applied sciences

MDPI

Article

Attend It Again: Recurrent Attention Convolutional Neural Network for Action Recognition

Haodong Yang [1,*], Jun Zhang [1,*], Shuohao Li [1], Jun Lei [1] and Shiqi Chen [2]

[1] Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, No. 109, Deya Road, Changsha 410073, China; lishuohao@nudt.edu.cn (S.L.); leijun1987@nudt.edu.cn (J.L.)

[2] College of Electronic Sciences, National University of Defense Technology, No. 109, Deya Road, Changsha 410073, China; chenshiqi12@nudt.edu.cn

* Correspondence: yanghaodong12@nudt.edu.cn (H.Y.); zhangjun1975@nudt.edu.cn (J.Z.)

Received: 29 January 2018; Accepted: 28 February 2018; Published: 6 March 2018

Abstract: Human action recognition in videos is an important task with a broad range of applications. In this study, we improve the performance of recurrent attention convolutional neural network (RACNN) by proposing a novel model, "attention-again". We consider the nature of video frames as sequences, which will cause the change of regions of interest in the frame, thus we cannot use an attention mechanism similar to that in images. "Attention-again" model is a variant from traditional attention model for recognizing human activities and is embedded in two long short-term memory (LSTM) layers. Different from hierarchal LSTM which change the LSTM structure to combine the hidden states from two LSTM layers, our proposals introduce "attention-again" model to avoid the change of LSTM structure. Furthermore, this model not only learns the relations in each frame, but also obtains the relations among all frames, and these relations instruct the next learning stage. Therefore, our proposed model outperform the baseline and is superior to methods with the same experimental conditions on three benchmark datasets: UCF-11, HMDB-51 and UCF-101. To understand how the model works, we also visualize the region of interest in the frame.

Keywords: human action recognition; attention; recurrent attention convolutional neural network; LSTM

1. Introduction

Human action recognition in videos is one of the fundamental challenging tasks in computer vision. Since studies in images have had great success, researchers are paying more attention to video. In the literature, recognizing actions plays a vital role in several applications, such as multimedia contexts [1,2], video surveillance [3,4], video streaming [5–7], health care systems [8] and smart indoor security systems [9–11]. Recently, human action recognition has achieved great success in depth video and sensor [12–18]. In this work, we focus on RGB video sequences from single camera.

Unlike action recognition in images, action recognition in videos relies on motion dynamics in addition to visual appearance. Traditional approaches use hand-crafted features, and then input the aggregated features to the classifier. In contrast, the recent surge of deep neural networks (DNNs) achieve a significant advancement in extracting features. Among them, convolutional neural networks (CNNs) have shown a great ability to produce a rich representation of the input and achieve promising results on several challenging tasks, including image classification, object detection, machine translation, and tracking [19–21]. For this task, a direct approach to recognize action is to arm the convolutional operation with temporal information such as 3D convolutional networks [22,23]. Although they acheive great performance, these works are still missing much temporal information. Recurrent neural networks (RNNs), another branch of DNNs, are a class of

artificial neural networks, where the recurrent architecture allows the networks to exhibit dynamic temporal behavior. Since the video frames are sequential, RNNs might be more suitable for this kind of task. Thus, many works [24,25] have combined RNN with CNN for action recognition and have shown good results. Specifically, LSTM [26] has been proven by Donahue et al. [27], Shi et al. [28] and Ng et al. [29] to be effective for action recognition task from video sequences. In [25], they stack three LSTM layers and get great performance, however, they only fed the bottom hidden states of LSTM to top layers.Baradel et al. [30] are aware of weakness of this simply structure, thus, they embed the input feature maps with spatial attention model and change the structure of LSTM to fit the different input. Moreover, these changes improve the performance and then [30] outperform [25]. Attention mechanism mentioned in above method has obtained many interests from computer vision community and received a great achievement in caption generation, neural machine translation and image recognition. By attention mechanism, a model can learn where should be paid attention to, for example, in Figure 2 , the boy is juggling soccer, attention mechanism leads the model to focus on the leg and football which are important factors to classify the video into "soccer_juggling". The attention mechanism due to great performance of catching region of interest is also introduced to action recognition [25,27,31]. For example, Sharma et al. [25] transferred the attention mechanism on the spatial domain to action recognition.

Although the above DNN architectures with attention mechanisms have achieved great performance, we believe that attention mechanisms in video are not the same as that in images. For instance, in the task of image caption, the model with attention mechanisms learns to pay attention to the part of image from the previous state, which focus on the other part of image. However, images are different from videos, which consist of many frames. Therefore, the regions where activities are happening will be changing, and the interesting areas in previous frames might not be the main portions in next video frame. In addition, the intermediate feature vectors are limited as they are generated by only looking at previous frames in order. If we simply utilize this attention mechanisms into videos, we would only focus on the spatial information, and take less global state into consideration. In [32], they inspired by human behavior of reading document that they cannot understand some words in first read. Therefore, they introduce a simple mechanism that first reads the input sequence before committing to a representation of each word, and then, they propose a simple copy mechanism that is able to exploit very small vocabularies. Taking this mechanism into account, we find that understanding videos is also similar to reading, thus, we aim to develop a model to obtain great performances for action recognition in videos.

Therefore, in this study, we propose "attention-again" model to combine the neighboring frames with current frame to get the interesting part of the frame. This model has two LSTM layers. In the bottom layer of LSTM, we use the traditional attention mechanisms and generate the hidden state of LSTM unit from previous hidden state and current input. Next step, we integrate the hidden state of previous LSTM unit in top layer, current input feature and the current output from the bottom layer of LSTM unit. By this way, we could capture spatial relations by bottom LSTM layer and obtain more global information from top LSTM layer. Consequently, we not only consider the information from current state, but also fuse context information. Besides, we incorporate our proposed "attention-again" model with DNN framework. Our architecture is named recurrent attention convolutional Neural Network (RACNN) and is shown in Figure 1, where CNNs encode the input video frames into feature sequences.

The contributions of our proposals can be summarized as follows: (1) We introduce a novel deep learning model "attention-again" for action recognition in videos which can pay attention to the important region in video. Compared with traditional attention mechanisms, this model has improved performance. (2) An extensive set of experiments were conducted to demonstrate that the proposed model "attention-again" outperform the methods with the same experimental condition on three benchmark datasets: UCF-11 [33], HMDB-51 [34] and UCF-101 [35].

The rest of the paper is organized as following: In Section 2, we review the related work. We also present the development of action recognition and show some state-of-the-art work. The proposed model and the principle of our method is presented in Section 3. Experimental results and some implied training details are discussed in Section 4. Finally, Section 5 concludes this paper.

