Advances in Artificial Intelligence, Big Data and Algorithms
G. Grigoras and P. Lorenz (Eds.)

doi:10.3233/FAIA230830

Robust Skeleton-Based Action Recognition Based on Multi-Stream Gradient Activation Graph Convolution Network

Xiaotian WANG, Qibin HU, Yanzhe HU and Qian WANG[1]
School of Electronic and Electrical Engineering,
Wuhan Textile University, China

Abstract. Current skeleton-based human action recognition methods usually apply to complete skeletons. However, in real scenarios, capturing incomplete or noisy skeletons is inevitable. When some joints information is occluded or interfered, it may significantly reduce the performance of current methods. To improve the robustness of action recognition models, a multi-stream dynamic graph convolutional network (GCN) is proposed to explore sufficient discriminative features distributed on all skeleton joints. By a multi-stream structure, gradient information of the graph structure is aggregated in a progressive manner. We introduce class activation map (CAM) techniques to extract the joints with the maximum amount of information in each stream. The activation maps in each stream are input to the next stream as mask matrices so that the new stream can explore the unactivated parts. Meanwhile, we set up a depth module so that we can still distinguish the characteristic values between nodes after multiple aggregations. Experiments prove that our model achieves the state-of-the-art performance on the NTU-RGB+D dataset. At the same time, it also shows strong robustness on the jittering dataset.

Keywords. Action Recognition, Graph Convolutional Network, Skeleton, Jittering, Activation Map

1. Introduction

Action recognition is a crucial task in computer vision that plays a significant role in accurately identifying human behaviors within a given scene, thereby enabling a greater understanding of the overall context. A fundamental problem in human behaviour recognition is how to extract differentiated and rich features that adequately describe the dynamics of the spatial and temporal in human behaviour.

Early deep learning-based methods treated human joints as a set of independent features, organizing them into feature sequences or pseudo-images, which were then fed into RNN or CNN models to predict action labels. These methods, which used RGB images and optical flow, outperformed previous handcrafted feature-based methods.[1][2][3] However, they disregarded the inherent correlation between joints, which reveal the topology of the human body's skeletal structure and, when compared

[1] Corresponding Author, Qian WANG, Wuhan Textile University, China;
Email: 2115363099@mail.wtu.edu.cn

to traditional RGB-based action recognition methods, skeleton-based representations are more robust to lighting, camera viewpoint, and other background changes.

Yan et al. [4] were the first to model the correlation between human joints using a graph structure and apply GCN and temporal convolutions to extract motion features. Subsequent research using GCN models has grown, with Shi et al. [5] proposing a dual-stream adaptive graph convolutional network that added an adaptive adjacency matrix. This adaptive adjacency matrix represented a topology that was no longer fixed according to the human body structure, as in ST-GCN, but could be learned along with other parameters. Chen et al. [6] proposed multi-scale spatial graph convolution modules and multi-scale temporal graph convolution modules to enrich the model's receptive field in both spatial and temporal dimensions. Recent methods [7][8] have achieved good results by refining the topology of the skeleton through various techniques, improving the model's upper limit. However, these methods have their limitations, as recognition becomes difficult when the skeleton dataset is incomplete or is plagued by noise interference. To address this issue, Weinland et al. [9] proposed a gradient-oriented 3D histogram of oriented gradients (HOG) descriptor with local partitioning and hierarchical classification to provide robustness against occlusion and viewpoint changes. Song et al. proposed the RA-GCN model [10], which used class activation maps (CAM) to learn unique features of missing joints across multiple streams. Yu et al. [11] trained their model by maximizing the mutual information between normal and noisy skeletons using a predictive coding approach.

