



# Spatial-temporal channel-wise attention network for action recognition

Lin Chen<sup>1</sup> · Yungang Liu<sup>1</sup>  · Yongchao Man<sup>1</sup>

Received: 19 September 2020 / Revised: 10 December 2020 / Accepted: 24 February 2021 /

Published online: 20 March 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

Video-based action recognition has become a challenging task in computer vision and attracted extensive attention from the academic community. Most existing methods for action recognition treat all spatial or temporal input features equally, thus ignoring the difference of contribution provided by different features. To address this problem, we propose a spatial-temporal channel-wise attention network (STCAN) that is able to effectively learn discriminative features of human actions by adaptively recalibrating channel-wise feature responses. Specifically, the STCAN is constructed on a two-stream structure and we design a channel-wise attention unit (CAU) module. Two-stream network can effectively extract spatial and temporal information. Using the CAU module, the interdependencies between channels can be modelled to further generate a weight distribution for selectively enhancing informative features. The network performance of STCAN has been evaluated on two typical action recognition datasets, namely UCF101 and HMDB51, and comparable experiments have been performed to demonstrate the effectiveness of the proposed STCAN.

**Keywords** Action recognition · Channel-wise attention · Spatial-temporal information · Two-stream network

## 1 Introduction

Video-based action recognition has been widely investigated in the past decade, owing to its widespread applications in video surveillance, human action analysis, human-computer interaction, and so on. For action recognition in videos, the traditional methods are mainly based on hand-crafted features, such as 3D-Hessian, 3D-Harris, and improved dense trajectory (IDT) [2, 33, 59]. However, these methods are limited, to a large extent, by sampled interest regions when extracting video features. Motivated by the great promise of deep learning shown in image understanding, object detecting and target tracking, deep learning methods, such as two-stream CNNs and 3D CNNs, have been applied to the problem of

---

✉ Yungang Liu  
lygr@sdu.edu.cn

<sup>1</sup> School of Control Science and Engineering, Shandong University, 250061, Jinan, P.R. China

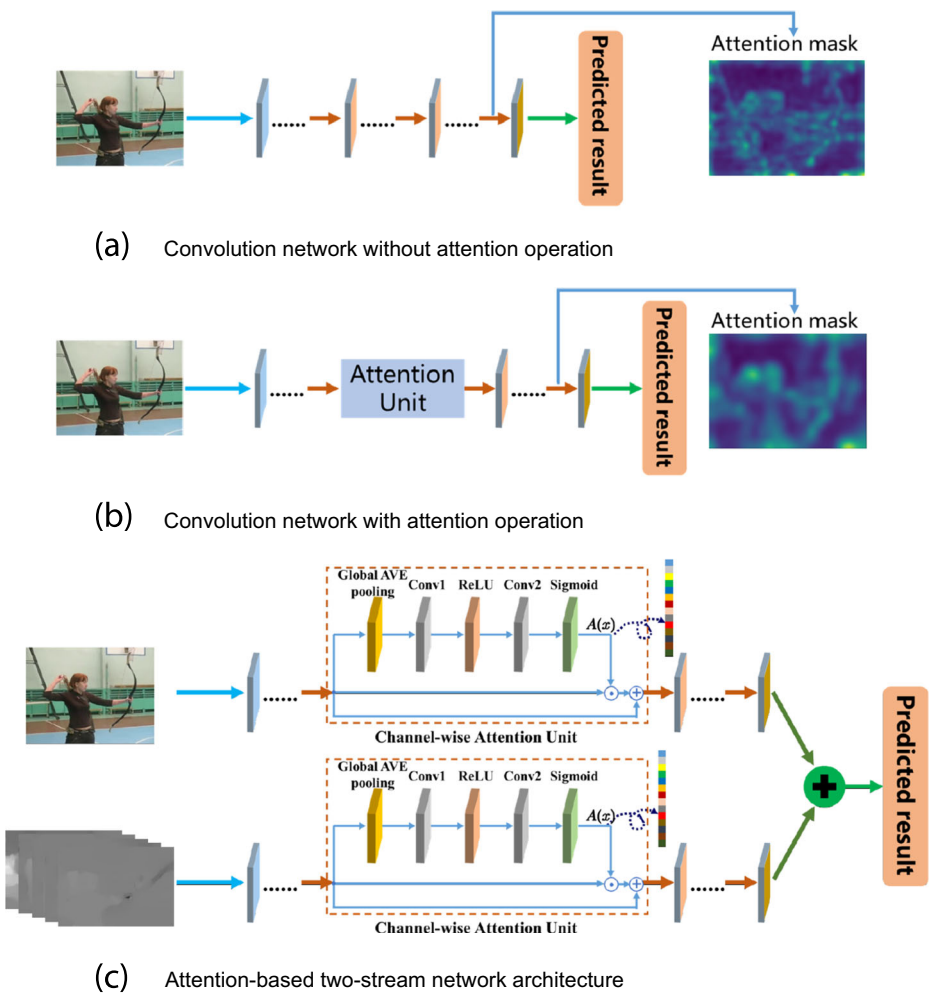
action recognition [8, 9, 11, 19, 25, 48]. Unlike traditional methods, deep learning methods are end-to-end frameworks and can automatically learn and extract video features from global images, showing great potential for video-based action recognition.

Two-stream CNN [37] has been proven to be a successful architecture, which contains a spatial stream and a temporal stream to extract appearance and motion features from videos separately. Two-stream CNN and most of its variants [10, 12, 19, 37, 49, 57] mainly focus on how to effectively fuse these two streams or how to operate segment-based sampling and effectively aggregate the different segment features, for example, the spatiotemporal distilled dense-connectivity network (STDDCN) [12] and the temporal segment network (TSN) [49]. 3D CNN has also been a promising method for action recognition, which extends 2D convolution with the temporal domain [5, 7, 13, 42, 56]. The Convolutional 3D (C3D) [42], Inflated 3D ConvNet (I3D) [5], and Temporal 3D ConvNet (T3D) [7] have achieved satisfying results on both UCF101 and HMDB51. However, most of these studies have not focused on how to selectively enhance informative features. Over the recent years, the attention mechanism has been widely exploited in many fields, such as natural language processing (NLP) [41] and image captioning [53], to help generate a more effective attention mask for the tasks. We argue that the relationship of feature channels plays an important role and emphasizing informative features selectively can help two-stream CNN yield superior performance for action recognition. Thus, we introduce attention mechanism into two-stream CNN to generate a new weight distribution of features in both spatial and temporal streams to better model the human actions [1, 26, 43]. The idea can be viewed in Fig. 1.

In this paper, a model named spatial-temporal channel-wise attention network (STCAN) is proposed based on TSN to recognize human action. Specifically, STCAN includes a flexible and effective module dubbed channel-wise attention unit (CAU), which can be embedded into CNNs expediently. Inspired by [17, 26], to selectively highlight the informative features while preserving the integrity of the original features, the CAU consists of two parts: a squeeze-and-excitation (SE) operation module and a shortcut connection module. The former can capture the channel-wise dependencies and generate a new weight distribution for the feature channels by feature compression operation, which can enhance informative features and suppress less relevant features selectively. Considering that the former module may destroy the discriminative properties of the original features, the latter is designed as a bypass structure, which connects the input and output of CAU. Notably, the CAU is simple and effective, and only increases slight computational complexity to networks. Furthermore, we argue that the network performance may have great relations with the embedding position and number of CAUs, thus it is crucial to explore the most suitable position and the most appropriate number of CAUs for superior performance.

