DL-SFA: Deeply-Learned Slow Feature Analysis for Action Recognition

Lin Sun*‡§, Kui Jia†, Tsung-Han Chan†, Yuqiang Fang*, Gang Wang♯, Shuicheng Yan*
*Department of Electrical and Computer Engineering, National University of Singapore
†Advanced Digital Sciences Center, Singapore
‡Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology
♯ Nanyang Technological University, Singapore
§Lenovo Corporate Research Hong Kong Branch

Abstract

Most of the previous work on video action recognition use complex hand-designed local features, such as SIFT, HOG and SURF, but these approaches are implemented sophisticatedly and difficult to be extended to other sensor modalities. Recent studies discover that there are no universally best hand-engineered features for all datasets, and learning features directly from the data may be more advantageous. One such endeavor is Slow Feature Analysis (SFA) proposed by Wiskott and Sejnowski [33]. SFA can learn the invariant and slowly varying features from input signals and has been proved to be valuable in human action recognition [34]. It is also observed that the multi-layer feature representation has succeeded remarkably in widespread machine learning applications. In this paper, we propose to combine SFA with deep learning techniques to learn hierarchical representations from the video data itself. Specifically, we use a two-layered SFA learning structure with 3D convolution and max pooling operations to scale up the method to large inputs and capture abstract and structural features from the video. Thus, the proposed method is suitable for action recognition. At the same time, sharing the same merits of deep learning, the proposed method is generic and fully automated. Our classification results on Hollywood2, KTH and UCF Sports are competitive with previously published results. To highlight some, on the KTH dataset, our recognition rate shows approximately 1% *improvement in comparison to state-of-the-art methods even without supervision or dense sampling.*

1. Introduction

Recognizing human action in realistic videos is challenging while has widespread applications, including unusual activity detection [21], human computer interactions (HCI) and so forth. Modern feature extraction methods or hand-designed features are innovating this task and achieving remarkable performance. However, these hand-designed features, such as HOG/HOF[16] and HOG3D[13], are designed with specific purpose. They may not be able to be generalized to other datasets in real-world scenarios since it is rarely known which features are important to the task at hand. Even with the manual assistance in selecting features and methods, accurate human action recognition is still a highly cumbersome task due to complex background, different illumination environment and significant intra-class variations. Given that human beings are able to reliably know what the video tells without assistance, why can't computers?

Learning features directly from the data could be a feasible solution since the learned features are expected to be more generalizable than hand-designed ones. We therefore can see growing interests in development of unsupervised feature learning methods recently; these include Slow Feature Analysis (SFA) [33], Deep Belief Nets (DBN) [7, 8] and other methods[19, 2, 25]. SFA intends to capture the invariant and slowly-varied features from input signals, and has been successfully applied to the self-organized receptive field of cortical neuron from synthetic image sequences [33], and to robust recognition of whole objects [6]. For human motion analysis, SFA can be adopted to reduce the semantic gap between the quickly varying image input signals and the slowly varying action categories. Deep learning models [7, 2, 8] have shown promising and plausible results in various applications. These methods focus on learning to extract multiple layers of features which can hierarchically represent the contents with increasingly abstract at each level. The convolutional neural networks (CNN) [18] is an important type of deep models in which trainable filters and local neighborhood pooling are applied alternatingly on the raw data, resulting in a hierarchy of increasingly complex features. Inspired by the achievement of deep learning in visual recognition and the fact that using SFA alone, which can be seen as one-layer SFA, may

not work well for the complex video action recognition, we integrate SFA into a two-layered neural network, termed as deeply-learned SFA (DL-SFA) to extract the hierarchical 'slow' features of the video. The proposed DL-SFA adopts the notion of 3D convolution, max-pooling to capture abstract, structural and translational invariant features. It is worth noting that the deep architectures already consist of feature detector units; and that lower layers detect simple features and feed into higher layers, which in turn detect more complex features. This indicates that the DL-SFA can be learned in a fully unsupervised manner, unlike the previous hand-designed methods which require an additional feature detector in the learning process [5][24].

To make a fair comparison, a standard processing pipeline is followed as described in Wang et al.[31], but only replacing the first stage of feature extraction with our method. By doing this, we can understand the contribution of the deeply-learned slow features compared with other hand-designed features. Our method is evaluated on three well-known human action recognition benchmark datasets: Hollywood2 [23], KTH [28] and UCF Sports[27]. According to the experimental results, our method outperforms most of the published methods using either hand-designed [31][27] or learned features [11] on challenging datasets even without supervision or dense sampling.

