Skeleton-based Action Recognition with Two-Branch Graph Convolutional Networks

Zhi Liu, Qici Xie, Yunhua Lu and Xian Wang
1School of Liangjiang AI, Chongqing University Of Technology, Chongqing, Chongqing, 043101, China
*Corresponding author’s e-mail: liuzhi@qcut.edu.cn

Abstract. Graph convolution is a popular technique for action recognition based on human skeleton data. Due to the fact that human skeleton data can be treated as a graph in three dimensions, Graph Convolutional Networks (GCN) represent the input data as a graph structure to perform the recognition task, and thus numerous approaches based on GCN to recognize actions have achieved great results. In spite of the input data being structured as a four-dimensional tensor in GCN, it still can not fully exploit the contained rich action-related information. Therefore, we proposed a new model that the three-dimensional skeleton data is put into both the Convolutional Neural Network (CNN) and the Graph Convolutional Neural Network branches to perform feature extraction in the spatiotemporal dimension separately, then the output information is fused for prediction. Given the richness of time-domain information characteristics, feature extraction is enhanced by increasing the model’s depth. After the pooling layer and the fully connected layer, we concatenate the outputs at the ends of the graph data stream and the convolution data stream to obtain the network’s final output. Finally, the prediction results can be obtained via the SoftMax layer. On the Kinetic 400 dataset, our suggested model outperforms Benchmark STGCN in terms of accuracy. The experiment results indicate that the proposed novel model successfully increases the generalization ability and classification performance for action recognition.

1. Introduction
In recent years, human action recognition has become a popular and important application area of research, which not only makes valuable contributions in application scenarios such as video surveillance and human-computer interaction, also has been widely used in real-life applications such as security, traffic management, intelligent care giving, entertainment and leisure. However, there are still some challenges in this research field such as the action of people in video affected by changes in viewpoint, background irrelevant data, and difficulties in defining the time of action occurrence and end. With the development of deep learning, human action recognition has achieved a shift from the traditional method of setting features manually to an end-to-end approach, and has made great progress. Several research methods have applied RNN (Recurrent
Neural Networks) and CNN (Convolutional Neural Networks) to this field, which process RGB video frame by frame and use CNN to extract the spatial features of single frames divided into small equal parts and the overall features of the entire video frame. Based on the advantages of RNN for feature extraction of sequential data, it can be used to process the connections between video frames to extract spatiotemporal features of human action. Despite some reported methods has achieved good results, human action recognition still face the impact of perspective, background environment and computational complexity. With the creation of skeleton joint detection algorithms such as Alpha Pose[7], Mask RCNN[18] and OpenPose[1], and the application of deep cameras, action recognition and skeleton predictive classification of human action based on skeleton data has gradually developed. The skeleton data is free from extraneous background factors, colour shading issues due to its better lightweight and specificity (focus on the human body itself).

Since the skeleton data exists in a Non-Euclidean geometric space, Graph Convolutional Networks(GCN)[6,12] has been introduced to skeleton-based action recognition and achieved many encouraging results. For the first time, STGCN[27] models human skeleton data as graph data and utilizes graph convolution operations to extract spatio-temporal features to obtain action class. In the graph convolutional network models, the nodes and bones in the skeleton data are modelled as graph nodes and paths between the nodes separately, and the skeleton graphs of multiple video frames are stacked together to form a complete STGCN graph.

However, STGCN still has two shortcomings. First, STGCN does not sufficiently uncover information from the skeleton graph data especially for many global features; second, STGCN only adopts a simple approach to extract features between frames in the time dimension, which may cause losing many important connections between frames. In this work, we combine a convolutional neural network with a graph convolutional neural network to process data. The former one forms one branch, called the C-Branch, and the latter one is another branch, called the G-Branch, and the data is passed to both branches for operation. In order to fully exploit the inherent rich information in the temporal dimension of the skeleton data, we enhance the temporal feature extractor by improving the depth of the model. Figure 1 presents the form of the input skeleton data. The vertical direction represents the number of input video frames. The forward direction represents the coordinate feature of the joint point. The horizontal direction represents the number of joint points.

