Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6670087, 15 pages
https://doi.org/10.1155/2021/6670087

Research Article

The Multidimensional Motion Features of Spatial Depth Feature Maps: An Effective Motion Information Representation Method for Video-Based Action Recognition

Hongshi Ou and Jifeng Sun

South China University of Technology, School of Electronic and Information Engineering, No. 381 Wushan Road, Tianhe District, Guangzhou 510641, China

Correspondence should be addressed to Hongshi Ou; ouhongshi@163.com

Received 8 October 2020; Revised 14 December 2020; Accepted 18 January 2021; Published 28 January 2021

Academic Editor: Florin Stoican

In video action recognition based on deep learning, the design of the neural network is focused on how to acquire effective spatial information and motion information quickly. This paper proposes a kind of deep network that can obtain both spatial information and motion information in video classification. It is called MDFs (the multidimensional motion features of deep feature map net). This method can be used to obtain spatial information and motion information in videos only by importing image frame data into a neural network. MDFs originate from the definition of 3D convolution. Multiple 3D convolution kernels with different information focuses are used to act on depth feature maps so as to obtain effective motion information at both spatial and temporal. On the other hand, we split the 3D convolution at space dimension and time dimension, and the spatial network feature map has reduced the dimensions of the original frame image data, which realizes the mitigation of computing resources of the multichannel grouped 3D convolutional network. In order to realize the region weight differentiation of spatial features, a spatial feature weighted pooling layer based on the spatial-temporal motion information guide is introduced to realize the attention to high recognition information. By means of multilevel LSTM, we realize the fusion between global semantic information acquisition and depth features at different levels so that the fully connected layers with rich classification information can provide frame attention mechanism for the spatial information layer. MDFs need only to act on RGB images. Through experiments on three universal experimental datasets of action recognition, UCF10, UCF11, and HMDB51, it is concluded that the MDF network can achieve an accuracy comparable to two streams (RGB and optical flow) that requires the import of both frame data and optical flow data in video classification tasks.

1. Introduction

Video-based action recognition technology has been developed for decades. It is a technology of understanding and classifying video content based on computers. This paper uses the method of deep learning for action recognition. The convolutional neural network (CNN) has a certain degree of translation invariance and scale invariance, and its computing mode is very similar to the visual system of mammals, so CNN has made great achievements in the field of image classification [1–5]. Since a video consists of multiple images, CNN also has diversified applications in the field of video-based action recognition [5–8]. Compared with image classification, the success of video-based action recognition mainly depends on the effective acquisition of temporal information of videos.

In video-based action recognition based on deep learning, the design of the neural network structure is focused on how to acquire temporal information based on spatial information. That means it is necessary to extract not only spatial information of video images but also earlier and later spatial information of the target at time dimension [9] so that neural network can better classify videos. For instance, when classifying brushing teeth and combing hair videos, temporal information can identify the swing of an arm back and forth in both videos. So, temporal information

alone cannot identify the above two kinds of videos. In this case, the difference of spatial information is a perfect identifier for classification and recognition. That is because the motion information of brushing teeth is concentrated on the mouth while the motion information of combing hair is concentrated on the head.

Recently, many researchers have made outstanding achievements in this regard. (1) Entering video frames into CNN to learn to acquire local spatial information [1–3, 10–12]; the disadvantage of this method is that effective motion information cannot be obtained; (2) Fusing the spatial information acquired by CNN with motion information which based on optical flow image extraction [10, 13–15]. Compared with the first method, the second method is more effective. In video action recognition task, neural network-based motion information acquisition mainly includes the following ways. (1) Two-stream CNN structure-based method—RGB images enter the spatial flow CNN to obtain spatial information, while optical flow images enter the temporal flow CNN to obtain local motion information [4, 5, 16]; (2) RGB images enter 3DCNN (3D convolutional neural network) [2] to acquire motion information; and (3) traditional representing methods of motion information [17]: (i) HOG (histogram of oriented gradient) [18]; (ii) IDT (improved dense trajectory) [19]; and (iii) motion vector [20]. Of the above motion information acquisition methods, although 3DCNN does not need to calculate optical flow images, the motion information obtained is not more effective than the two-stream method [2]; the network parameters of 3DCNN are more complicated than 2DCNN, and more computing resources are consumed. Furthermore, the motion information obtained by traditional HOG, IDT, motion vector, and so on is not so effective as by optical flow, either. The introduction of optical flow images is an effective approach for the neural network to obtain effective motion information. What we focus on is that without obtaining the optical flow information of the video frame in advance, the motion information equivalent to the optical flow features can be obtained only through end-to-end training of the neural network.

How to obtain motion information quickly and effectively? The acquisition of motion information should consider not only effectiveness but also computing economy. For deep learning-based motion information acquisition, the design of the neural network can introduce the guidance on the effective motion information acquisition method. In this paper, a motion information acquisition network based on spatial depth features is proposed, which has fast computing speed and can effectively obtain motion information of videos.

As shown in Figure 1, this paper proposes a method of obtaining multiscale motion information on spatial depth feature maps. Because the motion information features are acquired based on spatial depth feature maps, the method is named MDF (the multidimensional motion features of deep feature map). MDFs are obtained on different temporal and spatial scales based on the spatial flow CNN depth feature map. In the process of MDF acquisition, the motion information acquisition network module only acts on the pixels of the spatial flow depth feature map, which makes the network end-to-end trainable. Also, the network-level feature map receives dimensionality reduction by order of magnitudes on the basis of the original image, so the computational complexity of the MDF network is acceptable. Moreover, the introduction of MDF makes it unnecessary for the network to obtain motion information through optical flow images of the video and instead directly acts on the spatial depth of RGB image to obtain motion information.

2. Related Work

Video classification based on CNN deep learning can be divided into three types: (1) structure; (2) inputs; and (3) connection. Structure pays attention to network structure; inputs refer to the type and format of input data as well as related operations of data enhancement; connection mainly refers to the interaction of spatial-temporal information in the two-stream network.

Structure: This video classification method is focused on the design of network structure. An effective network structure is designed according to the spatial-temporal information features of a video to obtain spatial-temporal information. The current mainstream structures are almost derived from two-stream convolutional networks [4] and C3D [2]. Lan et al. [21] proposed an action recognition method based on local features. Firstly, a two-stream structure and video tag were used to extract a short video clip and import it into the network for training. The pretrained two-stream network was used as the module of local spatial-temporal feature extraction of the whole video. After obtaining the local spatial-temporal information of each part, the local information of each part was aggregated to realize video-based action recognition; Diba et al. [22] proposed DTLENs (deep temporal linear encoding networks) for video-based action recognition. The network adopted a two-stream structure. Three frames were input into spatial flow CNN, and the corresponding three frames of optical flow images were input into temporal flow CNN. Both spatial flow and temporal flow were encoded with a TLE (temporal linear encoding) layer to realize aggregation of local information and finally action recognition; Wang et al. [23] proposed TSN (temporal segment network), a novel framework for video-based action recognition which is based on the idea of long-range temporal structure modeling. It combines a sparse temporal sampling strategy and video-level supervision to enable efficient and effective learning using the whole action video; Feichtenhofer et al. [24] introduced a residual connection into the two-stream structure to realize the classification of early detailed information. The temporal flow is initialized using the pretraining network of temporal flow, and temporal flow spatial information also interacts with the temporal flow layer. Such structure realizes the full fusion of spatial information and temporal information; Zhu

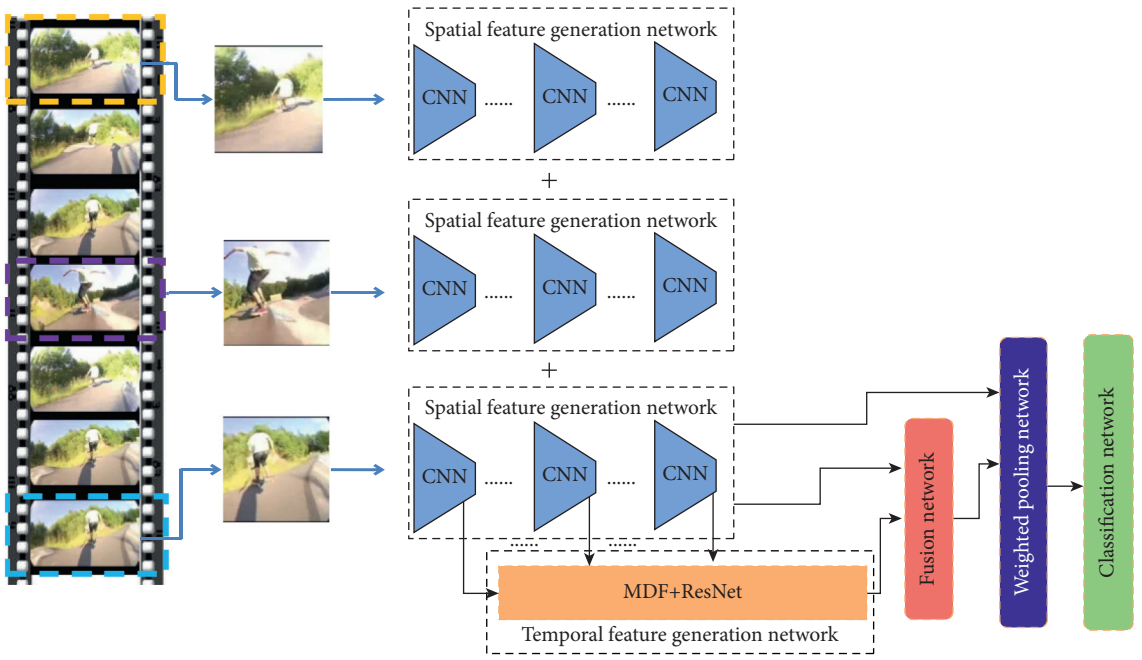

Figure 1: Network architecture overview.

