Neurocomputing 370 (2019) 109–117

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Convolutional relation network for skeleton-based action recognition

Jiagang Zhu[a,b], Wei Zou[a,b,c,*], Zheng Zhu[a,b], Yiming Hu[a,b]

[a] Institute of Automation, Chinese Academy of Sciences, Beijing, China
[b] University of Chinese Academy of Sciences, Beijing, China
[c] TianJin Intelligent Tech.Institute of CASIA Co., Ltd, China

ARTICLE INFO

Article history:
Received 11 September 2018
Revised 20 April 2019
Accepted 12 August 2019
Available online 27 August 2019

Communicated by Dr. Cheng Jun

Keywords:
Action recognition
Skeleton
Deep learning
Joint interaction
Dilation
Attention

ABSTRACT

In the skeleton-based action recognition, mining information from the joints and limbs of human skeletons plays a key role. Previous studies treated the skeleton data as vectors and could not explicitly capture the joint interactions (e.g., RNN-based methods), or modeled the joint interactions in a local manner and may lose important cues without global response mapping (e.g., CNN and GCN (Graph Convolution Network) based methods). In this work, we address these problems by considering the potential relations of all the node pairs and edge pairs on the skeleton graphs. A dilation group-specific convolution module is proposed to aggregate relation messages of all the unit pairs on the skeleton graphs. By enumerating all the pair relations, the joint interactions could be learned explicitly and globally. It is then enhanced by introducing the attention pooling including temporal attention, spatial attention and channel attention. By stacking such several blocks, the relation messages of the node pairs are augmented by multi-layer propagation. Finally, the late fusion of four streams is used to combine the predictions of different inputs including node pairs, edge pairs and corresponding frame differences. The proposed method, termed conv-relation network, achieves competitive performance on two large scale datasets, NTU RGB+D and Kinetics.

1. Introduction

Recent years have witnessed deep learning [1] being widely popular in the vision community, e.g., image classification [1], object detection [2], video classification [3], pose recovery [4], stereo matching [5], image ranking [6] and visual question answering [7]. Particularly, human action recognition has received increasing attention due to potential applications in human–robot interaction, behavior analysis and surveillance. According to the types of input data, human action recognition can be categorized into RGB-based [3,8–12] and skeleton-based approaches [13–21]. Compared with RGB images, skeleton data has the merits of being lightweight and robust against background noise.

In this paper, we focus on the problem of skeleton-based human action recognition. The interactions of skeleton joints play a key role in characterizing an action. Traditional methods [13–15] design hand-crafted features to extract co-occurrence patterns from skeleton sequences. With the resurgence of neural networks, Recurrent Neural Networks (RNN) and Convolution Neural Networks (CNN) have been widely used in the skeleton-based action recognition [16–20]. The RNN based methods [16–18] transform the skeleton data into a joint coordinates vector and then capture the sequence information of skeleton. Compared with RNN, CNN has good parallel ability and can benefit from pretraining on the large scale datasets. CNN-based methods [19,20] primarily represent a skeleton sequence as a pseudo-image and recognize the underlying action in the same way as image classification. However, local convolution cannot learn the global joint interactions efficiently and the underlying assumption of the joints being adjacent spatially in the input tensor may introduce unreliable prior. Recently, graph convolution networks (GCN) based methods [21] have been used to capture joint interactions on the skeleton graphs, explicitly considering the adjacent relationship between joints in a non-Euclidean space. Nevertheless, the skeleton graphs and their manually designed convolution kernels also limit joint interaction to being learned in a local manner. This motivates us to design a model which can break the limit of local convolution and learn the joint interactions explicitly and globally.

In this paper, this problem is solved by considering the potential relations of all the node pairs and edge pairs on the skeleton graphs. Firstly, a dilation group-specific convolution module is proposed to compute the interactions of all the node pairs on the skeleton graphs. By using this module, convolutions can be

* Corresponding author at: Institute of Automation, Chinese Academy of Sciences, Beijing, China.
E-mail addresses: zhujiagang2015@ia.ac.cn (J. Zhu), wei.zou@ia.ac.cn (W. Zou), zhuzheng2014@ia.ac.cn (Z. Zhu), huyiming2016@ia.ac.cn (Y. Hu).

https://doi.org/10.1016/j.neucom.2019.08.043
0925-2312/

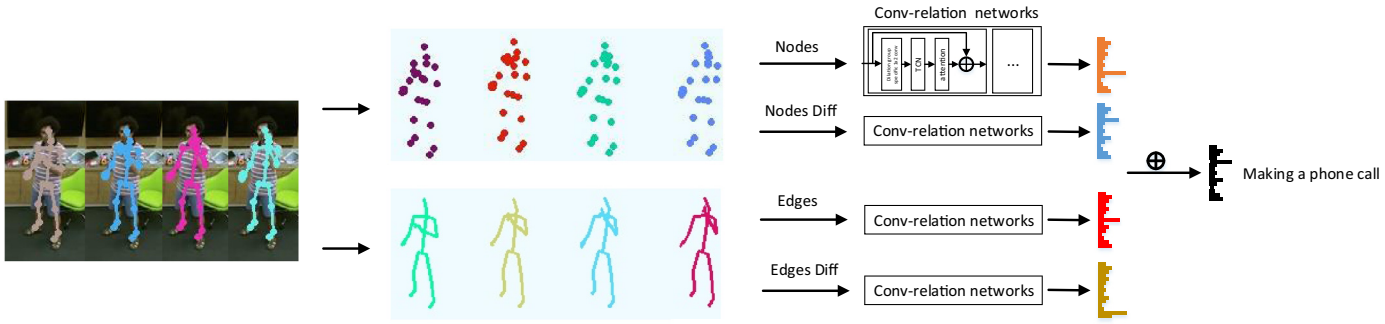

Fig. 1. Convolutional relation network for skeleton-based action recognition. The relations of node pairs, edge pairs and corresponding frame differences on the skeleton graphs are explicitly modeled respectively. Due to the space limit, we only show one building block of our model in the top, middle figure. The details of this block are given in Fig. 2. The late fusion of four streams is used to get the final prediction.

explicitly performed on all the node pairs. In this way, joint interactions are completely captured by pair interactions in a dense convolutional manner. Secondly, the attention pooling operations, including temporal attention, spatial attention and channel attention, are used to enhance the module. Finally, the late fusion of four streams is used to combine the predictions of different inputs including node pairs, edge pairs and corresponding frame differences. The proposed method, termed conv-relation network, is shown in Fig. 1.

