International Journal of Multimedia Information Retrieval (2023) 12:34
https://doi.org/10.1007/s13735-023-00301-9

REGULAR PAPER

ConvST-LSTM-Net: convolutional spatiotemporal LSTM networks for skeleton-based human action recognition

Abhilasha Sharma[1] · **Roshni Singh**[1]

Received: 11 February 2023 / Revised: 10 August 2023 / Accepted: 24 September 2023 / Published online: 27 October 2023

Abstract
Human action recognition (HAR) emphases on perceiving and identifying the action behavior done by humans within an image/video. The HAR activities include motion patterns and normal or abnormal activities like standing, walking, sitting, running, playing, falling, fighting, etc. Recently, it sparks the attention of researchers especially in 3D skeleton sequence. The actions of human can be represented via sequence of motions of skeletal keyjoints, although not all the skeleton keyjoints are informative in nature. Various approaches for HAR are used like LSTM, ConvLSTM, Conv-GRU, ST-LSTM, etc. Thus far, ST-LSTM approaches have shown tremendous performance in 3D skeleton sequence tasks but the detection of irrelevant keyjoints produce noise that deteriorates the performance of the model. So, the intent is to bring attention toward improving the efficacy of the model by focusing on informative keyjoint coordinates only. Therefore, the research paper introduces a new class of spatiotemporal LSTM approaches named as ConvST-LSTM-Net (convolutional spatiotemporal long short-term memory network) for skeleton-based action recognition. The prime focus of proposed model is to identify the informative keyjoints in each frame. The result of extensive experimental analysis exhibits that ConvST-LSTM-Net outperforms the state-of-the-art models on various benchmarks dataset, viz. NTU RGB + D 60, UT-Kinetics, UP-Fall Detection, UCF101, and HDMB51 for skeleton sequence data.

Keywords Activity recognition · LSTM · Spatiotemporal · Skeleton sequence · ConvLSTM

1 Introduction

Human action recognition has turn out to be a prominent & diligent research area in computer vision and image processing, which includes classification and recognition of normal & abnormal human activities of daily routine. It belongs to the automated recognition of human activity (normal & abnormal) in various application areas via analyzing the sequence of observations. Nowadays, crowded places with normal and abnormal activities are familiar due to population increase that turn toward suspicious activities. So, HAR has become an essential part in the automatic interpretation of human environment interaction in various online-offline applications such as auto-driving, intelligent surveillance [1–5], smart-gadgets analysis [6], object detection & tracking [7], video retrieval [8], and assisted daily living. Other HAR applications firmly coupled with the daily activities such as motion analysis [9–12], pose motion analysis [13, 14], health monitoring [15], classification or detection of actions or motions [16], and understanding human action behavior [17]. By recognizing and analyzing the human actions from the videos, one can clearly distinguish between normal and abnormal behaviors that can make significant improvements in public safety. Withal, HAR remains an ambitious challenge due to its clutter backgrounds, slight interclass segregation, and wide intra-class deviation. The main thing to recognize high accuracy & efficiency is to conquer both static appearances within each frame of the videos as well as temporal relationships throughout the multiple frames generated via videos. Some applications such as monitoring suspicious detection and early reporting for fall detection are also considered in human activity recognition. However, various techniques are there for the representation of human action based on motion, such as RGB-based

✉ Roshni Singh
roshnisingh1815@gmail.com

Abhilasha Sharma
abhi16.sharma@gmail.com

[1] Department of Software Engineering, Delhi Technological University, Shahbad Daulatpur, Delhi 110042, India

videos [18–20], RGBD-based videos [21–24], and skeleton-based videos [25–30]. Skeleton-based action recognition has become more prominent in recent times as it offers a focused and concise approach. The representation of human skeletons in videos typically involves a series of joint coordinates, which can be obtained through pose estimators or action prediction methods. By focusing solely on the action poses and disregarding contextual factors like background variations and lighting changes, skeleton sequences provide a compact and robust way to capture action information. On comparing these techniques, all the skeleton-based methods represent: (i) human motion via 3D coordinates positions for key-body points and (ii) are more robust to problems like variations of background clutter, observation viewpoints, illumination or intensity conditions, and so on. These advantages of skeleton motion sequences motivate researchers to develop new techniques for exploring informative features for human action recognition. These methods are gaining utmost importance in HAR since skeletons represent a compact sequence of data forms that depict dynamic motion within human body movements [31]. In respect of annotation, i.e., capturing the effective motion action representation from several unlabeled skeleton samples, manual annotation founds to be very expensive and challenging, nonetheless an unexplored area. These days, sensors are used for collecting data for their low-cost and high mobility. Some approaches are used for tracking and calculating the skeleton keyjoints with feature invariant to human key-body points, observation point, camera viewpoint, and so on.

Some approaches are used for tracking and calculating the skeleton keyjoints with feature invariant to human key-body points, observation point, camera viewpoint, and so on. The skeletal features within the human body are responsible for recognizing all the normal & abnormal activities. Besides this, they have also been used for evaluating (some activities such as falling, discriminating between jogging and running) the variation in the keyjoint coordinates between the center mass of the body, acceleration motion, velocity motion, for movement: angles between the keyjoint points within the skeleton. Some new methods like ST-LSTM and ST-GCN (spatiotemporal graph convolution network) are practices to extract these features also. The movement of body parts and the execution of various actions are made possible by the human skeletal system. When it comes to data modality, the use of skeleton-based information aligns with the structure of the human anatomy, which enhances the interpretability of ConvLSTM learning. This modality specifically focuses on 3D coordinates of keyjoints in the human body. By analyzing these skeleton sequences, the model is capable of recognizing and understanding human movements. Another advantage of the skeleton modality is its emphasis on privacy, as it is considered to be more privacy-friendly compared to other modalities.

In this work, the prime objective is to efficiently combine the important cues in CNN (convolutional neural networks), and LSTM using spatiotemporal data with skeleton-based recognition approaches call up as ST-LSTM. Here, a set of extracted skeleton features in conjunction with skeletal keyjoint is fed as input to the model. The skeletal tracking algorithms were used for detecting the keyjoints followed by the feature extraction that has been done through RGB frame data (extracted from videos) for improvising the efficiency of model. Some standard features, like angle between the keyjoint coordinates, velocity motion, acceleration motion, and human body position of the center of mass, for movement: angles between the joint key points, have been extracted from keyjoint coordinates of the human skeleton.

Once the feature extraction is done along with preprocessing thereupon, the preprocessed data are feed to the model, consisting of 17 extracted features among 25 skeleton coordinates. The overall pipeline of proposed ConvST-LSTM-Net model is illustrated in Fig. 1. The model exploits a spatiotemporal network consisting of CNNs, ST-LSTMs & fully connected dense layers. The model first detects the skeleton keyjoints of the persons using the skeleton-based recognition method. These keyjoints are fed to the CNN layers, followed by ST-LSTMs for the extraction of spatial–temporal features. Then, output from a hidden layer of ST-LSTMs is passed via FC dense layer (fully connected) for classification.

The key contributions of the research work can be summarized as follows:

1. A spatiotemporal ConvST-LSTM-Net model has been proposed that utilizes human body keyjoint coordinates from skeletal data obtained from RGB videos. The keyjoints are fed as an input to CNN layers for extracting the spatial–temporal features followed by ST-LSTM and output is passed to the time-distributed FC dense layer.
2. Motivated by the advances in CNN, ConvLSTM, and ST-LSTM, we have seamlessly combined the ideas of these models and integrated them to propose a new paradigm for skeleton-based action recognition termed as ConvST-LSTM-Net. The model brings the attention toward improving the efficacy by focusing only on informative keyjoint coordinates.
3. Among 25 keyjoints, a set of 17 extracted skeleton features along with 21 skeleton keyjoint coordinates are fed to the model as not all the skeleton keyjoints are informative in nature for recognizing the action classes.
4. The proposed ConvST-LSTM-Net model shows better performance in comparison to the existing models by using different modalities over various benchmarks, viz. NTU RGB + D dataset [32], UT-Kinect dataset [33], UP-Fall Detection [34], UCF101 [35], and HDMB51 dataset [36].

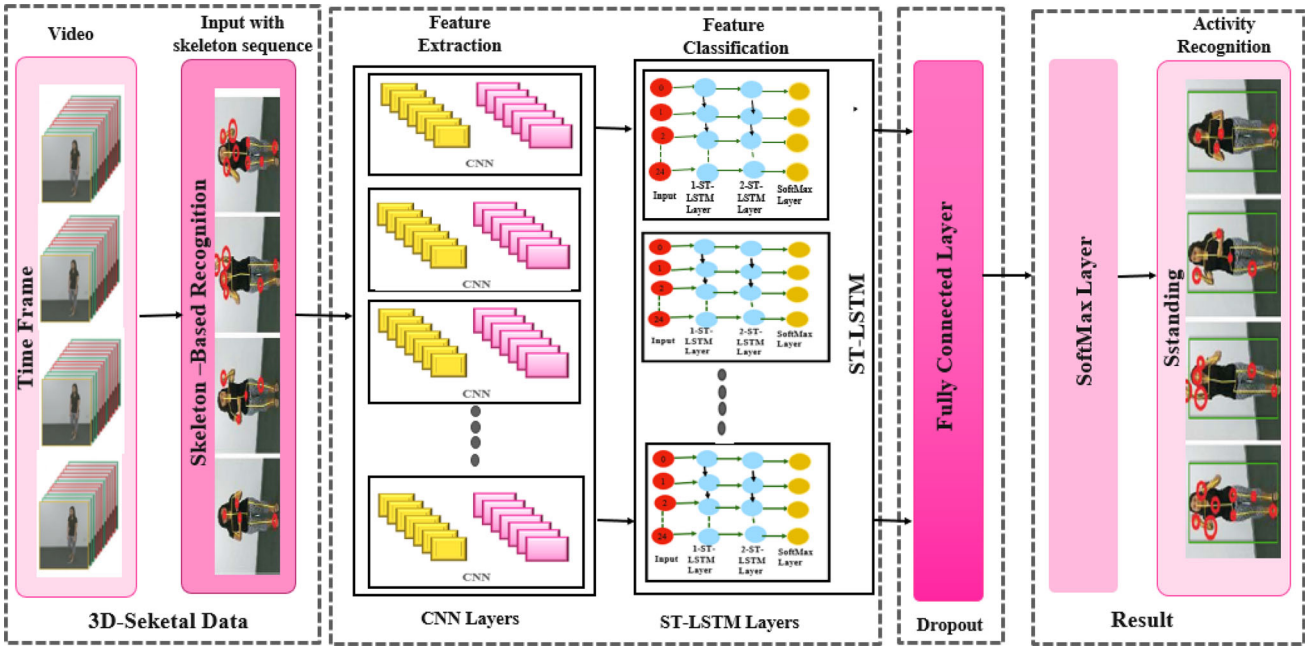


Fig. 1 Process informatics pipeline of ConvST-LSTM-Net. At first, the video frames are passed through the skeleton-based recognition feature method to extract the skeleton keyjoint coordinates. Then, the obtained keyjoint coordinates are fed to a modified ST-LSTM cell followed by ST-LSTM layers to evaluate the spatiotemporal feature. Further, the outputs are passed to FC dense layers. Ultimately, SoftMax shows the framewise prediction scores of human action behaviors

The rest of this paper is systematized as follows: Sect. 2 describes an overview of relevant studies available in literature. Section 3 briefly introduces the key terms and techniques along with the proposed methodology. Section 4 shows the experimental results and analysis. At last, the Sect. 5 discusses about the conclusion and future perspective.