2. Previous Work

Over the past years, the low-level 2-D features, such as SIFT [22], HOG [16] and SURF [1] have been successfully employed in static images, and some have been extended to 3-D for action recognition. Usually these methods have two steps: feature detection and feature extraction. Feature detection extracts interesting or saliency points by applying Harris (spacial)[24], Harris3D [15] (spacial-temporal), temporal Gabor filter [5], or Hessian [32] detector. HOG/HOF [16] and other techniques such as HOG3D [13] are all the feature extraction methods. Despite the noticeable success of the existing low-level features in particular datasets, choosing features that work on the data at hand is still very challenging. To this end, Wang et al.[31] recently combine various low-level feature detection, feature extraction methods and benchmark their performance on several action recognition datasets. They employ the same state-of-the-art processing pipeline with vector quantization, feature normalization and χ^2-kernel Support Vector Machines (SVMs). The only difference in this pipeline is the method of feature detection and feature extraction in order to evaluate the performance of variable detectors and descriptors along with their combinations. This paper also demonstrates that regular sampling consistently outperforms all the test space-time detector and there are no global features which work well on all datasets.

Recently, a novel and interesting work [34] proposed by Zhang et al. shows the effectiveness of slow features for action recognition. They develop three new supervised SFA learning strategies, including the supervised SFA (S-SFA), the discriminative SFA (D-SFA) learning, and the spatial discriminative SFA (SD-SFA), to address the classification task and enhance the selectivity of the learned slow features to different actions. However, their method is designed particularly for the KTH which may not be able to handle more complex datasets, such as Hollywood2 or UCF Sports. In addition, this method needs foreground and label information while training SFA kernels, and therefore may not be well-suited in general action recognition applications. As the authors extracted the slow features from the input data directly and only once, [34] can be somehow thought of as one-layer SFA, a special case of our proposed DL-SFA.

Another intriguing work is invariant spatio-temporal features for action recognition using Independent Subspace Analysis (ISA) proposed by Quoc et al. [17]. The authors present an extension of the ISA to learn invariant spatio-temporal features from unlabeled video data. It is similar to this work in that ISA is combined with deep learning techniques such as stacking and convolution to learn hierarchical representations. More specifically, both [17] and our method integrate existing (shallow) feature learning methods into a deep and convolutional feature extraction architecture. The difference is that [17] used ISA, while our method considers SFA. Objective function of ISA makes its learned feature representations tend to be invariant to spatial translation of stimuli, but highly sensitive to velocity change of the stimuli. Different from ISA, SFA is an interesting technique which aims at learning a mapping function from the noisy and quickly varying sensory data to a more stable and slowly varying feature representation, which is deemed to be a better representation of motion pattern. The notion of slowness in SFA includes not only spatial invariance, but also other high-frequency oscillations of input signals, such as velocity difference of a same action category performed by different human subjects. It may remain an open question about which feature learning method, ISA or SFA, is more useful for action recognition. However, according to our internal experiments on Hollywood2 dataset in which we use the source codes from the authors' website and set the parameters consistent with [17], the performance of Hierarchical ISA will reduce by about $8\% - 10\%$ without dense sampling. This reflects the advantage of our use of SFA over [17].

Recently, biologically-inspired networks or deep models that can learn features automatically from the hierarchical structure become dominant in machine learning area. These methods are generic since the features are directly learned from the raw pixels automatically. A novel convolutional GRBM method [29], an extension of convolutional RBMs [20] to 3D, is proposed for learning spatio-temporal

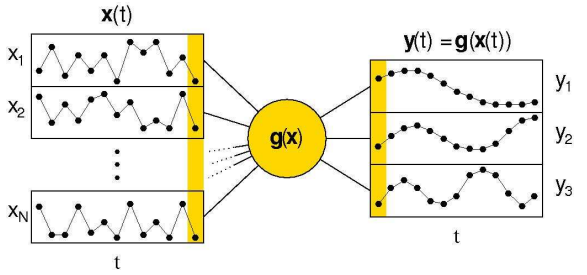
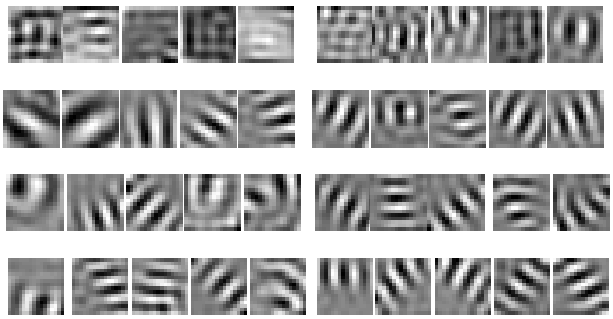


Figure 1. Schematics of the optimization problem solved by SFA. The SFA learns the functions g_j that transforms $\boldsymbol{x}(t)$ to slowly-varying output signals $y_j(t) = g_j(\boldsymbol{x}(t))$, so as to represent the input $\boldsymbol{x}(t)$ in a more abstract way.



