Spatial Residual Layer and Dense Connection Block Enhanced Spatial Temporal Graph Convolutional Network for Skeleton-Based Action Recognition

Cong Wu [1,2] Xiao-Jun Wu [1,2] Josef Kittler [3]
[1]School of IOT Engineering, Jiangnan University, China
[2]Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence
[3]CVSSP, University of Surrey, UK
congwu@stu.jiangnan.edu.cn, wu_xiaojun@jiangnan.edu.cn, j.kittler@surrey.ac.uk

Abstract

Recent research has shown that modeling the dynamic joint features of the human body by a graph convolutional network (GCN) is a groundbreaking approach for skeleton-based action recognition, especially for the recognition of the body-motion, human-object and human-human interactions. Nevertheless, how to model and utilize coherent skeleton information comprehensively is still an open problem. In order to capture the rich spatiotemporal information and utilize features more effectively, we introduce a spatial residual layer and a dense connection block enhanced spatial temporal graph convolutional network. More specifically, our work introduces three aspects. Firstly, we extend spatial graph convolution to spatial temporal graph convolution of cross-domain residual to extract more precise and informative spatiotemporal feature, and reduce the training complexity by feature fusion in the, so-called, spatial residual layer. Secondly, instead of simply superimposing multiple similar layers, we use dense connection to take full advantage of the global information. Thirdly, we combine the above mentioned two components to create a spatial temporal graph convolutional network (ST-GCN), referred to as SDGCN. The proposed graph representation has a new structure. We perform extensive experiments on two large datasets: Kinetics and NTU-RGB+D. Our method achieves a great improvement in performance compared to the mainstream methods. We evaluate our method quantitatively and qualitatively, thus proving its effectiveness.

1. Introduction

Action recognition is an important foundational work in visual understanding. It has extremely extensive application scenarios in automatic driving, human-computer interaction, crime detection and so on. Different from recognition tasks involving static pictures, video often contains a large number of frames, including many interference or redundant information frames, among which there is a certain spatiotemporal relationship. In other words, action is dynamic and consequently it is also manifested by a motion feature. Action recognition in video is difficult, with many open problems. In this paper we focus on modelling a long sequence of video frames, and on extracting and utilizing spatiotemporal information under the assumption that the imaging conditions do not change rapidly over time.

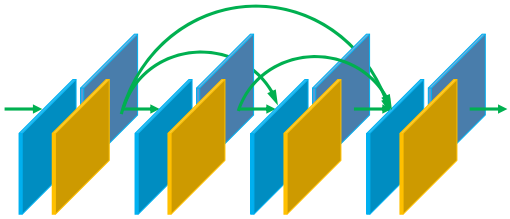

Figure 1. Illustration of our proposed method, which integrates 2D spatial convolution with 1D temporal convolution for spatiotemporal feature representation for a skeleton-based action recognition. The blue squares represent spatial graph convolution, and the yellow represent temporal convolution.

For video understanding, unlike image understanding, which only needs to pay close attention to static spatial characteristics, the modeling of spatiotemporal information is very critical. In order to incorporate spatiotemporal context, different structures are required [32, 43, 2] leading to different types of features. Among these, skeleton features have received considerable attention [41, 26, 16, 3]. For human-related action recognition, the skeleton is a robust source of information. Compared to RGB images or optical flow, the skeleton features have many advantages, such as being easy to calibrate and understand, and being concise, yet powerful. Clearly, if these powerful features are combined with other forms of information by a suitable fusion strategy, the action recognition accuracy may be expected

to improve even further.

How to efficiently exploit skeleton features is still an open problem. [3] used RNNs with a pose-attention mechanism to learn a spatiotemporal representation, but as is well known, RNNs are hard to optimize, and non-Euclidean spatial graph-structured data is incongruent with the input of a simple convolutional network, as it tends to lose the relation information among different joints, which is very crucial for skeleton-based action recognition. The problem of combining skeleton features with the graph convolution was considered in [41, 26, 16]. It is a promissing approach to modeling the non-Euclidean data. The mainstream approach is to use the graph structure to model the joint information of the skeleton, which can then be directly processed by graph convolution. This method is groundbreaking, but there are still many issues that need addressing.

