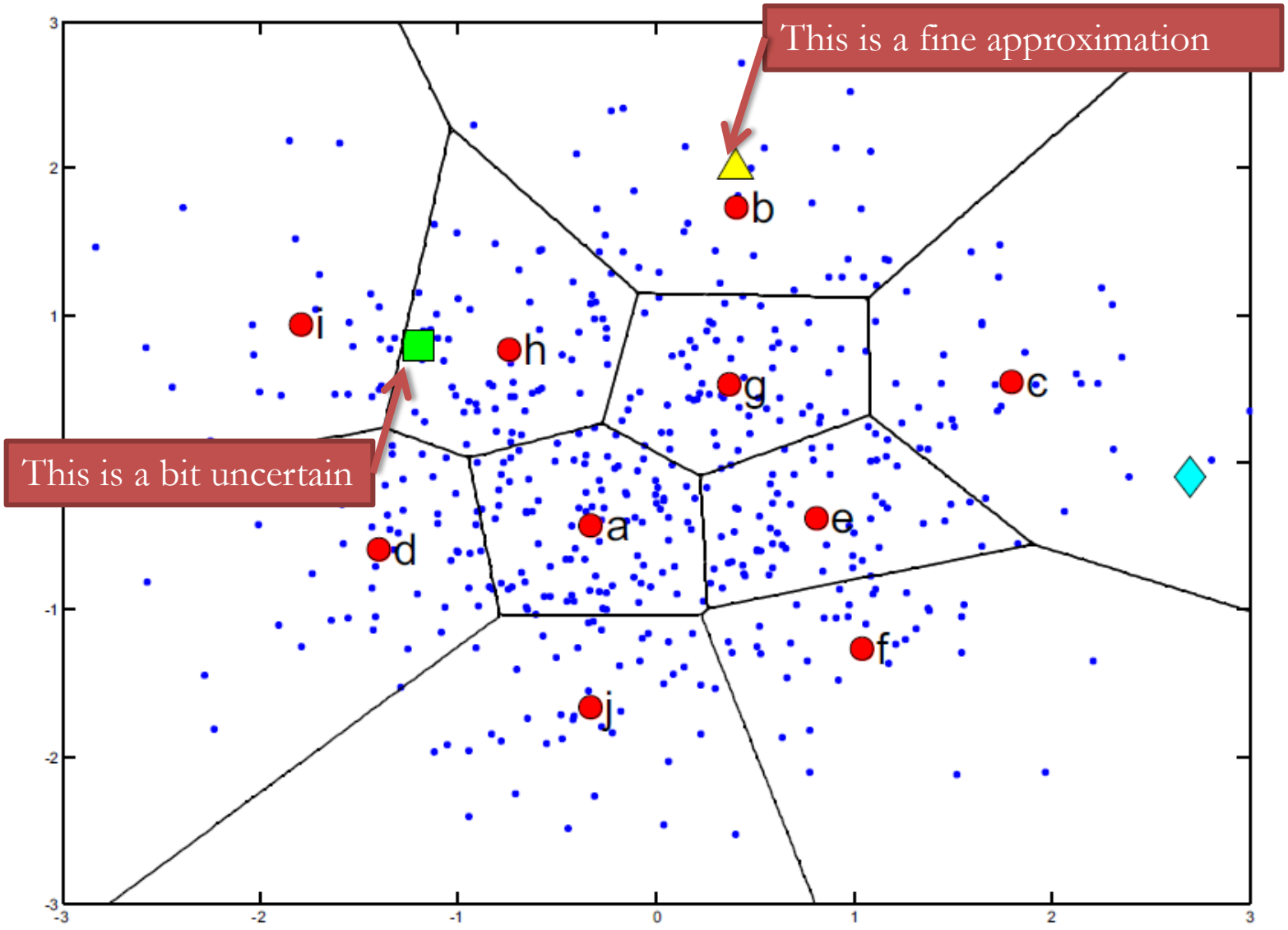
BOVW PROBLEMS



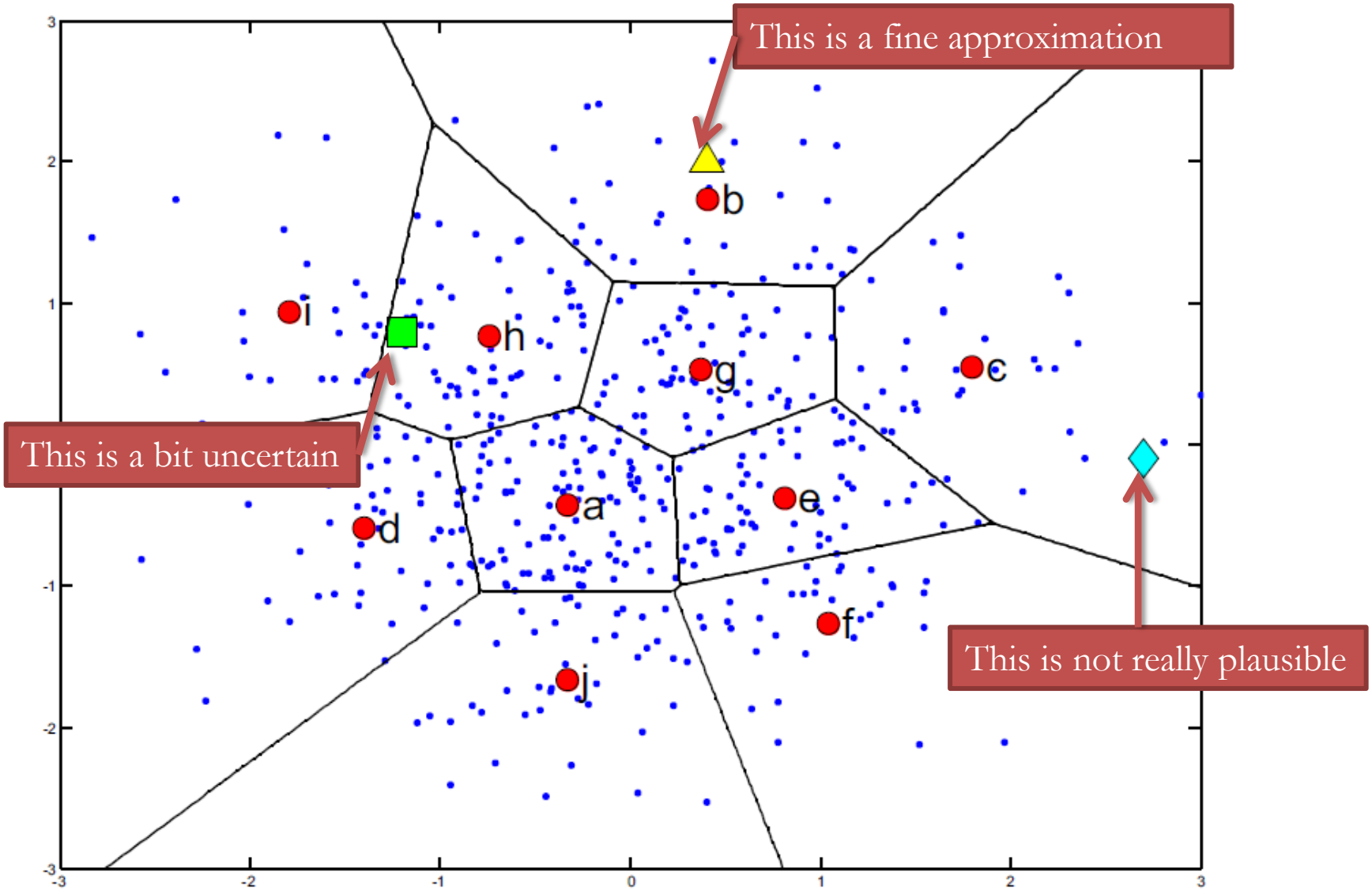
BOVW PROBLEMS



BOVW PROBLEMS



BOVW PROBLEMS



KERNEL CODEBOOK

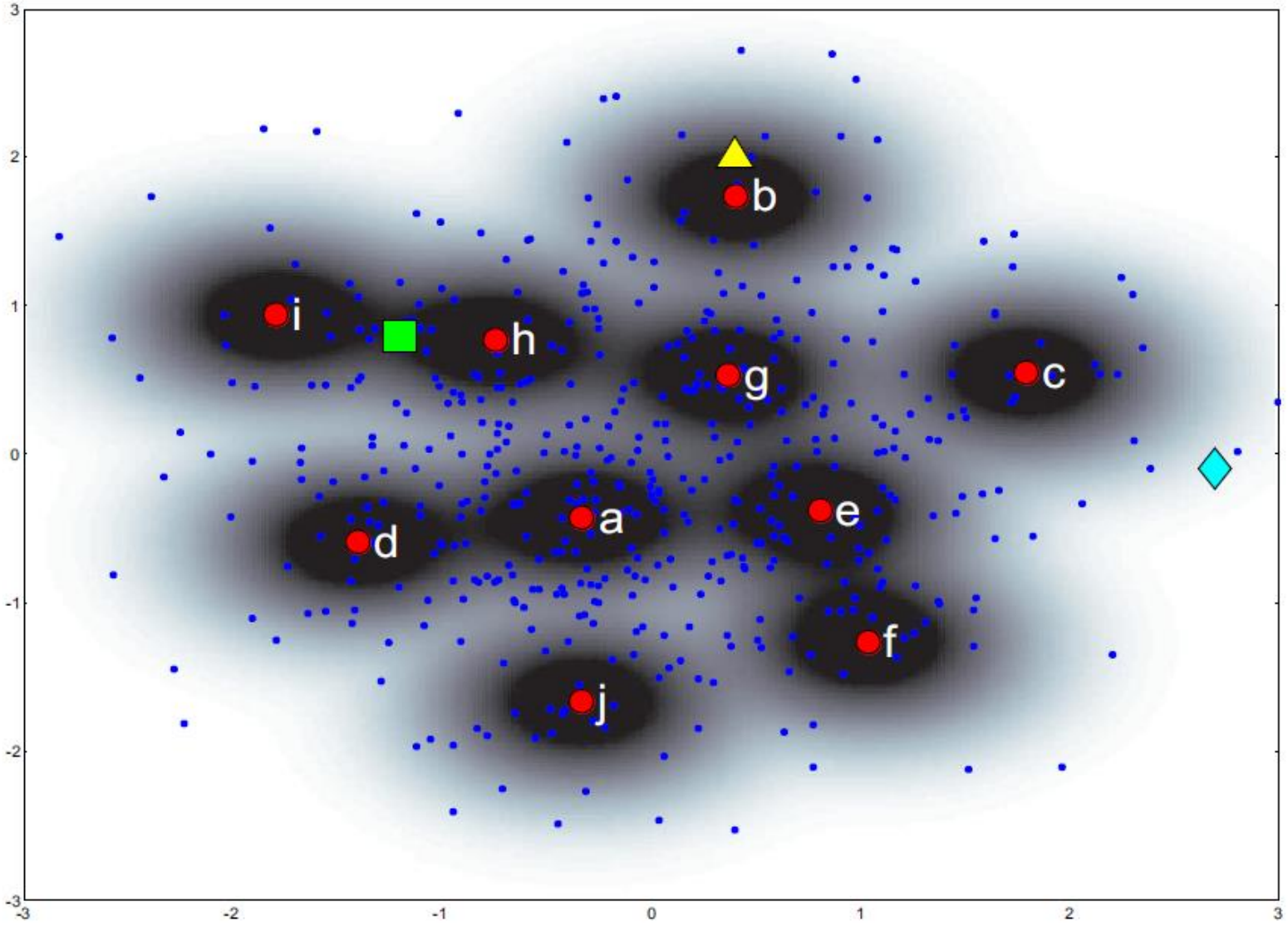
- van Gemert *et al.* start from the classical BOVW model calling it the traditional CodeBook, and defining it as:

$$CB(w) = \frac{1}{n}\sum_{i=1}^{n}\begin{cases}1 & \text{if } w = \underset{v \in V}{\operatorname{argmin}}\left(D(v, r_i)\right) \\ 0 & \text{otherwise}\end{cases}$$

- n is the number of descriptors, r_i is the i-th descriptor, V is the dictionary and $D(\cdot)$ is the distance function.
- Basically, every bin gets a 1 if descriptor r_i is assigned (*nearest*) to its centroid w.
- We can instead place a Gaussian kernel over every centroid and thus measure **how much** descriptor r_i is similar (*near*) to centroid w.
- The Gaussian kernel is:

$$K_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2}\frac{x^2}{\sigma^2}\right)$$

KERNEL CODEBOOK



KERNEL CODEBOOK

- The kernel codebook thus becomes:

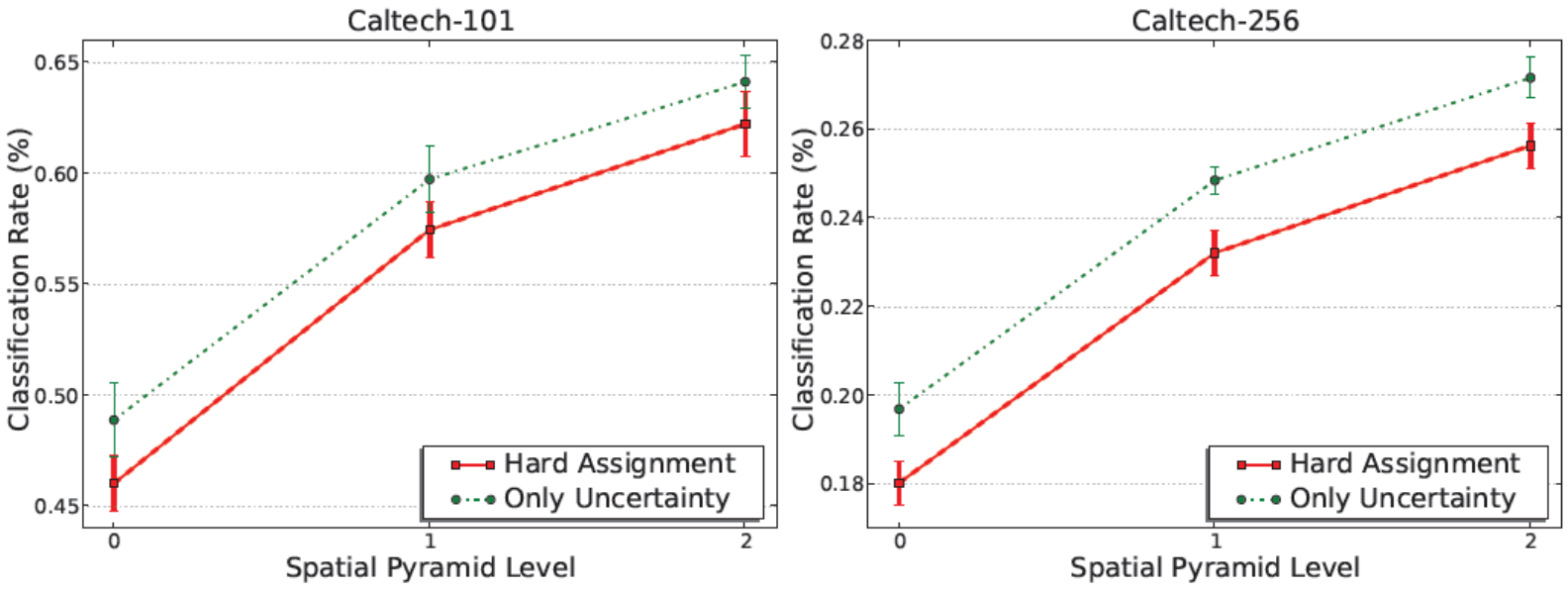
$$KCB(w) = \frac{1}{n}\sum_{i=1}^{n} K_\sigma\big(D(w, r_i)\big)$$

