International Journal of Machine Learning and Cybernetics (2021) 12:823–841
https://doi.org/10.1007/s13042-020-01204-5

ORIGINAL ARTICLE

Spatiotemporal attention enhanced features fusion network for action recognition

Danfeng Zhuang[1] · Min Jiang[1] · Jun Kong[1] · Tianshan Liu[2]

Received: 19 February 2020 / Accepted: 18 September 2020 / Published online: 12 October 2020

Abstract

In recent years, action recognition has become a popular and challenging task in computer vision. Nowadays, two-stream networks with appearance stream and motion stream can make judgment jointly and get excellent action classification results. But many of these networks fused the features or scores simply, and the characteristics in different streams were not utilized effectively. Meanwhile, the spatial context and temporal information were not fully utilized and processed in some networks. In this paper, a novel three-stream network spatiotemporal attention enhanced features fusion network for action recognition is proposed. Firstly, features fusion stream which includes multi-level features fusion blocks, is designed to train the two streams jointly and complement the two-stream network. Secondly, we model the channel features obtained by spatial context to enhance the ability to extract useful spatial semantic features at different levels. Thirdly, a temporal attention module which can model the temporal information makes the extracted temporal features more representative. A large number of experiments are performed on UCF101 dataset and HMDB51 dataset, which verify the effectiveness of our proposed network for action recognition.

Keywords Action recognition · Three-stream · Spatiotemporal attention · Features fusion

1 Introduction

The target of action recognition is to analyze the actions executed by the targets automatically. In earlier studies, manual features are applied in action recognition widely. Some manual feature-based methods can extract useful features in the videos and achieve excellent performance. People have tried to get more spatiotemporal local features in these methods. Some of these methods constructed local feature descriptors and extracted motion information around the interest points. Then, we could obtain the local feature vectors, such as cuboids [1], histogram of gradient and histogram of flow (HOG/HOF) [2], extended SURF (ESURF) [3] descriptors. Meanwhile, spatiotemporal trajectory-based action recognition methods such as [4] were an extension of local feature points in time and space. By tracking the key points of moving objects, [4] constructed more powerful local features. This method which was based on dense trajectories, has achieved good results in many public action recognition datasets.

With the development of deep learning, many deep learning networks [5–7] are utilized to extract features effectively. We can automatically extract the important features in RGB images by deep learning networks. To get the joint judgement with different patterns, two-stream networks such as [8, 9] trained the appearance stream and the motion stream separately. To capture the long-term temporal information and infer action labels after observing the execution of the entire action in the whole video, temporal segment networks (TSN) [10] was proposed. It evenly divided every video into segments and selected frames from different temporal segments as the objects of analysis to describe the whole video. The semantic information was extracted from the RGB appearance stream and optical flow motion stream. The video-level prediction was realized by segment consensus. Then prediction scores of the two streams were fused to obtain the final

✉ Min Jiang
minjiang@jiangnan.edu.cn

[1] Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi 214122, China

[2] Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong 999077, China

judgment result. In most of these two-stream networks for RGB video-based action recognition, RGB frames from videos have been used to express spatial information because they have rich detailed information and detailed information. Besides, various modalities with different characteristics such as optical flow data, optical-flow-like data and skeleton data are utilized in action recognition to complement the single RGB stream. The pre-extracted optical flow can describe the motion information of videos and is robust against dynamic circumstances. Therefore, optical flow is widely used in two-stream networks for action recognition.

Based on the deep learning networks above, the two-stream networks encounter several common questions as follows. Firstly, in most of the two-stream networks, using simple fusion methods on the extracted features or aggregating the classification scores of every stream cannot make full use of the characteristics from different streams. Therefore, effective features fusion methods are needed in the spatiotemporal stream, which can use different streams to complement each other and obtain promoted features. Secondly, in action recognition tasks, we need to focus on important information that can provide us with useful information. In deep learning networks, the first few layers of the networks learn low-level features and the last few layers learn high-level features. The spatial features in different levels are aggregated along the channel dimension, and it is necessary to extract and enhance this information effectively [11]. Therefore, how to get the representative semantic features and focus on the vital information in the frames are challenging tasks. Thirdly, for action recognition tasks based on long-term RGB videos, we need to deal with not only spatial context features, but also temporal features. The random segment sampling strategy provides us with an effective method to obtain a sequence of frames that can represent the entire video. In the process of segment consensus, the contribution of every frame is different. Meanwhile, frame sequences selected from different segments contain useful temporal information. Therefore, the interrelationships between frames should be established to model the temporal features and help us enhance the effect of segment consensus.

Based on the above description, this paper proposes a novel action recognition network named spatiotemporal attention enhanced features fusion network (ST-AEFFNet). Our contributions are as follows:

1. A novel joint training stream called multi-level features fusion (MFF) stream is designed to improve the interaction between different types of information. In this structure, we integrate the information from high representative levels in the original two streams. Meanwhile, we use different fusion blocks for different levels of features to take advantage of the characteristics of different high-level features.
2. Two attention modules named original spatial context guided attention (OSCGA) module and high-level channel grouped attention (HCGA) module are proposed in this paper for effectively extracting spatial semantic features. In the original input, the OSCGA module aggregates the spatial context to generate adaptive channel weights, which can guide the following network. HCGA module is designed to enhance the high-level information by modeling the relationships between the grouped channels.
3. To get better segment consensus effects and more accurate predictions, the temporal enhanced attention (TEA) module is proposed. The TEA module can model the temporal features, and the importance of each frame is evaluated to improve the effectiveness of important frames.

2 Related work

In this section, we briefly introduce the previous work in deep learning networks and attention mechanisms separately.

2.1 Deep learning networks for action recognition

With the great advance in deep learning, models based on deep convolutional neural networks have been widely used in action recognition. Deep learning networks with 2D-CNNs [5, 6, 12] have been used to get great performance in extracting local spatial information. The 2D convolutional operation performs on the single frame. Nowadays, most of the action recognition tasks are on long-term videos in which the interrelationships of frame sequences need to be built. To solve the problem, deep learning networks that can get long-term temporal information are proposed. Currently, recurrent neural networks (RNNs), two-stream networks with 2D-CNNs and 3D convolutional neural networks (3D-CNNs) are employed to capture fine motion, temporal ordering, and long-range dependencies.

RNNs such as [13–15] could utilize the previous information to understand the information. Donahue et al. [16] combined 2D-CNNs with long short-term memory (LSTM), in which 2D-CNNs aimed to extract the spatial features frame by frame and the goal of LSTM was to capture temporal dependencies. But recent academic researches based on LSTM find that the gradient still disappears when the sequence length exceeds the limit. At the same time, LSTM has some difficulties in parallelization. To represent spatial and temporal information in videos, two-stream networks such as [8] effectively learned the spatiotemporal characteristics by using RGB frames and optical flow frames as appearance representations and motion representations to describe videos. The single-stream was trained independently, and the features were sent to different streams to get the classification. Based on the traditional two-stream networks, features fusion methods are proposed to fuse different types of features effectively. Feichtenhofer et al. [17] introduced a residual connection from the motion stream to the appearance stream, and at the same time, it fine-tuned the network to learn spatiotemporal features. By building multiplicative interaction of appearance features and motion features coupled with identity mapping kernels, [18] could learn the long-term temporal information. The method used in these feature fusion models is to replace the two-stream structure with the features fusion structure directly. However, the RGB branch and optical flow branch which can provide sufficient information are also important.

Wang et al. designed a 3D convolutional network [19], which was effective in long-term tasks. By 3D convolution and pooling, the temporal and spatial features were achieved. Nevertheless, it still needed to optimize the parameters, and the performance should be supported by large-scale datasets in the experiments. Carreira et al. proposed [20] to combine 3D-CNNs with the structure of two-stream networks. It proposed a very efficient spatiotemporal model and achieved significant development by using the Kinetics dataset. However, this method had an excessive amount of 3D convolutional parameters and the optimization time was long. Qui et al. [21], Tran et al. [22] and Zhongxu et al. [23] attempted to go beyond 3D convolution and further figured out the problem of the heavy computation by joint spatiotemporal analysis. These methods could get a good balance between speed and accuracy by decomposing the 3D convolution filters into separate spatial and temporal operations. However, the large-scale datasets and calculation complexity of model bring huge training pressure on the network. Compared to 3D-CNNs, two-stream networks with 2D-CNNs show superiority on the challenging medium-scale datasets. Therefore, the two-stream structure with 2D-CNNs is still worth studying.

Basing on the two-stream structure with 2D-CNNs and features fusion stream, we propose a three-stream network with RGB appearance stream, optical flow motion stream and MFF stream. Different from the previous two-stream features interaction mechanisms which use the final output fusion and score fusion, this paper explores the effect of features interactions in different high-levels. RGB frames are employed in the appearance stream to achieve appearance representation, and the optical flow pattern is extracted as the motion representation. We also use the optical-flow-like pattern extracted by TVNet to compare the effects of different motion patterns in our experiments, and these data are shown in Fig. 1. The third stream is the MFF stream. In the MFF stream, we fuse the appearance information and the motion information by merging the multi-level features of the appearance stream and the

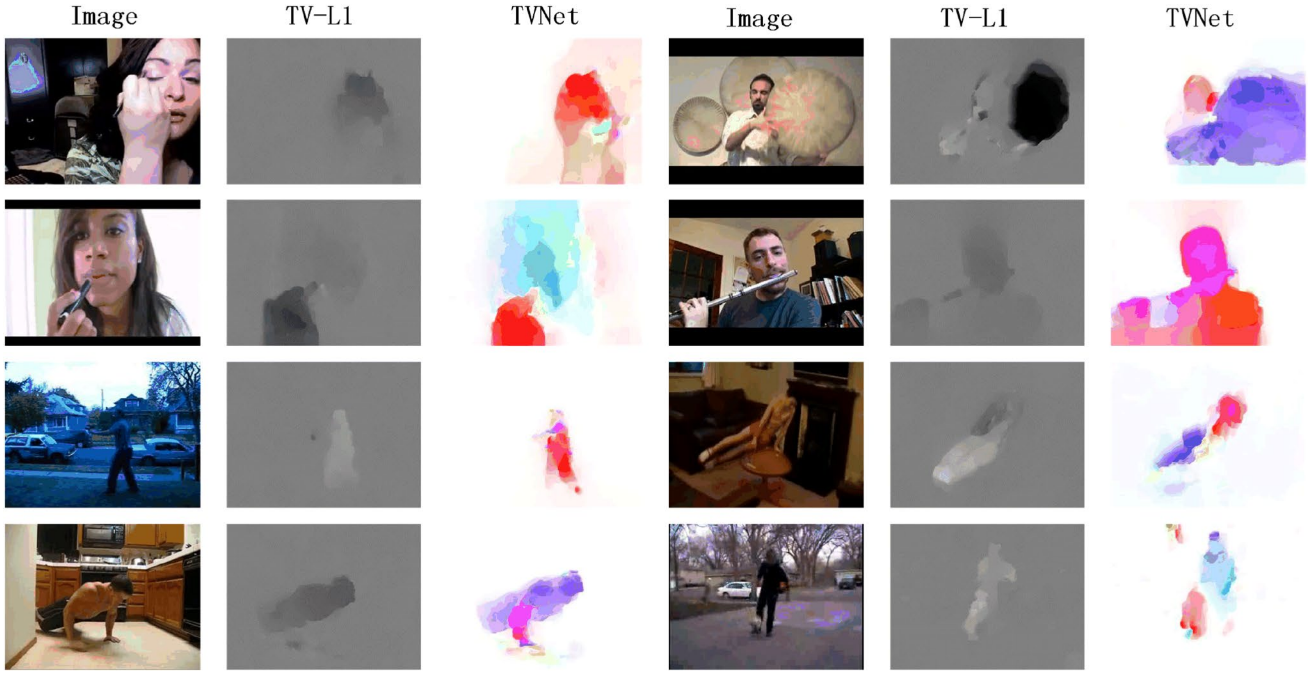


Fig. 1 Visualizations of the different motion patterns extracted by TV-L1 and TVNet. The action categories on the left are ApplyEyeMakeup, ApplyLipstick, Nunchucks and PushUps. The categories on the right are PlayingDaf, PlayingFlute, PommelHorse and SoccerJuggling

motion stream to obtain useful features that can supplement the original two-stream network.

