Feature pyramid graph convolutional networks for temporal action detection

Shanshan Du[a], Xiaomeng Zhang[b], Shanbang Zhu[c], Ruiwei Zhao[d], Weidong Xu[c*], Rui Feng[b,d#]
[a] School of Information Science and Technology, Fudan University, Shanghai, China; [b] School of Computer Science, Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai, China; [c] Department of Orthopedics, Changhai Hospital, Naval Medical University, Shanghai, China; [d] Academy for Engineering and Technology, Fudan University, Shanghai, China

ABSTRACT

In some long and untrimmed videos, locating the important and key segments can be a very challenging task for temporal action detection. Current methods make remarkable progress when it comes to RGB images. The aim of this paper is to develop a method with the assistance of dynamic model of human body skeletons to address this problem. To this end, we propose a Feature Pyramid Graph Convolutional Network (FP-GCN). The introduced model contains a Feature Encoding Module to encode skeleton data with graph convolutional networks, a Feature Pyramid Module to exploit the inherent pyramidal hierarchy and an Action Detection Module to generate the final prediction results of the detection. We experiment our approach on NTU RGB+D and THUMOS14 datasets and obtain a satisfactory result.

Keywords: Temporal action detection, feature pyramid, graph convolutional networks

1. INTRODUCTION

With its diverse values in various applications such as intelligent surveillance and human-computer interaction, human action detection in untrimmed videos is a fundamental which attracts lots of research attention yet remaining a very challenging computer vision problem[1, 2]. Large amount of research in action recognition and action detection have been conducted in the recent decade. Along with impressive progress in action recognition[3-5], the topic of temporal action detection has attracted considerable focuses from researchers all over the world in recent years[6].

The task of action detection is to predict temporal locations of the containing action segments or instances in the untrimmed videos, as well as to make estimation on the categories of each detected action segments or instances. Therefore, the traditional pipeline of action detection models can be mainly categorized into two consecutive major steps. The first major step is named as temporal proposal generation, which produces some candidate action locations from the long untrimmed video data. And the second major is called proposal categorization, which is to predict the semantic meaning of each generated action proposals. Most previous state-of-the-art methods rely on traditional hand-crafted visual features extracted from the video data, but their performances are always not so good to reach the bar of real applications. Recently, inspired by two-stage object detection framework, many newly proposed and reported competitive approaches[7, 8] adopt two-stage pipeline when address the temporal action detection issue. They propose temporal proposals first and then perform temporal boundary regression and classification. This kind of two-stage framework is tough to train and is not end-to-end. Besides, most of the existing action detection methods operate on raw videos. As a result, they are sensitive to the background and illumination changes in the videos. On the contrary, human body skeleton data is very robust under the changing video shooting environment. In other words, it is preferable to use the skeletal data of human bodies to address the issues and problems of action detection in untrimmed videos.

Our proposed Feature Pyramid Graph Convolutional Network, namely FP-GCN, is akin to the combination of FPN[9] and ST-GCN[10]. The former is an excellent feature pyramid network, and the latter is an extraordinary spatial-temporal skeleton modeling network. First, we establish human action spatial-temporal graphs on skeletal data from depth sensors or pose

* xuwdshanghai@126.com

fengrui@fudan.edu.cn

Third International Conference on Computer Science and Communication Technology (ICCSCT 2022)
edited by Yingfa Lu, Changbo Cheng, Proc. of SPIE Vol. 12506, 125060R
 · 0277-786X · doi: 10.1117/12.2662623

estimation algorithms, and then encode graph features by ST-GCN. Finally, we apply several lateral connections between the feature maps with different strides to obtain the high-level features and then try to estimate the boundaries of the action segments in the temporal axis based on these extracted features. Moreover, we conduct our experiments on the NTU RGB+D[11, 12] and THUMOS14[13] dataset, which are two widely used benchmarks in the field of action detection. The results show that our proposed method can achieve very competitive performance compared to the state-of-the-arts. The overall architecture of our network structure is shown in Figure 1.

