Multimedia Tools and Applications (2022) 81:40431–40449
https://doi.org/10.1007/s11042-022-12856-6

An information-rich sampling technique over spatio-temporal CNN for classification of human actions in videos

S. H. Shabbeer Basha[1] · **Viswanath Pulabaigari**[1] · **Snehasis Mukherjee**[2]

Received: 11 March 2021 / Revised: 27 January 2022 / Accepted: 9 March 2022 /
Published online: 9 May 2022

Abstract

We propose a novel video sampling scheme for human action recognition in videos, using Gaussian Weighing Function. Traditionally in deep learning-based human activity recognition approaches, either a few random frames or every k^{th} frame of the video is considered for training the 3D CNN, where k is a small positive integer, like 4, 5, or 6. This kind of sampling reduces the volume of the input data, which speeds-up the training network and also avoids overfitting to some extent, thus enhancing the performance of the 3D CNN model. In the proposed video sampling technique, consecutive k frames of a video are aggregated into a single frame by computing a Gaussian-weighted summation of the k frames. The resulting frame preserves the information in a better way than the conventional approaches and experimentally shown to perform better. In this paper, a 3-Dimensional deep CNN is proposed to extract the spatio-temporal features and follows Long Short-Term Memory (LSTM) to recognize human actions. The proposed 3D CNN architecture is capable of handling the videos where the camera is placed at a distance from the performer. Experiments are performed with KTH, WEIZMANN, and CASIA-B Human Activity and Gait datasets, whereby it is shown to outperform state-of-the-art deep learning based techniques. We achieve 95.78%, 95.27%, and 95.27% over the KTH, WEIZMANN, and CASIA-B human action and gait recognition datasets, respectively.

Keywords Human action recognition · 3-Dimensional convolutional neural networks (3D CNN) · Gaussian weighing function · Long short term memory (LSTM)

✉ S. H. Shabbeer Basha
shabbeer.sh@iiits.in

Viswanath Pulabaigari
viswanath.p@iiits.in

Snehasis Mukherjee
snehasis.mukherjee@snu.edu.in

[1] Computer Vision Group, Indian Institute of Information Technology Sri City, Chittoor, Andhra Pradesh, 517646, India

[2] Computer Science and Engineering Department, Shiv Nadar University, Greater Noida, India

1 Introduction and related works

Human action recognition in videos has been an active area of research, gaining the attention of Computer Vision and Machine Learning researchers during the last decade due to its potential applications in various domains, including intelligent video surveillance systems, *viz.,* Human-Computer Interaction (HCI), robotics, elderly and child monitoring systems and several other real-world applications. However, recognizing human actions in the real world remains a challenging task due to several challenges involved in real-life videos, including cluttered backgrounds, viewpoint variations, occlusions, varying lighting conditions and many more.

This paper proposes a technique for human activity recognition in videos, where the videos are captured by a camera placed at a distance from the performer.

The approaches for recognizing human actions from videos, found in the literature, can be broadly classified into two categories [61]. The first, make use of motion-related features (low, mid, and high level) for human action recognition [11, 29]. The other set of approaches experiment to learn a proper representation of the spatio-temporal features during action using deep neural networks [3, 48, 52, 56].

1.1 Human Action Recognition using hand-crafted features

Handcrafted features played a key role in various approaches for activity recognition [39]. Very recently, Ramya et al. [43] proposed a human action recognition method using distance transform and entropy features. Semantic features ease to identify similar activities that vary visually but have common semantics. Semantic features during an action contain human body parts (posture and poselet), background, motion and other features incorporating human perceptual knowledge about the activities. A study by Ziaeefard et al. [61] examined human action recognition approaches using semantic features. Malgireddy et al. [34] proposed a hierarchical Bayesian model which interconnects low-level features in videos with postures, motion patterns, and categories of activities.

Very recently, Nazir et al. [39] proposed a Bag of Expression (BOE) framework for activity recognition. The most common handcrafted feature, used for action recognition, is optical flow [6, 36, 38, 55]. Chaudhry et al. [6] introduced the concept of Histogram of Oriented Optical Flow (HOOF) for action recognition, where the optical flow direction is divided into octants. Mukherjee et al. [38] proposed Gradient-Weighted Optical Flow (GWOF) to limit the effect of camera shaking, where the optical flow of every frame is multiplied by the image gradient. Wang et al. [55] introduced another approach to reduce the camera shaking effect, called Warped Optical Flow (WOF), where gradient is computed on the optical flow matrix. In [36], the effect of background clutter is reduced by multiplying Weighted Optical Flow (WOF) features with the image gradients. Optical flow based approaches help in dissecting the motion, but gives too much unnecessary information such as, motion information at all the background pixels, which reduces the efficacy of the action recognition system in many cases.

