Action Recognition with Stacked Fisher Vectors

Xiaojiang Peng^{1,3,2}, Changqing Zou^{3,2}, Yu Qiao^{2,4,*} and Qiang Peng^{1}

^{1}Southwest Jiaotong University,Chengdu, China
^{2}Shenzhen key lab of CVPR, Shenzhen Institutes of Advanced Technology, CAS
^{3}Department of Computer Science, Hengyang Normal University, Hengyang, China
^{4}The Chinese University of Hong Kong, China

Abstract. Representation of video is a vital problem in action recognition. This paper proposes Stacked Fisher Vectors (SFV), a new representation with multi-layer nested Fisher vector encoding, for action recognition. In the first layer, we densely sample large subvolumes from input videos, extract local features, and encode them using Fisher vectors (FVs). The second layer compresses the FVs of subvolumes obtained in previous layer, and then encodes them again with Fisher vectors. Compared with standard FV, SFV allows refining the representation and abstracting semantic information in a hierarchical way. Compared with recent mid-level based action representations, SFV need not to mine discriminative action parts but can preserve mid-level information through Fisher vector encoding in higher layer. We evaluate the proposed methods on three challenging datasets, namely Youtube, J-HMDB, and HMDB51. Experimental results demonstrate the effectiveness of SFV, and the combination of the traditional FV and SFV outperforms state-of-the-art methods on these datasets with a large margin.

Keywords: Action recognition, Fisher vectors, stacked Fisher vectors, max-margin dimensionality reduction

1 Introduction

Action recognition in realistic videos has been an active research area in recent years due to its wide range of potential applications, such as smart video surveillance, video indexing, human-computer interface, etc. Though significant progresses have been made [31, 32, 26, 16], action recognition still remains a challenging task due to high-dimensional video data, large intra-class variations, camera motions and view point changes, and other fundamental difficulties [1].

By far, the most popular video representation for action recognition has been the Bag-of-Visual-Words (BoVW) model [29, 23] or its variants [21, 24] based on spatial-temporal local features. This representation mainly contains four steps: feature extraction, codebook generation, feature encoding and pooling, and normalization. As for traditional BoVW, we usually extract local features from videos, learn a visual dictionary in training set by k-means or Gaussian Mixture Model (GMM), encode features and pool them for each video, and finally

[$\star$] Corresponding Author.

normalize the pooled vectors as video representations. These representations are subsequently fed into a pre-trained SVM classifier. The good performance of BoVW model should be partly ascribed to the development of more elaborately designed low-level features (e.g., dense trajectory features [31, 32] and spatial-temporal co-occurrence descriptors [22]) and more sophisticated encoding methods (e.g., Fisher vector encoding [24]). Currently, the pipeline of Fisher vector encoding based on improved Dense Trajectory (iDT) features provides state-of-the-art results on most action datsets [32].

More recently, many efforts have focused on developing mid-level representations [19, 35, 34, 7, 37, 27] for action recognition. These methods usually mine discriminative action parts, such as attributes [19], motionlets [35], actons [37], and train a classifier for each type of parts, and then summarize the outputs of these classifiers as video representations by max-pooling. Therefore, the contribution of each subvolume for the final representation is summarized as a single value (if this subvolume obtains the highest response) or null (otherwise). This limits the capacity of the mid-level representations. From another aspect, hierarchical feature learning with deep network has attracted much attention for action recognition [18, 11, 12], which can partly alleviate the above dilemma. These works are partly inspired by the success of Deep Neural Network (DNN) for image representation and classification [14]. Though these methods can describe videos from low level features to more abstract and semantic representation using deep structures, they are very computationally expensive to directly learn effective deep neural network for video-based action recognition. Recently, improvement has also been observed in shallow but still hierarchically layered models based on traditional encoding methods for object classification [28, 25].

Inspired by these previous works, we propose Stacked Fisher Vectors (SFV), a new representation based on Fisher Vector (FV) encoding [24], for action recognition. Figure 1 compares the traditional single layer Fisher vector encoding method with our SFV. Unlike traditional single layer FV pipeline that directly encodes and summarizes all local descriptors of input video with Fisher vectors , our SFV pipeline first performs Fisher vector encoding in densely sampled subvolumes based on low-level features, and then discriminatively compresses these subvolume-level FVs, and finally employs another FV encoding layer based on compressed subvolume-level representations. Specially, subvolumes are extracted in multiple scales. As it is known, the raw FVs are too high-dimensional to serve as inputs for the next FV layer. To compress these high-dimensional vectors significantly, we learn a projection matrix via a max-margin learning framework (Section 4), which is very important for the performance of SFV. The compressed FVs delivered to the 2nd layer contain rich semantic information and are powerful to describe those large volumes as they come from high-dimensional space. Our experimental results on three popular datasets demonstrate that our SFV representation can provide significant complementary information w.r.t the traditional FV representation, and the SFV performs comparably with traditional FV. Specially, when combining SFV with traditional FV, we obtain significant-

