Feature Sequence Representation Via Slow Feature Analysis For Action Classification

Takumi Kobayashi
takumi.kobayashi@aist.go.jp

National Institute of Advanced Industrial Science and Technology
Tsukuba, Japan

Abstract

The recent advances in extracting motion descriptors, such as BoW and CNN features, enable us to effectively convert a video into a sequence of frame-based feature vectors. For improving the action classification performance, in this paper, we propose an efficient method to represent the feature sequence by exploiting the temporal patterns via slow feature analysis (SFA). The ordinary SFA suffers from small sample size (*SSS*) problem found in action video clips and thus we propose PCA-SFA to cope with the *SSS* problem by incorporating the information of PCA subspaces into SFA. The proposed method leverages the PCA-SFA projection vector to describe the sequence of even fewer frames by a fixed-dimensional video descriptor, capturing the essential temporal dynamics which is a slowly varying pattern embedded in the quickly varying input signals. The computational cost to produce the video descriptor is negligible compared to the feature extraction process such as BoW and CNN since the PCA-SFA is computed in a computationally efficient manner. In the experiments on action classification using various datasets, the proposed method exhibits favorable performance being competitive to the other methods.

1 Introduction

There is an increasing amount of multimedia data containing not only images but also videos through security cameras in the real world and web sites (such as YouTube) on the Internet. Thereby, it creates an urgent demand for automatic action recognition in computer vision communities. The action recognition has been tackled over the last two decades [3, 15, 33, 48]. The difficulty in the action recognition is primarily in extracting effective motion features, though the classifier also gains considerable research interest as in image classification. An input video is formulated in a spatio-temporal volume while the images are defined in a two-dimensional space. Such higher dimensionality of the input data makes it harder to design motion features in comparison with image feature extraction.

In recent years, along with the advances of image classification, the motion descriptors which extract motion characteristics are developed in the framework of bag-of-features including spatio-temporal interest points and/or dense trajectories, exhibiting successful performance in realistic videos [26, 44, 45, 53]. On the other hand, deep convolutional neural network (CNN) methods have been applied to various image recognition tasks with great success, and it is now being extended to motion recognition fields together with the large-scale video dataset [18, 19, 42]. The CNN methods can establish spatio-temporal features to effectively describe the motion patterns via end-to-end learning [42].

Although those motion descriptors are good at extracting rather temporally *local* motions, the other difficulty also exists in the action recognition, that is, how to describe temporally *global* action performed over a sequence. The (human) action/behavior is composed of several primitive motions which may be well characterized by the above-mentioned descriptors, and it is necessary to summarize or aggregate the local motion descriptors extracted such as at each frame. To be more concrete, through extracting local motion descriptors, a video is converted into a sequence of motion feature vectors (frames) and then we aim to extract global motion patterns from the feature sequence for action recognition.

As in a spatial pyramid of image classification [27], it is possible to assign temporal grids over the sequence on the assumption that the sequences are well aligned [26]. The transitions of feature frames in the sequence can be probabilistically modeled such as by HMM [41] and CRF [37], or LSTM [8] in recent years, to extract a higher-level motion patterns, though it requires substantial computation cost and thus unsuitable for large-scale classification. On the other hand, the feature sequence can be regarded as some sort of a feature *set*, and the subspace-based methods can be applied to such feature sets [20, 29, 30]. The subspace statistically well approximates the set by extracting the variance information from the feature frames. However, since the subspace-based methods assume the features to be i.i.d. samples, they completely lose the temporal patterns along the feature sequence.

In this paper, we propose an efficient method to represent the sequence of motion feature vectors for action classification. In the proposed method, we extract the characteristics of temporal dynamics on the feature sequence via slow feature analysis (SFA) [49, 51]. The SFA suffers from small sample size (*SSS*) problem usually found in action clips and thus we propose PCA-SFA to effectively cope with the *SSS* problem in the framework of SFA. As a result, the proposed method provides a fixed-dimensional video descriptor which well captures the temporal characteristics even in the shorter video clips and is favorably fed into a linear classifier for action recognition. The method is closely related to [11] which also extracts the sequence direction as a global motion pattern by a video evolution model based on a linear regression into time index. In contrast, our method represents the video sequence by leveraging the PCA-SFA to characterize temporal patterns more flexibly than the linear regression. It should be noted that the proposed method produces the video descriptor in a computationally efficient way and thus the computational overhead is negligible compared to the feature extraction process such as by BoW and CNN.

2 Slow feature analysis for small sample size problem

In this study, we focus on slow feature analysis (SFA) [49, 51] to extract temporal patterns from a feature sequence. The SFA is frequently applied to analyze the (temporal) sequence data in an unsupervised learning framework which is inspired by the visual system based on the slowness principle[1] [16]. The connection between the SFA and visual neurons is found in [2, 12] and it is also applied to recognize objects [13] and actions [40, 52]. As to the mathematical formulation, the SFA is extended to graph-based SFA [9, 10] for dealing with more

[1] While the primary sensory signals, such as in the retina, change on a faster timescale, our environment changes slowly. Thus, the internal representation of the environment should vary on a *slow* timescale.

general graph structure beyond a sequence. This section first describes the SFA and then proposes PCA-SFA to cope with small sample size (*SSS*) problem for action classification.

