Features for Large-Scale Visual Recognition

Florent Perronnin
Xerox Research Centre Europe

CVPR tutorial: Large-Scale Visual Recognition (LSVR)
June 23, 2013

Image description

Goal: convert an image into a mathematical representation such that

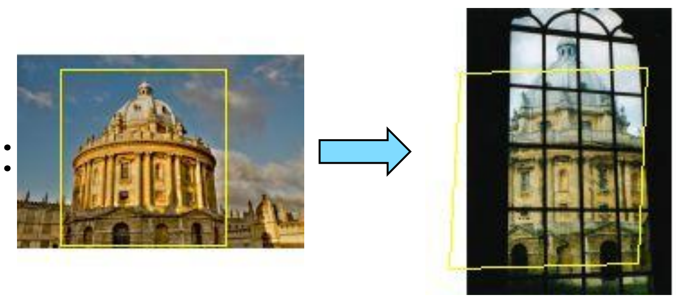
- “similar” images have similar representations
- “dissimilar” images have dissimilar representations

Difficulty: robustness to viewpoint, lighting, occlusion, intra-class variability, etc.

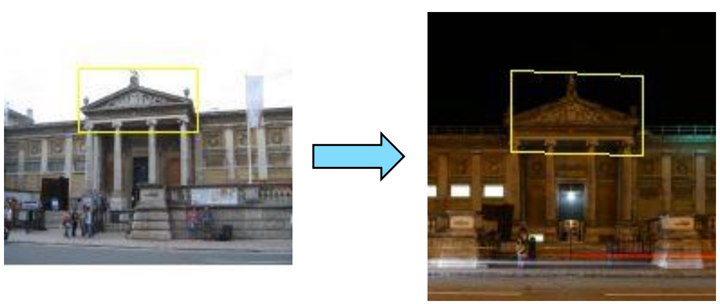
$\rightarrow$ need **invariant representation**

scale:

occlusion:



lighting:



view-point:

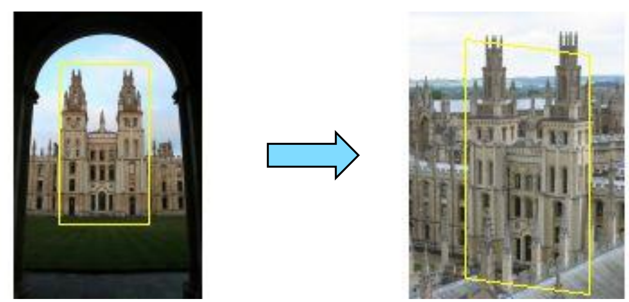


Image description

Goal: convert an image into a mathematical representation such that

- “similar” images have similar representations
- “dissimilar” images have dissimilar representations

But the representation should be **informative** enough:

Image description

Goal: convert an image into a mathematical representation such that

- “similar” images have similar representations
- “dissimilar” images have dissimilar representations

And it should be **efficient**:

- to compute
- to store in RAM / on disk, to transfer, etc.
- to process: e.g. fast comparison between images or between image and class model

Image description

Goal: convert an image into a mathematical representation such that

- "similar" images have similar representations
- "dissimilar" images have dissimilar representations

Trade-off between three conflicting requirements:

- robust to variations: scale, occlusion, lighting, etc.
- informative
- efficient: to compute, store, process

→ **trade-off is application-dependent**

Image description

Caveat: **there is no clear cut between description and learning!**

Better description can lead to simpler learning:

→ **see A. Vedaldi's part on explicit feature maps**

Features and classifiers learned jointly

→ **see M'A Ranzato's part on large-scale deep learning**

In this part: focus on features which are obtained through the **aggregation/pooling** of local **codes/statistics**

Outline

Global vs local descriptors

The bag-of-visual-words

Higher-order representations

Examples

Conclusion

Outline

Global vs local descriptors

The bag-of-visual-words

Higher-order representations

Examples

Conclusion

Global descriptors (of pixel statistics)

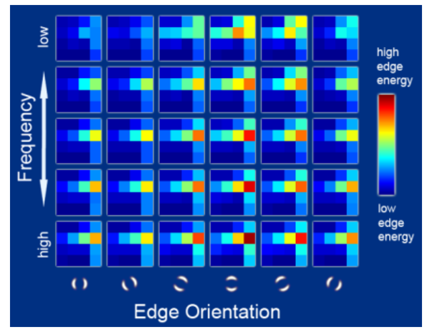
Color Histogram: high invariance but limited discriminative power

Swain, Ballard, “Color indexing”, IJCV’91.

GIST of a scene:

Oliva, Torralba, “Modeling the shape of the scene: a holistic representation of the spatial envelope”, IJCV’01.

Douze, Jegou, Sandhawalia, Amsaleg, Schmid, “Evaluation of GIST descriptors for web-scale image search”, CIVR’09.



CENTRIST: CENsus Transform hISTogram

Wu, Rehg, “CENTRIST: a visual descriptor for scene categorization”, TPAMI’11.

Highly efficient to compute and to match → **perfect for LSVR**

But **robustness vs informativeness tradeoff is hard to set**

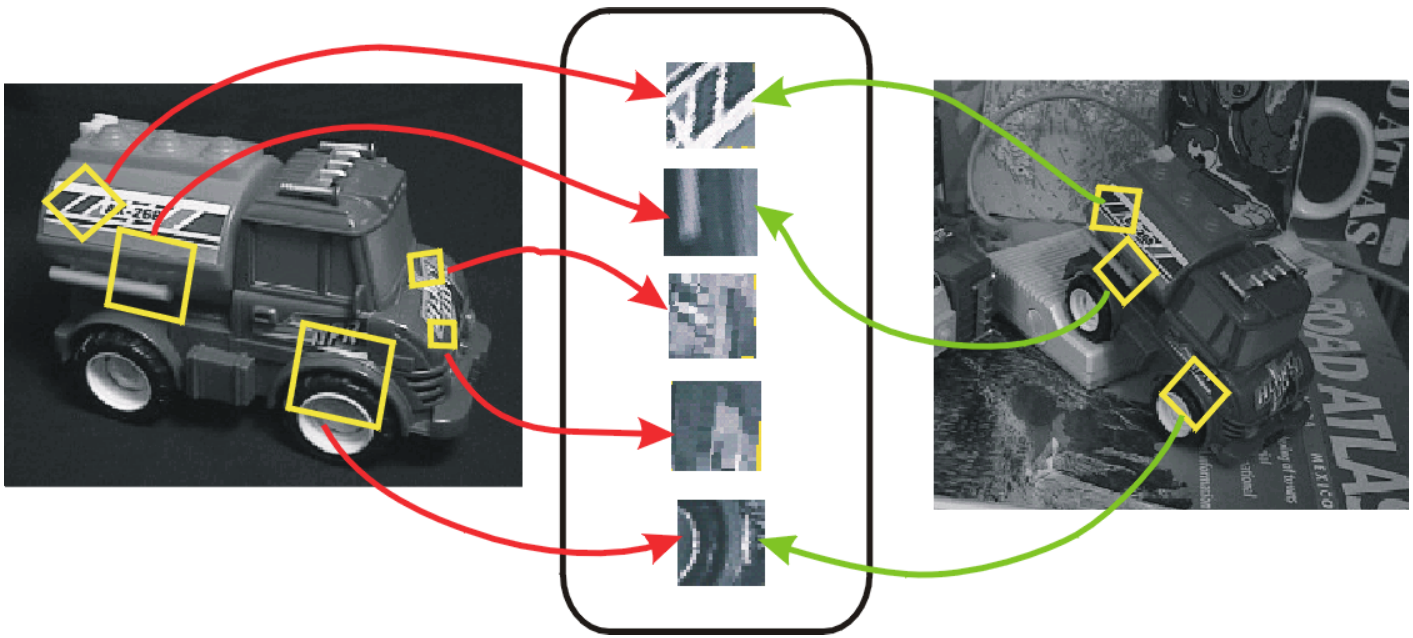
Local representations

Image content is transformed into a set of invariant descriptors (to photometric/geometric transformations) extracted from small image patches

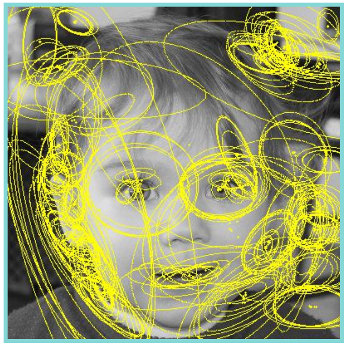
Very intuitive in retrieval / matching context:

Schmid, Mohr, "Local greyvalue invariants for image retrieval", TPAMI'97.

