Improved Fisher Vector for Large Scale Image Classification

XRCE's participation for ILSVRC

Jorge Sánchez, Florent Perronnin and Thomas Mensink

Xerox Research Centre Europe (XRCE)

Overview

- Fisher Vector
- Improved FV + results on VOC 07
- Compression
- Classification
- Results on VOC2010 & ILSVRC2010

Fisher Vector

- Exploiting Generative Models in discriminative classifiers [Jaakkola & Haussler 1999]
- Feature vector is derivative wrt probabilistic model
- Measure Similarity using the Fisher Kernel

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

- Fisher Information Matrix

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

- Learning a classifier on Fisher Kernel equals learning a linear classifier on $\boxed{\mathcal{G}_\lambda^X = L_\lambda G_\lambda^X}$ with $F_\lambda = L_\lambda' L_\lambda$

Fisher Vector (2)

- Fisher Kernels on visual vocabularies for image categorization
  [Perronnin & Dance 2007]
- $X = \{x_t, t = 1 \ldots T\}$ D-dimensional local features from an image
- GMM: $\boxed{u_\lambda(x) = \sum_{i=1}^{K} w_i u_i(x)}$
- Gradient:

$$\boxed{\begin{aligned} \mathcal{G}_{\mu,i}^X &= \frac{1}{T\sqrt{w_i}} \sum_{t=1}^{T} \gamma_t(i) \left(\frac{x_t - \mu_i}{\sigma_i} \right), \\ \mathcal{G}_{\sigma,i}^X &= \frac{1}{T\sqrt{2w_i}} \sum_{t=1}^{T} \gamma_t(i) \left[\frac{(x_t - \mu_i)^2}{\sigma_i^2} - 1 \right], \end{aligned}}$$

Fisher Vector (3)

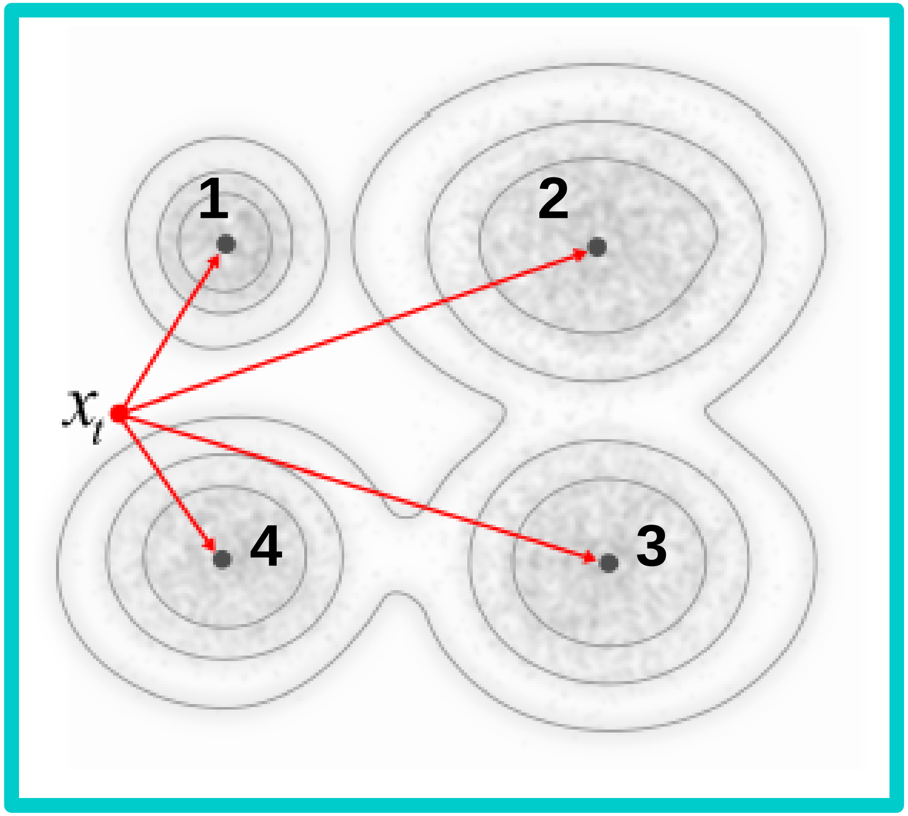
BOV

Hard Assignment

[0 0 0 1]