2.1 Slow feature analysis (SFA)

For analyzing videos in action recognition, an input video is generally converted to a set of feature vectors sequentially extracted along the time, denoted by $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T]$ of T feature vectors $\boldsymbol{x}_t \in \mathbb{R}^d$ with the time index t. The SFA finds a projection vector (linear mapping function) $\boldsymbol{w} \in \mathbb{R}^d$ for features $\boldsymbol{x}_t$ such that an output value $y_t = \boldsymbol{w}^\top \boldsymbol{x}_t + b$ *slowly* changes along the sequence under the following constraints: zero mean $\frac{1}{T}\sum_{t=1}^T y_t = 0$, unit variance $\frac{1}{T}\sum_{t=1}^T y_t^2 = 1$ and decorrelation $\frac{1}{T}\sum_{t=1}^T y_t^{(i)} y_t^{(j)} = 0\ \forall i < j$ where $y_t^{(i)} = \boldsymbol{w}_i^\top \boldsymbol{x}_t + b_i$ is the i-th output value by the i-th projection $\boldsymbol{w}_i$. This formulation analytically leads to the generalized eigenvalue problem [49, 51];

$$\left(\sum_t \dot{\boldsymbol{x}}_t \dot{\boldsymbol{x}}_t^\top\right) \boldsymbol{w} = \lambda \left(\sum_t (\boldsymbol{x}_t - \boldsymbol{\mu})(\boldsymbol{x}_t - \boldsymbol{\mu})^\top\right) \boldsymbol{w}, \tag{1}$$

where $\boldsymbol{\mu}$ indicates the mean $\boldsymbol{\mu} = \frac{1}{T}\sum_t \boldsymbol{x}_t$ and $\dot{\boldsymbol{x}}_t$ is the differential vector w.r.t t, practically computed as $\dot{\boldsymbol{x}}_t = \boldsymbol{x}_{t+1} - \boldsymbol{x}_t$. The eigenvectors $\boldsymbol{w}$ of the smaller eigenvalues λ in (1) are employed as the projection vectors of SFA that extract the *slow* features.

2.2 PCA-SFA

In some practical situations, especially on action classification of our interest, the number of samples T is significantly smaller than the feature dimensionality d of $\boldsymbol{x}$; a video clip containing action to be classified is composed of several hundreds of frames while the higher dimensional (>1000) features are extracted to improve recognition performance. In such a case, the eigenvalue problem (1) is ill-posed, producing a trivial solution no matter how the features are extracted [50]; it produces slow features of harmonic oscillation in disregard of feature distribution (see the supplemental material). To cope with the small sample size (*SSS*) problem, a hierarchical approach can be applied to gradually encode the subset of input features [51]. It, however, contains the difficulty in how to select the subsets and it is ineffective from the viewpoint that we leverage the projection vector $\boldsymbol{w}$ to represent the sequence as described in Sec. 3.1.

Thus, to cope with the *SSS* problem, we incorporate the information of PCA subspaces into the SFA (1). The small-sized samples $\{\boldsymbol{x}_t\}_{t=1}^T$ of $T < d$ can be projected into an arbitrary form of sequence due to over-fitting, and thus we regularize the projection vector $\boldsymbol{w}$ so as to follow the PCA subspace of the sequence for effectively extracting temporal information via SFA. The PCA subspace is obtained through maximizing the variance in the following way;

$$\max_{\boldsymbol{w}} \frac{\boldsymbol{w}^\top \{\sum_t (\boldsymbol{x}_t - \boldsymbol{\mu})(\boldsymbol{x}_t - \boldsymbol{\mu})^\top\} \boldsymbol{w}}{\boldsymbol{w}^\top \boldsymbol{w}} \Leftrightarrow \min_{\boldsymbol{w}} \frac{\boldsymbol{w}^\top \boldsymbol{w}}{\boldsymbol{w}^\top \{\sum_t (\boldsymbol{x}_t - \boldsymbol{\mu})(\boldsymbol{x}_t - \boldsymbol{\mu})^\top\} \boldsymbol{w}}. \tag{2}$$

On the other hand, the SFA (1) is equivalent to the optimization problem of

$$\min_{\boldsymbol{w}} \frac{\boldsymbol{w}^\top (\sum_t \dot{\boldsymbol{x}}_t \dot{\boldsymbol{x}}_t^\top) \boldsymbol{w}}{\boldsymbol{w}^\top \{\sum_t (\boldsymbol{x}_t - \boldsymbol{\mu})(\boldsymbol{x}_t - \boldsymbol{\mu})^\top\} \boldsymbol{w}}. \tag{3}$$