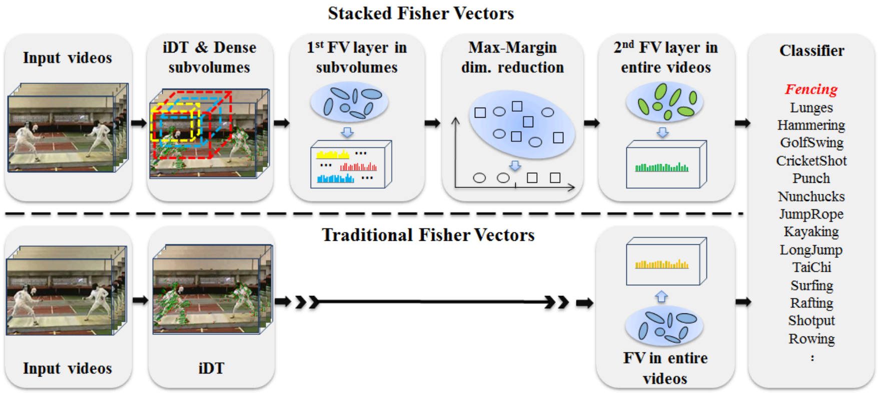


Fig. 1. Comparison between our approach and traditional Fisher vectors. Top: The pipeline of proposed Stacked Fisher vectors with two layers. Bottom: traditional pipeline of single layer Fisher vectors. The video representations of SFV are constructed based on large subvolumes which contain richer semantic information than those local cuboids.

ly superior recognition performance than the current state-of-the-art results on Youtube (93.77%), J-HMDB (69.03%), and HMDB51 (66.79%).

1.1 Related Work

Early researches in action recognition widely made use of low-level features with BoVW model. Typical low-level features in action videos include histogram of oriented gradients (HOG) [16], 3D-HOG [13], histogram of optical flow (HOF) [16] and motion boundary histogram (MBH) [30], which are computed in local cuboids obtained by spatial-temporal interesting points (STIP) detectors [17] or dense sampling schemes [33, 30]. These local features especially the dense trajectory features demonstrate excellent performance on many challenging datasets [33, 30, 32].

As discussed in [4, 36], selection of encoding methods is important to recognition performance in the BoVW framework. Recently, advanced feature encoding methods have been introduced for action recognition, such as soft-assignment [21, 36], vector of locally aggregated descriptors [9, 8], and Fisher coding [24, 36, 32]. In [36], Wang *et al.* evaluated most of these encoding methods for action recognition and observed that Fisher coding method performs the best among them. Wang *et al.* [32] also adopted this coding method with improved dense trajectory features, and obtained state-of-the-art results on many action datasets.

Besides those low-level features and encoding methods, recent efforts for action recognition have been devoted to mining discriminative mid-level action representations [19, 35, 7, 37, 27]. Wang *et al.* [35] developed motionlets which are defined as representative and discriminative 3D parts obtained by clustering and ranking algorithms. Jain *et al.* [7] learned discriminative cuboids by exemplar-SVM. Both Sapienza *et al.* [7] and Zhu *et al.* [37] adopted multiple instance learning framework to mine discriminative action parts or actons. Specially, all these methods made use of the part responses, and then pooled them as video representations. The mined 3D parts in this type methods are large subvolumes and expected to contain rich semantic information which is related to action categories. Along the line of this idea, but unlike these previous works, we do not mine discriminative action parts, and instead we encode densely sampled subvolumes via FV encoding, and project them to a low-dimensional subspace, and use another FV layer with those compressed FVs to construct video-level representations. Perhaps the most similar work to ours is Deep Fisher Networks proposed by Simonyan et al. [27]. Deep Fisher Networks used multiple layer of Fisher Vector encoding for image representation and classification. However, video based action recognition is different from image classification. A large portion of video is irrelevant to action, and the extracted features (such as iDT) mainly concentrate on foreground. The irregular distribution and spasticity of action related features makes it difficult to directly apply Deep Fisher Networks for action recognition. In our SFV, the sampling strategy and dimensionality reduction method are different from those of Deep Fisher Networks.

2 Fisher Vectors for Action Recognition

Fisher Vector (FV) coding method, derived from Fisher kernel, was originally proposed for large scale image categorization [24]. FV encoding assumes the generation process of local descriptors $\mathbf{X}$ can be modeled by a probability density function $p(\cdot;\theta)$ with parameters θ. The gradient of the log-likelihood w.r.t a parameter can describe how that parameter contributes to the generation process of $\mathbf{X}$ [6]. Then the video can be described by [6]:

$$G_\theta^{\mathbf{X}} = \frac{1}{N}\nabla_\theta \log p(\mathbf{X};\theta). \tag{1}$$