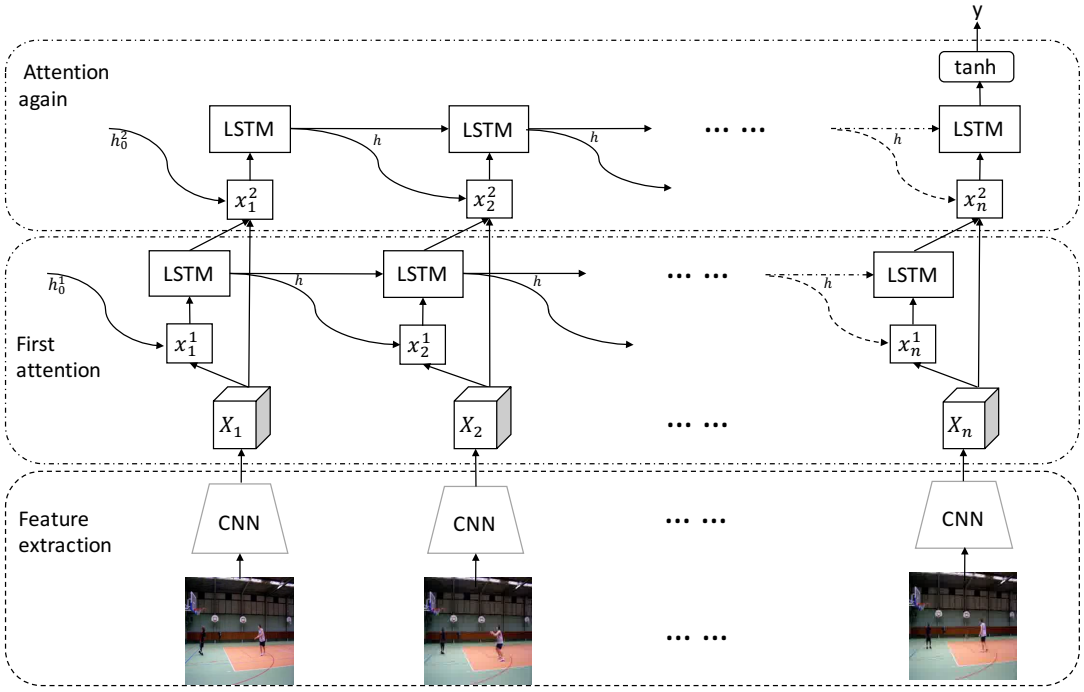


Figure 1. The architecture of the proposed recurrent attention convolutional neural network (RACNN) model. It contain three major components: (1) convolutional feature extraction; (2) long short-term memory (LSTM) sequence modeling and attention model; and (3) "attention-again" model. convolutional neural networks (CNNs) encode the input video frames into feature sequences. Traditional attention model in bottom layer refine the input feature sequences. The proposed model "attention-again" will get more information from input and hidden state.

2. Related Work

Human action recognition is a longstanding topic in computer vision community and is a well studied problem with various standard benchmarks. The literature on action recognition in video is vast and too broad for us to cover here completely. In this work, we present some related works on ConvNet architecture, LSTM-like architecture and attention mechanism.

2.1. ConvNet Architecture

Following the great success of CNNs for image classification, image segmentation and other computer vision tasks, deep learning algorithms have been used in video based human action recognition and have been proven that features learned from CNN are much better than hand-crafted features. In contrast to hand-craft shallow video representation, recent efforts try to learn the representation of video automatically from large scale labeled video data. For example, Karpathy et al. [23] directly applied CNNs to multiple frames in each sequence and obtain the temporal relations by pooling using single, late, early and slow fusion. Nevertheless, the results of this scheme are just marginally better than those of a single frame baseline, and indicate that motion features are difficult to obtain by simply and directly pooling spatial features from CNNs. To ameliorate this shortcoming, Simonyan et al. [36] proposed a two-stream ConvNet architecture, which combines optical flow and RGB video frames to train CNN and achieves comparable results with the state-of-the-art hand-craft based methods. Furthermore, dense trajectory pool CNN that combines

improved Dense Trajectories (iDT) [37] and two stream CNNs via the pooling layer achieves the state-of- the-art performance. Wang et al. [38] proposed a novel framework, temporal segment network (TSN), inspired from long-range temporal structure modeling, and obtained the state-of-the-art performance on the datasets of HMDB51 and UCF101. Another paradigm of action recognition, 3DCNNs, when solving above issues receive much attention from the community, which regard the video as a spatiotemporal blobs and train 3D filters to recognizing action. 3DCNNs is first introduced by Ji et al. [22] and learned 3D convolution kernels in both spatial and temporal space based on a straightforward extension of the established 2DCNNs. However, the performance of this method has been harder to scale for multi-stream methods. CNN based methods cannot accurately model the dynamics by simply averaging the scores across the temporal domain, even if the appearance features already achieve remarkable performance on other computer vision tasks.

2.2. LSTM-Like Architecture

To model the dynamics between frames, RNNs, particularly LSTM, have been considered in video based human action recognition. LSTMs are recurrent modules which can learn long-term dependencies using a hidden state augmented with nonlinear mechanisms to allow the state to propagate without modification. Most recent activity recognition methods [25,27,29] have CNNs underlying the LSTMs to mine the information in the feature map. Donahue et al. [27] proposed LSTMs that explicitly model short snippets of ConvNet activations. Meanwhile, Ng et al. [29] demonstrated that two-stream LSTMs outperform improved dense trajectories (iDT) [39] and two-stream CNNs, although they need to pre-train their architecture on sports 1-M videos. Srivastava et al. [24] also utilized pre-training on hundreds of hours of sports video, but they proposed an interesting LSTM based unsupervised training method and then fine-tuned this unsupervised pre-training LSTM to adapt human action recognition tasks. More importantly, they used an encoder LSTM to map an input sequence into a fixed-length representation, and used single or multiple decoder LSTMs to perform reconstruction of the input sequence or prediction of the future sequence. Then, Mahasseni et al. [40] proposed training LSTMs that are regularized using the output of another encoder LSTM (RLSTM-g3) grounded on 3D human-skeleton training data. Baccouche et al. [41] proposed a two-step learning method in which 3D features are extracted from CNNs, and then an RNN is trained to classify each sequence by considering the temporal evolution of the learned features.