Here we propose a general graph convolutional network structure to process the skeleton features. In this way, we capture the dependencies among joint points. Specifically, we introduce a spatial residual layer to capture and fuse spatiotemporal features. In the previous work, a spatial temporal layer included a spatial graph convolution and a temporal convolution. But serial superposition of different convolutions mixes the information of different domains, leading to inaccurate recognition. By introducing a cross-domain spatial residual convolution, the spatiotemporal information can be enhanced. Furthermore, we propose a dense connection block to extract the global information. It consists of multiple spatial residual layers. Among these layers, the information can be passed by means of dense connections. The benefit of adopting this approach are multifaceted [9]. The final structure developed in this paper consists of serval dense blocks. To verify the effectiveness of the proposed method, we test the model on two datasets: Kinetics [10] and NTU-RGB+D [22]. The experimental results prove that the cross-domain spatial residual layer and the dense connection block in a graph convolutional network bring notable performance gains for skeleton-based action recognition.

The main contributions in this paper are summarized as follows:

- We propose a cross-domain spatial residual layer which captures spatiotemporal information effectively and efficiently; see Fig. 2.
- We propose a dense connection block for ST-GCN to learn global information, and to improve the robustness of features; see Fig. 3.
- The spatial residual layer and the dense connection block enhanced spatial temporal graph convolutional network is comparable or outperforms state-of-art methods on two benchmarking data sets.

2. Related Work

2.1. Action Recognition.

Action recognition is a crucial but complex problem in visual understanding. For the task of action recognition, the mainstream works mainly focus on two aspects: feature selection and model design. Around these two aspects, many excellent methods have emerged.

Regarding features selection, RGB image is the most commonly used feature in static visual understanding. Clearly, it can also be used for video analysis [2, 30, 38]. In order to capture the dynamic characteristics of the video, optical flow is widely used as a feature [32, 43, 28] to represent motion information. Optical flow methods measure the change in pixel values in successive frames of the video. From the correspondence between the previous frame and the current frame, one can estimate the object motion. However, the calculation of optical flow is very time consuming, so some methods attempt to simulate optical flow by measuring image difference. Image difference can be obtained from adjacent frames, or directly from the video encoding information [42, 36, 25], but compared to optical flow, it is not fine enough and contains noise. Other more effective features are being sought. Recently, due to its characteristics, the skeleton feature has attracted the attention of the research community. As some features complement each other in the frame work of a multi-stream network, the accuracy of recognition can further be improved by fusion strategies [32, 43].

For the model design, the structures used for action recognition can be divided into 2D, 3D network and LSTM. Representatives of 2D networks include TSN [32], TRN [43], etc.. The 3D variations include I3D [2], R(2+1)D [30], S3D [38], etc.. However, for a 2D network, it is difficult to extract the information in a time series. Usually, the information is often supplemented by algorithm design and using multiple features [32, 43]; As a 3D network learns the information in space and time at the same time, but the optimization is complicated, and the hardware requirements are challenging. Usually, a lightweight 3D convolution can alleviate this problem to a certain degree [30, 38]. LSTM can also capture the spatiotemporal information [39, 26], but it is hard to train and optimize. Furthermore, it can not capture the graph structure of some specific datasets. Generally speaking, the design of the model and the selection of features are closely related [40, 15]. For a particular feature, we should select the appropriate model. Accordingly, a spatial-temporal graph convolutional network has been proposed for skeleton base action recognition [41, 37]. It has been shown that by optimizing the graph model, the performance can be improved.

2.2. Graph Convolutional Network.

Recently, the graph convolutional network (GCN) has received a lot of attention and has been the subject of in-depth research. GCN is mainly used to model non-Euclidean spatial graph-structured data. The topological graph adopted here uses vertices and edges to establish relationships, in which the number of neighbor nodes is usually not fixed. Through an iterative update of the data during the convolution process, the edges of the graph capture the relationship and structural information between the adjacent nodes. GCN is currently being applied to multiple tasks, such as clustering [35], detection [21], situation recognition [17], point clouds data analysis [7, 34], action recognition [41, 26, 37, 16], and so on.

The essential task of GCN is to extract the spatial features of a topological graph. The spatial domain and the spectral domain are the two main implementations used in the architecture. The spatial GCN [20] first selects adjacent points, and then the subgraphs are normalized so that they can directly be subject of a convolution operation. For the spectral GCN, which defines the Laplace transform and the Laplace inverse transform on the original data matrix, we can compute its Fourier transform by applying the convolution theorem to the graph. Accordingly, the Fourier transform of the convolution of two functions is the product of the Fourier transforms of the two functions. As an example of this approach, [13] introduced the spectral GCN for semi-supervised classification. In this work, we adopt GCN to model the relation among adjacent joints, which can be considered as skeleton features. It is an important method for recognizing human actions and interactions. In order to extract the spatiotemporal information of the skeleton features, and at the same time, keep the calculation efficient, we concatenate a temporal convolution after each spatial graph convolution. This structure is called a spatial temporal graph convolution network, abbreviated as ST-GCN.