The main contributions of this work can be summarized as follows:

- A dilation group-specific convolution module is proposed for aggregating dense relations of all the node pairs of the skeleton graphs. Besides, three attention models (i.e., temporal, spatial and channel attention) are proposed to enhance this convolution module.
- Different inputs on the skeleton graphs, including node pairs, edge pairs and their corresponding frame differences, are combined by the late fusion to improve the effectiveness.
- On two large scale datasets for skeleton-based action recognition, the proposed conv-relation network achieves competitive performance.

2. Related work

2.1. Skeleton-based action recognition

Hand-crafted features. Traditional methods designed handcrafted features to capture the dynamics of joint motion. These could be actionlet ensemble [13], covariance matrices of joint trajectories [14], rotations and translations between body parts [15], or temporal order information [22].

Deep learning methods. With the resurgence of neural networks, the long short term memory networks (LSTM) have been adopted for feature learning from skeleton sequences due to its ability of modeling long-term temporal dependency. In [17], a spatial–temporal LSTM model based on gating mechanism is proposed to filter out unreliable input due to the occlusion and sensor noise. [18] proposed an end-to-end fully connected deep LSTM network to learn co-occurrence features from skeleton data. In [23], a view-adaptive LSTM was introduced to transform the skeleton data to a suitable view adaptively for better action recognition. In recent years, more and more literatures adopted CNN for skeleton feature learning, due to the benefits of transferring knowledge from large scale image dataset. For example, [19] quantified the skeleton sequences into images and then fed them into CNN. In [24], a view-adaptive CNN was introduced to deal with the view variation challenge. Because of the operation of local convolution, CNN-based methods cannot capture the global relationship of all joints efficiently. Moreover, assuming the joints being adjacent in the Euclidean space may introduce unreliable prior. Human skeletons in videos can be treated as a spatial-temporal graph. Recently, graph convolution neural networks have been used to learn the spatial parts of human skeletons [21], which explicitly considers the adjacent relationship between joints in a non-Euclidean space. Nevertheless, the skeleton graphs and their manually designed convolution kernels also limit joint interaction to being learned in a local manner.

2.2. Video-based action recognition

One important stream of action recognition is video-based action recognition, where methods based on Deep CNN have dominated. Two-stream ConvNet [3] employs RGB images and optical flow stacks as the inputs of two networks and fuses their predictions by late fusion. Temporal Segment Network (TSN) [8] improves the performance of two-stream ConvNet by sparsely sampling video frames and learning video-level predictions. In 2017, DeepMind released a large-scale video action datasets Kinetics [25] and proposed Inflated 3D ConvNet (I3D).

Few-shot learning based methods [26] for action recognition, which aim to relieve the need for large amount of annotated data in the existing deep learning methods, have also developed rapidly. In traditional methods, the training and testing sets involve the same classes of samples. In a few-shot recognition setting, the network needs to effectively learn classifiers for novel concepts from only a few examples. Unlike traditional models trained on many data samples, the model in a few-shot setting is trained to generalize across different episodes.

Numerous methods for temporal context modeling in videos are also proposed, such as [27,28]. In this paper, the context information is obtained by the temporal convolution on the short clips, typically, less than 5 s. Note that this paper aims at exploring the information among skeleton joints and limbs (spatial) and the temporal context modeling is not the focus.

2.3. Relation reasoning

A simple plug-and-play module, Relation Network (RN) [29] was proposed to equip CNN with relation reasoning ability in several tasks. Recurrent relational networks [30] increased its ability of solving the tasks that require an order of magnitude more steps of relational reasoning. Non-local neural network [31] was designed to equip CNN with the ability of long range relation reasoning, including spatially in images and spatial–temporally in

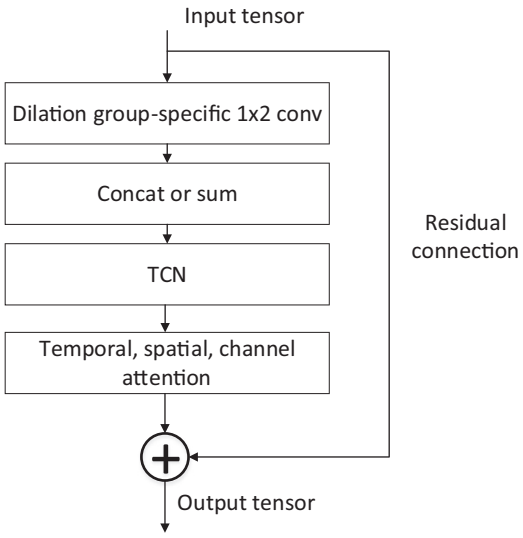

Fig. 2. The basic building block of convolutional relation module for feature extraction. It mainly consists of dilation group-specific 1×2 conv, Temporal Convolution (TCN), attention models.

Table 1
The structure of the conv-relation network. The number of input channels, output channels and the stride of each block are shown in the table. Data BN represents the data batch normalization layer. GAP represents the global average pooling layer.

	Input channels	Output channels	stride
Data BN	3	3	–
L1	3	72	1
L2	72	72	1
L3	72	72	1
L4	72	144	2
L5	144	144	1
L6	144	144	1
L7	144	288	2
L8	288	288	1
L9	288	288	1
GAP	–	–	–
softmax	–	–	–

videos. Temporal relation reasoning neural network (TRN) [32] was designed to learn and reason about the temporal dependencies of video frames at multiple time scales. The relations in these works were used to represent the underlying object interactions in spatial domain [29,30], temporal domain [32], or spatial–temporal domain [31]. In this paper, the relations of the skeletons joints are realized by 1×2 convolution in the spatial domain and the interactions in temporal domain are realized by the temporal convolutions.

3. Convolutional relation network

Fig. 2 shows the basic module of our conv-relation network. The module consists of 3 layers including dilation group-specific convolution with 1×2 kernel, temporal convolution with 9×1 kernel and attention pooling. A residual connection is added for each block. The dilation group-specific convolution module is designed for unit pairs (i.e., nodes, edges on graphs) relation extraction. By this module, the local feature aggregation of convolution is replaced by the global feature aggregation. The coordinates of different skeleton joints are allowed to explicitly interact with each other from the low layers of network. The number of joints captured by sensors or estimated by algorithms for skeleton-based action recognition is less than 30, e.g., 25 for NTU RGB+D, 18 for Kinetics, thus enumerating all the pair relations is affordable.

The conv-relation network is stacked with the blocks shown in Fig. 2. As shown in Table 1, there are 1 data batch normalization layer and 9 convolution relation modules. After that, global average pooling is performed and the final output is sent to a SoftMax classifier.