Lowe, "Distinctive image features from scale-invariant keypoints", IJCV'04.



Local representations: detectors



Roles of the detector:

- provide invariance to transformations
- **reduce the number of descriptors**

Popular detectors:

- Maximally Stable Extremal Regions (MSER)
  Matas, Chum, Urban, Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions", BMVC'02.
- Difference of Gaussians (DoG)
  Lowe, "Distinctive image features from scale-invariant keypoints", IJCV'04.
- Harris-Affine and Hessian-Affine
  Mikolajczyk, Schmid, "Scale and affine invariant interest point detectors", IJCV'04.

→ See also Mikolajczyk et al., "A comparison of affine region detectors", IJCV'05.

Dense descriptors are also possible

- Mainly for classification → let the classifier decide
  Leung, Malik, "Representing and recognizing the visual appearance of materials using 3D textons", IJCV'01.
- But also for image/scene/object retrieval
  Gordo, Rodriguez, Perronnin, Valveny, "Leveraging category-level labels for instance-level image retrieval", CVPR'12.

Local representations: descriptors

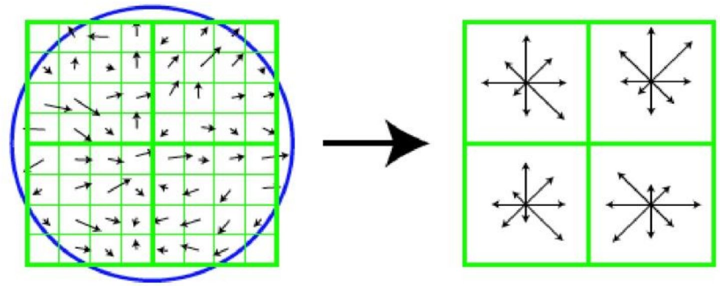
Description of a patch after orientation/scale/photometric normalization

Most widely-used patch descriptor: SIFT

Lowe, "Distinctive image features from scale-invariant keypoints", IJCV'04.

- 8 orientations of the gradient
- 4x4 spatial grid

$\rightarrow$ 128 dimensions



Many descriptors derive from SIFT:

- **More efficient: SURF**
  Bay, Tuytelaars, Van Gool, "SURF: speeded up robust features", ECCV'06.
- **More compact: CHOG, DAISY**
  Chandrasekhar et al, "Compressed histograns of gradients: a low-bit rate descriptor", IJCV'11.
  Tola, Lepetit, Fua, "DAISY: an efficient dense descriptor applied to wide baseline stereo", TPAMI'10.
- With color: color SIFT
  Van de Weijer, Schmid, "Coloring local feature extraction", ECCV'06.
  Burghouts and Geseborek, "Performance evaluation of local colour invariants", CVIU'09.

Outline

Global vs local descriptors

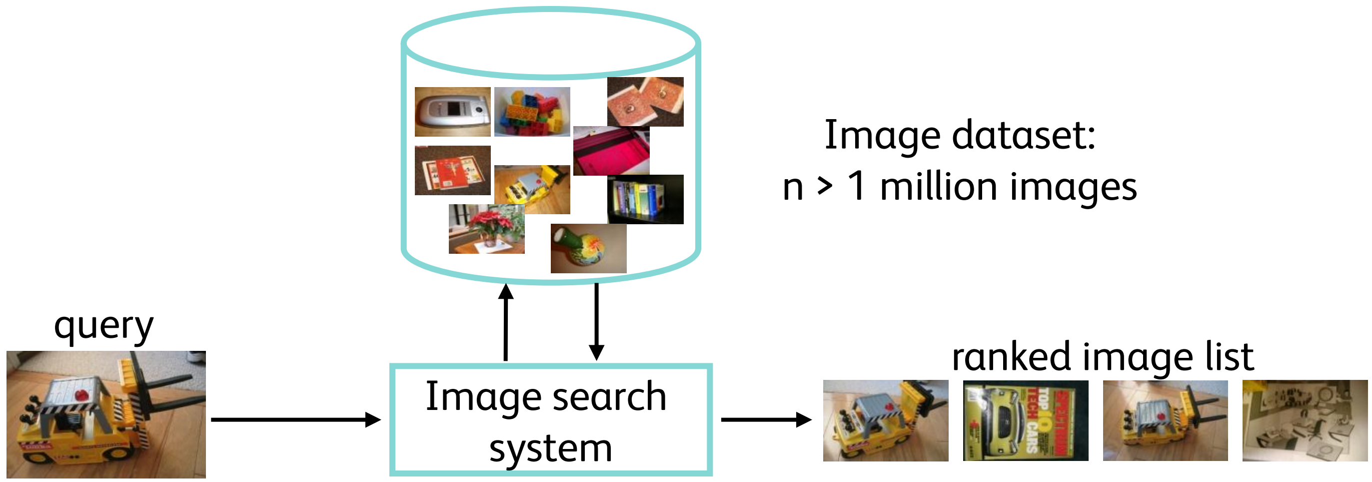
The bag-of-visual-words

Higher-order representations

Examples

Conclusion

Direct matching: a retrieval example



Assume an image described by m=1000 descriptors (dimension d=128)

- n*m=1 billion descriptors to index

Database representation in RAM: 128 GB with 1 byte per dimension

Search: $\mathbf{m^2}$ x n x d elementary operations

- $10^{14} \rightarrow$ **computationally intractable**
- The quadratic term m^2: severely impacts the efficiency

The bag-of-visual-words (BOV)

Concurrently introduced in image search and classification:

- in image search: “Video Google”

  Sivic, Zisserman, “Video Google: A Text Retrieval Approach to Object Matching in Videos”, ICCV’03.

- in image classification:

  Csurka, Dance, Fan, Willamowski, Bray, “Visual categorization with bag of keypoints”, ECCV SLCV’04.

  See also: Zhang, Marszalek, Lazebnik, Schmid, “Local features and kernels for classification of textures and object categories: a comprehensive study”, IJCV’07.

