Computer Science and Information Systems 19(3):1329–1348 https://doi.org/10.2298/CSIS220105043C

A Novel Motion Recognition Method Based on Improved Two-stream Convolutional Neural Network and Sparse Feature Fusion

Chen Chen

Sports Institute, Henan University of Technology
Zhengzhou City, 470001 China
byoungholee@qq.com

Abstract. Motion recognition is a hot topic in the field of computer vision. It is a challenging task. Motion recognition analysis is closely related to the network input, network structure and feature fusion. Due to the noise in the video, traditional methods cannot better obtain the feature information resulting in the problem of inaccurate motion recognition. Feature selection directly affects the efficiency of recognition, and there are still many problems to be solved in the multi-level feature fusion process. In this paper, we propose a novel motion recognition method based on an improved two-stream convolutional neural network and sparse feature fusion. In the low-rank space, because sparse features can effectively capture the information of motion objects in the video, meanwhile, we supplement the network input data, in view of the lack of information interaction in the network, we fuse the high-level semantic information and low-level detail information to recognize the motions by introducing attention mechanism, which makes the performance of the two-stream convolutional neural network have more advantages. Experimental results on UCF101 and HMDB51 data sets show that the proposed method can effectively improve the performance of motion recognition.

Keywords: motion recognition, two-stream convolutional neural network, attention mechanism, sparse feature fusion, low-rank space.

1. Introduction

Motion recognition is a challenging task. Influenced by various factors such as different illumination, complex backgrounds, multiple perspectives, and large intra-class differences [1,2], motion recognition algorithms are mainly divided into two categories: 1) based on traditional machine learning [3-5]; 2) based on deep learning [6-8]. The key of motion recognition algorithm based on traditional machine learning is feature extraction. In the process, it often takes effort to design features that meet the requirements and are easy to implement. However, its ability to represent motions is also limited by the extracted features. Deep learning-based motion recognition algorithms can automatically learn features. But it needs a lot of training data. The effectiveness of automatic feature extraction is closely related to network structure design and network parameter selection.

The most direct method of applying deep learning in motion recognition is to use convolutional neural network (CNN) to recognize each frame of a video, but this method does not take the motion information between continuous video frames into account. Ji et al., [9] proposed the concept of 3D convolution for the first time, and used 3D convolution kernel to extract spatial and temporal features for motion recognition. Feichtenhofer et al., [10] proposed a two-stream convolutional neural network for motion recognition, which was divided into two parts: spatial flow convolutional network and temporal flow convolution network. The spatial stream convolutional network took a single frame of RGB image as input to represent the static apparent information at a certain moment in the video. Time-flow convolutional network took several successive frames of optical flow images stacked together as input to represent the motion information of objects. Finally, the classification results of the two networks were fused to get the final results. This proposed model broke the leading position of improved dense trajectory extraction algorithm (IDT)[11] in the field of motion recognition.

Tran et al. [12] proposed a new 3D convolutional 3 dimension (C3D), where continuous video frames were stacked as the network input. The 3D convolutional kernel was used to make convolution in the cube formed after the stacking, which had more time dimensions than the 2D convolutional kernel. In this way, motion information could be obtained from continuous frames. The biggest advantage of this algorithm was that the recognition speed was much higher than that of the two-stream algorithm. So far, the motion recognition algorithm had formed two main directions: one was based on two-flow convolutional neural network for motion recognition; The other was based on 3D convolutional neural network for motion recognition.

At present, the mainstream motion recognition network input data sets are RGB images and optical flow images. For the spatial stream convolutional network, the input data is RGB images, and the initial spatial stream network adopts frame by frame input. However, the current publicly available data sets can often be identified by a single frame image. In this case, there is a lot of redundant information in the input of the spatial flow convolutional network. In order to reduce the frame by frame input redundancy between successive frames, Zhu et al. [13] put forward a key frame method to dig the decisive frame and key areas in video, which could improve the accuracy and efficiency. Although the extraction method of key frames could be integrated into one training network, it was similar to the object detection network RCNN, it first extracted the candidate boxes and then selected key frames. Kar et al. [14] proposed an ADASCAN feature aggregation method to judge the importance degree of different frames, and accordingly to achieve the purpose of improving accuracy and efficiency. The overall model of this method was simpler than the previous one. For the time flow convolutional network, the input data was the optical flow image, and the optical flow extraction was time-consuming and labor-consuming. The motion features contained in the optical flow might not be the optimal features.

Many researchers have improved optical flow for motion recognition. Zhu et al. [15] proposed a dual-flow convolutional network, which added MotionNet before the time-flow network to generate optical flow images and served as the input of the time-flow convolutional network. This method improved the quality of optical flow. Sevilla-Lara et al. [16] proved that the optical flow was effective for motion recognition because of its invariant apparent features, and the end-point-error (EPE) had no strong correlation with the accuracy of motion recognition. It could be seen from the tested optical flow algorithm that the accuracy of optical flow at the boundary and small displacement had a strong correlation with the performance improvement of the motion recognition algorithm. Meanwhile, the loss function value of motion recognition was used to improve the optical flow, so that the recognition accuracy could be improved. Similarly, due to the disadvantages of optical flow images, many researchers have done some works in finding features that can replace optical flow. Zhang et al. [17] used motion vectors to replace optical flow. The motion vector was originally used for video compression, and it could be extracted directly without extra calculation, which greatly accelerated the recognition speed of the two-stream convolutional network, but the accuracy was reduced. Choutas et al. [18] proposed a new posture feature, which could be used for motion recognition by extracting the trajectory of the key joints of the human body, which formed posture feature map for motion recognition. It was complementary to the features provided by RGB and optical flow images, but performed poorly with single feature. Only by changing the interaction mode of dual-stream network and extracting new motion features as network input, the problem of accuracy and speed could not be solved at the same time. The change of network structure also played a decisive role in the improvement of algorithm performance.