- So every descriptor contributes some information to every bin (of course depending on σ). In the paper, with a different motivation and analysis, the authors realize that different descriptors contribute differently to the total histogram, so they introduce the *Codeword uncertainty* which is a **normalized version** of the kernel codebook:

$$UNC(w) = \frac{1}{n}\sum_{i=1}^{n} \frac{K_\sigma\big(D(w, r_i)\big)}{\sum_{v \in V} K_\sigma\big(D(v, r_i)\big)}$$

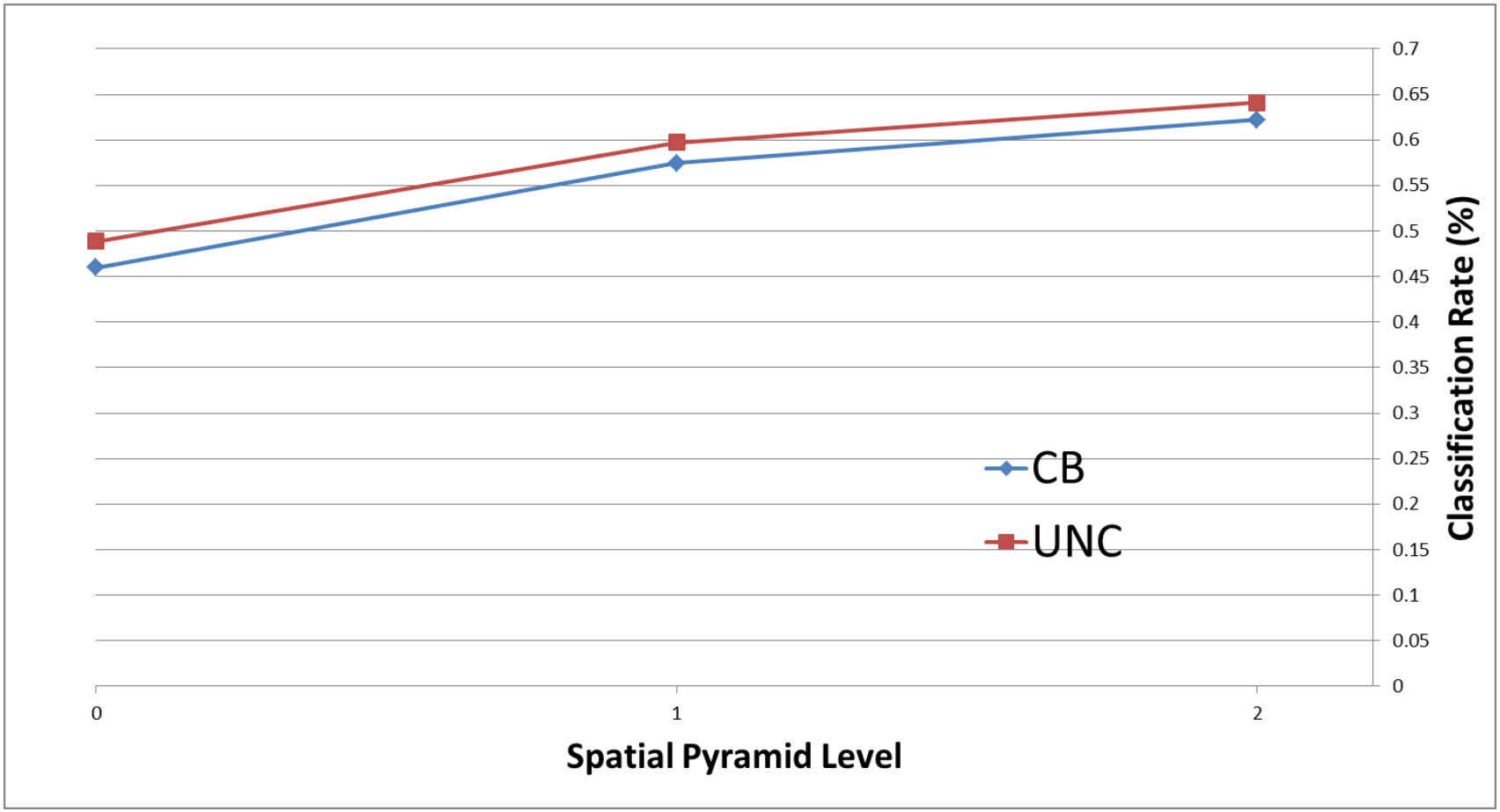
- In this case all descriptors contribute a constant amount of probability mass to all relevant codewords.

RESULTS



- So, soft assignment provides an improvement over the classical BOVW approach.
- But, wait, this chart doesn't start from 0! Let's show it with the correct scale.

RESULTS



- The improvement is there, but it is not so impressive.
- My point: a chart should start from 0. No matter what.

EFFICIENT MATCH KERNELS

- BOW is a way to measure the similarity between two images represented as sets of local features.
- As stated before, a better approach is to define kernels over sets of local features such as Pyramid Match Kernel.
- An interesting kernel is **Sum Match Kernel** which is obtained by adding local kernels over all combinations of local features from two different sets.
- However, the evaluation of this kernel requires an high computation cost and it is **impractical for large datasets**.
- In 2009, Bo and Sminchisescu proposed **Efficient Match Kernels** (EMK) over two sets of local features, which map local features to a low dimensional feature space, average the resulting feature vectors to form a set-level feature, then apply a linear classifier.
- They showed that EMK achieved the current state of the art in Caltech-101 and Caltech-256.

BOW IN MATCH KERNELS FORMALISM

- A set of S-dimensional local descriptors

$$X = [x_1, x_2, \dots, x_N] \in R^{S \times N}$$

- Codebook with D visual words

$$B = [b_1, b_2, \dots, b_M] \in R^{S \times M}$$

- In BOW, each local feature is quantized into a D dimensional binary vector

$$\mu(x) = [\mu_1(x), \mu_2(x), \dots, \mu_D(x)]$$

$$\mu_i(x) = \begin{cases} 1, & x \in R(b_i) \\ 0, & otherwise \end{cases}$$

Where $R(b_i) = \{x: \|x - b_i\| < \|x - b\|, \forall b \in B\}$

- When using BOW features with a linear classifier, the resulting kernel function is:

$$K_B(X, Y) = \frac{1}{|X||Y|} \sum_{x \in X} \sum_{y \in Y} \mu(x)^T \mu(y) = \frac{1}{|X||Y|} \sum_{x \in X} \sum_{y \in Y} \delta(x, y)$$

with

$$\delta(x, y) = \begin{cases} 1 & x, y \subset R(b_i), \exists i \epsilon \{1 \dots D\} \\ 0 & \text{otherwise} \end{cases}$$

EFFICIENT MATCH KERNELS

- This type of quantization can be too coarse when measuring the similarity of two set of local features. $\delta(x, y)$ can be replaced with a continuous kernel function as:

$$K_S(X, Y) = \frac{1}{|X||Y|} \sum_{x \in X} \sum_{y \in Y} k(x, y)$$

- Called **Normalized Sum Match Kernel.**
- **Problem:** high computation cost – $O(|X||Y|)$
- If the $k(x, y) = \phi(x)^T \phi(y)$ is a **finite dimensional kernel** (i.e. $\phi(\cdot)$ is finite dimensional), the **match kernel** can be simplified as:

$$K_S(X, Y) = \bar{\phi}(X)^T \bar{\phi}(Y) = \frac{1}{|X|} \sum_{x \in X} \phi(x)^T \frac{1}{|Y|} \sum_{y \in Y} \phi(y)$$

- Such kernel K_S is called an **Efficient Match Kernel (EMK).**
- **Since $\bar{\phi}(X)$ is finite and can be computed explicitly, it is possible to extract feature vectors on the set X, then apply linear classifier.**

EFFICIENT MATCH KERNELS

- Now the problem is to define **an appropriated kernel**.
- **The idea** is to project the high dimensional feature vectors $\phi(x)$ induced by the kernel $k(x, y) = \phi(x)^T \phi(y)$ to a low dimensional space spanned by D basis vectors.
- Given $\{\phi(z_i)\}_{i=1}^{D}$, a set of basis vectors z_i, $\phi(x)$ can be approximated as Hv_x by solving the following problem:

$$\overline{v_x} = \underset{v_x}{\operatorname{argmin}} \|\phi(x) - Hv_x\|^2$$

where $H = [\phi(z_1), \ldots., \phi(z_D)]$ and v_x are the projection coefficients.

- The analytic solution of this convex quadratic problem is:

$$\overline{v_x} = \left(\mathrm{H}^{\mathrm{T}}\mathrm{H}\right)^{-1}\left(\mathrm{H}^{\mathrm{T}}\phi(x)\right)$$

EFFICIENT MATCH KERNELS (DETAILS)

- The approximated kernel is:

$$k(x,y) = \phi(x)^T \phi(y) \cong [H\overline{v_x}]^T \left[H\overline{v_y}\right]$$

substituting $\overline{v_x}$ we obtain:

$$\begin{aligned} k(x,y) &= [H(H^TH)^{-1}(H^T\phi(x))]^T[H(H^TH)^{-1}(H^T\phi(y))] = \\ &= (H^T\phi(x))^T((H^TH)^{-1})^T\, H^TH(H^TH)^{-1}(H^T\phi(y)) = \\ &= (H^T\phi(x))^T((H^TH)^{-1})^T(H^TH)(H^TH)^{-1}(H^T\phi(y)) = \\ &= (H^T\phi(x))^T((H^TH)^T)^{-1}(H^T\phi(y)) = \\ &= (H^T\phi(x))^T(H^TH)^{-1}(H^T\phi(y)) \end{aligned}$$

since $H = [\phi(z_1), \ldots., \phi(z_D)]$

$$\begin{aligned} &= (H^T\phi(x))^T\ (H^TH)^{-1}(H^T\phi(y)) \\ &= k_Z(x)^T K_{ZZ}^{-1}\ \ k_Z(y) \end{aligned}$$

where $\{k_Z\}_i = k(x, z_i)$ and $\{K_{zz}\}_{ij} = k(z_i, z_j)$

EFFICIENT MATCH KERNELS

- The approximated kernel is:

$$k(x,y) \cong [H\overline{v_x}]^T \left[H\overline{v_y}\right] = k_Z(x)^T K_{zz}^{-1} k_Z(y)$$

where $\{k_Z\}_i = k(x, z_i)$ and $\{K_{zz}\}_{ij} = k(z_i, z_j)$

- Since K_{zz}^{-1} is always positive definitive, it can be decomposed Cholesky Decomposition by in

$$K_{zz}^{-1} = G^T G$$

- The resulting **approximated kernel** is then

$$k(x,y) = \phi(x)^T \phi(y) \cong k_Z(x)^T K_{zz}^{-1} k_Z(y) = k_Z(x)^T G^T G k_Z(y)$$

- Thus,

$$k(x,y) = (Gk_z(\mathrm{x}))^{\mathrm{T}} (Gk_z(y))$$

In their experiments they use $k(x,y) = \exp(\|x-y\|^2)$.

PRACTICAL EMK

- Select a kernel $k(x, y)$, for example $k(x, y) = \exp(\|x - y\|^2)$.
- Compute a set of basis vectors $\mathbf{z}_i$ using k-means over a set of local features
- Compute K_{ZZ}^{-1} matrix, where $\{K_{zz}\}_{ij} = k(z_i, z_j)$
- Obtain G by inverting K_{ZZ}^{-1} and performing Cholesky Decomposition

- Compute a feature vector for each image:
  - For each local feature x, compute $\mathrm{Gk_z(x)}$, where $\{k_Z\}_i = k(x, z_i)$
  - Sum pooling and l_2 normalization

- Use these feature vectors to learn Linear SVM classifier

RESULTS

- Caltech-101

Algorithms	15 training	30 training
PMK [5, 6]	50.0±0.9	58.2
HMAX [19]	51.0	56.0
ML+PMK [9]	52.2	62.1
KC [28]	N/A	64.0
SPM [14]	56.4	64.4±0.5
SVM-KNN [31]	59.1±0.5	66.2±0.8
kCNN [30]	59.2	67.4
LDF [4]	60.3	N/A
ML+CORR [9]	61.0	69.6
NBNN [1]	65.0±1.1	73.0
EMK-Fourier	60.2±0.8	70.1±0.8
EMK-CKSVD	60.5±0.9	70.3±0.8

Pyramid match kernel

Kernel Codebook
Spatial Pyramids

ANOTHER AGGREGATION

- As seen in the kernel codebook approach, we are providing a representation of the feature vectors of our dataset, and using it to provide a fixed length description of that of a specific sample (an image).
- We did it by counting how many descriptors of every "type" we found (BOVW), or by considering how similar every descriptor was to every centroid (KCB).
- At CVPR 2010, Jégou *et al.* came up with a simple representation which again changed the way to describe our image features distribution.
- Their Vector of Locally Aggregated Descriptors (VLAD) starts from the assignment of the BOVW or Bag Of Features (BOF) of descriptors to centroids. Instead of counting they aggregate the difference vectors with summation.
- Using their notation, x is an image descriptor, x_j is the j-th component of vector x, the codebook is $\mathcal{C} = \{c_1, \dots, c_k\}$:

$$v_{i,j} = \sum_{x|NN(x)=c_i} x_j - c_{i,j}$$

VLAD

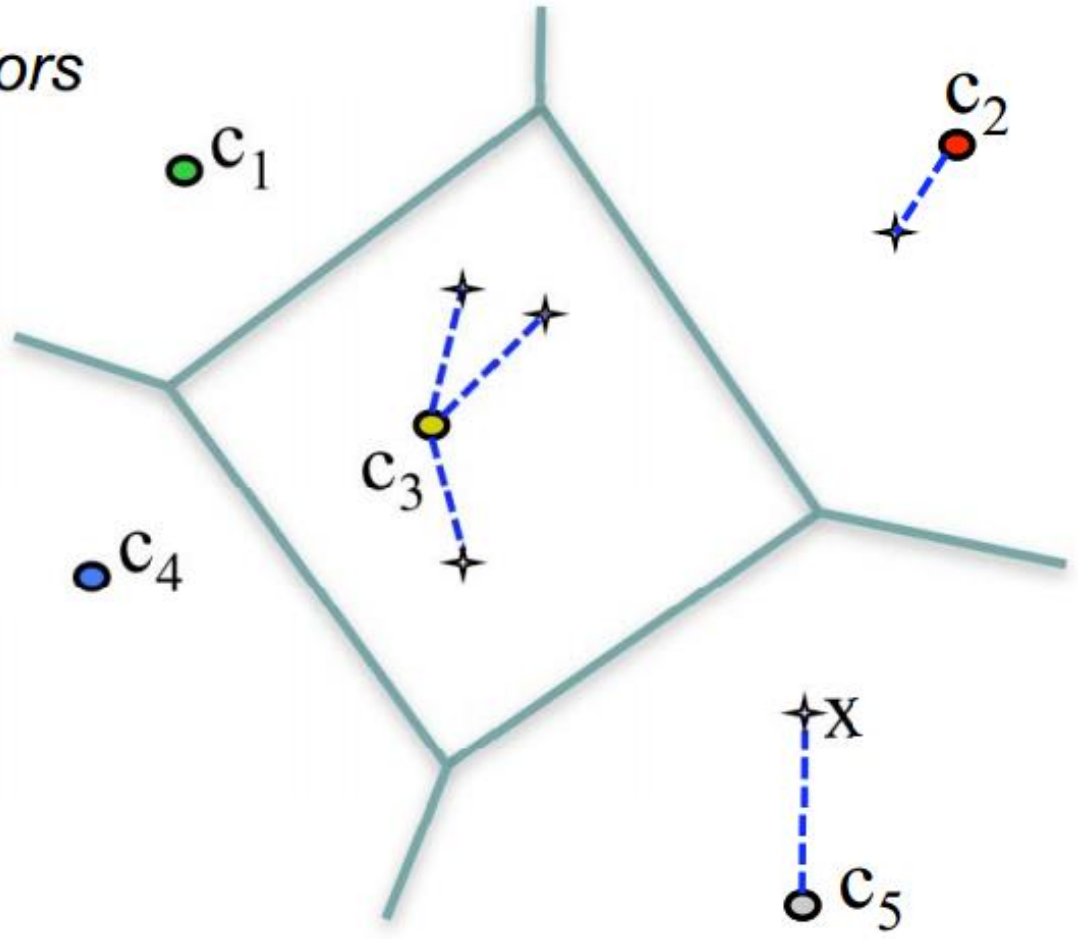
- Recalling the notation of the kernel codebook, you could write:

$$VLAD(w) = \sum_{i=1}^{n} \begin{cases} r_i - w & \text{if } w = \underset{v \in V}{\operatorname{argmin}}\big(D(v, r_i)\big) \\ 0 & \text{otherwise} \end{cases}$$

- Moreover, the vector v is subsequently L_2-normalized by $v := v/\|v\|_2$.
- It is important to stress that now the total dimensionality of the representation is $D = k \times d$, with k the number of centroids and d the dimensionality of the feature vectors. So, just as an example, if we build a 1000 centroid BOF representation, the VLAD vector of SIFT descriptors becomes 128000 dimensional.
- Interestingly, in their paper they test the descriptor against BOF with $k = 16$ for VLAD and $k = 20000$ for BOF on the Holidays dataset and the result is incredibly better for VLAD: 0.496 vs 0.446, with 1/10th of descriptor size.
- This clearly shows that a better representation of the image features leads to great improvements in the results.

