Modeling Temporal Dynamics and Spatial Configurations of Actions Using Two-Stream Recurrent Neural Networks

Hongsong Wang[1,3] Liang Wang[1,2,3]
[1]Center for Research on Intelligent Perception and Computing (CRIPAC),
National Laboratory of Pattern Recognition (NLPR)
[2]Center for Excellence in Brain Science and Intelligence Technology (CEBSIT),
Institute of Automation, Chinese Academy of Sciences (CASIA)
[3]University of Chinese Academy of Sciences (UCAS)
{hongsong.wang, wangliang}@nlpr.ia.ac.cn

Abstract

Recently, skeleton based action recognition gains more popularity due to cost-effective depth sensors coupled with real-time skeleton estimation algorithms. Traditional approaches based on handcrafted features are limited to represent the complexity of motion patterns. Recent methods that use Recurrent Neural Networks (RNN) to handle raw skeletons only focus on the contextual dependency in the temporal domain and neglect the spatial configurations of articulated skeletons. In this paper, we propose a novel two-stream RNN architecture to model both temporal dynamics and spatial configurations for skeleton based action recognition. We explore two different structures for the temporal stream: stacked RNN and hierarchical RNN. Hierarchical RNN is designed according to human body kinematics. We also propose two effective methods to model the spatial structure by converting the spatial graph into a sequence of joints. To improve generalization of our model, we further exploit 3D transformation based data augmentation techniques including rotation and scaling transformation to transform the 3D coordinates of skeletons during training. Experiments on 3D action recognition benchmark datasets show that our method brings a considerable improvement for a variety of actions, i.e., generic actions, interaction activities and gestures.

1. Introduction

Human action recognition [2] has become an active area in computer vision and there are many important research problems, such as event recognition [23], group based activities recognition [27], human object interactions [15] and activities in egocentric videos [29, 11]. Most approaches have been proposed to recognize actions in RGB videos recorded by 2D cameras. However, it still remains a challenging problem for three reasons. First, it is hard to well extract useful information from the high dimensional and low quality input data. Second, the RGB video is highly sensitive to some factors like illumination changes, occlusion and background clutter. Third, the identification of actions is related to high-level visual clues such as human poses and objects, which are very difficult to obtain from RGB videos directly.

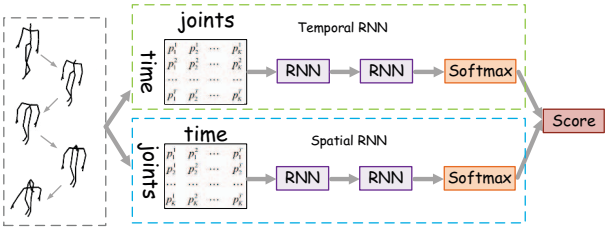


Figure 1. A two-stream RNN architecture for skeleton based action recognition. Here *Softmax* denotes a fully connected layer with a softmax activation function.

Humans can recognize actions with a few spots describing motions of the main joints of skeletons [24], and experiments show that a large set of actions can be recognized solely from skeletons [25]. In contrast to RGB based action recognition, skeleton based action recognition can avoid the awful task of feature extraction from videos and explicitly model the dynamics of actions. There are three ways to obtain skeletons: motion capture systems, RGB images and depth maps. Sophisticated motion capture systems are very expensive and require the user to wear a motion capture suit with markers. Extracting reliable skeletons from monocular RGB images or videos, i.e., pose estimation, is still an unsolved problem. Fortunately, with the recent advent of affordable depth sensors, it is much easier and cheaper to obtain 3D skeletons from depth maps. For example, Shot-

1063-6919/17 $31.00 © 2017 IEEE
DOI 10.1109/CVPR.2017.387

ton et al. [38] propose a method to quickly and accurately predict 3D positions of body joints from a single depth image. These advances excite considerable interest for skeleton based action recognition and various algorithms have been proposed recently.

Traditional skeleton based action recognition approaches are mainly divided into two categories: joint based approaches and body part based approaches. Joint based approaches consider the human skeleton as a set of points and use various positions based features such as joint positions [20, 31] and pairwise relative joint positions [46, 51] to characterize actions. While body part based approaches regard the human skeleton as a connected set of segments, and then focus on individual or connected pairs of body parts [50] and joint angles [33]. Based on handcrafted low-level features, both approaches employ relatively simple time series models, e.g., hidden Markov model [47, 49], to recognize actions. However, human-engineered features are limited to represent the complexity of the intrinsic characteristics of actions and the subsequent time series models do not unleash the full potential of the sequential data.

Inspired by the great success of deep learning for RGB based action recognition [39, 26, 21], there is a growing trend of using deep neural networks for skeleton based action recognition. Different structures of Recurrent Neural Networks (RNN), e.g., hierarchical RNN [7], RNN with regularizations [55], differential RNN [43] and part-aware Long Short-Term Memory (LSTM) [37], have been used to learn motion representations from raw skeletons. However, considering an action is a continuous evolution of articulated rigid segments connected by joints [54], these RNN-based methods only model the contextual information in the temporal domain by concatenating skeletons for each frame. In fact, different actions are performed with different spatial configurations of joints of skeletons. The dependency in the spatial domain also reflects the characteristics of actions and should not be neglected for skeleton based action recognition.

