Hindawi
Security and Communication Networks
Volume 2021, Article ID 2290304, 13 pages
https://doi.org/10.1155/2021/2290304

Research Article

A Lightweight Hierarchical Model with Frame-Level Joints Adaptive Graph Convolution for Skeleton-Based Action Recognition

Yujian Jiang,[1,2,3] **Xue Yang**,[1,2,3] **Jingyu Liu**,[1,2,3] **and Junming Zhang**[1,2,3]

[1]*State Key Laboratory of Media Convergence of Communication, Communication University of China, Beijing 100024, China*
[2]*Key Laboratory of Acoustic Visual Technology and Intelligent Control System, Communication University of China, Ministry of Culture and Tourism, Beijing 100024, China*
[3]*Beijing Key Laboratory of Modern Entertainment Technology, Communication University of China, Beijing 100024, China*

Correspondence should be addressed to Yujian Jiang; yjjiang@cuc.edu.cn

Received 23 June 2021; Revised 9 October 2021; Accepted 18 October 2021; Published 1 November 2021

Academic Editor: Zhenhua Tan

In skeleton-based human action recognition methods, human behaviours can be analysed through temporal and spatial changes in the human skeleton. Skeletons are not limited by clothing changes, lighting conditions, or complex backgrounds. This recognition method is robust and has aroused great interest; however, many existing studies used deep-layer networks with large numbers of required parameters to improve the model performance and thus lost the advantage of less computation of skeleton data. It is difficult to deploy previously established models to real-life applications based on low-cost embedded devices. To obtain a model with fewer parameters and a higher accuracy, this study designed a lightweight frame-level joints adaptive graph convolutional network (FLAGCN) model to solve skeleton-based action recognition tasks. Compared with the classical 2s-AGCN model, the new model obtained a higher precision with 1/8 of the parameters and 1/9 of the floating-point operations (FLOPs). Our proposed network characterises three main improvements. First, a previous feature-fusion method replaces the multistream network and reduces the number of required parameters. Second, at the spatial level, two kinds of graph convolution methods capture different aspects of human action information. A frame-level graph convolution constructs a human topological structure for each data frame, whereas an adjacency graph convolution captures the characteristics of the adjacent joints. Third, the model proposed in this study hierarchically extracts different levels of action sequence features, making the model clear and easy to understand; further, it reduces the depth of the model and the number of parameters. A large number of experiments on the NTU RGB + D 60 and 120 data sets show that this method has the advantages of few required parameters, low computational costs, and fast speeds. It also has a simple structure and training process that make it easy to deploy in real-time recognition systems based on low-cost embedded devices.

1. Introduction

Human action recognition can be used in various scenes, such as video retrievals and human-computer interactions [1], so it has been widely discussed in the literature. However, the diversity and complexity of human behaviours have introduced great challenges to the task of human action recognition. Biological research has shown that even in the absence of appearance information, it is possible to distinguish among action categories by analysing joint movements [2]. Skeleton data comprise the three-dimensional position data of several key joints in the human body, characterising rich depth information [3]. These data are not affected by the clothes worn by subjects, the lighting conditions, or environmental noise. Additionally, these data have strong robustness and can express advanced human movement characteristics. With the advent of 3D cameras, such as Kinect cameras [4], skeleton data have become easy to obtain, and action recognition studies based on skeleton data have attracted more attention and made great progress [5], becoming an important branch of human action recognition research.

In the initial stage, due to the limitations of data sets, skeleton-based human action recognition researchers have mainly used manual feature-extraction and machine-learning methods. Since Shahroudy et al. [6, 7] established the NTU RGB + D data set, a large-scale data set for 3D human activity analyses, deep learning has been widely used in skeleton-based human action recognition studies. The existing research has been divided mainly into two directions: models based on convolutional neural networks(CNNs) [8–12] and models based on recurrent neural networks (RNNs) [13–18]. CNN based-methods regard the X, Y, and Z coordinates of joints as image channels, whereas the frame number and joint number of each action sequence are regarded as the length and width of the corresponding image, respectively. RNN-based methods consider the time series characteristics of human behaviour and use RNNs to model these behaviours over time.

However, skeletons are non-Euclidean structural data in which joints are disordered. Different joints have different neighbouring nodes connecting the human skeleton. If joints are input into a convolutional network sequentially, the information obtained for the joints near any given node may not be adjacent in a real human skeleton. Therefore, it is difficult to extract local and global joint features using traditional convolution methods. Recurrent neural networks carry out only temporal modelling and cannot fully express the spatial information of joints in skeletons. Graph convolutional neural network (GCN) is a new type of convolutional neural network. Yan et al. first applied a graph convolution method in a skeleton-based human action recognition study [19] and proposed a spatiotemporal graph convolutional network (ST-GCN). The ST-GCN model constructs the spatial structure of the human skeleton according to the adjacency between two joints in the human body, significantly improving the recognition performance of the model and reflecting the applicability and superiority of the GCN in this task. Graph convolution has gradually become a mainstream research method for skeleton recognition, and researchers have carried out specific research based on the idea of graph convolution [20–32]. Combining the graph convolution with excellent network structures, such as attention networks [33, 34] or residual networks [35, 36], can further improve the human skeleton recognition accuracy.

The current mainstream researches based on ST-GCN improve the recognition accuracy of skeleton recognition task by multistream input [37], adding optimization module, improving loss function [38], improving convolution kernel [24, 39], and increasing attention [34]. These methods make the network deeper and the structure of each layer more complex; they often introduce many parameters and extremely difficult training processes and frequently require many computing resources and long training times. Additionally, these methods not only place high demands on the computing performance of the utilised equipment but also take a long time to predict action sequences in practical applications. Therefore, these models are difficult to be applied to real-time recognition applications based on low-cost embedded devices.

To solve the problems described above, this study proposed a lightweight hierarchical model called a frame-level joints adaptive graph convolutional network (FLAGCN). The hierarchical model consists of four parts: the data-processing level, point level, spatial level, and temporal level. There are six core layers, namely, the coordinate embedding layer, three frame-level joints adaptive graph convolutional (FLAGC) layers, and two CNN layers. The FLAGCN not only ensures a high recognition accuracy but also greatly reduces the required parameters and computational complexity of the model, thus reducing the training time and prediction time of the model and providing a solution for building a real-time recognition system. The main contributions of this study are as follows.

Three mainstream features (bones and the relative positions and motions of joints) are acquired and fused early in the modelling process, replacing the traditional multistream network. The model inputs can obtain useful discriminant information and reduce the required training parameters and computational costs. In addition, feature generation is integrated into the model, thus avoiding the extra previous feature generation operation.

The proposed model uses a three-layer frame-level joints adaptive graph convolution method to capture human motion information from two aspects: a frame-level graph convolution and an adjacent graph convolution. The frame-level graph convolution method adaptively constructs different graphs for each data frame of each action sequence and captures the spatial characteristics of each frame. The adjacency graph convolution method uses a predefined adjacency matrix to capture the relationships between adjacent joints and fully utilises the prior information characterising the human skeleton. The combination of these two graph convolution methods improves the ability of the proposed model to extract spatial features.

In this study, the features of skeleton sequences are extracted hierarchically. In contrast from the spatiotemporal graph convolution layer, spatial and temporal features are extracted at each layer. In the model proposed in this study, the three-dimensional coordinate features of joints are mainly extracted at the point level, whereas the spatial features of all joints in each frame are extracted at the spatial level and the temporal features of the whole sequence are extracted at the temporal level. Therefore, the model is simple, clear, and easy to understand. The ablation experiment confirms that the layered feature-extraction process utilised in this model can effectively improve the recognition accuracy of skeletons with a small number of required parameters.