In recent years, the main structure of motion recognition network is based on dual-stream network and 3D convolution network. Wang et al. [19] proposed a temporal segment network (TSN), which used multiple dual-stream networks to extract and fuse short-term motion information at different timing positions, so as to solve the problem that the traditional dual-stream only paid attention to apparent features and short-term motion information. Lan et al. [20] inherited the excellent characteristics of TSN and carried out weighted fusion for short-term motion information at different temporal positions. Zhou et al. [21] proposed the temporal inference network, which was based on TSN and added the three-layer fully connected network to learn the weight of video frames with different lengths, and carried out temporal inference for video frames with different lengths. Finally, it obtained the results by fusion. Xu et al. [22] proposed R-C3D(region convolutional neural network) by combining C3D and Faster-RCNN[23]. R-C3D used 3D convolution to extract video features. The idea of the Faster-RCNN is adopted, that is, the proposal is first generated, then the candidate region was pooled. Finally, the classification and boundary regression were performed. The network could recognize the behavior of video with any length. The speed was high and accuracy was improved. Chen et al. [24] modified the 3D convolution used in behavior recognition and proposed P3D residual net (pseudo 3D residual Net). 133 convolution and 311 convolution were used to replace 333 convolution. The former was similar to 2D convolution to extract spatial flow features, while the latter was used to obtain temporal flow features. This method greatly reduced the amount of calculation. Two-stream convolutional network and 3D convolutional neural network could extract time stream information, while long-short-term-memory (LSTM)[25] could also conduct time dimension modeling, which was also a popular direction in the field of motion recognition at present. Jiang et al. [26] proposed a multi-modal LSTM structure combining with attention mechanism with high stability. Du et al. [27] introduced the attitude attention mechanism combining with LSTM and CNN structure, which could effectively extract space-time features. In addition, other researchers have studied the common deep networks. Duan et al. [28] proposed a new non-local network structure, which regarded non-local operations as an efficient, simple and universal component that could be used to capture long-distance dependencies in neural networks. Deep learning algorithms mainly include dual-stream structure and 3D convolution, the dual-stream structure has high accuracy and slow speed. However, the 3D convolution is faster and has slightly low accuracy. They are all higher than the traditional machine learning algorithms on the whole. It has great advantages over traditional algorithms in dealing with complex backgrounds and large changes within the class.

Aiming at the limitation that the input data of mainstream two-stream convolution network is RGB image and optical flow image, this paper uses sparse features in low-rank space to effectively capture the information characteristics of motion objects in the video and supplement the network input data. At the same time, in view of the lack of information mutual characteristics in the network, we combine the high-level semantic information and low-level detail information to jointly identify motions, so that the network performance has more advantages.

The main contribution of this paper has three aspects:

- We research the dual-stream convolutional neural network based on temporal segmentation network, a temporal segmentation network combining sparse features is proposed to better focus on motion objects.
- A multi-layer feature fusion temporal segmentation network for motion recognition is proposed to solve the problem of low feature utilization.
- We fuse the high-level semantic information and low-level detail information to recognize the motions, which makes the performance of the two-stream convolutional neural network have more advantages.

2. Related Works

2.1. Two-stream convolutional neural network

Two-stream convolutional neural network is divided into spatial flow convolutional neural network and time flow convolutional neural network. The two convolutional neural networks process spatial dimension and temporal dimension of video, and extract spatial information and temporal information, respectively. The basic structure of two-stream convolutional neural network is shown in figure 1. Here, spatial information refers to the scene, object and other information in the video. Time information refers to the motion information of objects in the video.

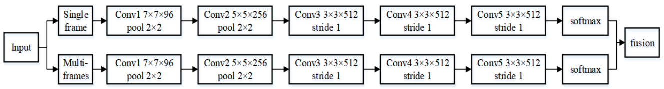


Fig. 1. Two-stream convolutional neural network

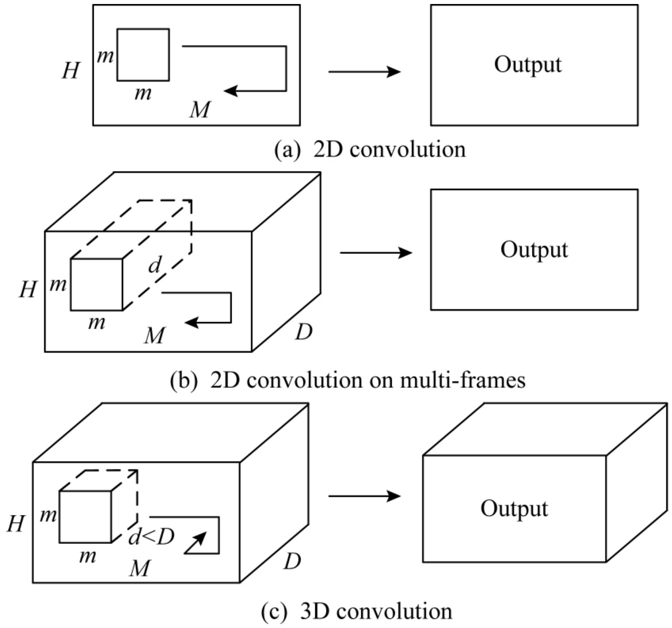
The input of spatial flow convolutional neural network is a single frame RGB image, which can effectively recognize human motion in static image. The network structure is similar to the commonly used image classification network, usually using Alexnet, VGG16, GoogleNet and other deep models as spatial flow convolutional neural network. Generally, pre-training is performed on ImageNet, and then the pre-trained parameters are transferred to spatial flow network to improve the speed and performance of network training. The input of the time stream convolutional neural network is the stacked continuous frame optical flow image, which can represent the motion information of the object in the video. It is a way to show the motion of the object by using the change of pixel in the time domain and its correlation in the continuous frame. By using this characteristic of optical flow, the human motion between successive frames can be recognized effectively. In order to match the feature dimension of spatio-temporal network fusion, the structure of time-flow network is usually the same as that of spatial flow convolution network.

The fusion of two-stream network refers to the fusion between spatial flow network and time flow network, which is generally divided into two forms. 1) The spatial flow and temporal flow convolution networks carry out result fusion after their Softmax layers. Usually, the average method and the weighted method are used to fuse the scores of different categories to get the final result. 2) The spatio-temporal network is fused at the middle feature layer. Generally, a hybrid spatio-temporal convolution network is formed after the fusion of spatio-temporal features at a certain network layer. Another fusion method is to retain pure spatial flow convolutional network or temporal flow convolutional network after forming a mixed spatio-temporal convolutional network. After the Softmax layer, the scores of different categories are fused again to get the final result.

2.2. 3D convolutional neural network

The most direct way to use convolutional neural network in video sequence is to use convolutional neural network to recognize each frame image. However, the processing of single frame image does not consider the information between successive frames. In motion recognition, the occurrence of motion generally lasts a process, and there is motion information between successive frames. Therefore, in order to effectively utilize the motion information between successive frames, reference [29] proposed a 3D convolutional neural network method, that is, 3D convolutional kernel was used in the structure of convolutional neural network for convolution. Compared with 2D convolution kernel, 3D convolution kernel increases the time dimension and can obtain the features of both time and space dimension at the same time, which is better than 2D convolution in the aspect of motion recognition feature representation. 2D convolution is carried out on the basis of a single frame image. Usually, a convolution kernel with the size of 33 is selected, and 2D convolution is applied to the image to output the image. Therefore, 2D convolution network will lose the time information of input signal after each convolution operation. 3D convolution is carried out on several adjacent frames, and the size of the convolution kernel is generally 333. Only 3D convolution can retain the time information of input signal, as shown in figure 2.

