**Skeleton-Based Action Recognition Using Few Shot Learning**

**Wangbin Ding1,2, Shengqin Lin2, Zhenze Dai2**

1Shanghai JJ Software Technology Co 2Fuzhou Institute of Technology

1[yomynan@gmail.com](mailto:yomynan@gmail.com)

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 skeleton image feature, a serial of dilated-dense layers is adopted in our model to both extend the receptive field of feature points and capture diversity representation of the skeleton image. After that, a prototypical network is introduced to recognize the specific action of the feature stands for. The skeleton image feature will be mapped 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 labeled samples. Moreover, it could deal with the samples from unseen classes that have not presented during the training phase. We evaluated our method with the seen and unseen class of samples, experiment result shows, the method achieved comparable performance on benchmark datasets even with a few support samples.

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

# **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 model. However, these methods always consume large-scale datasets during the training period. As human actions are varied, the demand for train sample will explode. Besides, when samples with unseen labels acquired, the deep model needs to be re-trained to fit the changing distribution of the data label. A sort of few-shot (one-shot, zero-shot) learning methods are proposed to resolve these issues. Non-parameter model from nearest neighbor to metric learning [13] have played a significant role in the progress of this field. Mishra [14] present a generative framework for Zero-Shot or Few-Shot action recognition. Yang [15] introduce a new example-based action detection on the Matching Network. Nevertheless, seldom attention had 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 [19]. Because we had engaged a serial of CNN layers to extract the feature of skeleton sequence, our method also can benefit from the performance of CNN. And The main contributions of this method are：

1. Only a few skeleton sequences are adequate to train an efficient action recognition model.
2. With few support samples, it is enough to recognize the action that had never seen before.
3. Dilated-dense layer is embedded to extract the feature maps, which can enhance the robustness and diversity of the 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 the representation of the skeleton using dilated-dense layer is introduced. Then the inference and training algorithm of our few shot learning based model will be fully described. In Sec.4 we report the experiments results on a series of datasets to show the performance of the method. Finally, in Sec.5 we discuss research directions in the future work.

# **Related Work**

From the perspective of feature representation, skeleton-based action recognition methods can be divided into 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[1] introduce a compact and effective method to encode spatiotemporal information carried in 3D skeleton sequences into Joint Trajectory Maps (JTM). Pham[2] and Li[3] rearrange the pixels in RGB skeleton-images to obtain a better representation of the movement. Furthermore, Liu[4] 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[1], multi-scale CNN[5], DeepResidualNeuralNetworks[2], multi-stream CNN[4]) 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 []. For better performance, they tend to adopt the LSTM to process the skeleton sequence data. Zhu [6] 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 [7] design a skeleton tree traversal method and a new gating mechanism to achieve a robust representation of the input sequence data. To further, Liu [8] 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 [9], which can be applied directly to the raw skeleton data. Shi [10] present a novel two-stream nonlocal graph convolutional network for the recognition task. Li [11] introduce multi-scale graphical convolutional kernels to encode motion variations and input state for feature extracting. Si [12] propose 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.

# Proposed Method

## Skeleton Image Encoder

It is common knowledge that a skeleton sequence can be represented as an RGB image []. Consider an frame action , each frame contains joints . And each joint is a 3-D coordinate point, which is corresponding to the RGB channel of a pixel in a skeleton image. Thus, the skeleton sequence can be encoded as an skeleton image. a sort of variant RGB encoder had proposed to achieve the translation-scale invariant representation of the skeleton sequences [4 16,17]. But our method hasn’t taken advantage of these mechanisms. Only a naive normalization, proposed by [18], is adopted.

Where and are the maximum and minimum value of the k-th channel (x; y; z) of a skeleton. In this way, each action sample can be encoded into a skeleton image .



Figure 1: the workflow of extracting a skeleton image from a skeleton sequence

## Dilated-Dense Layers

The learnable part of our model is the mapping function . For the sake of processing the RGB skeleton image, we construct it as a CNN 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 could convolve as much as pixels in the skeleton image. The other one is that, both the movement of the skeleton joints in temporal domain and the configuration of the joints in spatial space are the significate 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 joints and movement feature representation.