(a) (b)

Figure 2. Visualization of the slow feature functions learned from Hollywood2 dataset. (a) The optimal excitatory stimuli and (b) the optimal inhibitory stimuli.

features. [12] proposes to use a modified CNN for action recognition, operating on spatio-temporal outer boundaries volume. Jhuang et.al [10] present a biologically-motivated system in which an array of motion-direction sensitive unit is analyzed first. Ji et al. [11] propose a 3D CNN for action recognition with combination of multiple hand-wired features as input. Nonetheless, these methods either use the label information or hand-crafted features as the input.

3. Algorithm

In this section, we first give a brief introduction of slow feature analysis (SFA), mainly focusing on why the learned 'slow' features are effective in human motion analysis and how we use SFA to extract these features from image sequences (video). Then we elaborate the proposed DL-SFA algorithm for human action recognition.

3.1. Slow Feature Analysis

One can treat perception as the problem of reconstructing the external causes of the sensory input to allow generation of adequate behavior. For example, when looking at a picture on a computer screen, we see the objects and their relative positions in the image, rather than the colors of the individual pixels. An important idea in the field is that objects in the world have a common structure, which results in statistical regularities in the sensory input. Using these regularities as a guide, the brain is able to form a meaningful representation of the environment.

Many researchers have tried to develop algorithms to mimic the functions of visual cortex neurons using different computational principles. The slowness principle is one of them. For instance, behaviorally relevant visual elements (objects and their attributes) are visible for extended periods of time and change with time in a continuous fashion, on a time scale of seconds. On the other hand, the primary sensory signal, like the responses of individual retinal receptors or the gray-scale value of a single pixel in a video camera, are sensitive to very small changes in the environment, and thus vary on a much faster time scale. If explicitly representing the original visual elements, the internal representation of the environment in the brain should vary on a slow time scale again. This difference in time scales leads to the central idea of the slowness principle: by finding and extracting slowly varying output signals from the quickly varying input signal , we seek to recover the underlying external causes of the sensory input [4]. Fig. 1 (vivid example from [14]) shows one example of what SFA does.

Given an input signal $\boldsymbol{x}(t) = [x_1(t), \ldots, x_I(t)]^T$ with $t \epsilon [t_0, t_1]$ indicating time, SFA finds out a set of functions g_j for $j = 1, \ldots, J$ such that $y_j(t) = g_j(\boldsymbol{x}(t))$ for all j transforms the severely changed signals into the signals varying as slowly as possible; that is, SFA solves the following optimization problem:

$$\min_{g_j, \forall j} \sum_{j=1}^{J} \mathbb{E}[\dot{g}_j^2(\boldsymbol{x}(t))] \tag{1}$$

$$\text{s.t.} \quad \mathbb{E}[g_j(\boldsymbol{x}(t))] = 0, \quad \mathbb{E}[g_j^2(\boldsymbol{x}(t))] = 1, \tag{2}$$

$$\mathbb{E}[g_i(\boldsymbol{x}(t)) g_j(\boldsymbol{x}(t))] = 0, \tag{3}$$

$$\forall j \neq i, \ i = 1, ..., J. \tag{4}$$

where $\dot{g}_j$ denotes the operator of computing the first-order derivative of $g_j(\boldsymbol{x}(t))$ with respect to t and $\mathbb{E}$ denotes the sample mean operator over time t. The objective reduces the temporal variation of $y_j(t)$ by minimizing (1); that is, the mean of the power of the first-order derivative of $y_j(t)$. The constraint (2) enforces that the output $y_j(t)$ should have zero mean and unit variance, and the constraint (3) restricts that the J outputs $y_1(t), ..., y_J(t)$ are mutually uncorrelated. Problem (1) has a closed-form solution if g_j is linear in $\boldsymbol{x}(t)$; e.g., $g_j(\boldsymbol{x}(t)) = \boldsymbol{w}_j^T \boldsymbol{x}(t)$ where $\boldsymbol{w}_j$ is a weighting vector or sort of temporal filter. As such, SFA turns out to solve a generalized eigenvalue problem [33]:

$$\mathbb{E}[\dot{\boldsymbol{x}}(t)\dot{\boldsymbol{x}}(t)^T]\boldsymbol{W} = \mathbb{E}[\boldsymbol{x}(t)\boldsymbol{x}(t)^T]\boldsymbol{W}\boldsymbol{D}, \tag{5}$$

where the weighting vectors $\boldsymbol{W} = [\boldsymbol{w}_1, ..., \boldsymbol{w}_J] \in \mathbb{R}^{I \times J}$ are identical to the generalized eigenvectors and $\boldsymbol{D}$ is a di-

