Article

TBRNet: Two-Stream BiLSTM Residual Network for Video Action Recognition

Xiao Wu ^{1,2} and Qingge Ji ^{1,2,*}

¹ School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China; wuxiao28@mail2.sysu.edu.cn

² Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou 510006, China

* Correspondence: issjqg@mail.sysu.edu.cn

Received: 10 June 2020; Accepted: 11 July 2020; Published: 15 July 2020

Abstract: Modeling spatiotemporal representations is one of the most essential yet challenging issues in video action recognition. Existing methods lack the capacity to accurately model either the correlations between spatial and temporal features or the global temporal dependencies. Inspired by the two-stream network for video action recognition, we propose an encoder–decoder framework named Two-Stream Bidirectional Long Short-Term Memory (LSTM) Residual Network (TBRNet) which takes advantage of the interaction between spatiotemporal representations and global temporal dependencies. In the encoding phase, the two-stream architecture, based on the proposed Residual Convolutional 3D (Res-C3D) network, extracts features with residual connections inserted between the two pathways, and then the features are fused to become the short-term spatiotemporal features of the encoder. In the decoding phase, those short-term spatiotemporal features are first fed into a temporal attention-based bidirectional LSTM (BiLSTM) network to obtain long-term bidirectional attention-pooling dependencies. Subsequently, those temporal dependencies are integrated with short-term spatiotemporal features to obtain global spatiotemporal relationships. On two benchmark datasets, UCF101 and HMDB51, we verified the effectiveness of our proposed TBRNet by a series of experiments, and it achieved competitive or even better results compared with existing state-of-the-art approaches.

Keywords: action recognition; bidirectional long short-term memory; residual connection; temporal attention mechanism; two-stream networks

1. Introduction

With the rapid development of the mobile Internet and the continuous updating of video capture devices, the number of video resources is growing explosively. It is essential to find effective methods to analyze these videos intelligently. Therefore, video action recognition [1–6] has become a hotspot in the field of computer vision and has been widely applied in various fields such as surveillance, human–machine interfaces, and sports analysis. The goal of this task is to analyze the action in a video and determine its category. However, many factors in videos, such as background clutter, camera motion, and illumination changes, make this task more challenging. Hence, it is essential to model the appropriate spatiotemporal representations for this task as well as other video-based tasks. Unfortunately, when modeling the spatiotemporal representations of videos, there are still several specific challenges that need to be resolved.

One of the key challenges is accurately modeling the correlations between spatial and temporal features. Inspired by the considerable progress made in convolutional neural networks (CNNs), which are capable of automatically extracting hierarchical features from static images [7–10], some methods [11,12] simply apply CNNs to the video action recognition task. However,

these approaches do not notably surpass the performance of traditional approaches that use hand-crafted features [13–17], and the multiple kernel learning algorithm [18] with the aggregation of several hand-crafted temporal descriptors. These CNN-based approaches only capture the appearance features while ignoring the motion cues, which are confirmed to be significant for modeling effective spatiotemporal representations of videos. In some recent work that used the two-stream architecture [2,19], besides taking advantage of CNNs to extract the appearance features of videos, these approaches construct another CNN pathway to model motion information with the help of optical flow and then obtain the final results by averaging the prediction scores of the two streams. However, when observing cases that were misclassified, we noticed that, sometimes, the results of one stream were right while those of the other one were wrong. This usually happens when videos have similar backgrounds (Figure 1a) or motion patterns within short snippets (Figure 1b). Therefore, instead of only fusing the final predictions of two pathways, it is important to model more accurate correlations of spatial and temporal features and make them reinforce each other. In our work, we argue that constructing connections between the two pathways is helpful for this issue.

Figure 1. Examples of videos that have similar patterns in one pathway which will cause misclassification in the original two-stream network: (a) the “BreastStroke” versus the “FrontCrawl”, where they both have similar swimming pools as backgrounds though they are different swimming actions; (b) “PullUps” versus “RopeClimbing”, where they have similar motion patterns of raising the body in short snippets but different backgrounds.

The other key challenge was how to effectively capture the global temporal dependencies. Rank pooling [20] generates a fast video-level descriptor named “dynamic image” which is designed to summarize arbitrary temporal length of frames. Thus, a single dynamic image contains both the appearance and motion information of the entire video. In addition, in Reference [20], the authors point out that the long-term dynamics and temporal patterns are important cues for the recognition of actions. In a short fixed-length temporal window, by extending 2D convolution kernels to 3D, 3D-CNNs [4,12,21] are able to preserve temporal relationships and learn good spatiotemporal representations of consecutive frames for action classification. For the variable lengths of temporal windows with a long range, long short-term memory (LSTM) networks, which have been demonstrated to be powerful in sequential modeling tasks, are suitable for abstracting temporal dependencies among video segments. For example, existing approaches [5,6] use LSTM coupled with CNN features and are notably effective for long-term temporal perceptions. Therefore, we take both the short fixed-length temporal cues from 3D-CNNs and long-range temporal cues from LSTM into consideration to extract the overall information. Moreover, there is still room for improvement in comprehensively exploiting the global temporal structure of videos. By observing videos in datasets, such as UCF101 [22] and HMDB51 [23], we notice that there are usually some frames that are less relevant to the action, while some frames are salient and discriminative, as indicated by the frames outlined in red in Figure 2. Rather than treating every part of the video equally, it is important to comprehend the temporal context and automatically direct different levels of concentration to different parts. In our work, we applied the temporal attention mechanism to temporal sequences, which proved to be beneficial, according to our experiments.

Figure 2. The change in frames in a “basketball dunk” over time. The yellow circles mark the changes in action, and several discriminative frames are outlined in red.

In this paper, we introduce an encoder–decoder framework named Two-Stream BiLSTM Residual Network (TBRNet) for video action recognition. More specifically, in the encoding phase, in contrast to the original two-stream network that extracts appearance and motion features separately, our proposed two-stream encoder consisted of a spatial appearance stream and a temporal motion stream with multiplicative residual connections inserted between the two pathways. These connections affect the gradients and make the hierarchical spatiotemporal features interact earlier during processing. Moreover, the two pathways were constructed by proposed 3D CNN named Res-C3D network, which has 3D convolution kernels with residual blocks. Then, the spatial appearance features and temporal motion features from the fully connected layers of each pathway were fused together as short-term spatiotemporal representations of the encoder. In the decoding phase, the fused representations at every time step were sent to the temporal attention-based BiLSTM network which models the temporal dependencies for sequences in both the forward and backward directions and concentrates more on salient information. After that, the temporal dependencies were integrated with short-term spatiotemporal features by residual connections and then sent to another LSTM network, followed by a softmax classifier for the final prediction of the video action. Ablation experiments on two video action recognition datasets, named UCF101 and HMDB51, proved the effectiveness of our approaches. Our proposed model not only showed significant improvements over baseline models but also outperformed some state-of-the-art methods. To summarize, the contributions of this work are as follows:

- We accurately modeled the interactions between spatial and temporal features using proposed a two-stream encoder with cross-stream residual connections which were also benefit for backpropagation of gradients;
- We effectively captured the global spatiotemporal dependencies by incorporating the local features in a fixed-length window extracted by proposed Res-C3D and the long-term relationships among entire sequences extracted by proposed attention-based BiLSTM;
- We proposed a new model with encoder–decoder framework named TBRNet for video action recognition. Extensive experiments on two benchmark datasets named UCF101 and HMDB51 show the effectiveness of our proposed model, and it achieves competitive or even better results compared with some existing state-of-the-art approaches.