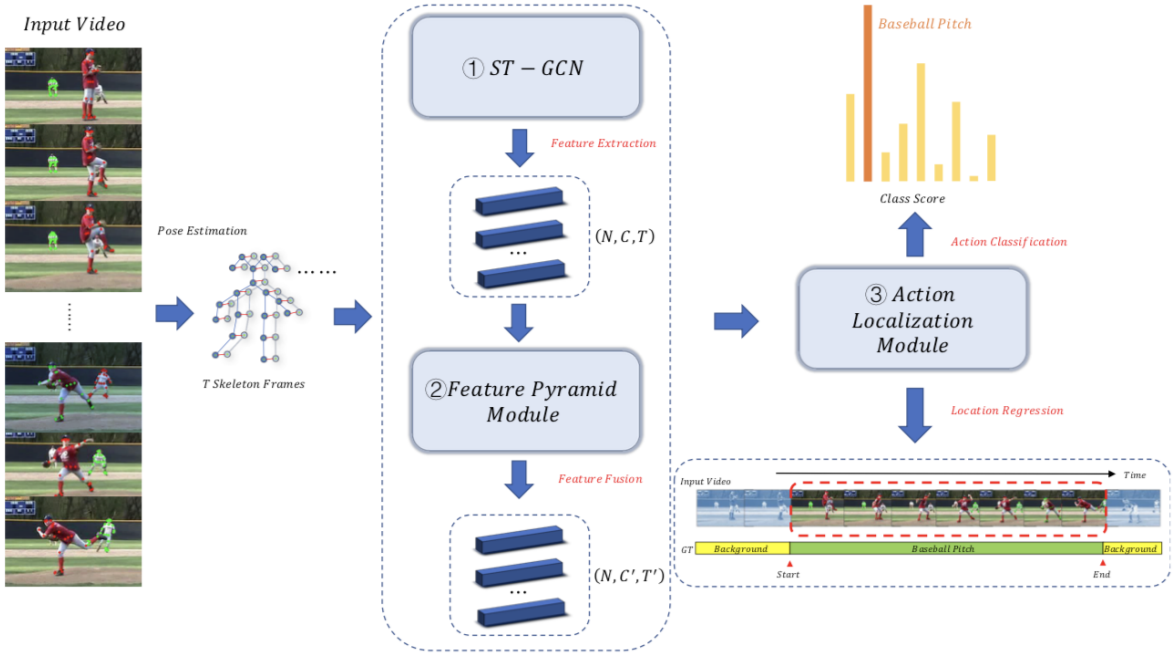

Figure 1. Overview of our FP-GCN.

FP-GCN adopts ST-GCN as the backbone to extract the high-level and representative features from the generated skeleton graph. Then we construct an in-network feature pyramid. Finally, we feed the features into the action detection module to obtain the results.

Our work applies human skeletal data in temporal action detection. Main contributions of our work are described below:

(1) We present a novel approach to detect human action time intervals directly on skeletal data instead of on RGB videos.

(2) We improve ST-GCN and combine it with a feature pyramid module to extract multi-scale features and help detect human actions in the untrimmed videos.

(3) On two popular action detection benchmarks, which are NTU RGB+D and THUMOS14, the achieved performances of our proposed method are reported. Its effectiveness and advantages are demonstrated and discussed in the experiment.

2. RELATED WORK

Graph Neural Network (GNN) is widely adopted in modeling graph-structured data such as social networks[14], knowledge graphs[15], recommendation systems[16] and physical systems[17]. A graph can be described in terms of its containing graph nodes and graph edges. Given the initial node features of a graph, we can construct neural networks to learn the feature representations of its containing vertices, or nodes, and edges. The way to construct a graph neural network typically follows two mainstreams respectively form the spatial or spectral perspectives. In the spatial domain, message passing in graph convolutional networks following the paradigm of propagating and aggregating information to update node features[18]. While in the spectral domain, the convolution operation is performed with the help of the eigenvectors plus the eigenvalues of the Laplacian matrix corresponding to the graph[19]. Our work follows the setting from the first spatial stream.

The motion of human skeletons and joints conveys quite much important information about the underlying human action. Usually, conventional skeleton-based action recognition methods rely on manually designed traversal rules which present unreasonable human factors and thus leading to unsatisfactory performance[20, 21]. Human skeleton data can be less affected by the diversity of illuminations and the variation of viewpoints. Generally speaking, there exist three deep learning methods in skeleton-based human action recognition. One is based on the RNN model. RNN is originally designed to model temporal data. Here it is used to model skeleton data as a collection of sequential joint features in vector form[22-24]. Another treats skeleton data as pseudo-images and applies the convolution operation on them[25-27]. But these two approaches are unable to model the relationship between space and time. Instead of representing skeleton data as sequences or images, the ST-GCN construct spatial-temporal graphs from the skeleton joint data and make further inference based on graph modeling results.