The above two minimization problem sharing the denominator can be merged into

$$\min_{\boldsymbol{w}} \frac{\boldsymbol{w}^\top(\eta \boldsymbol{I} + \sum_t \dot{\boldsymbol{x}}_t \dot{\boldsymbol{x}}_t^\top)\boldsymbol{w}}{\boldsymbol{w}^\top\{\sum_t(\boldsymbol{x}_t - \boldsymbol{\mu})(\boldsymbol{x}_t - \boldsymbol{\mu})^\top\}\boldsymbol{w}} \Rightarrow \left(\eta \boldsymbol{I} + \sum_t \dot{\boldsymbol{x}}_t \dot{\boldsymbol{x}}_t^\top\right)\boldsymbol{w} = \lambda\left(\sum_t(\boldsymbol{x}_t - \boldsymbol{\mu})(\boldsymbol{x}_t - \boldsymbol{\mu})^\top\right)\boldsymbol{w}, \tag{4}$$

where η is a balancing parameter between PCA (2) and SFA (3). The proposed method (4), called *PCA-SFA*, produces the projection vector to extract slow features while following the PCA subspace that captures the essential statistical structure in the sequence. It is obvious that (4) is a unified method of PCA and SFA, rendering the SFA by $\eta = 0$ and the PCA by $\eta \to \infty$. In this study, the balancing (regularization) parameter is determined as $\eta = \frac{1}{T}\sum_t \|\dot{\boldsymbol{x}}_t\|_2^2$ which becomes higher for the smaller-sized samples and lower for the larger-sized ones, compared to the SFA term $\sum_t \dot{\boldsymbol{x}}_t \dot{\boldsymbol{x}}_t^T$; the SFA can stably exploit temporal structure of the larger-sized samples without resorting to PCA. It is noteworthy that the PCA-SFA (4) satisfies the above-mentioned three constraints, zero mean, unit variance and decorrelation, as in SFA.

The PCA-SFA (4) is efficiently computed in the case of *SSS* as follows. First, we apply singular value decomposition (SVD) to the centered feature sequence; $\boldsymbol{X} - \boldsymbol{\mu}\boldsymbol{1}^\top = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \in \mathbb{R}^{d\times T}(T \ll d)$ where $\boldsymbol{U} \in \mathbb{R}^{d\times r}, \boldsymbol{\Sigma} \in \mathbb{R}^{r\times r}, \boldsymbol{V} \in \mathbb{R}^{T\times r}$ and $r(< T)$ indicates the rank, usually $r = T - 1$. Then, by reparameterizing the projection vector as $\boldsymbol{w} = \boldsymbol{U}\boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha}$, (4) results in

$$\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^\top\left(\eta \boldsymbol{I} + \sum_t \dot{\boldsymbol{x}}_t \dot{\boldsymbol{x}}_t^\top\right)\boldsymbol{U}\boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha}, \Leftrightarrow \left(\eta\boldsymbol{\Sigma}^{-2} + \boldsymbol{\Sigma}^{-1}\boldsymbol{U}^\top\sum_t \dot{\boldsymbol{x}}_t \dot{\boldsymbol{x}}_t^\top \boldsymbol{U}\boldsymbol{\Sigma}^{-1}\right)\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha}. \tag{5}$$

The above eigenvalue problem of $r \times r$ size and the SVD of $\boldsymbol{X} - \boldsymbol{\mu}\boldsymbol{1}^\top$ can be quite efficiently computed due to small T; on an average, the sequence of 4,096-dimensional BoW features is processed only in 78 msec by Xeon 3.4GHz PC on HOLLYWOOD2 dataset which contains videos of about 300 frames.

2.3 Discussion

As to a mathematical formulation, our method is similar to [39] in which PCA and Fisher discriminant analysis (FDA) are blended in terms both of objective function and constraints in the framework of semi-supervised learning. It should be noted that our method is formulated in an unsupervised learning framework without resorting to any class labels and reasonably merges the objective functions of PCA and SFA through simply reformulating the PCA into (2), due to which the PCA-SFA effectively retains the SFA constraints.

From a practical viewpoint, it might be possible to avoid the trivial solution in the practical SFA computation ($\eta = 0$ in (5)) by setting lower dimensional PCA subspace (of small r). It, however, is difficult to carefully tune the subspace dimension r while keeping essential temporal patterns in the subspace for each sequence. The proposed PCA-SFA takes the full rank r in (5) and thus only requires the parameter η to be set in advance; we present the pre-defined form of $\eta = \frac{1}{T}\sum_t \|\dot{\boldsymbol{x}}_t\|_2^2$, which is empirically validated in Sec. 4.

One might apply the regularization term $\eta\boldsymbol{I}$ to the right-hand-side of (1) as in regularized FDA [14], but it still provides trivial solution in SFA; the null space of $\boldsymbol{X}$ gives the smallest eigenvalue ($\lambda = 0$).

3 Feature sequence representation

After extracting temporally local (frame-based) motion features to produce a feature sequence $\boldsymbol{X}$, it is required to effectively describe the whole feature sequence by extracting global patterns, such as temporal dynamics, over the sequence for action recognition.

