Author Guidelines for the British Machine Vision Conference

BMVC 2019 Submission # ??

Abstract

This paper presents a new skeleton-based action recognition method using a few shot learning. Consider the sequence data contain the temporal and spatial information; the proposed method encodes each of skeleton as an RGB image. Nothing more but a naïve normalization is engaged to each channel of the encoded skeleton image. In order to acquire the discriminative feature of skeleton image, dilated-dense layer is adopted in our model to both extend the receptive field of feature points and capture diversity representation of the skeleton image. After that, the prototypical network is introduced to recognize the specific action of the feature stands for. It can map the skeleton image into a metric space in which action classification can be performed by the nearest neighbor search. Benefited from the nature of the few shot learning, our model can be trained with only a few training samples. Moreover, it can adjust to the unseen classes that had not presented during the training phase with a few labeled samples from unseen classes. We evaluated our method with the seen and unseen class of samples, experiment result shows, the method achieved comparable performance on benchmark datasets.

Index Terms: skeleton sequence, dilated-dense layer, few shot learning

1 Introduction

Human action recognition has been widely researched for a few decades. A lot of recognition methods are developed to serve for entertainment, surveillance, and video analysis. At present, the action recognition algorithm has made a great step forward by the wave of the deep learning. However, these methods consume large-scale of training samples to capture the inner pattern of every action. As human action category is varied, the demand for training sample will explode. Besides, when new unseen actions are acceded to the classification task, the deep model need to be retrained for changing quantity of the action. A kind of few-shot (one-shot, zero-shot) learning methods are proposed to resolve these issues. Usually, these method can quickly accommodate the new classes by given a few of action samples from new class. Instance-based learning, Non-parametric method, meta-learning and metric learning [3] have played a significant role in the progress of this field. Mishra [12] present a generative framework for Zero-Shot or Few-Shot action recognition. Yang [19] introduce a new example-based action detection on the Matching Network. Nevertheless, seldom attention had been paid to apply the few shot learning model to skeleton sequence yet. To address this challenge, we propose an action recognition method based on the prototypical network. It can learns a metric space in which classification can be performed by computing distances to prototype representations of each class [17]. The performance of our method also benefits from convolution network, because we had engaged a serial of convolution layers to extract the feature of skeleton sequence. And The main contributions of this method are:

1.Only a few skeleton sequence samples are adequate to train an efficient action recognition model.

2.With a few support samples, the model is capable of recognizing the new action that had not seen during the training period.

3. Dilated-dense layer is embedded to our model, which can enhance the robustness and diversity of the skeleton feature representation.

The remainder of the paper is organized as follows. In Sec.2 we briefly review methods proposed to deal with skeleton-based action recognition. In Sec.3 a simple encoder without bells and whistle is introduced to encode the sequence as a skeleton image. Then a convolution network with dilated-dense layers is learned to map the skeleton image into a suitable metric space. Finally, we fully describe the training and recognition process of our action recognition method. In Sec.4 we report the experiments results on two datasets to show the performance of the method with a few-shot samples. In Sec.5 we discuss research directions in the future work.

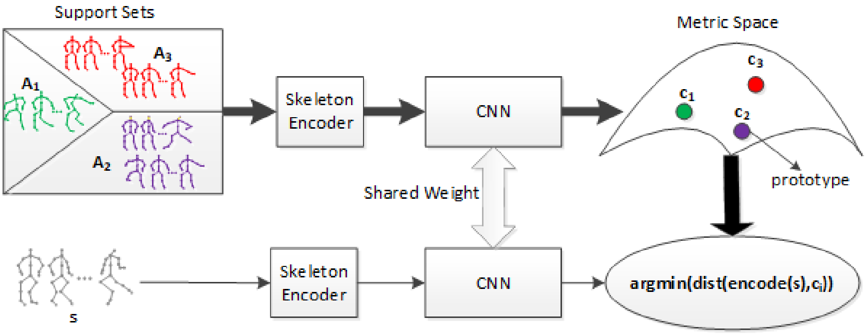

Figure 1: Proposed Model: The support sets (a few labeled sequences) are mapped into a metric space to estimate the prototype of each class. During the training phase, the convolution network with dilated-dense layers is learned as the mapping function. And for the inference, the unlabeled sample s is assigned to the class whose prototype has the shortest distance from itself in the metric space.