2. Related Work

2.1. Skeleton-Based Action Recognition. In traditional methods, machine learning is used to solve human action recognition tasks based on human skeletons. For example, Vemullapally et al. [40] used a combination of dynamic time warping, the Fourier time pyramid, and a linear support vector machine (SVM) to classify skeletons. Zanfir et al. [41] expressed each action by its associated joint velocity and

acceleration in the key frame and classified the actions using an improved k-nearest neighbour (KNN) classifier integrated with global time information. Continuously progressing deep learning methods have shown excellent data-processing abilities and allowed breakthrough progress to be made in the computer vision and natural language processing fields. With the emergence of large data sets [6, 7], deep learning has also been used for human action recognition based on skeletons. For example, Li et al. [8] proposed a hierarchical co-occurrence feature-learning framework based on the global aggregation capability of CNNs. They learned the point-level features of each joint independently and fused the motion features with a double-flow frame. Nie et al. [9] proposed two descriptors and input them into a CNN network. Pan et al. [17] constructed a dual-stream, long short-term memory (LSTM) network to extract multilevel attitude and trajectory features. Zheng et al. [18] introduced a recurrent relational network and designed an organic framework to simultaneously simulate the spatial allocation and temporal dynamics of joints. These works have achieved improved performances compared with previously utilised methods.

2.2. Graph Convolutional Network. Graph convolutional networks, which have arisen as a new network form in recent years, show advantages in unstructured data processing and are widely used in traffic flow predictions, network node classifications, and molecular activity predictions in biochemistry [42]. Inspired by these advantages, Yan et al. [19] proposed a spatiotemporal graph convolution method in which every joint in the human skeleton corresponds to every node in a skeleton graph, and the connections between joints are defined as the edges of the skeleton graph. There are two types of edges in action sequences. Spatial edges refer to natural joint connections; these edges are thus predefined by an adjacency matrix characterising human joints. Temporal edges refer to the virtual connections of the same joints between adjacent frames and are simulated by the selected temporal convolution method. Shi et al. [20] proposed a dual-stream, adaptive, graph convolutional network (2s-AGCN) that trains and updates the skeleton graph structure together with the convolutional parameters of the model. This data-driven method improves the flexibility of the resulting graph. At the same time, to utilise the second-order information (the lengths and directions of bones) of the skeleton data, the model adds bones as inputs in another stream. The lengths and directions of the bones are expressed as vectors pointing from the source joints to the target joints. This method compensates for the shortcomings of the ST-GCN predefined graph, such as its lack of flexibility and inclusion of only first-order information, and achieves a better recognition effect. In recent years, some researchers have devoted themselves to optimizing the structure of the skeleton graph to improve the utilised networks based on graph convolution methods, whereas others have combined additional theories with graph convolution. For example, the dynamic framework proposed by Ye et al. [27] takes advantage of both GCNs and CNNs. The shift GCN designed by Cheng et al. [28] is composed of a spatial-shift graph convolution method and a temporal-shift graph convolution method, and its computation costs are greatly lowered. Si et al. [36] proposed the attention-enhanced graph convolutional LSTM (AGC-LSTM) network, representing the first attempt to combine graph convolution with LSTM for the task of human action recognition. Zhao et al. [43] combined graph convolution with LSTM and further extended the network to a probability model following a Bayesian framework. Peng et al. [31] constructed a graph convolutional network using a neural architecture search.

Some of the methods mentioned above require large numbers of parameters and deep networks or a lot of calculations. If the model is applied to low-cost embedded devices with limited memory or computing power, it is difficult to ensure good real-time recognition performance. The frame-level, adaptive, graph convolutional model proposed in this study combines the advantages of frame-adaptive graphs and adjacency matrices to extract spatial features and uses a simple network and lightweight model to realize the high-precision recognition of human actions based on skeletons. The model can be adapted to such embedded devices with small cost.

3. Methodologies

In this section, the proposed model is introduced in three parts. The first part describes the feature fusion at the point level. The second part introduces the details of the frame-level, adaptive, graph convolutional layer used in the spatial layer, focusing on two graph convolutional mechanisms. In the third part, we analyse the proposed hierarchical feature-extraction model and introduce the data-processing level and temporal level.

3.1. Point Level: Early Feature Fusion. Although neural networks can autonomously learn data features, many studies have indicated that early feature processing can improve the performances of models, so it is necessary to select distinctive features [44–48]. For example, inspired by the Lie group-based skeleton descriptor [44], Jiang et al. [16] proposed a spatiotemporal skeleton transformation descriptor (ST-STD) to define the relative transformations of skeleton gestures, including rotation and translation during skeleton movement. Ahad et al. [45] used the linear joint position feature (LJPF) and angular joint position feature (AJPF) obtained based on the three-dimensional linear joint positions and angles between skeleton segments as distinctive features. Nie et al. [9] proposed two new viewpoint-invariant motion features: the Euler angle of joints (JEAs) and the Euclidean distance matrix between joints (JEDM). Li et al. [23] chose a total of six data modalities (joints, bones, their motions, and their relative positions) and independently fed these modalities into the network with a six-stream input.

Later, bones, and the relative position and motion information of joints became common features in skeleton-based action recognition because they are easy to obtain and

have a strong discrimination ability [18, 21, 23, 26, 29, 30, 44]. Therefore, we first generate these three features at the data-processing level of the model.

The relative position of a joint is obtained by subtracting the coordinates of the joint centre from the coordinates of any other joint. This value can be calculated using equation (1), where a is an arbitrary joint and c is a central joint. Because the distances and angles between the skeleton and observation points are uncertain, the relative positions of joints can be used to reduce the influence of position changes among people and observation points. If the centre of the frame was subtracted from each joint, the motion information of the central joint would be lost; considering this, we determine the middle of the spine in the first frame of a given action sequence as the central joint.

$$P = (x_a, y_a, z_a) - (x_c, y_c, z_c). \tag{1}$$

Bones refer to the edge vectors formed by the natural connections within the human body. In our model, 25 bone vectors defined in 2s-AGCN [20] are used. Each bone is calculated by the vector difference between the two joints constituting the bone, as shown in equation (2), where t is the target joint node and s is the source joint node.

$$B = (x_t, y_t, z_t) - (x_s, y_s, z_s). \tag{2}$$

The motion information of joints is obtained by calculating the coordinate differences between the adjacent frames representing the same joints as shown in equation (3), where t_2 represents the frame following t_1. The empty frame at the end is filled with the value of 0, causing less computation and a simpler operation than the interpolation frame-alignment method. Because the time interval between two adjacent frames is fixed, the motion information can indicate not only the change in joint position but also the speed of the joint motion.

$$M = (x_{t1}, y_{t1}, z_{t1}) - (x_{t2}, y_{t2}, z_{t2}). \tag{3}$$

The two-stream or multistream networks used in some researches [8, 14, 18, 20, 21, 23, 30] have achieved good performances, but they have also increased the required numbers of model parameters. Therefore, this article embeds bones and the relative positions and motions of joints into a high-dimensional space at the point level and then fuses these three features without multiplying the parameters. This data fusion can be expressed as follows:

$$\text{input} = \text{embed}(P) + \text{embed}(B) + \text{embed}(M). \tag{4}$$

In equation (4), the bones and relative positions and motion information of joints are described as in equations (1), (2), and (3) and embed $(\cdot)$ represents the embedding operation, which is composed of convolution operations with two convolution kernels with sizes of 1×1, similar to a dense layer; the operation realized by each layer is shown in the following:

$$x^l = \max\left(\left(W^l x^{l-1} + b^l\right), 0\right), \tag{5}$$

where x^{l-1} is the output of the upper layer, x^l is the output of the current layer, W^l is the weight, b^l is the bias, and Max is the rectified linear unit (ReLU) activation function.