Our main contributions are summarized as follows:

- A model named STCAN is proposed, which has the ability to model long-range temporal structure. And STCAN includes a simple but effective module named CAU, which can not only generate a channel-wise weight distribution to selectively enhance relevant features, but also maintain the discriminative properties of original features.
- The CAU proposed can be embedded into CNNs expediently with only a slight increase in computational complexity and enables end-to-end training. We explore several strategies of embedding the CAU for superior performance.
- Compared with the state-of-the-art approaches, STCAN achieves superior performance, which are evaluated on two challenging action recognition datasets, UCF101 and HMDB51.



**Fig. 1** a The attention mask generated by the network without attention unit. b The attention mask generated by the network with attention unit. c The channel-wise attention unit is embedded into basic two-stream architecture, which can enhance informative features for action recognition

The remaining of this paper is organized as follows. Section 2 discusses some related works, and Section 3 describes the proposed STCAN architecture in detail. In Section 4, the experimental details and results are provided. Section 5 gives some concluding remarks.

## 2 Review of related works

In this section, we will cover some works closely related to this paper, including human action recognition and attention mechanism.

## 2.1 Action recognition

Action recognition in videos has made significant progress in the past decade. State-of-the-art approaches can be roughly classified into two types: RGB-based ones and RGB-D-based ones.

RGB-based approaches can effectively extract appearance information and better represent action details, and they can be categorized into two types: ones using hand-crafted features and the others based on deeply-learned features. For the former approaches, many spatial-temporal feature detectors have been proposed, for example, 3D-Hessian [52], 3D-Harris [21], and improved dense trajectory (IDT) [46]. Additionally, to extract the appearance and dynamic information around interest regions, the histogram descriptors are developed, for example, Histogram of Oriented Optical Flow (HOF) [22], Histogram of Oriented Gradient (HOG) [22], Extended Speeded Up Robust Features (ESURF) [52], and Motion of Boundary History (MBH) [44]. Moreover, to form the feature representations, several encoding methods are applied. The classical encoding methods include Fisher Vector (FV), Bag of Visual Words (BoVW) [34], Vector of Locally Aggregated Descriptors (VLAD), and Multi-View Super Vector (MVSU) [4]. These traditional methods are simple to implement, but their ability to represent video features is limited by the sampled interest regions.

Unlike traditional methods, deep learning does not sample interest regions manually, instead, it can extract features autonomously through network training [3, 14]. There are two main research lines for action recognition in deep learning methods, one of which uses 3D CNNs to recognize the human action and the other is based on optical flow [5, 7, 13, 15, 30, 42, 56]. 3D CNN extends 2D convolution with the temporal domain, which can take the temporal information into consideration [5, 7, 13, 15, 42]. Carreira et al. [5] proposed two-stream I3D to learn seamless spatio-temporal feature extractors. Diba et al. [7] proposed T3D to introduce a temporal layer to efficiently model the temporal convolution kernel depths. He et al. [15] proposed a spatial temporal network architecture to model both local and global spatial-temporal features. One successful architect based on optical flow is two-stream CNN [10, 12, 33, 37, 49, 57], which uses optical flow images and RGB images respectively to extract motion and appearance features in parallel. Ng et al. [33] introduced the long short term memory (LSTM) module to two-stream CNN and could extract motion information more accurately from the images. Wang et al. [49] proposed a new method TSN to extract video features on the basis of two-stream CNN, which improved the accuracy dramatically. Feichtenhofer et al. [10] proposed a variety of methods to fuse the temporal and spatial networks, which further improved network performance. RGB-D-based approaches are not sensitive to illumination changes or dynamic camera views, and can accurately estimate the contour and skeleton of human body. The most typical ones are skeleton-based methods [35, 58]. Plizzari et al. [35] proposed a spatial-temporal transformer network to model dependencies between the joints. Zheng et al. [58] constructed a spatial and temporal graph convolution network to extract spatial-temporal features of skeleton for classification. However, most of the approaches lack the ability to distinguish the contribution of different features of the image.

Taking inspiration from the TSN architecture, we build a new model STCAN. In order to capture the channel-wise dependencies and enhance important features, we further introduce the attention mechanism.

## 2.2 Attention mechanism

Originated from human vision, the attention mechanism can assist the CNNs to focus on some specific features of the image and suppress less useful features [26]. We argue that through attention mechanism, CNNs can generate a weight distribution of image features, which can be applied to the original image to highlight the informative features.

Recently, the attention mechanism has been extensively studied in a lot of domains, such as human parsing, object tracking, image cropping, image captioning and so on [1, 17, 20, 23, 24, 26, 36, 41, 43, 47, 50, 51, 53, 55], and can also be applied in many practical applications [27–29]. For example, Tan et al. [41] designed an effective attention architecture for semantic role labeling. You et al. [53] combined bottom-up and top-down approaches through a semantic attention model to selectively focus on semantic concept proposals. Liao et al. [26] proposed a residual attention unit for 3D CNNs to highlight the foreground region for action recognition. Hu et al. [17] proposed a squeeze-and-excitation block to model channel-wise interdependencies to improve network performance. Li et al. [24] proposed a spatiotemporal attention to learn the discriminative feature representation. Shen et al. [36] designed an effective hierarchical attention Siamese network for object tracking. Zhang et al. [55] proposed a moving foreground attention model to pay more attention to the foreground targets. In this paper, we introduce attention mechanism into two-stream CNN and design an attention model CAU. Different from the previous works, the module CAU can not only emphasise informative features but also retain the discriminative properties of the original features.

In summary, inspired from TSN architecture and attention mechanism, we propose a new method STCAN, which can model long-term temporal structure and includes an attention unit CAU. The CAU, containing a SE operation module and a shortcut connection module, has the abilities to model the interdependencies among all the channels, selectively highlight the informative features, and retain the discriminative properties of the original features.

## 3 Spatial-temporal channel-wise attention network

In this section, we make a detailed description of the proposed STCAN. Firstly, the STCAN framework is introduced in Section 3.1. Then, the architecture of the CAU is presented in Section 3.2. Finally, the strategies to embed CAU are discussed in Section 3.3.

### 3.1 Spatial-temporal channel-wise attention network framework

The STCAN architecture is built on the basis of TSN [49]. Similarly, STCAN has the capacity to incorporate long-term temporal information into the learning of action models and learns action models efficiently with the segment-based sampling and aggregation scheme. Specifically, given a video  $V$ , STCAN firstly splits the video into  $K$  segments  $\{S_1, S_2, \dots, S_K\}$  with the same length, from which to randomly sample the snippets  $\{I_1, I_2, \dots, I_K\}$ . Each snippet  $I_i$  could be 1 frame for RGB or 5 frames for optical flow. Then, the two patterns of frames are treated as inputs of the networks to extract features and make clip-level recognition predictions. Finally, the video-level prediction can be obtained by fusing these clip-level recognition predictions. The formula can be represented as follows:

$$Net(I_1, I_2, \dots, I_K) = \mathcal{H}(\mathcal{G}(\mathcal{F}(I_1; W), \mathcal{F}(I_2; W), \dots, \mathcal{F}(I_K; W))). \quad (1)$$

where  $\mathcal{F}(\cdot; W)$  denotes the ConvNet function with parameters  $W$ , which takes  $I_i$  as input and produces scores over all the classes.  $\mathcal{G}(\cdot)$  refers to segmental consensus function to combine the outputs of multiple snippets.  $\mathcal{H}(\cdot)$  refers to the Softmax function to predict the probability of each class for the whole video.

