IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 5, 2017, accepted November 6, 2017. Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.2778011

Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features

AMIN ULLAH[1], (Student Member, IEEE), JAMIL AHMAD[1], (Student Member, IEEE), KHAN MUHAMMAD[1], (Student Member, IEEE), MUHAMMAD SAJJAD[2], SUNG WOOK BAIK[1], (Member, IEEE)

[1]Intelligent Media Laboratory, Digital Contents Research Institute, Sejong University , Seoul 143-747, South Korea
[2]Digital Image Processing Laboratory, Department of Computer Science, Islamia College Peshawar, Peshawar 25000, Pakistan

Corresponding author: Sung Wook Baik (sbaik@sejong.ac.kr)

This work was supported by the National Research Foundation of Korea Grant funded by the Korea Government (MSIP) under Grant 2016R1A2B4011712.

ABSTRACT Recurrent neural network (RNN) and long short-term memory (LSTM) have achieved great success in processing sequential multimedia data and yielded the state-of-the-art results in speech recognition, digital signal processing, video processing, and text data analysis. In this paper, we propose a novel action recognition method by processing the video data using convolutional neural network (CNN) and deep bidirectional LSTM (DB-LSTM) network. First, deep features are extracted from every sixth frame of the videos, which helps reduce the redundancy and complexity. Next, the sequential information among frame features is learnt using DB-LSTM network, where multiple layers are stacked together in both forward pass and backward pass of DB-LSTM to increase its depth. The proposed method is capable of learning long term sequences and can process lengthy videos by analyzing features for a certain time interval. Experimental results show significant improvements in action recognition using the proposed method on three benchmark data sets including UCF-101, YouTube 11 Actions, and HMDB51 compared with the state-of-the-art action recognition methods.

INDEX TERMS Action recognition, deep learning, recurrent neural network, deep bidirectional long short-term memory, and convolution neural network.

I. INTRODUCTION

Action recognition in video sequences is a challenging problem of computer vision due to the similarity of visual contents [1], changes in the viewpoint for the same actions, camera motion with action performer, scale and pose of an actor, and different illumination conditions [2]. Human actions range from simple activity through arm or leg to complex integrated activity of combined arms, legs, and body. For example, the legs motion for kicking a football is a simple action, while jumping for a head-shoot is a collective motion of legs, arms, head, and whole body [3]. Generally, human action is a motion of body parts by interacting with objects in the environment. In the context of videos, an action is represented using a sequence of frames, which humans can easily understand by analyzing contents of multiple frames in sequence. In this paper, we recognize human actions in a way similar to our observation of actions in real life. We use LSTM to consider the information of previous frames in automatic understanding of actions in videos.

One of the key motivations, which attracts researchers to work in action recognition, is the vast domain of its applications in surveillance videos [4], robotics, human-computer interaction [5], sports analysis, video games for player characters, and management of web videos [6]. Action recognition using video analysis is computationally expensive as processing a short video may take a long time due to its high frame rate. As each frame plays an important role in a video story, keeping information of sequential frames for long time, makes the system more efficient. Researchers have presented many solutions for this problem such as motion, space-time features [7], and trajectories [8]. The proposed method uses recurrent neural network "LSTM" to analyze frame to frame change of action videos. RNNs are building blocks of a connected neuron with input units, internal (or hidden) units, and output units, having an activation at time t, which can selectively process data in sequence. As it processes one element at a time, it can model outputs, consisting of sequence of elements that are not independent [9].

The RNN architecture provides strength to processing and finding hidden patterns in time-space data such as audio, video, and text. RNN processes data in sequential way such that at each time t, it gets input from the previous hidden state S_{t-1} and new data x_t. The data is also multiplied with weights, biases are added, and is fed to activation functions. Due to the large number of calculations, the effect of the initial inputs becomes negligible for the upcoming sequence of data after few layers, resulting in vanishing gradient problem. The solution to this problem is LSTM. The main idea of LSTM architecture is its memory cell, input gate, output gate, and forget gate, which can maintain its state over time T_N, and non-linear gating units which regulate the information flow into/out of the cell [10]. Researchers have presented different variations of LSTM such as multi-layer LSTM and bidirectional LSTM for processing sequential data. The proposed method analyzes the complex pattern in the visual data of each frame, which cannot be efficiently identified using simple LSTM and multi-layer LSTM [11].

In the proposed method, features of video frames are analyzed for action recognition. Deep features from every sixth frame of a video are extracted using pre-trained AlexNet [12]. Next, an architecture of DB-LSTM is developed with two layers at each forward and backward pass for learning sequence information in the features of video frames. The proposed method is capable of recognizing actions in long videos because the video is processed in N time steps. Our system has less computational complexity as it only processes five frames per second. The implementation of DB-LSTM has a high capacity of learning sequences and frame to frame change in features due to small change in visual data of videos. These properties make the proposed method more suitable for action recognition in videos. The rest of the paper is organized as the follows: Section 2 presents an overview of the related works. The proposed framework is explained in Section 3. Experimental results, evaluation of our technique, and comparison with other state-of-the-art methods are discussed in Section 4. Section 5 concludes the paper with future research directions.

II. RELATED WORKS

Over the last decade, researchers have presented many hand-crafted and deep-nets based approaches for action recognition. The earlier work was based on hand-crafted features for non-realistic actions, where an actor used to perform some actions in a scene with simple background. Such systems extract low level features from the video data and then feed them to a classifier such as support vector machine (SVM), decision tree, and KNN for action recognition. For instance, the geometrical properties of space-time volume (STV) called action sketch, were analyzed by Yilmaz and Shah [13]. They stacked body contours in time axis by capturing direction, speed, and shape of STV for action recognition. Gorelick et al. [14] presented human action as three-dimensional shapes made from the silhouettes in the STV. They used the poisson equation method to analyze 2D shapes of actions and extracted space time features (STF) containing local space-time saliency, action dynamics, shape structure, and orientation. Their method used a non-realistic dataset and, in certain cases, two different actions resulted the same 2D shapes in STV, making the representation of different actions difficult. Hu et al. [15] used two types of features: motion history image (MHI) and histogram of oriented gradients feature (HOG). The former is the foreground image subtracted from the background scenario whereas the later one is magnitudes and directions of edges. These features were then fused and classified through a simulated annealing multiple instance learning SVM (SMILE-SVM). Liu et al. [16] extracted motion and static features for realistic videos. They pruned the noisy motion feature by applying motion statistics to acquire stable features. In addition, they also used "PageRank" to mine the most informative static features and construct discriminative visual vocabularies. However, these hand-crafted features based methods have certain limitations. For instance, STVs based methods are not effective for recognizing multiple person actions in a scene. STF and MHI based techniques are more suitable for simple datasets. To process complex datasets, we need hybrid approaches which can combine different features and preprocessing such as motion detection [17], background segmentation [18], HOG, SIFT, and SURF. But such hybrid methods increase the computational complexity of the target system. These limitations can cause difficulty for lengthy videos and real-time applications with continuous video streaming.