2.2 Spatial and temporal attention mechanisms

Attention mechanisms have played an important role in human visual perception. Human vision does not capture the information of the entire scene at once, and it selectively captures the significant parts at first. In the process of action recognition, we not only need to pay attention to people's behavior in the videos but also need to capture and focus on representative objects which can provide sufficient information. For example, Fig. 2 shows actions such as walking with the dog, playing bowling, blowing candles, and so on. Dogs, bowling pins, and candles are also important in these actions. Therefore, how to focus on important areas to get the right classification is a challenging task. Now many networks with attention mechanisms are used for visual tasks to focus on the effective areas, which can achieve better results.

In the early stage, Wang et al. [24] designed the residual attention networks with encoded modules to improve the performances of convolutional neural networks (CNNs). Some attention models were proposed to get channel dependencies by modeling context information. By squeeze and excitation, [11] used the global average fusion function to calculate the channel attention weights. To accomplish spatial features aggregation and information redistribution in local areas, Hu et al. [25] took gather operator and excite operator to get channel descriptors. At the same time, Zhao et al. [26] tried to connect each position with all the other positions by a self-adaptively learned attention mask. By combining the channel attention mechanism with the spatial attention mechanism, Woo et al. [27] first proposed to infer attention maps in the two separate dimensions.

The previous methods used convolutions and iterative operations to construct attention blocks. To capture long-range dependencies, we need to repeat the convolutional operations and iterative operations in the local neighborhoods of space and channels, which have very limited process fields. To get non-local relationships between pixels of images in the spatiotemporal dimension, [28] directly captured the remote dependency by calculating the relationship descriptors between every two locations. Fu et al. [29] proposed two types of attention modules that could get the semantic interdependencies in spatial and channel dimensions. In the spatial module, the attention mechanism selectively aggregated the feature of each position with a weighted fusion at all positions. The channel module emphasized interdependent channel features by integrating associated features among all channel maps. To allow each neuron to adjust the size of its convolutional kernel adaptively, Li et al. [30] used a dynamic selection mechanism for the convolutional kernels in CNNs. Li et al. [31] generated attention weights for every position in each semantic group and adjusted the weights in each sub-feature. To model the temporal information better, temporal attention methods have been proposed. To get better temporal modeling, Zhou et al. [32] proposed a method, in which Bi-directional long short-term memory (BiLSTM) used attention mechanisms to establish temporal relationships.

Inspired by the spatiotemporal attention mechanism, we propose three attention modules to guide and enhance the network. The attention mechanism used in action recognition rarely explores the effects of attention weights on different levels. In this paper, the OSCGA module and HCGA

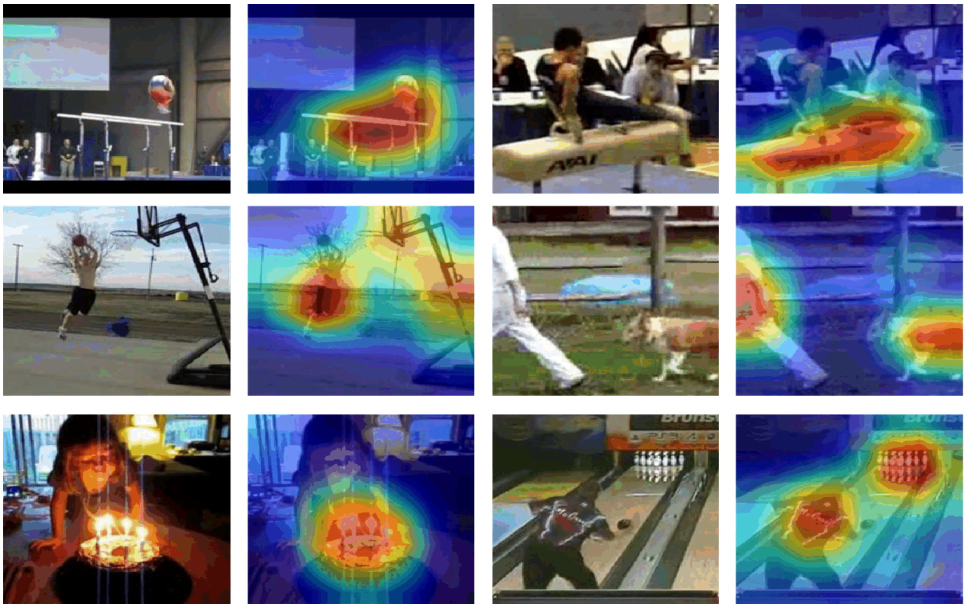


Fig. 2 The original RGB images and the heatmaps of important areas obtained by training RGB images in UCF101 dataset. Deep learning networks focus on different objects which are important to make judgement

module entirely use the spatial context and model the channel characteristics obtained by spatial context aggregation in different levels. To evaluate the importance of each frame and achieve effective segment consensus, the results of temporal modeling are used as temporal attention weights of frames in the TEA module.

3 Our work

In this section, we first propose the overall structure of ST-AEFFNet with the features fusion stream MFF in Sect. 3.1. The architecture is in Fig. 3. To obtain long-term temporal information from video sequences, we divide each input video into T segments and conduct sparse sampling. In the streams, these frames are input into the network in order. Firstly, the original input is sent into the OSCGA module, and spatial context features are modeled to get attention maps. The expression of OSCGA is shown in Sect. 3.2. The output of OSCGA is sent into InceptionV3, which is adopted as the backbone to extract semantic information. Next, these high-level features are sent into the HCGA module and TEA module separately. HCGA module can obtain effective spatial semantic information by channel attention mechanism, and the structure of HCGA is in Sect. 3.3. TEA module and segment consensus are employed on the temporal sequences to model the temporal information and are proposed in Sect. 3.4. In this section, the output of the HCGA module and the output of the TEA module are combined as the spatiotemporal attention enhanced features in the single stream. Segment consensus is used to fuse features from different segments, which can get the video-level prediction. Finally, these features are input into the linear classification layer to obtain classification scores.

3.1 Overall architecture

3.1.1 RGB Appearance Stream and Optical Flow Motion Stream

In RGB appearance stream and optical flow motion stream, the original inputs are expressed as $X_* \in R^{T_* \times C_* \times H_* \times W_*}$ $, * \in \{A, M\}$. Here T_* is the number of the segments. C_* is the channel dimension of frames in every segment. H_* and W_* are numbers of pixels in the height and width of frames. In RGB appearance stream, we select one frame in every segment. Different from RGB appearance stream, the serial stacked optical flow frames in x direction and y direction are selected in optical flow motion stream. $OSCGA_*(\cdot)$, $HCGA_*(\cdot)$ and $TEA_*(\cdot)$ are the three spatiotemporal attention functions. $INC_*(\cdot)$ is InceptionV3 function. In $INC_*(\cdot)$ function, features chosen from three high levels are expressed as $F^i_{INC*} \in R^{T_* \times C_i \times 8 \times 8}, i \in \{1, 2, 3\}$ and its last layer output after global pooling is expressed as $F^4_{INC*} \in R^{T_* \times C_4 \times 1 \times 1}$. i represents the selected three high levels. Here, F^4_{INC*} is sent into $HCGA_*(\cdot)$ and $TEA_*(\cdot)$. After extracted and enhanced by spatiotemporal attention functions, $F_* \in R^{T_* \times C' \times 1 \times 1}$ is

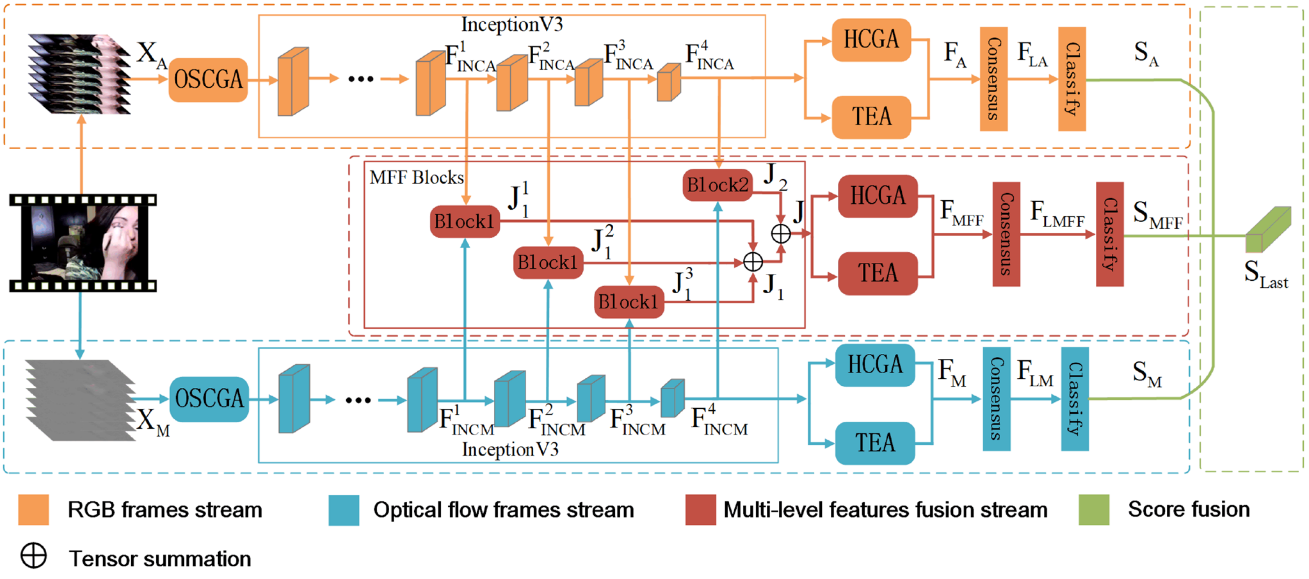


Fig. 3 The overall architecture of ST-AEFFNet. The orange represents an RGB appearance stream with stacked RGB frames. The blue represents an optical flow motion stream with an optical flow pattern. And red represents a multi-level fusion stream that trains RGB patterns and optical flow patterns jointly with MFF Blocks (Block1 and Block2). Spatiotemporal attention modules OSCGA, HCGA and TEA are added to the three-stream network for the features guidance and enhancement. Finally, the green represents a weighted fusion of the scores obtained as a fusion result of three streams

obtained. C' is the channel dimension of F_*. The formula to calculate F_* is expressed as follows:

$$\begin{aligned} F_* &= HCGA_*(INC_*(OSCGA_*(X_*))) \\ &\quad + TEA_*(INC_*(OSCGA_*(X_*))) \\ &= HCGA_*(F^4_{INC*}) + TEA_*(F^4_{INC*}) \end{aligned} \tag{1}$$

