Skeleton-Based Relational Modeling for Action Recognition

Wu Zheng*13
zhengwu2016@ia.ac.cn
Lin Li*13
lilin2015@ia.ac.cn
Zhaoxiang Zhang123
zhaoxiang.zhang@ia.ac.cn
Yan Huang23
huangyan@nlpr.ia.ac.cn
Liang Wang23
wangliang@nlpr.ia.ac.cn

1 Research Center for Brain-inspired Intelligence, CASIA
Beijing, China
2 National Laboratory of Pattern Recognition, CASIA
Beijing, China
3 University of Chinese Academy of Sciences
Beijing, China

Abstract

With the fast development of effective and low-cost human skeleton capture systems, skeleton-based action recognition has attracted much attention recently. Most existing methods use Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) to extract spatio-temporal information embedded in the skeleton sequences for action recognition. However, these approaches are limited in the ability of relational modeling in a single skeleton, due to the loss of important structural information when converting the raw skeleton data to adapt to the CNN or RNN input. In this paper, we propose an Attentional Recurrent Relational Network-LSTM (ARRN-LSTM) to simultaneously model spatial configurations and temporal dynamics in skeletons for action recognition. The spatial patterns embedded in a single skeleton are learned by a Recurrent Relational Network, followed by a multi-layer LSTM to extract temporal features in the skeleton sequences. To exploit the complementarity between different geometries in the skeleton for sufficient relational modeling, we design a two-stream architecture to learn the relationship among joints and explore the underlying patterns among lines simultaneously. We also introduce an adaptive attentional module for focusing on potential discriminative parts of the skeleton towards a certain action. Extensive experiments are performed on several popular action recognition datasets and the results show that the proposed approach achieves competitive results with the state-of-the-art methods.

1 Introduction

Action recognition provides a reasonable approach for video understanding and is under great demand, especially in the domains of intelligent surveillance and human-computer interaction. Traditional approaches are mainly based on the modeling of appearance and optical flow. However, the difficulty of extracting high-level features while eliminating...
noise interference in RGB video, such as illumination change, object occlusion and background clutter, dramatically obstructs the development of action recognition. Therefore, action recognition remains one of the most challenging problems in computer vision.

Benefitted from the advent of affordable depth sensors [1, 41] and efficient algorithms [28] of estimating joint positions based on depth maps, dynamic human skeleton becomes an available and effective modality for action recognition. Meanwhile, compared with RGB video, the characteristics of high-level representation and robustness to viewpoints, appearances and background noise [25] make skeletons play an essential role in action recognition. As a result, many early skeleton-based methods were proposed and showed encouraging improvements, such as [2, 9, 21, 31, 33, 33, 35, 39]. However, these approaches were significantly limited in either the lack of exploitation towards spatial structures [33] or the dependence for low-level hand-crafted features to analyze the spatial patterns [1, 3, 21, 33, 33, 33, 33].

Recently, various deep learning based methods have been proposed to conduct skeleton-based action recognition. In general, these approaches are mainly based on CNN [23] and RNN [24] for capturing spatio-temporal information in skeletons. Specially, the CNN-based methods utilize the powerful representation ability of CNN and achieve better performances than those methods using hand-crafted features. While the RNN-based models have shown great advantages in capturing temporal dynamics in sequential skeletons. However, the CNN usually lose important structural information in the process of encoding skeletons into spatial-temporal images [1, 1, 21, 21, 21, 21, 21], and RNN have the same weakness when learn the spatial features in a single skeleton [1, 21, 21, 21, 21]. Thus, when converting the raw skeleton data to match with the CNN or RNN input format, the destruction of the original relationship among the skeleton joints and lines leads to the difficulty in extracting enough spatial features in a single skeleton, which remains the main weakness of these frameworks.

