ITC 4/52
Information Technology and Control
Vol. 52 / No. 4 / 2023
pp. 952-965
DOI 10.5755/j01.itc.52.4.33239

CSST-Net: Channel Split Spatiotemporal Network for Human Action Recognition

Received 2023/01/18 | Accepted after revision 2023/10/18

HOW TO CITE: Zhou, X., Ma, J., Yi, J. (2023). CSST-Net: Channel Split Spatiotemporal Network for Human Action Recognition. *Information Technology and Control*, 52(4), 952-965. https://doi.org/10.5755/j01.itc.52.4.33239

CSST-Net: Channel Split Spatiotemporal Network for Human Action Recognition

Xuan Zhou, Jixiang Ma
School of Mechanical and Electrical Engineering, Xi'an Traffic Engineering Institute, Xi'an, 710300, China; e-mail: 1138845898@qq.com

Jianping Yi
School of Electronics and Information, Xi'an Polytechnic University, Xi'an, 710048, China; e-mail: 942749578@qq.com

Corresponding author: 1138845898@qq.com

Temporal reasoning is crucial for action recognition tasks. The previous works use 3D CNNs to jointly capture spatiotemporal information, but it causes a lot of computational costs as well. To improve the above problems, we propose a general channel split spatiotemporal network (CSST-Net) to achieve effective spatiotemporal feature representation learning. The CSST module consists of the grouped spatiotemporal modeling (GSTM) module and the parameter-free feature fusion (PFFF) module. The GSTM module decomposes features into spatial and temporal parts along the channel dimension in parallel, which focuses on spatial and temporal clues, respectively. Meanwhile, we utilize the combination of group-wise convolution and point-wise convolution to reduce the number of parameters in the temporal branch, thus alleviating the overfitting of 3D CNNs. Furthermore, for the problem of spatiotemporal feature fusion, the PFFF module performs the recalibration and fusion of spatial and temporal features by a soft attention mechanism, without introducing extra parameters, thus ensuring the correct network information flow effectively. Finally, extensive experiments on three benchmark databases (Sth-Sth V1, Sth-Sth V2, and Jester) indicate that the proposed CSST-Net can achieve competitive performance compared to existing methods, and significantly reduces the number of parameters and FLOPs of 3D CNNs baseline.

KEYWORDS: Temporal reasoning, Action recognition, Spatiotemporal representation learning, Spatiotemporal fusion.

1. Introduction

The objective of action recognition is to predict the class of action on the pre-segmented temporal sequences. As one of the hottest topics of video understanding, it has received extensive attention from academia and industry due to its widespread application scenarios in motion analysis, video surveillance, human-computer interaction, intelligent information retrieval, and so on.

Existing action video datasets are usually derived from real application scenarios. However, the method based on traditional hand-craft features [30] has poor generalization ability. The development of convolutional neural networks (CNNs), has shown strong generalization performance of image classification [7], [10], which has greatly inspired the research on CNNs-based action recognition. In addition, action video datasets are usually having complex motion patterns, so how to effectively capture the temporal information in the video is the focus of action recognition. As shown in Figure 1, the Something-Something V1 dataset [5] contains some fine-grained actions such as "pushing something from left to right" and "pushing something from right to left", using a single frame cannot effectively distinguish two actions. On the contrary, temporal reasoning is critical when processing temporal-related datasets.

To model temporal information effectively, some works [22], [37] via employing the temporal module on the top of the 2D CNNs for later temporal fusion. However, it generally focuses on a coarser and long-term temporal structure, but cannot represents finer temporal relations in a local window. Thus, some works [24], [32] try to leverage optical flow to encode the motion information between adjacent frames. In practice, calculating optical flow is an expensive and time-consuming task, which hinders the application of optical flow-based methods in the real world. Other works [1], [14], [29] have expanded existing 2D CNNs to 3D CNNs via 3D convolution and 3D pooling and extracted spatial-temporal features from RGB volume. Although 3D CNNs have achieved competitive performance, it has introduced a large number of computational costs as well, which caused the performance degradation of the CNNs. Although some works [3], [23], [25], [36] factorize the 3D convolution kernel into spatial and temporal parts to reduce computational overhead, it is still unknown whether the simple decomposition of the 3D convolution kernel can capture the spatiotemporal information effectively.

To tackle the limitation of existing methods, we propose a CNNs architecture, dubbed CSST-Net, for efficient video learning. The framework of the proposed method is shown in Figure 2, Specifically, CSST-Net consists of the stacked CSST block, which includes GSTM and PFFF modules. Different from previous works where the groups are symmetric, the GSTM

