IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 17, 2020, accepted September 24, 2020, date of publication September 28, 2020,
date of current version October 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3027386

Temporal Segment Connection Network for Action Recognition

QIAN LI, WENZHU YANG, XIANGYANG CHEN, TONGTONG YUAN, AND YUXIA WANG
School of Cyber Security and Computer, Hebei University, Hebei 071002, China
Corresponding author: Wenzhu Yang (wenzhuyang@163.com)

This work was supported in part by the Natural Science Foundation of Hebei Province under Grant F2020201011.

ABSTRACT Two-stream Convolutional Neural Networks have shown excellent performance in video action recognition. Most existing works train each sampling group independently, or just fuse at the last level, which obviously ignore the continuity of action in temporal and the complementary information between action fragments. In this paper, a temporal segment connection network is proposed to overcome these limitations. On the one hand, the forget gate module of the long short-term memory (LSTM) network is used to establish feature-level connections between each sampling group. This not only strengthens the information transmission between the sampling groups to enhance the temporal connectivity, but also extracts the complementary information between the sampling groups to enhance the overall representation of the action. On the other hand, a bi-directional long short-term memory (Bi-LSTM) network is used to automatically evaluate the importance weights of each sampling group based on the deep feature sequence. The experimental results on UCF101 and HMDB51 datasets show that the proposed model can effectively improve the utilization rate of temporal information and the ability of overall action representation, thus significantly improves the accuracy of human action recognition.

INDEX TERMS Action recognition, convolutional neural network, two-stream, forget-gate connection module, adaptive weighting module.

I. INTRODUCTION

Video-based action recognition attracts extensive attention due to its applications in many fields like security and behavior analysis. Hand-crafted features are mainly used in early works to represent actions [1]–[5], but now the method of automatic feature extraction based on deep-learning has become the mainstream [6]–[10]. Among the deep learning-based methods, the methods developed from the two-stream network have excellent recognition effects and promote action recognition to a new record [11]–[14].

Unlike image recognition, video action recognition includes both spatial appearance information and temporal motion information. Therefore, whether the spatiotemporal information can be fully utilized is the key to improving the performance of action recognition. Two-stream approaches train two independent CNNs, one operating on the appearance using RGB data, the other one processing motion based on optical flow images. However, the original two-stream convolutional neural network has some shortcomings. The original two-stream convolutional neural network can combine spatial and temporal information, but it only focuses on short-term motion changes and does not capture long-term information about the video. To address this issue, Wang and Xiong [15] proposed a Temporal Segment Network (TSN) to extract several sampling groups from a video to enhance the long-term modeling ability of the network. On the basis, some researchers believe that a spatiotemporal interaction mechanism should be added to the spatiotemporal networks to make full use of the complementary information between them. For example, Hao and Zhang [16] established dense connections between spatiotemporal networks based on DenseNet [17], and enhanced spatial information with temporal information to improve the interactivity between spatial and temporal networks. Zhang and Hu [18] added a spatiotemporal fusion network to extract additional spatiotemporal fusion information. However, each sampling group still trains independently, which obviously destroys the temporal connectivity and ignores the complementary information between the sampling groups. When temporal connectivity is broken, actions such as ''stand up'' and ''sit down'' may be misrecognized. The amount of complementary information between

The associate editor coordinating the review of this manuscript and approving it for publication was Varuna De Silva.

sampling groups depends on the heterogeneity between them. The more heterogeneity between the sampling groups contain more complementary information. Conversely, the more similar the sampling groups are, the less complementary information they contain. We believe that the sampling groups should be fully utilized in order to express actions more globally.

To satisfy the abovementioned requirements, a temporal segment connection network (TSCN) is designed to integrate and connect sampling groups in temporal. The TSCN first introduces a forget-gate connection module (FCM), which establishes feature-level connections between each sampling group to complete the transfer of information. Then an adaptive weighting module (AWM) is used to automatically evaluate the importance weights of each sampling group.

The contributions of this paper are summarized as follows:

- A novel Temporal Segment Connection Network (TSCN) is presented for action recognition. The feature-level forget-gate connections are established between adjacent sampling groups, which can not only enhance the temporal connection, but also extract the complementary information between the sampling groups.
- The strategy of endowing weights based on context information is introduced, which can automatically evaluate the importance weights of each sampling group.
- Our model obtains promising performance in action recognition on two benchmark datasets, including UCF101 and HMDB51 respectively.

The rest of this paper is organized as follows. In Section 2, related works are introduced. In Section 3, our TSCN is described in detail. Experimental results are presented in Section 4. Finally, in Section 5, we conclude the paper with future works.

II. RELATED WORKS

Action recognition has been extensively explored in past years [2], [8]. Previous related works fall into two categories: 1) video action recognition, 2) feature representation.

A. VIDEO ACTION RECOGNITION

Video action recognition has been extensively studied in recent years. Early works focus on developing good hand-crafted features for representation actions [1], [2], [5]. The performances of these methods are often restrained due to the limited differentiation capability of hand-crafted features.

With the development of deep learning, many deep learning-based methods are proposed for action recognition, which use ConvNets to automatically obtain the feature representation for actions. Simonyan and Zisserman [7] proposed a two-stream convolutional network which used two ConvNets to extract spatial and temporal features from RGB and Flow images respectively. The final action classification score is obtained by fusing the scores of the two streams. To further improve the action recognition performance, many extensions of two-stream ConvNet are proposed. In [15], Wang *et al*. proposed a Temporal Segment Network (TSN) for action recognition, aiming to improve the long-term modeling capabilities of two-stream network. Tu and Li [11] designed an action-stage emphasized spatiotemporal vector of locally aggregated descriptors (ActionS-ST-VLAD) method to aggregate informative deep features across the entire video. By combining the traditional streams with the novel human-related streams, a human-related multi-stream CNN (HR-MSCNN) architecture was designed [13]. Huang and Kuang [19] introduced an Optical Flow guided Feature (OFF), which can replace optical flow to quickly extract robust temporal information by convolutional neural network. Chen and Bai [20] proposed a spatiotemporal heterogeneous two-stream network, which employs two different network structures for spatial and temporal information.

