EFFICIENT IMAGE CATEGORIZATION WITH SPARSE FISHER VECTOR

Xiankai Lu[1], *Zheng Fang*[2], *Tao Xu*[2], *Haiting Zhang*[3], *Hongya Tuo*[2]

[1]Department of Automation, Shanghai Jiao Tong University,Shanghai, China
[2]School of Aeronautics and Astronautics, Shanghai Jiao Tong University, Shanghai, China
[3]School of Control Science and Engineering, Shan Dong University, Jinan, China

ABSTRACT

In object recognition, Fisher vector (FV) representation is one of the state-of-the-art image representations ways at the expense of dense, high dimensional features and increased computation time. A simplification of FV is attractive, so we propose Sparse Fisher vector (S-FV). By incorporating locality strategy, we can accelerate the Fisher coding step in image categorization which is implemented from a collective of local descriptors. Combining with pooling step, we explore the relationship between coding step and pooling step to give a theoretical explanation about SFV. Experiments on benchmark datasets have shown that SFV leads to a speedup of several-fold of magnitude compares with FV, while maintaining the categorization performance. In addition, we demonstrate how SFV preserves the consistence in representation of similar local features.

Index Terms— Sparse Fisher vector, locality strategy, Generalized Max Pooling, image categorization

1. INTRODUCTION

The Fisher vector approach [1–6] is an extension of the popular Bag-Of-Words (BOW) model by encoding for codeword the mean and variance of local descriptors. It consists of two steps (i) encoding step, encoding the descriptors into dense and high-dimensional features codes; and (ii) pooling step, pooling the codes into a vector. With several improvements [2, 3], Fisher vector has been one of the most effective ways for image categorization.

The success of FV representation is ascribed to its high dimensionality, but FV representation also suffers high computation cost when compared to BOW model [3], especially for large-scale image retrieval and object detection. For specific tasks, several simplified [4, 7, 8] and extended [2, 5, 6, 9] versions of Fisher coding have emerged. In [4], Jegou et al. proposed the VLAD representation in which each local descriptor is assigned to the nearest visual word, then the differences between codewords and corresponding descriptor are accumulated. In [5], Florent Perronnin et al. compressed the high-dimensional Fisher vectors through Local Sensitive Hashing. Recently, Dan Oneata et al. [7] presented approximations to normalizations in Fisher vector. In [8], authors realized a fast local area independent representation by representing the picture as sparse integral images. In this paper, we combine the locality strategy into Fisher vector to reduce the time consumption in feature coding step.

Locality strategy has been used in Linear Embedding and spectral clustering i.e. Local Linear Embedding [10] and local spectral clustering [11]. Inspired by this strategy, many localized coding ways or nearest search algorithms have emerged in BOW, for example, Locality-constrained Linear Coding (LLC) [12], Local Soft Coding (LSC) [13], Laplacain Sparse Coding [14], Local Coordinate Coding [15], local sparse coding [16]. Also this locality-preserving method has been used in pooling step [17]. This locality will produce an early cut off effect to remove the unreliable longer distances. Previous work has shown the effectiveness of preserving configuration space locality during coding, so that similar inputs lead to similar codes [14]. Also we can view them as a trick whose computational cost would be prohibitive with standard coding. Because all the coding coefficients can be regarded as the probability density to describe the feature which can be represented by histogram or fisher vector. In this paper, we will introduce the LLC, LSC and SFV from probabilistic perspective and reveal the relationships between LLC, LSC and SFV, and this part will be discussed in section 4.

For the pooling step in image categorization, Naila Murray [18] et al. tried to generalize max pooling (GMP) to Fisher vector by constructing object function with loss term. Based on this structure, we reformulate the sparse Fisher vector which is the origin Fisher vector combined with locality strategy.

A notable previous idea which is similar to our work is proposed in [3] with "posterior thresholding". But [3] only regarded this as an accelerating trick, and failed to provide the detailed theoretical proof and the effectiveness of the proposed method are not explained. Our paper provides the detailed explanation of the scheme and implement a experimental evaluation on image categorization task.