To this end, we introduce a novel two-stream RNN architecture which incorporates both spatial and temporal networks for skeleton based action recognition. Figure 1 shows the pipeline of our method. The temporal stream uses a RNN based model to learn the temporal dynamics from the coordinates of joints at different time steps. We employ two different RNN models, *stacked RNN* and *hierarchical RNN*. Compared with *stacked RNN*, *hierarchical RNN* is designed according to human body kinematics and has fewer parameters. At the same time, the spatial stream learns the spatial dependency of joints. We propose a simple and effective method to model the spatial structure that first casts the spatial graph of articulated skeletons into a sequence of joints, then feeds this resulting sequence into a RNN structure. Different methods are explored to turn the graph structure into a sequence for the purpose of better maintaining the spatial relationships. The two channels are then combined by late fusion and the whole network is end-to-end trainable. Finally, to avoid overfitting and improve generalization, we exploit data augmentation techniques by using 3D transformation, i.e., rotation transformation, scaling transformation and shear transformation to transform the 3D coordinates of skeletons during training.

In summary, the main contributions of this paper are listed as follows. First, we propose a two-stream RNN architecture to utilize both spatial and temporal relations of joints of skeletons. Second, we exploit and compare different architectures of both streams. Third, we propose data augmentation techniques based on 3D transformation and demonstrate the effectiveness for skeleton based action recognition. Finally, our method obtains the state-of-the-art results on three important benchmarks for a variety of actions, i.e., generic actions (NTU RGB+D), interaction activities (SBU Interaction) and gestures (ChaLearn).

2. Related work

In this section, we briefly review action recognition approaches related to ours. The two aspects are as follows.

2.1. Action recognition with deep networks

Deep neural networks have made great progress in the area of action recognition. 3D Convolutional Neural Networks (CNN) is proposed and different architectures are studied to take advantage of local spatio-temporal information [26, 21]. To capture complementary information between appearance and motion, a two-stream CNN architecture is developed for RGB based action recognition [39].

Recently, Recurrent Neural Networks (RNN) have been widely used for action recognition. Srivastava et al. [40] use multilayer Long Short Term Memory (LSTM) networks to learn representations of video sequences. Donahue et al. [4] develop an end-to-end trainable Long-term Recurrent Convolutional Networks (LRCN) architecture which can simultaneously learn temporal dynamics and convolutional perceptual representations from RGB videos. Deep Convolutional and Recurrent Neural Networks has also been proposed and applied for activity recognition [34, 19].

Prior to our work, several models have been proposed based on RNN for skeleton based action recognition. Du et al. [7, 6] first design an end-to-end hierarchical RNN architecture for skeleton based action recognition. Zhu et al. [55] propose a fully connected deep LSTM network with regularization terms to learn co-occurrence features of joints. Veeriah et al. [43] present differential RNN that extends LSTM structure by modeling the dynamics of states evolving over time. Shahroudy et al. [37] propose a part-aware extension of LSTM to utilize the physical structure

of the human body. These methods only model the motion dynamics in the temporal doamin and neglect the spatial configurations of articulated skeletons. Recently, Liu et al. [30] extend LSTM to spatial-temporal domain for the purpose of modeling the dependencies between joints. As temporal dynamics and spatial configurations are separate visual pathways [14], we employ a two-stream architecture to model them accordingly.

2.2. Features based on skeletons

Previous skeleton based action recognition methods mainly focus on handcrafted features [1]. To get representations of postures, one straightforward feature is the pairwise joint location difference, which can be simply concatenated [32], or casted into 3D cone bins to build a histogram of 3D joints locations [49] for action recognition.

Joint orientation is another good feature as it is invariant to the human body size. For example, Sempena et al. [36] apply dynamic time warping based on the feature vector built from joint orientation along time series. Bloom et al. [3] use AdaBoost to combine five types of features, i.e., pairwise joint position difference, joint velocity, velocity magnitude, joint angle velocity and 3D joint angle to recognize gaming actions, for real-time action recognition.

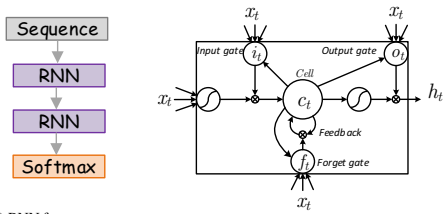
There are some work that groups the joints of skeletons to construct planes from joints and then measures the joint-to-plane distance and motion. Yun et al. [53] capture the geometric relationship between the joint and the plane spanned by three joints. Sung et al. [41] compute the joint's rotation matrix w.r.t. the person's torso, hand position w.r.t. the torso and joint rotation motion as features.

3. Overview of RNN

Different from feedforward neural networks that map from one input vector/matrix to one output vector/matrix, recurrent neural networks (RNN) map an input sequence X to another output sequence Y.

RNN architectures are naturally suitable for the sequence classification, where each input sequence is assigned with a single class. Layers of RNN can be stacked to build a deep RNN by considering the output sequence of the previous layer as the input sequence of the current layer. A typical structure of RNN for sequence classification is shown in Figure 2(a), which contains a stack of RNN layers with a softmax classification layer on top of the last hidden layer.