3.1. Dilation group-specific 1×2 convolution

As shown in Fig. 3, several groups of convolutions are used to enumerate all the unit pairs and get their relations. Different dilations are set in the different groups of convolutions, from 1 to R respectively. For the i_{th} group of convolutions, its dilation is set as i, where $i = 1, 2, \ldots, R$. That is, each group of convolutions has the same dilation as its group order. R denotes the number of groups of convolutions. When the inputs are nodes and edges on the skeleton graphs, the total number of groups of convolution R equals to $(V-1)$ and $(V-2)$ respectively, where V is the number of skeleton joints.

Given an input tensor with the shape of $N \times C \times T \times V$, it will be convolved by $(V-1)$ groups of 1×2 convolution for $(V-1)$ times. Normally, the convolution output corresponding to the j_{th} joint and the i_{th} group can be represented as

$$output_{ij} = conv_i(j, (j+i)\%N) \tag{1}$$

where $conv_i(x, y)$ denotes performing 1×2 convolution on the x_{th}, y_{th} spatial dimension of the input tensor with i_{th} group convolution, and then the result is assigned to $output_{ix}$. The modular operation in Eq. (1) is implemented as padding the i_{th} spatial dimension of the input tensor to the end of original input tensor incrementally before preforming the i_{th} group convolution. When the i_{th} group convolution is finished, its final result is

$$output_i = concat(conv_i(j, (j+i)\%N)), \quad j = 1, 2, \ldots, V \tag{2}$$

which is the convolution operation sliding on the spatial dimension of the input tensor. After finishing each group of 1×2 convolution, all outputs are concatenated or summed along the channel dimension.

$$output = f(output_i), \quad i = 1, 2, \ldots, V \tag{3}$$

where f denotes concatenating or summing the list of input tensors along the channel dimension. These two fusion methods have the same output dimension in our implementation. In the case of concatenation fusion, each group of dilation group-specific convolutions has output channel M/D, where M is the predefined number of output channel and D is the number of groups. So the total number of output channel after concatenation is $M/D \times D = M$. In the case of the summation fusion, each group of convolutions has output channel M, so the total number of output channel after summation is M. Typical values of M and D are 72 and 24 respectively, as shown in Table 1, layer *L1, L2* and *L3*.

Note that the above operations are performed on the V (spatial) dimension of the input tensor and shared on its T (temporal) dimension. Meanwhile, the temporal convolution network (TCN) with 9×1 kernel is applied to the T dimension of the input tensor and shared on its V dimension.

3.2. Attention pooling

Enumerating all the pair relations is likely to cause over-fitting due to the sensor noise of skeleton extraction. Furthermore, not all nodes and frames are needed to discriminate an action. Thus, three attention models are used to deal with such issues. A SE-block

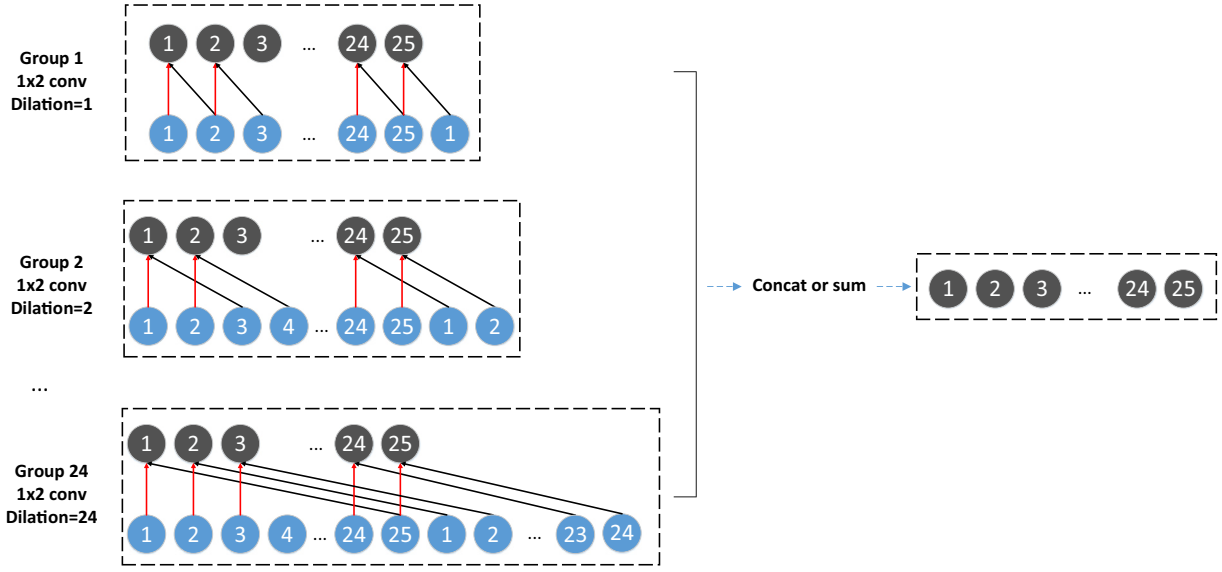

Fig. 3. An illustration of Dilation group-specific 1×2 convolution. In this example, there are 25 joints. Hence there are 24 groups of 1×2 convolution and each of them has dilation the same as its group order. The convolution outputs of all groups are concatenated or summed along the channel dimension to get the final output.

