GROUPED TEMPORAL ENHANCEMENT MODULE FOR HUMAN ACTION RECOGNITION

Hong Liu, Bin Ren, Mengyuan Liu, Runwei Ding✉

Key Laboratory of Machine Perception, Peking University, Shenzhen Graduate School
{hongliu, sunshineren, dingrunwei}@pku.edu.cn
Sun Yat-sen University, School of Intelligent Systems Engineering
nkliuyifang@gmail.com

ABSTRACT

Temporal information is a significant cue for recognizing human actions from videos. Different from 2D CNN which can only capture spatial information in an efficient way, 3D CNN is good at capturing both spatial and temporal information at the expense of high computational cost. Beyond both methods, this paper presents a Grouped Temporal Enhancement (GTE) module which even outperforms 3D CNN, meanwhile only needs similar low computational cost as 2D CNN. The GTE module firstly decomposes an input video into spatial and temporal groups along channel dimension, and then uses a learnable temporal shift (LTS) operation for efficient temporal modeling. Finally, a 2D convolution filter is used to enhance the ability of LTS for spatial modeling. Extensive experiments on three benchmark datasets validate the effect of our method.

Index Terms— Action Recognition, Spatial-Temporal Modeling, Video Classification

1. INTRODUCTION

Human action recognition can be used in a wide range of applications such as intelligent surveillance system, virtual reality, human behavior analysis [1]. Different from traditional image processing task, recognizing similar human actions from videos usually heavily depends on the usage of temporal information among video frames. Currently, efficient modeling of temporal information is still an open problem.

Existing methods about temporal modeling for video action recognition can be summarized into three categories. The first type is 3D CNN based methods like [2, 3, 4], due to temporal information can be mixed together with spatial features, exceeding results had been gained recently. However, it is hard to train and time consuming for its huge amounts of parameters. The second type is RNN/LSTM based methods like [5, 6], which is suitable for simple sequence data such as skeleton. When taking RGB frames or video as input, the performance will be limited because too much useful spatial information within backgroud is abandoned. Finally, two-stream based methods, the third kind strategy, [7, 8] generally consist of an RGB stream and a flow stream, and temporal information is captured by flow stream but costs over 90% of the run time because of the extraction of optical flow. Hence how to balance the performance and computation cost when taking both spatial and temporal relations into account still needs to be studied.

Relation to prior work: 3D CNN is the nature choice when recognition tasks transited from image to video for actions. C3D [3] firstly utilized 3D CNNs for learning spatiotemoral features but with huge quantity of parameters and hard to train shown in Fig. 1(b). P3D [9] explored sevral decomposition strategies to reduce parameters by dividing 3D filters. Recently, Slowfast [10] designed a two-pathway network with two input frame rates to capture spatial and motion cues inspired by retinal ganglion in primates, which still requiring heavy computation resources. To handle those nuts. TSM [11] shift a fixed proportion of input feature along time dimension with surprising results while with no extra parameters introduced shown in Fig. 1(a). What's more, GST [12] divided input feature into spatial and temporal groups along channels, but the 3D conv kernels used in temporal branch is still not an optimal choice for its huge number of parameters.

To this end, we propose the efficient Gouped Temporal Enhanced (GTE) module, which not only strengthen the temporal modeling ability with fewer parameters, but also maintain a good spatial modeling ability in an efficient way. The main contributions of our GTE can be summarized as follows: (i) An effective sub-module LTS is proposed to enhance the ability of temporal modeling with little extra computation cost; (ii) By grouping input feature and adding a 2D CNN after LTS, we design the proposed GTE which can be utilized in 2D CNN framework in an effective yet parameters saving way; (iii) Extensive experiments demonstrate that we gain

This work is supported by National Natural Science Foundation of China (NSFC, No.U1613209, 61673030), National Natural Science Foundation of Shenzhen (No.JCYJ20190808182209321).

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on May 30,2021 at 16:15:03 UTC from IEEE Xplore. Restrictions apply.