Soft Assignment

[.3 .1 .1 .5]



Fisher Vector

Gradient wrt w

[.15 -.2 -.35 .2]

Gradient wrt mean

[.8 -1.5 -3.7 -1.3 -3.8 1.2 -.9 1.4]

Gradient wrt var

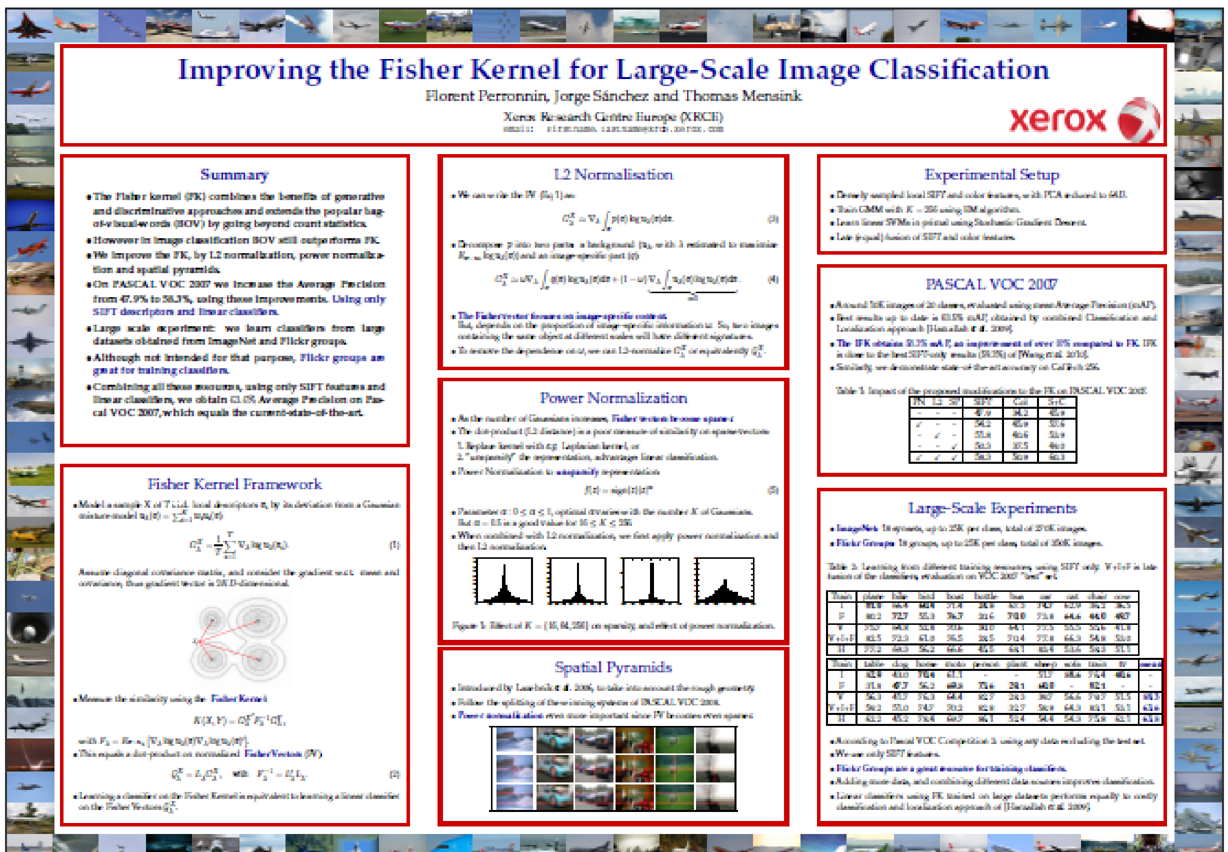
[-1.2 -.9 1.4 -.8 1.5 -3.7 1.3 -3.8]