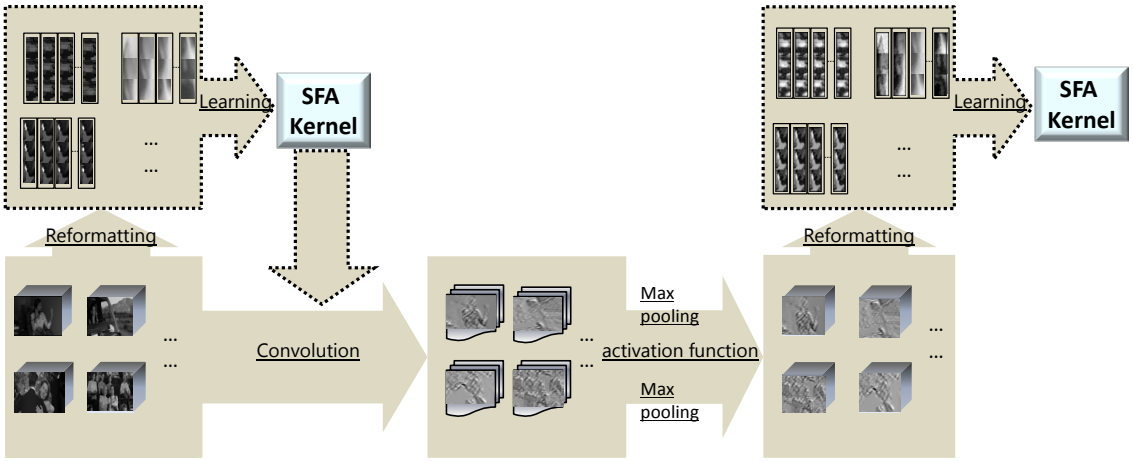


Figure 3. DL-SFA learning architecture for human action recognition. This architecture contains one convolution layer, two max-pooling operations, the process of reorganizing/reformatting and two SFA learning layers. Details are presented in the main text.

agonal matrix of the generalized eigenvalues. The order of slow features is determined by the eigenvalues, where the most slowly varying signal has the lowest index; please see [33] for details. The above SFA learning is expected to discover mapping functions between input image sequences that vary quickly and the corresponding high-level semantic concepts that vary slowly. We illustrate the optimal excitatory stimuli and the optimal inhibitory stimuli learned from the Hollywood2 dataset in Fig. 2. They correspond to the stimuli that elicit in the considered function, the highest and the lowest response, respectively. The optimal stimuli have the shape of localized gratings and largely resemble those of simple cells and complex cells.

3.2. Deeply-Learned Slow Feature Analysis (DL-SFA)

In this section, we elaborate how to incorporate SFA into the deep video feature learning structure. Applying SFA directly to the whole video volume is very time-consuming because the video sequences usually have high resolution with a large number of frames. Besides, learning a single global kernel from an entire action sequence may not be useful because of the complex variations in the high-dimensional image space. We, therefore, propose to use a local-based hierarchical approach to recognize human actions, that is sampling cuboids from the video sequence instead of using entire video as the training data. The whole learning diagram is presented in Fig. 3. In the beginning, a large number of local cuboids are collected by randomly sampling from within a large collection of video sequences. According to Berkes and Wiskott's work [3], we reorganize each input vector by δt successive frames, so SFA counts the temporal information in the neighbor frames. Fig. 4 visualizes the reorganizing procedure. Then these spatial-temporal volumes are fed into a succession of SFA learning

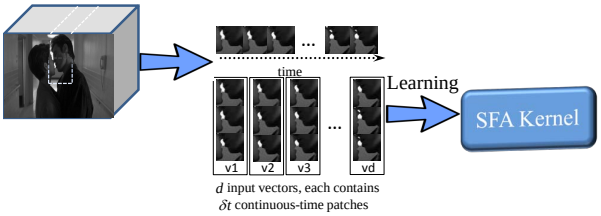


Figure 4. Video cuboid reorganizing procedure. Each cuboid contains $d + (\delta t - 1)$ frames and then are reformatted to d input vectors at each time including δt successive patches (in this example $\delta t = 3$). Hence, at last the original $d + (\delta t - 1)$ frames will be reformatted to d input vectors with δt successive patches.