The rest of the paper is organized as follows: In Section 2, related works of ours are briefly reviewed. In Section 3, we introduce our proposed framework TBRNet. Introduction of datasets and implement details, ablation study and the comparison with state-of-the-art approaches are discussed in Section 4. Section 5 concludes the paper.

2. Related Works

2.1. Video Action Recognition

Before the dramatic success achieved by CNNs, approaches based on hand-crafted features have gained initial progress, nevertheless most of them only suitable for non-realistic actions performed in simple backgrounds. Those approaches utilize the feature detectors such as 3D-SIFT [13] to represent videos in the form of Histogram of Oriented Gradient (HOG) [14], Histogram of Optical Flow (HOF) [16], Motion Boundary Histogram (MBH) [15] or dense trajectories [24], and the local representations are taken into consideration using a spatio-temporal pyramid [25] with Bag-of-Words (BoW) [16] or Fisher

vector-based encoding [17] to feed into classifiers. Improved dense trajectories (iDT) [17], which uses Fisher vector as an encoder and SVM as a classifier, becomes one of the most significant approaches based on them. Saleh et al. [18] aggregated the temporal coherent descriptors, such as HOG, HOF, MBH and fisher vectors, into a multiple kernel learning algorithm which computes the optimal kernel and parameters from a large set of kernels to reduce the bias.

Over several recent years, as CNNs have thrived, researchers tended to introduce them into video action recognition [2,11,12]. Karpathy et al. [11] proposed a method to extract the spatial feature of each frame by using a pre-trained 2D CNN, and then fuse the spatial features of continuous frames in the last stage to obtain the classification result. However, the results turned out to be less satisfying, since the model was unable to effectively extract temporal features. Inspired by the two-stream networks proposed by Simonyan et al. [2], many two-stream-based approaches, such as seen in Reference [19], were introduced which capture appearance features in spatial stream using RGB frames and motion features in temporal stream using optical flow frames, where each stream processes individually. With a simple but effective extension from 2D convolution to 3D convolution [12], 3D CNNs can encode the spatial and temporal information simultaneously and show promising results. However, the main deficiency of those 3D CNN methods [4,12,21,26] is the lack of the ability for modeling long-term temporal dependencies, since they only capture motion features from fixed-number frames. Fernando et al. [20] point out that the long-term temporal patterns are important for action recognition. They encode the video using dynamic images which are designed to summarize a series of frames on the video level.

Recurrent Neural Network (RNN), especially LSTM, attract much concern, since they are effective for handling temporal sequence encoding tasks. LSTMs show improvements when cooperating with features extracted by CNNs [1,5,6,27,28]. Long-term recurrent convolutional network (LRCN) [6] is an end-to-end framework that classifies the action in video sequences using LSTM with features abstracted by CNN. Varol et al. [28] utilize a longer temporal pooling window with 3D convolution kernels. Ullah et al. [1] use bidirectional deep LSTM instead of single-layer LSTM to analyze a certain time interval feature sequence.

2.2. Attention Mechanism

An attention mechanism is first introduced into image perception tasks, such as image classification, object detection, and image segmentation [29–31], which simulate the human visual system to focus on discriminative regions of an image and quickly glance at other regions. Besides the different importance of regions inside an image, there are always some redundant and irrelevant frames to the action category inside a video; thus, it is natural to employ an attention mechanism into the video-based tasks [32–36]. Kar et al. [32] propose a two-stream-based network with dynamic attention pooling of the video frames. Korbar et al. [33] propose to sample a set of salient clips of video by calculating the scores of all the clips. Sharma et al. [35] apply the soft attention model to RNN to concentrate on important frames in frame sequences. Yu et al. [36] propose a joint spatial–temporal attention model which combines the spatial convolution (2D) for the spatial attention cues and extra temporal convolution (1D) for temporal attention cues to obtain high-level representations of video. Sang et al. [34] propose a two-level attention model with the attention model in both the region level and frame level. By employing the attention mechanism, the results usually perform better as we proved by experiment in our own work.

2.3. Residual Learning

With the increasing level of network depth, it becomes harder to train a deep network owing to the gradient vanishing problem. The deep residual network proposed by He et al. [37] can reach a depth of hundreds of layers by building residual blocks, and this can effectively maintain the network's performances while learning extra deep features. Wang et al. [31] employ the attention modules in residual blocks for image classification. However, they do not take the temporal information into

consideration, which has a non-trivial influence for video tasks. Feichtenhofer et al. [38,39] propose a two-stream-based network with a residual connection between two pathways in order to model the spatiotemporal features during processing. Qiu et al. [40] propose Pseudo-3D Residual Network which splits the 3D convolution kernel ($3 \times 3 \times 3$) into a spatial convolution kernel ($1 \times 3 \times 3$) and a temporal convolution kernel ($3 \times 1 \times 1$) by residual connections; thus, it leads to a largely reduction in the computation cost. In our work, we employed the residual learning in three places of the whole model to obtain accurate global spatiotemporal features for video.

3. Proposed Approach

In this section, we describe our proposed TBRNet and its main components. The TBRNet can be divided into two modules: the two-stream CNN-based encoder and the BiLSTM-based decoder. The overall architecture of TBRNet is shown in Figure 3. Specifically, residual learning, as well as its key idea, is described in the first subsection, since it is applied to three places in our model. Then, we introduce the proposed Res-C3D model which is used to construct the two streams. After that, several types of interaction between the two pathways are discussed. Subsequently, we describe the decoder modeled with the BiLSTM network. Finally, the attention mechanism for BiLSTM and the integration of overall global spatiotemporal representations using residual connections are described in the last subsection.

Figure 3. The overall architecture of our proposed encoder–decoder model Two-Stream BiLSTM Residual Network (TBRNet) for video action recognition. First, we divided a video into N segments over time and sample t RGB frames and corresponding t optical flow frames for each segment with a fixed-time interval. Then, those stacked frames were fed into the encoder with two-stream networks to extract visual features, where each network had residual connections from the motion stream to the appearance stream for earlier interactions. Each stream was constructed using the proposed Residual Convolutional 3D (Res-C3D). In addition, “A-” and “M-” stand for the appearance stream and motion stream, respectively. After that, the short-term spatiotemporal features from the encoder at every time step were fed into attention-based BiLSTM to obtain the bidirectional temporal dependencies, and the “F-” and “B-” stand for the forward and backward directions of LSTM, respectively. Those temporal dependencies were integrated with the short-term spatiotemporal features in another merged LSTM (M-LSTM) for global spatiotemporal representations. Finally, a softmax layer was applied to obtain the final result of classification.