BOV Histogram has size: K
Fisher Vector (wrt to mean and var): 2 * D * K

* the numbers are only for illustrative purpose

Improving the Fisher Vector

- L2 Normalization
- Power Normalization
- Spatial Pyramid Matching

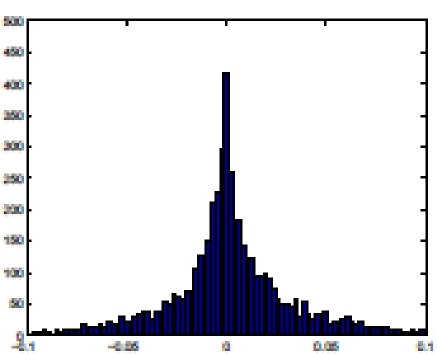


L2 Normalization

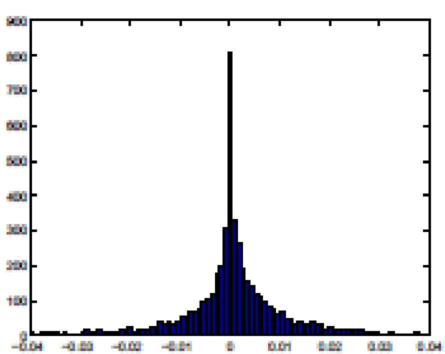
- By construction the Fisher Vector discards descriptors which are likely to occur in any image
- The FV **focus on image specific features**
- However, the FV depends on the amount of image specific information / background information
  - 2 images with same object on a different scale will have a different feature vector
- L2 Normalization to remove this dependence

Power Normalization

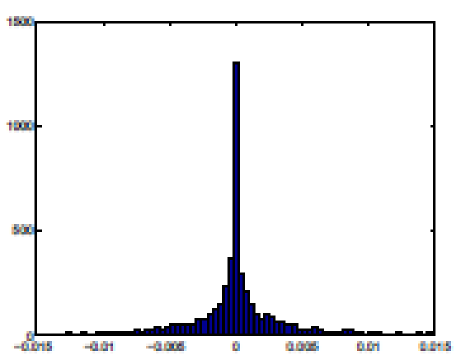
- As the number of Gaussians increase, the FV becomes sparser
  - Replace dot-product with other kernel
  - Unsparsify the representation
- Power normalization to unsparsify: $\boxed{f(z) = \text{sign}(z)|z|^{\alpha}}$