However, the aforementioned models have some shortcomings, such as lower recognition rates in the backbone network and a lack of use of gradient information between joints, thereby limiting their ability to recognize noisy skeleton models. To solve these problems, in this paper we propose a multi-stream dynamic topological activation map convolution, which enhances the robustness of the network by fusing gradient information with a multi-stream structure. MGA-GCN draws inspiration from Grad-CAM++[12] and improves upon the activation module in RA-GCN. Each stream in the model can dynamically explore feature parts on previously unactivated joints using CAM technology to extract the joint with the most informative feature in each stream, forming an activation map that is inputted to the next stream in the form of a mask matrix containing joint activation information, while the new stream explores the unactivated parts. At the same time, the joint information contained in the activation module is used to calculate the gradient information between each joint and its adjacent joints, thereby enhancing the model's recognition ability through local gradient features. The MGA-GCN model structure is shown in Figure 1.

Our main contributions are as follows:

1. We propose a method that uses a multi-stream structure to capture the gradient information between nodes and dynamically adjusts the weights to enhance the performance of the network.

2. In the spatial convolution module, we propose a depth module to address the convergence of node values caused by the deep layers of the GCN, allowing the deep multi-stream graph convolutional network to more effectively learn features.

3. Extensive experimental results demonstrate that our proposed MGA-GCN outperforms existing graph convolutional methods on two benchmarks in NTU RGB+D, and also outperforms other methods in robustness experiments.

2. Model Architecture

2.1. Dynamic Topological Non-Shared GCN

Whereas traditional CNNs use the same convolutional parameters for all samples and the convolutional kernel parameters are determined through training, the dynamic convolutional kernel in our method is data-dependent, meaning that different data samples have different convolutions. This data dependency indicates that the dynamic kernel has a stronger representation capacity than the single kernel in conventional convolutions.

Currently, there are two categories of methods based on two different perspectives: Static/Dynamic Methods and Topology-Shared/Topology-Non-Shared Methods. In the GCN, static methods refer to the pre-defined topology structure that remains unchanged during training, while dynamic methods infer the topology structure of the GCN dynamically during inference. In action recognition, different types of motion features are represented by different channels, and the correlation between joints varies with the motion features. Therefore, a shared topology is not optimal. In the Topology-Shared method, both dynamic and static methods share the same topology structure across all channels. In contrast, the Topology-Non-Shared method uses different topology structures for different channels or channel groups. The experiments in [7] also demonstrate that the dynamic topology non-shared GCN has the least constraints and the strongest representation capacity among other graph convolutions. A dynamic topological non-shared GC can be expressed as:

$$z_i^k = \sum_{v_j \in N(v_i)} x_j^k([r_{ij1}^k w_{:,1}, \cdots, r_{ijC'}^k w_{:,C'}]) \quad (1)$$

where r_{ijC}^k is the dynamic topology of the k^{th} sample in the c^{th} channel between v_i and v_j. $w_{:,C}$ is the c^{th} column of w, where k denotes the index of the input sample.

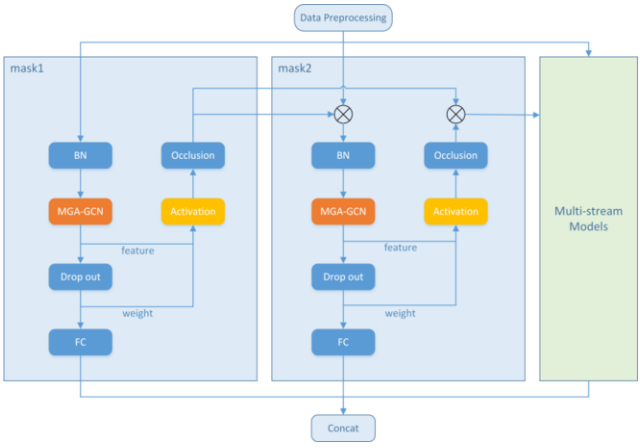

Figure 1. The multi-stream structure of the MGA-GCN. Each stream contains an MGA-GCN baseline network, the data is fed into each stream through the pre-processing module, the joint information is fed into the network through the mask matrix, the baseline network and the output of the Drop out layer are fed into the activation module and the mask matrix of the next stream is calculated. Finally, the outputs of each stream are joined to obtain the final output.

