applied sciences

MDPI

Article

Action Recognition Network Based on Local Spatiotemporal Features and Global Temporal Excitation

Shukai Li [1], Xiaofang Wang [2,*], Dongri Shan [1,*] and Peng Zhang [2]

[1] School of Mechanical Engineering, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250300, China; 10431200088@stu.qlu.edu.cn
[2] School of Information and Automation Engineering, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250300, China; zp@qlu.edu.cn
* Correspondence: wxf2012@stu.xjtu.edu.cn (X.W.); shandongri@qlu.edu.cn (D.S.); Tel.: +86-152-7515-9896 (X.W.); +86-138-6406-5008 (D.S.)

Abstract: Temporal modeling is a key problem in action recognition, and it remains difficult to accurately model temporal information of videos. In this paper, we present a local spatiotemporal extraction module (LSTE) and a channel time excitation module (CTE), which are specially designed to accurately model temporal information in video sequences. The LSTE module first obtains difference features by computing the pixel-wise differences between adjacent frames within each video segment and then obtains local motion features by stressing the effect of the feature channels sensitive to difference information. The local motion features are merged with the spatial features to represent local spatiotemporal features of each segment. The CTE module adaptively excites time-sensitive channels by modeling the interdependencies of channels in terms of time to enhance the global temporal information. Further, the above two modules are embedded into the existing 2DCNN baseline methods to build an action recognition network based on local spatiotemporal features and global temporal excitation (LSCT). We conduct experiments on the temporal-dependent Something-Something V1 and V2 datasets. We compare the recognition results with those obtained by the current methods, which proves the effectiveness of our methods.

Keywords: local spatiotemporal features; channel time excitation; action recognition; feature enhancement

Citation: Li, S.; Wang, X.; Shan, D.; Zhang, P. Action Recognition Network Based on Local Spatiotemporal Features and Global Temporal Excitation. *Appl. Sci.* **2023**, *13*, 6811. https://doi.org/10.3390/app13116811

Academic Editor: Oscar Reinoso García

Received: 27 April 2023
Revised: 24 May 2023
Accepted: 1 June 2023
Published: 3 June 2023

1. Introduction

Action recognition has aroused great interest in the field of computer vision due to its potential applications in fields such as virtual reality, human–computer interaction, and video surveillance [1,2]. Unlike still images, videos contain temporal properties that are crucial for recognizing human actions. Many actions, such as "pushing the table" and "pulling the table" in Figure 1, are similar in terms of background and can only be distinguished by accurately reasoning about temporal information in videos. Therefore, how to accurately model temporal information in videos is a key issue.

Researchers have proposed various temporal modeling approaches to recognize actions in videos. There are three primary types of approaches for action recognition. The first is the two-stream architecture, which consists of a spatial branch and optical flow branch [3–5]. The optical flow branch is designed to extract local motion features from optical flow, but it is computationally expensive and challenging to extract optical flow in real-word applications. The second approach is based on 3DCNNs [6–15], which have achieved good results by using 3D convolution kernels to extract temporal and spatial features from videos. However, the model of 3D CNN is larger, which brings higher computational cost.

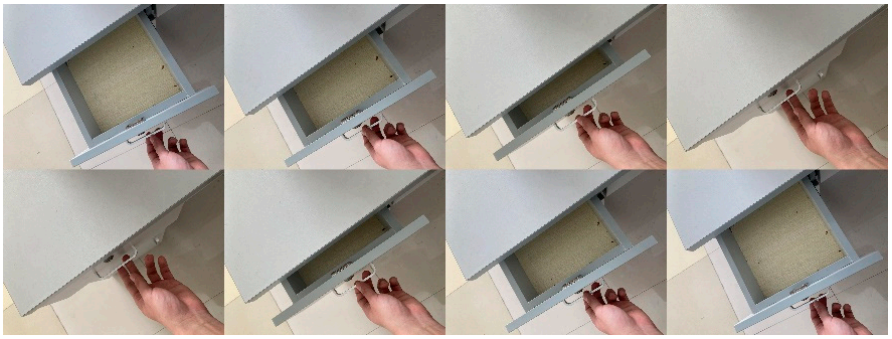

Figure 1. Two actions that are very similar in terms of background. First row: pushing the table; second row: pulling the table.

The third category of approaches model the temporal information based on 2DCNNs [16–20]. Initially, TSN [16] uses a segmented sampling strategy to split the video equally and then randomly captures a frame from each video clip to feed the network. However, TSN only simply splits the video segment and fuses equally the features of each segment at a later stage of the network, which lacks temporal modeling capability. To address this issue, multiple temporal modules [17–20] are designed to be embedded in 2DCNNs, where 2D convolution kernels extract spatial features while temporal modules extract temporal features. Based on the TSN, TSM [17] designs a time shift module that shifts partial channels in the time dimension to exchange temporal information of neighboring video frames. However, TSM samples video sequences using a segmented sampling strategy, which ignores the local temporal information within each video segment. We argue that accurate action recognition depends on the local temporal information. TDN [18] extracts appearance and local motion information to produce an effective video representation by supplying a single RGB frame with a temporal difference. Furthermore, considering different channels have different sensitivity for temporal information, SENet [21] models the interdependencies between channels to recalibrate feature responses and adaptively enhances the salient features. In the field of action recognition, TEI [19], TEA [20], etc., have designed motion excitation modules and embedded them into 2D CNNs of ResNet [22] architecture, which improves the capability of temporal modelling.

