3D-TDC: A 3D Temporal Dilation Convolution Framework for Video Action Recognition

Yue Ming[a,**], Fan Feng[a,*], Chao Li[a] and Jing-Hao Xue[b]

[a]Beijing Key Laboratory of Work Safety and Intelligent Monitoring, School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
[b]Department of Statistical Science, University College London, London WC1E 6BT, U.K.

ARTICLE INFO

Keywords:
3D convolution
temporal dilation
action recognition
temporal compression

ABSTRACT

Video action recognition is a vital area of computer vision. By adding temporal dimension into convolution structure, 3D convolution neural network owns the capacity to extract spatio-temporal features from videos. However, due to computing constraints, it is hard to input the whole video into the convolution network at one time, resulting in a limited temporal receptive field of the network. To address this issue, we propose a novel 3D temporal dilation convolution (3D-TDC) framework, to extract spatio-temporal features of actions from videos. First, we deploy the 3D temporal dilation convolution as the shallow temporal compression layer, enabling an effective capture of spatio-temporal information in a larger time domain with the reduced computational load. Then, an action recognition framework is constructed by integrating two networks with different temporal receptive fields to balance the long-short time difference. We conduct extensive experiments on three widely-used public datasets (UCF-101, HMDB-51, and Kinetics-400) for performance evaluation, and the experimental results demonstrate the effectiveness of our proposed framework in video action recognition with low computational load.

1. Introduction

Video action recognition encompasses a wide range of applications, such as human computer interaction, smart video surveillance, sports, and health care [1]. It has made great progress, due to the rapid development of deep networks. These deep networks can be mainly divided into 2D and 3D convolution networks. A 2D convolution network, although effective for image recognition, is not strong to model the temporal information, while a sequential reasoning structure, such as recurrent neural network, is not sufficiently effective in visual analysis. Therefore, for video action recognition, on the one hand, two-stream methods [2–4] construct 2D convolution networks with the input of both RGB image and optical flow. The optical flow, however, is computationally costly, which limits the practical applications of such a method. On the other hand, 3D convolution networks [5–10] directly construct an end-to-end model to extract both the temporal and spatial features of actions, but such a network usually entails a large number of parameters and computation. As a result, how to improve the network structure, such that the action model can extract the spatio-temporal features in a large time domain with finite computation, has become a research focus of video action recognition.

Because the duration of a video varies, the video needs to be cut into segments with a fixed temporal size determining the temporal receptive field of the network. The recognition accuracy of a 3D convolution network is thus limited by the temporal size of each segment. To address this issue, in this paper, we propose a novel action recognition framework called 3D temporal dilation convolution framework (3D-TDC, as shown in Figure 1). First, we regularize the video duration for the network input. Then, the shallow temporal compression layer is introduced to embed 3D temporal dilation convolution, for effectively capturing the spatio-temporal information with a reduced computational load. Finally, the action recognition framework is constructed by integrating two networks with different temporal receptive fields, which can effectively control the network computational load while improving the recognition accuracy. The main contributions of this work are threefold:

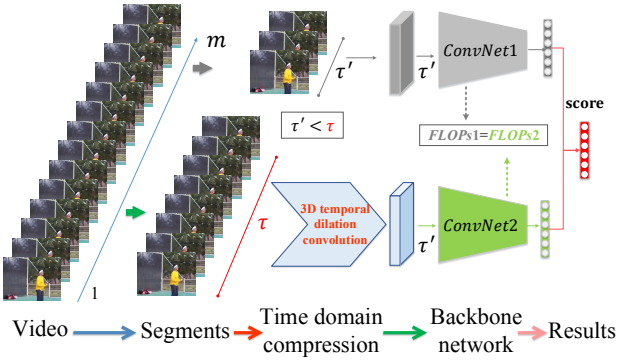

Figure 1: A novel 3D temporal dilation convolution (3D-TDC) framework. It consists of two branches with different time-domain sizes (τ and τ') but the same computational load ($FLOPs$). m is the total number of video frames.

1. *Better spatio-temporal feature extraction*: The 3D temporal dilation convolution embedded as the shallow temporal compression layer can effectively improve the temporal receptive field of the network for bet-

*Co-First author
**Corresponding author: Yue Ming
myname35875235@126.com (Y. Ming)
ORCID(s):

ter exploration of the spatio-temporal information and can reduce the computational load.

2. *Improved recognition performance*: By integrating two networks with different temporal receptive fields through parameter transfer, our proposed 3D-TDC framework can improve the accuracy of video action recognition.
3. *Good practicability*: Our 3D-TDC framework achieved superior performance on different benchmark datasets with a large range of tasks, including UCF-101, HMDB-51, and Kinetics-400. Moreover, our method can effectively balance the computational load and recognition accuracy of the network, vitally in practice.

The rest of the paper is organized as follows. In Section 2, we summarize the related work. Section 3 describes our proposed method and framework. The implementation details, experimental results, and analysis are presented in Section 4. Finally, conclusions are drawn in Section 5.

2. Related Work

Video action recognition methods can be roughly categorized into traditional and deep-learning methods. Traditional methods extracted low-level action features from videos. The deep neural networks extracted high-level action semantic features and adopted end-to-end models to carry out unified feature extraction and classification [11]. In this section, we focus on the related work in deep learning, which can be mainly divided into three groups: two-stream convolution framework, dilation convolution network, and spatio-temporal convolution network.

2.1. Two-stream convolution framework

Two-stream convolution neural networks [2] extracted dynamic time-domain motion (optical flow) features and static spatial RGB features with two independent networks. These networks can be summarized from three aspects: input, network, and optimization. In the input aspect, the sparse time sampling strategies [3, 4] were used to segment video in the time domain; features were extracted from different segments; multiple branches were used to fit a whole-video-level recognition. FlowNet [12] and hidden two-stream convolution networks [13] focused on using convolution networks to generate optical flow features. TVNet [14] proposed an end-to-end neural network to simulate optical features. DMC-Net [15] and [16] were based on the video compression domain, transferring motion information from optical flow. In the network aspect, [17] explored different fusion algorithms of networks. AssembleNet [18] sought neural architectures with better connectivity and spatio-temporal interactions. CTAN [19] integrated the channel-wise attention mechanisms into networks. Yang et al. [20] proposed a generic temporal pyramid network at the feature level to capture action instances at various tempos. Besides, recurrent neural network based on two-stream convolution frameworks [21, 22] were used to directly fuse the convolution features to complete the temporal reasoning. In the optimization aspect, VLAD [23] aggregated deep features across the entire video according to adaptive video feature segmentation and sampling. PBNets [24] designed a watch-and-choose mechanism to optimize the back-propagation algorithms during two-stream network training. However, the motion feature coding of the two-stream frameworks depended on the optical flow, leading to a huge computational load in the input stage. In this work, we only use RGB images as the input of a two-stream network to pursue comparable performance with optical flow.