Object detection on images remains fundamental yet challenging for a long time. Algorithms in this direction aim at determining the location and category of existing objects in each image. SSD[28] is one of the notable multi-class one-stage detector. This very kind of detector in single shot form with multiple boxes conducts all computation in one network structure and outperforms previous representative Faster R-CNN[29] from both aspects of speed and accuracy. Moreover, object detection method like Feature Pyramid Network (FPN)[9] exploits lateral connections that using a ConvNet's pyramidal feature, which can both take spatial resolutions and semantic strength into account. The design of our FP-GCN for the purpose of creating in-network feature pyramids is similar to the architecture of FPN in exploiting the pyramidal shape of the feature hierarchy in convolutional neural networks. Besides, our work applies focal loss[30] to deal with class imbalance typically caused by limited amount of positive examples and large amount of negative or background samples.

The R (2 + 1) D network[31] are also adopted to provide complementary information encoded from the raw input videos, which is called video content encoder. Compared with conventional network with 3D convolutions, R (2 + 1) D replaces a 3D convolutional layer with a two-dimensional convolutional layer followed by a one-dimensional convolution layer. Experiments show that such kind of convolution block can decompose spatial and temporal modeling and achieve lower error rate than 3D convolution block. Furthermore, it also follows the architecture of ResNet[32]. R (2 + 1) D takes T frames of the video as input and predicts the category. The output of the last global average pooling layer is kept as the encoded feature representation of the input video data. Moreover, to encode every anchor in the video, we sample the video content in original videos from the same positions of the corresponding anchor locations regarding the interested feature maps and get the features of these video clips by R (2 + 1) D. So given K anchors, we will get the encoded feature of shape (K, D), where D is the quantity of output feature map channels from the very last global average pooling layer of the adopted network architecture. The setting of the anchors will be described in Section 3.3.

3. APPROACH

3.1 Problem formulation

Temporal action detection is to locate the starting frame and ending frame of the action instances based on the content analysis on the input untrimmed videos and predict their action categories. Without loss of generality, any untrimmed video data can be treated as a collection of sequential frames denoted as $T = \{t_n\}_{n=1}^{T_n}$, where t_n is the n-th frame data in the form of RGB images. The annotation set of T is composed of two subsets. One contains temporal boundary information and the other contains category labels. Temporal boundary information is represented by a set of temporal annotations $\Phi_g = \{\phi_g = (t_{s,n}, t_{e,n})\}_{n=1}^{N}$, where N refers to the quantity of labeled action segments or instances in video which are treated as ground-truth. $t_{s,n}$ marks the starting frame of the action segment ϕ_n, and $t_{e,n}$, on the other side, indicates the ending frame of the same action instance. Category information is denoted by a set of class labels C_g corresponding to ground-truths in ϕ_g, where $C_g = \{c_{g,n}\}_{n=1}^{N}$. During inference, temporal action detection method should generate a temporal anchor set $\Phi_p = \{\phi_p = (t_{s,n}^p, t_{e,n}^p)\}_{n=1}^{N^p}$ and a label set $C_p = \{c_{p,n}\}_{n=1}^{N_p}$, where N_p is the total amount of predicted action segments or instances in T. Temporal anchor set should cover annotation set with high temporal overlap and the predicted category of each detected action instance should also be accurate.

In Spatial-Temporal GCN (ST-GCN), inspired from the natural properties and structure of human skeletons, the skeleton graph takes joints as vertices and bones as edges, and construct the graph thereby. Specifically, we consider a skeleton with joint nodes and edges as an undirected spatial-temporal graph $G = (V, E)$, where $V = \{v_{ti} | t \in \{1,2, \dots, T\}\}_{i=1}^{N_v}$ is the collection of N_v body joints and E is the set of m bones. There are two different sorts of edges in graph G, namely spatial

edges and temporal edges. Formally, the spatial edges feature intra-body connection at each frame t and can be denoted as $E_s = \{(v_{ti}, v_{tj}) | v_{ti}, v_{tj} \in \{1,2,\dots,T\}\}$. The temporal edges E_t feature inter-frame connection, which create connection between the same joints across consecutive frames, where $E_t = \{(v_{ti}, v_{(t+1)i}) | v_{ti} \in \{1,2,\dots,T-1\}\}$.

