Fisher Vectors for Fine-Grained Visual Categorization

Jorge Sánchez, Florent Perronnin and Zeynep Akata

Xerox Research Centre Europe (XRCE)

email: firstname.lastname@xrce.xerox.com

Abstract

- **Fine-Grained Visual Categorization (FGVC): fine distinction of closely related image categories (*e.g.* vehicles or plants).**
- **Bag-of-visual-words (BOV): most popular image representation for categorization → applied to FGVC in [Branson *et al.*, ECCV'10], [Deng *et al.*, ECCV'10].**
- **Our contribution: show that the Fisher Vector (FV) is an excellent alternative to the BOV for FGVC**
  - **present theoretical and practical motivations.**
  - **provide empirical evidence: comparison with the BOV.**

⇒ State-of-the-art on 4 fine-grained subsets of ImageNet

Fungus, Ungulates, Vehicles and ImageNet10K

The Fisher Vector (FV)

[Perronnin and Dance, CVPR'07] based on [Jaakkola and Haussler, NIPS'98]

- $X = \{x_t, t = 1 \dots T\}$ is a set of T i.i.d D-dim local descriptors (*e.g.* SIFT).
- $u_\lambda(x) = \sum_{i=1}^{N} w_i u_i(x)$ is a GMM with N Gaussians and parameters λ which models the distribution of descriptors in any image (visual vocabulary).
- The FV is the following **gradient vector**:

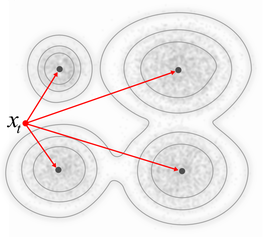
$$G_\lambda^X = \frac{1}{T} \sum_{t=1}^{T} \nabla_\lambda \log u_\lambda(x_t).$$

normalized by the Fisher Information Matrix (whitening).
→ Describe a sample X by its **deviation** from u_λ.

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

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

To be compared with the BOV: $\frac{1}{T} \sum_{t=1}^{T} \gamma_t(i)$.



⇒ Beyond counting: 1st and 2nd order statistics.

- Combine with **Spatial Pyramid (SP)**: compute one FV per image region and concatenate (*e.g.* $R = 1 + 3 = 4$).

- The FV is $E = 2DNR$-dim (NR-dim for BOV)
  e.g. $D = 64$, $N = 256$, and $R = 4$ → E=131,072-dim

⇒ Dense HD features at low computational cost.

FV Normalisation

[Perronnin, Sánchez and Mensink, ECCV'10]

- **L2 normalization:** Assume the x_t's are drawn from $p = \omega q + (1 - \omega) u_\lambda$

$$G_\lambda^X \approx \nabla_\lambda \int_x p(x) \log u_\lambda(x) dx \quad \text{(law of large numbers)}$$

$$\approx \omega \nabla_\lambda \int_x q(x) \log u_\lambda(x) dx + (1 - \omega) \underbrace{\nabla_\lambda \int_x u_\lambda(x) \log u_\lambda(x) dx}_{\approx 0 \text{ (MLE)}}.$$

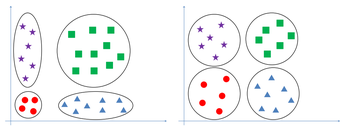
→ L2-normalize to remove dependence on ω

⇒ FV discards background information (TF-IDF).

- **Square-rooting**:
  Apply following transform:

$$f(z) = \text{sign}(z)|z|^{1/2}$$

If we accept the compound Poisson as a generative model of FVs, then f can be interpreted as a **variance stabilizing tranform**.



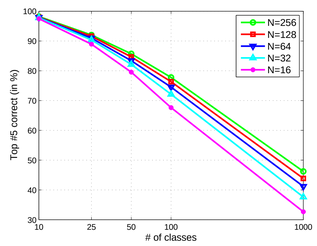
Before/after variance stabilization.

⇒ Makes FVs more linearly separable.

FV Compression

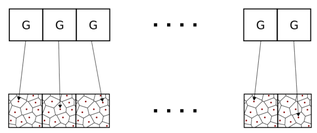
[Sánchez and Perronnin, CVPR'11]

- More (and denser) classes require larger features for linear separability:
  - ILSVRC2010: 1K classes (leaf nodes).
  - Train/validation/test split: 1.26M/50K/250K
  - Measure top-5 correct (in %).
  - Vary number of Gaussians N.
  - **Subsample number of classes** (repeat experiments ×5).



⇒ HD features are necessary on large datasets.

- Use **Product Quantization (PQ)** [Jégou *et al.*, TPAMI'11] to compress HD FV:
  - split feature into small sub-vectors.
  - **Training** step: k-means clustering for each sub-vector.
  - **Compression** step: encode each sub-vector by its closest codebook index.
    → vector of codebook indices.
  - **SGD learning**: on-the-fly decompression of features (look-up tables).



⇒ From 4.3TBs to 80GBs on ImageNet10K

Summary: Why the FV for FGVC?

- **The FV is informative:** FGVC requires subtle cues
  → keep as much of the raw patch information as possible.
  - The quantization process in the BOV is lossy.
  - The FV includes **higher-order statistics**: 1st and 2nd order
    → more raw patch information.
- **The FV is discriminative:**
  - The FV describes an image by what makes it different from others on average.
    → discards automatically "background" information.
  - The background can be adapted to each fine-grained problem by training the GMM u_λ on the relevant data: vehicle background, plant background, etc.
    → much more still needs to be done on this aspect...
- **The FV is scalable:** FGVC (typically) implies a large number of classes.
  → large-scale problem.
  - The FV is **efficient to compute**: small visual vocabularies ($\leq$ 1K Gaussians) are sufficient to generate HD signatures.
  - The FV **works well with costless linear classifiers**:
    → efficient SVM learning, *e.g.* with Stochastic Gradient Descent (SGD)
    → efficient runtime: cost independent of # of support vectors
  - The FV **can be easily and efficiently compressed**.
    → reduces storage/memory/IO issues.
    → enables working with HD signatures.

Large-Scale FGVC Experiments

- Datasets: Fungus, Ungulates, Vehicles and ImageNet10K.
  - Protocol: 1/2 of the data for training, 1/2 for testing
  - Accuracy measured as top-1 correct (in %).
- Comparison with [Deng *et al.*, ECCV'10]:
  - Feature: SIFT + BOV (1K codewords) + SP (21 regions) → 21K-dim
  - Learning: fast HIK SVM (explicit embedding [Maji *et al.*, ICCV'09] + LIBLINEAR [Fan *et al.*, JMLR'08]) → **6 CPU years.**
- Our system:
  - Feature: SIFT + FV (256 Gaussians) + SP (4 regions) → 131K-dim
  - Learning: linear SVM (SGD [Bottou]) → **70 CPU days.**

Dataset	Fungus	Ungulate	Vehicle	INet10K
#Classes	134	183	262	10K
#Images	88K	173K	226K	9M
BOV	11.6%	14.5%	24.1%	6.4%
FV	**19.4%**	**29.5%**	**42.3%**	**16.7%**

⇒ State-of-the-art for FGVC (and ILSVRC 2010)