2.2. Dilation convolution network

Dilation convolution [25] in action recognition usually adopted to model temporal features and extract larger contextual information. In [26], a dense dilated network was trained to recognize actions from clip-level to global-level, by fusing outputs from each densely-connected dilated convolutions layer. In temporal aggregation network (TAN) [27], a dedicated temporal aggregation block was designed to encode multi-scale spatio-temporal patterns, and larger temporal context can be captured by dilated convolutions effectively. For long videos [28, 29], encoder-decoder temporal convolutional networks (TCN) can capture spatio-temporal and contextual information from adjacent image frames, and share the parameters between all time steps. Dilated-TCN [30] fused residual connections and dilated convolutions to model long-range temporal relationship. After that, MS-TCN [31] combined multiple dilated-TCNs [30] to form a multi-stage framework, in each stage of which the prediction results of the previous stage were refined. In the untrimmed videos with densely distributed actions, selecting the key temporal information is particularly vital. In [32], dilated attention layers (DAL) were proposed to encode representative local features, by weighting attentional coefficients to different frames. Based on multiple DALs deploying different dilation rates, a pyramid dilated attention network (PDAN) can structure both short-term and long-term temporal relations. However, these approaches require a complex design of dilation convolution and multiple blocks of the network. In this work, we are mainly interested in designing dilation convolution only in the shallow layer, instead of multiple dilation convolution blocks throughout the network.

2.3. Spatio-temporal convolution network

Spatio-temporal convolution networks are usually end-to-end models, including 3D convolution network and its variants. The 3D convolution network [5] added a temporal dimension into the 2D convolution network, which enables a convolution network to simultaneously mine temporal and spatial features. For network structure, the mature topological structures of 2D convolution networks were transferred into the 3D convolution network. P3D [6] and R(2+1)D [8] split the 3D convolution kernel into convolution of space and convolution of time. TSM [33] proposed a temporal shift module to achieve the balance between the computational load of 2D CNN and the performance of 3D CNN. COST [10] proposed to extract spatio-temporal features from three video orthogonal views by three convolu-

tions with shared parameters. Song et al. [34] introduced a temporal-spatial mapping for capturing the temporal evolution of the frames by jointly analyzing all the frames of a video. Materzynska et al. [35] proposed a spatial-temporal interaction network to reason the geometric relations between constituent objects and an agent performing acting on compositional action recognition. [36] and [37] explored different structure constructions and the correlation between spatio-temporal channels. Different spatio-temporal coding structures [38–42] were proposed for improving the discrimination ability of spatio-temporal convolution networks. Besides, graph convolution networks (GCN) [43, 44] based on 3D convolution were adopted to capture the appearance features and the temporal relation between video sequences. For optimization, the video action transformer network introduced an attention mechanism to a 3D convolution network. Spatial-temporal attentive convolution neural network (STA-CNN) [45] incorporated a temporal attention mechanism and a spatial attention mechanism into a unified convolution network to recognize actions in videos. MARS [46] proposed a training method for a 3D convolution network under the supervision of optical flow. Kim et al. [47] presented random mean sampling (RMS) to relieve the overfitting in 3D residual networks. For framework construction, SlowFast networks [48] proposed a framework composed of two networks with different time scales. Sudhakaran et al. [49] introduced spatial gating in spatial-temporal decomposition of 3D kernels without additional parameters and computational overhead. Compared with 2D convolution networks, the spatio-temporal convolution framework has significant advantages in recognition performance for video actions. However, due to the existence of time domain in the hidden layers, the computational load of spatio-temporal convolution networks increases sharply with the expansion of the temporal receptive field. In this work, we adopt the dilation convolution to explore a tradeoff between computational load and temporal receptive field size.

3. Method

To balance the computational effectiveness and recognition accuracy, we propose a 3D temporal dilation convolution (3D-TDC) framework deployed as the shallow temporal compression layer, which can effectively extract the spatio-temporal features from a larger temporal receptive field without heavy computational load. In this section, we first introduce the action video prepossessing mechanism as the network input (section 3.1). Then, we describe the 3D temporal dilation convolution to demonstrate the effect of good temporal compression (section 3.2). Finally, we present our framework construction devoting to effective extraction of the spatio-temporal features of the action video, high recognition accuracy, and low computational load (section 3.3).

3.1. Network input

Suppose the original video input is $\mathbf{I}_T(:,:,\mathbf{t}) \in \mathbb{R}^{w\times h\times m}$, where m is the total number of the video frames ($\mathbf{t} \in \mathbb{R}^{m\times 1}$) and $w \times h$ is the spatial size of the frame. The temporal size of the video (m) varies from video to video. Therefore, for network input, the video needs to be segmented into different local temporal clips with a fixed temporal size as follows:

$$\mathbf{t} = (t_1, \dots, t_m)^T = (\mathbf{s}_\tau, \mathbf{s}_{2\tau}, \mathbf{s}_{3\tau}, \dots, \mathbf{s}_{n\tau}), \tag{1}$$

where $\mathbf{s}_\tau = (t_1, t_2, \dots, t_\tau)^T$, $\mathbf{s}_{2\tau} = (t_{\tau+1}, t_{\tau+2}, \dots, t_{2\tau})^T$, $\dots$, $\mathbf{s}_{n\tau} = (t_{n\tau+1}, t_{n\tau+2}, \dots, t_m)^T$; n is the number of clips; τ is the fixed temporal size. Then the video can be expressed as

$$\begin{aligned}\mathbf{I}_T(:,:,\mathbf{t}) &= \mathbf{I}_T(:,:,(t_1,\dots,t_m)^T)\\ &= \begin{bmatrix} \mathbf{I}_T(:,:,(t_1,t_2,\dots,t_\tau)^T)\\ \mathbf{I}_T(:,:,(t_{\tau+1},t_{\tau+2},\dots,t_{2\tau})^T)\\ \vdots \\ \mathbf{I}_T(:,:,(t_{n\tau+1},t_{n\tau+2},\dots,t_m)^T)\end{bmatrix}\\ &= \begin{bmatrix} \mathbf{I}_T(:,:,\mathbf{s}_\tau)\\ \mathbf{I}_T(:,:,\mathbf{s}_{2\tau})\\ \vdots \\ \mathbf{I}_T(:,:,\mathbf{s}_{n\tau})\end{bmatrix}.\end{aligned} \tag{2}$$

Convolution network $f^{ConvNet}$ encodes the clips separately and generates the output vector $\mathbf{score}_i \in \mathbb{R}^{1\times c}$ of the ith clip, where c is the number of action categories. Finally, the output vector $\mathbf{score} \in \mathbb{R}^{1\times c}$ of the whole video is obtained by average fusion of all clips:

$$\begin{aligned}\mathbf{score}_1 &= f^{ConvNet}(\mathbf{I}_T(:,:,\mathbf{s}_\tau))\\ \mathbf{score}_2 &= f^{ConvNet}(\mathbf{I}_T(:,:,\mathbf{s}_{2\tau}))\\ &\vdots\\ \mathbf{score}_n &= f^{ConvNet}(\mathbf{I}_T(:,:,\mathbf{s}_{n\tau}))\\ \mathbf{score} &= \frac{\sum_{i=1}^{n}\mathbf{score}_i}{n}\end{aligned} \tag{3}$$

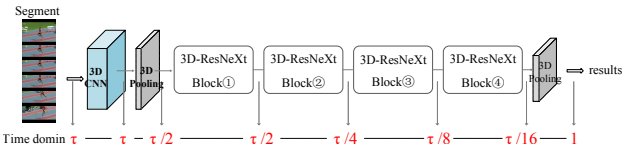

Figure 2: Temporal variation in 3D convolution network.

