# WALKING WALKing walking: Action Recognition from Action Echoes

**Qianli Ma[†], Lifeng Shen[†], Enhuan Chen[†], Shuai Tian[†], Jiabing Wang[†], Garrison W. Cottrell[‡]**
[†] School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
[‡] Department of Computer Science and Engineering, University of California, San Diego, CA, USA
qianlima@scut.edu.cn, scuterlifeng@foxmail.com

## Abstract

Recognizing human actions represented by 3D trajectories of skeleton joints is a challenging machine learning task. In this paper, the 3D skeleton sequences are regarded as multivariate time series, and their dynamics and multiscale features are efficiently learned from action echo states. Specifically, first the skeleton data from the limbs and trunk are projected into five high dimensional nonlinear spaces, that are randomly generated by five dynamic, training-free recurrent networks, i.e., the reservoirs of echo state networks (ESNs). In this way, the history of the time series is represented as nonlinear echo states of actions. We then use a single multiscale convolutional layer to extract multiscale features from the echo states, and maintain multiscale temporal invariance by a max-over-time pooling layer. We propose two multi-step fusion strategies to integrate the spatial information over the five parts of the human physical structure. Finally, we learn the label distribution using softmax. With one training-free recurrent layer and only layer of convolution, our Convolutional Echo State Network (ConvESN) is a very efficient end-to-end model, and achieves state-of-the-art performance on four skeleton benchmark data sets.

## 1 Introduction

Human action recognition is an active research branch in machine learning, with a wide range of applications in smart home scenes, human motion analysis and human-computer interaction, etc. In the past decades, most work on action recognition has focused on the analysis of 2D camera-based RGB video sequences. However, there are many difficulties with this modality, including illumination changes, viewpoint variations, and occlusions, etc. In other words, 2D cameras do not fully capture the human motion in 3D space. Currently, human 3D-skeleton positions can be accurately and easily extracted from a single depth image [Shotton *et al.*, 2013], greatly accelerating the progress in action recognition. Compared with 2D video-based images, 3D human skeleton points are more appropriate for representing the natures of human actions.

3D skeleton-based action recognition is still a challenging task. One of the biggest challenges is that semantically similar motions may not necessarily be numerically similar [Presti and Cascia, 2016]. The approaches to this problem can be broadly divided into two categories: feature-based and dynamics-based. Feature-based methods extract pose representations or discriminative joint subsets from the skeletons to capture the correlation of body joints, and evaluate the similarity of different actions using a suitable metric [Devanne *et al.*, 2013; Hussein *et al.*, 2013; Vemulapalli *et al.*, 2014; Gong *et al.*, 2014; Zhang *et al.*, 2016]. While feature-based approaches are efficient to some extent, designing representative features and selecting the appropriate metric is time-consuming and error-prone.

On the other hand, dynamics-based methods treat skeleton data as 3D trajectories of body joints [Lo Presti *et al.*, 2015; Slama *et al.*, 2015]. From the view of physical structure, the human body is a skeleton-based articulated system with five parts (two arms, two legs and a center trunk), and human actions are time-varying combinations of these parts (ignoring head movements). In this sense, the skeleton sequences can be regarded as multivariate time series, and recognizing human actions can be seen as a time series classification (TSC) problem [Gong *et al.*, 2014]. Popular dynamics-based methods include: Hidden Markov Models (HMMs) [Xia *et al.*, 2012; Wu and Shao, 2014], Long-Short Term Memory networks (LSTM) [Du *et al.*, 2015; Zhu *et al.*, 2016; Song *et al.*, 2016]. However, for the TSC problem, it is crucial to identify dynamical patterns in their temporal *context*, as many patterns are locally similar, but correspond to different actions depending on previous actions. HMMs lack the ability to maintain the long-term temporal history of a movement. Although LSTMs can learn contextual information for given sequences, they must be trained, which is very time-consuming. Moreover, it is known that time series often have features at different time scales, and LSTMs are poor at capturing multiscale features in time series. Hence, it is a challenge to simultaneously and efficiently model the dynamics and capture multiscale temporal features of 3D skeleton sequences.

To address these problems, we regard the 3D skeleton sequences as multivariate time series and propose a novel neural network approach called the Convolutional Echo State Network (ConvESN) for the skeleton-based action recognition