---

### Algorithm 1 Training process.

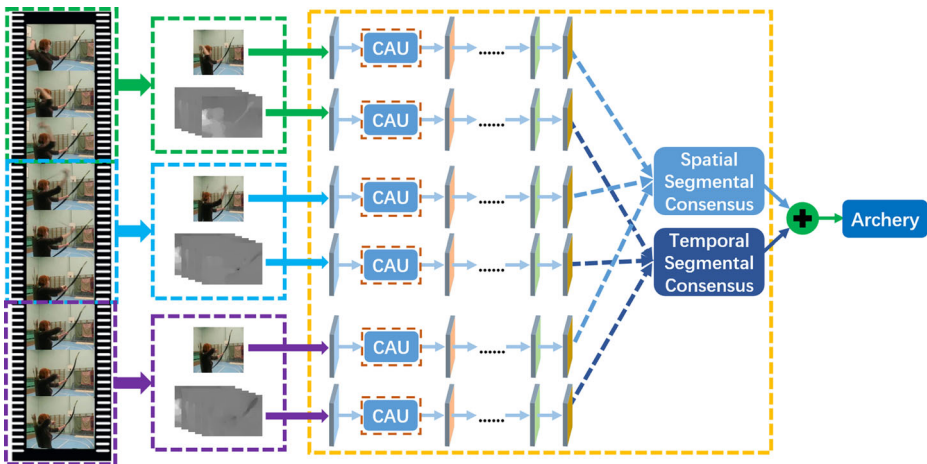
---

- 1: **Input:**  $N$  videos (each has been divided into  $K$  segments  $\{S_1, S_2, \dots, S_K\}$ ) with  $n$  classes.

The maximum iteration number  $Iter$ .

- 2: **Output:** Parameters of the model STCAN.
  - 3: Initialize the model with parameters pre-trained on ImageNet or Kinetics and randomly initialize the fully connected layers.
  - 4: Initialize the iteration number  $i = 0$ .
  - 5: **repeat**
  - 6: ① Forward propagation and get clip-level scores  $\mathcal{F}(I_i; W)$ .
  - 7: ② Get the video prediction  $\mathcal{H}(\cdot)$  according to (1).
  - 8: ③ Back Propagation to optimize network parameters.
  - 9: ④  $i = i + 1$ .
  - 10: **until**  $i = Iter$ .
- 

The model STCAN contains a simple but effective module CAU in both spatial and temporal streams, which can selectively highlight the informative features and preserve the discriminative properties of the original features. Additionally, the deep CNN architectures (e.g. BN-Inception and Inception-v3 [31, 40]) are employed as the backbone. The ConvNet parameters are transferred from pre-trained models and the learning procedure is sketched in Algorithm 1. Figure 2 illustrates the basic framework of the model STCAN.



**Fig. 2** The basic framework of Spatial-Temporal Channel-wise Attention Network. The sample video is divided into  $K$  segments ( $K = 3$  for example), from each to randomly sample a short snippet (1 frame for RGB or 5 frames for optical flow). The CAU is embedded into both streams to selectively highlight the informative features. The snippet scores are fused to yield a video-level prediction

### 3.2 Architecture of the channel-wise attention unit

Figure 3 shows the architecture of the CAU. Inspired by [17, 26], the CAU contains two parts: the shortcut connection module and the SE operation module. The former, connecting the input and output of the CAU, can prevent the loss of discriminative properties of the original features. The latter makes full use of the channel global information to obtain the channel-wise dependencies and adaptively recalibrate the feature responses, which works in the squeeze-and-excitation way.

The squeeze and excitation operations mentioned above are implemented by five layers, of which the first layer is to fulfill the squeeze operation and the remaining layers are designed to complete the excitation operation. In this paper, the global average pooling, a simple aggregation technique, is exploited as the first layer. The remaining four layers consist of two convolutional layers (namely Conv1 and Conv2), one ReLU layer and one Sigmoid layer. The function of these two convolutional layers is to fuse the features of each channel, and the ReLU layer makes this module capable of learning the non-linearity between channels. In addition, to ensure that multiple channels can be emphasized, we opt Sigmoid layer to learn non-mutually exclusive relationships.

Consider the input  $x \in \mathbb{R}^{N \times C \times W \times H}$ , where  $N$  refers to the number of training samples in each batch,  $C$  refers to the number of channels, and  $W$  and  $H$  represent the width and the height of the sampled image. Then, we can obtain the channel statistic  $z(x) \in \mathbb{R}^{N \times C \times 1 \times 1}$  by operating the global average pooling on individual feature channel of each training sample. The  $c$ -th channel on  $n$ -th sample of  $z(x)$ , denoted as  $z_{cn}(x) \in \mathbb{R}^{1 \times 1}$ , can be formulated as follows:

$$z_{cn}(x) = \frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H x_{cn}(i, j), \tag{2}$$

where  $x_{cn}(i, j)$  is the value at position  $(i, j)$  of the  $c$ -th channel on  $n$ -th sample of the input  $x$ . Furthermore, the output of excitation operation  $A(x) \in \mathbb{R}^{N \times C \times W \times H}$  can be written as follows:

$$A(x) = \sigma(\mathcal{F}_2(\delta(\mathcal{F}_1(z(x); W_1)); W_2)), \tag{3}$$

where  $\sigma$  refers to the Sigmoid function, and  $\mathcal{F}(\cdot; W)$  denotes the convolutional network function with parameters  $W$ . More specifically,  $\mathcal{F}_1(\cdot; W_1) \in \mathbb{R}^{N \times \frac{C}{r} \times 1 \times 1}$  denotes the Conv1 function and  $\mathcal{F}_2(\cdot; W_2) \in \mathbb{R}^{N \times C \times 1 \times 1}$  denotes the Conv2 function,  $\delta$  refers to the ReLU

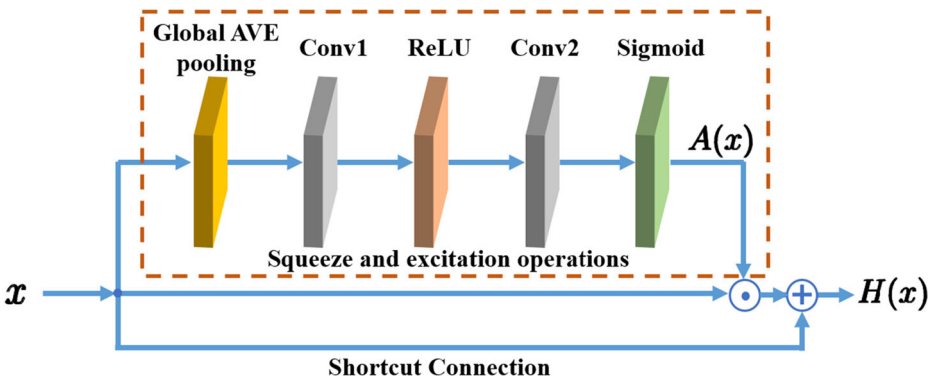


Fig. 3 The architecture of Channel-wise Attention Unit

function. The Conv1 layer is a dimensionality-reduction layer and the reduction ratio  $r$  is set to 16. The Conv2 layer is a dimensionality-increasing layer, which makes the output dimension return to the channel dimension of input  $x$ . Note that  $A(x)$  is a scalar, which represents the weights of channel feature maps of the input  $x$ , and is obtained by learning from the two convolutional layers and ReLU layer.

Thus, the final output of the CAU,  $H(x) \in \mathbb{R}^{N \times C \times W \times H}$ , can be represented as follows:

$$H(x) = x \oplus (A(x) \odot x), \tag{4}$$

where  $\oplus$  represents element-wise addition and this operation represents the shortcut connection, and  $\odot$  denotes element-wise multiplication, which means that a weight coefficient is added to the feature maps of each channel. For further explanation, the overview of squeeze and excitation operations is shown in Fig. 4.