On the other hand, 3D convolutional neural network can simultaneously learn the appearance and temporal information by using the 3D convolution operation [8]. Considering the large scale of related calculations, Qiu and Yao [21] reformed the 3D convolution and proposed Pseudo-3D Residual Network (P3D ResNet). P3D ResNet replaces the $3\times3\times3$ kernels with $1\times3\times3$ kernels operating on spatial domain and $3\times3\times1$ kernels operating on temporal domain.

Encouraged by the recent success of LSTM [22] in speech recognition, some researchers are trying to apply LSTM in video action recognition. Shi and Chen [23] presented a convolutional LSTM (ConvLSTM) network, which replaces weight operations with convolution operations so that the spatial features can be extracted. Then the Lattice LSTM [24] and the Correlational Convolutional LSTM [25] were proposed one after another, which both developed from ConvLSTM.

B. FEATURE REPRESENTATION

The previous action recognition works mainly used the method of image recognition to extract monotonous features to represent actions. Considering that video action recognition contains extra temporal information, how to improve the ability of action feature representation is the key to improving the effect of action recognition. Hao *et al.* proposed a Spatiotemporal Distilled Dense-Connectivity Network (STDDCN), which builds block-level dense connections between appearance and motion streams to enhance the spatiotemporal interactive function of the network. Zhang *et al.* presented a 3D Multi-Level Dense Fusion (MLSF-3D) model, which adds an additional spatiotemporal fusion stream to explore the potential relation of features extracted from two-stream network. Moreover, to explore more effective feature representation methods, [26] introduced a spatiotemporal relation feature representation model with only RGB input and [27] presented a pose motion representation (PoTion) method with some semantic key points.

Our approach is similar to multi-stream interaction methods, but with some differences: 1) The previous works transmitted spatiotemporal information between multi-stream networks. Comparatively, our approach establishes feature-level connections between sampling groups to obtain a more global feature representation. 2) The previous

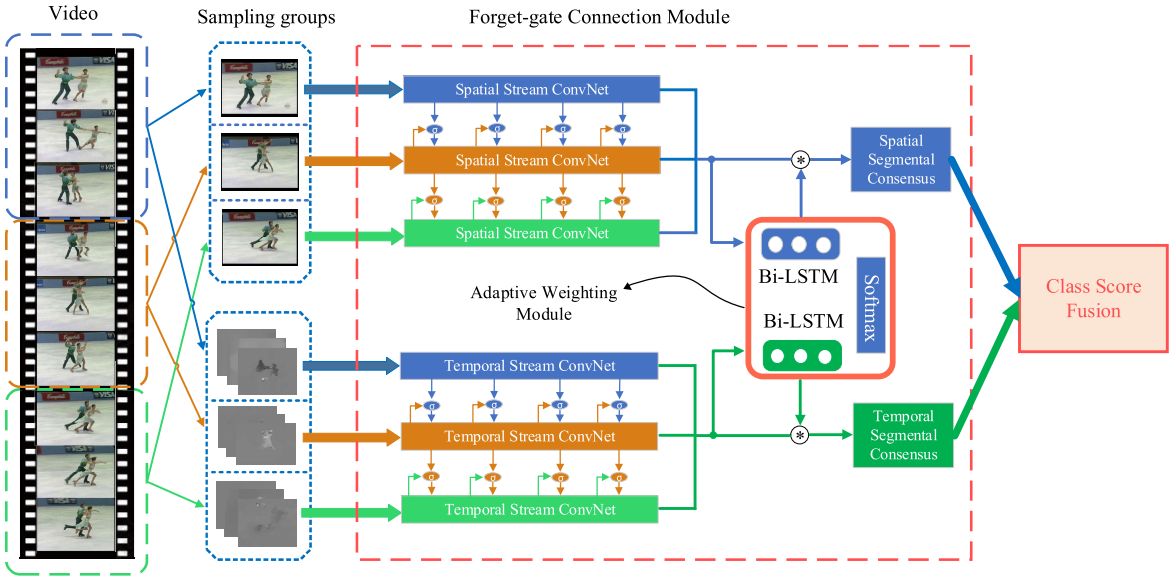


FIGURE 1. Temporal Segment Connection Network (TSCN): One input video is divided into K segments and a sampling group is randomly selected from each segment. Then, the sampling groups are input into a forget-gate connection module to calculate the deep features and predict the action label of the input video. At the same time, an adaptive weighting module is applied to evaluate importance weights of the sampling groups. Finally, the importance weights are combined with the predictions of each sampling group to obtain the final video-level prediction.

works do not differentiate the importance of different sampling groups. Comparatively, our approach introduces an adaptive weighting module to automatically evaluate the importance weights of each sampling group.

III. TEMPORAL SEGMENT CONNECTION NETWORK

The framework of the proposed TSCN is shown as Fig. 1. TSCN is mainly composed of one forget-gate connection module and one adaptive weighting module. The purpose of TSCN is to strengthen temporal information and extract complementary information between sampling groups to obtain a more global representation of actions.

A. ARCHITECTURE

The architecture of the proposed TSCN is developed from TSN, which aims at modeling long-term temporal structure. TSN operates on a sequence of short snippets sparsely sampled from the entire video instead of a single frame or frame stack. Each snippet generates an action class prediction, and finally gathers all predictions of sampling groups to obtain the final video-level prediction.

Based on TSN, our model TSCN establishes feature-level forget-gate connections between sampling groups to obtain a more global representation of actions. Then TSCN further introduces an adaptive weighting module to adaptively evaluate the importance of each sampling group. Specifically, our backbone network is DenseNet.

Formally, given a video V, we divide it into K segments $\{S_1, S_2, \ldots, S_K\}$ with equal durations. The sampling group s_i, $i \in [1, K]$ is randomly sampled from the corresponding sequence segment S_i. Then the sampling groups are input into the forget-gate connection module to calculate the deep feature and predict the action label of the input video. At the same time, an adaptive weighting module is applied to evaluate the importance weights of each sampling group. Finally, the importance weights are combined with the prediction of each sampling group to obtain the final video-level prediction.

B. FORGET-GATE CONNECTION MODULE

An Forget-gate Connection Module (FCM) is introduced to overcome the limitation that the entire video information is hardly represented due to the independent training for each sampling group. Instead of layer-to-layer feature connections, block-level feature connections are established between adjacent sampling groups, which can not only enhance the generalization ability of the network, but also increase the propagation speed. In detail, the feature-level forget-gate connections are established between adjacent sampling groups to extract temporal information G_t, and then G_t is input to the latter sampling group as the action supplement information of the previous sampling group. Finally, the feature-level forget-gate connections are established between all sampling groups to complete the transfer of global action information. Considering that the sampling groups are connected in the same way on Flow and RGB, only the connection method of RGB is shown here. The detailed structure of FCM is shown as Fig. 2.