system to learn the first layer and second layer SFA kernels, respectively. During stacking the second layer onto the first one, it has a convolution layer in which a 3D convolution is performed using the learned SFA kernels of the first layer and sliding it around over every pixel in the original video sequence both in spatial and temporal domain. The 3D convolution is achieved by convolving a 3D kernel with the cuboid formed by stacking multiple contiguous frames together. Formally, the cuboid unit at position (x, y, z) in the jth feature map of the ith layer with activation function is denoted:

$$c_{i,j,k}^{x,y,z} = tanh\Big(\sum_{w=0}^{K_{W_i}-1} \sum_{h=0}^{K_{H_i}-1} \sum_{t=0}^{K_{T_i}-1} w_{i,j}^{w,h,t} c_{(i-1),k}^{(x+w)(y+h)(z+t)} \Big), \tag{6}$$

where $tanh$ is the hyperbolic tangent activation function, k indexes the sample number in the $(i-1)$th layer, $w_{i,j}^{w,h,t}$ is the value of the jth SFA kernel at the position (w, h, t) with the size of $K_{W_i}, K_{H_i}, K_{T_i}$ in the ith layer, respectively. The output of convolutional video cuboid features is given by the maximum activation which can be defined as the max-pooling layer. The max pooling is applied to all

convolutional local features to obtain the robust representation,

$$c_{i,k}^{x,y,z} = \max_{j}(c_{i,j,k}^{x,y,z}) \tag{7}$$

After convolution and pooling, the k-th cuboid in the i-th layer is denoted as:

$$C_{i,k} = \{c_{i,k}^{x,y,z} \mid 0 \leq x \leq W_k - 1, 0 \leq y \leq H_k - 1, \\ 0 \leq z \leq T_k - 1\}, \tag{8}$$

where W_k, H_k, T_k denote the width, height and number of frames of the video cuboid, respectively. Then $C_{i,k}$ will pass through a 2×2 spatial max-pooling. This can characterize the slow features of video volume from both spatial and temporal dimensions. The utilization of max-pooling is useful here for two reasons. Firstly, since max-pooling reduces the learned features to one or a small number of features, it will reduce the computational complexity for upper layers. And secondly, it provides a form of translational invariance. In the pooling operation (spatial), the resolution of the feature maps is reduced by pooling over local neighborhood on the feature maps in the previous layer, thereby increasing invariance to distortions on the inputs and making the next layer capture more abstract details. The original video sequences are convolved with the learned first-layer SFA kernels and max pooled to generate the training data for the second layer. During the testing, we apply the DL-SFA kernels consecutively so as to obtain the features that are more abstract to represent the video sequence. The feature extraction procedure of DL-SFA is presented in Fig. 5. In order to obtain an abstract and robust descriptor of the video volume, we combine the features from both the first and second layer. Then we construct a codebook for the descriptor and use Vector Quantization to build the Bag of Visual Words (BoW) representation. We train the codebooks by clustering on the regularly collected video cuboids of the whole training video sequences using k-means implemented in [30]. According to our investigation, we could achieve the best performance when codebooks equals 4000. The whole procedure is shown in Algorithm 1.

Some exemplar features of Hollywood2 dataset extracted from the first layer of our structure are visible in Fig. 6. They demonstrate that the first layer mostly detect the edge and the contour. Besides, in the listed consecutive features of one cuboid, the temporal consistency is well kept which make our algorithm work well in detecting the meaningful features for video.

4. Experiments

In this section, the numerical results of different action recognition algorithms on three datasets are presented. After extracting local feature descriptors, vector quantization by k-means and classification by χ^2 kernel SVM is performed. We only replaced the feature extraction stage with the DL-SFA descriptor and other feature descriptors so as to make the comparison reasonable.

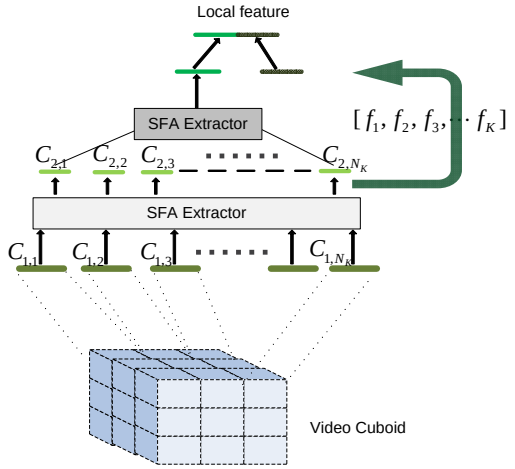


Figure 5. Features extraction structure of DL-SFA for videos. Herein, $C_{1,1}, C_{1,2}, \cdots C_{1,N_K}$ and $C_{2,1}, C_{2,2}, \cdots C_{2,N_K}$ are the first layer and seconde layer cuboids as indicated in the paper, where N_K is the number of cuboids generated. The local features combine the features extracted from the first-layer and second-layer.

