Video ViG for Action Understanding

Ming Li A0243411W
Jinbin Bai A0268268M
Haoqiang Zhu A0254392B

Abstract

In this report, we propose a Video GNN (VideoG) framework, inspired by ViG, for action recognition. Firstly, we construct a baseline method modified from ViG by regarding a video as an image (frame) sequence and dividing each frame into a series of patch nodes. Secondly, we propose some novel strategies of modelling temporal information among frames in a video, significantly improving the baseline method. Finally, we select a popular action recognition benchmark and conduct experiments on it, demonstrating the feasibility of our baseline method and the effectiveness of our proposals. Codes are released at https://github.com/JB-Bai/VideoG.

1. Contributions of Each Team Member

- **Ming Li**: Investigating action recognition literature, constructing the baseline, including video data preparation, data preprocessing pipeline, the baseline model, and the performance evaluation, and writing the report. The corresponding sections include Section 2, 4, 7.1, 7.2, 7.3, and 7.4.
- **Jinbin Bai**: Improving the baseline, adding some tricks to achieve better performance. Organizing the baseline code to make it readable and make it public. Writing the report which includes Section 5, 7.5 and 8.
- **Haoqiang Zhu**: Propose possible improvements, implement the correspond code and do the experiments. Writing the report:3, 6, 7.5.

2. Introduction

Action recognition plays a critical role in many fundamental applications, *e.g.,* smart city, domestic robotics, and autonomous driving. It has seen great progress in recent years [3]. However, modelling and exploiting interactions of different parts of a single object or between different objects are still not sufficiently underscored for action recognition [8]. Graph neural networks (GNNs) have innate advantages in modelling relationships of nodes in a graph. Moreover, Vision GNN (ViG) [2] has demonstrated that it is feasible to divide an image into a set of patch nodes and model their relationships for object classification. Naturally, we consider it a valuable direction to apply GNNs to learn inner- or inter-object interactions in videos for action recognition.

In this project, we propose a Video GNN (VideoG) framework, inspired by ViG, for action recognition. Firstly, we construct a baseline method modified from ViG by regarding a video as an image (frame) sequence and dividing each frame into a series of patch nodes. Secondly, we propose a novel strategy of modelling temporal information among frames in a video, significantly improving the baseline method. Finally, we select a popular action recognition benchmark and conduct extensive experiments on it, demonstrating the feasibility of our baseline method and the effectiveness of our proposals.

3. Vision GNN for Image Classification

The method is proposed in the paper "Vision GNN: An Image is Worth a Graph of Nodes". It is designed for efficiently processing and analyzing image data. The authors argue that representing images as graph structures can better capture local structural information in images, thus improving the performance of computer vision tasks. By representing pixels or regions as nodes and spatial relationships in feature space as edges, it can represent images as graphs. This allows graph neural networks to be applied directly to image processing tasks.

The ViG block proposed in the paper is the basic building unit in ViG network for extracting feature representations of images. It consists of Grapher module and FFN module. The Grapher module uses graph convolution operation and linear layers to exchange information between nodes and increase feature diversity. To further enhance feature transformation capacity and alleviate over-smoothing phenomenon in GCN, the FFN module is introduced to perform feed-forward network operation on each node. The max-relative graph convolution used in ViG block reduces computation and improves efficiency.

4. Baseline for Action Recognition

Based on the ViG framework, we regard a video as an image sequence and apply 3D convolutions to extract image patches from it. In this work, we mainly explore designing GNN mechanisms to model temporal correlations of nodes for action recognition. Therefore, we do not downsample the temporal dimension of a video when extracting nodes using 3D convolutions.

5. Modelling Temporal Information

To process video data, we require an input dimension of [batch size, frame number, channel, height, weight]. In order to convert the video into a visual word embedding, we can use a sequence of nn.Conv3d operations, as described in the publication by Wang et al. [9]. The resulting embedding input dimension, which will be fed to the GCN, which should be [batch size, filter number, temporal number, height number, weight number]. The filter number represents the dimension of each node embedding.

Our baseline method distributes the temporal number of nodes evenly to the height and weight numbers. However, this approach may result in the loss of temporal information. To address this issue, we choose to preserve the 3D patch structure prior to building the graph. For implementation purposes, we treat each of the three dimensions independently and add independent positional embedding.

6. Our Improvements

Method 1: Whole Graph Construction

In this method, the time, height, and width dimensions of the input data are merged, resulting in the construction of a graph where data from all time steps are included in the same graph. This approach considers both time and spatial dimensions, but they are processed within a single adjacency matrix. In other words, this method fuses the spatial information of the entire time sequence into one adjacency matrix.

