Journal of Engg. Research Vol. 9 No. (3A) September 2021 pp. 115-133 DOI : 10.36909/jer.v9i3A.8383

Bidirectional LSTM with saliency-aware 3D-CNN features for human action recognition

Sheeraz Arif*,**, Jing Wang*, Adnan Ahmed Siddiqui**, Rashid Hussain** and Fida Hussain***

**Department of Information and Communication Engineering, School of Information and Electronics, Beijing Institute of Technology, Beijing, 100081, China*
***Department of Computing, Faculty of Engineering Science and Technology, Hamdard University, Karachi, Pakistan*
****School of Electrical and Information Engineering, Jiangsu University, Nanjing, China*
**Corresponding Author: sheeraz.arif@bit.edu.cn; sheeraz.arif@hamdard.edu.pk*

Submitted: 01/07/2019
Revised: 20/03/2020
Accepted: 25/03/2020

ABSTRACT

Deep convolutional neural network (DCNN) and recurrent neural network (RNN) have been proved as an imperious research area in multimedia understanding and obtained remarkable action recognition performance. However, videos contain rich motion information with varying dimensions. Existing recurrent based pipelines fail to capture long-term motion dynamics in videos with various motion scales and complex actions performed by multiple actors. Consideration of contextual and salient features is more important than mapping a video frame into a static video representation. This research work provides a novel pipeline by analyzing and processing the video information using a 3D convolution (C3D) network and newly introduced deep bidirectional LSTM. Like popular two-stream convent, we also introduce a two-stream framework with one modification; that is, we replace the optical flow stream by saliency-aware stream to avoid the computational complexity. First, we generate a saliency-aware video stream by applying the saliency-aware method. Secondly, a two-stream 3D-convolutional network (C3D) is utilized with two different types of streams, i.e., RGB stream and saliency-aware video stream, to collect both spatial and semantic temporal features. Next, a deep bidirectional LSTM network is used to learn sequential deep temporal dynamics. Finally, time-series-pooling-layer and softmax-layers classify human activity and behavior. The introduced system can learn long-term temporal dependencies and can predict complex human actions. Experimental results demonstrate the significant improvement in action recognition accuracy on different benchmark datasets.

Keywords: Action recognition; Convolutional neural network (CNN); Long-short-term-memory (LSTM); Recurrent neural network (RNN); Saliency.

INTRODUCTION

Human activity recognition has been highlighted as an intensely researched area due to its numerous applications, especially in social activity analysis, intelligent services, and cyber-physical systems. The types of applications include surveillance, health care, human-computer interaction, robot vision, and video retrieval. Furthermore, automatic indexing and classification of videos from an enormous amount of available digital video data are also extremely demanding. Action recognition's main goal is to identify behaviors or actions performed by one or more subjects/ actors from video sequences. Despite increasing interest in this field, state-of-the-art action recognition systems are still not accurate as human-level performance because of the intraclass variations, background clutter, partial occlusion, varying motion speed, and the high dimension of video data. Actions in videos can be represented by a series of video frames and can be recognized by assessing the features and information reside in multiple frames of the video. Unlike image classification, the information in videos is not limited to a single domain. For ideal video recognition, both the

static and motion patterns are extremely important. In addition, there may be possibilities of motion changing, and different actors in the scene may have different kinds of appearances. A complex activity typically consists of many subactivities. The behavior and temporal evolution of the associated subevents in video scenes can be complicated in videos having complex activities. Therefore, it is important to capture temporal dynamics information of these subevents for accurate recognition of complex activities.

To solve the aforementioned issues related to the recognition of human activities in videos, many researchers put their efforts by providing the extension in image action methods in the form extraction of handcrafted spatial and temporal features in videos, such as SIFT (Lowe, 2004) that was extended to 3D-SIFT (Scovanner et al., 2007). The study (Poppe, 2010) detected the spatial-temporal interest points and captured Histogram of Oriented Gradients (HOG) (Dalal & Triggs, 2005) and Histogram of Optical Flow (HOF) (Dalal et al., 2006) at each interest point. Wang and Schmid (2013a) discovered an effective way of detecting dens points in multiple scales in each frame, and then dense trajectories are extracted by performing sampling operation and tracking step on the dense points to capture the motion information. They also collected HOF, HOG, and Motion Boundary Histogram (MBH) (Wang et al., 2013b) at each dense or key point, and their overall combination is extremely effective in boosting the final performance. In Wang and Schmid (2013a), the camera motion is estimated by providing an improved version of dense trajectories. Zhou et al. (2015) simultaneously identified the spatial and temporal information of action by employing dense trajectories in the joint learning framework. However, these methods cannot deal with long-term action, and their performance degrades when applied to complex and realistic videos.

Recently, deep-learning neural networks (DCNN) perform well in many areas of computer vision and obtained prominent results for many computer vision tasks. Deep-learning methods use the phenomena of backpropagation to identify and extract the hidden pattern in input visual information, so the main features are extracted automatically without using any artificial method. From the many studies, it is also verified that deep learning features have shown much better performance than handcrafted features in the context of action recognition. As compared to image activity classification, human action in videos consists of features related to visual appearance that dynamically change over time. Several methods have been proposed based on the extension of DCNN from images to videos. Ji et al. (2013) performed the extension on the traditional 2D CNN methods and developed a 3D-CNN framework to capture the Spatio-temporal representation using input video frames, and this is later improved by convolution 3D (C3D) (Trans et al., 2015). Simonyan and Zisserman (2014) introduced a two-stream ConvNets method, which incorporates both spatial (appearance) and temporal architectures and trained neural networks using ImageNet network. Although deep CNNs are extremely powerful in automatically learning of features and have obtained great achievements in multimedia understanding, however, deep learning-based ones require large training datasets and largely ignore the temporal characteristics of the video data.

More recently, the recurrent neural network (RNN) provides the ability to the networks to extract and find hidden dynamics in time-space visual data and sequentially process the data. However, these RNN based networks encounter two problems, i.e., vanishing gradient problem and the fact that there are a large number of complex calculations. This problem is addressed by long-short-term-memory (LSTM) (Hochreiter, 1997), which can update and forget the hidden states by using combinations of forget gates, memory units, input, and output gates. LSTM networks tend to identify long-term temporal dynamics and preserve sequential information from input data. This is the reason why LSTM is the potential candidate to solve different issues in sequential modeling tasks such as video description, human action recognition, machine translation, and speech recognition. Many investigators introduced various flavors of CNN-LSTM networks and different variations of LSTM like architectures, for example, bidirectional and multilayer based LSTM for processing of video data. However, in these pipelines, modeling of long-range dependencies is still problematic, and capturing complex and rich motion dynamics from a sequence of frames is not ideal in videos, where the activity is performed by multiple subjects. Besides, it is also extremely important to consider semantic and salient features, rather than representing the entire information of frame in a static way.

To address the aforementioned problems in visual modeling tasks, we introduce a new lightweight architecture integrating the deep neural network with salient action motion features. C3D layers are constructed to capture visual

features, while the following bidirectional LSTM layers are applied to handle temporal dynamics. This research effort is the continuous work of our published preliminary work (Arif et al., 2019), which proposed a simple and effective cyclic training model to capture motion maps using only RGB frames. This method can integrate useful information into a single motion map from each video frame, and unidirectional LSTM is used as an encoder and decoder for the processing of all spatial-temporal features. While, in this proposed research work, we adopt a two-stream CNN model and successfully utilized two-stream (RGB and saliency video), we performed one modification in this model, which is the introduction of a saliency-aware video stream instead of the optical flow field to avoid computational complexity. In addition, to extract long-term temporal dependencies, we utilize bidirectional LSTM and obtain better recognition accuracy. The introduced pipeline can solve the issues related to long-term motion dependencies without any computational complexity. The main contributions of the proposed model can be highlighted as follows:

1- We design a new pipeline, which extracts both static and salient motion features by intelligently integrating bidirectional LSTM with 3D-CNN and does the action classification.

2- To capture the important subjects or multiple subjects in video shot, we introduce a saliency stream, which enhances the performance of the C3D network and highlights the salient regions.

3- We utilized two-stream (RGB and optical flow stream) 3D-convolutional network architecture with one modification. We replaced the optical flow stream by salience-aware video stream and achieve better results.

