Skeleton-based one-shot action recognition on graph neural network

Hao Chen*, Yuzhuo Fu, Ting Liu
School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China

ABSTRACT

In our work, we implemented one-shot action recognition that using the skeleton data. In terms of data preprocessing, we used the form of mapping skeleton sequence coordinates into signal images. In the feature extraction module, we used feature extraction based on resnet18. In the few-shot learning part, we adopted the metric neural network model based on graph neural network. Finally, the leading accuracy is realized on the ntu-rgbd120 one-shot dataset.

Keyword: One-shot learning, metric learning, graph neural network

1. INTRODUCTION

The research in action recognition in computer vision mainly focuses on one-shot action recognition, pedestrian re-recognition, face recognition and so on. These methods have a good effect on one-shot action recognition, but only focus on a single pattern such as images or skeleton sequences. We consider using a signal-level representation that allows flexible encoding of signals into images and fusion of signals from different sensor modes.

There are several benefits of making representation for encoding skeleton data into signal images. First, as long as the sensor generates multivariable higher modal data forms and sequences, it allows generalization across different sensor modes. Secondly, the representation of similar images makes the using of a better classification architecture which performs well becomes possible and feasible.

In our research, the signals come from the three-dimension skeleton sequences which is collected by the RGB-D video camera, and other video data measurements. For details on signal representation, see Section 3.

As few-shot learning algorithm needs to make full use of the relationship between the support set data and the query set data[1], using GNN absolutely has great potential to come up with the solution of few-shot learning. Garcia and Bruna[2] construct a graph in which the support set and the query set are closely connected. We have performed experimental verification on the model on a small number of shot image classifications, matching the most advanced performance with fewer parameters.

Our method achieves the first use of one-shot action classification based on GNN structure on skeletal data processing. And we achieved the best accuracy on 5 way-1 shot task on the dataset NTU-RGBD120 one shot.

2. RELATED WORK

We briefly outline the methods related to representation of data and one-shot recognition methods and graph neural network in general.

2.1 Image representation

Our method is based on the image representation of the sensor sequence[3]. Wang et al.[4] encoded the joint trajectory image into an image through three different spatial perspectives. Liu et al.[5] proposed a combination of bone visualization methods and jointly trained these methods on multiple streams.

2.2 Few-shot classification

A matching network[1] created a new type of method which can use attention algorithm to calculate the classifier of the nearest neighbor that can perform calculus and a prototypical network[6] enlarges it by making the definition of prototypes as the average value of embedded support examples for each class. Meta-LSTM[7] updates the model by using LSTM,

*haolianyi@sjtu.edu.cn

International Conference on Computer Application and Information Security (ICCAIS 2021),
edited by Yingfa Lu, Changbo Cheng, Proc. of SPIE Vol. 12260, 122600E · © The Authors.
Published under a Creative Commons Attribution CC-BY 3.0 License · doi: 10.1117/12.2637828

making its model's parameters to be the hidden states. MAML[8] only concentrates on the initial parameters' initial values and simply uses SGD. Reptile[9] uses only first-order gradients.

2.3 Graph neural network

Gori et al.[10] firstly proposed the graph neural network, which acts like a trainable cyclic message transfer whose fixed points can adjust differently. Li et al.[11] further proposed a method using gated loop units and advanced optimization techniques. Kipf and Welling[12] groundbreakingly applied semi-supervised learning of graph structure data, which is scalable. They[2, 13] have applied the GNNs to few-shot learning and labeled the framework on the node level.

3. APPROACH

To solve the action recognition task across a series of sensor modalities we consider the action recognition problem on a signal level. The signals are encoded in a differentiated image representation. The representation of similar images allowed direct adaptation of the established image classification framework to extract features. To solve the one-shot problem, we apply a metric neural network based on graph neural network. An illustration of our approach is given in Figure 1.

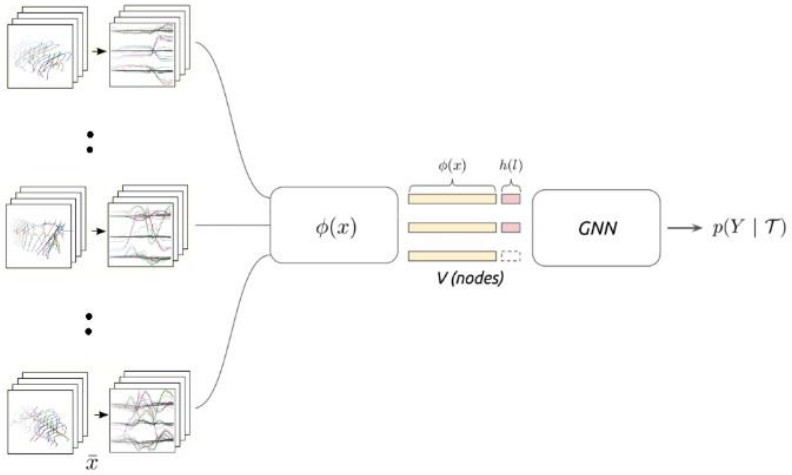

Figure 1. The framework of one-shot action recognition. $\Phi(x)$ denotes the embedding network (Resnet18 or Resw).

