Boosting VLAD with Supervised Dictionary Learning and High-Order Statistics

Xiaojiang Peng[1,4,3,⋆], Limin Wang[2,3,⋆], Yu Qiao[3], and Qiang Peng[1]

[1] Southwest Jiaotong University, Chengdu, China
[2] Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China
[3] Shenzhen Key Lab of CVPR, Shenzhen Institutes of Advanced Technology, CAS, Shenzhen, China
[4] Hengyang Normal University, Hengyang, China

Abstract. Recent studies show that aggregating local descriptors into super vector yields effective representation for retrieval and classification tasks. A popular method along this line is vector of locally aggregated descriptors (VLAD), which aggregates the residuals between descriptors and visual words. However, original VLAD ignores high-order statistics of local descriptors and its dictionary may not be optimal for classification tasks. In this paper, we address these problems by utilizing high-order statistics of local descriptors and peforming supervised dictionary learning. The main contributions are twofold. Firstly, we propose a high-order VLAD (H-VLAD) for visual recognition, which leverages two kinds of high-order statistics in the VLAD-like framework, namely diagonal covariance and skewness. These high-order statistics provide complementary information for VLAD and allow for efficient computation. Secondly, to further boost the performance of H-VLAD, we design a supervised dictionary learning algorithm to discriminatively refine the dictionary, which can be also extended for other super vector based encoding methods. We examine the effectiveness of our methods in image-based object categorization and video-based action recognition. Extensive experiments on PASCAL VOC 2007, HMDB51, and UCF101 datasets exhibit that our method achieves the state-of-the-art performance on both tasks.

1 Introduction

Effective representation of image and video is crucial for visual recognition such as object recognition and action recognition. One popular representation is Bag of Visual Words (BoVW) model with local descriptors [5,36,22]. Approaches along this line include vector quantization (VQ) [27], sparse coding (SC) [39], soft-assignment (SA) [19], locality-constrained linear coding (LLC) [34], Fisher vector (FV) [23], and vector of locally aggregated descriptors (VLAD) [12]. These methods start from extracting local low-level descriptors (e.g., SIFT [20] or HOG, HOF, MBH [32]), then learn a codebook or dictionary from training set, encode descriptors to new vectors, and finally aggregate them to a global vector. After normalization, these vectors are used to train a classifier for visual classification.

⋆ Indicates equal contribution.

D. Fleet et al. (Eds.): ECCV 2014, Part III, LNCS 8691, pp. 660–674, 2014.

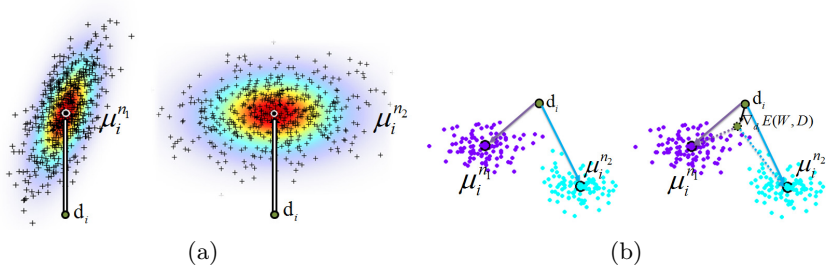


Fig. 1. The illustration of VLAD problem and dictionary refining process. (a) The descriptors ("+") assigned to word $\mathbf{d}_i$ in the n_1-th and n_2-th samples from different classes share the same mean. It results in two similar VLAD representations. However, the discrimination is preserved when incorporating high-order statistics of the assigned descriptors. (b) For the dictionary, our supervised learning method tunes the dictionary to minimize the classification error E and achieves better cosine-like similarity measure.

Recent study works show that super vector based encoding methods provide successful representations for visual recognition [5,36,22]. VLAD [12] is a kind of efficient super vector encoding method. For VLAD, it trains a codebook in the feature space using K-means. Each block of VLAD can be viewed as the difference between the mean of the descriptors assigned to the visual word and the word itself. VLAD can be efficiently computed and its effectiveness has been verified in several tasks, such as instance retrieval [12,1], scene recognition [8], and action recognition [11]. However, there are still two main issues about VLAD representation:

- It ignores the high order information of the descriptor distribution. As illustrated in Fig. 1 (a), the descriptors assigned to word $\mathbf{d}_i$ in the n_1-th and n_2-th samples share the same means. This results in two similar aggregated vectors by original VLAD method. However, the distributions of the two sets of descriptors are obviously different.
- The dictionary is another important issue for VLAD [1]. The similarity between two VLAD vectors is more sensitive to the visual words. As illustrated in the Fig. 1 (b), the two VLAD blocks, generated by $\mathbf{d}_i$ for the n_1-th and n_2-th samples are deemed to be similar due to the acute angle between them. But in practice, the two sets of descriptors may come from different categories, and their similarity is desired to be not large.

To address this issues, we introduce two important methods to boost the representation capacity of VLAD. Firstly, we leverage two high-order statistics in the VLAD-like framework, including diagonal covariance and skewness, to construct a high-order version of VLAD (H-VLAD). The covariance of descriptors reflects the distribution shape which is beneficial for classification. We utilize the residuals between the diagonal covariance from clusters and that from assigned descriptors to enhance the original VLAD. Mean and covariance are sufficient to describe a pure Gaussion distribution [9]. However, as shown in [13], there always exist heavy

tails for the distributions of gradient-based features. Therefore, we also test the third-order statistics, namely skewness, which capture the asymmetry of the descriptors around the mean.

