Fisher Vector Encoding

Gonzalo Vaca-Castano

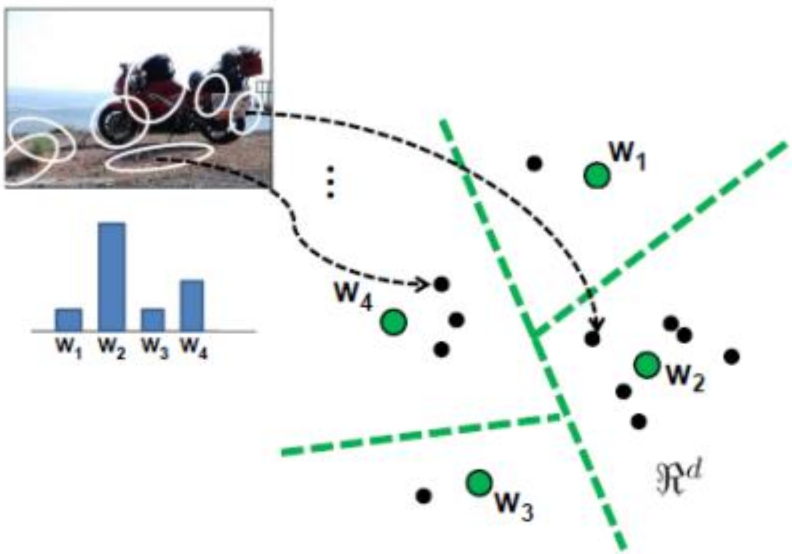
University of Central Florida

Papers used for this presentation

- **Fisher Kernels on Visual Vocabularies for Image Categorization** Florent Perronnin and Christopher Dance. CVPR 2007
- **Improving the Fisher Kernel for Large-Scale Image Classification.** Florent Perronnin, Jorge Sanchez, and Thomas Mensink. ECCV 2010
- **Image Classification with the Fisher Vector: Theory and Practice.** Jorge Sánchez , Florent Perronnin , Thomas Mensink , Jakob Verbeek.

Motivation

- BoW is the most typical representation method



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

- Define number of Gaussians
- MLE to estimate GMM
- Encode using fisher
- SVM

Algorithm 1 Compute Fisher vector from local descriptors

Input:

- Local image descriptors $X = \{x_t \in \mathbb{R}^D, t = 1, \ldots, T\}$,
- Gaussian mixture model parameters $\lambda = \{w_k, \mu_k, \sigma_k, k = 1, \ldots, K\}$

Output:

- normalized Fisher Vector representation $\mathscr{G}_\lambda^X \in \mathbb{R}^{K(2D+1)}$

1. **Compute statistics**
   - For $k = 1, \ldots, K$ initialize accumulators
     - $S_k^0 \leftarrow 0, \quad S_k^1 \leftarrow 0, \quad S_k^2 \leftarrow 0$
   - For $t = 1, \ldots T$
     - Compute $\gamma_t(k)$ using equation (15)
     - For $k = 1, \ldots, K$:
       * $S_k^0 \leftarrow S_k^0 + \gamma_t(k)$,
       * $S_k^1 \leftarrow S_k^1 + \gamma_t(k) x_t$,
       * $S_k^2 \leftarrow S_k^2 + \gamma_t(k) x_t^2$

$$\gamma_t(k) = \frac{w_k u_k(x_t)}{\sum_{j=1}^K w_j u_j(x_t)}$$

2. **Compute the Fisher vector signature**
   - For $k = 1, \ldots, K$:

$$\begin{aligned} \mathscr{G}_{\alpha_k}^X &= (S_k^0 - T w_k) / \sqrt{w_k} \\ \mathscr{G}_{\mu_k}^X &= (S_k^1 - \mu_k S_k^0) / (\sqrt{w_k} \sigma_k) \\ \mathscr{G}_{\sigma_k}^X &= (S_k^2 - 2\mu_k S_k^1 + (\mu_k^2 - \sigma_k^2) S_k^0) / \left(\sqrt{2 w_k} \sigma_k^2\right) \end{aligned}$$

   - Concatenate all Fisher vector components into one vector
     $\mathscr{G}_\lambda^X = \left(\mathscr{G}_{\alpha_1}^X, \ldots, \mathscr{G}_{\alpha_K}^X, \mathscr{G}_{\mu_1}^X{}', \ldots, \mathscr{G}_{\mu_K}^X{}', \mathscr{G}_{\sigma_1}^X{}', \ldots, \mathscr{G}_{\sigma_K}^X{}'\right)'$