2. GENERALIZED MAX POOLING REVISIT

Fisher vector is essentially the sum pooling of encoded SIFT features. It should be noted that the sum-pooled representation is more influenced by frequent descriptors in one image. While max-pooled representation only considers the greatest response, and therefore immune to this effect, but it does not apply to aggregation-based encoding such as FV representation. To alleviate the problem, [18] proposed the generalized max pooling method that mimics the desirable properties of max pooling. They denote ϕ_n the code vector of each feature, and ϕ^{max} the GMP vector. GMP demands that $\phi_{\mathrm{n}}^{\mathrm{T}}\phi^{\max} = \mathrm{Const}$, which indicates that ϕ^{max} is equally similar to frequent and rare features. In the BOF case, GMP is strictly equivalent to max pooling [18]. GMP can be formalized in two ways. The first is the primal formulation:

$$\phi^{gmp} = \arg\min_{\phi} \left\|\Phi^T\phi - \mathbf{1}_{\mathrm{N}}\right\|^2 \tag{1}$$

which directly gives the result of pooling ϕ^{gmp}, where $\mathbf{1}_{\mathrm{N}}$ is the N-dimensional vector of all ones. The second is the dual formulation:

$$\alpha_\lambda = \arg\min_{\alpha} \left\|\Phi^T\Phi\alpha - \mathbf{1}_{\mathrm{N}}\right\|^2 + \lambda\|\Phi\alpha\|^2 \tag{2}$$

*Corresponding author: carrierlxk@gmail.com

which gives the weight of each feature. ϕ^{gmp} is the result of weighted sum pooling.

3. SPARSE FISHER VECTOR THEORY

Let $X = \{x_1, ...x_N\}$ be a set of N local descriptors extracted from an image. We denote M the number of Gaussian Mixture Model(GMM) clusters, and D the dimension of SIFT descriptors after using PCA. Clearly. According to Section 2, the Fisher vector representation ϕ should be equally similar to each Fisher vector code, which is defined as:

$$\Phi^T \phi = \mathbf{1}_N \tag{3}$$

where Φ is the code matrix, of which each row represents a Fisher vector code corresponding to the descriptor x_n.

$$\Phi = \begin{pmatrix} G_1(x_1) & \cdots & G_M(x_1) \\ \vdots & \ddots & \vdots \\ G_1(x_N) & \cdots & G_M(x_N) \end{pmatrix} \tag{4}$$

where $G_m(x_n)$ is the sub-vector of cluster m of the Fisher vector code corresponding to x_n.

In [1], the normalization of the Fisher information matrix takes a diagonal form, which assumes the sub-vectors are independent of each other. Therefore it is natural to divide Eq. 3 into M subtasks. We denote by Φ_m the m-th column of Φ, which is the code matrix in the m-th subtask:

$$\Phi_m = \begin{pmatrix} G_m(x_1) \\ \vdots \\ G_m(x_N) \end{pmatrix} \tag{5}$$

And $\Phi = (\Phi_1 \cdots \Phi_M)$. If each subtask is fulfilled as follows, the whole task likes Eq. 3 will be fulfilled as:

$$\Phi_m^T \phi_m = \mathbf{1}_N \tag{6}$$

The objective function of the m-th subtask in the primal formulation is:

$$\phi_{m,gmp} = \arg\min \left\| \Phi_m^T \phi_m - \mathbf{1}_N \right\| + \lambda \left\| \phi_m \right\|_2^2 \tag{7}$$

Clearly the primal formulation does not have the sparsifying effect, so we turn to the dual formulation. According to Section 2, we denote by α_m the code weight so that $\Phi_m \alpha_m = \phi_m$, which means that ϕ_m is the pooling result of code matrix Φ_m with weight α_m [17]. α_m is consistent with the idea of Sparse Fisher vector because it can determine whether a Fisher vector code is valid in the final image representation.

