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

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

1Shanghai JJ Software Technology Co 2Fuzhou Institute of Technology

Abstract

This paper presents a new action recognition method using a few shot learning. Consider the skeleton sequence contain the temporal and spatial information; the proposed method encodes each of skeleton feature as an RGB image. Then a few shot learning method, prototypical network, is introduced to recognize the specific action of the RGB image stands for. Since a serial of dilated-dense layers is utilized to map the feature image into embedding space, our method could extend the receptive field of feature points to whole the skeleton sequence (RGB image). Experiment result shows, the method achieved state-of-the-art performance on a series of datasets. Besides that, it could deal with the samples from unseen classes that have not presented during the training phase with the help of a few support samples.

**Index Terms:** skeleton sequence, dilated filter, 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 the entertainment, surveillance, and video analysis. The action recognition algorithm has made a great step forward by the wave of the deep learning model. From the perspective of feature representation, skeleton-based action recognition methods could be divided into two main streams. One of the them usually encode the skeleton sequence into a feature image. then a CNN network is engaged to classify the indeed action of the movement. The other one is based on the RNN network, whose Recurrent structure can boil a sequence down into a high-level understanding.

these recognize means could divide into three types: First, local feature-based method. Such methods [1,2] extend 2D spatial feature descriptor into a 3D spatial-temporal local invariant feature descriptor (3D-Harris and 3D-Hessia) to capture the movement of the human body. Then a machine learning model will be engaged to recognize the indeed action of the movement. Second, global feature-based method. It usually needs a descriptor to capture the whole motion of the human body. Motion Energy Image (MEI) and Motion History Image (MHI) are two typical types of the global descriptor [3]. The last type method is Or it could interpret as a mixture of local and global feature descriptor method. Deep learning models like 3D-CNN[5] and LSTM[6] had achieved state of the art performance in the action recognition filed. But these methods always consume large-scale datasets during training time. As human actions are varied, the demand for train sample will explode. Besides, when samples with unseen labels gained, the deep model needs to be re-trained to fit the changing distribution of the data label.

A sort of Few-shot learning method is proposed to resolve these issues, Non-parameter model from nearest neighbor to metric learning [7] have played a significant role in the progress of this field. But seldom attention had paid to apply the few shot learning model to action recognition yet. For these reasons, we propose a method based on the few shot learning and global skeleton feature. Our method also could be considered as a local and global feature representation method. 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 could enhance the robustness and diversity(dense) 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 representation, few shot learning, and dilated-dense layer. Then the inference and training algorithm of our method will be fully described. In Sec.3 we report the experiments results on a series of datasets to show the performance of the method. Finally, in Sec.4 we discuss research directions in the future work.

# **Related Work**

## Skeleton Sequence Encoder

It is common knowledge that a skeleton sequence can be represented as an RGB image []. Consider an frame skeleton sequence action , each contains joints . And each is a 3-D coordinate point, which is corresponding to the RGB channel of a pixel in a feature image. Thus, the skeleton sequence can be encoded as an feature image. a sort of variant RGB encoder had proposed to achieve the translation-scale invariant representation of the skeleton sequences [8,9]. But our method hasn’t taken advantage of these mechanisms. Only a normalization, proposed by [10], is introduced.

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 feature image .



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

## Prototypical Network

A prototypical network learns a metric space in which classification can be performed by computing distances to prototype representations of each class[11]. Given K classes of labeled support sets, we define the k-th support set as：

each support set will be mapped by the function and the prototype of each class is the mean of the mapped support samples that belong to it.

when feeding an unlabeled feature image , the label of the image is decided by the nearest prototype.

## Dilated-Dense Layer

we construct the as a CNN network. Since each pixel scatted in the image feature represent a skeleton join, we trend to expands the size of the convolution kernel, so that is could involve as much as pixels in the feature image. Besides that, Both the movement of the skeleton joins between different frames and the distribution of the joins 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 when constructing the CNN network. Besides, the full connectivity of dense layers could lead to a variety of joins and movement features descriptors.



Figure 2: **CNN** architecture of the mapping function , The is corresponding to the weights and biases of the CNN network.