Besides hand-crafted features based approaches for action recognition, several deep learning based methods were also proposed in recent years. Deep learning has shown significant improvement in many areas such as image classification, person re-identification, object detection, speech recognition and bioinformatics [19]. For instance, a straight forward implementation of action recognition using deep networks is developed through 3D convolutional networks by Ji et al. [20]. They applied 3D convolutional kernels on video frames in a time axis to capture both spatial and temporal information. They also claimed that their approach can capture motion and optical flow information because frames are connected by fully connected layers at the end. A multi-resolution CNN framework for connectivity of features in time domain is proposed by [21] to capture local spatio-temporal information. This method is experimentally evaluated on a new "YouTube 1 million videos dataset" of 487 classes. The authors claimed to have speed up the training complexity by foveated architecture of CNN. They improved the recognition rate for large dataset up to 63.9% but their recognition rate on UCF101 is 63.3%, which is still too low for such important task of action recognition. A two-stream CNN architecture is proposed by [22] in which first stream captures spatial and temporal information between frames and second one demonstrates the dense optical flow of multiple frames. They have increased the amount of data for the training CNN model by combining two datasets. In [6], authors used two CNN models for processing each individual frame of the input video for

IEEE Access

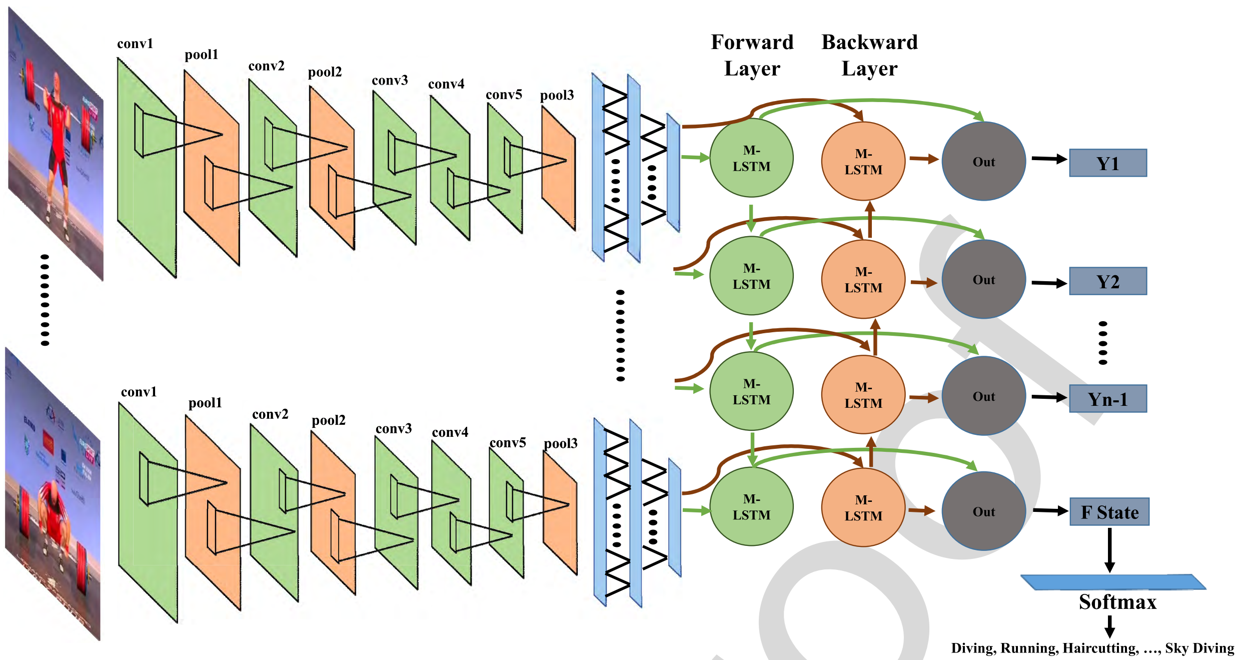

FIGURE 1. Framework of the proposed DB-LSTM for action recognition.

action recognition. The output of intermediate layers of both architectures is processed by special 1x1 kernels in fully connected layers. The method finally used 30 frames unrolled LSTM cell connected with the output of CNN in training. The feature maps of pre-trained model are analyzed by Bilen et al. [23] for video representation named as dynamic image. They added rank pooling operator and approximate rank pooling layer in fine tuning phase, which combine maps of all frames to a dynamic image as one representation of the video. Deep learning based approaches have the ability to accurately identify hidden patterns in visual data because of its huge feature representation pipeline. On the other hand, it requires huge amount of data for training and high computational power for its processing. In this work, we have balanced the complexity of the system and action recognition accuracy. Our method is computationally efficient as it analyzes only each sixth frame of the video, which is an optimal value for frame jump verified through different experiments. For better action recognition, we have intelligently combined CNN and LSTM due to its state-of-the-art results on visual and sequential data.

III. PROPOSED FRAMEWORK

In this section, the proposed framework and its main components are discussed in detail including the recognition of an action $\mathcal{A}_\mathcal{I}$ from the sequence of frames in video $\mathcal{V}_\mathcal{I}$ using DB-LSTM and features extraction through CNN for $\mathcal{F}_\mathcal{N}$ frames. The procedure for action recognition is divided into two parts: First, we extract CNN features from the frames of video $\mathcal{V}_\mathcal{I}$ with jump $\mathcal{J}_\mathcal{F}$ in sequence of frames such that the jump $\mathcal{J}_\mathcal{F}$ does not affect the sequence of the action $\mathcal{A}_\mathcal{I}$ in the video. Second, the features representing the sequence of action $\mathcal{A}_\mathcal{I}$ for time interval $\mathcal{T}_\mathcal{S}$ (such as $\mathcal{T}_\mathcal{S} = 1$ sec) are fed to the proposed DB-LSTM in $\mathcal{C}_\mathcal{N}$ chunks, where each $\mathcal{C}_\mathcal{I}$ chunk is the features representation of the video frame and input to one RNN step. At the end, the final state of each time internal $\mathcal{T}_\mathcal{S}$ is analyzed for final recognition of an action in a video. The proposed framework is shown in Fig. 1. Each step of the proposed method is discussed in separate section. The input and output parameters of the proposed method are given in Table 1.

A. PREPARATION AND FEATURES EXTRACTION

CNN is a dominant source for the representation and classification of images. In the case of video data, each individual frame is represented by CNN features, followed by finding the sequential information between them using DB-LSTM. A video is a combination of frames moving at 30 to N frames per second. Thirty to fifty frames in a unit time have many redundant frames, whose processing is a computationally expensive process. Considering this processing complexity, we jump six frames when processing a video for action recognition. It is evident from the experiments that a six frame jump does not affect the sequence of the action. The scenario of the features representation is given in Fig. 2, where the first row represents the frames in a sequence and second row shows features maps of the corresponding frames. A basketball is moving from one player to another where a small change in players' position and orientation can be observed. As CNN finds hidden patterns in images, it captures all the tiny changes in each frame. These changes in sequential form are learnt through RNN for action recognition in a video.

