Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition

Sijie Yan, Yuanjun Xiong, Dahua Lin
Department of Information Engineering, The Chinese University of Hong Kong
{ys016, dhlin}@ie.cuhk.edu.hk, bitxiong@gmail.com

GCN lecture 최종 발표
2018-23112 홍석일

Contents

1. Introduction to Human Action Recognition

2. Spatial Temporal Graph Convolutional Network (ST-GCN)

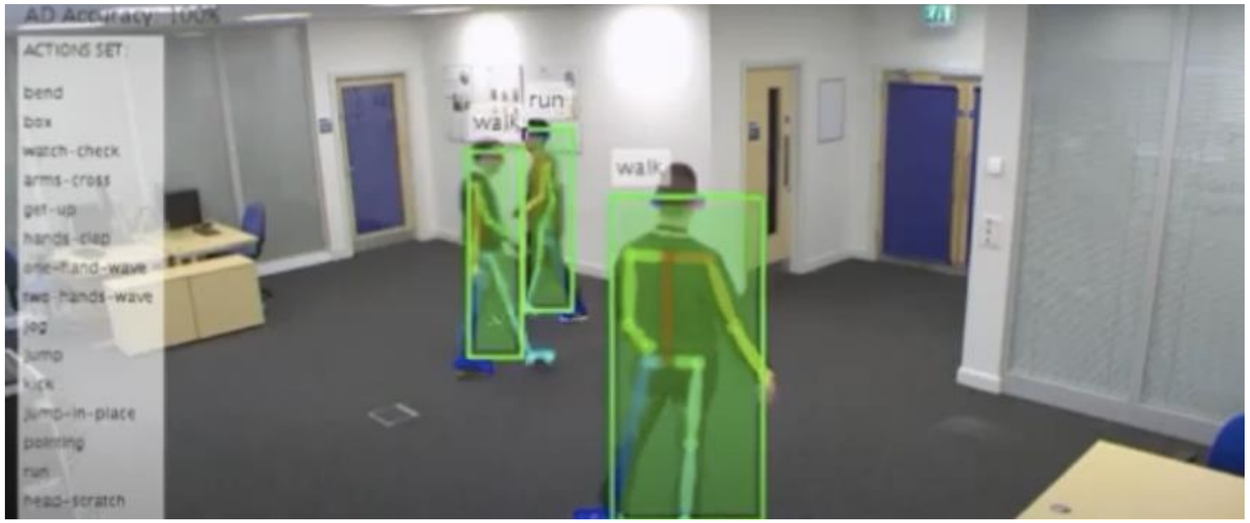
3. Experiments & Ablation study

4. Conclusion & Limitation

5. Project & Results

1. Introduction to Human Action Recognition

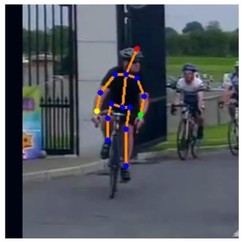
- Input = A (human) video segment containing activities
- Problems : Action classification, Action detection/localization, Action prediction/forecast
- Dataset : UCF101, HMDB, Kinetics, …
- Multiple modalities : Appearance, Depth, Optical flows, and Body skeletons