In this paper, we propose an Attentional Recurrent Relational Network [22]-LSTM(ARRN-LSTM) to model temporal dynamics and spatial configurations in skeletons for action recognition. Our approach is based on a two-stream architecture to learn sufficient relational information by exploiting the complementarity between joints and lines in the skeleton. In each stream, we use the Recurrent Relational Network to learn the spatial patterns in a single skeleton and exploit a multi-layer LSTM to extract temporal information in skeleton sequences. Following the Recurrent Relational Network, we introduce an adaptive attentional module for focusing on potential discriminative parts of a skeleton towards a certain

Figure 1: The expanded structure of Recurrent Relational Network(RRN), which is used to learn the spatial pattern in the single skeleton frame by modeling joints and lines separately.
action. Different from CNN and RNN, the Recurrent Relational Network is actually a fully-connected graph model with each node connecting to others, as depicted in Figure 1, which ensures the flexible flow of information in the graph and helps to capture relational information among joints or lines. Meanwhile, the skeleton itself has a graph structure and its spatial information is essential to action recognition. Therefore, we believe the Recurrent Relational Network could be taken as a good architecture for learning spatial patterns in the single skeleton. Overall, our contributions can be summarized as follows:

- We propose a two-stream Attentional RRN-LSTM to conduct skeleton-based action recognition. The RRN-LSTM module is for capturing spatio-temporal information among joints or lines in skeletons, while the attentional module is for focusing on potential discriminative parts of a skeleton towards a certain action.

- To the best of our knowledge, it may be the first attempt to apply the Recurrent Relational Network to action recognition by modeling relationship in a single skeleton. We believe that it could provide a brand new perspective to address this problem.

- We perform extensive experiments on several popular action recognition datasets and obtain competitive results with the state-of-the-art methods.

2 Related Work

2.1 Relational Network

Modern deep learning methods have made great progress in solving non-structural data problems, but many previous approaches have neglected to consider the connections and structures in data. Santoro et al. [26] propose a simple plug-and-play neural network module for relational reasoning. With this module, a neural network gains the ability of handling unstructured inputs and inferring their hidden relationship, which achieves state-of-the-art results on a visual question answering dataset [11]. Based on this work, Palm et al. [22] propose the Recurrent Relational Networks for complex relational reasoning, such as learning an iterative strategy to solve Sudoku. This framework constructs a graph model, with each node having information channels to and from all nodes for updating its internal state.

2.2 Skeleton-based Action Recognition with Deep Networks

To utilize the powerful representation ability of CNN, skeletons are usually encoded into spatial-temporal images to fit the inputs of CNN. Hou et al. [7] accumulate the raw skeleton frames directly and encode the color based on temporal information. Imran et al. [10] add an extra channel of Motion History Images for the complementarity of temporal information. Liu et al. [19] exploit the 3DCNN to extract spatio-temporal features for avoiding the loss of information in projecting process. Du et al. [5] divide the joints into five main parts according to human physical structures (four limbs and one trunk) and take the 3D coordinators of joints as 3 channels of RGB image. Other works [4, 12, 13, 18] consider the geometric features calculated by the distances between pair-wise joints. Li et al. [16] perform a linear transformation on raw skeleton joints to learn the optimal order of joints. Specially, Yan et al. [37] use the Graph Neural Network in its framework to form hierarchical representation of skeletons and achieve results outperforming the mainstream methods.
Figure 2: Framework of the proposed two-stream ARRN-LSTM model. It is recommended to view the digital version for details by zooming in.

RNN is good at processing sequential data due to the extraordinary ability of capturing structural information in sequences. Du et al. [1] divide the human skeleton into five parts according to human physical structure and separately feed them into different RNNs, and the neighbouring parts are then fused to be the inputs of higher layers. Wang et al. [2] propose a two-stream RNN architecture to model temporal dynamics and spatial configurations separately and fuse the scores of two streams finally for recognition. Zhu et al. [3] insert fully-connected layers between multi-layer LSTM to fuse information from joints in different parts, which is called co-occurrence exploration. Shahroudy et al. [4] propose a Part-Aware LSTM unit that builds full connections between all the memory cells and all the input features for acquiring richer information. Liu et al. [5] transform the joints in the form of tree structure based on traversal and propose a spatio-temporal LSTM framework to learn spatio-temporal information in joint sequences.