2 Related work

This section briefly discusses the work related to skeleton-based RNN, LSTM & ST-LSTM for human action recognition.

2.1 Skeleton-based action recognition

Earlier, conventional skeleton-based recognition in the HRA modal aimed at the handcrafted features [37, 38], a well-known method for describing and classifying image features. However, it fails to classify the adequate semantic-labeling information of the human body. Deep learning approaches such as convolutional neural networks (CNNs) [39–43], recurrent neural networks (RNNs) [44–49], and graph convolutional networks (GCNs) [50] have achieved the exotic performance for learning more informative features about skeleton sequence, which helps in HAR learning. Many works have been introduced to achieve high performance of the skeleton sequence model. Li et al. [41] introduced a framework on convolutional co-occurrence feature learning that gradually works on hierarchical methods to aggregate contextual information on various levels. Vemulapalli et al. [37] designed a rolling map based on relative 3D rotations among different human body parts. Liu et al. [44] elongate RNN-based technique into the spatial–temporal model for revisiting the result based on the action-related performance of human action. Zhu et al. [43] introduced a cuboid-CNN in skeleton actions, ultimately concluding a human's normal keyjoint movements. Zhang et al. [51] implemented a view-adaptive modal for auto-regulate angle viewpoints during any motion action & obtaining different viewpoint observations of human actions. However, for skeleton sequences, these models fail to extract the temporal–spatial correlation configurations & even fail to explore the graphical aspects of human body structure. Due to the popularity of graph-based techniques, Yan et al. [52] introduced an approach based on GCN for the skeleton-based HAR, then introduced the ST-GCN method for featuring the spatial & temporal dynamics configuration of keyjoint skeletons of humans synchronously. Song et al. [53] worked on solving the occlusion issues and implemented multi-stream GCN for extracting qualified features for activated skeleton keyjoints in human action. Furthermore, they proposed a non-local technique [54] by using

2-stream GCN approach: 2 s-AGCN for improving recognition accuracy. Also, Shi et al. [55] worked on GCN fusion feature and proposed the multi-stream architecture at the output layer. Cheng et al. [56] worked on shift operation based on graph and used the point-wise convolutions connected layer for lowering its computational complexity. Ye et al. [57] introduced novel work on DGCN (dynamic graph convolutional network), an approach used for skeleton-based action recognition under 2-stream-AGCN, which features global dependency via achieving preeminent accuracy. Zhang et al. [58] also work on GCN in the spatial attentive–temporal dilated network for feature extraction in skeleton frame sequences using distant spatial attention weights and temporal scales. In 2-Stream network, Shi et al. [59] confiscated on bones, i.e., bone stream and joint stream, but entirely independent of each other. Furthermore, directed graph neural networks (DGNNs) [60], graph edge convolutional neural networks (GECNNs) [61] were introduced, which depict the relation among joints-bones in terms of action, but they fail to represent the various methods to combine features in the motion action transmission field.

2.2 Skeleton-based action recognition using LSTM's and RNN's approaches

Currently, the deep learning area mainly focuses on recurrent neural network (RNN)-based techniques, since it manifests its growth in skeleton-based action detection. In ConvST-LSTM network, the basic principle of this proposed model is familiarized from the ST-LSTM approach, i.e., a sequential fusion of CNN followed by the spatiotemporal method with LSTM is also known for the extension version of RNNs. In this subsection, a brief survey is provided on RNN approaches and LSTM approaches since they are the basic building blocks of the proposed methodology. Veeriah et al. [62] worked on LSTM and introduced a differential gating method to affirm the rate of information change. Du et al. [63] work on the HRNN network (hierarchical recurrent neural network) approach for depicting skeleton structure of human body along with its temporal dynamics coordinates for keyjoints in 2D. In LSTM network, Zhu et al. [64] implemented a mixed regularization technique for normalization toward learning the co-occurrence of skeletal joint features. Meanwhile, they introduced a network for trained termed as 'in-depth-dropout mechanism.' Shahroudy et al. [32] worked in LSTM to learn long-term contextual representations of different body parts individually termed as part-aware LSTM model. Liu et al. [44, 65] intended a framework network based on 2D spatiotemporal LSTM for both temporal and spatial domains to explore the hidden input layer's information-related context in the human body. For 3D coordinates of skeletal joints, they proposed a 'trust-gate-mechanism' [44] to trade on imprecise 3D-coordinates inputted via depth sensors devices. Nowadays, skeleton-based action RNN and LSTM approaches also adapt toward action forecasting and detection [66, 67].

3 ConvST-LSTM-Net: the proposed methodology

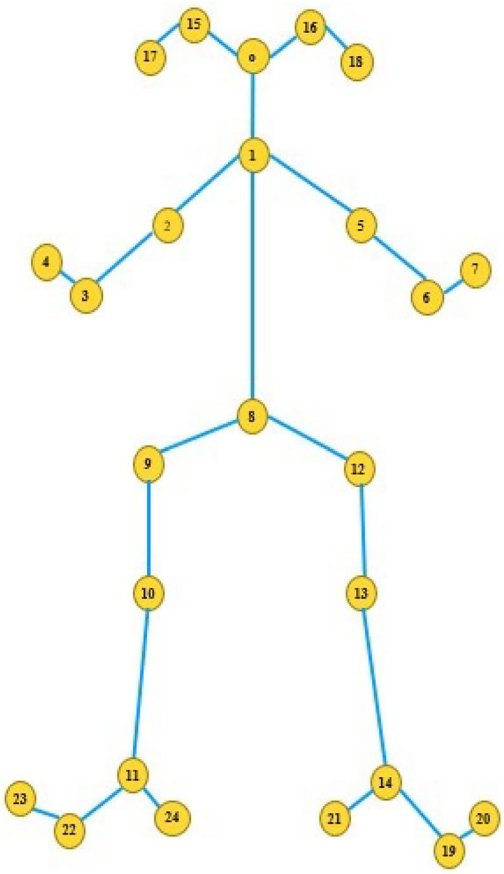
This section briefly canvasses crucial terms and approaches used in proposed ConvST-LSTM model that is divided into three models, namely CNN, ST-LSTM, and ConvST-LSTM-Net. The proposed methodology has been executed with the review of CNN along with the construction of skeleton body with coordinates and ST-LSTM, respectively.

Initially for dataset preparation, the human body frames from raw RGB videos are used to train the network and meanwhile track the 3D skeleton jointkey coordinates. For preprocessing, the 3D joint normalization method is applied that is helpful in making a bounding box above the tracked human body. Some features have been extracted for determining different activities, features such as velocity motion (v), acceleration motion (a), weight (w), depth (), height (H), angle (θ) within the consecutive skeleton joints, etc. After training, the feature extraction is done for input human activity behavior. Following subsections elucidates the whole network architecture of the ConvST-LSTM-Net model.

3.1 Keypoint detection and preprocessing

In preprocessing technique for RGB videos, the frames are inputted into the ST-LSTM network to evaluate the keyjoint locations of human body frames from skeleton coordinates. Figure 2 represents the 25-skeleton keyjoint coordinates, which have been tracked at each joint. Only 17 skeletal keyjoints are covered (since they are the informative skeletal keyjoints to specify the normal and abnormal human activities) and these are the right knee, right hip, left knee, left hip, left foot, left ankle, right foot, right ankle, head, spine mid, left wrist, spine base, right shoulder, left shoulder, right wrist, right elbow, and left elbow. Each frame tracked the human skeleton comprising X, Y, Z coordinates of human body keyjoints. After getting the 3D skeletal coordinates, normalization technique has been applied on 3D keyjoints to generate bounding boxes over the tracked human skeleton, which may vary as per the person's movement in the video.

Afterward, a feedforward network has been used based on a multi-CNN layer followed by ST-LSTM that takes input in the form of keyjoints coordinates from video frames using skeleton-based recognition. It learns the affiliation among the body parts of individuals within the frames. Table 1 presents



S.No.	Key-Joint Coordinates
0	Nose
1	Neck
2	Right Shoulder
3	Right Elbow
4	Right Wrist
5	Left Shoulder
6	Left Elbow
7	Left Wrist
8	Mid Hip
9	Right Hip
10	Right Knee
11	Right Ankle
12	Left Hip
13	Left Knee
14	Left Ankle
15	Right Eye
16	Left Eye
17	Right Ear
18	Left Ear
19	Left Thumb-Toe
20	Left Little-Toe
21	Left Heel
22	Right Thumb-toe
23	Right Little-Toe
24	Right Heel

Fig. 2 The 25-skeleton keyjoints for the human body track detection and preprocessing

Table 1 Detail of tracked skeletal keyjoint coordinates, derived features, and action class

Label	Description
Skeleton keyjoints	Right Hip, Left Hip, Left Foot, Right Foot, Right Knee, Left Knee, Right Ankle, Left Ankle, Head, Right Wrist, Left Wrist, Right Shoulder, Left Shoulder, Left Elbow, Right Elbow, Spine mid, Spine Base
Derived features	$\angle\theta$,v,α, h_d, $\not{d}$,w,H(Geometric & Kinematic features)
Action class	Sit, Stand, Walk, Run, Stand

the details of tracked skeletal keyjoints, a set of derived features, and action class. In this work, for normalizing the convergence of loss function, minimum–maximum normalization technique (i.e., min–max norm) has been used. Here, X indicates the training dataset, then normalization can be achieved as:

$$X_{\text{noramlize}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{1}$$

3.2 Construction & evaluation of feature vector: geometric & kinematic features

The skeleton keyjoint coordinates are used for constructing and calculating features vectors. The keyjoints coordinates of the human body that are tracked for different activities of humans are actually decided by using feature vectors. For particular activities, different features are utilized. These features and their evaluation are as follows:

$\angle\boldsymbol{\theta}$ *(Angle between keyjoints of skeleton coordinates)*:

Among 25 keyjoints coordinates, we consider those coordinates which are connected via straight line and then a skeleton structure of tracked human body is drawn using these coordinates as shown in Fig. 3. Accordingly, only 10 keyjoint comes out to be the most relevant ones, viz. left

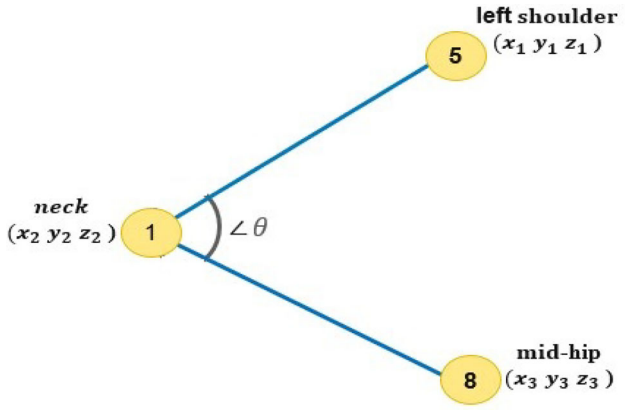


Fig. 3 Illustration for calculation of angle from the left side of the skeleton between left shoulder, neck, and mid-hip

shoulder, spine mid, right shoulder, spine base, left knee, right knee, left hip, right hip, left ankle, and right ankle are used for calculating the value of angle θ. It is the illustration for evaluating the keyjoint angles between the left shoulder, neck, and mid hip. If *A, B, C* are considered as the distance between the coordinate, then values are formulated as $A_1 = x_1 - y_1$, $B_1 = x_2 - y_2$, and $C_1 = x_3 - y_3$, then θ can be evaluated as follow:

$$\boldsymbol{\theta} = \frac{\boldsymbol{ABC}}{\boldsymbol{AB} * \boldsymbol{BC}} \tag{2}$$

where $ABC = (A1 * A2 + B1 * B2 + C1 * C2)$

$$\boldsymbol{AB} = \sqrt{\boldsymbol{A}_1^2 + \boldsymbol{B}_1^2 + \boldsymbol{C}_1^2} \text{ and } \boldsymbol{BC} = \sqrt{\boldsymbol{A}_2^2 + \boldsymbol{B}_2^2 + \boldsymbol{C}_2^2} \tag{3}$$

Velocity motion estimation ($\boldsymbol{v}$):

Velocity motion is calculated by taking the distance of positions of humans at time frames t and $t + 1$ in *x, y, z*-dimension. So, velocity of the tracked person is given by:

$$\boldsymbol{v} = \frac{d}{\boldsymbol{t}} \tag{4}$$

where $\mathscr{d}$ indicates distance of the tracked person between frames and t indicates the frame time.

acceleration motion estimation (α):

The rate of changes of velocity between consecutive frames in *x, y, z*-directions at time frame t. It is given by:

$$\alpha = \frac{v}{t} \tag{5}$$

where α is the acceleration of motion of the person. v indicates the velocity motion of tracked person between the frames. t indicates the frame time.

Head-floor distance ($\mathrm{h}_{\boldsymbol{d}}$):

It measures the distance between the head keyjoint coordinate & the floor where tracked person's location found.

Head-depth distance():

The distance measured from first camera view to the adjacent object is termed as depth. So, head-depth is calculated via the head keyjoint's coordinates in the z-dimension of a tracked person within the frame.

Width (w):

Width is defined as the difference between maximum of right-left keyjoint ($\boldsymbol{R}_{\mathbf{jMax}} - \boldsymbol{L}_{\mathbf{jMax}}$) coordinates of tracked person. The extreme left keyjoint width is estimated using a left elbow, left hip, left knee, left shoulder, left ankle, left foot, and head keyjoint values. In the same way, the right extreme keyjoints can be calculated by using all the right-side keyjoint coordinates. It is calculated as follows:

$$\boldsymbol{W} = \left|\boldsymbol{R}_{\mathbf{jMax}} - \boldsymbol{L}_{\mathbf{jMax}}\right| \tag{6}$$

where R_j indicates the right keyjoint coordinates, L_j indicates the left keyjoint coordinates.

Height (H):

Height is the measure between utmost bottom keyjoints and utmost top keyjoints of body coordinates. In extreme bottom, it includes keyjoint coordinates like left ankle, right knee, left knee, right ankle, right foot, left ankle, left foot, and right ankle and in utmost top, it includes keyjoint coordinates like head, right ankle, right elbow, left ankle, left elbow, right knee, and left knee keyjoints coordinates.

$$\boldsymbol{H} = \left|\mathbf{T}_j - \mathbf{B}_j\right| \tag{7}$$

where T_j indicates the top keyjoint coordinates, B_j indicates the bottom keyjoint coordinates.

3.3 ConvST-LSTM: the proposed model

In this section, the final preprocessed 3D keyjoint coordinates are inputted into the proposed deep learning network. We have used the sequential fusion of CNNs, Conv-LSTM & ST-LSTM to propose the ConvST-LSTM network.

3.3.1 Convolutional neural network architecture

Initially, the human action recognition has been executed by applying CNNs approach [68]. Consider, $X_t^0 = [X_1, X_2, X_3, \ldots\ldots\ldots\ldots, X_n]$ as the input vector, where n indicates the input samples and output of convolutional layers can be defined as follows:

$$\boldsymbol{C}_i^{l,j} = \boldsymbol{\sigma}\left(\boldsymbol{B}_{\mathbf{j}} + \sum_{\boldsymbol{m}=1}^{\boldsymbol{M}} \boldsymbol{W}_{\boldsymbol{m}}^{\boldsymbol{j}} * \boldsymbol{X}_{\boldsymbol{i}+\boldsymbol{m}-1}^{0,\boldsymbol{j}}\right) \tag{8}$$

here l corresponds to an index of convolutional layer; σ depicts the nonlinear sigmoid-activation function; whereas B represents the bias vector corresponds to jth feature-map; filter size of CNN is indicated by M; indicates the weight metrics for the jth feature map is indicated by W_m^j; mth is the filter index.

The input frames in the proposed model consist of three input channels, namely sequences, keyjoint, and coordinates, which resemble to the x, y, and z directions, respectively. Each input frame has a resolution of 125 × 25 × 3 pixels and contains information about the movement sequences, keyjoint positions, and spatial coordinates. In convolution layer, 6 filters are passed together with configured size of kernel, padding, and SoftMax functions in the hidden layer in order to avoid the vanishing gradient problem. Max-pooling is used as a pooling operation to estimate the maximum value for feature map, and diminish the processing time by reducing the dimensionality of the frame. Then, output from the hidden layer has passed to FC dense layers. Finally, SoftMax function shows the prediction score of the action classes.

3.3.2 Spatiotemporal LSTM

Before moving to ST-LSTM, let's recap LSTM [69], which consists of 3 memory cells (gates) and escape the vanishing gradient issue. These are: (a) forget cell: a binary gate that decides how much information to pass through. (b) Input cell: decides whether the current information can be stored in the unit cell and (c) Output cell: contains sigmoid-activation gate, which decides which information to show as output. Lastly, the *tanh* layer is used to pass the cell state and further multiply it with the final output obtained from the output cell.

The equations which define the activity of each cell can be formulated as follows:

$$\boldsymbol{i_t} = \boldsymbol{\sigma}(\boldsymbol{W_{X_i}X_t} + \boldsymbol{W_{H_i}H_{t-1}} + \boldsymbol{W_{C_i}C_{t-1}} + \boldsymbol{B_i}) \tag{9}$$

$$\boldsymbol{f_t} = \boldsymbol{\sigma}(\boldsymbol{W_{X_f}X_t} + \boldsymbol{W_{\mathrm{H}_f}H_{t-1}} + \boldsymbol{W_{\mathrm{C}_f}C_{t-1}} + \boldsymbol{B_f}) \tag{10}$$

$$\boldsymbol{o_t} = \boldsymbol{\sigma}\left(\boldsymbol{W_{X_o}X_t} + \boldsymbol{W_{H_o}H_{t-1}} + \boldsymbol{W_{\mathrm{C}_f}C_t} + \boldsymbol{B_o}\right) \tag{11}$$

$$\boldsymbol{C_t} = \boldsymbol{f_t}\mathrm{C}_{t-1} + \boldsymbol{i_t}\mathbf{tanh}\left(\boldsymbol{W_{X_c}X_t} + \boldsymbol{W_{\mathrm{H_c}}H_{t-1}} + \boldsymbol{B_c}\right) \tag{12}$$

$$\boldsymbol{H_t} = \boldsymbol{o_t}\mathbf{tanh}(\boldsymbol{C_t}) \tag{13}$$

here, W_i, W_f, W_o indicates weight matrices of forget (f), input (i) and output (o) gates, respectively; $X_t \in$ input fed to LSTM cells unit at t time; σ depicts the sigmoid-activation function, whereas *tanh* depicts the hyperbolic-tangent function (both nonlinear functions); C_t indicates memory cell state within the LSTM. B_i, B_f, B_o, and B_c indicates the bias

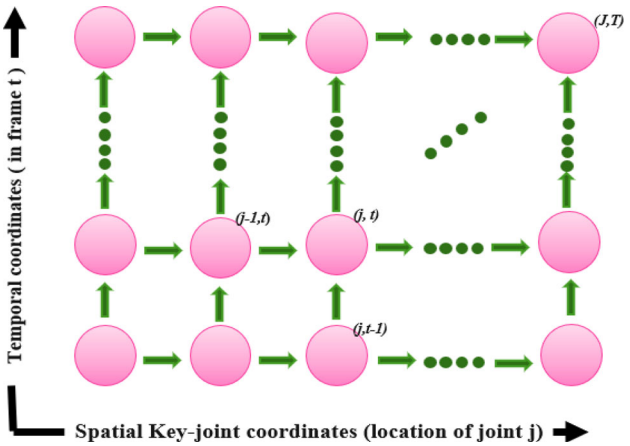


Fig. 4 Illustration of ST-LSTM cell [44]. For spatial domain, the skeletal keyjoints in each frame are aligned & feed sequentially. For temporal domain, the skeletal keyjoints are feed sequentially over the frames

vectors on forget, input & output gates, and memory cell c, respectively. Internal frame input keyjoint coordinates of each cell in the ST-LSTM model are represented in Fig. 4. The skeletal keyjoints are arranged in spatial direction and input as a chain whereas the corresponding keyjoints are inputted over various frames for temporal direction sequentially. Especially, each ST-LSTM cell is feed for a new input $(x_{j,t})$, where $x \in$ new input feed for 3D position of body keyjoint j in frame time t, the hidden layer $(h_{j,t-1})$ of the same keyjoint$j$ and the hidden layer $(h_{j-1,t})$ for the previous keyjoint *j-1* in same frame t, here j indicates the indices of keyjoint, i.e., $j \in \{1..j..J\}$ and t indicates the indices of frames, i.e.,$t \in \{1,..t..,T\}$,. An ST-LSTM unit cell consists of an input cell $(i_{j,t})$, 2-forget cells correspond to the sources of context information, i.e., temporal dimension $(f_{j,t}^T)$ & spatial domain $(f_{j,t}^S)$, in conjunction with an output gate $(o_{j,t})$.