3.1. Residual Network

Evidence [7,41] has revealed that with the increasing depth of a CNN, more hierarchical features can be integrated from deep networks, which translates to a more powerful ability to obtain semantic information. However, only increasing the number of network layers leads to the problem of vanishing gradients, which hampers the convergence from the start and ultimately results in the degradation of accuracy. To avoid this problem, we built our network by taking advantage of the idea of shortcut connections. Instead of using a “highway network” [42,43], weighted residual terms are replaced with identity mapping. As shown in Figure 4, through the “shortcut” skipping connections, the input signal can be directly propagated to any later layer of the network as the output.

$$H(x) = F(x, \{W_i\}) + x, \quad (1)$$

where x is the input vector of the considered layer, $H(x)$ is the desired underlying mapping, W_i denotes the convolution filter weights matrix, and the function $F(x)$ stands for the nonlinear residual mapping needed to learn. If we want to fit the underlying mapping $H(x)$ by a few stacked layers, we let these layers approximate another mapping $F(x) := H(x) - x$, and then the original mapping is recast into $F(x) + x$, which is easier to learn and better for network optimization.

Figure 4. Structure of the residual block, which has a shortcut from the input layer directly connected to later layers.

By introducing skipping connections, the residual learning has the advantages of not only maintaining the features from former layers while learning new information but also propagating gradients from the loss layer directly to any former layers during backpropagation. Thus, the problem of degradation as the network depth increases can be mitigated.

3.2. Two-Stream Interaction

In the original two-stream architecture, a video is represented as an RGB frame and a stack of optical flow frames, which are separately fed into the appearance stream and motion stream. Each of the two processing paths generates the spatial and temporal features on its own to obtain the softmax predictions which are averaged in the last step to calculate the final results of classification. Since this architecture extracts the appearance and motion features in parallel and has no interaction before the final fusion, it is incapable of modeling the subtle spatiotemporal cues, which are of significant importance for distinguishing videos that have similar appearances or motion patterns.

In our work, a video was first represented as N small segments, where N also corresponds to the time steps of the BiLSTM decoder. For each segment, we sampled t RGB frames and the corresponding t optical flow frames. One stack of RGB frames and optical flow frames covering one local video segment were fed into the appearance and motion streams, respectively, which had the same CNN structure for modeling the correlations of appearance and motion features between the same abstract levels of two streams.

Several types of cross-stream connection [39] can be introduced between the identical abstraction levels of the two pathways for an earlier interaction as depicted in Figure 5. However, the performances of structures with a simple direct connection, such as those shown in Figure 5a,b, are inferior to others, which is imputed to the huge changes in distribution directly induced by the other stream.

Figure 5. Several types of cross-stream connections for capturing spatial and temporal features simultaneously. (a–d) The connections of direction from motion to the appearance stream; (e) the bidirectional connections.

As depicted in Figure 5c, the additive connections injected from the motion stream to the appearance stream can be formalized as:

$$x_{l+1}^a = f(x_l^a) + F(x_l^a + f(x_l^m), W_l^a) \quad (2)$$

Here, x_l^a and x_l^m stand for the inputs of the l th layer of the appearance and motion streams, respectively, x_{l+1}^a stands for the inputs of the $(l+1)$ th layer of appearance stream, and W_l^a denotes the weight matrix of this unit in the appearance stream. $f(x)$ denotes ReLU activation function, and $F(x)$ denotes the operators of convolutional layers and ReLU activation layers shown in above figures. Then, during the backpropagation, given the inputs of the $(l+1)$ th layer of motion stream x_{l+1}^m , the gradient of the loss function L of the appearance stream $\partial L / \partial x_l^a$ and motion stream $\partial L / \partial x_l^m$ can be calculated through the chain rules as:

$$\frac{\partial L}{\partial x_l^a} = \frac{\partial L}{\partial x_{l+1}^a} \frac{\partial x_{l+1}^a}{\partial x_l^a} = \frac{\partial L}{\partial x_{l+1}^a} \left(\frac{\partial f(x_l^a)}{\partial x_l^a} + \frac{\partial}{\partial x_l^a} F(x_l^a + f(x_l^m), W_l^a) \right) \quad (3)$$

$$\frac{\partial L}{\partial x_l^m} = \frac{\partial L}{\partial x_{l+1}^m} \frac{\partial x_{l+1}^m}{\partial x_l^m} + \frac{\partial L}{\partial x_{l+1}^a} \frac{\partial}{\partial x_l^m} F(x_l^a + f(x_l^m), W_l^a) \quad (4)$$

Similarly, as depicted in Figure 5d, applying the multiplicative motion gating to two streams can be formalized as:

$$x_{l+1}^a = f(x_l^a) + F(x_l^a \odot f(x_l^m), W_l) \quad (5)$$

Here, $\odot$ denotes element-wise multiplication. Correspondingly, the gradient of the loss function L of the appearance stream in the backpropagation is formulated as:

$$\frac{\partial L}{\partial x_l^a} = \frac{\partial L}{\partial x_{l+1}^a} \frac{\partial x_{l+1}^a}{\partial x_l^a} = \frac{\partial L}{\partial x_{l+1}^a} \left(\frac{\partial f(x_l^a)}{\partial x_l^a} + \frac{\partial}{\partial x_l^a} F(x_l^a \odot f(x_l^m), W_l^a) f(x_l^m) \right), \quad (6)$$

where the gradient of residual unit is modulated by the signal $f(x_l^m)$ from the motion stream.

The loss function L of the motion stream is similarly calculated as:

$$\frac{\partial L}{\partial x_l^m} = \frac{\partial L}{\partial x_{l+1}^m} \frac{\partial x_{l+1}^m}{\partial x_l^m} + \frac{\partial L}{\partial x_{l+1}^a} \frac{\partial}{\partial x_l^a} F(x_l^a \odot f(x_l^m), W_l^a) x_l^a, \quad (7)$$

where the gradient of the residual unit is modulated by the signal x_l^a from the appearance stream and is added to the gradient of the motion stream $(\partial L / \partial x_{l+1}^m)(\partial x_{l+1}^m / \partial x_l^m)$ to become the gradient of the loss function L of the motion stream.

Finally, as depicted in Figure 5e, the bidirectional multiplicative residual connections can be considered as the multiplicative motion gating and multiplicative appearance gating applied on appearance stream and motion stream respectively, and the multiplicative appearance gating is similar as the motion gating discussed above.

As a result, the signals from each of the streams can be involved together and then affect the gradient during the backpropagation, which helps mitigate the problem of the original two-stream network's deficiency in modeling the subtle spatiotemporal correlations.