The CNN and RNN based approaches usually lose important structural information when converting the raw skeleton data to adapt to the CNN and RNN input, which is our motivation of adopting RRN to model spatial configurations in a single skeleton.

3 Proposed Method

3.1 Pipeline Overview

We propose a two-stream Attentional RRN-LSTM framework for action recognition, which is depicted in Figure 2. Our framework consists of two streams for modeling temporal dynamics and spatial configurations in the skeleton by utilizing joints and lines separately. In each stream, an embedding operation is first performed on each joint or line to fit the input size of the following RRN. The embedding results of joints or lines are then sent to RRN for relational modeling, which aims to capture the spatial patterns in a skeleton effectively. To focus more attention on potential discriminative parts of a skeleton towards a certain action, we take a mask to perform point-wise multiplication with node outputs of RRN, followed by a fully-connected layer to perform dimensionality reduction on the product. After that, we use a multi-layer LSTM to process the dimensionally reduced outputs, aiming to obtain temporal information in skeleton sequences. Finally, a fusion strategy is proposed to
combine the information from both streams by connecting a fully-connected layer after the multi-layer LSTM and taking the weighted outputs of the two streams as the final prediction.

3.2 The Construction of the Two-Stream ARRN-LSTM

3.2.1 Joint or Line Embedding

To model the skeleton with a graph model, we should first adjust the data format of skeletons to fit the graph model input. The raw skeleton is defined by a fixed number of joints in the form of 3D coordinators, with the number denoted as $J$. Our embedding is conducted by a fully-connected layer of a fixed output size $M$. Thus, given a joint $c_{t,i} = (c^x_{t,i}, c^y_{t,i}, c^z_{t,i})$ that means the 3D coordinator of the $i$-th joint in the $t$-th frame, the joint embedding result $v_{t,i}$ is:

$$v_{t,i} = Emb(c_{t,i}) = Emb(c^x_{t,i}, c^y_{t,i}, c^z_{t,i})$$ (1)

Compared with other graph-based models exploring relational patterns in both single skeleton and skeleton sequences, our RRN model realizes relational modeling only using the features from single skeleton without any temporal information. Thus, we try to provide RRN with more information from single skeleton than in traditional graph-based models to relieve this insufficiency. By modeling relational patterns with more information from single skeleton, we think it is also beneficial to the extraction of temporal information by the following multi-layer LSTM.

Except the original joints, we believe that the lines between pair-wise joints are also important geometric structures in the skeleton and contain rich spatial information. Thus, we obtain the lines by calculating the distances between joints in three dimensions separately to construct a second stream. And we combine the two streams information for action predictions in the end, by exploiting the complementarity between both geometries, we can achieve better results. We give an example of calculating lines between $c_{t,i}$ and other joints as follows:

$$l_{t,i} = (c^x_{t,i} - c^x_{t,1}, c^y_{t,i} - c^y_{t,1}, c^z_{t,i} - c^z_{t,1}, c^x_{t,i} - c^x_{t,2}, c^y_{t,i} - c^y_{t,2}, c^z_{t,i} - c^z_{t,2}, ..., c^x_{t,i} - c^x_{t,J}, c^y_{t,i} - c^y_{t,J}, c^z_{t,i} - c^z_{t,J})$$ (2)

The $l_{t,i}$ denotes all lines that connect the joint $c_{t,i}$ and all the other joints, but these distances exclude the one between $c_{t,i}$ and itself. Thus, if the dimension of one joint is 3, the dimension of the corresponding lines is $3 \times (J - 1)$. Similarly, the line embedding process is:

$$v_{t,i} = Emb(l_{t,i}) = Emb(l^1_{t,i}, l^2_{t,i}, l^3_{t,i}, ..., l^{3\times (J-1)}_{t,i})$$ (3)

Because the output dimensions of both joint and line embeddings are equal, we use the same symbol $v_{t,i}$ to denote the embedding result of lines $l_{t,i}$ in following expressions.