In a word, the main contributions of our work are as follows:(1)We design a double branch structure to make the spatiotemporal feature extraction more sufficient. (2)We make full use of temporal information by strengthening the depth of model. To evaluate the proposed method, we perform experiments on the dataset Kinetics 400[10]. Results show that proposed model achieves better accuracy than that of benchmark STGCN.
2. Related Works

2.1. Traditional Action Recognition
Action recognition feature extraction methods include calculation methods based on geometric features of the human body[9], feature extraction methods of motion information[2]; along with HOG (Histogram of oriented gradient)[5], SIFT (Scale-invariant feature transform)[17] and other multi-scale feature extraction algorithms with prior knowledge, HOG3D (Histogram of gradients 3D) combined with video sequence information and other feature extraction methods based on spatiotemporal interest points have been developed by leaps and bounds. Unfortunately, these methods are inevitably affected by a series of uncontrollable factors such as background independent optical flow interference, overlapping of characters and interactions between people. However, the current action recognition system based on human skeleton data can be immune to the above effects.

2.2. Skeleton-based Data Action Recognition
The skeleton data is highly focused on the human body and the corresponding skeleton data based methods[15, 24]show the stronger capability to avoid the interference such as optical flow, background, blurring and other problems than that of the standard RBG data-based methods[25, 23, 26, 13]. These methods use skeleton joint point coordinates to represent the skeleton, and action classification can be accomplished by extracting information from both the joint points and the skeleton in subsequent frames. However, the methodology based on human skeleton data is influenced by the viewpoint and performance of the skeleton identification algorithm, making it difficult to recognize minor motions such as cleaning teeth, drinking water, writing, and playing with a mobile phone. For the issues of viewpoint discrepancies, [28] proposed a viewpoint adaptive model to adjust the skeleton data to the suitable perspective to avoid overlap caused by incorrect angles.

2.3. Methods using graph convolution
Action recognition methods using graph convolution are based on human skeleton data, and STGCN first applied graph convolution to action recognition with good results. At present, Graph convolution has became a popular direction in the field of action recognition. Unlike normal CNN convolution, graph convolution operates on data in the form of a graph structure, and compared with CNN, GCN can make full use of the connections between local joints. The graph is made up of individual nodes and the links between them, and these nodes and links have their own corresponding parameters.

![Figure 2. Illustration of the overall architecture of the model we proposed.](image)
Skeleton joints and bones are modelled as nodes and paths in the graph separately. Multiple works have demonstrated that graph convolution can effectively extract the time-space features of skeleton graph data\cite{21, 22, 3, 16}. STGCN innovatively employs graph convolution to extract spatio-temporal features of skeleton data and proposes a novel region delineation strategy. The information from the skeleton vectors (new feature) are adopted to improve the STGCN, and the two-stream network\cite{21} is also redesigned and then the adaptive matrices are used in it to obtain better results. This model\cite{22} not only captures more discriminative features in spatial structure and temporal dynamics, but also explores co-occurrence relationships between spatiotemporal domains. Also, the authors propose a temporal hierarchy to increase the temporal receptive domain of the top-level AGC-LSTM layer, which improves its ability of learning high level semantic representations and significantly reduces computational costs. STGCN ignores certain implicit information, and its temporal feature extraction module is simple, resulting in the insufficient exploiting the connections that exist between successive frames in the temporal dimension. For this reason, our work focuses on these two aspects to enhance the information extraction. The experimental results prove the validity of our novel model.

3. Method
This section will outline the model we presented. We utilize a two-branch structure with one branch composed of convolutional neural networks (denoted as C-Branch), which is responsible for digging up and collecting the rich information of dependent relationships between skeleton data, as inspired by\cite{14}. The other one is a graph convolutional neural network (denoted as G-Branch), which is utilized to extract the connections between the joints and the skeleton’s spatial features. To produce classification predictions, we concatenated the outputs of the two branches, pooled and normalized them, and then fed them into the completely connected and SoftMax layers. Figure 2 shows a summary of the full model.

In the G-Branch, the yellow block represents the TCN module, which is responsible for extracting time domain information, while the blue block is the GCN module, which uses the graph convolution method to extract the connections between the joint points and the spatial characteristics of the skeleton. In C-Branch, the gray block is the Spatial-CNN module, which uses a common convolution method to extract the dependency information between the nodes in the skeleton data, and the yellow block represents the TCN module.

3.1. Two-Stream Branch
It is not sufficient to employ graph convolution networks alone to extract more valuable information. Therefore, we suggest a two-branch structure, one for the graph convolutional network and another for the convolutional network. In comparable to the two-stream network proposed in\cite{23}, the additional branch of the convolutional network(C-Branch) compensates for the loss of information extraction by examining the link between new skeleton nodes and the overall characteristics extracted from the input skeleton data, result in enhancing the network’s fitting abilities.