The features F_A and F_M can be expressed as $F^i_A = \left\{F^1_A, F^2_A, \dots, F^{T_A}_A\right\}, i \in \{1,2,\dots,T_A\}, F^i_A \in R^{1\times C'}$ and $F^i_M = \left\{F^1_M, F^2_M, \dots, F^{T_M}_M\right\},\ i \in \{1,2,\dots,T_M\},\ F^i_M \in R^{1\times C'}$ separately. Next, average segment consensus is performed on F^i_A and F^i_M to get video-level prediction $F_{LA} \in R^{1\times C'}$ and $F_{LM} \in R^{1\times C'}$ over the whole video. The formulas to calculate F_{LA} and F_{LM} are expressed as follows:

$$\begin{aligned} F_{LA} &= \frac{1}{T_A}\sum_{i=1}^{T_A} F^i_A \\ F_{LM} &= \frac{1}{T_M}\sum_{i=1}^{T_M} F^i_M \end{aligned} \tag{2}$$

After getting F_{LA} and F_{LM} in the streams, we input these features into the linear classification layer to get the scores S_A and S_M separately.

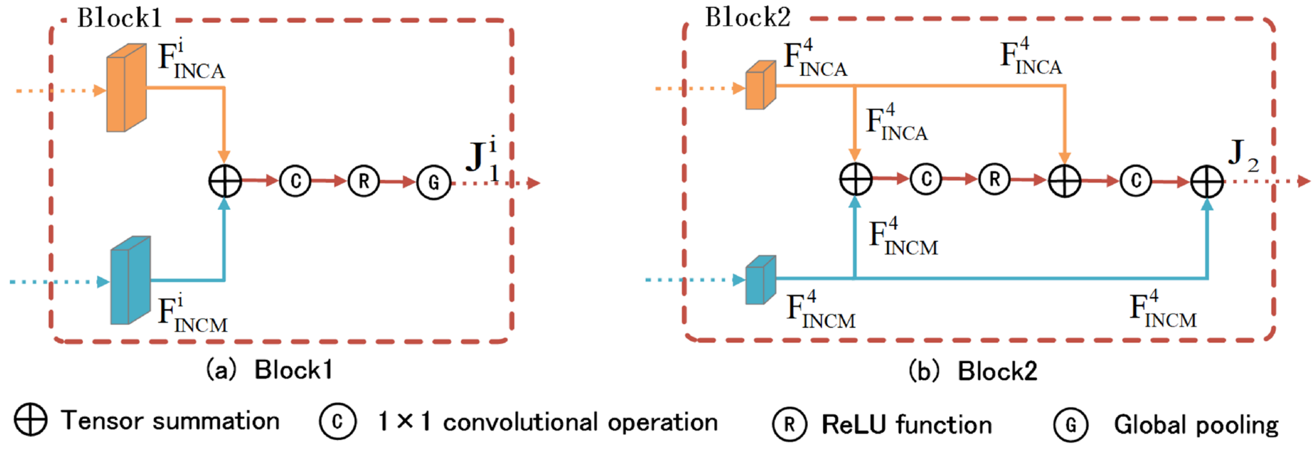
3.1.2 Multi-level features fusion stream

In the traditional two-stream networks, the RGB appearance stream and optical flow motion stream were trained separately. In the process of training, the network may overfit to the features in the single stream. Therefore, multi-level features fusion (MFF) stream is proposed to train the two streams jointly and achieve complementary features. To be specific, we select the features from three high levels and the last level in InceptionV3 backbone from the RGB appearance stream and optical flow motion stream. By fusing these features by Block1 and Block2, we can obtain more complementary features. These two fusion blocks are shown in Fig. 4.

The reason for the different structures of Block1 and Block2 is that various high-level features have different information. In general, in the higher level of a deep learning network, these are less detailed information such as texture information, and more semantic information. High-level semantic information can represent the information which is closest to human understanding. Therefore, the resolution ability of semantic features is stronger than that of detailed features. But sometimes the detailed information also plays a role in the judgment, and it is necessary to design different fusion blocks in different levels.

In Block1, features fusion is performed by add operation and convolution operation. The shallower high-level features can obtain sinformation in this block. However, this information is in the middle layers of the deep learning network, so in addition to containing semantic features, they also provide more information such as texture features. For example, the texture information of cats and dogs maybe the same, and high-level semantic information is needed to distinguish between two categories. By stacking the features which the network considers to be effective, we can obtain cross-layer information that can complement each other. The global pooling operation in Block1 uses the global max-pooling operation. This operation can reduce the deviation of the estimated mean caused by the parameter error in convolution layer. Besides, global max pooling can also be regarded as a feature selection function. By selecting the maximum value, the features with better classification recognition are chosen.

Block2 serves to fuse the last-layer high-level semantic features extracted by InceptionV3. High-level semantic



(a) Block1

(b) Block2

Fig. 4 Illustrations of Block1 and Block2 in multi-level fusion blocks (MFF blocks). We describe the effective fusion functions between RGB appearance features and optical flow motion features extracted by InceptionV3 backbone

features generally summarize the attributes of entities. Features at this layer have the essential semantic information obtained by deep learning networks, and most of the networks use the output of this layer for classification. Meanwhile, when the appearance features are integrated into the motion features in the fusion stream, the training loss of the motion stream reduces faster and ends with a lower value. It makes the fusion stream focus too much on appearance information. Therefore, differing from fusing the last-layer high-level semantic features of the appearance stream and motion stream in Block1, the structure of Block2 adds the motion features again to retain the characteristics of the motion stream most.

Features from three high levels in RGB appearance stream are expressed as $F^i_{INCA}, i \in \{1,2,3\}$, and corresponding features in optical flow motion stream are expressed as $F^i_{INCM}, i \in \{1,2,3\}$. We adopt Block1 with convolution operation $Conv(\cdot)$, activation function $ReLU(\cdot)$ and global pooling operation $G(\cdot)$ to combine F^i_{INCA} with F^i_{INCM}. Block1 is shown in Fig. 4a. J_1 is the fusion result of these features from three high levels. The formula is expressed in Eq. (3):

$$\begin{aligned} J^i_1 &= G\left(ReLU\left(Conv\left(F^i_{INCA} + F^i_{INCM}\right)\right)\right) \quad i \in \{1,2,3\} \\ J_1 &= \sum_{i=1}^{3} J^i_1 \end{aligned} \tag{3}$$

F^4_{INCA} and F^4_{INCM} are the last level outputs after global pooling of InceptionV3 backbone in RGB stream and optical flow stream separately. In the fusion of the two streams, we adopt Block2 which is shown in Fig. 4b. The definition of the attention-enhanced fusion result J_2 is indicated in Eq.(4):

$$\begin{aligned} J_2 = {} & Conv\left(ReLU\left(Conv\left(F^4_{INCA} + F^4_{INCM}\right)\right) + F^4_{INCA}\right) \\ & + F^4_{INCM} \end{aligned} \tag{4}$$

By adding J_2 to J_1, we obtain the multi-level fusion feature J in MFF stream. We input the obtained J into HCGA module and TEA module. The spatiotemporal attention enhanced feature in MFF is expressed as F_{MFF}. $HCGA_{MFF}(\cdot)$ and $TEA_{MFF}(\cdot)$ are HCGA function and TEA function in MFF stream. The formula to get F_{MFF} is expressed in Eq. (5):

$$F_{MFF} = HCGA_{MFF}(J) + TEA_{MFF}(J) \tag{5}$$

Then, average segment consensus is performed on $F^i_{MFF} = \left\{F^1_{MFF}, F^2_{MFF}, \dots, F^{T_{MFF}}_{MFF}\right\}, \quad i \in \{1,2,\dots,T_{MFF}\}$, $F^i_{MFF} \in R^{1\times C'}$ to get video-level prediction over the whole video. In MFF stream, the number of segments in RGB frame sequences is T_{MFF} which is the same as that in optical flow frame sequences. The feature $F_{LMFF} \in R^{1\times C'}$ in MFF stream which is obtained after segment consensus is calculated in the following formula:

$$F_{LMFF} = \frac{1}{T_{MFF}} \sum_{i=1}^{T_{MFF}} F^i_{MFF} \tag{6}$$

After getting F_{LMFF} in the fusion stream, we input it into the linear classification layer to get the classification score S_{MFF}.

3.1.3 Score fusion

The scores S_A, S_M and S_{MFF} are obtained in the three streams and a weighted fusion method is used to fuse these scores. Weights of the three streams are defined as ω_A, ω_M and ω_{MFF} separately. S_{Last} is the final prediction of our ST-AEFFNet and the formula is in Eq. (7).

$$\mathrm{S}_{Last} = \omega_A \times S_A + \omega_M \times S_M + \omega_{MFF} \times S_{MFF} \tag{7}$$

3.2 Original spatial context guided attention module

Generally, the original input contains a large number of details, which have messy backgrounds and fuzzy semantics. Applying this detailed information without processing cannot provide appropriate guidance to the following network. Most of the models send original features into the convolutional filters without processing so that the detailed information in the original input cannot be used effectively. In the OSCGA module, we use the spatial relationship of different location features to generate spatial attention maps. The generated spatial attention map is focused on where the informative part is. Inspired by the popular squeeze-excitation (SE) network [11], we enhance the original features by summarizing the same features from all locations. Firstly, we apply the average pooling and maximum pooling to generate an efficient feature descriptor. Secondly, we construct a context modeling function which aggregates the features of all locations to form a global context feature. The purpose of the above two functions is to enhance the features of query locations by aggregating information from other locations. Finally, the above two feature descriptors are merged into features at all positions in the space to get the attention maps. By getting where the informative part is with the attention map, the following feature extraction network can be guided to extract useful information. Therefore, the OSCGA module is designed to model the spatial context information and guide the following network. The illustration of OSCGA is in Fig. 5.