3.2 Feature pyramid graph convolutional network

3.2.1 Feature Encoding Module. The goal of convolution in graph is to capture the dynamic skeletal data within an image at time t along the spatial dimension, which can be formulated as[10]

$$F_{out}(v_{ti}) = \sum_{v_{ti} \in N(v_{ti}) \cup \{i\}} \frac{1}{C} \left(\Theta \cdot \left(F_{in}(v_{tj}) \right) \right) \quad (1)$$

where $F(v_{ti}) = (x_{ti}, y_{ti}, z_{ti})$ is the coordinate vector of the target node. v_{tj} denote its 1-hop neighbors. Θ is the weighting function responsible for feature transformation and the summation function aggregates the normalized results. Usually, the normalized constant C is the degree of v_{ti} aiming to balance the contribution of each neighbor. Following the idea of ST-GCN, we group v_{ti} and v_{tj} into three subsets, including (1) the target node itself, (2) the centripetal nodes near the skeleton gravity coordinates compared to the target node, and (3) otherwise the centrifugal ones. Then, equation (1) is transformed into

$$F_{out} = \sum_{k=1}^{3} A_k (\Theta_k \cdot F_{in}) \quad (2)$$

where $A_i \in \{0,1\}^{n \times n}$ refers to the adjacent matrix of each sub-graph with respect to three subsets, and element $A_{p,q} = 1$ if there exists an edge between joint p and joint q and 0 otherwise. In the temporal dimension, we capture high-level temporal motion features by applying a 179 kernel of $T_{k \times 1}$ dimension to perform graph convolution on T_k consecutive frames. Given a spatial-temporal human skeleton graph as constructed above, multiple layers of ST-GCN are built, which empower information to be fused along both the temporal and spatial domain. The ST-GCN encoder transforms feature map $(N \times C_{in} \times T_{in} \times V)$ into $(N \times C_{out} \times T_{out} \times V)$,where C_{in} and C_{out} records the numbers of channels, T_{in} and T_{out} writes the length of skeleton frames, while V denotes the amounts of vertexes.

We also use R (2 + 1) D to provide complementary information encoded from the raw input videos, which is called video content encoder. Comparing conventional network with 3D convolutions, R (2 + 1) D replaces a 3-dimensional convolution layer with a 2-dimensional convolution layer followed by a 1-dimensional convolution layer. Experiments show that such kind of convolution block can decompose spatial and temporal modelling and achieve lower error rate than 3D convolution block. R (2 + 1) D takes T frames of the video as input and predicts the category. The final encoded feature of the input video is formed by the output of the last global average pooling layer. Moreover, to encode every anchor in the video, we sample the video content in original videos from the same positions of the corresponding anchor locations with respect to interested feature maps and get the features of these video clips by R (2 + 1) D. So given K anchors, we will get the encoded feature with shape (K, D), where D is the number of output channels of the last global average pooling layer of R (2 + 1) D.

3.2.2 Feature Pyramid Module. The functionality of the Feature Pyramid Module (FPM) is to exploit the inherent pyramidal hierarchy of spatial-temporal graph convolution networks. Features in deeper GCN layers have finer temporal boundary information but semantically strong while features in lower GCN layers have coarser temporal boundary information but semantically weak. The bottom-up path involves the feed forward calculation implemented by the backbone networks ST-GCN. It contains several GCN-TCN blocks. Each GCN-TCN block consists of a first graph convolution module in spatial domain. In addition, it also contains another convolution module in temporal domain. We divide the whole graph convolution layers into two parts and each part is called one network stage. Graph convolution layers at the very end of each network stage have strides of {2, 4} frames with respect to input feature vectors. We select features coming from the ending layer in every major step, because the deepest layers of each major step are expected to boast the strongest semantic features. Concretely, we choose features generated from st_gcn_6 and st_gcn_9. Given the input skeleton graph of dimension $(N \times C_{in} \times T_{in} \times V \times M)$,the output feature maps have the dimension of $(N \times C_{out} \times T_{out} \times V \times M)$.

The top-down pathway includes two lateral connections and one top-down connections. S_2 lateral layer is a (1×1) feature fusion layer, where the input data channels and output channels are both 128. Features from the top pyramid level of the network are first fed into a (1×1) S_2 convolutional layer to half its channel to 128 and then go through an upsampling layer to get the (128×200) dimensional feature vectors. After that, an element-wise summation is laid upon the up-sampled feature maps and the fused feature maps to obtain the enhanced features. Then, a top-down connection is

established thereof. In short, low-level but spatial-detailed description and high-level but semantic-rich information complement each other to build an in-network feature pyramid.