The input data is a five-dimensional tensor with size \((N,C,T,V,M)\), \(N\) denotes batch, \(C\) is the input channel, \(T\) is the number of frames, \(V\) denotes the number of joints, \(M\) denotes the total number of persons in the video, as shown in Figure 1. The initial format of the input data is Json, but after the processing of OpenPose, the input data transformed into tensor. Both branches have 12 basic blocks. After normalization, the data is formally added to the network branch. The shape of the data of joints is \((x,y,c)\), in which \(x\) and \(y\) represent the relative coordinates of the joint points in the video, and \(c\) represents the confidence value of the \(x\) and \(y\) output by OpenPose, that is, in \((N,C,T,V,M)\), with \(C=3\). As illustrated in Figure 2, we concatenate the output of both branches in the channel dimension, the final size of the FC will be larger. A larger FC has the better mapping ability to the features that learned by two-branch. After the global average pooling layer, the data will enter the fully connected layer. Finally, the data will enter the SoftMax for prediction. In end part of the model, we didn’t use SoftMax directly, because our loss function is cross entropy.
3.2. G-Branch
In this branch, the graph convolution will act as a tool for extracting spatial features. The model uses graph convolution formula as shown in Equation 1, where $A$ is the adjacency matrix, $\lambda$ is the Laplace matrix, $W$ denotes the weights, $F_{in}$ denotes the input features, and $F_{out}$ is the output features.

$$F_{out} = \Lambda^{-\frac{1}{2}}(A + I)\Lambda^{-\frac{1}{2}}F_{in}W$$  \hspace{1cm} (1)$$

In (1), $\Lambda_{ii} = P_j(A_{ij} + \Lambda_{ij})$, the weight vectors of multiple output channels are superimposed to form a weight matrix $W$, this formula is under the Uni-labeling Partitioning strategy. When the partitioning strategy is Distance Partitioning, Equation 1 will be rewritten as Equation 2.

$$F_{out} = \sum_j \Lambda_{jj}^{-\frac{1}{2}} A_{jjj}^{-\frac{1}{2}} F_{in} W_j$$  \hspace{1cm} (2)$$

In (2), $A_{ii} = P_k(A_{ikj}) + \alpha$, $\alpha$ is to avoid empty lines in $A$. In kinetic, the number of joints is 18, hence the Adjacency Matrix is $18 \times 18$, $V$ will not change and remains at 18. This branch consists of 12 basic blocks, while there are only 10 basic blocks in STGCN. In order to increase the model’s generalizability and compensate for the model’s lack of information extraction strength in the temporal dimension, we increase the number of fundamental blocks from 9 to 12. Additionally, we reduced the size of the original channel from 64 to 32. This technique improves the extraction of information in both the temporal and spatial dimensions because of the data going through the TCN before entering the GCN (a module for spatial feature extraction).

In the basic block, data first enters into the TCN and subsequently into the GCN. TCN is composed of the following components: BN (Batch Normalization), Relu, Conv2d, BN, Dropout. GCN is composed of two components: Conv2d and graph convolution. Finally, the result is combined with the other blocks to generate the final result. As illustrated in Figure 3, the residual block is equal to the data entering TCN and will be added to the result from TCN following Conv and BN, and there are three processing units in every block, namely GCN, TCN and Residual. For G-Branch and C-Branch, the difference of the basic block is in that there is no GCN in C-Branch instead of Spatial-Cnn, and there is no Graph Conv in this module, just a module of Conv.

3.3. C-Branch
Many previous works have proved the effectiveness of convolutional neural networks for processing skeleton data[28, 14, 24], and the GCN-based method ignores the information implicit in the skeleton data. In order to make up for this lack, we added C-Branch to explore more features. For this branch, the input data’s dimension is $(N, C, T, V)$, where $N$ denotes the batch, $C$ is the channel, $T$ is the number of time frames, and $V$ is the number of skeleton nodes. For the data with size $(C, T, V)$, the graph convolution technique in G-Branch does not fully utilize the contained information. Thus, in order to address this
issue, CNN networks are employed in this branch to leverage more information. Further, to obtain sufficient information extraction in the temporal and geographical dimensions, we increase the depth of the C-Branch network from 9 to 12. The construction of the basic block in the dual branch is depicted in Figure 3.

The blocks of C-Branch are divided into two modules: Spatial-Cnn and TCN. The difference between Spatial-Cnn and the original structure is that we eliminated the graph convolution operation in favor of using standard CNN for spatial dimension information modeling and feature extraction. TCN’s module is identical to that of G-Branch, and it also includes a residual module. For TCN, we set the kernel-size to 9, which means that the spatio-temporal features are extracted every 9 frames. The step size is 1 when the channel does not change, and set to 2 when there is a change. This can enable TCN adapts to a small range of changes in the skeleton data. Before TCN, Spatial-Cnn is responsible for processing the information contained in each joint.