2.3. Attention Mechanism

To address this, attention models was introduced to capture where the model is focusing its attention when performing this task. Karpathy et al. [23] used a multi-resolution CNN architecture to perform action recognition in videos. They mentioned the concept of fovea but they fixed attention to the center of the frame. More recently, Jaderberg et al. [42] proposed a soft-attention mechanism called the Spatial Transformer module which they added between the layers of CNNs. Instead of weighting locations using a softmax layer, which we do, they applied affine transformations to multiple layers of their CNN to attend to the relevant part and get state-of-the-art results on the Street View House Numbers dataset. Yeung et al. [43] performed dense action labeling using a temporal attention based model on the input-output context and report higher accuracy and better understanding of temporal relationships in action videos. Fantastic work by Sharma [25] and Xu et al. [44] used both soft attention and hard attention mechanisms to generate descriptions. Our work directly builds upon these works. Iin this paper, we use a soft attention mechanism for action recognition in videos. Combined with proposed "attention-again" model, our model outperforms theirs.

3. The Proposed Method

The overall architecture of the proposed method is illustrated in Figure 1. We will describe the three major components of our method in detail in this section: (1) convolutional feature extraction; (2) LSTM sequence modeling and attention model; and (3) "attention-again" model.

3.1. Convolutional Features Extractor

CNN has shown a good ability to produce a rich representation of the input image by embedding it into a fixed feature vector. In many tasks, it is natural to use CNN as the encoder to extract the features and networks which are pre-trained on ImageNet dataset might obtain great performance. In general, our architecture can adapt any deep convolution networks for feature extraction, thus, we conduct our encoder network with GoogLeNet, VGG and ResNet-101,152.

First, we employ the opencv toolkit to split each video into 16 video frames with fixed interval, and then, we resize all frames into 224 × 224 and extract feature cubes from the different layers in different networks. For example, in GoogLeNet, we pick up the last convolutional layer, and we extract feature cubes from the last pooling layer and in both ResNet-101 and ResNet-152, also known as , block4 unit3 conv2. Simply, we represent these feature cubes as $D \times K \times K$ (e.g., 512 × 14 × 14 used in VGG Net). Especially, $K \times K$ is the size of feature map of the input image and D is the number of the feature dimensions. Hence, at each time-step t, we extract K^2 D-dimensional vectors. We refer to these vectors as feature slices into a feature cube, and denoted as:

$$\begin{aligned} X_t &= [X_{t,1}, \cdots, X_{t,K^2}] \\ X_{t,i} &\in R^D \end{aligned} \tag{1}$$

where each vertical feature vector is the same region of feature map in different dimensions. At time step t, these K^2 vertical feature slices to different overlapping regions in the input space and our model chooses to focus its attention on these K^2 regions.

3.2. LSTM Sequence Modeling and Attention Model

3.2.1. LSTM Sequence Modeling

Since the video is a sequence of frames, we establish a model $p(y|X_t, h_{t-1})$ with RNNs. After extracting the features in videos, we input all X_t in LSTMs. The LSTMs have been proven to get great performance in many tasks, thus, we utilize LSTMs to process the features. In this subsection, we briefly describe the LSTMs unit which is the basic part of our model. LSTM relies on a fantastic structure of gates (i_t, f_t, o_t, g_t) to control the flow of information to the hidden neurons and has the ability to preserve sequence information over time and capture long-term dependencies. Moreover, recent advances in computer vision also suggest that LSTM has potential to model videos for action recognition [25]. We follow the LSTM implementation in [45], which is given as follows:

$$\begin{aligned} i_t &= \sigma(W_{ix}x_{n,t} + W_{im}h_{t-1} + b_i) \\ f_t &= \sigma(W_{fx}x_{n,t} + W_{fm}h_{t-1} + b_f) \\ o_t &= \sigma(W_{ox}x_{n,t} + W_{om}h_{t-1} + b_o) \\ g_t &= \tanh(W_{gx}x_{n,t} + W_{gm}h_{t-1} + b_g) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \tag{2}$$

where i_t is the input gate, f_t is the forget gate, o_t is the output gate, and g_t is calculated, as shown in Equation (2). c_t is the memory cell state at t, h_t is the hidden state at t, and $x_{n,t}$ stands for input features at t. σ means the sigmoid function and $\odot$ denotes the element-wise multiplication. The weight matrices are denoted by W_{ij} and biases b_j, which are the trainable parameters. The cores of the LSTM unit are a memory cell c_t and three gates i_t, o_t, and f_t. Memory cell c_t encodes the information of previous memory cell c_{t-1} and current input by employing these three gates, which control the flow of information in LSTM units. For convenience, we denote $h_t, c_t = LSTM(x_t, h_{t-1}, c_{t-1})$ as the computation function for updating the LSTM internal state.

In [42], they conduct a hierarchical structure with two LSTM layers, which is similar to our method. However, they change the structure of LSTM to fit different input. Compared with [42], we take the traditional LSTM as our loop-connected processor, and attention-again model is used to solve the problem of input state.

To converge faster, we use the following initialization strategy (see Xu et al. [44]) for the cell state and the hidden state of the LSTM:

$$\begin{aligned} c_0 &= f_{init,c}(\frac{1}{T}\sum_{t=1}^{T}(\frac{1}{K^2}\sum_{i=1}^{K^2}X_{i,t})) \\ h_0 &= f_{init,h}(\frac{1}{T}\sum_{t=1}^{T}(\frac{1}{K^2}\sum_{i=1}^{K^2}X_{i,t})) \end{aligned} \tag{3}$$

where $f_{init,c}$ and $f_{init,h}$ are two multilayer perceptrons and T is the number of time-steps in the model. These values are used to calculate the first location softmax $a^1_{t,i}$ (see Section 3.2.2) which determines the initial input $x_{n,1}$.

3.2.2. Attention Model

In an image, the key to recognize what is in it is to find the important parts or regions of interest. In other words, different parts of the image have different weights. Compared with an image, not only the regions in a video, but also the frames in a video play different roles. Sharma et al. [25] obtained great achievement with attention mechanism and they have proven that soft attention could gain the performance in CNN-RNN model for action recognition task. To obtain this information, we employ soft attention mechanism.

