**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. considered 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, our method achieved state-of-the-art performance on a series of datasets. Besides that, our method is able to deal with new classes that have not presented during the training phase with 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 the field of entertainment, surveillance, and video analysis.

From the perspective of feature representation, these recognize methods could divide into three types: first, local feature-based method. Such methods design a set of 3D local invariant descriptor (3D-Hessia and 3D-Harris) to track 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. The last type method is developed by the latest wave of the deep learning model. Or it could interpret as a mixture of local and global feature descriptor method. deep learning models like 3D-CNN[1] and LSTM[2] 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 be exploded. Besides, when feeding a new class of sample, the deep model needs to be re-trained to fit the changing distribution of the data.

To address these difficulties, a sort of Few-shot learning method is proposed. Non-parameter model from nearest neighbor to metric learning [3] have played a great 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 could also be considered as a kind of global feature representation method. And The main contributions of this method are：

1. Only a few skeleton sequences are needed 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 of our 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 4. we report the experiments results on a series of datasets to show the performance of the method. Finally, in Sec.5 we discuss propose research directions in the future.

# **Proposed Method**

## Skeleton Sequnce 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 [4][5]. But our method hasn’t taken advantage of these mechanisms. Only a normalization, proposed by [6], 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[7]. given K classes of labeled support sets, we define the i-th support set as：

each support will be mapped by a function and the prototype of each class is the mean of the mapped support sample that belongs to it.

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

## Dilated-Dense Llayer

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 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.



Figure2． **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 descript of the inference phase.



Figure3. The structure of the skeleton-based action recognition model. Sample x is assigned to the class, who 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 illustrate the detail computation of the cross-entropy loss of the model. Having the loss defined, the parameters can be updated by the gradient descent method on it.



Figure 4. The computational graph of 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**

sThe 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 flattened layer. We implemented our method on Tensorflow with GeForce 920MX and evaluate 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[]. 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.

|  |  |
| --- | --- |
| method | accuracy |
| ELC-KSVD | 76.19% |
| kinect & Inertial | 79.10% |
| Cov3DJ | 85.58% |
| SOS | 86.97% |
| JTM | 87.90% |
| TSIIM-MSDCNN | **96.27%** |
| Our method | 96% |

## Deal with unseen classes

To verify the ability of dealing with the unseen classes, the whole dataset is divided into two unrelated training and inference sets. the training set contains 10 classes of the actions, and the inference set contains reminded 17 classes. During the training phase, we select ( support, query) samples from each class of action , with which the parameters of the model are optimized. During the inference phase, we randomly select 5 samples from each remined class to calculate the prototype of them . and the rest 27 samples is used to verify the accuracy of our model. In this scenario, the training and inference samples are totally from different actions. Table2 shows the recognition accuracy of our model with different train setting.

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

# Conclusion

We present an action recognition method based on a few shot learning. It is able to 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 classes. For future work, we will focus on to develop the model to work well with an arbitrary number of output classes. In addition, 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.

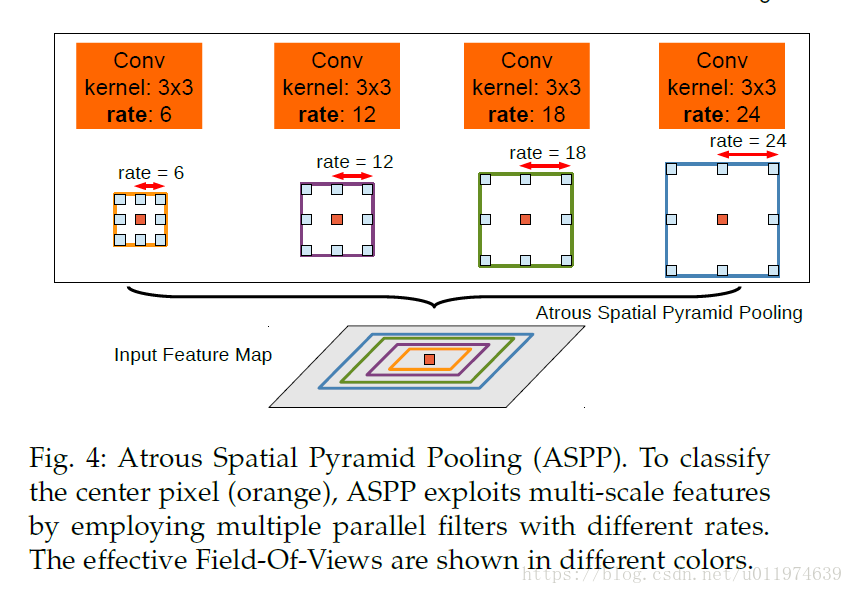
# Acknowledgment

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

# reference

1. 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.
2. 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.
3. Davis, J. V., Kulis, B., Jain, P., Sra, S., & Dhillon, I. S. (2007). Information-theoretic metric learning. *Icml 07: International Conference on Machine Learning*.
4. A New Representation of Skeleton Sequences for 3D Action Recognition
5. Enhanced skeleton visualization for view invariant human action recognition
6. SKELETON BOXES: SOLVING SKELETON BASED ACTION DETECTION WITH A SINGLE DEEP CONVOLUTIONAL NEURAL NETWORK
7. Prototypical Networks for Few-shot Learning
8. Skeleton Based Action Recognition Using Translation-Scale Invariant Image Mapping And Multi-Scale Deep CNN{论文的对比}

# Other



The atrous spatial pyramid pooling. To classify the center pixel, ASPP exploits multi-scale features by employing multiple parallel filter with different rates. The effective of Field-of-view are shown in different colors.