Training a deep learning model for image representation requires thousands of images and also requires high

TABLE 1. Description of input and output parameters used in the proposed DB-LSTM for action recognition.

V_I	Action video.
A_I	Action in video V_I.
F_N	Number of frames in video V_I
J_F	Jump between frames during extracting features
s_t	Output of current state of RNN
$b^{i,f,o}$	Biases of input, output, and forget gates of LSTM cell
T_S	Time interval of action feed to DB-LSTM.
C_N	Number of chunks in T_S.
DB-LSTM	Deep bidirectional LSTM.
FC8	Fully connected layer of CNN.
x_t	Input to RNN at time t.
$W^{i,f,o}$	Weights of input, output, and forget gates of LSTM cell

processing power such as GPU for the weight adjustment of the CNN model. Getting the required model using this strategy is an expensive process, which is solved using transform learning [24] where a trained model can be used for other purposes. In the proposed method, we used parameters of the pre-trained CNN model, called AlexNet [12] for feature extraction, which is trained on large scale ImageNet [25] dataset of more than 15 million images. The architecture of the model is given in Table 2. AlexNet has five convolution layers, three pooling layers, and three fully connected layers. Each layer is followed by a norm and ReLU nonlinear activation function. The extracted features vector from FC8 layer is one thousand dimensional. The features of each frame are considered as one chunk for one input step of RNN. $\mathcal{C}_N$ chunks for $\mathcal{T}_S$ time interval are feed to RNN. Thus for one second with six frame jump in video, we process six frames out of thirty frames. When we feed features of six frames, RNN processes it in six chunks. The final state of the RNN is counted for each $\mathcal{T}_S$ for final recognition. A detailed explanation of the RNN is given in the upcoming sub-sections.

B. RECURRENT NEURAL NETWORKS

RNNs are introduced for analyzing hidden sequential patterns in both temporal sequential and spatial sequential data [26]. Video is also sequential data in which movements in visual contents are represented in many frames such that sequence of frames help in understanding the context of an action. RNNs can interpret such sequences but forget the earlier inputs of the sequence in case of long term sequences. This problem is known as the vanishing gradient problem, which can be solved through a special type of RNN called LSTM [27]. It is capable of learning long term dependencies. Its special structure with input, output, and forget gates controls the long term sequence pattern identification. The gates are adjusted by a sigmoid unit that learns during training where it is to open and close. Eq. 1 to Eq. 7 [28] explain the operations performed in LSTM unit, where x_t is the input at time t (in our case it is chunk $\mathcal{C}$). f_t is the forget gate at time t, which clears information from the memory cell when needed and keeps a record of the previous frame whose information needs to be cleared from the memory. The output gate o_t keeps information about the upcoming step, where g is the recurrent unit, having activation function "tanh" and is computed from the input of the current frame and state of the previous frame s_{t-1}. The hidden state of an RNN step is calculated through tanh activation and memory cell c_t. As the action recognition does not need the intermediate output of the LSTM, we made final decision by applying softmax classifier on the final state of the RNN network.

$$i_t = \sigma((x_t + s_{t-1})W^i + b_i) \tag{1}$$
$$f_t = \sigma((x_t + s_{t-1})W^f + b_f) \tag{2}$$
$$o_t = \sigma((x_t + s_{t-1})W^o + b_o) \tag{3}$$
$$g = \tanh((x_t + s_{t-1})W^g + b_g) \tag{4}$$
$$c_t = c_{t-1} \cdot f_t + g \cdot i_t \tag{5}$$
$$s_t = \tanh(c_t) \cdot o_t \tag{6}$$
$$final\_state = soft\max(Vs_t) \tag{7}$$

Training large data with complex sequence patterns (such as video data) are not identified by the single LSTM cell. Therefore, in the proposed approach, we use ML-LSTM by stacking multiple LSTM cells to learn long term dependencies in video data.

C. MULTI LAYERS LSTM

The performance of the deep neural network has been boosted by increasing the number of layers in the neural network models. The same strategy is followed here for RNN by stacking two LSTM layers to our network. By adding this new layer, RNN captures higher level of sequence information [28]. In standard RNN, data is fed to single layer for activation and processing before output, but in time sequence problems, we need to process data on several layers. By stacking LSTM layers, each layer in the RNN is a hierarchy that receives the hidden state of the previous layer as input. Fig. 3 shows a multi-layer LSTM. Layer 1 receives input from data x_t while the input of layer 2 is from its previous time step $\mathrm{s}_{t-1}^{(2)}$, and the output of the current time step of layer one $\mathrm{s}_{t}^{(1)}$. The computation of LSTM cell is same as Eq. 1 to Eq. 7 but

FIGURE 2. Frame to frame features representation and changes in sequence of frames.

TABLE 2. Frame to frame fractures representation and changes in sequence of frames.

Layers	Conv1	Pool1	Conv2	Pool2	Conv3	Con4	Con5	Pool5	FC6	FC7	FC8
Kernel	11x11	3x3	5x5	3x3	3x3	3x3	3x3	3x3	-	-	-
Stride	4	2	1	2	1	1	1	2	-	-	-
Channels	96	96	256	256	384	384	256	256	4096	4096	1000

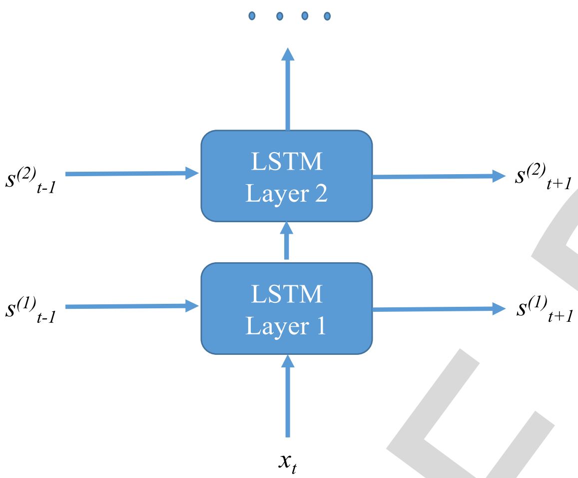

FIGURE 3. Two layer LSTM network.

only the layer's information has been added to the superscript of each i_t, f_t, o_t, c_t, and s_t. Eq. 8 shows the procedure of calculating the state of a layer.

$$s_t^l = \tanh(c_t^l) \cdot o_t^l \tag{8}$$

D. BIDIRECTIONAL LSTM

In bidirectional LSTM, the output at time t is not only dependent on the previous frames in the sequence, but also on the upcoming frames [29]. Bidirectional RNNs are quite simple, having two RNNs stacked on top of each other. One RNN goes in the forward direction and another one goes in the backward direction. The combined output is then computed based on the hidden state of both RNNs. In our work, we are using multiple LSTM layers, so our scheme has two LSTM layers for both forward pass and backward pass. Fig. 4 shows the overall concept of bidirectional LSTM used in the proposed method.

