Complex & Intelligent Systems (2023) 9:1377–1390
https://doi.org/10.1007/s40747-022-00858-8

ORIGINAL ARTICLE

Parallel temporal feature selection based on improved attention mechanism for dynamic gesture recognition

Gongzheng Chen[1] · **Zhenghong Dong**[1] · **Jue Wang**[1] · **Lurui Xia**[1]

Received: 1 May 2022 / Accepted: 16 August 2022 / Published online: 7 September 2022
© The Author(s) 2022

Abstract

Dynamic gesture recognition has become a new type of interaction to meet the needs of daily interaction. It is the most natural, easy to operate, and intuitive, so it has a wide range of applications. The accuracy of gesture recognition depends on the ability to accurately learn the short-term and long-term spatiotemporal features of gestures. Our work is different from improving the performance of a single type of network with convnets-based models and recurrent neural network-based models or serial stacking of two heterogeneous networks, we proposed a fusion architecture that can simultaneously learn short-term and long-term spatiotemporal features of gestures, which combined convnets-based models and recurrent neural network-based models in parallel. At each stage of feature learning, the short-term and long-term spatiotemporal features of gestures are captured simultaneously, and the contribution of two heterogeneous networks to the classification results in spatial and channel axes that can be learned automatically by using the attention mechanism. The sequence and pooling operation of the channel attention module and spatial attention module are compared through experiments. And the proportion of short-term and long-term features of gestures on channel and spatial axes in each stage of feature learning is quantitatively analyzed, and the final model is determined according to the experimental results. The module can be used for end-to-end learning and the proposed method was validated on the EgoGesture, SKIG, and IsoGD datasets and got very competitive performance.

Keywords Dynamic gesture recognition · Attention mechanism · Spatiotemporal features · The human–computer interaction · Video understanding

Introduction

With the popularization of computers in modern society, the interaction technology of people is gradually transferred from the computer as the center to the human center, and the technology of the cross-domain man–machine barrier has become a new research hotspot. Gestures are the most instinctive and common means of human communication. Compared with expressions and actions, gestures are not only more intuitive and natural but also can express rich semantic information. Therefore, gestures are the most common means of human communication. However, the gesture itself is highly flexible and diverse, so interaction through gesture is a challenging research direction.

✉ Gongzheng Chen
chengongzheng64@163.com

[1] Graduate School of Aerospace Engineering University, Beijing, China

Gestures are divided into static gestures and dynamic gestures according to the semantic expression of the gesture. The static gesture takes pictures as its data set. It focuses on gesture posture and shape features at a single time. The dynamic gesture takes video as its data set. It not only focuses on gesture posture and shape but also the time series of gesture input. The purpose of studying human–computer interaction is to make human–computer interaction as natural as human–human interaction. However, static pictures have limited semantic expression, so dynamic gesture interaction is more in line with people's usage habits and more suitable for the application of future human–computer interaction technology.

Early gesture recognition mainly relied on wearable devices and artificial design features, such as Soli [1] and MYO [2]. However, wearable devices are bulky and cumbersome to wear, so there are certain limitations. Artificial design features include Parcheta et al.'s [3] hidden Markov model (HMM), HOG algorithm, MEI algorithm,

etc. However, artificial design features cannot meet the needs of the actual gesture recognition system because it requires artificial design features and is time-consuming. With the advent of Alexnet and other convolutional neural networks, deep learning has made breakthroughs in image classification, image segmentation, target detection, scene recognition, face recognition [4], action recognition [5, 6], and gesture recognition. The method of dynamic gesture feature extraction based on deep learning has become a research hotspot. Gesture recognition is mainly based on the skeleton or video sequence, so time information plays an important role in gesture recognition. Unfortunately, compared with motion recognition and other video classification tasks, video background can improve the recognition results. For gesture recognition, complicated video background can reduce the accuracy of gesture recognition, bringing more challenges to gesture-based human–computer interaction. Because gesture recognition pays more attention to the gesture itself, namely the shape and movement track of the hand, a larger and complex background will affect the spatial features of the smaller hand and arm. Therefore, time information is more important in gesture recognition than other video classification tasks.

Many video classification algorithms have been applied to dynamic gesture recognition and achieved good performance and high accuracy. The first method is the two-flow convolutional network, which is the spatiotemporal features of gestures that people try to learn by using CNN. The second method is to inflate the 2D filters to 3D filters $T \times H \times W$. 3D ConvNet [7] (C3D) can directly learn spatiotemporal features by adding time dimension T, so it can directly process multiple frames and has made a breakthrough in video classification tasks. However, the convolution kernel is limited in size, so the learned feature receptive field is limited. In the same stage of the model, the mapping relation with the difference between frames greater than the time dimension of the convolution kernel cannot be learned. Therefore, such models can only learn short-term spatiotemporal features. The last one is to mimic natural language processing such as recursive neural networks and long and short-term memory to learn temporal features of gestures. This structure can remember the information learned before the network and apply it to the current calculation, so it can learn the information for a long time.

Since the convolution operation essentially fuses the spatial and channel information of the input feature maps, semantic information is captured from the input feature graph. However, not every feature graph has the same features, and its contribution to semantics is also different. Therefore, the attention mechanism emerges and is widely used in various computer vision tasks. Examples are SE [8], BAM [9] and CBAM [10].

For gesture recognition, the time series of hand and arm movements are very important. The learning of spatiotemporal features directly affects the recognition results. Our work is different from improving the performance of a single type of network with convnets-based models and recurrent neural network-based models or serial stacking of two heterogeneous networks. Instead, we propose an architecture that uses the attention mechanism to connect two heterogeneous networks in parallel in two independent dimensions, channel and spatial. And then the attention maps are multiplied to the input feature map for adaptive feature refinement. The module cannot only apply the channel attention module (CAM) and spatial attention module (SAM) to learn both global and local spatiotemporal features in parallel but also make the two heterogeneous networks learn the noteworthy information on the channel axis and spatial axis at each stage and automatically learn the contribution of two heterogeneous networks to classification results through backpropagation. It can discard some useless information and improve the weight of useful information. Res2plus1D and ConvLSTM are mainly used in this paper. More importantly, two heterogeneous networks can be replaced by other networks. Similar to the SeST [11], the proposed network not only selectively extracts the weight and dependencies for the two types of networks in the channel axis but also selectively extracts the weight and dependencies for the spatial axis.

The contributions of this paper are as follows:

1. Two heterogeneous networks (R2plus1D and ConvLSTM) are combined in parallel called the RPCNet module. The structure cannot only learn long-term and short-term spatiotemporal features by using the characteristics of the two networks but also the network is an end-to-end model.
2. The proposed RPCNet utilizes the attention mechanism to extract the features that need attention in the channel and the spatial axes and adjusts the contributions of the two networks by Softmax operation at each stage. By comparing the sequential and pooling operations of CAM and SAM and quantitatively analyzing the proportions of short-term and long-term spatiotemporal features on channel and spatial axes in each stage of feature learning, proves the effectiveness of our model.
3. In EgoGesture [12], SKIG [13], and IsoGD [14], the network gets very competitive performance in end-to-end networks.

The rest of this paper is organized as follows. In "Related work", we review the related work of gesture recognition from the aspects of manual feature extraction and deep learning. And in "Proposed method", we introduce the model architecture proposed in this paper in detail. In "Experiment", we discuss the combination of CAM and SAM and

the influence of maximum pooling and average pooling on the model accuracy, and we compared the proposed on Ego-Gesture [12] dataset, SKIG [13] dataset, and IsoGD [14] dataset. Finally, the output results of model weight are analyzed. The last section is for discussion and outlook.

Related work

With the continuous popularization of computers in society, the development of human–computer interaction technology based on gesture recognition will bring convenience to the use of computers, and can greatly improve the efficiency of interaction. Many kinds of research are based on this background. This section reviews the current researches from two aspects: hand-crafted features and gesture feature extraction based on deep learning.