2.2. Activation Module

The activation module's function is to guide the already activated joints in the previous stream to the next graph, in order to reinforce the reinforcement learning of the inactive joints in the next stream. Our inspiration came from Grad-CAM++ [12], and the previous RAGCN [10] proposed by Song et al. was the first to introduce the CAM technique into human skeleton recognition. CAM can display where the model's weights or centers of gravity are located during training, and which part of the features the classification model is based on. In human skeleton recognition, the gradient information between adjacent joints is also an important factor that cannot be ignored. When classifying, the contribution of each joint is also different. Therefore, we introduce a center-oriented local gradient feature to enhance the model's recognition ability. We separately calculate the gradient information between each joint and its adjacent joints, and set the weights based on this information. We replace the coordinates *(x, y)* in the feature map with the frame number *t* and joint number *i*, *j* represents nodes other than *i* in the skeleton sequence. This enables us to locate the activated joints and calculate their gradient information. Here, we define *i* as the source joint, and *j* as the target joint. The source joint refers to the joint closest to the center of gravity of the skeleton, while the target joint refers to the joint farthest from the center of gravity of the skeleton. As shown in Figure 2, the weight of node *i* at time *t* is as follows:

$$w_k^c = \sum_{i,j \in R} \left(\frac{\delta Y_i}{\delta A_{ti}^k} - \frac{\delta Y_j}{\delta A_{tj}^k} \right) \tag{2}$$

where *A* denotes the feature layer, *Y* denotes the score predicted by the network for category *c* before going through activation, and the score for each node is:

$$S_c(t,i) = relu\left(\sum_k w_k^c f_k(t,i) \right) \tag{3}$$

The predefined parameter σ is then used to determine which joints are activated by the corresponding stream, and $Score_c$ is used to represent the fractional graph of all joints in stream s^{th}. The activation graph of map_c^s is represented as follows,

$$map_c^s = \varepsilon\left(\frac{\sigma - min(Score_c)}{max(Score_c)} \right) \tag{4}$$

where $min()$ and $max()$ denote the minimum and maximum functions respectively, $\varepsilon()$ denotes the Heaviside step function and the mask matrix of the s^{th} stream is denoted as:

$$mask_s = \left(\prod_{i=1}^{s-1} mask_i \right) \otimes \left(1 - map_c^{s-1} \right) \tag{5}$$

Finally, the resulting $\mathrm{mask_s}$ is combined with the pre-processed skeleton data to obtain the final result by performing $\otimes$ operations.

Compared to the previous method, we weigh the contributions between nodes and gradient information and set the weights dynamically to enhance the correlation between nodes. This also allows for a better exploration of all joint information.

2.3. Depth Module

A significant portion of skeleton-based action recognition networks are multi-stream GCN. Graph convolutional neural networks can be understood as performing a fully connected transformation on the features during each aggregation operation, followed by taking the average of the aggregated results. However, too many aggregation operations on each vertex's neighboring nodes can lead to convergence of all vertex values to the same value, making it impossible to distinguish individual vertex features. To address this issue, we propose a depth module located in the spatial modeling module. The data is first processed through an activation function to produce an output, which is then refined by this module to extract joint features before generating the final output. The formula is as follows:

$$output = \frac{x}{\omega} min(\theta, max(0, x + \theta)) \tag{6}$$

Where ω and θ are balancing parameters, with this module we can differentiate feature values between nodes even after multiple aggregations, effectively improving the accuracy of the deep multi-stream map convolutional network model.