The objective function of the m-th task in the dual formulation is:

$$\alpha_{m,gmp} = \arg\min \left\| \Phi_m^T \Phi_m \alpha_m - \mathbf{1}_N \right\|_2^2 + \lambda \left\| \Phi_m \alpha_m \right\|_2^2 \tag{8}$$

For convenience, we substitute K for $\Phi_m^T \Phi_m$. The analytical solution to the dual formulation is:

$$\alpha_{m,gmp} = (K + \lambda I)^{-1} \mathbf{1}_N \tag{9}$$

The analytical solution indicates that we can leverage the individual items of α_m which are the weights of the Fisher vector codes in the m-th subtasks. If the weight is zero, then the corresponding descriptor makes no contributions in the pooling. In other words, the m-th component of the Fisher Vector code is sparsified, whose idea is like FV sparsity encoding in [3].

In LLC [12], weighted L2-norm constraint is used to assure that the local atoms are preserved, which inspires us to use a similar regularity to leverage the sparsity of α_m, let $\alpha_{m,sfv}$ denote the SFV representation,

$$\alpha_{m,sfv} = \arg\min \left\| K\alpha_m - \mathbf{1}_N \right\|_2^2 + \lambda \left\| d^m \odot \alpha_m \right\|_2^2 \tag{10}$$

where d gives different constraints to the individual items of α_m. Specially,

$$d_j^m = \begin{cases} 1 & if\ j \in \mathcal{N}_k^m \\ \infty & otherwise \end{cases} \tag{11}$$

$\mathcal{N}_k^m$ denotes the first k maximum posterior of m-th cluster. The analytical solution is:

$$\alpha_{m,sfv} = \left(K^T K + \lambda diag^2(d) \right)^{-1} K \mathbf{1}_N \tag{12}$$

When λ approaches infinity, $K^T K$ will be comparatively negligible, and the solution can be written as:

$$\alpha_{m,sfv} = \left(\lambda diag^2(d) \right)^{-1} K \mathbf{1}_N \tag{13}$$

Eq. 13 sparsifies the items in α_m that are heavily constrained by d, but the weights of the unsparsified descriptors are determined by $K\mathbf{1}_N$, which is time-costly. Therefore, we make a further simplification. K is the kernel matrix of patch-to-patch similarities. Clearly $\alpha_{m,sfv}$ only depends on K: when a feature shows little similarity with the other features, the corresponding weight α will be greater. Because the Fisher vector codes are all normalized, the diagonal items of K are all ones. If we ignore the non-diagonal items of K which means that the Fisher vector codes are orthogonal, Eq. 13 goes to: $\alpha_{m,sfv} = \left(\lambda diag^2(d) \right)^{-1} \mathbf{1}_N$.

Because λ will be eliminated by normalization, the individual item of $\alpha_{m,sfv}$ can be written as:

$$\alpha_{m,sfv}^{(j)} = \begin{cases} 1 & d_j^m = 1 \\ 0 & d_j^m = \infty \end{cases} \tag{14}$$

where $\alpha_{m,sfv}^j$ is the j-th term of $\alpha_{m,sfv}$, and d_j^m is the j-th term of d^m. As $\Phi_m \alpha_m$, sparse α_m makes Φ_m be sparsified, i.e., Sparse Fisher vector.

For $\lambda = 0$, we have $\alpha_{m,sfv} = \mathbf{1}_N / \lambda$, which corresponds to the original Fisher vector. Therefore, λ does not only play a role in regularization, but also realize a smooth transition between the solution to original Fisher vector ($\lambda = 0$) and Sparse Fisher vector ($\lambda \to \infty$).

4. EXPERIMENT EVALUATIONS

To verify the effectiveness of Sparse Fisher vector, we validate the proposed approach on image category task. Firstly, we describe the image classification datasets and experimental setup. We experimentally compare the Sparse Fisher vector against the canonical Fisher vector for two large data sets: Caltech-101 by Fei-Fei et al. [19] and the Pascal VOC sets of 2007 [20] .

4.1. Experimental setup

We compute all SIFT descriptors on overlapping 32×32 pixels patches with the step size of 4 pixels. We reduce their dimensionality to 64 dimensions with PCA, so as to better fit the diagonal covariance matrix assumption.