Spatio Temporal Interest Points (STIP) introduced in [28], identifies spatio-temporal interest points based on the extension of Harris Corner Detection approach [17] towards the temporal domain. Several researchers have shown interest to recognize human actions with the help of some other variants of spatio-temporal features like Motion- Scale Invariant Feature Transform (MoSIFT) [7] and sparse features [11]. A study on STIP based human activity recognition methods is published by Dawn et al. [9]. However, such spatio-temporal features are unable to handle the videos taken in real-world which suffers from background clutter and camera shake. Buddubariki et al. [5] combined the benefits of GWOF and STIP features by calculating GWOF on the STIP points. In [1], combination of 3-dimensional SIFT and HOOF features are used along with support vector machine (SVM) for classifying human actions.

1.2 Human action recognition using deep neural networks

Recently, deep learning based models are gaining the interest of researchers for recognizing human actions [3, 13, 19, 23, 48, 54]. Jaouedi et al. [19] developed a hybrid deep learning framework for human action recognition. Initially, they have extracted and detected the motion information using Gaussian Mixture Model [51] and Kalman filter. Later, Gated Recurrent Neural Networks (GRNN) [8] utilize these features to perform human action recognition. Han et al. [10] developed human action recognition framework called RegFrame for classifying the simple human actions. Taylor et al. [53] proposed a multi-stage network, where in a Convolutional Restricted Boltzmann Machine (ConvRBM) retrieves motion-related information from each pair of successive frames at the initial layer. In [48], a two-stream convolutional network is proposed that comprises spatial-stream ConvNet and temporal-stream ConvNet. Ji et al. [21] introduced a 3-dimensional CNN architecture for action recognition, where a 3 dimensional convolutions are used to extract the spatio-temporal features. Tran et al. [54] enhanced 3D CNN model by applying Fisher vector encoding scheme on the learned features. Karpathy et al. [23] proposed a deep neural network for spatio-temporal resolutions: high and low resolutions, then merged them to train the CNN.

Kar et al. [22] proposed a technique for temporal frame pooling in a video for human activity recognition. The Action Matching Network (AMN) is proposed to solve more challenging open-set action recognition problem [58]. A survey by Herath et al. [18] discusses both engineered and deep learning based human action recognition techniques.

Deep neural networks have been employed to solve another similar problem called Person re-identification [40–42]. To mention a few, Ning et al. [42] proposed a feature selection model that combines both global and local fine-grained features to perform person re-identification. A 3D face alignment algorithm is developed in [40] using Encoder-Decoder Network (EDNet) which uses feature enhancement and feature fusion to enable the information transfer between encoder and decoder.

In the literature of human action recognition, researchers have used either the fully observed video or a portion of the video to train the deep neural networks. Training the models using a portion of the video will take less amount of training time compared to training the model using entire video. However, considering a portion of the video (considering 9 frames from the entire video as in [3] and 7 frames as in [21]) results in information loss. Srivastava et al. [50] used multi layer LSTM network to learn the representations of video sequences. Video object segmentation is performed through Episodic Graph Memory Networks (EGMN) [32] in which an episodic memory network is used to represent frames as nodes and the cross-frame correlation as edges. Lu et al. [33] proposed a network termed CO-attention Siamese Network (COSNet), to solve the zero-shot video object segmentation. Recently, Bilen et al. [4] introduced dynamic image, a very compact representation of video used for analyzing the video with CNNs. However, dynamic images eventually dilute the importance of spatial information during action. The proposed sampling technique for video frames preserves both spatial and temporal information together.

”How many video frames are required to perform human action recognition ?” is well explored research problem in the literature [45]. For example, in [45], authors have claimed that 1-7 frames are sufficient for basic human action recognition. Recently, Sarfraz et al. [44] proposed a temporally-weighted hierarchical clustering algorithm to group the frames that are semantically consistent for action segmentation task. Other methods such as [3] utilizes every k^{th} frame as input to the 3D CNN model to perfrom human activity recognition. However, utilizing small amount of frames for action action recognition ignore the motion information present in the video. To better utilize the motion information of a video, we propose a sampling technique using Gaussian Weighting Function (GWF) that aggregates multiple frames into a single frame. The proposed video sampling method better represents the motion information compared to aggregating the frames by averaging the consecutive frames, which can be observed in Fig. 4. Along with introducing a video sampling technique, we develop a two-stage human activity recognition framework motivated by the method proposed in [3].