Remarkably, the purpose of CAU is twofold: (a) It can model the channel-wise interdependencies and adaptively recalibrate feature responses to selectively enhance informative features and suppress less relevant features. (b) It can prevent the loss of discriminative properties of the original features through a bypass structure. The proposed CAU is flexible and can be embedded into CNNs conveniently. Nevertheless, simply applying the CAU may not achieve satisfactory optimization performance, so how to embed the CAU into networks is worth further consideration.

### 3.3 Embedding strategies

In this subsection, the strategies of embedding CAU into CNNs are investigated. Since the network performance may have great relations with the embedding positions and number of CAUs, we design several different strategies to seek the most suitable position and the most appropriate number of CAUs for superior performance.

As mentioned above, Inception networks are employed as the network backbone, which contain a sequence of convolution Inception blocks. Taking Inception-v3 for example, we select the first four convolution Inception blocks (named Block1, Block2, Block3 and Block4) for experiment and exploit Kinetics dataset for pre-training. First, only one CAU

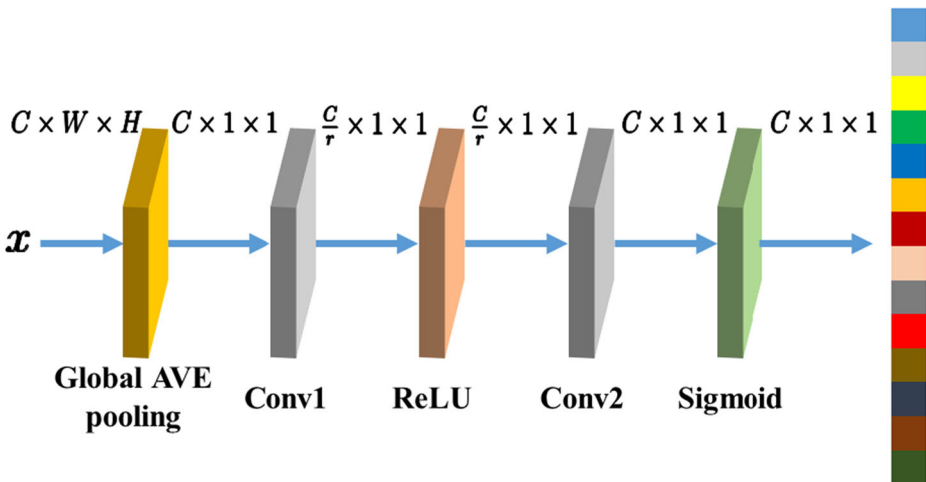


Fig. 4 The overview of squeeze and excitation operations



is embedded into the Inception-v3 for experiment, following Block1, Block2, Block3 and Block4 individually. For instance, Fig. 5 shows the structure when a CAU is embedded following Block1. Then, multiple CAUs are embedded following the chosen convolution Inception blocks for further experiments. By this method, we can not only find the appropriate positions to embed CAU, but also get whether multiple CAUs can further optimize the network performance.

## 4 Experiments

In this section, the datasets involved and implementation details are firstly introduced. Several good practices are then investigated to prevent overfitting in the process of training. Finally, we carry out experiments to evaluate the effectiveness of STCAN on two challenging datasets.

### 4.1 Datasets and implementation details

#### 4.1.1 Datasets

Two action recognition datasets are applied to verify the effectiveness of the proposed model STCAN, namely HMDB51 and UCF101. HMDB51 [18] consists of 51 action categories and 6766 short video clips with the resolution of  $320 \times 240$ . HMDB51 is divided into three sub-datasets (called split1, split2 and split3), and each contains 3570 clips for training and 1530 clips for testing. The experimental results are validated on these three sub-datasets, and we take the average value of these three results as the final accuracy on HMDB51.

UCF101 [39] consists of 101 action categories and 13320 short video clips with the resolution of  $320 \times 240$ . This dataset is also divided into three sub-datasets. Similarly, the experimental results of UCF101 are verified on three sub-datasets and we take the average value of these three results as the final accuracy on UCF101.

We downloaded HMDB51 dataset at <https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/>, and download UCF101 dataset at <https://www.crcv.ucf.edu/data/UCF101/UCF101.rar>. To extract RGB images and optical flow images of all frames of the datasets, we choose the OpenCV and DenseFlow toolkits compiled by GPU. Moreover, the TV- $L^1$  (Total Variation- $L^1$ ) [54] algorithm is used to calculate the optical flow.

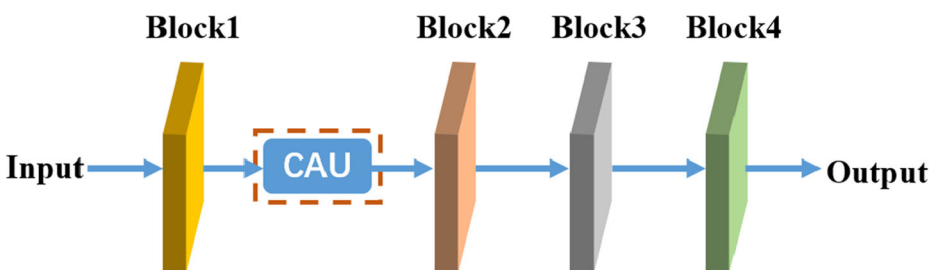


Fig. 5 The overview when a CAU is embedded following Block1

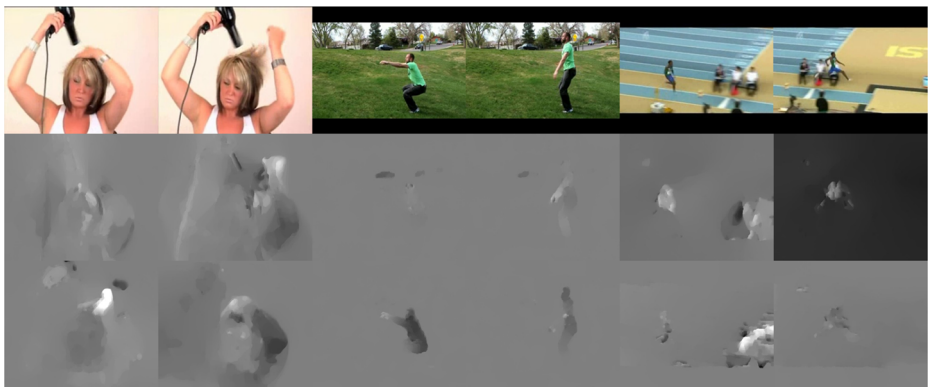
## 4.1.2 Implementation details

From the perspective of network inputs, a single RGB frame and 5 stacked consecutive optical flow frames that randomly sampled from each video clip are treated as inputs of the spatial and temporal streams respectively. Figure 6 shows the RGB images and their corresponding optical flow images. To extract the optical flow frames, we take the  $TV-L^1$  [54] optical flow algorithm. Specifically, it takes about 18 hours to extract frame images (RGB images, horizontal and vertical optical flow images) of all videos in UCF101, and takes about 5 hours in HMDB51.