EM algorithm is employed to learn the parameters of the GMM and the cluster number ranges from 64 to 256. By default, for Fisher vector, we calculate the gradient with respect to mean and standard

Table 1: Experiment results on PASCAL VOC 2007

Coding ways	Feature Dims	Accuracy(%)	Time per image(s)
FV(M=32)	4096	49.75	3.14
FV(M=64)	8192	52.48	5.43
FV(M=128)	16384	55.18	9.73
FV(M=256)	32768	58.42	16.97
SFV(M=32)	4096	49.76	1.04
SFV(M=64)	8192	52.49	1.81
SFV(M=128)	16384	55.18	2.39
SFV(M=256)	32768	58.25	3.72
BOW(M=8192)	8192	39.60	2.45
FV(M=256)[43]	32768	58.3	–

deviation. And for the Sparse Fisher vector we set the neighborhood as $k = 5$. We streamline the standard experimental setting and employ linear SVM. It is worth mentioning that the computing platform in our experiments is Intel Core Duo (4G RAM), so the results are slightly different with origin paper in computation time. We use the origin Fisher vector [1, 2] as the baseline and also the Sparse Fisher vector is improved based on origin Fisher vector.

4.2. PASCAL VOC 2007

The Pascal VOC 2007 database contains 9,963 images of 20 classes. We use the standard protocol which consists in training on the provided trainval set and testing on the test set and we set the BOW model as the baseline. The classification results are compared in Table 1, where M denotes the number of clusters in GMM. We compared three sections in different coding ways, including feature dimensions, accuracy and coding time per image.

For the same feature dimension, for example 8192, the FV achieves higher accuracy than BOW. This result shows that the FV is more discriminative than BOW with the double time cost. But for SFV, when the cluster number of Gaussian mixture distributions(GMM) is 64, we can obtain a comparable accuracy with FV but much faster image coding. This result is in accordance with the conditions of 32, 128 and 256 clusters number.

4.3. Caltech-101

Caltech 101 dataset consists of 9144 images of 102 classes like animals, flower and so on. Following the standard experimental setting, we use 30 images per class for training while leaving the remaining for test. Other experimental setting agrees with experiment setup above. Classification results are compared in Table 2. Table 2 shows

Table 2: Experiment results on Caltech-101

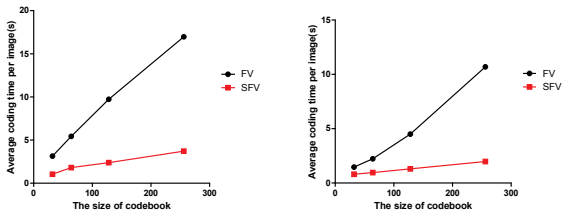
Coding ways	Accuracy(%)	Time per image(s)
FV(M=32)	61.00	1.46
FV(M=64)	65.09	2.33
FV(M=128)	67.85	4.50
FV(M=256)	70.79	10.69
SFV(M=32)	61.05	0.81
SFV(M=64)	65.03	0.96
SFV(M=128)	67.82	1.30
SFV(M=256)	70.75	1.98

the similar result as Table 1. Under the same size of codebook, SFV runs more quickly than FV with a comparable accuracy. And with the increase in codebook size, the difference of time consuming between these two coding ways is increasing. For example, when the codebook size is 256, coding time per image in FV is 10.69 s, while for SFV is 1.98 s which is nearly 5 times as fast as FV.

4.4. Experiment analysis

4.4.1. Computation cost analysis

To further show the advantage of SFV in computation cost, we demonstrate the average coding time per image with the size of codebook and analyze the computation complexity.



(a) PASCAL VOC 2007 (b) Caltech 101

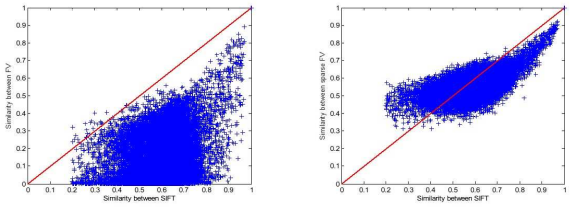
Fig. 1: Comparison between theoretical and empirical results, Black line indicate the origin FV and the red one indicate the SFV.