Secondly, in order to enhance the discriminative power of VLAD, we propose a supervised dictionary learning (SDL) method. The VLAD with supervised dictionary are called S-VLAD. Our novel SDL method jointly optimizes the dictionary and classifier, which is solved by updating the visual words and learning model alternately. A slight update of the visual word will revise the similarity at the desired direction as shown in Fig. 1 (b). It is worth **noting that** there are plenty of research works on SDL for traditional encoding methods [3,30], but to our best knowledge, we are the first to introduce SDL for VLAD encoding, which is independent with the recent work of Deep Fisher Kernel [29], where the GMM parameters are discriminatively tuned in Fisher Vector framework.

The main contributions of this paper can be summarized as follows: (i) we extend VLAD with high-order statistics while keeping both high performance and high extraction speed (Section 3). (ii) we are the first to explore supervised dictionary learning for VLAD and verify its effectiveness (Section 4). (iii) Our method obtains the state-of-the-art performance on several challenging benchmarks including PASCAL VOC 2007, HMDB51 and UCF101 datasets for object and action recognition (Section 5).

2 Related Works

Super Vector Based Encoding Method. Bag of Visual Words (BoVW) [5,36] model with local descriptors has become a popular method for visual recognition and super vector based encoding methods [40,12,23] have obtained the state-of-the-art performance in several tasks. Super vector based encoding methods yield very high dimensional representations by aggregating high order statistics and typical methods include Super Vector Coding (SVC) [40], Vector of Locally Aggregated Descriptors [12], and Fisher Vector [23]. SVC assumed to learn a smooth nonlinear function $f(x)$ defined on a high dimensional space and derive a good coding scheme $\phi(x)$ to approximate $f(x)$ in a linear form $\omega^\top \phi(x)$. The resulting super vector coding $\phi(x)$ can be viewed as a super vector aggregating zero order and first order statistics. FV [23] was derived from Fisher Kernel [10] by representing the sample using the parameter gradient vector of log likelihood. In practice, FV aggregates not only the first order statistics, but also the second order statistics. Thus, its performance is usually better than VLAD, where only the first order statistics is kept. The idea of augmenting VLAD with high order information is inspired by these super vector encoding methods. However, this augmenting method shares two advantages: (i) it is able to bridge the performance gap between VLAD and FV; (ii) it also shares the high speed of VLAD.

Supervised Feature Learning. Feature learning (or deep learning) [2] has become more popular in computer vision community. Among them, the discriminatively trained deep convolutional neural networks (CNN) [18] have recently

achieved impressive state-of-the-art results over a number of areas, including object recognition [16] and action recognition [26]. One of the main advantages of CNN is that it is able to supervised learn the network parameters according to specific task from a large dataset. The idea of supervised learning has been extended to traditional methods, such as sparse coding [3], soft assignment [30]. These methods mainly resorted to jointly optimize the dictionary and classifiers. However, they did not deal with super vector based encoding methods. The latest paper [29] designed an end-to-end learning method to discriminatively tune the GMM parameters for Fisher vector. Our work of supervised dictionary learning for VLAD is independent with them and we obtains much better results on the PASCAL VOC 2007 dataset.

3 Augmenting VLAD with High Order Statistics

In this section, we first review the original VLAD computation and its corresponding normalization operation. We then introduce adding high-order statistics in VLAD computation framework.

3.1 VLAD Review

VLAD is proposed by Jégou *et al.* in [12]. Similar to standard BoVW, a dictionary $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_K] \in \mathcal{R}^{d \times K}$ is first learned by K-means from training samples. Let $\mathbf{X} = [\mathbf{x}_1, ...\mathbf{x}_N] \in \mathcal{R}^{d \times N}$ denote a set of local descriptors from a video V. For each codeword $\mathbf{d}_k$, a vector $\mathbf{v}_k$ is yielded by aggregating the differences between the assigned descriptors and codeword $\mathbf{d}_k$:

$$\mathbf{v}_k = \sum_{\mathbf{x}_j : NN(\mathbf{x}_j) = k} (\mathbf{x}_j - \mathbf{d}_k), \tag{1}$$

where $NN(\mathbf{x}_j)$ denotes that the nearest neighborhood of $\mathbf{x}_j$ in $\mathbf{D}$. The VLAD representation is the concatenation of all the d-dimensional vectors $\mathbf{v}_k$ and is therefore a Kd-dimensional vector. The representative capacity of VLAD can be enhanced by pre-processing the local features with PCA-Whitening [6,22], and performing intra-normalization on the final representation [1]. Thus, the final representation of VLAD is expressed as follows:

$$\psi(\mathbf{X}, \mathbf{D}) = \left[\frac{\mathbf{v}_1}{||\mathbf{v}_1||_2}; ...; \frac{\mathbf{v}_i}{||\mathbf{v}_i||_2}; ...; \frac{\mathbf{v}_K}{||\mathbf{v}_K||_2} \right]. \tag{2}$$

3.2 High-Order VLAD

This section introduce a simple yet effective method to augment original VLAD, which is motivated by the fact that VLAD will lose discriminative capacity in two cases. The first case is as shown in Fig. 1(a) that it would yield the same VLAD representation when the two sets of assigned descriptors share the same mean. We call this problem as "*share-means*". Another one is that those zero

aggregated vectors are ambiguous, because of both the facts that no descriptor is assigned to the codeword and that the mean of assigned features is equal to the codeword. We call this problem as "*evil-zeros*". In order to solve these problems, we propose a higher-order VLAD (H-VLAD), which makes use of two high-order statistics in the VLAD-like framework, including diagonal covariance and skewness. The technical details are as follows.