The relative positions of joints represent first-order information, whereas the bones and joint motions represent second-order information. The early feature-fusion method described above combines the advantages of these three features at different levels to obtain distinctive features and avoid the use of multistream networks. The data-processing and point level details of our hierarchical model are shown in Figure 1. The original skeleton is directly input into the model, and three features are generated at the data-processing level and input to the point-level feature-extraction layer. After being embedded separately, these features are added to the model. Additional data-processing layer details are provided in-depth in Section 3.3.

3.2. Spatial Level: Frame-Level Graph Convolution and Adjacent Graph Convolution Methods. In traditional skeleton-based human action recognition methods, the skeleton is treated as structured data similar to an image, and the spatial relationships between joints are ignored. The ST-GCN introduced a graph convolutional neural network and defined a spatiotemporal skeleton sequence composed of nodes and edges, where nodes refer to the joints in the skeleton and edges are divided into two categories. In the same frame, the connecting relationships between human joints are considered as the first edge type, representing spatial information, and these connections are represented by an adjacency matrix. For the same joints, the connections between adjacent frames are considered to be the second edge type and are used to extract temporal information. The ST-GCN uses an adjacency matrix to perform graph convolution and extract spatial information. The graph convolution is realized using the following equation.

$$f_{\text{out}} = \sum_{k}^{k_v} W_k \cdot \left(f_{in} \cdot A_k\right) \odot M_k, \tag{6}$$

where f_{in} is the input of a given spatiotemporal graph convolutional layer, f_{out} is the output of the corresponding layer, W_k stands for the weight, A_k is the adjacency matrix, M_k is the attention mask, and K is the subset category number. In the ST-GCN, three connection mode subsets are identified: self-connection, centripetal connection, and centrifugal connection. The adjacency matrix A_k is determined using the connections between joints. The corresponding positions of connecting joints in the skeleton are defined as 1, and joints without connections are defined as 0. The spatial connections between joints are determined through multiplication operations within the adjacency matrix. The temporal connections are realized by a convolution operation in the time dimension.

ST-GCN directly multiplies M_k and A_k by their corresponding elements. If some elements in A_k have values of zero, the final multiplication result is zero regardless of the remaining values. This means that if a connection between two joints does not exist in the original skeleton, the network

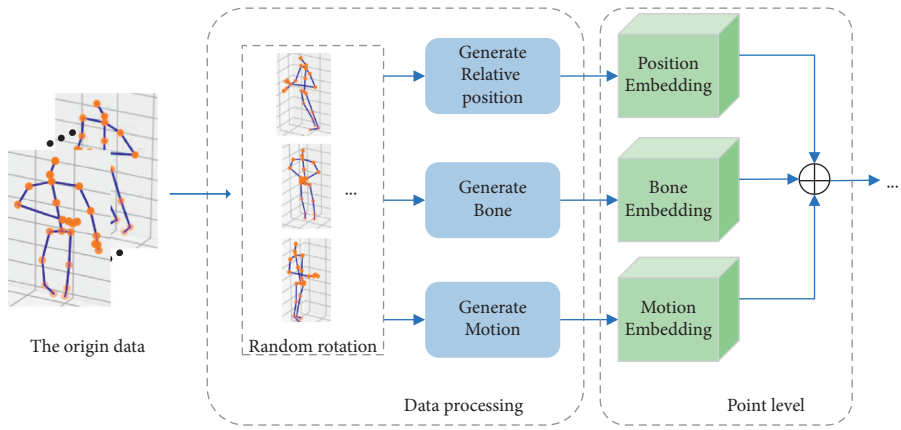


Figure 1: Details of data-processing level and point level.

ultimately does not produce this connection. However, in some behavioural actions, two unconnected joint nodes have notable relationships. For example, during actions such as "drinking water" and "eating," great correlations exist between the hands and head. However, the hands and head are not directly connected, and it is thus difficult for the network to capture this correlation. In view of this joint edge limitation in the ST-GCN, the 2s-AGCN adds an unconstrained, parameterized adjacency matrix (B_k) and an independently graph-calculated matrix (C_k) for each sample, enhancing the flexibility of the model. Their method is shown in the following equation:

$$f_{\text{out}} = \sum_{k}^{K_v} W_k \cdot f_{in} \cdot (A_k + B_k + C_k). \tag{7}$$

Compared with equation (6), equation (7) adds the sample-level adaptive parameters B_k and C_k; the other parameters are the same as those in equation (6). However, the addition of the adjacency matrix A_k, parameterized matrix B_k, and sample-level matrix C_k leads to the loss of spatial information. The 2s-AGCN does not consider the variations in graph variation among the different frames in each sample. In fact, during the process of each action, different frames show different graph characteristics. Therefore, we use the frame-level, adaptive, graph convolutional layer on the spatial layer of the model to capture the spatial features. Each frame-level, adaptive, graph convolutional layer includes two branches: the frame-level adaptive graph convolution branch and the adjacent graph convolution branch, in which predefined graphs are used. The whole calculation mechanism of the FLAGC layer is as follows:

$$f_{\text{out}} = W_k \cdot f_{in} \cdot G_f + \sum_{k}^{k_v} W_a \cdot f_{in} \cdot G_a. \tag{8}$$

The first half of equation (8) is part of the frame-level graph convolution, W_k is the weight of the graph convolution, and G_f is a frame-level graph of the action sequence. G_f is similar to the C_k term in the 2s-AGCN and uses a classical Gaussian embedding function to capture the similarity between joints. In contrast from the 2s-AGCN, we preserve the graph information of each frame and call this information the frame-level graph matrix. The calculation method is shown in the following equation:

$$G_f = \text{softmax}\left((W_\theta \cdot f_{\text{in}})^{P1} (W_\varphi \cdot f_{\text{in}})^{P2} \right), \tag{9}$$

where f_{in} is the input matrix with the shape of $n \times c \times v \times t$ and W_θ and$W_\varphi$are the weights of two embedding layers. The embedding operation used here is the same as that used at the point level and consists of two convolutional layers with a convolution kernel size of 1×1. $P1$ and $P2$ represent two different transposes. The obtained $G_f$term is a frame-level similar graph matrix with a scale of $n \times t \times v \times v$. The frame-level graph does not use prior information but adaptively trains the corresponding graph structure at each frame of each sample and extracts the spatial features of each frame in the skeleton.

The second half of equation (8) is the adjacency graph convolution module. The adjacency matrix, G_a, in the module consists of three matrices ($K_v = 3$). The first matrix represents the relationships between joints with distances of zero, i.e., the autocorrelation of joints. The second matrix represents the correlations between joints with distances of one. The third matrix represents the correlations between joints with distance of twos. Thus, adjacent features with different distances are extracted from these three matrices. The values of connected positions within the matrix are 1, and the values of the nonconnected positions are 0. The regularization process is shown in the following equation:

$$G_a = \Lambda^{-1/2} \cdot A \cdot \Lambda^{-1/2}, \tag{10}$$

where A is the adjacency matrix, which is defined according to the bone analysis, and Λ is used to normalize A; the adjacency graph matrix adds the known skeleton information to the spatial layer, making full use of prior information to further help the spatial layer extract more spatial features.