For the 3D convolution network, 3D ResNeXt-101 [9], the temporal size decreases gradually over the hidden layers to aggregate the spatio-temporal features, as shown in Figure 2. The temporal size of the first 3D-convolution-output feature maps remains the same as the original input, while after 3D pooling and four 3D-ResNeXt blocks, the temporal size of the hidden-layer feature maps is gradually compressed to encode the spatio-temporal features. For example, when $\tau = 16$, the temporal variation is 16-16-8-8-4-2-1-1. However, when the temporal size of the clip is small (e.g. $\tau = 16$), the temporal size in the network will be quickly compressed to a very low value, so it is difficult to obtain sufficient temporal correlation for the hidden layers in such a 3D convolution network.

The computational load $FLOPs$ (floating point of operations) of a 3D convolution layer is as follows:

$$FLOPs = \left[\left(k_w^{in} \cdot k_h^{in} \cdot C_{in} \cdot T_c\right) \cdot C_{out} + C_{out}\right] \cdot \left(H \cdot W \cdot \tau_{out}\right), \tag{4}$$

where k_w^{in}, k_h^{in} and T_c are the spatial and temporal sizes of convolution kernel; C_{in} and C_{out} are the numbers of input and output channels; H, W and τ_{out} are the spatial and temporal sizes of feature maps; for fixed T_c, temporal stride and temporal padding, τ_{out} is proportional to the temporal size τ. Therefore, when the clip temporal size τ is large, it will lead to a high $FLOPs$ for the network.

3.2. 3D temporal dilation convolution

To balance the clip temporal size τ and the convolution computational load ($FLOPs$), temporal dilation is introduced here into the time domain of the 3D convolution kernel, leading to a 3D temporal dilation convolution, as shown in Figure 3. Each 3D convolution kernel can skip a certain number of input frames, to improve the temporal receptive field. Temporal dilation coefficient D_t represents the number of intervals between frames in the 3D convolution kernel.

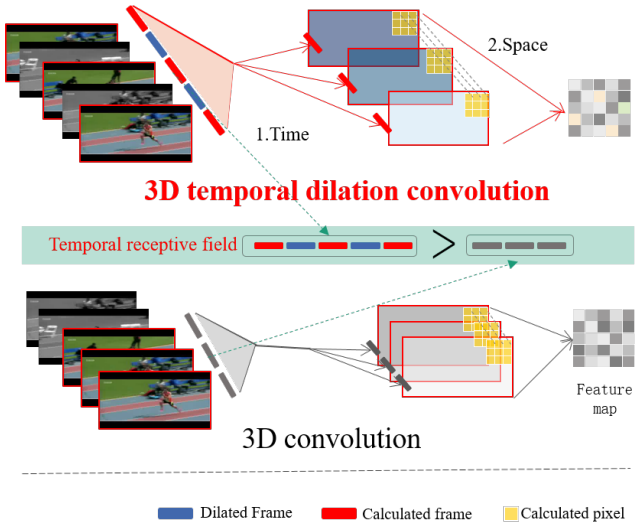

Figure 3: Comparison of temporal receptive fields of the 3D temporal dilation convolution and the basic 3D convolution.

As illustrated in Figure 3, the magnitude of the temporal receptive field R of basic 3D convolution operation is $R = T$, where T is the number of frames in a single convolution; in contrast, the magnitude of the temporal receptive field $R^{'}$ of 3D temporal dilation convolution is

$$R^{'} = T + (T-1) \cdot D_t. \tag{5}$$

Since $T \geqslant 1$ and $D_t \geqslant 1$, we can obtain

$$R^{'} \geqslant R. \tag{6}$$

That is, when T is fixed, the temporal receptive field of the 3D temporal dilation convolution kernel is larger than that of the original 3D convolution kernel.

Then the number of parameters $Params$ and computational load $FLOPs_{single}$ of a single 3D temporal dilation convolution (that is $\tau_{out} = 1$ in $FLOPs$) are

$$\begin{aligned} Params &= \left(k_w^{in} \cdot k_h^{in} \cdot C_{in} \cdot T_c\right) \cdot C_{out} + C_{out}, \\ FLOPs_{single} &= Params \cdot (H \cdot W). \end{aligned} \tag{7}$$

For the time domain, when T_c is fixed, the temporal dilation will not cause the above parameters to change. Therefore, compared with the original 3D convolution, 3D temporal dilation convolution will not increase the number of parameters $Params$ and the computational load $FLOPs_{single}$. Then, when τ_{out} is fixed, the computational load of a 3D convolution layer $FLOPs$ will remain unchanged. As a result, 3D temporal dilation convolution can enhance the temporal receptive field of the convolution without increasing the number of parameters and computational load.

3.3. Framework construction

As shown in Figure 1, we build a novel 3D temporal dilation convolution framework for action recognition. The temporal size in the hidden layers of the 3D convolution network decreases from the shallow layer to the deep layer, and the output temporal size of the former layer is the input of the latter layer. That is when the temporal size is reduced in the shallower layer, the computational load of the whole network decreases greatly. Therefore, we deploy the 3D temporal dilation convolution layer as the shallow layer of the network, which accomplishes the sparse expression of the larger time domain, as shown in Figure 4.

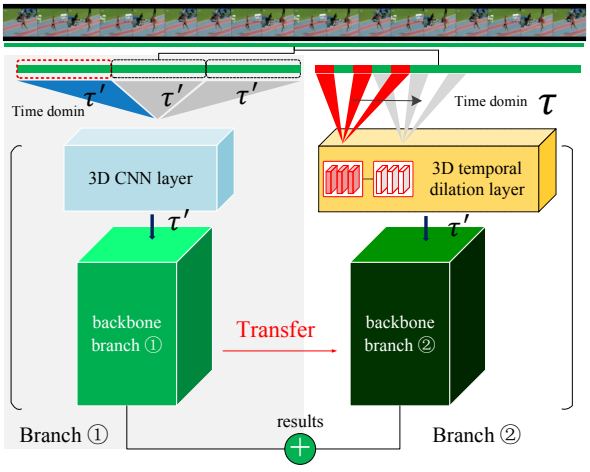

Figure 4: The 3D temporal dilation convolution (3D-TDC) framework. It consists of two branches, which input segments with two different temporal sizes $\tau^{'}$ and τ ($\tau^{'} < \tau$). Meanwhile, the computational load of the two branches remains the same.