In the OSCGA module, we do not consider the processing on the temporal dimension T and only operate on the features with the size of $C \times H \times W$. C is the channel dimension. H and W are numbers of pixels in the height and width of the original input. By modeling the information of original input and processing the cluttered

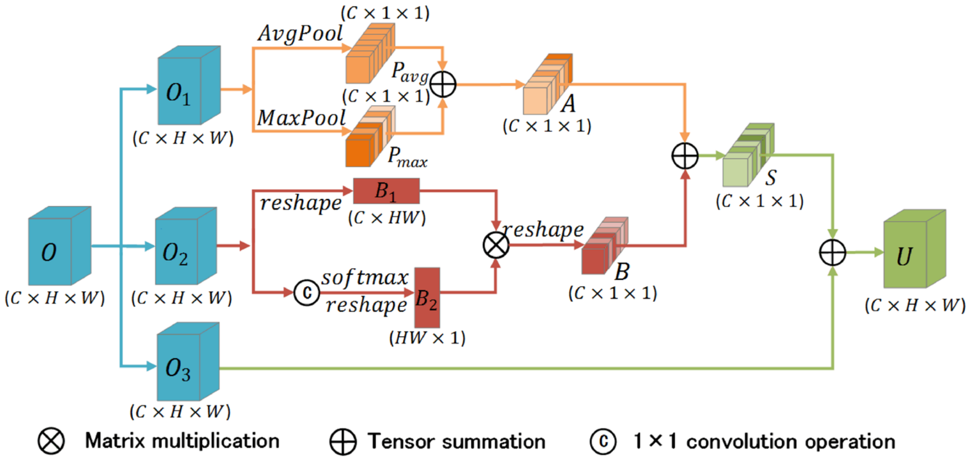


Fig. 5 Illustration of OSCGA module. It utilizes global pooling modeling and simplified non-local modeling to achieve original input channel descriptors. The results of OSCGA module are input into the following network as guidance

backgrounds and fuzzy context information, representative information is obtained along channels. These representations are used as weights to evaluate the importance of every channel and guide the following network. The original input is expressed as $O \in R^{C\times H\times W}$, and $O_1 \in R^{C\times H\times W}$, $O_2 \in R^{C\times H\times W}$, $O_3 \in R^{C\times H\times W}$ are three same inputs copied from O. Here two types of feature compression processing are adopted to squeeze the spatial information into channel descriptors.

One adopted processing is to transform every two-dimensional feature into a channel descriptor by combining the results of global average pooling and global maximum pooling. $P_{avg} \in R^{C\times1\times1}$ and $P_{max} \in R^{C\times1\times1}$ are the two generated descriptors obtained by global average pooling and global maximum pooling separately. The formula of $A \in R^{C\times1\times1}$ is as below:

$$A = P_{avg} + P_{max} \tag{8}$$

The other processing is to model the spatial context by the simplified non-local function which is proposed in [33]. With the non-local function, we can model the spatial context and extract effective information. Here O_2 is the original input. We perform the 1×1 convolutional operation $Conv(\cdot)$ on $O_2 \in R^{C\times H\times W}$ and input the convolutional result into softmax activation function $softmax(\cdot)$ to obtain a vector with the size of $R^{1\times HW}$. HW is to multiply H and W. Then the vector is reshaped to $B_2 \in R^{HW\times1}$. Here, we describe the calculation of the parameters for each subsequent position in details. Our update formula can be expressed in Eq. (9). The interrelationships are established between every location in O_2. The superscript i is the index of the position that needs to be processed in the feature map and B_2^i is the output value at i position obtained after the function. j is the index of position that enumerates all the possible positions and O_2^j is the value of j position in O_2. ω_k and ω_v are the linear transformation matrices. N is the number of positions in the feature map.

$$\begin{aligned} B_2 &= softmax(Conv(O_2)) \\ B_2^i &= \omega_v \sum_{j=i}^{N} \frac{\exp\left(\omega_k O_2^j\right)}{\sum_{h=1}^{N} \exp\left(\omega_k O_2^h\right)} O_2^j \end{aligned} \tag{9}$$

Then we aggregate the values of all locations to form a channel descriptor by simplified non-local function. $B_1 \in R^{C\times HW}$ is the vector reshaped from O_2. Next we multiply B_1 and B_2 to achieve a vector with the size of $R^{C\times1}$ and reshape it to $B \in R^{C\times1\times1}$. The features obtained by these two methods respond to the channels. $S \in R^{C\times1\times1}$ is the fusion result, which is obtained by combining A with B. Finally we add S to $O_3 \in R^{C\times H\times W}$ and get new vector $U \in R^{C\times H\times W}$ as the input of following network.

$OSCGA_A(\cdot)$ is the OSCGA function of RGB appearance stream in Eq. (1). In OSCGA module, it takes original input X_A as input O of OSCGA, and output U corresponds to $OSCGA_A(X_A)$. $OSCGA_M(\cdot)$ is the OSCGA function of optical flow motion stream in Eq. (1). It uses original input X_M as input O of OSCGA, and output U corresponds to $OSCGA_M(X_M)$.

According to the automatically acquired importance of each channel, useful features are enhanced, and others are suppressed for the current task. Then the output of the OSCGA module is sent into InceptionV3 backbone for depth feature extraction, and we obtain high-level features with rich semantic information. Next, we will introduce the HCGA module, which can handle high-level characteristics.

3.3 High-level channel grouped attention module

In this section, we process the high-level features obtained by the InceptionV3 backbone. High-level features are extracted and generated along the channels in the convolutional neural networks, and they have rich semantic information. Modeling the interrelationships of channel features that are obtained by aggregating high-level spatial features is an efficient way to improve the network with less computation. In the features obtained by the InceptionV3 backbone, different channels are sensitive to different spatial semantic information and provide diverse important information. Therefore, we have to design an effective channel enhancement function. With the development of CNNs, grouped methods such as grouped convolutions are designed to fuse and process features along the channel dimension. The illustration of the HCGA module is in Fig. 6.

To establish the connection between channels in the HCGA module, we operate on the grouped channels and global groups in order. Within each group, we use the 1 × 1 convolutional operation. We can obtain the fused channel features which are the representations of each group. To get global channels modeling, we need to establish the interrelationships between groups. The results of channel modeling are used as attention weights to enhance the important channels and suppress the influence of unimportant channels on the final evaluation results.

Firstly, these spatial features with rich semantic information are extracted by InceptionV3 backbone. After extracting features in RGB appearance stream and optical flow motion stream, feature maps with the size of $T \times C \times 1 \times 1$ are achieved after global pooling. T is the number of segments and C is the channel dimension. Because of not considering temporal processing, HCGA module is performed on the aggregated spatial feature $E \in R^{C\times1\times1}$. And the extracted feature E is divided into G groups along channels. These groups are represented as $E = \{E_1, E_2, \dots, E_G\}$. We achieve the enhanced features by aggregating the features in every group. $Conv(\cdot)$ is a 1 × 1 convolutional operation to fuse the channel information to a vector which represents the corresponding group. $Q \in R^{G\times1\times1}$ reflects the fused representation of all channels in each group. σ is a sigmoid function gate. The formula of Q is in Eq. (10).

$$Q = \sigma(Conv(E)) \tag{10}$$

Then we adjust the shape of Q to get $Q_1 \in R^{G\times1}$ and $Q_2 \in R^{1\times G}$. $H \in R^{G\times G}$ with two dimensions is achieved by multiplying Q_1 and Q_2 and it represents the relationships between every two groups. This method enables us to establish relationships between all the groups. We use $ReLU(\cdot)$ as the activation function. The formula to obtain H is presented in Eq. (11):

$$H = ReLU(Q_1 \times Q_2) \tag{11}$$

Then the convolutional operation is used to reshape the grouped channel feature which is obtained by multiplying H and Q. $F \in R^{C\times1\times1}$ is the feature after a 1 × 1 convolutional operation $Conv(\cdot)$. The formula is presented in Eq. (12):

$$F = Conv(H \cdot Q) \tag{12}$$

Finally, we obtain F and treat them as attention weights. We multiply E and F to get the output of HCGA module which is expressed as I.

$HCGA_A(\cdot)$ is HCGA function of RGB appearance stream in Eq. (1). In HCGA module, it takes output F^4_{INCA} which is the last layer output of InceptionV3 as input E, and output I corresponds to $HCGA_A(F^4_{INCA})$. $HCGA_M(\cdot)$ is HCGA function of optical flow motion stream in Eq. (1). It uses the last layer output of InceptionV3 F^4_{INCM} as input E, and output I corresponds to $HCGA_M(F^4_{INCM})$. $HCGA_{MFF}(\cdot)$ is HCGA function of MFF stream in Eq. (5). It uses the multi-level fusion feature J as input E of HCGA, and output I corresponds to $HCGA_{MFF}(J)$.

We enhance features by the HCGA module and obtain useful high-level semantic features. Next, we will introduce the TEA module for segment consensus.

3.4 Temporal enhanced attention module and segment consensus

Features have been extracted and enhanced in the network by the OSCGA module and HCGA module to get

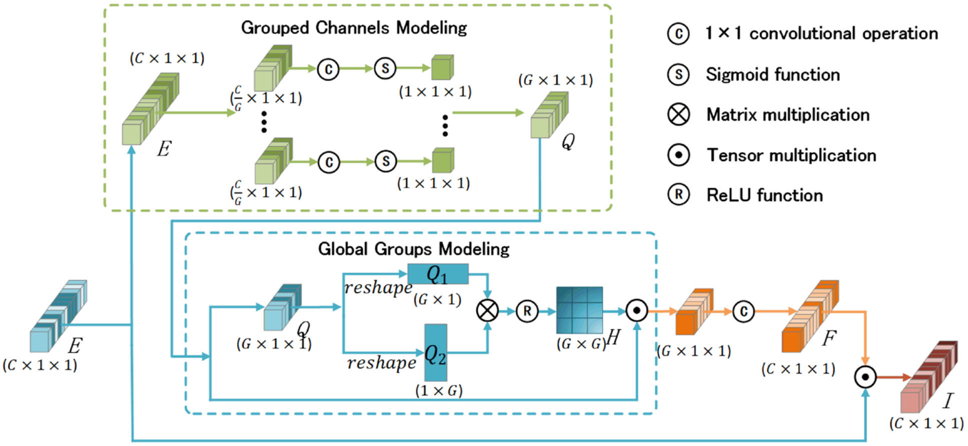


Fig. 6 Illustration of proposed HCGA module. In the green dotted box, channels are divided into several groups and the channel feature fusion is performed in every group. Then the relationships between the groups are established, and the weights of each channel are obtained by this module

sufficient and effective spatial semantic features. The next step is how to make full use of the features from different segments and promote the effect of segment consensus. In the segments we divide, frames in different temporal periods have different contributions to the classification. Consensus function is an important part of our network. The consensus function should have excellent modeling ability to aggregate the features of segments to get the final prediction over the whole video effectively. We optimize the entire network by using the differences between long-range segments.

Methods such as RNN and LSTM have been applied to action recognition tasks successfully. These methods can model the temporal information of the frame sequences. Through these temporal modeling methods, the obtained features can effectively represent the spatiotemporal variation of temporal series. However, as the distance between RNN increases, there is an issue that the gradient disappears and long-term temporal information cannot be learned. To overcome this problem, the LSTM model is proposed. Though LSTM has a memory door, it cannot completely remember all the past information. Compared with these previous methods, temporal convolutional network (TCN) [34] can retain long-range memory and minimize temporal information loss more realistically. At the same time, the TCN model is simpler than LSTM. Therefore, TCN is utilized in temporal modeling.