3.3 Action detection module

The setting of anchors is following faster RCNN. For the feature map with stride 2 and 4, the anchor size is 64 and 128 respectively. We use the parameterizations of 2 coordinates following

$$\begin{aligned} p_x &= (x - x_i)/w_i, \quad p_w = \log(w/w_i) \\ p_x^* &= (x^* - x_i)/w_i, \quad p_w^* = \log(w^*/w_i) \end{aligned} \tag{3}$$

where x stands for the center coordinates of the temporal box and w is its width. The x, x_i and x^* are calculated to predict the temporal box location, anchor box location and the ground-truth box location respectively.

The feature fusion block takes two kinds of features as input. One is from the graph encoder called graph features, and the other is from the video content encoder called video features. The graph features are denoted as a vector of $(N \times C_1 \times T \times V)$ dimension, where N refers to the batch size, C_1 is the output channel size, T is the output frame quantity and V is the joint counts. The video features are in a vector of $(N \times C_2 \times T)$, where C_2 is the amount of output channels. The graph features are then forwarded into an average pooling layer of kernel size $(1 \times V)$ and two convolutional layers to fuse the information of joints as shown in Figure 2. The video features are fed into a basic residual block to get an output tensor with dimension $(N \times C_1 \times T)$. Finally, the obtained graph features and the visual video features are combined together. The combination is implemented by the concatenation operation along the channel dimension.

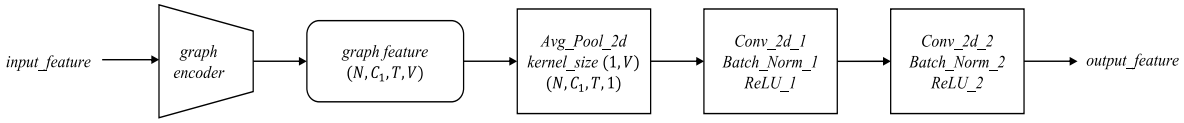

Figure 2. Average pooling and feature fusion of spatial-temporal graph.

The action recognition head is built with a kernel size 1 convolutional layer and the output channel size is $(K \times C)$, where K stands for the quantity of anchors and C is the action categories quantity including background. In this way, every value in each output channel of this head represents the classification results of each anchor.

The action recognition head is built with a convolutional layer with kernel dimension 1. Its output channel size is $(2 \times K)$, where $(2 \times K)$ means the predicted p_x and p_w of K anchors.

4. EXPERIMENT

4.1 Datasets and preprocess

NTU RGB+D 60 dataset is a widely used large-scale human action recognition benchmark collected by three cameras. There are totally 60 action classes in the dataset and it contains 56880 video clips gathered from 40 distinct subjects. Two kinds of standard evaluation benchmarks are provided, *i.e.*, cross-view benchmark x-view and cross-subject benchmark x-sub, which means different setting of camera and different people respectively. The NTU RGB+D 120 dataset is in fact an expanded version of the preliminary 60-classes dataset with wider range of performer ages, richer categories of action classes and finer action granularity. The NTU RGB+D 120 dataset involves a larger number of 114,480 video samples.

THUMOS14 is widely used in temporal action detection task which contains 20 action classes. The validation set and the testing set are both made up of temporal annotated untrimmed video which has 200 and 213 video samples respectively. In our experiments, we choose the processed training set and the original validation to train our model parameters and choose the testing set to evaluate our model performance.

We reconstruct the above two skeleton datasets to make it suitable for our temporal action detection task. In order to simulate the data characteristics of untrimmed videos, we concatenate together every two of short video whose frames is shorter than 150 to get a new long video and keep the other videos.

We utilize OpenPose[33] to estimate the 25 key joint coordinates (X, Y) of human skeletons on the image and normalize them according to size of the image and then combine them with the confidence score. According to the distribution of video length, we choose 400 as the target frame length of our input skeleton data. For the trimmed videos whose frame

larger than 400, we sample the other videos with an appropriate interval to restrict its frame length within the range of 400. For the other videos, we try clipping them with a window size of 400 which is designed to contain action instances as many as possible.

4.2 Training details

We use Focal Loss[30] in our classification task. It can be defined as:

$$L_{cls}(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{4}$$

Here α_t is the coefficient of target class and it is set to 0.25 for background and 0.75 for other classes. p_t is the calculated possibility by the model of the actual class, and γ is a hyperparameter that is set to 2. Focal loss can reduce the impact of large numbers of easy background samples and focus the model's attention more to the hard positive samples. The boundary regression loss is Smooth L1 Loss, which aims to gauge the gap between the position of the ground truths and the predicted anchors. The total loss is the weighted addition of the classification loss and the boundary regression loss. We define our loss as

$$L = L_{cls}(p_t) + \alpha L_{trg} \tag{5}$$

where α in this equation is the hyperparameter to balance the loss of classification and boundary regression and is set to 0.2 in our experiment.