2 Related Work

Papers must be 9 pages in length, *excluding* the bibliography. Skeleton sequence data contains strong interpretive feature for the action recognition task. A rich of deep action recognition models had proposed on such kind of data. Specifically, these methods can be sumaried as three main streams. The first streams usually encode the skeleton sequence into a skeleton-image. Different skeleton-image encoders are proposed to capture the feature of actions. Wang [18] introduce a compact and effective method to encode spatiotemporal information carried in 3D skeleton sequences into a Joint Trajectory Maps (JTM). Pham [14] and Li [7] rearrange the pixels in RGB skeleton images to obtain a better representation of the movement. Furthermore, Liu [11] design a skeleton visualization method to represent a skeleton sequence as a series of visual and motion enhanced color images. Having encoded skeleton sequences into skeleton images, a variety of CNN networks(CNN [18], multi-scale CNN [6], DeepResidualNeuralNetworks [14], multi-stream CNN [11]) are constructed to classify the indeed action of the skeleton image.

The second steam is inspired by the RNN network, whose recurrent structure can boil a sequence data down into a high-level understanding [13]. For better performance, they tend to adopt the LSTM to process the skeleton sequence data. Zhu [20] introduce an end-to-end fully connected deep LSTM network with a designed regularization, through which can learn the co-occurrence feature of the skeleton joints. Based on LSTM, Liu [9] design a skeleton tree traversal method and a new gating mechanism to achieve a robust representation of the input sequence data. To further, Liu [10] also add attention ability to the LSTM network, which is capable of focusing on the informative joints of the skeleton. The last one is based on graph convolutional network [1], which can be applied directly to the raw skeleton data. Shi[15] present a novel two-stream nonlocal graph convolutional network for the recognition task. Li [8] proposed a spatio-temporal graph convolution approach with multi-scale kernels to recognize the actions. Si[16] proposes a novel Attention Enhanced Graph Convolutional LSTM Network, which can not only capture discriminative features in spatial configuration and temporal dynamics but also explore the co-occurrence relationship between spatial and temporal domains.

3 Proposed Method

Skeleton Image Encoder

It is common knowledge that a skeleton sequence can be represented as an RGB image. Consider an n frame skeleton sequence $s = \{fr^1, fr^2, \cdots fr^n\}$, each frame fr^i includes m joints $fr^i = \{j_1^i, j_2^1, \cdots j_m^i\}$ And each joint $j_t^i = \{j_{tx}^i, j_{ty}^i, j_{tz}^i\}$ is a 3-D coordinate point, which is corresponding to the RGB channel of a pixel in a skeleton image. Thus, the skeleton sequence S can be encoded as an $m \times n$ skeleton image. a sort of variant RGB encoder had proposed to achieve the translation-scale invariant representation of the skeleton sequences [5, 11]. But our model hasn't taken advantage of these mechanisms. Only a naive normalization, proposed by [2], is adopted.

$$p'_k = floor(255 * \frac{(p_k - p_k^{min})}{\max_k(p_k^{max} - p_k^{min})})$$

Where p_k^{max} and p_k^{min} are the maximum and minimum value of the k-th channel $(x;y;z)$ of a skeleton. In this way, each skeleton s can be encoded into a skeleton image I.

The Mapping Function

Having the skeleton sequence encoded as an RGB image, we will map it into a metric space via a mapping function $F_W : \mathbb{R}^D \to \mathbb{R}^M$. The function is the learnable part of our action recognition model. For the sake of processing the RGB skeleton image, we construct the mapping function as a convolution network. There are two things that we considered before build the network. One is that, unlike natural image, each pixel scattered in the image has equally interpretive meaning for the final decision, we tend to expand the size of the convolution kernel, so that is can convolve as much as pixels in the skeleton image. The other