# Model

The output of the prototypical network is SoftMax over the distance; it is defined as a:

where, is the prototypical of the model, is the train feature image sample, is the parameter of the mapping function . Thus, the learnable part of this model can be solved via SGD optimization. For the sake of extending the receptive filed of the feature point and diversity feature representation, we had introduced the dilated-dense layers to construct the embedding function.

## Inference Phase

the inference phase of our model is similar to the nearest neighbor. Testing samples are divided into the support and test sets. samples from support set will be mapped into embedding space by . The prototype of our model is estimated by the mean of the different class of mapped support samples. When a sample from the testing set is feed into our model, it will be mapped by the same .then the assignment of the sample is depended on the model prototypes, from whom it had the shortest distance. 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. the unlabeled sample  **Output:** the assignment of the unlabeled sample  **Foreach i in**    **End** |

## Training Phase

The training algorithm of the skeleton-based action recognition method is provided in algorithm 2. The purpose of our training is to address the parameters of . the training samples are divided into support and query sets. Analogy to the inference, support sets is used to calculate the prototype of classes. The query sets, mapped into embedding space, will be used to adjust the parameter of . 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.



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:**    **Repeat:**  **Foreach i in**    **End**  **Foreach i in**        **End**  **Foreach i in**  **For each in**  **End**  **End**    **END** |

# **Experiments**

The trainable part of our model is the mapping function . as mentioned before, we construction the as a CNN network. The architecture of CNN is illustrated in Figure2. It composed of six layers: two dilated-dense layers, a convolution layer, a ReLU layer, a max-pool layer, a flatten layer. We implemented our method on Tensorflow with GeForce 920MX and evaluated it on a series of datasets.

## 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[12]. So, we get 32 samples for each action. From each class, we select 8(4 support, 4 query) samples for model training, and leave out 24 samples for testing. Table1 shows the recognition accuracy of different methods on UTD-MHAD dataset.

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

|  |  |
| --- | --- |
| metho | accuracy |
| ELC-KSVD | 76.19% |
| kinect & Inertial | 79.10% |
| Cov3DJ | 85.58% |
| SOS | 86.97% |
| JTM | 87.90% |
| TSIIM-MSDCNN[13] | **96.27%** |
| Our method | 94.3%~96.6% |

## work with 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 sequences that have not presented during the training phase. For this purpose, the UTD-MHAD dataset is split into two uncorrelated sets. The training set holds 10 classes of the actions from the whole dataset, and the remaining 17 is allotted to the inference set. During the training phase, we select ( support, query) samples from each type of action , with which the parameters of the model are optimized. During the inference phase, we randomly select 5 samples from each remained class to estimate the prototype of them. And the rest 27 samples are used to validate the accuracy of our model. In this situation, the sample labels are totally different between the training and inference sets. Table2 gives the recognition accuracy of our model with different training and testing settings.

Tables 2: the classification accuracies of the model with different training setting

|  |  |
| --- | --- |
| method | accuracy |
| support=5 n\_class =5 | 94% |
| support= n\_class = |  |
| support= n\_class = |  |
| support= n\_class = |  |

# Conclusion

We present an action recognition method based on a few shot learning. It could handle the unseen classes with a few support samples. But, the classification ability of the model is limited; it could only classify a sample into a fixed number of types. For future work, we will focus on to develop the model to work well with an arbitrary number of output classes. Besides, our method cannot be applied to the long-term skeleton sequence, which may contain a different type of actions. We will extend our action recognition task to the segmentation of the long-term action 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 Laptev, I. (2005). On Space-Time Interest Points. international conference on computer vision, 64(2), 107-123.

2 Willems, G., Tuytelaars, T., & Van Gool, L. (2008). An Efficient Dense and Scale-Invariant Spatio-Temporal Interest Point Detector. european conference on computer vision.

3 Bobick, A. F., & Davis, J. W. (2001). The recognition of human movement using temporal templates. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(3), 257-267.

4 Yilmaz, A., & Shah, M. (2005). Actions sketch: a novel action representation. computer vision and pattern recognition.

5 Zhao, R., Ali, H., & Der Smagt, P. V. (2017). Two-stream RNN/CNN for action recognition in 3D videos. intelligent robots and systems, 4260-4267.

6 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.

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

8 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.

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

10 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.

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

12 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.

13 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.