To accurately model temporal information in videos, combining the ideas of the temporal difference and motion information excitation, this paper proposes a local spatiotemporal extraction module (LSTE) and a channel time excitation module (CTE) to accurately model temporal information in video. In LSTE, difference features are first obtained by computing pixel-wise differences between adjacent video frames within each video segment, and the local motion features are obtained by stressing the effect of the feature channels sensitive to difference information. The local motion features are fused with the spatial features to represent the local spatiotemporal information in each segment. The CTE module excites time-sensitive channels by modeling the interdependences of channels in terms of time to enhance the global temporal information. Finally, we embed these two modules into the TSM [17] network to build an action recognition network based on local spatiotemporal features and global temporal excitation (LSCT).

The following is a summary of the contributions in this paper:

(1) We propose an LSTE module that extracts the local motion features and fuses them with the spatial features to obtain spatiotemporal features of each video segment.
(2) We propose a CTE module that excites time-sensitive channels by modeling the interdependences of channels in terms of time to enhance the global temporal information.
(3) These two modules are plug-and-play modules and are lightweight, in which the LSTE module can be embedded in the first layer of action recognition networks to extract local spatiotemporal information, and the CTE module can be embedded in the action recognition network based on the residual structure to enhance the global temporal information. Based on the TSM [17] network, we embed these two modules

into this network to build an LSCT network. We performed experiments on the Something-Something V1 and V2 datasets.

2. Related Work

Many researchers have begun to apply deep learning technology in the tasks of video action recognition because it brings a huge improvement in the field of images. A wide range of deep-learning-based action recognition methods have been suggested by numerous researchers, outperforming traditional methods in performance.

Methods based on 3DCNN. Videos contain an additional temporal dimension compared with images, making it more challenging to collect information from videos. 3DCNN-based action recognition methods have been suggested to recognize actions in videos by extracting spatial and temporal features. C3D [10] applied 3D convolution to extract temporal and spatial features from videos by convolving on adjacent frames. T3D [11] suggested a new migration learning method, which migrates the weight parameters of the previously trained 2D CNN to the 3D CNN, and captured time information at different time scales. SlowFast [7] used dual path network to sample frame sequences at unequal sampling speeds to extract spatial and temporal features separately. The slow path extracted spatial features, while the fast path extracted temporal features. Although these methods based on 3DCNN architecture have achieved good recognition results, they have some drawbacks such as a lot of parameters and slow convergence speed. To lower the computational cost of 3D CNN, S3D [13], P3D [14], and other similar methods decomposed the 3D convolution into 2D convolution, for extracting spatial information, and 1D convolution for extracting temporal information. ECO [15], GST [23] mixed 2D convolution, and 3D convolution are used in a network to improve computational efficiency. In contrast, our LSCT network utilizes temporal modeling modules to enhance the performance of 2D CNNs, which does not result in a large computational cost.

Methods based on 2DCNN. Karpathy et al. [24] proposed a 2D CNNs-based action recognition network that utilized a multi-resolution approach and was trained on the Sports-1M dataset. It used RGB frames as input, and tested various methods to combine temporal information. However, it performed poorly since it was unable to accurately capture the motion information of neighboring frames. Simonyan et al. [3] suggested a two-stream network to recognize actions in videos, where spatial stream with a RGB frame extracted spatial features while the flow stream with optical flow information extracted local motion features. However, extracting the optical flow was computationally expensive and required separate extraction, preventing end-to-end learning. TSN [16] proposed a segmented sampling strategy, which split the video into fixed segments equally and randomly captured a frame from each segment to feed the network. However, it lacked local temporal modeling and relied heavily on precomputed optical flow to extract local motion information. Based on this sampling strategy, subsequent studies have proposed multiple temporal modeling modules that were embedded in 2D CNNs to effectively model temporal information. TSM [17] designed a time shift module to exchange the temporal information of neighboring video frames by shifting partial channels in the time dimension. This module was embedded in 2D CNNs to model temporal features of video sequences. It has achieved high recognition performance with relatively little computational cost. TEI [19] employed an attention mechanism that utilized motion information to enhance motion-related features and leveraged a time shift module to model temporal features. STM [25], TIN [26], TEA [20], and TAM [27] etc. designed rich temporal modules which were embedded into 2DCNNs to effectively recognize actions in videos. The methods discussed above utilize the segmented sampling approach suggested by TSN to sample video sequences and have achieved good recognition results. However, the local motion information within each segment was ignored by this sampling strategy since it only selected one frame from each segment. To model fine temporal information, the input video can be split into more segments, but that will cause a higher processing cost. To solve this problem, we suggested an LSTE module that extracts the local motion features and fuses

them with the spatial features to obtain spatiotemporal features of each video segment. Our LSTE module compensates for the shortcomings of the above methods which ignore the local motion information of each video segment.

