EFFICIENT TEMPORAL-SPATIAL FEATURE GROUPING FOR VIDEO ACTION RECOGNITION

Zhikang Qiu[⋆], *Xu Zhao*[⋆], *Zhilan Hu*[†]

[⋆]Department of Automation, Shanghai Jiao Tong Univeristy
[†]The Central Media Technology Institute of Huawei Co., Ltd.

ABSTRACT

Temporal information plays an important role in action recognition. Recently, 3D CNN is widely used in extracting temporal features from videos. Compared to 2D CNN, 3D CNN has more parameters and brings heavy computation burden. It is necessary to improve the efficiency of action recognition. In this paper, inspired by group convolution and convolution kernel decomposition, we propose a novel module called grouped decomposed module (GDM) which separates channels into three groups and applies 3D, 2D and 1D convolution in parallel respectively. This module extracts spatial and temporal features efficiently. Based on GDM, we design a new network named grouped decomposed network (GDN). The grouped decomposed network achieves state-of-the-art performance on two temporal-related datasets (Something-Something V1 & V2) but requires few parameters and FLOPs.

Index Terms— Action recognition, group convolution, kernel decomposition

1. INTRODUCTION

Action recognition is a fundamental task in video analysis. Recently, intensive attention has been paid on action recognition for its wide applications on video understanding. Video is a type of media containing rich information on both spatial pattern and temporal relationship. Temporal relation is a crucial cue when recognizing actions from videos. For example, some actions change with reversed temporal order, as is shown in Figure 1. In the past several years, CNN has shown its great power on image-based tasks which focus on static pattern. Nowadays, there are many works that extract spatio-temporal information from videos based on convolution neural networks, but how to efficiently extract spatio-temporal information from videos remains a hard problem.

With the release of many large scale datasets, many works based on 2D [2, 3, 4] and 3D CNN [5, 6, 7, 8] have been proposed to improve the performance of action recognition.

This work has been supported in part by the funding from NSFC (61673269, 61273285), Huawei cooperation project, and the project funding of the Institute of Medical Robotics at Shanghai Jiao Tong University.
(Corresponding author: Xu Zhao. E-mail: zhaoxu@sjtu.edu.cn)

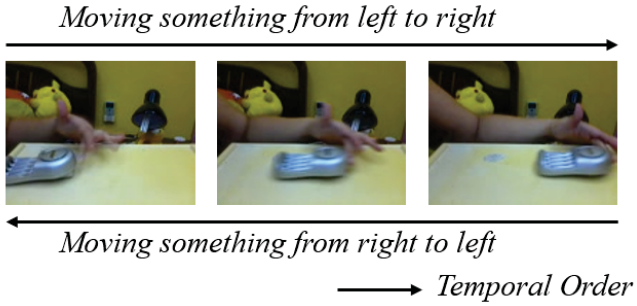

Fig. 1. A video whose actual label is *Moving something from left to right*, sampled from Something-Something [1] dataset, will be classified to a total different label as *Moving something from right to left* with reversed temporal frame order.

To extract spatial-temporal information, Karen et. al [9] propose a two-stream method that one branch takes RGB frames as input and another takes optical flow [10] frames as input. However, the application of optical flow is limited by its time-consuming calculation. Besides, 3D CNN is also used to extract spatial-temporal features from video clips. I3D [5] inflates filter kernels from 2D to 3D and initializes those kernels using ImageNet [11] pretrained weights. Compared with 2D CNN, the number of parameters and computation cost of 3D CNN are increased exponentially, which is harmful to network optimization. R2+1D [12] and P3D [13] decompose the spatial-temporal convolution into cascaded spatial convolution and temporal convolution so as to reduce the cost of 3D CNN. S3D [14] only inflates deep layers of 2D network so as to seek a balance between accuracy and speed. Compared with 2D CNN based methods, 3D CNN based methods still suffer from heavy computation burden. To solve this problem, Lin et. al [3] propose a temporal shift module (TSM) that shifts parts of the feature channels along the temporal dimension without additional parameters and computation. Recently, inspired by group convolution [15], Luo et. al [16] propose an efficient method called GST which separates channels into spatial-temporal group and spatial group where 3D CNN and 2D CNN are applied on each group.