3.2.2 Recurrent Relational Network

The temporal and spatial information in the skeleton is essential to recognize human action. The ARRN-LSTM module is expected to be a good architecture to extract these information as the RRN emphasizes the exchange of information among nodes and shows great performances on relational reasoning. Specially, we use all joints or lines in a skeleton to feed the RRN. In the process, each joint or line embedding $v_{t,i}$ will be the input to one node in RRN and the number of nodes in RNN is equal to $J$. This process is:

$$w_t = Rrn(v_t) = Rrn(v_{t,1}, v_{t,2}, v_{t,3}, ..., v_{t,J})$$ (4)
As for the detailed process in a single node, if we denote the states of node \( i \) and \( j \) in the \( e \)-th iteration of RRN as \( h^e_i \) and \( h^e_j \), we can define the information flow from node \( j \) to node \( i \) as:

\[
s^e_{i,j} = f(h^e_i, h^e_j)
\]

(5)

\( f \) is a message function. The messages from all neighbouring nodes to joint \( i \) are then summed up as:

\[
s^e_i = \sum_{j \in N(i)} s^e_{i,j}
\]

(6)

Then the output of node \( i \) can be updated by a trainable node function \( g \) as:

\[
h^e_i = g(h^{e-1}_i, v_{t,i}, s^e_i)
\]

(7)

Given the number of iterations in RRN as \( E \), the output of node \( i \) after the final iteration can be expressed as:

\[
w_{t,i} = h^E_i
\]

(8)

At this point, we calculate the relational modeling results of the joints or lines in a skeleton as \( w_t = (w_{t,1}, w_{t,2}, w_{t,3}, \ldots, w_{t,J}) \), in the dimension of \((J, M)\).

### 3.2.3 Attentional Module

For a certain action, humans usually recognize it by focusing the most discriminative parts. For example, the action of kicking can be identified through the legs, while the drinking action can be recognized by the arms. However, some different actions, such as flipping and reading a book, cannot be distinguished until the subtle differences in the hand part are identified. Thus, we propose an attentional module to address these problems, with the module following the RRN and taking the relational modeling results as inputs. We first take a mask to perform point-wise multiplication with node outputs of RRN and then use a fully-connected layer to reduce the product dimension. We express the process as follows:

\[
p_t = \text{Att}(w_t) = Fc(m_1 \cdot w_{t,1}, m_2 \cdot w_{t,2}, m_J \cdot w_{t,J})
\]

(9)

The mask \( m = (m_1, m_2, \ldots, m_J) \) is a \( J \)-dimensional vector and can be trained with the back-propagation algorithm, which aims to emphasize the impacts of some joints or lines while neglect other unimportant ones. The output \( p_t \) can expressed as \((p_{t,1}, p_{t,2}, p_{t,3}, \ldots, p_{t,J})\).

### 3.2.4 Multi-Layer LSTM

Following RRN, we exploit a multi-layer LSTM to learn the temporal dynamics in sequential skeletons. After concatenating all joints or lines features in a skeleton frame from the attentional module and performing a dimension reduction operation, we feed the result to one cell in LSTM. The number of cells in the single-layer LSTM is equal to the skeleton sequence length \( T \), so all frames in a skeleton sequence can exchange information with internal connections in LSTM. This process can be denoted as follows:

\[
(q_1, q_2, q_3, \ldots, q_T) = \text{Multi_lstm}(p_1, p_2, p_3, \ldots, p_T)
\]

(10)

The number of layers in multi-layer LSTM is \( H \) and the dimension of \( q_t \) is \( h \).
3.2.5 Score Fusion

Following the multi-layer LSTM, we connect a fully-connected layer to map the extracted spatial-temporal features to the categories of size $K$. Then we run a softmax operation on the output of the fully-connected layer to obtain the predicted probabilities. We take the process in the joint stream as an example and calculate the predicted probability vector $y_j$ as follows:

$$y_j = \text{softmax}(Fc(q_1,q_2,q_3,\ldots,q_T))$$  \hfill (11)

To fuse the scores from two streams, we calculate the weighted scores as our final prediction. We use $y_j$ and $y_l$ to denote the scores of joint and line stream, respectively. The final predicted probability vector $y$ is calculated as follows:

$$y = \alpha \cdot y_j + \beta \cdot y_l$$  \hfill (12)

$\alpha$ and $\beta$ are relative weights of the two stream predictions and their sum is equal to 1.