Figure 1
Sample of Fine-grained actions on Sth-Sth datasets

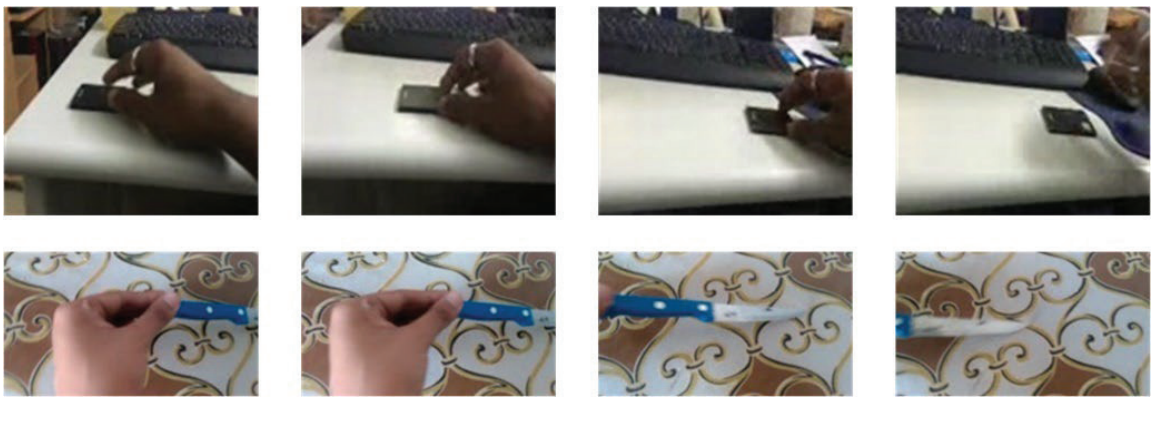
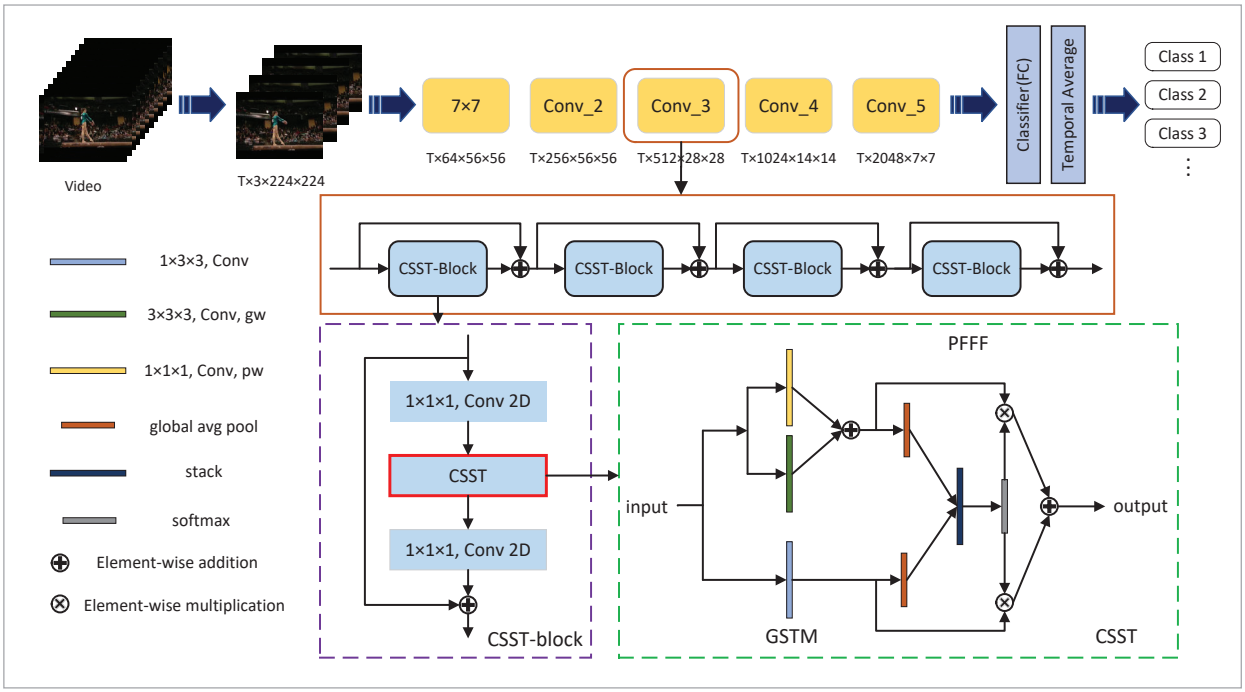


Figure 2
The framework of channel split spatiotemporal networks



module decomposes the input feature maps into the spatial branch and temporal branch along the channel dimensions in an asymmetric manner and performs different operations in parallel, to extract spatial and temporal features, respectively. Furthermore, inspired by MobileNet [26], we employ a parallel combination of 3D group-wise convolution and 3D pointwise convolution in the temporal branch to improve the efficiency of parameter learning and avoid damage to channel interaction information. Compared with vanilla 3D convolution, our model is more compact and facilitates the network to exploit features.

The PFFF module is a simplified version of the SKNet [16] for the integration of spatial and temporal features. In general, the PFFF module is based on the soft-attention, which allocates significance weights to the channels of the two branches, and then merges them via weighted summation to ensure the correct network information flow. In this manner, each channel integrates the spatial and temporal features of the previous layer. When the features are fed to the next layer, the PFFF is beneficial for multi-scale temporal modeling. Notably, the proposed module is efficient since it without introducing any extra parameters and can improve the performance of the network. We evaluated the proposed model on three benchmark databases, the evaluation results show that our CSST can yield a new state-of-the-art performance on three temporal relevant datasets such as Something-Something V1 (Sth-Sth V1), Something-Something V2 (Sth-Sth V2), and Jester [21].

We summarize our main contributions as follows:

1 To reduce the FLOPs and feature redundancy of the existing 3D CNNs-based models, we propose a novel grouped spatiotemporal feature extraction module. Through the channel split of intermediate feature maps, the spatial and temporal information was obtained effectively, which improved the efficiency of feature extraction.

2 To integrate spatiotemporal features and model multi-scale temporal information, we propose a parameter-free feature fusion module based on the soft-attention mechanism. The experimental results show that the module improves the network performance and ensures the correct flow of information without introducing any additional parameters.

3 Embedding CSST block into 2D ResNet50 only adds a limited extra computational cost, and achieves competitive performance on the three benchmark datasets for action recognition. Meanwhile, extensive ablation experiments demonstrate that the proposed CSST-Net is superiority over previous work methods.

2. Related Works

Two stream-based. Two-stream network including spatial and temporal streams, which extracts appearance and motion features from single RGB video frames and stacked optical flow images, respectively. Both streams make independent predictions based on their inputs, and then the prediction scores of the two streams are later fused. Because the optical flow only encodes the motion information between two adjacent frames, which limits the access of two-stream to the temporal context. Some works proposed a few novel feature coding methods such as spatiotemporal pyramid [38] and temporal linear coding [2], which encodes frame-level features into video-level representations, trying to solve the problem that two-stream networks have a weak ability for long-term temporal modeling. On the other hand, the calculation of optical flow is time-consuming, which brings challenges to real-world applications. Moreover, the feature learning of spatial and temporal streams is completely independent. Some works [28], [40] try to jointly optimize optical flow estimation and classification network or to implicitly simulate the optical flow of the RGB network.

3D CNNs-based. 3D CNNs via extending 2D convolution to the temporal domain to jointly extract spatiotemporal features from RGB volume. Most of the 3D CNNs for action recognition task is based on the expansion of the 2D CNNs architecture such as I3D [1], ARTNet [32], etc. Some works decomposed the 3D kernel into the 2D kernel and 1D kernel in spatial and temporal, respectively, to reduce parameters of 3D CNNs such as P3D [23], R(2+1)D [3], S3D [36], etc. Wang et al. [31] proposed a learnable correlation operator to capture the temporal relations between adjacent frames for explicitly encoding short-term temporal information. Zolfaghari [39] proposed to use 2D CNNs in the top layers and 3D CNNs in the bottom layers to trade-off accuracy and inference speed. Feichtenhofer [4] proposed a novel network design idea, which progressively expands tiny 2D CNNs into 3D CNNs for video understanding along with space, time, width, and depth, to explore the influence of various dimensions on video learning.

2D CNNs-based. TSM [14] is utilized to implicitly capture the temporal information via shifting the channel along the temporal dimension, but this local way significantly lacks explicit modeling of motion information. To model motion information effectively, TEA [18] and TEINet [19] proposed a motion excitation block, which calculates the feature-level difference between two adjacent segments, and then performs channel attention to activate motion-sensitive features. Kwon et al. [11] proposed a motion squeeze block to establish temporal relations across frames and convert them into motion features. Liu et al. [20] proposed a temporal adaptive module to generate a video-specific kernel, and use local and global branches to learn short-term and long-term temporal structures, respectively. Jiang et al. [9] proposed two modules for modeling spatiotemporal and motion features, respectively, to replace the original ResNet blocks, and confirmed the complementarity of spatiotemporal and motion information. Hussein [8] proposed multi-scale temporal convolution to learn the long-range temporal information of different receptive fields in a single layer, which is beneficial to modeling complex motion patterns.