To improve the efficiency of action recognition models, the number of parameters and FLOPs of the model should

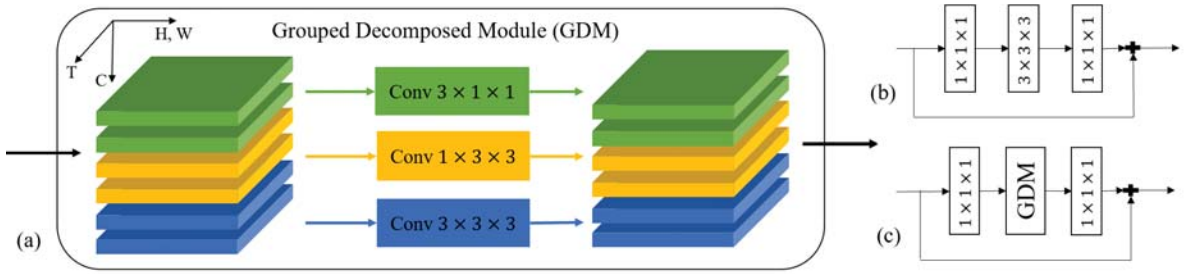

Fig. 2. (a) **Grouped Decomposed Module (GDM)** performs spatial and temporal modeling at the same layer. The input and output have the same shape. (b) The bottleneck block of 3D ResNet. (c) **Grouped Decomposed Network (GDN)** that replaces the $3 \times 3 \times 3$ convolution layers of bottleneck blocks by GDM.

be reduced while the performance of the model should be improved. Group convolution is widely used to reduce the model size and computation. Convolution kernel decomposition is another method for model slimming. Inspired by this, we propose a novel and much more efficient method named GDN. We divide the features into three groups along channel dimension and design a novel grouped convolution decomposition module (GDM) that processes the grouped features separately. 3D convolution is used in the first group to extract spatial-temporal information, 2D convolution is then concatenated in the second group for spatial information extraction and 1D convolution is applied in the last group for temporal relation modeling. Compared with other similar methods like P3D [13] and GST [16], we adopt a novel architecture which has three parallel grouped branches. What's more, we found that the spatial resolution is less important on temporal-related datasets. We reduce the height and width of input clips to half so that the computation burden is reduced to a quarter while the accuracy is almost unchanged.

To summarize, our contribution include two parts:

(a) We propose a novel and efficient method that extracts spatial-temporal, spatial and temporal features separately. Our method achieves state-of-the-art performance on Something-Something [1] datasets;

(b) We demonstrate that spatial resolution is not very important in temporal-sensitive action recognition.

2. METHOD

2.1. Decomposition of 3D Convolution Kernel

Videos can be seen as stacked images along time dimension. As a result, it's a natural way to expand the spatial convolution to spatio-temporal convolution. Assuming that T, H, W are the temporal and spatial dimension of convolution kernel size separately, the size of a 3D convolution kernel with C_{in} input channels and C_{out} output channels will be $C_{out} \times C_{in} \times T \times H \times W$, which is T times greater than its 2D counterpart. The increased kernel size brings much difficulties to model training.

In order to reduce the parameters, Qiu et. al [13] decompose the 3D convolution into cascaded spatial and temporal parts. They think that the spatial and temporal kernels are orthogonal to each other. The 3D convolution could be expressed as

$$x_o = ST\left(x_i\right) \tag{1}$$

where x_i and x_o denote the input and output features and ST is the spatio-temporal convolution. Then the decoupled 3D convolution could be formed as

$$x_o = S\left(T\left(x_i\right)\right) \tag{2}$$

where S is the spatial convolution and T is the temporal convolution. Under the decomposition, the number of nonlinear units is doubled so that the representation ability of the model is enhanced and experiments show that the model performs better after decomposition.