Due to the vanishing gradient and error blowing up problems [16], the standard RNN cannot store information for long periods of time or access the long range of context. Long short-term memory (LSTM) [17] addresses this problem by using additional gates to determine when the input is significant enough to remember, when it should continue to remember or forget the value, and when it should output the value. The LSTM unit has been shown to be capable of



(a) RNN for sequence classification.

(b) LSTM neuron.

Figure 2. (a) A two-layer stacked RNN for sequence classification. (b) A LSTM block with input, output, and forget gates [17].

storing and accessing information over very long timespans [13]. Figure 2(b) depicts a LSTM unit:

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\ f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\ c_t &= f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\ h_t &= o_t \tanh(c_t) \end{aligned} \quad (1)$$

where i, f, o correspond to the input gate, forget gate and output gate, respectively. All the matrices W are the connection weights and all the variables b are biases.

4. Two-stream RNN

The sequence of skeletons determines the evolution of actions, which has both spatial and temporal structures. The spatial structure displays a spot of the pictorial form of joints while the temporal structure tracks and represents the movement of joints. Accordingly, we devise an end-to-end two-stream architecture based on RNN, which is shown in Figure 1. Here the fusion is performed by combining the softmax class posteriors from the two nets.

4.1. Temporal RNN

We begin with the description of the temporal channel of RNN, which models the temporal dynamics of skeletons. Similar to the previous work [7, 55, 43, 37], it concatenates the 3D coordinates of different joints at each time step and handles the generated sequence with a RNN architecture. We focus on the following two model structures.

Stacked RNN. This structure feeds the RNN network with the concatenated coordinates of all joints at each time step. Here we stack two layers of RNN and find that adding more layers would not considerably improve the performance. As the length of skeleton sequences is relatively long (e.g., 50∼200), we adopt LSTM neurons for all layers. Although simple, *stacked RNN* has been widely used to process and recognize sequences of variable lengths.

Hierarchical RNN. The human skeleton can be divided into five parts, i.e., two arms, two legs and one trunk. We observe that an action is performed by either an independent

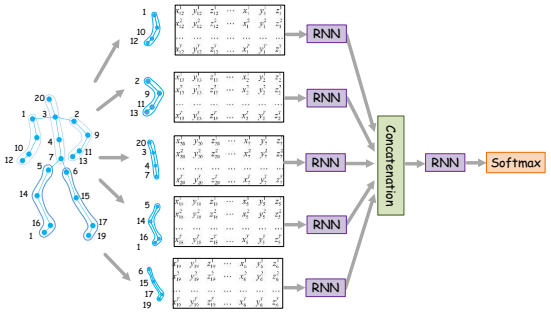


Figure 3. Hierarchical RNN for skeleton based action recognition.

part or a combination of several parts. For example, kicking depends on legs and running involves both legs and arms. Thus, a hierarchical structure of RNN is used to model the motions of different parts as well as the whole body. Figure 3 shows the proposed structure. To be consistent with the *stacked RNN* structure, our *hierarchical RNN* also has two layers vertically.

In the first layer, we use a corresponding RNN to model the temporal movement of each body part based on its concatenated coordinates of joints at each time step. In the second layer, we concatenate the outputs of the RNN of different parts and adopt another RNN to model the movement of the whole body. Compared with the pioneered hierarchical structure in [7], our structure is more succinct and straightforward, and does not use additional fully connected layers before the logistic regression classifier with softmax activation. Compared with the stacked structure, the hierarchical structure has relatively fewer parameters and is less likely to overfit.

4.2. Spatial RNN

Human body can be considered as an articulated system of rigid segments connected by joints. Take the MSR Action3D dataset [28] as an example, the physical structure of the 20 joints is represented by an undirected graph in Figure 4(a). Nodes denote the joints and edges denote the physical connections. When an action takes place, this undirected graph displays some varied patterns of spatial structures. For example, clapping is performed with the joints of the two palms striking together, and bending is acted when the joints of the trunk shape into a curve.

To model the spatial dependency of joints, we cast the graph structure into a sequence of joints and exactly develop a relevant RNN architecture. The input of the RNN architecture at each step corresponds to the vector of coordinates of a certain joint. As a joint has only three coordinates, we select a temporal window centered at the time step and concatenate the coordinates inside this window to represent this joint. This RNN architecture models the spatial relationships of joints in a graph structure and is called spatial RNN. The central problem is how to convert a graph into a sequence. We provide two alternative methods below.

Chain sequence. We assume the joints are arranged in a chain-like sequence with the order of arms, trunk and legs. The trunk is placed in the middle as it connects both arms and legs. For example, the 20 joints graph of the MSR Action3D dataset is arranged in a chain sequence in Figure 4(b). The *chain sequence* maintains the physical connections of joints of each body part (arms, trunk and legs), and the joints are placed in a sequence without duplication. One of the drawbacks is that there is no physical connections at the boundary of joints between hands, trunk and legs. For instance, the joint whose index is 13 is not connected with the joint whose index is 20. But the two joints are adjacent in the generated chain-like sequence.

Traversal sequence. To address the limitation of the *chain sequence*, we propose a graph traversal method to visit the joints in a sequence in the light of the adjacency relations, partly inspired by the tree-structure based traversal method [30].As illustrated in Figure 4(c), we first select the central spine joint as the starting point, and visit the joints of the left arm. While reaching an end point, it goes back. Then we visit the right arm, the upper trunk, etc. After visiting all joints, it finally returns to the starting point. We arrange the graph into a sequence of joints according to the visiting order. The *traversal sequence* guarantees the spatial relationships in a graph by accessing most joints twice in both forward and reverse directions.