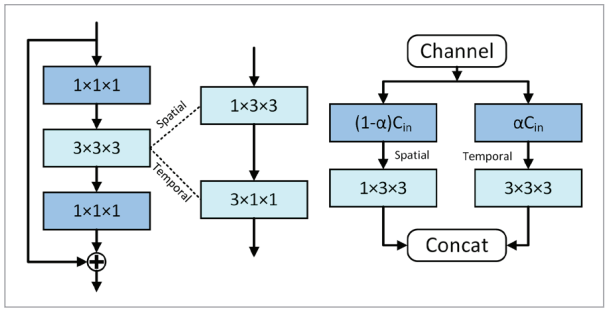
3. Research Methodology

In this section, we depict the CSST-Net in detail. We first explain the motivation of the proposed grouped spatiotemporal feature learning. Then, we give a technical description of the CSST block, and discuss the computational costs of CSST. Finally, we provide the implementation detail to instantiate CSST-Net with a ResNet50 backbone.

3.1. Grouped Spatiotemporal Feature Learning

For video understanding tasks, the input tensor $\mathbf{X}$ is represented as $[N, C_i, T, H, W]$, where N is the batch size. C_i is the channel, T, H, W denote temporal and spatial dimensions, respectively. Suppose the output tensor is $[N, C_o, T, H, W]$, the parameters of vanilla

Figure 3
The decomposition strategy for reducing the number of parameters of 3D CNNs. (a) Vanilla 3D convolution, (b) Cascade decomposition method, (c) Our proposed decomposition method



3D convolution can be calculated as $C_o \times C_i \times k \times k \times k$, it is about k times of the 2D kernel. Considering that the channel of existing CNNs is usually large, this will significantly increase the computational overhead of 3D CNNs.

As shown in Figure 3(b), to reduce the number of parameters of 3D CNNs, a common method is to factorize the 3D kernel into the 2D kernel and 1D kernel in spatial and temporal, respectively, and then the spatiotemporal features are modeled via a cascading way. This method can effectively reduce the overfitting of 3D CNNs by decoupling the spatiotemporal learning. The whole process can be illustrated as the Equation (1).

$$\mathbf{Y} = \mathbf{W_t} * (\mathbf{W_s} * \mathbf{X}), \tag{1}$$

where * denote convolution operation, $\mathbf{W_s} \in \mathbb{R}^{C_i \times C_o \times 1 \times k \times k}$, $\mathbf{W_t} \in \mathbb{R}^{C_i \times C_o \times k \times 1 \times 1}$ denote shapes of spatial and temporal filters, respectively.

3.2. Grouped Spatiotemporal Modeling Module

As shown in Figure 3(c), different from the cascade decomposition above, we propose to split the input feature maps into spatial and temporal parts along channel dimensions. The motivation for this decomposition is that in the channel, some are more relevant to static appearance clues, while others are more focused on dynamic motion clues. Thus, we perform 2D convolution and 3D convolution on spatial and temporal branches, respectively, implicitly encoding both appearance and motion information. This decomposition method can promote the two branches to learn complementary features, so can improve the efficiency of feature learning. Furthermore, a channel scaling factor α is utilized to quantize the ratio of the temporal branch to control the capacity of the model. The decomposition course is shown in Equation (2).

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{C'} \end{bmatrix} = \begin{bmatrix} W_{11} & \cdots & W_{1,\alpha C_i} \\ \vdots & \ddots & \vdots \\ W_{C_o,1} & \cdots & W_{C_o,\alpha C_i} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{\alpha C_i} \end{bmatrix} + \begin{bmatrix} w_{1,\alpha C_i+1} & \cdots & w_{1,C_i} \\ \vdots & \ddots & \vdots \\ w_{C_o,\alpha C_i+1} & \cdots & w_{C_o,C_i} \end{bmatrix} \begin{bmatrix} x_{\alpha C_i+1} \\ \vdots \\ x_{C_i} \end{bmatrix}, \tag{2}$$

where $W \in \mathbb{R}^{\alpha C_i \times C_o \times k \times k \times k}$ and $W \in \mathbb{R}^{(1-\alpha) C_i \times C_o \times 1 \times k \times k}$ denote the shape of the convolution kernel of temporal and spatial branches, respectively.

In general, the proposed method is different from the conventional kernel decomposition strategy in three aspects. (1) The proposed method is for channel decomposition, rather than the 3D kernel, (2) The processing methods of decomposed features: one is parallel, and the other is cascaded. (3) Our method flexibly controls model parameters and computational cost by changing α.

Meanwhile, inspired by Luo et al. [12], properly dropping the channel ratio of the temporal branch does not hurt performance significantly. We guess this phenomenon may be caused by feature redundancy, and local temporal relations may be related to sparse 3D kernels. Therefore, we further designed the temporal branch and proposed the GSTM module.

Specially, we utilize group-wise convolution in the temporal branch to reduce feature redundancy and improve the efficiency of parameter learning. Furthermore, we add point-wise convolution across all temporal channels to achieve cross-group information exchange among different groups feature maps. Group-wise convolution and point-wise convolution conduct operations on all channels of the temporal branch simultaneously, aiming at efficient temporal feature learning. The experimental results show that this asymmetric spatiotemporal decomposition can effectively allocate the utilization space of parameters, to achieve better performance. Finally, we merge the outputs via element-wise addition since group-wise convolution and point-wise convolution are the same channel origin. So the temporal branch of Equation (2) can be formulated as Equation (3).

$$\begin{bmatrix} W_{11}^g & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & W_{GG}^g \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ z_G \end{bmatrix} + \begin{bmatrix} W_{11}^p & \cdots & W_{1,\alpha C_i}^p \\ \vdots & \ddots & \vdots \\ W_{C_o,1}^p & \cdots & W_{C_o,\alpha C_i}^p \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{\alpha C_i} \end{bmatrix}. \tag{3}$$

Here, we divide the temporal branch αC_i into G groups and each group z_i contains $\alpha C_i/G$ channels. W_{ii}^g are parameters of group convolution kernel in i-th group.

3.3. Parameter Free Feature Fusion Module

So far, we have split the channel into two parts. For the temporal branch, we perform element-wise addition of 3×3×3 group-wise convolution and 1×1×1 point-wise convolution to extract temporal features and alleviate the feature redundancy. For the spatial branch, the 3×3 convolution is utilized to extract the spatial features. Because the input of the two branches comes from different channels, a fusion method is needed to control the information flow in the network. In this paper, we proposed a novel PFFF module, without extra parameters that can help improve performance.