Climbing
0.9645

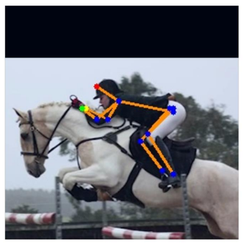

Shooting an arrow
0.9401


Pouring liquid
0.7095


Riding a bike
0.9222


Throwing frisby
0.9274

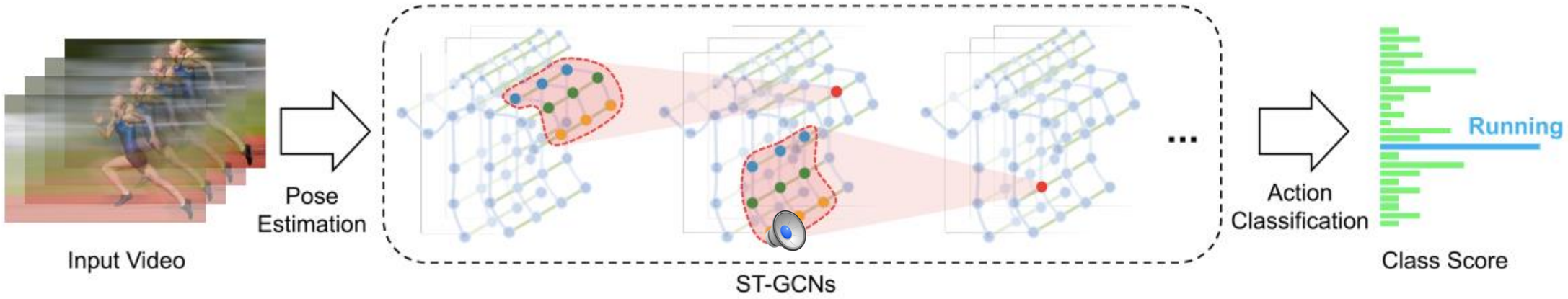

Cutting trees
0.8751


Riding a horse
0.9089


Cutting trees
0.9365

2. Spatial Temporal Graph Convolutional Network

- Pipeline Overview

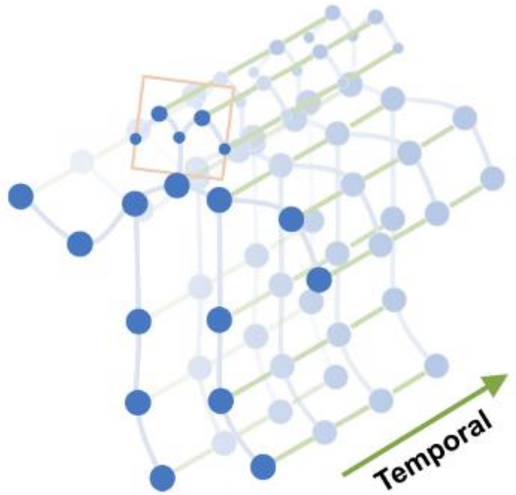


- Skeleton Graph Construction
  - Spatial temporal graph of a skeleton sequence
  - Edge of joint in one frame

$$E_S = \{v_{ti}v_{tj}|(i,j) \in H\}$$

  - Connection between joints in different frames

$$E_F = \{v_{ti}v_{(t+1)i}\}$$



2. Spatial Temporal Graph Convolutional Network

- ST-GCN
  - GCN

$$f_{out}(v_{ti}) = \sum_{v_{tj} \in B(v_{ti})} \frac{1}{Z_{ti}(v_{tj})} f_{in}(\mathbf{p}(v_{ti}, v_{tj})) \cdot \mathbf{w}(v_{ti}, v_{tj}),$$

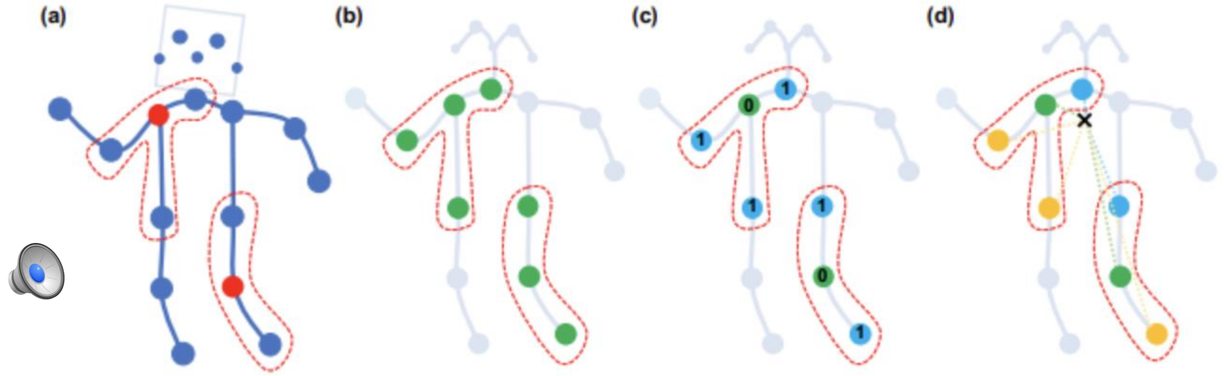
  - Sampling function
  - Weight function
  - Spatial Graph Convolution