Figure 2 shows the overall architecture of the FLAGC layer, and the calculation details are shown in Figure 3. The upper branch of Figure 2, corresponding to the left half of Figure 3, shows the frame-level graph convolution module. The module calculates the corresponding frame-level graph

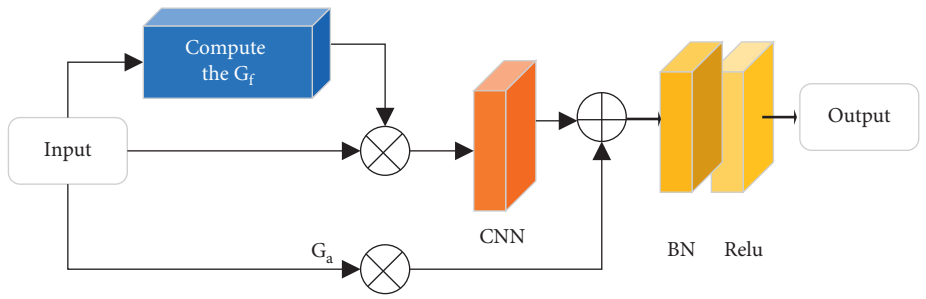


Figure 2: Architecture of FLAGC layer.

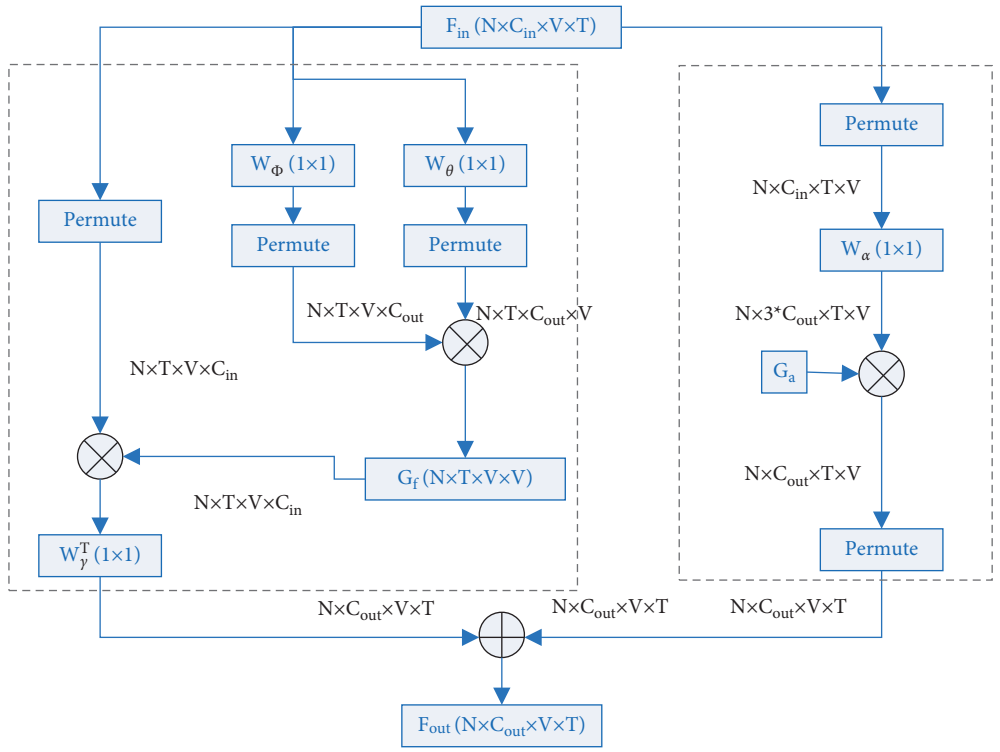


Figure 3: Calculation details of FLAGC layer.

matrix from the input samples and then performs a graph-convolution operation. The lower branch of Figure 2, corresponding to the right half of Figure 3, shows the adjacency graph convolution module, which performs the corresponding graph-convolution operation with the input information using the predefined adjacency matrix G_a. The FLAGC layer performs these two kinds of graph convolutions in parallel mode, making full use of the information contained in the samples and of the prior information.

3.3. Hierarchical Model: A Simple and Accessible Lightweight Model. Our proposed hierarchical model consists of four main parts, the data-processing level, point level, spatial level, and temporal level, as shown by the dotted box in Figure 4. The six core layers described above are marked with the Roman numerals I-VI, the coordinate embedding layer exists at the point level, the three-layer FLAGC exists at the spatial level, and the two-layer CNN exists at the temporal level.

In the data-processing layer, the skeleton rotates in the vertical direction. In practice, observed actions may not be collected with the subject completely facing the camera, and randomly rotating the skeleton is equivalent to increasing the amount of data sourced from different perspectives, playing a role in enhancing the data [11, 26, 41, 43].This random rotation operation is performed according to the following equation:

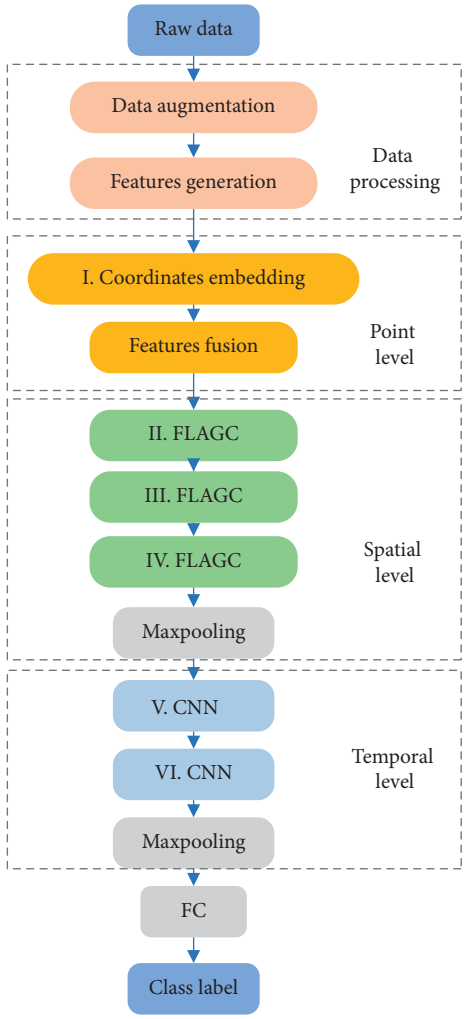


Figure 4: Illustration of the overall architecture of the model.

$$(x_r, y_r, z_r) = (x_o, y_o, z_o) \cdot \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (11)$$

where γ stands for the random rotation angle, o stands for the original coordinates, and r stands for the rotation coordinates. Subsequently, three features are calculated and generated by equations (1), (2), and (3). The feature-generation method integrated into the model reduces the workload of early data preparation and turns the model into an end-to-end recognition system.

In the spatial layer, the three-feature-fused data are taken as the inputs, the spatial information of the skeleton is extracted using three FLAGC layers, and the spatial dimensions of output data are converted into one dimension by a pooling layer to complete the spatial feature extraction.