Different from the temporal RNN, spatial RNN could recognize actions by a glimpse of one frame (when the size of temporal window equals 1). Here, we do not use a hierarchical structure based on body parts, as the number of joints is limited (e.g., 25 for the NTU RGB+D dataset).

4.3. 3D transformation of skeletons

For skeleton based action recognition, the input data is a sequence 3D coordinates of joints. As neural networks often require a lot of data to improve generalization and prevent overfitting, we exploit several data augmentation techniques based on 3D transformation to make the best use of limited supply of training data. Note that the 3D transformation techniques are only used during training.

Rotation. Based on Euler's rotation theorem, any 3D rotation can be given as a composition of rotations about three axes. The three basic rotation matrices in terms of rotate angles α, β, γ about the x, y, z axis in a counterclockwise direction are represented as below:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \tag{2}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \tag{3}$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

General rotations can be obtained from these three basic rotation matrices using matrix multiplication:

$$R = R_z(\gamma)R_y(\beta)R_x(\alpha) \tag{5}$$

where R is the general rotation matrix in the 3D coordinate system.

For the 3D coordinates of joints, we randomly rotate the input sequence of skeletons within a certain range for the x, y axis, as the rotation plane of the camera is perpendicular to the z axis. The rotation transformation simulates the viewpoint changes of the camera and improves the robustness of our model for cross view experimental settings. We find the recent work [6] also uses the rotation transformation for cross view recognition of actions.

Scaling. Scaling transformation is used to change the size of skeletons. The transformation matrix can be formulated as:

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \tag{6}$$

where s_x, s_y, s_z are scaling factors along with the three axes, respectively.

The scaling transformation can either expand or compress the dimensions of skeletons by using random scaling factors. As different action performers have varied heights and body sizes, the dimensions of their skeletons may be different. Thus the scaling transformation is beneficial for cross subject experimental settings.

Shear. Shear transformation is a linear map that displaces each point in a fixed direction. It slants the shape of the coordinates of joints and changes the angles between them. The transformation matrix can be represented as below:

$$Sh = \begin{bmatrix} 1 & sh_x^y & sh_x^z \\ sh_y^x & 1 & sh_y^z \\ sh_z^x & sh_z^y & 1 \end{bmatrix} \tag{7}$$

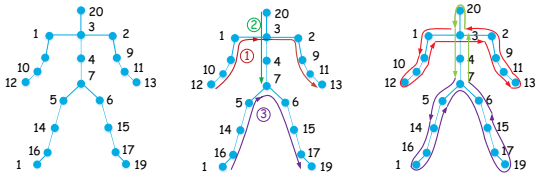
where $sh_x^y, sh_x^z, sh_y^x, sh_y^z, sh_z^x, sh_z^y$ are shear factors.

5. Experiments

The proposed model is evaluated on three datasets: NTU RGB+D dataset [37], SBU Interaction dataset [53], and ChaLearn Gesture Recognition dataset [9, 8].

5.1. Datasets

NTU RGB+D dataset. Currently, this is the largest depth-based action recognition dataset, providing 3D coordinates of 25 joints collected by Kincet v2. It contains more than 56 thousand sequences and 4 million frames, captured in various background conditions. The dataset has 60 different action classes including daily, mutual, and health-related actions. The actions are performed by 40 different human subjects, whose age range is from 10 to 35. Numerous variations in subjects and views, and large amount of samples make it highly suitable for deep learning methods. We follow the cross subject and cross view evaluations [37] and report the classification accuracy in percentage.



(a) Undirected graph of joints. (b) Chain sequence of joints. (c) Traversal sequence of joints.

Figure 4. (a) The physical structure of 20 joints. (b) Convert the joints graph into a sequence. The joints of arms come first, then that of body, finally is that of legs. (c) Use a traversal method to transform the joints graph into a sequence. The order of the sequence is the same as the visiting order of the arrow.

SBU Interaction dataset. This is a complex human activity dataset depicting two person interactions captured with Kinect. Each skeleton has 15 joints. It includes 282 skeleton sequences in 6822 frames. All videos are recorded in the same laboratory environment with 8 activities performed by 7 participants. The dataset is very challenging because the interactions are non-periodic, and have very similar body movements. Following the 5-fold cross validation [53], we split the 21 sets of this dataset into 5 folds and give the average recognition accuracy.

ChaLearn Gesture Recognition dataset. This dataset contains 20 Italian gestures performed by 27 different persons. There are 23 hours of Kinect data, consisting of RGB, depth, foreground segmentation and skeletons. The dataset has 955 videos in total. Each video lasts 1 to 2 minutes and contains 8 to 20 noncontinuous gestures. Here, we only use skeletons for gesture recognition. As done in the literature [9, 12], we report the precision, recall and F1-score measures on the validation set.

5.2. Implementation details

We normalize skeletons by subtracting the central joint, which is the average of 3D coordinates of the hip center, hip left and hip right. The sequences are converted to a fixed length T by sampling and zero padding to allow for batch learning. T should be larger than the length of most sequences to reduce loss of information caused by sampling.