The probability density function is usually modeled by Gaussian Mixture Model (GMM), and $\theta = \{\pi_1, \mu_1, \sigma_1, \cdots, \pi_K, \mu_K, \sigma_K\}$ are the model parameters denoting the mixture weights, means, and diagonal covariances of GMM. K and N are the mixture number and the number of local features, respectively. X denotes spatial-temporal local features (e.g., HOG and HOF) in action videos. Perronnin *et al.* [24] proposed an improved fisher vector as follows,

$$\mathcal{G}_{\mu,k}^{\mathbf{X}} = \frac{1}{N\sqrt{\pi_k}}\sum_{n=1}^{N}\gamma_n(k)\left(\frac{\mathbf{x}_n - \mu_k}{\sigma_k}\right), \tag{2}$$

$$\mathcal{G}_{\sigma,k}^{\mathbf{X}} = \frac{1}{N\sqrt{2\pi_k}}\sum_{n=1}^{N}\gamma_n(k)\left[\frac{(\mathbf{x}_n - \mu_k)^2}{\sigma_k^2} - 1\right], \tag{3}$$

where $\gamma_n(k)$ is the weight of local feature $\mathbf{x}_n$ for the i-th Gaussian:

$$\gamma_n(k) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n;\mu_k,\sigma_k)}{\sum_{i=1}^{K}\pi_i \mathcal{N}(\mathbf{x}_n;\mu_i,\sigma_i)}, \tag{4}$$

where $\mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k)$ is d-dimensional Gaussian distribution. The final fisher vector is the concatenation of all $\mathcal{G}^{\mathbf{X}}_{\mu,k}$ and $\mathcal{G}^{\mathbf{X}}_{\sigma,k}$ which is a $2Kd$-dimensional super vector.

Fisher vector encoding with dense features yields the best performance on both image classification [4] and video-based action recognition [32]. Compared with other coding methods such as vector quantization and sparse coding, FV encoding can easily obtain high-dimensional feature codes with small codebook size, which is very important for performance improvement when using linear classifiers. We apply power normalization followed by ℓ_2 normalization to each FV block $\mathcal{G}^{\mathbf{X}}_{\mu,k}$ and $\mathcal{G}^{\mathbf{X}}_{\sigma,k}$ before normalizing them jointly, which demonstrates good performance in previous works [27].

3 Stacked Fisher Vectors

The traditional FV effectively encodes the local features of action video in a high-dimensional space, and aggregates the codes into a super vector by sum pooling over the entire video. This representation describes the video from the local feature space (approximated by GMM), which can not directly depict more global and complex structures. Deep structures (e.g., DNN [14]) are able to capture complex structures by local spatial pooling and refining the representation from one layer to the next. In this section, we present a “deep” structure by stacking two FV encoding layers, which we call *Stacked Fisher Vectors*.

The motivation of SFV is to describe the entire video with higher level representation extracted from large cuboids, which contains rich semantic information. One may argue that increasing the size of spatial-temporal patches for feature extraction may address this motivation. But, unfortunately, extracting low-level features like HOG and HOF to depict large subvolumes is not robust due to huge pose and temporal variations in action videos [27], and it has been demonstrated that very large patches is inferior to small ones (e.g., 32×32) [31]. The pipeline of SFV is shown in Figure 1. In this paper, we consider SFV with two layers. One can generalize it to more layers without difficulty. The detailed description of each layer is as follows.

3.1 The First-layer FV

Given an video $\mathbf{V}$ with size $W \times H \times L$, we first extract improved dense trajectories [32] described by concatenated HOG, HOF, and MBH descriptors. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N] \in \mathbb{R}^{d \times N}$ be the trajectory features in the video. To meet the assumption of diagonal covariances for GMM, all the features are decorrelated using PCA+Whitening before feeding into the Fisher encoder, which shows good performance in previous works [32]. Then we perform FV on each trajectory feature ($N = 1$ in Equation (1)) using a pre-learned GMM with size of K_1 in training set. We call these sparse high-dimensional vectors $\mathbf{X}' = [\mathbf{x}'_1, \mathbf{x}'_2, \cdots, \mathbf{x}'_N] \in \mathbb{R}^{2K_1 d \times N}$ as *tiny FVs*.

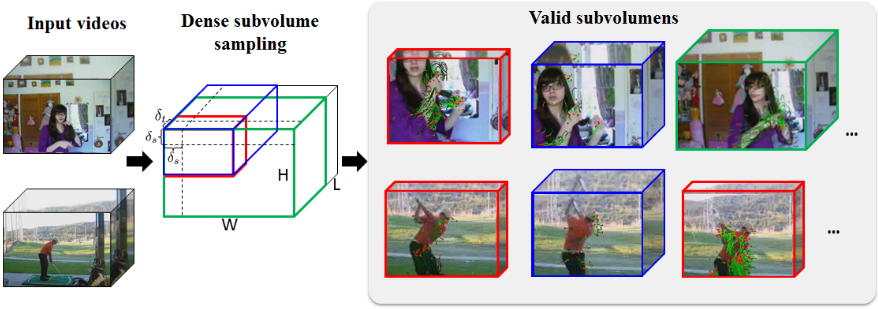


Fig. 2. Dense sampling strategy for subvolumes and some representive subvolumes from "brush hair" and "golf" action videos.