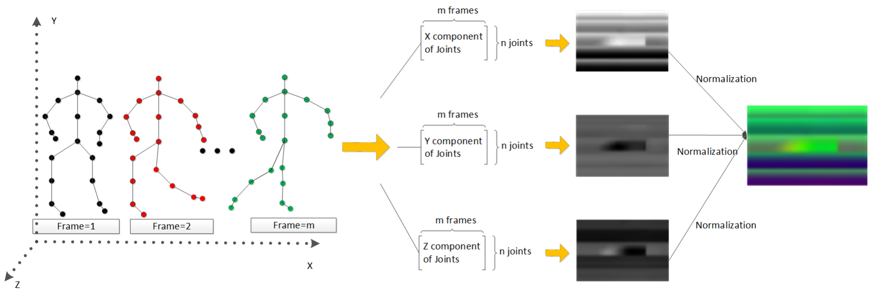

Figure 2: The workflow of the encoder. each component of the skeleton joints in a sequence is encoded as a channel of the RGB image. And the normalization is applied independently to each channel to generate the final skeleton image.

one is that, both the movement of the skeleton joints in temporal domain and the configuration of the joints in spatial space are the powerful features of action. For these reasons, we considered adding dilated-dense layers, whose dilated convolution kernel could enlarge the receptive field of the feature point and the densely connected layers can lead to a rich of configuration and movement feature representation.

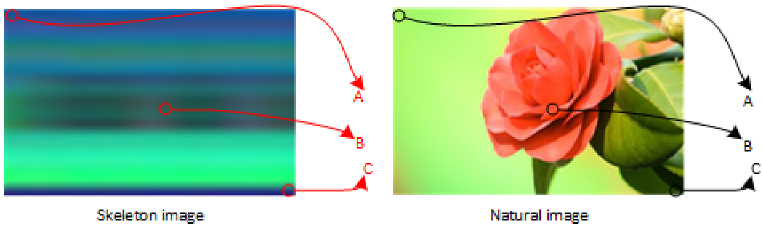

Figure 3: Each pixel represents a skeleton joint in skeleton image. Even the pixel around the corner(A,C) has equally importance to the one(B) located in the center. While things usually do not happen in natural image.

The Model

Figure 1 illustrate the whole structure of the action recognition model. our model includes two independent parts of parameters. One is the prototype of the classes $C_{support} = \{c_1, c_2, \cdots c_K\}$. it can be estimated by calculating the mean of the support set samples in metric space. The other one is the parameter of the mapping function F_W, which can be solved via SGD optimization. Given K classes of labeled skeleton samples, we randomly subsample a few of data from each class as the support sets. The k-th support set can be defined as:

$$S_k = s_1, s_2, \cdots s_{N_s}$$

The support sets will be mapped into a metric space by the convolution layers and the prototype of k-th class c_k is the mean of the mapped support samples that belong to it.

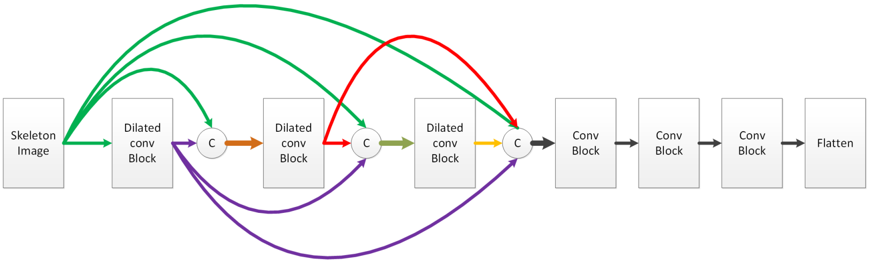

Figure 4: The architecture of the mapping function F_W. The first three dilated-conv-dense blocks are designed to enhance the robust representation of the skeleton image. Meanwhile, the later three convolution blocks are for projecting the feature into a discriminative space. And the final flatten layer will convert the feature into a one-dimension tensor. Weights and biases of the network are corresponding to the parameter of the mapping function.

$$c_k = \frac{1}{N_s} \sum_{s_i, y_i \in S_k} F_W(encode(s_i))$$

When feeding an unlabeled skeleton image s^*, the label of the image is decided by the nearest neighbour prototype, from whom it has the shortest distance.

$$\underset{i}{\mathbf{argmin}}(\|F_W(encode(s^*)) - c_i\|^2)$$

The inference phase of our model is similar to the nearest neighbor search. Algorithm 1 is a brief pseudo-code of the inference phase.