Fig. 4 (a) shows the external structure of the training phase, where the input data is fed to the bidirectional RNN, and the hidden states of forward pass and backward pass are combined in the output layer. The validation and cost is calculated after the output layer and weights and biases are adjusted through back-propagation. For validation, 20% of the data is separated from the dataset and cross entropy is used for error calculation of the validation data. Stochastic optimization [30] with a learning rate of 0.001 is used for cost minimization. Fig. 4 (b) shows the internal structure of the bidirectional RNN, where "fw" is forward pass and "bw" is backward pass. Both fw and bw consist of two LSTM cells, making our model a deep bidirectional LSTM. The proposed method outperforms other state-of-the-art methods due to its mechanism of computing the output. The output of a frame at time t is calculated from the previous frame at time $t-1$ and the upcoming frame at time $t+1$ because layers are performing processing in both directions.

IV. EXPERIMENTAL EVALUATION

In this section, the proposed technique is experimentally evaluated and the results are discussed on different benchmark action recognition datasets including UCF101 [2], Action YouTube [16], and HMDB51 [31]. A few sample images from each action category are give in Fig. 5. The datasets are divided by following machine learning three splits protocol in training, validation, and testing of 60%, 20%, and 20%, respectively. We have used Caffe toolbox for deep features extraction, tensorflow for DB-LSTM, and GeForce-Titan-X GPU for implementation. The training data is fed in mini batches of 512 size with a learning rate of 0.001 for cost minimization and one thousand iteration for learning the sequence patterns in the data. We have compared the proposed technique with recent state-of-the-art methods using the average accuracy score of confusion matrix as the recognition rate on each database. The comparisons with other methods are given in Table 3. The recognition scores are reported from the referenced papers. Some of the cells in Table 3 are blank because those methods have not reported the recognition score on the corresponding dataset.

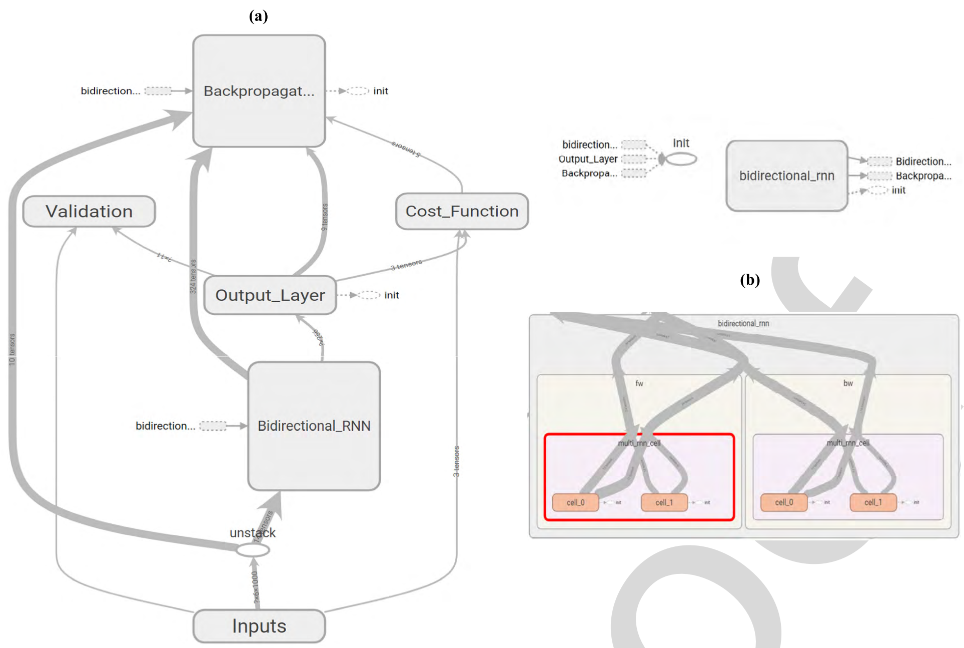

FIGURE 4. External and internal structure of the proposed DB-LSTM network.

TABLE 3. Comparison of average recognition score of the proposed DB-LSTM for action recognition with state-of-the-art methods.

Method	YouTube	HMDB51	UCF101
Multiresolution CNNs [21]	-	-	65.4%
LSTM with 30 frame unroll [6]	-	-	88.6%
Two-stream CNNs [22]	-	59.4%	88.0%
Multiple dynamic images [23]	-	65.2%	89.1%
RLSTM-g3 [32]	-	55.3%	86.9%
Hierarchical clustering multi-task [33]	89.7%	51.4%	76.3%
VideoDarwin [34]	-	63.7%	-
Discriminative representation [35]	91.6%	28.2%	79.7%
Ordered trajectories [8]	-	47.3%	72.8%
Factorized spatio-temporal CNNs [36]	-	59.1%	88.1%
Temporal pyramid CNNs [37]	-	63.1%	89.1%
Adaptive RNN-CNNs [38]	-	61.1%	-
Improved trajectories [39]	-	57.2%	-
Super-category exploration [40]	-	60.8%	-
Multi-layer fisher vector [41]	-	68.5%	-
Proposed DB-LSTM	**92.84**	**87.64**	**91.21**

A. UCF101 DATASET

UCF101 is one of the most popular action recognition datasets of realistic action videos. It consists of 13320 videos taken from YouTube, which are divided into 101 action categories. Each category contains videos between [100, 200]. UCF101 is comparatively more challenging dataset due to its large number of action categories from five major types: 1) human-object interaction, 2) body-motion only, 3) human-human interaction, 4) playing musical instruments, and 5) sports. Some categories have many actions such as sports, where most of the sports are played in a similar background, i.e., greenery. Some of the videos are captured in different illuminations, poses, and from different viewpoints. One of the major challenges in this dataset is its realistic actions performed in real life, which is unique compared to other datasets where actions are performed by an actor. The recognition scores of the proposed method and other methods are reported in column 4 of Table 3. Our method reported an increase of 2.11% in the accuracy, increasing it from 89.1% to 91.21%, which is the previous year best accuracy of TPC and MDI. The recognition accuracies are 65.4%, 88%, and 88.1% for other CNN based methods such as multi-resolution CNNs, two-stream CNNs, and factorized spatio-temporal CNNs, respectively. From the trajectories based methods, the ordered

IEEE Access

FIGURE 5. Sample action categories of UCF-101, HMDB51, and YouTube action dataset.