3D convolutional neural network reflects the time dimension by stacking multiple continuous image frames together to form a cube, and then using 3D convolutional kernel for convolution in the cube. The depth of the convolutional kernel is less than the number of stacked image frames. Therefore, each feature in 3D convolution is connected by the features of adjacent frames, and the representation on successive frames can obtain the motion information of objects in the video.



(a) 2D convolution

(b) 2D convolution on multi-frames

(c) 3D convolution

Fig. 2. 2D convolution and 3D convolution

2.3. Temporal segmentation network

Given one video V, it is divided into K segments $S_1, S_2, \cdots, S_K$, each segment is the same, then the temporal segmentation network can be expressed as:

$$Q_{TSN}(T_1, T_2, \cdots, T_K) = H(g(F(T_1; W), F(T_2; W), \cdots, F(T_K; W))). \quad (1)$$

Where $(T_1, T_2, \cdots, T_K)$ is a sequence composed of a single frame in video V. T_k is generated by random sampling of frames in its corresponding video sub-segment S_k, $k \in 1, 2, \cdots, K$. $F(T_k, W)$ is the input score prediction function belonging to different categories, that is, the video frame T_k gets a C-dimension vector through the convolution neural network with parameter W. It represents the predicted number of motions that T_k belongs to class C. $g(\cdot)$ is a segment consensus function, and the prediction results obtained by multiple sub-videos via convolutional neural network are fused to obtain the consistent prediction results $G = (G_1, G_2, \cdots, G_C)^T$ about the categories of videos. C represents the number of categories. Based on the above consistent prediction results, function $H(\cdot)$ is used to predict the probability of the entire video belonging to each behavior category. In here, $H(\cdot)$ uses Softmax function, the category with the highest probability is the category that video V belongs to. Combined with the cross-entropy loss commonly used in classification, the category prediction loss function of the final video V can be expressed as:

$$L(y,G) = -\sum_{i=1}^{C} y_i (G_i - \sum_{j=1}^{C} expG_j). \tag{2}$$

Where, y_i represents the true value of category i. This temporal segmentation network is differentiable determined by the function $g(\cdot)$. The back propagation algorithm and multiple sub-video frames can be used to jointly optimize the model parameter W. G is the consistent prediction result. In the process of back propagation, the gradient of model parameter W for the loss value L is:

$$\frac{\partial L(y,G)}{\partial W} = \frac{\partial L}{\partial G} \sum_{k=1}^{K} \frac{\partial G}{\partial F(T_k)} \frac{\partial F(T_k)}{\partial W}. \tag{3}$$

Where, K is the number of sub-video segments used by TSN. TSN learns model parameters from the entire video. At the same time, a sparse time sampling strategy is adopted for K, in which the sampled segment only contains a small portion of frames. Compared with the previous method using dense sampling frames, this method greatly reduces the computational overhead, and the structure of timing segmentation network is shown in figure 3.

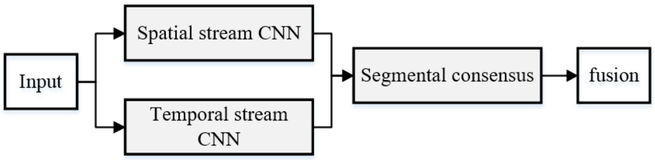


Fig. 3. Temporal segment network

3. Proposed Motion Recognition Method

This section will detailed introduce the network input data and network structure from two aspects. 1) The network input data of sparse features fusion is studied. The purpose is to focus the sparse features on the foreground target in the video, which can effectively extract the motion objects in the image, reduce the redundant information, and complement the information contained in RGB image and optical flow image. 2) It uses convolutional neural network visualization to verify that the shallow convolution can extract detailed features and deep convolution can extract semantic features. The combination of semantic information of high level features and detailed information of low level features in the deep network, and the complementary advantages of features between different convolution layers are helpful for the network to capture the overall features of human behavior and the detail features between different categories. It can improve the accuracy of motion recognition. Figure 4 is the flow chart of the proposed algorithm. The specific steps are as follows: (1) Evenly dividing the input video into three sub-videos, randomly sampling the three sub-videos to obtain the RGB, optical flow and sparse images of the samples, and input them into the convolutional network respectively; (2) Extracting the features of different convolutional layers of each data type, and fusing the features extracted by the convolutional network according to different sample types; (3) Using the Softmax function for motion classification.

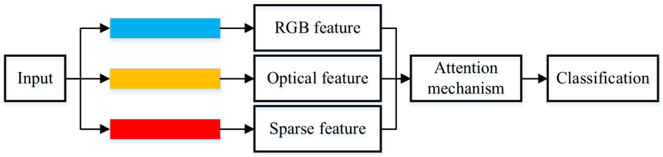


Fig. 4. Proposed temporal segment network based on feature fusion for motion recognition

3.1. Sparse feature

In many practical applications, the known data matrix D is often low-rank or approximately low-rank, but there are errors with arbitrarily large random amplitudes and sparse distribution, which destroy the low-rank of the original data. In order to restore the low-rank structure of matrix D, the matrix D can be decomposed into the sum of two matrices, that is, $D = A + E$, where matrices A and E are unknown, but A is low-rank and E is sparse.

When the elements of matrix E obey the independent Gaussian distribution, the classical principal component analysis method can be used to obtain the optimal matrix A, which is transformed into an optimization problem.

$$min_{A,E} = ||E||_F, s.t.rank(A) \leq r, D = A + E. \tag{4}$$

Where $||\cdot||_F$ denotes the Frobenius norm of the matrix.