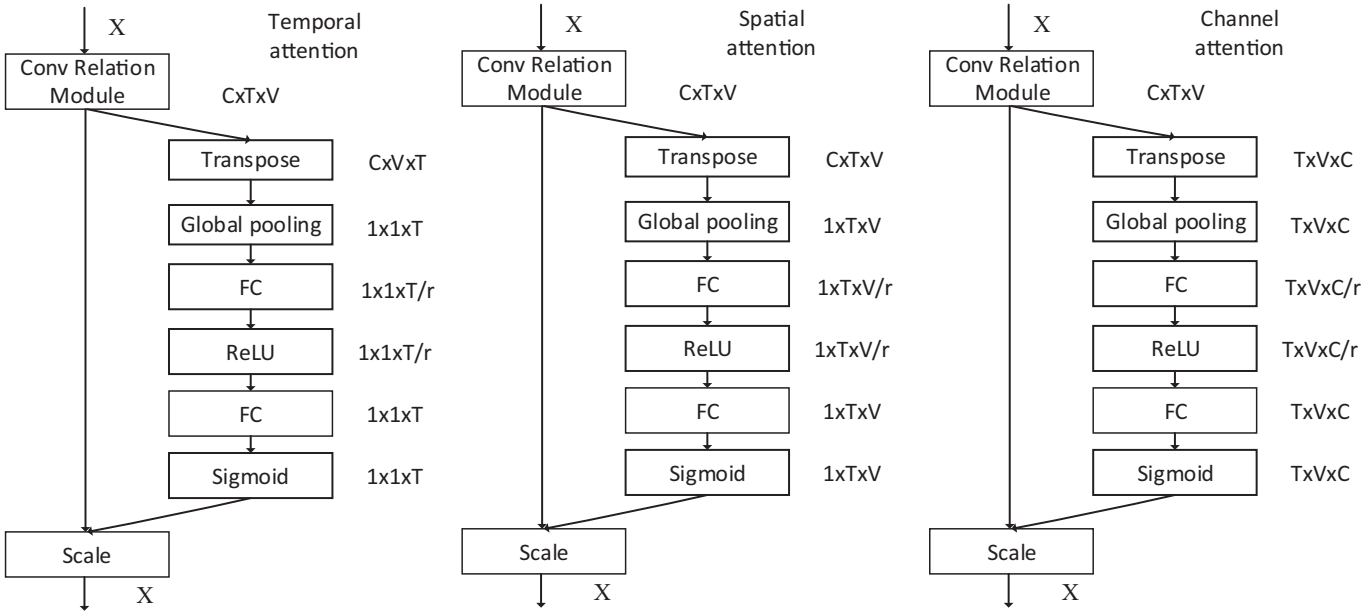

Fig. 4. Three attention modules, temporal attention, spatial attention, channel attention.

[33] is used to realize the attention module. SE-block is a bottleneck with 2 small fully connected layers whose final activation is sigmoid. It is originally used to adaptively recalibrate channel-wise feature responses. Here, it is employed to gate the channel, spatial and temporal dimension of an input tensor with the shape of $C \times T \times V$. Three attention models are shown in Fig. 4, where C denotes the channel dimension, T denotes the temporal dimension and V denotes the spatial dimension.

The operations of attention models consist of four steps, reshaping the input tensor, global average pooling, forward passing through two fully connected layers and channel-wise scaling.

Basic parameters of these operations are listed in Table 2. Take the temporal attention for example, the input tensor is firstly reshaped to $N \times T \times (V \times C)$. Then the global average pooling is performed on the dimension of $(V \times C)$. The obtained tensor is with the shape of $N \times T$. The final gating output is with the shape of $N \times T$. It denotes that the temporal attention is sample (video) specific. The similar analysis can also be applied to the spatial attention and channel attention.

Finally, the parallel multiplication is used to fuse these three attention models. As shown in Fig. 5, all attention weights are multiplied with the input tensor in a parallel fashion.

3.3. Input modalities

The main purpose of dilation group-specific convolution is to obtain all the joint interactions. However, the input information only relies on the single joint of skeletons. The natural connections

Table 2
Basic parameters of three attention models.

Attention	Reshape	Global average pooling dimension	Output shape	Characteristics
Temporal	$N \times T \times (V \times C)$	$(V \times C)$	$N \times T$	sample specific
Spatial	$(N \times T) \times V \times C$	C	$(N \times T) \times V$	sample and temporal specific
Channel	$(N \times T \times V) \times C$	Without pooling	$(N \times T \times V) \times C$	sample, temporal and spatial specific

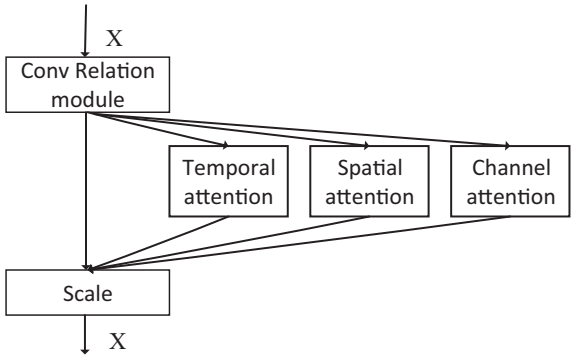

Fig. 5. Parallel attention fusion. All the attention weights are multiplied with the input tensor channel-wisely.

on the skeleton graphs are neglected, which may be important for capturing bone interactions. We also use the graph edges as the input of conv-relation network. Formally, each edge of the graph is the concatenation of its two nodes along the channel dimension. Then each edge is taken as the new spatial dimension of the input tensor with the shape of $N \times 2C \times T \times (V-1)$. Inspired from the optical flow stream in the two-stream networks [3], the frame differences are also considered as the input of conv-relation network to encode the explicit dynamics of skeletons. The frame differences are applied both on the nodes and edges respectively. That is, four kinds of inputs of the skeleton graphs, nodes, edges, node differences and edge differences, are taken as the inputs of conv-relation network. The final scores of four streams before softmax are averaged to get the final prediction.

4. Experiments

4.1. Datasets

NTU RGB+D [16]. This Kinect captured dataset is currently the largest dataset with RGB+D videos and skeleton data for human action recognition, with 56,880 video samples. It contains 60 different action classes including daily, mutual, and health-related actions. Each subject has 25 joints. The various setups of cameras, capturing views, and different facing orientations of the subjects, result in a great diversity of sample viewpoints. NTU contains two standard evaluations: Cross-Subject (CS) and Cross-View (CV). Specifically, Cross-Subject (CS) consists of 40 subjects that are randomly split into the training and the testing groups, while the training and the testing group of Cross-View (CV) come from the samples of cameras 2, 3 and camera 1 respectively. It is a challenging dataset for action recognition because of the large amount of videos, various subjects, and the difference of camera views. The top-1 recognition accuracy on both benchmarks is reported.

Kinetics [25]. Kinetics is a large-scale human action dataset which contains 300,000 videos clips in 400 classes. The video clips are sourced from YouTube videos. It only provides raw video clips without skeleton data. [21] estimate the location of 18 joints on every frame of the clips with the publicly available OpenPose toolbox [34]. Two persons are selected for multi-person clips based on the average joint confidence. Our model is evaluated on their released data. The dataset is divided into training set (240,000 clips) and validation set (20,000 clips). The top-1 and top-5 accuracies on the validation set are reported.

4.2. Experimental details

In order to make the model being insensitive to the initial position of an action, for each sequence, skeletons are normalized by subtracting the central joint of the first frame. The central joint is the average of 3D coordinates of the hip center, hip left and hip right. For the NTU RGB+D, average sequence length is about 80 and each sequence is divided into 32 segments. The number of segments is obtained by evaluating the model performance on a small validation set divided from training set. During training phase, we randomly sample a number from the range index of each segments and do bilinear interpolation. During testing phase, the center frame of each segment is sampled. For the Kinetics, its average sequence length is about 250. During training phase, a subsequence with the length of 150 is randomly cropped from the sequence. During testing phase, at most 2 subsequences are cropped from a sequence, since a sequence has a length of less than 300. The same bilinear interpolation is applied for Kinetics. The batch size is set to 64 and 4 GPUs are used for data parallel. The models are learned using stochastic gradient descent with an initial learning rate of 0.1. For the NTU RGB+D, the learning rate is decayed by 0.1 according to a schedule of [15, 60] and the total number of epochs is 90. For the Kinetics, the base learning rate is set to 0.2 and is decayed by 0.1 according to a schedule of [20, 50], the total number of epochs is 60. All experiments are conducted on PyTorch with 4 TITAN X GPUs. Code will be released.