In Figure 4, the gw represents group-wise convolution, the pw represents point-wise convolution. $\oplus$ denote element-wise addition, and $\otimes$ denote element-wise multiplication. As shown in Figure 4, the global spatiotemporal information of the output $\boldsymbol{U}$ is aggregated by 3D global average pooling to generate two channel-wise statistical vectors $\boldsymbol{S}_{1,2} \in \mathbb{R}^c$ for global information embedding. The c-th element of $\boldsymbol{S}$ can be calculated as Equation (4).

$$S_{lc}=F_{GAP}(U_{lc})=\frac{1}{T\times H\times W}\sum_{k=1}^{T}\sum_{i=1}^{H}\sum_{j=1}^{W}U_{lc}(i,j,k), l=[1,2]. \quad (4)$$

Secondly, vectors are stacked together and soft attention operations are conducted across channels to generate channel weight vectors $\boldsymbol{\beta} \in \mathbb{R}^c$ and $\boldsymbol{\gamma} \in \mathbb{R}^c$. The c-th element of β, γ can be calculated as Equation (5).

$$\beta_C=\frac{e^{S_{2C}}}{e^{S_{2C}}+e^{S_{1C}}}, \gamma_C=1-\beta_C. \quad (5)$$

Finally, the features $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ from spatial and temporal branches are channel-wise multiplied with weight vectors β and γ, respectively. Output $\boldsymbol{Y}$ is obtained by merging the activation of spatial and temporal features.

$$\boldsymbol{Y}=\boldsymbol{\beta}\boldsymbol{U}_1+\boldsymbol{\gamma}\boldsymbol{U}_2, \quad (6)$$

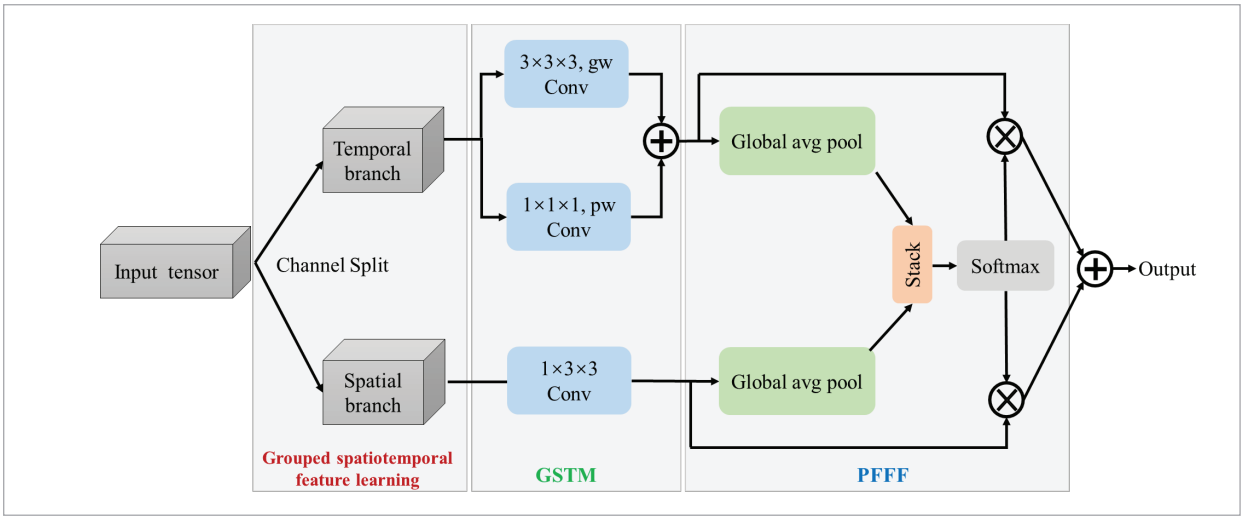
To sum up, $\boldsymbol{Y}$ can be formulated as Equation (7).

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{C_o} \end{bmatrix} = \boldsymbol{\beta}\begin{bmatrix} W_{11}^g & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & W_{GG}^g \end{bmatrix}\begin{bmatrix} z_1 \\ \vdots \\ z_G \end{bmatrix} + \boldsymbol{\beta}\begin{bmatrix} W_{11}^p & \cdots & W_{1,\alpha C_i}^p \\ \vdots & \ddots & \vdots \\ W_{C_o,1}^p & \cdots & W_{C_o,\alpha C_i}^p \end{bmatrix}\begin{bmatrix} x_1 \\ \vdots \\ x_{\alpha C_i} \end{bmatrix} + \boldsymbol{\gamma}\begin{bmatrix} w_{1,\alpha C_i+1} & \cdots & w_{1,C} \\ \vdots & \ddots & \vdots \\ w_{C_o,\alpha C_i+1} & \cdots & w_{C_o,C_i} \end{bmatrix}\begin{bmatrix} x_{\alpha C_i+1} \\ \vdots \\ x_{C_i} \end{bmatrix}, \quad (7)$$

3.4. Computational Costs Analysis

Since the parameters of the CSST block are concentrated in the temporal part. Hence, we employ α and g to control the complexity of CSST. Where, α denote

Figure 4
The overall architecture of the CSST block



the ratio of the temporal channels to the input channel, and g denote the number of groups of temporal convolutions. Furthermore, the number of output channels of the spatial branch and temporal branch are consistent, i.e, $C_{os} = C_{ot} = C_o$. The spatial branch performs the 2D convolution ($1\times k\times k$) with the number of parameters is $1\times k\times k\times(1-\alpha)C_i\times C_o$. The temporal branch performs the 3D group-wise convolution ($k\times k\times k$) and 3D point-wise convolution ($1\times1\times1$) in parallel, and the number of parameters is $k\times k\times k\times(\alpha C_i/g)\times(C_o/g)\times g+1\times1\times1\times\alpha C_i\times C_o$. The comparison of the number of parameters for CSST with four representative spatiotemporal learning blocks is shown in Table 1.

Table 1
Comparison of the number of parameters for spatial-temporal learning block

Model	#Params
C2D	$k\times k\times C_i\times C_o$
C3D	$k\times k\times k\times C_i\times C_o$
P3D	$(k\times k+k)\times C_i\times C_o$
C3Dg	$k\times k\times k\times C_i\times C_o/g$
CSST	$(k\times k\times(1-\alpha)+k\times k\times k\times\alpha/g+\alpha)\times C_i\times C_o$

When $\alpha = 0.5$ and $g = 2$, compare with the vanilla 3D convolution, the number of parameters of the proposed CSST can be reduced by 2.30x, and slightly smaller than the P3D model (P3D is reduced by 2.25x). When $\alpha = 0.5$ and $g = 4$, our CSST has an even fewer number of parameters than 2D Convolution. However, CSST has sufficient temporal convolution and excellent temporal reasoning ability.