et al. [16] used a convolution network to fuse the spatial flow depth features and the temporal flow depth features on the basis of TSN [23]. The fused information is used for the weighted pooling of spatial information and temporal information of TSN for final classification. This method realizes the weight distinction of spatial information and temporal information; Xie et al. [25] proposed T3D (temporal 3D ConvNet), which uses 3D convolution to acquire the temporal information of a video and meanwhile introduces the temporal transition layer (TTL) to acquire the temporal information at different temporal depths. Besides, a pretrained 2D network is introduced to achieve stable initialization of T3D through knowledge transfer; Zhou et al. [26] proposed the temporal relation network (TRN). Firstly, the feature maps of 12 frames are acquired using CNN, and the related information between random 2 frames, 3 frames, and 4 frames, respectively, among the 12 frames using correlation function is acquired. Then, the aggregate function is used to integrate the related information to realize classification recognition. This kind of network realizes the temporal information of multiscale time; Zhou et al. [27] used CNN to obtain multiframe depth information and then used the depth feature information of adjacent 3 frames to obtain motion information with correlation function: projection, distance, subtraction, multiplication, submul Relu, and so on. After obtaining the local spatial-temporal information, the aggregate function is used to realize the integration of local information. The integration functions contain average pooling, RNN and CNN, and the optimal local spatial-temporal information integration function is obtained by experiment.

Inputs: In the two-stream network, the input format of spatial network channel is usually a single RGB image or stack of RGB images. The spatial network is generally a fine-tune product of a classical network on ImageNet. While more and more attention is paid to motion information in recent years, some people criticize action recognition for overdependence on background and appearance characteristics and lack of modeling of motion itself. However, in fact, motion is neither a noun nor a verb, but a verb + noun form, e.g., play + basketball or play + football. Therefore, we believe that although more attention should be paid to temporal information, the important role of spatial characteristics is undeniable. The spatial network is mainly for capturing important object features in video frames. At present, most public datasets can only rely on a single frame to complete the classification of a video, and normally segmentation is unnecessary. In this case, the input of the spatial network generates considerable redundancy and may introduce additional noise. Can key video frames be extracted to improve the classification level? Zhu et al. [28] proposed KVMDF (key volume mining deep framework), a network for extracting key frames which borrows the idea of RCNN and introduces the key frame selection network. The key frame selection network needs to act on the whole video, which is an obvious disadvantage. To solve the shortcoming of the above method, Kar et al. [29] proposed AdaScan (adaptive scan pooling in deep convolutional neural networks) to attention on key frames while ignoring redundant frames at the time of convolution and pooling. This network structure realizes a lightweight network. However, the recognition accuracy is lower than that of KVMDF.

Connection: One is the interaction between layers within a single network, such as ResNet and Inception; the other is the interaction between two-stream networks, including explorations of different fusion ways. At present, a considerable practice is the connection of two-stream networks referring to the structure of ResNet. Feichtenhofer et al. [30] used residual structure for both spatial flow and temporal flow in upper structure and increased the interaction between temporal flow and spatial flow. They offered the experimental results of multiple interactive modes; according to Wang et al. [31], the key to action recognition lies in how to properly fuse spatial and temporal features. They found the following. Although the traditional two-stream network has a fusion process in the end, for those that are indeed independently trained, the prediction error of the final result tends to come from either network, and the spatial/temporal network has its strong points. This paper analyzes the causes of classification error; it is easier for the spatial network to make mistakes when the video backgrounds have a high similarity, while it is easier for the temporal network to make mistakes because of the limitation of snippets length in long-term actions. So, is it possible to make the two networks complement each other through interaction? In the aspect of interaction, this paper introduces an STCB module into the network, which fuses spatial information and temporal information while retaining independent spatial flow and temporal flow. Finally, the three channels are fused to produce the ultimate result.

The input of the two-stream network is video frame data and video frame optical flow information, so it is necessary to work out optical flow information first. In order that network can achieve video-based action recognition only by acting on video frame data, this paper proposes a motion information extraction method based on spatial depth features and introduces spatial information weighted pooling based on spatial-temporal information to realize an attention mechanism for spatial information, that is, to realize weight-based attention of effective spatial information on the basis of the effective fusion of spatial information and temporal information in the network.

3. Approach

For the action recognition neural network architecture proposed by this paper, a pretrained two-dimensional convolutional network (2DCNN) is used to extract the spatial information of video frames; a three-dimensional convolutional network (3DCNN) is used to act on deep spatial features to obtain temporal information (motion information). Local features of spatial CNN and local temporal information are fused at the pixel level to obtain local spatial-temporal information, and local spatial-temporal information is utilized to realize attention on effective spatial information. Finally, a multilevel LSTM is used to obtain the global spatial-temporal information and a spatial information attention mechanism given by semantic information to the network. Section 3.1 discusses the selection of CNN; Section 3.2 introduces the motion information acquisition method based on spatial depth features; Section 3.3 discusses the pixel level fusion method of local spatial information and motion information adopted in this paper; Section 3.4 discusses the weighted pooling method based on fusion information; Section 3.5 discusses the global representation method of the video in the action recognition model; Section 3.6 gives the overall network architecture; Section 3.7 describes the details of network implementation—network hyperparameters setting.

3.1. Convolutional Neural Network (CNN) Transfer Learning Implementation. Motivated by the literature [32], this paper uses the method of transfer learning to solve the problem of limited datasets. CNN is used for extracting the features of video frames in our network structure. Inspired by the literature [18, 33–36], we realized transfer learning in this way: the initial state of CNN was a VGG-16 model that had been pretrained with ImageNet, and its structure is shown in Figure 2. The VGG-16 model contains 13 convolutional layers and 3 fully connected layers. In Figure 2, a convolutional layer is represented with "conv<depth><number_of_channels>"; a fully connected layer is represented with "FC-<number_of_channels>."

3.2. The Multidimensional Motion Features of Depth Feature Maps. In order not to calculate the optical flow vector fields of videos in advance and improve real-time performance, we made 3DCNN act on spatial depth features to obtain motion information. In addition, spatial depth features had realized the compression of the original image data size, which further reduced the calculated quantity of motion information acquisition. 3DDCNN can effectively obtain local motion information but meanwhile increases network parameters and computation. In order to save computing resources, this paper decomposes 3DCNN into spatial 2DCNN and temporal 1DCNN. The corresponding convolution kernels are decomposed as follows:

$$\mathrm{3DConv}[3\times 3\times 3] \underline{\underline{\mathrm{def}}} \ \mathrm{3DConv}[3\times 3\times 1] + \mathrm{3DConv}[1\times 1\times 3], \tag{1}$$

where the computational complexity of $\mathrm{3DConv}[3\times 3\times 3]$ is $O(n) = n^3$ and the computational complexity of $\mathrm{3DConv}[3\times 3\times 1] + \mathrm{3DConv}[1\times 1\times 3]$ is $O(n) = n^2 + n$, which means the computational complexity is reduced by an exponential order. Moreover, in order to obtain multidimensional motion information, in the MD3DCNN (multidimensional 3DCNN) proposed in this paper, the convolution kernel of each channel uses different initialization to realize the acquisition of spatial and temporal motion information. The specific network structure is shown in Figure 3. In the MD3DCNN network module, the number of output channels of all convolutional modules is equal to the number of input channels.

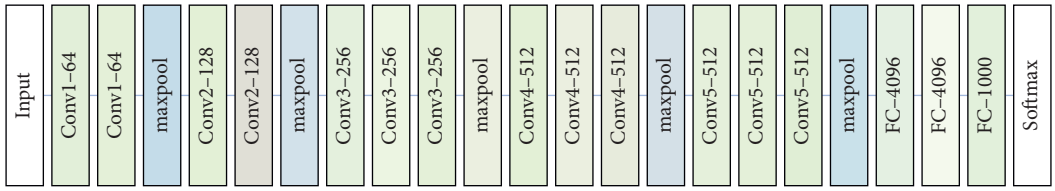

Figure 2: VGG-16 network.

