Channel-Wise Dense Connection Graph Convolutional Network For Skeleton-Based Action Recognition

***Michael Lao BanTeng*[1,2], *Zhiyong Wu*[1,2]**

[1] Tsinghua-CUHK Joint Research Center for Media Sciences, Technologies and Systems, Graduate School at Shenzhen, Tsinghua University, Shenzhen, China

[2] Department of Computer Science and Technology, Tsinghua University, Beijing, China

1.Introduction

Motivation

- Build an skeleton-based action recognition system
- ✓ Construct global information for action and select more related features
- ✓ Generate and utilize features to adapt for difference on action movements
- ✓ Extracted temporal feature representation

Challenges

- The importance of different channels varies in actions
- Some human actions only involve a small part of bodies
- Confusion on reversing actions

Contributions

- Extract the motion features from skeleton data and concatenating them with original spatial features
- Introduce a channel-wise attention module to emphasize channels with important features
- Use dense connection to ensure reuse of skeleton features and to generate a larger and sufficient features map
- Our model shows competitive performance with the state-of-the-art model on two large datasets, NTU-RGB+D and Kinetics

2. Background

Skeleton Graph

- Skeleton graph *G = (V,E)*
- ✓ Given M frames and N joints of skeleton sequence
- ✓ Joints as vertices (V) and the connection between joints as edges (E)
- ✓ Each vertices contains three channels of information, a two-dimensional coordinate of corresponding joint and its estimation confidence
- Adjacency matrix *A*
- ✓ $A_{i,j} = 1$ if *i*-th joint and *j*-th joint is connected, and 0 otherwise
- ✓ Joints are connected in one frame and adjacent frame

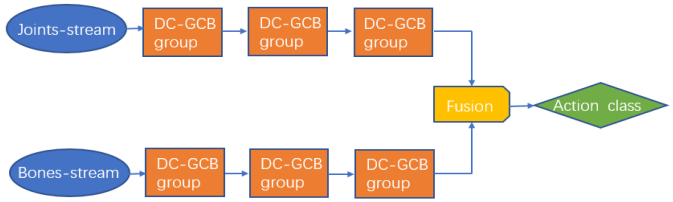
Baseline Model

- Spatial temporal graph convolutional networks (ST-GCN)
- ✓ Convolution operation on spatial and temporal dimension
- ✓ A mapping strategy to determine the size of convolution kernel and weight distribution of convolution on spatial dimension
- ✓ Temporal dimension convolution is similar to classical image convolution
- Two-stream adaptive graph convolutional networks (2s-AGCN)
- ✓ Based on ST-GCCN
- ✓ Introduced an adjacency matrix, the elements of it are parameterized and optimized together in the training process and can be arbitrary values
- ✓ Introduced a similarity matrix, whose elements denotes the similarity of two vertices
- ✓ Generate the connections and their importance between two vertices, which are not existed in the original graph

3. Two-stream Channel-wise Dense Connection GCN(2s-CDGCN)

Data Preprocessing

- Vertice's joint coordinate $v_i = (x_i, y_i)$
- Bone information, extracted from neighboring vertices, $b_{ij} = (x_i - x_j, y_i - y_j)$
- Motion information of joints and bones, extracted from consecutive frames of data, $m_i = (x_i^{t+1} - x_i^t, y_i^{t+1} - y_i^t)$
- Concatenate the information of joints and their motion in the frame dimension. The same procedure was conducted with the bones.



Framework of the proposed 2s-CDGCN

Model Structure

- Two-stream fashion, each stream consists of 12 graph convolution blocks with late fusion
- A graph convolution block (GCB) , consists of a spatial GCN, a temporal GCN and a channel-wise attention module (CAM), followed by a residual connection
- A DC-GCB group, consists of 4 GCBs with Dense connection implemented, followed by a transition layer