3.3. Res-C3D Network

We built each stream on the basis of 3D CNN with residual blocks instead of 2D CNN. We utilized C3D [4] as the base model which was equipped with 3D convolution kernels to directly extract the hidden patterns from a stack of RGB or optical flow frames ordered by time sequences. Therefore, it was an effective feature learning network for modeling the spatial features together with the local short fixed-range temporal relationships.

The original C3D model had five convolution blocks with eight convolution layers, five pooling layers, two fully connected layers, and a softmax loss layer. All the convolution layers had $3 \times 3 \times 3$ convolution kernels with stride $1 \times 1 \times 1$. The channel size for each convolution block was set to 64, 128, 256, 512, and 512 from the first block to the fifth block. The second to the fifth pooling layers all had $2 \times 2 \times 2$ pooling kernels with stride 2 in both the spatial and temporal dimensions in order to downsample the feature maps by a factor of eight. The first pooling layer had a kernel size of $1 \times 2 \times 2$, which was designed for preserving the temporal signals in the early steps. The two fully connected layers had a 4096 dimensional output size, followed by a softmax classifier to obtain the prediction results.

We generalized the idea of identical mapping with residual connections for the original C3D network and constructed an identical mapping “shortcut” by introducing residual connections between convolution blocks, which are depicted as the green curved arrows in Figure 6. These residual blocks are capable of preserving network performance by maintaining the learned features via identical mapping and integrating newly learned patterns to optimize the spatiotemporal representations.

Finally, considering that the fully connected layer was capable of perceiving the entire model so as to obtain high-level semantic information, we merged the outputs of the fully connected layers from two streams together to be the short-term spatiotemporal features of encoder. Since the deeper fully connected layers contained stronger signals, fc6 of appearance stream and fc7 of motion stream were combined together, and this type of combination strategy achieves better result compared to others. It can be explained by the fact that it is better to contain appearance signals with less strength and make the motion signals predominate during the training process.

Figure 6. The structure of the encoder. Each stream is constructed on the proposed Res-C3D, as depicted within the area of yellow and blue rounded rectangles. The orange arrows show the residual connections from the motion stream to the appearance stream, and $\odot$ indicates the element-wise multiplicative operation. The final feature vector is the combination of outputs of fc6 in the appearance stream and fc7 in the motion stream.

3.4. BiLSTM Network

Owing to the recurrent connections of each unit, RNNs yield good performances in modeling hidden sequential patterns of data. However, the “internal memory” property and the “vanishing gradient” problem of the RNN recurrent structure make it difficult to update the network parameters during backpropagation; thus, RNNs are deficient in interpreting early information and modeling the long-range temporal contexts of feature vector sequences.

Fortunately, LSTM conquers this fundamental weakness through its unique cell structure, which contains forget, input, and output gates controlled by sigmoid units to decide what information to update and store in memory cells. Linear connections across these LSTM units help to transmit the previous information to the present time step. The structure of an LSTM unit is illustrated in Figure 7.

Figure 7. The structure of an LSTM unit.

Given the current input vector x_t , the last hidden state h_{t-1} , and the last memory cell state c_{t-1} , the operations inside the LSTM cell can be formulated as

$$i_t = s(U_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (8)$$

$$f_t = s(U_{xf}x_t + W_{hf}h_{t-1} + b_f), \quad (9)$$

$$o_t = s(U_{xo}x_t + W_{ho}h_{t-1} + b_o), \quad (10)$$

$$g_t = y(U_{xt}x_t + W_{ht}h_{t-1} + b_c), \quad (11)$$

$$c_t = f_t \cdot c_{t-1} + g_t \cdot i_t, \quad (12)$$

$$h_t = o_t \cdot y(c_t), \quad (13)$$

where i_t , f_t , o_t , and g_t denote the input gate, forget gate, output gate, and input modulation gate at time t , respectively. U , W , and b denote the input weights, recursive weights, and bias vector. $s(x) = (1 + e^{-x})^{-1}$ denotes the sigmoid activation function, and $y(x) = (e^x - e^{-x}) / (e^x + e^{-x})$ denotes the hyperbolic tangent function. g_t is a vector that offers candidate values to update the memory cell, which is calculated from the present input and previous state by the tanh activation function. The input gate i_t and the input modulation gate g_t control what to write to the memory cell, while the forget gate f_t controls what previous information transmitted from the past to discard. The output gate o_t keeps the information for forthcoming operations that control the output of the cell at time t . After updating the memory cell as c_t , the hidden state h_t at time t can be calculated by element-wise multiplication of the output gate vector o_t and current memory cell state c_t after projection by the tanh function.

Stacking two LSTM layers of opposite directions together in our network serves as the point of departure for the use of the bidirectional temporal structures of feature vector sequences. In contrast to the outputs of standard single-layer LSTM at time t , the combined outputs of BiLSTM layers are decided by not only the cues from previous vectors but also cues from upcoming vectors. Therefore, by integrating the extra information from future data, BiLSTM networks are capable of generating higher-level global context relationships of sequential data from videos.

The CNN feature vectors from the encoder serve as the visual representations of each video clip, which are ordered as temporal sequences. Venugopalan et al. [27] employed the LSTM network to model the temporal dependencies of the features, which can be expressed as:

$$v_t = g(f_t | v_{t-1}), \quad (14)$$

where f_t indicates the feature vectors modeled at time t , v_t denotes the temporal information, and g denotes the modeling function for temporal relationships.

In contrast to Equation (14), which only takes the past information into consideration, in our work, we employed the BiLSTM network to abstract the temporal representation of both past and future information separately, and two hidden states of LSTM were merged to become the output, which can be illustrated as:

$$v_t = g(f_t | v_{t-1}, v_{t+1}). \quad (15)$$

In other words, by employing the BiLSTM network in our model, the long-term bidirectional global temporal relationships were abstracted by going forward and backward several times through the vector sequences encoded from all the video segments.

3.5. Temporal Attention Mechanism

When we are looking at a dog, we usually focus on the body of the dog and try to observe it clearly while roughly glancing at other regions, such as the background. The attention mechanism is first implemented as a spatial version for image perception, which directs the visual attention to the salient part of images and pays less attention to other useless areas. Similarly, the contribution provided by every single input element from sequential data to abstract semantic content is not equal. For example, the start and the end segments of videos usually carry less vital information compared with some key segments located in the middle of videos. The principle of the temporal attention mechanism is to decide “when” to look by automatically directing high levels of focus to those vectors that contain the most valuable information while directing relatively low-level focus to those containing less information according to the importance and relevance of the tasks. In our work, we employed the temporal attention mechanism in the BiLSTM network to simulate the process of focusing attention among all segments of videos, where the gradients of the loss function of BiLSTM can be backpropagated through both the BiLSTM network and the attention network.