The random view data augmentation is employed at the sequence level. Specifically, the skeleton is rotated around the *X, Y* and *Z* axis by some degrees which are generated randomly from −10 to 10. The probability of doing random view augmentation is 50%.

4.3. Ablation study

In this section, we verify the effectiveness of the proposed components in convolution relation network by three experiments on the cross view of NTU RGB+D dataset.

4.3.1. Convolutional relation module and Attention pooling

Firstly, the necessity of convolution relation module on the skeleton graphs is evaluated, as shown in Table 3. A baseline network is designed for comparison where all node interactions are replaced by 1×1 convolution, i.e. without joint interactions. For the conv-relation network, concatenation is adopted for relation channel fusion, shown as Concat relation. Concat relation outperforms the Conv1×1 by more than 3% in accuracy. It demonstrates that modeling joint interactions is important for skeleton-based action recognition. In addition, attention pooling constantly improves the performance of Concat relation and further parallel fusion leads to better performance. The relation channel fusion using summation achieves comparable performance as Concat relation.

Table 3
Recognition accuracy of different configurations of conv-relation module on the NTU RGB+D (cross view) dataset.

	Top1
Conv1 × 1	87.3%
Concat relation	91.1%
Concat relation, only with temporal attention	91.4%
Concat relation, only with channel attention	91.3%
Concat relation, only with spatial attention	91.5%
Concat relation, with all attention	91.8%
Sum relation, with attention	91.9%

Table 4
Recognition accuracy of conv-relation network with different inputs on the NTU RGB+D (cross view) dataset.

Input	Top1
Node	91.9%
Edge	91.9%
Node diff	92.1%
Edge diff	92.2%
Four Streams	94.5%

Table 5
Performance on the NTU RGB+D dataset. X-Sub and X-View are the cross subject split, cross view split of NTU RGB+D respectively.

	X-Sub	X-View
Lie Group [15]	50.1%	52.8%
H-RNN [35]	59.1%	64.0%
Deep LSTM [16]	60.7%	67.3%
PA-LSTM [16]	62.9%	70.3%
ST-LSTM+TS [17]	69.2%	77.7%
Temporal Conv [36]	74.3%	83.1%
C-CNN+MTLN [20]	79.6%	84.8%
ST-GCN [21]	81.5%	88.3%
Conv-Relation, node (ours)	82.5%	91.9%
Conv-Relation, edge (ours)	83.5%	91.9%
Conv-Relation, node, diff (ours)	82.3%	92.1%
Conv-Relation, edge, diff (ours)	84.0%	92.2%
Conv-Relation, Four Streams (ours)	86.2%	94.5%

By default, Sum relation with three attentions is used in the following experiments, unless otherwise specified.

4.3.2. Fusion of four streams

Another important improvement is the utility of different inputs, as shown in Table 4. The performances of using each kind of input data alone are compared, shown as Node and Edge, and its corresponding frame differences, shown as Node diff and Edge diff respectively. The performance of combining four streams is shown as Four streams. It demonstrates that combining the four kinds of data as input outperforms one stream based methods, which verifies the importance of natural edge connection and frame difference information for skeleton based action recognition.

4.4. Comparison with state-of-the-arts

As shown in Table 5, on NTU RGB+D dataset, our model is compared with Lie Group [15], Hierarchical RNN [35], Deep LSTM [16], Part-Aware LSTM (PA-LSTM) [16], Spatial Temporal LSTM with Trust Gates (STLSTM+ TS) [17], Temporal Convolutional Neural Networks (Temporal Conv.) [36], Clips CNN + Multi-task learning (C-CNN+MTLN) [20] and ST-GCN [21]. Our conv-relation network with single modality and four kinds of inputs as input outperforms previous state-of-the-art approaches on this dataset. Specifically, conv-relation network with four streams late fusion outperforms ST-GCN by more than 4% and 6% in top1 accuracy on cross subject and cross view respectively.

Table 6
Performance on the Kinetics dataset.

	Top1	Top5
Feature Encoding [22]	14.9%	25.8%
Deep LSTM [16]	16.4%	35.3%
Temporal Conv [36]	20.3%	40.0%
ST-GCN [21]	30.7%	52.8%
Conv-Relation, node (ours)	30.7%	52.99%
Conv-Relation, edge (ours)	31.5%	54.05%
Conv-Relation, node, diff (ours)	27.1%	49.44%
Conv-Relation, edge, diff (ours)	27.8%	50.1%
Conv-Relation, Four Streams (ours)	33.1%	55.8%

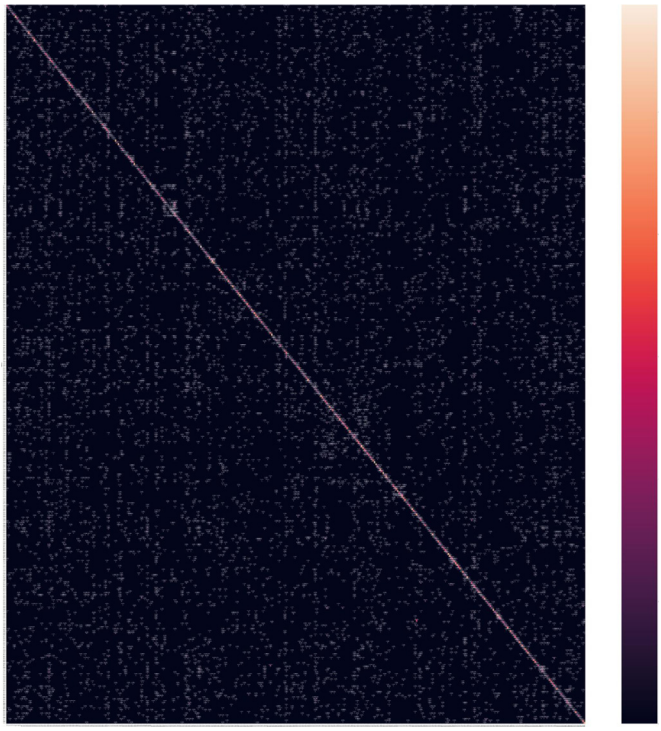

Fig. 6. Confusion matrix on Kinetics.