4- We introduced the joint-optimization-module, which comprises a fully connected (FC) layer along with a SoftMax layer. This practice optimizes our classifier and captures the internal relationship among feature vectors.

5- The performance and effectiveness of our introduced approach are tested on two public datasets and obtained at par results.

RELATED WORKS

Over the past years, various models have been discovered based on handcrafted and deep-learning in the context of accurate human action recognition. The conventional research works mostly rely on handcrafted features techniques, which are only specific to simple and non-realistic human actions present in videos. As the proposed framework is based on deep neural and RNN networks so this section only highlights the review research works based on CNN and RNN.

In recent years, many variants of deep learning approaches have been developed for robust recognition of human activity in videos and reached remarkable performance for many computer vision tasks. (Ji et al., 9) captured spatial and temporal features from action videos by applying 3D-convolutional filters on a sequence of frames in the video. (Karpathy et al., 2014) applied a deep learning method to the sequence of frames and achieved frame-level temporal relations among the frames by pooling operations using a different combination of fusion methods; however, these methods can only obtain the marginally better performance than single frame baseline methods. (Simonyan & Zisserman, 2014) utilized the two-stream deep CNN by incorporating two components i.e. RGB stream and pre-computed optical flow stream to extract temporal and spatial features. However, this architecture is only limited to learn short-term motion transitions and offer some computational complexity. However, the additional optical flow stream provides strength to the motion features and improved recognition accuracy. (Tran et al., 2015) proposed an efficient method, which can learn the temporal information by using deep neural networks without computing the optical flow features at the initial stage. However, C3D is only suitable to cover a short duration of video sequences. (Wang et al., 2016a) presented temporal segment networks (TSN) model, in which long-term temporal architecture can be modeled by applying sparse temporal sampling. (Feichtenhofer et al., 2016] find several techniques to fuse different CNN features and obtained spatial-temporal information from two-streams network i.e. spatial (appearance) and optical flow streams. However, these CNN based methods extract features related to only visual appearance and unable to model long-term temporal dependencies. Moreover, CNN based methods do not consider the internal relationship between the temporal and spatial and domains.

RNNs have been considered as the potential candidate to learn long-term temporal dependencies among video frames for video-based human activity recognition. RNNs can give the ability to network for accurate processing and finding of hidden dynamics in video-based data. This kind of system process the data in sequential order in such a way that it generates input data using its previous hidden status i.e. s_{t-1} at each time span t and obtains new data i.e. x_t. Many of the existing new models (Zaremba et al., 2014; Veeriah et al., 2015; Yue-Hei et al., 2015; Wu et al., 2015; Donahue et al., 2017) introduced a kind of recurrent network by combining the CNNs and RNNs architectures for human activity recognition and obtained impressive performance. However, they are not appropriate to overcome the vanishing gradient problem and there is a large number of complex calculations of parameters. LSTM (Donahue et al., 2017; Yue-Hei et al., 2015; Srivastava et al., 2015; Mahasseni et al., 2016) provides the solution to this problem, and its memory units and forget gates provide the LSTM tendency to extract long-range dependencies and preserve its sequential information over time by using memory units. (Hochreiter, 1997), first coined the term LSTM, now LSTMs have proven to be extremely successful for various sequential modeling tasks such as visual description, machine translation, speech recognition and achieved impressive performance. These networks extract the high-level features from the final fully connected layer of CNN and then these features are fed to the LSTM. LSTM unit consists of different memory units and multiplicative gates, which can control the identification of long-term hidden patterns as well as control the propagation of noise/error signal through the network.

Meanwhile, the traditional LSTMs do not consider the spatial correlation in the video frame and motion dynamics are not well presented. VideoLSTM (Li et al., 2018) and ConvLSTM (Xingjian et al., 2015) address this limitation by introducing convolutional operations on feature maps generated by convolutional layers. However, these networks are only ideal for activities with little movement (stationary motion) because only spatial patterns along the temporal domains are considered in these models. This issue has been addressed by some attention-based RNN (Yeung et al., 2015 & Sharma et al., 2015), in which attention techniques have been incorporated into LSTM. (Yeung et al., 2018) introduced LSTM based attention model by emphasizing the key temporal segments and (Sharma et al., 2015) identified some significant spatial locations at each time step of LSTM but attention mechanism largely ignores the spatial and motion cues. Attention in (Li et al., 2018 & Wang et al., 2016d) integrated motion and appearance cues into a unified architecture. However, attention factors still lack a rich spatial-temporal context among the video frames and this practice is insufficient to make a reasonable prediction.

Most of the researchers have also presented multilayer and bidirectional LSTM (Graves et al., 2005; Ullah et al., 2017; Yeung et al., 2018) for processing the sequential data and analyzed the complicated visual patterns which cannot be easily identified by single directional traditional LSTM unit. However, the aforementioned pipelines fail to extract complex motion information from adjacent frames or maybe multiple frames. Moreover, activities performed by multiple actors or subjects are difficult to recognize.

To address the above issues, we introduce a novel end-to-end framework by merging C3D and bidirectional LSTM in the presence of RGB stream and newly introduced saliency-aware stream. The saliency-aware stream improves the significance of foreground areas and captures important objects from video frames. Moreover, the introduction of two layers of stacked LSTM in both forward and backward directions are extremely useful to identify the complex hidden sequential information in the features as well as the processing of lengthy videos. Experimental results show significant improvement in action recognition.

THE PROPOSED FRAMEWORK

We first explain the overall pipeline of the introduced method and then define the details of the training method. As the overall methodology is depicted in Figure 1, our approach consists of two streams i.e. RGB stream for extracting appearance features and saliency stream for salient motion features. First, we apply a saliency-aware method to produce saliency-based action videos. Next, our novel end-to-end framework is designed by leveraging 3D-CNN and bidirectional LSTM, which effectively characterize long-term complex motion in videos. Then the action prediction is carried out by a time-series-pooling-layer and a softmax-layer. The proposed approach analyses complex hidden patterns in video data for every frame, which cannot be identified by a simple single LSTM unit.

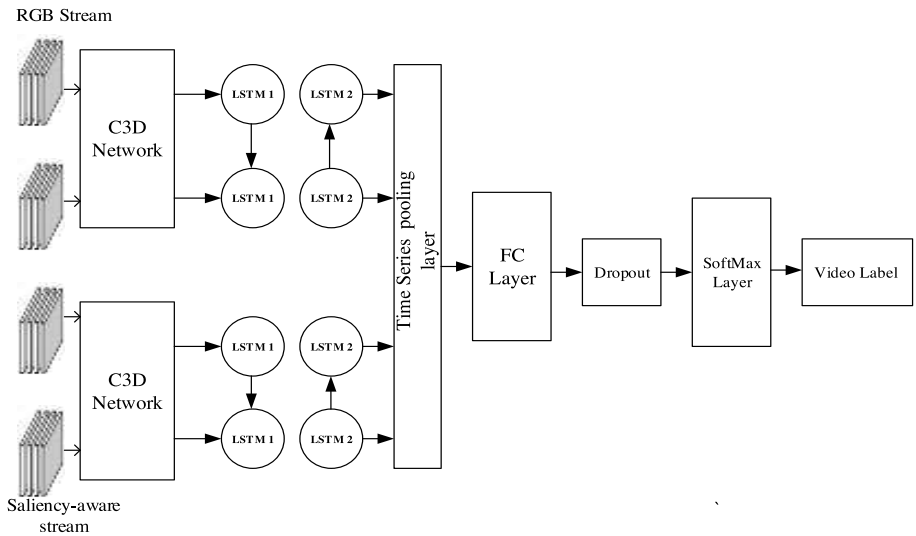

Figure 1. The overall pipeline of the proposed model.

Generation of saliency-aware videos

The main aim of saliency detection is to highlight the salient regions, which can require human visual attention in a video frame. The saliency detection mechanism has the characteristic of high scalability and has been widely used to access many other machine vision tasks and applications include object localization, frame cosegmentation, and action recognition. Compared with cosegmentation, saliency detection is closely related to human visual attention aims at discovering the important and common regions from the given image. Saliency detection approaches outputs probability maps to highlight the cosaliency regions, while cosegmentation methods directly segment out the common image regions to generate binary maps. Thus, saliency detection can be useful to obtain prior information to similar foregrounds for cosegmentation.

Figure 2. Saliency maps of the input video frames.