Attention mechanism in action recognition. The SENet [21] module modelled the interdependencies between feature channels using two fully connected layers, adaptively recalibrating feature responses of each channel. It effectively improves the capability of 2D CNNs in image recognition tasks. For action recognition tasks, TEA [20] designed a motion excitation module to enhance motion features by using the frame difference between segments to recalibrate channel-wise features responses. The ACTION-Net [28] successfully designed spatiotemporal attention, channel attention, and motion attention modules to improve the capability of 2D CNNs. Inspired by the above mechanism methods, this paper proposes a CTE module that models the interdependencies between feature channels in terms of time, and adaptively excites the time-sensitive channels to enhance the global temporal information.

3. Method

In this section, we present the technical principles of our LSCT network in detail. Firstly, we introduce the local spatiotemporal extraction module (LSTE) and the channel time excitation module (CTE). Then, we describe how these two modules are embedded into the baseline network TSM to form the LSCT network, in which we use the sampling method proposed by TSN to sample the video to feed the LSCT network.

3.1. LSCT-Net

LSTE and CTE can be embedded in common 2D convolutional neural networks, such as ResNet [22], MobileNet [29], and the action recognition models based on these networks [19,25,27]. TSM [17] proposed a time shift module embedded in ResNet-50 to extract temporal features and achieved good recognition results while maintaining a relatively low computational cost. We used TSM as the base network and embedded the LSTE module and the CTE module into it to construct our LSCT network, as shown in Figure 2. Specifically, we substituted the first 7×7 convolution layer of TSM [17] with the LSTE module to extract local spatiotemporal features of each segment, which are stacked along the time dimension and fed to the subsequent res2 stage. Meanwhile, we inserted the CTE module after the shift module to excite time-sensitive channels and enhance the global temporal information.

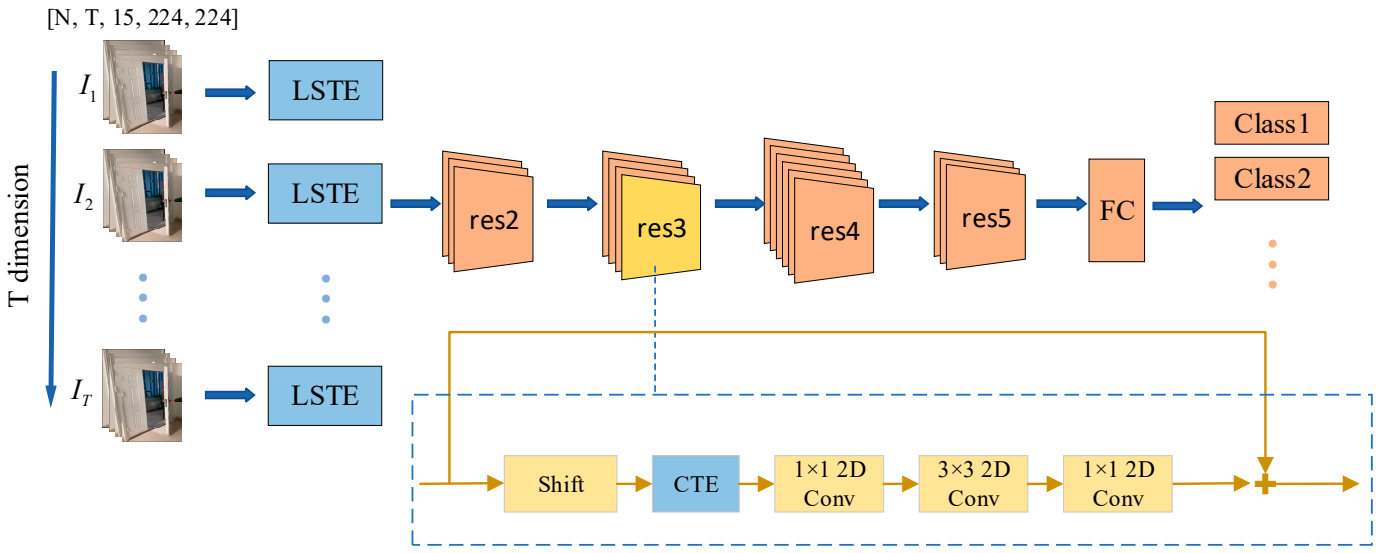

Figure 2. The framework of the LSCT network.

3.2. Local Spatiotemporal Extraction Module (LSTE)

The LSTE module obtained the local motion features through a motion branch, which is fused with the spatial features extracted through a spatial branch, as shown in Figure 3. In the spatial branch, a frame I_i randomly selected from each video segment was fed to a 7×7 convolutional layer to obtain the spatial features $X_s \in R^{N \times 64 \times \frac{H}{2} \times \frac{W}{2}}$. N represents the batch size, H represents the feature map's height, W represents the feature map's width. In the motion branch, the local motion features around a frame I_i was extracted. Firstly, we counted the pixel-wise differences of neighboring video frames $[I_{i-2}, I_{i-1}, I_i, I_{i+1}, I_{i+2}]$ and stacked them along channel dimension $D = [D_1, D_2, D_3, D_4]$, $D \in R^{N \times 12 \times H \times W}$. To reduce the computational cost, we performed an average pooling on D to halve the spatial size, and we obtained difference features X_D by a 7×7 2D convolutional layer, which is represented in Equation (1):

$$X_D = CNN(Avg\_pool(D(I_i))), X_D \in R^{N \times 64 \times \frac{H}{4} \times \frac{W}{4}} \quad (1)$$