The NTU RGB+D dataset has a variable (one or two) number of persons performing actions. For samples with two persons, we only process one sequence each time, and

average the predicted scores of the two. We set $T = 100$ for this dataset, as most sequences are less than 100 in length. For the SBU Interaction dataset with a pair of skeletons representing interactions of two persons, we concatenate the two 3D coordinates for each joint at each time step and regard it as one sequence of 6D coordinates. We set the normalized sequence length $T = 35$ for this dataset. For the ChaLearn Gesture Recognition dataset, we set $T = 50$.

For the NTU RGB+D dataset, the number of neurons of each layer of *stacked RNN* is 512. For *hierarchical RNN*, the number of neurons of the body part and the whole body are 128 and 512, respectively. For the ChaLearn Gesture Recognition dataset, the networks structures are the same as those of the NTU RGB+D dataset. Compared with the above two datasets, the SBU Interaction dataset has less number of training samples and the sequence length is shorter. So we reduce the number of neurons of *stacked RNN* of the temporal RNN to 256, and set the number of neurons of the body part and the whole body to 64 and 256, respectively. For all the datasets, the structure of the spatial RNN is the same as that of *stacked RNN* of the temporal RNN. We adopt LSTM neurons for all layers due to its excellent performance for sequence recognition.

To demonstrate the effectiveness of the two-stream RNN, we simply adopt *stacked RNN* for the temporal channel and *chain sequence* for the spatial channel. The weight of predicted scores of the temporal RNN is 0.9, and the temporal window size of the spatial RNN is one fourth of the fixed length T, both are determined by cross-validation. The networks are trained using stochastic gradient descent. The learning rate, initiated with 0.02, is reduced by multiplying it by 0.7 every 60 epochs during training. The implementation is based on Theano [42] and Lasagne [1]. One NVIDIA TITAN X GPU is used to run all experiments.

5.3. Experimental results

Comparison between models. The comprehensive results of our two-stream RNN on three datasets are shown in Table 1. We can see that the two-stream RNN consistently outperforms the individual temporal RNN and spatial RNN, which confirms that the spatial and temporal channels are both effective and complementary. In addition, for two activity recognition datasets, the 3D transformation techniques bring significant performance improvement for skeleton based recognition, especially for cross view evaluation. For example, on the NTU RGB+D dataset, the two-stream RNN with 3D transformation outperforms that without 3D transformation by 7.8% for cross view evaluation, much higher than the outperformed value of 2.7% for cross subject evaluation. The explanation is straightforward that rotation transformation randomly generates new skeletons from different views, thus making our two-stream RNN robust to the viewpoint changes.

Generally, the results of the temporal RNN are much better than those of the spatial RNN. This observation is consistent with the fact that most previous RNN based methods adopt the temporal RNN to recognize actions. For the temporal RNN, the hierarchical structure generally performs better than the stacked structure. For example, on the NTU RGB+D dataset, *hierarchical RNN* outperforms *stacked RNN* by an average of 1.6%. For the spatial RNN, the results of the *traversal sequence* are better than those of the *chain sequence* as the traversal method maintains better spatial relationships of the graph structure by visiting most joints twice in both forward and reverse directions.

Comparison between structures. In Section 5.2 we manually define the structures of both *stacked RNN* and *hierarchical RNN*. Here we empirically study the effects of the number of stacked layers and the number of neurons for each layer on the performance. Due to the limited space, we only give results on the NTU RGB+D dataset by cross view protocol in Table 2.

For *stacked RNN*, we observe that two stacked layers (*R512-512*) performs better than one layer (*R512*), and three stacked layers (*R512-512-512*) performs even better than two stacked layers. For the number of neurons of RNN layers, decreasing it to 256 (*R256-256*) reduces the accuracy and increasing it to 1024 (*R1024-1024*) does not necessarily improve the result. As adding more layers and increasing hidden neurons result more parameters and increase the computational complexity of our model, we adopt *R512-512* as the default structure for *stacked RNN*.

For *hierarchical RNN*, using two stacked RNN layers for the part (*P128-128, B512*) and increasing the number of neurons of the part from 128 to 256 (*P256, B512*) improve the performances. The accuracy can be further improved by increasing the number of neurons of both the part and the whole body (*P256, B1024*). To make a fair comparison with the stacked structure (*R512-512*) and reduce the computational cost, we keep the structure with two layers and choose 128 as the number of neurons for the part, which is one fourth of the number of neurons for the whole body.

5.4. Two-stream RNN versus temporal RNN

As previous RNN based methods merely use the temporal RNN, here we aim to show the superiority of our two-stream RNN over the temporal RNN.

We plot and compare the confusion matrices of our two-stream RNN and the temporal RNN on the SBU Interaction dataset in Figure 6. We can observe that there are three pairs of misclassified actions for the temporal RNN, but only one pair for our two-stream RNN. Moreover, for pushing, the samples are 22% misclassified as punching by the temporal RNN, while our two-stream RNN can correctly recognize all the samples.

[1] https://github.com/Lasagne/Lasagne

Table 1. Comprehensive evaluation results of two-stream RNN on three datasets.