2.2. Grouped Convolution Decomposition Module

Group convolution is firstly proposed in AlexNet [17] so that a deep neural network can be trained on less powerful GPUs with limited memory available at that time. After that, many works adopted this idea to reduce the size of models such as ResNeXt [18] and CSN [19]. It is worth noting that the structures of these groups are all the same and the features extracted by different groups have no preference on space and time. We design a grouped convolution decomposition module that uses different convolution kernels in different groups so that different groups can model different information separately.

As is shown in Figure 2 (a), the input features are divided into three groups. The first group is spatio-temporal group (blue part), 3D convolution with kernel size $3 \times 3 \times 3$ is applied on this group to extract spatio-temporal features. The second group is spatial group (yellow part) where 2D convolution with kernel size $1 \times 3 \times 3$ is applied to extract spatial features. The third group is temporal group (green part) for temporal feature extraction and the kernel size of temporal convolution is $3 \times 1 \times 1$. After that, the three output feature maps are concatenated together. Formally, the grouped convolution can be written as

Table 1. Comparison of the number of parameters between different blocks

Model	# parameters
C2D	$9 \times C_{out} \times C_{in}$
C3D	$27 \times C_{out} \times C_{in}$
P3D	$12 \times C_{out} \times C_{in}$
GST	$9 \times C_{out} \times C_{in}$
GDM	$33/8 \times C_{out} \times C_{in}$

$$x_o = [x_{o_{ST}}, x_{o_S}, x_{o_T}] = [ST(x_{i_{ST}}), S(x_{i_S}), T(x_{i_T})]$$

where $[\,]$ denotes the concatenation operation. Compared with I3D [5] and S3D [14], our module can model features with preference and avoid heavy computational burden. Later we will show that our model performs better than I3D [5].

The most significant effect of group convolution with N same groups is that the number of parameters is reduced to $1/N$ of the original. However, if we separate the channels evenly, the number of parameters of GDM will be greater than 2D CNN with kernel size 3×3. In order to reduce the parameters, we introduce two hyper-parameters to control the number of parameters of GDM. We use r_{st} and r_s to specify the channel proportion of spatio-temporal and spatial group, thus the proportion of temporal group is $1 - r_{st} - r_s$. The number of input channels and output channels of each group are same.

It is difficult to specify the values of r_{st} and r_s through theoretical analysis and searching the optimal values in unit space also requires lots of experiments. Fortunately, a lot of effective exploration has been done in GST [16]. According to GST [16], we set $r_{st} = 1/4$ and $r_s = 1/2$ empirically, it means that half channels are used in spatial group, a quarter channels are used in spatio-temporal group and other channels for temporal group.

With the kernel size $H = W = T = 3$, we list the parameter numbers of several different spatio-temporal architectures for comparison in Table 1. Obviously, GDM has the least parameters than almost any other previous work.

2.3. Network Architecture

We replace all the $3 \times 3 \times 3$ layers of 3D ResNet-50 [20] with GDM while keeping other layers unchanged, as is shown in Figure 2 (c). Similar to TSN [2], we fuse the prediction scores of all frames on average to get the final prediction.

Indeed, our model could be seen as a general representation of many different networks such as R3D [20], R2D [21] and S3D [14]. If the r_{st} of all layers equal to 1, then the model is R3D [20]. If the r_s of all layers equal to 1, then the model is R2D [21]. If the r_s of low layers equal to 1 and r_{st} of deep layers equal to 1, then the model is reduced as S3D [14].

3. EXPERIMENTS

3.1. Dataset

Our method is evaluated on two large scale temporal-related datasets, Something-Something V1 [1] and V2 [22], that require strong temporal modeling ability. It is difficult to recognize the action based on scene or typical objects shown in the video. Instead, more attention should be paid to temporal relation of the video. The dataset contains more than 100k (V1) and 220k (V2) videos across 174 classes, with duration ranging from 2 to 6 seconds.

3.2. Implementation Details

We choose 3D ResNet-50 as our backbone. We initialize the parameters of spatial group using ImageNet [11] pretrained model. The parameters of temporal group are randomly initialized. We use the same inflated method in I3D [5] to initialize the parameters of spatio-temporal group. We train the model on a GPU server. The method is implemented in PyTorch framework.