When E is a sparse large noise matrix, PCA cannot give ideal results. So the robust principal component analysis (RPCA) can be used to obtain the optimal matrix A, then the problem in equation (4) can be transformed into an optimization problem:

$$min_{A,E} rank(A) + \lambda ||E||_0, s.t.D = A + E. \tag{5}$$

Where the rank function $rank(\cdot)$ and 0-norm of matrix are non-convex. They become NP-hard problem, which needs to be relaxed. Since the kernel norm is the convex hull of the rank function, and the 1-norm is the convex hull of the 0-norm, the NP-hard problem of equation (5) can be transformed into a convex optimization problem after relaxation:

$$min_{A,E} ||A|| + \lambda ||E||_1, s.t.D = A + E. \tag{6}$$

Where A is the low-rank component and E is the corresponding sparse component. $||\cdot||_*$ represents the kernel norm of the matrix, it is the sum of the singular values of the matrix, and it is also the convex approximation of $rank(\cdot)$. $||\cdot||_1$ denotes the L1 norm.

$|| \cdot ||_1$ is a weighted parameter greater than zero to balance the two norms. Under certain conditions, it has been proved that as long as the error matrix E is sparse enough relative to matrix A, the low-rank component and the sparse component can be accurately recovered from the matrix D by solving the convex optimization problem (equation (4)), that is, the weighted combination of the above kernel norm and L1 norm can be minimized.

For the RPCA problem described in equation (6), the augmented Lagrange multiplier method can be used to optimize it. The Lagrange function is:

$$L(A, E, Y, \mu) = ||A||_8 + \lambda||E||_1 + < Y, D - A - E > + \frac{\mu}{2}||D - A - E||_F^2. \quad (7)$$

Where Y is the Lagrange multiplier and μ is a smaller positive number.

RPCA is widely used in image and video processing such as image correction, denoising, video background modeling, foreground target extraction, image segmentation, saliency detection [32]. For foreground target segmentation in video, the background is approximated as a low-rank component due to the correlation between frames. However, the foreground target only occupies a small part of the pixels in the image, such as the human motion. The moving part can be regarded as the sparse component. Through the above augmented Lagrange multiplier method to solve the RPCA problem, the sparse features shown in figure 5 can be obtained for the motion video.

In Figure 5, the first row represents RGB image, the second row represents the motion optical flow image in the x-axis direction, the third row represents the motion optical flow image in the y-axis direction, and the fourth row represents the sparse image. As can be seen from figure 5, the RGB image represents the apparent features of the image, including both background and foreground targets. The optical flow image represents the movement direction and speed of the moving object in the image. For the x-axis direction, white indicates the movement to the right, and the higher gray value denotes the faster speed. black indicates the movement to the left, and the lower gray value denotes the faster movement. The rest of the gray area means that nothing is moving. It is the same as the y-axis, white means moving up, black means moving down. Unlike color and optical flow images, sparse feature images can focus on the behaviors of foreground targets and effectively extract motion objects. Meanwhile, removing background can effectively reduce data redundancy and significantly improve the speed of network training.

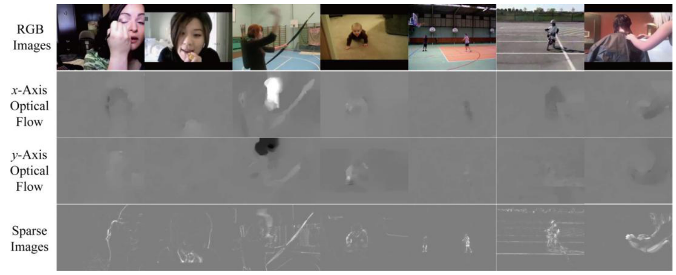


Fig. 5. Comparison of RGB, optical flow and low rank data

3.2. Network feature fusion with attention mechanism

In view of the lack of information interaction in the deep network, the deep network combines the high-level semantic information and low-level detail information to jointly identify the motion, so that the network performance has more advantages.

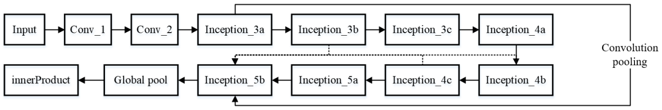


Fig. 6. Multilayer convolutional feature neural network

Multi-layer feature fusion is based on the low-level detail feature and high-level semantic feature of convolutional neural network, and the features of different deep convolutional layer features are used to achieve the fusion. We take InceptionV2 network as an example to illustrate the principle of the improved convolutional neural network as shown in figure 6. The network is composed of multi-stream convolutional neural networks. For the spatial-flow convolutional neural network, assuming that the input color image size is $224 \times 224 \times 3$, the convolution kernel of 7×7 and the step size of 2 are first selected. The convolution layer is used to extract the features of the input image, and $64 \times 112 \times 112$ feature maps are obtained. Then, the maximum pooling is performed to obtain the 5656 feature map. The convolution kernel of 3×3 and step size of 2 is selected, and the pooling features are extracted by reconvolution and pooling. The pooled feature size is $28 \times 28 \times 192$. Then, the obtained features are successively passed through 10 Inception structural units, from structural units Inception3a to Inception5b, and the size of the obtained features is $7 \times 7 \times 1024$. After one average pooling, it outputs the feature with $1 \times 1 \times 1024$, the 1D vector is expanded as the input of the fully connected layer. At the same time, the output features after shallow convolution are also expanded as 1D vectors and sent to the fully connected layer. Finally, the shallow convolution features and deep convolution features are input into the fully connected layer to form a 1×101 vector.

As shown in figure 6, the fusion process of multi-layer convolution features is illustrated by taking the output features of inception3a layer and Inception5b as an example. In order to clearly illustrate the fusion principle of features with high and low dimensions, table 1 lists the feature size output by each layer of the convolutional neural network.

Firstly, $28 \times 28 \times 192$ feature map is obtained after the input image goes through the first two convolution layer and pooling layer. The first 2-dimensional data represents the length and width of the feature map, and the third dimension data represents the number of channels. Then, the features are fed into the Inception3a layer, and four groups of features are obtained respectively through the four branches of the Inception structure unit. The four groups of features are connected in series as the input of the next layer. Meanwhile, the pooling operation is used for this feature. We select the average pooling, because it can reduce dimension and retain more image background information. It is beneficial to transfer the information to the next module for feature extraction, and make its size the same as the deep convolution feature size, which is convenient for feature fusion. In addition, because feature fusion will increase feature dimension and computational complexity, the