Channel	(%)	NTU RGB+D		SBU Interaction	ChaLearn Gesture		
		Cross subject	Cross view		Precision	Recall	F1-score
Temporal RNN	Stacked	66.1	68.9	89.0	89.5	89.6	89.5
	Hierarchical	67.8	70.5	90.2	89.8	89.9	89.7
Spatial RNN	Chain	53.7	58.9	82.2	81.9	82.1	81.9
	Traversal	55.2	60.5	86.6	84.0	84.2	84.0
Two-stream RNN	No transform	68.6	71.7	91.9	91.3	91.3	91.3
	3D transform	71.3	79.5	94.8	91.7	91.8	91.7

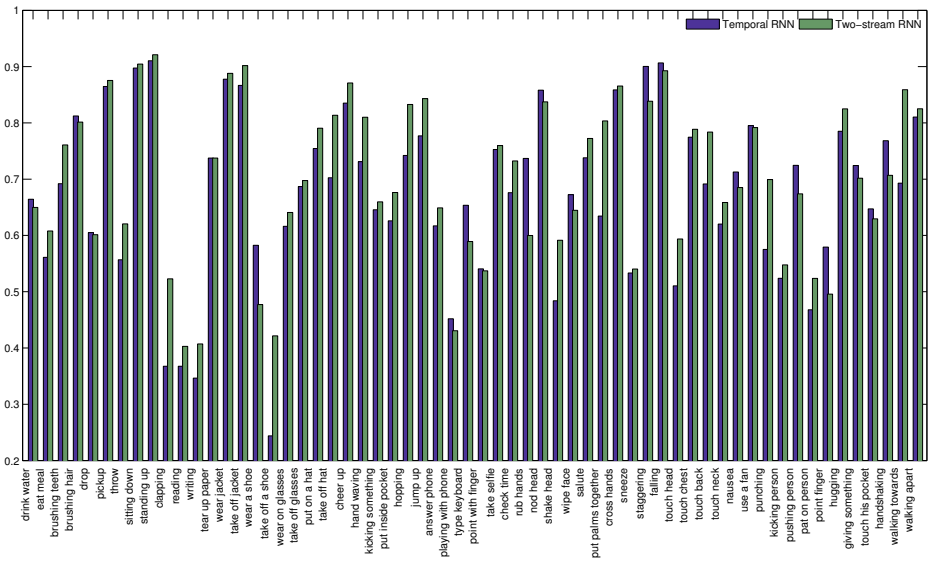


Figure 5. Accuracy for each action on the NTU RGB+D dataset.

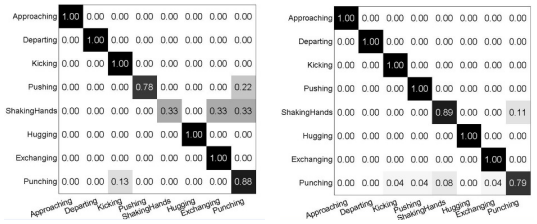
Table 2. Empirical study of networks structures. For *stacked RNN*, *R512-512* denotes two stacked layers of RNN with 512 hidden neurons. Similarly, *R1024* denotes one RNN layer with 1024 hidden neurons. For *hierarchical RNN*, *P128-128, B512* denotes two stacked RNN layers with 128 hidden neurons for the body part and one RNN layer with 512 hidden neurons for the whole body. And so on for the other symbols. The default structures of *stacked RNN* and *hierarchical RNN* are *R512-512* and *P128, B512*, respectively.

Stacked RNN		Hierarchical RNN	
R512-512	68.9	P128, B512	70.5
R512-512-512	69.2	P128-128, B512	71.4
R512	68.6	P256, B512	71.4
R1024-1024	68.9	P128, B1024	70.6
R256-256	68.2	P256, B1024	72.2

We also depict the accuracy of each action. Figure 5 shows the results of cross subject evaluation on the NTU RGB+D dataset. For most actions, the accuracy of our two-stream RNN is higher than that of the temporal RNN. For example, for brushing teeth, shaking head, and walking towards, the accuracy of the two-stream RNN is more than 8% higher than that of the temporal RNN.

5.5. Parameter sensitivity

In this section, we evaluate the impact of parameters on the performance. Our two-stream RNN has two parameters, i.e., the size of temporal window of the spatial channel, and the weight of the temporal channel, denoted by λ and τ, respectively. Figure 7 shows the evaluation results on the SBU Interaction dataset. It should be noted that similar results are observed for other datasets.

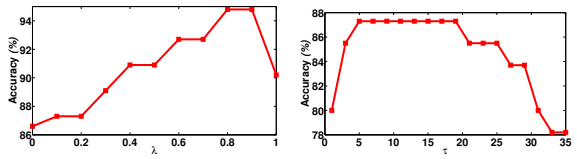


(a) Temporal RNN (a) Two-stream RNN

Figure 6. Comparison of confusion matrices on the SBU Interaction dataset.

Figure 7(a) shows the accuracy of the two-stream RNN w.r.t. the parameter λ, $\lambda \in \{0, 0.1, \cdots, 0.9, 1\}$. We can see the best performance is reached when λ=0.8 or λ=0.9. When $\lambda < 0.8$, the accuracy decreases with a smaller value of λ. The best result is much higher than the two extreme points where $\lambda \in \{0, 1\}$, which correspond to the spatial and temporal RNN, respectively.