3. **Apply normalizations**
   - For $i = 1, \ldots, K(2D+1)$ apply power normalization
     - $\left[\mathscr{G}_\lambda^X\right]_i \leftarrow \text{sign}\left(\left[\mathscr{G}_\lambda^X\right]_i\right) \sqrt{\left|\left[\mathscr{G}_\lambda^X\right]_i\right|}$
   - Apply ℓ_2-normalization:
     $\mathscr{G}_\lambda^X = \mathscr{G}_\lambda^X / \sqrt{\mathscr{G}_\lambda^X{}' \mathscr{G}_\lambda^X}$

$$\begin{aligned} S_k^0 &= \sum_{t=1}^T \gamma_t(k) \\ S_k^1 &= \sum_{t=1}^T \gamma_t(k) x_t \\ S_k^2 &= \sum_{t=1}^T \gamma_t(k) x_t^2 \end{aligned}$$

Problem

- 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

Motivation

- 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

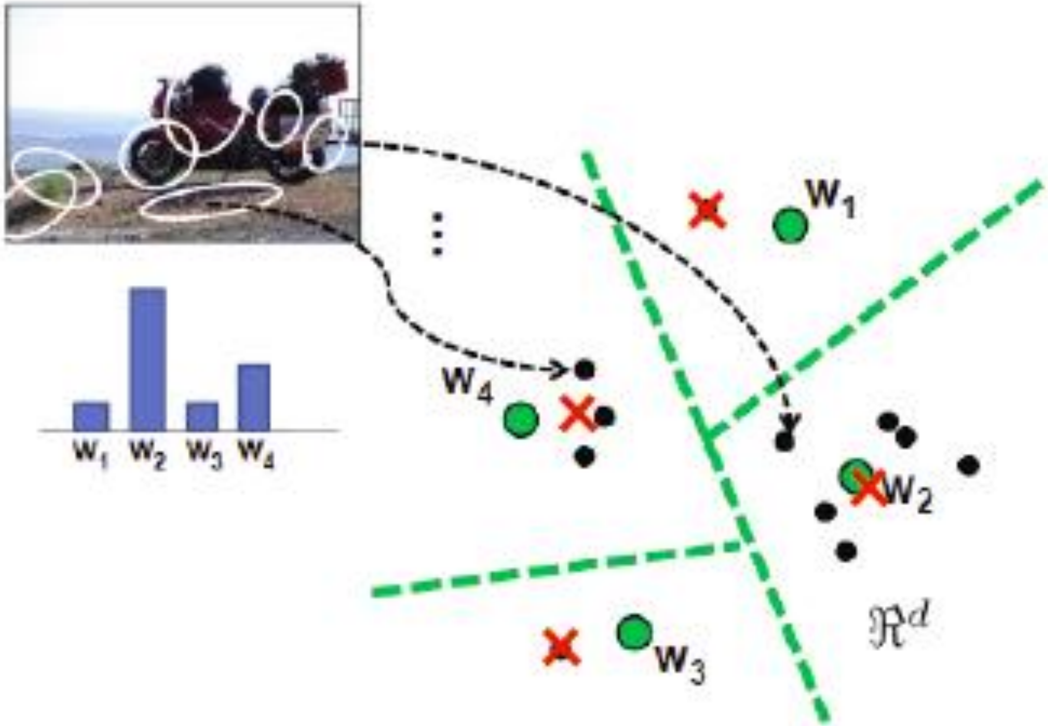
We need image representation which contain fine-grained information !