The relations in each frame are of great importance for recognizing action. To discovery these relations, we incorporate attention mechanism with LSTM. In this layer, hidden state h^1_{t-1} generated from LSTM integrates input features X^1_t extracted from ConvNet. Then, we design a function that takes h^1_{t-1} and X^1_t as the input and returns the unnormalized relevance score at time step t, which is shown in below:

$$e^1_{t,i} = W_1^T \tanh(W^1_a h^1_{t-1} + U^1_a X^1_t + b^1_a) \tag{4}$$

Once all relevance scores for feature sequence are computed, we normalize them to obtain the $a^1_{t,i}$:

$$a^1_{t,i} = \frac{exp(e^1_{t,i})}{\sum_{j=1}^{K\times K} exp(e^1_{t,j})} \qquad i \in 1...K^2 \tag{5}$$

The K^2 vectors in the appearance feature cubes correspond to K^2 regions in the t-th frame and our model chooses to focus its attention on these K^2 regions. $a^1_{t,i}$ is a softmax over $K \times K$ locations and denotes weight of the i-th region. At each time step, we update the parameters of attention model, and get the input x^1_t by the following function:

$$x^1_t = \sum_{i=1}^{K\times K} X^1_{t,i} a^1_{t,i} \tag{6}$$

After weighting the features from ConvNet, we get x^1_t and input it into LSTM, $h^1_t, c^1_t = LSTM^1(x^1_t, h^1_{t-1}, c^1_{t-1})$.

The illustration of the above process at the time step t is shown in Figure 2. In Equation (4), the W_1^T, W^1_a, U^1_a and b^1_a are estimated together in model training procedure.

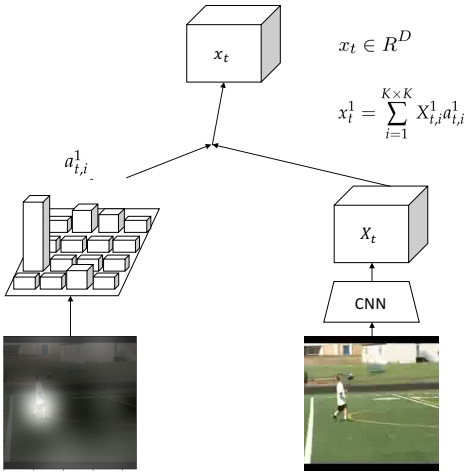


Figure 2. The illustration of attention process. $a_{t,i}^1$ is generated from hidden state and input, and then refines the input of first layer of LSTM through attention weight.

3.3. "Attention-Again" Model

"Attention-again" model combines the neighboring frames with current frame to get the interesting part of the frame. In other words, it could obtain more information than tradition attention model. To incorporate "attention-again" model into our framework, we stack another LTSM layer. Then, we utilize the current state in bottom LSTM network, the previous hidden state in top LSTM network and the current input features to generate unnormalized relevance score at time step t:

$$e_{t,i}^2 = W_2^T \tanh(W_{pa}^2 h_t^2 + W_a^2 h_{t-1}^2 + U_a^2 X_t^2 + b_a^2) \tag{7}$$

Similar to Equation (5), we normalize relevance score at time step and get $a_{t,i}^2$

$$a_{t,i}^2 = \frac{exp(e_{t,i}^2)}{\sum_{j=1}^{K\times K} exp(e_{t,j}^2)} \qquad i \in 1...K^2 \tag{8}$$

where h_t^1 is hidden state generated from bottom layer at time t, and h_{t-1}^2 represents hidden state generated from top layer at time $t-1$. In Equation (7), the $W_2^T, W_{pa}^2, W_a^2, U_a^2$ and b_a^2 are estimated together in model training procedure. After that, we obtain the input of top LSTM, which is determined by Equation (9):

$$x_t^2 = \sum_{i=1}^{K\times K} X_{t,i}^2 a_{t,i}^2 \tag{9}$$

Then, we feed this x_t^2 as features into top LSTM layer, $h_t^2, c_t^2 = LSTM^2(x_t^2, h_{t-1}^2, c_{t-1}^2)$.
In Section 4.3.1, we discuss the number of layer with "attention-again" model.

3.4. Decode Network

After embedding attention mechanism into our framework, we then input the result of above $LSTM^2$ to decode network. The main methods consider all hidden state, and set $softmax$ result as the final category, which is shown in below function:

$$y \sim argmaxSoftmax(W_y h_t^2 + b_y) \tag{10}$$

In our work, we only consider the hidden state from the last LSTM unit in top LSTM layer and acheive great performance. In addition, the loss will converge fluently compared to the main methods.

$$y \sim (Wh_n^2 + b) \tag{11}$$

4. Experiments

4.1. Datasets

To verify the effectiveness of our methods, we conduct experiments mainly on three public action recognition benchmark datasets, namely UCF-11 (YouTube Action), HMDB-51 and UCF-101, which are currently challenging annotated action recognition datasets.

UCF-11 dataset, also known as the YouTube Action dataset, consists of 1600 videos and 11 actions: basketball shooting, biking/cycling, diving, golf swinging, horse back riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog. The clips have a frame rate of 29.97 fps and each video has only one action associated with it. We use 1120 videos for training and 480 videos for testing (70% for training and 30% for testing).

HMDB51 dataset is a large collection of realistic videos from various sources, including movies and web videos. It is composed of 6766 video clips from 51 action categories, with each category containing at least 100 clips. These clips are labeled with 51 classes of human actions such as Clap, Drink, Hug, Jump, Somersault, Throw and many others. This dataset is more challenging than others because it contains scenes with more complex backgrounds. It has similar scenes in different categories, and it has a small number of training videos. HMDB-51 have three split settings to separate the dataset into training and testing videos. Our experiments follow the original evaluation scheme, but only adopt the first training/testing split. In this split, each action class has 70 clips for training and 30 clips for testing.

UCF101 dataset is composed of about 13,320 realistic user-uploaded video clips and 101 action categories, with each category containing at least 100 clips. The database is particularly interesting because it gives the largest diversity in terms of actions and with the presence of large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc, and it is the most challenging dataset to date. There are three splits for training and testing (70% training and 30% testing). We tested our model on the first training/testing split in the experiments. Classification accuracy is used as evaluation measure.

4.2. Training Details

As shown in Figure 1, we use a CNN-RNN network structure as in [25] to get our final prediction. It is believed that training a good deep convolutional neural network for videos understanding is more challenging. Similar to [25], we only train all the hyper-parameters in RNNs by inputing features extracted from pre-trained ConvNet.

In detail, the dimensionality of the LSTM hidden state, cell state, and the hidden layer were set to 1024. To avoid overfitting, we use dropout (similar to Srivastava et al. [24]) of 0.5 at all non-recurrent connections. After that, we employ Adam optimization algorithm (from Kingma et al. [46]) to train all models. Next step, the learning rate starts from 0.003, then we use exponential decay algorithm which was implied in TensorFlow toolbox and the decay rate is set to 0.9.