Algorithm 1: the inference process of action recognition model.

```
Input: $k$ classes of support set $S = \{S^1, S^2, \cdots S^k\}$, where each class of support
Output: the assignment of the unlabeled sample $y^*$
foreach i in K do
    $\mathbf{I^i} = \{I_1^i, I_2^i, \cdots I_N^i\} = \mathbf{encode}(S^i)$
    $\mathbf{F^i} = \{F_1^i, F_2^i, \cdots F^i\{N_s\}\} = \mathbf{F_w}(\mathbf{I^i})$
    $\mathbf{c_i} \leftarrow \mathbf{mean}(F^i)$
end
$I^* = \mathbf{encode}(a^*)$
$F^* = \mathbf{F_w}(I^*)$
$y^* = \underset{\mathbf{i}}{\mathbf{argmin}}\left(\|F_W(I^*) - c_i\|^2\right)$
```

The purpose of our training algorithm is to address the learnable parameter of w. To training the model, the labeled samples are divided into support and query sets. Support sets are used to estimate the prototype of each classes. While the query sets are used to calculate the loss of the model. For each labeled query sample (s^*, y^*), The cross-entropy loss of the prototypical model can be defined as:

$$loss(y = y^* | s^*, C_{support}, W) = \log(\frac{\exp(-d(F_W(encode(s^*)), c_{y^*}))}{\sum_{i=1}^{k} \exp((F_W(encode(s^*)), c_i)})$$

Figure 4 illustrates the detail computation of the cross-entropy loss of the model. Having the loss function defined, we optimize the parameters of the model in few shot fashion. For each training iteration, we randomly divided the labeled samples into the support and query set. With which the cross-entropy loss of the model will be calculated to updated the parameter of the network. A detail description of training phase is provided in algorithm 2.

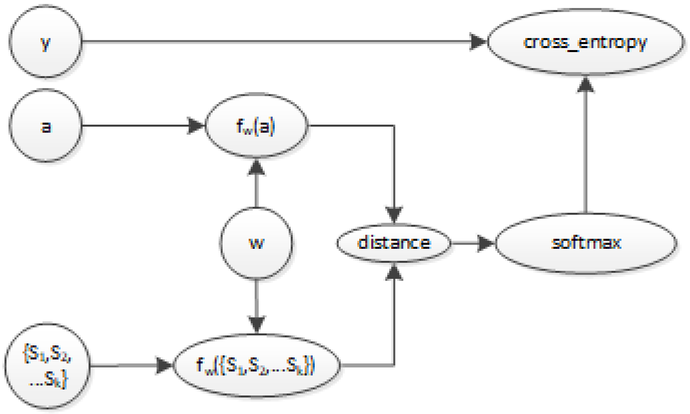

Figure 5: The architecture of the mapping function F_W. The first three dilated-conv-dense blocks are designed to enhance the robust representation of the skeleton image. Meanwhile, the later three convolution blocks are for projecting the feature into a discriminative space. And the final flatten layer will convert the feature into a one-dimension tensor. Weights and biases of the network are corresponding to the parameter of the mapping function.

4 Experiments

As mentioned before, the whole architecture of our model is provided in Figure 1. we construct the mapping function as convolution network, whose structure is illustrated in Figure 2. It composed of three dilated-conv-dense blocks, three convolution blocks and a flatten layer. We implemented our method on Tensorflow with GTX 1080 and evaluated it on two popular benchmarks. We train our model under few-shot setup. But it still achieves a comparable accuracy to the current existing fully supervised algorithm. Besides, we also show our model can adjust the unseen classes with a few labeled samples without retraining the parameter of the mapping function. A reproducible code is available in github. https://github.com/NanYoMy/human_action_recognition

UTD-MHAD

The UTD-MHAD dataset contains 27 classes of actions performed by 8 subjects (4 females and 4 males), Each subject repeated each action 4 times [2]. So, we get 32 samples per action. From each type of action, we select 8 sequences as the training set, and leave out 24 samples for testing. For evaluation, we randomly generated 4 support samples per class from the training set to estimate the prototypes of the model. Table 1 shows the recognition accuracy of different methods on UTD-MHAD dataset. Compare to other algorithm, our method only