Motivation

- BoW answer:
  - increase visual vocabulary size
- How to increase amount of information without increasing the visual vocabulary size?
  - BOV is only about counting
  - Include higher order statistics (mean, covariance) in representation

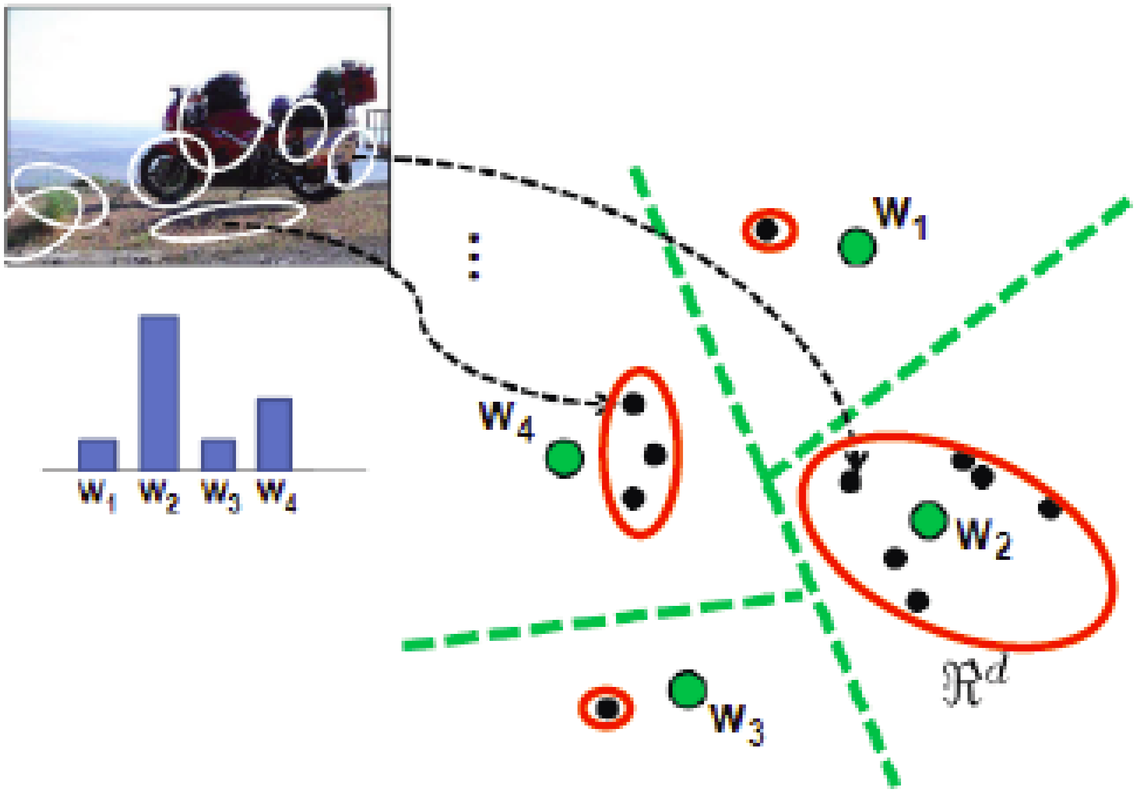
Motivation

- Mean



Motivation

- Variance



Fisher Vector Idea

- Characterizing a sample by its deviation from the generative model (GMM).
- Deviation is measured by computing the gradient of the sample log-likelihood with respect to the model parameters (w, μ, σ)

Fisher Vector

- T samples $X = \{x_t, t = 1, \ldots, T\}$
- Vector of M parameters $\lambda = [\lambda_1, \ldots, \lambda_M]' \in \mathbb{R}^M$
- Likelihood is: $u_\lambda(X) = p(X|\lambda)$
- In statistics, *score function* (informant) is given by $G_\lambda^X = \nabla_\lambda \log u_\lambda(X)$
- Intuition: direction in which the parameters λ of the model should we modified to better fit the data.

Fisher Vector

- The score function is a representation of the data using higher order statistics.
- gradient of the log-likelihood describes the direction in which parameters should be modified to best fit the data
- Dimensions depends in number of parameters M, not in number of samples
- It is important to **normalize** the input vectors since most discriminative classifiers use an inner product term.

How to Normalize ?

- Fisher information matrix (FIM)

$$F_\lambda = E_{x \sim u_\lambda} \left[G_\lambda^X G_\lambda^X{}' \right]$$

- FIM is the variance of the score G.
- $Var(G) = E[G^2] - (E[G])$^2.
- But $E[G] = 0$ (see next slide)
- $Var(G) = E[G^2]$ → $F_\lambda = E_{x \sim u_\lambda} \left[G_\lambda^X G_\lambda^X{}' \right]$

$$F_\lambda = E_X \left[\nabla_\lambda \log p(X|\lambda) \nabla_\lambda \log p(X|\lambda)' \right]$$

Score mean is zero

- $E_x \left[\frac{\partial}{\partial\lambda}\log p(x|\lambda)\right] =$
  $E_x \left[(\frac{\partial}{\partial\lambda}p(x|\lambda))/p(x|\lambda)\right] =$
  $\int \left[\frac{\frac{\partial}{\partial\lambda}p(x|\lambda)}{p(x|\lambda)} * p(x|\lambda)\right]dx = \frac{\partial}{\partial\lambda}\int p(x|\lambda)dx =$
  $\frac{\partial}{\partial\lambda}1 = 0$

How to measure distances ?