The BiLSTM with the temporal attention mechanism is depicted in Figure 8. Specifically, given the input vector x_{it} , where $t \in [1, T]$, the hidden state of each time step encoded by BiLSTM can be formulated as

$$\vec{h}_{it} = \vec{LSTM}(x_{it}), \quad (16)$$

$$\overleftarrow{h}_{it} = \overleftarrow{LSTM}(x_{it}), \quad (17)$$

where the forward and backward directions of the input sequences are denoted as $\leftarrow$ and $\rightarrow$, respectively. The final hidden state of BiLSTM can be expressed as:

$$h_{it} = [\vec{h}_{it}, \overleftarrow{h}_{it}]. \quad (18)$$

Then, the attention-based temporal attribute vector o_i is calculated by the weighted average:

$$o_i = \sum_{t=1}^T \beta_{it} \cdot h_{it} \quad (19)$$

Here, β_{it} denotes the attention weights of the i -th BiLSTM output vector at time t , which is computed by the following equations:

$$\gamma_{it} = \tanh(W_a h_{it} + b_a), \quad (20)$$

$$\beta_{it} = \frac{\exp(\gamma_{it})}{\sum_{j=1}^T \exp(\gamma_{it})}, \quad (21)$$

where W_a is the weight matrix used to project the current input h_{it} of BiLSTM into another space, and b_a is a bias vector. We utilize the nonlinear $\tanh$ activation function for the frame selection gate as the relevance vector β_{it} and normalize it through a softmax function. In other words, the temporal attention mechanism can be learned as a special representation of the time steps of interest to restrain the redundancy and noises.

Figure 8. Connections depicting the structure of the BiLSTM model with the attention mechanism at a time step.

Then, we employed another LSTM for the final representations of videos by taking the integration of the CNN visual features extracted by the encoder and the merged outputs of BiLSTM together as inputs. As illustrated in Figure 9, there is an identical mapping residual connection from the output layer of the encoder to the bidirectional representations, which constructs a highway between layers and is thereby more conducive to gradient propagation through the network and the abstraction of global spatiotemporal representations.

Figure 9. The approach to using the residual connection to combine the CNN features and the BiLSTM representations. $\oplus$ denotes the attention mechanism effects on the BiLSTM model.

4. Experiments

In this section, our proposed network is evaluated on two benchmark datasets named UCF101 and HMDB51 for video action recognition. In the first part, we introduce these two datasets and the implementation details of our experiments. An ablation study for demonstrating the effectiveness of our proposed method is described in the next part. In the last part, we compare our TBRNet with state-of-the-art approaches.

4.1. Datasets and Implement Details

Created by Florida University, the UCF101 dataset is one of the most challenging datasets for video action recognition in realistic scenes, where the videos are captured with a large diversity of poses, viewpoints, illumination conditions, and cluttered background. It contains 13,320 videos with a spatial resolution of 320×240 that can be categorized into 101 classes. Each class includes 100–200 videos, and these classes of action are divided into five types: (1) human–object interaction, (2) body motion only, (3) human–human interaction, (4) the playing of musical instruments, and (5) sports.

The HMDB51 dataset, a more challenging dataset published by American Brown University in 2011, contains more than 6000 video clips in total. The videos are organized as 51 distinct classes of action, and each class has more than 100 clips. The dataset covers highly diverse actions, such as object manipulations, facial actions, body movements, and human interactions. Problems such as poor video quality, a small number of training videos, significant camera motion, and a complex dynamic environment lead to higher intra-class variation, which makes it more challenging to reduce errors during training and testing periods. For both datasets, we adopted the original training/test splits and followed the standard evaluation protocols provided by the authors of these two datasets by averaging the accuracies of three splits as the final result.

For the two-stream CNN encoder, we adopted the pre-computed RGB and optical flow frames of videos from Reference [3] as the inputs of our model. We divided each video into $N = 12$ segments, which also relates to the time steps of the BiLSTM decoder sequences. We sampled $t = 8$ frames of RGB frames and the corresponding t optical flow frames for each segment and used them as the input modalities. The settings of N and t are based on our experiments, which show optimal results in complexity and accuracy. Since the number of videos provided by datasets is limited, we adopted the same data augmentation strategy as [4] to mitigate the effect of overfitting. We resized the frame to 128×171 and randomly cropped them into 112×112 . Thus, one input segment has a size of $8 \times 112 \times 112$. Additionally, we horizontally flipped them with a 55% probability to further augment the training samples.

As UCF101 is a larger dataset than HMDB51, we first trained our two-stream encoder on UCF101 to extract visual CNN features and then transferred it for HMDB51. As depicted in Figure 6, the merged output of fc6 from the appearance stream and fc7 from the motion stream were used to represent features of a video segment, which is a vector of 8192 dimensions. For each video, 12 feature vectors were fed into BiLSTM with 8192-dimensional hidden states. The outputs of the BiLSTM network with a weighted sum attention pooling were fed into another LSTM, together with the CNN visual features, to obtain the final global spatiotemporal representations. In the end, those spatiotemporal representations were optimized by the softmax cross-entropy loss to predict the actions.

We trained the model with the stochastic gradient descent (SGD) algorithm and set the batch size to 32 to perform the batch normalizations because of the memory constraints. The learning rate was set to 0.01 in the first 2000 iterations and then lowered at a fixed ratio every 100 iterations. For better training and validating, we also applied a momentum of 0.9 and an early-stopping strategy during the process of training. In addition, we applied a dropout probability of 0.8 to the fully connected layers in the encoder and also applied the decoder with a dropout of 0.5 to bidirectional connections.

4.2. Ablation Study

4.2.1. Analysis of Res-C3D Network

We first evaluate the performance of our proposed Res-C3D with residual connections by using the performance of C3D as a comparison. Table 1 shows the classification accuracies on UCF101 and HMDB51 for the different pathways employed with these two networks. By introducing residual blocks, our proposed Res-C3D outperformed the C3D not only in the single stream but also in the two-stream architecture. Especially, when applied to the motion stream, the proposed Res-C3D achieved a 3.7% increase in accuracy on HMDB51. Therefore, the residual learning of Res-C3D proved to be beneficial for feature generation which can be attributed to better signal propagation, especially the backward propagation during the training process. Table 1 also shows that the two-stream network architecture outperforms by a large margin compared with the single-stream architectures because it is capable of appropriately taking advantage of the information from both streams to model the spatiotemporal features rather than only using the information from a single one.

Table 1. Comparison of classification accuracies (%) for the two-stream encoder constructed with C3D and the proposed Res-C3D on UCF101 and HMDB51.