3.5. Network Architecture

As discussed above, the proposed CSST-Net is based on the sparse temporal sampling of the TSN baseline to dispose of the video with variable temporal length. Especially, the video is divided into T segments according to the uniform duration, and then a frame is randomly sampled from each segment to obtain the input sequence of T frames. As shown in Figure 2, we replace 3×3×3 convolution with the CSST from $Conv_2$ to $Conv_5$ since the spatiotemporal features are concentrated in the middle block. Finally, a temporal average pooling is utilized to average the prediction of all segments. Thus, the temporal interactions in the middle layers only come from CSST, which can better reflect the temporal modeling ability of CSST.

4. Experiments

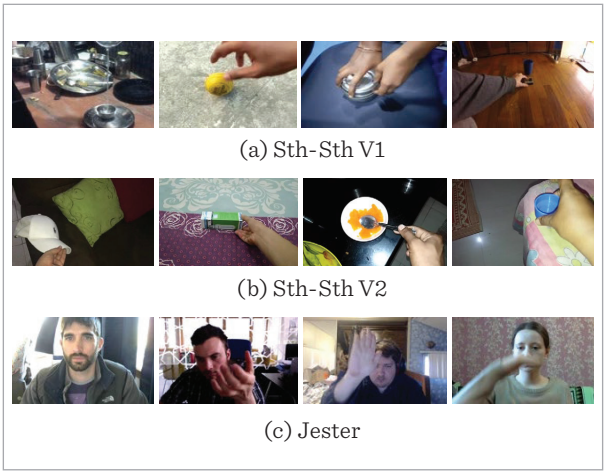
In this section, we first describe the benchmark databases and implementation details. Then, we perform ablation studies to measure the effectiveness of the proposed CSST-Net and to investigate its optimal setting. Finally, we compare CSST-Net with the previous state-of-the-art (SOTA) methods on three benchmark databases.

4.1. Datasets

Sth-Sth datasets. The Sth-Sth datasets focus on fine-grained actions, which contain the humans performing pre-defined actions with daily objects. Sth-Sth V1 contains 174 classes with 108,499 video clips. Sth-Sth V2 is an updated version of Sth-Sth V1, which a total of 220,847 video clips. The Sth-Sth datasets are to perform the same action with different objects ("something"), thus the model forcing to understand the action itself instead of recognizing the object.

Jester. The Jester dataset is a crowd-acted video dataset for generic human hand gesture recognition. It includes 27 categories with 118,562 training videos. Notably, the data augmentation technique of random flipping cannot be used since some gestures are symmetric with each other, such as "swiping left", "swiping right" and "sliding two fingers down", "sliding two fingers up". Sample actions of benchmark databases are demonstrated in Figure 5.

Figure 5
Sample actions of benchmark databases



(a) Sth-Sth V1

(b) Sth-Sth V2

(c) Jester

4.2. Implementation Details

We implement the model proposed in section 3.5 on PyTorch. We load the pre-trained ResNet50 on ImageNet to initialize the weight of the spatial branch and randomly initialize the parameters of the temporal branch.

Training. We use the SGD with a momentum of 0.9 and a weight decay of 0.0005 to train CSST-Net. The batch size is set to 8, and the input size is 224×224. The learning rate is initialized to 0.01, and the cosine learning rate schedule [13] is applied to ensure that the learning rate can be adjusted in each epoch. We train the network for 70 epochs on Syh-Sth V1, Sth-Sth V2, and 40 epochs on Jester. The first 10 epochs are used for linear warm-up [6]. To mitigate the over-fitting, we adopt dropout after the global pooling layer with a dropout ratio of 0.3, and apply the multi-scale cropping and randomly horizontal flipping as data augmentation.

Test. We use an evaluation protocol similar to TEA [18] to ensure reasoning speed (*center crop×1 clip*). Firstly, 1 clip with T frames is sampled from a video. Then, each frame is resized to 256×256. Finally, the region of the center cropping is limited to 224×224 for action prediction.

4.3. Ablation Studies

In the subsection, we carry out technical research on the hyper-parameters and effectiveness of the network. To perform fair comparisons, all methods follow sparse temporal sampling, sampling 8 frames from each video as input. Inference stage, we report Top-1 & Top-5 and computational efficiency or parameters utilization efficiency to comprehensively evaluate accuracy and efficiency.

First, we evaluate the effects of the depth of the backbone We use ResNet18, ResNet34, ResNet50, and ResNet101 to instantiate CSST-Net, and the α, g is set to 0.5 and 2, respectively. To better capture the relation between boosted performance and add-on computation, we define the computational efficiency formulated as Equation (8).

$$\gamma=\frac{\Delta Top\text{-}1}{\Delta FLOPs}, \tag{8}$$

where both ΔTop-1 and $\Delta FLOPs$ are in percent, γ is the computational efficiency that represents how many Top-1 accuracy in percent are increased to introducing 1% FLOPs (higher indicates more efficient). Specially, we choose CSST-ResNet18 as the baseline, and the evaluation results of CSST-ResNet at different depths are shown in Table 2.

From Table 2, the computational efficiency γ of ResNet34, ResNet50, and ResNet101 are 7.6%, 10.4%, and 4.7%, respectively, so the CSST-ResNet50 is the most computational efficiency when taking γ into account. Therefore, we choose ResNet50 as the backbone network in the following experiments.

Then, we utilize different combinations of the input channel proportion (α) and the group number (g) of the group-wise convolution to figure out the optimal hyper-parameters of the proposed CSST-Net. Although different stages might require different α, it would be too elaborate, so we provide a uniform global α for CSST-Net, and $\alpha\in[0.25, 0.50, 0.75]$. The quantitative comparison results of different settings are listed in Table 3. It can be noticed that our method with $\alpha = 0.5$ and $g = 2$ achieves the highest performance shown in Table 3, which will be applied in the following experiments.