Concretely, the feature-level forget-gate connection can be formulated as:

$$X_{R+1}^{i+1} = p(X_{R+1}^{i}) + G(p(X_{R+1}^{i}), p(X_{R}^{i})) \tag{1}$$

where X_R^i and X_{R+1}^i represent the inputs for the i^{th} block of the adjacent RGB sampling groups. $p(\cdot)$ presents the original function that transfers the input of the i^{th} block to the $(i+1)^{\text{th}}$ block in the corresponding ConvNet. Operator $+$ indicates the elementwise addition. $G(\cdot)$ denotes the forget-gate operation,

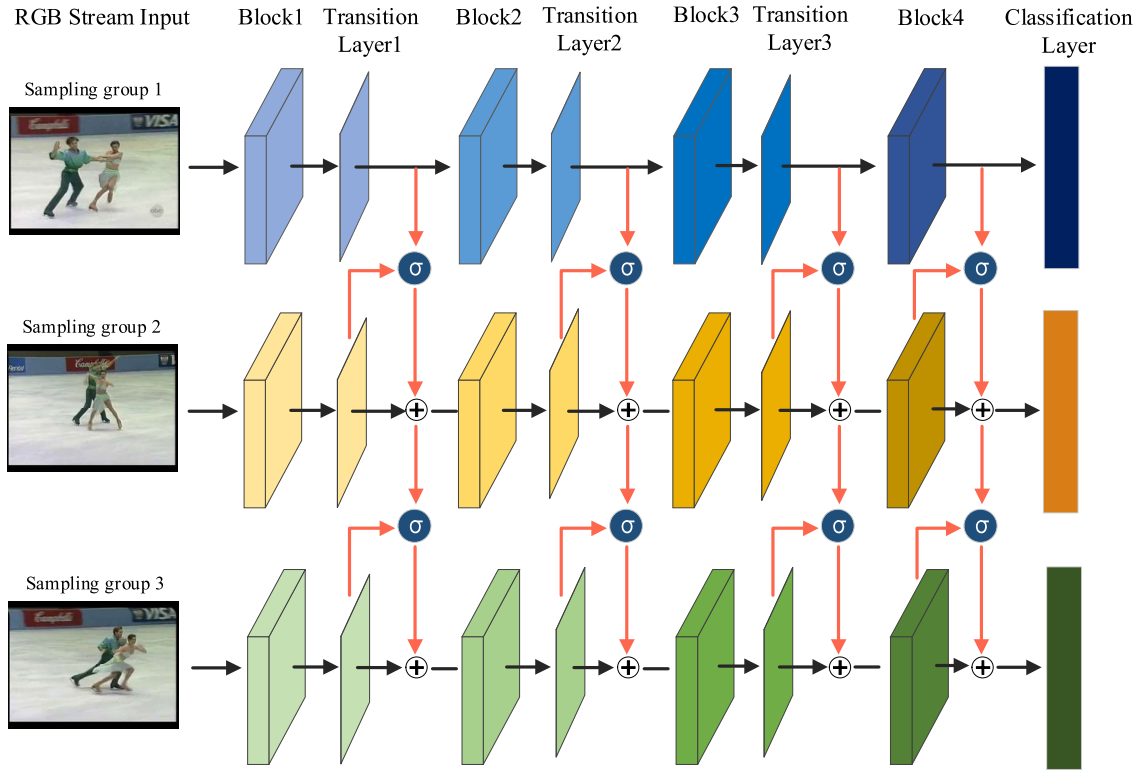


FIGURE 2. The architecture of Forget-gate Connection Module (FCM). σ represents the feature-level forget-gate connection, which is equivalent to the $G(\cdot)$ function in Eq. (1), and + represents the element-level addition.

which can be depicted as:

$$G(p(X^i_{R+1}), p(X^i_R)) = f * (p(X^i_{R+1}) \cdot p(X^i_R)) \tag{2}$$

$$f = \sigma(W_{R+1} \cdot (X^i_{R+1} + W_R \cdot p(X^i_R + b_f))) \tag{3}$$

where $\cdot$ represents elementwise multiplication, $*$ represents matrix multiplication. $\sigma(\cdot)$ presents the *Sigmoid* function, W_{R+1}, W_R and b_f represent weights and bias respectively. After the *Sigmoid* function operation, f will get a matrix whose elements range from 0 to 1. Then f is used to adjust the amount of information input into the next sampling group.

Eq. (1) illustrates that the input of the $(i+1)^{\text{th}}$ block of the current sampling group is the combination of the output of the i^{th} block and the output of the i^{th} block of the previous sampling group. Based on the above formulas, the gradient of the loss function L in backpropagation can be expressed as:

$$\frac{\partial L}{\partial X^i_R} = \frac{\partial L}{\partial X^{i+1}_R}\frac{\partial X^{i+1}_R}{\partial X^i_R} \tag{4}$$

$$\frac{\partial L}{\partial X^i_{R+1}} = \frac{\partial L}{\partial X^{i+1}_{R+1}}\frac{\partial X^{i+1}_{R+1}}{\partial X^i_{R+1}} = \frac{\partial L}{\partial X^{i+1}_{R+1}}\left(\frac{\partial P(X^i_{R+1})}{\partial X^i_{R+1}} + \frac{\partial G(P(X^i_{R+1}), P(X^i_R))}{\partial X^i_{R+1}}\right) \tag{5}$$

$$\cdots\cdots,$$

$$\frac{\partial L}{\partial X^i_{R+k-1}} = \frac{\partial L}{\partial X^{i+1}_{R+k-1}}\frac{\partial X^{i+1}_{R+k-1}}{\partial X^i_{R+k-1}} = \frac{\partial L}{\partial X^{i+1}_{R+k-1}} \times \left(\frac{\partial P(X^i_{R+k-1})}{\partial X^i_{R+k-1}} + \frac{\partial G(P(X^i_{R+k-1}), P(X^i_{R+k-2}))}{\partial X^i_{R+k-1}}\right) \tag{6}$$

where k indicates the number of sampling groups in the entire video.