Fig.1(a) and Fig. 1(b) show the average coding time per image as a function of the codebooks size on datasets above. As was the case on both datasets, SFV consistently outperforms the FV and the computation time difference increase with the codebooks size.

Considering the D dims of features and M clusters mentioned above, we can estimate the computation complexity. There are two sub-steps in FV encoding steps: the first sub-step is calculating the posterior probability and the second sub-step is calculating the derivation on the GMM. The computation complexity of the first step is $O[3MD]$ and 3 represents the number of mathematical operations(e.g., floating point multiplications) which is same for FV and SFV. The computation complexity of the second step is $O[(3+5)MD]$ and $O[(3+5)kD]$ respectively. As $M \gg k$, so the total time of SFV is much less than FV and the time difference increases with M which is consist with experiment results. What's more, because the computation complexity of second step in SFV is independent of the size of codebooks, the holistic computation complexity almost remains unchanged with the size increasing.

4.4.2. Similarity correspondence between SIFT and Sparse Fisher vector

One implicit contribution of our work is that SFV better preserves similarity. To demonstrate this, 200 SIFT features from PASCAL VOC 2007 are randomly selected. We calculate the pair-wise similarity by using cosine measure. The similarity correspondence is shown in Fig. 2. Fig. 2 indicates an obvious linear trend of the similarity between SFV against the similarity between SIFT features, while FV does not. The comparison confirmed that the effectiveness of preserving configuration space locality during coding, which makes similar inputs correspond to similar codes [14, 21].



(a) FV (b) SFV

Fig. 2: Experiment result on Pascal VOC 2007. The similarity correspondence relationship between the FV(left) or SFV(right) and the SIFT feature. A linear trend can be found in SFV.

4.5. Discussion about SFV

In Fisher vector, local features are described by deviation from a GMM. The probability representation of a feature by GMM can be represented as:

$$\begin{aligned} p(x|\theta) &= \sum_{m=1}^{M} \omega_m p_m(x|\theta) \\ p_m(x|\theta) &= \frac{exp(-\frac{1}{2}(x-\mu_m)^T \Sigma_m^{-1}(x-\mu_m))}{(2\pi)^{D/2}|\Sigma_m|^{1/2}} \end{aligned} \tag{15}$$

where ω_m denotes the prior of the codeword and $p_m(x|\theta)$ reflects the probability of feature x belongs to the m-th cluster. So we can regard the feature coding coefficient as the probability of a feature belonging to the codebook. We notice that no matter in LLC [12], or LSC [13], codewords in codebook are independent and there are no priors on them or we can regard the priors as equal. For LSC, Eq.15 can be rewritten as:

$$\begin{aligned} p(x|B) &= \sum_{m=1}^{M} p_m(x|B) \\ p_m(x|B) &= \exp\left(\|x - b_m\|_2^2 / \sigma\right) \end{aligned} \tag{16}$$

Eq.16 can be seen as the probability of input feature x belonging to the m-th codeword [22], where M denotes the number of codewords in codebook. So the object function of LSC can be represented as:

$$\begin{aligned} &\max P(x|B) = \sum_{m=1}^{M} P_m(x|B) \odot I(m) \\ &s.t \quad \|I\|_0 = k \end{aligned} \tag{17}$$

where I is a binary vector.

Also we need to notice that all dimensions of soft coding [13, 23] are independent of each other. In Fisher coding, the relations among different dimensions are represented by GMM. The object function of SFV can be represented as:

$$\begin{aligned} &\max \gamma(m) = P(m|x,\theta) = \frac{P(m)P_m(x|\theta)}{\sum P(m)P_m(x|\theta)} \odot I(m) \\ &s.t \quad \|I\|_0 = k \end{aligned} \tag{18}$$

where I is a binary vector.