We build the backbone of FP-GCN with 10 ST-GCN blocks, in which both the bottom-up pathway and the top-down pathway are connected by two lateral connections. The input channels in these 10 ST-GCN blocks are 64, 128, and 256 respectively. We train the model for a total amount of 150 epochs. The batch size is set to 128. We set the initial learning rate to 0.05. The training process warms up from 0 in the first 5 epochs and then scheduled by CosineAnnealingLR[34]. The optimizer is SGD-GC[35] and the momentum value is set to 0.9. Its weight decay value is set to 5E-4. We define the samples whose threshold of Intersection over Union (tIoUs) with the actual box locations are larger than 0.7 as positive samples and those whose tIOUs with ground truth are smaller than 0.3 as negative samples. The NMS threshold is 0.4.

4.3 Results and analysis

The results of FP-GCN on each dataset are displayed in Table 1 shows with details. The header line in the table indicates different level of threshold of tIoU). Thanks to the introduction of ST-GCN, our model can extract discriminative spatial-temporal features by analysing the massive skeleton data. In addition, the the lateral connections and top-down pathways for feature enrichment also play an important role during inference. It can be observed that our FP-GCN can successfully reach state-of-the-art performance and be very competitive on the evaluated NTU RGB+D datasets.

Table 1. Results of action detection on various datasets in terms of mAP@tIoU.

Dataset Name	0.3	0.4	0.5	0.6	0.7
NTU RGB+D 60 x-sub	67.53	67.51	67.33	66.65	63.86
NTU RGB+D 60 x-view	73.72	73.71	73.62	73.11	71.16
NTU RGB+D 120 x-sub	43.95	43.91	43.77	43.22	41.35
NTU RGB+D 120 x-view	64.75	64.73	64.67	64.24	62.73
THUMOS14	33.59	27.85	20.34	11.29	4.60

The reason why the mAP is relatively low on THUMOS 14 dataset is that for non-neglectable quantity of video samples, the skeleton results produced by OpenPose algorithm is not reliable. For example, for videos contain complex background contents, OpenPose pays its attention to the background contents rather than the action subjects, as shown in Figure 3.

On the contrary, for quite a few action categories, the performance of FP-GCN is excellent, as shown in Figures 4 and 5.

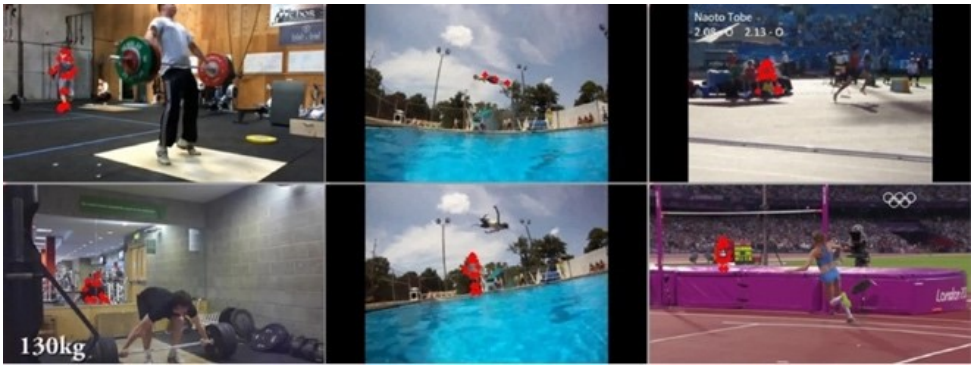

Figure 3. Examples of wrong skeleton estimation by OpenPose.