Based on these formulas, the propagation of the gradient traverses all sampling groups, which improves the temporal connection between the sampling groups and enhances the transfer of complementary information.

C. ADAPTIVE WEIGHTING MODULE

In order to distinguish the discrimination ability of different sampling groups, an Adaptive Weighting Module (AWM) is introduced as shown in Fig. 3. The information of the sampling group is extracted and transmitted to the next sampling group, intuitively, the later sampling group contains more information. In other words, the amount of information contained in each sampling group is different. Therefore, AWM is used to evaluate the importance weights of each sampling group dynamically.

Specifically, the sampling groups are input into the FCM to calculate the deep feature sequence $\{M_1, M_2, \ldots, M_k\}$.

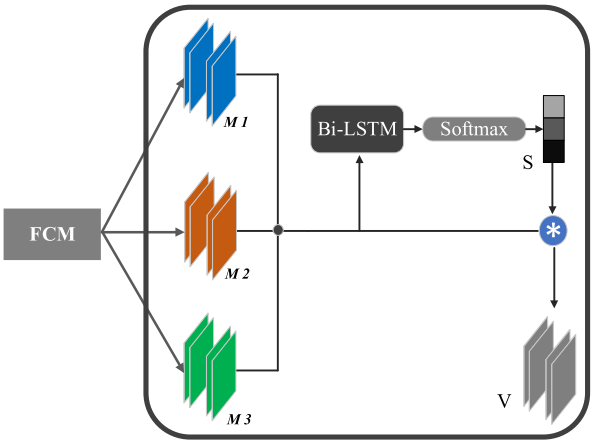


FIGURE 3. The architecture of Adaptive Weighting Module (AWM). The deep feature sequence $\{M_1, M_2, \ldots, M_k\}$ is input into Bi-LSTM to evaluate the importance weights of each sampling group.

To capture the temporal relationship of sampling groups, the deep feature sequence is input into Bi-directional LSTM for evaluation. Compared with the naïve LSTM, Bi-directional LSTM can utilize both the forward and backward direction context information to obtain a more global assessment. The specific formulation is defined as follows:

$$\begin{cases} h_t^f = \tanh(W_x^f x_t + W_h^f h_{t-1}^f + b_h^f), \\ h_t^b = \tanh(W_x^b x_t + W_h^b h_{t+1}^b + b_h^b), \\ u_t = W_u^t h_{t-1}^f + W_u^b h_{t+1}^b + b_u \end{cases} \tag{7}$$

where h_t^f is the forward propagation sequence, h_t^b is the backward propagation sequence and u_t is the output sequence. Bi-direction LSTM computes h_t^f, h_t^b, u_t by iterating the backward layer at time t.

Outputs of Bi-directional LSTM classifier are k fixed-dimension vectors $\{u_1, u_2, \ldots, u_k\}$ and the intermediate vector $u_{(k+1)/2} \in R^k$ is chosen as our proposal. Since the Bi-directional LSTM is adapted, intuitively the intermediate vector $u_{(k+1)/2}$ can get the information from both directions. Then our proposal $u_{(k+1)/2}$ is input into a Softmax layer to obtain a weight vector S. The weight vector S indicates the importance of each sampling group. Finally, the importance weights S are assigned to the deep feature sequence M to obtain the final feature representation V and the prediction result y, as shown:

$$V = \sum_{i=1}^{k} \frac{\exp(S(i))}{\sum_{j=1}^{k} \exp(S(j))} M_i \tag{8}$$

$$y = W^l V \tag{9}$$

where W^l is the parameter of linear transformation of consensus feature V to final prediction result y.

IV. EXPERIMENT AND ANALYSIS

In this section, we first give a brief introduction of two standard benchmarks used in experiments, namely HMDB51 [28] and UCF101 [29]. Then we present the implementation details of the proposed TSCN. Finally, we provide the experimental results and compare our model with current state-of-the-art models.

A. EVALUATION DATASETS

UCF101 is one of the most popular action recognition datasets of realistic action videos. It contains of 13320 videos taken from YouTube, which are divided into 101 action categories. Each category contains videos between [100, 200]. UCF101 is comparatively more challenging dataset due to its large number of action categories from five major types: 1) human-object interaction, 2) body-motion only, 3) human-human interaction, 4) playing musical instruments, and 5) sports.

The HMDB51 dataset contains a variety of actions related to human body movements including objects interaction with body, facial actions, and human interaction for body movements. It consists of 6766 action video clips, which are divided into51 classes, each containing more than one hundred clips. It is more challenging because the clips of each category are collected for a variety of subjects with different illuminations and 4 to 6 clips are recorded for each subject performing the same action on different poses and viewpoints.

We follow the evaluation scheme of the THUMOS13 challenge and adopt the three training/testing splits for evaluation. A few sample images from each action category are given in Fig.4.

B. IMPLEMENTATION DETAILS

We first train the spatial network and temporal network separately as describe in [15]. In our experiments, we use the pretrained DenseNet169 on ImageNet [30] as the backbone model of two-stream network. For the extraction of optical flow images, we use the TVL1 optical flow algorithm implemented in OpenCV with CUDA. During training, we first scale the image size to 256×340 and then crop the image. The width and height of the cropped region are randomly selected from {256, 224, 192, 168}. Next, all of the cropped images are resize to 224×224.

The mini-batch stochastic gradient descent algorithm is used to learn the network parameters, where the batch size is set to 8 to fit the GPU memory and momentum is set to 0.9. Then a smaller learning rate is set in our experiments. For spatial network, the learning rate is set as 0.001, which reduces to its 1/10 after 15,000 and 30,000 iterations. The maximum iteration is set as 50,000. For temporal networks, the learning rate is initiated as 0.005 and decrease to its 1/10 after 95,000 and 150,000 iterations. The whole training procedure stops at 200,000 iterations. In order to avoid over-fitting, typical data augmentation strategies are utilized, including location jittering, scale jittering, horizontal flipping and corner cropping.