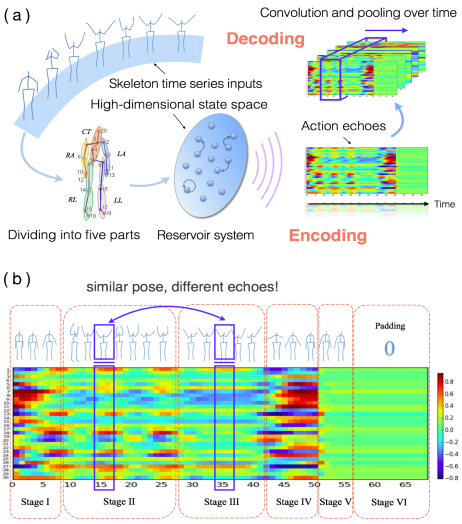



Figure 1: (a) Illustration of the proposed Convolutional Echo State Network (ConvESN) approach. Note that we drive five separated reservoirs using five parts of the skeleton inputs; only one is shown for simplicity. The upper-right representations feed into the fusion schema (see Figure 4). (b) Heat map of left arm trajectories of a person waving his two hands (54 frames) and the corresponding action echo states (echo-state representations). Note the representation of the same pose differs depending on the context.

task. ConvESN can automatically learn the dynamics and multiple time-scale features of time series from action echo states. The general architecture of the ConvESN is illustrated in Figure 1(a). It includes an encoding part and decoding part. In the encoding part, the skeleton data are projected into a high dimensional nonlinear space, which is randomly generated by a dynamic training-free recurrent net, i.e., the reservoir of an echo state network (ESN) [Jaeger and Haas, 2004]. In this high-dimensional space, the skeleton sequences are encoded into action echo states (we call "echo-state representations", ESRs) which contain actions' history information due to the short-term memory property of ESNs. In the decoding part, we use a layer of multiscale convolution and a max-over-time pooling to decode the ESRs (Figure 3). Furthermore, we propose two multi-step fusion strategies to integrate the spatial information over the five parts of human physical structure (Figure 4). Finally, we learn the label distribution using softmax. With one training-free recurrent layer and only one layer of convolution, ConvESN is a very efficient end-to-end model, and achieve state-of-the-art performance on four skeleton benchmark data sets.

The merits of using Echo State Networks over other RNN/LSTM networks are twofold. First, the weights of the reservoir are randomly initialized and fixed; no training is required. The recurrent layer can be regarded as a high-dimensional (usually 100-1000D) nonlinear temporal kernel and automatically provides abundant representations of the input time series. Our results compare favorably with existing RNN/LSTM models. Second, due to the sparse connectivity of neurons in a reservoir, a lot of loosely coupled oscillators are created, and information will persists in one part of the net without being propagated to other parts too quickly. These oscillators occur naturally at multiple time scales, so that ESRs contain actions' history information because of this multiscale memory. We give an illustrative example in Figure 1(b) with a visualization of the ESR activations over time. We can see that there are two very similar poses, 15 and 35, but the corresponding ESRs are very different. The ESRs distinguish the similar poses in the same action according to their different preceding pose sequences. Pose 15 is preceded by quite different poses (i.e., from 10 to 14) than those preceding pose 35 (i.e., from 30 to 34), part of a static pose. This can be attributed to the short-term memory property. Therefore, action echo states are very suitable and efficient in representing and capturing the temporal dynamics in 3D skeleton time series. However, only using linear combinations of action echo states and simple regression, as original ESNs used, are unable to decode complex ESRs (we've tried). Hence we use the CNN to understand action echo states.

Our main contributions can be summarized as follows. 1) We propose an efficient and effective end-to-end action recognition model learning from action echo states, which are well-suited to represent 3D skeleton-based action sequences. 2) Integrating the efficiency of the ESN in dealing with time series and the CNN for multiscale feature extraction into a unified framework, ConvESN bridges the gap between the reservoir computing paradigm [Lukoševičius and Jaeger, 2009] and the deep learning paradigm. 3) ConvESN can be easily generalized by plugging in other powerful deep learning decoders. 4) We achieve state-of-the-art performance on 3D-skeleton-based action recognition benchmarks, including the MSR-Action 3D dataset [Li *et al.*, 2010], the Motion Capture Dataset HDM05 [Müller *et al.*, 2007], the Florence3D-Action [Seidenari *et al.*, 2013] and the UTKinect-Action dataset [Xia *et al.*, 2012].

## 2 Proposed Method

To explain our model, we first briefly review the Echo State Network paradigm. Then we propose our ConvESN for 3D-skeleton-based action recognition.

### 2.1 Review of Echo State Networks

An Echo State Network (ESN) consists of three basic components: an input layer, a large recurrent hidden layer (called the *reservoir*) and an output layer. An ESN's input weights and reservoir weights are randomly initialized and fixed during all the stages. In particular, the reservoir contains very sparse connections, which encourages multiple oscillatory dynamics. The only adaptable parameters are the output weights, which usually can be obtained by linear regression. To more clearly understand the difference between ESN with other RNN/LSTM, their general architectures are illustrated in Figure 2.