Approach		UCF101	HMDB51
Appearance stream	C3D	81.5	46.2
	Res-C3D	82.8	48.4
Motion stream	C3D	85.9	53.3
	Res-C3D	86.1	57.0
Two-stream	C3D	89.3	61.6
	Res-C3D	90.8	63.4

4.2.2. Analysis of Cross-Stream Connections

For the purpose of learning the correspondences between the appearance and motion signals from identical levels of abstraction in the two streams, we introduced cross-stream connections into the two-stream network. Generally, as shown in Figure 5, there are several possible variants of connections for interactions between pixels at the same position of the two processing pathways. We use the classification errors on UCF101 and HMDB51 to evaluate the performances of these different types of interaction; as shown in Table 2, a lower value of error represents a more appropriate model. In Table 2, $\oplus$ denotes the additive operation, while $\odot$ denotes the element-wise multiplicative operation. In addition, we use $\leftarrow$ and $\rightarrow$ to denote the direction of flow from the motion to the appearance stream and its opposite direction, respectively, and use $\longleftrightarrow$ to represent the bidirectional connection between the two streams.

Firstly, we used the direct additive and multiplicative shortcut for cross-stream connections, as illustrated in Figure 5a,b. In the two-stream architecture equipped with the proposed Res-C3D, these kinds of structures allowed for the straightforward flow of the motion signals via the residual connections of each Res-C3D in both pathways, which, however, introduce large changes in input distribution into the appearance pathway. Those changes not only flow through the forward processing procedure to reach the deep layers but also are backpropagated to former layers after fusing with appearance features, thus resulting in the disturbance of the network capability of pattern encoding.

In other words, the straightforward shortcuts damaged the identity mapping of the original residual signal flow of the appearance Res-C3D, which increased the difficulty of optimization and increased the classification errors as listed in the first two rows of Table 2. We also note that the performance of the direct multiplicative shortcut connection is even worse compared with that of the additive shortcut connection. The explanation for this is that employing multiplicative operations on a straightforward connection amplifies the impacts of disturbing basic feature extraction on the two pathways. The results are similar when applying the connections from the appearance stream to the motion stream. Therefore, we continued the following experiments not based on the simple direct interactions but on the cross-stream residual connections illustrated in Figure 5c,d,e.

Table 2. Classification errors (%) on UCF101 and HMDB51 of the two-stream encoder with different cross-stream connections. We use $\oplus$ to denote an addition operation and use $\odot$ to denote an element-wise multiplication operation. The directions from the motion to the appearance stream and from the appearance to the motion stream are denoted as $\leftarrow$ and $\rightarrow$, respectively, while we use $\leftrightarrow$ to denote the bidirectional connection.

Connection Case	Figure	UCF101	HMDB51
direct $\oplus \leftarrow$	Figure 5a	21.72	57.15
direct $\odot \leftarrow$	Figure 5b	75.26	79.43
residual $\oplus \leftarrow$	Figure 5c	9.04	39.91
residual $\odot \leftarrow$	Figure 5d	7.53	36.69
residual $\oplus \rightarrow$	Figure 5c	14.51	48.03
residual $\odot \rightarrow$	Figure 5d	14.22	46.82
residual $\odot \leftrightarrow$	Figure 5e	13.87	46.75

Secondly, we compared the additive and multiplicative operations for cross-stream residual connections, which were introduced in Section 2. As we can see from Table 2, using multiplicative gating from the motion to the appearance stream led to errors of 7.53% and 36.69% on UCF101 and HMDB51, and additive gating achieved slightly worse results with errors of 9.04% and 39.91%. The multiplicative residual connections strengthen the information correspondences of the two pathways by modulating the gradient propagation with signals from both pathways. This type of reinforcement of two signals from each stream improves the ability of the two-stream encoder to learn spatiotemporal features.

Thirdly, we evaluated the performances of employing different directions of cross-stream connections in our network. As we can see from rows 4, 6, and 7 of Table 2, under the circumstance of multiplicative gating, the fusing direction from the motion to the appearance stream achieves superior performance compared with the other two types. Similar to the additive gating (rows 3 and 5 of Table 2), letting the motion signals flow to the appearance stream outperforms the same schema but in the opposite direction. Since the appearance features have stronger effects on recognizing the hidden patterns in frames, when enabling the flow of appearance signals to the motion stream, the whole network is dominated by appearance signals, which makes the loss of motion stream quickly reduce to a low value and finally leads to the overfitting to appearance patterns. Things are different when using the $\leftarrow$ direction, where both pathways learn the motion patterns together without suffering from the overfitting resulting from the domination of motion signals, thus modeling more appropriate spatiotemporal features. In the $\leftrightarrow$ bidirectional connections, the same problem occurs in which the appearance signals flowing in both streams force the whole network to overemphasize appearance information from RGB frames.

Therefore, we employed the multiplicative cross-stream residual connections illustrated in Figure 5d with a fusion of directions from the motion to the appearance stream for the two-stream CNN encoder, which is verified to be more effective for extracting spatiotemporal features than other variants.

4.2.3. Analysis of Fusion Strategies

Table 3 shows the results of fusing two signals from different layers of the appearance and motion streams. Fusing the last convolution layer of the two pathways had inferior performance compared with the fusion of fully connected layers because the fully connected layers are capable of perceiving the overall model so as to obtain higher-level semantic information. Fusing both fc7 layers from each stream outperformed the fusion of both layers to fc6, which demonstrates that the fusion of deeper layers led to deeper patterns of spatiotemporal representations extracted by more layers. However, the combination of fc6 of the appearance stream and fc7 of the motion stream achieved the best result among these fusion modalities, which can be explained by the fact that it is better to contain appearance signals with less strength and make the motion signals predominate during the training process by utilizing representations from lower fully connected layers of the appearance stream and deeper fully connected layers of the motion stream.

Table 3. The accuracies (%) of different fusion strategies for two streams on UCF101 and HMDB51. “A-” means the layer from the appearance stream, and “M-” means the layer from the motion stream.

Fusion Layers	UCF101	HMDB51
A-Conv5 and M-Conv5	80.1	51.6
A-fc6 and M-fc6	88.5	62.0
A-fc7 and M-fc7	90.8	63.4
A-fc7 and M-fc6	87.3	58.5
A-fc6 and M-fc7	92.0	65.2

4.2.4. Analysis of Attention-Based BiLSTM

In Table 4, we evaluated the performances of different variants of the visual representation decoder by using the sequential features from the two-stream CNN encoder as inputs. The result showed that adding the LSTM network was beneficial for decoding the long-term visual representations into temporal dependencies because of its capability of “remembering” information, which yields margins of at least 1.3% and 3.1% on UCF101 and HMDB51. Furthermore, the bidirectional structure outperformed the unidirectional LSTM structure, because it provides double temporal contexts by gathering information from both previous and future segments. Additionally, the temporal attention mechanism was applied to the LSTM network, and the results verified that with the additional temporal attention layer, both unidirectional and bidirectional LSTMs, performed better than the LSTMs without the attention mechanism. This was because the temporal attention mechanism simulates the attention of humans by focusing less on unimportant video segments and concentrating more on salient parts by assigning different attention weights.