Key idea: **aggregate** n local descriptors into 1 vector

- inherits invariance of the local descriptors
- (possibly) sparse vector $\rightarrow$ efficient comparison

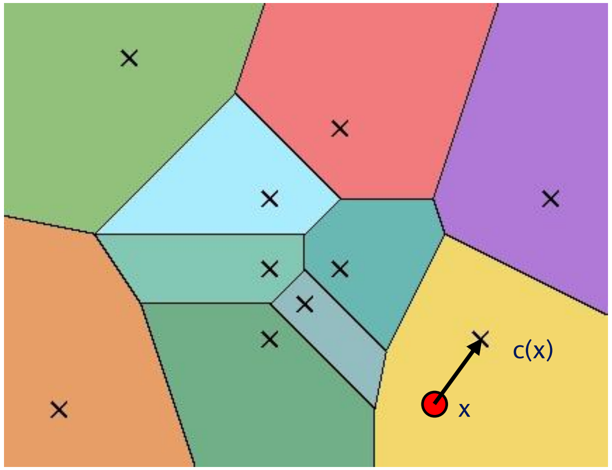
The bag-of-visual-words (BOV)

The goal: "put the images into words", namely **visual words**

- input local descriptors are continuous
- need to define what a visual word is
- done by a **quantizer**: q: $\mathbb{R}^d \rightarrow \omega$

$$x \rightarrow c(x) \in \omega$$

- q is typically a k-means



ω is called a **visual dictionary**

- A local descriptor is assigned to its nearest neighbor

$$q(x) = \arg\min_{w \in \omega} \|x-w\|^2$$

- Quantization is lossy: we cannot get back to the original descriptor
- But much more compact (few bytes per descriptor)

BOV and retrieval

Video Google system

Extract local descriptors

- Detector
- Describe the patch

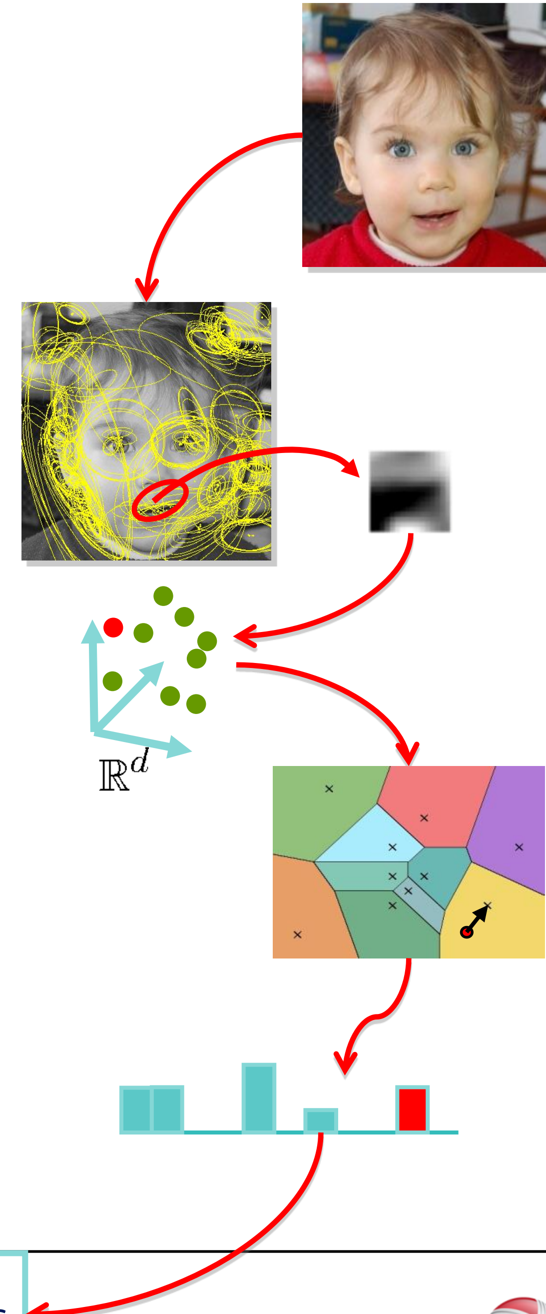
Quantize all descriptors

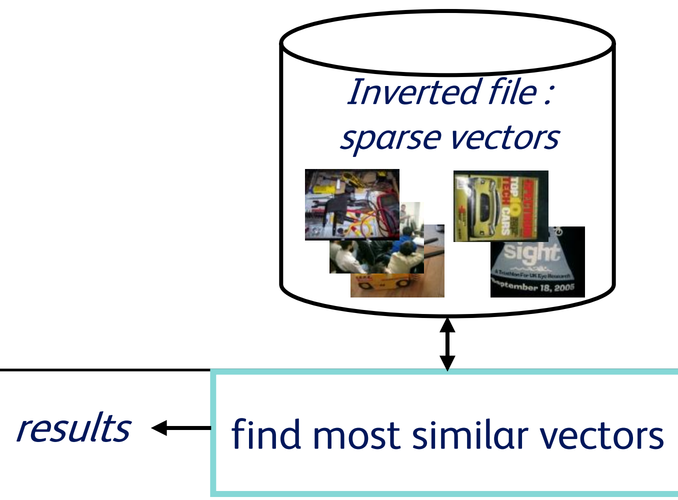
- Subsequently compute the vector of frequencies
- Weight by IDF (rare if more important) → TF-IDF vectors

Search similar vectors

Optionally: re-ranking

→ see O. Chum's part on large-scale geometry



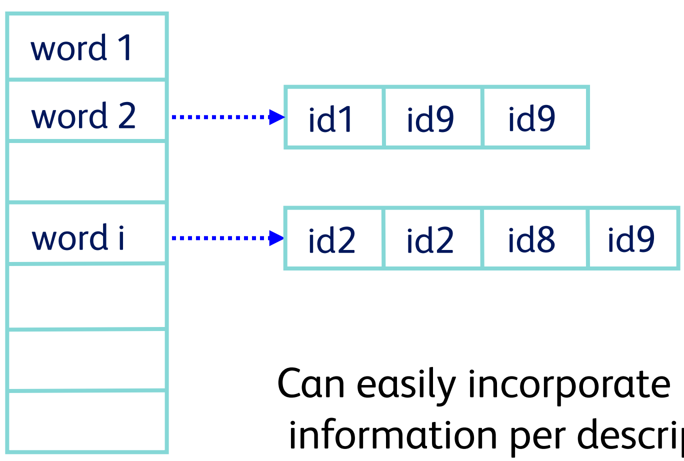


BOV and retrieval
Efficiency through inverted files

Set of lists that store the **sparse vector components**
→ useful if # descriptors << # visual words (retrieval)

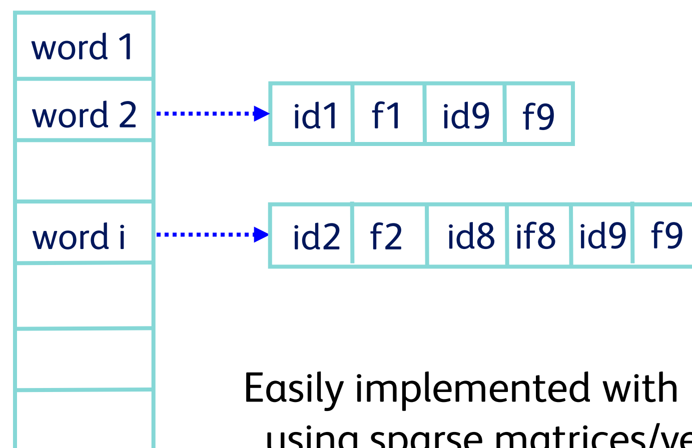
Two implementations:

- store one image id per descriptor:



Can easily incorporate meta information per descriptor (geometry, bundled features, etc)

- store image id+nb of descriptors:



Easily implemented with Matlab using sparse matrices/vectors

→ **histogram representation**

BOV and classification
Coding and pooling

Coding: how to go beyond VQ + hard coding?