Given $\mathbf{u} = (\mathbf{u}(0), \mathbf{u}(1), \ldots, \mathbf{u}(T-1))$ a $K$-dimensional input series, $N$-dimensional initial state $\mathbf{x}(0) \in \mathbb{R}^N$ in the reservoir and the $L$-dimensional output series $\mathbf{y} = (\mathbf{y}(0), \mathbf{y}(1), \ldots, \mathbf{y}(T-1))$, the update equations of entire system are as follows:

$$\mathbf{x}(t+1) = f(\mathbf{W}^{res}\mathbf{x}(t) + \mathbf{W}^{in}\mathbf{u}(t+1)) \quad (1)$$

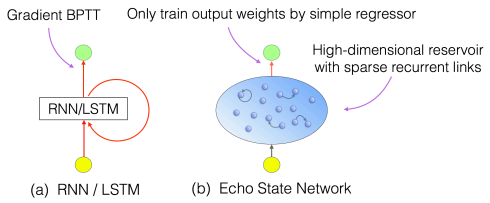



Figure 2: Comparison between RNN/LSTM and ESN. The red lines are adaptable, and the black ones are fixed.

$$\mathbf{y}(t+1) = f^{out}(\mathbf{W}^{out}\mathbf{x}(t+1)) \tag{2}$$

where $\mathbf{W}^{in}$, $\mathbf{W}^{res}$, $\mathbf{W}^{out}$ denote the connection weights from the input layer to the reservoir layer, the reservoir to itself and the reservoir to the output layer, respectively. $\mathbf{W}^{in}$, $\mathbf{W}^{res}$ are initialized randomly and fixed. Only $\mathbf{W}^{out}$ is adaptable. $f$ is the activation function in reservoir (usually $tanh(\cdot)$) and $f^{out}$ is the activation function in output layer (usually $identity(\cdot)$).

An ESN has two core properties for dynamical system identification.

1) *Temporal Kernel*: Input time series drive the large reservoir and produce echo responses in a high-dimensional state space, which enables reservoir to play a similar role as the kernel in kernel-based methods. That is, a reservoir can be regarded as a temporal kernel and the echo states are non-linear high-dimensional representations of the input time series.

2) *Echo-State Property (ESP)[Jaeger and Haas, 2004]*: The ESP means that inputs with more similar short-term history will evoke closer echo states, which ensure the dynamical stability of reservoir. ESP also provides the ESN an important capability called "fading memory" or "short-term memory". With this short-term memory, the input history information from some time past will not easily fade away. In practice, the ESP is guaranteed by keeping the spectral radius of the recurrent weight matrix below 1.

Therefore, if skeleton sequences are regarded as multivariate time series generated from an implicit dynamical system, we can take advantage of ESNs in representing their dynamics.

**Hyper-parameters and Initializations** There are three hyperparameters used for a ESN initialization, including $IS$-Input Scaling, $Sr$-Spectral Radius and $\alpha$-Sparsity.

1) $IS$ is used for the initialization of the matrix $\mathbf{W}^{in}$: the elements of $\mathbf{W}^{in}$ obey the uniform distribution of -$IS$ to $IS$. We set $IS$ to 0.1.

2) $Sr$ is the spectral radius of $\mathbf{W}^{res}$, given by

$$\mathbf{W}^{res} = Sr \cdot \frac{\mathbf{W}}{\lambda_{max}(\mathbf{W})} \tag{3}$$

where $\lambda_{max}(\mathbf{W})$ is the largest eigenvalue of matrix $\mathbf{W}$ and elements of $\mathbf{W}$ are generated randomly in [-0.5, 0.5]. To satisfy the Echo State Property, $Sr$ should be less than one; we use $Sr$=0.99.

3) $\alpha$ denotes the proportion of the non-zero elements in $\mathbf{W}^{res}$. We set $\alpha$ to 0.01.

## 2.2 Convolutional Echo State Network

Combining the benefits of Echo State Networks (temporal kernel, ESP) and Convolutional Neural Networks (multiscale feature learning, temporal shift-invariance), we propose an end-to-end deep neural network we call a Convolutional Echo State Network (ConvESN) for skeleton-based action recognition. The general structure is illustrated in Figure 1(a).

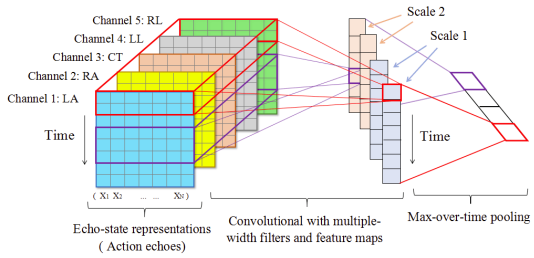



Figure 3: Multi-channel multiscale convolutional process (Here is an example with 5 channels, 2 filters and 2 time scales).

In the first stage, we divide the human skeleton data into five parts: left arm (LA), right arm (RA), left leg (LL), right leg (RL) and center trunk (CT). A simple action may be performed by one segment (e.g., kicking forward only depends on one leg) while complex actions require the coordination of several parts (e.g., running needs the cooperation of arms and legs).