4 Experiments and Results

In our experiments, we firstly train the proposed two-stream Attentional RRN-LSTM model and test it on three popular action recognition datasets, i.e., NTU RGB+D, Florence 3D and MSRAAction3D, with results and comparisons shown in Tables 1, 2 and 3 separately. Then, we perform detailed ablation study on two streams and the attentional module to validate each part of our framework, showing the results in Tables 4 and 5.

4.1 Implementation Details

We normalize the joint coordinators by subtracting the average value of the 5 joints close to the hip joint. The lines are calculated according to the normalized joints. We perform zero padding on videos with frames less than $T$ and random sampling on videos with frames more than $T$ to fix all videos as $T$ frames. According to the differences of frame numbers in different datasets, we set $T = 100, 25, 20$ for NTU-RGBD, Florence3D and MSRAAction3D, respectively. Besides, we conduct the embedding with a fully-connected layer and set the output size $M = 50, 20, 20$ for the three datasets based on cross-validation. The following RRN executes $E = 5$ iterations per frame with each node function $g$ realized by an GRU unit and the message function $f$ constructed with 3 fully-connected layers. The attentional module is built with a trainable mask $m$ and a fully-connected layer for reducing the product to a 256-dimensional vector. We use $H = 3$ layers LSTM to extract temporal information in skeleton sequences, with input and output set as a 256-dimensional and $h = 512$-dimensional vector separately. The mid-layer LSTM has an output size of 512,256,256 for NTU-RGBD, Florence3D and MSRAAction3D, respectively. We set both $\alpha$ and $\beta$ as 0.5 in our experiments. We use Stochastic Gradient Descent to train our model from scratch on NTU-RGBD and set the initial learning rate as 0.01, we multiply the learning rate with 0.1 when the accuracy gets saturated. On other datasets, we use Adam optimizer to train our model from scratch. Our model is trained on a NVIDIA TITAN X GPU with PyTorch [17].
4.2 Datasets

**NTU RGB+D Dataset.** This is the largest depth-based skeleton action recognition dataset currently with more than 56 thousand video samples and 4 million frames collected from 40 different subjects. It consists of 60 different action classes, including daily, mutual and health-related actions. We evaluate our model according to the metrics proposed in [27], including Cross-Subject(CS) and Cross-View(CV) evaluations.

**Florence 3D.** This dataset includes 215 action sequences performed by 10 subjects for 2 to 3 times. It is made up of 9 activities and each skeleton is represented by 15 joints. The difficulty of this dataset lies in its similarities between actions, such as drinking from a bottle, answering phone and reading watch. We follow the standard metric, i.e., leave-one-subject-out cross validation, to evaluate our model.

**MSRAction3D.** This dataset contains 20 actions performed by seven subjects for three times, which totally consists of 4020 action samples. The dataset is divided into three subsets and each subset has 8 actions. In each subset, the samples of subjects 1, 3, 5, 7, 9 are used for training while the samples of subjects 2, 4, 6, 8, 10 are used for testing. The final accuracy is calculated as the average accuracies of the three subsets.