- soft coding, e.g. using a mixture model or a “kernel” codebook
  Winn, Criminisi, Minka, “Object categorization by learned universal visual dictionary”, ICCV’05.
  Perronnin, Dance, Csurka, Bressan, “Adapted vocabularies for generic visual categorization”, ECCV’06.
  van Gemert, Geusebroek, Veenman, Smeulders, “Kernel codebooks for scene categorization”, ECCV’08.
- sparse coding
  Yang, Yu, Gong, Huang, “Linear spatial pyramid matching using sparse coding for image classification”, CVPR’09.
  Wang, Yang, Yu, Lv, Huang, Gong, “Locality-constrained linear coding for image classification”, CVPR’10.
  Boureau, Bach, LeCun, Ponce, “Learning mid-level features for reognition”, CVPR’10.

Pooling / aggregation:

- average pooling
  Csurka, Dance, Fan, Willamowski, Bray, “Visual categorization with bag of keypoints”, ECCV SLCV’04.
- max pooling
  Yang, Yu, Gong, Huang, “Linear spatial pyramid matching using sparse coding for image classification”, CVPR’09.
  Boureau, Ponce, LeCun, “A theoretical analysis of feature pooling in vision algorithms”, ICML’10.
- Lp pooling
  Boureau, Ponce, LeCun, “A theoretical analysis of feature pooling in vision algorithms”, ICML’10.

BOV and classification
Choice of classifier

BOV histograms are generally used together with **kernel classifiers**

Linear kernel classifiers:

- fast to learn and evaluate

→ **see Z. Harchaoui's part on large-scale learning**

- perform poorly on the BOV (at least with average pooling)

Non-linear kernel classifiers:

- perform well on the BOV (chi2 or intersection kernel)
- direct approach leads to slow learning and evaluation

→ **see A. Vedaldi's part on explicit feature maps**

Visual vocabulary size

For LSVR, we need image signatures which contain **fine-grained information**:

- in retrieval: the larger the dataset size, the higher the probability to find another similar but irrelevant image to a given query
- in classification: the larger the number of other classes, the higher the probability to find a class which is similar to any given class

BOV answer to the problem: increase visual vocabulary size

- practical problem: assignment of descriptors to visual words becomes costly

→ **see H. Jégou's part on efficient matching**

How to increase amount of information
without increasing the visual vocabulary size?

→ higher-order representations

Outline

Global vs local descriptors

The bag-of-visual-words

Higher-order representations

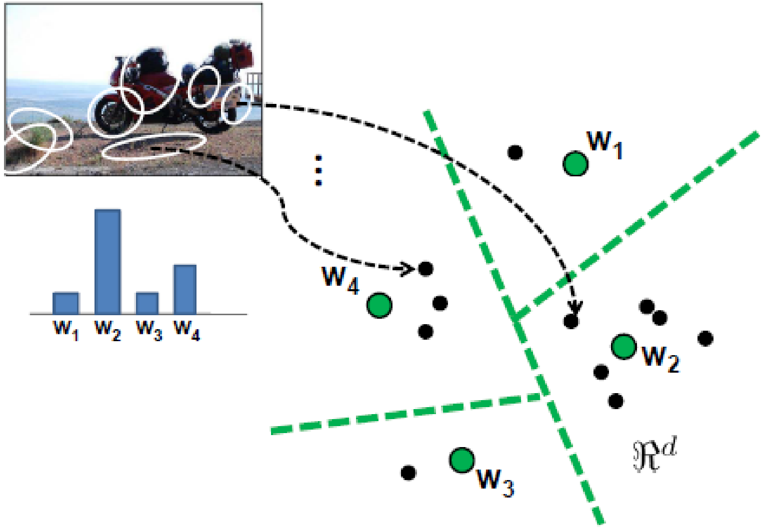
Examples

Conclusion

Motivation

BOV is only about **counting** the number of local descriptors assigned to each Voronoi region

Why not including **other statistics**?



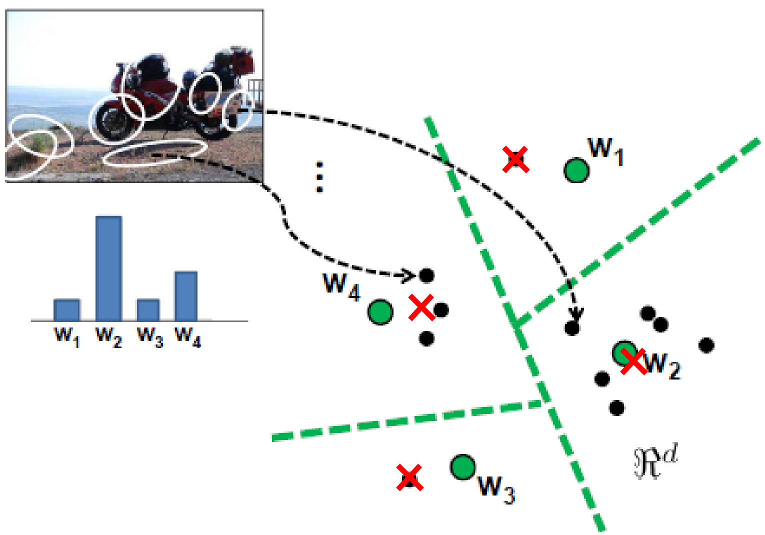
http://www.cs.utexas.edu/~grauman/courses/fall2009/papers/bag_of_visual_words.pdf

Motivation

BOV is only about **counting** the number of local descriptors assigned to each Voronoi region

Why not including **other statistics**? For instance:

- mean of local descriptors ×



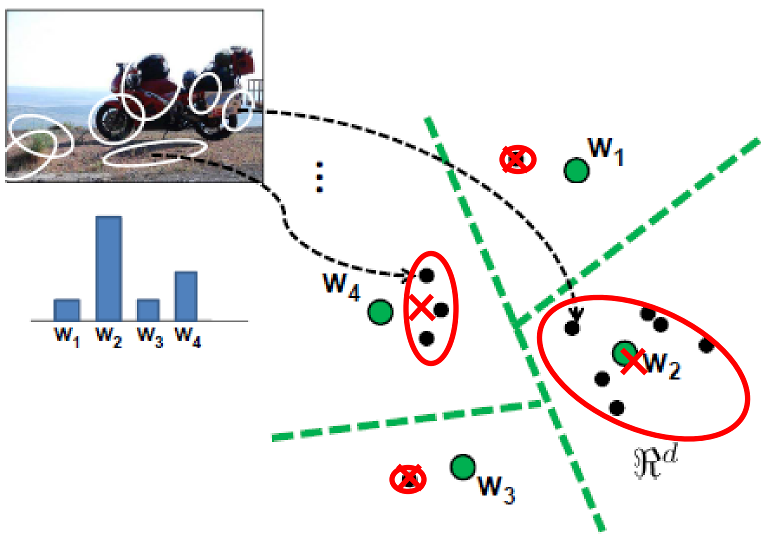
http://www.cs.utexas.edu/~grauman/courses/fall2009/papers/bag_of_visual_words.pdf

Motivation

BOV is only about **counting** the number of local descriptors assigned to each Voronoi region

Why not including **other statistics**? For instance:

- mean of local descriptors ×
- (co)variance of local descriptors



http://www.cs.utexas.edu/~grauman/courses/fall2009/papers/bag_of_visual_words.pdf

Motivation

BOV is only about **counting** the number of local descriptors assigned to each Voronoi region

Why not including **other statistics**? For instance:

- mean of local descriptors
- (co)variance of local descriptors

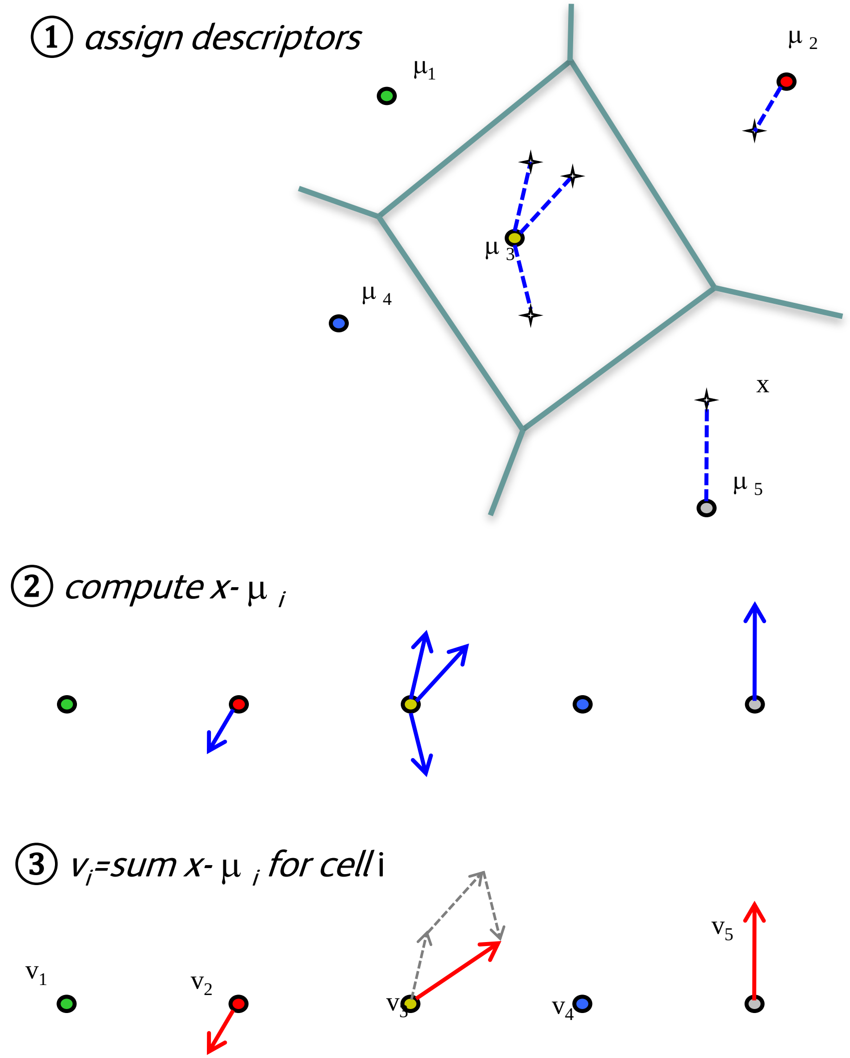
Model the approximate distribution of samples in each cell
$\rightarrow$ **aggregate higher order statistics**

The Vector of Locally Aggregated Descriptors (VLAD)

Given a codebook $\{\mu_i, i = 1 \ldots N\}$, e.g. learned with K-means, and a set of local descriptors $X = \{x_t, t = 1 \ldots T\}$:

- ① assign: $\mathrm{NN}(x_t) = \arg\min_{\mu_i} ||x_t - \mu_i||$
- ②③ compute: $v_i = \sum_{x_t : \mathrm{NN}(x_t) = \mu_i} x_t - \mu_i$
- concatenate v_i's + ℓ_2 normalize

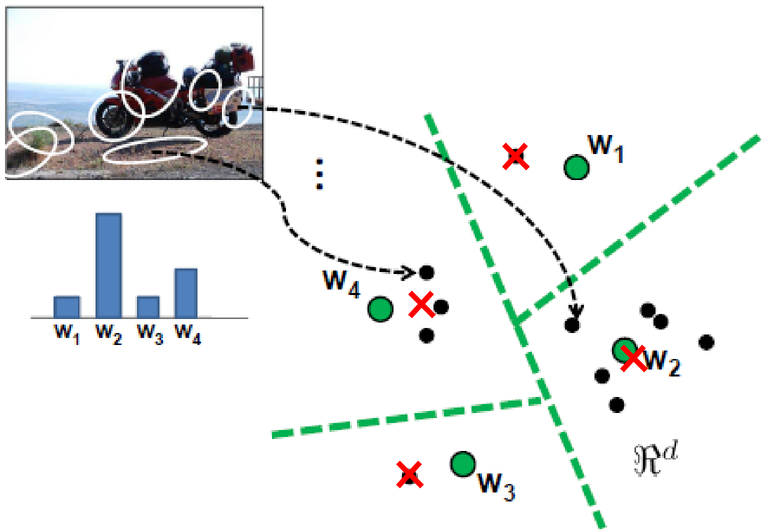
$\rightarrow$ the VLAD **is DxN dimensional**



Jégou, Douze, Schmid and Pérez, "Aggregating local descriptors into a compact image representation", CVPR'10.

The Vector of Locally Aggregated Descriptors (VLAD)

The distribution of samples in each cell is encoded by its mean (minus the centroid mean)



http://www.cs.utexas.edu/~grauman/courses/fall2009/papers/bag_of_visual_words.pdf

→ The VLAD is a special case of a more general descriptor: the Fisher Vector

The Fisher Vector (FV)

Score function

Given a likelihood function u_λ with parameters λ, the **score function** of a given sample X is given by:

$$\boxed{G_\lambda^X = \nabla_\lambda \log u_\lambda(X)}$$

$\rightarrow$ Fixed-length vector whose **dimensionality depends only on # parameters**.

Intuition: direction in which the parameters λ of the model should we modified to better fit the data.

The Fisher Vector (FV)

Fisher Kernel

Fisher information matrix (FIM) or negative Hessian:

$$F_\lambda = E_{x \sim u_\lambda} \left[\nabla_\lambda \log u_\lambda(x) \nabla_\lambda \log u_\lambda(x)' \right]$$

Measure similarity between gradient vectors using the **Fisher Kernel (FK)**:

$$\boxed{K(X,Y) = G_\lambda^{X\prime} F_\lambda^{-1} G_\lambda^Y}$$

Jaakkola and Haussler, "Exploiting generative models in discriminative classifiers", NIPS'98.

$\rightarrow$ can be interpreted as a score whitening

As the FIM, is PSD, it can be decomposed as: $F_\lambda^{-1} = L_\lambda' L_\lambda$

and the FK can be rewritten as a dot product between **Fisher Vectors** (FV):

$$\boxed{\mathcal{G}_\lambda^X \quad = \quad L_\lambda G_\lambda^X}$$

xerox

The Fisher Vector (FV)

Application to images

$X = \{x_t, t = 1 \ldots T\}$ is the set of T i.i.d. D-dim local descriptors (e.g. SIFT) extracted from an image:

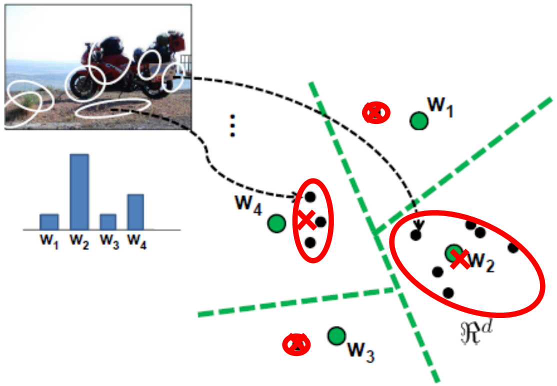
$$\boxed{G_\lambda^X = \frac{1}{T} \sum_{t=1}^{T} \nabla_\lambda \log u_\lambda(x_t)}$$