- Use FIM ($F_\lambda = E_X\left[\nabla_\lambda \log p(X|\lambda) \nabla_\lambda \log p(X|\lambda)'\right]$) to normalize distances
- **Fisher Kernel:** $K(X,Y) = G_\lambda^{X\,\prime} F_\lambda^{-1} G_\lambda^Y$
- F_λ is symmetric --> positive semi-definite
- Has a Cholesky decomposition $F_\lambda = L'_\lambda L_\lambda$
- Fisher Kernel becomes

$$K_{FK}(X,Y) = \mathscr{G}_\lambda^{X\,\prime} \mathscr{G}_\lambda^Y$$

- Where $\mathscr{G}_\lambda^X = L_\lambda G_\lambda^X = L_\lambda \nabla_\lambda \log u_\lambda(X)$ is the Fisher Vector

Important Observation

- Fisher Kernel is non-linear, $K(X,Y) = G_\lambda^{X\prime} F_\lambda^{-1} G_\lambda^Y$
- But is a linear kernel when you use the Fisher vector as feature vector

$$K_{FK}(X,Y) = \mathscr{G}_\lambda^{X\prime} \mathscr{G}_\lambda^Y$$

- Consequence: linear classifiers can be learned very efficiently.

Fisher Vector on Images

- Fisher vector is given by:

$$\mathcal{G}_\lambda^X = L_\lambda G_\lambda^X = L_\lambda \nabla_\lambda \log u_\lambda(X)$$

- Assuming that the samples (SIFT descriptors) are independent

  p(x1,x2,… xt)=p(x1)p(x2) … p(xt)

$$\mathcal{G}_\lambda^X = \sum_{t=1}^{T} L_\lambda \nabla_\lambda \log u_\lambda(x_t)$$

- FV is a sum of normalized gradient statistics $L_\lambda \nabla_\lambda \log u_\lambda(x_t)$ computed for each descriptor !!!

GMM case

- Model is GMM $\lambda = \{w_k, \mu_k, \Sigma_k, k = 1, \ldots, K\}$
- $$u_\lambda(x) = \sum_{k=1}^{K} w_k u_k(x),$$ $$u_k(x) = \frac{1}{(2\pi)^{D/2}|\Sigma_k|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_k)'\Sigma_k^{-1}(x-\mu_k)\right\},$$ $$\sum_{k=1}^{K} w_k = 1,$$
- Assuming that covariance matrices are diagonal (Uncorrelated data)

Score Function for GMM

$$\frac{\partial \mathcal{L}(X|\lambda)}{\partial w_i} = \sum_{t=1}^{T} \left[\frac{\gamma_t(i)}{w_i} - \frac{\gamma_t(1)}{w_1} \right] \text{ for } i \geq 2 ,$$

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

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

- Soft Assignment:

$$\gamma_t(i) = \frac{w_i u_i(x_t)}{\sum_{j=1}^{K} w_j u_j(x_t)} .$$

Fisher Normalization

- See appendix A

$$f_{w_i} = T\left(\frac{1}{w_i} + \frac{1}{w_1}\right),$$

$$f_{\mu_i^d} = \frac{T w_i}{\left(\sigma_i^d\right)^2},$$

$$f_{\sigma_i^d} = \frac{2T w_i}{\left(\sigma_i^d\right)^2}.$$

- Fisher vector

$$f_{w_i}^{-1/2} \partial \mathcal{L}(X|\lambda)/\partial w_i$$

$$f_{\mu_i^d}^{-1/2} \partial \mathcal{L}(X|\lambda)/\partial \mu_i^d$$

$$f_{\sigma_i^d}^{-1/2} \partial \mathcal{L}(X|\lambda)/\partial \sigma_i^d.$$

Fisher Vector

$$\gamma_t(i) = \frac{w_i u_i(x_t)}{\sum_{j=1}^{K} w_j u_j(x_t)}.$$

$$\mathcal{G}_{\alpha_k}^X = \frac{1}{\sqrt{w_k}} \sum_{t=1}^{T} \left(\gamma_t(k) - w_k\right),$$