On Kinetics, we compare with four characteristic approaches for skeleton based action recognition. The first is the feature encoding approach on hand-crafted features [22], referred to as Feature Encoding in Table 6. The second is based on LSTM, i.e. Deep LSTM [16]. The third one is based on CNN, i.e. Temporal ConvNet [36]. The last one is based on GCN, i.e. ST-GCN [21]. In Table 6, the conv-relation network with single modality and four kinds of inputs as input outperforms previous representative approaches. Specifically, conv-relation network with the late fusion of four streams outperforms ST-GCN by more than 2% on the metrics of top1 and top5 accuracy.

4.5. Confusion analysis on Kinetics

Due to the low performance of our model on Kinetics, we show the confusion matrix of our model (four stream) in Fig. 6. For better illustration, we show the top-10 error most classes and top-10 right classified classes in Table 7 and Table 8 respectively. It can be observed from Tables 7 and 8 that our model behaves bad in classes which are full of scenes (e.g., sled_dog_racing, cleaning_pool), objects (e.g., counting_money, juggling_balls, playing_monopoly), and much better in classes only involving

Table 7
Top-10 error most classes on the Kinetics dataset. Error rate, %. The numbers inside the brackets of the third column are the error rates of gt classes being mis-classified.

GT classes	Error rate	Mis-classified most
sled_dog_racing	100.0	washing_hair(8.0)
cleaning_pool	100.0	flipping_pancake(10.0)
counting_money	100.0	Squat(10.0)
juggling_balls	100.0	baking_cookies(8.3)
headbutting	100.0	catching_or_throwing_baseball(6.0)
snatch_weight_lifting	100.0	long_jump(6.0)
bobsledding	100.0	Squat(8.0)
playing_monopoly	100.0	front_raises(12.5)
tying_bow_tie	100.0	making_snowman(26.0)
throwing_ball	100.0	catching_or_throwing_baseball(10.0)

Table 8
Top-10 right classified classes on the Kinetics dataset. Top1 accuracy, %.

GT classes	Top1
somersaulting	94.0
playing_cymbals	92.0
jumping_into_pool	86.0
eating_carrots	83.6
knitting	83.6
playing_poker	83.3
drop_kicking	83.3
jogging	82.0
busking	80.0
pushing_car	79.5

human-centric action (e.g., somersaulting, drop_kicking, jogging). Because the skeleton-based methods only use person-centric pose for action recognition. In these cases, they are complementary to those video-based methods.

4.6. Timing

Conv-relation network runs at ∼12 ms (∼80 fps) per clip (32 frames) on an Nvidia TitanX GPU. It is also fast to train. Training on the cross-view of NTU RGB+D takes 15 h in our 4-GPU implementation. Training on the Kinetics takes about 30 h.

4.7. Visualization

For a detailed comparison, we further investigate the per category change in accuracy. Fig. 7 shows the results, where the categories are sorted by accuracy gain. It is worth noting that the performance of most actions have been improved, especially for those involving joint interactions. For example, over 10% absolute improvement is observed for touch head, pointing to something with finger, making a phone call. Fig. 8 shows some example clips, for which our model predictions are true, while the baseline model (conv1x1) fails. For the actions without obvious joint interaction such as shake head, pat on back of other person, the relation messages of joints are not critical.

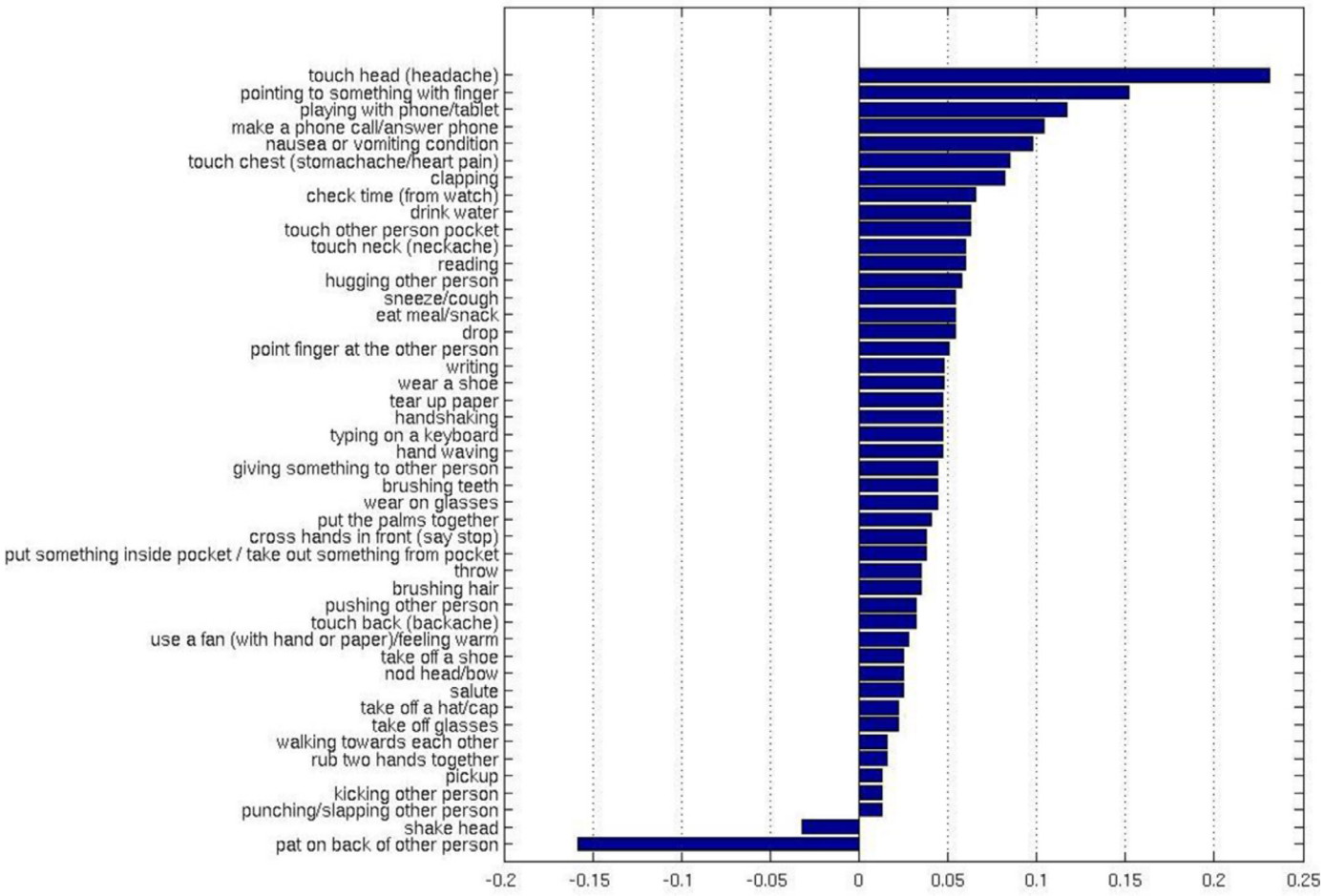

Fig. 7. Per-category change in accuracy of conv-relation over conv1x1 baseline on the NTU RGB+D dataset in the cross-view setting. For clarity only categories with change greater than 1% are shown.