3.1 SFA-based representation

As described in Sec. 2, SFA can extract the essential slow features from the sequence and thereby the projection vector $\boldsymbol{w}$ that produces such slow features is considered to reflect the essential temporal information, such as temporal evolution direction, embedded in the sequence. Therefore, we directly employ the first projection vector $\boldsymbol{w}$ of the smallest eigenvalue provided by PCA-SFA (4) as a d-dimensional feature vector to represent the feature sequence. Note that the previous works [40, 52] leverage SFA to extract *slow features* y as a basic feature extractor, paying less attention to the projection vector $\boldsymbol{w}$. In contrast to those previous works, we describe a sequence by using $\boldsymbol{w}$. And, the SFA naturally characterizes the temporal dynamics in the sequence unlike the method [11] which extracts the temporally evolving direction by means of linear regression into the (equally spaced) time index t from $\boldsymbol{x}_t$ without taking into account the feature distribution; the inherent time index in the sequence $\boldsymbol{X}$ would not be equally spaced nor always increasing due to non-linearity of the features. The SFA captures a non-linear manifold structure embedded in quickly varying input signals [50]. Thus, we can say that the intrinsic temporal information can be extracted by SFA. Note that in this study we do not apply the polynomial expansion of the feature vector $\boldsymbol{x}_t$ [51] which significantly increases the feature dimensionality.

The only problem is that the directional sign of the projection vector $\boldsymbol{w}$ is arbitrarily determined in (4); that is, both $\boldsymbol{w}$ and $-\boldsymbol{w}$ give the smallest eigenvalue in (4). To consistently align the direction of $\boldsymbol{w}$, we transform $\boldsymbol{w}$ such that the slow features $y_t = \boldsymbol{w}^\top(\boldsymbol{x}_t - \boldsymbol{\mu})$ exhibit positive correlation with the time index t;

$$\hat{\boldsymbol{w}} = \begin{cases} \boldsymbol{w} & \texttt{corr}(\boldsymbol{w}^\top \boldsymbol{x}_t, t) \geq 0 \\ -\boldsymbol{w} & \texttt{corr}(\boldsymbol{w}^\top \boldsymbol{x}_t, t) < 0 \end{cases}, \tag{6}$$

where corr computes the correlation coefficient between two sequences in $t \in \{1, \cdots, T\}$.

While temporal dynamics in a sequence is extracted by the PCA-SFA, the sequence is also characterized by the position in the feature space, *i.e.*, mean $\boldsymbol{\mu} = \frac{1}{T}\sum_t \boldsymbol{x}_t$, which is eliminated in the PCA-SFA (4). The mean $\boldsymbol{\mu}$ is regarded as the 0th order representation while the first SFA projection vector $\boldsymbol{w}$ is the 1st order one. Those two types of representations are simply concatenated into the video descriptor of the fixed dimensionality through normalization;

$$\boldsymbol{v} = \left[\frac{\boldsymbol{\mu}^\top}{\|\boldsymbol{\mu}\|_2}, \frac{\hat{\boldsymbol{w}}^\top}{\|\hat{\boldsymbol{w}}\|_2}\right]^\top \in \mathbb{R}^{2d}. \tag{7}$$

3.2 Transformation of feature sequence

The method [11] showed the performance improvement by transforming an input feature sequence into smooth one by means of cumulative summation. We generalize this approach

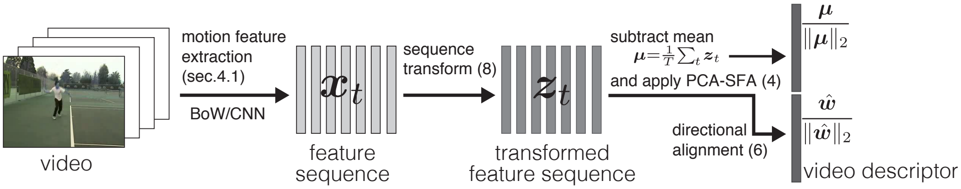
video
motion feature extraction (sec.4.1)
BoW/CNN
x_t
feature sequence
sequence transform (8)
z_t
transformed feature sequence
subtract mean $\mu = \frac{1}{T}\sum_t z_t$ and apply PCA-SFA (4)
directional alignment (6)
$\frac{\mu}{\|\mu\|_2}$
$\frac{\hat{w}}{\|\hat{w}\|_2}$
video descriptor

Figure 1: Pipeline of the proposed method.

Dataset	HOLLYWOOD2 [28]	HMDB51 [23]	UCF101 [38]
number of action classes	12	51	101
number of training videos	823	3570	9582 (avg.)
number of test videos	884	1530	3737 (avg.)
Averaged number of frames per video	286	93	186

Table 1: Action recognition datasets used in the experiments.

by introducing a *pooling* function for transforming the input feature sequence as follows.

$$z_t^{\text{fwd}} = \frac{\mathtt{l}(\mathtt{pool}(x_1,\cdots,x_t))}{\|\mathtt{l}(\mathtt{pool}(x_1,\cdots,x_t))\|_2},\quad z_t^{\text{bwd}} = \frac{\mathtt{l}(\mathtt{pool}(x_T,\cdots,x_t))}{\|\mathtt{l}(\mathtt{pool}(x_T,\cdots,x_t))\|_2},\quad t\in\{1,\cdots,T\}, \tag{8}$$