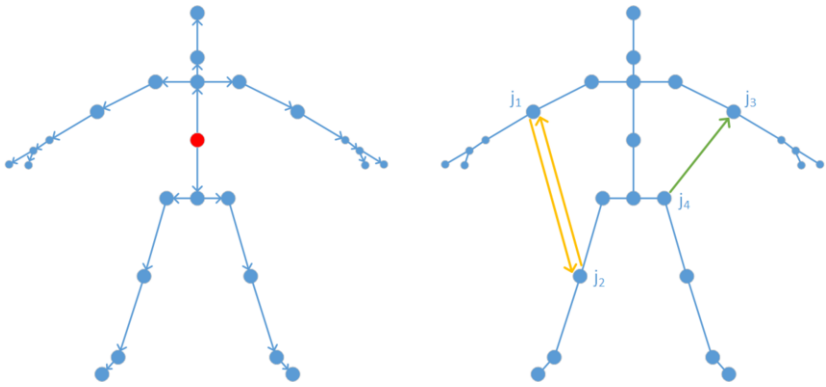

Figure 2. Description of the source and target joints. (a) Representation of joint orientation. The red joint is the centre of gravity of the skeleton and the arrow is pointing from the source joint to the target joint. (b) For joints that are adjacent we divided into two different ways of calculating the gradient: (1) When two joints are at the same distance from the central node, we calculate the gradient information with j_1 and j_2 as the centre respectively, that is, the gradient direction is j_1 to j_2 when calculating j_1, and the opposite when j_2 is the centre; (2) When two joints are at different distances from the central node, we only calculate the gradient information with the source joint as the centre, that is, the gradient direction is j_4 to j_3.

3. Experiments

3.1. Ablation Studies

In this part, We conduct ablation experiments on the importance of the depth module and the feasibility of our multi-stream structure, as well as an exploration of the optimal configuration of the depth module, using NTU RGB+D for the experimental dataset.

Firstly, we conduct experiments on the settings of the equilibrium parameters ω and θ. We used the dual-stream MGA-GCN without the addition of the depth module as a baseline and compared it on a benchmark of the CS. Without changing the other parameters and structure of the network, only the ω and θ parameters were modified to pursue the experiments. The experimental results are shown in Table 1. The optimal solution can be obtained when ω=3,θ=8. After that, we make different configurations of the positions of the depth modules to study their performance separately. There are three options as follows, option 1 is configured in the spatial convolution module alone, option 2 is configured in the temporal convolution module alone, and option 3 is configured in both the spatial and temporal convolution modules. The results are shown in Table 2.

To demonstrate the effectiveness of the depth module on multi-stream structured GCN, we used the 2s-AGCN, which is also a two-stream framework, as a baseline for comparison experiments on the CS benchmark of NTU60, and the results are shown in Table 3.

To explore the limits of the multi-stream model, we conducted experiments with the addition of the depth module, and the results are shown in Table 2, where the precision will drop when the number of streams in the network is greater than 3. For skeletal behaviour recognition, there are a limited number of joints in each action category that play a judgement role, and more streams do not improve the recognition ability of the network, but may lead to a decrease in accuracy due to overfitting.

Table 1. Comparison of different parameters of the depth module.

Model	Parameters	Accuracy(%)
2s MGA-GCN (None depth module)	-	89.72
	ω=1,θ=2	89.85
	ω=1,θ=4	90.37
	ω=1,θ=6	90.18
	ω=2,θ=4	90.52
	ω=2,θ=6	90.59
	ω=3,θ=6	90.64
	ω=3,θ=8	**90.74**
	ω=3,θ=9	90.22
	ω=4,θ=6	90.57
	ω=4,θ=8	90.02

Table 2. Comparison of different model setups on NTU60.

Model	Param	Position	CS(%)	CV(%)
2s MGA-GCN	ω=3,θ=8	-	89.72	-
		unit_gcn	90.74	-
		Temporalconv	89.78	-
		unit_gcn and Temporalconv	90.03	-
1s MGA-GCN	ω=3,θ=8	-	89.72	94.92
2s MGA-GCN		-	90.74	95.68
3s MGA-GCN		-	90.90	95.55
4s MGA-GCN		-	90.31	95.28

Table 3. Comparison of the accuracy of 2s-AGCN before and after setting up the depth module.