As illustrated in Figure 4, the overall framework has two branches. Each branch consists of a shallow layer and a backbone branch (3D ResNeXt-101 [9]). The shallow layer of the "branch 1" is the original 3D convolution layer to extract the spatio-temporal features with a small temporal size ($\tau^{'}$) of the clip. The shallow layer of the "branch 2" is the

3D temporal dilation convolution layer to extract the features with a large temporal size (τ) of the clip. Finally, the two branches are fused to obtain the final classification results. The 3D temporal dilation convolution layer of "branch 2" compresses the temporal size from τ to τ', while maintaining the same computational load as the original 3D convolution layer of "branch 1".

The network training adopts multi-part training and transfer strategies. First, we use the backbone parameters trained by the large temporal size τ of the clip as the initialization parameters. Then, the "branch 1" is trained by the small temporal size τ' of the clip. Meanwhile, the first layer parameters of the initialization are removed and transferred to the "branch 2", which consists of the 3D temporal dilation convolution layers and the backbone branch. The "branch 2" is trained by the large temporal size τ of the clip to obtain the final parameters. We use cross-entropy losses with softmax and back-propagate their gradients.

4. Experiments

We evaluate the proposed 3D-TDC framework on three datasets: UCF-101, HMDB-51 and Kinetics-400. The descriptions of datasets, data preprocessing and training details are presented in section 4.1. Then we present the details of our experiments on the effect of temporal input size (section 4.2.1), the comparison of different temporal compression structures (section 4.2.2), the performance and computational load of each branch (section 4.2.3), and the comparison with other state-of-the-art methods (section 4.3).

4.1. Data preprocessing and training details

Datasets. Our experiments are performed on three action recognition datasets: UCF-101 [50], HMDB-51 [51] and Kinetics-400 [52]. UCF-101 contains 101 categories of actions, with 13,320 instances. The videos are mainly from movies and Google videos. HMDB-51 contains 51 action categories, with 6,766 instances. Each category contains at least 101 videos. The Kinetic-400 contains 400 action categories, each of which contains more than 400 training video samples, and the temporal length of each video is about 10s.

Data preprocessing. We first transform the original video into the frame sequences (at 25FPS) through FFmpeg[1], and resize the frame such that the smallest dimension is 256. During the training, we apply the data augmentation methods including random clipping, subtracting ActivityNet mean (114.7748, 107.7354, 99.475), and vertical and horizontal flipping with the spatial size of 112. For the training, three different temporal sizes τ (16 frames, 32 frames and 64 frames) of clips are generated for the experiments. For the testing, the original frame sequences are cut into continuous clips without overlap. The recognition accuracy is obtained by the average fusion of multiple clip accuracy.

Training details. The selected backbone networks are 3D ResNeXt-101 [9] with different depths. When constructing a 3D temporal dilation convolution framework, the shallow layer of "branch 2" is removed and then connected with different temporal compression layers, and finally fused with the "branch 2". Here we list the detailed structure parameters of the network with 101 layers, as shown in Table 1. In the training, the momentum gradient descent method is used. The corresponding parameters of weight attenuation are set to 0.001 and 0.9, and the dropout is 0.9. The initial learning rate is 0.01 based on the large temporal size τ (64 frames) of clips and 0.0001 based on the small temporal size τ' (16 frames) of clips. Meanwhile, the structural parameters of the main network are frozen for transfer. After transfer, the overall learning rate is set to 0.000001 as the final fine-tuning. The learning rate attenuation strategy is that, when the accuracy of the last three rounds is not improved, the learning rate is halved. The model parameters are saved every two iterations, and the model with the best performance is finally selected. The experiments use the Kinetics pre-trained model [9], and the $FLOPs$ are generated by the THOP[2] module in Python. The experiment is carried out on a processor equipped with two NVIDIA GeForce GTX Titan X and eight 1080Ti GPUs, and the deep learning framework in experiments is PyTorch[3].

Table 1
The instantiation of 3D temporal dilation convolution framework (101 layers). The structure parameters of convolution kernel are expressed as $\{k^3, C_{out}\}$, where k is the spatio-temporal size of the kernel, and C_{out} is the channel number; g means $group$; "$\times N$" means the number of blocks.

stage	"branch 1"	"branch 2"
conv1	$7^3, 64$	$7^3, D_t = 1, 64$
block1	$\begin{pmatrix} 1^3, 128 \\ 3^3, g = 32, 128 \\ 1^3, 256 \end{pmatrix} \times 3$	$\begin{pmatrix} 1^3, 128 \\ 3^3, g = 32, 128 \\ 1^3, 256 \end{pmatrix} \times 3$
block2	$\begin{pmatrix} 1^3, 256 \\ 3^3, g = 32, 256 \\ 1^3, 512 \end{pmatrix} \times 4$	$\begin{pmatrix} 1^3, 256 \\ 3^3, g = 32, 256 \\ 1^3, 512 \end{pmatrix} \times 4$
block3	$\begin{pmatrix} 1^3, 512 \\ 3^3, g = 32, 512 \\ 1^3, 1024 \end{pmatrix} \times 23$	$\begin{pmatrix} 1^3, 512 \\ 3^3, g = 32, 512 \\ 1^3, 1024 \end{pmatrix} \times 23$
block4	$\begin{pmatrix} 1^3, 1024 \\ 3^3, g = 32, 1024 \\ 1^3, 2048 \end{pmatrix} \times 3$	$\begin{pmatrix} 1^3, 1024 \\ 3^3, g = 32, 1024 \\ 1^3, 2048 \end{pmatrix} \times 3$
	fully-connected	fully-connected

[1] https://github.com/FFmpeg/FFmpeg

[2] https://github.com/Lyken17/pytorch-OpCounter

[3] https://pytorch.org

4.2. Ablation studies

4.2.1. Effect of temporal input size on 3D convolution network

To verify the effect of different temporal size (τ) of the video clip on the recognition accuracy of the 3D convolution network with the original 3D convolution in the shallow layer, we explore different temporal sizes (16, 32, and 64 frames) of clip input. The experimental results of the computational load ($FLOPs$) and accuracy are shown in Table 2 and Figure 5 (UCF-101 and HMDB-51).