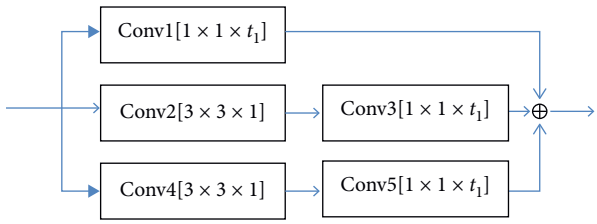

Figure 3: MD3DCNN module—3D convolutional layer for multidimensional motion information acquisition.

The initialization of each MD3DCNN convolution kernel is as follows. The tensor form of conv1 is $1 \times 1 \times 3 = [0, -1, 1]$. That means calculating the difference between adjacent frames to extract the motion information between frames; the tensor form of Conv2 is $3 \times 3 \times 1 = [[-1, 0, 1], [-1, 0, 1], [-1, 0, 1]]$. That means calculating the horizontal gradient of spatial; the tensor form of Conv3 is $1 \times 1 \times 3 = [0, 0, 1]$. Conv2 and Conv3 are cascaded to extract the horizontal gradient of a frame, that is, the horizontal motion information; the tensor form of Conv4 is $3 \times 3 \times 1 = [[-1, -1, 1], [0, 0, 0], [1, 1, 1]]$. That means calculating the spatial vertical gradient; the tensor form of Conv5 is $1 \times 1 \times 3 = [0, 0, 1]$. Conv4 and Conv5 are cascaded to extract the vertical gradient of a frame, that is, the vertical motion information.

The motion information acquisition network of each depth layer based on spatial depth features is shown in Figure 4. f_* is video frame. When the spatial depth feature map was above 128 channels, Conv 1 × 1 2DCNN was used to reduce the dimension of the spatial depth feature map to 128 channels. MD3DCNN acted on the spatial depth feature map to extract the motion information of adjacent frames at time dimension and multiple spatial dimensions in different depth layers.

3.3. The Architecture for Fusing the Spatial Information and Temporal Information. The motion information extraction network module based on spatial depth features is shown in Figure 5, and 5 ResNet modules form one 18-layer residual network (Resnet18). The output channels of all residual network modules are 128.

One of the main disadvantages of the spatial information and temporal information fusion structure mentioned in the literature [4] is that it is fused only in recognizable layers (FC layers); the spatial and temporal features at pixel level cannot be obtained by training and learning. For example, for actions with similar motion information like brushing teeth and combing hair, since a hand moves back and forth in spatial periodically in both videos, without fusion of motion

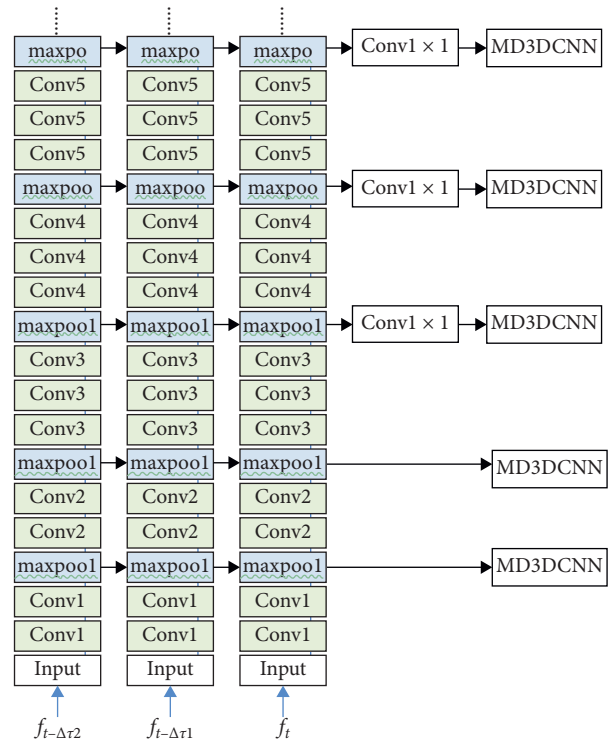

Figure 4: Motion information extraction network based on spatial depth feature.

information with spatial information at the same position at pixel level, the network cannot obtain the specific spatial target information (teeth or hair) corresponding to the motion information, thus resulting in failure to confirm whether it is brushing teeth or combing hair.

Since the motion information extracted based on spatial features can identify the above periodic motion and the spatial information can identify the position information of the motion (located in the tooth or the hair), the fusion feature of motion information and spatial information can identify whether to brush or comb the hair. For this reason, the fusion of feature maps on each channel is the corresponding fusion of pixels at the same position in the two-stream network fusion. If the motion information extraction network and the spatial information extraction network have the same network structure, the fusion method of the same spatial location on the feature map is very easy to implement; for example, it can simply overlay or stack the same location. However, the fusion layer of each network (spatial information extraction network and temporal information extraction network) has multiple channels, so

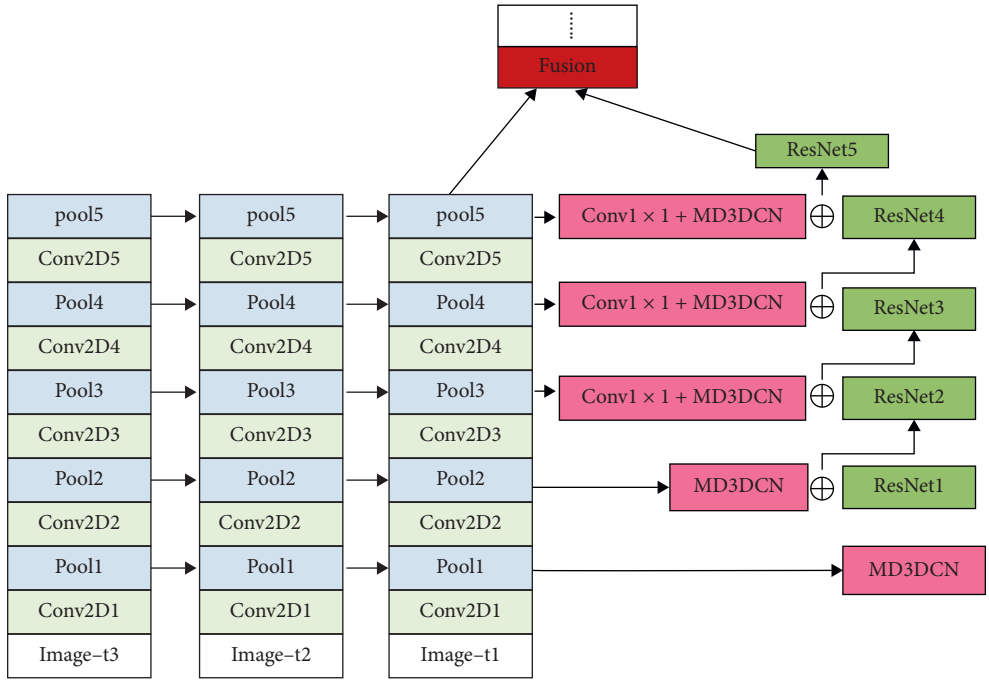

Figure 5: Spatial-temporal information fusion method.

how to determine the corresponding relationship between channels between different networks is the key problem.

It is assumed that different channels of the spatial network are responsible for extracting the features of different regions in the spatial, while different channels of the temporal network are responsible for extracting the motion features of different regions. Then, how to determine the corresponding relationships between channels of different networks is the key problem. Therefore, the fusion method must have the ability to learn to obtain the corresponding relationship between a channel of the spatial network and a channel of the temporal network so as to better distinguish different actions.

Based on the above considerations, we choose a spatial-temporal information fusion method based on convolution operation [37]. The mapping relation of feature fusion is $y_t = f(x_t^a, x_t^b)$, where "f" is the fusion function; the spatial network feature map is $x_t^a \in \mathbb{R}^{H\times W\times D}$; the temporal network feature map is $x_b^a \in \mathbb{R}^{H\times W\times D}$ and the output feature map after fusion is $y^t \in \mathbb{R}^{H'\times W'\times D'}$. In the above formulas, "t" stands for time (in the following text, t is removed from each formula since the operation is the same at each moment). W represents the width of the feature map; H represents the height of the feature map; D represents the number of channels on the fusion layer. And we have $H = H' = H''$, $W = W' = W''$, and $D = D'$. First, we cascade and stack the feature maps on the fusion layer; that is, $y^{\text{cat}} = f^{\text{cat}}(x^a, x^b)$. The specific stacking method is as follows:

$$y_{i,j,2d}^{\text{cat}} = x_{i,j,d}^a \cdot y_{i,j,2d-1}^{\text{cat}} = x_{i,j,d}^b, \tag{2}$$

where i and j are the spatial position of the feature map, d is the channel tab, and we have $y \in \mathbb{R}^{H\times W\times 2D}$. Convolutional fusion like formula (3) was carried out on this basis, which gave the network the ability to obtain the corresponding relationship between channels of the spatial information network and motion information network through learning.

$$y^{\text{conv}} = y^{\text{cat}} * f + b, \tag{3}$$

where f is a set of filters, and we have $f \in \mathbb{R}^{n\times m\times 2D\times D}$, $b \in \mathbb{R}^D$. The dimension of the filter is $n \times m \times 2D$; the number of channels output after convolutional fusion is D. The above set of filters was used to reduce the dimensionality of the number of channels while playing the role of fusing feature maps x^a and x^b at the same spatial position.