Here, the TEA module is designed as the function to operate on the ordered frame sequences. It is an adaptive weighted method based on the TCN model. In this module, we adaptively assign attention weights to each frame. In the process of training, the temporal relationships are built by the attention module. The attention mechanism can highlight important frames, and the segment consensus is enhanced by the TEA module. The structures of the TEA module and segment consensus are illustrated in Fig. 7. We reshape the feature extracted by InceptionV3 backbone after global pooling which has the shape of $R^{T\times C\times 1\times 1}$ to $X \in R^{T\times C\times 1}$. X is used as the input of the TEA module. $TCN(\cdot)$ can effectively model the temporal information, and the outputs of TCN are used as weights to evaluate the importance of each frame. By highlighting important frames in segment consensus, we can get better predictions in a single stream. The formula for calculating the temporal features $Y \in R^{T\times C\times 1}$ is as follows. We use $ReLU(\cdot)$ as the activation function.

$$Y = ReLU(X \cdot TCN(X)) \tag{13}$$

After getting the attention weights, we add these values to the feature $I' \in R^{T\times C\times 1}$. I' is reshaped from the output I of the HCGA module in Sect. 3.3. Then average segment consensus $AVG(\cdot)$ is used on the selected frames. Important segments are given higher weights by temporal attention module, which makes them play a crucial role in the segment consensus. In this way we can obtain the consensus feature representation $L \in R^{1\times C\times 1}$. The formula is in Eq. (14):

$$L = AVG\left(Y + I'\right) \tag{14}$$

In RGB appearance stream, $TEA_A(\cdot)$ is the TEA function in Eq. (1). It takes output F^4_{INCA} which is the last layer output of the InceptionV3 as input X, and output Y corresponds to $TEA_A(F^4_{INCA})$. The result $HCGA_A(F^4_{INCA})$ is reshaped as I'. The features Z and L correspond to F_A and F_{LA}, respectively. In optical flow motion stream, $TEA_M(\cdot)$ is the TEA function in Eq. (1). It uses the last layer output of InceptionV3 as input X of TEA, and output Y corresponds to $TEA_M(F^4_{INCM})$. The result $HCGA_M(F^4_{INCM})$ is reshaped as I'. The features Z and L correspond to F_M and F_{LM}, respectively. In MFF stream, $TEA_{MFF}(\cdot)$ is the TEA function in Eq. (5). It uses the multi-level fusion feature J as input X of TEA, and output Y corresponds to $HCGA_{MFF}(J)$. The features Z and L correspond to F_{MFF} and F_{LMFF}, respectively. The specific formulas of $AVG(\cdot)$ in three different streams are shown in Eqs. (2) and (6).

Then we use linear classification function to calculate the category score for the obtained consensus feature L, and get the score of the corresponding category in each stream. Finally, a weighted fusion method fuses scores of the three streams in Eq. (7).

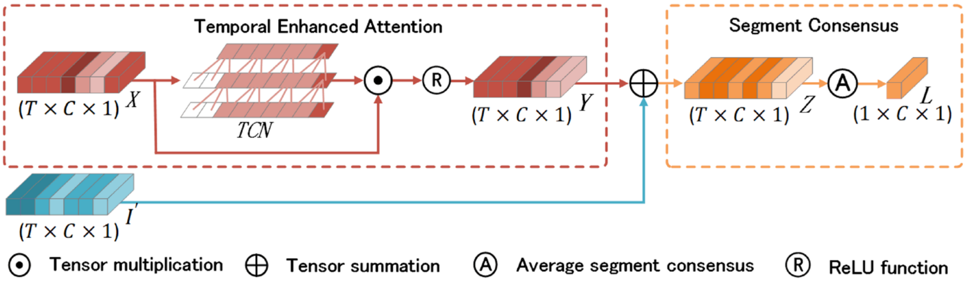


Fig. 7 Illustration of the proposed TEA module and segment consensus. Temporal series information is modeled by TCN and the importance of each frame is evaluated to enhance the role of effective frames in segment consensus

4 Experiments

To evaluate our proposed methods, experiments are performed on two publicly available human action datasets: UCF101 dataset and HMDB51 dataset. Firstly, we briefly introduce the datasets employed in the experiments. Next, the implementation details are described, and a series of ablation experiments are performed on the UCF101 dataset to explore the effects of the spatiotemporal attention modules we proposed. Furthermore, the validity of our features fusion stream with different blocks is tested, and a lot of experiments based on score fusions of different patterns are performed. Finally, our model is compared with state-of-the-art methods to discuss and analyze the experiment results for action recognition.

4.1 Datasets

4.1.1 UCF101 dataset

It is an action recognition dataset containing 13,320 video clips and 101 categories of actions. Each video contains one action. It has 101 types of actions, each of which is performed by 25 people and everyone performs 4–7 groups separately. The categories of actions are mainly human–object interaction, human–human interaction, physical movements, playing instruments and doing sports.

4.1.2 HMDB51 dataset

The videos in this dataset are selected from movies, public databases and video libraries such as YouTube. It contains 51 action categories with 6849 video sequences. HMDB51 dataset is challenging because it has different scales, different view angles, multiple activities, as well as complex backgrounds.

4.2 Implementation details

Firstly, our network processes appearance features and motion features. We use RGB frames extracted by OpenCV in the appearance stream. To get motion representation, we use two different models: TV-L1 model and TVNet model. Meanwhile, the MFF stream is proposed to train the appearance frames and the motion frames together to complement features.

In the RGB appearance stream and optical flow motion stream, to handle the long-term temporal information, the single video is divided into seven segments. In the RGB appearance stream, we randomly select frames from each segment to form frame sequences that can describe the whole video. In the optical flow motion stream, we select stacked optical flow frames from each segment to describe continuous action information. We clip the images of the dataset with a scale of 350×240 to a uniform size of 299×299. Our network is trained and tested on the UCF101 dataset and HMDB51 dataset. We report the Top-1 accuracies on the test dataset. We load the weights of the InceptionV3 backbone, which are pre-trained on ImageNet. On both the UCF101 dataset and HMDB51 dataset, the learning rate is 0.01 at the beginning. The learning rate decreases by 10 times every 30 rounds in the appearance stream, and in the motion stream, it decreases by 10 times every 120 rounds. The training in the appearance stream stops at 120 rounds and in the motion stream stops at 400 rounds. Our network is trained through Stochastic Gradient Descent (SGD) [35] with the momentum of 0.9. The dropout radio is set to 0.8, and the batch size is set to 24. In the branch of the MFF stream, to balance the computation and effect, the video is divided into 3 segments. The backbone weights which are obtained by appearance stream and motion stream are loaded as the pre-trained weights of the InceptionV3 backbone in the MFF stream. The learning rate is 0.01 at the beginning. It decreases by 10 times every 50 rounds, and the training stops at 200 rounds. We set the dropout radio to 0.8 and batch size to 48.

A weighted fusion method is used to fuse the scores of three streams. We tried different weights to get the best results and finally set the weights of appearance stream, motion stream and MFF stream to 1:0.5:0.5. Our studies are implemented with PyTorch [36]. All the model evaluation experiments are performed on four NVIDIA GTX1080Ti cards.

4.3 Ablation experiments

In ablation experiments, we will conduct the following studies on the UCF101 dataset. Firstly, we verify the validity of the three spatiotemporal attention modules in the two-stream network of RGB appearance stream and optical flow motion stream. Secondly, we analyze the experiments on the MFF stream, in which Block1 and Block2 are discussed to verify the effectiveness of the features fusion stream.

4.3.1 Spatiotemporal attention modules

Considering the period of ablation experiments on the three splits of the UCF101 dataset, we refer to the advanced methods such as TSN [10] and I3D [20] to conduct the ablation experiments on the split1. This method is proved to be representative and credible. More detailed experiments on these three attention modules are performed.

In Table 1, we can see that the three spatiotemporal attention modules have improved the accuracy in different

Table 1 Top-1 accuracies on split1 of UCF101 dataset show the influence of OSCGA (O), HCGA (H) and TEA (T) modules on RGB appearance stream (RGB) and optical flow motion stream (optical flow)

	RGB	Optical flow
	Recognition accuracy (%)	Recognition accuracy (%)
Original model	87.32	87.12
Our model (O)	87.77	87.31
Our model (H)	88.21	87.86
Our model (T)	87.85	87.47
Our model (O + H)	87.54	87.52
Our model (O + T)	87.91	88.19
Our model (H + T)	88.05	87.63
Our model (O + H + T)	88.73	88.28

streams of the traditional two-stream network. When each module is used independently in the network, the prediction accuracy can be improved. By the combination of the three modules, it achieves a 1.41% improvement in appearance stream and a 1.16% improvement in motion stream compared with the original model. As shown in the table, the OSCGA module slightly improves accuracy. It indicates that although original data contains a lot of noise, it can provide useful guidance for the following network by processing spatial context. Then, the HCGA module, which is added to the original model for efficient aggregation of high-level spatial features, improves the network more effectively. It indicates that the HCGA module can improve the representation of semantic information efficiently. The TEA module enhances the RGB appearance stream and the optical flow motion stream. It shows that temporal information is vital for RGB video action recognition and classification.

While using two or three attention modules, we find that the effect of the combination is not necessarily positive. The improvement from combining modules maybe not as good as expected. Some combinations have a smaller improvement on the model, and some are even lower than using a single module. For example, in the RGB stream, the accuracy of using OSCGA is 87.77% and using TEA is 87.85%, while the accuracy of combining OSCGA and TEA is 87.91%. There is only little improvement after the combination. But compared to the original model which does not use the attention mechanism, the accuracy has been improved. We analyze and explain the reasons for this phenomenon.

Firstly, the three different attention modules may focus on the same important features, which means that these modules are valid for the same category and video objects. In Fig. 8, we list different categories including ApplyEyeMakeup, ApplyLipstick, BoxingPunchingBag, TaiChi, TennisSwing, PullUps, PlayingViolin, PommelHorse, Rafting, Mixing. The following conclusions can be drawn from the figure. In categories such as ApplyLipstick, PlayingViolin, Rafting, etc., all the three attention modules can reduce the error rate. When the three modules are combined, we can achieve a lower error rate. It shows that the three modules can improve the classification effect and they focus on the same category, which may cause the improvement of combining three modules not as good as expected. Meanwhile, in some categories such as TiChi, the HCGA module improves recognition better. After the combination of three attention modules, the recognition error rate is slightly increased compared to using the HCGA module alone. For the ApplyEyeMakeup category, HCGA and TEA modules effectively reduce the error rate, the OSCGA module increases the error rate. In such cases, the effect of the combination is reduced. But the error rate of combining three modules is still lower than that of the original model. It shows that although the recognition ability of some categories after combining attention modules maybe not good enough as expected, our model can further improve the recognition effect generally.

Table 2 Top-1 accuracies on split1 of UCF101 dataset show the influence of combination strategies on HCGA (H) and TEA (T) modules on RGB appearance stream (RGB) and optical flow motion stream (optical flow)

Description	RGB	Optical flow
	Recognition accuracy (%)	Recognition accuracy (%)
Original model	87.32	87.12
Our model (H&T in order)	87.50	87.52
Our model (T&H in order)	87.71	87.18
Our model (H + T)	88.05	87.63

H + T is combined HCGA and TEA in parallel

Secondly, different combination strategies of the modules are crucial. For example, in the CBAM [27] method, the best combination strategy is obtained by comparing sequential combination and parallel combination, and the accuracy is further improved. In our paper, OSCGA is to aggregate the original spatial information and should be placed in the low layer of the network. HCGA is used for the global response to high-level channel features to strengthen important semantic features, and it should be placed after InceptionV3. Meanwhile, researches such as S3D [23] have found that the processing of temporal features is generally more effective at high levels. Therefore, the TEA module is also placed after InceptionV3. We strive to achieve positive results through a great combination strategy.