4. Experiments

Table 1: This table shows the compare test results on Kinetic 400, and the superscripts with 1, 2, 3, 4 correspond to EXP 1, 2, 3, and 4 respectively.

| Method                      | Top1(%) | Top5(%) |
|-----------------------------|---------|---------|
| G-Branch(128)              | 30.7    | 52.8    |
| G-Branch + C-Branch(128)   | 32.4    | 54.8    |
| G-Branch + C-Branch(256)   | 32.6    | 55.1    |
| G-Branch + C-Branch(384)   | 33.3    | 55.8    |

Table 2: Experiment on dataset Kinetics 400, the result shows that our proposed model is effective.

| Method          | Top1(%) | Top5(%) |
|-----------------|---------|---------|
| Feature Enc[8]  | 14.9    | 25.8    |
| Deep LSTM[20]   | 16.4    | 35.3    |
| Temporal Conv[11]| 20.3   | 40.0    |
| Baseline STGCN[27]| 30.7  | 52.8    |
| S-TR[19]        | 32.4    | 55.2    |
| SAN [4]         | 35.1    | 55.7    |
| Ours            | 33.3    | 55.8    |

4.1. Experiments Environment

Our experiment dataset called Kinetic 400. It contains films from YouTube in a total of 400 categories, with at least 400 videos in each category. Each video is around ten seconds long. Human-object interaction such as playing musical instruments, human-person interaction such as shaking hands and hugging, and sports, among others, are classified as person, person-person, and person-object. Apart from that, we utilize PyTorch 1.4 as our development framework, and Python’s vision is 3.7. The GPU is the RTX 5000.
4.2. Comparative Experiments
In order to validate the performance of the two-branch structure, we did some comparative experiments on the Kinetics 400[10] dataset. Table 1 summarizes the findings of several comparison experiments. In addition, Two-Branch consists of G-Branch and C-Branch, therefore, G-Branch + C-Branch presents Two-Branch. In order to explain the experiment better, we use Exp No. number to represent the experiment, as shown in the table. In Exp No.1, G-Branch(128) is same as STGCN, but the number of blocks is 12, this results a small degree of accuracy improvement. In Exp No.2, We added C-Branch on the basis of Experiment 1, this approach significantly improved the overall performance, top 1 reached 32.4, and top 5 reached 54.8, confirming the effectiveness of C-Branch. In order to further improve the fitting ability of the model, we increased the number of final channels of the model, and obtained results such as Exp No.3 and Exp No.4. In this process, our initial learning rate is different from STGCN. The initial learning rate is 0.05, it was reduced to (1e-2, 1e-3, 1e-4) in step (35, 45, 55), and the weight attenuation coefficient is changed to 0.0002. Table 2 summarizes the model’s accuracy, demonstrates that it outperforms both baseline STGCN and some previous studies.

4.3. Experiments Details
The optimizer Stochastic gradient descent (SGD) with Nesterov momentum (0.9) is applied as the optimization strategy, and we set the batch size to 128. Cross-entropy is selected as the loss function to backpropagate gradients. The weight decay is set to 0.0002. During the experiment, total epoch number is 65, the learning rate is 0.05, and the learning rate is divided by 10 at epoch 35, 45, 55.

5. Conclusion
For STGCN does not sufficiently uncover information from the skeleton graph data especially for many global features, we proposed a model that includes G-Branch and C-Branch to allowing for the extraction of more meaningful information from the skeleton data. Additionally, we added more basic blocks for feature extraction in both branches, which allows for more accurate extraction of information in the temporal dimension and makes up for the lack of STGCN learning in the temporal dimensional features. On the Kinetic 400 dataset, the proposed model achieves better accuracy performance than STGCN and other methods, demonstrates the effectiveness of our model.

However, for some reasons we did not have access to the NTU-RGBD dataset and only have the experimental results on Kinetics 400. In the future, we will continue to experiment on other specialized datasets in the future to further validate the effectiveness of the model. In addition, we will further improve the performance of the model by enhancing the feature extraction capability of our network through implementing multi-scale feature extraction and introducing attention mechanism.

Acknowledgment
This work was supported by the Natural Science Foundation of Chongqing, China (Grant No. cstc2019jcyj-msxmX0487) and Scientific Research Foundation of Chongqing University of Technology(Grant No.2020ZDZ026).