C. EXPERIMENTAL RESULTS AND DISCUSSIONS

In order to verify the effect of the TSCN model on action recognition, the basic model TSN is used for comparison on HMDB51 and UCF101. The comparison results are shown

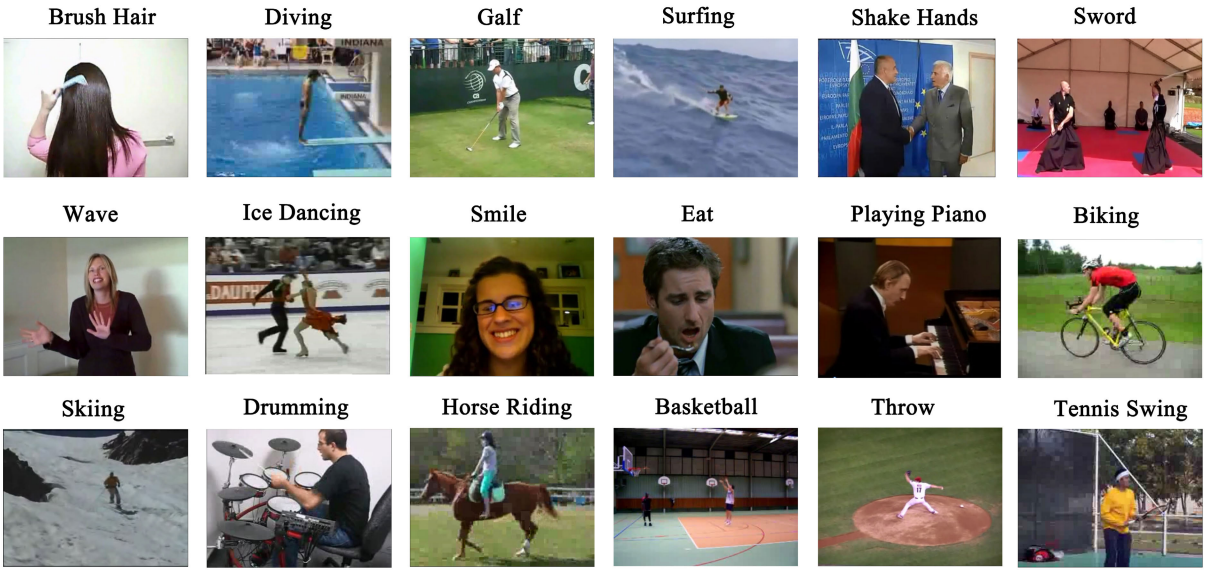


FIGURE 4. Sample action categories of UCF101 and HMDB51 action dataset.

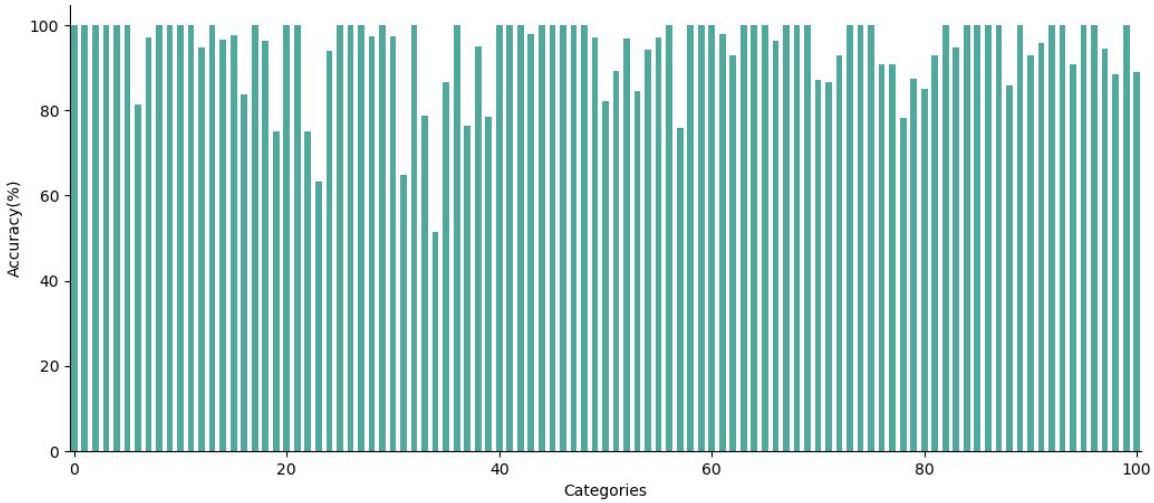


FIGURE 5. Class-wise accuracy of the proposed TSCN on UCF101 for action recognition.

TABLE 1. Comparison results of TSN and our TSCN on HMDB51 and UCF101.

dataset	HMDB51			UCF101
split	Split1	Split2	Split3	Split1
TSN	68.5%	67.4%	69.0%	93.3%
TSCN	**70.8%**	**69.2%**	**70.3%**	**94.2%**

in Table 1. During the training phase, the number of segments is set to 3. In the test phase, the number of segments is set to 24 instead of the original 25 to suit our network structure. The ConvNet used here is DenseNet169. If not stated, the basic ConvNet used in the following sections is DenseNet169.

From Table 1, it is observed that the TSCN yields consistent better results than TSN, which verifies the superiority of TSCN in all splits of HMDB51 and the first split of UCF101.

In order to analyze the experimental results in more detail, we first provide the class-wise accuracy of UCF101 dataset on the test data, as shown in Fig. 5. The horizontal axis represents categories and the vertical axis shows the accuracy of the corresponding category. From results, it can be seen that the results of most of the categories are greater than 80%; some of them reach 100%; and only one category has accuracy less than 60%. The proposed method improved the recognition rate on UCF101 dataset from 93.3% to 94.2%. Then we provide a comparison chart of the accuracy of TSCN and TSN on the test data of HMDB51 dataset, as shown in Fig. 6. The horizontal axis represents categories and the vertical axis shows the accuracy of the corresponding category. It can be seen that the accuracy of 30 categories of TSCN exceeds TSN, the accuracy of more than 10 categories