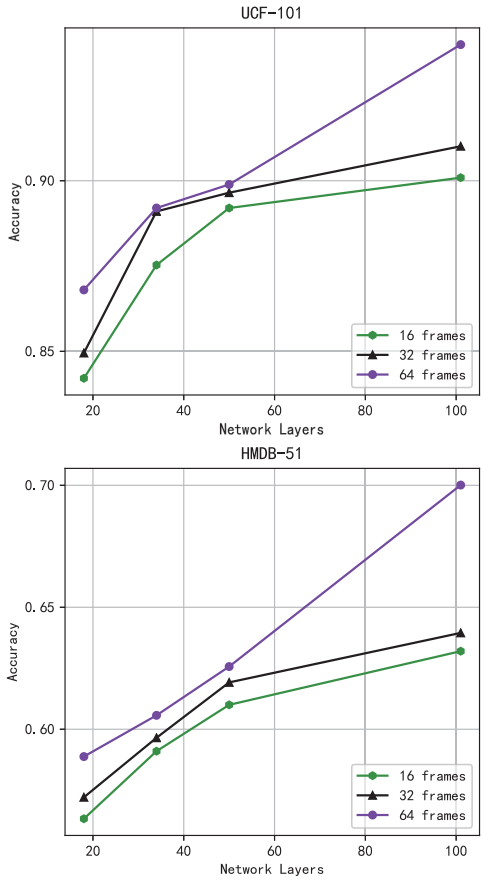

Figure 5: Recognition accuracy of 3D convolution network with different temporal sizes ($\tau = 16, 32, 64$) of clips from UCF-101 and HMDB-51.

As shown in Figure 5, the recognition accuracy of the 3D convolution network is significantly improved with the increase of temporal sizes from 16 to 64 frames, for any depth. It verifies that the spatio-temporal feature expressiveness of the network can be improved by increasing the temporal size τ of the clip. That is, for a 3D convolution network, a larger temporal receptive field can provide more sufficient temporal motion information. However, from Table 2, we can observe that: on the one hand, the computational load of the network increases gradually with the scale of model parameters; and on the other hand, when the temporal size τ of the clip is multiplied, the computational load of the network is multiplied, which requires higher computing capacity for devices. Therefore, the experiment demonstrates that the network recognition accuracy can be improved quickly by increasing the temporal size τ of the clip, but the network computational load will also increase sharply.

Table 2
$FLOPs$ of the 3D residual convolution network with different temporal sizes τ. "($xx.xx$ G)" is the computation load (FLOPs). "($xx.xx$ M)" is the number of parameters of network. The spatial size of input frame is 112×112.

τ	18 layers (33.25M)	34 layers (63.56M)	50 layers (26.07M)	101 layers (47.72M)
16	8.31G	12.71G	7.49G	9.61G
32	16.63G	25.41G	14.98G	19.20G
64	33.27G	50.83G	29.97G	38.40G

4.2.2. Comparison of different temporal compression structures

In this section, we take the 3D ResNeXt-101 [9] as the backbone, and combine it with the 3D temporal dilation convolution layer to build the 3D temporal dilation convolution framework, and compare it with other shallow temporal compression structures, including sparse sampling, $sliding^1$ and $sliding^2$, as shown in Figure 6.

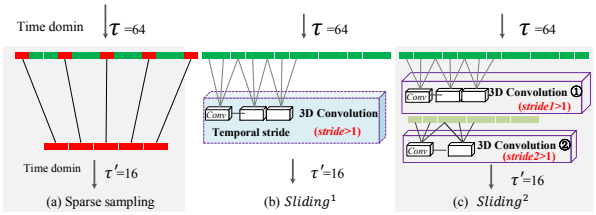

Figure 6: Different temporal compression structures.

For the video clips with temporal size $\tau = 64$, uniform sampling is carried out to obtain the temporal size $\tau' = 16$ of the clip. $sliding^1$ and $sliding^2$ adjust the temporal step: *stride* and *padding* of original 3D convolution layer. $sliding^1$ represents one layer, and $sliding^2$ represents two layers. $sliding^1$: convolution kernel size $k^1 = 7$, $stride^1 = 4$. $sliding^2$: $k_1^2 = 7$, $stride_1^2 = 2$; $k_2^2 = 3$, $stride_2^2 = 2$. The 3D-TDC layer: $k = 7$, $stride = 4$, $D_t = 1/2/3$. Then, we conduct an experimental comparison on the UCF-101 dataset, apply the above shallow temporal compression structures on the "branch 2" of the framework, and obtain the recognition accuracy and computational load ($FLOPs$) of the framework, as shown in Table 3 and Figure 7.

In sparse sampling, 64 frames are directly reduced to 16 frames, so that the temporal information is not compressed and mapped by parameter learning, but part of the infor-

Table 3
Recognition accuracy (%) of the frameworks with different shallow temporal compression structures on UCF-101. (sp stands for uniform sampling, sld^1 for $sliding^1$, and sld^2 for $sliding^2$.)

layers	original	sp	sld^1	sld^2	D_t=1	D_t=2	D_t=3
18	84.21	84.21	84.55	85.00	84.91	84.62	84.62
50	89.20	89.10	89.31	89.53	89.50	89.00	89.05
101	90.09	89.52	90.97	91.83	**91.85**	90.55	90.21

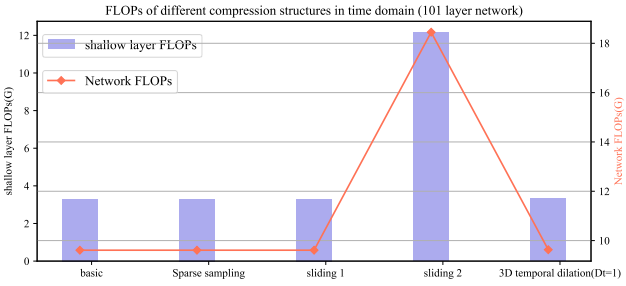

Figure 7: $FLOPs$ results of different temporal compression structures and networks. (101 layers)

mation in the long time domain is lost directly, resulting in low accuracy. In the shallow layer, the $sliding^1$ layer, $sliding^2$ layer, and 3D-TDC layer are to learn the temporal compression mapping relationship through convolution parameters. From Table 3, we can observe that, compared with the original network ($\tau = 16$), the accuracy of the frameworks using the above three temporal compression structures ($\tau = 64, \tau' = 16$) is significantly improved. Among them, the performances of $sliding^2$ and the 3D-TDC layer are the best. Both $sliding^1$ and $sliding^2$ compress the time domain by controlling the temporal stride, but the recognition accuracy of $sliding^2$ is higher. This is because, when the temporal stride is large, it is difficult to fully encode temporal relations, and $sliding^2$ learns better mapping relations through smaller stride with more parameters. However, the accuracy of 3D-TDC is almost the same as that of $sliding^2$ and is the highest one in the 101-layer network. Therefore, the larger temporal feature can be encoded by increasing the temporal size τ of the clip. When D_t increases, with the input temporal sparsity increasing, the input information loss increases during feature extraction, and thus the accuracy slightly decreases, but it is still higher than the original network.

We obtain the computational load $FLOPs$ of different temporal compression structures and the network, when the spatial size of the input frames is 112×112 and its temporal size is 64. According to Table 3, the frameworks with the $sliding^2$ layer and the 3D-TDC layer provide higher recognition accuracy, while the $FLOPs$ of the 3D-TDC layer is less than that of $sliding^2$ (Figure 7). This indicates that, by using the 3D-TDC layer, the recognition accuracy is improved and the computational load is also controlled.