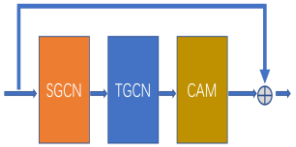


Illustration of GCB

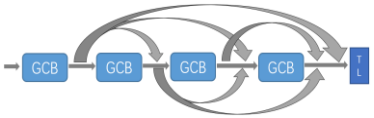


Illustration of DC-GCB group, grey curve arrows denote the dense connections

- Channel-wise attention module (CAM)
- ✓ Encode the entire spatial and temporal feature on a channel as a global feature descriptor
- ✓ Analyze the interdependence between channels, generate a set of attention weights of corresponding channels
- ✓ A channel-wise multiplication is made to represent a global information based on feature channels.
- Dense Connection
- ✓ Concatenation of all the preceding graph convolution block's output features maps
- ✓ Transition layer between blocks to reduce the number of features

Training

- Training batch size 64 and Test batch size 64
- Stochastic gradient descent (SGD) as optimizer with an initial learning rate 0.1 and a cosine learning rate decay
- The weight decay of 0.0001 and Nesterov momentum of 0.9 are set
- The hyperparameter reduction ratio r, used in channel-wise attention module, is set to 16

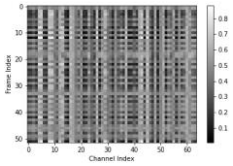
4.Experiments and Results

Dataset

- **NTU-RGB+D**
- ✓ 60 different action classes including daily and health-related actions
- ✓ 25 body joints collected by Microsoft Kinect v2
- ✓ 40 distinct subjects recorded from 3 different horizontal angles
- ✓ Cross-subject evaluation and cross-view evaluation
- **Kinetics**
- ✓ 400 action classes with at least 400 video clips
- ✓ 18 body joints obtained by OpenPose toolbox

Methods	Cross Subject (%)	Cross View (%)
2s-AGCN [4]	88.5	95.1
2s-CDGCN without DC	89.3	95.9
2s-CDGCN without CAM	89.5	95.5
2s-CDGCN	90.0	96.1

Ablation study experiments to validate modules



Example of activations in CAM

Ablation Study

- Channel-wise Attention Module
- ✓ Choose dataset NTU-RGB+D to test the Top-1 accuracy
- ✓ Valid effect on improving the performance of the model
- ✓ The module learns the non-linear relations between channels and the scale is not one-hot encoding
- ✓ Emphasize multiple channels with more importance
- Dense Connection
- ✓ Performance improvement shows that the network takes the advantage of Dense Connection
- ✓ Produces a larger and sufficient features map to achieve better results
- CAM, compared with DC, achieves higher accuracy improvements on cross-view benchmark, and vice versa, which can be explained by the relationship between modification modules and NTU-RGB+D setup
- Comparison with the State-of-the-Art methods
- ✓ Methods include hand-crafted methods [42][49], CNN-based methods [5][44][43][47], RNN-based methods [9][10][45][11] and GCN-based methods[1][14][13][15][48][46][4]

Methods	Top-1(%)	Top-5(%)
Feature [49]	14.9	25.8
Deep LSTM [20]	16.4	35.3
TCN [43]	20.3	40.0
ST-GCN [3]	30.7	52.8
AS-GCN [1]	34.8	56.5
2s-AGCN [4]	36.1	58.7
DGNN [48]	36.9	59.6
GCN-NAS [14]	37.1	60.1
2s-CDGCN	37.0	59.8

Comparison on Kinetics dataset

Methods	Cross-Subject(%)	Cross-View(%)
Lie Group [42]	50.1	52.8
Deep LSTM [20]	60.7	67.3
STA-LSTM [11]	73.4	81.2
TCN [43]	74.3	83.1
C-CNN + MTLN [44]	79.6	84.8
VA-LSTM [45]	79.4	87.6
ST-GCN [3]	81.5	88.3
SR-TSL [46]	84.8	92.4
HCN [5]	86.5	91.1
3scale ResNet152 [47]	85.0	92.3
RA-GCN [15]	85.9	93.5
DenseIndRNN [10]	86.7	93.7
PB-GCN [13]	87.5	93.2
AS-GCN [1]	86.8	94.2
AGC-LSTM [9]	89.2	95.0
2s-AGCN [4]	88.5	95.1
GCN-NAS [14]	89.4	95.7
DGNN [48]	89.9	96.1
2s-CDGCN	90.0	96.1

Comparison on NTU-RGB+D dataset

- Outperforms hand-crafted methods, CNN and RNN methods with a large margin
- A competitive result comparing with the state-of-the-art GCN-based methods