Since both HCGA and TEA are placed on the high level of the network, we verify that our strategy that combines HCGA and TEA modules in parallel is effective by experiments. We conduct experiments on split1 of UCF101. In Table 2, we find that if the HCGA and TEA modules are combined in sequence, compared to models that without

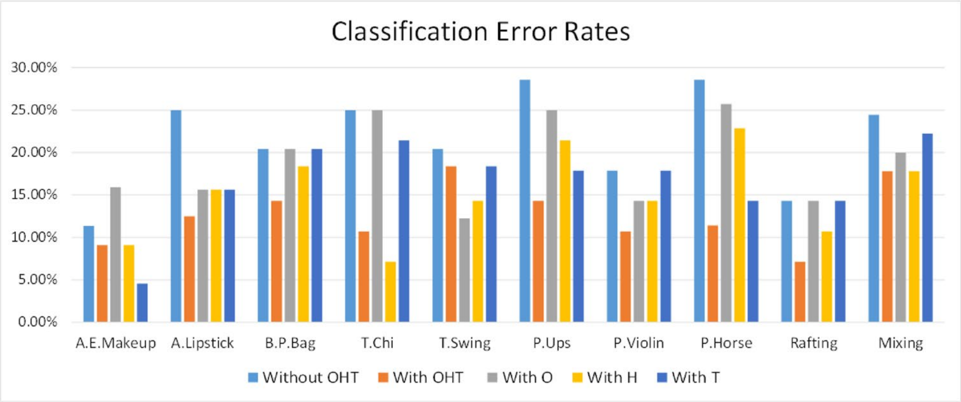


Fig. 8 Illustration of classification error rates. O stands for the OSCGA module, H stands for the HCGA module, and T stands for the TEA module. The light blue bars represent the recognition error rates in categories obtained by the network without using three attention modules. The orange bars represent the recognition error rates achieved by the network using all the three attention modules. The grey bars indicate the recognition error rate using only OSCGA modules. The yellow bars indicate the recognition error rates using only the HCGA module, and the dark blue bars represent using only the TEA module. These categories are ApplyEyeMakeup, ApplyLipstick, BoxingPunchingBag, TaiChi, TennisSwing, PullUps, PlayingViolin, PommelHorse, Rafting, Mixing

Table 3 Top-1 accuracies on UCF101 dataset show the influence of group numbers (N) in two-stream network with OSCGA (O) module, HCGA (H) module and TEA (T) module

N	Two-stream accuracy (%) (O+H+T)
1	94.87
4	95.01
8	94.90
16	94.71
32	94.90
64	94.50

two attention modules, there is a little improvement. However, when the parallel combination is applied, the accuracy of the entire model is greatly improved (88.05% vs. 87.32% in RGB appearance stream). It shows that enhancing spatial information and temporal information at the same time can improve the expressiveness of high-level features.

In the previous experiments, we have verified the effectiveness of the HCGA module. By using the HCGA module, we achieve higher accuracies. Next, we discuss the influence of group numbers in the HCGA module. Different numbers of groups are tested in the two-stream network with three spatiotemporal modules to find out which achieves better results. As shown in Table 3, we find that the best accuracy is obtained when the number of groups is 4. As the number of groups increases, the recognition effect continues to deteriorate. Because when the number of groups increases, local features are enhanced in a small local area, which makes the difference between local features increase. When the number of groups is 4, the coordination between the ability to mine local features and the enhancement of different effective local features is optimal.

For qualitative analysis on three spatiotemporal attention modules, referring to CBAM [27], Grad-CAM is applied to our networks on the UCF101 dataset. Grad-CAM is a visualization method that uses gradients to calculate the importance of different spatial locations in convolutional layers. The Grad-CAM results can clearly show the important areas obtained by the network. By observing areas that the network considers are important for prediction, we try to study the ability of our designed attention modules on extracting features. In Fig. 9, we compared the visualization results of our attention integrated model (InceptionV3 + OSCGA + HCGA + TEA) to that of the original network (InceptionV3) in RGB stream. In the figure, we can see that the Grad-CAM masks of attention integrated network can cover the target object areas. It shows that the designed attention modules can use the information in essential areas and aggregate features from these areas.

4.3.2 Multi-level features fusion stream

In the MFF stream, features from different levels are combined to complement the original two-stream network. Block1 and Block2 are used to fuse the features. These two blocks have different functions, and we then verify the effectiveness of these two blocks. The results of the ablation experiments on the MFF stream are in the Table 4. After using Block1 in four layers, it has only a 0.15% improvement compared to the original method of adding the features up. But such an improvement is not enough. Considering the rapid attenuation of optical flow characteristics during training, we designed Block2. However, the effect of using Block2 on the four layers is not ideal. The above two experimental results show that these fusion methods ignore the

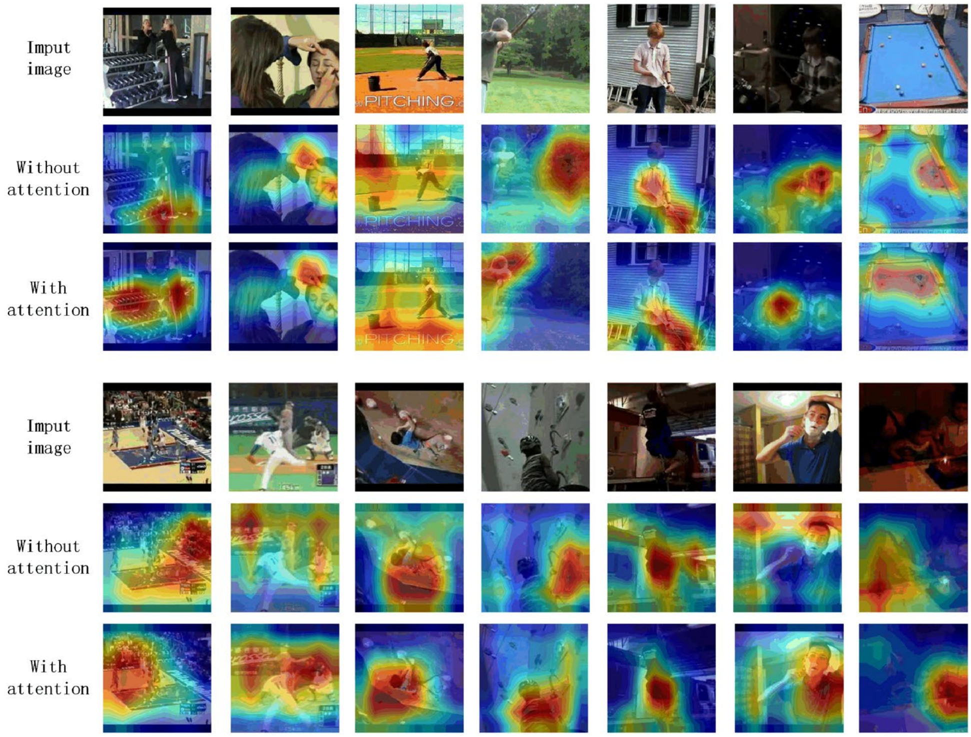


Fig. 9 Grad-CAM masks of the target object areas in RGB stream. The pictures in the first line are the original RGB images, the pictures in the second line are images with the Grad-CAM masks obtained by the original RGB stream, and the third line is images with the Grad-CAM masks obtained by the RGB stream with the attention modules

Table 4 Top-1 accuracies on UCF101 dataset show the influences of Block1 and Block2 in MFF stream

Models	UCF101 Video accuracy (%)
Basic fusion stream	91.58
Basic fusion stream (with 4 Block1)	91.73
Basic fusion stream (with 4 Block2)	90.71
Basic fusion stream (with 3 Block1)	92.45
Basic fusion stream (with 1 Block2)	92.77
Basic fusion stream (with 3 Block1 and 1 Block2)	93.27

difference in features between different levels, and there is a lot of interference information such as texture information in the shallower three layers. By using Block1 in the shallower high levels and adding the last-level features, the fusion stream gets 92.45% accuracy. It shows that the three high levels provide useful auxiliary judgment information, and at the same time, it has less interference with the final classification result. While only using one Block2 to fuse the features of the last layer, the accuracy reaches 92.77%, which shows that it is better to use Block2 for the features fusion of the last layer. Therefore, the shallower three layers extract auxiliary information by Block1, and the last-layer features obtained by the InceptionV3 backbone are fused with Block2 to get more significant effects. By using Block1 and Block2 together, better improvement is achieved (93.27%). In the process of fusion, the semantic information of features is enriched in the MFF stream. The visualization comparisons in RGB stream and our MFF stream are showed in Fig. 10. As can be seen in the figure, although the individual RGB streams can focus on important objects for recognition, the focused area is scattered. In the MFF stream, due to the fusion with optical flow motion characteristics, the focused area of network is more concentrated. For example, in the process of jumping the balance beam, the MFF network believes that the balance beam is a more important feature for recognition. This shows that using the MFF stream is effective to supplement the original stream.

4.4 Results of score fusion

In this section, we explore the impacts of different combinations of appearance stream and motion stream. The results in Table 5 show that the multi-stream network is more efficient than the single stream network. By fusing scores from different streams, the network can make a better judgment. Therefore, we compare the combinations of RGB stream, optical flow stream, optical-flow-like stream, and MFF stream.