Another key problem of feature fusion is which layer features are fused on. In this paper, the spatial-temporal information fusion method shown in Figure 5 is adopted. This method only carries out fusion on the last convolutional layer with rich spatial information, which can reduce the network parameters. Based on later verification, this fusion method can achieve good recognition accuracy.

3.4. Weighted Pooling Based on Spatial-Temporal Information. The attention model based on spatial-temporal information is a variant of the attention mechanism in a multimodal task [38–40]. We introduced this method into spatial-temporal information attention scenarios and then used the fusion information of motion information and spatial information to locate effective information regions on the spatial information feature map. Figure 6 shows a spatial motion clue attention mechanism network based on spatial-temporal information we proposed. We realized spatial information attention based on spatial-temporal information on the final convolutional layer (conv5 in VGGnet) of the spatial flow network because, on the one

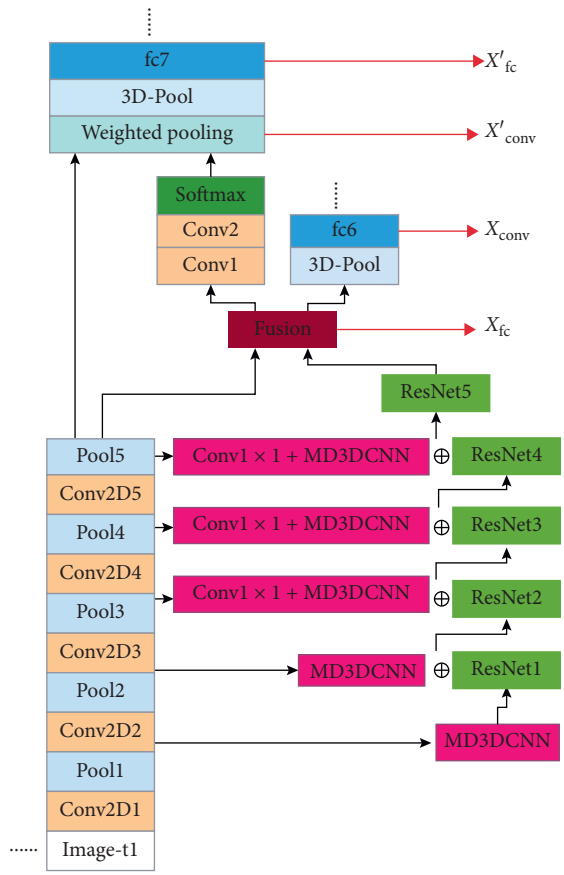

Figure 6: A spatial motion clue attention mechanism network based on spatial-temporal information.

hand, the final convolutional layer contains information that can be used for image classification, while lower-level convolutional layers contain fine grit information of the image, such as edge information, corner information, and texture information. On the other hand, that is because spatial-temporal information acts directly on the recognizable feature layer and realizes the weighted pooling of effective classification information. Such weighted pooling is more effective than maximum pooling of the recognizable information layer to obtain recognizable classification information and meanwhile avoids information loss caused by average pooling.

Attention pooling replaced the last maximum pooling layer and the first fully connected layer in VCCnet. A $7 \times 7 \times 512$ feature map was reduced to $1 \times 1 \times 512$ through attention pooling, which not only realized attention to spatial information but also reduced network parameters. Attention pooling, unlike maximum pooling or average pooling, has statistical logic and complicated mechanism because the attention pooling process is supervised by spatial-temporal information. And 3DCNN spatial-temporal fusion aggregates spatial information and corresponding motion information, which means spatial-temporal information can represent both appearance and motion clues. Two layers of the convolutional network and one layer of the softmax normalization layer were attached to the spatial-temporal information to obtain the spatial weight coefficient. The output feature map of Conv1, the first convolutional layer, is $64 \times 7 \times 7$; the output feature map of Conv2, the second convolution layer, is $1 \times 7 \times 7$. Then, the attention weight coefficient was given by softmax normalization. Based on spatial-temporal information, it became easier for spatial flow to abstract the moving object from still RGB images.

Our attention mechanism is similar to the attention mechanisms introduced into video classification by the literature [34], but paper has the following differences from others: (1) attention is given by the fusion of spatial information and temporal information based on depth features, which makes the network more able to obtain motion clues from spatial information; (2) motion information based on deep spatial features is introduced to supervise the attention module on the time span.

3.5. Long-Term Feature Presentation Methods in Video-Based Action Recognition

3.5.1. LSTM Network. The neural network needs to obtain the interrelations of a sequence of frame images in a video for action classification. In this paper, long short-term memory (LSTM) is used to extract the relational features of video sequence signals. An LSTM cycle network with layer normalization function is used to ensure the convergence of training. That is, the parameters of each input sigmoid activation function are normalized as follows:

$$\begin{aligned} \mu_\beta &\longleftarrow \frac{1}{m}\sum x_i, \\ \sigma_\beta^2 &\longleftarrow \frac{1}{m}\sum \left(x_i - \mu_\beta\right)^2, \\ \widehat{x}_i &\longleftarrow \frac{x_i - \mu_\beta}{\sqrt{\sigma_\beta^2 + \varepsilon}}, \\ y_i &\longleftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i). \end{aligned} \tag{4}$$

In the above four formulas, m is the number of samples used in a single training; x_i is No. i sample value in a single training; and y_i is the sample value of input sigmoid activation function after normalization.

3.5.2. CNN and LSTM Combination Method for Action Recognition. We used the combination method shown in Figure 7. The specific network feature layers of X_{conv}, X_{fc}, X'_{conv}, and X'_{fc} are shown in Figure 5. X_{conv} is the feature map after motion information and spatial information based on spatial depth features received 3DCNN fusion on the last convolutional layer. X_{fc} is the first fully connected layer of the branch subnetwork following the fusion layer; X'_{conv} is the feature map after weighted pooling was carried out to the last layer of spatial information; X'_{fc} is the first fully connected layer of the subnetwork following the weighted pooling layer.

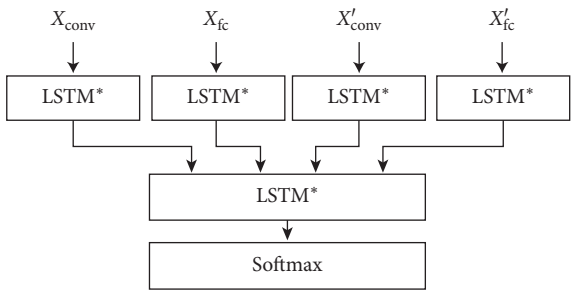

Figure 7: CNN and LSTM combination network.

The LSTM* network in Figure 7 is shown in Figure 8(a). It is a cycle network with temporal sequence signals as both input and output. The LSTM, as shown in Figure 8(b), is a traditional long short-term memory, that is, sequence input and single hidden layer unit output.

In our proposed multilayer long short-term memory, an LSTM of sequence input and sequence output () was firstly used to achieve the representation of hidden states of the video; then, the hidden states of LSTM* passed an LSTM of sequence input and a single hidden unit output to realize the representation of video sequence through a single hidden unit. Such a design realized the hierarchical acquisition of information. In addition, end-to-end backpropagation realized exchange of information between the spatial feature flow of convolutional layers (X_{conv}/X_{conv}' – LSTM*) and the semantic information flow of fully connected layers (X_{fc}/X_{fc}' – LSTM*). After the exchange between semantic information and spatial information was realized, the video frame regional attention mechanism of the network was realized, and meanwhile the convergence process of backpropagation training was improved.