Many researchers also capture object proposals in every video frame and utilize the object segmentation for the selection of object regions by using both appearance and motion information to compute the objectness scores. Motivated by the success of the saliency method, this work introduces the saliency technique in our pipeline to address the issues related to the recognition of complex motion activities. We adopt guidelines provided in (Wang & Shen, 2015) to capture the saliency-based map S for each frame of the video level sequence. Next, the saliency mask $s_{i,j}$ of the corresponding frame is computed by binarizing the saliency-aware map S on the following basis.

$$\left\{\begin{array}{l} s_{i,j} = 0; \text{if } s_{i,j} < mean(S) \\ s_{i,j} = 1; otherwise \end{array}\right\} \quad (1)$$

Next, to enhance the importance of foreground information i.e. actors and objects, we weaken the background region by conducting halves of the RGB values where $s_{i,j} = 0$. So, in this way, we can get the subject saliency information and the resulted video is known as saliency-aware video *Vs*. Figure 2 illustrates some examples of video frames with their respected saliency-aware maps by using our method.

Feature extraction from two-stream 3D-CNN

Deep Convolutional Networks have strong capabilities to learn discriminative power and extract the hidden sequential patterns in visual sequence-based data. According to the visual recognition mechanism, it has become the natural choice for action recognition to extract static/spatial features using DCNN and capture the motion information using LSTM. In our pipeline, we also successfully exploit the combination CNN and LSTM to capture clip-level salient motion information and frame-level spatial information from video data. Likewise, two-stream 3D ConvNets, we utilize two types of input streams, namely RGB stream and saliency-aware video stream. We just make one modification in the architecture, and we use a saliency-aware video stream instead of the optical flow field since its computation is too complicated and optical flow is only useful for short-term motion dynamics. RGB frame mainly represents static appearance at a particular period, while the saliency-based stream captures the salient motion information between consecutive frames. Both streams are given input to the C3D network and high-level semantic information is collected at the higher layers of the deep convolutional network. Next, the bidirectional LSTM network identifies long-term motion dependencies from the features extracted by C3D. So, our proposed coupled CNN-LSTM architecture can extract spatial as well as temporal dynamics in a good manner.

In our approach, we utilize C3D network architecture, which is considered as well-suited network for modelling sequential inputs. It can preserve and summarize the local temporal information within a video unit. C3D network comprises a series of 3D convolution and pooling operations to process both spatial and temporal dimensions. C3D network comprises five pooling layers, eight 3D convolution layers followed by two fully connected layers and finally softmax layer for action prediction. We make it simple by defining 3-dimensional convolution as C (k, d, f, s_t, s_p) and pooling kernels as P (d, f, st, s_p), where k represents the number of kernels, temporal depth can be denoted by d, f is considered as spatial size and s_t and s_p are the temporal stride and spatial stride respectively. The complete information of C3D architecture can be represented in Table 1.

Table 1. The architecture of the C3D network, illustrating the information about pooling and convolutional layers.

Layers	**Conv1a**	**Conv2a**	**Conv3a**	**Conv3b**	**Conv4a**	**Conv4b**	**Conv5a**	**Conv5b**
Filter	3 x3 x 3	3 x3 x 3	3 x3 x 3	3 x3 x 3	3 x3 x 3	3 x3 x 3	3 x3 x 3	3 x3 x 3
Stride	1 x 1 x 1	1 x 1 x 1	1 x 1 x 1	1 x 1 x 1	1 x 1 x 1	1 x 1 x 1	1 x 1 x 1	1 x 1 x 1
Channel	64	128	256	256	512	512	512	512
Ratio	1	1/2	1/4	1/4	1/8	1/8	1/16	1/1
Layers	**Pool-1**	**Pool-2**	**Pool-3**	**Pool-4**	**Pool-5**	**Fc-6**	**Fc-7**	
Kernal	1 x 2 x 2	2 x 2 x 2	2 x 2 x 2	2 x 2 x 2	2 x 2 x 2	-	-	Softmax Layer
Stride	1 x 2 x 2	2 x 2 x 2	2 x 2 x 2	2 x 2 x 2	2 x 2 x 2	-	-	
Channel	64	128	256	512	512	4096	4096	
Ratio	1/2	1/4	1/8	1/16	1/32	-	-	

For the t[th] frame of video, the feature maps $CM_t^+ \in_{\mathbb{R}}^{d_{cm} \times K \times K}$ can be extracted from the framework of two-stream C3D,

$$CM_t^+ = \{CM^+(t,1),...,CM^+(t,K^2)\}, \quad (2)$$

where either appearance (a) and motion (m) of the video stream can be represented by +, CM_t^+ contains d_{cm} feature maps with K x K spatial dimensions and also feature map can be defined as a set of feature vectors at different spatial locations, such as $CM^+(t,k) \in_{\mathbb{R}}^{d_{cm}}$ where k = 1, …, K^2. In the proposed model, we consider the output of first fully-connected-layer *fc6* and automatically extract spatial-temporal features φ_t^+ from all captured convolutional feature maps, i.e., $CM_1^+,....,CM_T^+$ where + denotes both input streams. So, each video clip can be defined as a sequence of extracted features of sampled frames, i.e., $\{\varphi_t^+\}_{t=1}^T$. Then, the bidirectional LSTM network is used to model this sequence and extracting long-term dependencies.

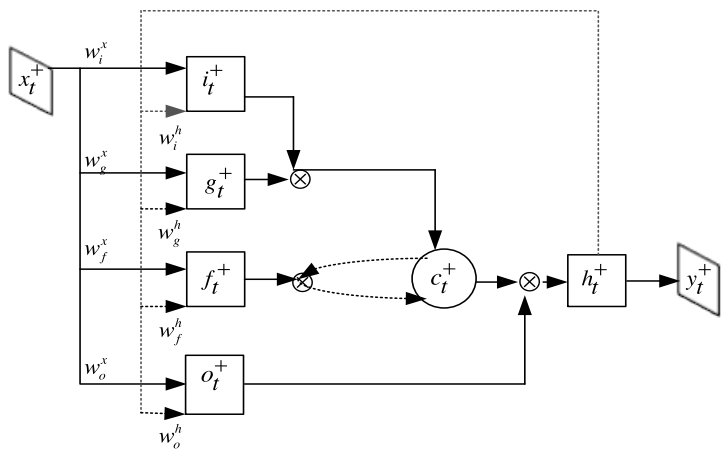

Figure 3. The LSTM unit.

Recurrent neural network

RNN is the natural way to process and encode temporal patterns of extracted semantic features from video frames. Visual information is presented in sequences of video frames, so we can easily understand the dynamics and context of action present in videos. RNNs tend to interpret these sequences but only limited to the short-term video sequences because it does not memorize the input frame sequence. To deal with vanishing issues LSTM has been designed which provides the ability to extract long-term contextual features from the temporal sequence. LSTM can extract the long-term temporal dependencies and stores sequential information over time. Moreover, LSTM keeps the earlier states of the input when trained with backpropagation through time. The LSTM architecture comprises memory units input, control gates and output gate, which can control the identification of long-term sequential patterns. The sigmoid unit controls the adjustment of the gates so that during training it can be opened and closed.