In this framework, we further observe the difference, between the proposed 3D-TDC framework ($D_t = 1$) and the original 3D CNN, in the recognition confidence difference of the random selected examples from the UCF-101 dataset (Figure 8). $\Delta a1$ is the difference between the highest confidence and the sub-high confidence ($\Delta a1 = max_1(\mathbf{score}) - max_2(\mathbf{score})$. $\Delta a2$ is the difference between the highest confidence and the third highest confidence ($\Delta a2 = max_1(\mathbf{score}) - max_3(\mathbf{score})$. Here $max_n(\mathbf{score})$ is the n-th largest value in $\mathbf{score}$. Note that, $\Delta a1$ and $\Delta a2$ can highlight the difference between the prediction category confidence and other categories. From Figure 8, compared with the original 3D CNN, the $\Delta a1$ and $\Delta a2$ of the 3D-TDC framework is larger, indicating that the 3D-TDC framework extracts more discriminative features of actions in these samples. In addition to randomly selected examples for visualization, we calculate the mean confidence of $\Delta a1$ and $\Delta a2$ for each class based on UCF-101, as illustrated in Figure 9. From Figure 9, we can observe that $\Delta a1$ of 59 categories and $\Delta a2$ of 63 categories in 3D-TDC have a distinct improvement, compared with $\Delta a1$ and $\Delta a2$ of original 3D CNN, respectively. This reveals that the 3D-TDC framework has a stronger ability to identify specified video categories; that is, in these categories, the 3D-TDC framework can encode more discriminative representations.

4.2.3. *Performance and computational load of each branch*

For further elaboration of the motivation for designing the two-branch framework, we show in Table 4 the performance, computational load, and temporal receptive field size of "conv1" in each branch. The experiment results are based on UCF-101. "branch 1" is with a small temporal size $\tau' = 16$, while "branch 2" uses a large temporal size $\tau = 64$. "branch 1$*$" has a same model configuration as "branch 1" except for a 64-frame input. The only difference between "branch 1" and "branch 2" is the shallow layer (i.e. "conv1"), where the latter adopts temporal dilation convolution to achieve a larger temporal receptive field. From Table 4, we can observe that "branch 2" has a larger temporal receptive field due to temporal dilation convolution, and it has the same model parameters as "branch1" which has a small temporal size $\tau' = 16$. Note that, when temporal size is 64 frames, the computational load of "branch 2" is about a quarter of that of "branch 1$*$". Although "branch 2" gains a larger temporal receptive field, temporal dilation convolution decreases the resolution in temporal dimension and makes the model difficult to optimize, which results in performance degradation. To address this issue, we fuse the results of "branch 1" and "branch 2" by using a weighted average. Note that, the computational load of "branch (1+2)" is about half that of "branch 1$*$", while "branch (1+2)" acquires a comparable performance compared with "branch 1$*$".

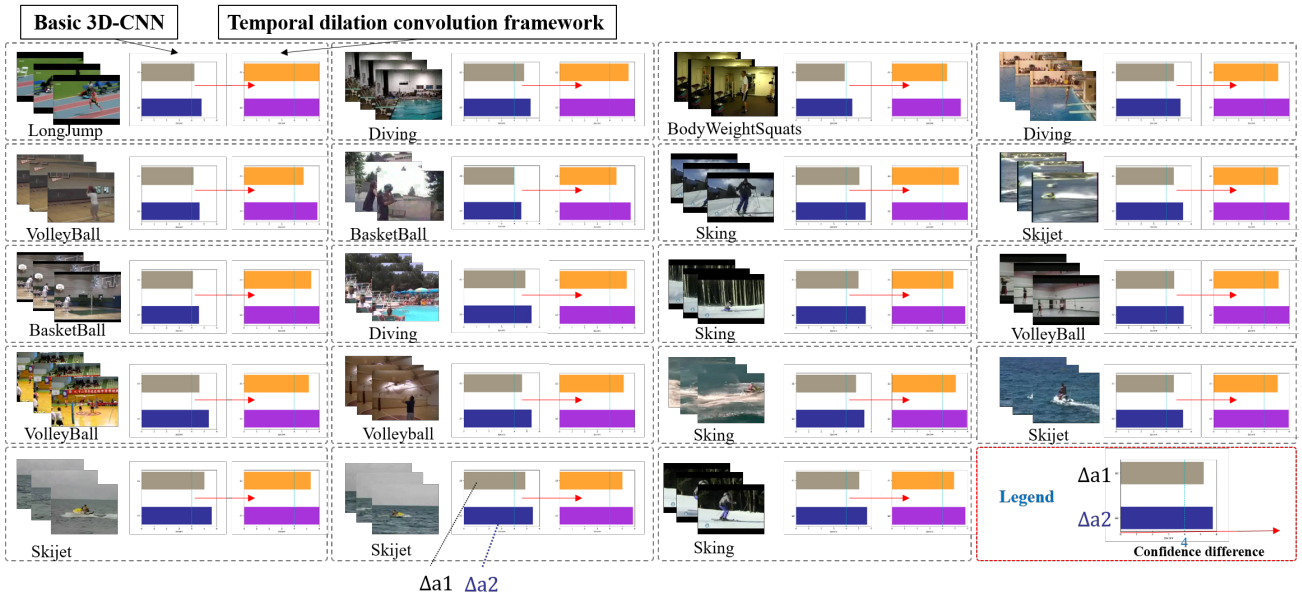

Figure 8: Recognition confidence difference of samples from UCF-101. In each block, the middle panel is the recognition confidence difference of basic 3D CNN, and the right-hand panel is that of the proposed 3D-TDC framework. $\Delta a1$ is the difference between the highest confidence and the sub-high confidence; $\Delta a2$ is the difference between the highest confidence and the third highest confidence.

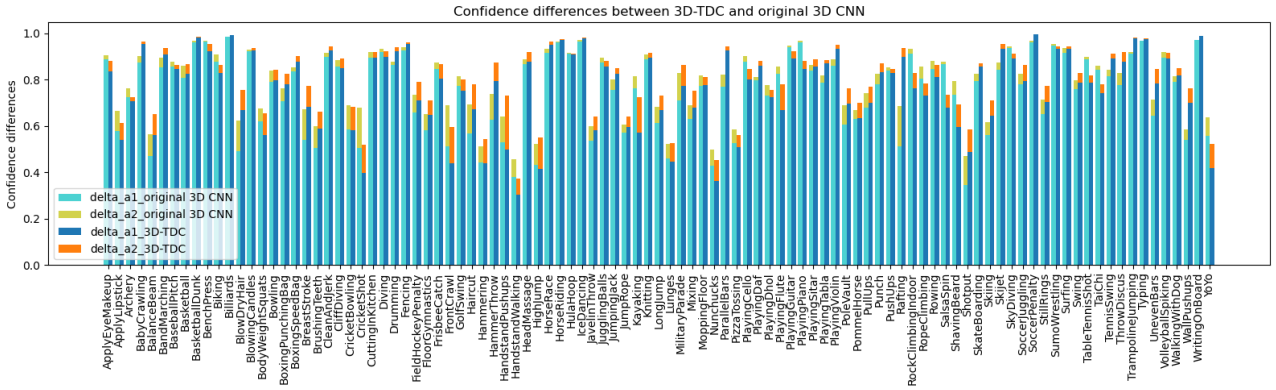

Figure 9: Confidence differences between original 3D CNN and 3D-TDC for all categories in UCF-101.