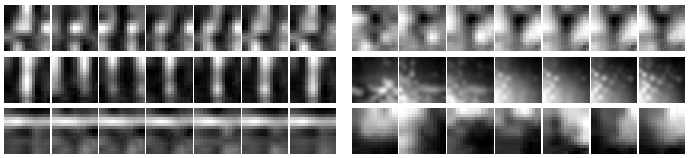


Figure 6. Examples of the features extracted using our first-layer SFA from the Hollywood 2 dataset (in order to have a better visualization we use higher resolution than the original one). On a given row, each frame corresponds to one spatial frame with time index fixed, so that times goes from left to right.

In the experiment we set the first layer's cuboids sampling size as 14×14 multiplies 17 consecutive frames. The cuboids sampling size is 7×7 times 17 consecutive frames in the second layer. In the reorganizing process we accept $\delta t = 8$ in order to balance the motion information and complexity. Finally we store 300 and 200 features for the first layer and the second layer, respectively. Here we carry out our experiments on three human action recognition datasets.

Hollywood2 action dataset is collected from 69 Hollywood movies which are divided into 33 training movies and 36 testing movies. Hollywood2 provides a dataset with 12 classes of human actions: answering phone, driving car, Eat, fighting person, get out of the car, hand shaking, hug person, kissing, running, sitting down, sitting up and standing up. This dataset (some samples shown in Fig. 7), is more challenging than other datasets because it has more complex backgrounds and context environments. In our experiments, we use 823 video sequences as the training data and 884 video sequences for testing. The performance is

Algorithm 1 DL-SFA for Action Recognition

Input: All the training video sequences and testing video sequences.

Training Procedure:
1. Randomly sample cuboids from unlabeled training video sequences and reorganize them according to Fig. 4, generating $C_{1,k}$ for $k = 1, ..., S$; S indexes the number of sampling cuboids;
2. Learn the first layer SFA kernels $\boldsymbol{W}_1$ [Eq. (1)-(5)];
3. Convolve the training video sequences with $\boldsymbol{W}_1$ and then max pooling to get the $C_{2,k}$ (Eq. (6)-(8));
5. Learn the second layer SFA kernels $\boldsymbol{W}_2$ [Eq. (1)-(5)];

Testing Procedure:
1. Regularly and non-overlapped scanning testing video sequences with SFA kernel W_1 to generate the first layer feature f^{L1}. Then convolve SFA kernel W_2 to the cuboid which has been processed through convolution and max pooling to obtain the second layer feature f^{L2}. Then the video sequences can be represented as a bag of local spatio-temporal features, denoted by $\{f_{(i,j)}, i = 1, ..., K,\ j = 1, ..., N_K\}$, where $f_{(i,j)} = [f^{L1}_{(i,j)}\ \ f^{L2}_{(i,j)}]$ where K is the number of video sequences and N_K is the number of cuboids in the Kth testing video.

2. Quantization into $V = v_1, v_2, \cdots v_K$ visual words and features are assigned to the closest vocabulary word using Euclidean distance.

3. Classification using non-linear support vector machine with χ^2 kernel:

$$K(H_i, H_j) = exp(-\frac{1}{2A}\sum_{n=1}^{K}\frac{h_{in} - h_{jn}}{h_{in} + h_{jn}})$$

where $H_i = \{h_{in}\}$ and $H_j = \{h_{jn}\}$ are the frequency histograms of word occurrences for training data and testing data over the visual words. A is the mean value of distances between all the samples.

Output: The classification results: $mean\_AP$ which is the mean of average precision over all classes. $Accuracy$ which is the average accuracy over all classes.

evaluated by the mean average precision for all the classes.

KTH action dataset contains six types of human actions: walking, jogging, running, boxing, hand waving and hand clapping which are performed several times by 25 subjects in four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes and indoors. The background is homogeneous and static. Thus the performance reported has been very high. In the implementation, we also follow the original experiment setup as indicated in [28] and report the average accuracy over all the classes as the performance measure.