$$f_{out}(v_{ti}) = \sum_{v_{tj} \in B(v_{ti})} \frac{1}{Z_{ti}(v_{tj})} f_{in}(v_{tj}) \cdot \mathbf{w}(l_{ti}(v_{tj}))$$

  - Spatial Temporal Modeling

$$B(v_{ti}) = \{v_{qj} | d(v_{tj}, v_{ti}) \leq K, |q - t| \leq \lfloor \Gamma/2 \rfloor\}.$$

- Partition Strategies



  - Uni-labeling partitioning
  - Distance partitioning
  - Spatial configuration partitioning

$$l_{ti}(v_tj) = \begin{cases} 0 & \text{if } r_j = r_i \\ 1 & \text{if } r_j < r_i \\ 2 & \text{if } r_j > r_i \end{cases}$$

3. Experiments & Ablation study

- Skeleton based action recognition performance on NTU-RGB+D datasets

	X-Sub	X-View
Lie Group (Veeriah, Zhuang, and Qi 2015)	50.1%	52.8%
H-RNN (Du, Wang, and Wang 2015)	59.1%	64.0%
Deep LSTM (Shahroudy et al. 2016)	60.7%	67.3%
PA-LSTM (Shahroudy et al. 2016)	62.9%	70.3%
ST-LSTM+TS (Liu et al. 2016)	69.2%	77.7%
Temporal Conv (Kim and Reiter 2017).	74.3%	83.1%
C-CNN + MTLN (Ke et al. 2017)	79.6%	84.8%
ST-GCN	**81.5%**	**88.3%**

- Mean class accuracies on the “Kinetics Motion” subset of the Kinetics dataset.

Method	RGB CNN	Flow CNN	ST-GCN
Accuracy	70.4%	72.8%	72.4%

3. Experiments & Ablation study

- Exploration using ST-GCN to capture motion information in two-stream style action recognition

	RGB TSN	Flow TSN	ST-GCN	Acc(%)
Single Model	✓			70.3
		✓		51.0
			✓	30.7
Ensemble Model	✓	✓		71.1
	✓		✓	71.2
	✓	✓	✓	71.7

4. Conclusion & Limitation

Contribution

- Proposed ST-GCN, a generic graph-based formulation for modeling dynamic skeletons, which is the first that applies graph-based neural networks for this task.
- Proposed several principles in designing convolution kernels in ST-GCN to meet the specific demands in skeleton modeling.
- On two large scale datasets for skeleton-based action recognition, the proposed model achieves superior performance as compared to previous methods.

Limitation

- The proposed method only finds the relationship between joints in the local area.