The network model equation is shown as follows. First, on the first layer, the features of convolutional layers and fully connected layers fused by LSTM* enter an LSTM under sequence input + sequence output work pattern to realize the sequence hidden state representation of the video:

$$\begin{aligned} h_{conv}^{i,t} &= \text{LSTM}^*\left(x_{conv}^{i,t}, h_{conv}^{i,t-1}\right), \\ h_{conv}^{i} &= \left[h_{conv}^{i,1}, h_{conv}^{i,2}, \ldots, h_{conv}^{i,T}\right], \\ h_{fc}^{i,t} &= \text{LSTM}^*\left(x_{fc}^{i,t}, h_{fc}^{i,t-1}\right), \\ h_{fc}^{i} &= \left[h_{fc}^{i,1}, h_{fc}^{i,2}, \ldots, h_{fc}^{i,T}\right]. \end{aligned} \tag{5}$$

On the first layer, the sequence hidden state output by LSTM* is again input into an LSTM under sequence input + sequence output work pattern:

$$\begin{aligned} h^{i} &= \text{LSTM}\left(W\left[h_{conv}^{i}, h_{fc}^{i}\right]\right), \\ y^{i} &= \text{softmax}\left(h^{i}\right). \end{aligned} \tag{6}$$

3.6. Proposed Architecture. Based on the descriptions in sections 3.1–3.5, the network architecture in Figure 9 can be given. After the last convolutional layers (after ReLU output) of the spatial network and the temporal network received

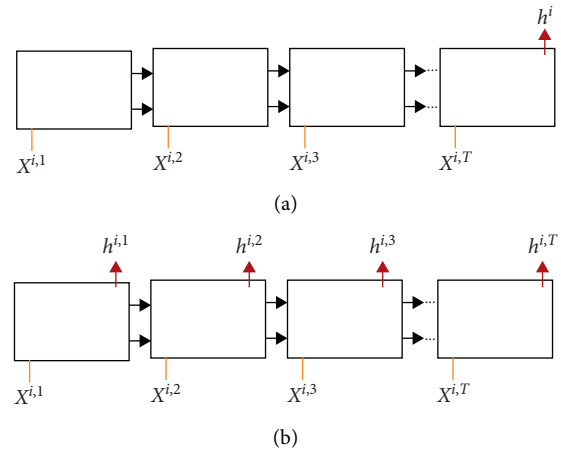

Figure 8: Two types of LSTM networks used in the proposed network: (a) sequence-to-one LSTM; (b) sequence-to-sequence LSTM.

3DCNN fusion, the postfusion features were output and then sent into LSTM*. The features output by fusion received C3D and then entered another LSTM*. A two-layer CNN acted on the spatial-temporal information to realize weighted pooling of spatial information based on spatial-temporal information. And the same weighted pooling spatial features entered another LSTM*. The weighted pooling features received C3D before entering another LSTM*. Lastly, an LSTM with sequence input and single hidden layer unit output was used to fuse the sequence features output by the abovementioned four LSTM*. And the last layer of the network is a softmax classification layer.

To reduce the network requirement for computing resources, we did not send all video frames into the network to achieve action classification. Instead, we did it this way: a fixed number ($T=17$) of frames were sampled from each video and sent into the network. As shown in Figure 9, the sampling time of the video frames was $t, t+\tau, \ldots, t+T\tau$, respectively.

3.7. Implementation Details. The CNN we used for spatial features extraction is a VGG-16 network model. The VCC-16 model contains 13 convolutional layers and 3 fully connected layers. In the network structure shown in Figure 9, the initial state of each convolutional layer of the spatial network is a VGG-16 model pretrained with ImageNet. From each video, we extracted $T(17)$ frames at equal intervals and sent them into the network. The frame number "T" has direct impact on the validity of semantic information obtained by the network. If the number of frames is too small, the network cannot reach the correct rate given by Table 1; too many frames may reduce the recognition efficiency. The size of each video frame input into the network is 224×224.

Like the fusion structure described in Section 3.3, the dimension of the 3D convolution kernel "f" used in spatial-temporal information fusion is $3 \times 3 \times 3 \times 640 \times 512$. That is, the dimension of spatial-temporal filter is H''

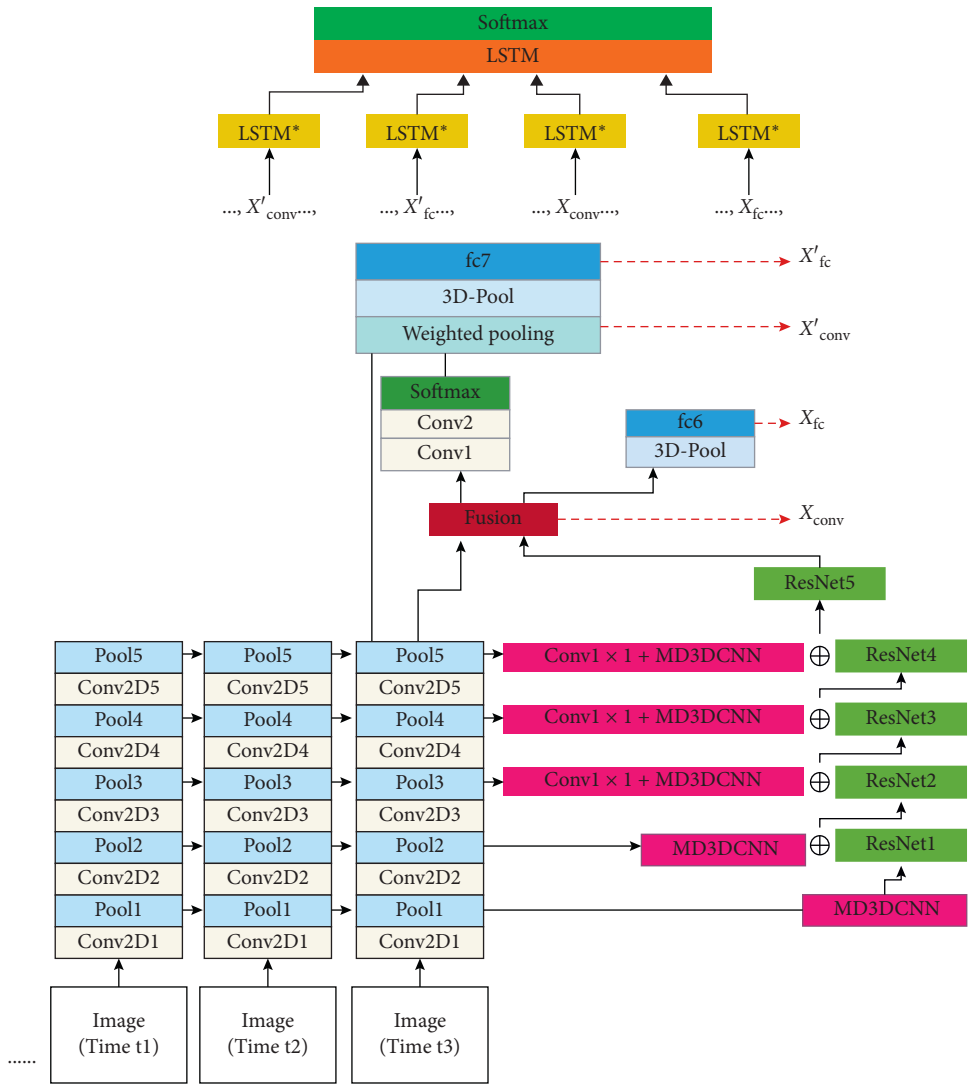

Figure 9: Structure of the multidimensional motion features of depth feature maps (MDFs).

Table 1: Experimental results on the effectiveness and necessity of MD3DCNN and 4LSTM*.

UCF101	
Method	Accuracy (%)
MDF:SD3DCNN+4LSTM*	86.2
MDF:MD3DCNN+3LSTM*	89.1
MDF:MD3DCNN+4LSTM*	96.2

$\times W'' \times T'' = 3 \times 3 \times 3$; $D = 640$ is the number of channels after the feature maps of spatial CNN, and motion information extraction CNN was stacked on the last convolutional layer (after ReLu output); $D' = 512$ is equal to the number of input channels of Fc6 and Conv1 in the next layer of the network; the hidden state dimensions of all LSTM networks used in this paper are 101. The tensor forms of X_{conv} and X_{fc} input into LSTM* in Figure 9 are, respectively, $X_{\text{conv}} = (512 \times 7 \times 7)$and $X_{fc} = (2048 \times 1)$; the tensor forms of X'_{conv} and X'_{fc} are, respectively, $X'_{\text{conv}} = (512 \times 7 \times 7)$ and $X'_{fc} = (2048 \times 1)$.

In the design of this paper, there is no need for step-by-step training. That is, the adopted network training mode is end-to-end training. The first fully connected layer after fusion in the network (Fc6 in Figure 9) reached dropout ratios of 0.15 in training. The dropout ratios of the first fully connected layer after weighted pooling (Fc7 in Figure 9) were 0 in training. The dropout ratios of LSTM* input and output were both 0; the dropout ratios of LSTM input and output were both 0.25. The experimental results suggest that if the dropout ratios of Fc6 were greater than 0.15 or the output dropout ratios of LSTM were greater than 0.25, the convergence rate of the network would be reduced, but the recognition accuracy would not be improved; the learning rate of network learning and training was 10^{-6}. The value of learning rate has a direct impact on whether network training can converge. The experimental results show that when the learning rate was greater than 10^{-4}, the network could not converge.

For the multidimensional motion features of deep feature map net (MDFs) proposed in this paper, the number of

video frames that need to be fed into the network during training is big (17 frames). In order to avoid overfitting and strengthen the generalization ability of the network, we used the method of increasing training datasets to overcome overfitting. That is, the following random image frames acquisition process was added. Upon the premise of sampling T frames, the start frame was randomly generated, and the sampling interval τ was random selected within [2, 10]. Meanwhile, the range of the sampling interval τ has an effect on the semantic information obtained by the network. If the interval is too big, local semantic information will be missing; if too small, the whole semantic information will be missing; the loss function used for the training of the MDF in this paper is a cross-entropy loss function.