The equations for ST-LSTM are formulated as introduced in [44]:

$$\begin{pmatrix} i_{j,t} \\ f_{j,t}^{(S)} \\ f_{j,t}^{(T)} \\ o_{j,t} \\ u_{j,t} \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \left(W \begin{pmatrix} x_{j,t} \\ h_{j-1,t} \\ h_{j,t-1} \end{pmatrix} \right) \tag{14}$$

$$\boldsymbol{C}_{j,t} = \boldsymbol{i}_{j,t} \odot \boldsymbol{u}_{j,t} + \boldsymbol{f}_{j,t}^{(S)} \odot \boldsymbol{c}_{j-1,t} + \boldsymbol{f}_{j,t}^{(T)} \odot \boldsymbol{c}_{j,t-1} \tag{15}$$

$$\boldsymbol{h}_{j,t} = \boldsymbol{o}_{j,t} \odot \mathbf{tanh}(\boldsymbol{c}_{j,t}) \tag{16}$$

where $c_{j,t}$ indicates the cell state; $h_{j,t}$ indicates the hidden input layer in ST-LSTM unit at the spatiotemporal steps for

keyjoint j and frame time t; the modulated input frame is indicated by $u_{j,t}$; $\odot$ represents framewise product for each unit and W indicates an affine transformation within the weight.

3.3.3 ConvST-LSTM-Net architecture

Several works [53, 54] have demonstrated that each action sequence has a subset of informative keyjoints. In contrast, some keyjoints may be irrelevant in order to recognize the action classes with proper information. Therefore, for obtaining high accuracy in human action recognition the informative skeletal keyjoints have been identified while focusing on their features vector. At the same time in order to recognize human behavior, we must preferentially concentrate on the informative keyjoints (coordinates for feed), ignoring the features of the irrelevant keyjoints.

This model has been executed by taking a sequential fusion of CNN, ST-LSTM (combination of LSTM and spatiotemporal based recognition), and FC layers. Here, CNNs are pre-owned for feature extraction, ST-LSTMs are used in sequence prediction for spatiotemporal feature extraction, and the features dense layers are used for mapping. For classification, the outputs from CNN's hidden layer are fed to the ST-LSTM layers and then GAP (Global Pooling Layer) is used to flatten the data followed by FC layers within the model.

The transformation equations for ConvST-LSTM-Net can be given as:

$$\mathcal{F}_{j,t}^{(T)} = \sigma\left(W_{X_{\mathcal{F}}} X_{j,t} + W_{H_{\mathcal{F}}} H_{j,t-1} + B_{\mathcal{F}}\right) \tag{17}$$

$$\mathcal{F}_{j,t}^{(S)} = \sigma\left(W_{X_{\mathcal{F}}} X_{j,t} + W_{H_{\mathcal{F}}} H_{j,t-1} + B_{\mathcal{F}}\right) \tag{18}$$

$$\tilde{\mathrm{I}}_{j,t} = \sigma\left(W_{X_{\tilde{\mathrm{I}}}} Xj_{,\,t} + W_{\mathrm{H}_{\tilde{\mathrm{I}}}} H_{j,t-1} + B_{\tilde{\mathrm{I}}}\right) \tag{19}$$

$$\tilde{\mathrm{O}}_{j,t} = \sigma\left(W_{X\ O} X_t + W_{H\ O} H_{t-1} + B_o\right) \tag{20}$$

$$C_{j,t} = f_{j,t} C_{t-1} + i_{j,t} \tanh(W_{X_c} X_t + W_{H_c} H_{j,t-1} + B_c \tag{21}$$

$$u_{j,t} = \tanh(W_{X_u} * X_t + W_{H_u} H_{t-1} + B_u \tag{22}$$

$$H_t = o_t \odot \tanh(C_t) \tag{23}$$

where $X_{j,t}$, $Cj,_t$, $H_{j,t}$, $F_{j,t}$, $I_{j,t}$ indicates inputs states, cells states, hidden states, forget cells, input cells for keyjoint j in frame time t; u_t input modulation gates and $\tilde{O}_t$ is the output cells; C_t is the memory cell used for aggregating the states information controlled by the cells. Figure 5 depicts about the ST-LSTM layer for each unit cell.

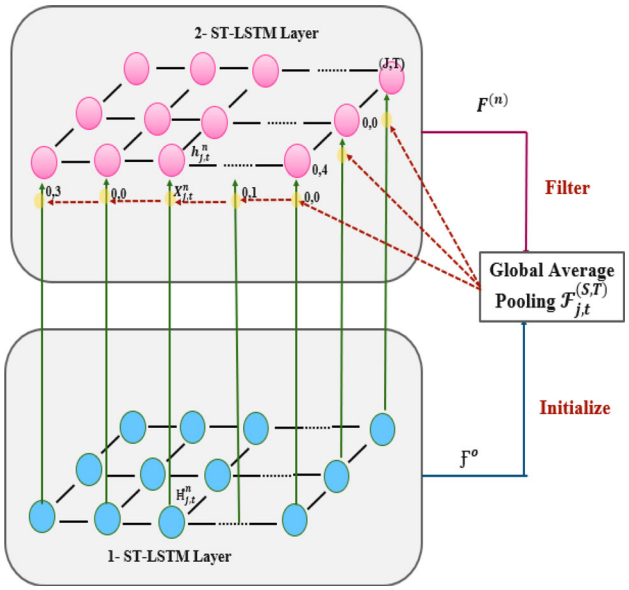


Fig. 5 The ConvST-LSTM network for ST-LSTM layer in each unit cell

Model training ConvST-LSTM-Net was trained on the frame samples obtained from the videos, keypoint recognition, followed by the fusion of spatiotemporal model consisting of ConvLSTMs. The model was trained on 150 epochs on a machine having AMDRyzen7 5800H processor, 8 GB RAM, and Graphics: NVIDIA GeForce RTX 3050 M, GPU, having a learning rate of 0.001 after repeated hyperparameter tuning.

For setup, Keras API version 2.3 of Python along with TensorFlow version 2.3.0 has been used in the backend to build the spatiotemporal model. To increase the code's reusability and readability, some helper functions are initially defined from the python libraries. Along with an optimum value has been set for the user-defined hyperparameters like size, no. of layers, iteration, epochs, no. of batch sizes, and learning rate. The training sample data with various batch size is feed to the model and get trained over 150 epochs. In first time-distributed CNN layer, we use 32 filters with kernel size 3 and its output is then regularized to attain faster convergence. Then, max-pooling is added to reduce computational costs. Dropout layer benefited to avoid the overfitting where 50% of weights are dropped randomly. For next time-distributed CNN layers, different size of filters is practices after performing feature extraction, we apply an additional dropout layer with a rate of 40%. At step 3, we use GAP layer through which the output of CNN layer is flattened to 1*56 dimension.

Further, ST-LSTM is used to handle the sequential action data of the tracked person's keyjoints coordinates. The ST-LSTM layer's output is passed to the time-distributed FC dense layer. At last, SoftMax layer gives the framewise probabilities for each action classes. The architecture of the

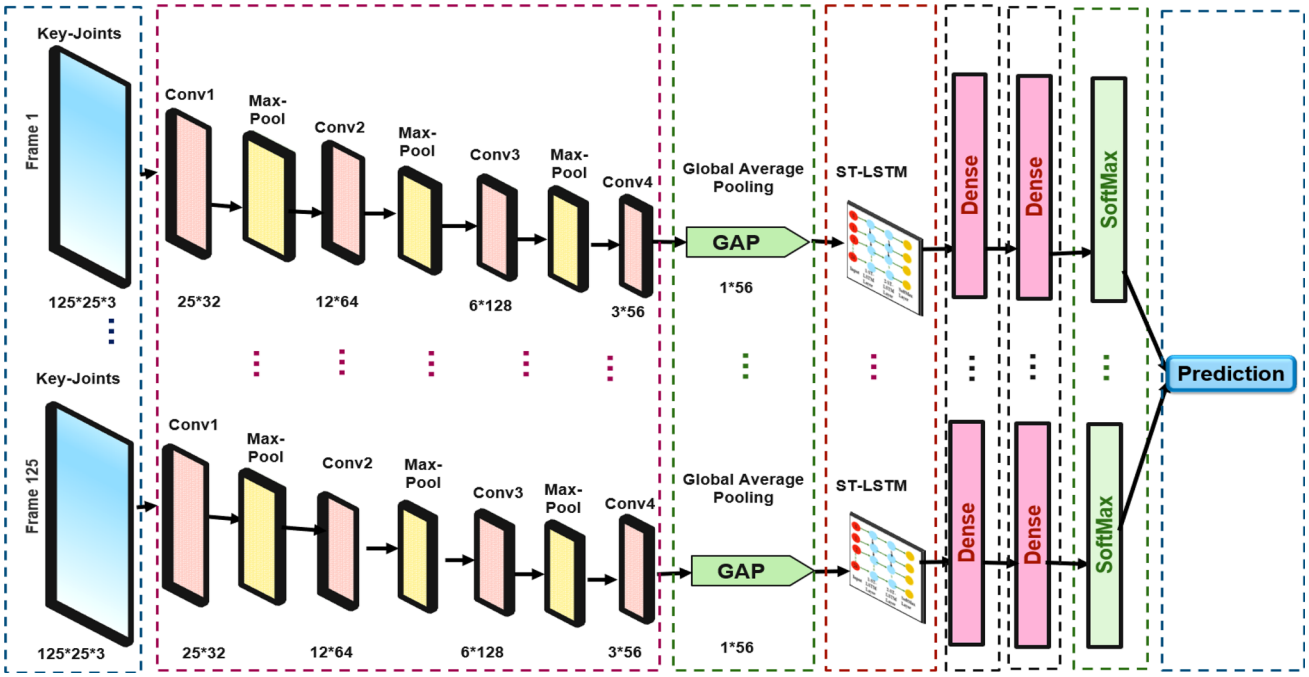


Fig. 6 Block architectural diagram of ConvST-LSTM-Net model for human action recognition. Starting from left side, the input frames clipped from videos; time-distributed convolutional layers including max-pooling, GAP, ST-LSTMs, FC dense layer followed by SoftMax function layer that results as a prediction of action

proposed ConvST-LSTM model is illustrated in Fig. 6. Further, Adam optimizer is help in optimizing the cost function and uses gradient clipping within the code. The hyperparameters such as checkpoint-path, saver-function, epochs, iterations, filter size, kernel size, and test & train data have been set for training purpose. Moreover, the proposed model utilized the stopping criteria with the value of 50. That means the training will be terminated only if there is no improvement in the monitor performance measure for 50 epochs or iterations in a row. This helps to prevent the entire model fall in to local optima. It has been found that the performance is excessively upgraded by using a sequential way.