UCF Sports action dataset consists ten different human actions: swinging (on the pommel horse and on the floor), diving, kicking, weight-lifting, horse riding, running, skateboarding, swinging (at the high bar), golf swinging and walking. This dataset contains 150 video sequences in total which show large intra-class variabilities. In the experiments we follow the protocol in [31]: extending the dataset by adding a horizontally flipped version of each sequence to the dataset. Similar to the KTH, we train a multi-class classier and report the average accuracy over all classes.

Table 1, Table 2 and Table 3 present the performance results of the Hollywood2, KTH and UCF Sports, respectively. Since all the experiments in the table follow the same pipeline, the performance results become comparable. When testing, we only apply non-overlapped sampling with regularly scanning. Our method is totally unsupervised without taking any advantages of label information or interesting region. However, our method outperforms all the hand-designed methods compared in the experiments on three datasets due to the complex video representation of the DL-SFA. We also present a comparable performance with state-of-the-art Hierarchical ISA in which it does dense sampling, interesting points detection [17] (for KTH) and norm-thresholding (for KTH). We even achieve better performance than [17] on KTH and UCF datasets. Particular for KTH, ours is 93.5% while the performance of [17] is 91.4% (dense sampling without norm-thresholding). However, it exists a small gap between our method and [17] on Hollywood2 dataset. Here we should state again that the performance of [17] largely relies on the dense sampling and some other tricks. Without these tricks the performance will be about $8\% \sim 10\%$ worse than the paper has reported. We can also expect our performance to be better after applying some tricks such as regularly dense sampling or interesting points detection.

4.1. Deep learning structure

In the tables , the experimental results of one-layer unsupervised SFA learning based feature extraction are also presented for three datasets. All the results in the tables show that our deeply-learned SFA is better than using unsupervised SFA directly to the video. The performance gap can be 9.4%, 5.2% and 6.8% for Hollywood2, KTH and UCF Sports, respectively. This result is also consistent with the conclusion summed up in [9] that the hierarchical layer learning is better than shallow learning (such as one layer). This also reflects that the features of higher layer (second layer or higher) can provide a vivid representation which the one-layer features can not.

Figure 7. Sample frames from video sequences of Hollywood2

4.2. Max pooling

To some extent, action recognition in videos share similar issues with object recognition in static images. Both tasks have to deal with significant intra-class variations, back-ground clutter, occlusions and so forth. The standard implementation is pooling all local features to obtain a robust and general video representation. In the proposed method, max pooling is used to obtain spacial and temporal invariant features. What is more, it reduces the computational complexity for the next layer.

Table 1. Mean AP on the Hollywood2 dataset.

Algorithm	Mean AP
Hessian[32] + ESURF [16]	38.2%
Harris3D [24] + HOG/HOF [16] (from [31])	45.2%
Dense + HOG3D [13]	45.3%
Hessian[32] + HOG/HOF [16]	46.0%
Cuboid [5] + HOG/HOF [16]	46.2%
GRBM [29]	46.6%
Dense + HOG/HOF [16]	47.7%
Hierarchical ISA [17]	53.3%
One Layer SFA	**38.7%**
Our Method (DL-SFA)	**48.1%**

Table 2. Average accuracy on the KTH dataset.

Algorithm	Accuracy
Hessian[32] + ESURF [16]	81.4%
pLSA [26]	83.3%
Dense + HOF[16]	88.0%
Cuboid [5] + HOG3D [13]	90.0%
GRBM [29]	90.0%
3D CNN [11]	90.2%
Hierarchical ISA with dense sampling[17]	91.4%
HMAX[10]	91.7%
Harris3D [24] + HOG/HOF [16] (from [31])	91.8%
Harris3D [24] + HOF [16] (from [31])	92.1%
Hierarchical ISA with dense sampling [17]	91.4%
One Layer SFA	**87.9%**
Our Method (DL-SFA)	**93.1%**

Table 3. Average accuracy on the UCF sports dataset.

Algorithm	Accuracy
Hessian[32] + ESURF [16]	77.3%
Harris3D [24] + HOG/HOF [16] (from [31])	78.1%
Hessian[32] + HOG/HOF [16]	79.3%
Dense + HOF [16]	82.6%
Cuboids [5] + HOG3D [13]	82.9%
Dense + HOG3D [13]	85.6%
Hierarchical ISA [17]	86.5%
One Layer SFA	**79.8%**
Our Method (DL-SFA)	**86.6%**

5. Conclusion

In this paper we propose a hierarchical deeply-learned SFA (DL-SFA) structure to extract more abstract and robust features in comparison with shallow SFA for video volumes. Our structure incorporates the concept of convolution and max pooling which make our algorithm have the property of translation invariance and hierarchy. This hierarchical 'slow' features make the final performance comparable if not better than state-of-the-art method.