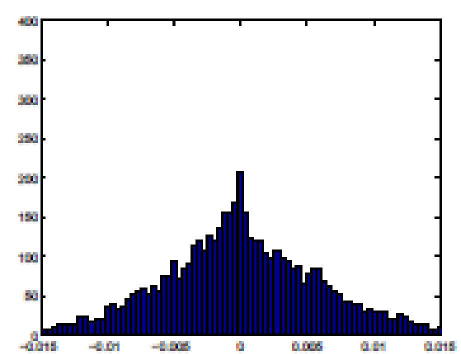
K = 16



K = 64



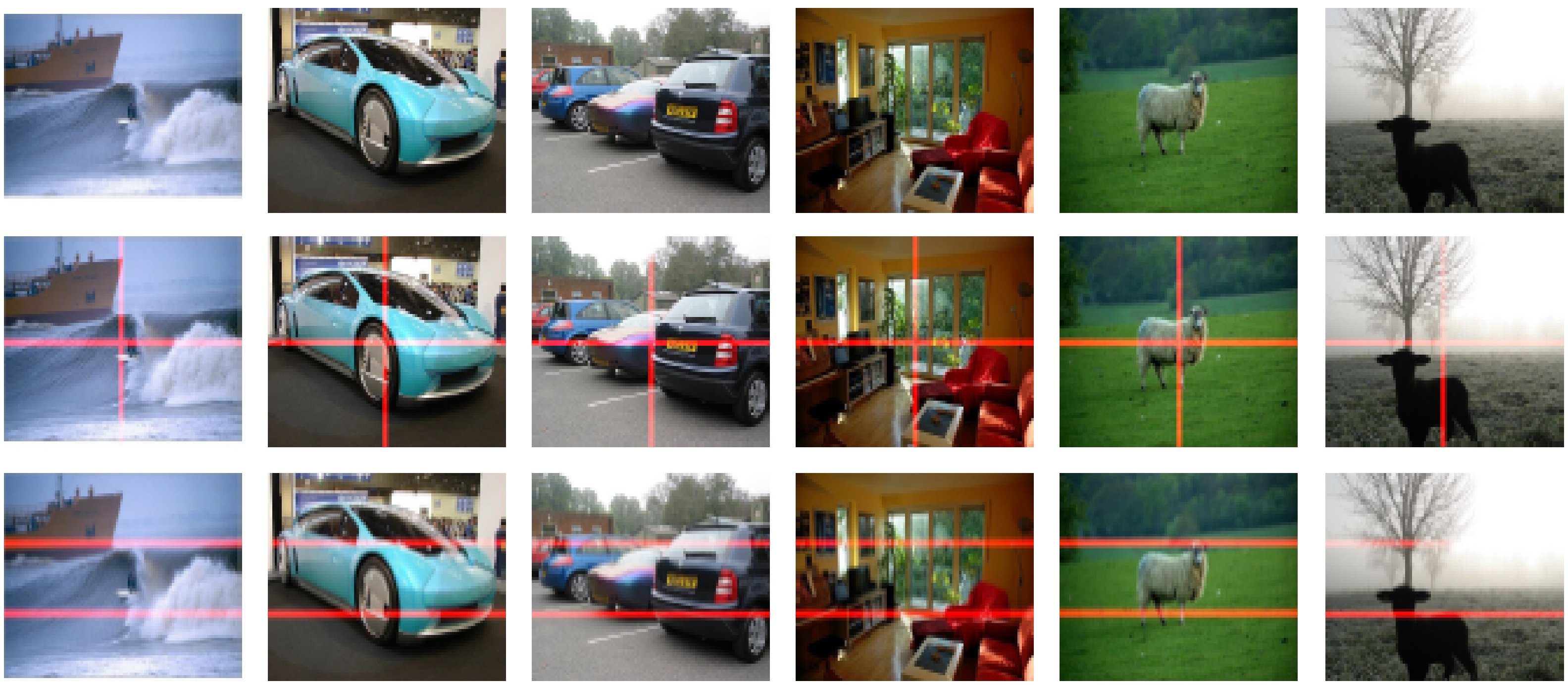
K = 256



K = 256
Power Normalized

Spatial Pyramids

- Take rough geometry into account [Lazebnik 2006]



- Power normalization becomes even more important (FV is sparser)

Experiments VOC 2007

- Improved Fisher Vector [ECCV 2010]
- Dense multiscale sampling, PCA, K=256
- Linear SVM

PN	L2	SP	SIFT	Col	S+C
-	-	-	47.9	34.2	45.9
✓	-	-	54.2	45.9	57.6
-	✓	-	51.8	40.6	53.9
-	-	✓	50.3	37.5	49.0
✓	✓	✓	58.3	50.9	60.3

Experiments VOC 2007 (2)

- Improved Fisher Vector [ECCV 2010]
- Larger Scale
  - Flickr Group Images
  - Up to 25k per class / 350k in total
  - Late fusion with VOC07 trainset
  - 63.3% (SIFT only)
- Best results 63.5% Localization and Classification [Harzallah et al. 2009]
- Flickr Groups are a great resource for labelled images
  - No additional labelling used!
- More training data improves performance

So far...

- FV is a rich representation, extends BOV.
- High dimensional (2 D K S) but allows for linear SVM
- Performance is compatible to state of the art

However...