where pool is a pooling function to produce a d-dimensional vector from a set of d-dimensional feature vectors and l is a feature transformation function which is specified in Sec. 4.1. Note that the forward pooling in z_t^{fwd} deals with the frame vectors x_t from the start ($t = 1$), while the backward one in z_t^{bwd} is applied from the end ($t = T$). In the case that pool outputs the last vector of an input set, $\mathtt{pool}(x_1,\cdots,x_t) = x_t$, the above transformation results in an identical mapping, that is, $z_t^{\text{fwd}} = z_t^{\text{bwd}} = \frac{\mathtt{l}(x_t)}{\|\mathtt{l}(x_t)\|_2}$, and the transformation used in [11] corresponds to (8) of the sum-pooling $\mathtt{pool}(x_1,\cdots,x_t) = \frac{1}{t}\sum_{i=1}^{t} x_i$. The transformation (8) is also regarded as a variant of motion history image [4] by replacing a frame image with a frame feature vector x_t, for further enhancing the temporal information on the feature sequence. The final video descriptor is produced by applying (7) to those bi-directional sequences $\{z_t^{\text{fwd}}\}_{t=1}^{T}$ and $\{z_t^{\text{bwd}}\}_{t=1}^{T}$, respectively;

$$v_{seq} = [v(\{z_t^{\text{fwd}}\}_{t=1}^{T})^\top, v(\{z_t^{\text{bwd}}\}_{t=1}^{T})^\top]^\top \in \mathbb{R}^{4d}. \tag{9}$$

By separately dealing with those transformed sequences, the temporal characteristics of the forward and backward dependencies are analyzed, as in bi-directional LSTM [34], and exploited as features. As a result, the proposed method extracts detailed structure of the sequence; the pipeline to extract our video descriptor is shown in Fig. 1.

4 Experimental results

We apply the proposed method to two types of features (Sec. 4.1) on three action recognition datasets, HMDB51 [23], UCF101 [38] and HOLLYWOOD2 [28], following the evaluation protocol provided in the respective datasets. The summary of those datasets are shown in Table 1; for details of the datasets, refer to the respective papers.

4.1 Motion features

This section details two types of motion features to which the proposed method is applied: convolutional neural network (CNN) [36, 42] to extract holistic frame-based features and bag-of-word (BoW) [44, 45] based on local descriptors [5, 26, 46].

CNN features. As in [1, 7], we transfer the CNN features pretrained on the other datasets to our action recognition tasks. We employ C3D features [42] that are trained on sports-1M dataset [19] for extracting motion features from 16 RGB image frames as well as very deep CNN (VGG19) features [36] trained on ImageNet dataset [6] for extracting frame-based *image* features from 1 RGB image frame; those CNNs extract spatially holistic features. We extract at every 4 frames these CNN features which are the outputs ($\boldsymbol{x}_t \in \mathbb{R}^{4096}$) of the first fully connected layer in respective CNN models.

The max-pooling is a key step in the CNN models [22, 36, 42] together with the convolution layers. We can say that taking maximum over local region is considered to improve discriminativity as well as increase robustness. Following this mechanism, we apply *max*-pooling across rather longer temporal region in the sequence transform (8) (Sec. 3.2); $\mathtt{pool}(\boldsymbol{x}_1,\cdots,\boldsymbol{x}_t) = [\max_{i=1,\cdots,t}(x_{i1}),\cdots,\max_{i=1,\cdots,t}(x_{id})]^\top$, where x_{ij} indicates the j-th component of the i-th feature vector $\boldsymbol{x}_i$, and we apply the identity mapping $\mathtt{l}(\boldsymbol{x}) = \boldsymbol{x}$.

BoW features. We extract BoW motion features in the framework of dense trajectories [44, 45]. In that framework, the local descriptors of HoF [26] and MBH [5] as well as the trajectory-pooled CNN local descriptors[2] [46] are extracted on the trajectories densely extracted in a video sequence, and then those are coded into 4,096 visual words. The word counting histogram feature $\boldsymbol{x}_t \in \mathbb{R}^{4096}$ at time t is computed by aggregating local descriptors (words) whose trajectories end at t and thereby it results in a rather sparse feature vector. In the sequence transform (8), the Dirichlet-FK feature transformation[3] $\mathtt{l}(\boldsymbol{x}) = \log(\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_1} + \tau) - \varepsilon$ [21] is applied to transform the BoW histogram features.

The above feature extraction processes produce two types of CNN motion feature sequences and three types of BoW ones, each to which the proposed method (Fig. 1) is applied, and then the video descriptors (9) are concatenated. Finally, the linear SVM classifier [43] is applied to categorize the video descriptors into action classes.

4.2 Performance analysis

On HOLLYWOOD2 dataset, we analyze the performance of the proposed method (Fig. 1) regarding the sequence transformation (Sec. 3.2) and PCA-SFA based descriptor (Sec. 3.1) with comparison to the other sequence representations.

Sequence transformation. There are three ways to transform a feature sequence (Sec. 3.2) by the function pool; the simplest *identical mapping*, *sum pooling* and *max pooling* in two directions of forward (*fwd*) and backward (*bwd*). The performance comparison is shown in Table 2, demonstrating that pooling-based sequence transformation favorably improves performance. In BoW features, the pooling-based method significantly outperforms *identical mapping* since the BoW features are rather sparse at each frame and thus pooling (aggregation) along the time sequence is effective to enhance discriminative power. And, the combination of the bi-directional sequences (*fwd*+*bwd*) exhibits the best performance. The *max*-pooling over larger temporal region is the better way to accumulate the CNN features

[2] We utilize the neuron activations at the conv4 layer in the temporal CNN; for the detail, refer to [46].