The original version of VLAD defined in Equation (1) can be rewritten as:

$$\mathbf{v}_k = N_k(\frac{1}{N_k}\sum_{j=1}^{N_k}\mathbf{x}_j - \mathbf{d}_k) = N_k(\mathbf{m}_k - \mathbf{d}_k), \tag{3}$$

where N_k is the number of descriptors assigned to codeword $\mathbf{d}_k$, which can be omitted when the intra-normalization is used, and $\mathbf{m}_k$ is the mean of these descriptors assigned to codeword $\mathbf{d}_k$. Thus, the original VLAD can be interpreted as the difference between the mean of descriptors and codeword. Similarly, using covariance, we formulate the second-order super vector as follows:

$$\mathbf{v}_k^c = \hat{\sigma}_k^2 - \sigma_k^2 = \frac{1}{N_k}\sum_{j=1}^{N_k}(\mathbf{x}_j - \mathbf{m}_k)^2 - \sigma_k^2, \tag{4}$$

where the square of a vector is element-wise one and σ_k^2 is the diagonal elements of covariance matrix of the k-th cluster.

As for standard Gaussian distribution, the first and second statistical information is sufficient to determine the distribution. However, low-level descriptors (e.g., SIFT) are not usually Gaussian distribution in reality [13]. Therefore, we also employ the third-order statistics (skewness) to exploit extra complementary information. Skewness is a measure of the asymmetry of the data around the sample mean. We formulate third-order super vector as follows,

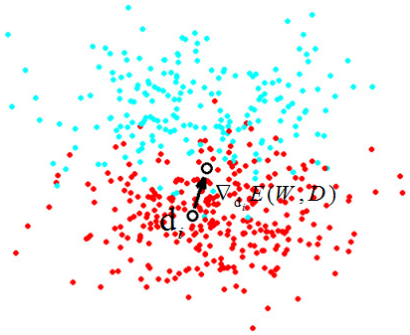
$$\mathbf{v}_k^s = \hat{\gamma_k} - \gamma_k = \frac{\frac{1}{N_k}\sum_{j=1}^{N_k}(\mathbf{x}_j - \mathbf{m}_k)^3}{(\frac{1}{N_k}\sum_{j=1}^{N_k}(\mathbf{x}_j - \mathbf{m}_k)^2)^{\frac{3}{2}}} - \gamma_k, \tag{5}$$

where the power of a vector is also the element-wise one. The γ_k is the skewness of k-th cluster.

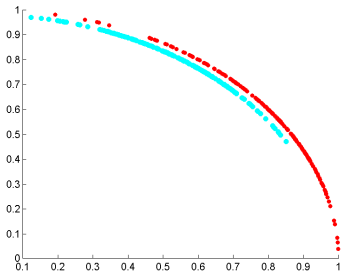
After intra-normalization separately, these two extra vectors are concatenated to the original VLAD to form a longer representation which is *the final representation of our H-VLAD*. Note that the statistics in Equation (3,4,5) can be quickly computed using Matlab toolbox. The H-VLAD requires no soft weight computation and contains higher statistical information compared with FV.

4 Supervised Dictionary Learning for VLAD

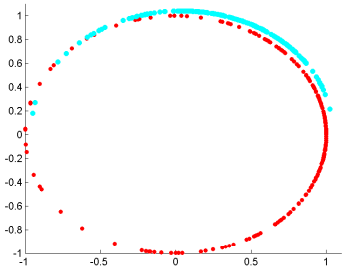
In this section, we first discuss the importance of dictionary for VLAD. Then, we formulate the supervised dictionary learning method for VLAD, and finally extend it to the spatial pyramid situation.



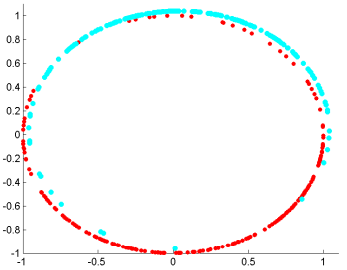
(a) Mean values: $\{\mathbf{m}_i^{c1}\}$ and $\{\mathbf{m}_i^{c2}\}$



(b) The normalized $\{\mathbf{m}_i\}$



(c) The normalized $\{\mathbf{m}_i\text{- }\mathbf{d}_i\}$



(d) The normalized $\{\mathbf{m}_i\text{- }\hat{\mathbf{d}}_i\}$

Fig. 2. Graphical interpretation of the importance of subtraction operation and a good dictionary. Two sets of mean values $\{\mathbf{m}_i^{c1}\}$ and $\{\mathbf{m}_i^{c2}\}$ from class $c1$ and $c2$ (a) are normalized by L2-norm and distribute in the 1st quadrant of a standard circle (b). After minus a particular anchor $\mathbf{d}_i$ (e.g., learned by K-means), they embed in the whole circle (c). An optimized anchor make the two sets of $\mathbf{m}_i$ more separate (d). Note that we make a micro shift to the points in (b), (c) and (d) for better visualization.