Hand-crafted feature of gesture recognition

Due to the lack of computer computing power, researchers can only use artificial features for image recognition. And, most artificial features are mainly used to extract spatiotemporal features for gesture recognition, that is, the interesting part of the input image is converted into a set of feature vectors. Common gesture features are divided into global feature-based and local feature-based. Common global features include image color, shape, texture, and other features, which are easy to understand and require less computation. Ibrahim et al. [15] used tone-based skin detection to segment the hand region and skin spot tracking technology to recognize and track gestures. But this kind of feature description does not apply to the case of occluded images. In recent years, gesture feature extraction based on local features has been widely used. Yang et al. [16] used depth motion mapping (DMMs) to capture motion cues from three different perspectives of the front, side, and top, and finally used the HOG descriptor to represent actions. Reviewing previous related work, the IDT proposed by Wang [17] is the most successful method of hand-crafted. Due to the high cost of hand-crafted, it is difficult to apply it to real scenes because it cannot take into account factors such as video background. With the rapid development of deep learning, many methods have been applied to gesture recognition, and gradually replace artificial features.

Gesture feature extraction based on deep learning

In recent years, deep neural networks have been applied to computer vision and other fields. The two-flow network was the first attempt to learn spatiotemporal features, and then some improvements were made according to the two-flow network, such as Wang et al. [18]. Wu et al. [19]. The above network uses optical flow as the input of time flow, which requires a lot of complex calculations and thus reduces the speed of the network model. Therefore, the C3D network emerges and is gradually applied in gesture recognition [20–23]. Based on the great success of the ResNet network in graph classification, Miao proposed the ResC3D [21] model, which makes use of the advantages of residual and C3D model and gets a place in ChaLearn LAP [24] gesture recognition. But 3D convolution can only handle small temporal windows, not the whole gesture in the video. Therefore, this kind of network cannot capture the long-term spatiotemporal features of gesture, thus affecting the performance of the network in the gesture recognition task. LSTM is a variant of the recurrent neural network, which can encode videos of different lengths for a long time. Since LSTM was proposed in 1995, several LSTM variants have been proposed by researchers. Among them, Shi et al. [25] extended fully connected LSTM (FC-LSTM) by using convolution structure in input to state and state to state transition and proposed convolution LSTM (ConvLSTM) network to process sequential images of precipitation near prediction. Therefore, convolutional LSTM can be used in video tasks such as motion recognition and gesture recognition, and it solves the defect that the model based on a convolutional neural network lacks time modeling ability. Molchanov et al. [26] combined C3D and RNN to construct a cyclic 3D convolutional neural network for gesture recognition. Nunez et al. [27] proposed the network model constructed by CNN and LSTM, but the model ignored the spatial information, thus reducing the accuracy. To make up for this deficiency, Zhang et al. [28] introduced ConvLSTM to replace RNN or LSTM to learn the spatial features of videos. Elboushaki et al. [29], Peng et al. [30], and Wang et al. [31] are conducting video classification tasks through the combination of 3D convolution and ConvLSTM. All of the above methods are serial cascades of two heterogeneous networks, and they are not simultaneous learning.

The attention mechanism is widely used in gesture recognition because it enables neural networks to recognize key information. Hou [32] and Wiederer [33], for example, have integrated the attention mechanism into gesture recognition. Dhingra [34] proposes a 3D attention based on the residual network, which can generate multiple stacked attention blocks for traffic gestures. Zhang [35] uses the attention module in different gates in LSTM and combines it with 3DCNN for gesture recognition.

Proposed method

Although convolutional network can realize end-to-end learning, its convolution kernel is limited. For example, the convolution kernel of C3D is $T \times H \times W$, which are time,

height, and width, respectively. Therefore, the receptive field of feature extraction is fixed. In the convolution of the same layer, features can only be extracted from adjacent T frames, and feature relations with frame spacing greater than T cannot be extracted. Therefore, 3D convolution cannot encode long-term time information. A large number of previous studies focused on sequential networks such as CNN and RNN, but none of these networks could learn spatiotemporal features at the same time, and the information loss caused by a single network would continue to accumulate. The proposed RPCNet module uses the attention mechanism to learn the contribution and dependence of R2plus1D and ConvLSTM to classification results on the channel and spatial axes respectively, so that the module can learn the long-term and short-term spatiotemporal features in parallel, thus overcoming the problem of a single network, and the model can realize end-to-end learning. Greatly improve the accuracy in video tasks. Its structure is shown in Fig. 1. Firstly, the gesture sequence is input into a 3D convolutional network (Layer0) for feature extraction and dimensionality reduction, which is indispensable. Because ConvLSTM will generate more training parameters and increase training time if the gesture sequence is input directly into the RPCNet module. Then it is input to the 4-layer network in turn, among which the first three layers (Layer1, Layer2, and Layer3) are the RPCNet module proposed in this paper, and the last layer (Layer4) is the 3D convolutional network.

The section begins with a review of the R2plus1D and ConvLSTM networks, it is our main framework for the proposed method. R2plus1D network is an outstanding 3D network that can learn the short-term spatiotemporal features of videos. And ConvLSTM can encode information over a long period. By combining the two architectures in parallel, the long-term and short-term spatiotemporal features are fully utilized for dynamic gesture recognition. Table 1 shows the hyperparameters used in various components of our work.

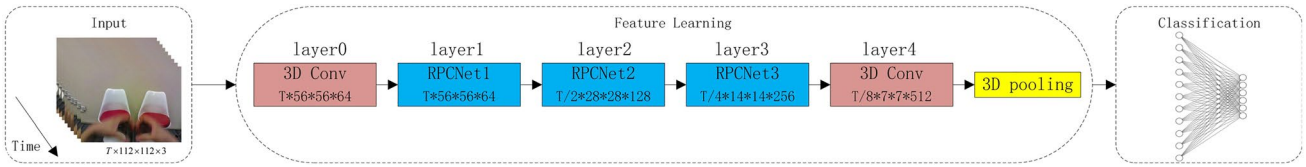


Fig. 1 The framework of the proposed deep architecture

Table 1 The hyperparameters used in various components of our work

Layer name	Output size ($T \times H \times W$)	Convnets-based model	Recurrent neural network-based model
Layer0	32×56×56	$\begin{bmatrix} 1\times7\times7, \text{stride}(1,2,2), 64 \\ 3\times1\times1, \text{stride}(1,1,1), 64 \end{bmatrix}$	None
Layer1	32×56×56	$\begin{bmatrix} 1\times3\times3, \text{stride}(1,1,1), 64 \\ 3\times1\times1, \text{stride}(1,1,1), 64 \end{bmatrix} \times 4$	$\begin{bmatrix} 3\times3, \text{stride}(1,1), 64 \\ FC-192 \end{bmatrix} \times 2$
Layer2	16×28×28	$\begin{bmatrix} 1\times3\times3, \text{stride}(1,2,2), 128 \\ 3\times1\times1, \text{stride}(2,1,1), 128 \end{bmatrix} \times 1$	$\begin{bmatrix} 3\times3, \text{stride}(1,1), 64 \\ FC-192 \end{bmatrix} \times 2$
		$\begin{bmatrix} 1\times3\times3, \text{stride}(1,1,1), 128 \\ 3\times1\times1, \text{stride}(1,1,1), 128 \end{bmatrix} \times 3$	$3\times7\times7, \text{stride}(1,2,2), 128$
Layer3	8×14×14	$\begin{bmatrix} 1\times3\times3, \text{stride}(1,2,2), 256 \\ 3\times1\times1, \text{stride}(2,1,1), 256 \end{bmatrix} \times 1$	$\begin{bmatrix} 3\times3, \text{stride}(1,1), 128 \\ FC-384 \end{bmatrix} \times 2$
		$\begin{bmatrix} 1\times3\times3, \text{stride}(1,1,1), 256 \\ 3\times1\times1, \text{stride}(1,1,1), 256 \end{bmatrix} \times 3$	$3\times7\times7, \text{stride}(1,2,2), 256$
Layer4	4×7×7	$\begin{bmatrix} 1\times3\times3, \text{stride}(1,2,2), 256 \\ 3\times1\times1, \text{stride}(2,1,1), 256 \end{bmatrix} \times 1$	None
		$\begin{bmatrix} 1\times3\times3, \text{stride}(1,1,1), 512 \\ 3\times1\times1, \text{stride}(1,1,1), 512 \end{bmatrix} \times 3$	
	1×1×1	Average pool, FC	

R2plus1D component

The earliest spatiotemporal features learning component is C3D extended from C2D, which expands the 2D convolution kernel $H \times W$ into 3D convolution kernel $T \times H \times W$ by introducing an additional time dimension T. Inspired by the ResNet network, ResC3D applies residual network to 3D convolution, which reduces the parameters of the network, avoids the problem of gradient disappearance, and greatly improves the performance of the model. In this paper, the RPCNet module is based on the R2plus1D model, which splits the convolution kernel of $3 \times 3 \times 3$ in the ResC3D model into $1 \times 3 \times 3$ and $3 \times 1 \times 1$ for spatial feature extraction respectively, and sequential feature extraction. The structure is shown in Fig. 2. We use the R2plus1D-18 in our work.