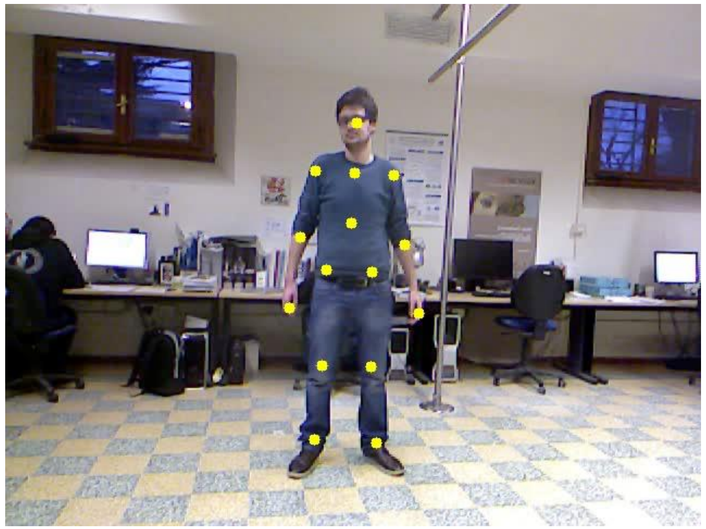
5. Project & Results

Task

- Human action classification by ST-GCN (reviewed paper)

Data

- Florence 3D action dataset
  - 9 classes : wave, drink from a bottle, answer phone,clap, tight lace, sit down, stand up, read watch, bow.
  - 215 video samples
  - There are 3d human skeleton location ground truth
  - Access : https://www.micc.unifi.it/resources/datasets/florence-3d-actions-dataset/



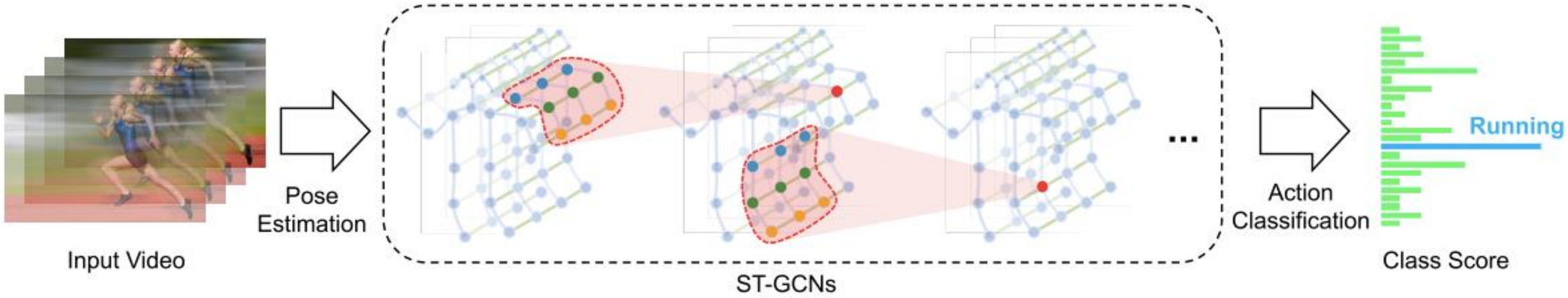
5. Project & Results

Code

- Project repository :
  - https://github.com/hongsi96/STGCN
- Dataset repository :
  - https://www.micc.unifi.it/resources/datasets/florence-3d-actions-dataset/
- Benchmark code repository :
  - https://pytorch-geometric.readthedocs.io/en/latest/
  - https://github.com/yysijie/st-gcn
  - https://github.com/yongqyu/st-gcn-pytorch
  - https://github.com/kenziyuliu/st-gcn
  - https://github.com/srijandas07/st-gcn

5. Project & Results

Model



- Input : Skeletons of video, concatenated 32 frames for one video
- Networks : ST-GCN 3 layer (kernel size = 5)
- Fully connected layer : 192 to 9
- Loss function : Cross entropy loss for classification