In Table 4, we compare the proposed CSST-Net with four competitive spatiotemporal learning models, such as C2D, C3D, $C3D_g$, and P3D. Specially, all models use the ResNet50 as the backbone, and choose the

Table 2
Ablation study on model depths

Models	Backbone	FLOPs	ΔFLOPs	Top-1(%)	ΔTop-1	γ
CSST	ResNet18	18.81G	-	43.1	-	-
	ResNet34	38.27G	19.46G(+103.5%)	46.5	+3.4(7.9%)	7.6%
	ResNet50	37.69G	18.88G(+100.4%)	47.6	+4.5(10.4%)	10.1%
	ResNet101	72.43G	53.62G(+285.1%)	48.2	+5.1(11.8%)	4.1%

Table 3
Ablation studying on hyper-parameter settings

Models	α,g	FLOPs	Top-1(%)	Top-5(%)
CSST-ResNet50	0.25, 2	35.43G	47.0	76.7
	0.50, 2	37.69G	47.6	77.1
	0.75, 2	39.96G	46.9	76.2
	0.50, 2	37.69G	47.6	77.1
	0.50, 4	32.14G	46.8	76.2
	0.50, 8	29.37G	46.5	76.0

C2D as baseline. To better capture the relation between boosted performance and add-on parameters, we define the parameters utilization efficiency formulated as Equation (9).

$$\eta=\frac{\Delta Top\text{-}1}{\Delta param}, \tag{9}$$

where both ΔTop-1 and $\Delta param$ are in percent, η is the parameters utilization efficiency that represents how many Top-1 accuracy in percent are increased to introducing 1% parameters (higher indicates more efficient).

Table 4
Comparison with counterparts of spatiotemporal learning on Sth-Sth V1

Model	#Param	$\Delta param$	Top-1	ΔTop-1(%)	η
C2D [32]	23.86M	-	20.4%	-	-
C3D [29]	46.50M	$+22.64_{(+94.9\%)}$	46.0%	$+25.6_{(+125.5\%)}$	1.3
P3D [23]	29.40M	$+5.54_{(+23.2\%)}$	45.6%	$+25.2_{(+123.5\%)}$	5.3
$C3D_g$ [3]	29.52M	$+5.66_{(+23.7\%)}$	45.0%	$+24.6_{(+120.6\%)}$	5.1
CSST-Net	27.34M	$+3.48_{(+14.6\%)}$	47.6%	$+27.2_{(+133.3\%)}$	9.1

As can be seen from Table 4, our model yields a superior performance of ~27% higher than C2D since 2D CNNs fail to process the temporal information. When α = 0.5 and g = 2, our CSST significantly performs better than P3D and $C3D_{g=2}$. This indicates that the asymmetric parallel decomposition can better utilize parameters than the cascaded method like P3D. Meanwhile, this shows replacing a set of spatiotemporal convolutions with spatial-only convolution is beneficial. Even compared with C3D, CSST still performs better, because vanilla 3D CNNs have feature redundancy when modeling spatiotemporal features. We split the channel into spatial and temporal parts, and perform 3D group-wise convolution and point-wise convolution on the temporal branch in parallel, which effectively reduces the parameters and feature redundancy of the network, and enhances the generalization ability of the network. In addition, from the perspective of parameter utilization efficiency, the proposed CSST-Net is ~10x higher than C2D, and it is better than other spatiotemporal learning models, which further demonstrates that the proposed CSST-Net can significantly improve the parameter utilization efficiency and has more advantages in temporal reasoning.

4.4. Comparison with Other Methods

Performance analysis. In Table 5, we give a performance comparison of CSST-Net with other methods on three challenging datasets. On Sth-Sth datasets, CSST-Net only samples 8 frames as input, and its performance already outperforms most existing models. Specifically, CSST-Net is superior to later or medium temporal fusion approaches such as TRN, ECO, since CSST-Net via embedding CSST blocks globally, temporal information can be encoded more efficiently. Meanwhile, our model outperforms the 2D CNNs counterparts with small extra computational costs, TSM, TEINet, and SmallBigNet. This is owing to our complementary design of the GSTM module and PFFF module, the PFFF module can recalibrate and merge the spatiotemporal features, so that when the features are fed to the next layer, the GSTM module can perform multi-scale temporal feature learning in a single layer. It is worth noting that our model can achieve almost the same performance as S3D-G (48.4% vs 48.2%) using only a few frames (12*f* vs 64*f*), and it even outperforms the complex models such as Non-local [35] with graph convolution [34]. This shows that CSST can allocate the parameter utilization space effectively, and alleviate the phenomenon of feature redundancy on 3D CNNs.

To further demonstrate the temporal reasoning ability of the proposed CSST-Net, we also conduct experiments on the Sth-Sth V2 and Jester datasets. The Sth-Sth V2 contains more video clips than the Sth-Sth V1, which can further release the full abilities of

Table 5
Comparison with other methods on benchmark datasets. (All models only taking RGB frames as inputs are listed in the table)

Method	Backbone	#Frames	#Param	FLOPs	Sth-Sth V1	
					Top-1(%)	Top-5(%)
TSN [32]	BNInception	8	10.7M	16G	19.5	-
TRN [37]	BNInception	8	18.3M	16G	42.0	-
TSM [14]	ResNet50	8	24.3M	33G	45.6	74.2
TSM [14]		16	24.3M	65G	47.2	77.1
TAM [20]		8	25.6M	33G	46.5	75.8
TAM [20]		16	25.6M	66G	47.6	77.7
SmallBigNet [17]		8	-	52G	47.0	77.1
TEINet [19]		8	30.4M	33G	47.4	-
GSM [27]		8	10.5M	17G	47.2	-
I3D [1]	3D ResNet50		28.0M	153G×2	41.6	72.2
NL I3D [35]	3D ResNet50	32×2clip	35.3M	168G×2	44.4	76.0
NL I3D+GCN [34]	3D ResNet50+GCN		62.2M	303G×2	46.1	76.8
ECO [39]	BNInception+3D Res18	8	47.5M	32G	39.6	-
ECO [39]		16	47.5M	64G	41.4	-
ECO_{EN}*lite* [39]		92	150M	267G	46.4	-
S3D-G [36]	BNInception	64	11.6M	71G	48.2	78.7
CorrNet-26 [31]	R(2+1)D-26	32	-	78G	47.4	-
GST [12]	3D ResNet50	8×2clip	21.0M	29.5G×2	47.6	76.6
CSST-Net	ResNet50	8	27.3M	37.7G	47.6	77.1
CSST-Net	ResNet50	12	27.3M	55.4G	48.4	77.8

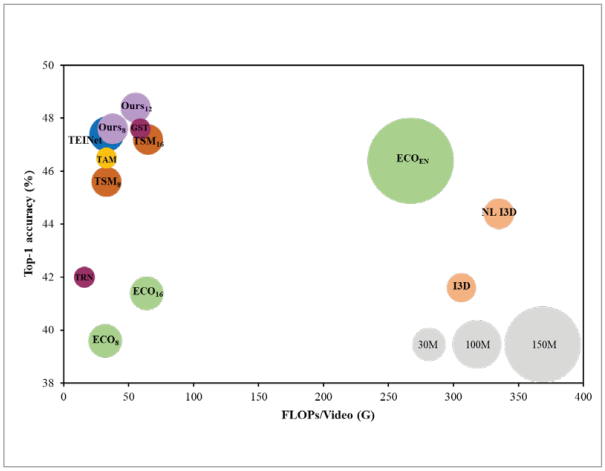
CSST-Net without suffering overfitting. We sample 8 frames from the video as input, and use 2 clips with 3 crops to report the accuracy in Table 6. On the Sth-Sth Vl dataset, compared with others with the same backbone, our method achieves a preferable performance, which shows the advantages of CSST-Net in modeling long-term temporal models. Secondly, on the Jester dataset, compared with previous works, our proposed method outperforms most architectures with similar structures, further verifying that CSST has significant performance in temporal modeling. Notably, our method is slightly lower than STM, but the latter obviously uses an evaluation protocol that is more computationally expensive.