ConvLSTM component

LSTM is mainly used in natural language processing to learn the timing features of text vectors, and it mainly deals with one-dimensional vectors. However, both video and image are two-dimensional vectors, which cannot be encoded for a long time. ConvLSTM was proposed to change the feedforward method of LSTM from Hadamard product to convolution. ConvLSTM has a large number of parameters because of the convolution operation. GateConvLSTM proposed in the literature [36] reduces the spatial dimension by performing a global average pooling on input features and hidden states, so that the convolution operation can be replaced by the fully connected operation. The number of parameters and calculation costs are greatly reduced. The performance of GateConvLSTM is better than that of ConvLSTM. The GateConvLSTM structure is shown in Fig. 3, the GateConvLSTM can be formulated as:

$$\overline{X}_t = \text{GlobalAveragePooling}(X_t), \tag{1}$$

$$\overline{H}_{t-1} = \text{GlobalAveragePooling}(H_{t-1}), \tag{2}$$

$$i_t = \sigma\left(W_{xi}\overline{X}_t + W_{hi}\overline{H}_{t-1} + b_i\right), \tag{3}$$

$$f_t = \sigma(W_{xf}\overline{X}_t + W_{hf}\overline{H}_{t-1} + b_f), \tag{4}$$

$$o_t = \sigma(W_{xo}\overline{X}_t + W_{ho}\overline{H}_{t-1} + b_o), \tag{5}$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c), \tag{6}$$

$$H_t = o_t \circ \tanh(C_t), \tag{7}$$

where σ is the sigmoid function, X_t is the input, C_t is the cell state, H_t is the hidden state. where $W_{x\sim}$ and $W_{h\sim}$ are 2D convolution kernels. i_t,f_t,o_t are three-dimensional tensors, * represents convolution operator, and ∘ represents Hadamard product.

However, the input and output dimensions of GateConvLSTM operation are unchanged, so we adopt the structure in Fig. 4 to enable GateConvLSTM to reduce sampling and achieve the purpose of end-to-end training. The input of this structure is $T \times H \times W \times C$, and compared with the information before $T/2$, the information after $T/2$ has more features to learn. So we sent the last $T/2$ information to the C3D structure to learn spatial features.

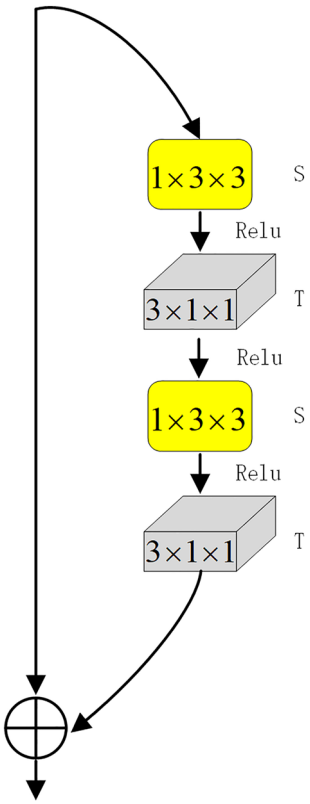
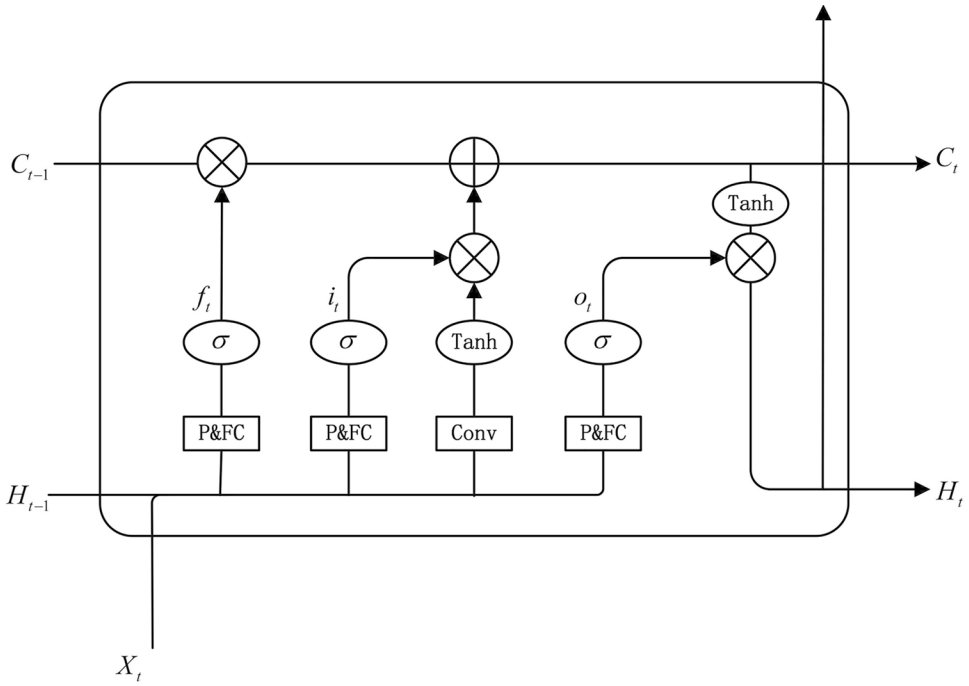


Fig. 2 The R2plus1D component

RPCNet

To better classify gestures, we need to learn the long-term and short-term spatiotemporal features of gestures, and the lack of any information will greatly affect the classification results. The representative long-term and short-term spatiotemporal features models are ConvLSTM and 3DCNN. 3DCNN is limited by the size of the convolution kernel, and the size of the receptive field is limited, so it can only learn short-term spatiotemporal features. Although ConvLSTM adopts a memory gate structure to learn long-term