The structure of an LSTM cell is illustrated in Figure 3. We can define input vector as x_t^+ at time t, and h_t^+, c_t^+, and y_t^+ can be termed as a hidden state, cell state and output at time t, respectively. The hidden state h_t^+ is extremely important for output y_t^+, while hidden state h_t^+ depends on its previous state as well as the cell state c_t^+. At any time, LSTM can process and maintains the information over time by writing and reading to its internal memory. LSTM neuron has a forget gate f_t^+ to clear the un-important record from the memory cell, an input gate i_t^+, a memory cell c_t^+, and an output gate o_t^+. So, LSTM utilizes these gates at each time step t to read, write, or reset the memory cell. This technique benefits the cell to retrieve and store information in different steps. Equations 3 to 8 demonstrate the mechanism of temporal modelling carried out in the LSTM unit.

$$i_t^+ = G\left(W_{i_+}^x x_t^+ + W_{i_+}^\varphi \varphi_t^+ + W_{i_+}^h h_{t-1}^+ + b_{i_+}\right), \tag{3}$$

$$f_t^+ = G\left(W_{f_+}^x x_t^+ + W_{f_+}^\varphi \varphi_t^+ + W_{f_+}^h h_{t-1}^+ + b_{f_+}\right), \tag{4}$$

$$o_t^+ = G\left(W_{o_+}^x x_t^+ + W_{o_+}^\varphi \varphi_t^+ + W_{o_+}^h h_{t-1}^+ + b_{o_+}\right), \tag{5}$$

$$g_t^+ = \tanh\left(W_{g_+}^x x_t^+ + W_{g_+}^\varphi \varphi_t^+ + W_{g_+}^h h_{t-1}^+ + b_{g_+}\right), \tag{6}$$

$$c_t^+ = f_t^+ \odot c_{t-1}^+ + i_t^+ \odot g_t^+, \tag{7}$$

$$h_t^+ = o_t^+ \odot \tanh(c_t^+), \tag{8}$$

where LSTM cell has some parameters such as bias term and weights which can be denoted *b* and *W* respectively. Whereas, appearance or motion can be represented by symbol +. A sigmoid function can be given by *G*, element-wise-multiplication can be denoted by a symbol ⊙ and *tanh* as the activation function. The long-term dependencies can be extracted after step by step computing the cell state and output. The feature vector x_t^+ is the input to LSTM which is collected from the first fully-connected-layer of the C3D architecture. All the un-necessary stored information can be cleared using forget gate and the output gate stores the information about the current and next step. There is a recurrent unit g_t^+, which can be calculated from previous state h_{t-1} and current input frame using the activation function *tanh*. The memory cell c_t^+ and tanh activation function are used to compute the hidden state of the LSTM step.

Bidirectional LSTM

Single LSTM unit is not capable of identifying the complex hidden sequence pattern. Therefore, in our introduced pipeline, the multi-LSTM scheme has been adopted by stacking multiple layers of LSTM unit to capture long-term dependencies in visual sequential data. One of the main benefits of multilayer LSTM is that it provides a flexible and expandable temporal receptive field at both the input and output side of an LSTM unit. This practice is indeed extremely helpful to achieve refined action predictions and also provide a direct path for referencing previously observed frames. Figure 4 presents a two-layer LSTM structure, in which Layer-1 gets the data from input vector x_t^+ and the input of layer-2 is from its previous time step h_{t-1}^2 and the output of the current time step of layer-1 h_t^1.

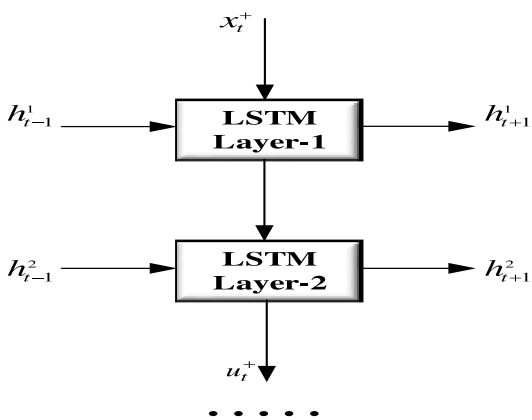

Figure 4. The working of two-layer LSTM architecture.

In bidirectional LSTM, the output is dependent on both previous and upcoming frames in the sequence at time *t*. In our method, we adopt two-layer LSTM architecture for backward and forward directions. The combined output is calculated based on hidden states of both backward and forward directions. The biases and weights and are adjusted

with the help of backpropagation and validation, and the cost function is computed after the output layer. The operations of LSTM can be computed from Equation 3 to 8, but for multilayer LSTM, we only add information of layer *l* to the superscript of each h_t^+, c_t^+, and o_t^+. So the state of the layer can be calculated by using Equation 9. Due to the b-directional processing of the layers, the output of a frame at *t* time step is computed from the upcoming frame at time *t+1* and previous frame at time step *t-1*.

$$h_t^{(+l)} = o_t^{(+l)} \odot \tanh(c_t^{(+l)}) \tag{9}$$

Optimization

Next, we apply our contextual feature φ_t^+ to LSTM through Equations (3)-(8), to extract h_t^a and h_t^m which are the current hidden states of both appearance and motion components. Both these components can be combined to train our model in an end-to-end manner into softmax as follows:

$$\hat{y} = soft\max(W_a h_t^a + W_m h_t^m + b_{am}), \tag{10}$$

where $\hat{y}$ is the prediction vector for a class and $\{W_a, W_m, b_{am}\}$ are the weight parameters. The vectors $\hat{y}$ interpret the un-normalized log probability of each human action category. However, it can be expected that the initial classifications of action may have some degree of uncertainty and prediction is not that accurate. To address this fact, we can modify the loss function as shown in Equation (11). Given training action label y_t, L_{action} is the cross-entropy loss between prediction label vector y_t and its true label vector $\hat{y}$ is given as follows:

$$L_{action=} -\sum_{t=1}^{T}\sum_{n-1}^{N} y_{t,n} \log \hat{y}_{t,n}, \tag{11}$$

where *N* is the class action categories, *T* denotes the of total time steps.

Action classification process

For the activity classification, we follow the joint optimization by using a time-series-pooling-layer and a Softmax-layer. Spatial-temporal features are aggregated by the time-series-pooling- layer, these features are the output of two stacked LSTM layers. We utilize a special dropout operation between FC and a softmax-layer, which prevents our approach from over-fitting. So in this way, joint optimization can be performed to get the prediction of human activity based on final scores.

EXPERIMENTS AND EVALUATION

For the verification of the effeteness of our proposed model, the series of extensive experiments have been conducted in the context of human activity recognition. Two well-known benchmark human action datasets: UCF101 (Soomro et al., 2012) and HDMB51 (Kuehne et al., 2011) have been used. First, we will give the descriptions of each dataset following by experiment setting/implementation detail as well as the experimental results and discussion.

Benchmark datasets

The UCF101 dataset is derived from the UCF50 dataset, it has a total of 101 different action classes and each class has at least 100 video clips. This dataset contains a total of 13,320 video sequences. Most of the videos are uploaded by users which are real and clean videos with camera motion, cluttered background, and illumination. The dataset can be partitioned into two sets of videos i.e. training set which contains 9.5K videos in total and testing set of videos containing 3.8K videos. We follow the guidelines provided in the THUMOS-3 challenge (Jiang et al.,2013) and follow the three splits of training/testing for evaluation of recognition performance by computing average recognition accuracy using three splits of the dataset.

The HDMB51 dataset comprises different realistic videos picked up from different resources such as YouTube and Google video. Dataset has 6,766 video clips with 51 different action classes and each action class has around 100 video clips. We adopt the three-test splits pattern evaluation schemes for experimental setting, and we keep 70

action categories for training and 30 sequences for testing evaluation. The final average recognition accuracy has been measured over these three-splits of this dataset.

Implementation details

To implement our work, we use a single node of GeForce GTX Titan Z CPU with RAM of 8 GB. Caffe toolbox (Jia et al., 2014) is used for extraction of deep features and TensorFlow for the implementation of bidirectional LSTM. We use the UCF-101 dataset to train the C3D model initially because UCF-101 is a larger dataset than the HMDB51 dataset, and then the learned model is transferred to HMDB51 for feature extraction. We process the video as video frames because of the nature of our network. To improve the generalization performance of our framework, we extract a constant number of frames/sec according to the video frame rate. **We extract 16 video frames from each video clip using an overlap of 8 frames. Each video clip is individually fed into the C3D network stream with a frame size of 16 × 112 × 112. We perform the data augmentation on the training video samples by cropping video frames from its corner to extracts salient regions from the corners and the center of the video frames. To make 16 frames of the video clip, we use the procedure of mirroring. In this way, we can prevent our network from bias towards the center area of the input data. The video frames are horizontally flipped with a 50% probability. We utilize stochastic gradient descent (SGD) with a mini-batch size of 30 samples in our experiments for the training of data. We do not perform truncating gradients for the optimization of our overall C3D network and update the weights of each data stream based on the full gradient.** The output of fc6 is used as input to BD-LSTM. FC6 is the first fully-connected-layer of C3D having the dimension of 4096.