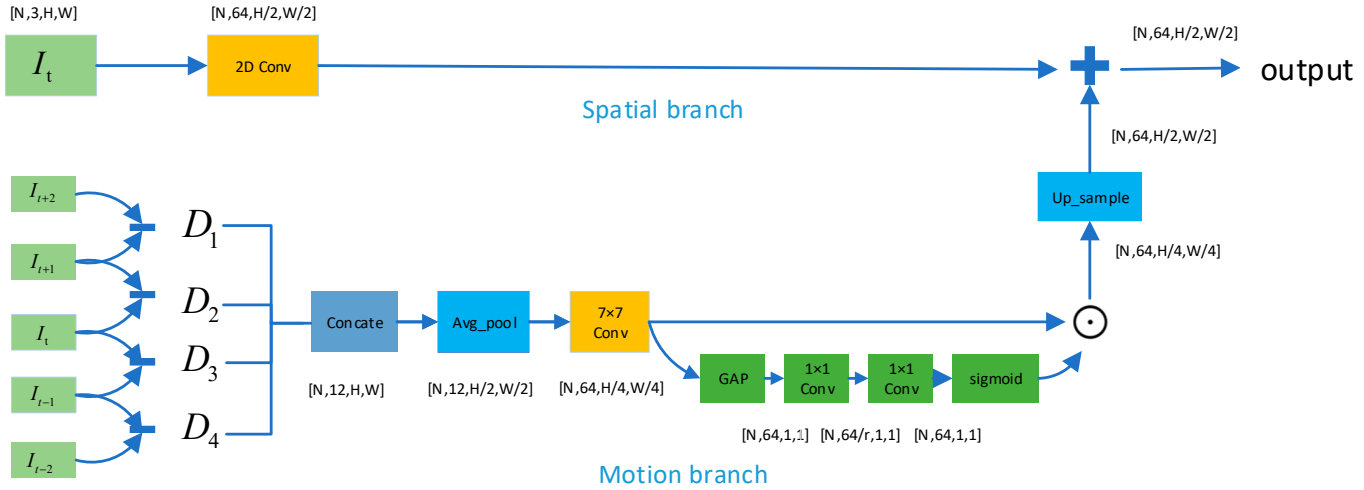

Figure 3. The framework of the LSTE module.

Then, we strengthened the difference features X_D by stressing the effect of the feature channels sensitive to difference information, which can be divided into the following three operations. In the first operation, we scaled the spatial dimension to 1×1 by using a global average pooling on X_D to obtain the global information F, which is represented in Equation (2):

$$F = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_D[:, :, i, j], F \in R^{N \times 64 \times 1 \times 1} \quad (2)$$

In the second operation, we performed two 1×1 convolution operations on F and obtained the importance weight s corresponding to each channel through a Sigmoid activation function, which is represented in Equation (3):

$$s = \sigma(W_2 * \delta(W_1 * F)) \quad (3)$$

where W_1 is the first 1×1 convolution, which was used to fully capture the interdependencies between each channel and reduce channel dimension by a ratio r ($r = 16$). δ is the ReLU activation function. W_2 is the second 1×1 convolution, which was used to recover the number of feature channels. σ is Sigmoid activation function. In the third operation, the local motion features $\overline{X}_D$ were obtained by performing a channel-wise multiplication between the difference features X_D and the importance weight s, which is shown in Equation (4):

$$\overline{X}_D = s \odot X_D, \overline{X}_D \in R^{N \times 64 \times \frac{H}{4} \times \frac{W}{4}} \quad (4)$$

Finally, we up-sampled $\overline{X}_D$ to match the spatial features X_s and fused them, which is shown in Equation (5):

$$X_L = X_s + Upsample(\overline{X}_D), X_L \in R^{N \times 64 \times \frac{H}{2} \times \frac{W}{2}} \tag{5}$$

where X_L is the output feature of the LSTE module. Through the above operations, we obtained the local spatiotemporal features of each video segment.

3.3. Channel Time Excitation Module (CTE)

The CTE module adaptively excites time-sensitive feature channels by modeling the interdependences of channels in terms of time to strengthen the global temporal information, as shown in Figure 4. Since our CTE module focuses on capturing temporal dynamics, we firstly used a spatial domain global average pooling to compress the spatial information of the given input feature $X \in R^{N \times T \times C \times H \times W}$,where T represents the number of segments, as represented in Equation (6):

$$F = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X[:, :, :, i, j], F \in R^{N \times T \times C \times 1 \times 1} \tag{6}$$

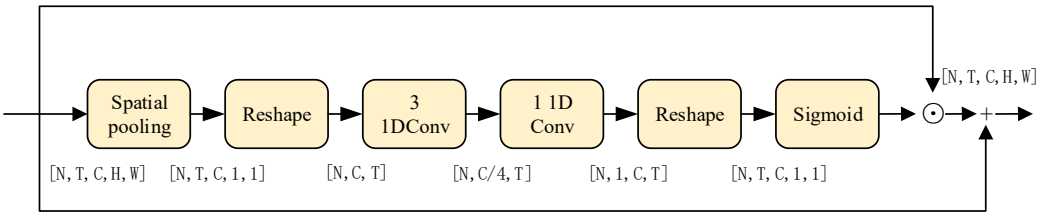

Figure 4. The framework of the CTE module.