Fig. 3 GateConvLSTM



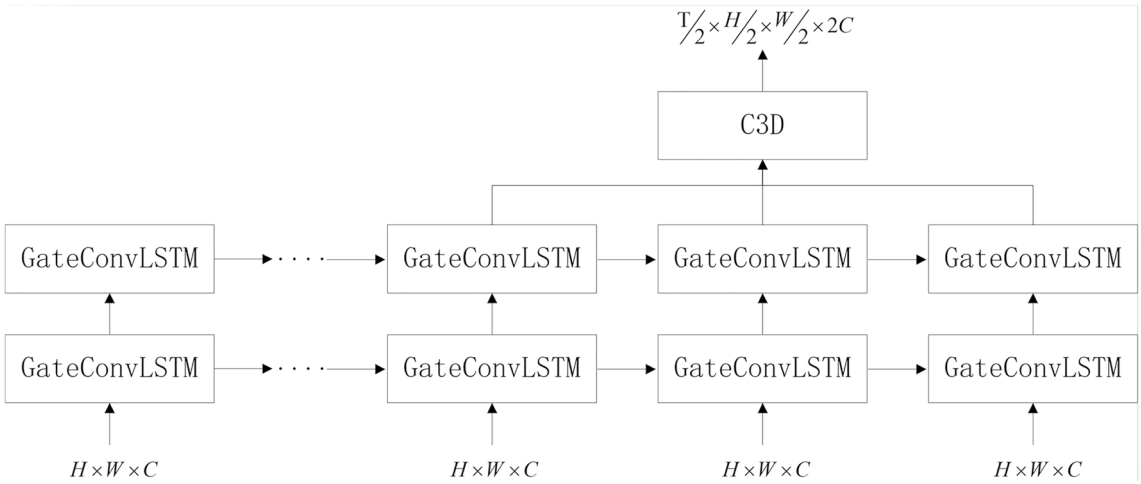


Fig. 4 Improved GateConvLSTM

spatiotemporal features, with the increase of input sequence, the earlier time information will be forgotten. However, the serial connection of 3D CNN and ConvLSTM could not learn both short-term and long-term spatiotemporal features at the same stage of learning. Therefore, the short-term and long-term spatiotemporal features are simultaneously learned by connecting two heterogeneous networks in parallel. In the process of network fusion, an attention mechanism is used to extract the features that need attention from two heterogeneous networks in the channel and spatial axes respectively, and the contribution and dependencies of two heterogeneous networks in the process of model learning are automatically selected. As shown in Fig. 5, the model is divided into two parts, the channel fusion module, and the spatial fusion module respectively. These are shown in Figs. 6 and 7.

Channel fusion module

Firstly, the input frames are sent to R2plus1D and GateConvLSTM networks respectively, and the outputs of the two models are summed element by element to integrate the feature maps of the two heterogeneous models. We downsize features maps from $T \times H \times W \times C$ to $T/2 \times H/2 \times W/2 \times 2C$.

Secondly, the global average pooling operation and the global maximum operation are used to aggregate the

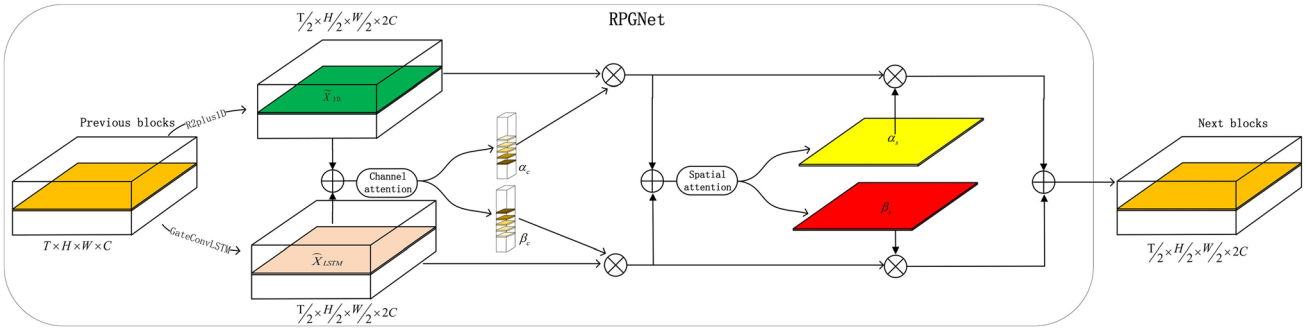


Fig. 5 RPCNet

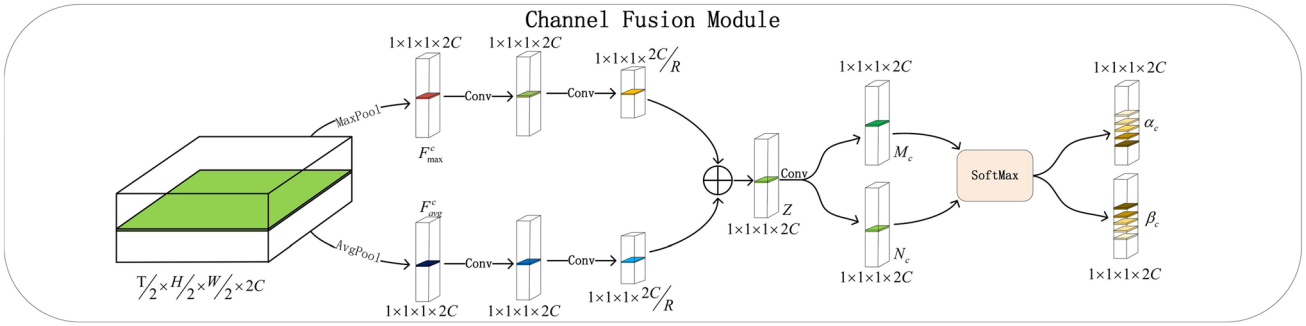


Fig. 6 Channel fusion module

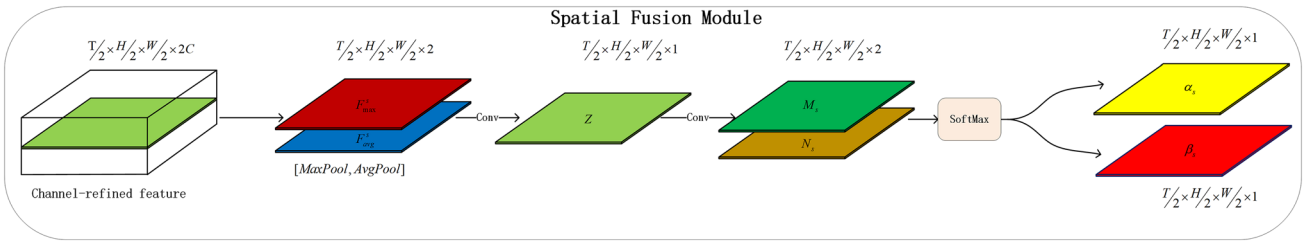


Fig. 7 Spatial fusion module

spatiotemporal features of X and generate channel-wise statistics F^c_{avg} and F^c_{max} whose dimension is $1 \times 1 \times 1 \times 2C$ used to represent the global average-pooled features and global max-pooled features. The resulting global average-pooled features are forwarded into a $1 \times 1 \times 1$ convolution to capture contextual information between channel axis. And then forwarded it into a $1 \times 1 \times 1$ convolution and we downsize features maps from $1 \times 1 \times 1 \times 2C$ to $1 \times 1 \times 1 \times {}^{2C}/_{r}$, reducing the dimensionality by reducing ratio r and capturing the channel-wise dependencies completely. F^c_{max} is the same as F^c_{avg}. And then, add up the features and get Z, and it is forwarded into a $1 \times 1 \times 1$ convolution for Softmax operation. In the experiment, the r was selected as 16 following reference [10].

Finally, a Softmax operation can automatically select the weight α_c and β_c of the two branches on the channel axis. The process can be formulated as:

$$X = \text{R2plus1}(I) + \text{GateConvLSTM}(I) = \tilde{X}_{\text{3D}} + \widehat{X}_{\text{LSTM}}, \quad (8)$$

$$z = \mathrm{Conv}(\mathrm{Conv}(\mathrm{Avgpool}(X))) + \mathrm{Conv}(\mathrm{Conv}(\mathrm{Maxpool}(X))) = \delta(W_2(W_0(F^c_{\mathrm{avg}}))) + \delta(W_3(W_1(F^c_{\mathrm{max}}))), \tag{9}$$

$$M_c = \mathrm{Conv}(z) = W_4(z), \tag{10}$$

$$N_c = \mathrm{Conv}(z) = W_5(z), \tag{11}$$

$$\alpha_c = \frac{\exp(M_c)}{\exp(M_c) + \exp(N_c)} \tag{12}$$

$$\beta_c = \frac{\exp(N_c)}{\exp(M_c) + \exp(N_c)}, \tag{13}$$

$$Y_c = \alpha_c \bullet \tilde{X}_{3D} + \beta_c \bullet \widehat{X}_{LSTM}, \alpha_c + \beta_c = 1, \tag{14}$$

where I represents the input feature whose size is $T \times H \times W \times C$,$T, H, W, C$ represents the time length, height, width, and the number of input channels of the feature, respectively. W_0 and W_1 is the convolution kernel of $1 \times 1 \times 1$, $W_2 \in R^{\frac{2C}{r} \times 2C}$, $W_3 \in R^{\frac{2C}{r} \times 2C}$, $W_4 \in R^{2C \times \frac{2C}{r}}$, $W_5 \in R^{2C \times \frac{2C}{r}}$.$Y_c \in R^{\frac{T}{2} \times \frac{H}{2} \times \frac{W}{2} \times 2C}$,$\delta$ denotes the ReLU function.