References
[1] Cao, Z. Simon, T. Wei, S.-E. , Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7291–7299.
[2] Chaudhry, R. Ravichandran, A. Hager, G. , Vidal, R. (2009). Histograms of oriented optical flow and binetcauchy kernels on nonlinear dynamical systems for the recognition of human actions. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 1932–1939. IEEE.
[3] Cheng, K. Zhang, Y. He, X. Chen, W. Cheng, J. , Lu, H. (2020). Skeleton-based action recognition with shift graph convolutional network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 183–192.
[4] Cho, S. Maqbool, M. Liu, F. , Foroosh, H. (2020). Self-attention network for skeleton-based human action recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 635–644.

[5] Dalal, N , Triggs, B. (2006). Histograms of oriented gradients for human detection. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), volume 1, pages 886–893. Ieee.

[6] Defferrard, M. Bresson, X. , Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. arXiv preprint arXiv:1606.09375.

[7] Fang, H.-S. Xie, S. Tai, Y.-W. , Lu, C. (2017). Rmpe: Regional multi-person pose estimation. In Proceedings of the IEEE International Conference on Computer Vision, pages 2334–2343.

[8] Fernando, B. Gavves, E. Oramas, J. M. Ghodrati, A. , Tuytelaars, T. (2015). Modeling video evolution for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5378–5387.

[9] Fujiyoshi, H. Lipton, A. J. , Kanade, T. (2004). Real-time human motion analysis by image skeletonization. IEICE TRANSACTIONS on Information and Systems, 87(1):113–120.

[10] Kay, W. Carreira, J. Simonyan, K. Zhang, B. Hillier, C. Vijayanarasimhan, S. Viola, F. Green, T. Back, T. Natsev, P. et al. (2017). The kinetics human action video dataset. arXiv preprint arXiv:1705.06950.

[11] Kim, T. S , Reiter, A. (2017). Interpretable 3d human action analysis with temporal convolutional networks. In 2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW), pages 1623–1631. IEEE.

[12] Kipf, T. N , Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

[13] Lan, Z. Zhu, Y. Hauptmann, A. G. , Newsam, S. (2017). Deep local video feature for action recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 1–7.

[14] Li, C. Zhong, Q. Xie, D. , Pu, S. (2017). Skeleton-based action recognition with convolutional neural networks. In 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), pages 597–600. IEEE.

[15] Li, C. Zhong, Q. Xie, D. , Pu, S. (2018). Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. arXiv preprint arXiv:1804.06055.

[16] Liu, Z. Zhang, H. Chen, Z. Wang, Z. , Ouyang, W. (2020). Disentangling and unifying graph convolutions for skeleton-based action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 143–152.

[17] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In Proceedings of the seventh IEEE international conference on computer vision, volume 2, pages 1150–1157. Ieee.

[18] Massa, F, Girshick, R. (2018). maskrcnn-benchmark:Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. https://github.com/facebookresearch/ maskrcnn-benchmark, 2018. Accessed: [Insert date here].

[19] Plizzari, C. Cannici, M. , Matteucci, M. (2021). Skeleton-based action recognition via spatial and temporal transformer networks. Computer Vision and Image Understanding, 208:103219.

[20] Shahroudy, A. Liu, J. Ng, T.-T. , Wang, G. (2016). Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1010–1019.
[21] Shi, L. Zhang, Y. Cheng, J. , Lu, H. (2019). Two-stream adaptive graph convolutional networks for skeletonbased action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12026–12035.

[22] Si, C. Chen, W. Wang, W. Wang, L. , Tan, T. (2019). An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1227–1236.

[23] Simonyan, K , Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. arXiv preprint arXiv:1406.2199.

[24] Song, S. Lan, C. Xing, J. Zeng, W. , Liu, J. (2017). An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In Proceedings of the AAAI conference on artificial intelligence, volume 31.

[25] Wang, H , Schmid, C. (2013). Action recognition with improved trajectories. In Proceedings of the IEEE international conference on computer vision, pages 3551–3558.

[26] Wang, L. Xiong, Y. Wang, Z. Qiao, Y. Lin, D. Tang, X. , Van Gool, L. (2016). Temporal segment networks: Towards good practices for deep action recognition. In European conference on computer vision, pages 20–36. Springer.

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

[28] Zhang, P. Lan, C. Xing, J. Zeng, W. Xue, J. , Zheng, N. (2017). View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In Proceedings of the IEEE International Conference on Computer Vision, pages 2117–2126.