In the training stage, we use the segmental sampling strategy proposed in TSN [2], the size of the short side of these frames is fixed to 256 and 128 and then randomly cropped to 224×224 and 112×112 respectively. We use randomly scale jittering and horizontal flipping for data augmentation. As for optimization, We use a mini-batch SGD optimizer with an initial learning rate of 0.02. The mini-batch size is 48, the total training epoch is about 40 and the learning rate decayed by a factor of 10 in epoch 20 and 30.

In the inference stage, we use the same sampling method as in the training stage but choose the central frame of every segment and do central crop for each frame. We don't use any multiple clips or crops fusion strategies to boost the performance. All results are reported on single clip and crop.

3.3. Experiment Results and Analysis

In this section, we compare our method with other popular methods. It's computational expensive to boost the accuracy by increasing the input frames. As a result, we fix the number of input frames to 8 and 16. As is shown in Table 2 and Figure 3, our method obtains the best accuracy among other methods. Compared with TSM [3], our GDN network gains 0.3% and 3.1% on 8 and 16 frames inputs respectively with few parameters and FLOPs. One thing needs to be noted is that the TSN [2] with single RGB stream input performs worst on Something-Something [1] datasets because it has no ability of temporal modeling. Something-Something V2 is derived from V1 by reducing noise and collecting more data. Results on Something-Something V2 dataset are listed on Table 4. Our method also gets the state-of-the-art result.

Our GDN model achieves the very competitive performance with great efficiency and low computation cost for fast

Table 2. Compared with other methods on the validation split of Something-Something V1 dataset. We only take RGB frames as input for it is the only practical data so far

Method	Backbone	#Frame	#Params	GFLOPs	Top1	Top5
TSN [2]	ResNet50	8	24.3M	33	19.7	46.6
ECO [23]	BNInception+ 3D ResNet-18	8	47.5M	32	39.6	-
ECO [23]	BNInception+ 3D ResNet-18	16	47.5M	64	41.4	-
ECO_{En}Lite [23]	BNInception+ 3D ResNet-18	92	150M	267	46.4	-
I3D [24]	3D ResNet50	$32\times2clips$	28.0M	153×2	41.6	72.2
NL I3D [24]	3D ResNet50	$32\times2clips$	35.3M	168×2	44.4	76.0
NL I3D+GCN [25]	3D ResNet50	$32\times2clips$	62.2M	-	46.1	76.8
TSM [3]	ResNet50	8	24.3M	33	43.4	73.2
TSM [3]	ResNet50	16	24.3M	65	44.8	74.5
TSM_{En} [3]	ResNet50	8+16	48.6M	98	46.8	76.1
GDN(ours)	ResNet50	8	17.7M	26	43.7	73.1
GDN(ours)	ResNet50	16	17.7M	52	47.9	77.4
GDN_{En}(ours)	ResNet50	8+16	35.4M	78	**50.2**	**78.6**

Table 3. The Top1 accuracy with two scales of input size and #Frame

Input Size	#Frame	GFLOPs	Top1	
			SthV1	SthV2
224×224	8	26	43.7	57.6
112×112	8	6.6	42.7	56.2
224×224	16	52	**47.9**	**59.2**
112×112	16	13	46.1	58.9

Table 4. Results on the validation split of Something-Something V2 dataset, * denotes results of 5 crops

Method	#Frame	Backbone	Top1	Top5
TSN [3]	8	-	30.0	60.5
TSM [3]	8	ResNet50	59.1*	85.6*
GDN(ours)	8	ResNet50	57.6	84.6
GDN(ours)	16	ResNet50	59.2	85.1
GDN_{En}(ours)	8+16	ResNet50	**61.1**	**86.8**