We choose $\tau \in \{1, 3, 5, \cdots, T\}$ and plot the accuracy



(a) Weight of temporal RNN (b) Temporal window size

Figure 7. Parameter sensitivity analysis on the SBU Interaction dataset. Here $0 \leq \lambda \leq 1$ and $1 \leq \tau \leq T$, where $T = 35$ is the sequence length after preprocessing.

of the spatial RNN in Figure 7(b). We find that when $5 \leq \tau \leq 17$, i.e., $T/7 \leq \tau \leq T/2$, the temporal RNN obtains the best result. The performance drops when τ is not in this range. We conclude that our result is not sensitive to τ for a wide range.

5.6. Comparison with the state-of-the-art

We compare our two-stream RNN method with the recent methods in the literature. Table 3 shows the results on the NTU RGB+D dataset. We first compare our method with three traditional methods, i.e., 3D skeletons representation in a Lie group [44], Fisher vector encoding of *skeletal quads* [10] and *FTP dynamic* [18]. We observe that our performances are significantly higher, which shows the superiority of deep learning methods over the methods based on handcrafted features. Then our method is compared with other deep learning methods based on RNN. Our results are much better than the reported results of *HBRNN* [7] and *Part-aware LSTM* [37], both of which only model temporal dynamics of actions. Moreover, our method outperforms the newest spatio-temporal LSTM with trust gates [30] by 2.1% and 1.8% for both cross subject evaluation and cross view evaluation, respectively.

The results on the SBU Interaction dataset are shown in Table 4. Our result is 7.9% higher than the best result based on handcrafted features (Joint Feature [22]). In addition, our approach is superior than recent RNN based approaches by outperforming the existing best result by 1.5%. This experiment demonstrates our two-stream RNN model can recognize interactions performed by two persons very well.

The results on the Chalearn Gesture Recognition dataset are summarized in Table 5. Here our two-stream RNN is only compared with the methods solely based on skeletons. For precision, recall and F1-score, our approach yields state-of-the-art performance, outperforming the recently proposed VideoDarwin [12] by more than 16%.

6. Conclusion

In this paper, we have proposed an end-to-end two-stream RNN architecture for skeleton based action recognition, with the temporal stream modeling temporal dynamics and the spatial stream processing spatial configurations.

Table 3. Comparison of the proposed approach with the state-of-the-art methods on the NTU RGB+D dataset.

Method	Cross subject	Cross view
Lie Group [44]	50.1	52.8
Skeletal Quads [10]	38.6	41.4
FTP Dynamic [18]	60.2	65.2
HBRNN [7]	59.1	64.0
Part-aware LSTM [37]	62.9	70.3
Trust Gate ST-LSTM [30]	69.2	77.7
Two-stream RNN	**71.3**	**79.5**

Table 4. Comparison of the proposed approach with the state-of-the-art methods on the SBU Interaction dataset.

Method	Accuracy
Joint Feature [53]	80.3
Joint Feature [22]	86.9
HBRNN [7]	80.4
Deep LSTM [55]	86.0
Co-occurrence LSTM [55]	90.4
Trust Gate ST-LSTM [30]	93.3
Two-stream RNN	**94.8**

Table 5. Comparison of the proposed approach with the state-of-the-art methods on the ChaLearn Gesture Recognition dataset.

Method	Precision	Recall	F1-score
Skeleton Feature [48]	59.9	59.3	59.6
Portfolios [52]	–	–	56.0
Gesture Spotting [35]	61.2	62.3	61.7
HiVideoDarwin [45]	74.9	75.6	74.6
CNN for Skeleton [5]	91.2	91.3	91.2
VideoDarwin [12]	75.3	75.1	75.2
Two-stream RNN	**91.7**	**91.8**	**91.7**

We explore two structures to model the sequence of joints of skeletons for the temporal stream. For the spatial stream, we also devise two methods to convert the structure of skeleton into a sequence before using a RNN to handle the spatial dependency. Moreover, to improve generalization and prevent overfitting for deep learning based methods, we employ rotation transformation, scaling transformation and shear transformation as data augmentation techniques based on 3D transformation of skeletons. Our experiments have shown that two-stream RNN outperforms existing state-of-the-art skeleton based approaches on datasets for generic actions (NTU RGB+D), interaction activities (SBU Interaction) and gestures (ChaLearn). In the future, we will consider to learn the structure patterns for the spatial channel and further improve the results.

Acknowledgement

This work is jointly supported by National Key Research and Development Program of China (2016YFB1001000), National Natural Science Foundation of China (61525306, 61633021, 61420106015) and Beijing Natural Science Foundation (4162058). This work is also supported by grants from NVIDIA and the NVIDIA DGX-1 AI Supercomputer.

References

[1] J. K. Aggarwal and X. Lu. Human activity recognition from 3d data: A review. *Pattern Recognition Letters*, 2014.

[2] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 2011.

[3] V. Bloom, D. Makris, and V. Argyriou. G3d: A gaming action dataset and real time action recognition evaluation framework. In *CVPR Workshop*. IEEE, 2012.

[4] 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*. IEEE, 2015.

[5] Y. Du, Y. Fu, and L. Wang. Skeleton based action recognition with convolutional neural network. In *ACPR*. IEEE, 2015.