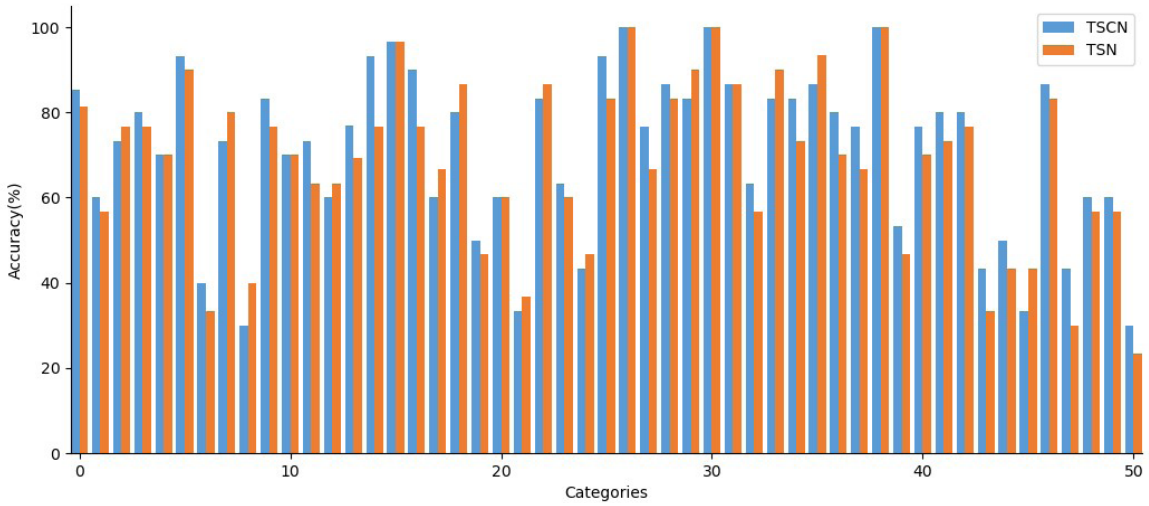


FIGURE 6. Class-wise accuracy of the proposed TSCN and TSN on HMDB51 for action recognition.

TABLE 2. Comparison results of different positions and numbers of FCM on UCF101.

position	RGB	Flow	RGB+Flow
Block1	83.9%	86.9%	93.1%
Block2	83.7%	87%	93.2%
Block3	84%	87.2%	93.3%
Block4	84.1%	87.4%	93.3%
Block3-4	84.2%	87.4%	93.5%
Block2-3-4	84.5%	87.5%	93.7%
Block1-2-3-4	85%	88.1%	94%

is equal to TSN, and the accuracy of 10 categories of TSCN is lower than TSN. The proposed method increased the recognition rate on HMDB51 from 68.3% to 70.3%.

In order to further explore the influence of the position and number of FCM insertion on the experimental results, 7 groups of experiments were compared as shown in Table 2. Wherein "Block n" means adding FCM after the n^{th} dense block. "Block2-3-4" means insert FCM after the second, third and fourth dense blocks respectively. Our basic network, DenseNet169, contains 4 dense blocks in total. Though the comparative experiment, two main phenomena are found. One is that the FCM is inserted in different positions and the experimental results have little difference. The other is that the recognition accuracy will increase as the number of FCM increases.

Then we visualize the block-level features in the network structure to explore the effect of TSCN in more detail, as shown in Fig. 7. For the sake of display, we only show the features extracted from the second and fourth blocks. Two different actions are compared, one is the action "ApplyEyeMakeup" with high similarity between the sampling groups, and the other is the action "Flic_Flac" with strong heterogeneity and more complementary information. In Fig. 7 (a), the sampling groups are similar, so there is little complementary information between the sampling groups. In this case, the performance of our TSCN is similar to that of TSN. In Fig. 7 (b), each sampling group has strong heterogeneity and compact temporal connection. It can be seen that compared with previous sampling group, the latter

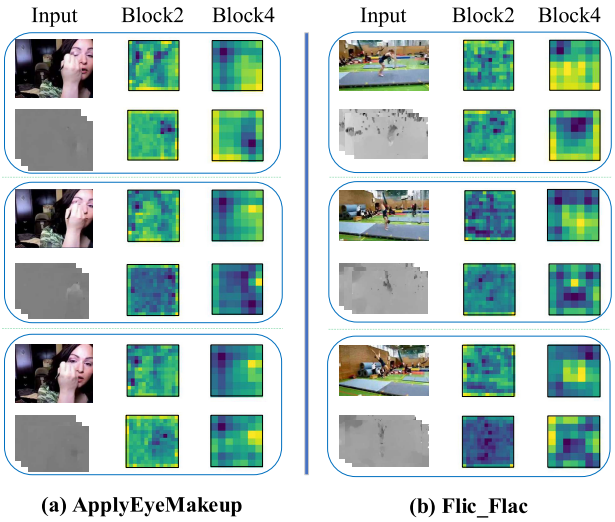


FIGURE 7. Visualize the feature maps of the sampling groups on the second dense block and the fourth dense block of TSCN.

sampling group contains more abundant and abstract information. Rich information can improve the accuracy of action recognition, so TSCN gets a better recognition result on the action "Flic_Flac": 93.3% vs 76.7%. It shows that TSCN can strengthen the temporal connection, improve the utilization of complementary information, and have better recognition results for the more heterogeneous actions.

Establishing feature-level connections between sampling groups will result in different amounts of information contained in different sampling groups, so we introduce an AWM to automatically evaluate the importance weights of each sampling group. To examine the effectiveness of our AWM, the average weighting method (AVG) in TSN is used for comparison as shown in Table 3. From Table 3, it can be seen that the recognition accuracy using our AWM is about 0.2% higher than the accuracy using AVG. In order to show the effect of AWM more clearly, we visualize the importance weight given to each sampling group, as shown