Experiments are carried out on three realistic human action datasets with identical processing pipeline. The results present the efficiency of our method. Since our method does not need any assistance of label information in the training processing, this paper also suggests that learning features directly from the raw video data is a very important trend for action recognition. In addition, in this paper, balancing the performance and complexity, we only consider a 2-layer SFA. We expect to obtain a better performance and a more abstract video representation when proper layers added.

6. Acknowledgment

This research is partially supported by the Singapore National Research Foundation under its International Research Centre @Singapore Funding Initiative and administered by the IDM Programme Office.

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008.

[2] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, U. D. Montral, and M. Qubec. Greedy layer-wise training of deep networks. In *In NIPS*. MIT Press, 2007.

[3] P. Berkes and L. Wiskott. Applying slow feature analysis to image sequences yields a rich repertoire of complex cell properties. In *Proceedings of the International Conference on Artificial Neural Networks*, ICANN '02, pages 81–86, London, UK, UK, 2002. Springer-Verlag.

[4] P. Berkes and L. Wiskott. Slow feature analysis to image sequences yields a rich repertoire of complex cell properties. volume 5, pages 579–602, 2005.

[5] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Proceedings of the 14th International Conference on Computer Communications and Networks*, ICCCN '05, pages 65–72, Washington, DC, USA, 2005. IEEE Computer Society.

[6] M. Franzius, N. Wilbert, and L. Wiskott. Invariant object recognition with slow feature analysis. In *Proceedings of the 18th international conference on Artificial Neural Networks, Part I*, ICANN '08, pages 961–970, Berlin, Heidelberg, 2008. Springer-Verlag.

[7] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507, 2006.

[8] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006.

[9] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *ICCV*, pages 2146–2153, 2009.

[10] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *In ICCV*, pages 1–8, 2007.

[11] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):221–231, Jan. 2013.

[12] H.-J. Kim, J. S. Lee, and H.-S. Yang. Human action recognition using a modified convolutional neural network. In *Proceedings of the 4th international symposium on Neural Networks: Part II–Advances in Neural Networks*, ISNN '07, pages 715–723, Berlin, Heidelberg, 2007. Springer-Verlag.

[13] A. Klaeser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *Proc. BMVC*, pages 99.1–99.10, 2008. doi:10.5244/C.22.99.

[14] T. Kühnl, F. Kummert, and J. Fritsch. Monocular road segmentation using slow feature analysis. In *Intelligent Vehicles Symposium*, pages 800–806, 2011.

[15] I. Laptev and T. Lindeberg. Space-time interest points. In *IN ICCV*, pages 432–439, 2003.

[16] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Conference on Computer Vision & Pattern Recognition*, jun 2008.

[17] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 3361–3368, Washington, DC, USA, 2011. IEEE Computer Society.

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Intelligent Signal Processing*, pages 306–351. IEEE Press, 2001.

[19] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 801–808. MIT Press, Cambridge, MA, 2007.

[20] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 609–616, New York, NY, USA, 2009. ACM.

[21] W. Lin, M.-T. Sun, R. Poovendran, and Z. Zhang. Human activity recognition for video surveillance. In *ISCAS*, pages 2737–2740. IEEE, 2008.

[22] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society.

[23] M. Marszałek, I. Laptev, and C. Schmid. Actions in context. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2009.

[24] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer Vision-Part I*, ECCV '02, pages 128–142, London, UK, UK, 2002. Springer-Verlag.

[25] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *Int. J. Comput. Vision*, 79(3):299–318, Sept. 2008.

[26] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *Int. J. Comput. Vision*, 79(3):299–318, Sept. 2008.

[27] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach: a spatio-temporal maximum average correlation height filter for action recognition. In *In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.

[28] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *International Conference on Pattern Recognition*, 2004.

[29] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *Proceedings of the 11th European conference on Computer vision: Part VI*, ECCV'10, pages 140–153, Berlin, Heidelberg, 2010. Springer-Verlag.

[30] A. Vedaldi and B. Fulkerson. Vlfeat: an open and portable library of computer vision algorithms. In *Proceedings of the international conference on Multimedia*, MM '10, pages 1469–1472, New York, NY, USA, 2010. ACM.

[31] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *Proc. BMVC*, 2009.

[32] G. Willems, T. Tuytelaars, and L. Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, pages 650–663, Berlin, Heidelberg, 2008. Springer-Verlag.

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

[34] Z. Zhang and D. Tao. Slow feature analysis for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(3):436–450, Mar. 2012.