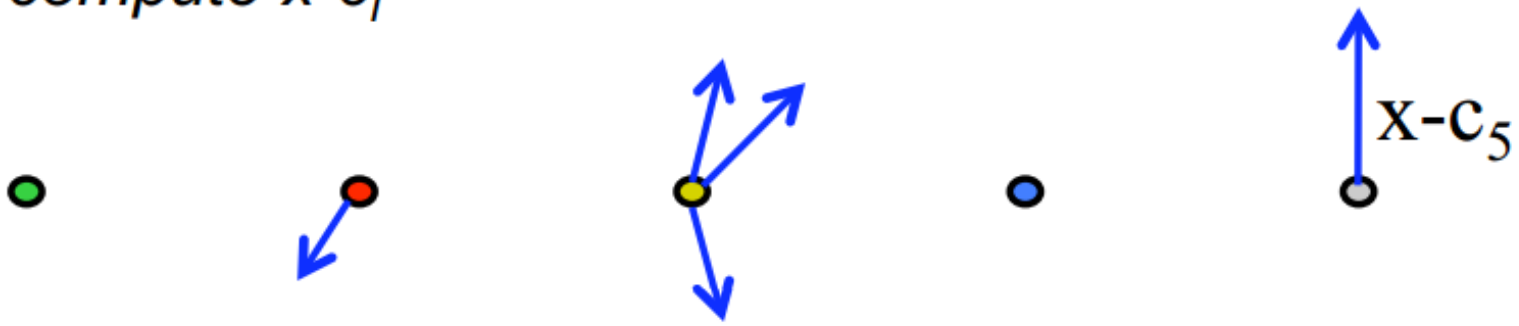
VLAD (VISUALLY)

① *assign descriptors*

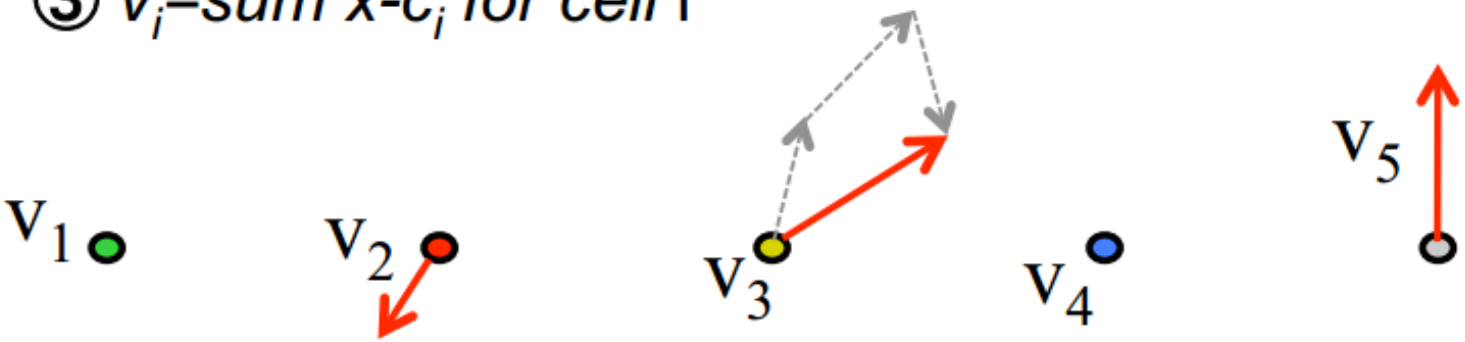


VLAD (VISUALLY)

② *compute* $x\text{-}c_i$



③ v_i=*sum* $x\text{-}c_i$ *for cell* i



VLAD (NUMBERS)

- The comparison is performed using an Hessian affine-invariant region detector and SIFT descriptors.
- Tests are performed also reducing the final descriptor to 128 and 64 dimensions (which is somehow surprising, given the original size).

Aggregator	k	D	D'=D (no reduction)	D'=128	D'=64
BoF	1,000	1,000	41.4	44.4	43.4
BoF	20,000	20,000	44.6	45.2	44.5
BoF	200,000	200,000	54.9	43.2	41.6
VLAD	16	2,048	49.6	49.5	**49.4**
VLAD	64	8,192	52.6	**51.0**	47.7
VLAD	256	32,768	**57.5**	50.8	47.6

THE FISHER KERNEL

- In our chronological analysis of literature, we need to make a little jump back to 2007, when Perronnin and Dance proposed a very interesting extension of the BOV description, by applying the Fisher Kernel (FK) to image classification. (They use bag-of-visterms to refer to BOVW, so they drop the W).
- This representation was shown to extend the BOV: it is not limited to the number of occurrences of each visual word, but it also encodes additional information about the distribution of the descriptors.
- Yet, in practice, the FK didn't receive much consideration because the BOV performed similarly and VLAD was shown to work better.
- In 2010, Perronnin *et al.* published an improvement of the original proposal which instead incorporated all mentioned techniques and introduced a different normalization which drastically changed the FK performance.
- This version of the FK instantly became the state of the art and is basically also at the present time one of the best techniques.

THE FISHER KERNEL

- The first observation which has to be made, in order to get the Fisher Kernel representation, is that we want to model the probability density function (pdf) which is generating our samples, i.e. the descriptors of our image.
- Instead of using *k*-means, they employ a Gaussian Mixture Model (GMM), which is of course a richer description of our training set.
- The likelihood that observation x_t (a D-dimensional feature vector) was generated by the GMM is

$$p(x_t|\lambda) = \sum_{i=1}^{N} w_i p_i(x_t|\lambda)$$

- with $\sum_{i=1}^{N} w_i = 1$, and

$$p_i(x|\lambda) = \frac{\exp\left\{-\frac{1}{2}(x-\mu_i)'\Sigma_i^{-1}(x-\mu_i)\right\}}{(2\pi)^{D/2}|\Sigma_i|^{1/2}}$$

- $\lambda = \{w_i, \mu_i, \Sigma_i, i = 1 \dots N\}$ is the set of parameters of the GMM.

THE FISHER KERNEL

- We are now interested in describing a set of low-level feature vectors $X = \{x_t, t = 1 \dots T\}$. The BOV could be applied here considering the *occupancy probability* of a sample and a component:

$$\gamma_t(i) = p(i|x_t, \lambda) = \frac{w_i p_i(x_t|\lambda)}{\sum_{j=1}^{N} w_j p_j(x_t|\lambda)}$$

- that is the probability for observation x_t to have been generated by the i-th Gaussian.
- Something very similar to the BOV approach could be obtained with

$$CB_i = \frac{1}{T}\sum_{t=1}^{T} \begin{cases} 1 & \text{if } i = \underset{i=1\dots N}{\text{argmax}}(\gamma_t(i)) \\ 0 & \text{otherwise} \end{cases}$$

- and another version of the Kernel Codebook could be

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

THE FISHER KERNEL

- Let's refer to the log likelihood of set X with

$$\mathcal{L}(X|\lambda) = \log p(X|\lambda)$$

- We assume that the observations (SIFT descriptors) are independent, so

$$\mathcal{L}(X|\lambda) = \sum_{t=1}^{T} \log p(x_t|\lambda)$$

- The idea of FK is to expand on these descriptions characterizing the samples with the following gradient vector:

$$\nabla_\lambda \mathcal{L}(X|\lambda)$$

- Since we plan to use this vector with inner product, it is important to correctly normalize the vector, so, following prior results, the authors apply the Fisher information matrix: $F_\lambda = E_X[\nabla_\lambda \mathcal{L}(X|\lambda)\ \nabla_\lambda \mathcal{L}(X|\lambda)']$ which gives:

$$\mathcal{G}_\lambda^X = F_\lambda^{-1/2}\ \nabla_\lambda \mathcal{L}(X|\lambda)$$

- the *Fisher vector* of X.

THE FISHER KERNEL (DERIVATION)

- It's probably possible to compute the gradient of a GMM also with full covariance matrices, but in all Fisher kernel derivations, these are supposed to be diagonal.
- This greatly helps in the derivation, in fact if $\sigma_i^2 = \mathrm{diag}(\Sigma_i)$, we have

$$p_i(x|\lambda) = \frac{\exp\left\{-\frac{1}{2}(x-\mu_i)'\Sigma_i^{-1}(x-\mu_i)\right\}}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} =$$

$$p_i\left(x|\mu_i^d,\sigma_i^d\right) = \prod_{d=1}^{D}\mathcal{N}\left(x;\mu_i^d,\sigma_i^d\right) = \prod_{d=1}^{D}\frac{1}{\sqrt{2\pi}\sigma_i^d}\exp\left(-\frac{1}{2}\left(\frac{x-\mu_i^d}{\sigma_i^d}\right)^2\right)$$

- We thus need to derive the following function:

$$\mathcal{L}\left(X|w_i,\mu_i^d,\sigma_i^d\right) = \sum_{t=1}^{T}\log\sum_{i=1}^{N}w_i p_i\left(x|\mu_i^d,\sigma_i^d\right)$$

- w.r.t. $N + N \times D + N \times D$ parameters.

THE FISHER KERNEL (DERIVATION)

- Deriving a sum is summing derivatives, so we can get rid of the first one.
- Moreover:

$$\frac{\partial \log \sum_{i=1}^{N} f(x_i)}{\partial x_j} = \frac{1}{\sum_{i=1}^{N} f(x_i)} \frac{\partial \sum_{i=1}^{N} f(x_i)}{\partial x_j} =$$

$$= \frac{1}{\sum_{i=1}^{N} f(x_i)} \sum_{i=1}^{N} \frac{\partial f(x_i)}{\partial x_j} = \frac{1}{\sum_{i=1}^{N} f(x_i)} \frac{\partial f(x_j)}{\partial x_j}$$

- So

$$\nabla_\lambda \mathcal{L}(X|\lambda) = \sum_{t=1}^{T} \frac{1}{\sum_{j=1}^{N} w_j p_j(x_t|\lambda)} \left[\frac{\partial w_i p_i(x|\lambda)}{\partial w_i} \quad \frac{\partial w_i p_i(x|\lambda)}{\partial \mu_i^d} \quad \frac{\partial w_i p_i(x|\lambda)}{\partial \sigma_i^d} \right]$$

- Note that the fraction is the denominator of $\gamma_t(i)$, thus if we can extract a $w_i p_i(x_t|\lambda)$ from every partial derivative, we can use it to get a compact formulation.

THE FISHER KERNEL (DERIVATION)

- Clearly $\frac{\partial w_i p_i(x|\lambda)}{\partial w_i} = p_i(x|\lambda)$, so:

$$\frac{\partial \mathcal{L}(X|\lambda)}{\partial w_i} = \sum_{t=1}^{T} \frac{1}{\sum_{j=1}^{N} w_j p_j(x_t|\lambda)} p_i(x|\lambda) = \sum_{t=1}^{T} \frac{\gamma_t(i)}{w_i}$$

- which takes care of the derivation w.r.t. the weights. Now:

$$\frac{\partial \prod_j f(x_j)}{\partial x_i} = \frac{\partial f(x_i)}{\partial x_i} \prod_{j \neq i} f(x_j) = \frac{\partial f(x_i)}{\partial x_i} \frac{\prod_j f(x_j)}{f(x_i)}$$

- thus

$$\frac{\partial w_i p_i(x|\lambda)}{\partial \mu_i^d} = \frac{w_i p_i(x|\lambda)}{\mathcal{N}(x; \mu_i^d, \sigma_i^d)} \frac{\partial \mathcal{N}(x; \mu_i^d, \sigma_i^d)}{\partial \mu_i^d}$$

$$\frac{\partial w_i p_i(x|\lambda)}{\partial \sigma_i^d} = \frac{w_i p_i(x|\lambda)}{\mathcal{N}(x; \mu_i^d, \sigma_i^d)} \frac{\partial \mathcal{N}(x; \mu_i^d, \sigma_i^d)}{\partial \sigma_i^d}$$

THE FISHER KERNEL (DERIVATION)

$$\frac{\partial}{\partial \mu_i^d}\frac{1}{\sqrt{2\pi}\sigma_i^d}\exp\left(-\frac{1}{2}\left(\frac{x-\mu_i^d}{\sigma_i^d}\right)^2\right)=\frac{1}{\sqrt{2\pi}\sigma_i^d}\frac{\partial}{\partial \mu_i^d}\exp\left(-\frac{1}{2}\left(\frac{x-\mu_i^d}{\sigma_i^d}\right)^2\right)=$$

$$=\mathcal{N}\left(x;\mu_i^d,\sigma_i^d\right)\frac{\partial}{\partial \mu_i^d}\left(-\frac{1}{2}\left(\frac{x-\mu_i^d}{\sigma_i^d}\right)^2\right)$$

- The Gaussian will disappear with the corresponding one at the denominator, so we derive:

$$\frac{\partial}{\partial \mu_i^d}-\frac{1}{2}\left(\frac{x-\mu_i^d}{\sigma_i^d}\right)^2=-\frac{-2x+2\mu_i^d}{2\left(\sigma_i^d\right)^2}=\frac{x-\mu_i^d}{\left(\sigma_i^d\right)^2}$$

- Finally:

$$\frac{\partial \mathcal{L}(X|\lambda)}{\partial \mu_i^d}=\sum_{t=1}^{T}\gamma_t(i)\left[\frac{x-\mu_i^d}{\left(\sigma_i^d\right)^2}\right]$$

THE FISHER KERNEL (DERIVATION)

$$\frac{\partial}{\partial \sigma_i^d}\frac{1}{\sqrt{2\pi}\sigma_i^d}\exp\left(-\frac{1}{2}\left(\frac{x-\mu_i^d}{\sigma_i^d}\right)^2\right)=$$

$$=\frac{1}{\sqrt{2\pi}}\frac{1}{\left(\sigma_i^d\right)^2}\left(\sigma_i^d\frac{\partial}{\partial \sigma_i^d}\exp\left(-\frac{1}{2}\left(\frac{x-\mu_i^d}{\sigma_i^d}\right)^2\right)-\exp\left(-\frac{1}{2}\left(\frac{x-\mu_i^d}{\sigma_i^d}\right)^2\right)\right)=$$

$$=\frac{1}{\sqrt{2\pi}\sigma_i^d}\exp\left(-\frac{1}{2}\left(\frac{x-\mu_i^d}{\sigma_i^d}\right)^2\right)\left(\frac{\partial}{\partial \sigma_i^d}\left(-\frac{1}{2}\left(\frac{x-\mu_i^d}{\sigma_i^d}\right)^2\right)-\frac{1}{\sigma_i^d}\right)=$$

$$=\mathcal{N}\left(x;\mu_i^d,\sigma_i^d\right)\left(-\frac{1}{2}\left(x-\mu_i^d\right)^2\frac{\partial}{\partial \sigma_i^d}\left(\frac{1}{\left(\sigma_i^d\right)^2}\right)-\frac{1}{\sigma_i^d}\right)=$$

$$=\mathcal{N}\left(x;\mu_i^d,\sigma_i^d\right)\left(-\frac{1}{2}\left(x-\mu_i^d\right)^2\left(-\frac{2}{\left(\sigma_i^d\right)^3}\right)-\frac{1}{\sigma_i^d}\right)=$$

$$=\mathcal{N}\left(x;\mu_i^d,\sigma_i^d\right)\left[\frac{\left(x-\mu_i^d\right)^2}{\left(\sigma_i^d\right)^3}-\frac{1}{\sigma_i^d}\right]$$

THE FISHER KERNEL (RESULT)

- The final expressions for the derivations become:

$$\frac{\partial \mathcal{L}(X|\lambda)}{\partial w_i} = \sum_{t=1}^{T} \frac{\gamma_t(i)}{w_i}$$

$$\frac{\partial \mathcal{L}(X|\lambda)}{\partial \mu_i^d} = \sum_{t=1}^{T} \gamma_t(i) \left[\frac{x - \mu_i^d}{\left(\sigma_i^d\right)^2} \right]$$

$$\frac{\partial \mathcal{L}(X|\lambda)}{\partial \sigma_i^d} = \sum_{t=1}^{T} \gamma_t(i) \left[\frac{\left(x - \mu_i^d\right)^2}{\left(\sigma_i^d\right)^3} - \frac{1}{\sigma_i^d} \right]$$

- To normalize the dynamic range of the different dimensions of the gradient vectors, we need to compute the diagonal of the Fisher information matrix F. But it is really complex and the results are available in the papers! Approximately you divide by T and multiply by sigma.
- Let us just look at the form of the derivatives: the zeroth order is basically a kernel codebook while the first order is really similar to the VLAD descriptor.