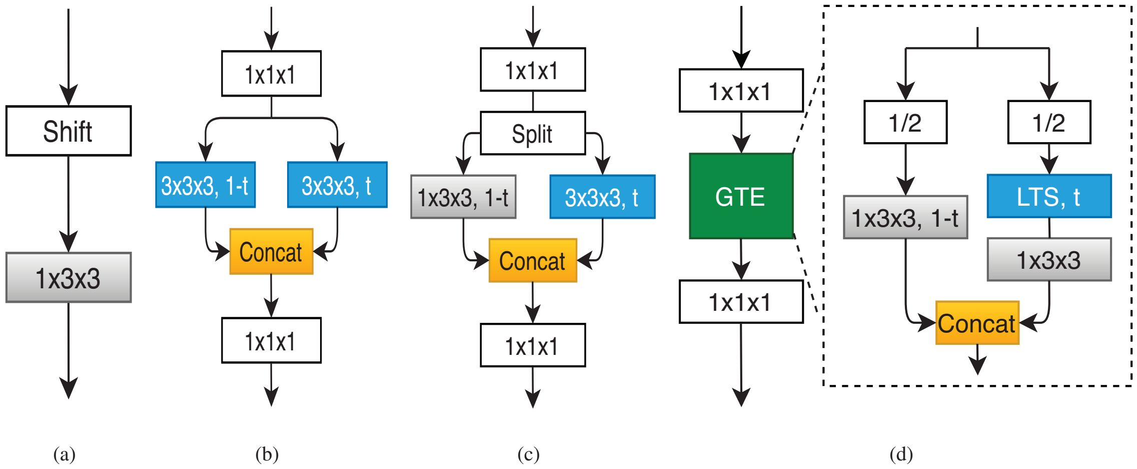


Fig. 1. Residual part comparision of C3D decomposition approaches and the proposed GTE. **(a)** TSM. **(b)** C3D-Equivalent methods like C3D. **(c)** Grouped Spatial-Temporal Aggregation. **(d)** The framework of our Grouped Temporal Enhancement (GTE) module. In GTE, input feature map is divided into two groups for spatial and spatial-temporal modeling respectively. The parameter t used in those schematics is to specify the proportion of spatial and temporal branches.

competitive performances on three public benchmark datasets with less parameters and FLOPs compared to related state-of-the-art methods.

2. PROPOSED METHOD

2.1. Overall Framework

Most methods [3, 13] reduce the complexity by decouping 3D convolution kernels as shown in Fig. 1(b). Unlike those approaches, the proposed Grouped Temporal Enhancement (GTE) module shown in Fig. 1(d) firstly splits input along channels into two branches similar to GST [12]. One branch is for spatial modeling using standard 2D convolution and then another branch is used for temporal modeling via our sub-module Learnable Tempral Shift (LTS) operation inspired by TSM [11] shown in Fig. 1(a). Compared with original 3D CNN used in GST shown in Fig. 1(c), LTS can reduce parameters more efficiently and a standard 2D convolution filter followed after LTS also strengthen the whole spatial modeling ability.

2.2. Learnable Temporal Shift Module

To make full use of the given video data and strengthen the ability of bakebones, we replace the original 3D CNN in GST with the proposed simple yet efficient sub-module named learnable temporal shift (LTS). After LTS, a standard 2D convolution filter is added, as a supplement for spatial branch to strengthing the spatial modeling ability.

Precisely, the input activation in video classification problem can be represented as X in shape of (N, C, T, H, W) where N, C, T are batch size, channel number, temporal dimension respectively, H is the height and W is the width. Unlike original TSM [11], for a certain frame, which just utilized shift operation before convolution layers along temporal dimension with different fixed portion of channels towards its previous and next frames just like Fig. 2(a). LTS transforms shift operation into learnable shift module, which makes shift portion and ratio more suitable for specific actions in a flexibly learnable way shown in Fig. 2(b), while only with very few extra parameters increase compared with original TSM.

It's obvious that temporal shift operation in TSM can be replaced with two specific forward/backward shift kernels $W_f = (a1, a2, a3) \in R^3$, $W_b = (b1, b2, b3) \in R^3$, and more specificly in TSM $W_f = (0, 0, 1)$, $W_b = (1, 0, 0)$ in temporal dimension. However, the kernels are fixed, so a certain frame could only get features from its neighbor frames in forward and backward direction. Inspired by the task of video inpainting [14, 15], which need more information across temporal dimension in a wide range. In GTE, we replace original hard-coded shift operation with soft learnable weight, making the shifting kernels also learnable instread of the fixed $W_f = (0, 0, 1)$, $W_b = (1, 0, 0)$.