4.1 Importance of Dictionary

The dictionary plays an important role of subtractor in original VLAD. As illustrated in Fig. 2(a), two sets of VLAD blocks $\mathbf{m}_i^{c1}$ and $\mathbf{m}_i^{c2}$ defined in Equation (3) are from class $c1$ and $c2$ and scattered in a high-dimensional space. The separability of $\{\mathbf{m}_i^{c1}\}$ and $\{\mathbf{m}_i^{c2}\}$ determines the performance of final discriminative classifier. After normalization, all the vectors are projected to the 1st quadrant of a unit circle due to the positive property of descriptors (e.g., SIFT, HOG, etc) as shown in Fig.2(b). A linear classifier will produce a large error since almost half of samples overlap in this case. As for original VLAD, both sets of $\mathbf{m}_i^{c1}$ and $\mathbf{m}_i^{c2}$ will subtract an anchor vector $\mathbf{d}_i$ (the cluster center from K-means), then the normalized vectors will project to the whole unit circle, which make the samples more easily be separated (Fig. 2(c)). Since $\mathbf{d}_i$ affects the distribution of normalized vectors, we may find an optimized one to minimize the classification

error. When we update the $\mathbf{d}_i$ according to the negative gradient direction of the cost function as shown in Fig. 2(a), the normalized vectors will become much more easily to be separated using the discriminative dictionary (Fig. 2(d)).

4.2 Algorithm and Formulation

The original VLAD makes use of K-means to learn the dictionary. As discussed in the previous subsection, this may not be optimal for classification tasks. In this subsection, we formulate the dictionary learning and classification in a unified framework.

Due to its good performance, we consider the intra-normalization version of original VLAD [1]. The VLAD representation for the n-th training image or video is given by:

$$\phi_n = \psi(\mathbf{X}_n, \mathbf{D}) = [\phi_{n1}, \cdots, \phi_{nK}], \tag{6}$$

where ϕ_{ni} denotes the super vector of codeword $\mathbf{d}_i$ as defined in Equation (2). We aim to learn the dictionary $\mathbf{D}$ and classifier parameters $\mathbf{w}$ given N training samples by minimizing the following objective function:

$$E(\mathbf{w}, \mathbf{D}) = \sum_{n=1}^{N} \ell(y_n, f(\psi(\mathbf{X}_n, \mathbf{D}), \mathbf{w})) + \lambda ||\mathbf{w}||_2^2 \tag{7}$$

where y_n denotes the label of the n-th sample, $f(\psi(\mathbf{X}_n, \mathbf{D}), \mathbf{w})$ is the prediction model, ℓ denotes the loss function, and λ is a regularization parameter. Minimizing $E(\mathbf{w}, \mathbf{D})$ can be approached by optimizing alternately over $\mathbf{w}$ and $\mathbf{D}$. We utilize logistic regression and softmax activation function for binary and multi-class classification, respectively. This is because the performance of them are similar to that of linear SVMs but their cost functions are differentiable [3].

Consider the binary classification problem first, where $y_n \in \{0, 1\}$ and $f_n = \sigma(\mathbf{w}^T \phi_n)$. The σ denotes the sigmoid function $\sigma(x) = 1/(1 + \exp(-x))$. With cross-entropy loss, the cost function E is given by:

$$E = -\sum_{n=1}^{N} \{y_n \mathrm{In} f_n + (1 - y_n) \mathrm{In}(1 - f_n)\} + \lambda ||\mathbf{w}||_2^2. \tag{8}$$

The gradient of E over $\mathbf{w}$ is:

$$\nabla_{\mathbf{w}} E = \sum_{n=1}^{N} (f_n - y_n)\phi_n + 2\lambda \mathbf{w}. \tag{9}$$

Given dictionary $\mathbf{D}$, we can use gradient descent method to find optimal $\mathbf{w}$. Given the model $\mathbf{w}$, in order to optimize E over $\mathbf{D}$, we apply the chain rule to compute the gradient as follows:

$$\nabla_{\mathbf{d}_i} E = \sum_{n=1}^{N} \frac{\partial \ell}{\partial f_n} \frac{\partial f_n}{\partial \phi_n} \frac{\partial \phi_n}{\partial \mathbf{d}_i}. \tag{10}$$

The computational processes of the first two gradients in the right side are similar with that of $\nabla_{\mathbf{w}} E$. The key problem is reduced to compute the gradient of the VLAD representation ϕ_n over $\mathbf{d}_i$. Ignoring the effect of $\mathbf{m}_i$ (we set a small learning rate to meet this condition in practice), we can compute the gradient as follows:

$$\frac{\partial \phi_n}{\partial \mathbf{d}_i} = \left[\mathbf{0}, \cdots, \frac{\partial \phi_{ni}}{\partial \mathbf{d}_i}, \cdots, \mathbf{0}\right], \tag{11}$$

$$\frac{\partial \phi_{ni}}{\partial \mathbf{v}_{ni}} = \frac{1}{\|\mathbf{v}_{ni}\|_2}(\mathbf{I} - \phi_{ni}\phi_{ni}^{\top}), \tag{12}$$

$$\frac{\partial \mathbf{v}_{ni}}{\partial \mathbf{d}_i} = -N_{ni}\mathbf{I}, \tag{13}$$

where $\mathbf{I} \in \mathcal{R}^{d \times d}$ is a unit matrix. Therefore, the gradient of E over $\mathbf{d}_i$ is given by:

$$\nabla_{\mathbf{d}_i} E = -\sum_{n=1}^{N}(f_n - y_n)\frac{N_{ni}(\mathbf{I} - \phi_{ni}\phi_{ni}^{\top})}{||\mathbf{v}_{ni}||_2}\mathbf{w}_{(i)}, \tag{14}$$

where $\mathbf{w}_{(i)}$ is the i-th block of $\mathbf{w}$ with the same size of $\mathbf{d}_i$.