[3] Those parameter values in the Dirichlet-FK are set to $\tau = 0.0001, \varepsilon = -8.5$ determined by the method [21].

for action recognition, as described in Sec. 4.1. On the other hand, BoW features prefer the *sum*-pooling since the Dirichlet-FK feature transform [21] is built upon the probabilistic property of the histogram features and *max*-pooling might violate such probabilistic nature. Thus *fwd+bwd* with *max*-pooling is suitable for CNN features while that with *sum*-pooling works on BoW features. We hereafter employ those types of sequence transformation for CNN and BoW features.

Regularization parameter in PCA-SFA. Then, we analyze the regularization parameter η in PCA-SFA (4) which balances SFA and PCA. Note again that the smaller η makes the method close to SFA (1) and the larger one leads to PCA. We change η based on $C \triangleq \frac{1}{T}\sum_t \|\dot{\boldsymbol{x}}_t\|_2^2$ (Sec. 2.2). Table 2b shows the performance results on various η. The performance is deteriorated by $\eta = 0$ where PCA-SFA is reduced to simple SFA (1). In contrast, by increasing η, the performance is accordingly improved and the best one is achieved around $\eta = C$. This experimental result demonstrates that the proposed PCA-SFA with the appropriate regularization parameter value $\eta = C$ favorably exploits the temporal information to represent a sequence as a feature vector, while the SFA ($\eta = 0$) suffers from the *SSS* problem found in the action clip and fails to extract effective temporal patterns. Thus, we use $\eta = C$ in the PCA-SFA representation throughout these experiments.

Comparison to the other sequence representation. The PCA-SFA based representation is compared with the other methods that represent a video sequence by a fixed-dimensional descriptor. For fair comparison, all the methods are applied to bi-directional *fwd+bwd* sequence with *sum/max* pooling on BoW/CNN features.

The method is compared to *mean* $\boldsymbol{\mu} = \frac{1}{T}\sum_t \boldsymbol{z}_t$, the *temporal grid* on the sequence [26], the *subspace*-based representation and *VideoDarwin* [11]. We apply two temporal grids along the sequence around which the frame-based feature vectors are aggregated (summarized), resulting in the $2d$-dimensional representation for the sequence. The subspace-based method extracts the two-dimensional subspace from the samples $\{\boldsymbol{z}_t\}_{t=1}^T$ in the sequence and concatenates the (orthonormal) subspace basis vectors into the $2d$-dimensional sequence representation as in (7). As in the PCA-SFA representation, the subspace bases are arbitrarily assigned with directional signs and thus we align them so as to produce the projection positively correlated with the time index t by (6). Note that the subspace representation ignores the temporal information (temporal order t) in the sequence. The *VideoDarwin* [11] extracts sequence direction through linear regression from the frame-based features $\boldsymbol{z}_t$ to the time index t, and in this experiment, for fair comparison, we add the mean $\boldsymbol{\mu}$ to the VideoDarwin representation to produce the $2d$-dimensional sequence representation. As a result, the methods other than *mean* produce the same dimensional video descriptor.

Table 2c shows the performance comparison. The performance is degraded by the method of *temporal grid*. The method requires the sequences to be temporally aligned across videos so that the feature vector summarized at each grid is properly matched. Videos, however, are not so aligned due to temporal sifts and variation of actions. In contrast, the other methods statistically exploits the (temporal) variation without such alignment. The *subspace* method considers only the variance of the feature distribution without the temporal ordering information and is inferior to the methods that incorporates such ordering information in the representation. The *VideoDarwin* [11] is still inferior to even SFA and PCA-SFA. The linear regression employed in the VideoDarwin projects the frame-based features $\boldsymbol{z}_t$ into the equally-spaced time index t which is fixed in any sequences without considering the distribution (manifold) in the sequence. The SFA (1) is close to the linear regression in the sense that it also embeds the frame-based features into the fixed (slowest) harmonic oscillation due

(a) Sequence transformation (Sec. 3.2).

The pooling function `pool` in (8) is defined as

identical: $\texttt{pool}(\boldsymbol{x}_1,\cdots,\boldsymbol{x}_t)=\boldsymbol{x}_t$

sum: $\texttt{pool}(\boldsymbol{x}_1,\cdots,\boldsymbol{x}_t)=\sum_{t'}\boldsymbol{x}_{t'}$

max: $\texttt{pool}(\boldsymbol{x}_1,\cdots,\boldsymbol{x}_t)=\max_{t'}(\{\boldsymbol{x}_{t'}\}_{t'=1}^{t})$

Note that in *identical* the feature vectors are not pooled (aggregated) on a sequence.

`pool`	direction	BoW	CNN
identical	–	35.91	55.36
sum	*fwd*	66.39	56.38
max	*fwd*	65.58	55.88
sum	*bwd*	67.17	52.51
max	*bwd*	65.40	53.72
sum	*fwd+bwd*	**68.62**	59.61
max	*fwd+bwd*	67.91	**59.92**

(b) Regularization parameter η in PCA-SFA (4).
η is changed based on $C\triangleq\frac{1}{T}\sum_t\|\dot{\boldsymbol{x}}_t\|_2^2$.