To learn the network weights, we adopt mini-batch stochastic-gradient (SGD) with a momentum setting of 0.9 and a weight decay of 5×10^{-4}. We consider 16 frames/video as one sample for each RGB and saliency stream. The starting learning rate for RGB stream is set as 10^{-3} and then divided by 10 after iterations 20K and 40K, and we keep maximum iterations as 50K. For the saliency-aware stream, we keep the initial learning rate of 10^{-3} and decreased to 10^{-4} after 30K iterations. Then it is reduced to 10^{-5} after 50K iterations and the maximum training iterations are 60K. To train our coupled C3D-LSTM framework, we set the starting learning rate as 10^{-3} and is divided by 10 every after 10K iterations, and training is stopped at 30K iteration. Within LSTM all superposition kernels and convolutional kernels are used with the setting of 3×3.

Results and discussions

To discover the significant properties and effectiveness, we tested the performance of our model on UCF 101 and HDMB51 datasets. To explore the different aspects of our method, the number of experiments are carried out, their details with discussions are given in subsequent sections.

Exploration experiments

Since our method supports different kinds of modalities, we study the exploration of different network structures with their combinations and also analyze the complementary properties of the RGB and saliency-based stream on the HDMB51 dataset. Table 2 demonstrates the obtained results. We carry out different experiments with pure 3D CNN without single layer LSTM in the presence of RGB and saliency-aware streams. First, we examine the performance of 3D CNN on the RGB and saliency stream individually and also with their combinations. We can see that saliency stream 3D CNN gets the worst performance than using RGB stream with a 3D CNN model but obtained better results with a combination of both streams. The possible reason is that some activities become ambiguous when only considering the saliency stream. We also test the performance of 3D CNN in the presence of a single-layer LSTM module and results indicate that the saliency stream does not improve the recognition results even working with LSTM. However, in the presence of both streams, recognition accuracy increase by a significant margin. The underlying reason is that LSTM considers more information to drive correct prediction when fusing RGB and saliency stream. From the results, it also reflects that the addition of the time-series-pooling layer at the bidirectional LSTM

output obtains slightly better recognition results than in the absence of time-series-pooling layer and accuracy is boosted by 0.8 %. We can thus conclude that the combination of some modalities has a distinct advantage over the single modality based approaches and RGB and saliency streams provide complementary information for the action recognition tasks. Thus, achieved 73.9% accuracy verifies that the recognition performance can be improved by complementary properties of both streams and interrelation between salience regions and video clips are extremely important for accurate action recognition.

Table 2. Exploration results of different combinations of network structure on the HDMB51 dataset.

Different Model	Accuracy (%)
RGB Stream+3D CNN	60.2%
Saliency Stream+ 3D CNN	59.0%
(RGB + Saliency) Stream + 3D CNN	64.9%
RGB Stream+3D CNN + Single LSTM	67.9%
Saliency Stream+ 3D CNN+ Single LSTM	66.8%
(RGB + Saliency) Stream + 3D CNN+ Single LSTM	71.1%
Proposed method (without Time_pooling)	72.9%
Proposed method (with Time_pooling)	73.9%

Evaluation of videos with complex movement and multiple subjects

In this section, we evaluate our approach to videos with complex movement and also consider those videos in which activities are performed by multiple subjects. We use UCF 101 dataset with all its three splits. Except for 101 action classes, UCF 101 dataset has some coarse definitions with different sets of activities such as human and human interaction, human and object interaction and videos with complex body motion only.

Table 3. Performance evaluation on videos with complex movements and multiple subjects of the UCF-101 dataset.

Video Types	ConvLSTM	L2STM	Our method
Coarse Definition			
Body Motion Only	88.1	88.6	91.2
Object-Human Interaction	76.0	86.7	90.2
Human-Human Interaction	89.1	95.4	95.8
Multiple Subjects			
Soccer Penalty	87.2	93.2	93.9
Military Parade	86.3	92.8	92.1
Horse Race	88.1	92.1	92.3
Basket Ball Shooting	88.0	92.2	93.4
Cricket Bowling	87.9	90.2	93.5
Complex Videos			
Salsa Spins	86.0	100.0	98.9
Pizza Tossing	21.2	72.7	82.2
Playing Tabla	81.6	100.0	100.0
Ice Dancing	97.8	100.0	100.0
Mixing Batter	55.6	86.7	93.2
Breast Stroke	82.1	90.3	94.7

We also select some action classes in which activities are performed by multiple subjects and videos with complex actions and recognition performance is listed in Table 3. We compare the proposed method with ConvLSTM (Xingjian et al., 2015) and L2STM (Sun et al., 2017) as they also use two streams of input to the network and some of the results on these selected categories are available. We can observe from the table that, for various action classes and coarse definitions of the dataset, the proposed model obtains better activity recognition results specifically on an object and human interaction ↑ 3.7 ~ 14.2 and human to human interaction ↑ 0.7 ~ 6.3. In the case of body motion, both methods ConvLSTM and L2STM perform similarly but our method obtains better results as compared to both methods. On the other hand, the middle part of the table shows the comparison on some selected categories in which activities are performed by multiple subjects and our method obtains better results than L2STM except only one category i.e., Military Parade in which L2STM performs marginally well. The bottom portion of the given table demonstrates the accuracies for complex action classes in which movements of subjects and objects are extremely complex and fast.

We can also see from the table that the percentage of performance is increased as the action becomes more complex in videos. This suggests that both spatial and salient information are necessary for the temporal domain and the relationship between salient regions and video clips are extremely crucial for the ideal performance of action recognition.

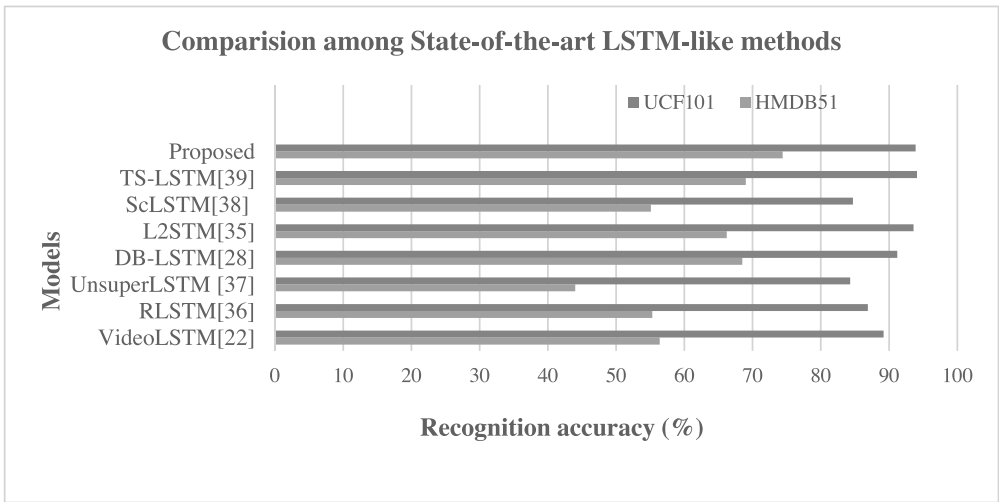

Figure 5. Performance comparison analysis among LSTM-like architectures.

Comparison among LSTM-like network architectures

In Figure 5, the proposed approach is compared with all those existing models which have similar architecture as LSTM for human activity recognition. All experiments are carried out on the first split of the UCF-101 and HMDB51 dataset. In the UCF-101 dataset, most of the videos having realistic actions are performed in real life, and the HMDB51 dataset consists of actions with complex body movements. From the results, we can notice that our introduced method achieves better recognition performance than other LSTM-like architectures. Our method obtains similar results as L2STM and TS-LSTM (Ma et al., 2019), however, in the case of the HMDB51 dataset, our model appears the best amongst the all listed LSTM-like architectures. This indicates that the proposed method can learn long-term complex motion dynamics in videos and the presence of a saliency-aware stream is certainly more helpful to extract salience regions that are more discriminative for action classification.