4.3. Comparison with state-of-the-art methods

For evaluating the accuracy of our proposed 3D-TDC framework more comprehensively, we conduct experiments on UCF-101, HMDB-51, and Kinetics-400, and we compare the accuracy of 3D-TDC with other action recognition methods. Note that the listed results of other methods are taken from their original papers. From Table 5, we can observe that the 3D-TDC (101) framework with only RGB input has the accuracy advantage and is closer to the methods that integrate optical flow as additional input. As can be observed from Table 6, when using R(2+1)D [8] as the backbone to construct our framework, we can improve the recognition accuracy to 93.82% on UCF-101 and 66.83% on HMDB-51, and the performance of our 3D-TDC (R(2+1)D) is also closer to that of the optical flow method, such as Mars [46]. Compared with other optical flow methods, such as Two-stream CNN [53] and Two-stream I3D [53], our 3D-TDC (R(2+1)D) achieves better recognition accuracy, and the recognition accuracy of our 3D-TDC (101) is better than that of the Two-stream CNN [53]. The FLOPs of 3D-TDC (R(2+1)D) are far below the FLOPs of Two-stream I3D [53] that adopts RGB and optical flow as input. As for 3D CNN methods, the recognition accuracy of our 3D-TDC (101) is better than C3D [5], P3D [6], T3D [54], I3D [53] and 3D-ResNeXt-101 [9]. Furthermore, the FLOPs of our 3D-TDC (101) are lower than C3D [5], P3D [6], I3D [53], R(2+1)D [8] and 3D-ResNeXt-101 [9]. As for dilation convolution methods, such as DDN [26], our 3D-TDC (101) and 3D-TDC (R(2+1)D) increase the accuracy by 2.16% and 4.13% on UFC-101, respectively. Compared with TSM [33], our 3D-

Table 4
The accuracy (%), computational load (FLOPs), and temporal receptive field size of "conv1" in each branch are based on UCF-101. "R" denotes temporal receptive field size. "branch (1+2)" represents the proposed 3D-TDC framework. The temporal size of "branch 1∗" is 64 frames.

	branch 1	branch 1∗	branch 2	branch (1+2)
R	7	7	13	-
FLOPs	9.61G	38.40G	9.61G	19.22G
Accuracy	90.09	94.10	84.11	91.85

TDC achieves the comparable recognition accuracy, and the FLOPs of 3D-TDC (101) are lower than TSM [33]. All of these verify the effectiveness and general applicability of our proposed 3D-TDC framework in achieving comparable recognition accuracy with less computational consumption.

Table 5
Accuracy (%) on Kinetics-400 compared with the state-of-the-art methods.

Input	Method	accuracy
RGB+Flow	Two-stream CNN [53]	61.0
	Two-stream I3D [53]	**71.6**
	Mars [46]	68.9
	3D-ResNext-101 [46]	69.1
RGB	CNN+LSTM [53]	57.0
	3D-ResNeXt-101 [9]	65.1
	3D-TDC (101)	67.5

5. Conclusion

We propose a new action-recognition framework based on 3D temporal dilation convolution. We introduce the 3D temporal dilation convolution structure into the shallow layer, to enlarge the temporal receptive field of the whole network and compress the large temporal information. Then we build the 3D temporal dilation convolution framework for action recognition. Through extensive experiments and analysis on the various benchmark datasets, the performance advantages of our framework are verified. In the future, we will further explore the deployment of the 3D temporal dilation structure over the whole network and investigate a temporal feature extraction network suitable for a longer time domain.

Acknowledgement

This work was partly supported by partly supported by Natural Science Foundation of China (Grant No. 62076030), Beijing Natural Science Foundation of China (Grant No. L201023 and L182033) and the Fundamental Research Funds for the Central Universities (2019PTB-001).

Table 6
Accuracy (%) and computation load (FLOPs) on UCF-101 and HMDB-51 compared with the state-of-the-art methods.

Input	Method	FLOPs	UCF-101/HMDB-51
RGB+Flow	Two-stream CNN [53]	-	87.76/58.00
	Two-stream I3D [53]	213.85G	93.40/66.40
	Mars [46]	18.13G	**95.80/75.00**
RGB	C3D [5]	38.57G	82.30/-
	P3D [6]	18.51G	88.60/-
	T3D [54]	-	90.30/59.20
	I3D [53]	111.33G	84.50/49.80
	R(2+1)D [8]	41.69G	93.60/66.60
	3D-ResNeXt-101 [9]	38.40G	90.09/63.20
	TSM [33]	32.88G	95.50/73.60
	DDN [26]	-	89.69/74.51
	TRN [55]	23.50G	83.83/-
	3D-TDC (R(2+1)D)	84.39G	93.82/66.83
	3D-TDC (101)	19.22G	91.85/64.12

References

[1] J. Cai, J. hu, X. Tang, H. Tzu-Yi, Y.-P. Tan, Deep historical long short-term memory network for action recognition, Neurocomputing 407 (24) (2020) 428–438.
[2] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Advances in Neural Information Processing Systems, 2014, pp. 568–576.
[3] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal segment networks: Towards good practices for deep action recognition, in: European Conference on Computer Vision, Springer, 2016, pp. 20–36.
[4] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal segment networks for action recognition in videos, IEEE Transactions on Pattern Analysis and Machine Intelligence 41 (11) (2018) 2740–2755.
[5] S. Ji, W. Xu, M. Yang, K. Yu, 3D convolutional neural networks for human action recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (1) (2012) 221–231.
[6] Z. Qiu, T. Yao, T. Mei, Learning spatio-temporal representation with pseudo-3D residual networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5533–5541.
[7] D. Tran, J. Ray, Z. Shou, S.-F. Chang, M. Paluri, ConvNet architecture search for spatiotemporal feature learning, arXiv preprint arXiv:1708.05038.
[8] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, M. 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.
[9] K. Hara, H. Kataoka, Y. 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.
[10] C. Li, Q. Zhong, D. Xie, S. 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.

[11] X. Qin, Y. Ge, J. Feng, D. Yang, F. Chen, S. Huang, L. Xu, DTMMN: Deep transfer multi-metric network for rgb-d action recognition, Neurocomputing 406 (17) (2020) 127–134.

[12] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, T. Brox, FlowNet: Learning optical flow with convolutional networks, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 2758–2766.