	BoW	CNN	BoW+CNN
$\eta=0$ [SFA (1)]	67.10	58.92	70.75
$\eta=0.1C$	67.83	59.36	71.69
$\eta=0.5C$	68.56	60.07	**72.83**
$\eta=1C$	68.62	59.92	**72.83**
$\eta=5C$	68.98	59.58	72.67
$\eta=10C$	69.06	59.11	72.56

(c) Comparison to other representation.

	BoW	CNN	BoW+CNN
mean	65.39	55.04	68.48
temporal grid [26]	59.70	55.25	66.41
subspace	65.87	52.00	68.83
VideoDarwin [11]	66.26	51.98	70.56
SFA (1)	67.10	58.92	70.75
PCA-SFA (4)	68.62	59.92	**72.83**

Table 2: Performance comparison on HOLLYWOOD2 dataset in terms of (a) sequence transformation (Sec. 3.2), (b) regularization parameter η in PCA-SFA (4) with (c) comparison to the other sequence representations. The mean average precisions (%) are reported.

to the *SSS* problem (see supplemental material). On the other hand, the proposed PCA-SFA (4) effectively extracts the temporal characteristics by taking into account of the distribution of the frame-based feature vectors via PCA.

4.3 Comparison to the other methods

Based on the above analyses, we suggest to apply PCA-SFA (4) with $\eta=\frac{1}{T}\sum_t\|\dot{\boldsymbol{x}}_t\|_2^2$ to BoF feature sequence with sum-pooling and to CNN feature sequence with max-pooling through the bi-directional (forward and backward) sequence transform (8). The proposed method is also tested on HMDB51 and UCF101 as in HOLLYWOOD2 and the performance results are shown in Table 3; we report the classification accuracies (%) on those two datasets.

We can see that the CNN features are inferior to BoW features on HOLLYWOOD2 dataset in which the task is to recognize rather primitive actions, while the holistic CNN features are suitable for classifying actions related to scenes in HMDB51 and UCF101. Then, we combine those CNN and BoW features to further improve classification performance. It is noteworthy that the proposed method renders the fixed-dimensional video descriptors from any types of feature sequence which can be further integrated with the other ones in a simple manner (concatenation). The combination of the two types of motion features favorably boosts performance, being competitive to the other methods; especially on HMDB51, it outperforms the others.

We can analyze the results of the proposed method as follows. In UCF101, the action classes of *"BrushingTeeth"* and *"Shotput"* tend to be misclassified into *"ShavingBeard"* and *"ThrowDiscus"*, respectively, which are composed of similar actions as those classes. Similarly, in HMDB51, *"SwordExercise"* is misclassified to *"DrawSword"*, and in HOLLYWOOD2, *"SitUp"* is occasionally confused with *"StandUp"*. By incorporating more contextual information beyond action itself, we could further improve performance. On the other hand, the classes of smaller number of frames are favorably recognized; 100% on

	HOLLYWOOD2	HMDB51	UCF101
PCA-SFA (BoW)	68.62	60.24	83.12
PCA-SFA (CNN)	59.92	61.66	88.77
PCA-SFA (BoW+CNN)	72.83	**71.83**	93.82
subspace (BoW+CNN)	68.83	68.69	92.70
SFA (BoW+CNN)	70.75	70.68	93.85
Others	64.3 [44]	65.1 [25]	87.9 [31]
	66.3 [24]	65.9 [17]	88.0 [35]
	68.0 [25]	66.8 [32]	89.1 [25]
	73.6 [17]	69.4 [47]	**94.2** [47]

Table 3: Comparison to the other action recognition methods. We show the classification performances reported in the respective papers.

"FloorGymnastics" (avg. 71 frames) in UCF101, 93.33% on *"Dribble"* (avg. 36 frames) in HMDB51, 84.01% on *"StandUp"* (avg. 160 frames) in HOLLYWOOD2.

5 Conclusion

In this paper, we have proposed a method to effectively represent a sequence of frame-based feature vectors for action classification. The proposed method extracts the temporal dynamics patterns from the feature sequence via slow feature analysis (SFA). While the ordinary SFA suffers from small sample size (*SSS*) problem usually found in action clips, we propose PCA-SFA to effectively cope with the *SSS* problem by incorporating PCA subspaces into SFA. In contrast to such as a linear regression, the PCA-SFA exploits the essential temporal pattern in a non-linear manifold based on the slowness principle. We leverage the PCA-SFA projection vector to represent the feature sequence capturing temporal characteristics even in the shorter video clips. It should be noted that the proposed method produces a video descriptor in a computationally efficient way and the computational overhead is negligible compared to the feature extraction process such as by BoW and CNN. The experimental results on action classification using three standard datasets show that the proposed method improves classification performance, being favorably compared with the other methods.

References

[1] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. From generic to specific deep representations for visual recognition. In *CVPR Workshop*, pages 36–45, 2015.