4 Experimental results and analysis

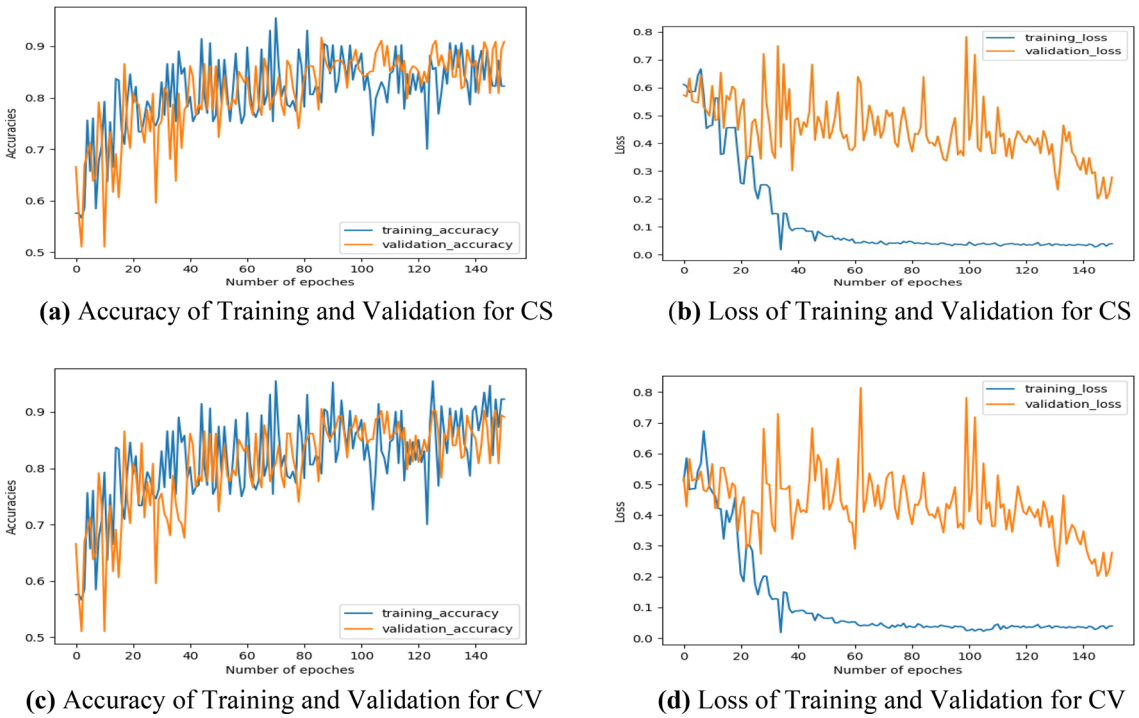
This section discusses about the implementation trait of the proposed model on various benchmarks with its training hyperparameters. We has evaluate the performance of ConvST-LSTM-Net model on three publicly available benchmarks, i.e., the NTU RGB + D 60 dataset [32], UT-Kinect dataset [33], UP-Fall-Detection Dataset [34], UCF101 [35], and HDMB51 dataset [36] for skeleton-based data.

4.1 Experiments on NTU RGB + D 60 dataset

The NTU RGB + D60 [32] is publicly available dataset used for human action recognition consisting of total 56,880 samples having 60 activity classes collected over 40 subjects in it. In this dataset, activities are classified into three categories having 40 daily living activities (drinking, standing, reading, happing, etc.), 9 medical conditions-related activities (sneezing, staggering, falling, vomiting, etc.), and 11 common activities (punching, kicking, hugging, etc.) based on multimodal information of the daily action characterization, along with 3D skeletal keyjoint, RGB-videos, masked-depth maps, full-depth maps, and infrared sequences data. The annotations provide the 3D location in x, y, z-dimension of each keyjoint in the camera coordinate system. It has total 25 key points per subject and each clip has 2 subjects. The evaluation has done on two protocols: Cross-Subject (CS) and Cross-View (CV).

For performing experiments, we choose 5 action classes (Stand, Sit, Run, Walk, Fall) contains 150 clips in each class. The two benchmarks for evaluation are set as: (1) Cross-subject (CS) contains 400 clips from 5 subjects, used for training; and the 100 clips for validation. (2) Cross-view (CV) contains 450 from 5 subjects used for training and 150 clips for validation. The proposed ConvST-LSTM-Net model surpasses the ConvLSTM network in [68] by 4.3% with the CS evaluation protocol and 3.1% with the CV evaluation protocol. This demonstrates that spatiotemporal skeleton-based recognition approaches in LSTM networks bring significant improvement. The comparative analysis for the results of the proposed ConvST-LSTM-Net model with state-of-the-art approaches is enumerated in Table 2.

The trade-off curves for training accuracy & loss and validation accuracy & loss on the benchmark of NTU RGB +



(a) Accuracy of Training and Validation for CS

(b) Loss of Training and Validation for CS

(c) Accuracy of Training and Validation for CV

(d) Loss of Training and Validation for CV

Fig. 7 Trade-off curves for model's Training and Validation Accuracy versus Training and Validation Loss on the NTU RGB + D 60 benchmark dataset

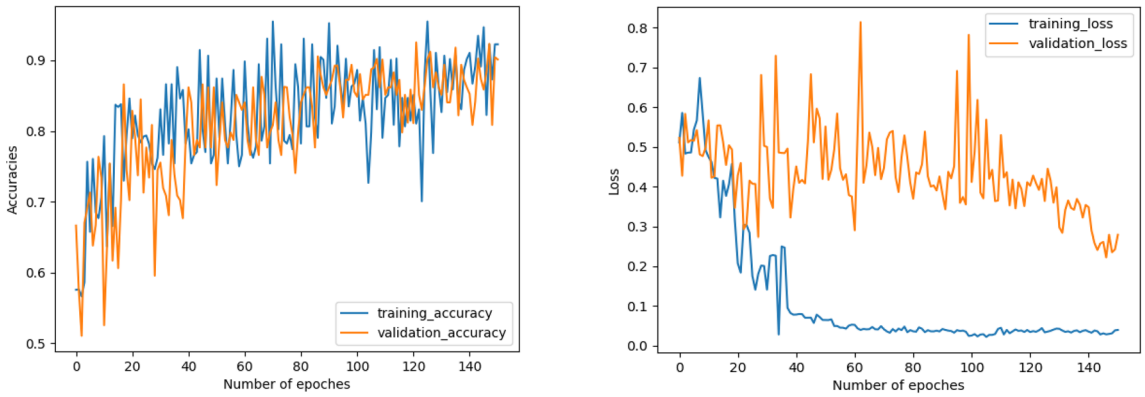
Table 2 Experimental Results on NTU RGB + D 60 Dataset for skeletal sequence data

Methods	CS (%)	CV (%)
Deep-LSTM [70]	56.3	64.1
ST-LSTM [44]	69.2	77.7
ST-LSTM + Global(1) [71]	70.5	79.5
ST-LSTM + Global(2) [71]	70.7	79.4
Conv-LSTM [68]	76.2	83.2
Conv-GRU [72]	88.9	90.1
LA-GCN [73]	90.9	89.28
TD-GCN [74]	91.82	94.2
SkeletonGCL [75]	89.2	90.3
ConvST-LSTM-Net	91.72	90.5

D 60 dataset for its two-evaluation protocol, i.e., CS and CV is illustrated in Fig. 7. Training and validation accuracy increases with time as depicted in Fig. 7a, c and finally, the growth rate reaches a steady-state value. Figure 7b and d depict the loss curve, illustrating the gradual decrease of validation loss with increasing epochs. To evaluate the model's performance, the weights are saved from the epochs that achieve the highest validation accuracy. The loss curve is shown in Fig. 7b and d, which demonstrates how the validation loss gradually decreases by increasing epoch. For testing, the weights from the epochs with the maximum validation accuracy are saved.

4.2 Experiments on the UT-Kinect dataset

The UT-Kinect dataset [34] is publicly available and was taken through a single stationary Kinect comprised of total 10 subjects that took total 10 action types (walking, stand up, pick up, carry, sit down, throw, push, pull, wave hands, clap hands). Each subject performs each action twice. 3-channels were captured for (i) RGB, (ii) depth, and (iii) skeleton keyjoint locations. We have only recorded the frames when the skeleton of human body was tracked. To assess the proposed method on this dataset, the standard leave-one-out-cross-validation protocol has been followed. Table 3 provides the comparative result of the proposed ConvST-LSTM-Net model with state-of-the-art approaches. The trade-off curves for the model accuracy and loss on the UT-Kinect dataset has been illustrated in Fig. 8. It is observed from these curves that the proposed methodology offers exceptional accuracy during training and moderate accuracy in the validation process. For training process, the model causes low and for validation process it causes moderate loss.