We implement our model with TensorFlow [47], and our experiments are carried out on a workstation with a 4GHz Intel(R) Core (M) i7-4790 CPU, 128G RAM and an NVIDIA(R) GeForce 1080 GPU. In the process of training and testing, our model uses CUDA to accelerate the experimental process. For UCF-11 dataset, we set batch size as 16, while the batch size of both HMDB-51 and UCF-101 is set as 8.

4.3. Results and Analysis

In this part, we analyze our experiment in two aspects. (1) Quantitative analysis: We compare the result measured on three benchmark datasets with other state-of-art works under the same condition; (2) Qualitative analysis: We show some examples that illustrate the attention in the frame vividly.

4.3.1. Quantitative Analysis

The Effect of Different CNN Encoders

In the famous imageNet competition, ResNet have get great performance due to deep structure. Since action recognition is not the same as image classification, we verify the result of different CNN encoders. To date, there are four widely used CNN encoders, namely GoogLeNet, VGG, ResNet-101 and ResNet-152, to extract visual features. In this sub-experiment, we study the influence of different versions of CNN encoders on our framework. The experiments are conducted on RGB data on the UCF11 and first split of HMDB51 and UCF101 datasets. The results are shown in Table 1. These above networks are all pre-trained on imageNet dataset. We can easily find that, by taking ResNet-152 as the visual decoder, our method perform best with 91.2% on UCF11, 54.4% on HMDB51 and 87.7% on UCF101.

Table 1. Convolutional neural networks (CNNs) encoder analysis on UCF11 and first split of HMDB51 and UCF101. Classification accuracy is used as evaluation measure which represented as recognition accuracies (%). The **bold** text represents the best result.

Model	UCF11	HMDB51	UCF101
GoogLeNet	89.7	52.3	85.4
VGG	90.1	52.6	85.8
ResNet-101	90.9	53.8	87.2
ResNet-152	**91.2**	**54.4**	**87.7**

The Effect of Every Component

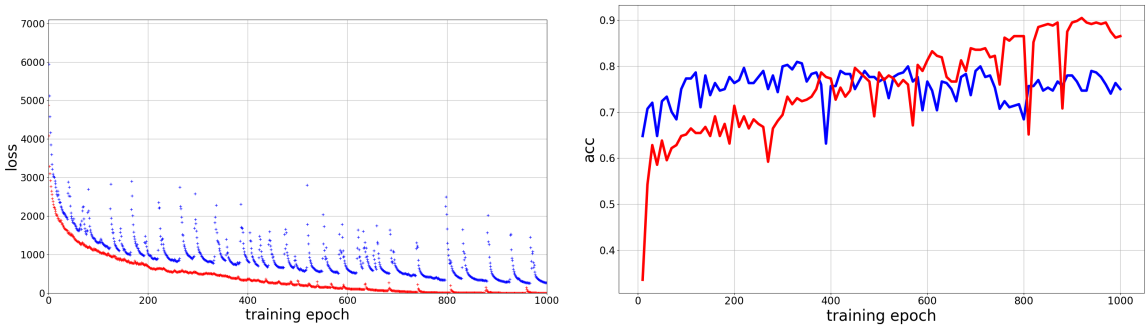
The baseline of these methods is a simple "encode–decode" model with CNN encoding the input videos and LSTM decoding features into categories. In [48], they compare baseline performance of LSTM, Attention-LSTM (ALSTM), ConvLSTM [49] and ConvALSTM. In Table 2, we list the performance of all these LSTM variants and our main component for action recognition. To conduct a fair comparison, we use VGG net as our encode network and conduct experiment on split 1 of UCF-101 and HMDB51. Our proposed method significantly outperforms the variants on each dataset. From the component analysis, we find that the three LSTM layers would perform worse than two LSTM layers, thus we use two LSTM layers as our architecture. Moreover, attention mechanism greatly improves the performance, and "attention-again" model gains the result of recognizing actions.

Table 2. Performance of LSTM variants and different components on first split of HMDB51 and UCF101. Classification accuracy is used as evaluation measure which represented as recognition accuracies (%).

Model	HMDB51	UCF101
LSTM	41.3	77.5
ALSTM	40.9	77.0
ConvLSTM	41.8	77.6
ConvALSTM	43.3	79.6
ConvLSTM + hierarchical LSTM (Three layers)	45.2	81.7
ConvLSTM + hierarchical LSTM (Two layers)	46.6	82.4
+ attention mechanism	50.9	84.1
+ "attention-again" model	52.6	85.8

Furthermore, in decode network, we only use the output of the last LSTM unit. To verify the performance of this small change, we compare with conventional methods which make softmax operation among the outputs of every LSTM unit. Then, we conduct experiments on UCF11 and use the same encode network (VGG). The results are shown in Figure 3. The loss is shown in Figure 3a, while Figure 3b represents accuracy. The red and blue lines in Figure 3 represent our method and

softmax method, respectively. From Figure 3, it is easily concluded that our method is more fluent than softmax. More importantly, the accuracy of our method is higher than softmax. Therefore, this structure of decode network can be proven.



(**a**) (**b**)

Figure 3. The comparison of the result of output of the last LSTM unit with softmax operation among the outputs of every LSTM unit: (**a**) comparison of loss; and (**b**) comparison of accurary. The red line represents our method, and blue line denotes softmax.

Comparison with LSTM-like architecture

In this sub-experiment, we list all state of the art methods that use a LSTM-like architecture for action recognition tasks by only using RGB data. To present the comparison completely and clearly, we elaborate on some factors, such as pre-training type and network architecture. In Table 3, our proposed method clearly performs the best among the LSTM-like architectures.

Table 3. State-of-the-art comparison with LSTM-like architectures. Classification accuracy is used as evaluation measure which represented as recognition accuracies (%). The **bold** text represents the best result.

Method	Pre-Train ImageNet	Networks GoogLeNet	VGG-M	VGG16	UCF11	HMDB51	UCF101
LRCN [27]	✓	-	✓	-	-	-	82.9
Soft-attention [25]	✓	✓	-	-	84.86	41.31	77
VideoLSTM [48]	✓	-	-	✓	-	43.3	79.6
JAN [50]	✓	-	-	✓	-	50.2	81.6
L^2STM [51]	✓	-	-	✓	-	*	83.2
Ours	✓	-	-	✓	**90.1**	**52.6**	**85.8**

The performance of RGB data from HMDB51 dataset are not shown in [51].

Comparison with state of the art

In addition to the LSTM-like comparison in Table 3, another state of the art algorithm comparison is presented in Table 4, which shows the results from RGB data.

It is obvious that our "attention-again" model outperforms these above methods significantly in three datasets. Although there are some other methods that combine the RGB data and flow data together to improve the performance in benchmark datasets, our model performs competitively against RACNN models in its category (models using RGB features only).