- 2 * 64 * 256 * 8 = 262,144 dimensions
- Almost dense feature
- ~ 1MB per image / per modality
- ImageNet Train/Test/Val → 1.4 TB (per modality)

Compression [unpublished]

Two options

A) Dimension Reduction

- PCA / Dense Random projections
  - is costly in high dimensional dense space
- Hash Kernels
  - Observation: performance decreases rapidly (already by factor 4)
- Can improve learning speed (not necessary)

B) Data Compression

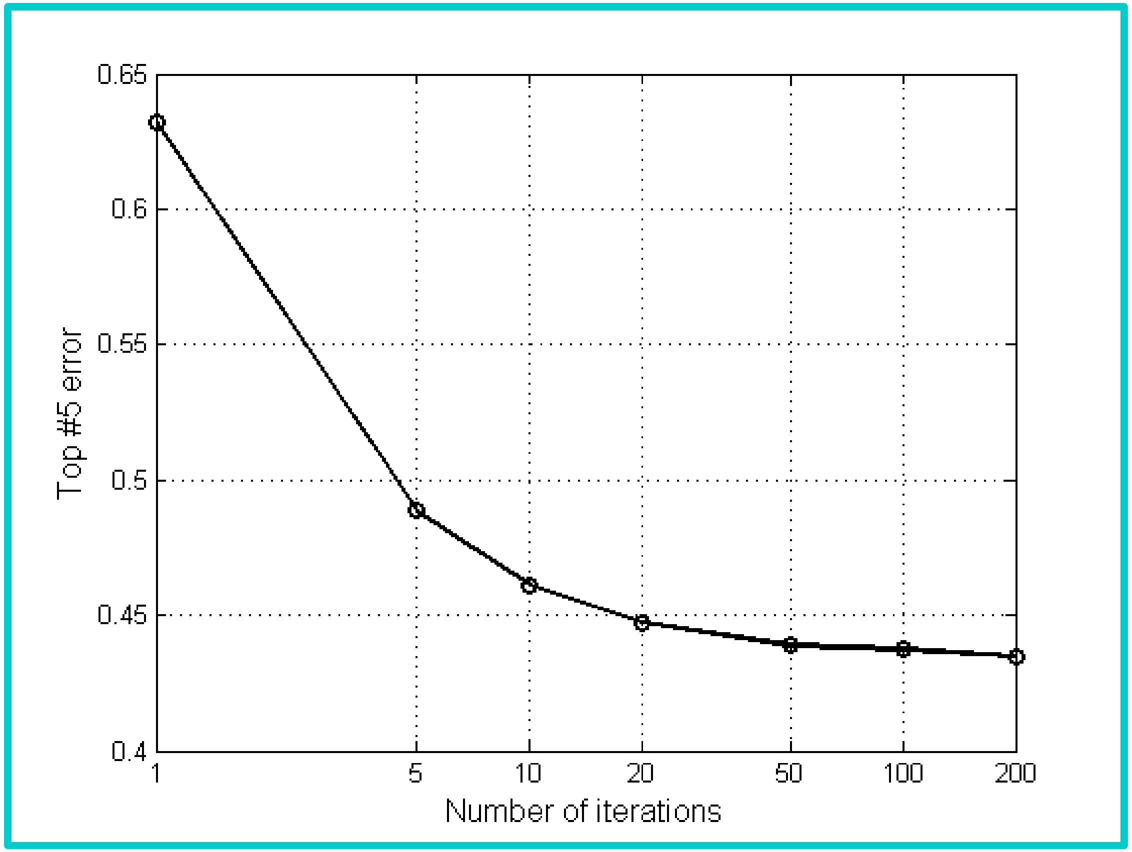
- Use same dimensionality
- But lossy compression up to factor 64 possible.
- 1.4TB → 20GB (per modality)
- Not able to learn in compressed space

Stochastic Gradient Descent (SGD)

- Learn Linear SVM in the primal, PEGASOS
  [Shalev-Shwartz et al. 2007]
- SGD inspired on Pegasos by L. Bottou
  [http://leon.bottou.org]
- Online algorithm, using one sample at the time
- Our approach is:
  1. Load compressed vector
  2. Decompress vector
  3. SGD Update

Stochastic Gradient Descent (2)