Once these tiny FVs are obtained, we aggregate them within multi-scale subvolumes scanned densely over spatial-temporal domain with strides of δ_s and δ_t. The subvolumes range from small cuboids to larger ones, allowing for two scales in width (i.e. $W/2$ and W), two scales in height (i.e. $H/2$ and H), and three scales in time (i.e. $L/3$, $2L/3$, and L), where the largest scale stretches over the entire video. To avoid meaningless statistics in near motion empty subvolumes, we check the number of trajectories within subvolumes and only perform aggregating over those subvolumes where the number of trajectories is more than a given threshold T. Figure 2 summarizes the sampling process and shows some examples of valid subvolumes. We observe that most of the valid subvolumes can deliver sufficient characteristics to discriminate action categories. We call these locally aggregated FVs $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_M] \in \mathbb{R}^{2K_1 d\times M}$ as *local FVs*, where M is the number of valid subvolumes in the video. It is worth noting that M can be varied in different videos. The local FVs are sequently normalized by power+ℓ_2 normalization per component before performing ℓ_2-normalization jointly.

3.2 The Second-layer FV

The FVs from the 1st FV layer are too high-dimensional to be directly used as the inputs of next FV layer. Here we adopt a max-margin dimensionality reduction algorithm to compress the local FVs, which make the dimensions of compressed local FVs comparable to those local features at the first layer. The details of max-margin dimensionality reduction algorithm will be described in Section 4.

The compressed local FVs are sequently decorrelated by PCA+Whitening, and then serve as the inputs of the 2nd FV layer. After learning a GMM with size of K_2, we perform another FV layer with these pre-processed local FVs and aggregate them over the entire video. The output vector is sequently normalized using the same scheme as that in the 1st layer, and serves as the final video representation of SFV.

4 Max-Margin Dimensionality Reduction

This section presents the max-margin dimensionality reduction algorithm used to compress the local FVs of subvolumes.

As explained in Section 3.2, we need to learn a projection matrix $\mathbf{U} \in \mathbb{R}^{p\times 2Kd}, p \ll 2Kd$ to significantly reduce the dimension of local FV. Note that only the whole video is assigned action label. Taking into account the fact that there are too many local FVs to us in the leaning process, we learn U from a subset of local FVs. Specially, to make all the labels of local FVs in this subset available, we sample those local FVs from entire videos and large subvolumes with size of $W \times H \times 2L/3$ which inherit the labels of their corresponding videos.

Suppose the selected local FVs and their labels are $\{\phi_i, y_i\}_{i=1,\cdots,N_l}$, where N_l denotes the number of local FVs. We aim to find the projection $\mathbf{U}$ where $\{U\phi_i\}_{i=1,\cdots,N_l}$ are as linearly separable as possible. In this paper, we perform multi-class classification with a *one-vs-all* approach, and impose there is a margin of at least one between positive and negative local FVs. This results in the following constraints,

$$y_i(\mathbf{w}U\phi_i + b) > 1, \quad i = 1, \cdots, N_l, \tag{5}$$

where $\mathbf{w} \in \mathbb{R}^{p\times 1}$ is the linear model of a certain category, and $y_i \in \{+1, -1\}$. Incorporating regularization to all the model parameters of C categories and the projection U with hinge-loss, we obtain the following objective function,

$$\arg\min_{U,W,b} \frac{\lambda}{2}\|U\|_F^2 + \frac{\beta}{2}\sum_{j=1}^{C}\|\mathbf{w}_j\|^2 + \sum_{i=1}^{N_l}\sum_{j=1}^{C}\max\{0, 1 - y_i(\mathbf{w}_jU\phi_i + b)\}, \tag{6}$$

where λ and β are the regularization constants. Though the learning objective is non-convex w.r.t $\mathbf{w}$ and U, it is convex w.r.t one of them when fixing the other. The optimum can be obtained by alternately solve the convex optimization problems of $\mathbf{w}$ and U. Both of the problems can be solved by sub-gradient method. Specially, we leverage standard SVM [3] to optimize linear model $\mathbf{w}$ and use sub-gradient algorithm to optimize U. We initialize U by PCA-Whitening matrix U_0 obtained from the local FVs, and perform the following update for U in the j-th model at each iteration,

$$\mathbf{U}_{t+1}^j = \begin{cases} -\gamma\lambda\mathbf{U}_t^j, \text{if } y_i(\mathbf{w}_jU_t\phi_i + b) > 1, \forall\ i \in \{1, \cdots, N_l\} \\ -\gamma(\lambda\mathbf{U}_t^j + \sum_i -y_i\mathbf{w}_j\phi_i), \text{ otherwise,} \end{cases} \tag{7}$$

where $\gamma > 0$ is a given learning rate, and the final updated projection matrix is $\mathbf{U}_{t+1} = \mathbf{U}_t + \sum_{j=1}^{C}\mathbf{U}_{t+1}^j$ at the t-th iteration. Once both optimization objectives have converged, the model $\mathbf{w}$ is discarded, and only $\mathbf{U}$ is saved.