#### Table 1: Accuracies of different methods on the NTU RGB+D (%).

| Methods                | CV  | CS  |
|------------------------|-----|-----|
| Lie Group [30]         | 52.8| 50.1|
| Dynamic Skeletons [8]  | 65.2| 60.2|
| Deep LSTM [27]         | 67.3| 60.7|
| PA-LSTM [27]           | 70.3| 62.9|
| ST-LSTM+TS [20]        | 77.7| 69.2|
| STA-LSTM [26]          | 81.2| 73.4|
| ST-NBMIM [36]          | 84.2| 80.0|
| Deep STGCK [17]        | 86.3| 74.9|
| VA-LSTM [38]           | 87.6| 79.4|
| C-CNN + MTLN [13]      | 84.8| 79.6|
| ST-GCN [37]            | 88.3| 81.5|
| ARRN-LSTM              | 89.6| 81.8|

#### Table 2: Accuracies of different methods on the Florence 3D (%).

| Methods                | Accuracy |
|------------------------|----------|
| Riemannian Manifold [3] | 87.04    |
| Lie Group [31]         | 90.88    |
| Graph-Based [31]       | 91.63    |
| MIMTL [33]             | 95.29    |
| P-LSTM [26]            | 95.35    |
| STGCK [26]             | 97.67    |
| Deep STGCK [17]        | 99.07    |
| ARRN-LSTM              | 98.52    |

#### Table 3: Accuracies of different methods on the MSRAction3D (%).

| Methods                | Accuracy |
|------------------------|----------|
| Lie Group [31]         | 92.5     |
| SCK+DCK [13]           | 94.0     |
| HBRNN [6]              | 94.5     |
| ST-LSTM [20]           | 94.8     |
| Graph-Based [26]       | 94.8     |
| ST-NBNN [37]           | 94.8     |
| ST-NBMIM [36]          | 95.3     |
| ARRN-LSTM              | 95.0     |
Table 4: Results of ablation study on NTU RGB+D.

| Streams    | With Attention | Without Attention |
|------------|----------------|-------------------|
|            | CS  | CV  | CS  | CV  |
| Joint      | 79.6| 87.8| 74.6| 83.1|
| Line       | 76.4| 87.2| 74.5| 83.3|
| Two Stream | 81.8| 89.6| 77.6| 84.2|

Table 5: Comparisons of iteration times on NTU RGB+D.

| Iteration Times | Joint Stream | Line Stream |
|-----------------|--------------|-------------|
|                 | CS  | CV  | CS  | CV  |
| 3               | 72.6| 82.0| 72.3| 81.1|
| 5               | 74.6| 83.1| 74.5| 83.3|
| 7               | 73.8| 82.3| 73.1| 81.7|

4.3 Comparisons with State of the Arts

NTU RGB+D Dataset. We compare our ARRN-LSTM model with previous state-of-the-art methods on this dataset in Table 1. It is clear from the table that our ARRN-LSTM method outperforms the mainstream methods and achieves the state-of-the-art results. Similar to the ST-GCN model, our framework is also based on graph model, but ST-GCN [37] takes the graph convolution to learn the spatial patterns and temporal features in skeleton sequences, while our method takes the RRN to perform relational modeling in a single skeleton and uses a multi-layer LSTM to obtain the temporal information in skeleton sequences.

Florence 3D. As shown in Table 2, the proposed ARRN-LSTM framework is superior to the mainstream methods that are based on LSTM, CNN and traditional algorithms. Compared with the state-of-the-art Deep STGCK [17], although our result is slightly lower on Florence 3D, we achieve much better results on the larger dataset NTU RGB+D.

MSRAAction3D. As shown in Table 3, our method achieves 95.0% accuracy and outperforms the mainstream methods. The result is competitive with the state-of-the-art method ST-NBMIM [36], which validates the effectiveness of our method on the small dataset. Though our result is slightly weaker than ST-NBMIM, our approach achieves much better results on NTU RGB+D dataset. We believe that it may be caused by the fact that LSTM is easy to be overfitting on small datasets, which is the issue that we are going to address in the future.

4.4 Ablation Study

We examine the effectiveness of the proposed ARRN-LSTM and study the impact of both streams and attentional module in this section, with results shown in Table 4. We perform the ablation study on NTU RGB+D and calculate the accuracies under both cross subject and cross view metrics. We study the performances of each stream with or without attentional module separately.