As for multi-class problem, y_n and f_n are C dimensional vectors and the activation function is $f_{nc} = \exp(\mathbf{w}_c\phi_n)/\Sigma_{j=1}^{C}\exp(\mathbf{w}_j\phi_n)$. Using cross-entropy loss, the cost function with regularization is defined by:

$$E = -\sum_{n=1}^{N}\sum_{c=1}^{C} y_{nc}\mathrm{In}f_{nc} + \lambda||\mathbf{w}||_2^2. \tag{15}$$

Applying the chain rule to compute the gradient of E over $\mathbf{d}_i$, we obtain:

$$\begin{aligned}\nabla_{\mathbf{d}_i} E &= \sum_{n=1}^{N}\sum_{c=1}^{C}\frac{\partial \ell}{\partial f_{nc}}\frac{\partial f_{nc}}{\partial \phi_n}\frac{\partial \phi_n}{\partial \mathbf{d}_i} \\ &= -\sum_{n=1}^{N}\frac{N_{ni}(\mathbf{I} - \phi_{ni}\phi_{ni}^{\top})}{||\mathbf{v}_{ni}||_2}\sum_{c=1}^{C}(f_{nc} - y_{nc})\mathbf{w}_{(i)c}.\end{aligned} \tag{16}$$

We summarize our supervised dictionary learning method for both binary and multi-class classification in Algorithm 1. The dictionary $\mathbf{D}$ is usually initialized by K-means from a subset of local descriptors.

Supervised Dictionary Learning with Spatial Pyramid. Spatial pyramid (SPM) is usually beneficial to image classification. Here we present the supervised dictionary learning method with spatial pyramid. Suppose M cells are used for the n-th sample, then the final VLAD representation with SPM is $\Phi_n = [\phi_n^1, ..., \phi_n^M]$. The predict model is then changed to $f(\Phi_n, \mathbf{w}) = \sigma(\sum_m \mathbf{w}^m\phi_n^m)$. Then, for the binary case, the gradient of E over $\mathbf{D}$ can be written as follows:

Algorithm 1. Supervised Dictionary Learning Algorithm for VLAD

Input : Local descriptors and labels in training data: $\{(\mathbf{X}_n, y_n)\}$. Parameters: K, $\lambda, \lambda_d, \lambda_w$

Output: Dictionary: $\mathbf{D}$, Predict Model: $\mathbf{w}$

Initialization: $\mathbf{D}^0 \leftarrow$ init by K-means, $\mathbf{w} \leftarrow$ randomly select from normal distribution;

0. Compute VLAD using $\mathbf{D}^0$

while $t < T$ *and* $\delta > \epsilon$ **do**

1. Update $\mathbf{D}$ fix $\mathbf{w}$: $\mathbf{D}^t \leftarrow \mathbf{D}^{t-1} - \lambda_d \nabla_{\mathbf{d}_i} E$;
2. Recompute VLAD using $\mathbf{D}^t$;
3. Optimize $\mathbf{w}$ fix $\mathbf{D}$: $\arg\min_{\mathbf{w}} E$;
4. $\delta \leftarrow$ error(t) - error(t-1);

end

$$\begin{aligned} \nabla_{\mathbf{d}_i} E &= \sum_{n=1}^{N} \frac{\partial \ell}{\partial f_n} \sum_{m=1}^{M} \frac{\partial f_n}{\partial \phi_n^m} \frac{\partial \phi_n^m}{\partial \mathbf{d}_i}. \\ &= -\sum_{n=1}^{N} (f_n - y_n) \sum_{m=1}^{M} \frac{N_{ni}^m (\mathbf{I} - \phi_{ni}^m \phi_{ni}^{m\top})}{||\mathbf{v}_{ni}^m||_2} \mathbf{w}_{(i)}^m, \end{aligned} \tag{17}$$

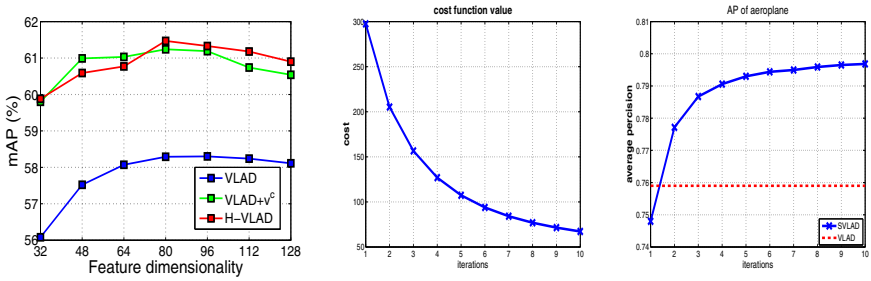
where N_{ni}^m denotes the number of descriptors assigned to word $\mathbf{d}_i$ at the m-th cell in n-th sample. Similar with Algorithm 1, the desired dictionary and the model with spatial pyramid can be optimized alternately.