$u_\lambda(x) = \sum_{i=1}^{K} w_i u_i(x)$ is a Gaussian Mixture Model (GMM) with parameters $\lambda = \{w_i, \mu_i, \Sigma_i, i = 1 \ldots N\}$

The FV is typically 2xDxN dimensional

With respect to the VLAD:

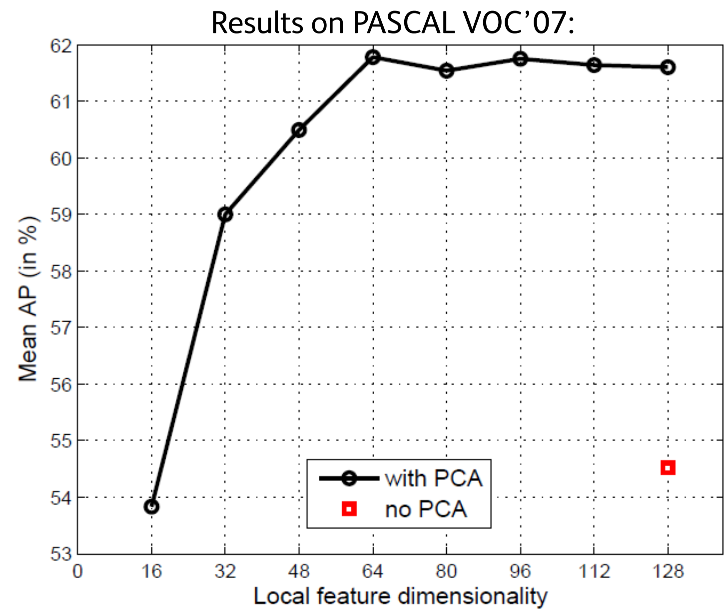
- add 2nd order moments
- soft-assignment of local descriptors
- per-dimension whitening



Perronnin and Dance, “Fisher kernels on visual categories for image categorization”, CVPR’07.

The Fisher Vector (FV)

Practical considerations



PCA on the local descriptors is necessary:

- because of the GMM diagonal approximation

ℓ_2 -normalization:

- to make the FV more compliant with the dot-product assumption

Power-normalization: $\boxed{f(z) = \text{sign}(z)|z|^\alpha \text{ with } 0 \leq \alpha \leq 1}$

- to correct the patch independence assumption

  Cinbis, Verbeek, Schmid, "Image categorization using Fisher kernels of non-iid image models", CVPR'12.

$\rightarrow$ For a detailed analysis see: Sánchez, Perronnin, Mensink, Verbeek, "Image Classification with the Fisher Vector: Theory and Practice", IJCV'13.

Embedding view of the BOV and FV

BOV: $\varphi_{BOV}(x_t) = [0, \ldots, 0, 1, 0, \ldots, 0]$

FV: $$\varphi_{FV}(x_t) = \left[0, \ldots, 0, \overbrace{\frac{1}{\sqrt{w_i}} \left(\frac{x_t - \mu_i}{\sigma_i} \right), \frac{1}{\sqrt{2w_i}} \left(\frac{(x_t - \mu_i)^2}{\sigma_i^2} - 1 \right)}^{2D \text{ non-zero dim}}, 0, \ldots, 0 \right]$$

A linear classifier on these representations induces in the descriptor space:

Csurka and Perronnin, “An efficient approach to semantic segmentation”, IJCV ‘10.

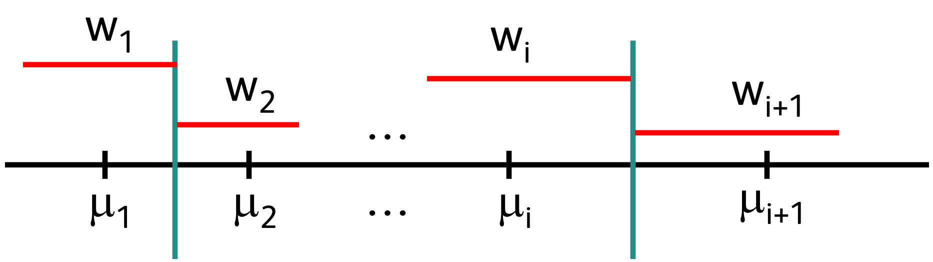
Embedding view of the BOV and FV

BOV: $\varphi_{BOV}(x_t) = [0, \ldots, 0, 1, 0, \ldots, 0]$

FV: $$\varphi_{FV}(x_t) = \left[0, \ldots, 0, \overbrace{\frac{1}{\sqrt{w_i}} \left(\frac{x_t - \mu_i}{\sigma_i} \right), \frac{1}{\sqrt{2 w_i}} \left(\frac{(x_t - \mu_i)^2}{\sigma_i^2} - 1 \right)}^{2D \text{ non-zero dim}}, 0, \ldots, 0 \right]$$

A linear classifier on these representations induces in the descriptor space:

- in the BOV case: a piece-wise constant decision function



Csurka and Perronnin, "An efficient approach to semantic segmentation", IJCV '10.

xerox

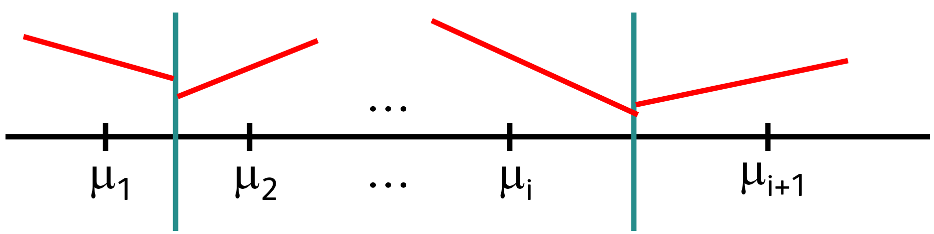
Embedding view of the BOV and FV

BOV: $\varphi_{BOV}(x_t) = [0, \ldots, 0, 1, 0, \ldots, 0]$

FV: $$\varphi_{FV}(x_t) = \left[0, \ldots, 0, \overbrace{\frac{1}{\sqrt{w_i}} \left(\frac{x_t - \mu_i}{\sigma_i} \right), \frac{1}{\sqrt{2w_i}} \left(\frac{(x_t - \mu_i)^2}{\sigma_i^2} - 1 \right)}^{2D \text{ non-zero dim}}, 0, \ldots, 0 \right]$$

A linear classifier on these representations induces in the descriptor space:

- in the BOV case: a piece-wise constant decision function
- in the FV case: a piecewise linear / quadratic decision function



Csurka and Perronnin, “An efficient approach to semantic segmentation”, IJCV ‘10.

Embedding view of the BOV and FV

BOV: $\varphi_{BOV}(x_t) = [0, \ldots, 0, 1, 0, \ldots, 0]$

FV: $\varphi_{FV}(x_t) = \left[0, \ldots, 0, \overbrace{\frac{1}{\sqrt{w_i}} \left(\frac{x_t - \mu_i}{\sigma_i} \right), \frac{1}{\sqrt{2 w_i}} \left(\frac{(x_t - \mu_i)^2}{\sigma_i^2} - 1 \right)}^{2D \text{ non-zero dim}}, 0, \ldots, 0 \right]$

A linear classifier on these representations induces in the descriptor space:

- in the BOV case: a piece-wise constant decision function
- in the FV case: a piecewise linear / quadratic decision function

→ **FV leads to more complex decision functions for same vocabulary size**

Csurka and Perronnin, "An efficient approach to semantic segmentation", IJCV '10.