It is clear that the ARRN-LSTM in a single stream with attentional module can achieve comparable performances with those mainstream methods, which validates the effectiveness of the Recurrent Relational Network. Our results of single stream are slightly weaker than ST-GCN model, we believe this is because the RRN only extracts the relational features in
single skeleton, unlike ST-GCN extracting information from both single skeleton and skeleton sequences. But by combining the two streams, the accuracy of our model is increased by 2 ∼ 3 points, which proves we can achieve better results by adding more information from single skeleton to RRN, and also proves there exists complementarity between joints and lines in the skeleton for action recognition. Besides, the attentional module increases the accuracy by 2 ∼ 5 points under both metrics, validating the insight of the attention mechanism.

We also study the impacts of iteration times of RRN on two streams without attention module separately and summarize the results in Table 5. The results include the accuracies of 3, 5, 7 iteration times on joint stream and line stream separately under CV and CS metrics. From the table, we discover that the performances are weaker when the iteration times are small, while the performances can hardly be improved when the iteration times transcend 5. This illustrates that the RRN can model the relationship in the skeleton effectively with limited iterations.

4.5 Discussion

We propose a ARRN-LSTM model to learn the spatio-temporal features for skeleton-based action recognition. Although we can validate the effectiveness of RRN in modeling single skeleton, which may be the main contribution of this work, there still exists possibility of modeling both single skeleton and skeleton sequences with relation network for action recognition. In other words, integrating the function of multi-layer LSTM that models the temporal patterns into relation network, this deserves to be explored in the future.

5 Conclusion

In this paper, we have proposed a two-stream Attentional RRN-LSTM for skeleton-based action recognition and achieve competitive results with the state-of-the-art methods. To the best of our knowledge, the proposed ARRN-LSTM for modeling temporal dynamics and spatial configurations is may be the first attempt to apply the Recurrent Relational Network to skeleton-based action recognition. The designed attentional module also plays an important role in boosting the performance by focusing potential discriminative parts in skeleton. And we utilize a two-stream architecture to exploit the complementarity between joints and lines for obtaining enough spatial information. In the future, we will consider to learn more fine-grained patterns for further improving the results.

References

[1] Intel realsense. https://software.intel.com/en-us/realsense.

[2] Rizwan Chaudhry, Ferda Ofli, Gregorij Kurillo, Ruzena Bajcsy, and Rene Vidal. Bio-inspired dynamic 3d discriminative skeletal features for human action recognition. In CVPRW, pages 471–478. IEEE, 2013.

[3] Maxime Devanne, Hazem Wannous, Stefano Berrettii, Pietro Pala, Mohamed Daoudi, and Alberto Del Bimbo. 3-d human action recognition by shape analysis of motion trajectories on riemannian manifold. IEEE Transactions on Cybernetics, 45(7):1340–1352, 2015.
[4] Zewei Ding, Pichao Wang, Philip O Ogunbona, and Wanqing Li. Investigation of different skeleton features for cnn-based 3d action recognition. In ICMEW, pages 617–622. IEEE, 2017.

[5] Yong Du, Yun Fu, and Liang Wang. Skeleton based action recognition with convolutional neural network. In ACPR, pages 579–583. IEEE, 2015.

[6] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In CVPR, pages 1110–1118. IEEE, 2015.

[7] Yonghong Hou, Zhaoyang Li, Pichao Wang, and Wanqing Li. Skeleton optical spectra based action recognition using convolutional neural networks. IEEE Transactions on Circuits and Systems for Video Technology, 2016.

[8] Jian-Fang Hu, Wei-Shi Zheng, Jianhuang Lai, and Jianguo Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. In CVPR, pages 5344–5352. IEEE, 2015.

[9] 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 IJCAI, pages 2466–2472, 2013.

[10] Javed Imran and Praveen Kumar. Human action recognition using rgb-d sensor and deep convolutional neural networks. In ICACCI, pages 144–148. IEEE, 2016.

[11] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In CVPR, pages 1988–1997. IEEE, 2017.