5 Experiments

In this section, we evaluate the performance of the proposed SFV and traditional FV for action recognition on three popular datasets, and compare it with several

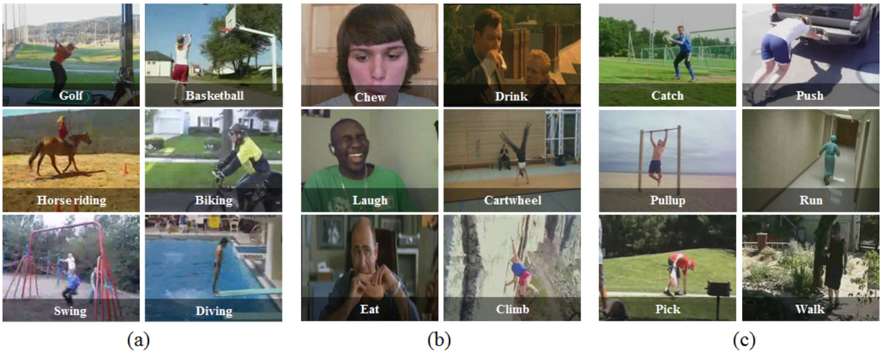


Fig. 3. From left to right, example frames from (a)YouTube, (b)HMDB51, and (c) J-HMDB.

state-of-the-art methods. Moreover, we also provide evaluations on the mixture number of GMM used in the 2nd FV layer, and on the parameters of dense sampling (i.e. δ_s, δ_t, etc.).

5.1 Datasets

We conduct experiments on three action datasets, namely Youtube [20], HMDB51 [15], and J-HMDB [10]. Some example frames are illustrated in Figure 3. We summarize them and the experimental protocols as follows.

The **Youtube** dataset [20] is collected from YouTube videos. It contains 11 action categories: basketball *shooting*, volleyball *spiking*, trampoline *jumping*, soccer *juggling*, horse back *riding*, *cycling*, *diving*, *swinging*, *golf-swinging*, *tennis-swinging*, and *walking* (with a dog). A total of 1,168 video clips are available. Following [20], we use Leave-One-Group-Out cross-validation and report the average accuracy over all classes.

The **HMDB51** dataset [15] is a large action video database with 51 categories. Totally, there are 6,766 manually annotated clips which are extracted from a variety of sources ranging from digitized movies to YouTube. It contains facial actions, general body movements and human interactions. It is a very challenging benchmark due to its high intra-class variation and low video quality. We follow the experimental settings in [15] where three train/test splits are available, and report the mean average accuracy over three splits.

The **J-HMDB** dataset [10] is a subset of HMDB51 with 21 action categories, which is annotated in details. This dataset excludes categories from HMDB51 that contain facial expressions like smiling, interactions with others such as shaking hands, and focuses on single body action. From Figure 3(c), we observe that most of the videos contain the actor in a relative small region. This ensures that sampled subvolumes can cover most of action region. The person in each frame is annotated with his/her 2D joint positions, scale, viewpoint, segmentation, puppet mask and puppet flow, which are used to evaluate the mid-level (e.g.,

Table 1. Performance of traditional FV, the proposed SFV, and their combination.

Method (Dim.)	Youtube (%)	HMDB51 (%)	J-HMDB (%)
Traditional FV (102,400)	90.69	57.29	62.83
Stacked FV (102,400)	88.68	56.21	59.27
Combination (204,800)	93.38	66.79	67.77

bounding box) and high-level features (e.g., pose feature and joints). We follow the experimental settings in [10], and report the mean average accuracy.

5.2 Experimental Setup

In all the following experiments, we densely extract improved trajectories using the code from Wang [32]. Each trajectory is described by concatenating HOG, HOF, and MBH descriptors, which is a 396-dimensional vector. We reduce the dimensionality of these descriptors to 200 by performing PCA and Whitening. For traditional FV pipeline and the first layer of SFV, we randomly sample 1,000,000 features and learn the GMM with 256 components via the EM algorithm [2], which has been shown to empirically give good results for a wide range of datasets [32]. The default values of δ_s, δ_t, and T are 10, 5, and 100, respectively. These parameters are closely related to the number of valid subvolumes, which are evaluated in Section 5.4.

We reduce the dimensionality of local FV to 400 by default. The discriminative projection matrix is initialized by PCA-Whitening matrix and learned in the training set for each dataset. λ and γ are fixed as 0.1 and 0.01, respectively. We stop the iteration once the training accuracy keeps unchanged. For the second layer of SFV, we decorrelate those compressed local FV by PCA and Whitening and further reduce the dimensionality from 400 to 200. And then we learn GMM with 256 components from a randomly sampled subset of 100,000 decorrelated local FVs. In our experiments, we choose linear SVM as our classifier with the implementation of LIBSVM [3]. For multi-class classification, we use the *one-vs-rest* approach and select the class with the highest score.

5.3 Experimental Results

We evaluate the recognition performance by default parameters in this experiment. Table 1 shows the results of traditional FV, SFV, and their combination. The FV and SFV are combined in representation level since this strategy exhibits high performance [23]. Combining the FV and SFV can double the dimension of video representation. As for higher dimension of traditional FV, please refer to our recent study in [23].