Super-Vector (SV) coding

Given a codebook $\{\mu_i, i = 1 \dots N\}$ and a patch x_t we have:

$$f(x_t) \approx f(\mu_i) + \nabla f(\mu_i)'(x_t - \mu_i) = w' \varphi_{SV}(x_t)$$

with $\varphi_{SV}(x_t) = \left[0, \dots, 0, \overbrace{s, (x_t - \mu_i)}^{(D+1) \text{ non-zero dim}}, 0, \dots, 0 \right]$

and $w = \left[0, \dots, 0, \frac{f(\mu_i)}{s}, \nabla f(\mu_i), 0, \dots, 0 \right]$ (to be learned)

Zhou, Yu, Zhang and Huang, "Image classification using super-vector coding of local image descriptors", ECCV'10.

Super-Vector (SV) coding

Given a codebook $\{\mu_i, i = 1 \ldots N\}$ and a patch x_t we have:

$$f(x_t) \approx f(\mu_i) + \nabla f(\mu_i)'(x_t - \mu_i) = w'\varphi_{SV}(x_t)$$

with $$\varphi_{SV}(x_t) = \left[0, \ldots, 0, \overbrace{s, (x_t - \mu_i)}^{(D+1) \text{ non-zero dim}}, 0, \ldots, 0 \right]$$

average pooling $\rightarrow$ **SV ≈ BOV + VLAD**

Zhou, Yu, Zhang and Huang, "Image classification using super-vector coding of local image descriptors", ECCV'10.

Super-Vector (SV) coding

Given a codebook $\{\mu_i, i = 1 \ldots N\}$ and a patch x_t we have:

$$f(x_t) \approx f(\mu_i) + \nabla f(\mu_i)'(x_t - \mu_i) = w'\varphi_{SV}(x_t)$$

with $\varphi_{SV}(x_t) = \left[0, \ldots, 0, \overbrace{s, (x_t - \mu_i)}^{(D+1) \text{ non-zero dim}}, 0, \ldots, 0 \right]$

average pooling $\rightarrow$ **SV ≈ BOV + VLAD**

$f : \mathbb{R}^D \rightarrow \mathbb{R}$ is Lipschitz smooth if $\forall (x, y) \in \mathbb{R}^D \times \mathbb{R}^D$:

$$|f(x) - f(y) - \nabla f(y)'(x - y)| \leq \frac{\beta}{2} ||x - y||^2$$

$\rightarrow$ bound in Lipschitz smooth inequality provides argument for k-means.

Zhou, Yu, Zhang and Huang, "Image classification using super-vector coding of local image descriptors", ECCV'10.

Memory issue

Higher-order representations are typically:

- high-dimensional $\rightarrow$ few 100Ks of dims
- dense $\rightarrow$ on the order of 50 % sparsity for the FV

$\rightarrow$ storing a dataset such as ImageNet can take tens of TBs

Solution: **compression**, e.g. with Product Quantization (PQ)

Jégou, Perronnin, Douze, Sánchez, Pérez and Schmid, "Aggregating local descriptors into compact codes", TPAMI'11.
Sánchez and Perronnin, "High-dimensional signature compression for large-scale image classification", CVPR'11.

$\rightarrow$ see H. Jégou's part on efficient matching

$\rightarrow$ see also A. Vedaldi's part on how to combine compression and learning

Outline

Global vs local descriptors

The bag-of-visual-words

Higher-order representations

Examples

Conclusion

Examples
Retrieval

Example on Holidays:

From: Jégou, Perronnin, Douze, Sánchez, Pérez and Schmid, “Aggregating local descriptors into compact codes”, TPAMI’11.

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

Examples

Retrieval

Example on Holidays:

From: Jégou, Perronnin, Douze, Sánchez, Pérez and Schmid, “Aggregating local descriptors into compact codes”, TPAMI’11.

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

$\rightarrow$ second order statistics are not essential for retrieval

Examples
Retrieval

Example on Holidays:

From: Jégou, Perronnin, Douze, Sánchez, Pérez and Schmid, “Aggregating local descriptors into compact codes”, TPAMI’11.

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

$\rightarrow$ second order statistics are not essential for retrieval

$\rightarrow$ even for the same feature dim, the FV/VLAD can beat the BOV

Examples

Retrieval

Example on Holidays:

From: Jégou, Perronnin, Douze, Sánchez, Pérez and Schmid, "Aggregating local descriptors into compact codes", TPAMI'11.

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

$\rightarrow$ second order statistics are not essential for retrieval

$\rightarrow$ even for the same feature dim, the FV/VLAD can beat the BOV

$\rightarrow$ soft assignment + whitening of FV helps when number of Gaussians $\uparrow$

Examples

Retrieval

Example on Holidays:

From: Jégou, Perronnin, Douze, Sánchez, Pérez and Schmid, “Aggregating local descriptors into compact codes”, TPAMI’11.

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

$\rightarrow$ second order statistics are not essential for retrieval

$\rightarrow$ even for the same feature dim, the FV/VLAD can beat the BOV

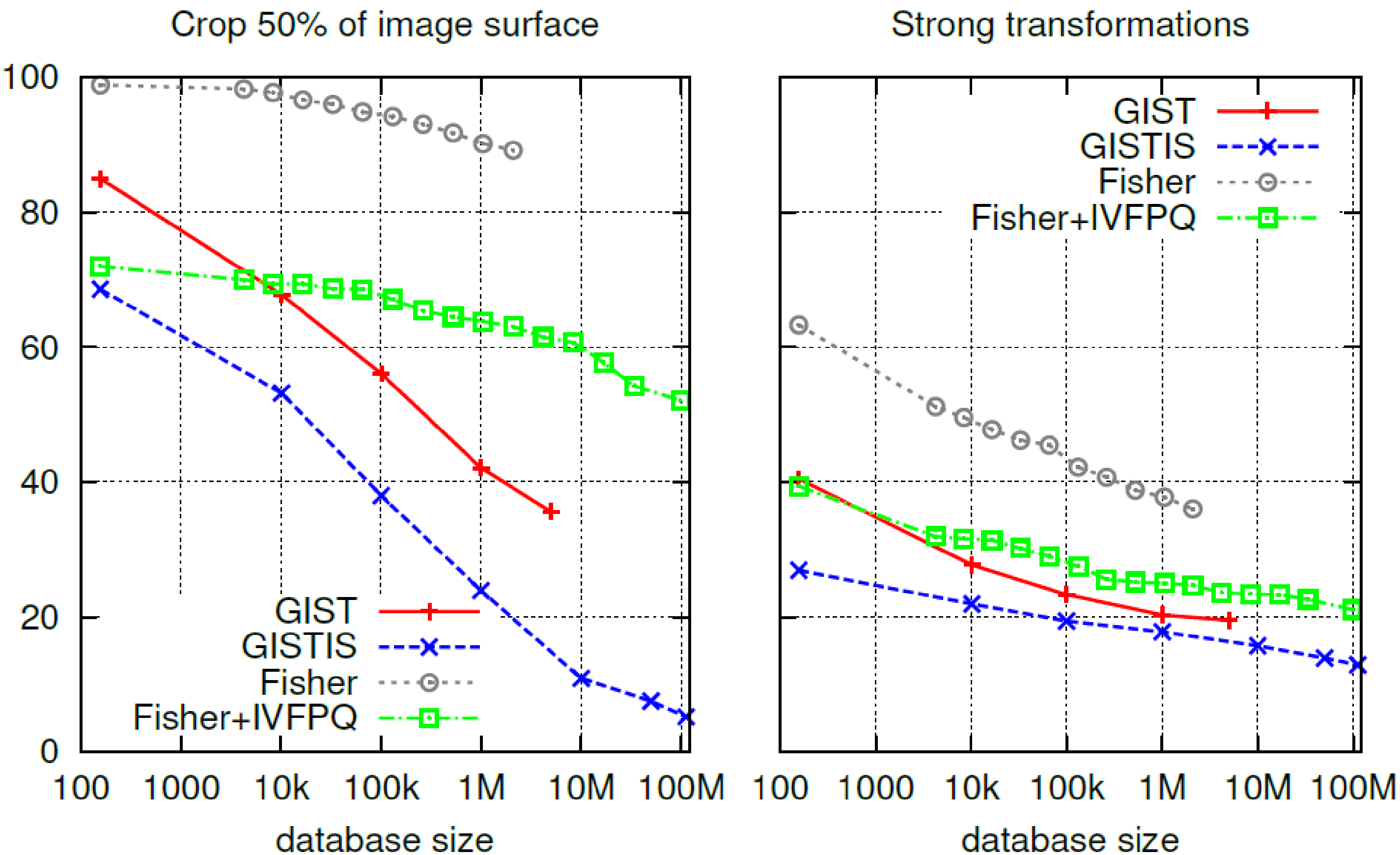
$\rightarrow$ soft assignment + whitening of FV helps when number of Gaussians $\uparrow$