In the training process, four data augmentation methods, namely random clipping, corner clipping, horizontal flipping, and scale jittering [38], are adopted for pre-processing. The clipping size of scale jittering is randomly determined according to the jittering ratios, which are set to 1, 0.875, 0.75, and 0.66. Then, the clipping areas are uniformly adjusted to  $224 \times 224$  for network training. The stochastic gradient descent (SGD) algorithm is used to learn the network parameters. The batch size is set to 32 and the momentum is set to 0.9. BN-Inception and Inception-v3 are employed as the backbone. In addition, pre-trained models from Kinetics or ImageNet are exploited to initialize the network weights. Moreover, the number of segments is set to 3. Since the average pooling can model multiple segments jointly and capture visual information from the entire video, we apply average pooling as aggregation function to calculate the prediction scores of the segments. The integration ratio of spatial and temporal streams is set to 1:1.6.

For the temporal stream parameters on UCF101, the total number of iterations is set to 9000. The learning rate is initialized as 0.001, and reduces to its 0.1 times after 5000 and 8000 iterations. For the spatial stream, the total number of iterations is set to 4000. The learning rate is initialized as 0.0005, and changes to its 0.1 times for every 1500 iterations. On HMDB51, the overall process is similar as that of UCF101. For temporal stream, the total number of iterations is set to 5000. The learning rate is initialized as 0.001, and reduces to its 0.1 times after 2500 and 3500 iterations. For the spatial stream, the total number of iterations is set to 3000. The learning rate is initialized as 0.0005, and changes to its 0.1 times for every 1000 iterations.

In order to prevent overfitting [32], we adopt two methods: gradient clipping and dropout [16]. The parameter of gradient clipping (clip\_gradient) is set to 20 in temporal stream



**Fig. 6** The RGB images and their corresponding optical flow images. The first row is the RGB images, and the second and third rows are the optical flow images in  $x$  and  $y$  component

and set to 40 in spatial stream to make the gradient restrained in a fixed range to prevent gradient explosion during training. Moreover, the dropout layers are added between the global pooling layer and the FC layer of spatial and temporal streams, and the parameters (dropout\_ratio) are set to 0.7 and 0.8 respectively.

The experiments are carried out in the Ubuntu 18.04 with RTX 2080 TI and the proposed model is built on Caffe. Since only a single RTX 2080 TI is used, the network training is time-consuming. For instance, when training BN-Inception network on UCF101, the total training time of the temporal stream is about 20 hours, and the spatial stream about 2.5 hours.

## 4.2 Experimental results

### 4.2.1 Embedding experiments of CAU

In this subsection, we will explore the appropriate position and number of CAUs for spatial and temporal stream networks, and experiments are carried out on HMDB51. Meanwhile, Inception-v3 architecture is employed as the backbones.

Since the bottom features and top features contain different information, the former represent low-level texture information and the latter represent semantic information, it is worth exploring which features are appropriate for CAU. To this end, we evaluate the performance of embedding a single CAU following different convolution Inception blocks, such as Block1, Block2, Block3 and Block4. Additionally, we argue that embedding multiple CAUs into multiple positions may achieve satisfactory optimization performance, thus it is worth pondering whether multiple CAUs can further increase the network performance.

Tables 1 and 2 show the spatial and temporal stream results of embedding different numbers and different positions of CAUs into networks. It figures out that embedding CAUs into Inception-v3 can significantly improve the accuracies in both spatial and temporal streams, and inserting a single CAU following Block1 can get the best performance. Note that embedding multiple CAUs dose not further improve performance as expected. Conversely, the accuracy decreases as the number of CAU increases. From the results in Tables 1 and 2, we deduce that the bottom features can fit the CAU better, which can be explained that low-level features contain rich texture information and are more conducive to generating attention mask. Otherwise, high-level features are more abstract and may lead to

**Table 1** Spatial stream results of embedding different numbers and different positions of the CAUs into Inception-v3 on HMDB51 split1

Embedding strategy	Accuracy
Baseline	57.06%
Block1	61.76%
Block2	60.85%
Block3	60.78%
Block4	60.72%
Block1 + Block2	60.46%
Block2 + Block3	60.26%
Block3 + Block4	59.54%
Block2 + Block3 + Block4	59.48%
Block1 + Block2 + Block3	59.28%
Block1 + Block2 + Block3 + Block4	58.63%

**Table 2** Temporal stream results of embedding different numbers and different positions of the CAUs into Inception-v3 on HMDB51 split1

Embedding strategy	Accuracy
Baseline	69.61%
Block1	72.75%
Block2	72.48%
Block3	72.35%
Block4	71.90%
Block1 + Block2	71.76%
Block2 + Block3	71.37%
Block3 + Block4	71.24%
Block1 + Block2 + Block3	70.98%
Block2 + Block3 + Block4	70.92%
Block1 + Block2 + Block3 + Block4	70.39%

disharmony with CAU. Additionally, if the bottom features are modulated repeatedly by CAU in multi-layers, the attention mask already processed may be destroyed and lose some important information.

#### 4.2.2 Comparison of pre-trained datasets

As the two-stream CNNs take images as inputs, it is feasible to apply the models trained on Kinetics or ImageNet as initialization. Additionally, experiments are performed to investigate the impact of pre-trained datasets on recognition performance. Furthermore, BN-Inception is employed and the parameters can be optimized with back propagation algorithm. Table 3 gives comparison results of different pre-trained datasets on UCF101.

As shown in Table 3, exploiting the model pre-trained on Kinetics can obtain better results compared with that pre-trained on ImageNet, which indicates that the capacity of dataset can affect the recognition performance. Moreover, we argue that the generalization ability and the recognition performance of the network may improve with the increase of the dataset capacity. Therefore, the model pre-trained on Kinetics is exploited as initialization for the subsequent experiments by default, unless stated otherwise.

**Table 3** Comparison results of BN-Inception pre-trained on ImageNet or Kinetics on UCF101

CNN architecture	Stream	UCF101		
		Split1	Split2	Split3
BN-Inception(ImageNet)	Spatial	86.02	84.96	84.55
	Temporal	87.62	90.18	91.28
	Two	93.52	94.16	94.38
BN-Inception(Kinetics)	Spatial	86.40	87.18	85.88
	Temporal	90.75	93.77	93.34
	Two	94.72	95.51	95.26

### 4.2.3 Evaluation of proposed STCAN

As stated above, embedding the single CAU following Block1 will obtain the best performance, thus this strategy is used for subsequent experiments. In addition, we adopt the transfer learning method and exploit the model trained on Kinetics as initialization, which can effectively solve the problem of insufficient training samples. The experiment performance are evaluated on UCF101 and HMDB51, and BN-Inception and Inception-v3 architectures are employed. The comparison results of STCAN and the baseline(Not embedded with CAU) are shown in Table 4, and the detailed accuracies are summarized in Tables 5 and 6.

As shown in Table 4, STCAN yields superior performance than baseline on both UCF101 and HMDB51. Furthermore, by comparing the accuracies in Tables 5 and 6, we can make the following three observations. *First*, both the spatial and temporal networks embedded with CAU perform better than those without CAU, which verifies the feasibility of the method proposed. Particularly, we find an interesting thing that the improvement generated by spatial network is greater than that of temporal network. We suspect it may be because that the useless information, which is mostly high-frequency, often appears in RGB images but not optical flow images. Also, it may be because the less useful information, such as background parts, in the optical flow images are often static. *Second*, when embedding CAU, BN-Inception can achieve comparable or even better performance than Inception-v3, which may indicate that CAU has a more obvious improvement on BN-Inception. *Third*, the improvement on HMDB51 is more significant than that on UCF101. We suggest that there may be two reasons: (a) From the perspective of dataset capacity, the actual training dataset is not infinite. CAU is designed to generate a weight distribution to selectively enhance informative features. When the training data becomes sufficient, the generalization ability of the network will be improved, and thus the effect of CAU will become less significant. (b) From the perspective of video clip content, many clips of different categories on HMDB51 have the same scene and embedding CAU may allow the network to focus more on the action features.