3. Background

Before describing our work, in this section, we first introduce the basic concepts of spatial temporal graph convolutional network for skeleton-based action recognition.

3.1. Graph definition

In a graph $G = (V, E)$, V is the set of points, E is the set of edges which connect the adjacent points. The graph G represents the entire data distribution. The information about each data is represented by a specific point in the graph, and each edge between two points in the graph represents correlation between the data points. A is the adjacency matrix of the graph. If the i-th and the j-th points are connected, $A_{i,j} = 1$; Otherwise,$A_{i,j} = 0$. The normalized adjacency matrix of G is defined as $\tilde{A} = D^{-1/2}AD^{-1/2}$, where D is the degree matrix of the graph , $D_{i,i} = \sum_j A_{i,j}$.

For the skeleton data $X \in \mathbb{R}^{n\times d\times T}$, the entire human skeleton constitutes a graph structure, where the joints and the bones of skeleton are points and edges of the graph. In order to model group operations, we define the feature matrix and adjacency matrix. The feature matrix can be expressed as the coordinates information of the joints. During implementation, n is the number of joints in one frame, d is the dimension of the joint spatial coordinates, which can be 2D or 3D coordinates ((x, y) or (x, y, z)), and T denotes the number of frames in one video. $X_t = X_{:,:,t} \in \mathbb{R}^{n\times d}$ is the joint position information at the t-th frame, $X_t^i = X_{i:,:,t} \in \mathbb{R}^d$ is the joint position information of the i-th joint at t-th frame.

3.2. Spatial temporal graph convolution network

Spatial temporal graph convolution network includes many GCN layers. Each layer consists of a spatial convolution and a temporal convolution operation, which model the spatiotemporal feature. They can be seen as 2D convolution and 1D convolution respectively. This form is similar to the R(2+1)D [30] and S3D [38] network, which decouples the 3D convolution into 2D spatial convolution and 1D temporal convolution. In this way, we can not only reduce the computational complexity, but also reduce the entanglement of spatial and temporal information. Thus the spatiotemporal information can be effectively extracted.

Let $X \in \mathbb{R}^{n\times d_{in}}$ be the input feature of the joints, $Y \in \mathbb{R}^{n\times d_{out}}$ be the output feature obtained by the graph convolution operation, where n is the number of data points, and d_{in} and d_{out} are the input and output feature dimensionality. Under normal circumstances, $d_{in} = d_{out}$, the graph convolution operation is generally computed as:

$$Y = \tilde{A}X, \tag{1}$$

$\tilde{A}$ is the normalized adjacency matrix, which can be computed as $\tilde{A} = D^{-1/2}AD^{-1/2}$.

The attention mechanism can also be applied to this expression, and for possible changes in dimensions, an auxiliary matrix is needed. For example,

$$Y = M \circ \tilde{A}XW, \tag{2}$$

where $\circ$ is the Hadamard product, $M \in \mathbb{R}^{n\times n}$ and $W \in \mathbb{R}^{n\times d_{out}}$ are trainable weights to indicate the importance of edges and points of the graph respectively, which can be seen as a simple attention mechanism. W also plays the role of dimensional transformation.

A complete spatial temporal convolution module includes spatial convolution and temporal convolution. Spatial convolution is achieved by the above-mentioned graph convolution. For the temporal convolution, since the skeleton structure in each frame is fixed, that is, the selected joint

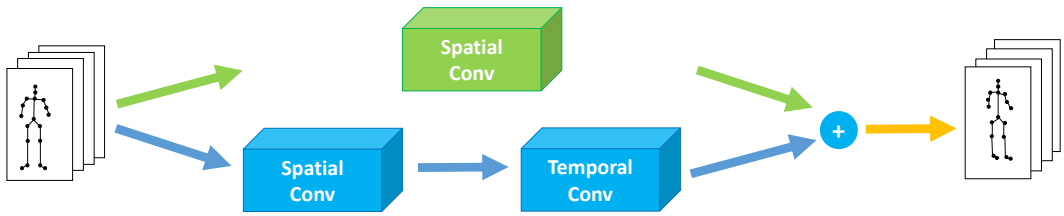

Figure 2. Spatial residual ST-GCN layer. The lower part is the ST-GCN layer, which consists of spatial graph convolution and temporal convolution, and the upper is a spatial graph convolution. Just like in the ResNet, the input of the residual connection is the same as ST-GCN layer, and the output obtained from the residual connection is then added to the output of the ST-GCN layer, the result of the addition is the final output.