Methods	Position	Param	Accuracy(%)
2s-AGCN	-	-	86.32
2s-AGCN	unit_gcn	ω=3,θ=8	86.68

3.2. Experimental Results on Standard Dataset

We compare our model with existing methods on the NTU RGB+D dataset, and the network's parameters are set to the best values obtained from ablation studies. As can be seen from the experimental results in Table 4, our multi-stream network outperforms the current state-of-the-art GCN method on both CS and CV evaluation benchmarks. Compared to the previous state-of-the-art methods, the MGA-GCN dual-stream network has improved by 0.78% and 0.6% on CS and CV benchmarks, respectively, and the three-stream network has improved by 0.94% and 0.47% on CS and CV benchmarks, respectively. In comparison with the similarly robust RA-GCN and PeGCN, 2s MGA-GCN has achieved an improvement of 4.14% and 5.14% on the CS benchmark, and 2.35% and 2.28% on the CV benchmark, respectively; 3s MGA-GCN has achieved an improvement of 4.3% and 5.3% on the CS benchmark, and 2.22% and 2.15% on the CV benchmark, respectively.

Table 4. Accuracy comparison with state-of-the-art methods on the NTU60 dataset.

Model	NTU-RGB+D	
	X-Sub (%)	X-View (%)
ST-GCN[4]	81.50	88.30
AS-GCN[13]	83.30	93.30
UNIK[14]	86.26	93.01
2s-AGCN[5]	86.32	94.20
RA-GCN(2s)[10]	85.39	93.02
RA-GCN(3s)[10]	86.60	93.33
PeGCN[11]	85.60	93.40
PA-ResGCN-B19[15]	86.63	94.33
MS-G3D[16]	87.29	94.32
CTR-GCN[7]	89.96	95.08
MGA-GCN (1s)	89.72	94.92
MGA-GCN (2s)	90.74	**95.68**
MGA-GCN (2s)	**90.90**	95.55

3.3. Experimental Results on Jittering Datasets

In order to evaluate the effects of jittered skeletons, we formulated two different jitter datasets by introducing varying levels of Gaussian noise via methods proposed in [11]. The resulting experimental outcomes are comprehensively reported in Tables 5 and 6. The jitter probability of the joints is set to 0.02, 0.04, 0.06, 0.08, 0.10. As shown in the tables, the multi-stream network using activation maps for construction performs significantly better on the jittered datasets of NTU 60 compared to other networks. Moreover, the MGA-GCN model demonstrates high stability even at higher jitter probabilities.

Table 5. Experimental results(%) of the jitter skeleton (α = 0, ν = 0.05) of the NTU60 (CS benchmark).

α = 0 ν = 0.05	**Probability of jitter**					
	0	0.02	0.04	0.06	0.08	0.10
ST-GCN[4]	81.50	77.16	68.48	55.54	41.95	28.32
2s-AGCN[5]	86.32	81.68	75.04	70.19	66.91	62.74
2s RA-GCN[10]	85.39	84.94	84.34	82.22	77.23	71.20
3s RA-GCN[10]	86.60	86.52	86.07	85.24	84.08	82.08
PeGCN[11]	84.36	80.65	73.23	65.13	51.70	42.02
CTR-GCN[7]	89.96	86.89	80.55	76.34	72.13	58.94
MGA-GCN (2s)	90.74	90.06	89.32	88.62	88.23	87.17
MGA-GCN (3s)	**90.90**	**90.34**	**90.12**	**89.68**	**89.15**	**88.23**

Table 6. Experimental results(%) of the jitter skeleton (α = 0, ν = 0.10) of the NTU60 (CS benchmark).