Spatial fusion module

Firstly, X' generated by channel fusion module is used to perform global average-pooled and max-pooled features along the channel dimension to highlight the effective information region, and spatial descriptor F^s_{avg} and F^s_{max} with dimension $T/2 \times H/2 \times W/2 \times 1$ is generated to represent the average-pooled feature and max-pooled feature. The resulting average-pooled feature and max-pooled are then forwarded into a $3 \times k \times k$ convolution to capture contextual information between spatial axis. And then, it forwarded into a $3 \times k \times k$ convolution for Softmax operations. According to the experimental results in reference [10], the model k value was set to 7.

Finally, a Softmax operation can automatically select the weight α_s and β_s of the two branches on the spatial axis. The process can be formulated as:

$$X' = Y_c = \alpha_c \bullet \tilde{X}_{3D} + \beta_c \bullet \widehat{X}_{\mathrm{LSTM}}, \tag{15}$$

$$z = \mathrm{Conv}([\mathrm{Avgpool}(X'); \mathrm{MaxPool}(X'))) = \delta(f_1^{3\times7\times7}([F^s_{\mathrm{avg}}; F^s_{\mathrm{max}}])), \tag{16}$$

$$M_s = \mathrm{Conv}(z) = f_2^{3\times7\times7}(z), \tag{17}$$

$$N_s = \mathrm{Conv}(z) = f_3^{3\times7\times7}(z), \tag{18}$$

$$\alpha_s = \frac{\exp(M_s)}{\exp(M_s) + \exp(N_s)} \tag{19}$$

$$\beta_s = \frac{\exp(N_s)}{\exp(M_s) + \exp(N_s)}, \tag{20}$$

$$Y_s = \alpha_s \bullet \alpha_c \bullet \tilde{X}_{3D} + \beta_s \bullet \beta_c \bullet \widehat{X}_{\mathrm{LSTM}}, \alpha_s + \beta_s = 1, \tag{21}$$

where $f^{3\times7\times7}$ represents a convolution operation with the filter size of $3 \times 7 \times 7$,δ represent the sigmoid function.

Experiment

To verify the effectiveness of the RPCNet, we evaluate our network on the EgoGesture [12], SKIG [13], and IsoGD [14] datasets. The model can be validated in RGB and Depth modalities.

EgoGesture [12] is a recent multimodal large-scale dataset, which was an egocentric gesture recognition dataset published by the Chinese Academy of Sciences, and its video format is a first-person view. The dataset format is RGB-D, and the resolution of each video frame is 320×240 which was collected by 50 people in 6 different indoor and outdoor scenarios. There are 83 gesture categories in this dataset, including 33 static gestures and 50 dynamic gestures. The dataset splits in a 3:1:1 ratio by distinct subjects which resulted in 1239 training, 411 validation, and 431 testing videos, having 14,416,4768, and 4977 gesture samples, respectively.

SKIG [13] dataset is one of the used datasets for hand gesture recognition published by Sheffield. The dataset format is RGB-D and contains 1080 gesture sequences. It collected 10 different gestures from 6 subjects, who were asked to complete gestures with fist, flat, and index in 2 illumination conditions and 3 different backgrounds.

IsoGD [14] dataset is a large gesture dataset containing 47,933 gesture videos and the format is RGB-D. It's derived from the CGD dataset which was collected by 21 different individuals and it has 249 categories.

Data sets and implementation details

For each video, the proposed RPCNet selected 32 frames as the input, and for the clips with more than 32 frames, we keep the middle 32 frames and remove the non-important information at both ends. We randomly repeat the frames until the number of video frames equals 32 for the clips with

less than 32 frames. When training, we randomly cut each frame to 224 × 224 to achieve the purpose of data expansion. And center clipping when testing. Then adjust the image to 112 × 112. One Quadro GV100 is used to train our proposed, which is implemented on the Pytorch platform. We use the mini-batch stochastic gradient descent (SGD) algorithm to optimize the parameters of the network. The momentum is set as 0.9 and weight decay is set at 0.0005. The batch size is set to 8 and the initial learning rate follows a polynomial decay from 0.001 to 0.000001 within a total of 50 epochs. In this paper, to shorten the training time, We first pre-trained our model on the Jester dataset [37].

Explorative study

This section will prove the validity of the model through experiments. In this experiment, we used the EgoGesture [12] dataset. And R2plus1D and GateConvLSTM as the basic model. Firstly, RPCNet modules are used in the last four stages of the model, namely Layer1, Layer2, Layer3, and Layer4. The experiments are divided into three parts, which discuss the sequential of CAM and SAM and the influence of maximum pooling and average pooling on the model accuracy. The final model is determined by comparing the weights of the two heterogeneous networks on the channel axis and spatial axis.

Sequential of CAM and SAM

In this experiment, three different combinations of CAM and SAM are compared: sequential channel-spatial, sequential spatial-channel, and parallel use of both attention modules. Because CAM is more concerned with global information and SAM is concerned with local information, the two functions are different. Therefore, the combination mode will change the performance of the model.

Table 2 compares the experimental results of the three permutations. It can be found that the effect of using CAM first is better than using SAM first. Both CAM and SAM were better than CAM alone. SeST [11] only uses CAM to learn spatiotemporal features in parallel and ignores the importance of SAM.

Table 2 Comparison methods of the CAM and SAM on EgoGesture [12]

Description	Accuracy (%)	
	RGB	Depth
R3D + channel (SeST [11])	93.20	93.35
R2plus1D	92.76	92.94
R2plus1D + channel	93.26	93.42
R2plus1D + channel + spatial	93.53	93.68
R2plus1D + spatial + channel	93.32	93.45
R2plus1D + channel & spatial in parallel	93.42	93.51

Maximum pooling and average pooling

This experiment compares the influence of three model variations on model accuracy. Average pooling alone, maximum pooling alone, and both pooling operations are used in parallel. Because they capture different information, the two pooling approaches can complement each other.

Table 3 compares the three variants. You can see that average pooling alone is more accurate than maximum pooling alone. And the best results can be obtained by using both the average pooling and the maximum pooling. Therefore, it is recommended to use both average pooling and maximum pooling.

Comparing the weights of two heterogeneous networks in the channel and spatial axes

We selected the Photo Frame category of EgoGesture [12] validation to the trained network and output the weight of each layer's channel axis. Figure 8 shows the weights of learning on channel axis for two heterogeneous networks. In the first, second, and third layers, weights (α_c and β_c) learned by the 3D convolutional network and ConvLSTM network are the same (values are around 0.5). Because the weights are processed by Softmax, it indicates that the 3D convolutional network and ConvLSTM network are equally important in the channel axis at the initial stage of the model. In the fourth layer, the weight of the 3D convolutional network is significantly higher than that of the ConvLSTM network. The reason may be that at the early stage of the model, the time dimension T is much larger than the width of the 3D convolution kernel, and CAM is concerned with global information, so the short-term and long-term temporal features at the early stage are equally important. In the last layer, the time dimension T (4) is slightly larger than the width of the 3D convolution kernel (3). When T is small, the feature learning ability of ConvLSTM is lower than that of 3D convolution network, so the features learned by 3D convolution are more important. It shows that the model can use two heterogeneous networks to learn gestures simultaneously in the short and long time.