(a) Accuracy of Training and Validation

(b) Loss of Training and Validation

Fig. 8 Trade-off curves for Model Training and Validation Accuracy & Model Training and Validation Loss on the UT-Kinect dataset

Table 3 Experimental Results on UT-Kinect

Method	Accuracy (%)
Histogram of 3d joints [33]	88.9
ST-LSTM [44]	87.0
ST-LSTM + Global(1) [71]	91.9
ST-LSTM + Global(2) [71]	90.8
Conv-LSTM [68]	90.2
Conv-GRU [72]	89.99
ConvST-LSTM-NET	92.0

Table 4 Experimental Results on UP-Fall Detection Dataset

Method	Accuracy (%)
GCA-LSTM [71]	88.5
Conv-LSTM [68]	87.6
Conv-GRU [72]	88.8
ConvST-LSTM	89.0

Table 5 Experimental Results of ConvST-LSTM on the UCF101 Dataset

Method	Accuracy (%)
GCA-LSTM [71]	84.2
Conv-LSTM [68]	83.3
Conv-GRU [72]	86.28
PYSKL [76]	88.89
ConvST-LSTM	92.8

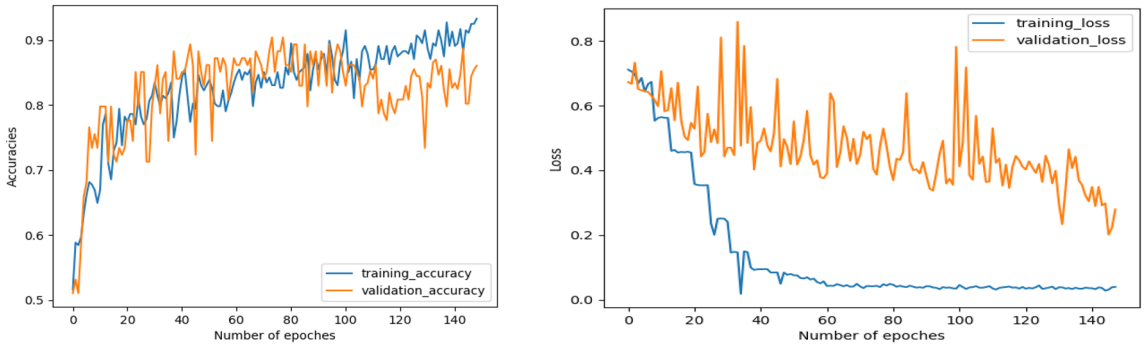
4.3 Experiments on the UP-fall detection dataset

The UP-fall detection [34] is the large-scale multimodal dataset collected by using vision-wearable, and ambient sensors. It includes Activity for Daily Livings (ADLs-850 GB), collected by 17 healthy persons including 9 male, 8 females individuals. It has total 11 actions, i.e., 6 basic actions for daily living: walk, sit, stand, picking-up an item, laying, jump and 5 fall-actions: fall-forward via knees, fall-forward via hands, fall-sitting in an empty chair, fall backward and fall-sideward). Two cameras were set up to capture the subject's front views as well as its side views. A total of 589,418 sample image frames are there taken from both cameras. Total size of this vision dataset was 277 GB. For performing experiments, we choose 5 action classes (i.e., Stand, Sit, Run, Walk, Fall) contains 1000 clips in each class in which 800 clips used for training, and the 200 clips for validation. Table 4 gives the comparative results of the proposed ConvST-LSTM-Net with various state-of-the-art methods.

Figure 9 illustrates the trade-off curves for (a) accuracy of training and validation vs. (b) loss of training and validation. It is observed from these curves that the proposed methodology offers exceptional accuracy during training and moderate accuracy in the validation process. For training process, the model causes low and for validation process it causes moderate loss.

4.4 Experiments on the UCF101 dataset

The UCF101 [35] is a popular action recognition dataset that contains 13,320 video clips from 101 action categories. The action videos are clustered in 25 groups, where each group contains 4–7 videos of an action. The action categories can be classified into five distinct types, i.e., (a) Human-Object Interaction (b) Body-Motion (c) Human–Human Interaction (d) Playing Musical Instruments (e) Sports. For performing experiments, we choose 5 action classes from body motion categories contains total 17 body motion clips. Table 5 gives



(a) Accuracy of Training and Validation

(b) Loss of Training and Validation

Fig. 9 Trade-off curves for Model Training and Validation Accuracy versus Model Training and Validation Loss on UP-Fall Detection dataset

Table 6 Experimental Results on the HMDB51 Dataset

Method	Accuracy (%)
GCA-LSTM [71]	82.3
Conv-LSTM [68]	81.2
Conv-GRU [72]	80.8
PYSKL [76]	69.4
ConvST-LSTM	91.86

the comparative results of the proposed ConvST-LSTM-Net with various state-of-the-art methods.

The trade-off curves for the model accuracy and loss on the UCF101 dataset has been illustrated in Fig. 10, i.e., (a) accuracy of training and validation versus (b) loss of training and validation. For training process, the model causes low and for validation process it causes moderate loss.

4.5 Experiments on the HMDB51 dataset

The HMDB51 [36] dataset is a commonly used benchmark dataset for action recognition in videos, which consists of video clips from various sources like movies, YouTube. From 2 GB, total 7000 clips distributed in 51 action classes. The actions categories can be divided into five types: (a) General facial actions. (b) Facial actions with object manipulation. (c) General body movements. (d) Body movements with object interaction. (e) Body movements for human interaction. The video clips have varying durations and resolutions. For performing experiments, we select the general body movements action classes in which 5 action clips are taken (i.e., Stand up, Sit down, Run, Walk, Fall). Each action classes contains minimum of 101 clips. Among them 80% are used of training and 20% are used for validation. Table 6 gives the comparative results of the proposed ConvST-LSTM-Net with various methods.

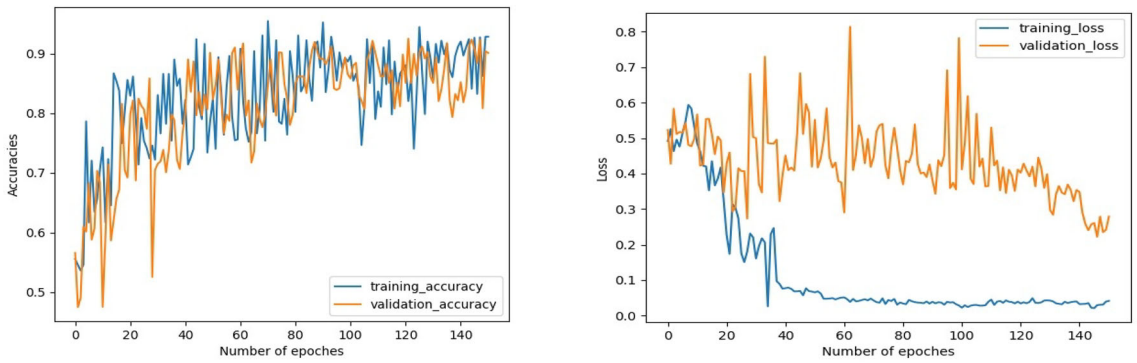
Figure 11 illustrates the trade-off curves for (a) accuracy of training and validation versus (b) loss of training and validation on HDMB51 dataset. Form this, it is observed that the proposed methodology achieves outstanding accuracy during the training process and moderate accuracy during the validation process. The model exhibits low loss during training and moderate loss during validation.

4.6 Multimodal analysis over standard performance measures

This section discusses the results analysis gained on the proposed ConvST-LSTM-Net. The performance of the model has been measured on different performance metrics, i.e., Precision, Recall, F1-score, and Accuracy. Figure 12 displays the accuracy, precision, recall, and F1-score on various benchmarks. The accuracies and losses are plotted for 150 epochs. The proposed ConvST-LSTM-Net results in a better accuracy. The effectiveness of the proposed model is verified on various benchmarks, i.e., NTU RGB + D 60, UT-Kinect, UP-Fall Detection, UCF101, and HDMB51 datasets, where the model outperforms state-of-the-art methods. Figure 13 illustrates the human action recognition results obtained in different benchmarks datasets with framing the bounding box over the tracked human. We observed that the performance of the model is sufficiently high.

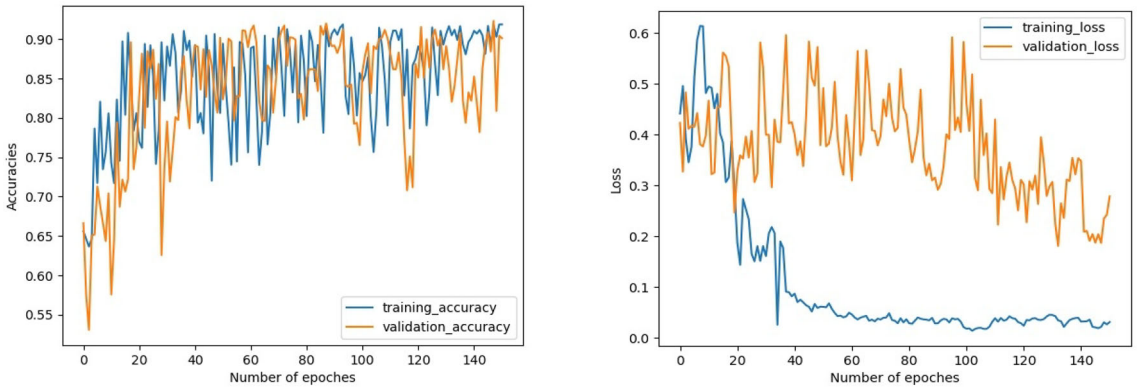
5 Conclusions and future prospective

Human action recognition has gained a large prominence in today's era, but few limitations are there in their application areas despite having networks that could achieve good results. In this paper, we improved the internal cell structure of the ST-LSTM unit and successfully proposed a ConvST-LSTM having high accuracy & reliability. The model is based on a spatiotemporal LSTM module, uses video frames



(a) Accuracy of Training and Validation **(b)** Loss of Training and Validation

Fig. 10 Trade-off curves for Model Training and Validation Accuracy versus Model Training and Validation Loss on UCF101 Dataset



(a) Accuracy of Training and Validation **(b)** Loss of Training and Validation

Fig. 11 Trade-off curves for Model Training and Validation Accuracy versus Model Training and Validation Loss on HDMB51 Dataset