[13] Y. Zhu, Z. Lan, S. Newsam, A. Hauptmann, Hidden two-stream convolutional networks for action recognition, in: Asian Conference on Computer Vision, Springer, 2018, pp. 363–378.

[14] L. Fan, W. Huang, C. Gan, S. Ermon, B. Gong, J. Huang, End-to-end learning of motion representation for video understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6016–6025.

[15] Z. Shou, X. Lin, Y. Kalantidis, L. Sevilla-Lara, M. Rohrbach, S.-F. Chang, Z. Yan, DMC-Net: Generating discriminative motion cues for fast compressed video action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1268–1277.

[16] B. Zhang, L. Wang, Z. Wang, Y. Qiao, H. Wang, Real-time action recognition with deeply transferred motion vector CNNs, IEEE Transactions on Image Processing 27 (5) (2018) 2326–2339.

[17] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1933–1941.

[18] M. S. Ryoo, A. Piergiovanni, M. Tan, A. Angelova, AssembleNet: Searching for multi-stream neural connectivity in video architectures, arXiv preprint arXiv:1905.13209.

[19] J. Lei, Y. Jia, B. Peng, Q. Huang, Channel-wise temporal attention network for video action recognition, in: IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2019, pp. 562–567.

[20] C. Yang, Y. Xu, J. Shi, B. Dai, B. Zhou, Temporal pyramid network for action recognition, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 591–600.

[21] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 2625–2634.

[22] W. Du, Y. Wang, Y. Qiao, Recurrent spatial-temporal attention network for action recognition in videos, IEEE Transactions on Image Processing 27 (3) (2017) 1347–1360.

[23] Z. Tu, H. Li, D. Zhang, J. Dauwels, B. Li, J. Yuan, Action-stage emphasized spatiotemporal vlad for video action recognition, IEEE Transactions on Image Processing 28 (6) (2019) 2799–2812.

[24] W. Huang, L. Fan, M. Harandi, L. Ma, H. Liu, W. Liu, C. Gan, Toward efficient action recognition: Principal backpropagation for training two-stream networks, IEEE Transactions on Image Processing 28 (4) (2018) 1773–1782.

[25] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, International Conference on Learning Representations.

[26] B. Xu, H. Ye, Y. Zheng, H. Wang, T. Luwang, Y.-G. Jiang, Dense dilated network for video action recognition, IEEE Transactions on Image Processing 28 (10) (2019) 4941–4953.

[27] X. Dai, B. Singh, J. Y.-H. Ng, L. Davis, Tan: Temporal aggregation network for dense multi-label action recognition, in: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2019, pp. 151–160.

[28] X. Wang, R. Girshick, A. Gupta, K. He, Non-local neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7794–7803.

[29] X. Long, C. Gan, G. De Melo, J. Wu, X. Liu, S. Wen, Attention clusters: Purely attention based local feature integration for video classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7834–7843.

[30] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, G. D. Hager, Temporal convolutional networks for action segmentation and detection, in: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 156–165.

[31] Y. A. Farha, J. Gall, Ms-tcn: Multi-stage temporal convolutional network for action segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3575–3584.

[32] R. Dai, S. Das, L. Minciullo, L. Garattoni, G. Francesca, F. Bremond, Pdan: Pyramid dilated attention network for action detection, in: WACV 2021-Winter Conference on Applications of Computer Vision 2021.

[33] J. Lin, C. Gan, S. Han, Tsm: Temporal shift module for efficient video understanding, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 7083–7093.

[34] X. Song, C. Lan, W. Zeng, J. Xing, X. Sun, J. Yang, Temporal–spatial mapping for action recognition, IEEE Transactions on Circuits and Systems for Video Technology 30 (3) (2020) 748–759.

[35] J. Materzynska, T. Xiao, R. Herzig, H. Xu, X. Wang, T. Darrell, Something-else: Compositional action recognition with spatial-temporal interaction networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1049–1059.

[36] C. Luo, A. L. Yuille, Grouped spatial-temporal aggregation for efficient action recognition, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 5512–5521.

[37] A. Diba, M. Fayyaz, V. Sharma, M. Mahdi Arzani, R. Yousefzadeh, J. Gall, L. Van Gool, Spatio-temporal channel correlation networks for action classification, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 284–299.

[38] X. Zhu, C. Xu, L. Hui, C. Lu, D. Tao, Approximated bilinear modules for temporal modeling, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 3494–3503.

[39] J. Wang, A. Cherian, F. Porikli, S. Gould, Video representation learning using discriminative pooling, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1149–1158.

[40] S. Xie, C. Sun, J. Huang, Z. Tu, K. 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.

[41] Y. Chen, Y. Kalantidis, J. Li, S. Yan, J. Feng, Multi-fiber networks for video recognition, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 352–367.

[42] B. Jiang, M. Wang, W. Gan, W. Wu, J. Yan, STM: Spatiotemporal and motion encoding for action recognition, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 2000–2009.

[43] J. Zhang, F. Shen, X. Xu, H. T. Shen, Temporal reasoning graph for activity recognition, IEEE Transactions on Image Processing 29 (2) (2020) 5491–5506.

[44] X. Wang, A. Gupta, Videos as space-time region graphs, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 399–417.

[45] H. Yang, C. Yuan, L. Zhang, Y. Sun, W. Hu, S. J. Maybank, STA-CNN: Convolutional spatial-temporal attention learning for action recognition, IEEE Transactions on Image Processing 29 (3) (2020) 5783–5793.

[46] N. Crasto, P. Weinzaepfel, K. Alahari, C. Schmid, MARS: Motion-augmented RGB stream for action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7882–7891.

[47] J. Kim, S. Cha, D. Wee, S. Bae, J. Kim, Regularization on spatio-temporally smoothed feature for action recognition, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 12103–12112.

[48] C. Feichtenhofer, H. Fan, J. Malik, K. He, Slowfast networks for video recognition, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 6202–6211.

[49] S. Sudhakaran, S. Escalera, O. Lanz, Gate-shift networks for video action recognition, in: IEEE/CVF Conference on Computer Vision

and Pattern Recognition (CVPR), 2020, pp. 1102–1111.

[50] K. Soomro, A. R. Zamir, M. Shah, UCF101: A dataset of 101 human actions classes from videos in the wild, arXiv preprint arXiv:1212.0402.

[51] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, HMDB: a large video database for human motion recognition, in: International Conference on Computer Vision, IEEE, 2011, pp. 2556–2563.

[52] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, A. Zisserman, The kinetics human action video dataset, arXiv preprint arXiv:1705.06950.

[53] J. Carreira, A. 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.

[54] A. Diba, M. Fayyaz, V. Sharma, A. H. Karami, M. M. Arzani, R. Yousefzadeh, L. Van Gool, Temporal 3D ConvNets: New architecture and transfer learning for video classification, arXiv preprint arXiv:1711.08200.

[55] B. Zhou, A. Andonian, A. Oliva, A. Torralba, Temporal relational reasoning in videos, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 803–818.