$\rightarrow$ after dim-reduction however, the FV and VLAD perform similarly

Examples

Very large-scale retrieval

Results on INRIA copydays:



Jégou, Perronnin, Douze, Sánchez, Pérez and Schmid, “Aggregating local descriptors into compact codes”, TPAMI’12.

Examples
Classification

Example on PASCAL VOC 2007:

From: Chatfield, Lempitsky, Vedaldi and Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods", BMVC'11.

	Feature dim	mAP
VQ	25K	55.30
KCB	25K	56.26
LLC	25K	57.27
SV	41K	58.16
FV	132K	61.69

Examples
Classification

Example on PASCAL VOC 2007:

From: Chatfield, Lempitsky, Vedaldi and Zisserman, “The devil is in the details: an evaluation of recent feature encoding methods”, BMVC’11.

	Feature dim	mAP
VQ	25K	55.30
KCB	25K	56.26
LLC	25K	57.27
SV	41K	58.16
FV	132K	61.69

→ FV outperforms BOV-based techniques including:

- VQ: plain vanilla BOV
- KCB: BOV with soft assignment
- LLC: BOV with sparse coding

Examples
Classification

Example on PASCAL VOC 2007:

From: Chatfield, Lempitsky, Vedaldi and Zisserman, “The devil is in the details: an evaluation of recent feature encoding methods”, BMVC’11.

	Feature dim	mAP
VQ	25K	55.30
KCB	25K	56.26
LLC	25K	57.27
SV	41K	58.16
FV	132K	61.69

→ FV outperforms BOV-based techniques including:

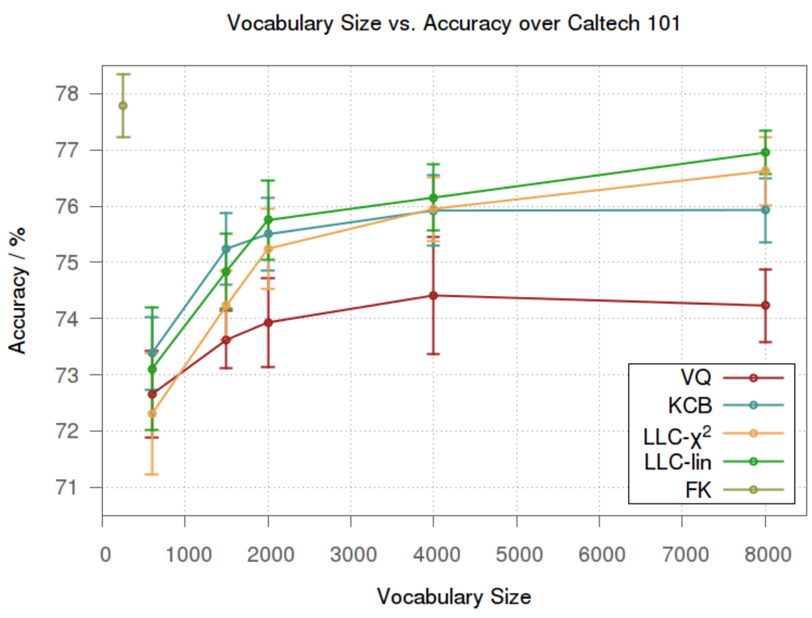
- VQ: plain vanilla BOV
- KCB: BOV with soft assignment
- LLC: BOV with sparse coding

→ including 2nd order information is important for classification

Examples
Classification

Example on CalTech 101:

From: Chatfield, Lempitsky, Vedaldi and Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods", BMVC'11.



→ FV outperforms BOV-based techniques including:

- VQ: plain vanilla BOV
- KCB: BOV with soft assignment
- LLC: BOV with sparse coding

→ including 2nd order information is important for classification

→ with higher-order information, one can get excellent results with tiny vocabularies

Examples

Very large-scale classification

Results on ImageNet10K:

- 10,184 classes (leaves and internal nodes)
- ≈ 9M images: ½ training / ½ test
- accuracy measured as % top-1 correct

Examples

Very large-scale classification

Results on ImageNet10K:

- 10,184 classes (leaves and internal nodes)
- ≈ 9M images: ½ training / ½ test
- accuracy measured as % top-1 correct

SIFT + BOV (21K-dim) + explicit embedding + linear SVM (SGD)

→ accuracy = 6.4 %

→ training time ≈ 6 CPU years

Deng, Berg, Li and Fei-Fei, "What does classifying more than 10,000 image categories tell us?", ECCV'10.

Examples
Very large-scale classification

Results on ImageNet10K:

- 10,184 classes (leaves and internal nodes)
- ≈ 9M images: ½ training / ½ test
- accuracy measured as % top-1 correct

SIFT + BOV (21K-dim) + explicit embedding + linear SVM (SGD)

→ accuracy = 6.4 %

→ training time ≈ 6 CPU years

Deng, Berg, Li and Fei-Fei, "What does classifying more than 10,000 image categories tell us?", ECCV'10.

SIFT + FV (130K-dim) + PQ compression + linear SVM (SGD)

→ accuracy = **19.1%**

→ training time ≈ 1 CPU year (**trick: do not sample all negatives**)

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

Outline

Global vs local descriptors

The bag-of-visual-words

Higher-order representations

Examples

Conclusion

Conclusion

Global descriptors should not be automatically discarded

$\rightarrow$ still useful for such things as near duplicate detection or pre-filtering

Conclusion

Global descriptors should not be automatically discarded

$\rightarrow$ still useful for such things as near duplicate detection or pre-filtering

Among patch-based aggregation techniques, higher-order techniques seem to have an edge.

Conclusion

Global descriptors should not be automatically discarded

→ still useful for such things as near duplicate detection or pre-filtering

Among patch-based aggregation techniques, higher-order techniques seem to have an edge.

The standard SIFT + BOV pipeline can be viewed as a first step toward a deep architecture as it combines multiple layers made of:

- coding
- pooling
- non-linearity

→ see M'A Ranzato's part on large-scale deep learning

Packages

The INRIA package:

http://lear.inrialpes.fr/src/inria_fisher/

The Oxford package:

http://www.robots.ox.ac.uk/~vgg/software/enceval_toolkit/

Questions?