### 4.2.4 Evaluation of proposed module CAU

To evaluate the influence of the proposed module CAU, we perform a comparative experiment of embedding a SE block [17] into the same position of BN-Inception or Inception-V3 and the results are shown in Table 7. The baseline is carried out on BN-Inception or Inception-V3 with no module embedded in. As shown in Table 7, the module proposed can obtain better results, which demonstrates the effectiveness of the proposed module CAU.

In addition, to evaluate the computational complexity that CAU increases, we compare the models with FLOPs (floating point operations) and report the results in Table 8. From

**Table 4** Comparison results of STCAN and Baseline on UCF101 and HMDB51

Dataset	BN-Inception		Inception-V3	
	Baseline	STCAN	Baseline	STCAN
UCF101	95.16	96.18	95.46	96.13
HMDB51	71.74	75.17	72.26	74.45

**Table 5** Comparison results of STCAN and Baseline with BN-Inception

Dataset	Structure	Stream	Split1	Split2	split3
UCF101	Baseline	Spatial	86.40	87.18	85.88
		Temporal	90.75	93.77	93.34
		Two	94.72	95.51	95.26
	STCAN	Spatial	90.27	90.80	89.90
		Temporal	91.44	94.92	93.98
		Two	95.88	96.65	96.01
HMDB51	Baseline	Spatial	55.82	52.29	55.29
		Temporal	69.41	69.93	70.26
		Two	72.94	70.98	71.31
	STCAN	Spatial	61.63	60.26	59.08
		Temporal	71.76	71.63	71.90
		Two	75.82	74.84	74.84

**Table 6** Comparison results of STCAN and Baseline with Inception-v3

Dataset	Structure	Stream	Split1	Split2	Split3
UCF101	Baseline	Spatial	87.33	87.63	87.93
		Temporal	91.70	94.32	93.58
		Two	94.76	95.85	95.76
	STCAN	Spatial	90.55	89.89	90.21
		Temporal	91.74	94.40	93.86
		Two	95.53	96.37	96.50
HMDB51	Baseline	Spatial	57.06	56.01	55.36
		Temporal	69.61	70.98	70.92
		Two	73.20	71.37	72.22
	STCAN	Spatial	61.76	59.48	59.48
		Temporal	72.75	71.50	72.55
		Two	75.36	73.14	74.84

**Table 7** Comparison of different embedding modules

Structure	BN-Inception		Inception-V3	
	Spatial	Temporal	Spatial	Temporal
Baseline	55.82	69.41	57.06	69.61
SE block	60.46	71.18	60.92	71.57
CAU	61.63	71.76	61.76	72.75

**Table 8** Comparison on complexity of Inception networks with CAU

Method	GFLOPs	Params(M)
BN-Inception (RGB+Flow)	4.34614	20.698
Inception-v3 (RGB+Flow)	5.79717	43.987
BN-Inception + CAU (RGB+Flow)	4.34621	20.766
Inception-v3 + CAU (RGB+Flow)	5.79719	44.004

the results, we can infer that embedding CAU into BN-Inception and Inception-v3 will only lead to a small amount of computation while the performance increases significantly.

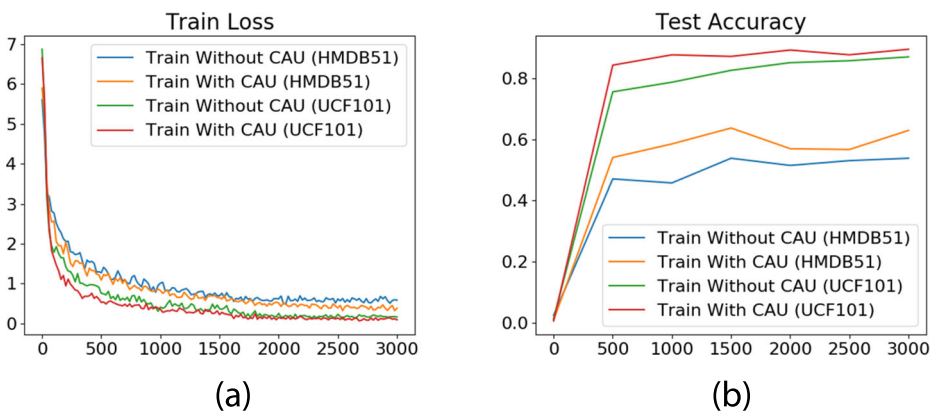
Moreover, to provide a clearer comparison of the influence of CAU on model training and testing, the sample training and testing curves for runs of the spatial stream architectures with and without CAU are depicted in Fig. 7. We observe that the network embedded with CAU can produce a steady improvement during both training and testing process, and this result is verified in both HMDB51 and UCF101 datasets.

#### 4.2.5 Comparison of all categories on the two datasets

Furthermore, we compare the accuracies of all action categories on the two dataset (UCF101 and HMDB51). We evaluate the performance of Inception-v3 with or without CAU on UCF101 and HMDB51, and the comparison detail results are shown in Figs. 8 and 9 respectively. It is worth pointing out that the approach proposed performs better in almost all categories, taking UCF101 for example, especially HandstandWalking, JumpingJack, JumpRope, etc. However, it performs worse in the categories, such as ThrowDiscus, which may be due to the similarity between the category Shotput and ThrowDiscus.

#### 4.2.6 Comparison with the state-of-the-art

In this subsection, We compare the method STCAN with several state-of-the-art approaches, including the traditional methods [34, 45, 46], the two-stream methods [10, 33, 37, 49], and the 3D convolution methods [5, 13, 15, 24, 42]. The comparison results are summarized in



**Fig. 7** Training and testing curves of Spatial stream architectures with and without CAU. (a) Training results for the split1 of two datasets. (b) Testing results for the split1 of two datasets

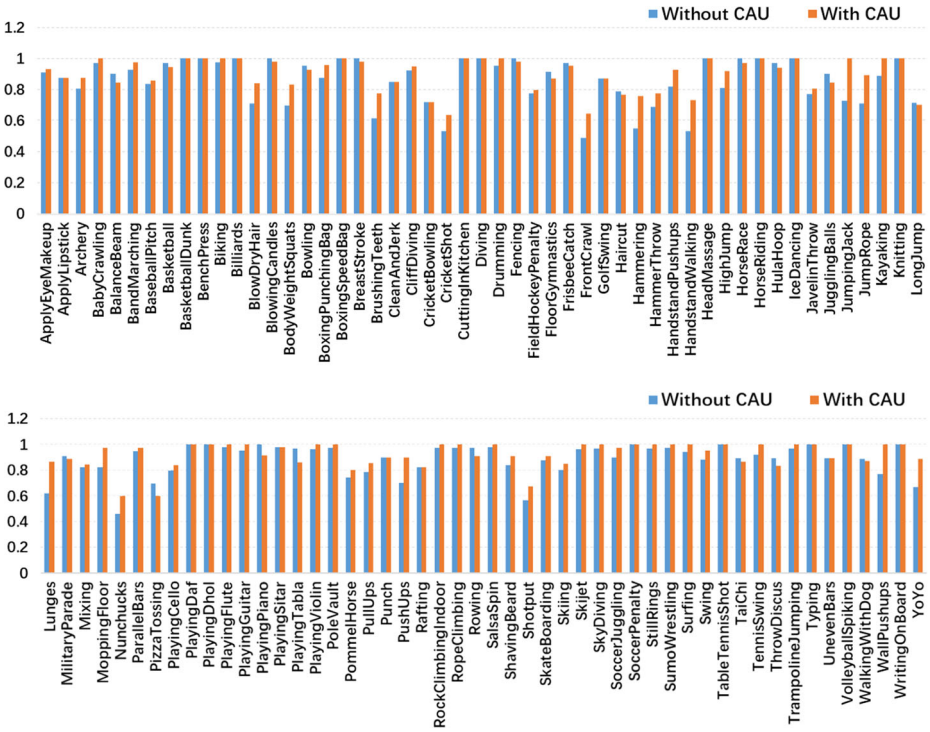


Fig. 8 Comparison results of all categories on UCF101 (split1)

Table 9. The best accuracies are 88.30% (UCF101) and 61.70% (HMDB51) in traditional methods, and 98.40% (UCF101) and 81.40% (HMDB51) in deep learning methods.