5 Experiments

We verify the effectiveness of our proposed method on two recognition tasks, namely visual object categorization (PASCAL VOC2007 [7]) and human action recognition (HMDB51 [17] and UCF101 [28]). In this section, we first conduct extensive experiments on PASCAL VOC 2007 to evaluate the performance of our H-VLAD and S-VLAD. And then we apply them to video-based action recognition with a large number of classes.

5.1 Evaluation on Object Classification

Our first and most extensive experiments are conducted on the well-known PASCAL VOC2007 dataset [7]. This challenge is known as one of the most difficult image classification tasks due to significant variations both in appearances and poses even with occlusions. It consists of about 10,000 images with 20 different object categories. There are 5,011 training images (train+val sets) and 4,952 test images. The performance is evaluated by the standard PASCAL protocol which computes average precision (AP) based on the precision-recall curve. We also report the mean of AP (mAP) over 20 categories.



(a) (b) (c)

Fig. 3. (a) Performance of the VLAD and our H-VLAD with various local feature dimensionality on PASCAL VOC2007 without SPM. (b) The training cost curve for the supervised learning process. (c) The test accuracy curve of class aeroplane in the supervised learning process.

Implementation Details. We densely extract local SIFT descriptors with a spatial stride of 4 pixels at 9 scales, and the width of SIFT spatial bins is fixed as 8 pixels, which are the default parameters setting in the VLFeat toolbox [31], version 0.9.17. We learn dictionary from a subset of 256k SIFT descriptors. All descriptors are whitened after PCA processing. The regularized factor λ is fixed to 0.01, and the learning rates for $\mathbf{D}$ and $\mathbf{w}$ (λ_d and λ_w) are set to 0.001 and 0.01, respectively. To ensure the effective estimation, we compute high-order statistics only when N_i is larger than a threshold which is set to d (the same size as the dimension of feature) empirically. For all the encoding methods, the long vectors are post-processed by "power+intra" normalization. Note that during supervised dictionary learning process, no power-normalization is performed due to the gradient computation. As for SPM, we divide the image in $1 \times 1, 2 \times 2, 3 \times 1$ grids.

H–VLAD. Fig.3(a) illustrates the mAPs of VLAD, VLAD with the 2nd-order statistics, and H-VLAD (VLAD with 2nd and 3rd order statistics). The dictionary size are all fixed to 256 as common setting. All the approaches perform better when feature dimensionality increases, while reach the upper bounds at the dimension of 80. Adding the 2nd-order statistics boosts the performance (by 2.4%–3.8%) of original VLAD as expected. We find the result obtained by the single 2nd-order statistics ($\mathbf{v}^c$) is inferior to that of VLAD by 1.5%–3.5% when testing them separately. We argue that the locations of the means from local descriptors are more discriminative than the distribution shapes of descriptors, and the distribution shapes contain complementary information to the mean of distribution, which is beneficial for classification. We also find the result from only usage of 3rd-order statistics is inferior to that of 2nd-order. Our H-VLAD is superior to the others when the dimensions are larger than 80. This indicates the skewness can provide complementary information to the 1st and 2nd statistics.

For fair comparison, we extend the dimension of VLAD to the same as that of H-VLAD by increasing dictionary size. The first three columns in Table 1 show the detailed results for each category in VOC2007 dataset. From the 2nd

Table 1. Detailed results of VLAD and our H-VLAD with and without supervised dictionary. The number in the brackets of the 2nd and 3rd columns denote dictionary size.

category	VLAD(256)	VLAD(768)	H–VLAD	S–VLAD	SH–VLAD	SH–VLAD(SPM)
aeroplane	75.9	76.5	80.3	79.7	84.0	85.1
bicycle	65.8	67.6	69.7	68.7	69.7	70.5
bird	51.1	48.9	55.9	50.6	55.6	61.5
boat	73.6	74.5	74.2	74.7	74.3	79.9
bottle	28.9	29.3	32.1	32.8	31.1	*32.1*
bus	63.6	62.3	64.8	68.0	72.4	74.0
car	80.3	79.9	82.0	82.2	82.1	83.0
cat	59.3	59.7	61.5	60.3	62.9	66.6
chair	52.1	55.6	53.6	53.2	54.5	57.9
cow	44.0	46.7	49.4	47.9	50.7	53.6
diningtable	50.5	49.2	55.5	53.2	59.5	61.1
dog	42.9	45.9	44.4	45.3	44.3	48.3
horse	78.8	79.6	81.0	81.3	81.9	83.7
motorbike	62.4	62.3	65.4	67.7	68.9	69.4
person	85.0	84.8	86.6	86.4	86.7	87.3
pottedplant	26.1	26.8	32.1	31.2	33.1	36.6
sheep	46.1	46.0	46.2	48.2	49.2	51.8
sofa	49.6	51.1	53.6	54.2	57.8	57.9
train	76.3	74.9	80.2	78.8	84.6	85.3
tvmonitor	52.8	54.2	55.2	54.3	57.9	58.9
mAP	58.3	58.8	61.2	60.9	63.1	65.2

column of Table 1, it is clear that the performance improvement is very limited from increasing dictionary size and our H-VLAD is also superior to VLAD with the same dimension.

Supervised Dictionary. We evaluate the impact of dictionary on VLAD and our S-VLAD algorithm by fixing the PCA dimensionality as 80. First, we test the performance of three random dictionaries (randomly selected SIFT descriptors) and the mAPs are [51.0%, 50.9%, 50.0%]. This result verifies the importance of a good dictionary. Then, we initialize the supervised dictionary by K-means with the size fixed to 256 and learn a supervised dictionary with logistic function. Fig. 3(b) and (c) illustrates the optimization process of S-VLAD for class "aeroplane". We plot the cost function value and AP for each iteration. As expected, the AP increases and the cost function value decreases during iteration. The improvement is limited when the iteration reaches 8. Therefore, we fix the iterations to 8 for all the categories.