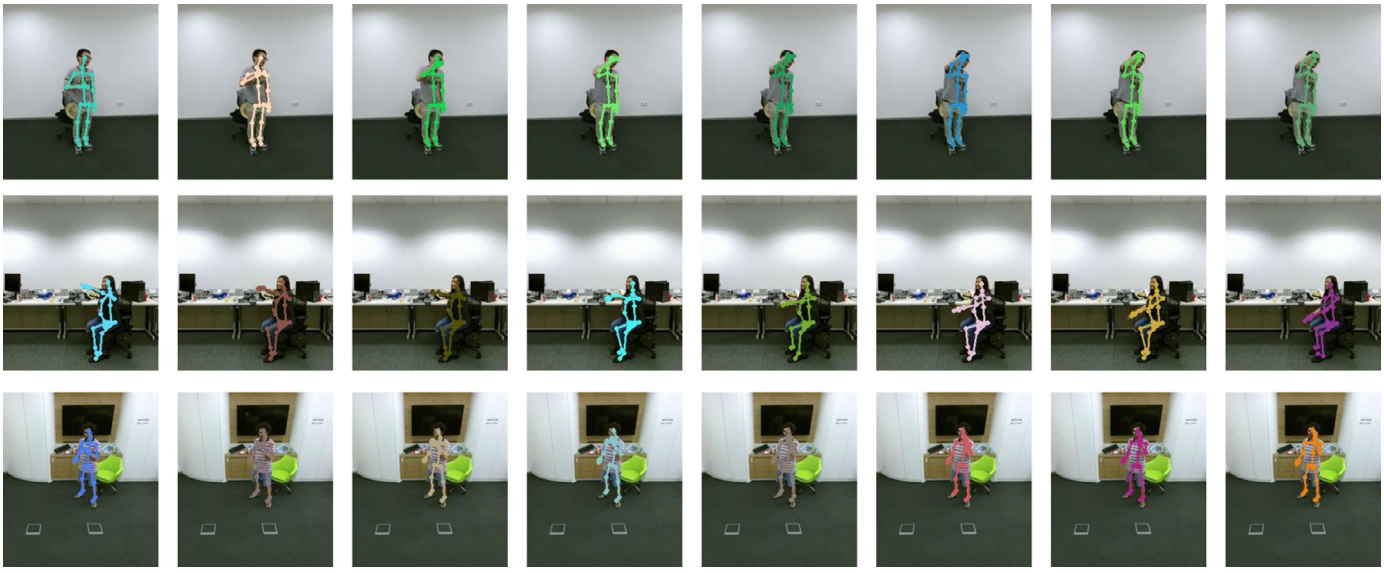

Fig. 8. Visualizing some example clips on NTU RGB+D cross-view, for which our model predictions are true, while the baseline model (conv1x1) fails. The clips of three action classes, i.e., *touch head (headache), pointing to something with finger, making a phone call* are shown from top to the bottom. The wrong predictions of the baseline model are *touch neck (neckache), check time, drop*, respectively.

5. Conclusion

In this paper, a convolutional relation neural network is proposed for skeleton-based action recognition. A convolutional-like relation module called dilation group-specific convolution is designed for modeling the potential relations of all the pairs of nodes and edges of the skeleton graphs. Attention pooling is used to enhance the relation module. The final prediction is obtained by the late fusion of four streams. Extensive experiments are conducted to show the effectiveness of our model. On two large datasets, Kinetics and NTU RGB+D, our method achieves competitive performance. Future work would include extending deformable convolution kernel [37] into skeleton-based action, which is perhaps a more efficient way of joints interaction modeling than conv-relation network. Another direction is combining skeleton data with RGB images in a multi-task manner, where the fine-grained feature of skeleton data and appearance feature of RGB images can complement each other. Moreover, combining the proposed method with data selection methods such as active learning [38], focal loss [39], online hard negative mining [40] is also a direction.

Declaration of Competing Interest

None.

Acknowledgments

This work is supported in part by the National Key Research and Development Program of China under Grant No. 2017YFB1300104, the National Natural Science Foundation of China under Grant No. 61773374, and in part by Project of Development In Tianjin for Scientific Research Institutes Supported By Tianjin government under Grant No. 16PTYJGX00050.

References

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, in: NIPS'12, Curran Associates Inc., USA, 2012, pp. 1097–1105. http://dl.acm.org/citation.cfm?id=2999134.2999257.

[2] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, in: NIPS, 2015.

[3] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 27, Curran Associates, Inc., 2014, pp. 568–576.

[4] C. Hong, J. Yu, J. Wan, D. Tao, M. Wang, Multimodal deep autoencoder for human pose recovery, IEEE Trans. Image Process. 24 (12) (2015) 5659–5670, doi:10.1109/TIP.2015.2487860.

[5] M. Yang, Y. Liu, Z. You, The euclidean embedding learning based on convolutional neural network for stereo matching, Neurocomputing 267 (2017) 195–200.

[6] J. Yu, X. Yang, F. Gao, D. Tao, Deep multimodal distance metric learning using click constraints for image ranking, IEEE Trans. Cybern. 47 (12) (2017) 4014–4024, doi:10.1109/TCYB.2016.2591583.

[7] Z. Yu, J. Yu, C. Xiang, J. Fan, D. Tao, Beyond bilinear: generalized multimodal factorized high-order pooling for visual question answering, IEEE Trans. Neural Netw. Learn. Syst. 29 (12) (2018) 5947–5959, doi:10.1109/TNNLS.2018.2817340.

[8] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal segment networks: Towards good practices for deep action recognition, in: Computer Vision – ECCV 2016, 2016, pp. 20–36.

[9] J. Zhu, W. Zou, Z. Zhu, End-to-end video-level representation learning for action recognition, in: ICPR, 2018.

[10] J. Zhu, W. Zou, Z. Zhu, Two stream gated fusion convnets for action recognition, in: ICPR, 2018.