Algorithm 2: the training process of action recognition model.

```
Input: $k$ classes of labeled sample $A = \{A^1, A^2, \cdots A^k\}$, where the i-th class of
       sample $A^i = \{a^i_1, a^i_2, \cdots a^i_N\}$ is labeled as $y^i$ and the $N_s$ and $N_q$ are the
       quantity of the support and query samples.
Output: $W$
Init $W$
repeat
    foreach i in K do
        $\mathbf{I^i} = \{I^i_1, I^i_2, \cdots I^i_N\} = \mathbf{encode}(S^i)$
        $\mathbf{F^i} = \{F^i_1, F^i_2, \cdots F^i\{N_s\}\} = \mathbf{F_w}(\mathbf{I^i})$
    end
    foreach i in K do
        $F^i_{supprot} \leftarrow RandomSample(F^i, N_s)$
        $F^i_{query} \leftarrow F^i - F^i_{support}$
        $c_i \leftarrow mean(F^i_{support})$
    end
    $C_{support} \leftarrow \{c_1, c_2, \cdots c_k\}$
    $J \leftarrow 0$
    foreach i in K do
        foreach $F_{it}$ in $F^i_{query}$ do
            $J \leftarrow J + \frac{1}{N_q * K} log\{p(y^i | F_it, C_{support}, W)\}$
        end
    end
    $W^{new} \leftarrow W^{old} - \varepsilon \bigtriangledown_W J$
until;
```

needs a quarter of samples for training without any data augmentation mechanism [14, 14], but still, we obtain an improvement of 1% over the state-of-the-art .

KARD

The KARD [4] dataset contains 18 actions, performed by 10 subjects and each subject repeated each action 3 times for creating a number of 540 sequences. Following the evaluation protocol in [14], the whole dataset is divided into three subsets. For each subset, one-third samples is used for training in our model and the rest is for testing. Our model requires less training examples than existing algorithm with a comparable accuracy. Furthermore, we also train our model on the whole KARD dataset. Although the quantity of the action is increased to 18, the model still achieves 99.38% recognition accuracy.

Few (one) shot action recognition

To investigate the performance of the model when dealing with the samples from the unseen classes, we must evaluate it with the action that have not presented during the training phase. For this purpose, the all class of the dataset is split into two disjoint sets as showed in Table 3. The parameters of the mapping function are optimized on training set. For evaluation,

Method	Accuracy
ELC-KSVD	76.19%
kinect & Inertial	79.10%
Cov3DJ	85.58%
SOS	86.97%
JTM	96.27%
TSIIM-MSDCNN[21]	86.97%
Our Model	**97.62%**

Table 1: comparison of different action recognition methods on UTD-MHAD.

Method	Exp1	Exp2	Exp3
Gaglio et al.	89.73%	94.50%	88.27%
Cippitelli et al.; P = 11	96.47%	98.27%	96.87%
Ling et al.	98.90%	99.60%	99.43%
DRNN[2]	99.87%	98.27%	96.87%
Our Model	**99.37%**	**99.72%**	**99.37%**
Our Model	**99.38%**		

Table 2: comparison of different action recognition methods on KARD.

we randomly select a few(one) support samples per class from the testing set to estimate the prototype of the classes. And the rest testing samples are used to validate the accuracy. Table 4 gives the performance of the model on the unseen action without retraining the parameter of the mapping function.

5 Conclusion

We present an action recognition method based on a few shot learning. It can achieve a comparable performance even with a few training samples. And we also demonstrate that our method can classify the unseen actions without retrain the parameter of the model. But, the ability of the model is limited, our method cannot be applied to the long-term skeleton sequence, which may contain a different type of continue actions. In the future, we will extend our action recognition task to the segmentation of the long-term skeleton sequence based on the few shot learning.