α = 0 ν = 0.10	**Probability of jitter**					
	0	0.02	0.04	0.06	0.08	0.10
ST-GCN[4]	81.50	72.32	55.46	42.45	21.35	12.30
2s-AGCN[5]	86.32	76.76	62.89	56.03	49.84	31.31
2s RA-GCN[10]	85.39	84.18	81.79	76.13	67.40	58.23
3s RA-GCN[10]	86.60	85.70	84.18	82.48	78.73	72.92
PeGCN[11]	84.36	76.31	67.84	58.72	36.71	27.87
CTR-GCN[7]	89.96	82.19	76.56	62.49	51.46	40.90
MGA-GCN (2s)	90.74	89.09	87.00	84.43	84.02	80.35
MGA-GCN (3s)	**90.90**	**89.88**	**88.47**	**87.93**	**86.94**	**83.64**

4. Conclusion

In this paper, we proposed a new multi-stream gradient activation map graph convolutional network for skeleton-based action recognition. MGA-GCN captures gradient information between nodes through a multi-stream structure and dynamically sets weights to improve network performance. We also introduced a depth module to address the problem of node value convergence resulting from multiple aggregations of nodes. Additionally, our network exhibits strong effectiveness and robustness for noisy skeletons. Through extensive experiments, MGA-GCN has been shown to outperform current state-of-the-art methods on both standard and noisy datasets.

References

[1] Wang L, Xiong Y, Wang Z, et al. Temporal segment networks: Towards good practices for deep action recognition[C]//European conference on computer vision. Springer, Cham, 2016: 20-36.

[2] Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos[J]. Advances in neural information processing systems, 2014, 27.

[3] Punnakkal A R, Chandrasekaran A, Athanasiou N, et al. BABEL: bodies, action and behavior with English labels[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 722-731.

[4] Yan S, Xiong Y, Lin D. Spatial temporal graph convolutional networks for skeleton-based action recognition[C]//Proceedings of the AAAI conference on artificial intelligence. 2018, 32(1).

[5] Shi L, Zhang Y, Cheng J, et al. Two-stream adaptive graph convolutional networks for skeleton-based action recognition[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 12026-12035.

[6] Chen Z, Li S, Yang B, et al. Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(2): 1113-1122.
[7] Chen Y, Zhang Z, Yuan C, et al. Channel-wise topology refinement graph convolution for skeleton-based action recognition[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 13359-13368.
[8] Ye F, Pu S, Zhong Q, et al. Dynamic gcn: Context-enriched topology learning for skeleton-based action recognition[C]//Proceedings of the 28th ACM International Conference on Multimedia. 2020: 55-63.
[9] Weinland D, Özuysal M, Fua P. Making action recognition robust to occlusions and viewpoint changes[C]//European Conference on Computer Vision. 2010 (CONF).
[10] Song Y F, Zhang Z, Shan C, et al. Richly activated graph convolutional network for robust skeleton-based action recognition[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 31(5): 1915-1925.
[11] Yoon Y, Yu J, Jeon M. Predictively encoded graph convolutional network for noise-robust skeleton-based action recognition[J]. Applied Intelligence, 2022: 1-15.
[12] Chattopadhay A, Sarkar A, Howlader P, et al. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks[C]//2018 IEEE winter conference on applications of computer vision (WACV). IEEE, 2018: 839-847.
[13] Li M, Chen S, Chen X, et al. Actional-structural graph convolutional networks for skeleton-based action recognition[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 3595-3603.
[14] Yang D, Wang Y, Dantcheva A, et al. Unik: A unified framework for real-world skeleton-based action recognition[J]. arXiv preprint arXiv:2107.08580, 2021.
[15] Song Y F, Zhang Z, Shan C, et al. Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition[C]//proceedings of the 28th ACM international conference on multimedia. 2020: 1625-1633.
[16] Liu Z, Zhang H, Chen Z, et al. Disentangling and unifying graph convolutions for skeleton-based action recognition[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 143-152.