On all the datasets we used, the proposed SFV achieves comparable performance w.r.t traditional FV. This may be explained by the fact that the number of local FVs for the 2nd layer of SFV is about one-tenth of that of traditional FV. However, somewhat surprisingly, the proposed SFV provides significant

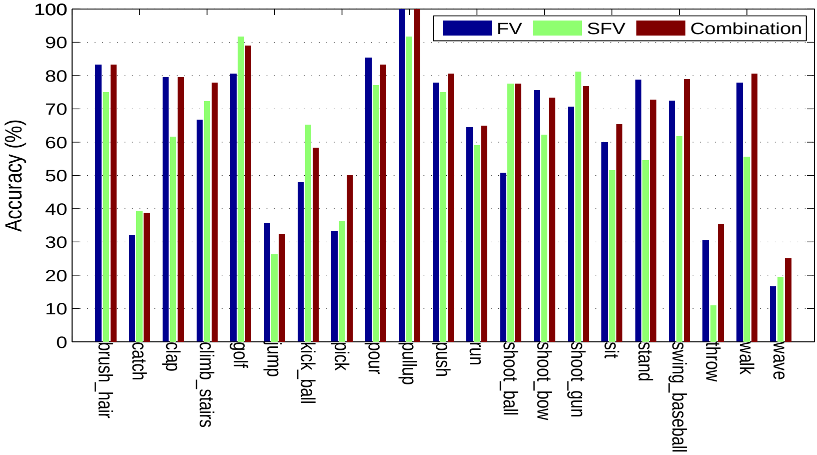
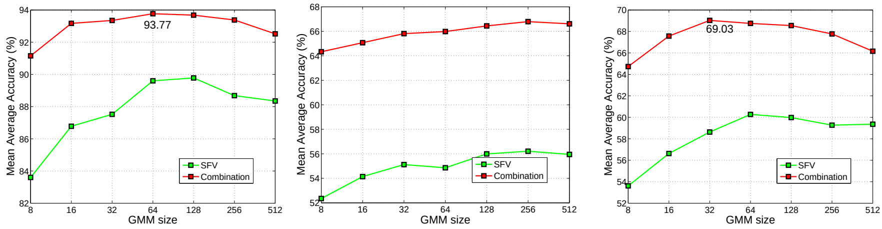


Fig. 4. The results of all the action categories from the J-HMDB dataset.



(a) Youtube (b) HMDB51 (c) J-HMDB

Fig. 5. Performance of SFV and FV+SFV with varying GMM size on the Youtube, HMDB51, and J-HMDB datasets.

complementary information to traditional FV. When combining SFV to FV, we improve the results by 2.69% on Youtube, 9.5% on HMDB51, and 4.94% on J-HMDB. To further investigate the effects of SFV on traditional FV, we illustrate the individual recognition results of all the action classes of J-HMDB dataset in Figure 4. From Figure 4, we observe that the proposed SFV is effective for the actions with less variations like golf, kick ball, shoot_ball, and shoot_gun. This can be interpreted by the properties of large volumes (described by local FVs): *global* and *discriminative* [35]. However, the global nature makes them sensitive to intra-class variation and deformation. Therefore, the performance of our SFV representation is not high for those actions with large variations.

Considering that there are less local FVs or subvolumes than local features, we also evaluate the GMM size for the 2nd layer of SFV. Figure 5 shows the results of SFV and FV+SFV with different GMM sizes. It is worth noting that the GMM sizes of both the traditional FV and the 1st layer of SFV are fixed as 256, and only that of the 2nd layer of SFV is changed. For all the datasets,

Table 2. Evaluation of multi-scale sampling strategy for dense subvolumes on Youtube, and J-HMDB.

	Youtube		J-HMDB	
sizes	volumes/video	accuracy	volumes/video	accuracy
$0.5W \times 0.5H \times \frac{L}{3}$	~1,500	83.60	~600	53.82
$0.5W \times \{0.5H, H\} \times \frac{L}{3}$	~3,200	86.35	~1,100	56.22
$\{0.5W, W\} \times 0.5H \times \frac{L}{3}$	~3,200	86.52	~1,000	56.29
$\{0.5W, W\} \times \{0.5H, H\} \times \frac{L}{3}$	~4,600	86.78	~1,600	57.96
Default	~6,000	88.68	~2,500	60.27

increasing the GMM size K improves the performance in the beginning. However, the recognition performance decreases when GMM sizes are larger than 128 and 64 on Youtube and J-HMDB, respectively. For the combination performance, the best results are observed with GMM sizes 64, 256, and 32 on Youtube, HMDB51, and J-HMDB datasets, respectively.