In the second stage, we input trajectories of skeleton joints of each part to five separated reservoirs over time and obtain five parts of echo-state representations. The definition of the echo-state representation is given by Def.1.

**Definition 1 (Echo-State Representation)** *Assume an $N$-neuron reservoir satisfies the ESP, given a $K$-dimensional input denoted as $\boldsymbol{u}(t)$ at time step $t$, then according to Eq.1 (update equation), we have a response of $N$-dimensional echo-state vector $\boldsymbol{x}(t)$ at time step $t$, where $\boldsymbol{x}(t) = (x_1(t), x_2(t), \ldots, x_N(t))$. For a $T$-length input series $\boldsymbol{u}$, we call a matrix $\boldsymbol{X}$ the echo-state representation of $\boldsymbol{u}$, if $\boldsymbol{X}$ satisfies the following equation:*

$$\boldsymbol{X} = \mathscr{F}(\boldsymbol{u}) = \mathscr{F}((\boldsymbol{u}(0), \boldsymbol{u}(1), \ldots, \boldsymbol{u}(T-1))^T) \tag{4}$$

$$= (\boldsymbol{x}(0), \boldsymbol{x}(1), \ldots, \boldsymbol{x}(T-1))^T \tag{5}$$

$$= \begin{pmatrix} x_1(0) & x_2(0) & \ldots & x_N(0) \\ x_1(1) & x_2(1) & \ldots & x_N(1) \\ \vdots & \vdots & \ldots & \vdots \\ x_1(T-1) & x_2(T-1) & \ldots & x_N(T-1) \end{pmatrix} \tag{6}$$

*where $\mathscr{F}$ denotes the reservoir update operator (Eqn. 1). $\boldsymbol{x}(t) \in \mathbb{R}^N$ denotes $t$-th row of $\boldsymbol{X}$, $t \in [0, T-1]$. In addition, we use the notation $\boldsymbol{x}_j$ to denote $j$-th column of $\boldsymbol{X}$, $j \in 1, \ldots, N$.*

**Convolution and Pooling** The multi-channel multiscale convolutional procedure (an example with 5 channels, 2 filters and 2 time scales) is shown in Figure 3. Specifically, the convolution layer uses multiple filter widths and feature maps to extract multiscale features from the echo-state representations and maintains multiscale temporal invariance by max-over-time pooling layers.

Let $\mathbf{X}^i$ be the echo-state representation from the $i$-th channel (here, $i \in 1, 2, \ldots, 5$), $\mathbf{x}^i(t)$ denote its echo-state vector

at time step $t$ (as $t$-th row), then we could describe $T$-length echo-state representation of $i$-th channel by

$$\mathbf{z}^i_{0:T-1} = \mathbf{x}^i(0) \oplus \mathbf{x}^i(1) \oplus \cdots \oplus \mathbf{x}^i(T-1) \tag{7}$$

where $\oplus$ is the concatenation operator and $\mathbf{z}^i_{0:T-1}$ is matrix with size $T \times N^i$, $N^i$ is the size of reservoir of $i$-th channel. In a similar way, $\mathbf{z}^i_{t:t+k-1}$ denotes $k$-length echo-state representation from $t$ to $t+k-1$.

Let $w_{k,j} \in \mathbb{R}^{k \times N}$ denote the $j$th filter with $k$-width. Given $\mathbf{X}^i \in \mathbb{R}^{T \times N}$ and a stride of 1, we have temporal windows as $\mathbf{z}^i_{0:k-1}, \mathbf{z}^i_{1:k}, \ldots, \mathbf{z}^i_{T-k+1:T}$. Then the convolution result with filter $w_{k,j}$ is given by

$$\mathbf{c}_{k,j} = (c_0, c_1, \ldots, c_{T-k+1})^T \tag{8}$$

where $c_m, m = 1, 2, \ldots, T-k+1$ is the convolution result of $m$-th sliding window and is defined as

$$c_m = f(\sum_i \alpha^i_{k,j} \cdot (w_{k,j} * \mathbf{z}^i_{m:m+k-1}) + b) \tag{9}$$

where $f$ is the nonlinear activation function (e.g., $tanh(\cdot)$), $\alpha^i_{k,j}$ is the connection weight from the $i$-th channel reservoir to the $j$th filter with $k$-width and $*$ denotes the dot-product operation.

In the pooling layer, we use the max-over-time pooling operation proposed in [Collobert *et al.*, 2011]. In this way, the feature extracted by the filter $w_{k,j}$ is defined as $d_{k,j} = \max\{\mathbf{c}_{k,j}\}$, where $\{\mathbf{c}\}$ denotes the element set of vector $\mathbf{c}$.

The fusion layer is a fully-connected layer, which could contain several fully-connected layers in our multi-step fusion strategies. In this stage, we fuse the relevant pooled features $\{d_{k,j}\}$ into a single vector. We believe the relationship of the features between two arms or two legs is close. Hence, we fuse the information of the two arms and that of two legs, respectively, and combine all the information in the following steps. In section 2.4, we will introduce two multi-step fusion strategies under the ConvESN paradigm.