points are invariable, we can use the traditional convolution for the temporal convolution operation. Analogous to the convolution operation in the image domain, the feature map here usually corresponds to the feature map $X \in \mathbb{R}^{C\times W\times H}$ (C, W, H are the number of channels, width and height respectively) in an image, which is $X \in \mathbb{R}^{D\times N\times T}$ (D, N, T are the joint feature dimensionality, the number of joint points and the number of frames respectively). In temporal convolution, temporal features are obtained by convolving the same joints of different frames..

4. New Structure of ST-GCN

In this section, we introduce our main work: the spatial residual layer and the dense connection block enhanced spatial temporal graph convolutional network. The proposed structure, SDGCN, is described in detail.

4.1. Spatial Residual Layer (SRL)

The concept of residual connection was first proposed in ResNet [8], by introducing a residual structure, where the input node information is passed through an identity mapping. The idea of residual mapping is to remove the same main part, thus highlight minor changes. By introducing residual mapping, the entire structure is more sensitive to changes in output. The residual layer can be regarded as an amplifier, subject to reasonable settings, the sensitive information is amplified, so the residual connection only needs to care about what it needs to learn.

The GCN with cross domain spatial residual connection, can be regarded as a spatial residual ST-GCN layer, abbreviated as ST-GCN, with SRL, as illustrated in Fig. 2. We introduce a 2D spatial residual structure between the input and output of the spatial temporal GCN layer, to learn the spatiotemporal information more efficiently. Let the temporal convolution kernel be denoted by B, the input feature of this layer by X_{in}, and the output feature by Y. Other symbols are the same as aforementioned. Then we can write

$$Y = B(M \circ \tilde{A}XW) + M \circ \tilde{A}XW. \tag{3}$$

It is desirable to decompose space-time convolution to avoid the prohibitive cost of the calculation, but making it a simple tandem structure lacks the capability to extract joint spatiotemporal features. In order to overcome this problem, we incorporate the residual connection into our network. Unlike in the previous works [8, 5], the spatial residual connection is cross domain. The spatial temporal fusion network is composed of a spatial graph convolution branch and a spatial temporal convolution branch. The identity mapping is the lower stream in the figure. We introduce this cross domain residual connection, acknowledging that video usually contains a lot of redundancy information in the time dimension, which needs to be suppressed. As the spatiotemporal information is different from both the spatial feature, and the temporal feature, the simple solution of superimposing the two convolutions to produce the final result is not very effective. In contrast, the residual connection can help to solve this problem. Unlike the original ResNet, the identity map here is composed of a graph convolution, that can also be seen as a special two-stream structure, where one stream learns static features and the other learns spatiotemporal features. By the means of the 2D spatial graph convolution, the static spatial feature can be extracted. Thanks to the residual connection, the residual map will pay attention to the static spatial information. However, the original layer only needs to attach importance to spatiotemporal information. This design makes GCN learn the important information from video more effectively.

4.2. Dense Connection Block (DCB)

DenseNet [9] is a groundbreaking concept. Its structure is very simple, consisting of several dense connection blocks. In each block, the feature map from each layer is concatenated with all previous features of the same scale. By introducing a dense connection, the features of each layer will be reused. On one hand, using a small amount of computation we obtain a much richer feature map. On the other hand, the reused feature is more robust, so that the dependence between different layers is reduced. However, the excessive GPU memory usage of Dense Connection is

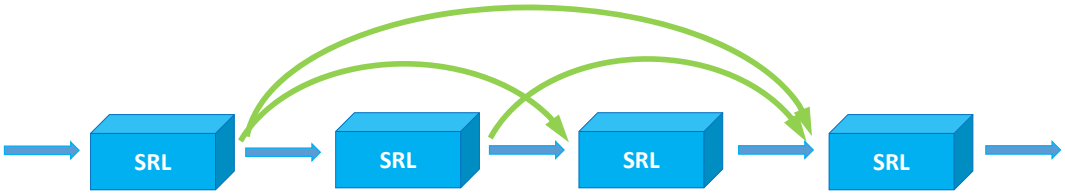

Figure 3. Dense connection block. Each dense connection block consists of several spatial residual layers as described in the previous section. Here, except the first layer or the last layer, the size of the input feature of each layer is exactly the same as the output size of the previous layer, as in the dense block in [9]. Therefore, in addition to the normal sequential connection, the four layers also have three dense connections, as shown in the picture.

a serious problem. Here we introduce a dense connection into the spatial temporal graph convolutional network at the same time, while keeping the structural efficiency.