Figure 1: Each pixel represent 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 didn’t usually happen in natural image.



Figure 2: CNN architecture of the mapping function : Three dilated-dense blocks are designed to enhance the robust representation of the skeleton image. The parameter of the mapping function is corresponding to the weights and biases of the CNN network.

## The Model

Given K classes of labeled skeleton image samples, we randomly subsample a few of samples from each class as the support sets. The k-th support set can be defined as：

The support sets will be mapped into a metric space by the convolution layers and the prototype of k-th class is the mean of the mapped support samples that belong to it. Figure 3 illustrate the structure of the action recognition model.

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

The inference phase of our model is similar to the nearest neighbor search.

The algorithm 1 is a brief pseudo-code of the inference phase.



Figure 3 : The structure of the skeleton-based action recognition model. Sample is assigned to the class whose prototype had the shortest distance from it

|  |
| --- |
| **Algorithm 1: the inference process of action recognition model.** |
| **Input：** classes of support set , where each class of support set contains labeled samples. is the unlabeled sample  **Output:** the assignment of the unlabeled sample  **Foreach i in**    **End** |

The parameter of our model can be divided into two subsets. One is the support sets , it can be provided by a few labeled samples in real recognition scenarios. The other one is the parameter of the mapping function , which can be solved via SGD optimization. The purpose of our training algorithm is to address the learnable parameter of . The training samples are divided into support and query sets. Support sets are used to estimate the prototype of each classes. While the query sets will be mapped into metric space to adjust the learnable parameter of . For each labeled query sample , The cross-entropy loss of the prototypical model can be defined as:

Figure 4 illustrates the detail computation of the cross-entropy loss of the model. Having the loss function defined, the parameters can be updated via the gradient descent method on it. We optimize the parameters of the model in few shot fashion[]. For each training iteration, we randomly divided the training samples into the support and query set. With which the cross-entropy loss of the model will be calculated. A detail description of training phase is provided in algorithm 2.



Figure 4: The computational graph of the model.

|  |
| --- |
| **Algorithm 2: the training process of action recognition model.** |
| **Input：** classes of action training set , where the i-th class of training sample is labeled as and contains samples. and are the number of support and query sample.  **Output:**    **Foreach i in**    **End**  **Repeat:**  **Foreach i in**        **End**  **Foreach i in**  **For each in**  **End**  **End**    **END** |

# **Experiments**

The whole architecture of our model is provide in Figure 3, as mentioned before, we construction the mapping function as a serial of dilated-dense layers. The structure of network is illustrated in Figure2. It composed of three dilated-dense blocks, three convolution blocks and a flatten layer. We implemented our method on Tensorflow with GeForce 920MX and evaluated it on two popular benchmarks. Our model requires less training examples than existing algorithm with a comparable accuracy. A reproducible piece of code is available in **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[20]. 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 needs a quarter of samples for training without any data augmentation mechanism [2,21], but it still outperform the other algorithms in Table1.

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

|  |  |
| --- | --- |
| **method** | **Accuracy** |
| ELC-KSVD | 76.19% |
| kinect & Inertial | 79.10% |
| Cov3DJ | 85.58% |
| SOS | 86.97% |
| JTM | 87.90% |
| TSIIM-MSDCNN[21] | 96.27% |
| Our Model | **97.62%** |

## KARD

The KARD[22] 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 [2], the whole dataset is divided into three subsets. For each subset, one-third samples is used for training and the rest is for testing. Furthermore, we also train our model on the whole KARD dataset. Although the output class number is increased to 18, the model still achieves 99.38% recognition accuracy.

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

|  |  |  |  |
| --- | --- | --- | --- |
| 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 | 100.0 | 99.93 |
| **Our Model** | **99.37%** | **99.72%** | **99.37%** |
| **Our Model** | **99.38%** | | |

## unseen classes

To investigate the performance of the model when dealing with the samples from the unseen class, 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. During the inference phase, we randomly select a few of support samples per class from the testing set to estimate the prototypes of the model. 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.

Table 3: list of action in two disjoint sets(Training set and Testing set).

|  |  |  |
| --- | --- | --- |
|  | 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 | 11 Take Umbrella, 12 Bend, 13 Hand Clap, 14 Walk, 15 Phone Call, 16 Drink, 17 Sit down, 18 Stand up |

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.