Table 1. Map size in each network layer

Network layers	Kernel size	Stride	Output size
Convolution-1	7×7	2	$112 \times 112 \times 64$
Pooling	3×3	2	$56 \times 56 \times 64$
Convolution-2	3×3	1	$56 \times 56 \times 192$
Pooling	3×3	2	$28 \times 28 \times 192$
Inception3a	——	—-	$28 \times 28 \times 256$
Inception3b	——	—-	$28 \times 28 \times 320$
Inception3c	2	2	$28 \times 28 \times 576$
Inception4a	——	—-	$14 \times 14 \times 576$
Inception4b	——	—-	$14 \times 14 \times 576$
Inception4c	——	—-	$14 \times 14 \times 576$
Inception4d	——	—-	$14 \times 14 \times 576$
Inception5a	——	—-	$7 \times 7 \times 1024$
Inception5b	——	—-	$7 \times 7 \times 1024$
Pooling	——	—-	$1 \times 1 \times 1024$

shallow convolution features are obtained by reducing dimension with convolution kernel of 1×1. The shallow convolution feature is connected in series with the output feature of inception5b layer and expanded into a 1-dimensional vector as the input of the fully connected layer. Time flow convolutional network and sparse convolutional neural network are similar to spatial flow convolutional neural network. Shallow convolutional features are obtained according to the above networks, and they are fused with the deep convolutional features output by the last layer of Inception structural unit to participate in the final classification. For the two feature maps $x_t^a \in R^{H \times M \times D}$ and $x_t^b \in R^{H' \times M' \times D'}$, they are will be utilized to generate the various feature map $y_t \in R^{H'' \times M'' \times D''}$. In here, t is the time. H, M and D represent the height, width and channel number of the three feature maps respectively. Because the cascade fusion is simple and efficient, this paper uses the cascade fusion method to fuse the low-level detail information and the high-level semantic information. The low-level detail information mainly extracts the color, texture and other detail features. While the high-level semantic information is more representative, it can extract the detailed feature for motion recognition. Through fusion process, features can be fully utilized to improve the recognition accuracy.

Series fusion means that the feature maps of the corresponding channels of the two features are connected in sequence, and the combined features are taken as new features, namely:

$$y_{i,j,2d} = x^a_{i,j,d}. \tag{8}$$

$$y_{i,j,2d-1} = x^b_{i,j,d}. \tag{9}$$

Where $1 \leq i \leq H$, $1 \leq j \leq M$, $1 \leq d \leq D$, and , $x^a, x^b \in R^{H \times M \times D}$, $y \in R^{H \times M \times D}$.

Drawing on the signal processing mechanism of the human brain, the attention mechanism quickly scans all features to obtain the feature categories that need to be focused on, and assigns corresponding attention weights according to the critical degree of feature categories, so that the brain can process huge information with limited resources. The application in CNN is reflected in the difference of importance of each generated feature map. The core objective of attention mechanism is to obtain the difference of importance between each feature map by calculation, allocate computing resources according to its importance, and use the execution effect to guide the feature in reverse map weights are updated, and the task is finally completed efficiently and accurately. The implementation principle of attention module is shown in Figure 7.

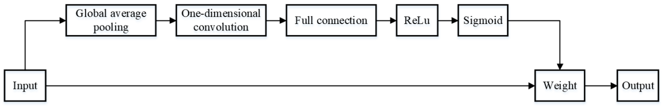


Fig. 7. Schematic of attention mechanism

Specific implementation methods are as follows. Firstly, each feature map obtained by convolution is global average pooling operation, and each feature map is extruded into a real number, as shown in Equation (10). The squeezed real numbers of each feature map are combined into a vector, namely the weight of each feature. After the weight vector is obtained, the full connection, ReLu activation function and sigmoid activation function are shown in Equation (11). The weighted feature map is obtained by assigning weights to each feature category. Finally, equation (12) is used to guide the feature map to update in a direction conducive to the recognition task.

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(ij). \tag{10}$$

$$s = \sigma(W_2 \delta(W_1 z)). \tag{11}$$

$$x_c = s_c u_c. \tag{12}$$

Where H and W represent the length and width of feature map. u_c stands for the result after convolution. z_c represents the importance of each feature map. s_c is the weight vector of all feature maps. σ stands for Relu activation function. δ represents Sigmoid activation function. W_1 and W_2 are two different fully connected operations.

4. Experiments and Analysis

In this section, experiments are conducted on two large motion datasets to verify the effectiveness of the feature fusion temporal segmentation network. The two datasets are UCF101 and HMDB51 respectively. UCF101 dataset contains 101 motion categories and 13320 video clips. The HMDB51 dataset is a large number of realistic videos from a variety of sources, such as movies and web videos. The dataset consists of 6849 video clips from 51 motion categories. The experiment followed the original evaluation scheme using three training/test groups, namely dataset group 1, dataset group 2, and dataset group 3. It takes the average accuracy of these groups as the final motion recognition accuracy.

The experiment in this section uses the small-batch random gradient descent algorithm to learn network parameters. The batch size is set to 32 and the momentum is set to 0.9. In addition, the model trained in advance with the dataset is used to initialize the network weights, and a small learning rate is set in the experiment. For spatial network, the learning rate is initialized to 0.001 and reduced by 1/10 per 2000 iterations. The entire training process stops at 10000 iterations. For temporal network and sparse network, the initial learning rate is set to 0.005, which is reduced to 110 after 12000 and 18000 iterations. The maximum iteration number is set to 20000. In order to extract optical flow quickly, the TVL1 optical flow algorithm implemented by CUDA in OpenCV is selected. To speed up the training, multiple GPUs are adopted implemented by using Caffe and OpenMP12.

4.1. Experiment datasets

UCF101 has 13320 videos including 101 categories. There are great changes in the aspects of camera movement, object appearance and posture, object proportion, viewpoint, messy background, lighting conditions, etc,. In addition, the motion videos are all edited rather than performed by actors, which is the most challenging dataset to a certain extent. Some motion categories in the dataset are shown in figure 8.

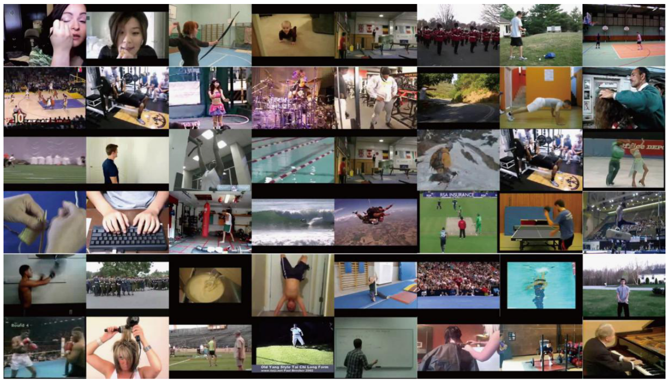


Fig. 8. Partial action categories in UCF101 dataset

The HMDB51 dataset contains 6766 video clips divided into 51 motion categories. Each action category contains at least 101 video clips. Some of the motion categories are shown in figure 9. Most samples of the HMDB51 dataset come from movies, while a small amount of samples come from public video sites such as Prelinger Archive, YouTube and Google Video.