[2] P. Berkes and L. Wiskott. Slow feature analysis yields a rich repertoire of complex cell properties. *Journal of Vision*, 5(6):579–602, 2005.

[3] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, pages 1395–1402, 2005.

[4] A. Bobick and J. Davis. The representation and recognition of action using temporal templates. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 23(3): 257–267, 2001.

[5] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, pages 428–441, 2006.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[7] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, pages 647–655, 2014.

[8] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, pages 2625–2634, 2015.

[9] A. N. Escalante-B and L. Wiskott. How to solve classification and regression problems on high-dimensional data with a supervised extension of slow feature analysis. *Journal of Machine Learning Research*, 14:3683–3719, 2013.

[10] A. N. Escalante-B. and L. Wiskott. Theoretical analysis of the optimal free responses of graph-based sfa for the design of training graphs. *Journal of Machine Learning Research*, 17:1–36, 2016.

[11] B. Fernando, E. Gavves, M. J. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, pages 5378–5387, 2015.

[12] M. Franzius, H. Sprekeler, and L. Wiskott. Slowness and sparseness lead to place, head-direction, and spatial-view cells. *PLoS Computational Biology*, 3(8):1605–1622, 2007.

[13] M. Franzius, N. Wilbert, and L. Wiskott. Invariant object recognition with slow feature analysis. In *International Conference on Artificial Neural Networks*, pages 961–970, 2008.

[14] J. H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989.

[15] T. Hassner. A critical review of action recognition benchmarks. In *CVPR Workshop*, pages 245–250, 2013.

[16] G. E. Hinton. Connectionist learning procedures. *Artificial Intelligence*, 40(1–3):185–234, 1989.

[17] M. Hoai and A. Zisserman. Improving human action recognition using score distribution and ranking. In *ACCV*, pages 3–20, 2014.

[18] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1): 221–231, 2013.

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014.

[20] T.-K. Kim, J. V. Kittler, and R. Cipolla. On-line learning of mutually orthogonal subspaces for face recognition by image sets. *IEEE Transaction on Image Processing*, 19 (4):1067–1074, 2010.

[21] T. Kobayashi. Dirichlet-based histogram feature transform for image classification. In *CVPR*, pages 3278–3285, 2014.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[23] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition. In *ICCV*, pages 2556–2563, 2011.

[24] T. Lan, Y. Zhu, A. R. Zamir, and S. Savarese. Action recognition by hierarchical mid-level action elements. In *ICCV*, pages 4552–4560, 2015.

[25] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj. Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. In *CVPR*, pages 204–212, 2015.

[26] I. Laptev, M. Marzalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[27] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169–2178, 2006.

[28] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, pages 2929–2936, 2009.

[29] M. Nishiyama, O. Yamaguchi, and K. Fukui. Face recognition with the multiple constrained mutual subspace method. In *AVBPA*, pages 71–80, 2005.

[30] E. Oja. *Subspace Methods for Pattern Recognition*. Research Studies Press, 1983.

[31] X. Peng, L. Wang, X. Wang, and Y. Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *CoRR*, abs/1405.4506, 2014.

[32] X. Peng, C. Zou, Y. Qiao, and Q. Peng. Action recognition with stacked fisher vectors. In *ECCV*, pages 581–595, 2014.

[33] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR*, pages 32–36, 2004.

[34] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[35] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[37] Y. Song, L.-P.Morency, and R. Davis. Action recognition by hierarchical sequence summarization. In *CVPR*, pages 3562–3569, 2013.

[38] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human action classes from videos in the wild. In *CRCV-TR-12-01*, 2012.

[39] M. Sugiyama, T. Ide, S. Nakajima, and J. Sese. Semi-supervised local fisher discriminant analysis for dimensionality reduction. *Machine Learning*, 78(1–2):35–61, 2010.

[40] L. Sun, K. Jia, T. Chan, Y. Fang, G. Wang, and S. Yan. Dl-sfa : Deeply-learned slow feature analysis for action recognition. In *CVPR*, pages 2625–2632, 2014.

[41] K. Tang, L. Fei-Fei, and D. Koller. Learning latent temporal structure for complex event detection. In *CVPR*, pages 1250–1257, 2012.

[42] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015.

[43] V.N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[44] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, pages 3551–3558, 2013.

[45] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories andmotion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103:60–79, 2013.

[46] L. Wang and Y. Qiao. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, pages 4305–4314, 2015.

[47] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36, 2016.

[48] A. D. Wilson and A. F. Bobick. Parametric hidden markov models for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:884–900, 1999.

[49] L. Wiskott. Learning invariance manifolds. In *Joint Symposium on Neural Computation*, pages 196–203, 1998.

[50] L. Wiskott. Slow feature analysis: A theoreticalanalysis of optimal free responses. *Neural Computation*, 15(9):2147–2177, 2003.

[51] L. Wiskott and T. J. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002.

[52] Z. Zhang and D. Tao. Slow feature analysis for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):436–450, 2012.

[53] G. Zhao, T. Ahonen, J. Matas, and M. Pietikäinen. Rotation-invariant image and video description with local binary pattern features. *IEEE transactions on image processing*, 21(4):1465–77, 2012.