5. Project & Results

Results

```
[epoch 1 ] Train loss: 2.726305960388909 Train Acc: 0.14619883040935672
[epoch 2 ] Train loss: 2.789683940989232 Train Acc: 0.08187134502923976
Val loss: 4.0056541193472714 Val Acc: 0.10526315789473684
[epoch 3 ] Train loss: 2.3220801199214502 Train Acc: 0.2046783625730994
[epoch 4 ] Train loss: 1.6903783955464238 Train Acc: 0.4093567251461988
Val loss: 1.5880341843554848 Val Acc: 0.42105263157894735
[epoch 5 ] Train loss: 1.4712061687040399 Train Acc: 0.5087719298245614
[epoch 6 ] Train loss: 1.2785616190613884 Train Acc: 0.5321637426900585
Val loss: 0.8734155791370493 Val Acc: 0.631578947368421
[epoch 7 ] Train loss: 1.155952106278978 Train Acc: 0.5321637426900585
[epoch 8 ] Train loss: 1.1028945031548627 Train Acc: 0.5906432748538012
Val loss: 0.781858212461597 Val Acc: 0.7368421052631579
[epoch 9 ] Train loss: 0.9935713876583423 Train Acc: 0.6081871345029239
[epoch 10 ] Train loss: 0.8738961743942478 Train Acc: 0.6608187134502924
Val loss: 0.4096413409631503 Val Acc: 0.7894736842105263
[epoch 11 ] Train loss: 0.8359170377973402 Train Acc: 0.7192982456140351
[epoch 12 ] Train loss: 0.6896112568251658 Train Acc: 0.7719298245614035
Val loss: 0.3120671329333594 Val Acc: 0.8947368421052632
[epoch 13 ] Train loss: 0.5782215058618383 Train Acc: 0.8070175438596491
[epoch 14 ] Train loss: 0.5530537161431466 Train Acc: 0.8128654970760234
Val loss: 0.3962192135421853 Val Acc: 0.8947368421052632
[epoch 15 ] Train loss: 0.504748245162007 Train Acc: 0.8304093567251462
[epoch 16 ] Train loss: 0.3907653574902586 Train Acc: 0.8362573099415205
Val loss: 0.46070418389219986 Val Acc: 0.8947368421052632
[epoch 17 ] Train loss: 0.5696957375355914 Train Acc: 0.8362573099415205
[epoch 18 ] Train loss: 0.4297809437004446 Train Acc: 0.8830409356725146
Val loss: 0.2110244537654676 Val Acc: 1.0
[epoch 19 ] Train loss: 0.2361900110848858 Train Acc: 0.9415204678362573
[epoch 20 ] Train loss: 0.13300178904217064 Train Acc: 0.9590643274853801
Val loss: 0.13508479532442594 Val Acc: 1.0
[epoch 21 ] Train loss: 0.07759352296329382 Train Acc: 0.9766081871345029
[epoch 22 ] Train loss: 0.0501077430219286 Train Acc: 0.9824561403508771
Val loss: 0.15183017442100927 Val Acc: 1.0
[epoch 23 ] Train loss: 0.03303506839693638 Train Acc: 0.9941520467836257
[epoch 24 ] Train loss: 0.02282898357867846 Train Acc: 1.0058479532163742
Val loss: 0.13941689607638277 Val Acc: 1.0
[epoch 25 ] Train loss: 0.0168000951147916 Train Acc: 1.0058479532163742
[epoch 26 ] Train loss: 0.01223723487982973 Train Acc: 1.0058479532163742
Val loss: 0.1407097687846736 Val Acc: 1.0
[epoch 27 ] Train loss: 0.009580390190171916 Train Acc: 1.0058479532163742
[epoch 28 ] Train loss: 0.006891008401126193 Train Acc: 1.0058479532163742
Val loss: 0.1065081703035455 Val Acc: 1.0
[epoch 29 ] Train loss: 0.005069180173875644 Train Acc: 1.0058479532163742
[epoch 30 ] Train loss: 0.004016913266645538 Train Acc: 1.0058479532163742
Val loss: 0.09767571954350722 Val Acc: 1.0
Test loss: 0.15568707561628384 Test Acc: 0.9545454545454546
```

Validation accuracy : 1.0

Test accuracy : 0.9545

Code : https://github.com/hongsi96/STGCN

Run : sh run.sh