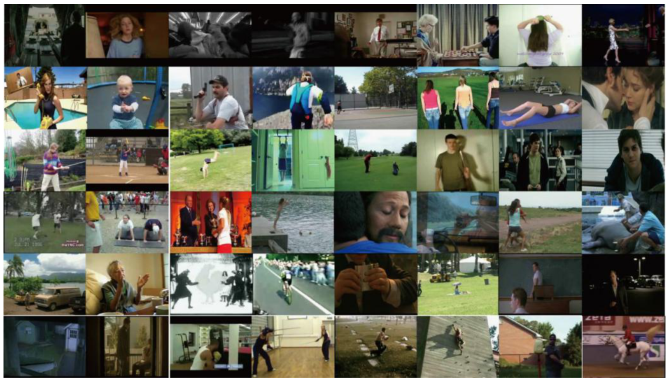


Fig. 9. Partial action categories in HMDB51 dataset

4.2. Experimental?Environment

Deep learning hardware environment: CPU E5-2696V4, GPU, two GTX1080Tis, 256 GB SSD, 32 GB memory. Network learning and test environment: Ubuntul6.04, NVIDIA CUDA8.0, cudnnv5, Caffe, opencv3.0, Python.

4.3. Effect of sparse feature

The experiment is conducted on the two public motion recognition datasets UCF101 and HMDB51, and we compare the proposed method with some classical algorithms and commonly used algorithms in recent years. The comparison results are shown in table 2.

As can be seen from table 2, the compared algorithms are divided into three categories. And figure 10 is the visual diagram for table 2. The first category is the traditional classical machine learning algorithm without deep learning. This algorithm manually extracts motion features and has high stability. The recognition rate can reach about 88% on UCF101 and exceed 61% on HMDB51. For example, the MoFAP method is a combinatorial motion feature way, which consists of three parts: local motion feature, motion atom, and motion statement. The motion atom refers to a certain sub-stage in the process of motion, and the motion statement is the combination of these sub-stages. For example, the high jump is divided into three sub-stages, the run-up, take-off and landing, namely the motion atom. The different combinations of the three become motion statement. In this way, the ability of features to represent motion is stronger, so as to improve the recognition accuracy.

The second category is the deep learning algorithm based on 3D convolution. This algorithm is fast, it can achieve real-time requirement, and the recognition rate is higher than that of the traditional algorithm. For example, reference [48] verified that different actions had different temporal and spatial patterns, and some motions could require a long time to be recognized. So LTC network structure was proposed to improve the recognition accuracy by increasing the duration of input video.

The third category is based on two-stream convolutional neural networks. This kind of algorithm has the highest accuracy, which can exceed 88%. As can be seen from table 2,

Table 2. Accuracy comparison with different methods on UCF101 and HMDB51/%

Method	UCF101	HMDB51
DT+MVSV[30]	83.7	56.1
IDT+FV[31]	86.1	57.4
IDT+HSV[32]	88.1	61.3
MoFAP[33]	88.5	61.9
C3D+IDT[12]	90.6	62.8
TDD+IDT[34]	91.7	66.1
LTC[35]	91.9	65.0
LTC+IDT[35]	92.8	67.4
P3D ResNet+IDT[36]	93.9	68.2
Two streams[37]	88.2	59.6
Two streams+LSTM[38]	88.8	63.4
Two streams fusion[10]	92.7	65.6
Transformations[39]	92.6	62.2
TSN(RGB+Optical Flow)[19]	94.2	69.4
RCA[40]	95.7	72.2
FFN[41]	96.9	74.3
Sparse+TSN	97.1	76.6

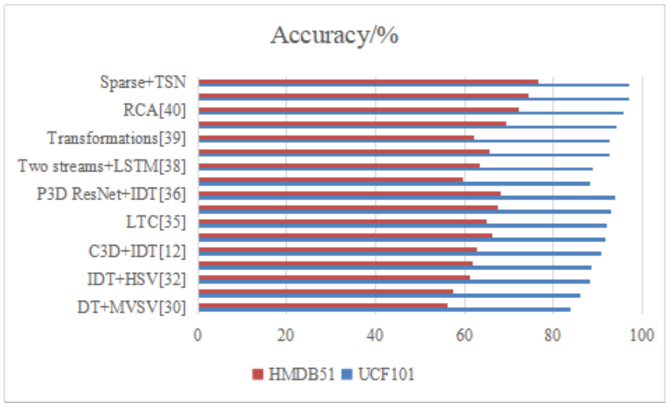


Fig. 10. Visual diagram of table 2

the recognition rate of the temporal segmentation network with sparse features are 97.1% and 76.6% on UCF101 and HMDB51 respectively.

4.4. Multi-layer feature fusion experiment

In order to verify the effectiveness of multi-layer convolution feature fusion convolution network, the experiment of UCF101 group 1 is taken as an example. The outputs from inception3a to inception5a are fused with the feature of inception5b. Table 3 lists the recognition rate of the temporal segmentation network trained by RGB, optical flow images and sparse images after adding the multi-layer feature fusion method. Similar to RGB image, the temporal segmentation network trained by optical flow image and sparse image is also fused with the convolution feature output by Incepteion5a layer and incepteion5b layer, and the highest recognition rate is obtained, which reaches 93.68% and 86.22% respectively. The optical flow basically remains unchanged. The recognition rate of sparse network is more than 0.6% higher than that of network without shallow convolution feature fusion, indicating that the addition of shallow convolution feature can improve the network performance.

Table 3. Comparison recognition rate with different convolution layers fusion on UCF101/%

Fusion layer	RGB	Optical flow	Sparse
Inception3a → Inception5b	87.82	92.71	85.14
Inception3b → Inception5b	87.97	93.15	84.92
Inception3c → Inception5b	87.95	93.10	85.06
Inception4a → Inception5b	87.97	92.98	85.48
Inception4b → Inception5b	87.25	93.03	85.98
Inception4c → Inception5b	87.22	92.89	85.57
Inception4d → Inception5b	87.83	93.12	85.83
Inception4e → Inception5b	88.21	92.89	86.09
Inception5a → Inception5b	88.34	93.68	86.22

In order to further verify the effect of the multi-layer feature fusion on temporal segmentation network, the experiment is conducted on UCF101 and HMDB51 data sets. Compared with some classical algorithms and commonly used algorithms, the comparison results are shown in table 4.