Mathematically, learnable temporal shift can be expressed in a simple convolutional way as:

$$Feature_{input} = (Input)^{\wedge} \tag{1}$$

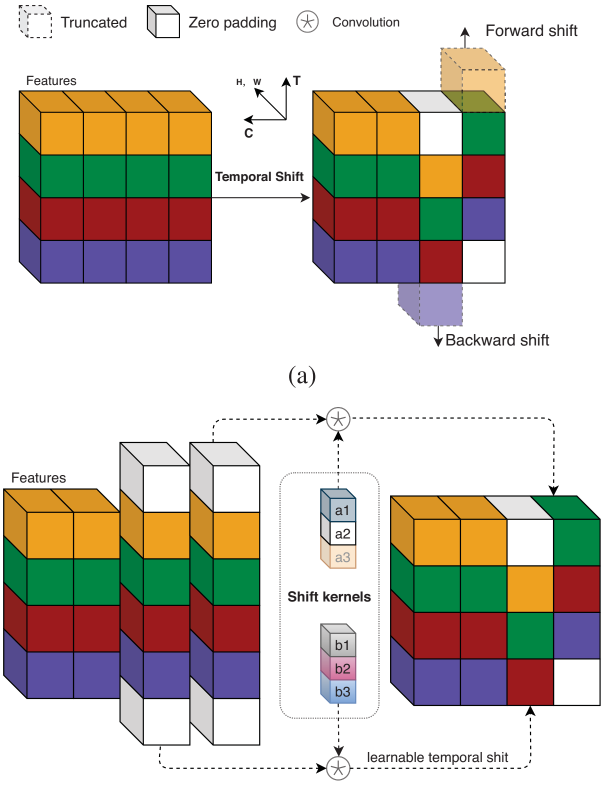
$$Feature_{output} = W_l * Feature_{input} \tag{2}$$

$$Output = (Feature_{output})^{\vee} \tag{3}$$

Here the operator $(.)^{\wedge}$ and $(.)^{\vee}$ are used to keep the $Output$ and $Input$ share the same dimension in (N, C, T, H, W) for consistency, the dimensions of $Feature_{input}$ and $Feature_{output}$ are adjusted to fit the learnable parameteres W_l, and the operator $*$ indicates the lenrable temporal shift operation.

According to the proposed learnable temporal shift module, the model could get useful features from more further neighbor frames along temporal dimension. Compared to the hard-coded time-shift operations in naive TSM with only

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on May 30,2021 at 16:15:03 UTC from IEEE Xplore. Restrictions apply.



(a)

(b)

Fig. 2. Clarification of the learnable temporal shift (LTS). Each row in the same color means a certain frame and a row with different color indicates a temporal dimension mixed frame. **(a)** Naive temporal shift module which shifts a fixed portion channels towards both forward and backward along the temporal dimension. **(b)** The proposed learnable shift (LTS) operation which replace the naive shift with forward/backward shifting kernels on temporal-channel map.

two shift kernels for backward and forward, the shift portion and number of our learnble kernels now also become learnable. And especially for video action recognition, this module enhances the model's capability greatly, which almost achieves the equal performance to 3D CNN while with fewer parameters compared with 2D CNN based methods.

After learnable temporal shift module, we add an extra 2D filter to strenghten the spatial modeling ability within the temporal modeling branch. Compared with the 3D convolution kernel used in temporal branch of GST, our method can sharply reduce parameters in temporal branch.

2.3. Efficient Spatial-Temporal Decomposition

Group convolution is not an unusual strategy to reduce parameters in image recognition tasks. However, conventional group convolution used in video tasks would result in symmetric groups which can't fully utilize the information behind spatial scenes and temporal cues, like [16]. So for a more optimal and purpose, we firstly divide the input feature into two groups along channels, then apply different operations for each branch instead of the same operation on all channels.

Table 1. Comparision of the number of parameters for each spatial-temporal block for listed methods.

Method	#Params
C2D	$k_H k_W C_i C_o$
C3D	$k_T k_H k_W C_i C_o$
GST	$(1 - t + tk_T) k_H k_W C_i C_o / 2$
GTE(ours)	$(k_H k_W + tk_T) C_i C_o / 2$