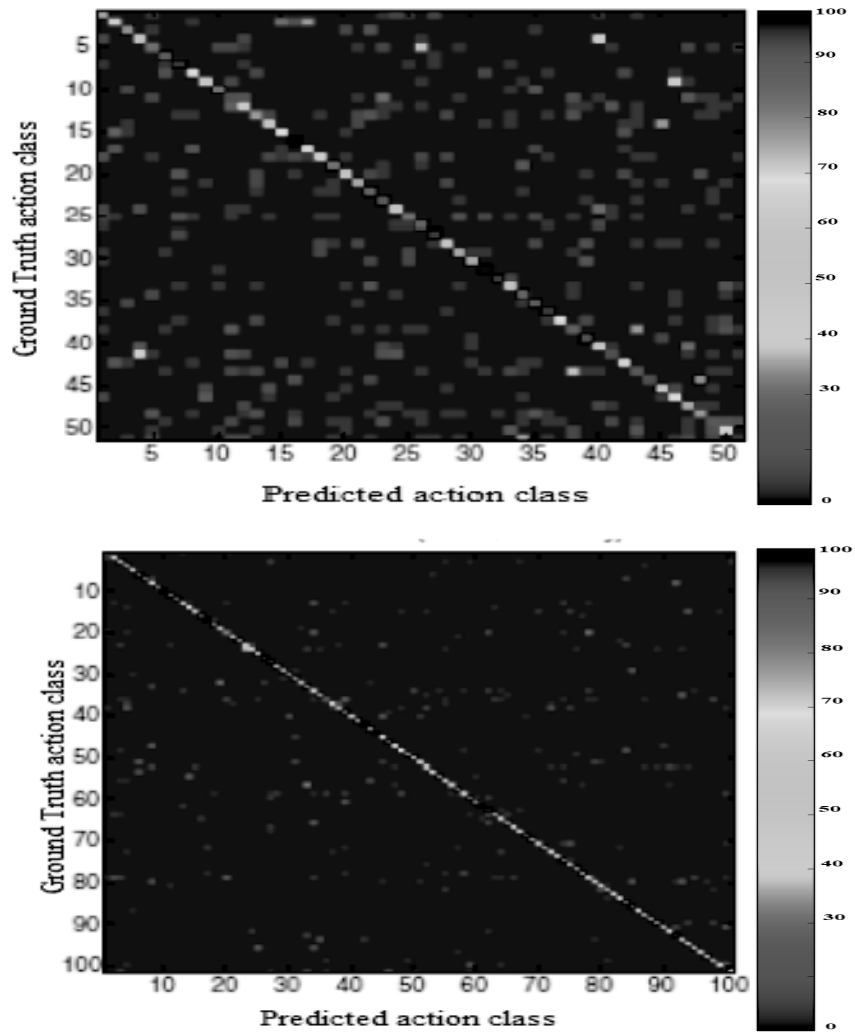

Figure 6. Confusion matrices of the proposed model on HMDB51 and UCF-101 datasets.

Confusion matrixes and recognition visualization

The confusion matrixes for our introduced system on both HMDB51 and UCF-101 datasets are shown in Figure 6. For UCF-101 and HMDB51, we consider all 101 and 51 action categories respectively. Due to the large collection of action classes in these datasets and limited page space, it is not possible to present the confusion matrixes in tabular form. The accuracies in the diagonal cells are indicated by different colors and red cells show the 100% accuracy achieved for the particular action class. The confusion matrix shows the relationship between the classified action class and the ground-truth class. The confusion matrix for the UCF-101 dataset is well diagonalized, where the diagonal portion gives high intensity as recognition accuracy for each action class, and extremely few categories are mixed up when classifying. However, it can be observed from the confusion matrix that some action classes are interfering and misclassifying each other and reporting low scores. The possible reasons for interfering and misclassification are the motion similarity in actions or the same background, objects and scene which shows a similar appearance and motion-based features. For example, action categories throwing and swinging baseball have a similar type of motions as object position is over the head and throwing it away. Motion in both categories produce similar motion features, so the correct action classification is extremely confusing and difficult. Also, two other categories shooting the ball

and dribbling performed in the basketball court and have similar objects and backgrounds so there is a possibility of generation of similar appearance-based features. However, from both given figures, we can observe that the diagonal accuracies dominate in most of the columns, which confirms the good performance of our model.

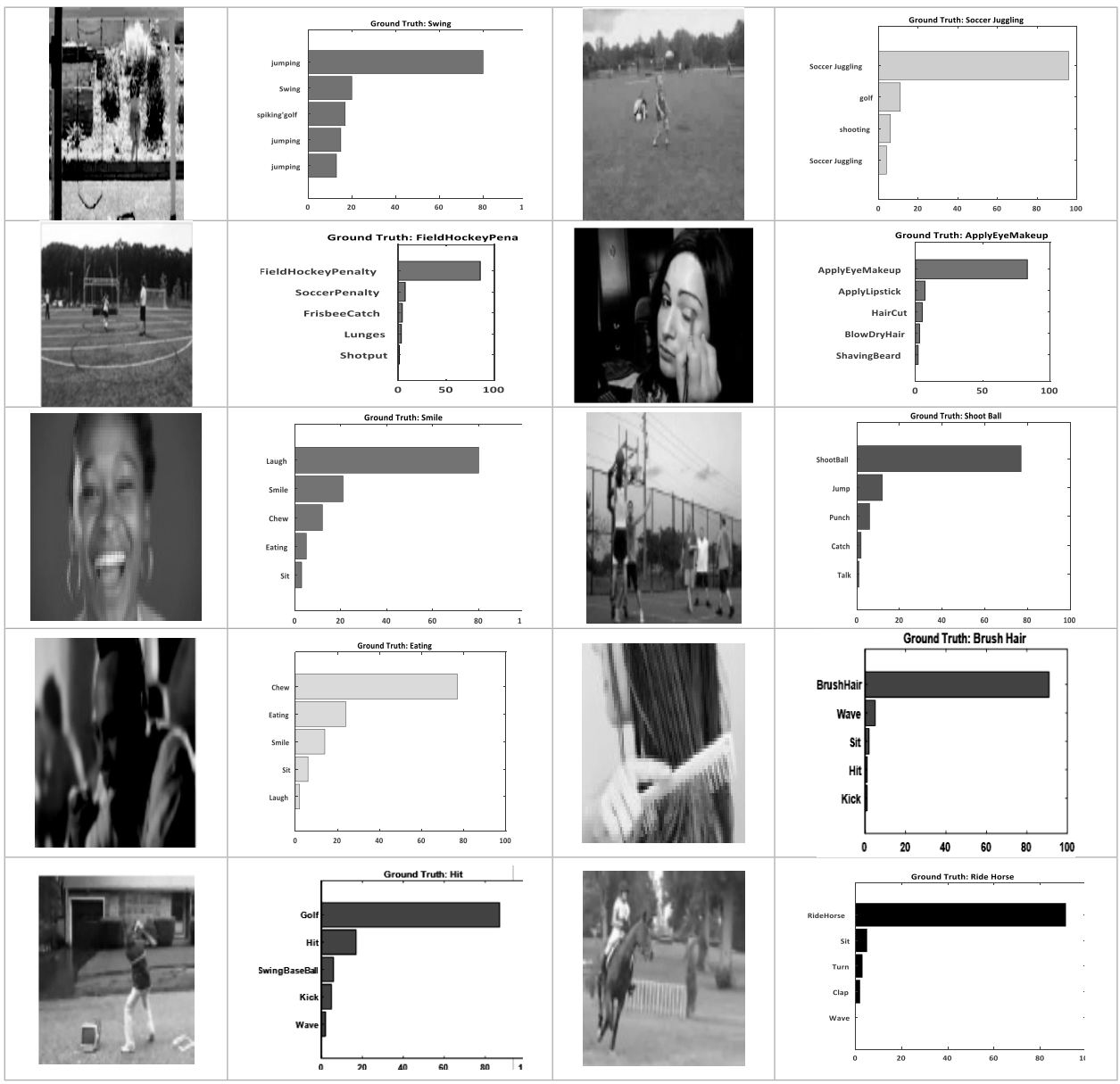

Figure 7. Failed and successful predictions on HMDB51 and UCF-101 datasets.