Table 4. Comparison with state of the art methods, the result from RGB data only. The **bold** text represents the best result.

Method	UCF11	HMDB51	UCF101
Soft attention model [41]	84.96	41.31	77
TSN [38]	-	51. 0	84.5
Average pooling [52]	-	52. 2	-
Attention-again model (our model)	**91.2**	**54.4**	**87.7**

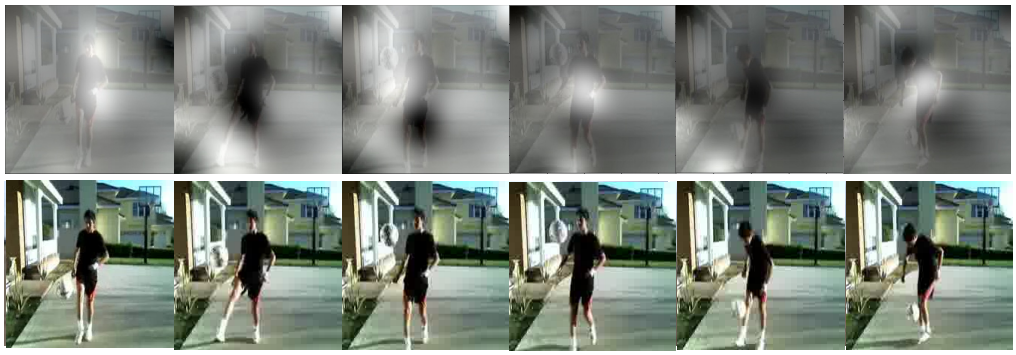
4.3.2. Qualitative Analysis

In this part, we show some test examples of what our model is paying attention to in the videos. As we know, attention model will teach network to focus on regions of interesting. Figure 4 shows the correct example in test on UCF-11 dataset, where we can see that our model mainly focuses on where the action is happening. Figure 4a is a horse-riding example; our model pays attention to the leg and head of horse, and the leg of human is also given attention. Figure 4b is soccer example; our model mainly attend to the person who is playing soccer as well as the ball. Figure 4c shows two man playing volleyball; our model focuses on the net and surrounding in addition to players. Wang et al. [53] suggest that the true meaning of an action lies in "'the change or transformation an action brings to the environment", e.g., kicking a ball. Thus, to recognize action, we not only need the feature from the human, but also need the feature of environment. In other words, we have to combine the foreground and background together to gain better performance. Fortunately, our model tackles this problem by using attention-again model. For example, in Figure 4b, our model not only attends to the person who is playing soccer, but also focuses on the playground. It is naturally believed that, when people judge whether the man in the image is playing soccer, the information of man with a ball is not enough; other contextual information such as playground plays a critical role. Similar to Figure 4b, Figure 4c shows that our model pays more attention to the net and surrounding information. Additionally, it is obvious to observe that the person in Figure 4b,c is not as highlighted as that in Figure 4a.

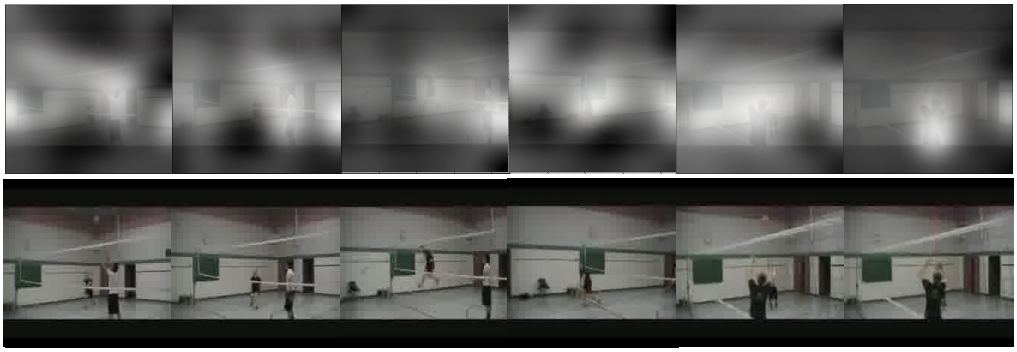
Simultaneously, we can also better understand failures of the model using the attention mechanism. Figure 5a shows the incorrect example (Ground truth: biking; Result:tennis), while Figure 5b is correct example of tennis and Figure 5c is correct example of biking. We could understand how the model performs the wrong result to some extent by reviewing Figure 5b,c. As Figure 5b shows, our model attempts to focus on the person playing tennis and some surroundings to recognize "'tennis", and the interesting regions are mostly located in the red blocks. In Figure 5c, our model pays attention to the position of bike, apart from region of where the person appears. Meanwhile, the interesting regions actually belong to not the red blocks but the yellow ones. However, we could find that the model only focuses on the person and cannot concentrate on where the bike appears in Figure 5a, thus our model incorrectly classifies the biking into tennis.

(**a**)

Figure 4. *Cont.*

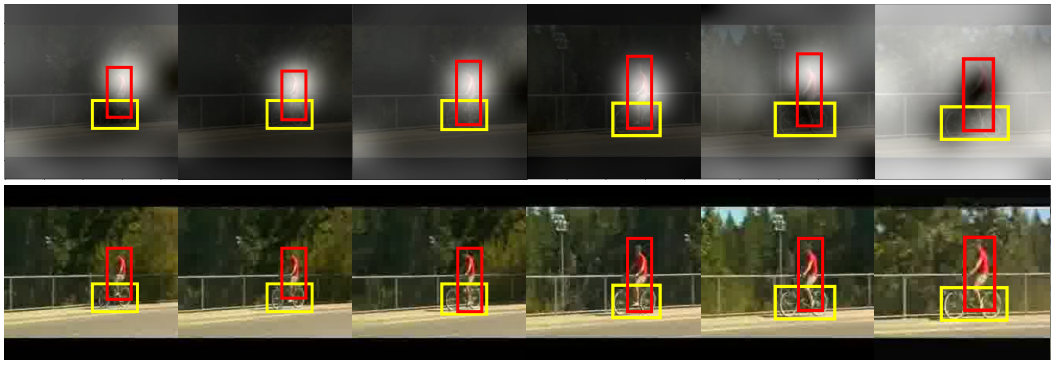


(**b**)