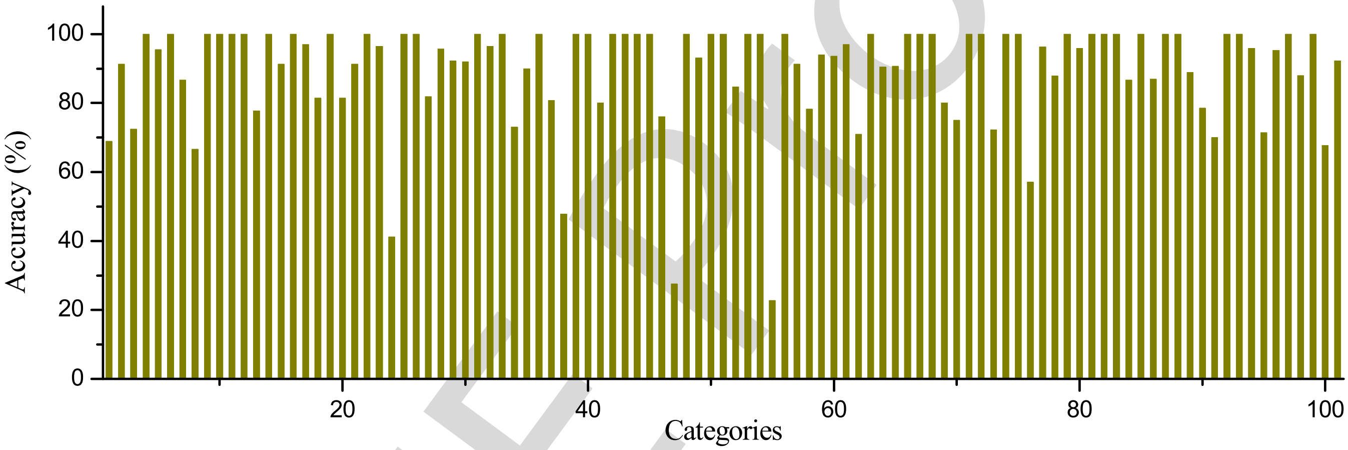

FIGURE 6. Class wise accuracy of UCF-101 dataset on the proposed DB-LSTM for action recognition.

trajectories reported 72.8% accuracy while the improved trajectories based method has however not reported the recognition rate for UCF101 dataset. Fig. 6 shows the class wise accuracy of UCF101 dataset on test data. The horizontal axis represents categories and the vertical axis shows the percentage accuracy of corresponding category. From results, it can be seen that the results of most of the categories are greater than 80%; some of them reach 100%; and only three categories have accuracies less than 50%. The proposed method improved the recognition rate on UCF101 dataset from 89.1% to 91.21%. The confusion matrix is given in Fig. 7, where the intensity of true positives (diagonal) is high for each category, proving the efficiency of the proposed method on UCF101 dataset.

B. HMDB51 DATASET

The HMDB51 dataset contains a variety of actions related to human body movements including objects interaction with body, facial actions, and human interaction for body movements. It consists of 6849 action video clips, which are divided into 51 classes, each containing more than one hundred clips. It is more challenging because the clips of each category are collected for a variety of subjects with different illuminations and 4 to 6 clips are recorded for each subject performing the same action on different poses and viewpoints. The proposed method is capable of learning frame to frame changes regardless of its view point, pose, and subject. The proposed approach outperformed on HMDB51 dataset as is evident from the comparisons with other methods in column 3 of Table 3. The proposed method boosted the accuracy on this dataset from 68.5% to 87.64% with 19.14% increment while the accuracy of other CNN based methods is far behind. The confusion matrix is given in Fig. 7, where the intensity of true positives (diagonal) is high for each category. Fig. 8 shows category wise result for the proposed method, which is consistent for all categories. The horizontal axis represents categories and the vertical axis shows the percentage accuracy of the corresponding category. It can be

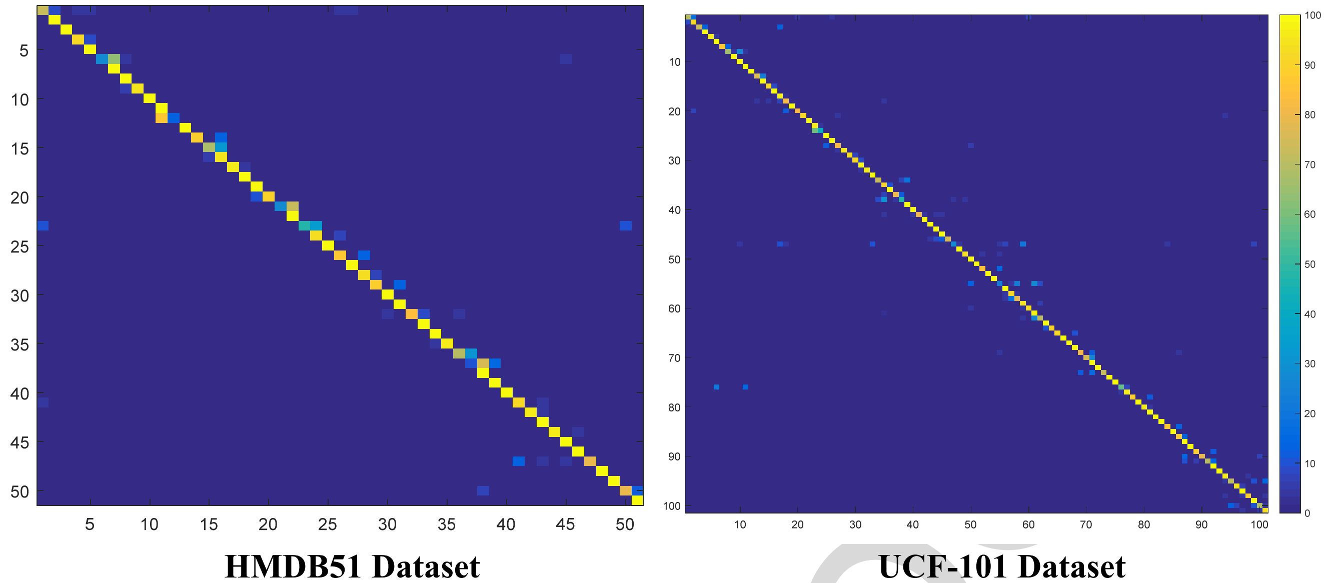

FIGURE 7. Confusion matrixes of HMDB51 and UCF-101 datasets for the proposed DB-LSTM for action recognition.

TABLE 4. Confusion matrix of YouTube actions dataset for the proposed DB-LSTM for action recognition.

Categories	Basketball	Biking	Diving	Golf-Swing	Horse-Riding	Soccer	Swing	Tennis	Jumping	Volleyball	walking
Basketball	**96.55**	0	0	0	0	0	0	3.44	0	0	0
Biking	0	**99.2**	0	0	0	0	0	0	0	0.8	0
Diving	0	0	**90.62**	0	9.37	0	0	0	0	0	0
G-Swing	0	0	0	**98.7**	0	0.3	0	0	0	0.9	0
H-Riding	0	0	0	0	**100**	0	0	0	0	0	0
Soccer	0	0	0	0	0	**78.12**	0	0	0	3.12	18.75
Swing	0	0.3	0.2	0	0.5	0	**99.0**	0	0	0	0
Tennis	0	0	0	0	0	8.082	0	**70.58**	0	20.58	0
Jumping	0	0	0	0	0	0	0	16.66	**83.33**	0	0
Volleyball	0	0	0	0	0	0	0	0	0	**100**	0
walking	0	0	0	0	0	0	0	0	0	0	**100**
Average Accuracy											**92.656%**

seen that more than 20 classes reported 100% accuracy. The variation in accuracy of other classes is between 80% and 100%. Among the other classes, only two classes reported test accuracy less than 20%. The proposed method increased the recognition rate on HMDB51 from 68.5% to 87.64%.