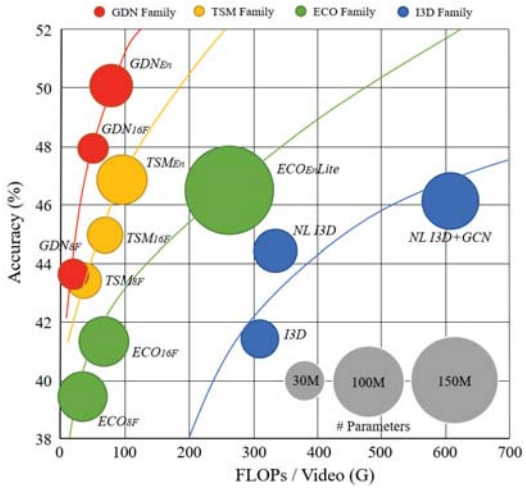

Fig. 3. GDN obtains better trade-off than other methods.

inference. Although spatial information is very important on action recognition, we find that we can get better trade-off between accuracy and computation by reducing the spatial size but increasing the number of input frames. We show the FLOPs of different input scales and frames in Table 3. We use two input scales, 224×224 and 112×112. Compared with 224×224, the 112×112 input only requires $1/4$ FLOPs but gets similar results. On Something-Something V1 dataset, the 224×224 outperforms 112×112 by 1.0% and 1.8% on 8 and 16 input frames respectively but requires $4\times$ FLOPs. If we increase the #Frame from 8 to 16, we will get increased accuracy (+4.2%, +3.4%) with only $2\times$ FLOPs. It means that the spatial resolution is not very important when the action is highly related to temporal order. We think that most spatial information is kept when spatial resolution is resized to 112×112. At the same time, it's difficult to recognize a temporal-sensitive action at a quick glance because of the shortage of temporal information. As a result, it's useful to increase the input frames. Under this circumstance, we can focus more on temporal information rather than spatial information.

4. CONCLUSION

In this paper, we propose a very simple but effective network GDN which modeling the spatial and temporal relation of videos. The GDM is obtained by applying channel grouping and kernel decomposition on 3D convolution kernels. Thus, the parameters and FLOPs are reduced greatly. In the future, we will explore the best channel ratios of different groups and better trade-off between spatial size and input frames.

5. REFERENCES

[1] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al., "The" something something" video database for learning and evaluating visual common sense.," in *ICCV*, 2017, vol. 1, p. 3.

[2] 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 *European conference on computer vision*. Springer, 2016, pp. 20–36.

[3] Ji Lin, Chuang Gan, and Song Han, "Tsm: Temporal shift module for efficient video understanding," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7083–7093.

[4] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu, "Collaborative spatiotemporal feature learning for video action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7872–7881.

[5] Joao Carreira and Andrew Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[6] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[7] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He, "Slowfast networks for video recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6202–6211.

[8] Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Xinmei Tian, and Tao Mei, "Learning spatio-temporal representation with local and global diffusion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12056–12065.

[9] Karen Simonyan and Andrew Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576.

[10] Frank Steinbrücker, Thomas Pock, and Daniel Cremers, "Large displacement optical flow computation withoutwarping," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 1609–1614.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[12] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459.

[13] Zhaofan Qiu, Ting Yao, and Tao Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5533–5541.

[14] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 305–321.

[15] Ting Zhang, Guo-Jun Qi, Bin Xiao, and Jingdong Wang, "Interleaved group convolutions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4373–4382.

[16] Chenxu Luo and Alan L Yuille, "Grouped spatial-temporal aggregation for efficient action recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5512–5521.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[18] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[19] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli, "Video classification with channel-separated convolutional networks," *arXiv preprint arXiv:1904.02811*, 2019.

[20] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6546–6555.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[22] Farzaneh Mahdisoltani, Guillaume Berger, Waseem Gharbieh, David Fleet, and Roland Memisevic, "On the effectiveness of task granularity for transfer learning," *arXiv preprint arXiv:1804.09235*, 2018.

[23] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox, "Eco: Efficient convolutional network for online video understanding," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 695–712.

[24] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.

[25] Xiaolong Wang and Abhinav Gupta, "Videos as space-time region graphs," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 399–417.