In the temporal layer, the temporal features in the skeleton sequence are extracted continuously by two convolutional layers, and the kernel size of the convolutional layer is 1×3. Convolution with a kernel size of $1 \times N$ is similar to one-dimensional temporal convolution, in which the information is convoluted only in the temporal dimension; this convolution method can be expressed by the following equation:

$$x_j^l = \sum_i x_i^{l-1} \cdot k_{ij}^l + b_j^i, \quad (12)$$

where x_i^{l-1} is the input of the upper layer, k_{ij}^l is the convolution kernel, and b_j^i is the bias. Each CNN is followed by BN and ReLU layers. After the two-layer convolution, AdaptiveMaxPool2d pools the temporal dimension of the output data into one dimension to complete the spatial and temporal information extraction, and the dense dual-layer completes the final action classification. The model hierarchically extracts action sequence features with different dimensions in a process that is clear and easy to understand. Compared with the ST-GCN [19], which adopts spatio-temporal layers to simultaneously extract spatial and temporal features, the method proposed herein simplifies the model structure, number of layers, and computational costs.

4. Experiment

4.1. Data Sets. NTU RGB+D 60 data set [5] is one of the earliest available, large-scale, multimodal, human action recognition data sets and was created in 2016. It contains RGB videos, depth information, skeleton information (the three-dimensional positions of 25 main joints), and infrared data. The 60 actions contained within the data set were collected from 40 subjects, and three Kinect v2 cameras were placed in 17 different shooting positions. This data set solves the problems associated with the use of a single visual angle, limited action categories, and changeless backgrounds that often arise in human action recognition studies based on deep learning. This data set provides two evaluation criteria: the cross-subject (CS) and cross-view (CV) criteria. The CS tasks include 40,320 training samples and 16,560 test samples; these samples are divided by categorizing the 40 subjects into two groups. The CV training video samples are collected by cameras 2 and 3 (37,920 samples), and the videos collected by camera 1 (18,960 samples) are regarded as the test set.

NTU RGB+D 120 data set [6] is an extended version of NTU RGB+D 60 and was created in 2019; an additional 60 actions and 57,600 samples are included in this extended data set. The resulting data set contains videos from 155 different camera perspectives. A total of 106 subjects of different ages (10 to 57 years old) and different cultural backgrounds (15 countries) recorded in 96 different environments (different backgrounds or light conditions) are comprised in the data set. A total of 114,480 samples are included; in these samples, the actions are mainly divided into three aspects, daily, medical, and interactive human behaviours, covering the most common behaviours in human life. In the cross-subject evaluation, the 106 subjects are randomly divided into two groups. The fifty-three people in each group are used for either training (63026 samples) or testing (50919 samples). Among the 32 different Kinect setting methods, odd-numbered methods are used for training (54,468 samples), and the rest are used for testing (59,477 samples) in the cross-view evaluation. Notably, the 535 missing samples should be ignored.

4.2. Experimental Details. To align the frame numbers of the action samples, we counted the frame numbers of the NTU RGB + D 120 samples and found that most of the samples were contained within 100 frames except the "reading," "writing," "wearing a jacket," and "taking off a jacket" samples. The proportion of samples within 100 frames was close to 70%. The distribution of sample frame numbers is shown in Figure 5. The horizontal axis represents the frame ranges, and the vertical axis represents the number of samples falling into the ranges. Understandably, many common actions in life are completed in 3 seconds. Therefore, our data-processing layer samples only nonzero frames and randomly and evenly selects 20 frames as training input, as in the study by Zhao et al. [43]; the sampling interval is determined by the following equation: where f_o represents the original total number of frames and f_s represents the standard number of frames to be aligned. After the sampling interval is obtained, we randomly select one frame in each interval. For example, if the total number of data frames is 100, one frame is extracted every 5 frames. Such a random uniform extraction can truly reflect the behaviours and actions contained within the samples and makes the training process easy and fast. In addition, this random extraction method allows each round of training samples to not be completely the same and reduces the overfitting of the training sets.

$$\text{interval} = \frac{f_o}{f_s}, \tag{13}$$

In addition, considering the diversity of the camera perspectives contained in the data sets and practical applications, skeletons are randomly rotate [−30°, 30°] to enhance the data and adapt to changes in perspective at the data-processing level.

We implement this model with PyTorch and train it on a Titan graphics processing unit (GPU). The Adam optimizer is used for the optimization, and the weight decay is set to 0.0001. The initial learning rate is set at 0.001, and this value is decreased by a factor of 0.1 in the 60^{th}, 80^{th}, and 100^{th} rounds. The maximum number of training periods is set to 100. The batch size of the two data sets is 64. The channel of each layer is shown in Figure 6. The embedded layer consists of two convolutional layers, and the size of the cube indicates the size of the data.

4.3. Ablation Study

4.3.1. Fusion Modes of Different Features. With regard to the early fusion method in which three features (bones and the relative positions and motions of joints) are fused, we test various combinations, such as individual features, feature connections, and feature additions. Finally, we confirm that the feature-fusion method proposed in Section 3.1 can be used to optimize the action recognition accuracy.

In Table 1, P stands for the relative position, B stands for the skeleton vector, M stands for the motion information, + stands for the addition operation, and "cat" stands for the point dimension splicing. We try to use "cat" to minimize the number of parameters and finally determine that the method in which three features are added after embedding results in the best accuracy performance. Table 1 shows that the use of multiple inputs can bring about increased accuracies by 1%–5%.

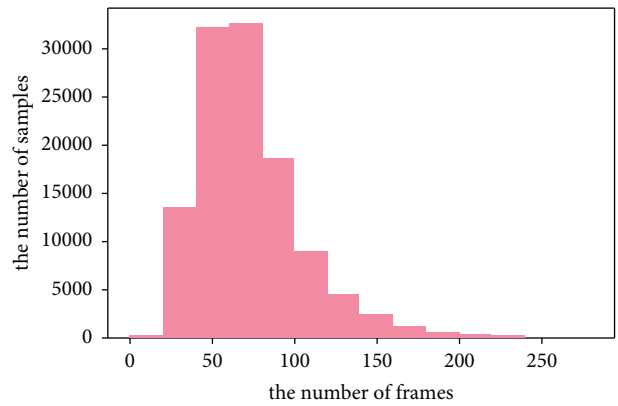


Figure 5: Frame distribution histogram of NTU 120 data set.

4.4. Effectiveness of the Frame-Level Graph Convolution. We make many attempts to explore the validity of two kinds of graph-convolution structures in frame-level adaptive graph convolutional layers. First, we address the frame-level graph convolution structure using two methods to calculate the global graph of each sample instead of the frame-level graph, representing are the maximum and average graph matrix values in the temporal dimension. The results are shown in Table 2.

From Table 2, we can see that the accuracy obtained using the mean value is slightly higher than that obtained using the maximum value. The frame-level graph convolution method achieves the best performance without requiring additional parameters or computational costs (floating-point operations (FLOPs)). The parameter and FLOP units are 10^6 and 10^9, respectively.

To confirm that the two utilised kinds of parallel graph convolution play a positive role in the overall modelling process, we put the two kinds of graph-convolution structures into the spatial layer separately and after superposition, as is conducted in the 2s-AGCN. The results are shown in Table 3, which indicates that the parallel structures of the two convolution operations induce an accuracy improvement of more than 1%.