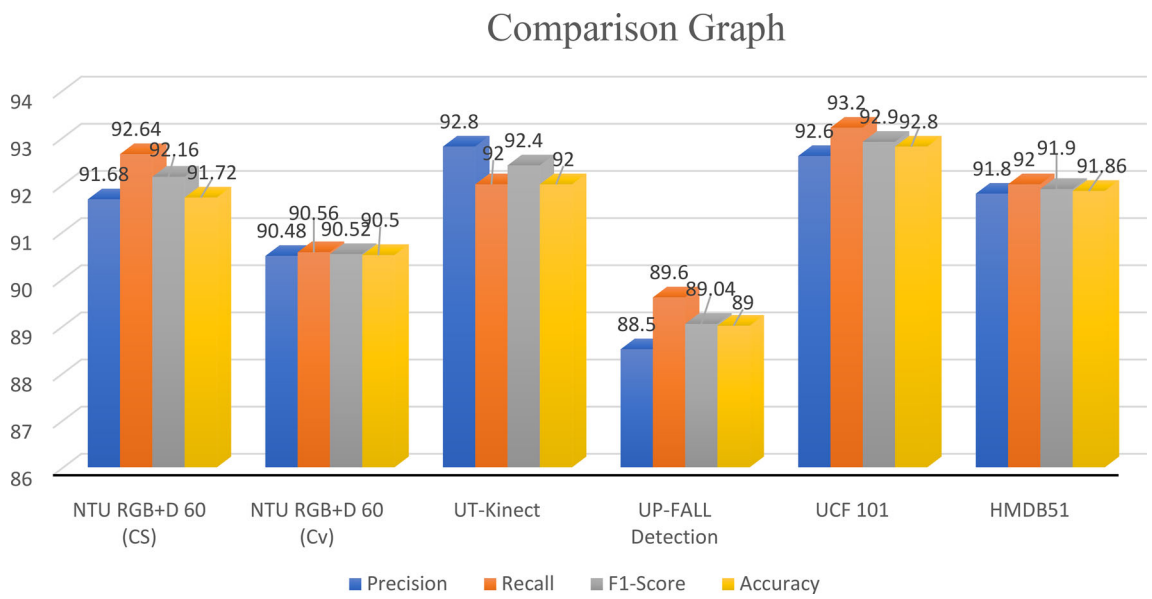


Fig. 12 Comparative Stats of Standard Performance Measure over different datasets

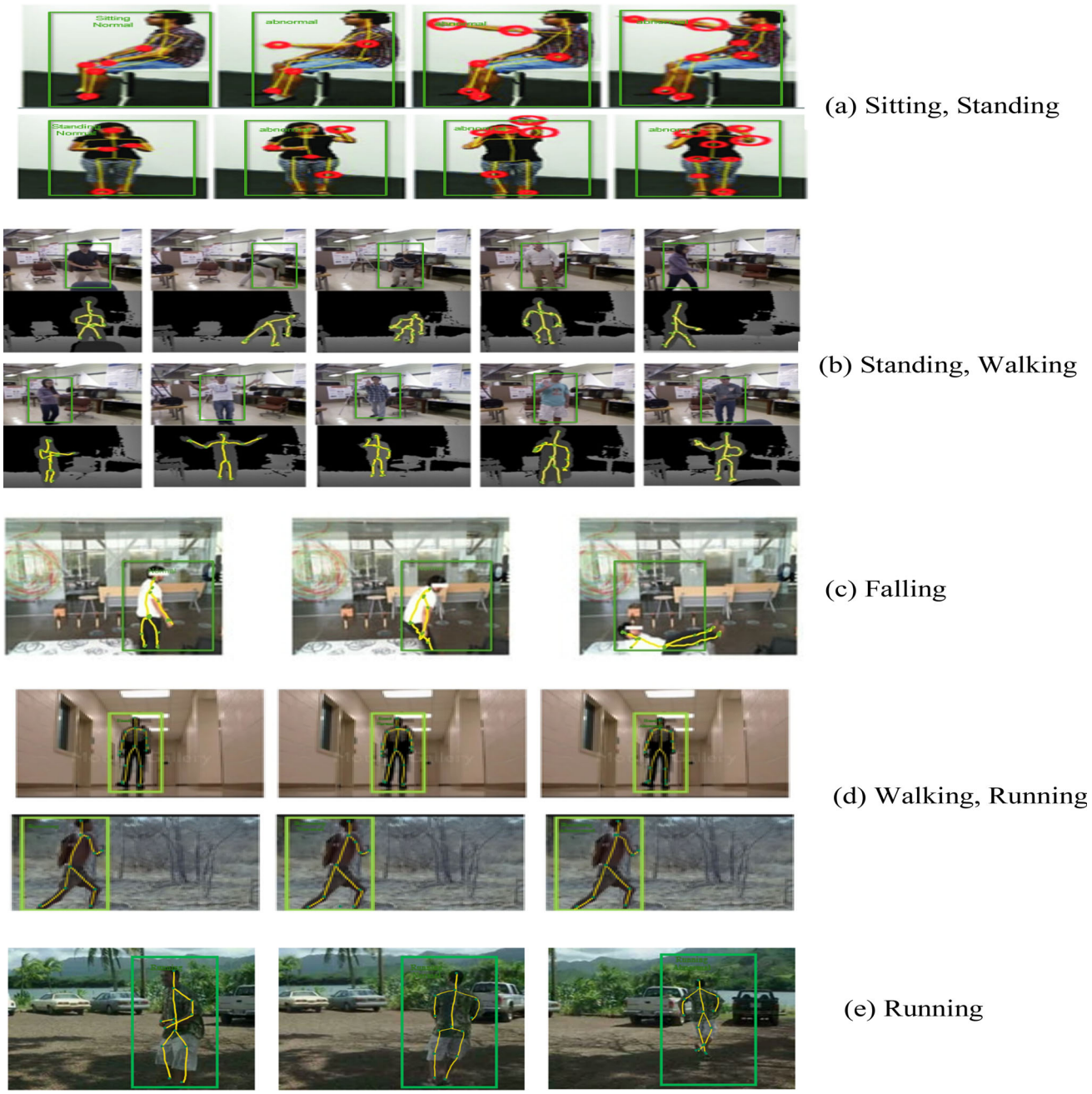


Fig. 13 Illustration of the Human Action Recognition on various benchmarks. Starting from left–right **a** NTU RGB + D 60 Dataset: Sitting, Standing **b** UT-Kinect Dataset: Standing, Walking **c** UP-Fall Detection Dataset: Fall **d** UCF101 Dataset: Walking, Running **e** HMDB51 Dataset: Running

and skeleton-based features and has the robust capability for selecting the informative keyjoints in each frame while ignoring the irrelevant keyjoints of the skeleton sequence. The model is independent of the camera orientation, clothing, background noise, etc., which can effectively recognize suspicious actions related to human activity. Finally, the experimental results show better performance and achieve good accuracy for skeleton-based anomaly activity recognition. However, the consequences of growing population and rise in ever-challenging activities fosters the need to introduce a more promising predictive methodology for recognizing human behavior that proffers a practical alternative solution for the security and protection of people from daily risks in life. With the future perspective, we can use a graph oriented spatiotemporal base data to represent humans and

objects. Moreover, GCN can also be used for the classification and detection of unsuspicious activity of human behaviors.

Acknowledgements Not applicable.

Author contributions Roshni Singh, carried out the related studies, participated in the sequence alignment and drafted the manuscript along with performances and statistical analysis. Dr. Abhilasha Sharma, conceived of the study and participated in its design and coordination and helped to draft the manuscript. All authors read and approved the final manuscript.

Funding Not applicable.

Availability of data and materials Not applicable.

Declarations

Conflict of interest The authors have no relevant financial or nonfinancial interest to disclose. The authors have no competing interest to declare that are relevant to the content of this article. All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or nonfinancial interest in the subject matter or materials discussed in this manuscript. The authors have no financial or proprietary interest in any material discussed in this article.

References

1. Chang Y, Tu Z, Xie W, and Yuan J (2020). Clustering driven deep autoencoder for video anomaly detection. In European Conference on Computer Vision (pp. 329–345). Springer, Cham.
2. Zhang D, He L, Tu Z, Zhang S, Han F, Yang B (2020) Learning motion representation for real-time spatio-temporal action localization. Pattern Recogn 103:107312
3. Niu W, Long J, Han D and Wang Y-F, Human activity detection and recognition for video surveillance, in 2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763), vol. 1, June 2004, pp. 719–722 Vol.1.
4. Valera M, Velastin SA (2005) Intelligent distributed surveillance systems: a review. IEE Proc–Vision, Image Signal Process 152(2):192–204
5. Lin W, Sun MT, Poovandran R, and Zhang Z (2008), Human activity recognition for video surveillance, in 2008 IEEE International Symposium on Circuits and Systems, pp. 2737–2740.
6. Kalimuthu S, Perumal T, Yaakob R, Marlisah E, and Babangida L (2021), Human activity recognition based on smart home environment and their applications, challenges. In 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE) (pp. 815–819). IEEE.
7. Poppe R (2010) A survey on vision-based human action recognition. Image Vis Comput 28(6):976–990
8. Chaquet JM, Carmona EJ, Fernández-Caballero A (2013) A survey of video datasets for human action and activity recognition. Comput Vis Image Underst 117(6):633–659
9. Patrona F, Chatzitofis A, Zarpalas D, Daras P (2018) Motion analysis: Action detection, recognition, and evaluation based on motion capture data. Pattern Recogn 76:612–622
10. Vishwakarma DK, Dhiman A, Maheshwari R, Kapoor R (2015) Human motion analysis by fusion of silhouette orientation and shape features. Procedia Comput Sci 57:438–447
11. Yao H, Hu X (2023) A survey of video violence detection. Cyber-Phys Syst 9(1):1–24
12. Yang Y, Liu G, Gao X (2022) Motion guided attention learning for self-supervised 3D human action recognition. IEEE Trans Circuits Syst Video Technol 32(12):8623–8634
13. Duan H, Wang J, Chen K, and Lin D (2022), DG-STGCN: dynamic spatial-temporal modeling for skeleton-based action recognition. arXiv preprint arXiv:2210.05895.
14. Liu S, Bai X, Fang M, Li L, Hung CC (2022) Mixed graph convolution and residual transformation network for skeleton-based action recognition. Appl Intell 52(2):1544–1555
15. Abdulhussein AA, Hassen OA, Gupta C, Virmani D, Nair A, and Rani P (2022), Health monitoring catalogue based on human activity classification using machine learning. Int J Electrical Comput Eng, 12(4): (2088–8708).
16. Andrade-Ambriz YA, Ledesma S, Ibarra-Manzano MA, Oros-Flores MI, Almanza-Ojeda DL (2022) Human activity recognition using temporal convolutional neural network architecture. Expert Syst Appl 191:116287
17. Qiu S, Zhao H, Jiang N, Wang Z, Liu L, An Y, Fortino G (2022) Multi-sensor information fusion based on machine learning for real applications in human activity recognition: State-of-the-art and research challenges. Inform Fusion 80:241–265
18. Wu L, Zhang C, Zou Y (2023) SpatioTemporal focus for skeleton-based action recognition. Pattern Recogn 136:109231
19. Mahdikhanlou K, Ebrahimnezhad H (2023) 3D hand pose estimation from a single RGB image by weighting the occlusion and classification. Pattern Recogn 136:109217
20. Dallel M, Havard V, Dupuis Y, and Baudry D (2022), A sliding window based approach with majority voting for online human action recognition using spatial temporal graph convolutional neural networks. In 2022 7th International Conference on Machine Learning Technologies (ICMLT) (pp. 155–163).
21. Sánchez-Caballero A, Fuentes-Jiménez D, and Losada-Gutiérrez C (2022) Real-time human action recognition using raw depth video-based recurrent neural networks. Multimedia Tools Appl, 1–23.
22. Yue R, Tian Z, and Du S (2022) Action recognition based on RGB and skeleton data sets: A survey. Neurocomputing.
23. Khaire P, Kumar P (2022) Deep learning and RGB-D based human action, human–human and human–object interaction recognition: a survey. J Vis Commun Image Represent 86:103531
24. Ding C, Wen S, Ding W, Liu K, Belyaev E (2022) Temporal segment graph convolutional networks for skeleton-based action recognition. Eng Appl Artif Intell 110:104675
25. Setiawan F, Yahya BN, Chun SJ, Lee SL (2022) Sequential inter-hop graph convolution neural network (SIhGCN) for skeleton-based human action recognition. Expert Syst Appl 195:116566
26. Khowaja SA, & Lee SL (2022) Skeleton-based human action recognition with sequential convolutional-LSTM networks and fusion strategies. Journal of Ambient Intelligence and Humanized Computing, 1–18.
27. Hou R, Wang Z, Ren R, Cao Y, and Wang Z (2022). Multi-channel network: constructing efficient GCN baselines for skeleton-based action recognition. Comput Gr.
28. Gao BK, Dong L, Bi HB, Bi YZ (2022) Focus on temporal graph convolutional networks with unified attention for skeleton-based action recognition. Appl Intell 52(5):5608–5616
29. Xu W, Wu M, Zhu J, Zhao M (2021) Multi-scale skeleton adaptive weighted GCN for skeleton-based human action recognition in IoT. Appl Soft Comput 104:107236
30. Song YF, Zhang Z, Shan C, and Wang L (2020), Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition. In proceedings of the 28th ACM international conference on multimedia (pp. 1625–1633).
31. Wang L, Suter D (2007) Learning and matching of dynamic shape manifolds for human action recognition. IEEE Trans Image Process 16(6):1646–1661