The ST-GCN with dense connection blocks, which is referred to as ST-GCN with DCB, is illustrated in Fig. 3. In each dense connection block, each layer has been connected to all subsequent layers. Let the l^{th} layer receive all features from its previous layers, $x_0, x_1, ..., x_{l-1}$ as input features of this layer. Let C be the SRL operation. Then for the output of l^{th} layer, we have

$$x_l = C([x_0, x_1, ..., x_{l-1}]), \tag{4}$$

where $[x_0, x_1, ..., x_{l-1}]$ is the concatenation of all the feature outputs from layers $0, 1, \ldots, l-1$. After concatenating them all together in a channel, the final feature is used as the input to the x_l layer. All these operations together make up a dense connection block. Through this block, most information exacted by the previous layers can be reused in later layers. Just as the DenseNet, this block allows the entire network to take full advantage of global information. Above all, from the perspective of features, through the feature reuse and bypass settings, the number of parameters of the network is greatly reduced, and the vanishing gradient problem is alleviated to some extent. On the other hand, the input of each layer includes not only the output from the previous layer, but also other preceding layers. This also improves the robustness of the network.

4.3. Model Architecture

By introducing the spatial residual layer and the dense connection block, we create a generic spatial temporal GCN, as can be seen from Fig. 4.

Here we combine the spatial residual layer and dense connection block together to make the final architecture, denoted as SDGCN. Note that several spatial residual layers make up a block. We introduce the dense connection to connect these layers in each block. The entire network structure consists of 3 dense connection blocks. In each block, the settings of channel size allow us to make full use of dense connections. We adopt the original ST-GCN's settings. This ensures that the proposed method can be applied to the commonly used ST-GCN structure. In order to explore the final impact of dense connections on the model, we set up the above two frameworks, as shown in the Fig. 4. The right model follows a standard setup, where each block consists of four layers and three dense connections. Compared with that, except the first block, each block in the left model has a reduced number of layers. The detailed comparison is carried out in the experimental section. Through this design, on the one hand we try to explore the impact of dense connections on the model, on the other hand, we try to balance the amount of computation and the beneficial impact as much as possible.

5. Experiments

5.1. Datasets

Kinetics. Kinetics [10] is a very large action recognition dataset collected from YouTube, which contains 400 classes, over 240K training samples and 20K validation samples, which have been trimmed. Here we obtain the skeleton data of Kinetics by OpenPose [1] toolbox. Specifically, we first extract frames at video rate (30 frames per second), and resize them to 340×256. We then enter the resized data into OpenPose. The output data of each frame includes two-dimensional coordinates of 18 joint points and a confidence score. If the number of people in a frame is more than two, we choose two people with the highest average confidence score for the joint points.

NTU-RGB+D. NTU-RGB+D [22] contains 56,800 action samples, including three categories: daily actions, mutual actions and medical conditions. These samples are captured by three cameras. The data have 4 modalities: RGB videos, depth map sequences, 3D skeleton and infrared videos. Here we focus on the skeleton data set. The 3D skeleton includes three-dimensional coordinates of the spatial locations of the 25 joint points marked in each frame of the video. Two evaluation standards are recommended for this data set: Cross–Subject and Cross-View. For Cross-