6 Acknowledgment

This work was supported in part by the Educational research projects for young and middle-aged teachers of Fujian under Grants JAT170796

	Training Set	Testing Set
UTD-MHAD	(1) right arm swipe to the left, (2) right arm swipe to the right, (3) right hand wave, (4) two hand front clap, (5) right arm throw, (6) cross arms in the chest, (7) basketball shoot, (8) right hand draw x, (9) right hand draw circle (clockwise), (10) right hand draw circle (counter clockwise), (11) draw triangle, (12) bowling (right hand), (13) front boxing, (14) baseball swing from right, (15) tennis right hand forehand swing, (16) arm curl (two arms), (17) tennis serve	(18) two hand push, (19) right hand knock on door, (20) right hand catch an object, (21) right hand pick up and throw, (22) jogging in place, (23) walking in place, (24) sit to stand, (25) stand to sit, (26) forward lunge (left foot forward), (27) squat (two arms stretch out)
KARD	1 Horizontal arm wave, 2 High arm wave 3 Two hand wave, 4 Catch Cap, 5 High throw, 6 Draw X, 7 Draw Tick, 8 Toss Paper, 9 Forward Kick, 10 Side Kick	1 Take Umbrella, 12 Bend, 13 Hand Clap, 14 Walk, 15 Phone Call, 16 Drink, 17 Sit down, 18 Stand up

Table 3: list of action in two disjoint sets (training set and testing set).

	UTD-MHAD	KARD
5 support samples	(1) 93.5%	95.88%
1 support samples	82.54%	87.3%

Table 4: the classification accuracies of the model with different number of support samples on UTD-MHAD and KARD datasets. All accuracy result is averaged over 1000 test episodes.

References

[1] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[2] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *2015 IEEE International conference on image processing (ICIP)*, pages 168–172. IEEE, 2015.

[3] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.

[4] Salvatore Gaglio, Giuseppe Lo Re, and Marco Morana. Human activity recognition process using 3-d posture data. *IEEE Transactions on Human-Machine Systems*, 45 (5):586–597, 2015.

Method	Frobnability
Theirs	Frumpy
Yours	Frobbly
Ours	Makes one’s heart Frob

Table 5: Results. Ours is better.

[5] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3288–3297, 2017.

[6] Bo Li, Yuchao Dai, Xuelian Cheng, Huahui Chen, Yi Lin, and Mingyi He. Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep cnn. In *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 601–604. IEEE, 2017.

[7] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Skeleton-based action recognition with convolutional neural networks. In *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 597–600. IEEE, 2017.

[8] Chaolong Li, Zhen Cui, Wenming Zheng, Chunyan Xu, and Jian Yang. Spatio-temporal graph convolution for skeleton based action recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[9] Jun Liu, Amir Shahroudy, Dong Xu, Alex C Kot, and Gang Wang. Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):3007–3021, 2018.

[10] Jun Liu, Gang Wang, Ling-Yu Duan, Kamila Abdiyeva, and Alex C Kot. Skeleton-based human action recognition with global context-aware attention lstm networks. *IEEE Transactions on Image Processing*, 27(4):1586–1599, 2018.

[11] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017.

[12] Ashish Mishra, Vinay Kumar Verma, M Shiva Krishna Reddy, S Arulkumar, Piyush Rai, and Anurag Mittal. A generative approach to zero-shot and few-shot action recognition. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 372–380. IEEE, 2018.

[13] Chris Olah and Shan Carter. Attention and augmented recurrent neural networks. *Distill*, 1(9):e1, 2016.

[14] Huyhieu Pham, Louahdi Khoudour, Alain Crouzil, Pablo Zegers, and Sergio A Velastin. Learning and recognizing human action from skeleton movement with deep residual neural networks. *international conference on pattern recognition*, 2017.

[15] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Non-local graph convolutional networks for skeleton-based action recognition. *arXiv: Computer Vision and Pattern Recognition*, 2018.

[16] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. *arXiv preprint arXiv:1902.09130*, 2019.

[17] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

[18] Pichao Wang, Zhaoyang Li, Yonghong Hou, and Wanqing Li. Action recognition based on joint trajectory maps using convolutional neural networks. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 102–106. ACM, 2016.

[19] Hongtao Yang, Xuming He, and Fatih Porikli. One-shot action localization by learning sequence matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1450–1459, 2018.

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