As can be seen from table 4, the motion recognition temporal segmentation network with multi-layer feature fusion has a certain improvement compared with the original temporal segmentation network with sparse feature fusion. The recognition rates of UCF101 and HMIDB51 are 97.3% and 76.9%, indicating that shallow convolutional layer and deep convolutional layer fusion have a certain effect on the improvement of network performance. The confusion matrix is shown in figure 11 and figure 12.

Table 4. Accuracy comparison with different methods on UCF101 and HMDB51/%

Method	UCF101	HMDB51
C3D+IDT	90.6	62.8
TDD+IDT	91.7	66.1
LTC	91.9	65.0
LTC+IDT	92.8	67.4
P3D ResNet+IDT	93.9	68.2
Two streams	88.2	59.6
Two streams+LSTM	88.8	63.4
Two streams fusion	92.7	65.6
Transformations	92.6	62.2
TSN(RGB+Optical Flow)	94.2	69.4
Sparse+TSN	97.3	76.9

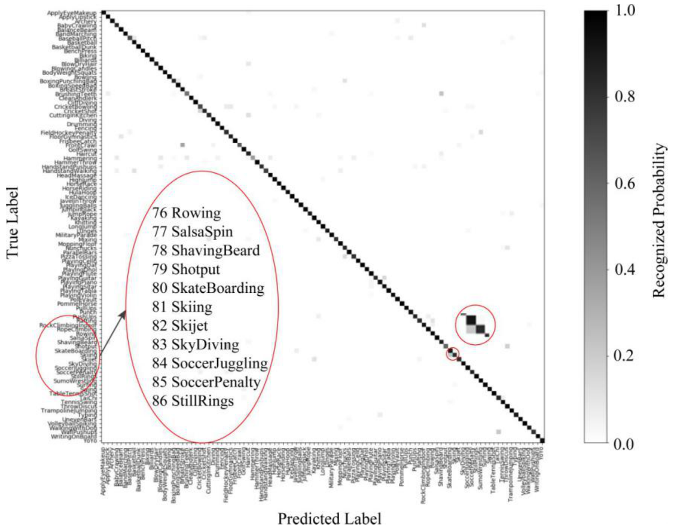


Fig. 11. Confusion matrix on UCF101 dataset

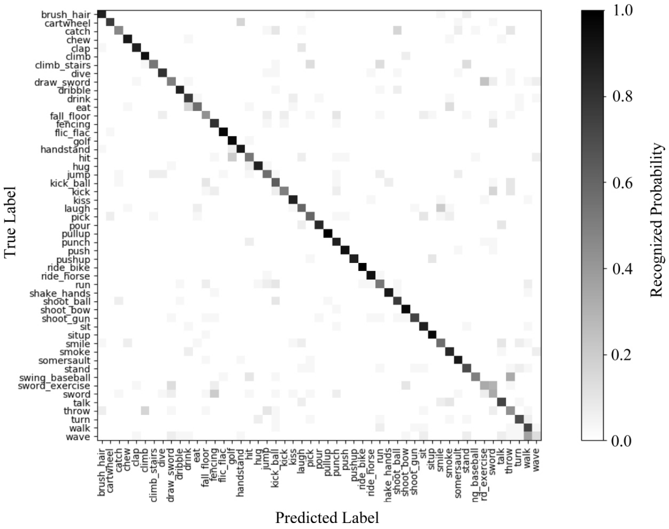


Fig. 12. Confusion matrix on HMDB51 dataset

5. Conclusion

In this paper, the two-stream convolutional neural network based on temporal segmentation network is studied and a temporal segmentation network with sparse features is proposed. Meanwhile, to solve the problem of low feature utilization, a multi-layer feature fusion temporal segmentation network for motion recognition is proposed. Based on the motion recognition network of sparse features and multi-layer feature fusion, the recognition effect of the new algorithm on the UCF101 and HMDB51 is better than other algorithms. In the future, more deep learning-based models will be utilized for action recognition.

References

1. Yao, G., Lei, T., Zhong, J. ”A Review of Convolutional-Neural-Network-Based Action Recognition,” *Pattern Recognition Letters,* vol. 118, pp. 14-22. (2018)
2. Li, H., Ding, Y., Li, C., et al,. ”Action recognition of temporal segment network based on feature fusion,” *Journal of Computer Research and Development*, Vol. 57, No. 1, pp. 145-158. (2020)
3. Olivieri, D. N., Conde, I.G., Sobrino, X.A.V. ”Eigenspace-based fall detection and activity recognition from motion templates and machine learning,” *Expert Systems with Applications,* Vol. 39, No. 5, pp. 5935-5945. (2012)
4. Zheng, D., Li, H., Yin, S. ”Action Recognition Based on the Modified Two-stream CNN,” *International Journal of Mathematical Sciences and Computing (IJMSC),* Vol. 6, No. 6, pp. 15-23. (2020)
5. J. Long, X. Wang, W. Zhou, J. Zhang, D. Dai and G. Zhu. ”A Comprehensive Review of Signal Processing and Machine Learning Technologies for UHF PD Detection and Diagnosis (I): Preprocessing and Localization Approaches,” *IEEE Access*, vol. 9, pp. 69876-69904, (2021).