3.1 Problem definition

We consider the one-shot action recognition problem as a few-shot metric learning problem. First, we encode the sequence of actions. Convert the signal level to an image representation. The input in our example is a signal matrix $S \in \mathbb{R}^{N \times M}$ where each row vector represents a discrete 1-dimensional signal and each column vector represents a sample of all sensors at one specific time step. The matrix is transformed to an RGB image $I \in \{0,...,255\}^{H \times W \times 3}$ by normalizing the signal length M to W and the range of the signals to H. The identity of each signal is encoded in the color channel. So our dataset consist of $D = \{(I_i, y_i)\}_{i=1}^{N}$ of N training images $I_{1,...,N}$ with labels $y_i \in \{1,...,C\}$. We consider input-output pairs $(\mathcal{T}_i Y_i)_i$ drawn from a distribution P of partially-labeled image collections

$$\mathcal{T} = \left\{ \left\{ (x_1, l_z), ..., (x_s, l_s) \right\}, \left\{ \tilde{x}_1, ..., \tilde{x}_r \right\}, \left\{ \bar{x}_1, ..., \bar{x}_t \right\}; l_i \in \{1, K\}, x_i, \tilde{x}_1, \bar{x}_j \sim \mathcal{P}_l\left(\mathbb{R}^N\right) \right\} \tag{1}$$

and $Y = (y_1, ..., y_t) \in \{1, K\}^t$ for arbitrary values of s, r, t and K. s is the number of labeled samples, r is the number of unlabeled samples and t is the number of samples to classification. K is the number of classes. $\mathcal{P}_l(\mathbb{R}^N)$ denotes a class-specific image distribution over $\mathbb{R}^N$. Given a training set $\{(\mathcal{T}_i, Y_i)_i\}_{i \le L}$, we consider the standard supervised learning

objective $\min_{\mathfrak{E}} \frac{1}{L}\sum_{i\le L}\mathcal{L}\left(\Phi\left(\mathcal{T}_i;\mathfrak{E}\right),Y_i\right)+\mathcal{V}\left(\mathfrak{E}\right)$, using the model $\Phi\left(\mathcal{T};\mathfrak{E}\right)=p\left(Y\mid\mathcal{T}\right)$ specified in Section 5 and is $\mathcal{V}$ a standard regularization objective.

When $r=0,t=1$ and $s=qK$, there is a single image in the collection with unknown label. In addition, if each label occurs exactly q times, this setting is called q-shot, K-way learning. And in our work, we only focus on one-shot problem.

3.2 Representation

Our method is based on discernible image representation. Therefore, we propose a novel and compact signal level representation. Multivariable signals or higher-level feature sequences are recombined into a 3-channel image. Each row of the result image corresponds to one joint, and each channel corresponds to one sample in the sequence. The color channels, red, green and blue, represent respectively the signals' *x*-, *y*- and *z*-values. The resulting images are normalized to the range of 0-1.

3.3 Feature extraction

The feature extraction module of our model mainly used a Resnet network-based structure. We consider that after the joint training is implemented, the feature extraction module can be strongly combined with few-shot classifier module. We used Resnet18 and Resw (the network optimized by our approach) for feature extraction.

On one hand, we apply a 1×1 convolution layer before the 2 layers of 3×3 convolution in every main stage of Resnet18, i.e., the residual block of Resnet. With this optimization, the nonlinearity of the model can be increased before each residual calculation module in the network is down-sampled, and the results show that the effectiveness of feature extraction is improved. On the other hand, we also postponed the down-sampling of the residual module by adjusting the stride of the two-layer convolution.

After the last feature layer, we use a two-layer perceptron to transform the features into the embedding size. The embedding is refined by the metric learning approach.

3.4 Metric network

The goal of few-shot learning is to propagate label information from labeled samples to unlabeled query images. This propagation of information can be formalized as a posterior inference over a graphical model determined by the input images and labels. The input $\mathcal{T}$ contains a collection of labeled and unlabeled images. This information dissemination can be formalized as a posteriori reasoning of the graphical model determined by the input image and the label. We associate $\mathcal{T}$ with a fully-connected graph $G_{\mathcal{T}}=(V,E)$ where nodes $v_a\in V$ correspond to the images present in $\mathcal{T}$ (both labeled and unlabeled).