C. YouTube ACTIONS DATASET

YouTube actions dataset is a small but very challenging dataset for action recognition. The dataset is collected from 11 sports action categories including volleyball, basketball, golf, horse riding, biking/cycling, tennis, diving, football, swinging, jumping, and walking with a dog. The dataset contains 25 different subjects with more than four video clips for each subject. The video clips in the same subject share some common features such as the same actor, similar background, and similar viewpoint. There is large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, and illumination conditions, making this dataset more challenging. We achieved an average accuracy of 92.84% on this dataset as given in column 2 of Table 3, dominating the hierarchical clustering multi-task and discriminative representation method having 89.7% and 91.6% accuracy, respectively. The confusion matrix for this dataset is given in Table 4. Our method has achieved more than 90% accuracy for eight classes. The class "soccer" reported 78.1% accuracy and the class "interfere with walking" has 18.75% false prediction. This is due to the fact that in soccer, a performer walks around a football, leading to less accuracy. Similarly, the class "tennis" and "volleyball" are interfering because the background of these activities has the same scenarios, i.e., players are jumping and playing around a net. The recognition score is low for three categories including "walking and soccer", "jumping", and "volleyball" due to similar features, i.e. motion of body parts of an actor in performing actions.

D. VISUAL RESULTS AND COMPUTATION

The proposed method is tested on 20% videos of each dataset. Some of the correct and miss classified visual results are shown in Fig. 9. The intermediate frames of an action are

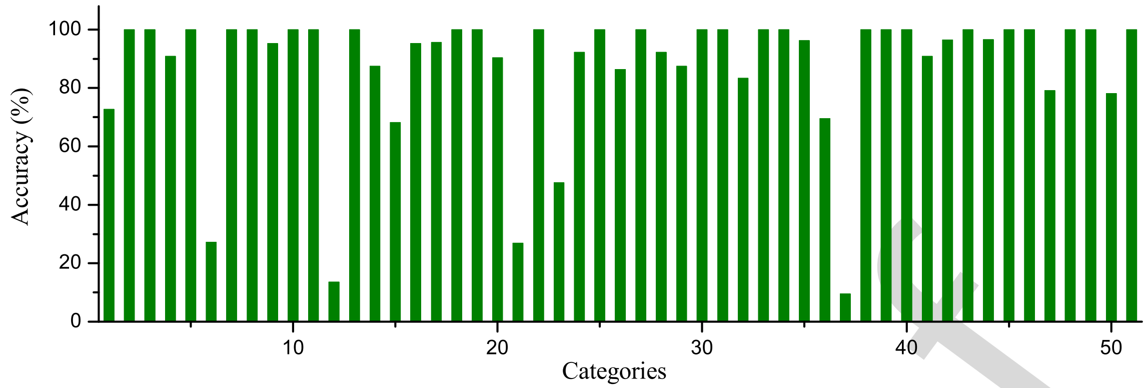

FIGURE 8. Class-wise accuracy of HMDB51 dataset for the proposed DB-LSTM for action recognition.

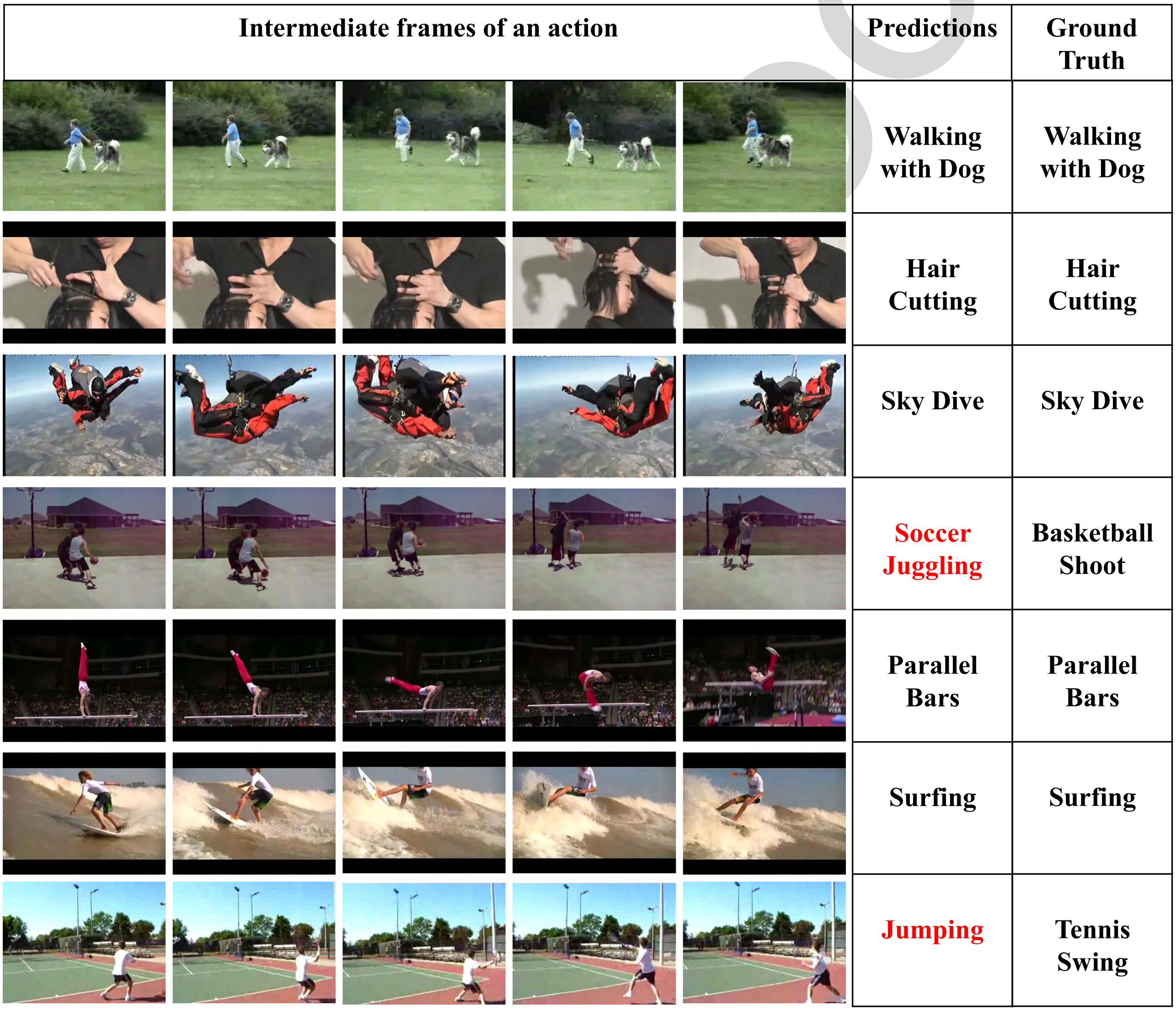

FIGURE 9. Predictions of the proposed DB-LSTM for action recognition for sample clips. The red font indicates wrong prediction of our method.

given for understanding of an action in Fig. 9. Our method takes a test video as input and extracts features from its frames with six frame jump. The extracted features are fed to the proposed DB-LSTM in chunks for time interval T. The DB-LSTM returns output for each chuck and finally the video is classified for the highest frequency class

TABLE 5. Average time complexity and accuracy on different frame jumps for 30 FPS video clip.