Similar to GST [12], to control the complexity of GTE, the parameter t is introduced to specify the proportion of spatial and temporal branches. So for output channels C_o, we have $C_{ot} = tC_o$ channels for the temporal modeling branch, while the rest for spatial modeling branch. C_i means the input channels. Parameters of spatial P_s and temporal path P_t are:

$$P_s = (1 - t) k_H k_W C_{i_s} C_o \quad (4)$$

$$P_t = P_{LTS} + P_{2dCNN} = t(k_T + k_H k_W) C_{i_t} C_o \quad (5)$$

where k_H and k_W indicate the spatial kernel size and k_T indicates temporal kernel size, P_{LTS} and P_{2dCNN} mean the parameters in learnable shift module (LTS) and the followed standard 2D convolution filter respectively. Especially, in GTE, we utilize 3D CNN with kernel size $(3, 1, 1)$ to make temporal shift module learnable, also divide input features into two groups along channels. So $C_{i_s} = C_{i_t} = C_i/2$ and the final GTE $Param_{GTE}$ can be calculated with the following equations:

$$Param_{GTE} = P_s + P_t = (k_H k_W + tk_T) C_i C_o / 2 \quad (6)$$

To summarize, we also list the number of parameters for different methods in Table 1.

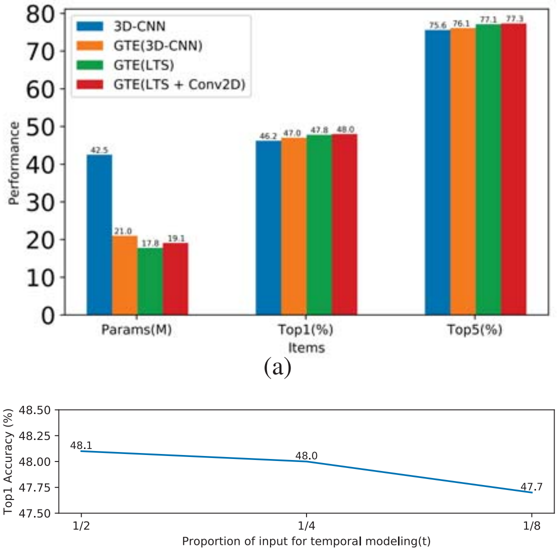
3. EXPERIMENTS

3.1. Datasets & Implementation Detail

Datasets: We train and evaluate GTE on three temporal related standard action recognition benchmarks. **Something-Something** [17] v1 and v2 are two large video datasets which consist of a large collection of densly-labeled video clips showing human pre-defined basic actions with daily objects in 174 classes. The number of videos in something-v2 is greatly increased from $108,499$ to $220,847$, and most actions within these two dataset cannot be recognized without the temporal relationship. **Diving48** [18] is also a new released long-term temporal information relied dataset which contains more than $18K$ video clips for 48 classes. Ablation study is mainly conducted in Something-v1 datasets.

Implementation Detail: For a fair comparision with other related methods, we choose ResNet-50 pretrained on ImageNet as backbone. And for LTS, the learnable parameters are intialized with $W_f = (0, 0, 1)$, $W_b = (1, 0, 0)$, this can be seen as a TSM [11]-like intialization. For temporal branch, the same strategy was adopted as TSN [8].

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on May 30,2021 at 16:15:03 UTC from IEEE Xplore. Restrictions apply.



(a)

(b)

Fig. 3. Ablation Study of GTE on Something-v1. For all experiments, frames are set to 8 for each video clips and backbone is ResNet-50 [19]. **(a)** GTE with different temporal modeling strategies and t is $1/4$. **(b)** The relation between performance and parameter t which means the proportion of input for temporal modeling.

Table 2. Comparision with the state-of-the-art results on Something-v1 datasets. Within these methods, S3D take BN-Inception as backbone while others utlize ResNet-50.

Method	#Frames	GFLOPs	Top1(%)	Top5(%)
TSM [11]	16	33	47.2	77.1
S3D [20]	64	66.8	47.3	78.1
GST [12]	8	29.5	47.0	76.1
GST [12]	16	59	48.6	77.9
GTE(ours)	8	**25.3**	48.1	77.4
GTE(ours)	16	50.6	**49.4**	**78.9**

We train GTE with 4 GTX 1080TI GPUS and batch-size is set to 48. Similary to GST [12], SGD is adopted to optimize our model with an initial learning rate of 0.01 for about 40 epochs. The learning rate is decaied by a factor of 10 every 10 epochs. Total training epochs are 60 and droupout ratio is set to 0.3. For inference, middle frame in each segment is sampled and center crop is also used for those middle frame.