So when we execute the localization operation in Eq. 16, we calculate the codewords which belong to the k-nearest neighborhood of the feature. This can be regarded as the soft maximum of the likelihood of conditional probability. This is also true for LLC model. But in SFV, when we execute the early cut off operation, the prior of the codeword is incorporated. So we calculate the codewords which belong to the k-nearest neighborhood of the feature as Eq. 15. This can be regarded as a soft maximum of the posterior probability.

5. CONCLUSION

In this paper, we have introduced a 'localized' Fisher vector called Sparse Fisher vector. Based on GMP, we sparsified the Fisher vector code matrix by adding local regular term. These ways allow efficient image categorization without undermining its performance on several public datasets and coding outputs preserve the similarity among input features.

Fisher vector origins from the natural gradient in [24], so Sparse Fisher vector can be seen as partial gradient descent. Also, from probabilistic perspective, Sparse Fisher vector can be regarded as a soft maximum of the posterior probability. Since GMP considers the uniqueness of features and weight them according to uniqueness, we will combine it in our future work.

6. ACKNOWLEDGEMENT

This work was supported in part by the National Basic Research Program of China(2012CB719903).

7. REFERENCES

[1] Florent Perronnin and Christopher R. Dance. Fisher kernels on visual vocabularies for image categorization. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*, 2007.

[2] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV*, pages 143–156, 2010.

[3] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob J. Verbeek. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*, 105(3):222–245, 2013.

[4] Herve Jegou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 3304–3311, 2010.

[5] Florent Perronnin, Yan Liu, Jorge Sánchez, and Herve Poirier. Large-scale image retrieval with compressed fisher vectors. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 3384–3391, 2010.

[6] Ramazan Gokberk Cinbis, Jakob J. Verbeek, and Cordelia Schmid. Segmentation driven object detection with fisher vectors. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 2968–2975, 2013.

[7] Dan Oneata, Jakob J. Verbeek, and Cordelia Schmid. Efficient action localization with approximately normalized fisher vectors. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 2545–2552, 2014.

[8] Koen E. A. van de Sande, Cees G. M. Snoek, and Arnold W. M. Smeulders. Fisher and VLAD with FLAIR. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 2377–2384, 2014.

[9] Jie Lin, Ling-Yu Duan, Tiejun Huang, and Wen Gao. Robust fisher codes for large scale image retrieval. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, pages 1513–1517, 2013.

[10] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[11] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pages 849–856, 2001.

[12] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas S. Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 3360–3367, 2010.

[13] Lingqiao Liu, Lei Wang, and Xinwang Liu. In defense of soft-assignment coding. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 2486–2493, 2011.

[14] Shenghua Gao, Ivor Wai-Hung Tsang, Liang-Tien Chia, and Peilin Zhao. Local features are not lonely - laplacian sparse coding for image classification. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 3555–3561, 2010.

[15] Kai Yu and Tong Zhang. Improved local coordinate coding using local tangents. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 1215–1222, 2010.

[16] Jianchao Yang, Kai Yu, and Thomas S. Huang. Efficient highly over-complete sparse coding using a mixture model. In *Computer Vision - ECCV 2010 - 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part V*, pages 113–126, 2010.

[17] Y-Lan Boureau, Nicolas Le Roux, Francis Bach, Jean Ponce, and Yann LeCun. Ask the locals: Multi-way local pooling for image recognition. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 2651–2658, 2011.

[18] Naila Murray and Florent Perronnin. Generalized max pooling. pages 2473–2480, 2014.

[19] Fei-Fei Li, Robert Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.

[20] Mark Everingham, Luc J. Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge, 2010.

[21] Shenghua Gao, Ivor Wai-Hung Tsang, and Liang-Tien Chia. Laplacian sparse coding, hypergraph laplacian sparse coding, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):92–104, 2013.

[22] Yongzhen Huang, Zifeng Wu, Liang Wang, and Tieniu Tan. Feature coding in image classification: A comprehensive study. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(3):493–506, 2014.

[23] Jan van Gemert, Cor J. Veenman, Arnold W. M. Smeulders, and Jan-Mark Geusebroek. Visual word ambiguity. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(7):1271–1283, 2010.

[24] Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.