We observe that the method STCAN outperforms the traditional methods and most 2D convolution methods on both datasets. Moreover, our results significantly outperform the two-stream baseline [37] by 8.18 and 15.77 percent. When compared with 3D CNN methods, the method STCAN achieves comparable performance. Specifically, I3D

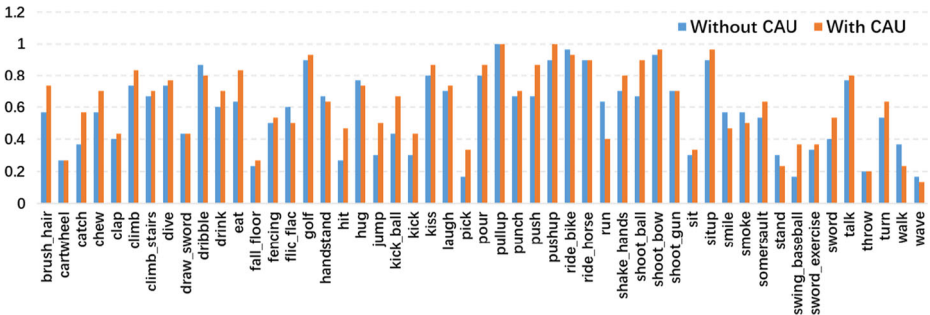


Fig. 9 Comparison results of all categories on HMDB51 (split1)



**Table 9** Comparison results with current state-of-the-art methods on UCF101 and HMDB51

Method	UCF101	HMDB51
IDT+FV [46]	85.90	57.20
IDT+BoVW [34]	87.90	61.10
MoFAP [45]	88.30	61.70
Two-stream [37]	88.00	59.40
Two-stream+LSTM [33]	88.60	–
Two-stream+IDT [10]	93.50	69.20
TSN(3 seg) [49]	94.20	70.70
TSN(7 seg) [49]	94.90	71.00
C3D (RGB) [42]	82.30	–
I3D (RGB) [5]	95.10	74.30
I3D (RGB+Flow) [5]	97.80	80.90
STA-ResNeXt-101(64f) [24]	95.50	74.10
STA-MARS+RGB+Flow [24]	98.40	81.40
StNet+IRv2(T=7) [15]	95.70	–
3D Resnext-101+64f+RAU [13]	95.27	71.83
Ours(Inception-v3)	96.13	74.45
Ours(BN-Inception)	96.18	75.17

(RGB+ Flow) and STA-MARS+RGB+Flow perform better than STCAN, probably because that I3D can learn seamless spatio-temporal feature extractors through its 3D filters and STA-MARS+RGB+Flow can efficiently learn the discriminative feature representation of actions. MARS (Motion-Augmented RGB Stream) [6] is a 3D CNN model that trained using a linear combination of feature-based loss and standard cross-entropy loss, and its performance is improved by 0.2% and 1.7% respectively to 98.40% (UCF101) and 81.40% (HMDB51) with the help of module STA [24]. For further comparison, the complexity of these methods is shown in Table 10. From the observation, we can find that the method proposed has smaller computational complexity. Remarkably, our approach improves the recognition performance, and the accuracies are up to 96.18% (UCF101) and 75.17% (HMDB51).

**Table 10** Comparison on complexity with other methods

Method	GFLOPs
C3D (RGB) [42]	38.5
STA-ResNeXt-101(64f) [24]	38.502
StNet+IRv2(T=7) [15]	123
I3D (RGB) [5]	111.5
I3D (RGB+Flow) [5]	214.2
Inception-v3 + CAU (Ours)	5.79719
BN-Inception + CAU (Ours)	4.34621

## 5 Conclusion

In this paper, a new method, named STCAN, has been proposed, which can model long-range temporal structure. Specifically, STCAN includes a simple and effective module named CAU, which can be embedded into the spatial and temporal streams expediently. The CAU can not only model channel-wise interdependencies to adaptively recalibrate the feature responses, but also prevent the loss of discriminative properties of the original features. Moreover, the model proposed can be trained end-to-end and has smaller computational complexity compared with the ones that achieve comparable results. Since this method has the ability to selectively highlight the informative features, future work will extend the ideas of this paper to other interesting domains of recognition, such as video surveillance and target tracking.

**Acknowledgements** The research is supported by the National Natural Science Foundations of China (62033007, 61873146, 61973186, 61821004 and 62073192), the Key and Development Plan of Shandong Province (Grant No. 2019JZZY010433) and the Taishan Scholars Climbing Program of Shandong Province.

## References

1. Anderson P, He X, Buehler C, Teney D, Johnson M (2018) Bottom-up and top-down attention for image captioning and visual question answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6077–6086
2. Beddiar DR, Nini B, Sabokrou M, Hadid A (2020) Vision-based human activity recognition: a survey. *Multimed Tools Appl*. <https://doi.org/10.1007/s11042-020-09004-3>
3. Bianco S, Ciocca G, Cusano C (2016) CURL: Image classification using co-training and unsupervised representation learning. *Comput Vis Image Underst* 145:15–29
4. Cai Z, Wang L, Peng X, Qiao Y (2014) Multi-view super vector for action recognition. In: Proceedings IEEE conference on computer vision and pattern recognition, pp 596–603
5. Carreira J, Zisserman A (2017) Quo vadis, action recognition? a new model and the kinetics dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4724–4733
6. Crasto N, Weinzaepfel P, Alahari K, Schmid C (2019) MARS: Motion-augmented RGB stream for action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7874–7883
7. Diba A, Fayyaz M, Sharma V, Karami AH, Arzani MM, Yousefzadeh R (2017) Temporal 3d convnets: new architecture and transfer learning for video classification. arXiv:1711.08200
8. Dong X, Shen J (2018) Triplet loss in siamese network for object tracking. In: Proceedings of the European conference on computer vision (ECCV), pp 472–488
9. Dong X, Shen J, Wu D, Guo K, Jin X, Porikli F (2019) Quadruplet network with one-shot learning for fast visual object tracking. *IEEE Trans Image Process* 28(7):3516–3527
10. Feichtenhofer C, Pinz A, Zisserman A (2016) Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1933–1941
11. Guo S, Qing L, Miao J, Duan L (2019) Action prediction via deep residual feature learning and weighted loss. *Multimed Tools Appl* 79(7-8):4713–4727
12. Hao W, Zhang Z (2019) Spatiotemporal distilled dense-connectivity network for video action recognition. *Pattern Recognit* 92:13–24
13. Hara K, Kataoka H, Satoh Y (2018) Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet? In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6546–6555
14. He P, Jiang X, Su T, Li H (2018) Computer graphics identification combining convolutional and recurrent neural networks. *IEEE Signal Proc Lett* 25(9):1369–1373
15. He D, Zhou Z, Gan C, Li F, Liu X, Li Y, Wang L, Wen S (2019) StNet: Local and global spatial-temporal modeling for action recognition. In: Proceedings of the AAAI conference on artificial intelligence, pp 8401–8408