$$\mathcal{G}_{\mu_k}^X = \frac{1}{\sqrt{w_k}} \sum_{t=1}^{T} \gamma_t(k) \left(\frac{x_t - \mu_k}{\sigma_k}\right),$$

$$\mathcal{G}_{\sigma_k}^X = \frac{1}{\sqrt{w_k}} \sum_{t=1}^{T} \gamma_t(k) \frac{1}{\sqrt{2}} \left[\frac{(x_t - \mu_k)^2}{\sigma_k^2} - 1\right].$$

Fisher Vector

$$\gamma_t(i) = \frac{w_i u_i(x_t)}{\sum_{j=1}^{K} w_j u_j(x_t)}.$$

Closely related to BoW (Soft assignment) →

$$\mathscr{G}_{\alpha_k}^X = \frac{1}{\sqrt{w_k}} \sum_{t=1}^{T} (\gamma_t(k) - w_k),$$

Closely related to Vlad →

$$\mathscr{G}_{\mu_k}^X = \frac{1}{\sqrt{w_k}} \sum_{t=1}^{T} \gamma_t(k) \left(\frac{x_t - \mu_k}{\sigma_k} \right),$$

$$\mathscr{G}_{\sigma_k}^X = \frac{1}{\sqrt{w_k}} \sum_{t=1}^{T} \gamma_t(k) \frac{1}{\sqrt{2}} \left[\frac{(x_t - \mu_k)^2}{\sigma_k^2} - 1 \right].$$

Fisher Vector: Comparison with BOW

Advantages

- BoV is a particular case of the FV where the gradient computation is restricted to the mixture weight parameters of the GMM.
- FV is that it can be computed from much smaller vocabularies and therefore at a lower computational cost.
- it performs well even with simple linear classifiers