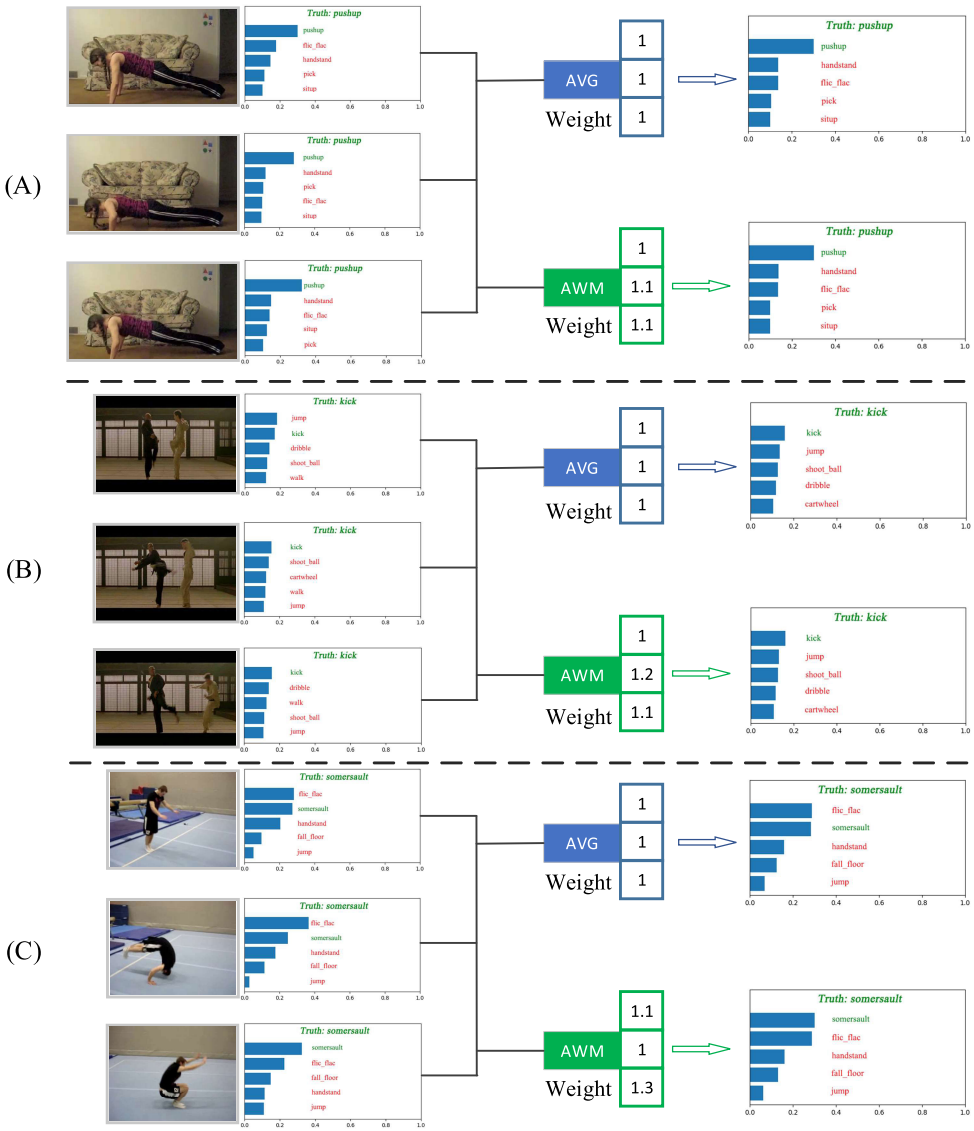


FIGURE 8. Visualize the weight distribution of AWM to the sampling groups. Three situations (A), (B) and (C) are used for comparative analysis to prove the effectiveness of our AWM.

TABLE 3. Comparison of methods with average weighting and adaptive weighting.

dataset	HMDB51			UCF101
Split	Split1	Split2	Split3	Split1
AVG	70.6%	69.0%	70.2%	94.0%
AWM	**70.8%**	**69.2%**	**70.3%**	**94.2%**

TABLE 4. Comparison of TSCN with different model depth on HMDB51 and UCF101.

dataset	HMDB51			UCF101
split	Split1	Split2	Split3	Split1
DenseNet169	70.8%	69.2%	70.3%	94.2%
DenseNet201	**71.5%**	**70.0%**	**71.1%**	**94.6%**

in Fig. 8. In Fig. 8, three situations (A), (B) and (C) are used for comparative analysis to prove the effectiveness of our AWM. In Fig. 8(A), three sampling groups all give correct predictions, and the results obtained by AWM are similar to those obtained by AVG. In Fig. 8(B), two sampling groups give correct predictions and one sampling group gives wrong predictions. AVG made the correct prediction, and AWM slightly strengthened the significance of the result to AVG.

In Fig. 8(C), one sampling group gives correct predictions, and two sampling groups give wrong predictions. AVG made the wrong prediction, while AWM gave the correct prediction through adaptive weight distribution. It can be seen from the experiment that AWM mainly plays a fine-tuning role. The stronger the heterogeneity of sampling groups, the more obvious the effect of AWM. It also proves that AWM can dynamically assign different weights to different sampling groups.

To assess whether TSCN can generalize well with different model depth or not, the results of TSCN based on DenseNet169 and DenseNet201 are presented as shown in Table 4. Table 4 illustrates that the TSCN with deeper architecture achieves uniformly better results, which verifies

TABLE 5. Comparison with current state-of-the-art methods on UCF101 and HMDB51.

Model	UCF101	HMDB51
IDT [2]	85.9%	57.2%
C3D + IDT [8]	90.4%	-
P3D ResNet [22]	93.7%	-
Two-stream [7]	88.0%	59.4%
Two-Stream TSN (BN-Inception) [15]	94.0%	68.5%
Two-Stream TSN (DenseNet169) [15]	93.3%	68.3%
C^2LSTM [24]	92.8%	61.3%
L^2LSTM [25]	93.6%	66.2%
HR-MSCNN + IDT [13]	94.5%	69.8%
STDDCN [16]	93.8%	66.9%
MLDF-3D [18]	93.5%	68.6%
Two-Stream Heterogeneity [20]	94.4%	67.2%
STRN [26]	93.2%	64.9%
Ours (DenseNet169)	94.2%	70.3%
Ours (DenseNet201)	94.6%	70.9%

the generalization capacity of TSCN in terms of model depth.

D. COMPARISON WITH THE STATE-OF-THE-ART

At last, our model is compared with the state-of-the-art methods on both UCF101 and HMDB51 datasets. The comparison results are summarized in Table 5. Compared with TSN, under the same basic network DenseNet169, our model can at least improve the accuracy by 1.8% on the HMDB51 dataset, and 0.9% on the UCF101 dataset. The superior performance of our model demonstrates that it is necessary to establish feature-level connections between the sampling groups for a global action representation. However, compared to the recently developed models, TSCN does not show a significant advantage. We mainly summarize the reasons in three points: First, TSCN shows advantages on the more heterogeneous datasets, while the exiting datasets have weak heterogeneity. Second, there is no doubt that the establishment of feature-level connections between sampling groups can improve the recognition accuracy, but whether there is a better connection method is still worthy of study. Third, TSCN is developed from TSN, so it is difficult for TSCN to break through the inherent bottleneck of TSN.