The final layer is a softmax layer, whose inputs are the fusion features passed through the fully connected layers and output is defined as the conditional distribution $p(C_s|\mathbf{u})$ over skeleton action labels, where $C_s$ denotes $s$-th class of human actions, and $p(C_s|\mathbf{u})$ is output of the softmax function.

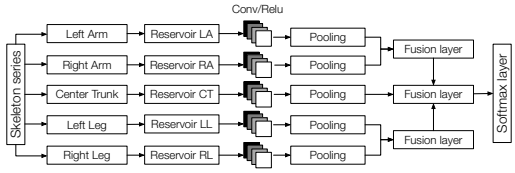
## 2.3 Training

The loss function of ConvESN is to maximize the logarithm likelihood function:

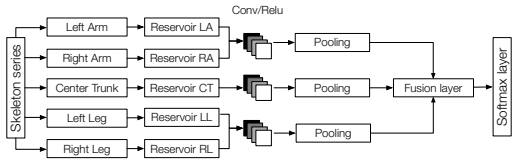
$$\mathcal{L}(\mathbf{u}) = \sum_{n=1}^{N'} \sum_{s=0}^{C-1} \delta(s-r) \ln p(C_s|\mathbf{u}_n) \tag{10}$$

where $\mathbf{u}$ denotes the skeleton training data, $C_s$ is s-th class of human actions, $N'$ denotes the size of train set, $\delta(\cdot)$ is the Kronecker delta and $r$ is the groundtruth label of sample $\mathbf{u}_n$.

In the following experiments, we use the back-propagation algorithm and the gradient optimization method ADAM [Kingma and Ba, 2014] to optimize parameters in ConvESN. Note that the parameters in reservoirs do not need to be learned, but are randomly fixed.




(a) ConvESN-MSSC (multi-step & single-channel)




(b) ConvESN-MSMC (multi-step & multi-channel)

Figure 4: Our models of ConvESN corresponding to different fusion strategies. (a) and (b) are our models with multi-step fusion strategy.

## 2.4 Fusion strategy

To capture the spatial correlation of five skeleton parts, we first fuse the features of two arms and two legs respectively. And then we fuse all the representations. There are two multi-step fusion strategies to integrate the skeleton information.

**ConvESN-MSSC** As shown in Figure 4(a), this model is called ConvESN with Multi-Step Single-Channel fusion (ConvESN-MSSC). In this model, we apply five separate single-channel CNNs to convolve echo-state representations of five reservoirs respectively. After that, we fuse the pooled features corresponding to five parts by three layer-wise fusion layers.

**ConvESN-MSMC** ConvESN with Multi-Step Multi-Channel fusion (ConvESN-MSMC) is illustrated in Figure 4(b), we regard the reservoir LA and the RA as two-channel inputs to the first filter groups, CT as the single-channel input of the second filter group, and LL and RL as the two-channel inputs of the third filter groups. After pooling, all the features are combined in a fully-connected layer.

# 3 Experimental Results

## 3.1 Evaluation Datasets and Settings

We evaluate our proposed ConvESN on four skeleton-based action recognition benchmarks:

**MSR-Action 3D (MSRA3D).** [Li *et al.*, 2010] Captured by Microsoft Kinect-like depth sensor at 15FPS, it provides skeleton (20 joints) 3D coordinates data for 20 actions performed 2-3 times by 10 subjects, which gives a total of 567 sequences with 23797 frames. The MSR-Action 3D task is very challenging due to its data corruption. In [Zhang *et al.*, 2016], 10 sequences with excessive noise were removed. However, we use the complete original datasets in order to evaluate its anti-noise capacity.

**HDM05.** [Müller *et al.*, 2007] An optical marker-based motion capture dataset sampled at 120Hz. It contains 3D coordinates of 31 joints for 130 actions performed by 5 non-professional actors. Following the same protocol in [Cho and Chen, 2014], we classify some samples of 130 actions into

the same category, in which samples represent the same action (e.g., jogging starting from air and from floor). After this processing, we obtain 65 action categories.

**Florence3D-Action.** [Seidenari *et al.*, 2013] Captured by using Kinect sensor, it contains 3D locations of 15 joints for 9 actions performed 2-3 times by 10 different subjects, giving a total of 215 sequences. Its challenge is the high intra-class variations (left-hand actors and right-hand ones) and the presence of actions like drink from a bottle and answer phone which are quite similar to each other.

**UTKinect-Action (UTKA).** [Xia *et al.*, 2012] Provides the 20-joints skeleton coordinates for 10 actions performed 2 times by 10 subjects, captured at 15FPS. There are 195 sequences. The frame length of them ranges from 5 to 170 with an average value of $30.5 \pm 20$. Again, the challenge is intra-class variations as in Florence3D-Action, as well as multiple views.