Table 3 Comparison methods of pooling mode on EgoGesture [12]

Networks	Accuracy (%)	
	RGB	Depth
R2plus1D + channel + spatial(avg)	93.53	93.68
R2plus1D + channel + spatial(max)	93.25	93.49
R2plus1D + channel + spatial(avg & max)	93.65	93.81

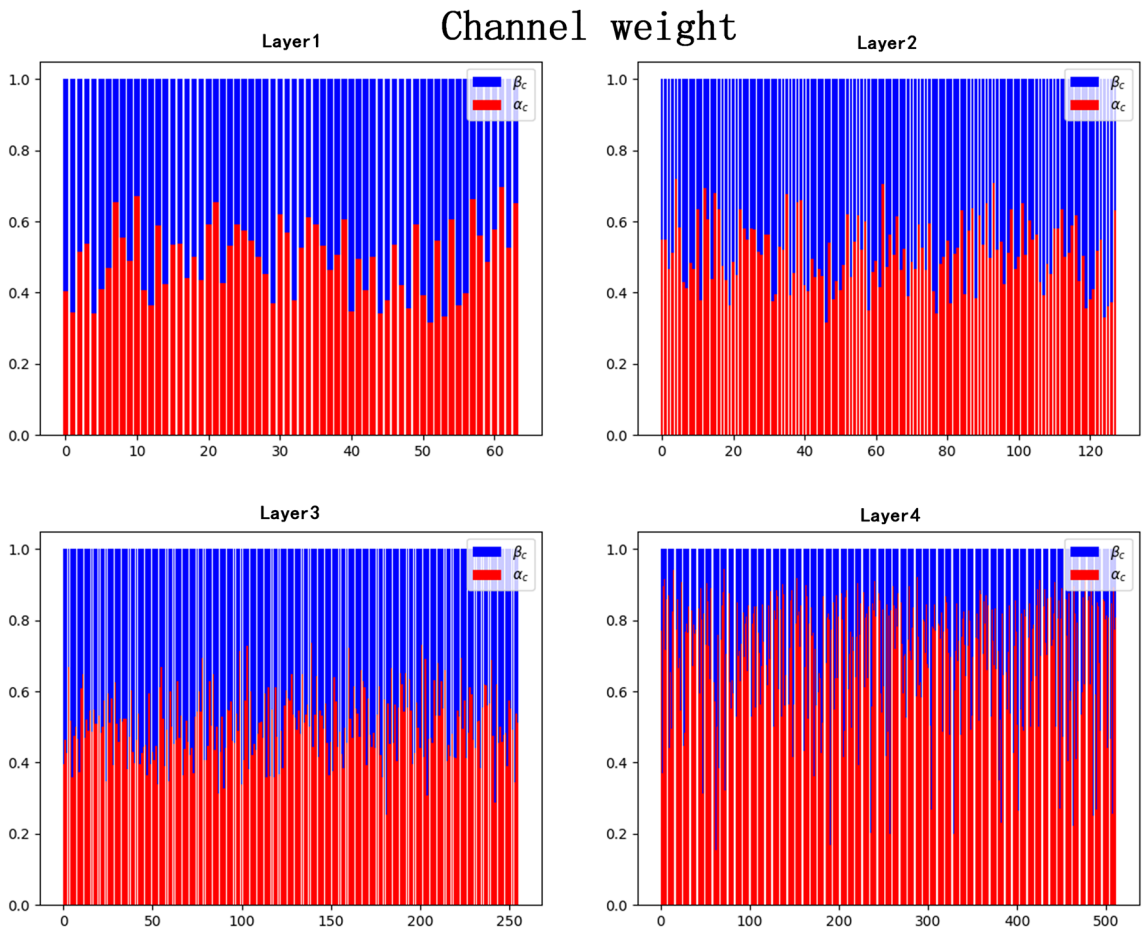


Fig. 8 Compare the weights of two heterogeneous networks on channel axis. Layer1, Layer2, Layer3, and Layer4 represent the output of model weights of the first, second, third, and fourth layers of the model respectively

We selected the Photo Frame category of EgoGesture [12] validation to the trained network and output the weight of each layer's spatial axis. Since the weight of the spatial axis is three-dimensional ($T \times H \times W$), we select the weight of the fixed time dimension ($^{T}/_{2}$) in each layer. Figure 9 shows the weight of learning on spatial axis of two heterogeneous networks. In the first and second layers, the weight learned by the 3D convolutional network (α_s) is less than that learned by the ConvLSTM network (β_s). It indicates that in the first two layers of the model, the features learned by ConvLSTM on the spatial axis are more important than the 3D convolutional network. In the third layer, the weight of two heterogeneous networks is close, and the weight of the 3D convolutional network in the last layer is significantly higher than that of the ConvLSTM network. The reason may be that the time dimension T is large in the initial stage of the model, and the improved GateConvLSTM network can learn the spatial features for a long time through the memory gate structure. As SAM pays more attention to local information, the local information learned by ConvLSTM is richer and the weight (β_s) learned is higher. In the later time of the model, dimension T is close to the width of the convolution kernel, so the weight (α_s) of 3D convolution increases and significantly higher than ConvLSTM.

The experiment proves that the contribution of the ConvLSTM network can be ignored in Layer4, whether it is the channel axis or the spatial axis. So we set Layer4 as a 3D convolutional network. The final model is shown in Fig. 1.

Final results of the EgoGesture, SKIG, and IsoGD datasets

We compare our method with the other state-of-the-art methods [38–49] on EgoGesture [12] SKIG [13] and IsoGD [14] datasets. It can be seen in Tables 4, 5, and 6. It shows that the final model achieves higher accuracy in the EgoGesture [12] data set compared with the RPCNet module used in the last layer. The final model is used on the SKIG [13] and IsoGD [14] datasets and the accuracy is also higher than SeST [11]. To the best of our knowledge, our model is state-of-the-art on SKIG [13] and IsoGD [14] datasets in RGB and Depth modes. Although our model is not state-of-the-art on