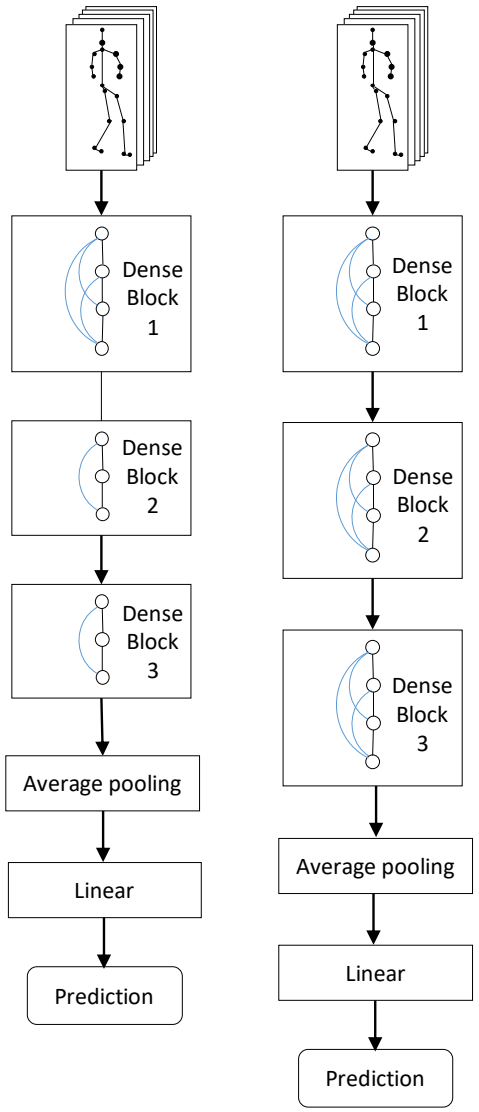

Figure 4. Example Model Architecture. Left: SDGCN based on DCB1. We follow the standard ST-GCN model to design our model. We do not introduce additional structural elements. Right: SDGCN based on DCB2. Compared to left, in order to fully explore the role of dense connections, we introduce more layers and dense connections. We conduct a detailed analysis of their effect in the experimental section. As shown in the figure, for the sake of brevity, SRL is represented as a hollow circle.

Subject, there are 40,320 samples including 20 subjects, which make up a training set. The rest are used as a test set; For Cross-View, there are 37,920 training samples and 18,960 test samples. The samples captured by the first two cameras form the training set. The rest serve as test samples. We conduct experiments on these sets separately.

Methods	Top-1	Top-5
ST-GCN	30.7	52.8
ST-GCN with SRL	33.31	55.9
ST-GCN with DCB1	32.21	54.64
ST-GCN with DCB2	33.18	55.99

Table 1. Ablation study on the Kinetics data set. The SRL means spatial residual layer, DCB means dense connection block (DCB1 and DCB2), as detailed in Fig. 4 . We report TOP-1 and TOP-5 accuracies on the Kinetics data set.

Methods	Cross Subject	Cross View
ST-GCN	81.5	88.3
ST-GCN with SRL	83.25	91.06
ST-GCN with DCB	83.04	90.33

Table 2. Ablation study on the NTU-RGB+D data set. We report the accuracies on the Cross-Subject and Cross-View data. Other symbols are the same as Tab. 1. Only the TOP-1 accuracy is reported.

5.2. Implementation Details

We use the Kinetics and NTU-RGB+D data sets to compare the proposed algorithms with two baseline methods: ST-GCN [41], and 2s-AGCN [24]. The main difference between those models is in the modeling of the graph. The ST-GCN introduces a spatial temporal convolutional network to model the spatiotemporal information of skeleton. 2s-AGCN exploits an adaptive mechanism capable of capturing multi-scale joint information and a two stream processing method based on bone flow and joint flow, which is inspired by Non-Local network [33] and two-stream network [28]. Even though they are different in terms of the graph model, and some other aspects, their main framework is still the superposition of separate layers. The feature scale of the multiple adjacent layers is the same, which contrasts with the versatility of our method. In the experiments, if there is no special mention, we use the original structure. We did all the experiments on two GeForce RTX2080Ti.

5.3. Ablation Studies

In this section, we perform a detailed experimental comparison on Kinetics and NTU-RGB+D to analysis our methods.

Spatial residual layer. We first explore the effectiveness of the cross-domain spatial residual layer, using ST-GCN [41] as a baseline. As described in Section 4.1, compared with the original structure, for each spatial temporal structure, which consist of spatial graph convolution operation and temporal convolution in series, we introduce a spatial residual connection to the original network, abbreviated as SRL, and keep other conditions unchanged. Referring to Tab. 1 2, we find that compared with the baseline

Methods	Top-1	Top-5
Feature Enc. [6]	14.9	25.8
Deep LSTM [22]	16.4	35.3
TCN [12]	20.3	40.4
ST-GCN [41]	30.7	52.8
AS-GCN [16]	34.8	56.5
Js-AGCN [24]	35.1	57.1
Bs-AGCN [24]	33.3	55.7
2s-AGCN [24]	36.1	58.7
DGNN [23]	36.9	59.6
SDGCN (ours)	34.06	56.33
Js-SDGCN (ours)	35.25	58.21
Bs-SDGCN (ours)	35.35	58.16
2s-SDGCN (ours)	37.35	60.38