We reshaped F into $F \in R^{N \times C \times T}$ to reason the temporal information. To learn the temporal information of neighboring frames, we used a 1D convolutional with a kernel size of three to perform F in the temporal dimension, which decreased the amount of feature channels by a ratio r ($r = 4$) to control the computational cost and obtained good performance; this is shown in Equation (7):

$$F_s = Conv * F, F_s \in R^{N \times \frac{C}{r} \times T} \tag{7}$$

Another 1D convolution with a kernel size of one was adopted to perform F_s, which recovers the amount of channels, and the time-sensitive weight P is obtained by a Sigmoid activation function σ, which is shown in Equation (8):

$$P = \sigma(Conv_{\exp} * F_s), P \in R^{N \times C \times T \times 1 \times 1} \tag{8}$$

Finally, the input feature X and time-sensitive weight P were multiplied in a channel-wise way to excite time-sensitive channels. The original background that is beneficial for action recognition was preserved by a residual link, as shown in Equation (9):

$$X_{out} = X + X \odot P \tag{9}$$

where X_{out} is the output feature of the CTE module. Through the above operations, the temporal of input feature X was enhanced.

4. Experiments

In this section, we first go through the specifics of the LSCT network's implementation. Then, we carry out experiments on temporal-dependent datasets Something-Something V1 and V2. Meanwhile, we also conduct ablation experiments for the CTE module and LSTE module on Something-Something V1.

4.1. Datasets and Evaluation Metrics

For Something-Something V1 and V2 [30], we used the code provided by TSM to divide them into training sets, verification sets, test sets according to the official label files, and their ratio is 8:1:1. These two datasets contain interactions between humans and commodities (such as spoons, bottles, paper) in daily life and both have 174 categories. Something-Something V1 has 86,017 training videos, 11,522 validation videos, and 10,960 testing videos. Something-Something V2 has 168,913 training videos, 24,777 validation videos, and 27,157 test videos. The training sets and the validation sets were mainly employed in previous studies, and the accuracy of the validation sets was reported in the end. For comparison purposes, we also reported the accuracy of validation sets in this paper.

These two datasets differ from other datasets in the recognition of actions in videos is highly dependent on temporal information, while actions in other datasets are more dependent on scene information and can be identified by only relying on one or a few frames in videos. The action changes of characters in these two datasets are very similar in terms of scene, such as two actions ('Tearing something into two pieces' and 'Tearing something just a little bit'). Therefore, accurately modeling temporal information in videos is a key issue.

The evaluation metrics: Top-1 (%) accuracy, Top-5 (%) accuracy, FLOPs:

Top-1 accuracy: we only consider the highest predicted labels for a given sample. If it matches the true label for that sample, it is considered to be the correct classification.

Top-5 accuracy: we consider the top five predicted labels for a given sample. If the true label is one of the top five predicted labels, it is considered to be the correct classification. Top-5 accuracy is usually more lenient than Top-1 accuracy.

FLOPs calculate the number of floating-point operations performed by a deep learning model. FLOPs are often used to estimate the computational complexity of a model, which is important for determining the efficiency and usefulness of a model in real-world scenarios.

4.2. Implementation Details

Training. In accordance with TSN's segmented sampling strategy, the input video was split into T segments equally. Then, a sequence of frames, which were randomly selected from each segment, was input to the LSCT network. The short side of these frames was adjusted to 256, as seen in [17]. For the purpose of enhancing the data, we applied corner cropping and random scaling to these frames. Finally, the cropped region of each frame was scaled to 224×224 for training the LSCT network. The size of data inputted to the LSCT network was $N \times T \times 15 \times 224 \times 224$. We used the weights from ResNet-50, which was trained on the ImageNet dataset, to initialize the LSCT network's parameters. For the Something-Something V1 and V2 datasets, we totally trained for 60 epochs. When the sampled frame was $T = 8$, the batch size was set to $N = 32$, and when sampled sample was $T = 16$, the batch size was set to $N = 16$. For these two datasets, we set the initial learning rate as 0.01 and decreased it by 0.1 at 20, 40, and 50 epochs. The stochastic gradient descent (SGD) was used as the optimizer to optimize the network parameters, where momentum is 0.9 and weight decay is 5×10^{-4}. Dropout was set as 0.8 to prevent the overfitting of the network.

Validation. We present Top-1%, Top-5% accuracy and FLOPs of the validation set. We used the center-crop strategy following [17] for inference. We only used one clip per video and the center 224×224 crop.

4.3. Comparisons with the State of the Art

We compared LSCT networks with state-of-the-art action recognition approaches on Something-Something V1 and V2. Tables 1 and 2 display the combined data, which includes the method, backbone, frames, FLOPs (G), top-1 (%), and top-5 (%).

Table 1. Results compared with state of the arts on Something-Something V1.