In the two-stream network, RGB frames are used to express appearance information in the appearance stream. To express motion information, two different motion representations are used. One is an optical flow pattern extracted by TV-L1, and the other is an optical-flow-like pattern extracted by TVNet [37]. With the same number of segments divided, the prediction result of a single optical flow motion stream

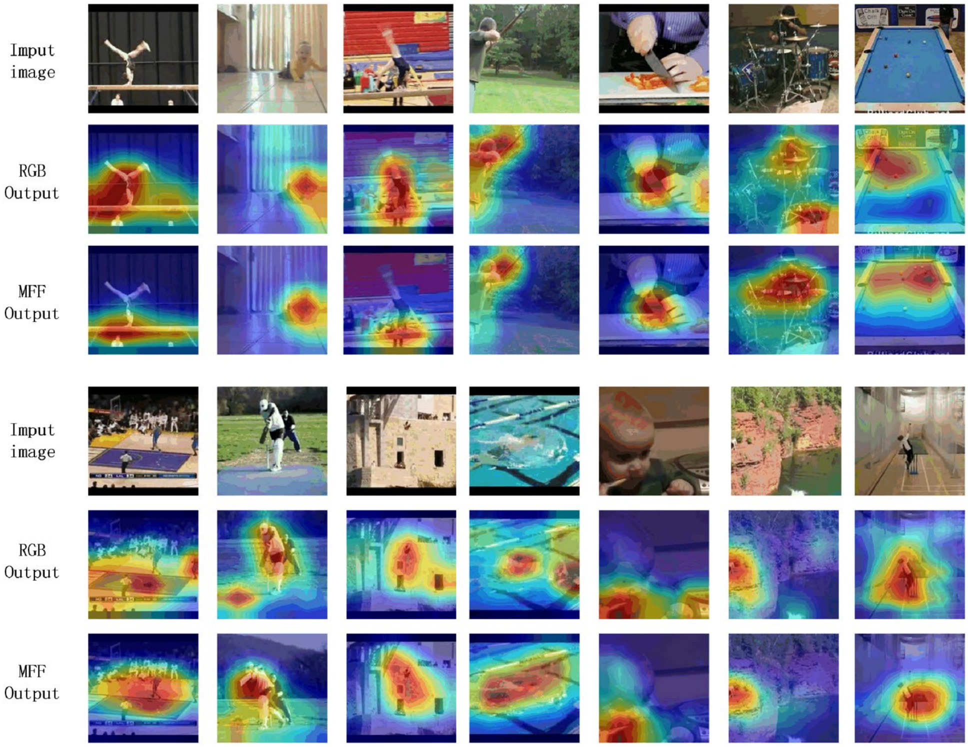


Fig. 10 Grad-CAM masks of the target object areas in the RGB appearance stream and MFF stream. The first line is the original RGB images, the second line is the images with the Grad-CAM masks in the appearance stream. The third line is the images with the Grad-CAM masks in the MFF stream. It shows that the network of the MFF stream focuses on the important areas better

Table 5 Top-1 accuracies on UCF101 dataset show the result of single stream and fusion results of different combinations of RGB stream (RGB), optical flow stream (Optical flow), optical-flow-like pattern stream (TVNet) and MFF stream (MFF)

Streams	UCF101 Video accuracy (%)
RGB	88.01
Optical flow	89.90
TVNet	86.15
MFF	93.27
RGB + Optical flow	95.01
RGB + TVNet	94.06
RGB + Optical flow + TVNet	95.10
RGB + Optical flow + MFF	95.20

Table 6 Results of different weights of RGB appearance stream, optical flow motion stream and MFF stream (Three-stream) in the weighted fusion method (weighted)

Models	UCF101 Video accuracy (%)
Three-stream + weighted (0.5:0.5:1)	93.73
Three-stream + weighted (0.5:1:0.5)	94.24
Three-stream + weighted (0.5:1:1)	93.61
Three-stream + weighted (1:0.5:0.5)	95.20
Three-stream + weighted (1:0.5:1)	94.71
Three-stream + weighted (1:1:0.5)	95.01
Three-stream + weighted (1:1:1)	94.78
Our model	**95.20**

Bold value indicates the best score of the weighted fusion method

is better than that of an optical-flow-like motion stream. The accuracy of the two-stream network with the optical flow motion pattern is nearly 1% higher than that with an optical-flow-like motion pattern. Therefore, in the basic two-stream network, we still use optical flow as the motion representation.

Based on the two-stream network with RGB pattern and optical flow pattern, the MFF stream and TVNet optical-flow-like motion stream are taken into account to supply more effective features. We perform experiments to find out which approach is more effective. In the table, we can see that both of the two streams have a little improvement. The three-stream network using TVNet optical-flow-like motion pattern is 0.1% higher than the two-stream network. And in the three-stream network with the MFF stream, the improvement is 0.2%. The optical-flow-like features extracted by TVNet do not perform well in two-stream and three-stream networks, while traditional optical flow performs well. This shows that the traditional optical flow is still superior in action recognition tasks and can represent motion features. At the same time, although the MFF stream can extract RGB appearance features and optical flow motion features at the same time, the result still cannot exceed the two-stream architecture. It shows that the separate RGB stream and optical flow stream can extract more targeted appearance features and motion features, so they cannot be discarded. The features extracted by the MFF stream contain additional information to supplement the original two streams.

In Table 6, the score fusions of different streams with different ratios are tried. RGB appearance stream, optical flow motion stream and MFF stream are chosen for score fusion because of the good fusion performance. The three-stream network includes spatiotemporal attention modules: OSCGA, HCGA and TEA. The highest precision is achieved in the weighted fusion method. In the score fusion, different streams have different importance degrees for predictions. Each stream is given a weight of 0.5 or 1 for fusion. When the ratio of RGB stream, optical flow stream and MFF stream is 1:0.5:0.5, the best fusion result is achieved in our experiment. This shows that the RGB features are the most important in the three-stream network. RGB features provide rich color information and detailed information. The optical flow stream and fusion stream can provide important motion and supplementary information.

4.5 State-of-the-art comparisons

First, we analyzed the statistics of ST-AEFFNet to evaluate the complexity of the model and compared it with other typical methods. Second, ST-AEFFNet can effectively improve the accuracies of action recognition and we compare this network with state-of-the-art methods.

4.5.1 Computational complexity

The comprehensive statistics such as corresponding FLOPs, run times, the number of network parameters and the classification results are important criteria for evaluating the model.

In Table 7, 3D-CNNs or the mixup of 2D and 3D CNNs such as C3D, S3D-G and I3D have excellent performance for action recognition in recent years. Due to the high computational cost of these networks, the FLOPs of these methods is usually higher than other methods, but their parameter amounts are relatively low. Compared to the 3D-CNNs, the algorithmic complexity of our ST-AEFFNet is lower. At the same time, it has achieved better performance in accuracy improvements. Among all the existing methods, the most typical one is TSN and its FLOPs is 33G. Compared to TSN, the proposed ST-AEFFNet uses the InceptionV3 backbone, which increases the algorithmic complexity, but its performance has been improved. Compared to R-STAN which also uses attention mechanisms to get useful features, ST-AEFFNet with the spatiotemporal attention mechanism has higher Flops and more parameters, but it improves the accuracy (88.01% vs. 86.62%).

4.5.2 Accuracy comparison

On the UCF101 dataset and HMDB51 dataset, we compare our network to state-of-the-art methods. By comparing these different methods, the effects of our model are shown more comprehensively. Our model is compared to manual feature-based methods, deep learning-based methods and attention mechanism-based methods. In our comparisons, spatiotemporal attention enhanced network (ST-AENet) represents original two-stream architecture with OSCGA, HCGA and TEA spatiotemporal modules. Our whole network ST-AEFFNet represents three-stream architecture with MFF stream and OSCGA, HCGA, TEA spatiotemporal modules. On the UCF101 dataset, ST-AENet achieves 95.0% and ST-AEFFNet improves the accuracy to 95.2%. The results of the UCF101 dataset are shown in Table 8. On the HMDB51 dataset, ST-AENet and ST-AEFFNet achieve 71.3% and 71.9% accuracy, respectively. The results on the HMDB51 dataset are in Table 9.

In recent years, some manual feature-based methods are still competitive. Compared to traditional methods such as IDT and MIFS, ST-AEFFNet has a significant advantage on the UCF101 dataset and HMDB51 dataset due to the diversity of data. Meanwhile, it is found that although the deep learning networks are very useful, they may still fail to pay attention to the most representative features for recognition during the process of feature extraction. By combining spatiotemporal attention modules on different levels, it can

Table 7 The comprehensive statistics comparisons with the classic models in RGB appearance stream, including corresponding FLOPs, run times, the number of network parameters and the classification results in RGB stream

Methods	Frames	Backbone	GFLOPs	Params (M)	Speed (fps)	UCF101 accuracy (%)
C3D-RGB [19]	16	–	38.5	–	20	85.2
Res3D-RGB [38]	12	–	19.3	–	106	85.8
TSN-RGB [23]	8	BNInception	33	24.3	14	86.4
I3D-RGB [20]	64	Inception	107.9	12.1	<14	84.5
R-STAN-RGB [39]	12	ResNet152	11.4	–	14	86.62
S3D-G [23]	64	Inception	71.4	11.56	–	–
ST-AEFFNet-RGB	8	InceptionV3	43.3	30.3	10	88.01

Table 8 Comparisons with the state-of-the-art models on the UCF101 dataset

Models	UCF101 Video accuracy (%)
IDT + FV [40]	85.9
IDT + HSV [41]	87.9
MIFS [42]	89.1
Two-stream (SVM) [8]	88.0
Two-stream LSTM [43]	88.6
TDD [44]	90.3
C3D [19]	85.2
TSN (2 modalities) [10]	94.0
ST-ResNet [17]	93.4
Convolutional two-stream [9]	92.5
ST-Pyramid network [45]	94.6
Multiplier [18]	94.2
CO2FI + ASYN [46]	94.3
Hidden two-stream [47]	93.1
STRN [48]	93.2
PBNet [49]	94.9
D-Res-LSTM [50]	92.3
STMN ResNet + IDT [51]	94.5
Key volume mining [52]	93.1
AdaScan + IDT + C3D [53]	93.2
CatNet Two-Stream [54]	93.8
Two-stream TCLSTA [55]	94.0
TS FCAN-comp [56]	92.0
Two-stream TCLSTA [57]	94.0
STAN [58]	93.6
R-STAN [39]	94.5
ST-HConvLSTM [59]	95.0
STACNet [60]	94.3
TFV + LDS [61]	91.6
DDSPP [62]	92.5
ST-AENet	**95.0**
ST-AEFFNet	**95.2**

We compare our model to manual feature-based methods, deep learning-based methods and attention mechanism-based methods. Spatiotemporal attention enhanced network (ST-AENet) represents original two-stream architecture with OSCGA, HCGA and TEA spatiotemporal modules and without MFF stream. The results of ST-AENet and ST-AEFFNet are listed in the table

Bold values indicate the best scores of our ST-AENet and ST-AEFFNet on UCF101 dataset

Table 9 Comparisons with the state-of-the-art models on the HMDB51 dataset

Models	HMDB51 Video accuracy (%)
IDT [40]	61.7
MIFS [42]	65.4
C3D [19]	51.6
TDD + IDT [44]	65.9
Transformation [63]	62.0
Two-stream fusion [9]	65.4
Two-stream fusion [9]+IDT	69.2
TSN [10]	69.4
ST-ResNet + IDT [17]	70.3
Res3D [38]	54.9
Long-term ConvNets [64]	70.3
ST Multiplier [18]	68.9
CO2FI + ASYN [46]	69.0
STRN [58]	64.9
DMC-Net [65]	61.8
IP TSN [66]	69.1
SemIN (2 modalities) [67]	70.1
Key volume mining [52]	63.3
AdaScan + IDT + C3D [53]	66.9
TS FCAN-comp [56]	66.7
Two-stream TCLSTA [55]	68.7
STACNet [60]	69.14
MFCD-Net [68]	66.9
XDC [69]	67.4
ST-AENet	**71.3**
ST-AEFFNet	**71.9**

We compare our model to manual feature-based methods, deep learning-based methods and attention mechanism-based methods. The results of ST-AENet and ST-AEFFNet are listed in the table

Bold values indicate the best scores of our ST-AENet and ST-AEFFNet on HMDB51 dataset

make the network focus on important areas. Many studies introduce various attention mechanisms to optimize features. These networks such as STRN and STACNet have introduced the spatiotemporal attention mechanism to improve the network. Compared to these models, our accuracy is improved, which shows the effectiveness of spatiotemporal attention modules with different functions.