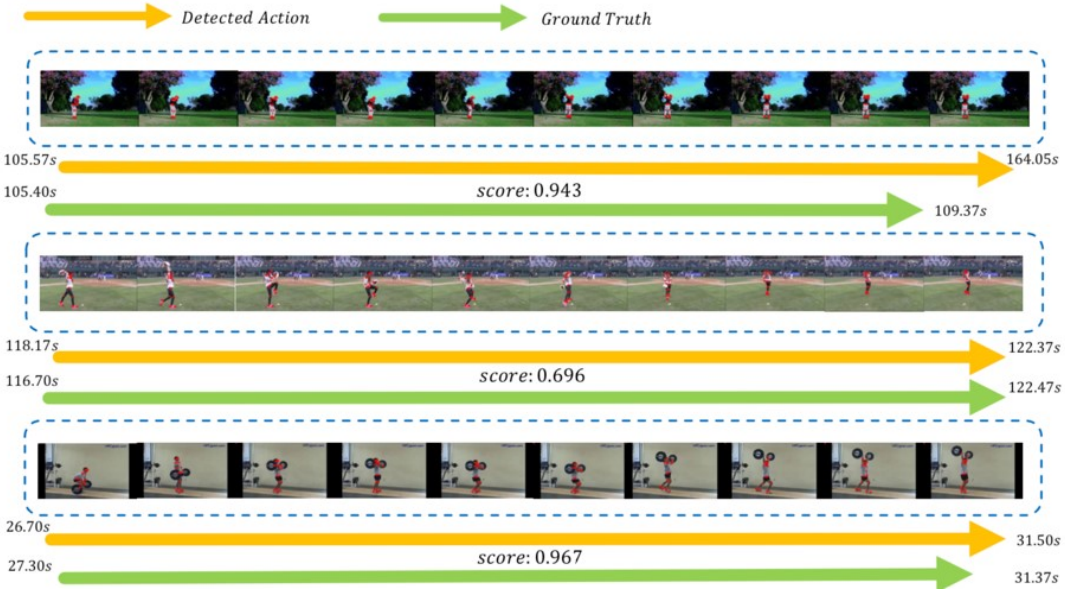

Figure 4. Qualitative examples of action instances detected ed by FP-GCN on THUMOS14.

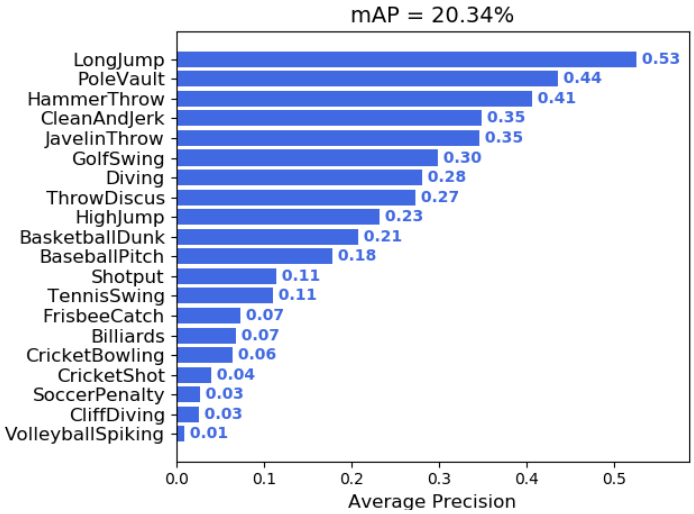

Figure 5. Statistics of action instances detected by FP-GCN on THUMOS14.

5. CONCLUSION

In this work, we put forward a new solution for temporal action detection. We name our proposed algorithm as feature pyramid graph convolutional networks (FP-GCN). We also introduce skeleton modality data into the temporal action detection task. In addition, our study suggests the effectiveness of build-in feature pyramids in graph convolutional networks, which can enhance the features to better fit in the subsequent action detection and action classification tasks.

ACKNOWLEDGEMENTS

This work was supported (in part) by National Natural Science Foundation of China (No. 62172101), Science and Technology Commission of Shanghai Municipality (No. 21511100500, No. 20DZ1100205), and Science and Technology Major Project of Commission of Science and Technology of Shanghai (No. 2021SHZDZX0103).