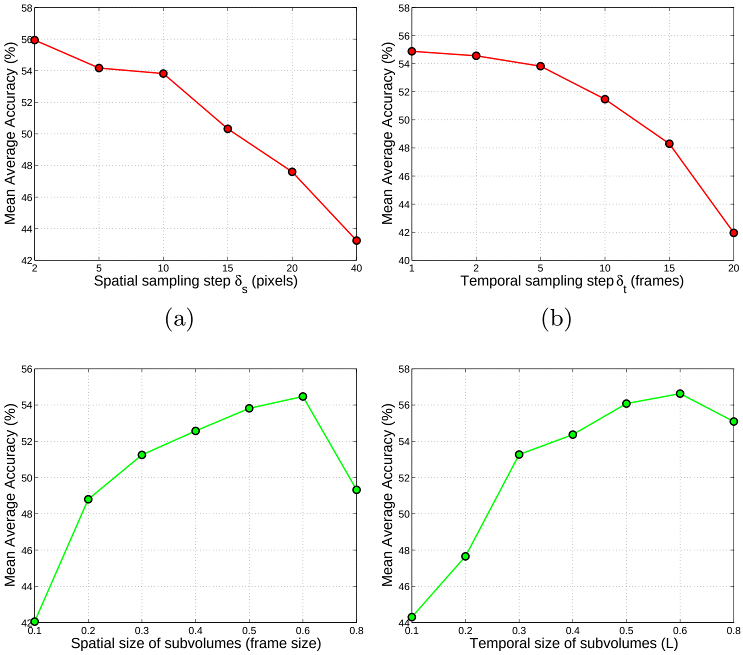
5.4 Evaluation of Sampling Parameters

In this section, we evaluate the impact of the sampling parameters for subvolumes on the performance. We report results for Youtube and J-HMDB datasets. Specially, we study the impact of multi-scale, spatial and temporal sampling steps, spatial and temporal sizes of subvolumes. In these experiments, unless otherwise stated, we carry out the evaluation for one parameter at a time, and fix the other ones to the default values, i.e., 12 scales for subvolumes, spatial sampling step $\delta_s = 10$, temporal sampling step $\delta_t = 5$.

Multi-scale vs. single scale. Results for multi-scale sampling are shown in Table 2. Considering various multi-scale schemes can can lead to different numbers of subvolumes, we also show the approximate number of valid subvolumes per video. From Table 2, it is clear that using multi-scale subvolumes is beneficial compared to a single scale on both datasets. The results from single scale $0.5W \times 0.5H \times \frac{L}{3}$ are inferior to the default settings by 5.08% and 6.45% on Youtube and J-HMDB, respectively. The main reason is that there is not enough subvolumes to cover the entire video.

Sampling step. We evaluate the spatial and temporal sampling steps on J-HMDB dataset with single scale $0.5W \times 0.5H \times \frac{L}{3}$. With respect to the spatial sampling step δ_s, Figure 6(a) presents the results for $\delta_s = 2$ pixels to $\delta_s = 40$ pixels. The performance increases with a higher sampling density. Figure 6(b) shows the results of different temporal sampling steps. For both spatial and temporal sampling steps, lower sampling density obtains less number of valid subvolumes, which is harmful to the recognition performance.

Volume size. We also evaluate the spatial and temporal sizes of subvolumes with single scale on J-HMDB dataset. Figure 6(c) and Figure 6(d) show the results of various spatial and temporal sizes, respectively. The worst results are those from the smallest spatial and temporal sizes. Small sizes of subvolumes



(a) (b)

(c) (d)

Fig. 6. Evaluation of the sampling parameters and subvolume sizes on the J-HMDB dataset with single scale. (a) spatial sampling step, (b) temporal sampling step, (c) the spatial size of subvolumes related to frame size, (d) the temporal length of subvolumes related to video length L.

suffer from two issues. On one hand, there are a small number of local features within subvolumes which results in very few valid local FVs. On the other hand, pooling tiny FVs (from local features) in a small 3D patch may lead to less meaningful statistics [27]. Enlarging the size of subvolumes boosts the performance up to 60 percent of frame size and video length L. However, subvolume sizes larger than 60 percent of frame size and L decrease the performance, as there is a limited sampling space for subvolumes.

5.5 Comparison with State of the Art

In this section, we compare our results to the state of the art on each dataset. Table 3 displays our best results and several recently published results in the literature.

These methods (e.g., motionlets [35], mid-level parts [27], and actons [37]) that utilize the responses of discriminative action parts combined with low-level features perform inferior to our method (FV+SFV) with a certain margin on all three datasets. This demonstrates the effectiveness of the proposed stacked

Table 3. Comparison of our approach (FV+SFV) with the state-of-the-art results on Youtube, HMDB51, and J-HMDB. *Our own implementation. $^+$ It leverages human annotation on actors with person mask or pose, but ours don't require.

Youtube		HMDB51		J-HMDB	
Liu *et al.* [20]	71.2	Actons [37]	54.0		
Ikizler *et al.* [5]	75.21	Motionlets [35]	42.1	DT+BoVW [10]	56.6
Mid-level parts [27]	84.5	Mid-level parts [27]	37.2	iDT+FV*	62.8
DT+BoVW [31]	85.4	DT+BoVW [31]	46.6	Masked DT+BoVW [10]$^+$	69.0
iDT+FV *	90.69	iDT+FV [32]	57.2	Pose+BoVW [10] $^+$	76.0
Our method	**93.77**	Our method	**66.79**	Our method	69.03

Fisher vectors. For a fair comparison, we use the results of "iDT+FV" as baselines. From Table 3, our approach outperforms the best previous results by 3.08% on Youtube, and 9.59% on HMDB51. As for J-HMDB, the method [10] using annotated pose features with BoVW model provides the highest performance. Without the high-level human annotated pose information, our method significantly improves the baseline by 6.2%.