3.2. Experimental Results

Ablation Study: In this part, we report the ablation analysis conducted on Something-v1 dataset. The results of general 3D CNN and GTE with different strategies for temporal modeling in temporal branch are shown in Fig. 3(a). GST can be seen as a special parameter configuration case of our GTE where the temporal branch is 3D CNN, and Fig. 3(b) shows the relation between performance and t. It can be easily found that GTE improves the accuracy with a large increasement compared with 3D CNN, and parameters are also sharply reduced. What's more, with the decrease of channels for temporal branch, performance will decrease, and parameters will increase from $18.0M$ to $19.3M$, we conclude this is because our GTE is more efficient for its LTS compared with original 3D filter utilized in GST. And when add a 2D convolution after LTS module shown within line of dashes in Fig. 1(d), the temporal branch will also become spatially related. So accompanied with the original spatial branch, GTE not only performs temporal modeling in a more efficient manner, but also strengthens the spatial modeling ability without extra cost.

Table 3. Comparision with the state-of-the-art results on Something-v2 datasets with validate set.

Method	#Frames	Top1(%)	Top5(%)
TSM [11](our impl.)	16	60.3	85.4
GST [12]	8	61.6	87.2
GST [12]	16	62.6	87.9
GTE(ours)	8	62.3	87.8
GTE(ours)	16	**63.2**	**88.3**

Table 4. Results on Diving48 datasets. For C3D and GST, the results are conducted by [12].

Method	Pre-training	Accuracy(%)
TSM [11](our impl.)	ImageNet	36.3
C3D [3, 12]	ImageNet	34.5
GST [12]	ImageNet	38.8
GTE(ours)	ImageNet	**39.7**

Comparison with state-of-the-arts. Table 2 shows the results of GTE and several related state-of-the-art methods on Something-v1 dataset. Our model with 8 frames could outperforms TSM which using 16 frames, and there is also an obvious improvement about 1% for GTE compared with GST [12]. It can be concluded that GTE achieves a comparable result in temporal related dataset such as Something-v1 with lower GFLOPs in a more efficient way.

What's more, we also train and test GTE on validation set of Something-v2 and dataset Diving48, GTE again outperforms other methods shown in Table3 and Tab4. For Diving48 [18], to make the results of the GTE comparable, 16 frames from each video clip are sampled for testing our model.

4. CONCLUSION

In this work, Grouped Temporal Enhancement (GTE) is proposed which sharply reduce the parameters in human action recognition problem. Within GTE, we design the learnable temporal shift (LTS) module which flexibly improve temporal modeling ability with few extra cost through more further neighbor frames. And a 2D convolution filter after LTS can make still further progress, which can be seen as a supplement of spatial modeling branch. As a consequence, at least competitive results have been achieved via our GTE on three standard datasets. Our further work will focus on how to divide the channels flexible in a data driven way instead of hard-coded two groups.

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on May 30,2021 at 16:15:03 UTC from IEEE Xplore. Restrictions apply.

5. REFERENCES

[1] Mengyuan Liu, Hong Liu, and Chen Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognition*, vol. 68, pp. 346–362, 2017.

[2] 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.

[3] 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.

[4] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2012.

[5] Yin Fan, Xiangju Lu, Dian Li, and Yuanliu Liu, "Video-based emotion recognition using cnn-rnn and c3d hybrid networks," in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, 2016, pp. 445–450.

[6] Amin Ullah, Jamil Ahmad, Khan Muhammad, Muhammad Sajjad, and Sung Wook Baik, "Action recognition in video sequences using deep bi-directional lstm with cnn features," *IEEE Access*, vol. 6, pp. 1155–1166, 2017.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] Lin Sun, Kui Jia, Dit-Yan Yeung, and Bertram E Shi, "Human action recognition using factorized spatio-temporal convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4597–4605.

[14] Chuan Wang, Haibin Huang, Xiaoguang Han, and Jue Wang, "Video inpainting by jointly learning temporal structure and spatial details," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 5232–5239.

[15] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu, "Learnable gated temporal shift module for deep video inpainting," *arXiv preprint arXiv:1907.01131*, 2019.

[16] 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.

[17] 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 *Proceedings of the IEEE International Conference on Computer Vision*, 2017, vol. 1, p. 3.

[18] Yingwei Li, Yi Li, and Nuno Vasconcelos, "Resound: Towards action recognition without representation bias," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 513–528.

[19] 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.

[20] 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.

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on May 30,2021 at 16:15:03 UTC from IEEE Xplore. Restrictions apply.