Method	Backbone	Frames	FLOPs (G)	Pre-Trained	Top-1 (%)	Top-5 (%)
I3D [31]	Resnet-50	32 × 2	153 × 2	ImageNet + K400	41.6	72.2
ECO [15]	BNInception + ResNet-18	8	32	Kinetics-400	39.6	-
ECO_{En} [15]	BNInception + ResNet-18	92	267	Kinetics-400	46.4	-
SAST [8]	BNInception + ResNet-18	16	-	Kinetics-400	44.3	-
SAST [8]	BNInception + ResNet-18	32	-	Kinetics-400	45.6	-
TSN [16]	Resnet-50	8	33	Kinetics-400	19.5	46.6
TSN [16]	Resnet-50	16	66	Kinetics-400	19.7	47.3
TSM [17]	Resnet-50	8	33	ImageNet	45.6	74.2
TSM [17]	Resnet-50	16	66	ImageNet	47.2	77.1
TRN [32]	BNInception	8	16	ImageNet	40.6	-
GST [23]	Resnet-50	8	29.5	ImageNet	47.0	76.1
GST [23]	Resnet-50	16	59	ImageNet	48.6	77.9
STM [25]	Resnet-50	8 × 30	33 × 30	ImageNet	49.2	79.3
STM [25]	Resnet-50	16 × 30	67 × 30	ImageNet	51.0	80.4
TEI [19]	Resnet-50	8	33	ImageNet	47.4	-
TEI [19]	Resnet-50	16	66	ImageNet	49.9	-
LSCT (Ours)	Resnet-50	8	34	ImageNet	49.3	78.4
LSCT (Ours)	Resnet-50	16	68	ImageNet	50.6	79.6

Table 2. Results compared with state of the arts on Something-Something V2.

Method	Backbone	Frames	FLOPs (G)	Top-1 (%)	Top-5 (%)
TSN [16]	Resnet-50	8	33	27.8	57.6
TSN [16]	Resnet-50	16	66	30.0	60.5
TRN [32]	BNInception	8	16	48.8	77.6
CPNet [33]	Resnet-18	16 × 6	-	54.1	82.1
CPNet [33]	Resnet-34	16 × 6	-	57.7	84.0
TSM [17]	Resnet-50	8 × 6	198	59.1	85.6
TSM [17]	Resnet-50	16 × 6	390	63.4	88.5
GST [23]	Resnet-50	8	29.5	61.6	87.2
GST [23]	Resnet-50	16	59	62.6	87.9
TEI [19]	Resnet-50	8	33	61.3	-
TEI [19]	Resnet-50	16	66	62.1	-
LSCT (Ours)	Resnet-50	8	34	61.4	86.9
LSCT (Ours)	Resnet-50	16	68	62.3	87.6

Something-Something v1 dataset. According to Table 1, the baseline approach TSN [16] receives very low recognition results compared with the other methods, which shows the significance of temporal modeling in action recognition. In contrast to the TSM baseline approach [17], LSCT network achieves a higher accuracy with relatively low FLOPs. When sampling eight frames as input, our LSCT network achieves 3.7% accuracy improvement over TSM, with only a slight increase in FLOPs to 34G. Action recognition approaches of Table 1 can be classified into two types. The first type is 3DCNN-based methods, including I3D [31], ECO [15], SAST [8], and GST [23]. The 3DCNN-based methods have a large network model, which causes high FLOPs. In contrast, our proposed LSCT network achieves superior performance with low FLOPs, outperforming these 3DCNN-based methods. Specifically, our LSCT network achieves a 5% accuracy improvement over SAST and achieves 2.3% accuracy improvement over GST when sampling eight frames as input. The second category is 2DCNN-based methods, including TSM [17], STM [25], TRN [32], and TEI [19]. Our proposed LSCT network outperforms these methods. When sampling eight frames as input, despite a slight increase in FLOPs to 34G, our LSCT network achieves 1.9% accuracy improvement over TEI. Moreover, it also achieves competitive results compared with STM, which employs a three-crop strategy.

Among the above methods, our LSCT network achieves the highest accuracy while maintaining relatively low FLOPs, demonstrating the effectiveness of our LSCT net-

work. These findings prove the ability of the LSCT network for improving action recognition performance.

Something-Something v2. Table 2 compares the results on the Something-Something V2 dataset and demonstrates a considerable improvement over the baseline methods TSN and TSM. When sampling eight frames as input, the Top-1 accuracy of our LSCT network is 2.3% higher than that of TSM. Since TSM uses two-clip and three-crop strategy, the eight sampled frames are increased to 48, which results in 198G FLOPs. Moreover, our LSCT network also achieves competitive results compared with advanced action recognition methods such as GST and TEI. Our LSCT network also outperforms other popular methods, which demonstrates its effectiveness in action recognition tasks.

4.4. Ablation Studies

In this section, we conduct ablation studies of the LSCT network on the Something-Something V1 dataset, which demonstrate the effectiveness of the LSTE and CTE. Baseline methods TSN and TSM are used for comparison.