- Performance vs number of passes through data



Categorization Pipeline

- Extract dense sampled features (SIFT, Colour)
- Project (with PCA) to 64D
- Learn codebook with K (256) Gaussians on 1M features
- Learn Compressor (on small set of FV)
- Compute and compress FV
- Learn Linear Classifiers using SGD
- Classify test images

Categorization Pipeline (2)

- Computation time for Learning ImageNET

Intel Xeon double quadcore (16 proc) @ 2.53GHz, 32GB RAM

	CPU	Wall-Clock
Extract SIFT+Projection	36h	18h
GMM	minutes	
Learn Compressor	48h	3h
Extract FV + Compression	96h	6h
500 SGD Iterations*	960h	66h
Total (SIFT)	1140h	93h

1.2M train images, training ~ 4 CPU sec per image / modality

* Without significant loss of performance 500 → 50 iterations

Categorization Pipeline (3)

- Computation time for Testing on ImageNET

	CPU
Feature Extraction + Projection	2.5h
FV Extraction	30m
Classifiers	12h

- Total SIFT + Col = 30h
- 150K images / 1000 classes

Classification << 1ms per image/class/modality

Results

- Pascal VOC 2010 and ImageNet Challenge
- Same approach and settings
- K = 256
- FV + L2 & Power Norm + pyramids
- Compression
  - Except for VOC train/val set
- Linear SVM in primal
  - Number of SGD iterations are different

Pascal VOC 2010

- 10K test images / 20 classes / multi-label
- Challenge 1: Only provided train/val data

Rank		MAP
1	NUSPSL	73.8
2	NLPR	71.2
3	NEC	70.9
9	**XRCE Improved FV**	**61.2**

Pascal VOC 2010 (2)

- 10K test images / 20 classes / multi-label
- Challenge 2: Any data except test data
- 1M Flickr Group images of 18 classes
  - Tv/monitor and sofa are missing
  - No additional labelling, just the group labels

Rank		MAP	# Classes
4	BIT	26.9	20
3	UCI	51.7	9
2	**XRCE Flickr 1M**	**65.5**	**18**
1	**XRCE Optimal Fuse F & V**	**68.3**	**20**

More data helps: 61.2 → 68.3 MAP

ImageNet Challenge

- 150k test images / 1000 classes / single labelled.
  - Flat cost: is the correct label in the top 5?
  - Hierarchical cost: distance to lowest common ancestor

	flat cost	hie cost
NEC-UIUC	0.28191	2.1144
XRCE	**0.33649**	**2.5553**
ISIL	0.44558	3.6536
UCI	0.46624	3.6288

Conclusions

- Improved Fisher Vector for Image Classification
- Linear Classification → scales to larger scale
- More data (Flickr Groups) helps
  - Pascal VOC 2010: 61.2 → 68.3 MAP

- Using compression → scales to LARGE scale

- Very very fast: training (8s/i) & classifying (2s/i)

Questions?

L2 Normalization (2)

- Fisher Vector

$$\boxed{\begin{aligned} G_\lambda^X &= \frac{1}{T}\sum_{t=1}^{T} \nabla_\lambda \log u_\lambda(x_t) && 1 \\ G_\lambda^X &\approx \nabla_\lambda E_{x\sim p} \log u_\lambda(x) = \nabla_\lambda \int_x p(x) \log u_\lambda(x) dx. && 2 \\ G_\lambda^X &\approx \omega \nabla_\lambda \int_x q(x) \log u_\lambda(x) dx + (1-\omega) \nabla_\lambda \int_x u_\lambda(x) \log u_\lambda(x) dx && 3 \end{aligned}}$$

- GMM Trained with Maximum Likelihood, ie maximize $\boxed{E_{x\sim u_\lambda} \log u_\lambda(x)}$

$$\boxed{\nabla_\lambda \int_x u_\lambda(x) \log u_\lambda(x) dx = \nabla_\lambda E_{x\sim u_\lambda} \log u_\lambda(x) \approx 0}$$

- Fisher Vectors automatically **focus on image specific features** and discard image independent/background features

- L2 Normalization to remove dependence on ω