Experiments	Frame Jump	Average Time Complexity	Average Accuracy
1	4	1.725 sec	92.2%
2	6	1.12 sec	91.5%
3	8	0.90 sec	85.34%

in outputs. In Fig. 9, row 4 and row 7 are miss-classified, where "basketball shoot" is classified as "soccer juggling" and "tennis swing" is classified as "jumping". These incorrect predictions are due to the similarity of visual content, motion of camera, and changes in parts of an actor body in both action categories. We have evaluated the proposed method using different experiments with various number of frame jumps for analyzing action in videos. Table 5 shows the statistics of the conducted experiments. We have used 6 frame jump in overall experiments because of its optimal results in complexity and accuracy. The proposed method is evaluated on GeForce-Titan-X GPU for feature extraction, training, and testing. The system takes approximately 0.23 sec for feature extraction per frame. Feeding the extracted features to DB-LSTM for classification takes 0.53 sec for 30 frames per second video clip. Overall, the proposed method takes approximately 1.12 seconds for processing of a 1-second video clip. With these statistics, our method can process 25 frames per second, making it a suitable candidate for action recognition in real-time video processing applications.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed an action recognition framework by utilizing frame level deep features of the CNN and processing it through DB-LSTM. First, CNN features are extracted from the video frames, which are fed to DB-LSTM, where two layers are stacked on both forward and backward pass of the LSTM. This helped in recognizing complex frame to frame hidden sequential patterns in the features. We analyzed the video in N chunks, where the number of chunks depend on the time interval "T" for processing. Due to these properties, the proposed method is capable of learning long term complex sequences in videos. It can also process full length videos by providing prediction for time interval "T". The output for small chunks is combined for the final output. The experimental results indicate that the recognition score of the proposed method successfully dominates other recent state-of-the-art action recognition techniques on UCF-101, HMDB51, and YouTube action video datasets. These characteristics make our proposed method more suitable for processing of visual data and can be an integral component of smart systems. The proposed method extracts features from the whole frame of the video. In future, we aim to analyze only the salient regions of the frames for action recognition. Furthermore, we have intention to extend this work for activity recognition in videos [42]–[44]. Finally, the proposed method can be combined with people counting techniques to intelligently analyze the people crowded behavior and dense situations [45].

REFERENCES

[1] A. Nanda, D. S. Chauhan, P. K. Sa, and S. Bakshi, "Illumination and scale invariant relevant visual features with hypergraph-based learning for multi-shot person re-identification," *Multimedia Tools Appl.*, pp. 1–26, Jun. 2017, doi: https://doi.org/10.1007/s11042-017-4875-7

[2] K. Soomro, A. R. Zamir, and M. Shah. (2012). "UCF101: A dataset of 101 human actions classes from videos in the wild." [Online]. Available: https://arxiv.org/abs/1212.0402

[3] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey," *Image Vis. Comput.*, vol. 60, pp. 4–21, Apr. 2017.

[4] A. Nanda, P. K. Sa, S. K. Choudhury, S. Bakshi, and B. Majhi, "A neuromorphic person re-identification framework for video surveillance," *IEEE Access*, vol. 5, pp. 6471–6482, 2017.

[5] S. A. Aly, T. A. Alghamdi, M. Salim, and A. A. Gutub, "Data dissemination and collection algorithms for collaborative sensor devices using dynamic cluster heads," *Trends Appl. Sci. Res.*, vol. 8, no. 2, pp. 55–72, 2013.

[6] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4694–4702.

[7] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, 2004, pp. 32–36.

[8] O. V. R. Murthy and R. Goecke, "Ordered trajectories for human action recognition with large number of classes," *Image Vis. Comput.*, vol. 42, pp. 22–34, Oct. 2015.

[9] Z. C. Lipton, J. Berkowitz, and C. Elkan. (2015). "A critical review of recurrent neural networks for sequence learning." [Online]. Available: https://arxiv.org/abs/1506.00019

[10] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[11] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional LSTM networks for improved phoneme classification and recognition," in *Proc. 5th Int. Conf.*, Warsaw, Poland, Sep. 2005, p. 753.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[13] A. Yilmaz and M. Shah, "Actions sketch: A novel action representation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 984–989.

[14] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007.

[15] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. S. Huang, "Action detection in complex scenes with spatial and temporal ambiguities," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 128–135.

[16] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos 'in the wild,'" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 1996–2003.

[17] S. K. Choudhury, P. K. Sa, S. Bakshi, and B. Majhi, "An evaluation of background subtraction for object detection vis-a-vis mitigating challenging scenarios," *IEEE Access*, vol. 4, pp. 6133–6150, 2016.

[18] S. K. Choudhury, P. K. Sa, K.-K. R. Choo, and S. Bakshi, "Segmenting foreground objects in a multi-modal background using modified Z-score," *J. Ambient Intell. Hum. Comput.*, pp. 1–15, Apr. 2017, doi: https://doi.org/10.1007/s12652-017-0480-x

[19] S. K. Choudhury, P. K. Sa, R. P. Padhy, S. Sharma, and S. Bakshi, "Improved pedestrian detection using motion segmentation and silhouette orientation," *Multimedia Tools Appl.*, pp. 1–40, Jun. 2017, doi: https://doi.org/10.1007/s11042-017-4933-1

[20] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[21] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.

[22] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[23] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould, "Dynamic image networks for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3034–3042.

[24] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 806–813.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255.

[26] K.-I. Funahashi and Y. Nakamura, "Approximation of dynamical systems by continuous time recurrent neural networks," *Neural Netw.*, vol. 6, no. 6, pp. 801–806, 1993.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. 15th Annu. Conf. Int. Speech Commun. Assoc.*, 2014, pp. 338–342.

[29] A. Ogawa and T. Hori, "Error detection and accuracy estimation in automatic speech recognition using deep bidirectional recurrent neural networks," *Speech Commun.*, vol. 89, pp. 70–83, May 2017.

[30] D. P. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980

[31] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2556–2563.