4.5. Effectiveness of the Hierarchical Model Structure. To further explore the effectiveness of layered feature extraction in the studied model, similarly to the methods used in previous studies [19, 20], we add the temporal feature-extraction layer to each spatial-extraction layer to form a similar spatiotemporal graph scroll layer. The results are shown in Table 4. When more parameters are included, the spatiotemporal graph convolution method performs poorly in cross-object tasks, and its accuracy is 1.6% lower than that of the hierarchical model. To better extract temporal

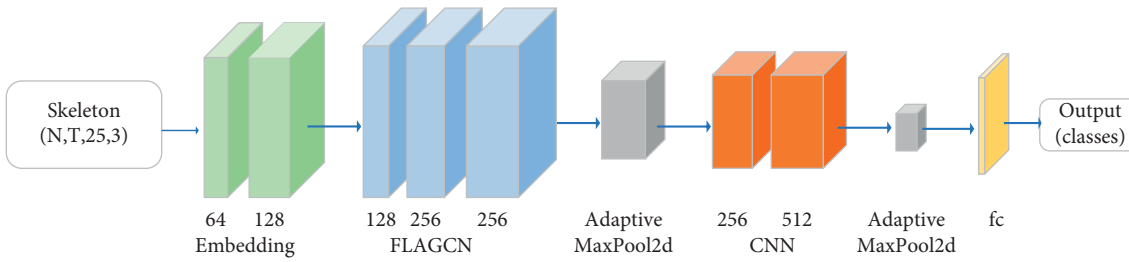


Figure 6: Diagram of data channel changes in the model.

Table 1: Feature fusion partial ablation experiment (based on NTU60).

Methods	Params (M)	CS (%)	CV (%)
P	0.81	84.2	91.6
B	0.81	83.6	90.8
$P+B$	0.82	87.5	93.2
$P+M$ cat B	0.82	87.9	93.6
$P+M+B$	0.83	89.4	94.8

Table 2: Ablation experiments of frame-level map effectiveness (based on NTU60).

Methods	Prams (M)	FLOPs (G)	CS (%)	CV (%)
Mean	0.83	4.1	88.3	93.6
Max	0.83	4.1	87.5	92.9
FLAGCN	0.83	4.1	89.4	94.8

Table 3: Ablation experiments on the effectiveness of two kinds of graph convolution structures (based on NTU60).

Methods	Prams (M)	FLOPs (G)	CS (%)	CV (%)
FLGCN	0.66	3.1	86.6	92.2
AGCN	0.70	3.7	87.8	93.6
(FL + A) GCN	0.83	3.9	88.0	93.4
FLAGCN	0.83	4.1	89.4	94.8

Table 4: Ablation experiments of the hierarchical model, temporal level, and random rotation (based on NTU60).

Method	Prams (M)	CS (%)	CV (%)
Spatial-temporal in spatial level	0.94	88.0	93.9
LSTM in temporal level	0.86	86.6	92.2
GRU in temporal level	0.85	87.3	92.5
Without rotation in point level	0.83	88.6	92.6
FLAGCN	0.83	89.4	94.8

features, we also try to use the LSTM and gated recurrent unit (GRU) methods to construct time modules. Similarly, in cases considering more parameters, the accuracies of both methods are reduced by more than 2%. The third row in Table 4 shows the accuracy of the data-processing layer without random rotation and shows that because random rotation simulates the characteristics of visual angle changes, this situation performs poorly in the cross-visual angle-recognition task.

4.6. Comparison with the SOTA Method. Table 5 shows a performance comparison between our model and other excellent methods based on different networks with the NTU RGB + D 60 data set. The accuracies of our model are 89.4% on CS, which is superior to the accuracy of 0.4% obtained with a previously established method [26], and 94.8% on CV, which is superior to the accuracy of 0.3% obtained with the same previous method [26].

Because the NTU RGB + D 120 data set is new, some established methods have not been tested this data set, and many methods do not provide the number of parameters, calculation amount, and prediction speed. We have obtained some data marked with " $*$ " in Table 6 through our own test. Table 6 compares the accuracies, parameters, FLOPs, and prediction speeds of these models. The parameters, FLOPs, and prediction speeds are all based on the NTU RGB + D 60 data set. The prediction speeds listed in Table 6 represent the average time required for the trained model to predict a given action sequence. The result shows that FLAGCN has accuracies of 81.6% in the CS task and 82.9% in the CV; these accuracies are slightly higher than those of the 2s-AGCN. At the same time, the number of required parameters is reduced to less than 1/8, the calculation cost is reduced to less than 1/9, and the prediction speed is 7 times faster than those of the 2s-AGCN. In Table 6, SGN is shown to have obtained the smallest number of parameters and the fastest speed, but its accuracy is slightly lower than those of other models. The Sybio-GNN achieves the highest accuracy, but its required parameters are numerous. The number of required parameters in ResGCN-N51 is lower than that of our method and its accuracy is higher when applied to the NTU RGB + D 120 data set. But when the method is applied to the NTU RGB + D 60 data set, the accuracy is lower than our method. In contrast from our model, the ResGCN-N51 model uses a parallel extraction structure to obtain spatiotemporal features.

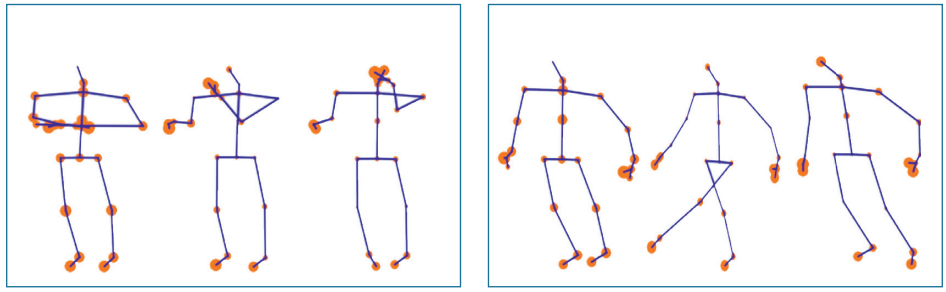
4.7. Visualization of the FLAGCN. To further confirm that the FLAGCN can model the spatial structures embodied in human actions and to explain the model by displaying the features extracted from each layer of the model, we made two visual displays: a presentation of the three-layer FLAGC and a frame-level graph matrix. To make the skeleton appear clearer, we display it in 2D rather than 3D, so there is some occlusion.

Table 5: Performance comparison of NTU60.

	Methods	Year	CS	CV
CNN	HCN [8]	2018	86.5	91.1
	View-invariant CNN [9]	2019	86.7	91.8
RNN	Ind-RNN [13]	2018	81.8	88.0
	2s ARRN-LSTM [14]	2019	81.8	89.4
	AGC-LSTM [36]	2019	89.2	95.0
GCN	ST-GCN [19]	2018	81.5	88.3
	2s-AGCN [20]	2019	88.5	95.1
	SGN [26]	2020	89.0	94.5
Ours	FLAGCN	–	89.4	94.8

Table 6: Comparison of precision (%), parameters (M), FLOPs (G), and prediction speed (ms).