4. Experiments

We trained and verified our network using three publicly available datasets: (1) UCF101, UCF11, and HMDB51. It is very challenging to recognize the video actions from the above three datasets because the videos have varied light conditions, camera states (mobile or fixed in the shooting process), and background complexities.

4.1. Datasets

(1) UCF101 contains 13,320 videos of 101 categories [41]

(2) UCF11 contains 1,600 videos of 11 action categories: shooting at the basket, cycling, diving, golf swing, horse riding, bouncing a football, swing, flapping tennis, trampolining, volleyball spike, and dog walking

(3) HMDB51 contains 6,765 videos of 51 action categories [15]

4.2. Experimental Procedures. The datasets have different number of frames for their videos with the minimum frame number. In order to make the network universal to each dataset, we sent a few frames into the network, which is 17 frames. That is, the number of frames sent into the network $T = 17$. During testing, assuming that N is the total number of frames of a video sequence, the acquisition interval "t" is as follows:

$$t = \frac{N}{T}. \tag{7}$$

If the integer value of rounded t is t', the video frame sequence acquired "S" is as follows:

$$S = [1 \times t', 2 \times t', \ldots, T \times t']. \tag{8}$$

In this paper, the video frames "T", respectively, sampled from the three datasets were all 17 frames. In the experiment, the data of the video datasets were randomly divided into "training set" and "test set." The training set was used to train the network model, while the test set was used to test and verify the recognition accuracy after the network training. The ratio of the training set to the test set was 7:3.

4.3. Experimental Results and Comparative Analysis. MD3DCNN is used for local motion information extraction based on spatial depth features. To verify the necessity of multidimensional 3DCNN, we compared it with SD3DCNN (single-dimensional 3DCNN). That is, in the MD3DCNN module shown in Figure 3, only the correct recognition rates of Conv1 submodule and MD3DCNN were retained. The correct recognition rates of both are shown in Table 1. Table 1 shows that the correct recognition rate of the single-channel network was lower than that of the multichannel network when verified using UCF101; it was unnecessary for both to use 4 LSTM* in the network structure as shown in Figure 9. We used UCF101 to verify the correct recognition rate. When any LSTM* was subtracted (4 LSTM* changed to 3 LSTM*), the correct recognition rate was reduced, and the specific data are shown in Table 1. Moreover, in Table 1, "SD3DCNN+4LSTM*" and "MD3DCNN+3LSTM*" had lower training convergence rates than "MD3DCNN+4LSTM*." The specific experimental results are shown in Figure 10.

Figure 10(a) is a relation curve graph between the loss function and the frequency of training obtained by training MDF:MD3DCNN+4LSTM* with UCF101. According to Figure 10(a), the MDF network training proposed by this paper could converge quickly. From this, we can judge that the MDF designed by us underwent no vanishing gradient or explosion. Figure 10(b) is a relation curve graph between the correct recognition rate and the frequency of training obtained by training MDF:MD3DCNN+4LSTM* with UCF101. According to Figure 9(b), the correct recognition rate of UCF101 tended to be stable with rising frequency of training (the specific verification of the correct recognition rate is shown in Table 2); Figure 9(c) is a relation curve graph between the loss function and the frequency of training obtained by training MDF:MD3DCNN+3LSTM* with UCF101. According to Figure 10(c), the proposed MDF: MD3DCNN+3LSTM* could converge in network training. Figure 10(d) is a relation curve graph between the correct recognition rate and the frequency of training obtained by training MDF:MD3DCNN+3LSTM* with UCF101. According to Figures 10(b) and 10(d), MDF: MD3DCNN+3LSTM* had a lower correct recognition rate than MDF:MD3DCNN+4LSTM*.

Figure 11(a) is a relation curve graph between the loss function and the frequency of training obtained by training MDF:MD3DCNN+4LSTM* with HMDB51. According to Figure 11(a), the MDF proposed by this paper could converge quickly in network training.

From this, we can judge that the MDF designed by us underwent no vanishing gradient or explosion. Figure 11(b) is a relation curve graph between the correct recognition rate and the frequency of training obtained by training MDF: MD3DCNN+4LSTM* with HMDB51. According to Figure 11(b), the correct recognition rate of HMDB51 tended to stabilize with the rising frequency of training (the specific verification of the correct recognition rate is shown in Table 2).

On the top of using our proposed method to verify action recognition on UCF101 and HMDB51, other methods were

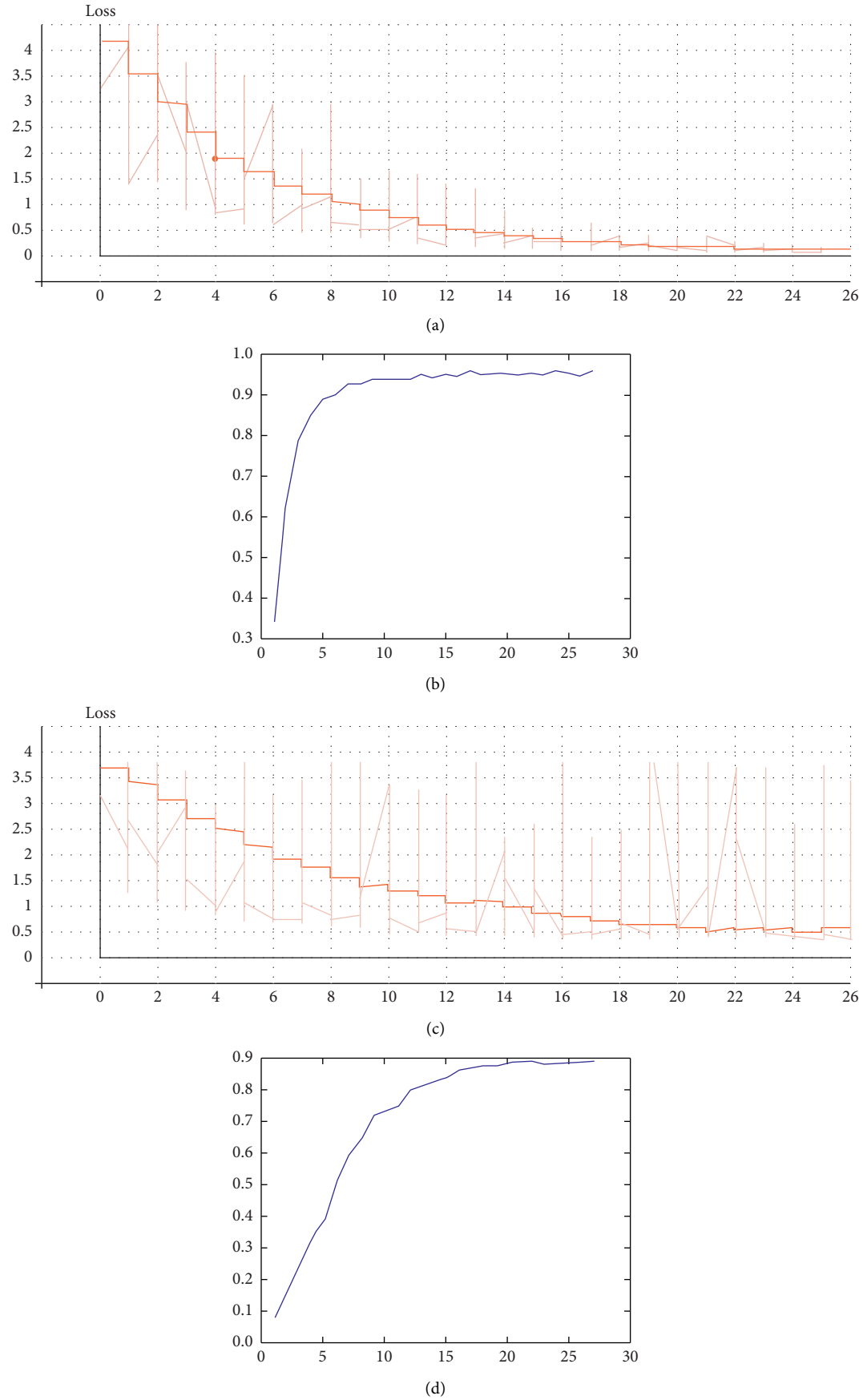

Figure 10: (a, b) The loss function curve and the correct recognition rate curve of MDF:MD3DCNN+4LSTM* trained with UCF101, respectively; (c, d) The loss function curve and the correct recognition rate curve of MDF:MD3DCNN+3LSTM* trained with UCF101, respectively.

Table 2: Comparison of our results to the state-of-the-arts on action recognition datasets UCF101 and HMDB51.