Table 4. The accuracies (%) in ablation study of the attention-based BiLSTM decoder on UCF101 and HMDB51. Here, $\rightarrow$ denotes the chronological order, and $\leftarrow$ denotes the reverse chronological order.

Decoder	UCF101	HMDB51
TBRNet without decoder	92.0	65.2
LSTM $\rightarrow$	93.5	69.0
LSTM $\leftarrow$	93.3	68.3
LSTM $\rightarrow$ + attention	94.1	69.7
BiLSTM	94.7	71.4
BiLSTM + attention	95.2	72.5
BiLSTM + attention + shortcut	95.4	72.8

We also compared the performance of using the residual connection to integrate CNN visual representations with BiLSTM temporal relationships as depicted in Figure 9. From the comparison of the last two rows of Table 4, we noticed that the network with the residual connections achieved

higher accuracy, and we attribute this improvement to the better propagation of global spatiotemporal representations within the whole network resulting from the overall fusion.

4.3. Comparison with State-of-the-Art Models

In this section, we compare our proposed model with existing state-of-the-art video action recognition approaches on both UCF101 and HMDB51 benchmark datasets. We categorize these approaches into three types according to the types of extracted features, as reported in Table 5.

Table 5. Compared performances of our proposed TBRNet with state-of-the-art approaches on UCF101 and HMDB51.

	Approach	UCF101	HMDB51
Hand-crafted	HOG [17,44]	72.4	40.2
	HOF [17,44]	76.0	48.9
	HOF+MBH [17,44]	82.2	54.7
	iDT+ Fisher vector [17]	85.9	57.2
Deep-learned	Two-stream [2]	88.0	59.4
	Two-stream fusion [3]	92.5	65.4
	Spatiotemporal resnets [38]	93.4	66.4
	TDD [44]	90.0	63.2
	C3D [4]	85.2	-
	3D Convolution [12]	91.8	64.6
	Res3D [26]	85.8	54.9
	LSTM [5]	88.6	-
	LRCN [6]	82.9	-
	Two-stream + LSTM [28]	88.6	-
	Multi-LSTM [45]	90.8	-
Hybrid	TDD + iDT [44]	91.5	65.9
	C3D + iDT [4]	90.4	-
	3D Convolution + iDT [12]	93.5	69.2
	TSN [19]	94.2	69.4
Ours	Encoder of TBRNet	92.0	65.2
	TBRNet	95.4	72.8

Compared with the approaches that use hand-crafted features [17,44], our proposed TBRNet yielded significant margins as high as 9.5% and 15.6% on UCF101 and HMDB51.

Our encoder of TBRNet was based on a two-stream network [2] and C3D [4] which outperformed these two base models by 4.0% and 6.8% on UCF101. We note that some two-stream-based approaches, such as Two-stream fusion [3] and Spatiotemporal ResNets [38], lead to higher accuracies compared with our proposed two-stream encoder, and this can be explained by the fact that these two approaches with similar fusion strategies among streams use deeper CNNs as base models which need to learn more network parameters. However, with the help of the temporal attention-based BiLSTM decoder, our entire TBRNet achieved improvements of 2.9% and 2.0% on UCF101 and 7.4% and 6.4% on HMDB51, which verifies the benefit of our decoder in capturing long-term dependencies for temporal sequences. Compared with other 3D CNN methods, such as 3D Convolution [12], Res3D [26], and other LSTM-based methods, such as LSTM [5], LRCN [6], Two-stream + LSTM [28], and Multi-LSTM [45], our proposed TBRNet still performed better.

From the performances of approaches that use hybrid features, such as TDD [44], C3D, and 3D Convolution, higher accuracies were obtained by taking iDT into consideration. Dense trajectories, calculated by the iDT algorithm, were stronger representations that contained more spatiotemporal features. However, they require higher computation resources compared with the usage of only optical flow. Note that without the BiLSTM decoder to model the long-range temporal dependencies, our two-stream CNN encoder still outperformed C3D and TDD with iDT versions by 1.6% and 0.5%

on UCF101, which shows the strong ability of our proposed encoder to capture spatiotemporal features from RGB frames and optical flow frames in a fixed-length temporal window. In addition, TSN [19] with iDT is pre-trained with the help of a large-scale ImageNet dataset which provides high diversity for training. Despite that, our proposed TBRNet also showed superior performance.

Finally, we demonstrate that our TBRNet with an encoder–decoder framework provided global spatiotemporal information complementarily and achieved classification accuracies of 95.4% and 72.8% on UCF101 and HMDB51.

The performances for all action classes in UCF101 and HMDB51 are given in Figure 10, and these confusion matrixes are well diagonalized. Some categories are easy to recognize, such as “pull-up” and “ride bike” in HMDB51, “BenchPress” and “BandMarching” in UCF101. However, some categories are hard to recognize, such as “laugh” in HMDB51 and “ApplyLipstick” in UCF101, since the videos in “ApplyLipstick” are easily confused with “ApplyEyeMakeup” and “BrushingTeeth”. Nonetheless, our proposed model still performs well with most categories.

Figure 10. The confusion matrixes of UCF101 and HMDB51 dataset.

Some correct and misclassified examples of the prediction results are shown in Figure 11a and 11b, respectively. We used the top two categories scores predicted by our TBRNet. It can be seen that similar actions can disturb the prediction of classifiers. For example, in the particular video, the action “Cartwheel” looks like “HandupStand”, which makes our classifiers confused. We attribute the misclassification of these actions to their high within-class variance and low between-class variance to their targeted false class.

Figure 11. Some examples of the prediction results. For each video, we used the top two categories' predicted scores. The gray bars indicate the ground truth label, the green and orange bars distinguish the correct and incorrect predictions, respectively. The length of each bar indicates the confidence. (a) The correct examples of the prediction results. (b) The incorrect examples of the prediction results.

5. Conclusions

In this paper, we proposed a new model with an encoder–decoder framework named TBRNet to learn more powerful spatiotemporal representations for video action recognition. To model the interactions between spatial and temporal features, we applied multiplicative residual motion gating on proposed two-stream encoder. To capture the global spatiotemporal dependencies, we extracted the local features in a fixed-length window by using proposed Res-C3D and extracted the long-term bidirectional attention-pooling temporal dependencies by using proposed attention-based BiLSTM. With systematic experiments to evaluate our proposed TBRNet on two benchmark datasets UCF101 and HMDB51, the boosting results demonstrate the effectiveness of our approaches. We have also compared our TBRNet with different types of existing state-of-the-art approaches and our model obtains competitive performance. In the future, we intend to work on the salient regions of the video frames, and cascade multi-head attention for action recognition. Moreover, knowledge distillation can be taken into consideration, since it is able to compress the network scale and reduce the computational complexity of the model.