16. Hinton GE, Srivastava N, Krizhevsky A (2012) Improving neural networks by preventing co-adaptation of feature detectors. arXiv:[1207.0580v1](https://arxiv.org/abs/1207.0580v1)
17. Hu J, Shen L, Albanie S, Sun G, Wu E (2020) Squeeze-and-excitation networks. *IEEE Trans Pattern Anal Mach Intell* 42(8):2011–2023
18. Kuehne H, Jhuang H, Garrote E, Poggio TA, Serre T (2011) HMDB51: A large video database for human motion recognition. In: *Proceedings of the IEEE international conference on computer vision*. IEEE, pp 2556–2563
19. Kwon H, Kim Y, Lee J, Cho M (2018) First person action recognition via two-stream ConvNet with long-term fusion pooling. *Pattern Recognit Lett* 112:161–167
20. Lai Q, Wang W, Sun H, Shen J (2020) Video saliency prediction using spatiotemporal residual attentive networks. *IEEE Trans Image Process* 29:1113–1126
21. Laptev I (2005) On space-time interest points. *Int J Comput Vis* 64(2-3):107–123
22. Laptev I, Marszalek M, Schmid C, Rozenfeld B (2008) Learning realistic human actions from movies. In: *Proceedings IEEE conference on computer vision and pattern recognition*, pp 1–8
23. Li T, Liang Z, Zhao S, Gong J, Shen J (2020) Self-learning with rectification strategy for human parsing. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 9260–9269
24. Li J, Liu X, Zhang W, Zhang M, Song J, Sebe N (2020) Spatio-temporal attention networks for action recognition and detection. *IEEE Trans Multimed* 22(11):2990–3001
25. Liang Z, Shen J (2020) Local semantic siamese networks for fast tracking. *IEEE Trans Image Process* 29:3351–3364
26. Liao Z, Hu H, Zhang J, Yin C (2019) Residual attention unit for action recognition. *Comput Vis Image Underst* 189:102821
27. Lv Z, Halawani A, Feng S, Li H, Réhman S (2013) Multimodal hand and foot gesture interaction for handheld devices. In: *Proceedings of the 21st ACM international conference multimedia*, pp 621–624
28. Lv Z, Halawani A, Feng S, Réhman S, Li H (2015) Touch-less interactive augmented reality game on vision-based wearable device. *Personal Ubiquit Comput* 19(3-4):551–567
29. Lv Z, Penades V, Blasco S, Chirivella J, Gagliardo P (2016) Evaluation of kinect2 based balance measurement. *Neurocomputing* 208:290–298
30. Ma Z, Sun Z (2018) Time-varying LSTM networks for action recognition. *Multimed Tools Appl* 77(24):32275–32285
31. McNeely D, Beveridge J, Draper B (2020) Inception and ResNet features are (almost) equivalent. *Cogn Syst Res* 59:312–218
32. Murphy PK (2012) *Machine learning: a probabilistic perspective*. MIT Press, Cambridge
33. Ng JYH, Hausknecht M, Vijayanarasimhan S, Vinyals O, Monga R, Toderici G (2015) Beyond short snippets: Deep networks for video classification. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 4694–4702
34. Peng X, Wang L, Wang X, Qiao Y (2016) Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *Comput Vis Image Underst* 150:109–125
35. Pizzari C, Cannici M, Matteucci M (2020) Spatial temporal transformer network for skeleton-based action recognition. arXiv:[2008.07404](https://arxiv.org/abs/2008.07404)
36. Shen J, Tang X, Dong X, Shao L (2020) Visual object tracking by hierarchical attention siamese network. *IEEE Trans Cybern* 50(7):3068–3080
37. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. In: *Proceedings of the 27th International conference on neural information process system*, pp 568–576
38. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: *Proceedings of the international conference Learning representations*, pp 1–14
39. Soomro K, Zamir AR, Shah M (2012) UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv:[1212.0402](https://arxiv.org/abs/1212.0402)
40. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 2818–2826
41. Tan Z, Wang M, Xie J, Chen Y, Shi X (2017) Deep semantic role labeling with self-attention. arXiv:[1712.01586](https://arxiv.org/abs/1712.01586)
42. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3D convolutional networks. In: *Proceedings of the IEEE international conference on computer vision*, pp 4489–4497
43. Vaswani A, Shazeer N, Parmar N, Uszkoreit J (2017) Attention is all you need. arXiv:[1706.03762](https://arxiv.org/abs/1706.03762)
44. Wang H, Kläser A, Schmid C, Liu CL (2011) Action recognition by dense trajectories. In: *Proceedings IEEE conference on computer vision and pattern recognition*, pp 3169–3176

45. Wang L, Qiao Y, Tang X (2016) MoFAP: A multi-level representation for action recognition. *Int J Comput Vis* 119(3):254–271
46. Wang H, Schmid C (2013) Action recognition with improved trajectories. In: *Proceedings of the IEEE international conference on computer vision*, pp 3551–3558
47. Wang W, Shen J, Ling H (2019) A deep network solution for attention and aesthetics aware photo cropping. *IEEE Trans Pattern Anal Mach Intell* 41(7):1531–1544
48. Wang W, Shen J, Shao L (2018) Video salient object detection via fully convolutional networks. *IEEE Trans Image Process* 27(1):38–49
49. Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Gool LV (2019) Temporal segment networks for action recognition in videos. *IEEE Trans Pattern Anal Mach Intell* 41(11):2740–2755
50. Wang W, Zhang Z, Qi S, Shen J, Pang Y, Shao L (2019) Learning compositional neural information fusion for human parsing. In: *International conference on computer vision*, pp 5702–5712
51. Wang W, Zhu H, Dai J, Pang Y, Shen J, Shao L (2020) Hierarchical human parsing with typed part-relation reasoning. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 8926–8936
52. Willems G, Tuytelaars T, Gool LJV (2008) An efficient dense and scale-invariant spatio-temporal interest point detector. In: *Proceedings European conference on computer vision*. Springer, Berlin, pp 650–663
53. You Q, Jin H, Wang Z, Fang C, Luo J (2016) Image captioning with semantic attention. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 4651–4659
54. Zach C, Pock T, Bischof H (2007) A duality based approach for realtime TV-L<sup>1</sup> optical flow. In: *Proceeding of the 29th DAGM symposium pattern recognition*, pp 214–223
55. Zhang J, Hu H, Lu X (2019) Moving foreground-aware visual attention and key volume mining for human action recognition. *ACM Trans Multimed Comput Comm Appl* 15(3):1–16
56. Zhang B, Wang L, Wang Z, Qiao Y, Wang H (2018) Real-Time action recognition with deeply transferred motion vector CNNs. *IEEE Trans Image Process* 27(5):2326–2339
57. Zhang K, Zhang L (2017) Extracting hierarchical spatial and temporal features for human action recognition. *Multimed Tools Appl* 77(13):16053–16068
58. Zheng W, Jing P, Xu Q (2019) Action recognition based on spatial temporal graph convolutional networks. In: *Proceedings of the 3rd international conference on computer science and application engineering*, pp 1–5
59. Zhu J, Zou W, Zhu Z, Xu L, Huang G (2019) Action machine: Toward person-centric action recognition in videos. *IEEE Sig Proc Lett* 26(11):1633–1637

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.