In the graph neural network part, given an input signal $F\in\mathbb{R}^{V\times d}$ on the vertices of a weighted graph G. We apply a family $\mathcal{M}$ of graph intrinsic linear operators that act locally on this signal. And the adjacency operator A: $F\mapsto A(F)$ where $(AF)_i:=\sum_{j\sim i}w_{i,j}F_j$, with $i\sim j$ if $(i,j)\in E$ and $w_{i,j}$ is associated weight. A GNN layer Gc($\cdot$) receives as input a signal $x^{(k)}\in\mathbb{R}^{V\times d_k}$ and produces $x^{(k+1)}\in\mathbb{R}^{V\times d_{k+1}}$ as

$$x_l^{(k+1)}=Gc\left(x^{(k)}\right)=\rho\left(\sum_{B\in\mathcal{M}}Bx^{(k)}\epsilon_{B,l}^{(k)}\right),l=d_1d_2...d_{k+1} \tag{2}$$

where $\mathfrak{C}=\left\{\epsilon_1^{(k)},...,\epsilon_{|\mathcal{M}|}^{(k)}\right\}_k,\epsilon_B^{(k)}\in\mathbb{R}^{d_k\times d_{k+1}}$, are trainable parameters and $\rho(\cdot)$ is a point-wise non-linearity. Based on this basic formula, we did some exploration. Particularly, inspired by message-passing algorithms, we consider a Multilayer Perceptron stacked after the absolute difference between two vector nodes.

$$\psi_{\tilde{c}}\left(x_i^{(k)},x_j^{(k)}\right)=MLP_{\tilde{c}}\left(abs\left(x_i^{(k)}-x_j^{(k)}\right)\right) \tag{3}$$

where ψ is a symmetric function parametrized with a neural network, etc. And ψ is a metric, which is learned by nonlinear combination of the absolute difference between the individual features of the two nodes. Then, we normalize the trainable adjacency to a random kernel by using a softmax along each row. By adding the edge feature kernel $A^{(k)}$ into the generator family $\mathcal{M}$ and applying (2), the update rule of the node feature is obtained.

Adjacency learning is especially important in the following applications: the input set is considered to have a certain geometric structure, but the metric is unknown a priori, as in our case. In a general graph, the depth of the network is selected in the order of the diameter of the graph, so that all nodes obtain information from the entire graph. However, in our context, since the graphics are closely connected, depth is simply interpreted as giving the model greater expressive power.

3.5 Composition of initial node features

The input set $\mathcal{T}$ is mapped into node features, as shown below. For signal images $x_i \in \mathcal{T}$ with known label l_i, the one-hot encoding of the label is concatenated with the embedding features of the image at the input of the GNN.

$$x_i^{(0)} = \left(\varphi\left(x_i\right), h\left(l_i\right)\right) \tag{4}$$

where φ is the Resnet embedding network and $h(l) \in \mathbb{R}_+^K$ is a one-hot encoding of the label.

4. TRAINING

Our model was aimed to predict the label Y corresponding to the signal image to be classified $\bar{x} \in \mathcal{T}$, with node $*$ in the graph. Therefore, the last layer of GNN is softmax, which maps node features to K-simplex. Then we apply the Cross-entropy loss evaluated at node $*$:

$$\mathcal{L}\left(\Phi\left(\mathcal{T};\mathfrak{E}\right), Y\right) = -\sum_k y_k \log P\left(Y_* = y_k \mid \mathcal{T}\right) \tag{5}$$

5. EXPERIMENT

We applied our methods to two datasets. First, we used skeleton sequences from the NTU RGB+D 120[14] as our large-scale one-shot recognition dataset. And we also used the UTD-MHAD[15] dataset to prove the generalization of our model.

5.1 Datasets

NTU RGB+D 120: The NTU RGB+D 120[14] dataset is a wide-ranging action recognition dataset involving RGB+D mode data and skeleton sequences data. The dataset consists of 114,480 sequences containing 120 action classes from 106 subjects in 155 different views. This article uses the one-shot standard given by the author of the dataset. The categories of these standards are divided into two parts: training set and test set. The categories contained in the training set and the test set do not overlap. There are 100 categories for training and 20 categories for testing. A1, A7, A13, A19, A25, A31, A37, A43, A49, A55, A61, A67, A73, A79, A85, A91, A97, A103, A109, A115 are previously unseen.

b) UTD-MHAD: The UTD-MHAD[15] is a dataset containing 27 actions of 8 individuals performing 4 repetitions each. RGB-D image source data, skeleton sequences and inertial data are involved. The RGB-D camera is placed frontal to the demonstrating person. This article used 23 classes for training and other classes for testing. Besides the 23/4 training/testing setting, we also testing the setting of 19/8 and 15/12 training/testing setting. Other works even applied the setting of 11/16 and 7/20, we believe our three setting is Forcefully and credibly.