Furthermore, we also give recognition visualizations on video frames from two standard datasets for a better understanding of our proposed model. The proposed approach is tested on 30% videos of both UCF-101 and HMDB51 datasets. Some of the examples of intermediate frames of action categories along with correct and failed visual recognition results are given in Figure 7. In this figure, two initial rows show some successful and failed results from UCF 101 dataset and the last three rows indicate the successful and failed results from the HMDB51 dataset. Our models receive test video stream as input and C3D captures spatial and motion features which are given

input to the bidirectional LSTM in small segments for time interval *T*. The bidirectional LSTM identify the temporal dynamics and generates output for each segment and then the video action is predicted for the highest frequency class for obtained outputs. In Figure 7, we can see that some action classes are interfering each other, where "Hit" is classified as "Golf", "Eating" is classified as "Chew" "Smile" is predicted as "Laugh". These actions are taken from the HMDB51 dataset, at which our method cannot discriminate fined-grained examples. The possible reason is that the HMDB-51 dataset is not that large in terms of both diversity and scale and these failed predictions are because of the similarity in the camera motion, visual content, appearances, and changes in subject body parts in both confused action classes. While in the case of UCF-101 "Swing" is classified as "Jumping" but most of the action categories are correctly classified. UCF-101 dataset is a comparatively large dataset than HMDB51 for training which can recognize fine-grained examples. Thus, from the qualitative examples, we conclude that our proposed framework can obtain impressive performance in practice.

Table 4. Comparison with the state-of-the-art methods on UCF-101 and HMDB51 datasets.

Features	State of the art approach	Year	UCF101	HMDB51	Combined
Traditional	iDT-FV	2013	84.7%	57.2%	70.9%
	Ordered Trajectories	2013	72.8%	47.3%	60.0%
	MPR	2015	-	65.5%	65.5%
	MoFAP	2016	88.3%	61.7%	75.0%
	Trajectory Rejection	2017	85.7%	58.1%	71.9%
DCNN	Two-stream ConvNets	2016	88.0%	59.4%	73.7%
	FSTCN	2015	88.1%	59.1%	73.6%
	C3D	2015	85.2%	-	-
	EMV-CNN	2016	86.4%	-	-
	DANN	2016	89.2%	63.3%	76.2%
	Dynamic Images	2016	89.1%	65.2%	77.1%
	3D Convolution	2013	91.8%	64.6%	78.2
	FCNs-16	2017	90.5%	63.4%	76.9%
	LTC-CNN	2018	92.7%	67.2%	79.9%
RNN	LSTM	2015	88.6%	-	-
	LRCN	2017	82.9%	-	-
	VideoLSTM	2018	89.2%	56.4%	72.8%
	STPP-LSTM	2017	91.6%	69.0%	80.3%
	Hidden-Two-Stream	2018	90.3%	58.9%	
	RMDN	2017	82.8%	-	-
	L2STM	2017	93.6%	66.2%	79.9%
	TS-LSTM	2019	94.1%	69.0%	81.5%
	Multi-LSTM	2018	90.8%	-	-
	RSTAN [54]	2018	**94.6**%	70.5%	82.55%
Hybrid Model	TDD-iDT+FV	2015	91.5%	65.9%	78.7%
	C3D-iDT	2015	90.4%	-	-
	TSN	2016	94.2%	69.4%	81.8%
	3D Convolution + iDT	2013	93.5%	69.2%	81.3%
	SCLSTM	2017	84.0%	55.1%	69.5%
	FCNs-16 + iDT	2017	93.0%	70.2%	81.6%
	LTC-iDT	2018	92.7%	67.2%	79.9%
Ours	SC-BDLSTM		94.2%	73.9%	84.0%

Comparison to the existing state-of-the-art methods

In our previous subsections, we already explore our proposed model in different aspects. This subsection further verifies the effectiveness and feasibility of our model. The recognition accuracy of our proposed approach is compared with various existing successful and prominent Human Action Recognition approaches on UCF101 and HMDB51 video datasets. The comparison performance in terms of accuracy is listed in Table 4. We categorize these baseline models concerning the type of extracted features and network architecture being used, including traditional (handcrafted), deep convolutional neural networks (DCNNs), recurrent neural networks (RNNs) and hybrid features. Because of the non-availability of recognition results of some methods on particular datasets, some cells are left blanks in the table.

Most of the hand-crafted feature-based techniques make use of trajectories such as iDT-FV (Wang et al., 2013a), ordered trajectories (OD) (Murthy et al., 2013), trajectory rejection (TR) (Seo et al., 2017), Motion part regularization (MPR) (Ni et al., 2015) and Mofap (Wang et al., 2016b) perform well and have competitive results, however, our approach outperforms them by a fair margin on both datasets. Compared with prominent deep learning models such as 3D Convolution, C3D, FCN-16 (Yu et al., 2017), FSTCN (Sun et al., 2015), DANN (Wang et al., 2016c), (Bilen et al., 2016) and LTC-CNN (Varol et al., 2018), the proposed method reported slightly better results in accuracy on UCF-101 and HMDB51 dataset respectively against the best accuracy of LTC-CNN. It can be also seen that some RNN based methods such STPP-LSTM (Wang et al., 2017a), L2STM, RMDN (Bazzani et al., 2017), Hidden-Two-Stream (Zhu et al., 2018), Multi-LSTM (Yeung et al., 2018) and TS-LSTM obtained extremely competitive results on UCF-101 datasets and specially RSTAN (LSTM based attention model) (Wenbin et al., 2018) shows better performance by a minimal margin on the UCF101 dataset. However, our introduced method outperforms these RNN based methods on the HDMB51 dataset and show a higher recognition rate on the small-scale dataset. In contrast to the manually crafted features and DCNN and RNN models, some frameworks like TDD+ iDT+ FV (Wang et al., 2015), FCNs-16+iDTand LTC-IDT, which integrates deep-learning features and hand-crafted features also produced state-of-the-art results but still, our model outperforms these hybrid models by a fair margin. We can conclude that a combination of bidirectional LSTM with the 3D convolutional network for RGB and Saliency-aware streams achieves better results and obtains the recognition rate of 94.2% and 73.9% on UCF101 and HDMB51 datasets respectively. Moreover, we also demonstrate the combined recognition accuracy by considering both datasets and introduced model achieved better results than all. For combined recognition accuracy, we only consider the accuracy of those models which provides the recognition results for both of the datasets. The symbol "-" means results are not reported for this particular dataset. Thus, our model in the presence of RGB and saliency-aware stream explores more relationships between video clips and salient regions and the introduction of bidirectional LSTM captures the long-term temporal dependencies from video frames.

CONCLUSIONS

This research work introduced an effective framework for human action recognition which combines both modalities i.e., C3D and bidirectional LSTMs. Two streams, RGB and saliency-based streams, are used to obtain strong video representation for action prediction in videos. Firstly, we apply the saliency-based method to capture saliency-aware videos, which are extremely useful in enhancing the significance of foreground objects and regions in the video frame. This method also avoids the computation complexity as we usually find in optical flow data. Frame level features are extracted by C3D and clip level features are processed by BD-LSTM and the time-series pooling layer. Our BD-LSTM comprises two stacking layers in both backward and forward directions. This proposed architecture performs well to discover the hidden and complex sequential patterns in video features. The achieved recognition results demonstrate that the recognition performance of our model outperforms the other existing prominent models on HMDB51 and UCF-101 datasets. Successful validations proved that our approach is suitable for the processing of sequential visual data. In future research work, we can retrain our framework on larger datasets such as Kinetic human action video dataset, ActivityNet, 1M-sports, and Charades for further evaluation and improvement of our proposed framework. Also, attention mechanisms can be incorporated into our proposed method to enhance the learning of video representation and successful recognition of complex human actions such as actions with a series of subactivities.

REFERENCES

Arif, S., Wang, J., Ul Hassan, T. & Fei. Z. 2019. 3D-CNN-Based Fused Feature Maps with LSTM Applied to Action Recognition. Innovative Topologies and Algorithms for Neural Networks, Future Internet (MDPI). **11**(2): 1-17.

Bazzani, L., Larochelle, H. & Torresani, L. 2017. Recurrent mixture density network for spatiotemporal visual attention. 5th International Conference on Learning Representations ICLR. Toulon, France.

Bilen, H., Fernando, B. & Gavves. 2016. Dynamic image networks for action recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA.

Dalal, N. & Triggs, B. 2005. Histograms of oriented gradients for human detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). San Diego, CA, USA.

Dalal, N., Triggs, B. & Schmid. C. 2006. Human detection using oriented histograms of flow and appearance. In European Conference on Computer Vision. Graz, Austria.

Donahue, J., Hendricks, L. & S. Guadarrama, S. 2017. Long-term recurrent convolutional networks for visual recognition and description. IEEE Transactions on Pattern Analysis and Machine Intelligence. **39**(4): 677-691.