6. Wang, P., Li, W., Ogunbona, P., et al. ”RGB-D-based Human Motion Recognition with Deep Learning: A Survey,” *Computer vision and image understanding,* Vol. 171, pp. 118-139. (2017)
7. Kim, K., Yong, K.C. ”Effective inertial sensor quantity and locations on a body for deep learning-based worker’s motion recognition,” *Automation in Construction,* Vol. 113. (2020)
8. Yin, S., Li, H. ”GSAPSO-MQC:medical image encryption based on genetic simulated annealing particle swarm optimization and modified quantum chaos system,” *Evolutionary Intelligence,* vol. 14, pp. 1817-1829. (2021)
9. Ji, S., Xu, W., Yang, M., and Yu, K. ”3D Convolutional Neural Networks for Human Action Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 35, No. 1, pp. 221-231. (2013)
10. Feichtenhofer, C., Pinz, A., Zisserman, A. ”Convolutional Two-Stream Network Fusion for Video Action Recognition,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2016, pp. 1933-1941.
11. Wang, H., Schmid, C. ”Action Recognition with Improved Trajectories,” *2013 IEEE International Conference on Computer Vision, 2013*, pp. 3551-3558.
12. Tran, D., Bourdev, L., Fergus, R., et al. ”Learning Spatiotemporal Features with 3D Convolutional Networks,” *2015 IEEE International Conference on Computer Vision (ICCV),* 2015, pp. 4489-4497.
13. Zhu, W., Hu, J., Sun, G., Cao X., et al. ”A Key Volume Mining Deep Framework for Action Recognition,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2016, pp. 1991-1999.
14. Kar, A. Rai, N. Sikka K. and Sharma, G. ”AdaScan: Adaptive Scan Pooling in Deep Convolutional Neural Networks for Human Action Recognition in Videos,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017,* pp. 5699-5708.
15. Yi, Z., Lan, Z., Newsam, S., et al. Hidden Two-Stream Convolutional Networks for Action Recognition. 2017. arXiv:1704.00389
16. Sevilla-Lara, L., Liao, Y., Güney, F., et al. ”On the Integration of Optical Flow and Action Recognition,” *Pattern Recognition. GCPR 2018. Lecture Notes in Computer Science,* vol. 11269, pp. 281-297, Springer, Cham. (2019)
17. Zhang, B., Wang, L., Wang, Z., et al. ”Real-Time Action Recognition With Deeply Transferred Motion Vector CNNs,” *IEEE Transactions on Image Processing,* Vol. 27, No. 5, pp. 2326-2339. (2018)
18. Choutas, V., Weinzaepfel, P., Revaud J. ”PoTion: Pose MoTion Representation for Action Recognition,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 2018, pp. 7024-7033.
19. Wang, L., et al. ”Temporal Segment Networks: Towards Good Practices for Deep Action Recognition,” *Computer Vision-ECCV 2016. ECCV 2016. Lecture Notes in Computer Science,* vol. 9912, pp. 20-36, Springer, Cham. (2016)
20. Lan, Z., Zhu, Y., Hauptmann, A. G., and Newsam, S. ”Deep Local Video Feature for Action Recognition,” *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),* pp. 1219-1225. (2017)
21. Zhou, B., Andonian, A., Oliva, A., et al. ”Temporal Relational Reasoning in Videos,” *Computer Vision-ECCV 2018. ECCV 2018. Lecture Notes in Computer Science,* vol. 11205, pp. 831-846, Springer, Cham. (2018)
22. Xu, H., Das, A., and Saenko, K. ”R-C3D: Region Convolutional 3D Network for Temporal Activity Detection,” *2017 IEEE International Conference on Computer Vision (ICCV),*, pp. 5794-5803. (2017)
23. Yin, S., Li, H., Teng, L. ”Airport Detection Based on Improved Faster RCNN in Large Scale Remote Sensing Images,” *Sensing and Imaging,*?Vol. 21. (2020).
24. Chen, J., Kong, J., Sun, H. et al. ”Spatiotemporal Interaction Residual Networks with Pseudo3D for Video Action Recognition,” *Sensors,* Vol. 20, No. 11, 3126. (2020)

25. Jiang, D., Li, H., Yin, S. ”Speech Emotion Recognition Method Based on Improved Long Short-term Memory Networks,” *International Journal of Electronics and Information Engineering,* Vol. 12, No. 4, pp. 147-154. (2020)
26. Jiang, Y., Wu, Z., Tang, J., et al. ”Modeling Multimodal Clues in a Hybrid Deep Learning Framework for Video Classification,” *IEEE Transactions on Multimedia,* vol. 20, no. 11, pp. 3137-3147. (2018)
27. Du, W., Wang, Y., Qiao, Y. ”RPAN: An End-to-End Recurrent Pose-Attention Network for Action Recognition in Videos,” *2017 IEEE International Conference on Computer Vision (ICCV),* 2017, pp. 3745-3754.
28. Duan, Z., Zhang, T., Tan, J. et al. ”Non-Local Multi-Focus Image Fusion With Recurrent Neural Networks,” *IEEE Access,* Vol. 8, pp. 135284-135295. (2020)
29. Byeon, Y.H., Kwak, K.C. ”Facial Expression Recognition Using 3D Convolutional Neural Network,” *International Journal of Advanced Computer Science & Applications,* Vol. 5, No. 12. (2014).
30. Cai, Z., Wang, L., Peng, X., Qiao, Y. ”Multi-view Super Vector for Action Recognition,” *2014 IEEE Conference on Computer Vision and Pattern Recognition,* 2014, pp. 596-603.
31. Luong, V. D., Wang, L., Xiao, G. ”Action Recognition Using Hierarchical Independent Subspace Analysis with Trajectory,” *Springer International Publishing,* 2015.
32. Peng, X., Wang, L., Wang, X., et al. ”Bag of Visual Words and Fusion Methods for Action Recognition: Comprehensive Study and Good Practice,” *Computer Vision & Image Understanding,* Vol. 150, pp. 109-125. (2016)
33. Wang, L., Qiao, Y., Tang, X. ”MoFAP: A Multi-level Representation for Action Recognition,” *International Journal of Computer Vision,* Vol. 119, No. 3, pp. 254-271. (2016)
34. Wang, L., Qiao, Y., Tang, X. ”Action recognition with trajectory-pooled deep-convolutional descriptors,” *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2015, pp. 4305-4314.
35. Varol, G., Laptev, I., Schmid, C. ”Long-Term Temporal Convolutions for Action Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 40, No. 6, pp. 1510-1517. (2018)
36. Qiu, Z., Yao, T., Mei, T. ”Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks,” *2017 IEEE International Conference on Computer Vision (ICCV). IEEE,* 2017.
37. Simonyan, K., Zisserman, A. ”Two-stream convolutional networks for action recognition in videos,” *Neural Information Processing Systems,* Vol. 1, No. 4, 568576. (2014)
38. Joe Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga and G. Toderici. ”Beyond short snippets: Deep networks for video classification,” *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2015, pp. 4694-4702.
39. Wang, X., Farhadi A., and Gupta, A. ”Actions Transformations,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2016, pp. 2658-2667.
40. Dianhuai Shen, Xueying Jiang, Lin Teng. ”Residual network based on convolution attention model and feature fusion for dance motion recognition,” *EAI Endorsed Transactions on Scalable Information Systems,* 21(33), e8, 2021. http://dx.doi.org/10.4108/eai.6-10-2021.171247
41. Jisi A and Shoulin Yin. ”A New Feature Fusion Network for Student Behavior Recognition in Education,” *Journal of Applied Science and Engineering,* vol. 24, no. 2, pp. 133-140. (2021)

Chen Chen is a lecturer and doctor at the School of Physical Education of Henan University of Technology. Research direction: Social sports, humanities and Sociology of sports.

Received: May 01, 2022; Accepted: September 08, 2022.