IMPROVING THE FK

- In their ECCV 2010 paper, Perronnin *et al.* analyzed and included three improvements over the original formulation:
  1. L2 normalization of the Fisher Vector
  2. Power normalization (with an empirical motivation). This consists in applying in each dimension the following function:

     $$f(z) = \text{sign}(z)\,|z|^{\alpha}$$

     where $0 \leq \alpha \leq 1$ is a parameter of the normalization. In practice, $\alpha = 0.5$ is the chosen value, so later works called this normalization Signed Square Rooting (SSR).
  3. Spatial Pyramids: following Lazebnik's work, they include the usual three level pyramid decomposition. The Fisher Vector is computed for every spatial region and separately L2-normalized.
- The reported results use K=256 Gaussians with 128-D SIFT descriptors, and drop the gradient with respect to the weight parameters. The FV becomes $2 \times 256 \times 128 \times 21 = 1\,376\,256$ dimensional. This requires some care when using a classifier.

LOCALITY-CONSTRAINED LINEAR CODING FOR IMAGE CLASSIFICATION

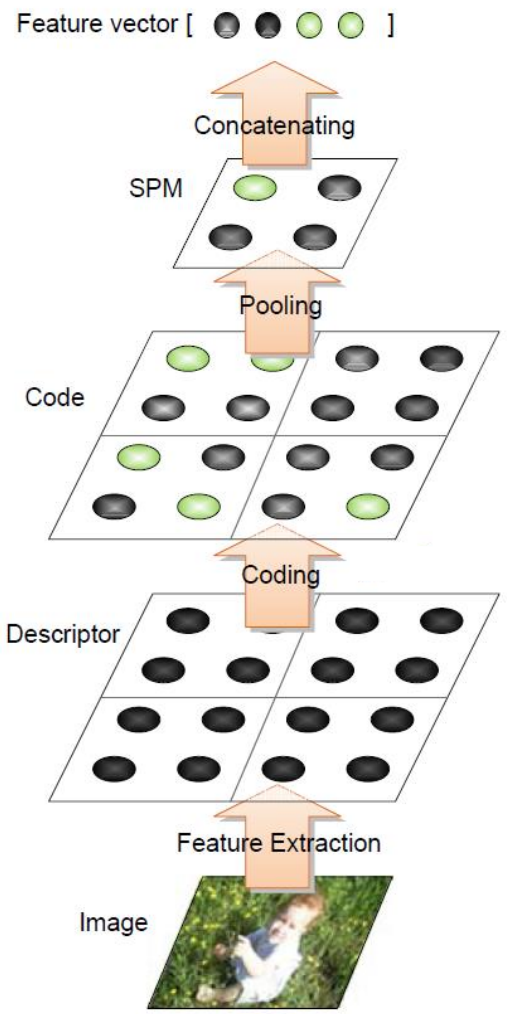
- As observed before, the standard image classification pipeline can be roughly divided in:

Feature extraction

- SIFT
- HOG
- etc

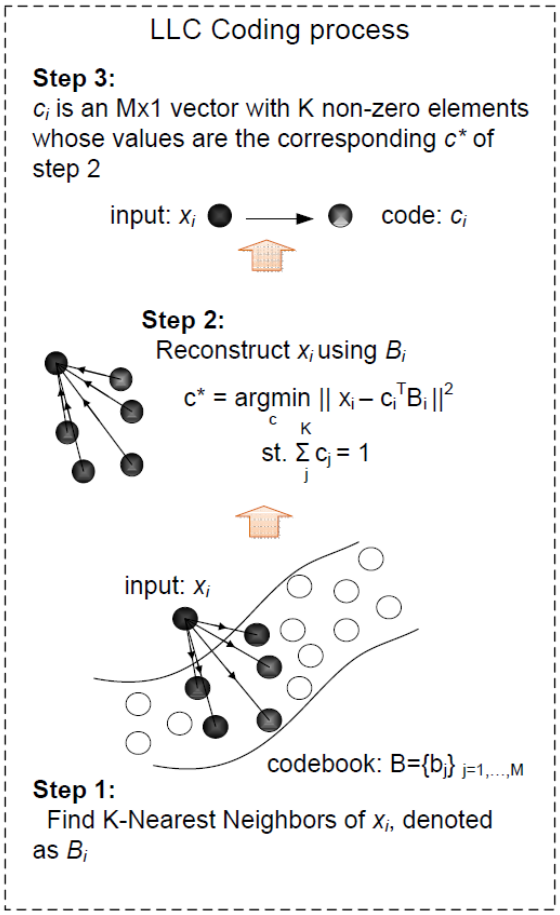
- **Coding**
  - Vector Quantization
  - Sparse coding
  - etc

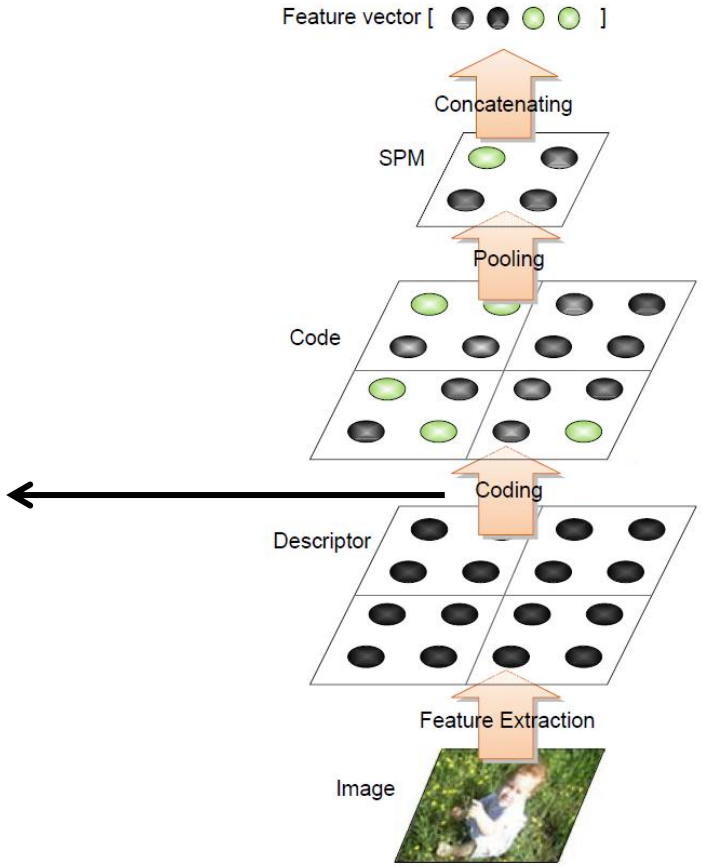
- **Pooling**
  - Max pooling
  - Sum pooling



LOCALITY-CONSTRAINED LINEAR CODING FOR IMAGE CLASSIFICATION

In 2010 Wang et al. proposed a method based on space coding obtaining the best result on IMAGENET Large Scale Visual Recognition Challenge 2010 (ILSVRC2010)





VECTOR QUANTIZATION (VQ)

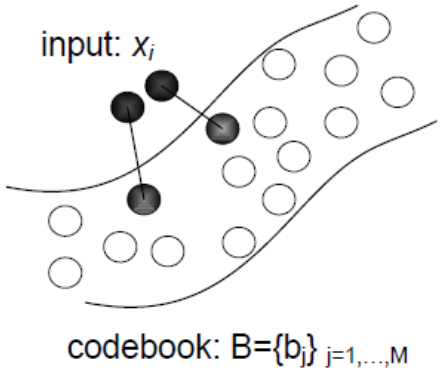
Hard quantization method

- A set of D-dimensional local descriptors

$$X = [x_1, x_2, \dots, x_N] \in R^{D \times N}$$

- Codebook with M entries

$$B = [b_1, b_2, \dots, b_M] \in R^{D \times M}$$



VQ

- Objective function

$$\underset{C}{argmin} \sum_{i=1}^{N} \|x_i - Bc_i\|^2$$

$$s.t. \|c_i\|_{l^0} = 1, \|c_i\|_{l^1} = 1, c_i \succcurlyeq 0, \forall i$$

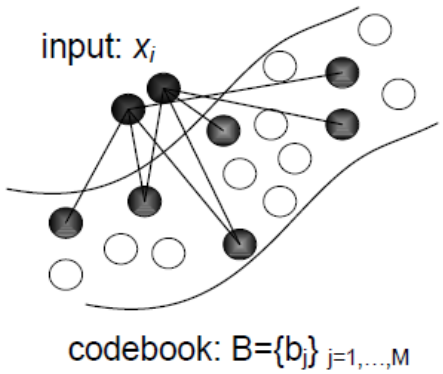
where $C = [c_1, c_2, \dots, c_N]$ is the set of codes for X

$\|c_i\|_{l^0} = 1, \|c_i\|_{l^1} = 1, c_i \succcurlyeq 0, \forall i$ mean that only one element of c_i is non-zero and equal to 1

SPARSE CODING (SC)

- But why should the feature be assigned to only one codebook entry?
- Ameliorate the quantization loss of VQ by removing the constraint $\|c_i\|_{l^0} = 1$ and instead using a sparsity regularization term to restrict the number of non-zero bases:

$$arg\min_{C} \sum_{i=1}^{N} \|x_i - Bc_i\|^2 + \lambda \|c_i\|_{l^1}$$

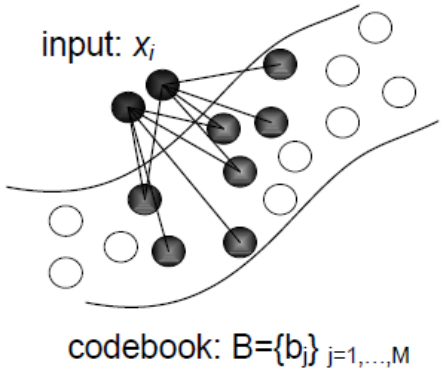


SC

- By assigning to multiple bases we overcome the quantization errors introduced by VQ
- On Caltech-101 using dense SIFT, it yields 10% improvement over VQ, and 5~6% improvement over soft-assignment using kernel codebooks with a **linear SVM**

LOCALITY-CONSTRAINED LINEAR CODING (LLC)

- The effectiveness of distance-based soft-assignment suggests that the locality of the visual words used to describe any feature is also important
- LLC account for this by replacing the sparsity regularization with a locality constraint:



LLC

$$\arg\min_{C} \sum_{i=1}^{N} \|x_i - Bc_i\|^2 + \lambda \|d_i \odot c_i\|^2$$

$$s.t.\ 1^T c_i = 1, \forall i$$

- $\odot$: the element-wise multiplication
- $d_i \in R^M$

$$d_i = \exp\left(\frac{dist(x_i, B)}{\sigma}\right)$$

where $dist(x_i, B) = [dist(x_i, b_1), dist(x_i, b_2), \ldots, dist(x_i, b_M)]^T$ and σ is used for adjusting the weight decay speed for the locality adaptor.

APPROXIMATED LLC FOR FAST ENCODING

- The distance regularization of LLC effectively performs **feature selection**
- This suggests we can develop a fast approximation of LLC by removing the regularization completely and instead using the K $(K < D < M)$ nearest neighbors of x_i as the local bases B_i, and solve a much smaller linear system to get the codes

$$\min_{\tilde{C}} \sum_{i=1}^{N} \|x_i - \tilde{c}_i B_i\|^2$$
$$s.t.\, 1^T \tilde{c}_i = 1, \forall i$$

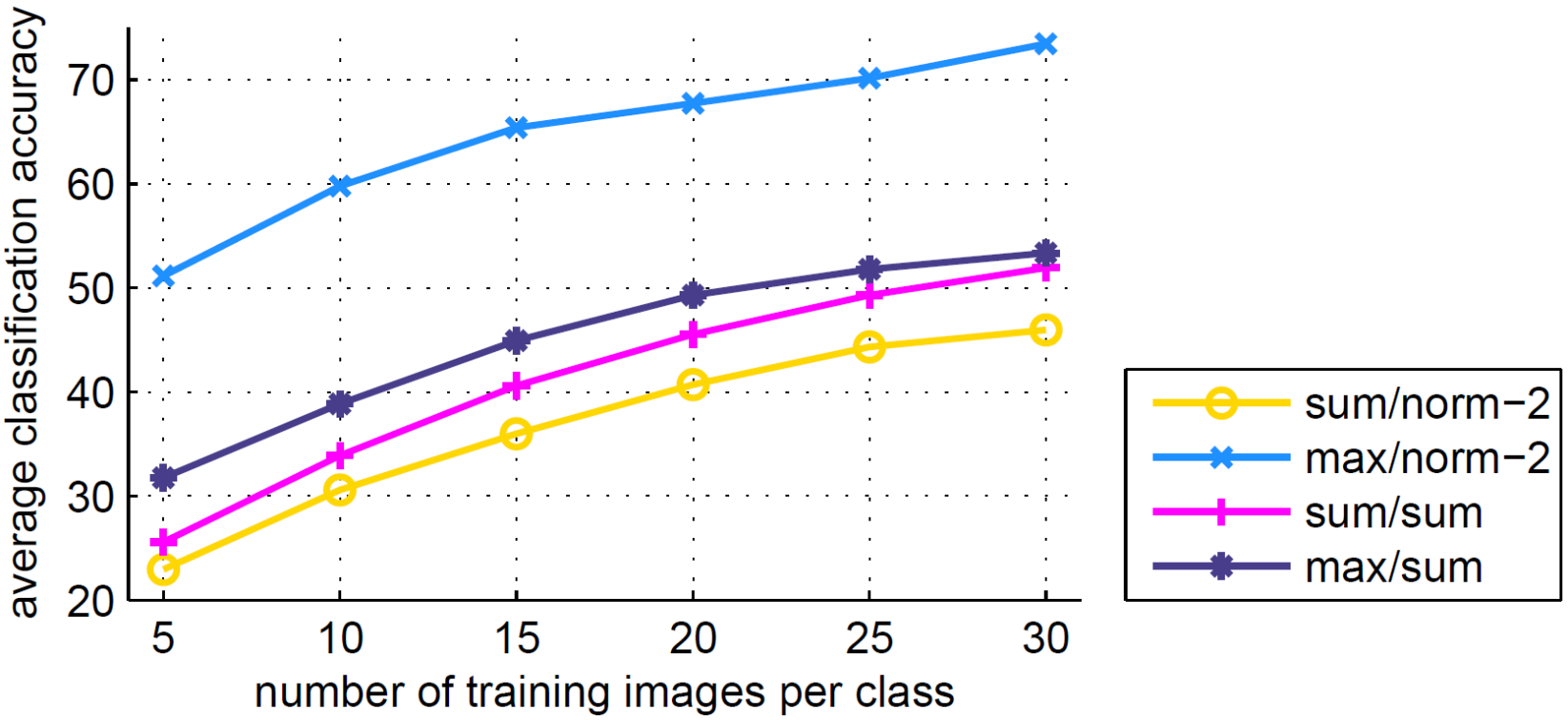
- As K is usually very small (in the experiments the authors set it to 5), solving the equation is very fast.