Method	UCF101 (%)	HMDB51 (%)
LRCN [13]	82.9	
Dense trajectories [42]	84.2	
Composite LSTM model [33]	84.3	
Soft attention [34]	84.9	39.87
Two-stream ConvNet [4]	88.0	59.4
C3D [2]	90.4	—
TDD [43]	91.5	65.9
Action transformations [44]	92.4	62.0
TSN [23]	94	68.5
DOVF [21]	94.9	71.7
TLE [22]	95.6	71.1
Ours (MDF:MD3DCNN+4LSTM*)	96.2	`65.5

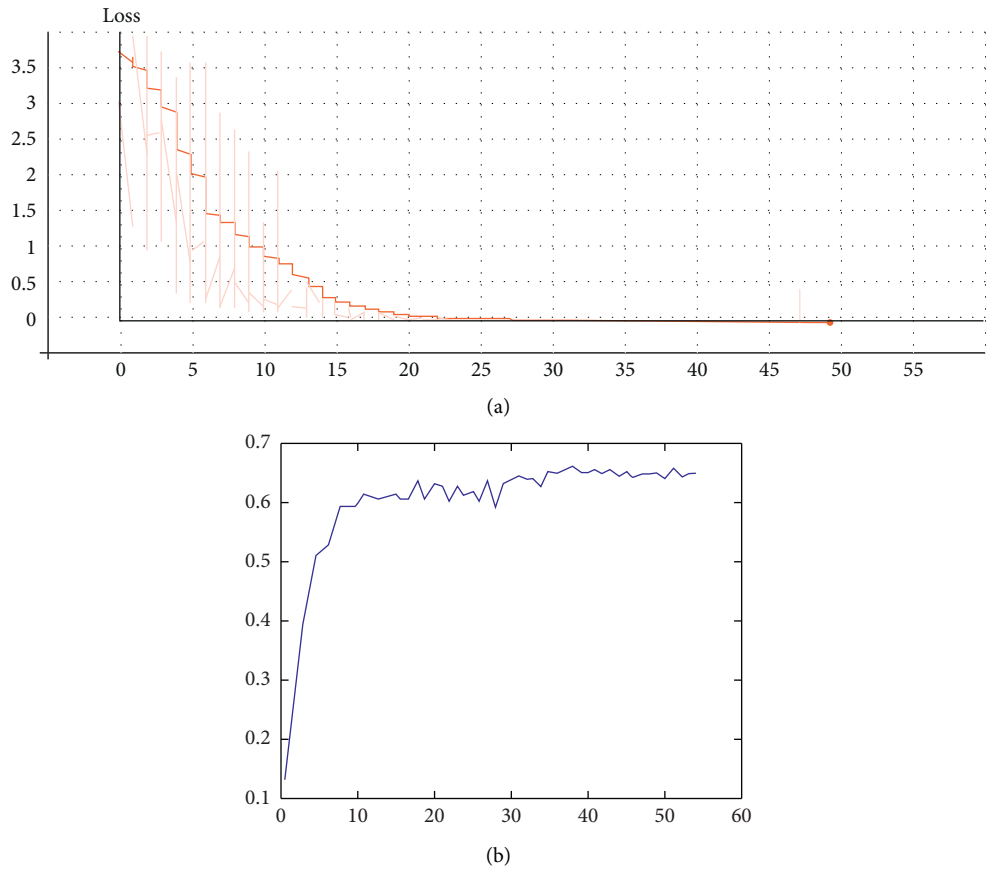

Figure 11: (a, b) The loss function curve and the correct recognition rate curve of MDF:MD3DCNN+4LSTM* trained with HMDB51, respectively.

also used to recognize and classify videos of UCF101 and HMDB51. The experimental results of each method are shown in Table 2.

There are no two-stream network structure and no optical flow information feature: LRCN [13], Dense LSTM Model [42], Composite LSTM Model [33], Soft attention [34], in which the accuracy of video action recognition is low.

Both C3D [2] and two-stream ConvNet [4] require simultaneous input of video frame data and optical flow data. And both face the problem of limited time-dimensional information; the whole process of TDD [43] is as follows: the trajectory-constrained sample and pooling strategy are adopted to aggregate convolutional features to form descriptor. Fisher vector is used to aggregate these local TDDs into a global long vector and then classified by SVM. TDD is actually the fusion of all kinds of hand-craft features (HOG, HOF, and MBH); action transformations [44], which use a linear system to describe the dynamic change of the high-level visual information of an action, also adopt two-stream

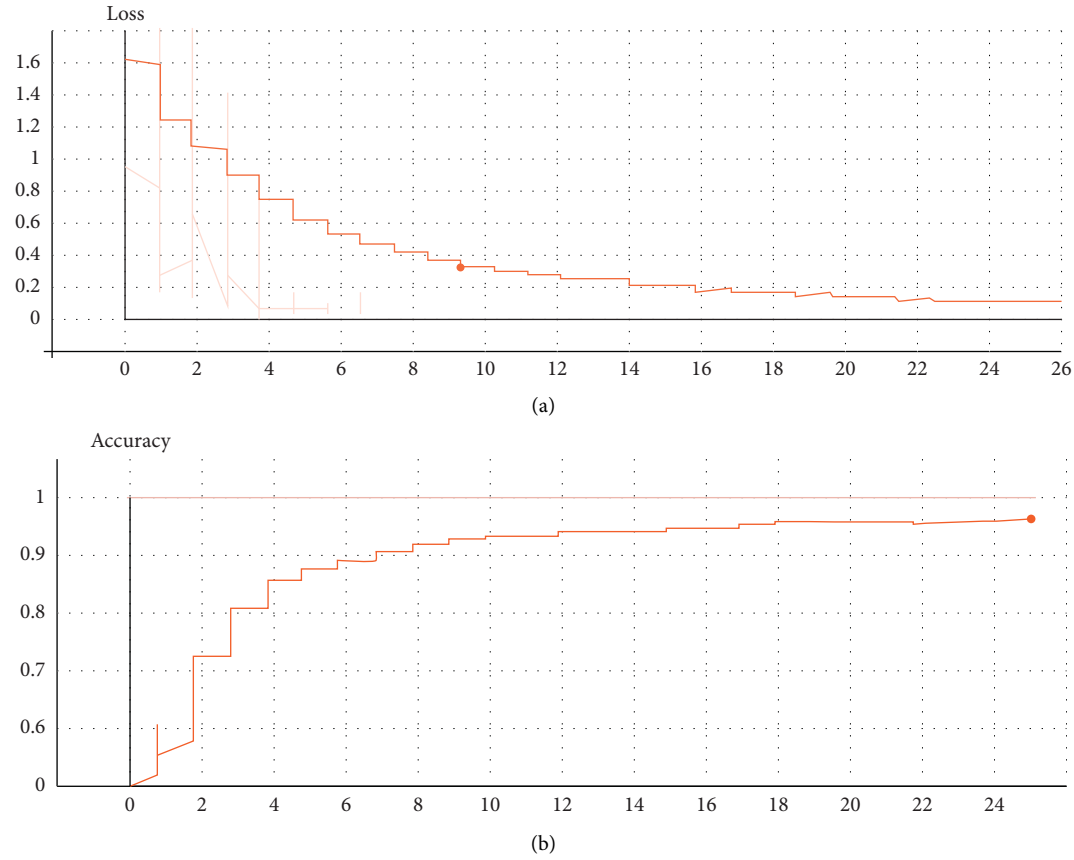

Figure 12: (a, (b) The loss function curve and the correct recognition rate curve of MDF:MD3DCNN+4LSTM* trained with UCF11, respectively.

network structure, but only video frames are sent into the two-stream structure. This paper uses a linear system to learn the transformation from embedding latent representation of precondition state to embedding latent representation of effect state. Therefore, it can be assumed that this method only considers the motion information on the time dimension of adjacent frames. Once the motion information changes slowly and the adjacent frames cannot give the motion information, it will not be recognized; TSNs (temporal segment networks) [23] inherit the structure of the two-stream network. In order to solve the long-term problem, the author proposes to use multiple two-stream networks to capture the short-term information at different positions of the temporal sequence and then fuses them to obtain the final result. The limitation of the literature [23] is that it is only about the fusion of multiple local features; DOVF [21] is also based on two-stream structure. It is an improved version of TSN. The improvement mainly lies in the fusion part—different segments are assigned different weights. And this part is obtained by network learning, and the final result is achieved by SVM classification; TLE [22] uses Tempora linear encoding to, respectively, encode the spatial information and motion information of 3 adjacent frames in two-stream network structure and finally fuses the encoding results. The network only acts on a small number of video frames; based on the characteristics of the abovementioned two-stream methods and the analysis and introduction of the network architecture proposed by this paper, our method does not use two-stream structure, but its effect is comparable to two-stream networks and better than non-two-stream networks. It can be concluded that the proposed method can effectively extract the spatial information and motion information of video frames, thus effectively realizing video-based action recognition.