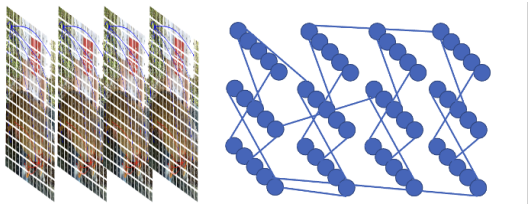

Figure 1. Whole Graph Construction

Method 2: Separate Time Step Graph Construction

In this method, the batch and time dimensions of the input data are merged, meaning that when constructing the graph, each time step has its own independent graph. This processing approach allows for handling data along the time dimension, where each time step has its own adjacency matrix. This method considers spatial information separately for each time step when dealing with data that has a time dimension (such as video).

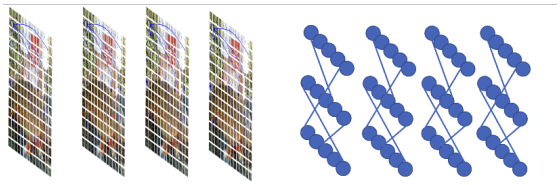

Figure 2. Separate Time Step Graph Construction

7. Experiments

7.1. Dataset

HMDB51 [4] is one of the most popular action recognition benchmarks, which contains 6,849 videos and 51 human actions. These videos are collected from movies, public databases, and the YouTube website. The action categories can be grouped in five types, *i.e.,* general facial actions, facial actions with object manipulation, general body movements, body movements with object interaction, and body movements for human interaction. Each action category contains at least 101 clips.

7.2. Evaluation Metric

The performance of action recognition is evaluated by the widely used top-1 accuracy, where the class of a video is determined by the averaged prediction of 5 clips $\times$ 3 crops [7]. 5 clips are sampled from original videos along the temporal dimension with an equal interval. Similarly, 3 crops are obtained from original videos along the height dimension with an equal interval. The best values are marked in bold.

7.3. Data Augmentation

During training, random cropping, horizontally flipping and color jitter are performed as data augmentation strategies. Besides, the AutoAugment policy with rand-m7-n4-mstd0.5-inc1 [1] is applied to augment videos. None of them is adopted to process testing images.

7.4. Implementation Details

Our VideoG is implemented using PyTorch and based on ViG. The learning objective is smoothed cross-entropy loss [6] derived from action labels. The optimization is carried out using AdamW [5] with mixed precision and a weight decay of 0.05. The frame sampling rate is set at 2. The video size and node patch size are set at $16 \times 224 \times 224$ and $1 \times 16 \times 16$, respectively. Therefore, the total number of nodes derived from a video is $16 \times 14 \times 14$. The number of training epochs is 120. The learning rate is 2e-3 and the warm-up strategy is applied in the first 10 epochs. The experiments are conducted on two NVIDIA TESLA V100 GPUs. For more details, please refer to the attached code.

7.5. Result Comparisons

We report the Top-1 accuracy of action recognition on HMDB51 with different methods.

As can be seen from the table, our methods achieve better results than the baseline with less computation.

Method	Params(M)	FLOPs(B)	Top-1
ViG-Ti (baseline)	7.8	47.6	33.922
M2,T=4	34.9	9.8	36.286
M2,T=16	63	27	38.929
M1,T=4,k=36	34.9	10.8	30.221

Table 1. Accuracy at Epoch 120 for different methods on HMDB51. M1 means Method 1 and M2 is method 2. k is the number of nearest nodes, default 9. T is the time steps we keep before consturcting the graph

8. Conclusion

In this project, we dive deep into Vision GNN, which is designed for efficiently processing and analyzing image data with graph neural networks. Inspired by ViG, we propose a Video GNN (VideoG) for action recognition. Specifically, we first construct a baseline method and then propose a novel strategy of modelling temporal information among frames in a video, finally, we select a popolar action recognition benchmark and conduct experiments on it, demonstrating the feasibility of our baseline method and effectiveness of our proposals.

References

[1] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 113–123, 2019. 2

[2] Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. Vision gnn: An image is worth graph of nodes. In *Advances in Neural Information Processing Systems*. 1

[3] Yu Kong and Yun Fu. Human action recognition and prediction: A survey. *International Journal of Computer Vision*, 130(5):1366–1401, 2022. 1

[4] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pages 2556–2563. IEEE, 2011. 2

[5] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[6] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 2

[7] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *arXiv preprint arXiv:2203.12602*, 2022. 2

[8] Tiancai Wang, Tong Yang, Martin Danelljan, Fahad Shahbaz Khan, Xiangyu Zhang, and Jian Sun. Learning human-object interaction detection using interaction points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4116–4125, 2020. 1

[9] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 1