SOME DETAILS

- Performance under different neighbors (Caltech-101)

SOME DETAILS

- Performance with different pooling strategies (Caltech-101)



RESULTS

- Caltech-101

training images	5	10	15	20	25	30	
Zhang [25]	46.6	55.8	59.1	62.0	-	66.20	
Lazebnik [15]	-	-	56.40	-	-	64.60	Spatial Pyramids
Griffin [11]	44.2	54.5	59.0	63.3	65.8	67.60	
Boiman [2]	-	-	65.00	-	-	70.40	
Jain [12]	-	-	61.00	-	-	69.10	
Gemert [8]	-	-	-	-	-	64.16	Kernel Codebook
Yang [22]	-	-	**67.00**	-	-	73.20	Sparse Coding
Ours	**51.15**	**59.77**	65.43	**67.74**	**70.16**	**73.44**	LLC

RESULTS

- VOC 2007

object class	aero	bicyc	bird	boat	bottle	bus	car	cat	chair	cow
Obj.+Contex [20]	**80.2**	61.0	49.8	69.6	21.0	66.8	**80.7**	51.1	51.4	35.9
Best PASCAL'07 [6]	77.5	63.6	**56.1**	**71.9**	**33.1**	60.6	78.0	58.8	53.5	42.6
Ours	74.8	**65.2**	50.7	70.9	28.7	**68.8**	78.5	**61.7**	**54.3**	**48.6**

object class	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	average
Obj.+Contex [20]	**62.0**	38.6	69.0	61.4	84.6	28.7	**53.5**	**61.9**	**81.7**	**59.5**	58.4
Best of PASCAL'07 [6]	54.9	**45.8**	**77.5**	64.0	**85.9**	**36.3**	44.7	50.9	79.2	53.2	59.4
Ours	51.8	44.1	76.6	**66.9**	83.5	30.8	44.6	53.4	78.2	53.5	59.3

- **Computational performance**: using a codebook with 2048 entries, a 300x300 image requires only 0.24 second on average for processing (including dense local descriptors extraction, LLC coding and SPM pooling to get the final representation).

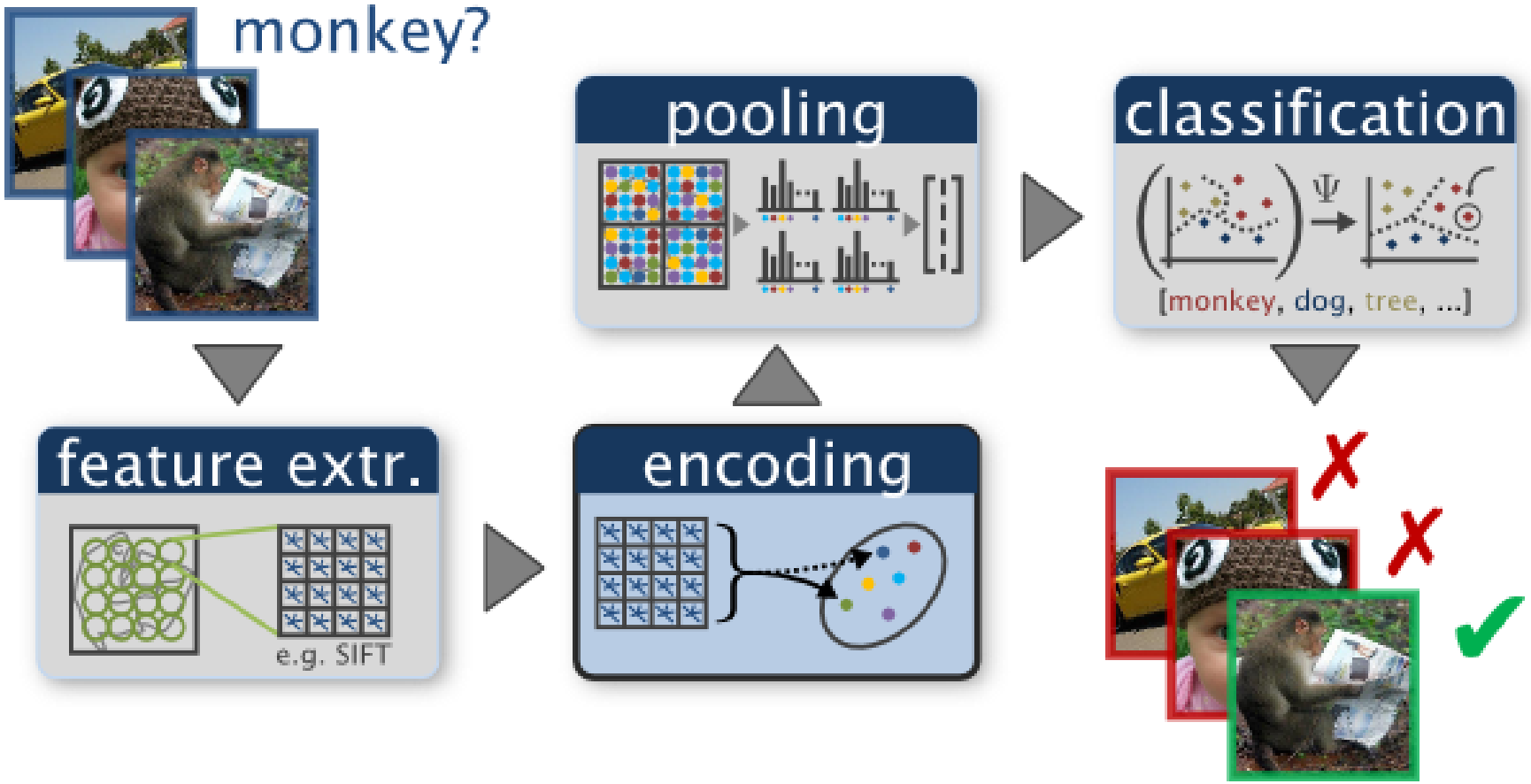
THINGS TO NOTE AND TO REMEMBER WHEN APPLYING THESE TECHNIQUES

IMPLEMENTATION DETAILS

THE DEVIL IS IN THE DETAILS

- We went through a lot!
- Due to differences in the image feature extraction and learning methodology, often published results cannot be directly compared.
- In ECCV 2011, Chatfield *et al.* published a really well conducted experiment, which marked a clear methodology and a set of well defined conditions.
- Results were compared on two datasets: Caltech-101 and PASCAL VOC 2007. For Caltech-101 three random splits were selected and used for all techniques.
- SIFT descriptors were extracted with a spatial stride of between two and five pixels, and at four scales, defined by setting the width of the SIFT spatial bins to 4, 6, 8 and 10 pixels respectively. The rotation of the SIFT features is fixed to a constant value. Additionally, low contrast SIFT descriptors are detected by measuring the average of the gradient magnitude (in the descriptor support) and dropped when this magnitude is below a certain threshold (about 5% of the SIFT features in the PASCAL dataset satisfy this condition).

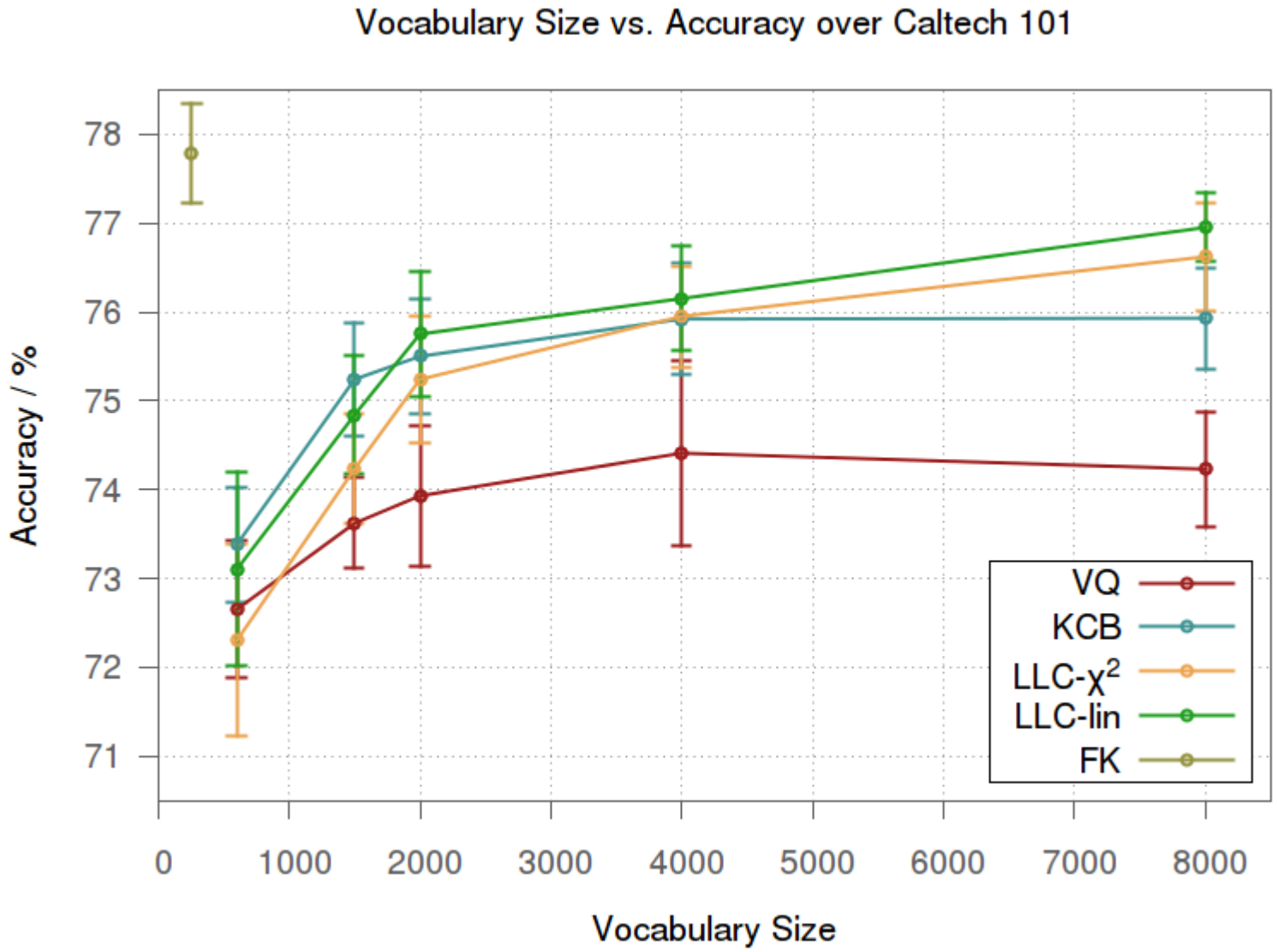
PIPELINE


monkey?
pooling
classification
Ψ
[monkey, dog, tree, ...]
feature extr.
e.g. SIFT
encoding

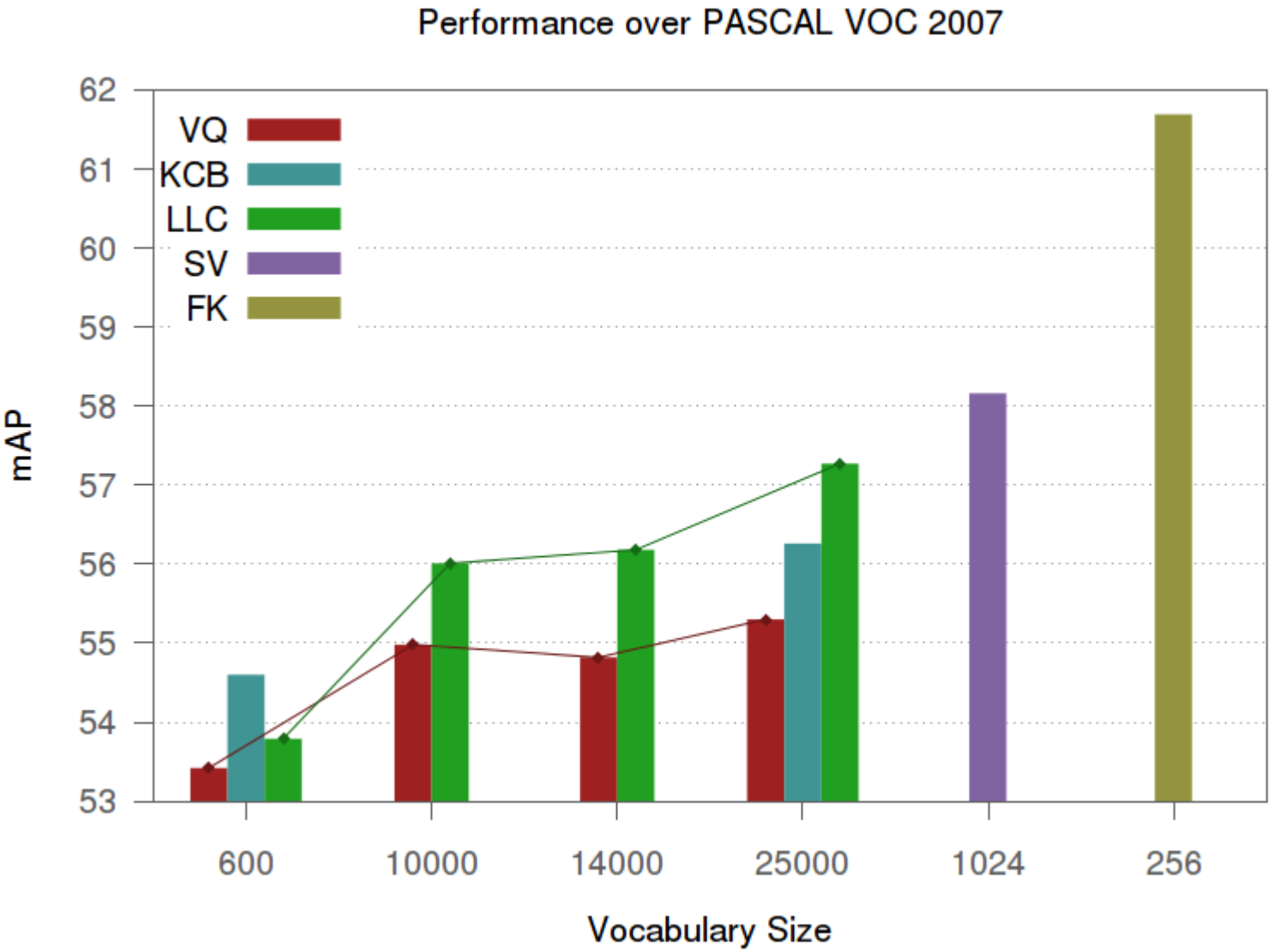
ENCODING AND POOLING

- Five **encoding** techniques are compared:
  - Histogram encoding (VQ) - the standard BOVW
  - Kernel codebook encoding (KCB)
  - Locality constrained linear coding (LLC)
  - Fisher encoding (FK)
  - Supervector encoding (SV) - we didn't analyze this proposal
- Codes are then **pooled** spatially with sum or max using the spatial pyramid.
- Particular attention is devoted to normalization: L1 is used with VQ and KCB, while L2 is used with other encodings.
- Each spatial region is normalized, then all are stacked together and another L2 global normalization is performed.
- The spatial regions are the classic $1 \times 1, 2 \times 2, 4 \times 4$ for Caltech-101, while for PASCAL VOC a simpler $1 \times 1, 3 \times 1, 2 \times 2$ is used.
- Classification is performed with the dual formulation (precomputed kernel matrix) of a linear SVM (C is optimized on a validation set).