Efficiency analysis. Figure 6 analyzes the memory overhead and computational complexity of the proposed CSST-Net and other representative approaches to evaluate the trade-off in accuracy, parameters, and FLOPs on the Sth-Sth V1 dataset. Specially, the area of bubbles represents the number of parameters for each method. As can be seen from Figure 6, compared with other methods, our method has more competitive performance, and has a more compromised computational cost and memory, which further demonstrates that our method can efficiently allocate the parameter space, and achieves the preferable balance between accuracy and computational costs.

Table 6
Comparison to the state-of-the-art on Sth-Sth V2 and Jester datasets

Method	Backbone	#Frames	Sth-Sth V2		Jester	
			Top-1(%)	Top-5(%)	Top-1(%)	Top-5(%)
TSN[32]	ResNet50	8×2×3	27.8	57.6	82.3	99.2
TRN[37]	BNInception	8×1×1	48.8	77.6	95.0	-
TSM[14]	ResNet50	8×2×3	58.6	85.3	94.7	99.7
TAM[20]		8×2×3	62.0	87.6	-	-
SmallBigNet[17]		8×2×3	61.6	81.7	-	-
STM[9]		8×3×10	62.3	88.8	96.6	99.9
TEA[18]		8×2×3	-	-	96.5	99.8
CSST-Net	ResNet50	8×2×3	62.3	88.6	96.5	99.9

Figure 6
Video classification performance on Sth-Sth V1



5. Conclusions

In this paper, we propose a novel channel split spatiotemporal (CSST) block to efficiently capture the spatial and temporal information in videos, and built a powerful video architecture (CSST-Net). The proposed CSST block achieves a good trade-off in terms of FLOPs and accuracy with the well-designed temporal module. Furthermore, we propose an efficient spatiotemporal fusion method, which can perform the recalibration and fusion of spatial and temporal features, and without introducing extra parameters. Experiments results on three challenging datasets (Sth-Sth V1, Sth-Sth V2 and Jester) indicate that CSST-Net is superior to the existing temporal modules significantly, demonstrating the effectiveness of CSST-Net on temporal reasoning.

Acknowledgement

This work was supported by the Scientific Research Program Funded by Education Department of Shaanxi Provincial Government (Program No. 23JK0529).

Ethical Permit

The authors declare that they have no conflict of interest. All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and national research committee. This article does not contain any studies with animals performed by any of the authors. Informed consent was obtained from all individual participants included in the study.

References

1. Carreira, J., Zisserman, A. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2017), Honolulu, HI, USA, July 21-26, 2017, 6299-6308. https://doi.org/10.1109/CVPR.2017.502
2. Diba, A., Sharma, V., Gool, L. V. Deep Temporal Linear Encoding Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2017), Honolulu, HI, USA, July 21-26, 2017, 1541-1550. https://doi.org/10.1109/CVPR.2017.168
3. Du, T., Wang, H., Torresani, L., Ray, J., LeCun, Y. A Closer Look at Spatiotemporal Convolutions for Action Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2018), Salt Lake City, UT, USA, June 18-22, 2018, 6450-6459.
4. Feichtenhofer, C. X3D: Expanding Architectures for Efficient Video Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2020), Virtual, Online, USA, June 14-19, 2020, 203-213. https://doi.org/10.1109/CVPR42600.2020.00028
5. Goyal, R., Kahou, S. E., Michalski, V., Materzynska, J., Westphal, S. The "something something" Video Database for Learning and Evaluating Visual Common Sense. Proceedings of the IEEE International Conference on Computer Vision, (ICCV 2017), Venice, Italy, October 22-29, 2017, 5842-5850. https://doi.org/10.1109/ICCV.2017.622
6. Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L. Accurate, Large Minibatch SGD: Training Imagenet in 1 Hour. arXiv preprint, 2017, arXiv:1706.02677.
7. He, K., Zhang, X., Ren, S., Sun, J. Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2016), Las Vegas, NV, USA, June 26-July 1, 2016, 770-778. https://doi.org/10.1109/CVPR.2016.90
8. Hussein, N., Gavves, E., Smeulders, W. M. Timeception for Complex Action Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2019), Long Beach, CA, USA, June 15-20, 2019, 254-263. https://doi.org/10.1109/CVPR.2019.00034
9. Jiang, B., Wang, M., Gan, W., Wu, W., Yan, J. STM: Spatiotemporal and Motion Encoding for Action Recognition. Proceedings of the IEEE International Conference on Computer Vision, (ICCV 2019), Seoul, Republic of Korea, October 27-November 2, 2019, 2000-2009. https://doi.org/10.1109/ICCV.2019.00209
10. Krizhevsky, A., Sutskever, I., Hinton, G. E. Imagenet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems, 2012, 1106-1114.
11. Kwon, H., Kim, M., Kwak, S., Cho, M. Motionsqueeze: Neural Motion Feature Learning for Video Understanding. arXiv preprint, 2020, arXiv:2007.09933. https://doi.org/10.1007/978-3-030-58517-4_21
12. Luo, C., Alan, Y. Grouped Spatial-Temporal Aggregation for Efficient Action Recognition. Proceedings of the IEEE International Conference on Computer Vision, (ICCV 2019), Seoul, Republic of Korea, October 27-November 2, 2019, 5512-5521. https://doi.org/10.1109/ICCV.2019.00561
13. Loshchilov, I., Hutter, F. SGDR: Stochastic Gradient Descent with Warm Restarts. arXiv preprint, 2016, arXiv:1608.03983.
14. Lin, J., Gan, C., Han, S. TSM: Temporal Shift Module for Efficient Video Understanding. Proceedings of the IEEE International Conference on Computer Vision, (ICCV 2019), Seoul, Republic of Korea, October 27-November 2, 2019, 7083-7093. https://doi.org/10.1109/ICCV.2019.00718
15. Liu, K., Liu, W., Gan, C., Tan, M., Ma, H. T-C3D: Temporal Convolutional 3D Network for Real-time Action Recognition. AAAI Conference on Artificial Intelligence, (AAAI 2018), New Orleans, USA, February 2-7, 2018, 7138-7145. https://doi.org/10.1609/aaai.v32i1.12333
16. Li, X., Wang, W., Hu, X., Yang, J. Selective Kernel Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2019), Long Beach, CA, USA, June 15-20, 2019, 510-519. https://doi.org/10.1109/CVPR.2019.00060
17. Li, X., Wang, Y., Zhou, Z., Qiao, Y. Smallbignet: Integrating Core and Contextual Views for Video Classification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2020), Virtual, Online, USA, June 14-19, 2020, 1092-1101. https://doi.org/10.1109/CVPR42600.2020.00117