REFERENCES

[1] Duan, X., Huang, W., et al., "Weakly supervised dense event captioning in videos," Advances in Neural Information Processing Systems (NeurIPS), (2018).
[2] Gan, C., Wang, N., et al., "DevNet: A deep event network for multimedia event detection and evidence recounting," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), (2015).
[3] Shi, L., Zhang, Y., et al., "Skeleton-based action recognition with directed graph neural networks," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), (2019).
[4] Shi, L., Zhang, Y., et al., "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), (2019).
[5] Zhang, P., Lan, C., et al., "View adaptive neural networks for high performance skeleton-based human action recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(8), 1963-1978(2019).
[6] Zeng, R., Huang, W., et al., "Graph convolutional networks for temporal action localization," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), (2019).
[7] Gao, J., Yang, Z. and Nevatia, R., "Cascaded boundary regression for temporal action detection," arXiv preprint arXiv:1705.01180, (2017).
[8] Xu, H., Das, A. and Saenko, K., "R-C3D: Region convolutional 3D network for temporal activity detection," IEEE Inter. Conf. on Computer Vision (ICCV), (2017).
[9] Lin, T. Y., Dollár, P., et al., "Feature pyramid networks for object detection," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), (2017).
[10] Yan, S., Xiong, Y. and Lin, D., "Spatial temporal graph convolutional networks for skeleton-based action recognition," AAAI Conf. on Artificial Intelligence (AAAI), (2018).
[11] Liu, J., Shahroudy, A., et al. 2019 NTU RGB+D 120: A large-scale benchmark for 3D human activity understanding IEEE Transactions on Pattern Analysis and Machine Intelligence 42 10 2684-2701.
[12] Shahroudy, A., Liu, J., et al., "NTU RGB+D: A large scale dataset for 3D human activity analysis," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), (2016).
[13] Jiang, Y. G., Liu, J., et al., "THUMOS challenge: Action recognition with a large number of classes," (2014).
[14] Hamilton, W., Ying, Z. and Leskovec, J., "Inductive representation learning on large graphs," Advances in Neural Information Processing Systems (NeurIPS), (2017).
[15] Wang, Z., Zhang, et al., "Knowledge graph embedding by translating on hyperplanes," AAAI Conf. on Artificial Intelligence (AAAI), (2014).
[16] Ying, R., He, R., et al., "Graph convolutional neural networks for web-scale recommender systems," ACM SIGKDD Inter. Conf. on Knowledge Discovery & Data Mining (SIGKDD), (2018).
[17] Battaglia, P., Pascanu, R., et al., "Interaction networks for learning about objects, relations and physics," Advances in Neural Information Processing Systems (NeurIPS), (2016).
[18] Gilmer, J., Schoenholz, S. S., et al., "Neural message passing for quantum chemistry," Inter. Conf. on Machine Learning (ICML), (2017).
[19] Kipf, T. N. and Welling, M., "Semi-supervised classification with graph convolutional networks," arXiv preprint arXiv:1609.02907, (2016).
[20] Fernando, B., Gavves, E., et al., "Modeling video evolution for action recognition," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2015).
[21] Vemulapalli, R., Arrate, F. and Chellappa, R., "Human action recognition by representing 3D skeletons as points in a lie group," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), (2014).

[22] Du, Y., Wang, W. and Wang, L., "Hierarchical recurrent neural network for skeleton based action recognition," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), (2015).
[23] Liu, J., Shahroudy, A., et al., "Spatio-temporal LSTM with trust gates for 3D human action recognition," European Conf. on Computer Vision (ECCV), (2016).
[24] Song, S., Lan, C., et al., "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," AAAI Conference on Artificial Intelligence (AAAI), AAAI Press, (2017).
[25] Ke, Q., Bennamoun, M., et al., "A new representation of skeleton sequences for 3D action recognition," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017).
[26] Kim, T. S. and Reiter, A., "Interpretable 3D human action analysis with temporal convolutional networks," IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), (2017).
[27] Li, C., Zhong, Q., et al., "Skeleton-based action recognition with convolutional neural networks," IEEE Inter. Conf. on Multimedia & Expo Workshops (ICMEW), (2017).
[28] Liu, W., Anguelov, D., et al., "SSD: single shot multibox detector," European Conf. on Computer Vision (ECCV), (2016).
[29] Ren, S., He, K., et al., "Faster R-CNN: Towards realtime object detection with region proposal networks," Advances in Neural Information Processing Systems (NeurIPS), Curran Associates Inc., (2015).
[30] Lin, T. Y., Goyal, P., et al., "Focal loss for dense object detection," IEEE Inter. Conf. on Computer Vision (ICCV), (2017).
[31] Tran, D., Wang, H., et al., "A closer look at spatiotemporal convolutions for action recognition," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), (2018).
[32] He, K., Zhang, X., et al., "Deep residual learning for image recognition," Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), (2016).
[33] Cao, Z., Martinez, G. H., et al., "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(1), 172-186(2019).
[34] Loshchilov, I. and Hutter, F., "SGDR: stochastic gradient descent with warm restarts," arXiv preprint arXiv:1608.03983, (2016).
[35] Yong, H., Huang, J., et al., "Gradient centralization: A new optimization technique for deep neural networks," arXiv preprint arXiv:2004.01461, (2020).