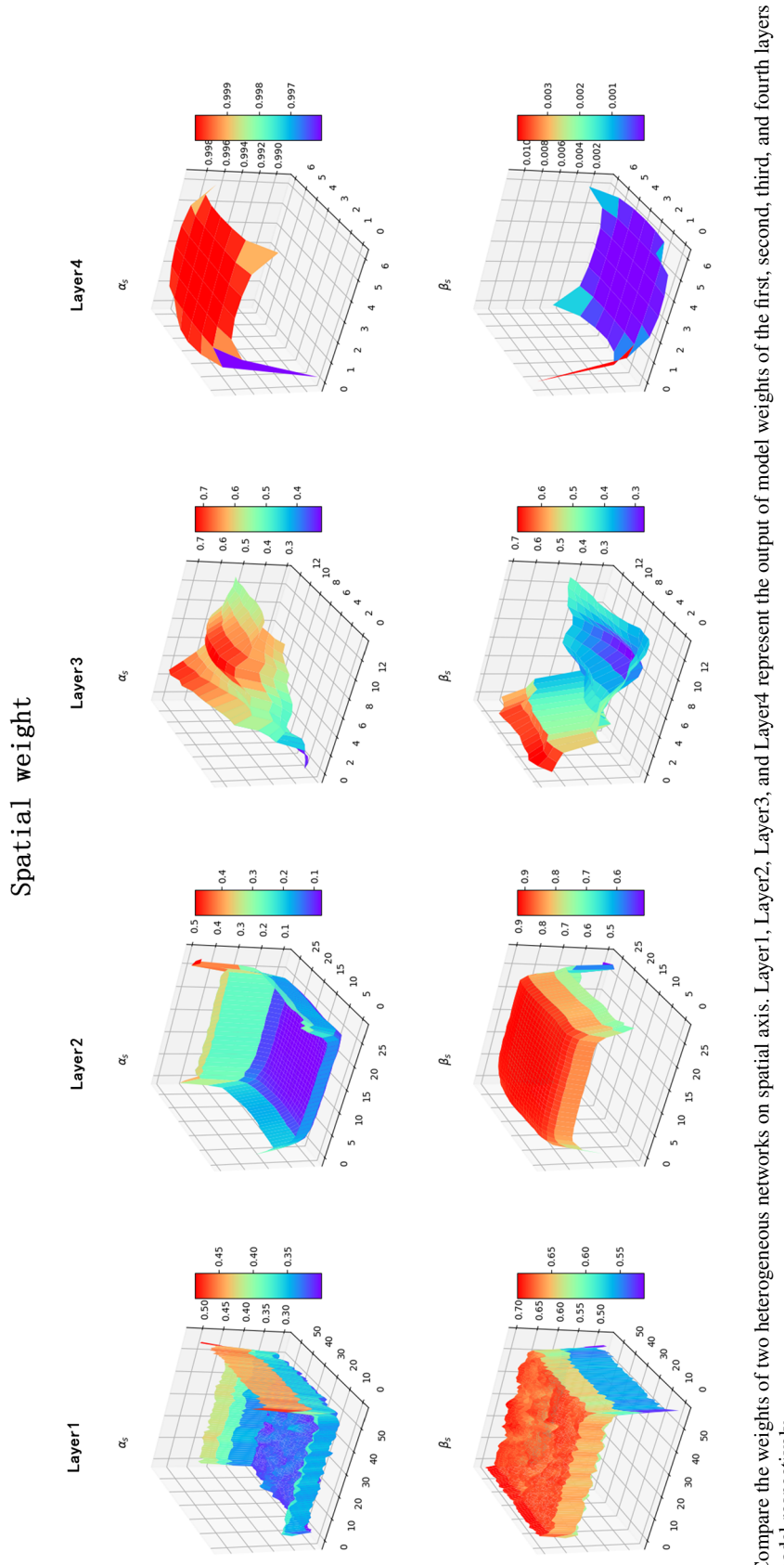


Fig. 9 Compare the weights of two heterogeneous networks on spatial axis. Layer1, Layer2, Layer3, and Layer4 represent the output of model weights of the first, second, third, and fourth layers of the model respectively

Table 4 Comparison results of our models with other state-of-the-art methods on the test set of EgoGesture [12] dataset

Networks	Accuracy (%)	
	RGB	Depth
VGG16 [38]	62.50	62.30
C3D [7]	86.88	88.45
C3D + LSTM + RSTTM [39]	89.30	90.60
CatNet [40]	90.05	90.09
MTUT [41]	92.48	91.96
SeST [11]	93.20	93.35
ResNeXt-101 [42]	93.75	94.03
STCA-R(2 + 1)D [43]	94.00	–
ACTION-Net [44]	94.40	–
RPCNet (initial model)	93.65	93.81
RPCNet (final model)	93.93	94.14

Table 5 Comparison results of our models with other state-of-the-art methods on the test set of SKIG [13] dataset

Networks	Modality	Accuracy (%)
RGGP + RGBD [45]	RGB	99.63
MRNN [46]	RGBD	97.80
DesNet + Bi-LSTM [47]	RGBD	99.07
SeST [11]	RGB	99.63
RPCNet (ours)	RGB	99.70

Table 6 Comparison results of our models with other state-of-the-art methods on the test set of IsoGD [14] dataset

Networks	Accuracy (%)	
	RGB	Depth
ResNet50 [48]	33.22	33.22
3DCNN + ConvLSTM + 2DCNN [28]	51.31	49.81
Res3D + ConvLSTM [29]	53.81	56.35
Res3D + GateConvLSTM + Pyramid [36]	57.42	54.18
SeST [11]	60.27	57.02
AlexNet + LSTM [49]	63.51	51.29
RPCNet (ours)	69.30	69.65

EgoGesture [12], the two heterogeneous networks can be replaced by other networks. We can replace it with a very advanced convnets-based models and recurrent neural network-based models.

Discussion

Through the experiment in "Experiment", it can be concluded that the simultaneous connection of 3D convolutional network and ConvLSTM network can simultaneously learn the short-term and long-term spatiotemporal features of gestures. Compared with the SeST [11] model, which only uses channel information to connect two heterogeneous networks in parallel, this paper proposes to connect two heterogeneous networks using CAM and SAM simultaneously and proves through the melting experiment that the performance of the model using CAM and SAM at the same time is higher than that using CAM only. At the same time, the sequence and pooling operation of CAM and SAM were compared. In the final model shown in Figs. 5, 6, and 7, we followed the order of CAM and SAM, using both average pooling and maximum pooling. In addition, the proportion of short-term and long-term features on channel and spatial axes in each stage of feature learning is quantitatively analyzed. Through the final experiment, it is proved that our model can learn the short-term and long-term spatiotemporal features of gestures in the first three stages of gesture learning, whether on the spatial axis or the channel axis and improved the classification results. To the best of our knowledge, our model is the state-of-the-art on SKIG [13] and IsoGD [14] datasets in RGB and Depth modes. The confusion matrix of the model on the EgoGesture [12] dataset is shown in Fig. 10. Although our model is not state-of-the-art on EgoGesture [12], the two heterogeneous networks can be replaced by other networks. We can replace it with a very advanced convnets-based models and recurrent neural network-based models. The core of our work is to build an architecture that combines convnets-based models and recurrent neural network-based models in parallel so that the fusion architecture can simultaneously learn the long-term and short-term spatiotemporal features of gestures.