RESULTS


Vocabulary Size vs. Accuracy over Caltech 101
Accuracy / %
78
77
76
75
74
73
72
71
0
1000
2000
3000
4000
5000
6000
7000
8000
Vocabulary Size
VQ
KCB
LLC-χ2
LLC-lin
FK

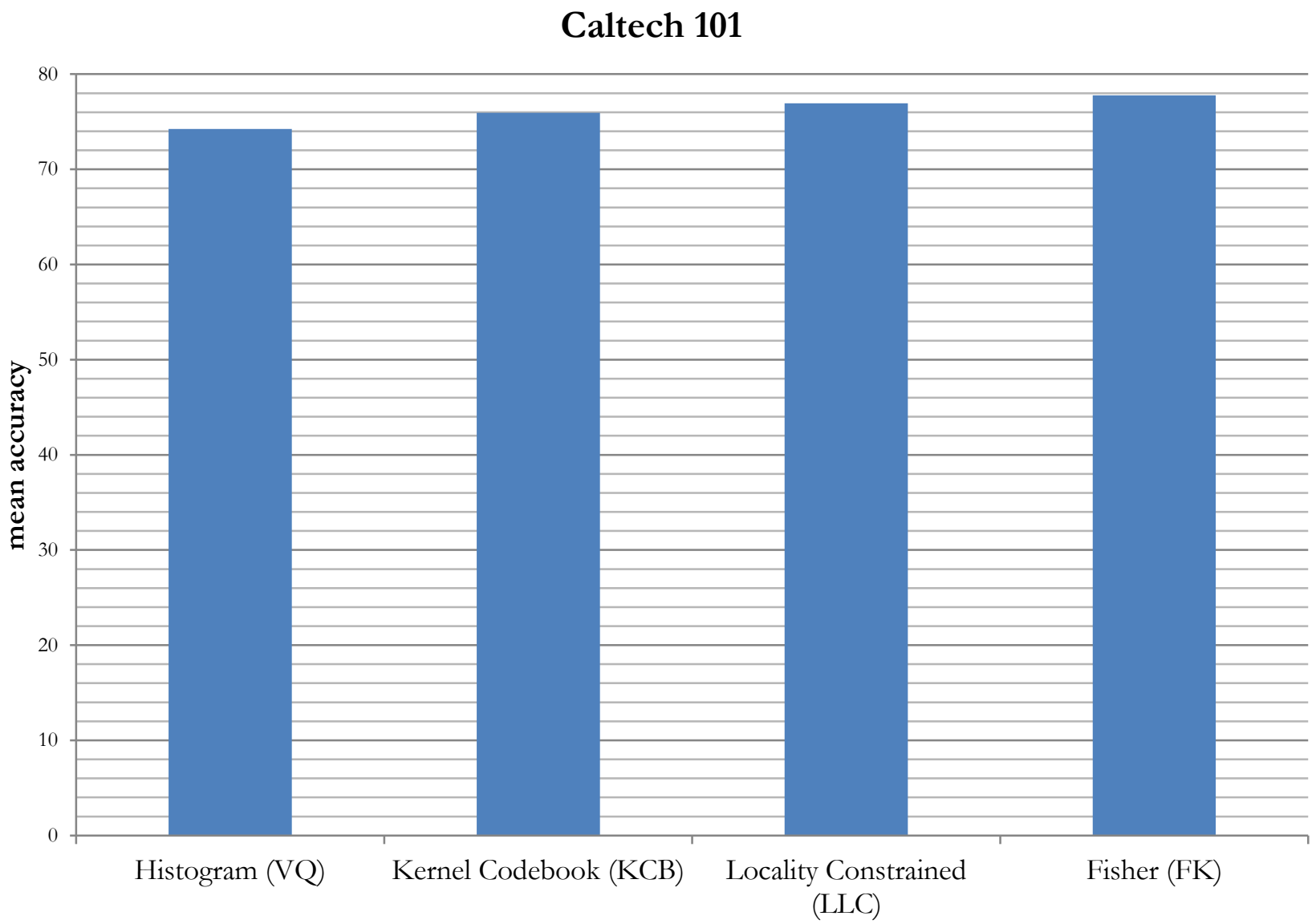
RESULTS


Performance over PASCAL VOC 2007
VQ
KCB
LLC
SV
FK
mAP
62
61
60
59
58
57
56
55
54
53
600
10000
14000
25000
1024
256
Vocabulary Size

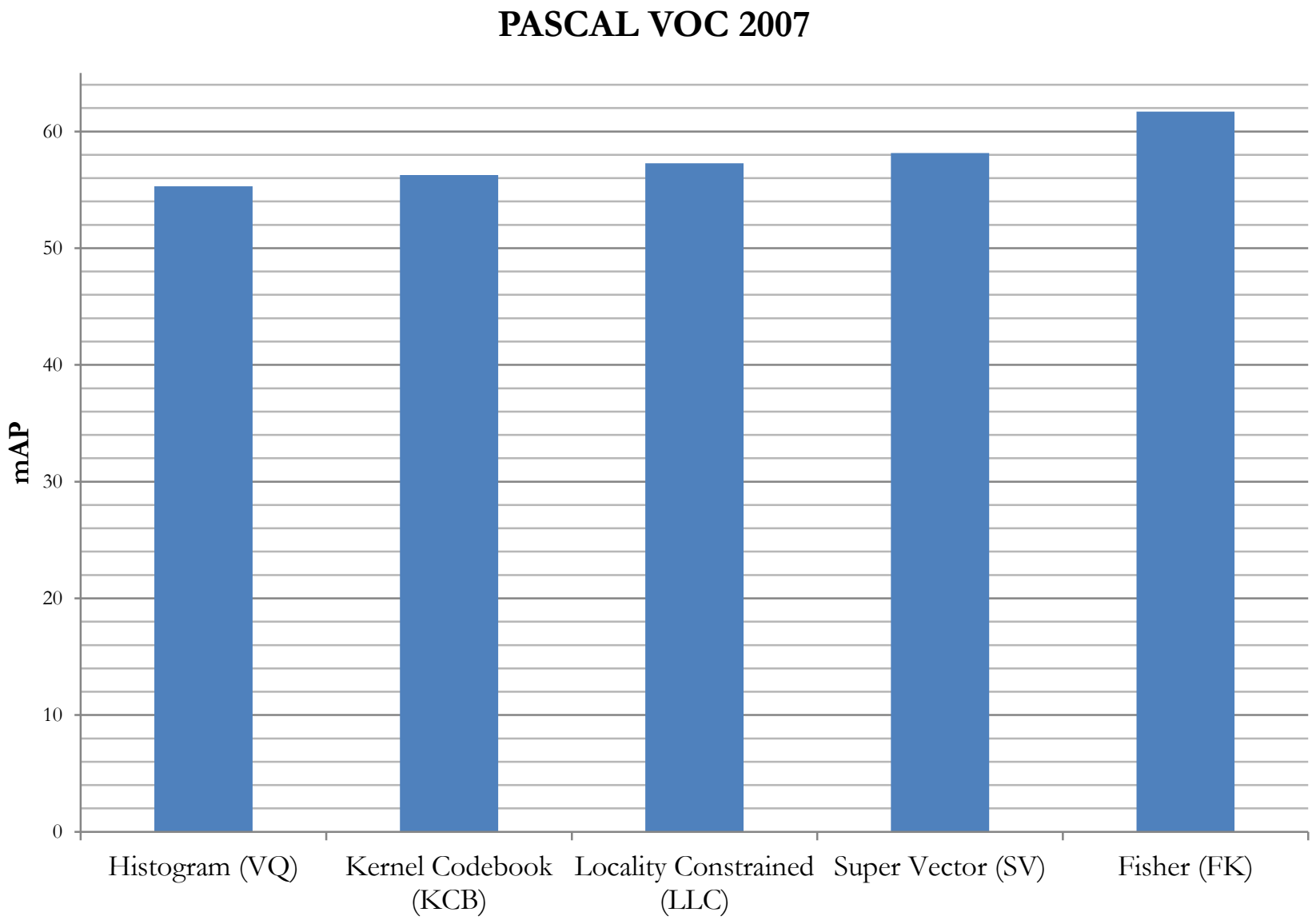
RESULTS

Caltech 101	vocabulary size (+ dimensionality)	mean accuracy
Histogram (VQ)	8,000	74.23 ± 0.65
Kernel Codebook (KCB)	8,000	75.93 ± 0.57
Locality Constrained (LLC)	8,000	76.95 ± 0.39
Fisher (FK)	256 (~41k)	77.78 ± 0.56
VOC 2007	vocabulary size (+ dimensionality)	mAP
Histogram (VQ)	25,000	55.30
Kernel Codebook (KCB)	25,000	56.26
Locality Constrained (LLC)	25,000	57.27
Fisher (FK)	256 (~41k)	61.69
Super Vector (SV)	1024 (~132k)	58.16

RESULTS


Caltech 101
mean accuracy
80
70
60
50
40
30
20
10
0
Histogram (VQ)
Kernel Codebook (KCB)
Locality Constrained (LLC)
Fisher (FK)

RESULTS


PASCAL VOC 2007
mAP
0
10
20
30
40
50
60
Histogram (VQ)
Kernel Codebook (KCB)
Locality Constrained (LLC)
Super Vector (SV)
Fisher (FK)

FURTHER IMPROVEMENTS

- In CVPR 2012, Arandjelović and Zisserman contributed three improvements, which can be included in most of the solutions we described:
  - RootSIFT: take the square root of every SIFT descriptor value. This is an explicit feature mapping using the Hellinger kernel.
  - Discriminative query expansion (DQE): it re-ranks results by their distance from the decision boundary of a linear SVM learned on positive and negative examples provided by the basic BOVW results.
  - Database-side feature augmentation (AUG): images are augmented with visual words from relevant neighboring regions.

Retrieval Method	SIFT		RootSIFT	
	Ox5k	Ox105k	Ox5k	Ox105k
Philbin *et al.* [23]: tf-idf ranking	0.636	0.515	0.683	0.581
Philbin *et al.* [23]: tf-idf with spatial reranking	0.672	0.581	0.720	0.642
Chum *et al.* [6]: average query expansion (AQE)	0.839	0.726	0.850	0.756
Turcot and Lowe [29]: database-side feature augmentation (AUG)	0.776	0.711	0.827	0.759
This paper: discriminative query expansion (DQE)	**0.847**	0.752	0.861	0.781
This paper: spatial database-side feature augmentation (SPAUG)	0.785	0.723	0.838	0.767
This paper: SPAUG + DQE	0.844	**0.795**	**0.881**	**0.823**

SOME DETAILS

- We present some experiments in order to show how details are very important to reach good results.
- We used VLFeat to train an test an image classifier on the Caltech-101 data. The classifier achieves 65% average accuracy by using a single feature and 15 training images and 15 testing images per class.
- The code uses:
  - PHOW features (dense multi-scale SIFT descriptors)
  - Elkan k-means for fast visual word dictionary construction
  - A nonlinear Coding: the homogeneous kernel map to transform a χ^2 kernel support vector machine (SVM) into a linear one
  - Linear SVM classifiers

SINGLE SCALE - MULTIPLE

N Train	N Test	Codebook	SP	GRID STEP	SIFT SCALE	ACCURACY
15	15	600	[1 1]	3	4	46.93
15	15	600	[1 1]	3	[4 6]	49.02
15	15	600	[1 1]	3	[4 6 8]	51.90
15	**15**	**600**	**[1 1]**	**3**	**[4 6 8 10]**	**52.34**
15	15	600	[1 1]	3	[4 6 8 10 12]	52.55

GRID SPACE

N Train	N Test	Codebook	SP	GRID STEP	ACCURACY
15	15	600	[1 1]	15	39.15
15	15	600	[1 1]	12	42.88
15	15	600	[1 1]	9	47.45
15	15	600	[1 1]	6	50.39
15	**15**	**600**	**[1 1]**	**3**	**52.35**
15	15	600	[1 1]	1	52.03

PYRAMID LEVEL

N Train	N Test	Codebook	SP	GRID STEP	ACCURACY
15	15	600	[1]	3	52.35
15	15	600	[1 2]	3	63.79
15	**15**	**600**	**[1 2 4]**	**3**	**68.56**
15	15	600	[1 2 4 6]	3	68.89

A LITTLE NOTE THAT WE ADD ON TOP OF ALL THIS!

OUR 2 CENTS

LIMITATION OF BAG OF WORDS

- Introducing a quantization of the feature space is a critical step and tightly ties dataset characteristics to the features representation.

- An example J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman in "Lost in quantization: Improving particular object retrieval in large scale image databases" (CVPR 2008) show that

Codebook Generation	Image Retrieval task	mAP
Oxford dataset	Oxford dataset	0.67
Paris dataset	Oxford dataset	0.49

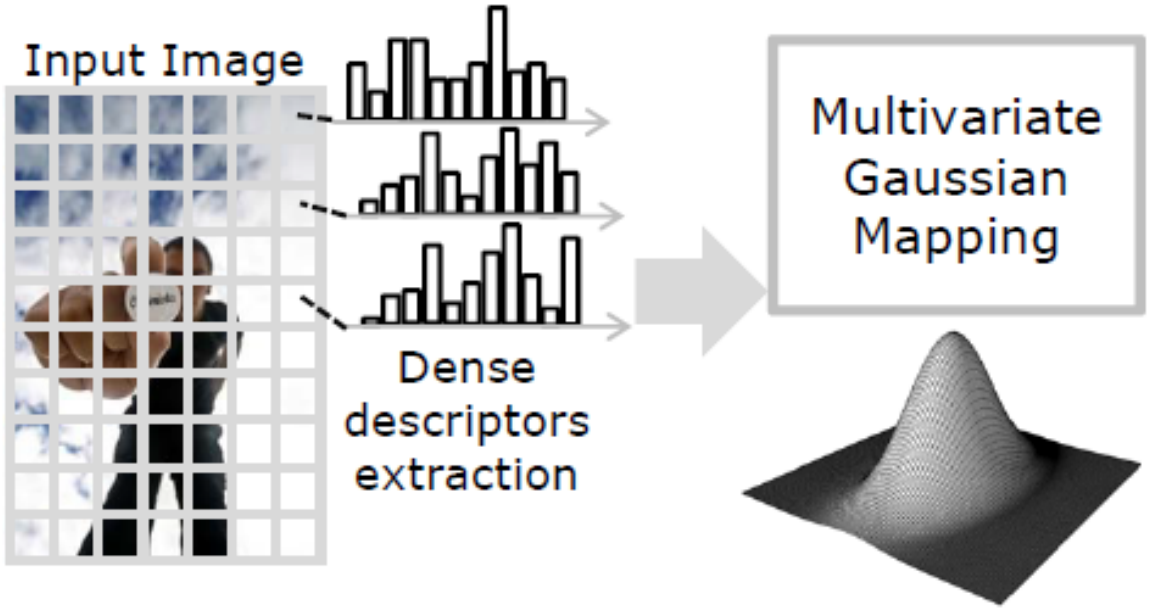
AVOID CODEWORD GENERATION

- In 2005 a different strategy was proposed in a technical report by *Farquhar et al.* in order to **remove any dataset dependency**.
- **Their idea was**:
  - first model each set of vectors by a probability density function
  - compare them with a kernel defined over the pdfs
- This solution received little attention, because this method requires a **specific kernel** and thus is **computationally expensive**.