Disadvantages

- Requires more storage

  (2*D+1)*N – 1

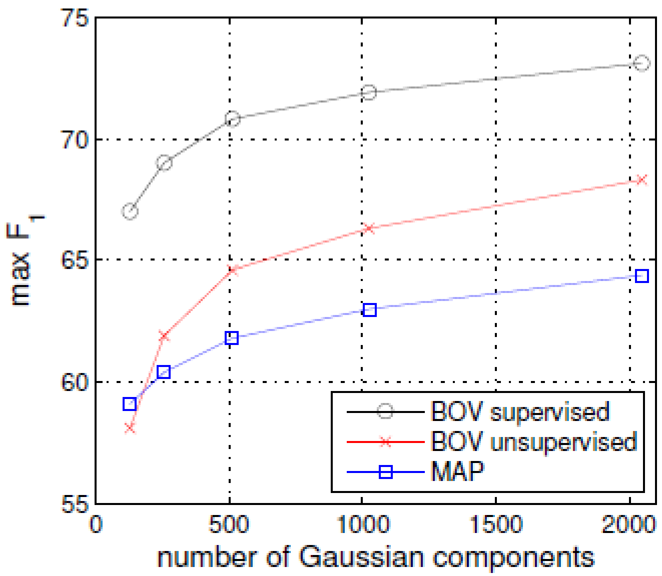
D= feature Dimension

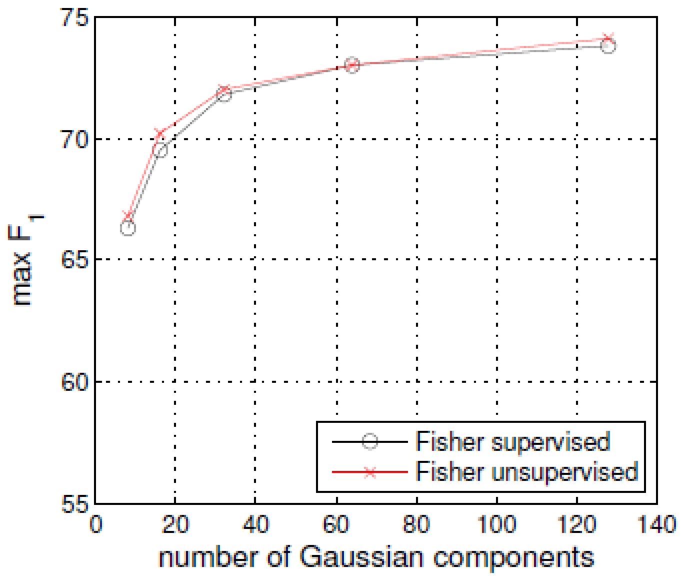
N = Num codewords

Specific Dictionary or Global Dictionary

- Case 1:
  - Train the GMM in an unsupervised manner with the low-level feature vectors from all categories or even on a separate dataset
- Case 2:
  - Train a vocabulary for each class
  - For one image, a representation is generated for each class

Experiments





Algorithm 1 Compute Fisher vector from local descriptors

Input:

- Local image descriptors $X = \{x_t \in \mathbb{R}^D, t = 1, \ldots, T\}$,
- Gaussian mixture model parameters $\lambda = \{w_k, \mu_k, \sigma_k, k = 1, \ldots, K\}$

Output:

- normalized Fisher Vector representation $\mathscr{G}_\lambda^X \in \mathbb{R}^{K(2D+1)}$

1. **Compute statistics**
   - For $k = 1, \ldots, K$ initialize accumulators
     - $S_k^0 \leftarrow 0, \quad S_k^1 \leftarrow 0, \quad S_k^2 \leftarrow 0$
   - For $t = 1, \ldots T$
     - Compute $\gamma_t(k)$ using equation (15)
     - For $k = 1, \ldots, K$:
       * $S_k^0 \leftarrow S_k^0 + \gamma_t(k)$,
       * $S_k^1 \leftarrow S_k^1 + \gamma_t(k) x_t$,
       * $S_k^2 \leftarrow S_k^2 + \gamma_t(k) x_t^2$
2. **Compute the Fisher vector signature**
   - For $k = 1, \ldots, K$:

$$\begin{aligned} \mathscr{G}_{\alpha_k}^X &= (S_k^0 - T w_k) / \sqrt{w_k} \\ \mathscr{G}_{\mu_k}^X &= (S_k^1 - \mu_k S_k^0) / (\sqrt{w_k} \sigma_k) \\ \mathscr{G}_{\sigma_k}^X &= (S_k^2 - 2\mu_k S_k^1 + (\mu_k^2 - \sigma_k^2) S_k^0) / \left(\sqrt{2 w_k} \sigma_k^2\right) \end{aligned}$$

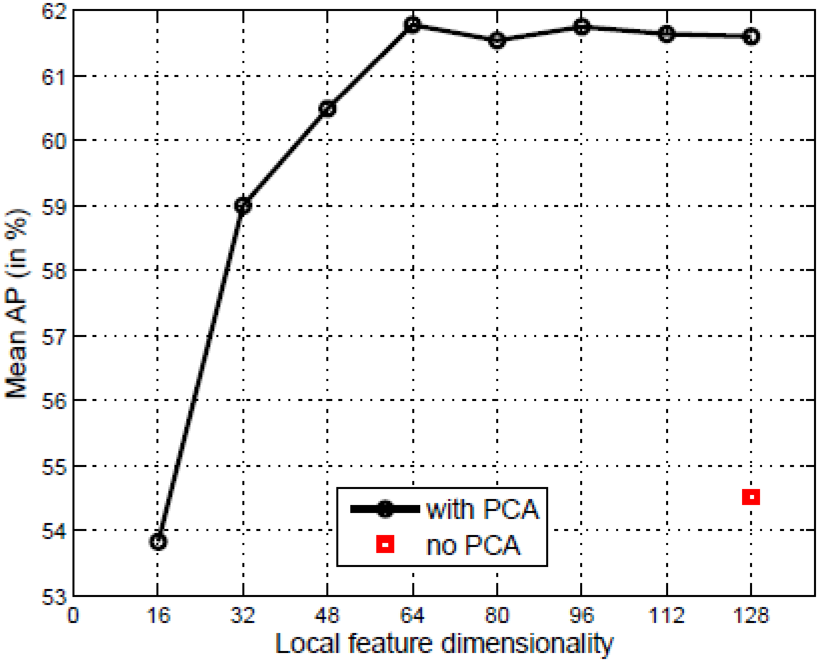
   - Concatenate all Fisher vector components into one vector $\mathscr{G}_\lambda^X = \left(\mathscr{G}_{\alpha_1}^X, \ldots, \mathscr{G}_{\alpha_K}^X, \mathscr{G}_{\mu_1}^{X\prime}, \ldots, \mathscr{G}_{\mu_K}^{X\prime}, \mathscr{G}_{\sigma_1}^{X\prime}, \ldots, \mathscr{G}_{\sigma_K}^{X\prime}\right)'$
3. **Apply normalizations**
   - For $i = 1, \ldots, K(2D+1)$ apply power normalization
     - $\left[\mathscr{G}_\lambda^X\right]_i \leftarrow \text{sign}\left(\left[\mathscr{G}_\lambda^X\right]_i\right) \sqrt{\left|\left[\mathscr{G}_\lambda^X\right]_i\right|}$
   - Apply ℓ_2-normalization: $\mathscr{G}_\lambda^X = \mathscr{G}_\lambda^X / \sqrt{\mathscr{G}_\lambda^{X\prime} \mathscr{G}_\lambda^X}$