Methods	NTU60 CS	NTU60 CV	NTU120 CS	NTU120 CV	Params	FLOPs	Speed
ST-GCN [19]	81.5	88.3	71.7*	72.4*	3.10*	16.3*	30.9*
HCN [8]	86.5	91.1	73.9*	76.5*	10.02*	1.8*	52.9*
2s-AGCN [20]	88.5	95.1	80.5*	82.6*	6.93*	37.3*	150.3*
SGN [26]	89.0	94.5	79.2	81.5	0.69	3.4	15.2
RA-GCN [49]	87.3	93.6	—	—	6.21	—	41.2
MS-G3D [50]	91.5	96.2	86.9	88.4	6.44	98.0	—
Sybio-GNN [39]	90.1	95.4	—	—	14.85	—	60.3
Tripool [38]	88.0	95.3	80.1	82.8	3.6	11.76	—
MS-AAGCN [37]	90.0	96.2	—	—	15.04*	74.8*	—
DGNN [50]	89.9	96.1	—	—	8.18*	—	41.7*
ST-TR [51]	90.3	96.3	84.3	86.7	4.83	26.26*	—
Shift-GCN [28]	87.8	95.1	—	—	—	2.5	—
ResGCN-N51 [34]	89.1	93.5	84.0	84.2	0.77	—	16.8
FLAGCN	89.4	94.8	81.6	82.9	0.83	4.1	21.5

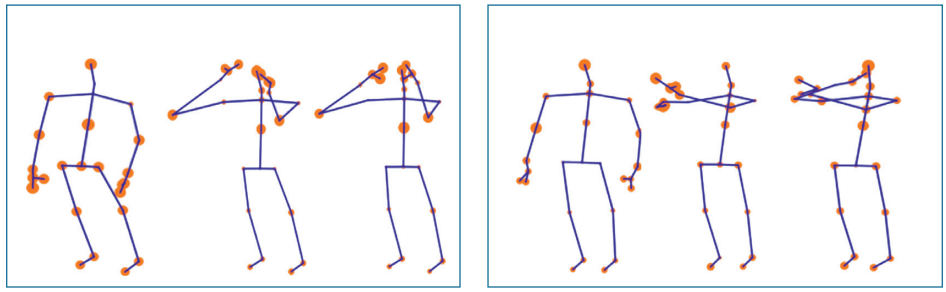


(a) (b)

Figure 7: Visualization of the output of each layer in the spatial level. (a) Wipe face. (b) Kicking something.

First, to confirm that spatial features can be gradually extracted from the three-layer FLAGC, we show the output of each layer at the spatial level. We obtain the shape data with a size of 25×1 by averaging all dimensions except the joint dimensions. The data are normalized and enlarged 100 times as joint size. We choose two representative hand and foot movements as examples, as shown in Figure 7: panel (a) represents the "wipe face" movement, whereas panel (b) displays the "kicking something" movement. The three subgraphs of each action are the output of the first FLAGC layer, the second layer, and the third layer. The weights of all joints in the output of the first layer do not differ extensively, but the weights of the hands and feet increase in the outputs of the second and third layers.

In addition, we visualize the frame-level graph in our proposed frame-level graph convolution, indicating that the weight of each frame in a given action is different and that the FLAGC layer captures this difference. After the previous iteration, the frame-level graph matrix of the third FLAGC layer can most clearly support our viewpoint, so we choose it for the visualisation. In Figure 8, panel (a) displays the "headache" movement and panel (b) shows the "cross hands in front" movement. We choose the figures of the second, tenth, and nineteenth frames of these two action sequences to represent the early stage, middle stage, and late stage of each action. The graph parameters calculated by the network are normalized and expanded to 100 times the size of the corresponding joint. The weights of all joints do not differ extensively in the early stage, but in panel (a), the weights of the hands and head increase in the middle and late stages; panel (b) shows that the head and elbow have more weight in the early stage. The weight of the arms increases in the middle stage, that of the head



(a) (b)

Figure 8: Visualization of the last FLAGC layer parameter. (a) Headache. (b) Cross hands in front.

decreases, and that of the middle part of the spine begins to enlarge due to the arms reaching the vicinity of the middle part of the spine in the later stage. This is consistent with our perspective; that is, the structure of the graph may constantly change during the described action. At the same time, the observed changes in key joints at different movement times also align with common sense. Because the visualisation results of time layer have no practical significance, they are not displayed.

5. Conclusion

This study proposed a lightweight hierarchical model with early feature fusion and frame-level adaptive graph convolution, which can be applied to resource-constrained embedded devices. Our model uses 6-layer network architecture instead of the traditional 9-layer network architecture. The proposed model reduces the number of required parameters and the computational costs of the model and provides a simple method for real-time human skeleton action recognition. In this model, the early feature-fusion process integrates the advantages of multiple-feature utilisation without a multi-stream network. The FLAGCN is divided into two branches to capture spatial information: the frame-level graph convolution branch calculates the graphic structure of each frame, whereas the adjacent graph convolution branch extracts the relationship between adjacent nodes using the adjacency characteristics of the corresponding joints, and the combination of these two graph convolution methods allows spatial information to be extracted in action sequences more comprehensively. The network is designed as a hierarchical, effective, end-to-end model. To test this model, many explorations are made. The final model is verified on the NTU RGB + D 60 and 120 data sets and uses only 1/8 of the parameters and 1/9 of the FLOPs required by the 2s-AGCN while achieving a higher recognition accuracy and faster prediction speed.

Next, we hope to deploy the model in our own embedded system and apply it in a variety of application scenarios, such as display, performance, game, and so on. We will focus on the applicability of the model in different hardware conditions with limited storage and computing performance.

Data Availability

The data sets used in this paper are public, free, and available at https://rose1.ntu.edu.sg/dataset/actionRecognition/.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by Funds for Key Laboratory of Ministry of Culture and Tourism (WLBSYS2005) and the Fundamental Research Funds for the Central Universities (CUC19ZD005).

References

[1] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, no. 6, pp. 976–990, 2010.

[2] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception & Psychophysics*, vol. 14, 1973.

[3] P. Elias, J. Sedmidubsky, and P. Zezula, "Understanding the gap between 2D and 3D skeleton-based action recognition," in *Proceedings of the 2019 IEEE International Symposium on Multimedia (ISM)*, pp. 192–1923, IEEE, San Diego, CA, USA, December 2019.

[4] Microsoft Kinect. https://dev.windows.com/en-us/kinect [OL].

[5] L. Wang, D. Q. Huynh, and P. Koniusz, "A comparative review of recent kinect-based action recognition algorithms," *IEEE Transactions on Image Processing*, vol. 29, pp. 15–28, 2020.

[6] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu RGB+D: a large scale dataset for 3D human activity analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

[7] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "Ntu RGB+D 120: a large-scale benchmark for 3D human activity understanding," in *Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, NW Washington, DC, April 2019.

[8] C. Li, Q. Zhong, D. Xie, and S. Pu, "Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, Stockholm, Sweden, July 2018.

[9] Q. Nie, J. Wang, X. Wang, and Y. Liu, "View-invariant human action recognition based on a 3d bio-constrained skeleton model," *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 3959–3972, 2019.

[10] Y. Zhengyuan, L. Yuncheng, Y. Jianchao, and L. Jiebo, "Action recognition with spatio-temporal visual attention on skeleton image sequences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, pp. 2405–2415, 2018.

[11] F. Meng, H. Liu, Y. Liang, and J. Tu, "Sample fusion network: an end-to-end data augmentation network for skeleton-based human action recognition," *IEEE Transactions on Image Processing*, vol. 99, p. 1, 2019.