The last three columns in Table 1 show the results of all the categories by using supervised dictionary for VLAD. The notation "SH-VLAD" denotes the combination of H-VLAD and S-VLAD. We only conduct supervised dictionary learning for VLAD in current implementation. The supervised dictionary for both VLAD and H-VLAD can improve the performance, and the improvements are 2.6% and 1.9% for VLAD and H-VLAD, respectively. The performance becomes even

better when performing the proposed SH-VLAD with spatial pyramid scheme (the last column).

5.2 Application to Action Recognition

With the observation in VOC2007, we also perform experiments on the HMDB51 and UCF101 action datasets. The HMDB51 dataset [17] consists 51 action categories with 6,766 manually annotated clips which are extracted from a variety of sources ranging from digitized movies to YouTube. We follow the experimental settings in [17] and report the mean average accuracy over all classes. The UCF101 dataset [28] has been the largest action recognition dataset so far, and exhibits the highest diversity in terms of actions, with the presence of large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions and so on. It contains 13,320 videos collected from YouTube and includes total number of 101 action classes. We perform evaluation on three train/test splits [1] and report the mean average accuracy over all classes.

For both datasets, we densely extract improved trajectories using the code from Wang [33]. Each trajectory is described by HOG, HOF, and MBH descriptors. We reduce the descriptor dimensions by a factor of two using PCA+Whiten pre-processing. We use soft-max function in Algorithm 1 to learn dictionaries for each type of descriptor. All the super vectors are consistently normalized using the same strategy as VOC2007. To combine different features, we concatenate their final representations. A linear *one-vs-all* SVM with C=100 is used for classification.

Table 2 compares our methods with VLAD and the improved FV [23] on HMDB51 and UCF101 action datasets. For each individual type of features, our H–VLAD outperforms VLAD with a large margin on both datasets, and achieves very similar results as FV. This indicates the importance of high-order information. Supervised dictionaries are beneficial to recognition for all types of feature. Our SH–VLAD obtains best results on both datasets with a slight better than that of FV. Besides, we also present the time costs of VLAD, H–VLAD, and FV in Table 2. We implement these encoding methods in Matlab without any parallel processing and use KD-Tree to search the nearest neighbor for VLAD and H–VALD. We randomly select 10 videos, and compute the average cost per video for encoding all types of feature. Note that the dictionary does not affect the time cost in test phase. From Table 2, it is clear that the cost of our H–VLAD is very close to that of original VLAD, but largely lower than that of FV.

5.3 Comparison and Discussion

Table 3 compares our best results to several recently published results in the literature on each dataset. Our method outperforms these previously reported results on all datasets. As for VOC2007 dataset, the most similar performance

[1] `http://crcv.ucf.edu/ICCV13-Action-Workshop/`

Table 2. Performance and computational cost of VLAD, S–VLAD, H–VLAD, SH–VLAD, and FV on HMDB51 and UCF101 action datasets. Note that the time costs of VLAD and H–VLAD are largely less than that of FV.

	HMDB51					UCF101				
	VLAD	H–VLAD	S–VLAD	SH–VLAD	FV	VLAD	H–VLAD	S–VLAD	SH–VLAD	FV
HOG	35.9	42.1	37.5	45.1	42.7	67.5	73.3	68.9	74.1	73.1
HOF	46.7	49.0	47.9	50.7	50.8	75.2	76.8	76.4	78.3	77.3
MBHx	38.4	43.0	44.4	44.1	44.3	70.2	76.0	74.6	76.6	75.1
MBHy	43.2	47.9	48.5	50.0	49.0	73.6	78.1	76.9	78.6	77.6
Combined	55.5	58.3	57.1	59.8	58.5	84.8	86.5	85.9	87.7	86.7
Time (s)	**2.63**	**3.38**	–	–	**57.21**	–	–	–	–	–

Table 3. Comparison of our results to the state of the arts

VOC 2007		HMDB51		UCF101	
Methods	mAP	Methods	Accuracy	Methods	Accuracy
winner (2007)	59.4%	iDT+FV[33] (2013)	57.2%	winner [14] (2013)	85.9%
[5] (2011)	61.7%	[35] (2013)	42.1%	[4] (2014)	83.5%
[24] (2012)	57.2%	[32] (2013)	46.6%	[37] (2014)	84.2%
[15] (2013)	62.2%	[25] (2013)	47.6%		
[38] (2013)	64.1%	[21] (2013)	52.1%		
Our result	**65.2**%	Our result	**59.8**%	Our result	**87.7**%

with ours comes from [38]. They applied a layered model to PHOG and SIFT features to obtain a desired mid-level representation, and learned this representation in a *supervised* way. Compared with [38], our approach only use SIFT descriptors and is simpler, but achieves better performance. This should be partly ascribed to high-dimensional representations we used. For the HMDB51 and UCF101 datasets, our approach improve the state-of-the-art performance (57.2% and 85.9%) [33] by 2.6% and 1.8%, respectively, which may be because of the supervised dictionary.