- Recently Carreira *et al.* proposed to use second-order analogues of the most common first-order pooling operators to describe image regions.
- Following the technique proposed da Tuzel et al. (PAMI 2008), they managed to obtain a descriptor suitable for linear classifiers.

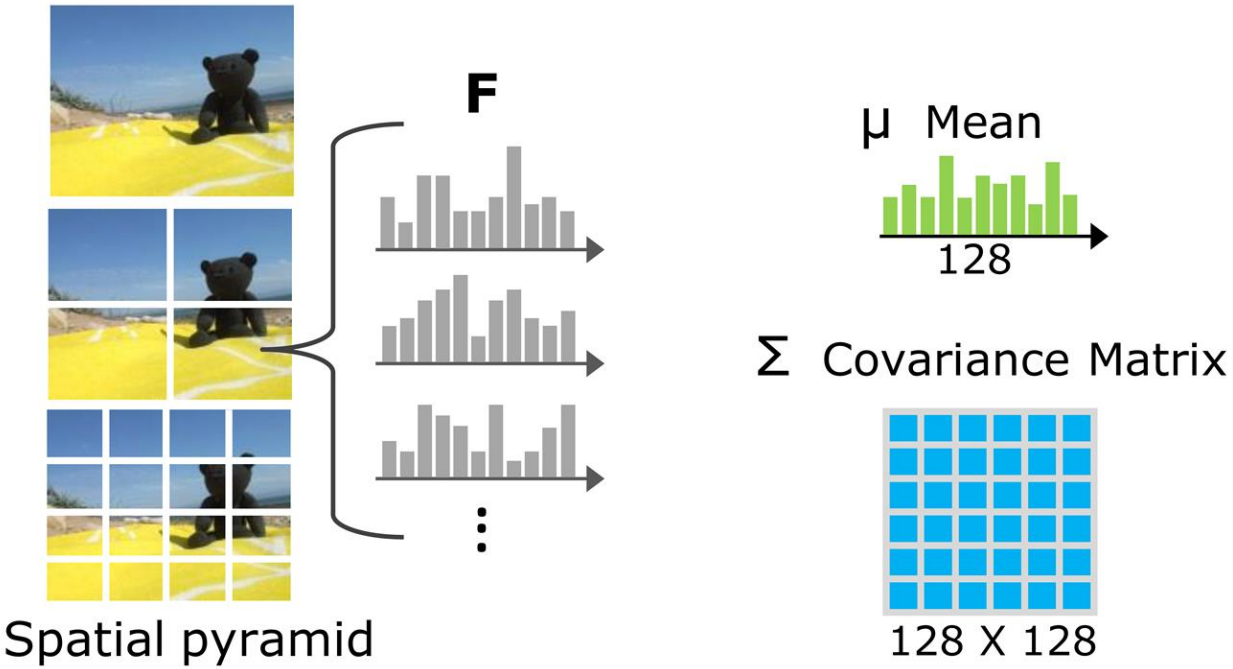
GOLD: GAUSSIAN OF LOCAL DESCRIPTORS

- In 2013 we propose to follow this latter way of modeling local features distributions, by choosing a **multivariate Gaussian distribution**.



GAUSSIAN OF LOCAL DESCRIPTORS

- We partition the image into regions following the Spatial Pyramid approach for example $1 \times 1, 2 \times 2, 4 \times 4$.
- Extract local features (e.g. SIFT) from a region on a regular grid
- Describe the local features distribution with a **Multivariate Gaussian Distribution,** thus obtaining a fixed length descriptor, composed of the mean and the full rank covariance matrix.



GAUSSIAN OF LOCAL DESCRIPTORS

- More formally, let $F = \{f_1 \dots f_N\}$ be the set of d-dimensional local features and suppose that they are independent and identically distributed samples form a multivariate Gaussian distribution, defined as

$$N(f; m, C) = \frac{1}{|2\pi C|^{\frac{1}{2}}} e^{-\frac{1}{2}(f-m)^T C^{-1}(f-m)}$$

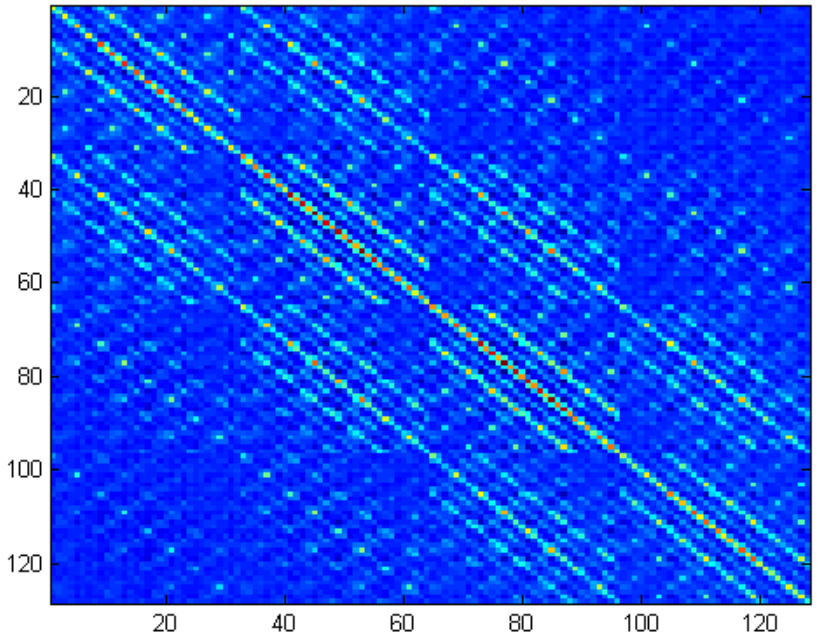
- where $|\cdot|$ is the determinant, m is the mean vector and C is the covariance matrix estimated from F as follows:

$$m = \frac{1}{N}\sum_{i=1}^{N} f_i$$

$$C = \frac{1}{N-1}\sum_{i=1}^{N}(f_i - m)(f_i - m)^T$$

COVARIANCE MATRIX

- Although the **covariance matrix** encodes information about the variance of the features and their correlation, it does **not lie in a vector space**.
- The covariance space is a Riemannian manifold and is not closed under multiplication with a negative scalar.
- Our basic idea is to map the Riemannian manifolds to Euclidean spaces as suggested by Tuzel *et al.*

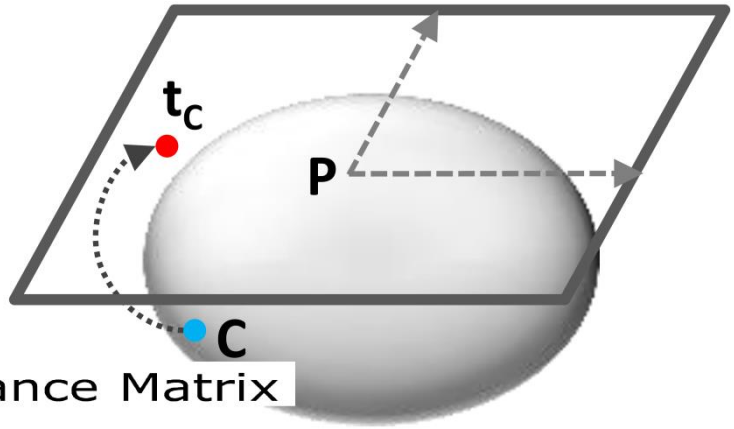


MULTIVARIATE GAUSSIAN DESCRIPTOR

- The first step is the projection of the covariance matrices on an Euclidean space tangent to the Riemannian manifold, on a specific tangency matrix P.
- The second step is the extraction of the orthonormal coordinates of the projected vector in order to obtain a minimal representation.
- The projection of C on the hyperplane tangent to P is

$$c = vec_I(\log(P^{-\frac{1}{2}} C P^{-\frac{1}{2}}))$$

where $vec_I(Y) = [y_{1,1}\ \sqrt{2}y_{1,2}\ \sqrt{2}y_{1,3} \dots y_{2,2}\ \sqrt{2}y_{2,3} \dots y_{d,d}]$ and $\log$ is the matrix logarithm.



COVARIANCE PROJECTION

- More formally, the projection of C is given by:

$$t_C = \log_P(C) \triangleq P^{\frac{1}{2}} \log\left(P^{-\frac{1}{2}} C\, P^{-\frac{1}{2}}\right) P^{\frac{1}{2}}$$

where log is the matrix logarithm.

- The orthonormal coordinates of the projected vector t_C in the tangent space at point P are given by the vector operator:

$$vec_P(t_C) = vec_I(P^{-\frac{1}{2}} t_C P^{-\frac{1}{2}})$$

where I is the identity matrix, while the vector operator is defined as:

$$vec_I(Y) = [\, y_{1,1}\ \sqrt{2}y_{1,2}\ \sqrt{2}y_{1,3} \cdots y_{2,2}\ \sqrt{2}y_{2,3} \cdots y_{d,d}]$$

- Substituting t_C in $vec_P(t_C)$ the projection of C on the hyperplane tangent to P becomes:

$$c = vec_I\left(\log\left(P^{-\frac{1}{2}} C\, P^{-\frac{1}{2}}\right)\right)$$

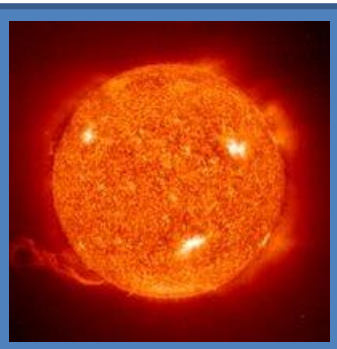
RESULTS ON CALTECH-101

	15 Training	30 Training	
GOLD	**73.39**	**80.92**	
Bo et al. [40]	60.50	73.86	EMK
Grauman et al. [41]	50.00	58.20	Pyramid Match Kernel
Jia et al. [42]	-	75.30	
Jiang et al. [43]	67.50	75.30	
Liu et al. [44]	-	74.21	
C. Zhang et al. [45]	69.58	75.68	
Tuytelaars et al. [46]	69.20	75.20	
Wang et al. [13]	65.43	73.40	LLC
Yang et al. [47]	67.00	73.20	Sparse Coding
Carreira et al. [5]	-	79.20	Second-order pooling
Lazebnik et al. [6]	56.40	64.60	Spatial Pyramids

IMAGECLEF 2013 EXPERIENCE

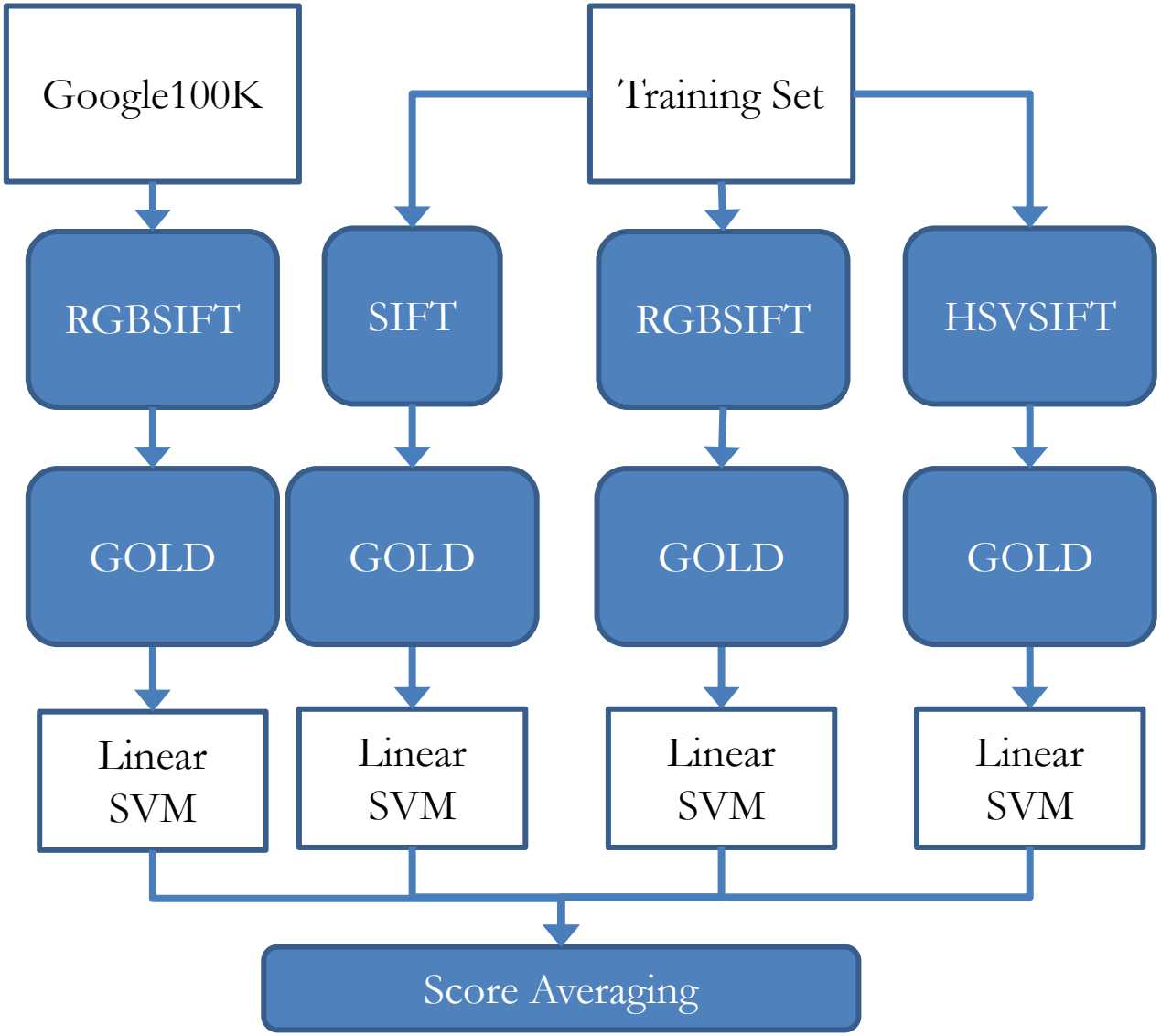
We participated at **IMAGECLEF 2013** Annotation Task that consists**:**

- **Training set** (250,000 crawled images and respective webpages)
- **Development set** (1,000 images, labeled for 95 concepts)
- **Test set** (2,000 images, which have to be labeled with 116 concepts)
- **Concepts**: Were defined as WordNet synsets and for most of them, also a Wikipedia article was associated.



Images from a web search query of "sun"

UNIMORE@IMAGECLEF 2013



UNIMORE@IMAGECLEF 2013

Test set participants' results

Group and Run #	MF-samples (%)	MF-concepts (%)	MF-concepts unseen in dev. (%)	MAP-samples (%)
UNIMORE_5	31.5 [30.6--32.5]	31.9 [28.3--36.7]	31.9 [23.1--46.9]	45.6 [44.6--46.7]
TPT_6	42.6 [41.4--43.7]	34.1 [30.3--38.9]	45.1 [34.4--57.5]	44.4 [43.3--45.5]
TPT_5	32.5 [31.7--33.3]	26.7 [23.8--31.2]	27.3 [20.2--42.2]	44.3 [43.3--45.5]
UNIMORE_2	27.5 [26.4--28.7]	33.1 [29.4--37.9]	34.8 [26.0--48.8]	44.1 [43.1--45.2]
UNIMORE_6	31.1 [30.2--32.0]	32.0 [28.5--36.7]	31.3 [22.9--46.2]	44.1 [43.1--45.2]
TPT_3	31.9 [31.1--32.7]	24.8 [22.0--29.1]	24.7 [19.2--38.8]	43.6 [42.5--44.7]
TPT_4	41.8 [40.8--42.9]	33.7 [30.2--38.3]	45.3 [34.9--57.2]	43.2 [42.2--44.3]
MIL_1	33.2 [32.5--34.0]	32.6 [29.3--37.0]	33.8 [25.6--47.6]	42.1 [41.1--43.0]

COMMENTS

- Modeling local features with a Multivariate Gaussian is effective and leads to state-of-the-art results;
- Using several SIFT variations in a late fusion approach is useful and enhance considerably the performance;
- Retrieving images from Google gives 4-5 percentage points of MAP

WHAT'S NEXT?