6 Conclusions

Mid-level action parts prove to be effective for action recognition [35, 7, 37, 27]. However, previous methods only leveraged the responses of discriminative part-s for subvolumes which have limited representative ability. In this paper, we propose stacked Fisher vectors, which is a hierarchical structure based on the off-the-shelf Fisher coding. It describes the densely sampled subvolumes by high-dimensional super vectors. The high-dimensional nature allows it to preserve richer information for each subvolume. After discriminative dimensionality reduction by a max-margin approach, we utilize another Fisher coding layer to construct a global representation for videos. Extensive experiments on three widely-used datasets indicate the effectiveness of our SFV representation. Combining our SFV and standard Fisher vectors, we achieve superior performance on the Youtube and HMDB51 datasets than state-of-the-art methods .

Acknowledgements: This work is partly supported by Natural Science Foundation of China (91320101, 61036008, 60972111), Shenzhen Basic Research Program (JC201005270350A, JCYJ20120903092050890, JCYJ20120617114614438), 100 Talents Program of CAS, Guangdong Innovative Research Team Program (201001D0104648280), and the construct program of the key discipline in Hunan province.

References

1. Aggarwal, J., Ryoo, M.S.: Human activity analysis: A review. ACM Computing Surveys 43(3), 16 (2011)
2. Bishop, C.M., Nasrabadi, N.M.: Pattern recognition and machine learning, vol. 1 (2006)
3. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) 2(3), 27 (2011)
4. Chatfield, K., Lempitsky, V., Vedaldi, A., Zisserman, A.: The devil is in the details: an evaluation of recent feature encoding methods. In: BMVC (2011)
5. Ikizler-Cinbis, N., Sclaroff, S.: Object, scene and actions: combining multiple features for human action recognition. In: ECCV. pp. 494–507 (2010)
6. Jaakkola, T., Haussler, D., et al.: Exploiting generative models in discriminative classifiers. NIPS pp. 487–493 (1999)
7. Jain, A., Gupta, A., Rodriguez, M., Davis, L.S.: Representing videos using mid-level discriminative patches. In: CVPR. pp. 2571–2578 (2013)
8. Jain, M., Jégou, H., Bouthemy, P.: Better exploiting motion for better action recognition. In: CVPR. pp. 2555–2562 (2013)
9. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: CVPR. pp. 3304–3311 (2010)
10. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M.J., et al.: Towards understanding action recognition. In: ICCV (2013)
11. Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. TPAMI pp. 221–231 (2013)
12. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR (2014)
13. Klaser, A., Marszałek, M., Schmid, C., et al.: A spatio-temporal descriptor based on 3d-gradients. In: BMVC (2008)
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. vol. 1, p. 4 (2012)
15. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: a large video database for human motion recognition. In: ICCV. pp. 2556–2563 (2011)
16. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR. pp. 1–8 (2008)
17. Laptev, I.: On space-time interest points. IJCV 64(2), 107–123 (2005)
18. Le, Q.V., et al.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: CVPR. pp. 3361–3368 (2011)
19. Liu, J., Kuipers, B., Savarese, S.: Recognizing human actions by attributes. In: CVPR. pp. 3337–3344 (2011)
20. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos in the wild. In: CVPR. pp. 1996–2003 (2009)
21. Liu, L., Wang, L., Liu, X.: In defense of soft-assignment coding. In: ICCV. pp. 2486–2493 (2011)
22. Peng, X., Qiao, Y., Peng, Q., Qi, X.: Exploring motion boundary based sampling and spatial-temporal context descriptors for action recognition. In: BMVC. pp. 1–11 (2013)
23. 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)

24. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: ECCV. pp. 143–156 (2010)
25. Ren, X., Ramanan, D.: Histograms of sparse codes for object detection. In: CVPR. pp. 3246–3253 (2013)
26. Sadanand, S., Corso, J.J.: Action bank: A high-level representation of activity in video. In: CVPR. pp. 1234–1241 (2012)
27. Sapienza, M., Cuzzolin, F., Torr, P.H.: Learning discriminative space–time action parts from weakly labelled videos. IJCV pp. 1–18 (2014)
28. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep fisher networks for large-scale image classification. In: NIPS. pp. 163–171 (2013)
29. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: ICCV. pp. 1470–1477 (2003)
30. Wang, H., Klaser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: CVPR. pp. 3169–3176 (2011)
31. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. IJCV pp. 1–20 (2013)
32. Wang, H., Schmid, C., et al.: Action recognition with improved trajectories. In: ICCV (2013)
33. Wang, H., Ullah, M.M., Klaser, A., Laptev, I., Schmid, C., et al.: Evaluation of local spatio-temporal features for action recognition. In: BMVC (2009)
34. Wang, L., Qiao, Y., Tang, X.: Mining motion atoms and phrases for complex action recognition. In: ICCV. pp. 2680–2687 (2013)
35. Wang, L., Qiao, Y., Tang, X.: Motionlets: Mid-level 3d parts for human motion recognition. In: CVPR. pp. 2674–2681 (2013)
36. Wang, X., Wang, L., Qiao, Y.: A comparative study of encoding, pooling and normalization methods for action recognition. In: ACCV, pp. 572–585 (2013)
37. Zhu, J., Wang, B., Yang, X., Zhang, W., Tu, Z.: Action recognition with actons. In: ICCV (2013)