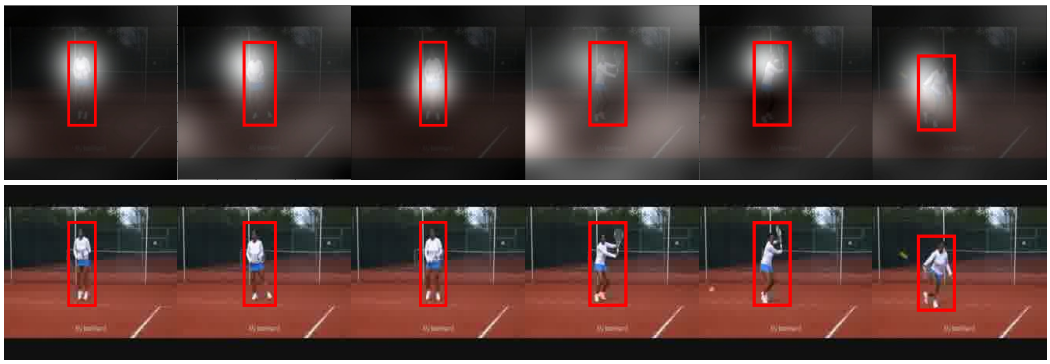


(**c**)

Figure 4. The illustration of some correct examples: (**a**) horse-riding example; (**b**) soccer example; and (**c**) volleyball example.

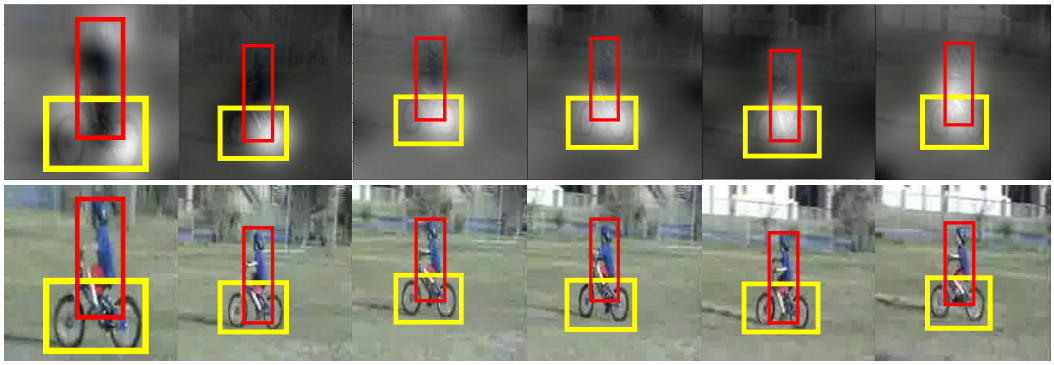


(**a**)



(**b**)

Figure 5. *Cont*.



(**c**)

Figure 5. The illustration of difference between incorrect example and correct example. The label of (**a**) is biking, but our model recognizes it as tennis, while (**b**) is correct example of tennis and (**c**) is correct example of biking. The red case labels the location of person and the yellow one represents the location of bike in image.

5. Conclusions

In this paper, we present a human-inspired model, namely "attention-again", for action recognition in videos. "Attention-again" model is a variant of the normal attention mechanism for classification and recognition of human activities in videos. Our framework consists of three major components: (1) convolutional feature extraction; (2) LSTM sequence modeling and attention model; and (3) the proposed "attention-again" model. We conduct several experiments on three benchmark datasets to demonstrate the great performance of model. From results shown in Section 4, we can easily conclude that our model outperform the state-of-the-art work with RGB data. Moreover, we visualize the region our model focuses on to understand how the model works. In future work, we will conduct our model using other state-of-the-art works and combine the advantages of different works to improve the accuracy of action recognition.

Acknowledgments: This work was primarily supported by National Natural Science Foundation of China (NSFC) with grant number 6117019.

Author Contributions: Haodong Yang, Shuohao Li, and Jun Zhang conceived and designed the network structure; Haodong Yang performed the experiment; Haodong Yang and Shiqi Chen analyzed the data; Haodong Yang and Jun Lei wrote the paper.

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

References

1. Yang, A.Y.; Iyengar, S.; Kuryloski, P.; Jafari, R. Distributed segmentation and classiffication of human actions using a wearable motion sensor network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
2. Jalal, A.; Kamal, S. Real-time life logging via a depth silhouette-based human activity recognition system for smart home services. In Proceedings of the IEEE International Conference on Advanced Video and Signal-based Surveillance, Seoul, Korea, 26–29 August 2014; pp. 74–80.
3. Jalal, A.; Sarif, N.; Kim, J.T.; Kim, T.S. Human activity recognition via recognized body parts of human depth silhouettes for residents monitoring services at smart homes. *Indoor Built Environ.* **2013**, *22*, 271–279.
4. Song, Y.; Tang, J.; Liu, F.; Yan, S. Body surface Context: A new robust feature for action recognition from depth videos. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *24*, 952–964.
5. Jalal, A.; Kamal, S.; Kim, D. Shape and motion features approach for activity tracking and recognition from Kinect video camera. In Proceedings 29th International Conference on Advanced Information Networking and Applications Workshops, Gwangiu, Korea, 24–27 March 2015; pp. 445–450.