Table 3. Comparison with the state of the art methods on Kinetics. 'Js' and 'Bs' mean the joint-based and bone-based stream networks, '2s' is the final fusion result.

method, ST-GCN with SRL exhibits a clear improvement. On Kinetics, the performance has increased by 2.61%, and for NTU-RGB+D, the performance has increased by 1.75% and 2.76% for Cross-Subject and Cross-View respectively. This verifies the merits of our SRL structure .

Dense connection block. We also evaluated the performance of the Dense connection block of our network. For the details of the design, the reader can refer to Section 4.2. We call it DCB for short. Dense connection is a very important structure, which has been applied in many different areas. Based on a previous design, we build ST-GCN with DCB. The results achieved with this enhancement are listed in Tab. 1 2. Note, there are two dense blocks, DCB1 and DCB2, as introduced in the previous section. DCB1 is based on the original structure, which consists of 10 layers. In order to demonstrate the role of dense connection, we designed DCB2, which contains 12 layers. In Tab. 1, the performance of ST-GCN with DCB1 has increased by 1.51%, and with DCB2 by 2.48%. Clearly, the dense connection contributes a lot to our network. However, with the increase of dense connections, the number of network parameters increases rapidly. In addition, the blind cumulative network complexity may cause model over-fitting on certain data sets. So in the following experiment, we do not make a distinction, but adopt the DCB1 structure as DCB. As shown in Tab. 2, the performance of ST-GCN with DCB has reached 83.04% in the Cross-Subject and 90.33% in the Cross-View tests.

5.4. Comparison with the State of the Art

Here we compare our method with the state of the art methods. For a comprehensive comparison, we choose to relate our methods to two important baselines: ST-GCN [41] and 2s-AGCN [24]. The first baseline is the ground-breaking work on skeleton based action recognition, and 2s-AGCN is the latest best performing method. We combine the SRL and DCB together to report the final result. The final model is denoted as SDGCN. As shown in the Tab. 3 4, compared to the ST-GCN based method, our method has improved the accuracy to 34.06% on Kinetics, 84.04% and 91.43% in the Cross-Subject and the Cross-View tests respectively. Compared with the original method, the improvements achieved are 3.36%, 2.54% and 2.1% respectively, which have surpassed most methods. When we choose 2s-AGCN as the baseline, the final performance is further improved. On Kinetics, the proposed method has achieved the accuracy of 37.35%. On NTU-RGB+D, the accuracy is 89.58% and 95.74% in the Cross-Subject and the Cross-View data respectively. The model developed is superior to most mainstream approaches.

Methods	Cross Subject	Cross View
Lie Group [31]	50.1	52.8
H-RNN [4]	59.1	64.0
Deep LSTM [22]	60.7	67.3
PA-LSTM [22]	62.9	70.3
ST-LSTM+TS [18]	69.2	77.7
TCN [12]	74.3	83.1
Visualize CNN [19]	76.0	82.6
C-CNN+MTLN [11]	79.6	84.8
ST-GCN [41]	81.5	88.3
DPRL [29]	83.5	89.8
SR-TSL [27]	84.8	92.4
HCN [14]	86.5	91.1
AS-GCN [16]	86.8	94.2
Js-AGCN [24]	-	93.7
Bs-AGCN [24]	-	93.2
2s-AGCN [24]	88.5	95.1
DGNN [23]	89.9	96.1
SDGCN (ours)	84.04	91.43
Js-SDGCN (ours)	87.54	94.34
Bs-SDGCN (ours)	88.10	94.54
2s-SDGCN (ours)	89.58	95.74

Table 4. Comparison with the state of the art methods on NTU-RGB+D. Other symbols are the same as Tab. 3.

6. Conclusion

We have proposed a unified spatial temporal graph convolution network framework, referred to as SDGCN, to improve the performance of skeleton based action recognition. By introducing a cross-domain spatial graph residual layer and a dense connection block, our method fully utilizes spatiotemporal information. It enhances the effectiveness of spatiotemporal information processing. It can easily be incorporated in a mainstream spatial temporal graph network.