Author Contributions: Q.J. conceived and designed the experiments; X.W. performed the experiments and wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Key R & D Program of Guangdong Province, grant number 2018B010107005 and the Natural Science Foundation of Guangdong Province, grant number 2016A030313288.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ullah, A.; Ahmad, J.; Muhammad, K.; Sajjad, M.; Baik, S.W. Action recognition in video sequences using deep bi-directional LSTM with CNN features. *IEEE Access* **2017**, *6*, 1155–1166. [[CrossRef](#)]

2. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, Canada, 8–13 December 2014; pp. 568–576.
3. Feichtenhofer, C.; Pinz, A.; Zisserman, A. Convolutional two-stream network fusion for video action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1933–1941.
4. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 4489–4497.
5. Yue-Hei Ng, J.; Hausknecht, M.; Vijayanarasimhan, S.; Vinyals, O.; Monga, R.; Toderici, G. Beyond short snippets: Deep networks for video classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4694–4702.
6. Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; Darrell, T. Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2625–2634.
7. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
8. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
9. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, CA, USA, 3–6 December 2012; pp. 1097–1105.
10. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
11. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Fei-Fei, L. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1725–1732.
12. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*; IEEE: Piscataway, NJ, USA, 2012; Volume 35, pp. 221–231.
13. Scovanner, P.; Ali, S.; Shah, M. A 3-dimensional sift descriptor and its application to action recognition. In Proceedings of the 15th ACM International Conference on Multimedia, Augsburg, Germany, 24–29 September 2007; pp. 357–360.
14. Hu, Y.; Cao, L.; Lv, F.; Yan, S.; Gong, Y.; Huang, T.S. Action detection in complex scenes with spatial and temporal ambiguities. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 27 September–4 October 2009; pp. 128–135.
15. Dalal, N.; Triggs, B.; Schmid, C. Human detection using oriented histograms of flow and appearance. In Proceedings of the European Conference on Computer Vision, Graz, Austria, 7–13 May 2016; pp. 428–441.
16. Laptev, I.; Marszalek, M.; Schmid, C.; Rozenfeld, B. Learning realistic human actions from movies. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 24–26 June 2008; pp. 1–8.
17. Wang, H.; Schmid, C. Action recognition with improved trajectories. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 3551–3558.
18. Saleh, A.; Abdel-Nasser, M.; Garcia, M.A.; Puig, D. Aggregating the temporal coherent descriptors in videos using multiple learning kernel for action recognition. *Pattern. Recogn. Lett.* **2018**, *105*, 4–12. [[CrossRef](#)]
19. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; Van Gool, L. Temporal segment networks: Towards good practices for deep action recognition. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 20–36.

20. Fernando, B.; Gavves, E.; Oramas, J.; Ghodrati, A.; Tuytelaars, T. Rank pooling for action recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*; IEEE: Piscataway, NJ, USA, 2016; Volume 39, pp. 773–787.
21. Carreira, J.; Zisserman, A. Quo vadis, action recognition? A new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017; pp. 6299–6308.
22. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A dataset of 101 Human Actions Classes from Videos in the Wild. Available online: <https://arxiv.org/pdf/1212.0402> (accessed on 3 December 2012).
23. Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; Serre, T. HMDB: A large video database for human motion recognition. In *Proceedings of the 2011 International Conference on Computer Vision*, Barcelona, Spain, 6–13 November 2011; pp. 2556–2563.
24. Wang, H.; Kläser, A.; Schmid, C.; Liu, C.L. Action recognition by dense trajectories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, USA, 20–25 June 2011; pp. 3169–3176.
25. Lazebnik, S.; Schmid, C.; Ponce, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, 17–22 June 2006; Volume 2, pp. 2169–2178.
26. Tran, D.; Ray, J.; Shou, Z.; Chang, S.F.; Paluri, M. Convnet Architecture Search for Spatiotemporal Feature Learning. Available online: <https://arxiv.org/pdf/1708.05038> (accessed on 16 August 2017).
27. Venugopalan, S.; Rohrbach, M.; Donahue, J.; Mooney, R.; Darrell, T.; Saenko, K. Sequence to sequence-video to text. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, 7–12 June 2015; pp. 4534–4542.
28. Varol, G.; Laptev, I.; Schmid, C. Long-term temporal convolutions for action recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*; IEEE: Piscataway, NJ, USA, 2017; Volume 40, pp. 1510–1517.
29. Ba, J.; Mnih, V.; Kavukcuoglu, K. Multiple Object Recognition with Visual Attention. Available online: <https://arxiv.org/pdf/1412.7755> (accessed on 23 April 2015).
30. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
31. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Tang, X. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017; pp. 3156–3164.
32. Kar, A.; Rai, N.; Sikka, K.; Sharma, G. Adascan: Adaptive scan pooling in deep convolutional neural networks for human action recognition in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017; pp. 3376–3385.
33. Korbar, B.; Tran, D.; Torresani, L. SCSampler: Sampling salient clips from video for efficient action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, Seoul, Korea, 27 October–2 November 2019; pp. 6232–6242.
34. Sang, H.; Zhao, Z.; He, D. Two-Level Attention Model Based Video Action Recognition Network. *IEEE Access* **2019**, *7*, 118388–118401. [[CrossRef](#)]
35. Sharma, S.; Kiros, R.; Salakhutdinov, R. Action Recognition Using Visual Attention. Available online: <https://arxiv.org/pdf/1511.04119> (accessed on 14 February 2016).
36. Yu, T.; Guo, C.; Wang, L.; Gu, H.; Xiang, S.; Pan, C. Joint spatial-temporal attention for action recognition. *Pattern Recognit. Lett.* **2018**, *112*, 226–233. [[CrossRef](#)]
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
38. Feichtenhofer, C.; Pinz, A.; Wildes, R.P. Spatiotemporal residual networks for video action recognition. In *Proceedings of the Advances in Neural Information Processing Systems*, Barcelona, Spain, 5–10 December 2016.
39. Feichtenhofer, C.; Pinz, A.; Wildes, R.P. Spatiotemporal multiplier networks for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017; pp. 4768–4777.

40. Qiu, Z.; Yao, T.; Mei, T. Learning spatio-temporal representation with pseudo-3d residual networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5533–5541.
41. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. Available online: <https://arxiv.org/pdf/1409.1556> (accessed on 10 April 2015).
42. Srivastava, R.K.; Greff, K.; Schmidhuber, J. Training very deep networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 2377–2385.
43. Srivastava, R.K.; Greff, K.; Schmidhuber, J. Highway Networks. Available online: <https://arxiv.org/pdf/1505.00387> (accessed on 3 November 2015).
44. Wang, L.; Qiao, Y.; Tang, X. Action recognition with trajectory-pooled deep-convolutional descriptors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4305–4314.
45. Yeung, S.; Russakovsky, O.; Jin, N.; Andriluka, M.; Mori, G.; Fei-Fei, L. Every moment counts: Dense detailed labeling of actions in complex videos. *Int. J. Comput. Vis.* **2018**, *126*, 375–389. [[CrossRef](#)]

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).