**Implementation Details.** The $IS$ of reservoir is 0.1, the $Sr$ is 0.99, and the reservoir sizes range 100 to 300. For the multiscale convolutional layer, we choose the width of sliding windows as 2, 3, & 4 and the number of filters under each width ranges from 16 to 128. The size of the final fusion layer is set as 144. It's worth noting that our ConvESN only consists of a shallow architecture with a convolution layer and a max-over-time pooling layer as Figure 4.

Preprocessing is an important step for action recognition, because include that raw skeleton joints are not in an unified coordinate system, different joints trajectories have different levels of smoothness, and samples have various length, etc. Our preprocessing details are as follows. We normalize the coordinate system by setting the origin to the average of the hip center, left, and right joints. We then apply the popular Savitzky-Golay smoothing filter [Steinier *et al.*, 1972] to smooth the joint trajectories. To reduce computational cost on the HDM05 dataset we sample every 4 frames. We do not down-sample the other datasets. Finally, for variable length trajectories, we pad them with zeros up to a given max-length value. The machine setup is on an Intel Core i5-6500, 3.20-GHz CPU 32-GB RAM and a GeForce GTX 980-Ti 6G.

## 3.2 Experimental Results and Analysis

**MSR-Action 3D:** We use a standard validation protocol used by [Li *et al.*, 2010] on the MSR-Action 3D dataset. In this protocol, we split the whole dataset into three overlapping subsets (AS1, AS2, AS3) of 8 classes for each one. Within each set, we adopt cross-subject validation: the subjects 1, 3, 5, 7, 9 are used for training and 2, 4, 6, 8, 10 are used for testing. The results (average accuracy of AS1, AS2 and AS3) are reported in Table 1.

As seen from Table 1, ConvESN-MSMC achieves the best average accuracy with 97.88%. ConvESN-MSSC also performs well with 97.56%. Without removing the 10 excessively noisy sequences, they both outperform the existing approaches listed in Table 1. The best of other methods is the work of Zhang et al. [Zhang *et al.*, 2016] with 96.97%, which was with the 10 noisy sequences removed.

**HDM05:** Following the protocol used in previous work [Du *et al.*, 2015], we perform 10-fold cross validation on this dataset. As shown in Table 2, the best two models with average accuracy are ConvESN-MSMC and Deep LSTM [Zhu *et al.*, 2016] with 97.25%.

Table 1: Recognition accuracy (%) on MSR-Action 3D dataset (Cross-subject Test).

| Methods | Ave.(%) |
|---|---|
| Covariance [Hussein *et al.*, 2013] | 88.10 |
| HOD [Gowayyed *et al.*, 2013] | 91.26 |
| Skeletons Lie group [Vemulapalli *et al.*, 2014] | 92.46 |
| DHMM+SL [Lo Presti *et al.*, 2015] | 92.91 |
| Random Forest+depth [Zhu *et al.*, 2013] | 94.30 |
| Hierarchical LSTM [Du *et al.*, 2015] | 94.49 |
| SGWT+SVM [Kerola *et al.*, 2014] | 94.77 |
| DMMs+Fisher vectors [Chen and Liu, 2016] | 95.97 |
| Gram matrices Representations [Zhang *et al.*, 2016] | 96.97 |
| ConvESN-MSSC | 97.56 |
| ConvESN-MSMC | **97.88** |

Table 2: Recognition accuracy on HDM05 dataset (10-fold cross validation).

| Methods | Ave.(%) |
|---|---|
| DNN [Cho and Chen, 2014] | 95.59 |
| Hierarchical LSTM [Du *et al.*, 2015] | 96.92 |
| Deep LSTM [Zhu *et al.*, 2016] | **97.25** |
| ConvESN-MSSC | 97.08 |
| ConvESN-MSMC | **97.25** |

Table 3: Recognition accuracy on Florence3D-Action dataset (10-fold cross validation).

| Methods | Ave.(%) |
|---|---|
| Multi-Part Bag-of-Poses [Seidenari *et al.*, 2013] | 82.00 |
| Skeletons Lie group [Vemulapalli *et al.*, 2014] | 90.88 |
| ConvESN-MSSC | 91.17 |
| ConvESN-MSMC | **91.72** |

The confusion matrices of ConvESN with MSSC and MSMC on the HDM05 task are shown in Figure 5. As shown in Figure 5, the proposed models (ConvESN-MSSC and ConvESN-MSMC) can recognize most of HDM05 actions very well. However, there exist several easily-misclassified action pairs, including class 4 "depositFloorR" vs. class 9 "grabFloorR", class 5 "depositHighR" vs. class 10 "grabHighR", class 6 "depositLowR" vs. class 11 "grabLowR". These action pairs have very similar trajectories because they all contain two key actions: "deposit" and "grab". We cannot distinguish these action pairs well unless we obtains more information from the context of action processes.