Feichtenhofer, C., Pinz, A. & Zisserman, A. 2016. Convolutional two-stream network fusion for video action recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA.

Graves, A., Fernández, S. & Schmidhuber, J. 2005. Bidirectional LSTM networks for improved phoneme classification and recognition. International Conference on Artificial Neural Networks: Formal Models and Their Applications–ICANN. Warsaw, Poland.

Hochreiter, S., & J. Schmidhuber, J. 1997. Long short-term memory. 1997. Neural Computation, **9**(8): 1735-1780.

Ji, S., Xu, W. & Yang, M. 2013. 3D convolutional neural networks for human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. **35**(1): 221-231.

Jia, Y., Shelhamer, E. & Donahue, J. 2014. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the ACM International Conference on Multimedia. Pp: 675-678.

Jiang, Y.G., Liu, J. & Roshan, Z.A. 2013. THUMOS challenge: Action recognition with a large number of classes. THUMOS'13 International Workshop on Action Recognition with a Large Number of Classes Program. Sydney, Australia.

Karpathy, A., Toderici, G., Shetty, S. & Leung, T., Sukthankar, R & Li, F.F. 2014. Large-scale video classification with convolutional neural networks. IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH, USA.

Kuehne, H., Jhuang, H. & Garrote, E. 2011. Hmdb: a large video database for human motion recognition. IEEE International Conference on Computer Vision, Barcelona, Spain.

Li, Z., Gavves, E. & Jain, M. 2018. VideoLSTM convolves, attends and flows for action recognition. Computer Vision and Image Understanding. **166**: 41-50.

Lowe, DG. 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision. **60**(2): 91-11.

Ma, C.Y., Chen, M.H. & Kira, Z. 2019. Ts-lstm and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition. Signal Processing: Image Communication. **71**: 76-87.

Mahasseni, B. & Todorovic, S. 2016. Regularizing long short term memory with 3D human-skeleton sequences for action recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA.

Murthy, O.V. & Goecke, R. 2013. Ordered trajectories for large scale human action recognition. In proceeding IEEE conference on computer vision and pattern recognition. Sydney, NSW, Australia.

Ni, B., Moulin, P. & Yang, X. 2015. Motion part regularization: Improving action recognition via trajectory selection. In Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, MA, USA.

Poppe, R. 2010. A survey on vision-based human action recognition. Image and Vision Computing. **28**(6): 976-990.

Scovanner, P., Ali, S. & Shah, M. 2007. A 3-dimensional SIFT descriptor and its application to action recognition. Proceedings of the 15th ACM international conference on Multimedia. Pp: 357-360.

Seo, J., Kim, H & Ro, Y.M. 2017. Effective and efficient human action recognition using dynamic frame skipping and trajectory rejection. Journal Image and Vision Computing, **58**:76-85.

Sharma, S., Kiros, R. & Salakhutdinov, R. 2015. Action recognition using visual attention. Mathematics, Computer Science, International Conference on Learning Representations ICLR.

Simonyan, K., & Zisserman, A. 2014. Two-stream convolutional networks for action recognition in videos. Proceedings of the 27th International Conference on Neural Information Processing Systems, **1**: 568-576.

Soomro, K., Zamir, A.R. & Shah, M. 2012. UCF101: A dataset of 101 human action classes from videos in the wild. Center Res. Comput. Vis. Univ. Florida, Orlando, USA.

Srivastava, N., Mansimov, E & Salakhutdinov, R. 2015. Unsupervised learning of video representations using lstms. Proceedings of the 32nd International Conference on International Conference on Machine Learning, **37**: 843-852.

Sun, L., Jia, K. & Shi, B.E. 2015. Human action recognition using factorized spatio-temporal convolutional networks. In Proceedings of IEEE International Conference on computer vision (ICCV), Santiago, Chile.

Sun, L., Jia, K. & Chen, K. 2017. Lattice Long Short-Term Memory for Human Action Recognition. IEEE International Conference on Computer Vision, Venice, Italy.

Tran, D., Bourdev, L. & Fergus, R. 2015. Learning spatiotemporal features with 3d convolutional networks. IEEE International Conference on Computer Vision, Santiago, Chile.

Ullah, A., Ahmad, J. & Muhammad, K. 2017. Action Recognition in Video Sequences using Deep Bi-directional LSTM with CNN Features. Visual Surveillance and Biometrics: Practices, Challenges, and Possibilities, IEEE Access **6**: 1155-1166.

Varol, G., Laptev, I. & Schmid, C. 2018. Long- term temporal convolutions for action recognition. IEEE transactions on pattern analysis and machine intelligence, **40**: 1510-1517.

Veeriah, V., Zhuang, N. & Qi, G.J. 2015. Differential recurrent neural networks for action recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Santiago, Chile.

Wang, H & Schmid, C. 2013a. Action recognition with improved trajectories. IEEE International Conference on Computer Vision, Sydney, NSW, Australia.

Wang, H., Klaser, A. & Schmid, C. 2013b. Dense trajectories and motion boundary descriptors for action recognition. International Journal of Computer Vision, **103**: 60-79.

Wang, J., Wang, W. & Wang, R. 2016c. Deep alternative neural network: exploring contexts as early as possible for action recognition. Proceedings of 30th Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain.

Wang, L., Qiao, Y. & Tang, X. 2015. Action recognition with trajectory-pooled deep-convolutional descriptors. Proceedings of the IEEE conference on computer vision and pattern recognition. Boston, MA, USA

Wang, L., Xiong, Y. & Z. Wang, Z. 2016a. Temporal segment networks: towards good practices for deep action recognition. European Conference on Computer Vision, Amsterdam, The Netherlands.

Wang, L., Qiao, Y. & Tang, X. 2016b. Mofap: a multi-level representation for action recognition. International Journal of Computer Vision, **119**: 119-254.

Wang, W. & Shen, J. 2015. Saliency-aware geodesic video object segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, MA, USA.

Wang, X., Gao, L. & Wang, P. 2017a. Two-stream 3D ConvNet Fusion for Action Recognition in Videos with Arbitrary Size and Length. IEEE transaction on multimedia, **20**(3): 634-644.

Wang, X., Gao, L. & Song, J. 2017b. Beyond Frame-level CNN: Saliency-Aware 3-D CNN with LSTM for Video Action Recognition. IEEE signal processing letters, **24**(4): 510-514.

Wang, Y., Wang, S. & Tang, J. 2016d. Hierarchical attention network for action recognition in videos. In ArXiv.

Wenbin, D., Wang, Y. & Qiao, Y. 2018. Recurrent Spatial-Temporal Attention Network for Action Recognition in Videos. IEEE Transactions on image processing, **27**(3): 1-14.

Wu, Z., Wang, X. & Jiang, Y. 2015. Modelling spatial-temporal clues in a hybrid deep learning framework for video classification.

In Proceedings of the 23rd ACM international conference on Multimedia. Pp: 461-470.

Xingjian, S., Chen, Z. & Wang, H. 2015. Convolutional lstm network'. A machine learning approach for precipitation nowcasting. Proceedings of the 28th International Conference on Neural Information Processing Systems. Pp: 802-810.

Yeung, S., Russakovsky, O. & N. Jin. 2018. Every Moment Counts: Dense Detailed Labeling of Actions in Complex Videos. International Journal of computer vision, **126**: 375-389.

Yu, S., Cheng, Y. & Xie, L. 2017. Fully convolutional networks for action recognition. Institution of Engineering and Technology (IET). **11**(8): 744-749.

Yue-Hei, J., Hausknecht, M. & Vijayanarasimhan, S. 2015. Beyond short snippets: Deep networks for video classification. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Boston, MA, USA.

Zaremba, W., Sutskever, I. & Vinyals. O. 2014. Recurrent neural network regularization. arXiv preprint arXiv:1409.2329.

Zhang, B., Wang, L. & Wang, Z.Y. 2016. Real-time action recognition with enhanced motion vector CNNs. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, USA.

Zhou, Z., Shi, F. & Wu, W. 2015. Learning spatial and temporal extents of human actions for action detection. IEEE Transaction on Multimedia, **17**(4): 512-525.

Zhu, Y., Zhengzhong, L. & Newsam, S. 2018. Hidden two-stream convolutional networks for action recognition. Asian Conference on Computer Vision (ACCV). Perth, WA, Australia.