5 Conclusion

Based on the two-stream network, a novel three-stream network architecture ST-AEFFNet with spatiotemporal attention mechanism enhanced and the MFF stream complemented is proposed. RGB appearance features and optical flow motion features are used in our network. Firstly, the MFF stream complements the two-stream network by combining different representative multi-level features to obtain supplemental information. To achieve better improvement in features fusion method, attention fusion function is added in the MFF stream. Secondly, we observe that original inputs have detailed information with noise, and features in high levels have rich semantic information. Therefore, the OSCGA module is designed to utilize the spatial context modeling of original inputs and guide the following network.

HCGA module is used to enhance high-level spatial semantic information by modeling the channel interrelationships. Thirdly, we propose a TEA module to improve the effectiveness of segment consensus by modeling temporal information and assigning different weights to different frames. We demonstrate the effects of MFF stream and spatiotemporal attention modules by performing a variety of experiments on the datasets. At the same time, compared with other advanced action recognition models, our network has good performance on both the UCF101 dataset and the HMDB51 dataset.

Acknowledgements This work was partially supported by the National Natural Science Foundation of China (61362030, 61201429), China Postdoctoral Science Foundation (2015M581720, 2016M600360), Jiangsu Postdoctoral Science Foundation (1601216C), Scientific and Technological Aid Program of Xinjiang (2017E0279), 111 Projects under Grant B12018.

References

1. Dollár P, Rabaud V, Cottrell G, Belongie S (2005) Behavior recognition via sparse spatio-temporal features. In: 2005 IEEE international workshop on visual surveillance and performance evaluation of tracking and surveillance. IEEE, pp 65–72
2. Laptev I, Marszalek M, Schmid C, Rozenfeld B (2008) Learning realistic human actions from movies. In: 2008 IEEE conference on computer vision and pattern recognition. IEEE, pp 1–8
3. Willems G, Tuytelaars T, Van Gool L (2008) An efficient dense and scale-invariant spatio-temporal interest point detector. In: European conference on computer vision. Springer, pp 650–663
4. Wang H, Kläser A, Schmid C, Liu CL (2011) Action recognition by dense trajectories. In: 2011 IEEE conference on computer vision and pattern recognition. IEEE, pp 3169–3176
5. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp 1097–1105
6. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
7. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861
8. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems. pp 568–576
9. 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
10. Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Van Gool L (2016) Temporal segment networks: towards good practices for deep action recognition. In: European conference on computer vision. Springer, pp 20–36
11. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 7132–7141
12. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 1–9
13. Mikolov T, Kombrink S, Burget L, Černocky J, Khudanpur S (2011) Extensions of recurrent neural network language model. In: 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 5528–5531
14. Luo J, Wu J, Zhao S, Wang L, Xu T (2019) Lossless compression for hyperspectral image using deep recurrent neural networks. Int J Mach Learn Cybern 1–11
15. Zhou T, Li Z, Zhang C (2019) Enhance the recognition ability to occlusions and small objects with robust faster r-cnn. Int J Mach Learn Cybern 10(11):3155–3166
16. Donahue J, Hendricks LA, Guadarrama S,Rohrbach M, Venugopalan S, Saenko K, Darrell T (2015) Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 2625–2634
17. Feichtenhofer C, Pinz A, Wildes R (2016) Spatiotemporal residual networks for video action recognition. In: Advances in neural information processing systems. pp 3468–3476
18. Feichtenhofer C, Pinz A, Wildes RP (2017) Spatiotemporal multiplier networks for video action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 4768–4777
19. 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
20. 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 6299–6308
21. Qiu Z, Yao T, Mei T (2017) Learning spatio-temporal representation with pseudo-3d residual networks. In: Proceedings of the IEEE international conference on computer vision. pp 5533–5541
22. Tran D, Wang H, Torresani L, Ray J, LeCun Y, Paluri M (2018) A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 6450–6459
23. Zhongxu H, Youmin H, Liu J, Bo W, Han D, Kurfess T (2018) 3d separable convolutional neural network for dynamic hand gesture recognition. Neurocomputing 318:151–161
24. Wang F, Jiang M, Qian C, Yang S, Li C, Zhang H, Wang X, Tang X (2017) Residual attention network for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3156–3164
25. Hu J, Shen L, Albanie S, Sun G, Vedaldi A (2018) Gather-excite: exploiting feature context in convolutional neural networks. In: Advances in neural information processing systems. pp 9401–9411
26. Zhao H, Zhang Y, Liu S, Shi J, Loy CC, Lin D, Jia J (2018) Psanet: point-wise spatial attention network for scene parsing. In: Proceedings of the European conference on computer vision (ECCV). pp 267–283
27. Woo S, Park J, Lee J-Y, Kweon IS (2018) Cbam: convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV). pp 3–19
28. Wang X, Girshick R, Gupta A, He K (2018) Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 7794–7803
29. Fu J, Liu J, Tian H, Li Y, Bao Y, Fang Z, Lu H (2019) Dual attention network for scene segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 3146–3154

30. Li X, Wang W, Hu X, Yang J (2019) Selective kernel networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 510–519
31. Li X, Hu X, Yang J (2019) Spatial group-wise enhance: improving semantic feature learning in arXiv preprint arXiv:1905.0964
32. Zhou P, Shi W, Tian J, Qi Z, Li B, Hao H, Xu B (2016) Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th annual meeting of the association for computational linguistics (vol 2: short papers). pp 207–212
33. Cao Y, Xu J, Lin S, Wei F, Hu H (2019) Gcnet: non-local networks meet squeeze-excitation networks and beyond. arXiv preprint arXiv:1904.11492
34. Bai S, Kolter JZ, Koltun V (2018) An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271
35. Bottou L (2010) Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010. Springer, pp 177–186
36. Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Antiga L, Lerer A (2017) Automatic differentiation in pytorch, Alban Desmaison
37. Fan L, Huang W, Gan C, Ermon S, Gong B, Huang J (2018) End-to-end learning of motion representation for video understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 6016–6025
38. Tran D, Ray J, Shou Z, S-F Chang, Paluri M (2017) Convnet architecture search for spatiotemporal feature learning. arXiv preprint arXiv:1708.05038
39. Liu Q, Che X, Bie M (2019) R-stan: residual spatial-temporal attention network for action recognition. IEEE Access 7:82246–82255
40. Wang H, Schmid C (2013) Action recognition with improved trajectories. In: Proceedings of the IEEE international conference on computer vision. pp 3551–3558
41. Peng X, Wang L, Wang X, Qiao Yu (2016) Bag of visual words and fusion methods for action recognition: comprehensive study and good practice. Comput Vis Image Underst 150:109–125
42. Lan Z, Lin M, Li X, Hauptmann AG, Raj B (2015) Beyond gaussian pyramid: multi-skip feature stacking for action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 204–212
43. Ng JY-H, 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
44. Wang L, Qiao Y, Tang X (2015) Action recognition with trajectory-pooled deep-convolutional descriptors. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 4305–4314
45. Wang Y, Long M, Wang J, Yu PS (2017) Spatiotemporal pyramid network for video action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 1529–1538
46. Lin W, Mi Y, Wu J, Lu K, Xiong H (2017) Action recognition with coarse-to-fine deep feature integration and asynchronous fusion. arXiv preprint arXiv:171107430
47. Zhu Y, Lan Z, Newsam S, Hauptmann A (2018) Hidden two-stream convolutional networks for action recognition. In: Asian conference on computer vision. Springer, pp 363–378
48. Liu Z, Hu H (2019) Spatiotemporal relation networks for video action recognition. IEEE Access 7:14969–14976
49. Huang W, Fan L, Harandi M, Ma L, Liu H, Liu W, Gan C (2018) Toward efficient action recognition: principal backpropagation for training two-stream networks. IEEE Trans Image Process 28(4):1773–1782
50. Song S, Liu J, Li Y, Guo Z (2020) Modality compensation network: Cross-modal adaptation for action recognition. IEEE Trans Image Process 29:3957–3969
51. Li C, Zhang B, Chen C, Ye Q, Han J, Guo G, Ji R (2019) Deep manifold structure transfer for action recognition. IEEE Trans Image Process 28(9):4646–4658
52. Zhu W, Hu J, Sun G, Cao X, Qiao Y (2016) A key volume mining deep framework for action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 1991–1999
53. Kar A, Rai N, Sikka K, Sharma G (2017) Adascan: adaptive scan aooling in deep convolutional neural networks for human action recognition in videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 3376–3385
54. Liu Z, Wang L, Zheng N (2018) Content-aware attention network for action recognition. In: IFIP international conference on artificial intelligence applications and innovations. Springer, pp 109–120
55. Peng Y, Zhao Y, Zhang J (2018) Two-stream collaborative learning with spatial-temporal attention for video classification. IEEE Trans Circuits Syst Video Technol 29(3):773–786
56. Tran A, Cheong L-F (2017) Two-stream flow-guided convolutional attention networks for action recognition. In: Proceedings of the IEEE international conference on computer vision. pp 3110–3119
57. Peng Y, Zhao Y, Zhang J (2019) Two-stream collaborative learning with spatial-temporal attention for video classification. IEEE Transactions on Circuits and Systems for Video Technology 29(3):773–786
58. Li D, Yao T, Duan L, Mei T, Rui Y (2019) Unified spatio-temporal attention networks for action recognition in videos. IEEE Transactions on Multimedia 21(2):416–428
59. Xue F, Ji H, Zhang W, Cao Y (2019) Attention-based spatial-temporal hierarchical convlstm network for action recognition in videos. IET Computer Vision 13(8):708–718
60. Liu S, Ma X, Wu H, Li Y (2020) An end to end framework with adaptive spatio-temporal attention module for human action recognition. IEEE Access 8:47220–47231
61. Luo G, Wei J, Hu W, Maybank SJ (2019) Tangent fisher vector on matrix manifolds for action recognition. IEEE Trans Image Process 29:3052–3064
62. Rahimi S, Aghagolzadeh A, Ezoji M (2020) Human action recognition using double discriminative sparsity preserving projections and discriminant ridge-based classifier based on the gdwl-l1 graph. Expert Syst Appl 141:112927
63. Wang X, Farhadi A, Gupta A (2016) Actions transformations. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 2658–2667
64. Varol G, Laptev I, Schmid C (2017) Long-term temporal convolutions for action recognition. IEEE Trans Pattern Anal Mach Intell 40(6):1510–1517
65. Shou Z, Lin X, Kalantidis Y, Sevillalara L, Rohrbach M, Chang S, Yan Z (2019) Dmc-net: generating discriminative motion cues for fast compressed video action recognition. pp 1268–1277
66. Huang S, Lin X, Karaman S, Chang SF (2019) Flow-distilled ip two-stream networks for compressed video action recognition. arXiv preprint arXiv:191204462
67. Khowaja SA, Lee SL (2020) Semantic image networks for human action recognition. Int J Comput Vis 128(2):393–419
68. Battash B, Barad H, Tang H, Bleiweiss A (2020) Mimic the raw domain: Accelerating action recognition in the compressed domain. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. pp 684–685
69. Alwassel H, Mahajan D, Torresani L, Ghanem B, Tran D (2019) Self-supervised learning by cross-modal audio-video clustering. arXiv preprint arXiv:191112667

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