4.4.1. Study of LSTE and CTE Modules

To evaluate the impact of the LSTE module and CTE module for action recognition, we conducted experiments with the TSM model as our backbone. The results of Table 3 indicate that these two modules can significantly enhance the performance of the baseline approaches TSN and TSM. Specifically, the LSTE module increases the Top-1 accuracy by 2.5% compared with TSM. In contrast to TSM, the LSTE module operates directly on the input video sequence at the first layer of the network to capture local spatiotemporal information by computing pixel-wise differences of neighboring frames, leading to improved action recognition accuracy. The CTE module increases the Top-1 accuracy by 0.9% compared with TSM by modeling the independences between channels in terms of time to excite time-sensitive feature channels. The combination of the LSTE module and CTE module can help us learn richer local and global spatiotemporal features and achieves higher accuracy.

Table 3. Study of LSTE and CTE modules.

Method	Top-1	Top-5
TSN	19.7	46.6
TSM	45.6	74.2
Only LSTE	48.1	77.0
Only CTE	46.5	75.1
LSCT	49.3	78.4

4.4.2. Study on the Number of CTE

ResNet-50′s architecture can be seen as six stages, namely conv1, res2, res3, res4, res5, and FC. We used TSM as the backbone in this experiment. With the LSTE replacing conv1, we tested the impact of the number of residual stages, including the CTE module. We respectively embedded the CTE module into the res2 stage, res2 and res3 stages, res2, res3, and res4 stages, re2, res3, res4, and res5 stages, respectively. Table 4 displays the results, and it is clear that more residual stages including the CTE can yield better performance.

Table 4. Study on the number of the CTE module.

Stage	Top-1	Top-5
res2	48.5	77.3
res2,3	48.7	77.6
res2,3,4	49.1	78.1
res2,3,4,5	49.3	78.4

4.4.3. Results Compared with Other Temporal Modules

When each video was split into eight segments, our LSCT network actually sampled 40 frames to feed the LSCT network. To better test the effectiveness of LSCT network, we compared the LSCT network with other action recognition methods TSM and TEI. For these two methods, we sampled 8 and 40 frames as input of the TSM and TEI networks, respectively, and compared them with our LSCT network. The outcomes are displayed in Table 5. When sampling eight frames as input, our LSCT network outperforms the TSM and TEI networks with a slight increase in FLOPs to 34G. When sampling 40 frames as input, our LSCT network still outperforms the TSM and TEI. Although our LSCT network also inputs 40 frames, only 8 frames were involved in computation. The remaining 32 frames were only used to capture local spatiotemporal information within each segment in the first layer of LSCT network. Thus, our LSCT network does not cause high FLOPs. This demonstrates the efficiency of the LSTE module which calculates spatial features and local motion features in each segment and fuses them to obtain local spatiotemporal features.

Table 5. Results compared with other temporal modules.

Model	FLOPs (G)	Top1	Top5
TSM [17]	33	45.6	74.2
TSM_{40} [17]	165	47.6	77.9
TEI [19]	33	47.4	77.2
TEI_{40} [19]	165	49.0	79.0
LSCT	34	49.3	78.4

4.4.4. Analysis of Real-Time Operation

We present the latency of real-time operation on the Tesla V100 in Table 6. 'sec/video' represents how many seconds it takes to recognize a video. We used the batch size of 64 to test the latency. It can be seen that although our model is slightly slower than TSM, it is still guaranteed to run in real time.

Table 6. Analysis of real-time operation.

Method	Frames	Latency (Sec/Video)	Top-1 (%)
TSM	$8 \times 1 \times 1$	0.016	45.6
TSM	$16 \times 1 \times 1$	0.025	47.2
LSCT	$8 \times 1 \times 1$	0.033	49.3

5. Conclusions

In the paper, we propose a local spatiotemporal extraction module (LSTE) and a channel time excitation module (CTE). The LSTE module first obtains difference features by computing the pixel-wise differences of neighboring video frames within each video segment, and then obtains local motion features by stressing the effect of the feature channels sensitive to difference information. The local motion features are fused with the spatial features to represent the local spatiotemporal information in each segment. The CTE module adaptively excites time-sensitive channels by modeling the interdependencies of channels in terms of time to enhance the global temporal information.

Furthermore, we embed the LSTE module and the CTE module into the TSM network to build an action recognition network based on local spatiotemporal features and global temporal excitation (LSCT). On the Something-Something V1 and V2 datasets, we perform experiments and contrast the results with those results obtained by advanced action recognition methods. On the Something-Something V1, the accuracy of the LSCT network is 3.7% higher than the baseline method TSM, 2.3% higher than GST, and 1.9% higher than TEI. At the same time, we also conduct ablation studies, and the accuracy of the LSTE module and the CTE module increases by 2.5% and 0.9%, respectively, compared with the baseline method TSM. The results prove the effectiveness of the LSCT network.

Although the LSCT network achieved good recognition accuracy, there is still room for improvement. First, the LSCT network utilizes the LSTE module to fully sample video frames to capture temporal information, which may sample redundant frames. Next, we will modify the network for how to extract key frames. Second, the parameters of the LSCT network are slightly higher than those of the TSM. The next step is to modify the network model to improve computing efficiency.