5.2 Results

On the NTU RGB+D 120 one-shot dataset we compare against APSR[14] and SL-DML[3]. Table 1 shows the results with a training set size of 100 action classes and a test set size of previously unseen 20 action classes. Our method (using our own feature extraction backbone Resw) is 1.2% higher than the best method (SL-DML) before. Table 2 shows results for an increasing amount of training classes (100 training classes and 20 test classes are considered as the standard protocol). And Table 2 shows the results of using different feature extraction networks to affect the classification effect. We use Resnet18 and our own feature extraction network Resw optimized based on Resnet. It can be seen that using our backbone, we can achieve better results than regular Resnet18. All our results are presented in percentage. The best

results are bolded. By the results, we can see that our method performs better when the train classes is small, our method has a stronger robustness.

On the UTD-MHAD dataset, we conducted experiments based on two criteria. And we compare our results with the SL-DML on different training testing ratio, the results are shown in Table 3. On UTD-MHAD dataset we only use our backbone Resw, and all the results below are based on it.

Table 1. One-shot results on NTU-RGB+D 120 one-shot.

Approach	Accuracy (%)
Resw+GNN (Our)	**52.1**
SL-DML	50.9
APSR	45.3
Average pooling[3]	42.9
Fully connected[3]	42.1

Table 2. One-shot results with different train classes on NTU-RGB+D 120 one shot.

Train classes	Resnet18 (%)	Resw	SL-DML	APSR
20	38.9	**39.3**	36.7	29.1
40	42.9	**43.8**	42.4	34.8
60	48.7	48.5	**49.0**	39.2
100	51.5	**52.1**	50.9	45.3

Table 3. One-shot results with different train classes on UTD-MHAD.

Train/test	Resw-skl. (%)	Resw-fused	SL-DML-skl.	SL-DML-fused
23/4	91.5	89.6	**92.7**	90.2
19/8	74.6	**76.8**	74.8	76.0
15/12	80.7	78.5	**81.1**	78.7

6. CONCLUSION

In our work, we have achieved the states-of-arts results of the one-shot action recognition on the NTURGB+D120 dataset. We are the first to apply graph neural network to the task of one-shot action recognition based on skeleton data. We have also optimized the structure of the embedding network Resnet, increase the nonlinearity of the model and increase the effectiveness of feature extraction part, leading to a better result on our new backbone.

REFERENCES

[1] Vinyals, O., Blundell, C. and Wierstra, D., "Matching networks for one shot learning," NIPS, 3630-3638 (2016).

[2] Garcia, V. and Bruna, J., "Few-shot learning with graph neural networks," ICLR, (2018).

[3] Memmesheimer, R., Theisen, N. and Paulus, D., "SL-DML: Signal level deep metric learning for multimodal one-shot action recognition," ICPR, 4573-4580 (2020).

[4] Wang, P., Li, W., Li, C. and Hou, Y., "Action recognition based on joint trajectory maps with convolutional neural networks," Knowledge-Based Systems, 158, 43-53 (2018).

[5] Liu, M., Liu, H. and Chen, C., "Enhanced skeleton visualization for view invariant human action recognition," Pattern Recognition, 68, 346-362 (2017).
[6] Snell, J., Swersky, K. and Zemel, R., "Prototypical networks for few-shot learning," NIPS, 4077-4087 (2017).
[7] Ravi, S. and Larochelle, H., "Optimization as a model for few-shot learning," ICLR, (2017).
[8] Finn, C., Abbeel, P. and Levine, S., "Model-agnostic meta-learning for fast adaptation of deep networks," ICML, (2017).
[9] Nichol, A., Achiam, J. and Schulman, J., "On first-order meta-learning algorithms," CoRR, abs/1803.02999, (2018).
[10] Gori, M., Monfardini, G. and Scarselli, F., "A new model for learning in graph domains," Proc. IJCNN, (2005).
[11] Li, Y., Tarlow, D., Brockschmidt, M. and Zemel, R., "Gated graph sequence neural networks," arXiv:1511.05493, (2015).
[12] Kipf, T. N. and Welling, M., "Semi-supervised classification with graph convolutional networks," ICLR, (2017).
[13] Liu, Y., Lee, J., Park, M., Kim, S. and Yang, Y., "Transductive propagation network for few-shot learning," ICLR, (2019).
[14] Liu, J., Shahroudy, A., Perez, M. L., Wang, G., Duan, L. Y. and Chichung, A. K., "NTU RGB+ d 120: A large-scale benchmark for 3d human activity understanding," IEEE Transactions on Pattern Analysis and Machine Intelligence, 816-833 (2019).
[15] Chen, C., Jafari, R. and Kehtarnavaz, N., "Utd-MHAD: A multimodaldataset for human action recognition utilizing a depth camera and a wearable inertial sensor," 2015 IEEE Inter. Conf. on Image Processing (ICIP), 168-17 (2015).