Experimental setup

- In House dataset, Pascal 2006
- Best results:
  - Num Gaussians= 128
  - Gradient respect to mean and variance concatenated.
  - Dimension Reduction using PCA
  - L2 Normalization
  - Power normalization $f(z) = \text{sign}(z)|z|^{\alpha}$

Additional notes (Pascal 2007)

- Effect of PCA



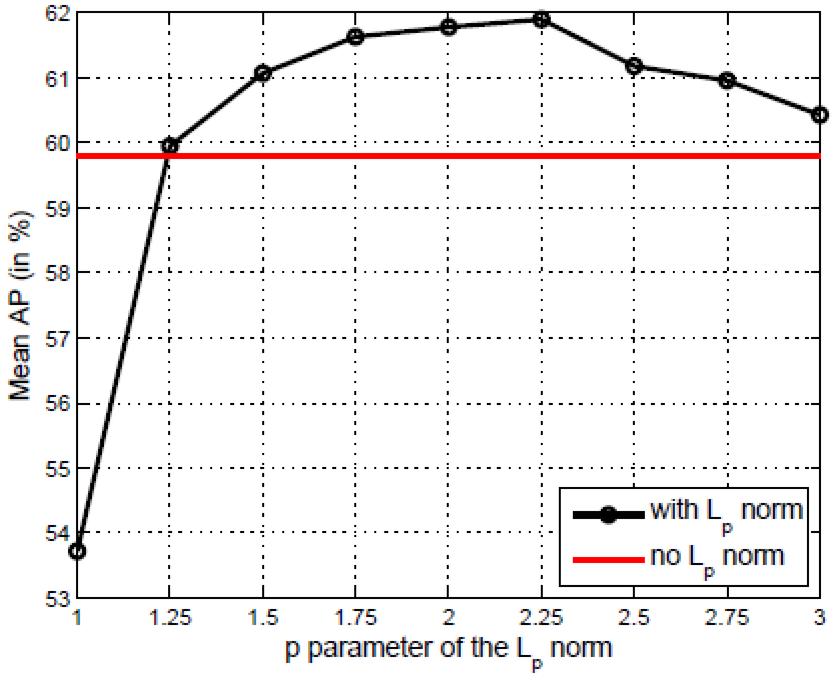
Additional Notes (Pascal 2007)

- Effect of Normalization

PN	ℓ_2	SP	SIFT		LCS	
No	No	No	49.6		35.2	
Yes	No	No	57.9	(+8.3)	47.0	(+11.8)
No	Yes	No	54.2	(+4.6)	40.7	(+5.5)
No	No	Yes	51.5	(+1.9)	35.9	(+0.7)
Yes	Yes	No	59.6	(+10.0)	49.7	(+14.7)
Yes	No	Yes	59.8	(+10.2)	50.4	(+15.2)
No	Yes	Yes	57.3	(+7.7)	46.0	(+10.8)
Yes	Yes	Yes	61.8	(+12.2)	52.6	(+17.4)

Additional Notes (Pascal 2007)

Effect of Lp normalization



Additional Notes

∇	MAP (in %)
w	46.9
μ	57.9
σ	59.6
$\mu\sigma$	61.8
wμ	58.1
wσ	59.6
w$\mu\sigma$	61.8