[12] Y. Tang, Y. Tian, J. Lu, P. Li, and J. Zhou, "Deep progressive reinforcement learning for skeleton-based action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5323–5332, USA, 2018.

[13] S. Li, W. Li, C. Cook, Ce Zhu, and Y. Gao, "Independently recurrent neural network (indrnn): building a longer and deeper RNN," *Proceedings of the IEEE Conferenceon Computer Vision and Pattern Recognition*, vol. 1, no. 2, pp. 5457–5466, 2018.

[14] D. Avola, M. Cascio, L. Cinque, G. L. Foresti, C. Massaroni, and E. Rodolà, "2-D skeleton-based action recognition via two-branch stacked LSTM-RNNs," *IEEE Transactions on Multimedia*, vol. 22, no. 10, pp. 2481–2496, 2020.

[15] S. Song, C. Lan, J. Xing, and W. Zeng, "Spatio-temporal attention-based LSTM networks for 3D action recognition and detection," *IEEE Transactions on Image Processing*, vol. 2018, Article ID 2818328, 1 page, 2018.

[16] X. Jiang, K. Xu, and T. Sun, "Action recognition scheme based on skeleton representation with DS-LSTM Network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 99, p. 1, 2019.

[17] G. Pan, Y. H. Song, and S. H. Wei, "Combining pose and trajectory for skeleton based action recognition using two-stream RNN," in *Proceedings of the 2019 Chinese Automation Congress (CAC)*, Hangzhou, China, November 2019.

[18] W. Zheng, L. Li, Z. Zhang, and Y. Huang, "Relational network for skeleton-based action recognition," in *Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 826–831, IEEE, Shanghai, China, July 2019.

[19] Y. Sijie, X. Yuanjun, and L. Dahua, "Spatial temporal graph convolutional networks for skeleton-based action recognition," *AAAI*, vol. abs/1801.07455, 2018.

[20] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12018–12027, Long Beach, CA, USA, June 2019.

[21] Z. Bai, Q. Ding, and J. Tan, "Two-Steam fully connected graph convolutional network for skeleton-based action recognition," in *Proceedings of the 2020 Chinese Control And Decision Conference (CCDC)*, Hefei, China, August 2020.

[22] Z. Zhang, Z. Wang, S. Zhuang, and H. Fuyu, "Structure-feature fusion adaptive graph convolutional networks for skeleton-based action recognition," *IEEE Access*, vol. 8, pp. 228108–228117, 2020.

[23] F. Li, A. Zhu, Y. Xu, and R. Cui, "Multi-stream and enhanced spatial-temporal graph convolution network for skeleton-based action recognition," *IEEE Access*, vol. 99, p. 1, 2020.

[24] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with directed graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7912–7921, Long Beach, CA, USA, June 2019.

[25] M. Li, S. Chen, X. Chen, and H. Lu, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3595–3603, Long Beach, CA, USA, June 2019.

[26] P. Zhang, C. Lan, W. Zeng, J. Xing, J. Xue, and N. Zheng, "Semantics-guided neural networks for efficient skeleton-based human action recognition," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1109–1118, Long Beach, CA, USA, June 2020.

[27] F. Ye, S. Pu, Q. Zhong, C. Li, D. Xie, and H. Tang, "Dynamic GCN: context-enriched topology learning for skeleton-based action recognition," in *Proceedings of the 28th ACM International Conference on Multimedia (MM '20)*, pp. 55–63, Association for Computing Machinery, New York, NY, USA, October 2020.

[28] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu, "Skeleton-based action recognition with shift graph convolutional network," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 180–189, Long Beach, CA, USA, June 2020.

[29] D. Sun, F. Zeng, B. Luo, J. Tang, and Z. Ding, "Information enhanced graph convolutional networks for skeleton-based action recognition," in *Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, USA, 2020.

[30] H. Xia and X. Gao, "Multi-scale mixed dense graph convolution network for skeleton-based action recognition," *IEEE Access*, vol. 9, pp. 36475–36484, 2021.

[31] W. Peng, X. Hong, and G. Zhao, "Learning graph convolutional network for skeleton-based human action recognition by neural searching," in *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*, USA, 2020.

[32] L. Huang, Y. Huang, W. Ouyang, and L. Wang, "Part-level graph convolutional network for skeleton-based action recognition," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, pp. 11045–11052, 2020.

[33] Z. Hu and E.-J. Lee, "Dual attention-guided multiscale dynamic aggregate graph convolutional networks for skeleton-based human action recognition," *Symmetry*, vol. 12, no. 10, p. 1589, 2020.

[34] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, "Stronger, faster and more explainable: a graph convolutional baseline for skeleton-based action recognition," in *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 1625–1633, Association for Computing Machinery, New York, NY, USA, October 2020.

[35] C. Wu, X. J. Wu, and J. Kittler, "Spatial residual layer and dense connection block enhanced spatial temporal graph convolutional network for skeleton-based action recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, Seoul, Korea, October 2019.

[36] C. Si, W. Chen, W. Wang, and L. Wang, "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*, pp. 1227–1236, Long Beach, CA, USA, June 2019.

[37] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with multi-stream adaptive graph convolutional networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 9532–9545, 2020.

[38] Z. Qin, Y. Liu, P. Ji et al., "Leveraging third-order features in skeleton-based action recognition," arXiv preprint arXiv: 2105.01563, 2021.

[39] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Symbiotic graph neural networks for 3d skeleton-based human action recognition and motion prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 2021, Article ID 3053765, 2021.

[40] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 588–595, NW Washington, DC, USA, June 2014.

[41] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "The moving pose: an efficient 3D kinematics descriptor for low-latency action recognition and detection," in *Proceedings of the 2013 IEEE International Conference on Computer Vision*, pp. 2752–2759, Sydney, NSW, 2013.

[42] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021.

[43] R. Zhao, K. Wang, H. Su, and Q. Ji, "Bayesian graph convolution LSTM for skeleton based action recognition," in *Proceedings of the International Conference on Computer Vision (ICCV)*, Sydney, NSW, 2019.

[44] C. Caetano, F. Bremond, and W. R. Schwartz, "Skeleton image representation for 3D action recognition based on tree structure and reference joints," in *Proceedings of the 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, October 2019.

[45] M. Ahad, M. Ahmed, A. D. Antar, Y. Makihara, and Y. Yasushi, "Action recognition using kinematics posture feature on 3D skeleton joint locations," *Pattern Recognition Letters*, vol. 145, 2021.

[46] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Learning clip representations for skeleton-based 3D action recognition," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2842–2855, 2018.

[47] J. Liu, A. Shahroudy, Xu Dong, and G. Wang, "Spatio-temporal lstm with trust gates for 3d human action recognition," *ArXiv*, vol. abs/1607.07043, 2016.

[48] A. Haoran, B. Baosheng, A. Kun, L. Jiaqi, and Z. Xin, "Skeleton edge motion networks for human action recognition," *Neurocomputing*, vol. 423, pp. 1–12, 2021.

[49] Yi-F. Song, Z. Zhang, and W. Liang, "Richly activated graph convolutional network for action recognition within complete skeletons," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, October 2019.

[50] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and unifying graph convolutions for skeleton-based action recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 143–152, Long Beach, CA, USA, June 2019.

[51] C. Plizzari, M. Cannici, and M. Matteucci, "Skeleton-based action recognition via spatial and temporal transformer networks," *Computer Vision and Image Understanding*, vol. 208-209, Article ID 103219, 2021.