**Florence3D-Action:** Table 3 shows that ConvESN-MSSC and ConvESN-MSMC both outperform previous methods.

**UTKinect-Action:** Table 4 shows our models also achieve the best performance of 100%. This accuracy is also reached by the most recent work Zhang et al. [Zhang *et al.*, 2016].

## 3.3 Contrast Baselines

To demonstrate the efficacy of our combined ESN and CNN, we compare it to each component separately: 1) ESN alone with mean-pooling and softmax; 2) CNN alone with the fusion strategy MSSC; 3) CNN alone with MSMC.

Figure 6 shows that muli-step-fusion based ConvESN

Table 4: Recognition accuracy on UTKinect Action dataset (10-fold cross validation).

| Methods | Ave.(%) |
|---|---|
| Random Forest+depth [Zhu *et al.*, 2013] | 87.90 |
| LTBSVM [Slama *et al.*, 2015] | 88.50 |
| HOJ3D+HMM [Xia *et al.*, 2012] | 90.92 |
| DP+KNN [Devanne *et al.*, 2013] | 91.50 |
| Skeletons Lie group [Vemulapalli *et al.*, 2014] | 97.08 |
| Gram matrices Representations [Zhang *et al.*, 2016] | **100.00** |
| ConvESN-MSSC | **100.00** |
| ConvESN-MSMC | **100.00** |

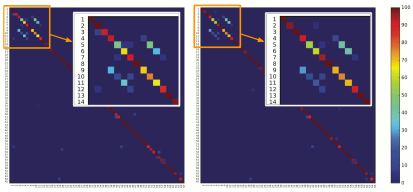


Figure 5: Confusion matrices on HDM05: ConvESN-MSSC (Left) -97.08%; ConvESN-MSMC (Right) -97.25%

(ConvESN-MSMC and ConvESN-ESSC) are both better than the contrast baselines CNN alone without action echoes (CNN-MSMC and CNN-MSSC). In additional, we also demonstrate that the original ESN with soft-max is a poor classifier with the CNN decoder. Thus, the combination of an ESN to represent the temporal dynamics of the signal, with the decoding capacity of the shallow CNN, gives the best of both worlds.

### 3.4 Computational Efficiency Discussions

Compared with the standard ESN, our ConvESN does't add significant computational cost, because it still processes the signal in linear time and the feed-forward CNN is a fixed cost on top of that.

Compared with other, more complex recurrent models such as Hierarchical or Deep LSTMs, the ESN automatically produces ESRs in a high-dimensional echo state space without training, and has been effective in the field of (chaotic) time series forecasting [Jaeger and Haas, 2004]. LSTM networks typically require extensive training via back-propagation through time (BPTT). For the HDM05, ConvESN-MSMC took 157s to produce all of the 200-D action echoes from 2339 sequences. Training the convnet took 23s per epoch. During testing, it runs at 780 sequences per second. Hierarchical LSTM [Du *et al.*, 2015] reported their training time at about real time. Our model trains about 7 times faster in equivalent circumstances, not counting the one-time generation of the echo states. On the other hand, the CNN in our framework retains the merits of weight sharing and structural conciseness, which largely reduces the size of the parameter space.

## 4 Conclusion

In this paper, we study the problem of 3D skeleton-based human action recognition and introduce a novel end-to-end neural network, the Convolutional Echo State Network (ConvESN). ConvESN incorporates modeling dynamics, multiscale temporal features extraction and actions classification in an unified framework. Experiments on benchmarks achieve state-of-the-art accuracy. These results verify that, with abundant action echo states, the dynamics and multiscale features of action recognition can be efficiently learned by an additional convolutional layer and a pooling layer of the CNN. ConvESN is a novel framework that bridges the reservoir computing and deep learning research fields, balancing high performance and model complexity. The action echo states are well-suited to represent 3D skeleton-based action sequences and have the potential to be combined with deeper learning models for more complex sequence recognition.

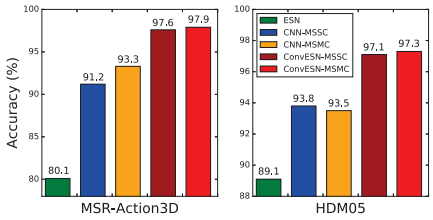



Figure 6: Results (Accuracy(%) of three contrast baselines from ConvESN on MSR-Action 3D and HDM05.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 61502174, 61402181), the Natural Science Foundation of Guangdong Province (Grant No. S2012010009961, 2015A030313215), the Science and Technology Planning Project of Guangdong Province (Grant No. 2016A040403046), the Guangzhou Science and Technology Planning Project(Grant No. 2014J4100006, 201704030051), and the Fundamental Research Funds for the Central Universities (Grant No. D2153950). It was also supported by the National Science Foundation (USA) grant SMA 1041755 to the Temporal Dynamics of Learning Center, an NSF Science of Learning Center.