18. Li, Y., Ji, B., Shi, X., Zhang, J., Kang, B. TEA: Temporal Excitation and Aggregation for Action Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2020), Virtual, Online, USA, June 14-19, 2020, 909-918. https://doi.org/10.1109/CVPR42600.2020.00099

19. Liu, Z., Luo, D., Wang, Y., Wang, L., Tai, Y. TEINet: Towards An Efficient Architecture for Video Recognition. Proceedings of the AAAI Conference on Artificial Intelligence, (AAAI 2020), New York, NY, USA, February 7-12, 2020, 11669-11676. https://doi.org/10.1609/aaai.v34i07.6836

20. Liu, Z., Wang, L., Wu, W., Qian, C., Lu, T. TAM: Temporal Adaptive Module for Video Recognition. Proceedings of the IEEE International Conference on Computer Vision, (ICCV 2021), Montreal, QC, Canada, October 10-17, 2021, 13688-13698. https://doi.org/10.1109/ICCV48922.2021.01345

21. Materzynska, J., Berger, G., Bax, I., Memisevic, R. The Jester Dataset: A Large-Scale Video Dataset of Human Gestures. Proceedings of the IEEE International Conference on Computer Vision Workshop, (ICCVW 2019), Seoul, Republic of Korea, October 27-28, 2019, 2874-2882. https://doi.org/10.1109/ICCVW.2019.00349

22. Ng, Y. H., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R. Beyond Short Snippets: Deep Networks for Video Classification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2015), Boston, MA, USA, June 7-12, 2015, 4694-4702.

23. Qiu, Z., Yao, T., Mei, T. Learning Spatio-Temporal Representation with Pseudo-3d Residual Networks. Proceedings of the IEEE International Conference on Computer Vision, (ICCV 2017), Venice, Italy, October 22-29, 2017, 5534-5542. https://doi.org/10.1109/ICCV.2017.590

24. Simonyan, K., Zisserman, A. Two-stream Convolutional Networks for Action Recognition in Videos. Advances in Neural Information Processing Systems, 2014, 568-576.

25. Sun, L., Jia, K., Yeung, D., Shi, B. Human Action Recognition Using Factorized Spatio-Temporal Convolutional Networks. Proceedings of the IEEE International Conference on Computer Vision, (ICCV 2015), Santiago, Chile, December 11-18, 2015, 4597-4605. https://doi.org/10.1109/ICCV.2015.522

26. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L. Mobilenetv2: Invertedresiduals and Linear Bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2018), Salt Lake City, UT, USA, June 18-22, 2018, 4510-4520. https://doi.org/10.1109/CVPR.2018.00474

27. Sudhakaran, S., Escalera, S., Lanz, O. Gate-Shift Networks for Video Action Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2020), Virtual, Online, USA, June 14-19, 2020, 1102-1111. https://doi.org/10.1109/CVPR42600.2020.00118

28. Sun, S., Kuang, Z., Sheng, L., Ouyang, W. L., Zhang, W. Optical Flow Guided Feature: A Fast and Robust Motion Representation for Video Action Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2018), Salt Lake City, UT, USA, June 18-22, 2018, 1390-1399. https://doi.org/10.1109/CVPR.2018.00151

29. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M. Learning Spatiotemporal Features with 3D Convolutional Networks. Proceedings of the IEEE International Conference on Computer Vision, (ICCV 2015), Santiago, Chile, December 11-18, 2015, 4489-4497. https://doi.org/10.1109/ICCV.2015.510

30. Wang, H., Schmid, C. Action Recognition with Improved Trajectories. Proceedings of the IEEE International Conference on Computer Vision, (ICCV 2013), Sydney, NSW, Australia, December 1-8, 2013, 3551-3558. https://doi.org/10.1109/ICCV.2013.441

31. Wang, H., Tran, D., Torresani, L., Feiszli, M. Video Modeling with Correlation Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2020), Virtual, Online, USA, June 14-19, 2020, 352-361. https://doi.org/10.1109/CVPR42600.2020.00043

32. Wang, L., Li, W., Li, W., Gool, L. V. Appearance-and-Relation Networks for Video Classification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2018), Salt Lake City, UT, USA, June 18-22, 2018, 1430-1439. https://doi.org/10.1109/CVPR.2018.00155

33. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. Proceedings of the European Conference on Computer Vision, (ECCV 2016), Amsterdam, Netherlands, October 8-16, 2016, 20-36. https://doi.org/10.1007/978-3-319-46484-8_2

34. Wang, X., Gupta, A. Videos as Spacetime Region Graphs. Proceedings of the European Conference on Computer Vision, (ECCV 2018), Munich, Germany, September

8-14, 2018, 399-417. https://doi.org/10.1007/978-3-030-01228-1_25

35. Wang, X., Girshick, R., Gupta, A., He, K. Non-local Neural Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2018), Salt Lake City, UT, USA, June 18-22, 2018, 7794-7803. https://doi.org/10.1109/CVPR.2018.00813
36. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K. Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-Offs in Video Classification. Proceedings of the European Conference on Computer Vision, (ECCV 2018), Munich, Germany, September 8-14, 2018, 318-335. https://doi.org/10.1007/978-3-030-01267-0_19
37. Zhou, B., Andonian, A., Oliva, A., Torralba, A. Temporal Relational Reasoning in Videos. Proceedings of the European Conference on Computer Vision, (ECCV 2018), Munich, Germany, September 8-14, 2018, 803-818. https://doi.org/10.1007/978-3-030-01246-5_49
38. Zhu, J., Zhu, Z., Zou, W. End-to-End Video-Level Representation Learning for Action Recognition. Proceedings of the IEEE International Conference on Pattern Recognition, (ICPR 2018), Beijing, China, August 20-24, 2018, 645-650. https://doi.org/10.1109/ICPR.2018.8545710
39. Zolfaghari, M., Singh, K., Brox, T. ECO: Efficient Convolutional Network for Online Video Understanding. Proceedings of European Conference on Computer Vision, (ECCV 2018), Munich, Germany, September 8-14, 2018, 713-730. https://doi.org/10.1007/978-3-030-01216-8_43
40. Zhu, Y., Lan, Z., Newsam, S., Hauptmann, A. Hidden Two-Stream Convolutional Networks for Action Recognition. arXiv preprint, 2017, arXiv:1704.00389.