32. Shahroudy A, Liu J, Ng T-T, and Wang G (2016), Ntu rgb+d: a large scale dataset for 3d human activity analysis, in CVPR, 2016.
33. Xia L, Chen C-C, and Aggarwal JK, View invariant human action recognition using histograms of 3D joints, in Proc. CVPR, 2012, pp. 20–27 (2012)
34. Martínez-Villaseñor L, Ponce H, Brieva J, Moya-Albor E, Núñez-Martínez J, Peñafort-Asturiano C (2019) UP-fall detection dataset: a multimodal approach. Sensors 19(9):1988
35. Soomro K, Zamir AR, and Shah M (2012) UCF101: a dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402
36. Kuehne HH, Jhuang E, Garrote T, Poggio and Serre T (2011) HMDB: a large video database for human motion recognition, 2011 International Conference on Computer Vision, 2011, 2556–2563, https://doi.org/10.1109/ICCV.2011.6126543.
37. Vemulapalli R, Arrate F, and Chellappa R (2014), Human action recognition by representing 3d skeletons as points in a lie group, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 588–595.
38. Vemulapalli R and Chellapa R, Rolling rotations for recognizing human actions from 3d skeletal data, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4471– 4479.
39. Ke Q, Bennamoun M, An S, Sohel F, and Boussaid F (2017), A new representation of skeleton sequences for 3d action recognition, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 3288–3297.
40. Li, B, Dai, Y, Cheng X, Chen H, Lin Y, and He M (2017), Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep cnn, in 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW). IEEE, 2017, pp. 601–604.
41. Li C, Zhong Q, Xie D and Pu S (2017), Skeleton-based action recognition with convolutional neural networks, in 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW). IEEE, 2017, pp. 597–600.
42. Liu M, Liu H, Chen C (2017) Enhanced skeleton visualization for view invariant human action recognition. Pattern Recogn 68:346–362
43. Zhu K, Wang R, Zhao Q, Cheng J, Tao D (2019) A cuboid cnn model with an attention mechanism for skeleton-based action recognition. IEEE Trans Multimedia 22(11):2977–2989
44. Liu J, Shahroudy A, Xu D, and Wang G (2016), Spatio-temporal lstm with trust gates for 3d human action recognition," in European Conference on Computer Vision. Springer, 2016, pp. 816–833.
45. Cao C, Lan C, Zhang Y, Zeng W, Lu H, Zhang Y (2018) Skeleton-based action recognition with gated convolutional neural networks. IEEE Trans Circuits Syst Video Technol 29(11):3247–3257
46. Zhao R, Wang K, Su H, and Ji Q (2019), Bayesian graph convolution lstm for skeleton based action recognition, in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 6882–6892.
47. Song S, Lan C, Xing J, Zeng W, and Liu J (2017), An end-to-end spatiotemporal attention model for human action recognition from skeleton data, in Thirty-first AAAI Conference on Artificial Intelligence, 2017.
48. Zhang S, Yang Y, Xiao J, Liu X, Yang Y, Xie D, Zhuang Y (2018) Fusing geometric features for skeleton-based action recognition using multilayer lstm networks. IEEE Trans Multimedia 20(9):2330–2343
49. Fan Z, Zhao X, Lin T, Su H (2018) Attention-based multiview reobservation fusion network for skeletal action recognition. IEEE Trans Multimedia 21(2):363–374
50. Xie J, Miao Q, Liu R, Xin W, Tang L, Zhong S, Gao X (2021) Attention adjacency matrix based graph convolutional networks for skeleton-based action recognition. Neurocomputing 440:230–239
51. Zhang P, Lan C, Xing J, Zeng W, Xue J, Zheng N (2019) View adaptive neural networks for high performance skeleton-based human action recognition. IEEE Trans Pattern Anal Mach Intell 41(8):1963–1978
52. Yan S, Xiong Y, and Lin D (2018) Spatial temporal graph convolutional networks for skeleton-based action recognition. In Thirty-second AAAI conference on artificial intelligence.
53. Song YF, Zhang Z, Shan C, and Wang L (2020). Stronger, faster and more explainable: a graph convolutional baseline for skeleton-based action recognition. In proceedings of the 28th ACM international conference on multimedia (pp. 1625–1633).
54. Song YF, Zhang Z, Shan C, Wang L (2022) Constructing stronger and faster baselines for skeleton-based action recognition. IEEE Trans Pattern Anal Mach Intell 45(2):1474–1488
55. Shi L, Zhang Y, Cheng J, Lu H (2020) Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. IEEE Trans Image Process 29:9532–9545
56. Cheng K, Zhang Y, He X, Chen W, Cheng J, & Lu H (2020). Skeleton-based action recognition with shift graph convolutional network. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 183–192).
57. Ye, L., & Ye, S. (2021, April). Deep learning for skeleton-based action recognition. In Journal of Physics: Conference Series (Vol. 1883, No. 1, p. 012174). IOP Publishing.
58. Zhang J, Ye G, Tu Z, Qin Y, Zhang J, Liu X, and Luo S, A spatial attentive and temporal dilated (satd) gcn for skeleton-based action recognition, CAAI Transactions on Intelligence Technology, (2020).
59. Shi L, Zhang Y, Cheng J, and Lu H (2019), Two-stream adaptive graph convolutional networks for skeleton-based action recognition, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 12 026–12 035
60. Shi L, Zhang Y, Cheng J, and Lu H (2019), Skeleton-based action recognition with directed graph neural networks, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7912–7921.
61. Zhang X, Xu C, Tian X, Tao D (2019) Graph edge convolutional neural networks for skeleton-based action recognition. IEEE Transact Neural Netw Learn Syst 31(8):3047–3060
62. Veeriah V, Zhuang N, and Qi G-J (2015), Differential recurrent neural networks for action recognition, in ICCV, 2015.
63. Du Y, Wang W and Wang L (2015), Hierarchical recurrent neural network for skeleton based action recognition, in CVPR, 2015.
64. Zhu W, Lan, C, Xing J, Zeng W, Li Y, Shen L, and Xie X (2016), Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks, in AAAI, 2016.
65. Liu J, Shahroudy A, Xu D, Kot AC, and Wang G (2017), Skeleton-based action recognition using spatio-temporal lstm network with trust gates, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.
66. Jain A, Zamir AR, Savarese S, and Saxena A (2016), Structural-rnn: deep learning on spatio-temporal graphs, in CVPR, 2016.
67. Li Y, Lan C, Xing J, Zeng W, Yuan C, and Liu J (2016) Online human action detection using joint classification-regression recurrent neural networks, in ECCV, 2016.
68. Yadav SK, Tiwari K, Pandey HM, & Akbar SA (2022), Skeleton-based human activity recognition using ConvLSTM and guided feature learning. Soft Comput, 1–14.
69. Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 5(2):157–166
70. Hu JF, Zheng W-S, Lai, J and Zhang J (2015), "Jointly learning heterogeneous features for RGB-D activity recognition," in Proc. CVPR, 2015, pp. 5344–5352.

71. Liu J, Wang G, Duan LY, Abdiyeva K, Kot AC (2017) Skeleton-based human action recognition with global context-aware attention LSTM networks. IEEE Trans Image Process 27(4):1586–1599
72. Yadav SK, Luthra A, Tiwari K, Pandey HM, Akbar SA (2022) ARFDNet: an efficient activity recognition & fall detection system using latent feature pooling. Knowl-Based Syst 239:107948
73. Xu, H, Gao Y, Hui Z, Li J, and Gao X (2023), Language knowledge-assisted representation learning for skeleton-based action recognition. arXiv preprint arXiv:2305.12398.
74. Liu J, Wang X, Wang C, Gao Y, and Liu M (2023) Temporal decoupling graph convolutional network for skeleton-based gesture recognition. IEEE Transactions on Multimedia
75. Huang, X., Zhou H, Feng B, Wang X, Liu W, Wang J, Feng H, Han J, Ding E, and Wang J (2023) Graph contrastive learning for skeleton-based action recognition. arXiv preprint arXiv:2301.10900 (2023).
76. Duan, H, Wang J, Chen K, and Lin D (2022) Pyskl: towards good practices for skeleton action recognition. In Proceedings of the 30th ACM International Conference on Multimedia, pp. 7351–7354. 2022.
77. Duan, H, Zhao Y, Chen K, Lin D, and Dai B (2022) Revisiting skeleton-based action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2969–2978.