Figure 12(a) is a relation curve graph between the loss function and the frequency of training obtained by training MDF:MD3DCNN+4LSTM* with UCF11. By comparing Figures 10(a) and 12(a), it is found that the training convergence was faster when the proposed MDF network architecture was applied to datasets with fewer samples. Figure 12(b) is a relation curve graph between the correct recognition rate and the epoch of training obtained by training MDF:MD3DCNN+4LSTM* with UCF11. According to Figure 12(b), the correct recognition rate about UCF11 tended to be stable with the increase in the epoch of training. In the case of limited training samples and verification sets, the correct rate was 97.7%, which indicates that our network has good generalization ability.

Similarly, on the top of using our proposed method to verify action recognition on UCF11, other methods were also used to recognize and classify videos of UCF11. The experimental results of each method are shown in Table 3.

Table 3: Comparison of our results to the state-of-the-arts on action recognition dataset UCF11.

UCF11	
Method	Accuracy (%)
Dense trajectories [42]	84.2
Soft attention [34]	84.9
Cho et al. [35]	88.0
Snippets [36]	89.5
Ours (MDF:MD3DCNN+4LSTM*	97.7

Cho et al. [44] used the traditional multicore sparse representation method to represent the local motion features and global motion features of videos. This method needs to learn the representation of local motion features in the obtained dictionary. This method is limited by the representation ability of the dictionary; the snippets method uses CNN to extract spatial features as the key frames extraction tool and then uses the following SVM to carry out category recognition to key frames. This kind of method lacks complete time dimension information; compared with other algorithms, in the case of limited datasets, our proposed method has a higher correct recognition rate and better generalization ability.

5. Conclusion

We proposed an MDF network for motion information extraction based on spatial depth features, which can be applied to video-based action recognition. In the case of limited training samples, the spatial information of video frames was fully obtained using a pretrained CNN. The network extracted motion information based on spatial depth features. Local spatial information and local motion information were fused to obtain local spatial-temporal information, and local spatial-temporal information was used to realize attention to effective spatial information. Finally, a 4-channel LSTM was used to realize the fusion of local spatial-temporal information and finally extracted the global representation of the video action. Through experimental comparison and verification on UCF101, HMDB51, and UCF11 datasets, our proposed method is comparable to the current two-stream network architecture in terms of recognition accuracy. This proves the effectiveness of the following four mechanisms in the proposed network architecture: (1) motion information extraction based on spatial depth features, (2) fusion of local spatial information and corresponding motion information, (3) focus on spatial effective information based on spatio-temporal fusion information, and (4) the ability of 4-channel LSTM and LSTM cascaded structure to correlate local spatial-temporal information to achieve long-term representation of the video action.

Data Availability

The data used to support the findings of this study are available in UCF101 (https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/) and HMDB (https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank Professor Sun Jifeng for his guidance. This work was supported by the National Natural Science Foundation of China, no. 62071183.

References

[1] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1725–1732, Columbus, OH, USA, June 2014.

[2] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, December 2015.

[3] Z. Liu, C. Zhang, and Y. Tian, "3D-based Deep Convolutional Neural Network for action recognition with depth sequences," *Image and Vision Computing*, vol. 55, pp. 93–100, 2016.

[4] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proceedings of the Proceedings of the Advances in Neural Information Processing Systems*, Montreal, Canada, December 2014.

[5] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao, "Towards good practices for very deep two-stream ConvNets," 2015, https://arxiv.org/abs/1507.02159.

[6] A. Diba, A. M. Pazandeh, and L. V. Gool, "Efficient two-stream motion and appearance 3D CNNs for video classification," 2016, https://arxiv.org/abs/1608.08851.

[7] Yi Zhu, Z. Lan, S. Newsam, and A. G. Hauptmann, "Hidden two-stream convolutional networks for action recognition," 2017, https://arxiv.org/abs/1704.00389.

[8] H. Chen, J. Chen, R. Hu, C. Chen, and Z. Wang, "Action recognition with temporal scale-invariant deep learning framework," *China Communications*, vol. 14, no. 2, pp. 163–172, 2017.

[9] C. Y. Wu, C. Feichtenhofer, H. Fan et al., "Long-term feature banks for detailed video understanding," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, November 2019.

[10] C.-Y. Ma, M.-H. Chen, Z. Kira, and G. AlRegib1, "TS-LSTM and temporal-inception: exploiting spatiotemporal dynamics for activity recognition," 2017, https://arxiv.org/abs/1703.10667.

[11] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: a unified embedding for face recognition and clustering," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, Boston, MA, USA, June 2015.

[12] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 7, p. 8, Boston, MA, USA, 2015.

[13] J. Donahue, L. Anne Hendricks, S. Guadarrama et al., "Long-term recurrent convolutional networks for visual recognition

and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2625–2634, Boston, MA, USA, June 2015.

[14] M. Xin, H. Zhang, H. Wang, M. Sun, and Y. Ding, "ARCH: a daptive recurrent-convolutional hybrid networks for long-term action recognition," *Neurocomputing*, vol. 178, pp. 87–102, 2016.

[15] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops*, Barcelona, Spain, November 2011.

[16] J. Zhu, W. Zou, and Z. Zhu, "Learning gating ConvNet for two-stream based methods in action recognition," 2017, https://arxiv.org/abs/1709.03655.

[17] Z. Fan, L. Shao, J. Xie, and Yi Fang, "From handcrafted to learned representations for human action recognition: a survey," *Image and Vision Computing*, vol. 55, pp. 42–52, 2016.

[18] A. Klaser, M. Marszalek, and C. Schmid, "A spatio-temporal descriptor based on 3D-gradients," in *Proceedings of the 19th British Machine Vision Conference*, Leeds, UK, September 2008.

[19] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3551–3558, Sydney, Australia, December 2013.

[20] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with enhanced motion vector CNNs," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2718–2726, Las Vegas, NV, USA, June 2016.

[21] Z. Lan, Y. Zhu, and A. G. Hauptmann, "Deep local video feature for action recognition," 2017, https://arxiv.org/abs/1701.07368.

[22] A. Diba, V. Sharma, and L. Van Gool, "Deep temporal linear encoding networks," 2016, https://arxiv.org/abs/1611.06678.

[23] L. Wang, Y. Xiong, Z. Wang et al., "Temporal segment networks: towards good practices for deep action recognition," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, Netherlands, October 2016.

[24] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal residual networks for video action recognition," 2017, https://arxiv.org/abs/1611.02155.

[25] S. Xie, C. Sun, J. Huang et al., "Rethinking spatiotemporal feature learning: speed-accuracy trade-offs in video classification," 2017, https://arxiv.org/abs/1712.04851.

[26] B. Zhou, A. Andonian, and A. Torralba, "Temporal relational reasoning in videos," 2018, https://arxiv.org/abs/1711.08496.

[27] Y. Zhou, J. Ren, J. Li et al., "Video classification via relational feature encoding networks," 2017.

[28] W. Zhu, J. Hu, G. Sun et al., "A key volume mining deep framework for action recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Las Vegas, NV, USA, June 2016.

[29] A. Kar, N. Rai, K. Sikka, and G. Sharma, "AdaScan: adaptive scan pooling in deep convolutional neural networks for human action recognition in videos," 2017, https://arxiv.org/abs/1611.08240.

[30] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal multiplier networks for video action recognition," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Honolulu, HI, USA, July 2017.

[31] Y. Wang, M. Long, J. Wang et al., "Spatiotemporal pyramid network for video action recognition," 2019, https://arxiv.org/abs/1903.01038.

[32] D. Ghadiyaram, M. Feiszli, D. Tran et al., "Large-scale weakly-supervised pre-training for video action recognition," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Long Beach, CA, USA, November 2019.

[33] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised learning of video representations using LSTMs," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 2, p. 8, Lille, France, July 2015.

[34] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," 2015, https://arxiv.org/abs/1511.04119.

[35] J. Cho, M. Lee, H. J. Chang, and S. Oh, "Robust action recognition using local motion and group sparsity," *Pattern Recognition*, vol. 47, no. 5, pp. 1813–1825, 2014.

[36] M. Ravanbakhsh, H. Mousavi, M. Rastegari, V. Murino, and L. S. Davis, "Action recognition with image based CNN features," 2015, https://arxiv.org/abs/1512.03980.

[37] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," 2016, https://arxiv.org/abs/1604.06573.

[38] K. Xu, J. Ba, R. Kiros et al., "Show, attend and tell: neu-ral image caption generation with visual attention," in *Proceedings of the International Conference on Machine Learning (ICML)*, Lille, France, July 2015.

[39] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," 2015, https://arxiv.org/abs/1511.02274.

[40] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, Barcelona, Spain, December 2016.

[41] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: a dataset of 101 human actions classes from videos in the wild," 2012, https://arxiv.org/abs/1212.0402.

[42] H. Wang, A. KlŁser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3169–3176, IEEE Computer Society, Colorado Springs, CO, USA, June 2011.

[43] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2015.

[44] X. Wang, A. Farhadi, and A. Gupta, "Actions transformations," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.