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

# Conclusion

We present an action recognition method based on a few shot learning. It is able to 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.

# Acknowledgment

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

# Reference

1 Wang, P., Li, Z., Hou, Y., & Li, W. (2016). Action Recognition Based on Joint Trajectory Maps Using Convolutional Neural Networks. *Acm on Multimedia Conference*.

2 Pham, Huy-Hieu, Khoudour, L., Crouzil, A., Zegers, P., & Velastin, S. A. (2018). Learning and recognizing human action from skeleton movement with deep residual neural networks.

3 Li, C. , Zhong, Q. , Xie, D. , & Pu, S. . (2017). Skeleton-based action recognition with convolutional neural networks.

4 Liu, M. , Liu, H. , & Chen, C. . (2017). Enhanced skeleton visualization for view invariant human action recognition. Pattern Recognition, 68, 346-362.

5 Bo, L., He, M., Cheng, X., Chen, Y., & Dai, Y. (2017). Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep cnn.

6 Zhu, W. , Lan, C. , Xing, J. , Zeng, W. , Li, Y. , & Shen, L. , et al. (2016). Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks.

7 Liu, J. , Shahroudy, A. , Xu, D. , Kot Chichung, A. , & Wang, G. . (2017). Skeleton-based action recognition using spatio-temporal lstm network with trust gates. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1-1.

8 Liu, J., Wang, G., Duan, L., Abdiyeva, K., & Kot, A. C. (2018). Skeleton-Based Human Action Recognition With Global Context-Aware Attention LSTM Networks. IEEE Transactions on Image Processing, 27(4), 1586-1599.

9 Bruna, J. , Zaremba, W. , Szlam, A. , & Lecun, Y. . (2013). Spectral networks and locally connected networks on graphs. Computer Science.

10 Shi, L. , Zhang, Y. , Cheng, J. , & Lu, H. . (2018). Non-local graph convolutional networks for skeleton-based action recognition.

11 Li, C. , Cui, Z. , Zheng, W. , Xu, C. , & Yang, J. . (2018). Spatio-temporal graph convolution for skeleton based action recognition.

12 Si, C., Chen, W., Wang, W., Wang, L., & Tan, T. (2019). An attention enhanced graph convolutional lstm network for skeleton-based action recognition.

13 Davis, J. V., Kulis, B., Jain, P., Sra, S., & Dhillon, I. S. (2007). Information-theoretic metric learning. Icml 07: International Conference on Machine Learning.

14 Mishra, A. , Verma, V. K. , Reddy, M. S. K. , Arulkumar, S. ,. , Rai, P. , & Mittal, A. . (2018). A generative approach to zero-shot and few-shot action recognition.

15 Yang, H., He, X., & Porikli, F. (2018). One-Shot Action Localization by Learning Sequence Matching Network. computer vision and pattern recognition.

16 Ke, Q., Bennamoun, M., An, S., Sohel, F. A., & Boussaid, F. (2017). A New Representation of Skeleton Sequences for 3D Action Recognition. computer vision and pattern recognition, 4570-4579.

17 Liu, M., Liu, H. W., & Chen, C. (2017). Enhanced skeleton visualization for view invariant human action recognition. Pattern Recognition, 68(68), 346-362.

18 Li, B., Chen, H., Chen, Y., Dai, Y., & He, M. (2017). Skeleton boxes: Solving skeleton based action detection with a single deep convolutional neural network. international conference on multimedia and expo, 613-616.

19 Snell, J., Swersky, K., & Zemel, R. S. (2017). Prototypical Networks for Few-shot Learning. neural information processing systems, 4077-4087.

20 Chen, C., Jafari, R., & Kehtarnavaz, N. (2015). UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. international conference on image processing.

21 Li, B., Dai, Y., Cheng, X., Chen, H., Lin, Y., & He, M. (2017). Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep CNN. international conference on multimedia and expo, 601-604.

22 Gaglio, S., Re, G. L., & Morana, M. (2015). Human Activity Recognition Process Using 3-D Posture Data. *IEEE Transactions on Human-Machine Systems*, 45(5), 586-597.