[32] B. Mahasseni and S. Todorovic, "Regularizing long short term memory with 3D human-skeleton sequences for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3054–3062.

[33] A.-A. Liu, Y.-T. Su, W.-Z. Nie, and M. Kankanhalli, "Hierarchical clustering multi-task learning for joint human action grouping and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 102–114, Jan. 2017.

[34] B. Fernando, E. Gavves, J. O. M, A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5378–5387.

[35] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1290–1297.

[36] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi, "Human action recognition using factorized spatio-temporal convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4597–4605.

[37] P. Wang, Y. Cao, C. Shen, L. Liu, and H. T. Shen. (2015). "Temporal pyramid pooling based convolutional neural networks for action recognition." [Online]. Available: https://arxiv.org/abs/1503.01224

[38] M. Xin, H. Zhang, H. Wang, M. Sun, and D. Yuan, "ARCH: Adaptive recurrent-convolutional hybrid networks for long-term action recognition," *Neurocomputing*, vol. 178, pp. 87–102, Feb. 2016.

[39] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3551–3558.

[40] Y. Yang, R. Liu, C. Deng, and X. Gao, "Multi-task human action recognition via exploring super-category," *Signal Process.*, vol. 124, pp. 36–44, Jul. 2016.

[41] M. Sekma, M. Mejdoub, and C. B. Amar, "Human action recognition based on multi-layer Fisher vector encoding method," *Pattern Recognit. Lett.*, vol. 65, pp. 37–43, Nov. 2015.

[42] Y. Liu, L. Nie, L. Han, L. Zhang, and D. S. Rosenblum, "Action2Activity: Recognizing complex activities from sensor data," in *Proc. IJCAI*, 2015, pp. 1617–1623.

[43] L. Liu, L. Cheng, Y. Liu, Y. Jia, and D. S. Rosenblum, "Recognizing complex activities by a probabilistic interval-based model," in *Proc. AAAI*, 2016, pp. 1266–1272.

[44] I. Kaysi, B. Alshalalfah, A. Shalaby, A. Sayegh, M. Sayour, and A. Gutub, "Users' evaluation of rail systems in mass events: Case study in Mecca, Saudi Arabia," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2350, pp. 111–118, Dec. 2013.

[45] H. Abdelgawad, A. Shalaby, B. Abdulhai, and A. A.-A. Gutub, "Microscopic modeling of large-scale pedestrian–vehicle conflicts in the city of Madinah, Saudi Arabia," *J. Adv. Transp.*, vol. 48, no. 6, pp. 507–525, 2014.

AMIN ULLAH (S'17) received the B.C.S. degree in computer science from the Islamia College Peshawar, Peshawar, Pakistan. He is currently pursuing the M.S. degree leading to the Ph.D. degree with the Intelligent Media Laboratory, Sejong University, South Korea. His research interests include features extraction, image analysis, content-based image retrieval, and deep learning for multimedia understanding.

JAMIL AHMAD (S'16) received the B.C.S. degree (Hons.) in computer science from the University of Peshawar, Pakistan, in 2008, the master's degree with specialization in image processing from the Islamia College Peshawar, Peshawar, Pakistan, in 2014. He is currently pursuing the Ph.D. degree with Sejong University, Seoul, South Korea. He is currently a Regular Faculty Member with the Department of Computer Science, Islamia College Peshawar. His research interests include deep learning, medical image analysis, content-based multimedia retrieval, and computer vision. He has published several journal articles in these areas in reputed journals, including the *Journal of Real-Time Image Processing*, *Multimedia Tools and Applications*, *Journal of Visual Communication and Image Representation*, *PLOS One*, the *Journal of Medical Systems, Computers and Electrical Engineering*, *SpringerPlus*, the *Journal of Sensors*, and the *KSII Transactions on Internet and Information Systems*. He is an Active Reviewer of the *IET Image Processing, Engineering Applications of Artificial Intelligence*, the *KSII Transactions on Internet and Information Systems, Multimedia Tools and Applications*, the IEEE Transactions on Image Processing, and IEEE Transactions on Cybernetics.

KHAN MUHAMMAD (S'16) received the bachelor's degree in computer science from the Islamia College Peshawar, Pakistan, in 2014, with research in information security. He is currently pursuing the M.S. degree leading to the Ph.D. degree in digital contents with Sejong University, Seoul, South Korea. He has been a Research Associate with the Intelligent Media Laboratory since 2015. His research interests include image and video processing, information security, image and video steganography, video summarization, diagnostic hysteroscopy, wireless capsule endoscopy, computer vision, deep learning, and video surveillance. He has published over 20 papers in peer-reviewed international journals and conferences, such as *Future Generation Computer Systems*, the IEEE Access, the *Journal of Medical Systems, Biomedical Signal Processing and Control*, *Multimedia Tools and Applications, Pervasive and Mobile Computing*, *SpringerPlus*, the *KSII Transactions on Internet and Information Systems*, *Journal of Korean Institute of Next Generation Computing*, the *NED University Journal of Research*, *Technical Journal*, the *Sindh University Research Journal*, the *Middle-East Journal of Scientific Research*, MITA 2015, PlatCon 2016, and FIT 2016.

MUHAMMAD SAJJAD received the master's degree from the Department of Computer Science, College of Signals, National University of Sciences and Technology, Rawalpindi, Pakistan, and the Ph.D. degree in digital contents from Sejong University, Seoul, South Korea. He is currently an Assistant Professor with the Department of Computer Science, Islamia College Peshawar, Pakistan. He is also the Head of the Digital Image Processing Laboratory, Islamia College Peshawar, where students are involved in research projects under his supervision, such as social data analysis, medical image analysis, multi-modal data mining and summarization, image/video prioritization and ranking, fog computing, Internet of Things, virtual reality, and image/video retrieval. His primary research interests include computer vision, image understanding, pattern recognition, and robot vision and multimedia applications, with current emphasis on raspberry-pi and deep learning-based bioinformatics, video scene understanding, activity analysis, fog computing, Internet of Things, and real-time tracking.

SUNG WOOK BAIK (M'16) received the B.S degree in computer science from Seoul National University, Seoul, South Korea, in 1987, the M.S. degree in computer science from Northern Illinois University, Dekalb, in 1992, and the Ph.D. degree in information technology engineering from George Mason University, Fairfax, VA, USA, in 1999. He was with Datamat Systems Research Inc., as a Senior Scientist of the Intelligent Systems Group from 1997 to 2002. In 2002, he joined the faculty of the College of Electronics and Information Engineering, Sejong University, Seoul, where he is currently a Full Professor and the Dean of Digital Contents. He is also the Head of Intelligent Media Laboratory, Sejong University. His research interests include computer vision, multimedia, pattern recognition, machine learning, data mining, virtual reality, and computer games. He served as a Professional Reviewer for several well-reputed journals, such as the *IEEE Communication Magazine*, *Sensors*, *Information Fusion*, *Information Sciences*, the IEEE TIP, MBEC, MTAP, SIVP, and JVCI.

• • •