6 Conclusion

This paper first proposes to enhance the VLAD representation by aggregating high-order information of local descriptors, which is called H–VLAD. The covariance and skewness are demonstrated to be complementary with the original VLAD in our experiments. We then discuss the importance of a good dictionary and propose the supervised dictionary learning method for VLAD, which we refer to S–VLAD. Adding supervised dictionary can further boost the performance of VLAD. Theoretically, our supervised dictionary learning method can be easily extended for other super vector based methods. We verify the effectiveness of our method for the tasks of object and action recognition. We conduct experiments on three challenging benchmarks: PASCAL 2007, HMDB51, and UCF101, and conclude that our method achieves the state-of-the-art performance.

Acknowledgements. This work is partly supported by Natural Science Foundation of China (91320101, 61036008, 60972111), Shenzhen Basic Research Program (JC201005270350A, JCYJ20120903092050890, JCYJ201 20617114614438), 100 Talents Program of CAS, Guangdong Innovative Research Team Program (201001D0104648280), and Jiangxiao Peng is supported by the construct program of the key discipline in Hunan province. Yu Qiao is the corresponding author.

References

1. Arandjelovic, R., Zisserman, A.: All about VLAD. In: CVPR (2013)
2. Bengio, Y., Courville, A.C., Vincent, P.: Representation learning: A review and new perspectives. TPAMI 35(8) (2013)
3. Boureau, Y.L., Bach, F., LeCun, Y., Ponce, J.: Learning mid-level features for recognition. In: CVPR (2010)
4. Cai, Z., Wang, L., Peng, X., Qiao, Y.: Multi-view super vector for action recognition. In: CVPR (2014)
5. Chatfield, K., Lempitsky, V.S., Vedaldi, A., Zisserman, A.: The devil is in the details: An evaluation of recent feature encoding methods. In: BMVC (2011)
6. Delhumeau, J., Gosselin, P.H., Jégou, H., Pérez, P., et al.: Revisiting the vlad image representation. In: ACM MM (2013)
7. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC 2007) Results (2007)
8. Gong, Y., Wang, L., Guo, R., Lazebnik, S.: Multi-scale orderless pooling of deep convolutional activation features. CoRR abs/1403.1840 (2014)
9. Hogg, R.V., Craig, A.: Introduction to mathematical statistics (1994)
10. Jaakkola, T., Haussler, D., et al.: Exploiting generative models in discriminative classifiers. In: NIPS (1999)
11. Jain, M., Jégou, H., Bouthemy, P.: Better exploiting motion for better action recognition. In: CVPR (2013)
12. Jégou, H., Perronnin, F., Douze, M., Schmid, C., et al.: Aggregating local image descriptors into compact codes. TPAMI (2012)
13. Jia, Y., Darrell, T.: Heavy-tailed distances for gradient based image descriptors. In: NIPS (2011)
14. Jiang, Y.G., Liu, J., Roshan Zamir, A., Laptev, I., Piccardi, M., Shah, M., Sukthankar, R.: THUMOS challenge: Action recognition with a large number of classes (2013), `http://crcv.ucf.edu/ICCV13-Action-Workshop/`
15. Kobayashi, T.: BoF meets HOG: Feature extraction based on histograms of oriented pdf gradients for image classification. In: CVPR (2013)
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
17. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: A large video database for human motion recognition. In: ICCV (2011)
18. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11) (1998)
19. Liu, L., Wang, L., Liu, X.: In defense of soft-assignment coding. In: ICCV (2011)
20. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV (2004)
21. Mihir, J., Jegou, H., Bouthemy, P.: Better exploiting motion for better action recognition. In: CVPR (2013)

22. Peng, X., Wang, L., Wang, X., Qiao, Y.: Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. CoRR abs/1405.4506 (2014)
23. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010)
24. Russakovsky, O., Lin, Y., Yu, K., Fei-Fei, L.: Object-centric spatial pooling for image classification. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 1–15. Springer, Heidelberg (2012)
25. Shi, F., Petriu, E., Laganiere, R.: Sampling strategies for real-time action recognition. In: CVPR (2013)
26. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. CoRR abs/1406.2199 (2014)
27. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: ICCV (2003)
28. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. ArXiv:1212.0402 (2012)
29. Sydorov, V., Sakurada, M., Lampert, C.H.: Deep fisher kernels - end to end learning of the fisher kernel gmm parameters. In: CVPR (2014)
30. Tariq, U., Yang, J., Huang, T.S.: Maximum margin gmm learning for facial expression recognition. In: FG Workshops (2013)
31. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008)
32. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. IJCV (2013)
33. Wang, H., Schmid, C., et al.: Action recognition with improved trajectories. In: ICCV (2013)
34. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: CVPR (2010)
35. Wang, L., Qiao, Y., Tang, X.: Motionlets: Mid-level 3D parts for human motion recognition. In: CVPR (2013)
36. Wang, X., Wang, L., Qiao, Y.: A comparative study of encoding, pooling and normalization methods for action recognition. In: ACCV (2012)
37. Wu, J., Zhang, Y., Lin, W.: Towards good practices for action video encoding. In: CVPR (2014)
38. Wu, R., Yu, Y., Wang, W.: Scale: Supervised and cascaded laplacian eigenmaps for visual object recognition based on nearest neighbors. In: CVPR (2013)
39. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR (2009)
40. Zhou, X., Yu, K., Zhang, T., Huang, T.S.: Image classification using super-vector coding of local image descriptors. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 141–154. Springer, Heidelberg (2010)