[6] Y. Du, Y. Fu, and L. Wang. Representation learning of temporal dynamics for skeleton-based action recognition. *IEEE Transactions on Image Processing*, 2016.

[7] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*. IEEE, 2015.

[8] S. Escalera, X. Baró, J. Gonzalez, M. A. Bautista, M. Madadi, M. Reyes, V. Ponce-López, H. J. Escalante, J. Shotton, and I. Guyon. Chalearn looking at people challenge 2014: Dataset and results. In *ECCV Workshop*. Springer, 2014.

[9] S. Escalera, J. Gonzalez, X. Baro, M. Reyes, O. Lopes, I. Guyon, V. Athitsos, and H. J. Escalante. Multi-modal gesture recognition challenge 2013: dataset and results. In *ICMI*. ACM, 2013.

[10] G. Evangelidis, G. Singh, and R. Horaud. Skeletal quads: Human action recognition using joint quadruples. In *ICPR*. IEEE, 2014.

[11] A. Fathi, Y. Li, and J. Rehg. Learning to recognize daily actions using gaze. *ECCV*, 2012.

[12] B. Fernando, E. Gavves, M. J. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *CVPR*. IEEE, 2015.

[13] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber. Learning precise timing with lstm recurrent networks. *Journal of machine learning research*, 2002.

[14] M. A. Goodale and A. D. Milner. Separate visual pathways for perception and action. *Trends in Neurosciences*, 1992.

[15] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *TPAMI*, 2009.

[16] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. A field guide to dynamical recurrent neural networks. IEEE Press, 2001.

[17] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

[18] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. In *CVPR*. IEEE, 2015.

[19] Y. Huang, W. Wang, and L. Wang. Bidirectional recurrent convolutional networks for multi-frame super-resolution. In *NIPS*, 2015.

[20] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *IJCAI*, 2013.

[21] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *TPAMI*, 2013.

[22] Y. Ji, G. Ye, and H. Cheng. Interactive body part contrast mining for human interaction recognition. In *ICME Workshop*. IEEE, 2014.

[23] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, and M. Shah. High-level event recognition in unconstrained videos. *IJMIR*, 2013.

[24] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics*, 1973.

[25] G. Johansson. Visual motion perception. *Scientific American*, 1975.

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

[27] T. Lan, Y. Wang, W. Yang, and G. Mori. Beyond actions: Discriminative models for contextual group activities. In *NIPS*, 2010.

[28] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In *CVPR Workshop*. IEEE, 2010.

[29] Y. Li, Z. Ye, and J. M. Rehg. Delving into egocentric actions. In *CVPR*. IEEE, 2015.

[30] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *ECCV*. Springer, 2016.

[31] F. Lv and R. Nevatia. Recognition and segmentation of 3-d human action using hmm and multi-class adaboost. In *ECCV*. Springer, 2006.

[32] S. Z. Masood, C. Ellis, A. Nagaraja, M. F. Tappen, J. J. LaViola, and R. Sukthankar. Measuring and reducing observational latency when recognizing actions. In *ICCV Workshop*. IEEE, 2011.

[33] E. Ohn-Bar and M. Trivedi. Joint angles similarities and hog2 for action recognition. In *CVPR Workshop*. IEEE, 2013.

[34] F. J. Ordóñez and D. Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 2016.

[35] T. Pfister, J. Charles, and A. Zisserman. Domain-adaptive discriminative one-shot learning of gestures. In *ECCV*. Springer, 2014.

[36] S. Sempena, N. U. Maulidevi, and P. R. Aryan. Human action recognition using dynamic time warping. In *ICEEI*. IEEE, 2011.

[37] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *CVPR*. IEEE, 2016.

[38] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 2013.

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

[40] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. In *ICML*. ACM, 2015.

[41] J. Sung, C. Ponce, B. Selman, and A. Saxena. Human activity detection from rgbd images. *Plan, Activity, and Intent Recognition*, 2011.

[42] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[43] V. Veeriah, N. Zhuang, and G.-J. Qi. Differential recurrent neural networks for action recognition. In *ICCV*. IEEE, 2015.

[44] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *CVPR*. IEEE, 2014.

[45] H. Wang, W. Wang, and L. Wang. Hierarchical motion evolution for action recognition. In *ACPR*. IEEE, 2015.

[46] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*. IEEE, 2012.

[47] D. Wu and L. Shao. Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition. In *CVPR*. IEEE, 2014.

[48] J. Wu, J. Cheng, C. Zhao, and H. Lu. Fusing multi-modal features for gesture recognition. In *ICMI*, 2013.

[49] L. Xia, C.-C. Chen, and J. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *CVPR Workshop*. IEEE, 2012.

[50] Y. Yacoob and M. J. Black. Parameterized modeling and recognition of activities. In *ICCV*. IEEE, 1998.

[51] X. Yang and Y. L. Tian. Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In *CVPRW*. IEEE, 2012.

[52] A. Yao, L. Van Gool, and P. Kohli. Gesture recognition portfolios for personalization. In *CVPR*. IEEE, 2014.

[53] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, and D. Samaras. Two-person interaction detection using body-pose features and multiple instance learning. In *CVPR Workshop*. IEEE, 2012.

[54] V. Zatsiorski. Kinematics of human motion. *Human Kinetics, Champaign, IL*, 1998.

[55] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *AAAI*. AAAI, 2016.