PREDICTIONS

- We all know that the future is easy to predict: movies keep doing it all the time...
- So, let's leverage their wisdom!

SMART WATCHES

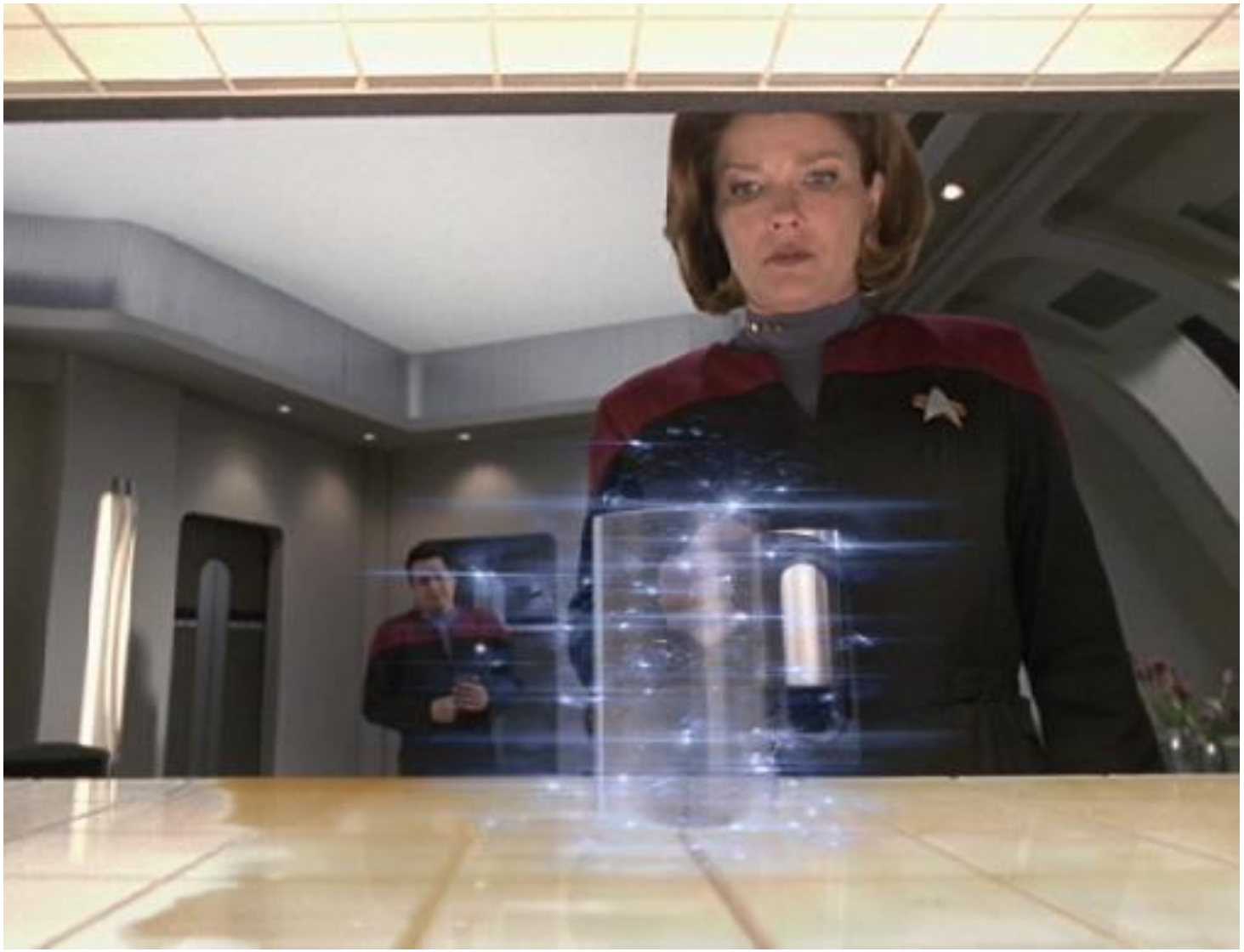
CELL PHONES

IPAD

MILITARY ROBOTS

GOOGLE GLASS

3D PRINTING



DRIVERLESS CARS

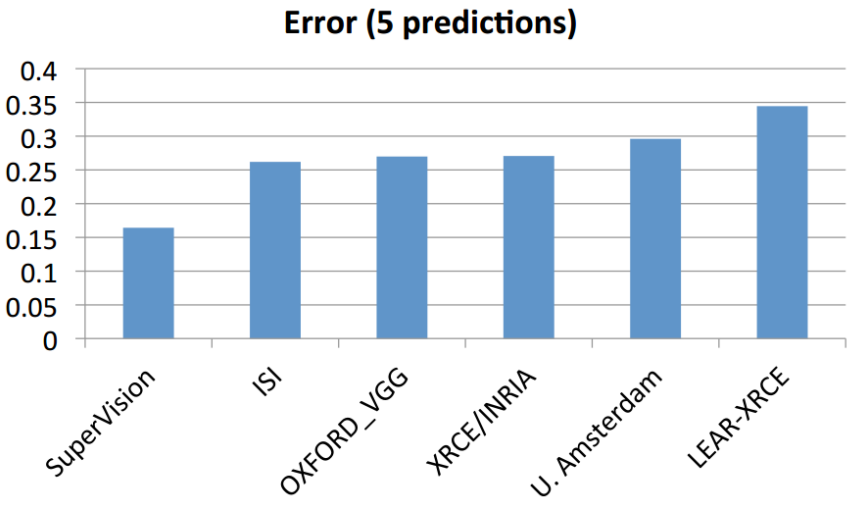
MULTITOUCH DISPLAYS

YOU KNOW WHAT'S COMING...

- *Los Angeles, year 2029. All stealth bombers are upgraded with neural processors, becoming fully unmanned. One of them, Skynet, begins to learn at a geometric rate. It becomes self-aware at 2:14 a.m. eastern time, August 29.*
- ...

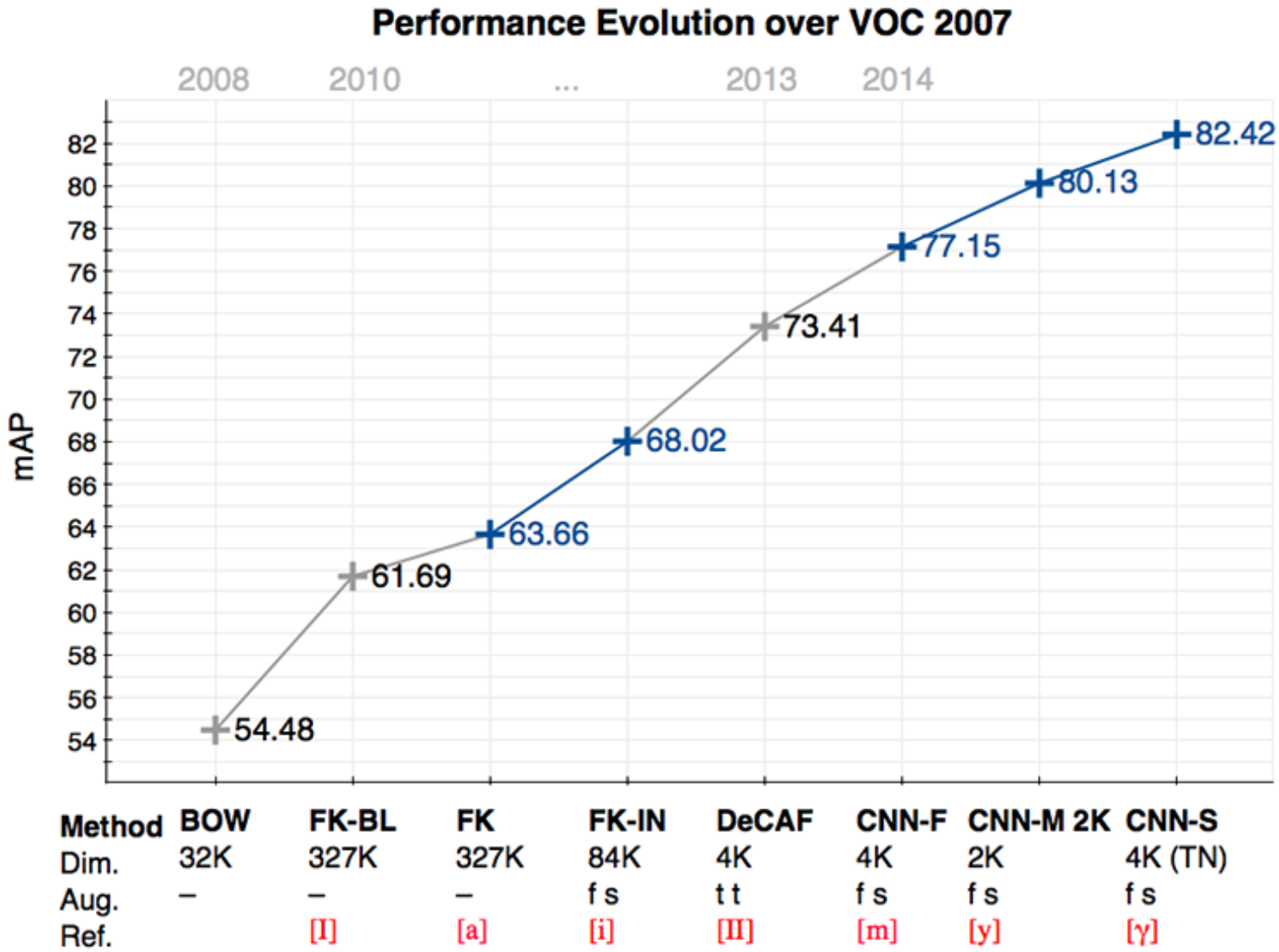
RISE OF THE CONVOLUTIONAL NEURAL NETWORKS

- From BOVW to the FV, we have been considering handcrafted techniques.
- Recently, these handcrafted approaches have been substantially outperformed by the introduction of the latest generation of Convolutional Neural Networks (CNNs) to the computer vision field.
- In the latest instances of the IMAGENET Large Scale Visual Recognition Challenge (ILSVRC) this is a constant. In ILSVRC 2012 the gap was like this:



- In ILSVRC 2013 all teams used CNNs.

STATE OF THE ART



THE END OF OTHER TECHNIQUES?

- In parallel to ILSVRC 2013 a specific challenge was the Fine-Grained Challenge 2013 (FGComp 2013).

Team	**Additional Details**	**Overall**
Inria-Xerox		77.0712
CafeNet*	Results of all 5 verticals using bounding boxes at test time.	75.8178
Inria-Xerox		73.1811
VisionMetric*	Distance metric learning with combo features	71.7211
Symbiotic		71.6323
Inria-Xerox		70.0655
CognitiveVision*	Using ILSVRC-2012 training set (~1.3M images) to pre-training the DNN.	69.9979
DPD_Berkeley*	Track 1 results on test data using the method described above.	69.1615
VisionMetric	Distance metric learning with LLC features	64.1091
CognitiveVision	Results by the method described in the abstract.	57.6757
MPG	SIFT, RGB-SIFT, Opponent-SIFT, C-SIFT. Fisher Vector with 256 Gaussians, 8 regions. Logistic regression classifiers.	52.9257
MPG	Different configurations for dogs and cars data (single scale)	43.818
Infor_FG*	DCNN	15.9907
InterfAIce	IGBA v1.2, 4 cycles, 4h20min	4.48281

THE END OF OTHER TECHNIQUES?

- In parallel to ILSVRC 2013 a specific challenge was the Fine-Grained Challenge 2013 (FGComp 2013).

Team	**Additional Details**	**Overall**
Inria-Xerox		77.0712
CafeNet*	Results of all 5 verticals using bounding boxes at test time.	75.8178
Inria-Xerox		73.1811
VisionMetric*	Distance metric learning with combo features	71.7211
Symbiotic		71.6323
Inria-Xerox		70.0655

Fisher Vectors

CNNs

- Details available in Gosselin *et al.* 2014, INRIA Technical Report no 8431

CONCLUSIONS

- Probably, it's too early to forget all BOVW techniques.

- Nevertheless, Convolutional Neural Networks are something which every Computer Vision researcher must add to their knowledge.

- We made a short journey of about 10 years in image classification and concept detection techniques, and it's pretty much clear that a lot of lessons have been learned and some techniques become part of the standard pipeline.

- Implementation details are important and sometimes make the difference between success and failure of a recognition system.

THANK YOU!

REFERENCES

1. J. Sivic, and A. Zisserman, "Video Google: a text retrieval approach to object matching in videos," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2003, pp. 1470-1477.

2. G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Statistical Learning in Computer Vision Workshop*, 2004, pp. 1–12.

3. L. Fei-Fei, and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005, pp. 524-531.

4. K. Grauman, and T. Darrell, "The pyramid match kernel: discriminative classification with sets of image features," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005, pp. 1458-1465.

5. S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2169–2178.

6. A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *ACM International Conference on Image and Video Retrieval*, 2007, pp. 401–408.

7. F. Perronnin, and C. Dance, "Fisher Kernels on Visual Vocabularies for Image Categorization," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2007, pp.1-8.

REFERENCES

8. J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. Smeulders, "Kernel Codebooks for Scene Categorization," in *European Conference on Computer Vision*, 2008, pp. 696–709.

9. L. Bo and C. Sminchisescu, "Efficient Match Kernel between Sets of Features for Visual Recognition," in *Neural Information Processing Systems*, 2009, pp. 135–143.

10. H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3304–3311.

11. F. Perronnin, J. S´anchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *European Conference on Computer Vision*, 2010, pp. 143–156.

12. K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *British Machine Vision Conference*, 2011, pp. 76.1–76.12.

13. R. Arandjelovic, and A. Zisserman, "Three things everyone should know to improve object retrieval," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2911-2918.

14. J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Lost in quantization: Improving particular object retrieval in large scale image databases, in*: IEEE International Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1-8.

REFERENCES

15. J. Farquhar, S. Szedmak, H. Meng, J. Shawe-Taylor, Improving bag-ofkeypoints "image categorisation: Generative Models and PDF-Kernels", Tech. rep., University of Southampton (2005).

16. O. Tuzel, F. Porikli, P. Meer, "Pedestrian Detection via Classification on Riemannian Manifolds", *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (10) (2008) 1713-1727.

17. G. Serra, C. Grana, M. Manfredi, R. Cucchiara, "Modeling Local Descriptors with Multivariate Gaussians for Object and Scene Recognition", in *ACM International Conference on Multimedia*, Barcelona, Spain, pp. 709-712, Oct. 21-25, 2013.

18. Xavier Pennec , Pierre Fillard , Nicholas Ayache, "A Riemannian Framework for Tensor Computing" in *International Journal of Computer Vision* 66(1) (2006) 41-66.

19. K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the Devil in the Details: Delving Deep into Convolutional Nets," *Technical Report*, arXiv:1405.3531v3

20. P.-H. Gosselin, N. Murray, H. Jégou and F. Perronnin, "Revisiting the Fisher vector for fine-grained classification," in *Pattern Recognition Letters*, 2014, to appear.