Author Contributions: Conceptualization, D.S., X.W. and S.L.; methodology, S.L. and X.W.; software, S.L.; validation, D.S., X.W. and S.L.; formal analysis, S.L. and X.W.; investigation, S.L.; resources, D.S. and X.W.; data curation, S.L.; writing—original draft preparation, S.L.; writing—review and editing, X.W. and P.Z.; visualization, S.L.; supervision, D.S., X.W. and P.Z.; project administration, D.S. and X.W.; funding acquisition, D.S. and X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Project of Shandong Provincial Major Scientific and Technological Innovation, grant No. 2019JZZY010444, No. 2019TSLH0315; in part by the Project of 20 Policies of Facilitate Scientific Research in Jinan Colleges, grant No. 2019GXRC063; and in part by the Natural Science Foundation of Shandong Province of China, grant No. ZR2020MF138.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yao, G.; Lei, T.; Zhong, J. A review of Convolutional-Neural-Network-based action recognition. *Pattern Recognit. Lett.* **2019**, *118*, 14–22. [CrossRef]
2. Wu, C.-Y.; Girshick, R.; He, K.; Feichtenhofer, C.; Krahenbuhl, P. A multigrid method for efficiently training video models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 153–162.
3. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. *arXiv* **2014**, arXiv:1406.2199.
4. Feichtenhofer, C.; Pinz, A.; Zisserman, A. Convolutional two-stream network fusion for video action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1933–1941.
5. Feichtenhofer, C.; Pinz, A.; Wildes, R.P. Spatiotemporal multiplier networks for video action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4768–4777.
6. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 221–231. [CrossRef] [PubMed]
7. Feichtenhofer, C.; Fan, H.; Malik, J.; He, K. Slowfast networks for video recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6202–6211.
8. Wang, F.; Wang, G.; Huang, Y.; Chu, H. SAST: Learning semantic action-aware spatial-temporal features for efficient action recognition. *IEEE Access* **2019**, *7*, 164876–164886. [CrossRef]
9. Hara, K.; Kataoka, H.; Satoh, Y. Learning spatio-temporal features with 3d residual networks for action recognition. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 3154–3160.
10. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4489–4497.
11. Diba, A.; Fayyaz, M.; Sharma, V.; Karami, A.H.; Arzani, M.M.; Yousefzadeh, R.; Van Gool, L. Temporal 3d convnets: New architecture and transfer learning for video classification. *arXiv* **2017**, arXiv:1711.08200.
12. Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; Paluri, M. A closer look at spatiotemporal convolutions for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6450–6459.
13. Xie, S.; Sun, C.; Huang, J.; Tu, Z.; Murphy, K. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 305–321.
14. Qiu, Z.; Yao, T.; Mei, T. Learning spatio-temporal representation with pseudo-3d residual networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5533–5541.
15. Zolfaghari, M.; Singh, K.; Brox, T. Eco: Efficient convolutional network for online video understanding. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 695–712.

16. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; Van Gool, L. Temporal segment networks: Towards good practices for deep action recognition. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 20–36.
17. Lin, J.; Gan, C.; Han, S. Tsm: Temporal shift module for efficient video understanding. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7083–7093.
18. Wang, L.; Tong, Z.; Ji, B.; Wu, G. Tdn: Temporal difference networks for efficient action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Montreal, QC, Canada, 11–17 October 2021; pp. 1895–1904.
19. Liu, Z.; Luo, D.; Wang, Y.; Wang, L.; Tai, Y.; Wang, C.; Li, J.; Huang, F.; Lu, T. Teinet: Towards an efficient architecture for video recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 11669–11676.
20. Li, Y.; Ji, B.; Shi, X.; Zhang, J.; Kang, B.; Wang, L. Tea: Temporal excitation and aggregation for action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 909–918.
21. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
23. Luo, C.; Yuille, A.L. Grouped spatial-temporal aggregation for efficient action recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 5512–5521.
24. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Li, F.-F. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1725–1732.
25. Jiang, B.; Wang, M.; Gan, W.; Wu, W.; Yan, J. Stm: Spatiotemporal and motion encoding for action recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 2000–2009.
26. Shao, H.; Qian, S.; Liu, Y. Temporal interlacing network. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 11966–11973.
27. Liu, Z.; Wang, L.; Wu, W.; Qian, C.; Lu, T. Tam: Temporal adaptive module for video recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 13708–13718.
28. Wang, Z.; She, Q.; Smolic, A. Action-net: Multipath excitation for action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Montreal, QC, Canada, 11–17 October 2021; pp. 13214–13223.
29. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
30. Goyal, R.; Ebrahimi Kahou, S.; Michalski, V.; Materzynska, J.; Westphal, S.; Kim, H.; Haenel, V.; Fruend, I.; Yianilos, P.; Mueller-Freitag, M. The "something something" video database for learning and evaluating visual common sense. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5842–5850.
31. Carreira, J.; Zisserman, A. Quo vadis, action recognition? a new model and the kinetics dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Venice, Italy, 22–29 October 2017; pp. 6299–6308.
32. Zhou, B.; Andonian, A.; Oliva, A.; Torralba, A. Temporal relational reasoning in videos. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 803–818.
33. Liu, X.; Lee, J.-Y.; Jin, H. Learning video representations from correspondence proposals. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4273–4281.