6. Jalal, A.; Kamal, S.; Kim, D. A depth video sensor-based life-logging human activity recognition system for elderly care in smart indoor environments. *Sensors* **2014**, *14*, 11735–11759.
7. Jalal, A.; Kim, J.T.; Kim, T.S. Development of a life logging system via depth imaging-based human activity recognition for smart homes. In Proceedings of the International Symposium on Sustainable Healthy Buildings, Seoul, Korea, 19 September 2012; pp. 91–95.
8. Jalal, A.; Lee, S.; Kim, J.T.; Kim, T.S. Human activity recognition via the features of labeled depth body parts. In Proceedings of the 10th International Conference on Smart Homes and Health Telematics, Seoul, Korea, 22–24 June 2012; pp. 246–249.
9. Jalal, A.; Kim, S. Global security using human face understanding under vision ubiquitous architecture system. *World Acad. Sci. Eng. Technol.* **2006**, *13*, 7–11.
10. Jalal, A.; Rasheed, Y.A. Collaboration achievement along with performance maintenance in video streaming. In Proceedings of the IEEE Conference on Interactive Computer Aided Learning, Villach, Austria, 26–28 September 2007; pp. 1–8.
11. Jalal, A.; Kim, S. Advanced performance achievement using multi-algorithmic approach of video transcoder for low bit rate wireless communication. *ICGST Int. J. Gr. Vis. Image Process.* **2005**, *5*, 27–32.
12. Jalal, A.; Kim, Y. Dense depth maps-based human pose tracking and recognition in dynamic scenes using ridge data. In Proceedings of the IEEE International Conference on Advanced Video and Signal-based Surveillance, Seoul, Korea, 26–29 August 2014; pp. 119–124.
13. Jalal, A.; Kim, Y.; Kim, D. Ridge body parts features for human pose estimation and recognition from RGB-D video data. In Proceedings of the IEEE International Conference on Computing, Communication and Networking Technologies, Hefei, China, 11–13 July 2014; pp. 1–6.
14. Jalal, A.; Kamal, S.; Kim, D. Human depth sensors-based activity recognition using spatiotemporal features and hidden markov model for smart environments. *J. Comput. Netw. Commun.* **2016**, *2016*, 1–11.
15. Jalal, A.; Kamal, S.; Kim, D. Depth map-based human activity tracking and recognition using body joints features and self-organized map. In Proceedings of the IEEE International Conference on Computing, Communication and Networking Technologies, Hefei, China, 11–13 July 2014; pp. 1–6.
16. Jalal, A.; Kim, Y.H.; Kim. Y.J. Robust human activity recognition from depth video using spatiotemporal multi-fused features. *Pattern Recognit.* **2017**, *61*, 295–308.
17. Jalal, A.; Kim, Y.; Kamal, S. Human daily activity recognition with joints plus body features representation using Kinect sensor. In Proceedings IEEE International Conference on Informatics, Electronics and Vision, Fukuoka, Japan, 15–18 June 2015; pp. 1–6.
18. Kamal, S.; Jalal, A. A hybrid feature extraction approach for human detection, tracking and activity recognition using depth sensors. *Arabian J. Sci. Eng.* **2016**, *41*, 1043–1051.
19. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2015**, arXiv:1409.0473
20. Mnih, V.; Heess, N.; Graves, A.; Kavukcuoglu, K. Recurrent models of visual attention. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 8–13 December 2014; pp. 2204–2212
21. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the International Conference on Neural Information Processing Systems, Lake Tahoe, Nevada, 3–6 December 2012; pp. 1097–1105.
22. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 221–231.
23. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Fei, L.F. Large-scale video classification with convolutional neural networks. In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 1725–1732.
24. Srivastava, N.; Mansimov, E.; Salakhutdinov, R. Unsupervised learning of video representations using lstms. *arXiv* **2015**, arXiv:1502.04681.
25. Sharma, S.; Kiros, R.; Salakhutdinov, R. Action recognition using visual attention. *arXiv* **2015**, arXiv:1511.04119.
26. Hochreiter, S.; Schmid Huber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780.

27. Donahue, J.; Hendricks, L.A.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; Dar-rell, T. Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 8–10 June 2015; pp. 2625–2634.
28. Shi, Y.; Tian, Y.; Wang, Y.; Huang, T. Sequential deep trajectory descriptor for action recognition with three-stream cnn. *IEEE Trans. Multimed.* **2017**, *19*, 1510–1520.
29. Ng, J.Y.; 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 (CVPR), Boston, MA, USA, 8–10 June 2015; pp. 4694–4702.
30. Baradel, F.; Wolf, C.; Mille, J. Pose-conditioned Spatio-Temporal Attention for Human Action Recognition. *arXiv* **2017**, arXiv:1703.10106.
31. Cai, Z.; Wang, L.; Peng, X.; Qiao, Y. Multi-view super vector for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 596–603.
32. Zeng, W.; Luo, W.; Fidler, S.; Urtasun, R. Efficient summarization with read-again and copy mechanism. *arXiv* **2016**, arXiv:1611.03382.
33. Liu, J.; Luo, J.; Shah, M. Recognizing realistic actions from videos "in the wild". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 25–25 June 2009,
34. Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; Serre, T. HMDB: A large video database for human motion recognition. In Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 2556–2563.
35. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A Dataset of 101 human action classes from videos in the wild. *arXiv* **2012**, arXiv:1212.0402.
36. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 8–13 December 2014; pp. 568–576.
37. Shi, Y.; Zeng, W.; Huang, T.; Wang, Y. Learning deep trajectory descriptor for action recognition in videos using deep neural networks. In Proceedings of the IEEE International Conference on Multimedia and Expo, Turin, Italy, 29 June–3 July 2015; pp. 1–6,
38. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; Gool, L.V. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. *ACM Trans. Inf. Syst.* **2016**, *22*, 20–36.
39. Wang, H.; Schmid, C. Action recognition with improved trajectories. In Proceedings of the IEEE International Conference on Computter Vision, Sydney, Australia, 1–8 December 2013; pp. 3551–3558.
40. Mahasseni, B.; Todorovic, S. Regularizing long short term memory with 3D human-skeleton sequences for action recognition. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
41. Baccouche, M.; Mamalet, F.; Wolf, C.; Garcia, C.; Baskurt, A. Sequential deep learning for human action recognition. In *Human Behavior Understanding*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2011; pp. 29–39.
42. Jaderberg, M.; Simonyan, K.; Zisserman, A.; Kavukcuoglu, K. Spatial transformer networks. *arXiv* **2015**, arXiv:1506.02025.
43. Yeung, S.; Russakovsky, O.; Jin, N.; Andriluka, M.; Mori, G.; Li, F.F. Every moment counts: Dense detailed labeling of actions in complex videos. *arXiv* **2015**, arXiv:1507.05738.
44. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.C.; Salakhutdinov, R.; Zemel, R.S.; Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 2048–2057.
45. Zaremba, W.; Sutskever, I.; Vinyals, O. Recurrent neural network regularization. *arXiv* **2014**, arXiv:1409.2329.
46. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2015**, arXiv:1412.6980.
47. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv* **2016**, arXiv:1603.04467.
48. Li, Z.; Gavves, E.; Jain, M.; Snoek, C.G.M. VideoLSTM Convolves, Attends and Flows for Action Recognition. *Comput. Vis. Image Underst.* **2018**, 166, 41–50.

49. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *arXiv* 2015, arXiv:1506.04214.
50. Shi, Y.; Tian, Y.; Wang, Y.; Huang, T. Joint Network based Attention for Action Recognition. *arXiv* **2016**, arXiv:1611.05215.
51. Sun, L.; Jia, K.; Chen, K.; Yeung, D.Y.; Shi, B.E.; Savarese, S. Lattice Long Short-Term Memory for Human Action Recognition. *arXiv* **2017**, arXiv:1708.03958.
52. Girdhar, R.; Ramanan, D. Attentional Pooling for Action Recognition. *arXiv* **2017**, arXiv:1711.01467.
53. Wang, X.; Farhadi, A.; Gupta, A. Actions transformations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2658–2667.