Acknowledgement

The paper is supported by the National Science Foundation of China (GRANT No.61672265, U1836218), the 111 Project of Ministry of Edycation of China (GRANT No. B12018), and UK EPSRC (Grant No. EP/N007743/1, MURI/EPSRC/DSTL, EP/R018456/1).

References

[1] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, pages 7291–7299, 2017.

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017.

[3] Wenbin Du, Yali Wang, and Yu Qiao. Rpan: An end-to-end recurrent pose-attention network for action recognition in videos. In *ICCV*, pages 3725–3734, 2017.

[4] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, pages 1110–1118, 2015.

[5] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In *NIPS*, pages 3468–3476, 2016.

[6] Basura Fernando, Efstratios Gavves, Jose M Oramas, Amir Ghodrati, and Tinne Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, pages 5378–5387, 2015.

[7] Lasse Hansen, Jasper Diesel, and Mattias P Heinrich. Multi-kernel diffusion cnns for graph-based learning on point clouds. In *ECCV*, pages 456–469, 2018.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[9] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017.

[10] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[11] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In *CVPR*, pages 3288–3297, 2017.

[12] Tae Soo Kim and Austin Reiter. Interpretable 3d human action analysis with temporal convolutional networks. In *CVPRW*, pages 1623–1631. IEEE, 2017.

[13] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[14] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In *IJCAI*, pages 786–792. AAAI Press, 2018.

[15] Hui Li and Xiao-Jun Wu. Densefuse: A fusion approach to infrared and visible images. *IEEE Transactions on Image Processing*, 28(5):2614–2623, 2018.

[16] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *CVPR*, pages 3595–3603, 2019.

[17] Ruiyu Li, Makarand Tapaswi, Renjie Liao, Jiaya Jia, Raquel Urtasun, and Sanja Fidler. Situation recognition with graph neural networks. In *ICCV*, pages 4173–4182, 2017.

[18] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *ECCV*, pages 816–833, 2016.

[19] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *PR*, 68:346–362, 2017.

[20] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *ICML*, pages 2014–2023, 2016.

[21] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. Learning human-object interactions by graph parsing neural networks. In *ECCV*, pages 401–417, 2018.

[22] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu-rgb+d: A large scale dataset for 3d human activity analysis. In *CVPR*, pages 1010–1019, 2016.

[23] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *CVPR*, pages 7912–7921, 2019.

[24] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, pages 12026–12035, 2019.

[25] Zheng Shou, Xudong Lin, Yannis Kalantidis, Laura Sevilla-Lara, Marcus Rohrbach, Shih-Fu Chang, and Zhicheng Yan. Dmc-net: Generating discriminative motion cues for fast compressed video action recognition. In *CVPR*, pages 1268–1277, 2019.

[26] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *CVPR*, pages 1227–1236, 2019.

[27] Chenyang Si, Ya Jing, Wei Wang, Liang Wang, and Tieniu Tan. Skeleton-based action recognition with spatial reasoning and temporal stack learning. In *ECCV*, pages 103–118, 2018.

[28] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014.

[29] Yansong Tang, Yi Tian, Jiwen Lu, Peiyang Li, and Jie Zhou. Deep progressive reinforcement learning for skeleton-based action recognition. In *CVPR*, pages 5323–5332, 2018.

[30] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, pages 6450–6459, 2018.

[31] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *CVPR*, pages 588–595, 2014.

[32] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36, 2016.

[33] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018.

[34] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829*, 2018.

[35] Zhongdao Wang, Liang Zheng, Yali Li, and Shengjin Wang. Linkage based face clustering via graph convolution network. In *CVPR*, pages 1117–1125, 2019.

[36] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Compressed video action recognition. In *CVPR*, pages 6026–6035, 2018.

[37] Jianchao Wu, Limin Wang, Li Wang, Jie Guo, and Gangshan Wu. Learning actor relation graphs for group activity recognition. In *CVPR*, pages 9964–9974, 2019.

[38] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, pages 305–321, 2018.

[39] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS*, pages 802–810, 2015.

[40] Tianyang Xu, Zhen-Hua Feng, Xiao-Jun Wu, and Josef Kittler. Joint group feature selection and discriminative filter learning for robust visual object tracking. In *ICCV*, pages –, 2019.

[41] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, pages 7444–7452, 2018.

[42] Bowen Zhang, Limin Wang, Zhe Wang, Yu Qiao, and Hanli Wang. Real-time action recognition with enhanced motion vector cnns. In *CVPR*, pages 2718–2726, 2016.

[43] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, pages 803–818, 2018.