[12] Qiuhong Ke, Senjian An, Mohammed Bennamoun, Ferdous Sohel, and Farid Boussaid. Skeletonnet: Mining deep part features for 3-d action recognition. Signal Processing Letters, 24(6):731–735, 2017.

[13] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In CVPR, pages 4570–4579. IEEE, 2017.

[14] Piotr Koniusz, Anoop Cherian, and Fatih Porikli. Tensor representations via kernel linearization for action recognition from 3d skeletons. In ECCV, pages 37–53. Springer, 2016.

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[16] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Skeleton-based action recognition with convolutional neural networks. In ICMEW, pages 597–600. IEEE, 2017.

[17] Chaolong Li, Zhen Cui, Wenming Zheng, Chunyan Xu, and Jian Yang. Spatio-temporal graph convolution for skeleton based action recognition. arXiv preprint arXiv:1802.09834, 2018.
[18] Chuankun Li, Yonghong Hou, Pichao Wang, and Wanqing Li. Joint distance maps based action recognition with convolutional neural networks. *Signal Processing Letters*, 24(5):624–628, 2017.

[19] Hong Liu, Juanhui Tu, and Mengyuan Liu. Two-stream 3d convolutional neural network for skeleton-based action recognition. *arXiv preprint arXiv:1705.08106*, 2017.

[20] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *ECCV*, pages 816–833. Springer, 2016.

[21] Eshed Ohn-Bar and Mohan M Trivedi. Joint angles similarities and hog2 for action recognition. In *CVPRW*, pages 465–470. IEEE, 2013.

[22] Rasmus Berg Palm, Ulrich Paquet, and Ole Winther. Recurrent relational networks for complex relational reasoning. *arXiv preprint arXiv:1711.08028*, 2017.

[23] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPSW*, 2017.

[24] Fernando J Pineda. Generalization of back-propagation to recurrent neural networks. *Physical Review Letters*, 59(19):2229, 1987.

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

[26] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Tim Lillicrap. A simple neural network module for relational reasoning. In *NIPS*, pages 4974–4983, 2017.

[27] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. *arXiv preprint arXiv:1604.02808*, 2016.

[28] 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 *CVPR*, pages 1297–1304. IEEE, 2011.

[29] 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. In *AAAI*, volume 1, page 7, 2017.

[30] Vivek Veeriah, Naifan Zhuang, and Guo-Jun Qi. Differential recurrent neural networks for action recognition. In *ICCV*, pages 4041–4049. IEEE, 2015.

[31] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *CVPR*, pages 588–595. IEEE, 2014.

[32] Hongsong Wang and Liang Wang. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In *CVPR*. IEEE, 2017.

[33] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, pages 1290–1297. IEEE, 2012.
[34] Pei Wang, Chunfeng Yuan, Weiming Hu, Bing Li, and Yanning Zhang. Graph based skeleton motion representation and similarity measurement for action recognition. In *ECCV*, pages 370–385. Springer, 2016.

[35] Pichao Wang, Wanqing Li, Philip Ogunbona, Zhimin Gao, and Hanling Zhang. Mining mid-level features for action recognition based on effective skeleton representation. In *DICTA*, pages 1–8. IEEE, 2014.

[36] Junwu Weng, Chaoqun Weng, Junsong Yuan, and Zicheng Liu. Discriminative spatio-temporal pattern discovery for 3d action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[37] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:1801.07455*, 2018.

[38] Yanhua Yang, Cheng Deng, Shangqian Gao, Wei Liu, Dapeng Tao, and Xinbo Gao. Discriminative multi-instance multitask learning for 3d action recognition. *IEEE Transactions on Multimedia*, 19(3):519–529, 2017.

[39] Mihai Zanfir, Marius Leordeanu, and Cristian Sminchisescu. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *ICCV*, pages 2752–2759. IEEE, 2013.

[40] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *ICCV*, pages 2136–2145. IEEE, 2017.

[41] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE Multimedia*, 19(2):4–10, 2012.

[42] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, Xiaohui Xie, et al. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *AAAI*, volume 2, page 8, 2016.