[11] Y. Sun, X. Wu, W. Yu, F. Yu, Action recognition with motion map 3d network, Neurocomputing 297 (2018) 33–39, doi:10.1016/j.neucom.2018.02.028. http://www.sciencedirect.com/science/article/pii/S0925231218301632

[12] M. Huang, G.-R. Cai, H.-B. Zhang, S. Yu, D.-Y. Gong, D.-L. Cao, S. Li, S.-Z. Su, Discriminative parts learning for 3d human action recognition, Neurocomputing 291 (2018) 84–96, doi:10.1016/j.neucom.2018.02.056. http://www.sciencedirect.com/science/article/pii/S0925231218302029

[13] J. Wang, Z. Liu, Y. Wu, J. Yuan, Mining actionlet ensemble for action recognition with depth cameras, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1290–1297, doi:10.1109/CVPR.2012.6247813.

[14] M.E. Hussein, M. Torki, M.A. Gowayyed, M. El-Saban, Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations, in: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, in: IJCAI '13, AAAI Press, 2013, pp. 2466–2472. http://dl.acm.org/citation.cfm?id=2540128.2540483.

[15] R. Vemulapalli, F. Arrate, R. Chellappa, Human action recognition by representing 3d skeletons as points in a lie group, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 588–595, doi:10.1109/CVPR.2014.82.

[16] A. Shahroudy, J. Liu, T.-T. Ng, G. Wang, Ntu rgb+d: a large scale dataset for 3d human activity analysis, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[17] J. Liu, A. Shahroudy, D. Xu, G. Wang, Spatio-temporal lstm with trust gates for 3d human action recognition, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, Springer International Publishing, Cham, 2016, pp. 816–833.

[18] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, X. Xie, Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, in: AAAI'16, AAAI Press, 2016, pp. 3697–3703. http://dl.acm.org/citation.cfm?id=3016387.3016423.

[19] Y. Du, Y. Fu, L. Wang, Skeleton based action recognition with convolutional neural network, in: 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), 2015, pp. 579–583, doi:10.1109/ACPR.2015.7486569.
[20] Q. Ke, M. Bennamoun, S. An, F. Sohel, F. Boussaid, A new representation of skeleton sequences for 3d action recognition, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4570–4579, doi:10.1109/CVPR.2017.486.
[21] S. Yan, Y. Xiong, D. Lin, Spatial temporal graph convolutional networks for skeleton-based action recognition, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'18, 2018. AAAI Press
[22] B. Fernando, E. Gavves, M.J. Oramas, A. Ghodrati, T. Tuytelaars, Modeling video evolution for action recognition, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 5378–5387, doi:10.1109/CVPR.2015.7299176.
[23] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, N. Zheng, View adaptive recurrent neural networks for high performance human action recognition from skeleton data, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2136–2145, doi:10.1109/ICCV.2017.233.
[24] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, N. Zheng, View adaptive neural networks for high performance skeleton-based human action recognition, IEEE T. Pattern Anal. 41 (8) (2019) 1963–1978, doi:10.1109/TPAMI.2019.2896631.
[25] J. Carreira, A. Zisserman, Quo vadis, action recognition? a new model and the kinetics dataset, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4724–4733, doi:10.1109/CVPR.2017.502.
[26] L. Zhu, Y. Yang, Compound memory networks for few-shot video classification, in: The European Conference on Computer Vision (ECCV), 2018.
[27] L. Zhu, Z. Xu, Y. Yang, Bidirectional multirate reconstruction for temporal modeling in videos, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1339–1348, doi:10.1109/CVPR.2017.147.
[28] L. Zhu, Z. Xu, Y. Yang, A.G. Hauptmann, Uncovering the temporal context for video question answering, Int. J. Comput. Vis. 124 (3) (2017) 409–421, doi:10.1007/s11263-017-1033-7.
[29] A. Santoro, D. Raposo, D.G.T. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, T.P. Lillicrap, A simple neural network module for relational reasoning, CoRR (2017). abs/1706.01427.
[30] R. Palm, U. Paquet, O. Winther, Recurrent relational networks, in: Advances in Neural Information Processing Systems 31, 2018, pp. 3368–3378. URL: http://papers.nips.cc/paper/7597-recurrent-relational-networks.pdf.
[31] X. Wang, R. Girshick, A. Gupta, K. He, Non-local neural networks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 7794–7803, doi:10.1109/CVPR.2018.00813.
[32] B. Zhou, A. Andonian, A. Oliva, A. Torralba, Temporal relational reasoning in videos, in: European Conference on Computer Vision, 2018.
[33] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018.
[34] Z. Cao, T. Simon, S. Wei, Y. Sheikh, Realtime multi-person 2d pose estimation using part affinity fields, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1302–1310, doi:10.1109/CVPR.2017.143.
[35] Y. Du, W. Wang, L. Wang, Hierarchical recurrent neural network for skeleton based action recognition, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1110–1118, doi:10.1109/CVPR.2015.7298714.
[36] T.S. Kim, A. Reiter, Interpretable 3d human action analysis with temporal convolutional networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 1623–1631, doi:10.1109/CVPRW.2017.207.
[37] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 764–773, doi:10.1109/ICCV.2017.89.
[38] Y. Yang, Z. Ma, F. Nie, X. Chang, A.G. Hauptmann, Multi-class active learning by uncertainty sampling with diversity maximization, Int. J. Comput. Vis. 113 (2) (2015) 113–127, doi:10.1007/s11263-014-0781-x.
[39] T. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017.
[40] A. Shrivastava, A. Gupta, R. Girshick, Training region-based object detectors with online hard example mining, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

Jiagang Zhu received his B.S. degree from China University of Petroleum, China, in 2015. He is currently Ph.D. Candidate in Institute of Automation, Chinese Academy of Sciences, China. His research interests include deep learning and computer vision.

Wei Zou received his B.S. degree in Control Science and Engineering from Baotou University of Iron and Steel technology, China in 1997, the M.S. degree in Control Science and Engineering from Shandong University of Technology, China in 2000, and the Ph.D. degree in control science and engineering from Institute of Automation, Chinese Academy of Sciences, China in 2003. Currently, he is a professor at the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences. His research interests include intelligent robotics, visual servo, robot localization and navigation.

Zheng Zhu received his B.S. degree from Zhengzhou University, China, in 2014. He is currently a Ph.D. Candidate in Institute of Automation, Chinese Academy of Sciences, China. His research interests include computer vision, deep learning and robotics.

Yiming Hu received his B.S. degree from China University of Geosciences, Wuhan, China, in 2016. He is currently pursuing the Ph.D. degree in the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current research interests include deep learning and computer vision.