## References

[Chen and Liu, 2016] Chen Chen and Mengyuan Liu. 3d action recognition using multi-temporal depth motion maps and fisher vector. In *International Joint Conference on Artificial Intelligence*, pages 3331–3337, July 2016.

[Cho and Chen, 2014] Kyunghyun Cho and Xi Chen. Classifying and visualizing motion capture sequences using deep neural networks. In *International Conference on Computer Vision Theory and Applications*, pages 122–130, 2014.

[Collobert *et al.*, 2011] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November 2011.

[Devanne *et al.*, 2013] Maxime Devanne, Hazem Wannous, Stefano Berretti, Pietro Pala, Mohamed Daoudi, and Alberto Del Bimbo. Space-time pose representation for 3d

human action recognition. In *International Conference on Image Analysis and Processing*, pages 456–464. Springer, 2013.

[Du *et al.*, 2015] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *2015 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1110–1118, June 2015.

[Gong *et al.*, 2014] Dian Gong, Gerard Medioni, and Xuemei Zhao. Structured time series analysis for human action segmentation and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(7):1414–1427, July 2014.

[Gowayyed *et al.*, 2013] Mohammad A. Gowayyed, Marwan Torki, Mohamed E. Hussein, and Motaz El-Saban. Histogram of oriented displacements (hod): Describing trajectories of human joints for action recognition. In *International Joint Conference on Artificial Intelligence*, pages 1351–1357, 2013.

[Hussein *et al.*, 2013] Mohamed E Hussein, Marwan Torki, Mohammad Abdelaziz Gowayyed, and Motaz El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *International Joint Conference on Artificial Intelligence*, volume 13, pages 2466–2472, 2013.

[Jaeger and Haas, 2004] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, 2004.

[Kerola *et al.*, 2014] Tommi Kerola, Nakamasa Inoue, and Koichi Shinoda. Spectral graph skeletons for 3d action recognition. In *Asian Conference on Computer Vision*, pages 417–432. Springer, 2014.

[Kingma and Ba, 2014] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.

[Li *et al.*, 2010] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3d points. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 9–14, June 2010.

[Lo Presti *et al.*, 2015] Liliana Lo Presti, Marco La Cascia, Stan Sclaroff, and Octavia Camps. Hankelet-based dynamical systems modeling for 3d action recognition. *Image Vision Comput.*, 44(C):29–43, December 2015.

[Lukoševičius and Jaeger, 2009] Mantas Lukoševičius and Herbert Jaeger. Survey: Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.*, 3(3):127–149, August 2009.

[Müller *et al.*, 2007] Meinard Müller, Tido Röder, Michael Clausen, Bernhard Eberhardt, Björn Krüger, and Andreas Weber. Documentation mocap database hdm05. Technical Report CG-2007-2, Universität Bonn, June 2007.

[Presti and Cascia, 2016] Liliana Lo Presti and Marco La Cascia. 3d skeleton-based human action classification: A survey. *Pattern Recognition*, 53:130 – 147, 2016.

[Seidenari *et al.*, 2013] Lorenzo Seidenari, Vincenzo Varano, Stefano Berretti, Alberto Del Bimbo, and Pietro Pala. Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses. In *The IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 479–485, June 2013.

[Shotton *et al.*, 2013] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1297–1304, 2013.

[Slama *et al.*, 2015] Rim Slama, Hazem Wannous, Mohamed Daoudi, and Anuj Srivastava. Accurate 3d action recognition using learning on the grassmann manifold. *Pattern Recogn.*, 48(2):556–567, February 2015.

[Song *et al.*, 2016] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. *arXiv:1611.06067*, 2016.

[Steinier *et al.*, 1972] Jean. Steinier, Yves. Termonia, Jules. Deltour, and Anal. Chem. Smoothing and differentiation of data by simplified least square procedure. *Analytical Chemistry*, 44(11):1906–9, 1972.

[Vemulapalli *et al.*, 2014] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–595, 2014.

[Wu and Shao, 2014] Di Wu and Ling Shao. Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–731, 2014.

[Xia *et al.*, 2012] Lu Xia, Chia Chih Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–27, June 2012.

[Zhang *et al.*, 2016] Xikang Zhang, Yin Wang, Mengran Gou, Mario Sznaier, and Octavia Camps. Efficient temporal sequence comparison and classification using gram matrix embeddings on a riemannian manifold. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 4498–4507, June 2016.

[Zhu *et al.*, 2013] Yu Zhu, Wenbin Chen, and Guodong Guo. Fusing spatiotemporal features and joints for 3d action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 486–491, June 2013.

[Zhu *et al.*, 2016] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *The AAAI Conference on Artificial Intelligence*, pages 3697–3703, 2016.