V. CONCLUSION

This paper presents a novel Temporal Segment Connection Network (TSCN) for action recognition. Our framework consists of two key ingredients: 1) a Forget-gate Connection Module, which extracts and integrates deep feature from multiple sampling groups to obtain a more global feature representation for actions. 2) an Adaptive Weighting Module which learns to endow different weights for different sampling groups. With this framework, we achieve promising performance on both UCF101 and HMDB51 datasets. However, our model is still not perfect. On the one hand, our model has higher memory requirements; on the other hand, there is still much room for improvement in the connection method of the sampling groups. Future works will explore more effective interaction strategies between sampling groups to improve action recognition.

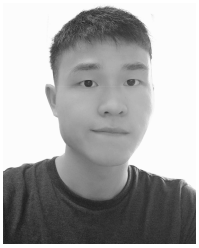
ACKNOWLEDGMENT

The authors thank the editor and reviewers for their work on this manuscript.

REFERENCES

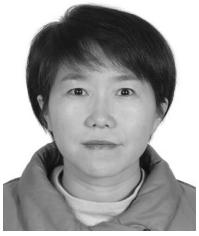
[1] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, nos. 2–3, pp. 107–123, Sep. 2005.
[2] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3551–3558.
[3] L. Wang, Y. Qiao, and X. Tang, "Motionlets: Mid-level 3D parts for human motion recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2674–2681.
[4] L. Wang, Y. Qiao, and X. Tang, "MoFAP: A multi-level representation for action recognition," *Int. J. Comput. Vis.*, vol. 119, no. 3, pp. 254–271, Sep. 2016.
[5] T. de Campos, M. Barnard, K. Mikolajczyk, J. Kittler, F. Yan, W. Christmas, and D. Windridge, "An evaluation of bags-of-words and spatio-temporal shapes for action recognition," in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Jan. 2011, pp. 344–351.
[6] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.
[7] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.
[8] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.
[9] L. Wang, Y. Xu, J. Cheng, H. Xia, J. Yin, and J. Wu, "Human action recognition by learning spatio-temporal features with deep neural networks," *IEEE Access*, vol. 6, pp. 17913–17922, 2018.
[10] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional LSTM with CNN features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018.
[11] Z. Tu, H. Li, D. Zhang, J. Dauwels, B. Li, and J. Yuan, "Action-stage emphasized spatiotemporal VLAD for video action recognition," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2799–2812, Jun. 2019.
[12] A. Piergiovanni and M. S. Ryoo, "Representation flow for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9945–9953.
[13] Z. Tu, W. Xie, Q. Qin, R. Poppe, R. C. Veltkamp, B. Li, and J. Yuan, "Multi-stream CNN: Learning representations based on human-related regions for action recognition," *Pattern Recognit.*, vol. 79, pp. 32–43, Jul. 2018.
[14] B. M. Martinez, D. Modolo, Y. Xiong, and J. Tighe, "Action recognition with spatial-temporal discriminative filter banks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5482–5491.
[15] L. Wang, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, and Y. Xiong, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 20–36.
[16] W. Hao and Z. Zhang, "Spatiotemporal distilled dense-connectivity network for video action recognition," *Pattern Recognit.*, vol. 92, pp. 13–24, Aug. 2019.
[17] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
[18] J. Zhang and H. Hu, "Deep spatiotemporal relation learning with 3D multi-level dense fusion for video action recognition," *IEEE Access*, vol. 7, pp. 15222–15229, 2019.
[19] S. Sun, Z. Kuang, L. Sheng, W. Ouyang, and W. Zhang, "Optical flow guided feature: A fast and robust motion representation for video action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1390–1399.
[20] E. Chen, X. Bai, L. Gao, H. C. Tinega, and Y. Ding, "A spatiotemporal heterogeneous two-stream network for action recognition," *IEEE Access*, vol. 7, pp. 57267–57275, 2019.
[21] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3D residual networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5533–5541.

[22] F. A. Gers and J. Schmidhuber, "Learning to forget: Continual prediction with LSTM," in *Proc. 9th Int. Conf. Artif. Neural Netw. (ICANN)*, Sep. 1999, pp. 850–855.
[23] X. Shi, H. Wang, D. Y. Yeung, W. K. Wong, W. C. Woo, and Z. Chen, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Proc. Syst.*, 2015, pp. 802–810.
[24] M. Majd and R. Safabakhsh, "Correlational convolutional LSTM for human action recognition," *Neurocomputing*, vol. 396, pp. 224–229, Jul. 2020.
[25] L. Sun, K. Jia, K. Chen, D. Y. Yeung, B. E. Shi, and S. Savarese, "Lattice long short-term memory for human action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2147–2156.
[26] Z. Liu and H. Hu, "Spatiotemporal relation networks for video action recognition," *IEEE Access*, vol. 7, pp. 14969–14976, 2019.
[27] V. Choutas, P. Weinzaepfel, J. Revaud, and C. Schmid, "PoTion: Pose MoTion representation for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7024–7033.
[28] H. Kuehne, R. Stiefelhagen, T. Serre, and H. Jhuang, "HMDB51: A large video database for human motion recognition," in *High Performance Computing in Science and Engineering*. Berlin, Germany: Springer, 2013, pp. 571–582.
[29] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*. [Online]. Available: http://arxiv.org/abs/1212.0402
[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.



QIAN LI received the B.S. degree from the Central South University of Forestry and Technology, China, in 2016. He is currently pursuing the M.S. degree in computer science and technology with Hebei University. His current research interests include computer vision and action recognition.

WENZHU YANG received the B.S. and M.S. degrees from Hebei University, China, in 1992 and 2002, and the Ph.D. degree from China Agricultural University, China, in 2010. He is currently a Professor with the School of Cyber Security and Computer, Hebei University. His current research interests include computer vision and smart systems.



XIANGYANG CHEN received the B.S. degree from Yanshan University, China, in 2000, and the M.S. degree from Hebei University, China, in 2007. Her current research interests include computer vision and action recognition.

TONGTONG YUAN received the B.S. degree from Agricultural University of Hebei Province, China, in 2017. She is currently pursuing the M.S. degree in computer science and technology with Hebei University. Her current research interests include computer vision, deep learning, and object tracking.

YUXIA WANG received the B.S. degree from Hebei North University, China, in 2018. She is currently pursuing the M.S. degree in computer technology with Hebei University. Her current research interests include computer vision, deep learning, and object detection.