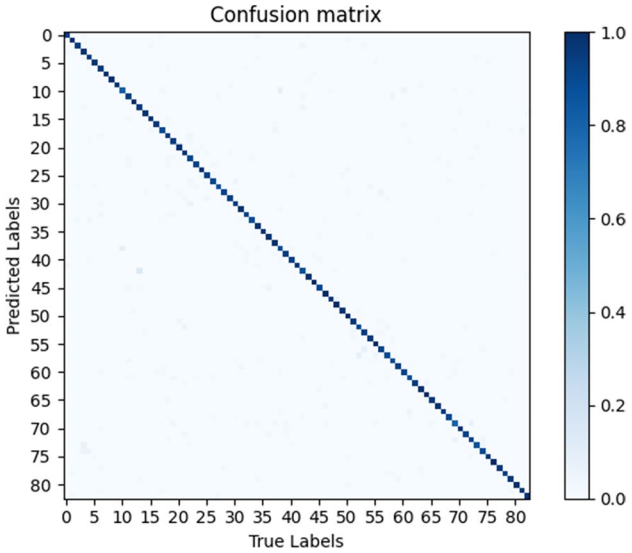


Fig. 10 Confusion matrix of EgoGesture

Conclusion

In this paper, a ConvLSTM which can be sampled down is designed and combined with R2plus1D in parallel. And a deep network structure for learning dynamic gesture features is proposed by using CAM and SAM. And discuss the combination of CAM and SAM and the influence of maximum pooling and average pooling on the model accuracy. The effectiveness of our model is proved by quantitative analysis of the proportions of short-term and long-term features on channel and spatial axes in each stage of feature learning. This structure cannot only use two heterogeneous networks R2plus1D and GateConvLSTM to learn long-term and short-term spatiotemporal features, respectively but also use an attention mechanism to automatically allocate the contributions and dependencies of the two networks in the channel and spatial axes. The network is an end-to-end model. Finally, the effectiveness of the proposed method is verified by comparing it with the existing method on three public dynamic gesture datasets. However, this model still has many limitations. For example, the dynamic gesture recognition method in this paper requires the operator to complete a complete gesture to be recognized, which causes a certain delay in the human–computer interaction based on gesture recognition. And our model takes a video as input. In real human–computer interaction applications, the model generates error detection because it cannot distinguish between intentional and subconscious interactions. In the future, we will try to use the most advanced 3D convolutional network and ConvLSTM network to learn the short-term and long-term spatiotemporal features of gestures in parallel, and design a more designed and lightweight model for online gesture recognition.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Lien J, Gillian N, Karagozler ME, Amihood P, Schwesig C, Olson E, Raja H, Poupyrev I (2016) Soli: ubiquitous gesture sensing with millimeter wave radar. ACM Trans Graph 35(4):1–19
2. Nymoen K, Haugen MR, Jensenius AR (2015) Mumyo–evaluating and exploring the myo armband for musical interaction. In: Proceedings of the international conference on new interfaces for musical expression
3. Parcheta Z, Martínez-Hinarejos C-D (2017) Sign language gesture recognition using HMM. In: Iberian conference on pattern recognition and image analysis. Springer, pp.419–426
4. Wieczorek M, Sika J, Wozniak M, Garg S, Hassan M (2021) Lightweight CNN model for human face detection in risk situations. IEEE Trans Ind Inf 18(7):4820–4829
5. Basak H, Kundu R, Singh PK, Ijaz MF, Woźniak M, Sarkar R (2022) A union of deep learning and swarm-based optimization for 3D human action recognition. Sci Rep 12(1):1–17
6. Yan G, Woźniak M (2022) Accurate key frame extraction algorithm of video action for Aerobics online teaching. Mobile Netw Appl 1–10
7. 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
8. 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
9. Park J, Woo S, Lee J-Y, Kweon IS (2018) BAM: Bottleneck attention module. http://arxiv.org/abs/1807.06514
10. 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
11. Tang X, Yan Z, Peng J, Hao B, Wang H, Li J (2021) Selective spatiotemporal features learning for dynamic gesture recognition. Expert Syst Appl 169:114499
12. Zhang Y, Cao C, Cheng J, Lu H (2018) Egogesture: a new dataset and benchmark for egocentric hand gesture recognition. IEEE Trans Multimed 20(5):1038–1050
13. Klaser A, Marszałek M, Schmid C (2008) A spatio-temporal descriptor based on 3d-gradients. In: BMVC 2008–19th British machine vision conference. British Machine Vision Association, pp 271–275
14. Wan J, Zhao Y, Zhou S, Guyon I, Escalera S, Li SZ (2016) Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 56–64
15. Ibrahim NB, Selim MM, Zayed HH (2018) An automatic Arabic sign language recognition system (ArSLRS). J King Saud Univ Comput Inf Sci 30(4):470–477
16. Yang X, Zhang C, Tian Y (2012) Recognizing actions using depth motion maps-based histograms of oriented gradients. In: Proceedings of the 20th ACM international conference on multimedia, pp 1057–1060
17. Wang H, Schmid C (2013) Action recognition with improved trajectories. In: Proceedings of the IEEE international conference on computer vision, pp 3551–3558
18. Wang L, Xiong Y, Wang Z, Qiao Y (2015) Towards good practices for very deep two-stream convnets. http://arxiv.org/abs/1507.02159.
19. Wu J, Ishwar P, Konrad J (2016) Two-stream CNNs for gesture-based verification and identification: learning user style. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 42–50

20. Funke I, Bodenstedt S, Oehme F, von Bechtolsheim F, Weitz J, Speidel S (2019) Using 3D convolutional neural networks to learn spatiotemporal features for automatic surgical gesture recognition in video. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 467–475
21. Miao Q, Li Y, Ouyang W, Ma Z, Xu X, Shi W, Cao X (2017) Multimodal gesture recognition based on the resc3d network. In: Proceedings of the IEEE international conference on computer vision workshops, pp 3047–3055
22. Pigou L, Van Den Oord A, Dieleman S, Van Herreweghe M, Dambre J (2018) Beyond temporal pooling: recurrence and temporal convolutions for gesture recognition in video. Int J Comput Vis 126(2):430–439
23. Shi L, Zhang Y, Hu J, Cheng J, Lu H (2019) Gesture recognition using spatiotemporal deformable convolutional representation. In: 2019 IEEE international conference on image processing (ICIP). IEEE, pp 1900–1904
24. Wan J, Escalera S, Anbarjafari G, Jair Escalante H, Baró X, Guyon I, Madadi M, Allik J, Gorbova J, Lin C (2017) Results and analysis of chalearn lap multi-modal isolated and continuous gesture recognition, and real versus fake expressed emotions challenges. In: Proceedings of the IEEE international conference on computer vision workshops, pp 3189–3197
25. Shi X, Chen Z, Wang H, Yeung DY, Wong WK, Woo WC (2015) Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems, pp 802–810
26. Molchanov P, Yang X, Gupta S, Kim K, Tyree S, Kautz J (2016) Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4207–4215
27. Nunez JC, Cabido R, Pantrigo JJ, Montemayor AS, Velez JF (2018) Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. Pattern Recognit 76:80–94
28. Zhang L, Zhu G, Shen P, Song J, Afaq Shah S, Bennamoun M (2017) Learning spatiotemporal features using 3dcnn and convolutional lstm for gesture recognition. In: Proceedings of the IEEE international conference on computer vision workshops, pp 3120–3128
29. Elboushaki A, Hannane R, Afdel K, Koutti L (2020) MultiD-CNN: a multi-dimensional feature learning approach based on deep convolutional networks for gesture recognition in RGB-D image sequences. Expert Syst Appl 139:112829
30. Peng Y, Tao H, Li W, Yuan H, Li T (2020) Dynamic gesture recognition based on feature fusion network and variant ConvLSTM. IET Image Proc 14(11):2480–2486
31. Wang P, Li W, Gao Z, Tang C, Ogunbona PO (2018) Depth pooling based large-scale 3-d action recognition with convolutional neural networks. IEEE Trans Multimed 20(5):1051–1061
32. Hou J, Wang G, Chen X, Xue J-H, Zhu R, Yang H (2018) Spatial-temporal attention res-TCN for skeleton-based dynamic hand gesture recognition. In: Proceedings of the European conference on computer vision (ECCV) workshops
33. Wiederer J, Bouazizi A, Kressel U, Belagiannis V (2020) Traffic control gesture recognition for autonomous vehicles. In: 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 10676–10683
34. Dhingra N, Kunz A (2019) Res3atn-deep 3d residual attention network for hand gesture recognition in videos. In: 2019 international conference on 3D vision (3DV). IEEE, pp 491–501
35. Zhang L, Zhu G, Mei L, Shen P, Shah SAA, Bennamoun M (2018) Attention in convolutional LSTM for gesture recognition. In: Proceedings of the 32nd international conference on neural information processing systems, pp 1957–1966
36. Zhu G, Zhang L, Yang L, Mei L, Shah SAA, Bennamoun M, Shen P (2019) Redundancy and attention in convolutional LSTM for gesture recognition. IEEE Trans Neural Netw Learn Syst 31(4):1323–1335
37. Materzynska J, Berger G, Bax I, Memisevic R (2019) The jester dataset: a large-scale video dataset of human gestures. In: Proceedings of the IEEE/CVF international conference on computer vision workshops
38. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. http://arxiv.org/abs/1409.1556
39. Zhang L, Zhu G, Mei L, Shen P, Shah SAA, Bennamoun M (2018) Attention in convolutional LSTM for gesture recognition. In: Advances in neural information processing systems, p 31
40. Wang Z, She Q, Chalasani T, Smolic A (2020) Catnet: class incremental 3d convnets for lifelong egocentric gesture recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pp 230–231
41. Abavisani M, Joze HRV, Patel VM (2019) Improving the performance of unimodal dynamic hand-gesture recognition with multimodal training. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1165–1174
42. Köpüklü O, Gunduz A, Kose N, Rigoll G (2019) Real-time hand gesture detection and classification using convolutional neural networks. In: 2019 14th IEEE international conference on automatic face & gesture recognition (FG 2019). IEEE, pp 1–8
43. Han X, Lu F, Yin J, Tian G, Liu J (2022) Sign language recognition based on R (2+ 1) D With spatial–temporal–channel attention. IEEE Trans Hum Mach Syst 1–12
44. Wang Z, She Q, Smolic A (2021) Action-net: Multipath excitation for action recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 13214–13223
45. Liu L, Shao L (2013) Learning discriminative representations from RGB-D video data. In: Twenty-third international joint conference on artificial intelligence, pp 1493–1500
46. Nishida N, Nakayama H (2015) Multimodal gesture recognition using multi-stream recurrent neural network. In: Image and video technology. Springer, pp 682–694
47. Li D, Chen Y, Gao M, Jiang S, Huang C (2018) Multimodal gesture recognition using densely connected convolution and blstm. In: 2018 24th international conference on pattern recognition (ICPR). IEEE, pp 3365–3370
48. Narayana P, Beveridge R, Draper BA (2018) Gesture recognition: focus on the hands. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5235–5244
49. Rastgoo R, Kiani K, Escalera S (2021) Hand pose aware multimodal isolated sign language recognition. Multimed Tools Appl 80(1):127–163

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