We propose a 3D CNN to learn spatio-temporal features and then apply LSTM to classify human actions. The proposed method uses small sized filters throughout the 3D CNN architecture, which helps to learn minute information present in the videos, which can help in recognizing the action of performers appearing very small in the video, due to the distance of the camera.

Our contributions in this paper are three-folds.

1. A novel sampling technique is introduced to aggregate the entire video into a fewer number of frames.
2. A 3-dimensional (3D) CNN architecture is proposed for better classification of human actions in videos where the performer looks significantly small. The choice of smaller filter size enables the proposed model to work well in such scenarios where the performer looks small due to distance from the camera.
3. We conduct experiments over KTH, WEIZMANN, and CASIA-B human activity and gait recognition datasets. We also experiment with the proposed deep learning model using transfer learning technique, by transferring the knowledge learned from KTH dataset to fine-tune over WEIZMANN dataset and vice versa.

The proposed pre-processing method is presented in Section 2. Section 3 illustrates the proposed 3D CNN architecture. The experiments and results are described in Section 4. Finally, Section 5 concludes and provides scope for future research.

2 Pre-processing using an information sampling approach

The primary objective of this pre-processing step is to reduce the amount of training time and at the same time motion information should be given utmost importance. We propose a novel sampling technique to aggregate a large number of frames into a fewer set of frames using Gaussian Weighing Function (GWF), which minimizes the information loss. The proposed video pre-processing scheme is shown in Fig. 1.

Considering the frames that are more informative (as a pre-processing step) and ignoring the less important frames for human action recognition may be an effective way. However, this step requires a method to decide whether the given frame is informative or not (which may take considerable amount of time). A key-frame selection technique might help. However, key frame selection techniques are aimed to find out frames with more relevance. The relevance of the frames are measured based on video content as a whole. This content based relevance measure generally does not work for action recognition tasks in which

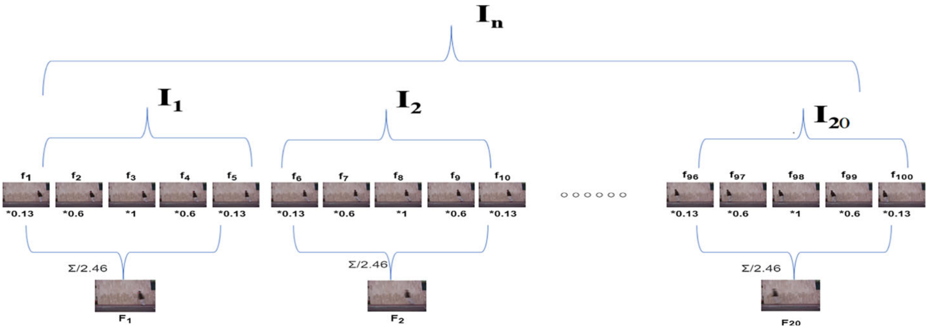

Fig. 1 The proposed pre-processing procedure using Gaussian Weighing Function. An entire video (collection of all frames) is represented as an exhaustive non-overlapping sequence I_n, which further has sub-sequences $\{I_1, I_2, \cdots\}$. A single pre-processed frame (for example $F1$) is obtained by performing weighted summation of consecutive five frames (for instance $f1$, $f2$, $f3$, $f4$, and $f5$ belongs to sub-sequence I_1) as shown in (2)

the performer covers a small portion of the entire frame [37]. In this paper, we introduce a mechanism to aggregate the consecutive k frames into a single frame.

Gaussian Weighing Function (GWF) is used to aggregate the entire video into a fewer number of frames. Let us consider $\{I_n\}_{n \in N}$, an exhaustive non-overlapping sequence (collection of all frames of a video), which is given by

$$\{I_n\} = \{I_1, I_2, \ldots, I_k, \ldots\}, \tag{1}$$

where $\{I_k\}$ is the k^{th} sub-sequence of $\{I_n\}$ and $k < n$. Mathematically, Gaussian Weighing Function G, for a sub-sequence $\{I_k\}$, is given as follows:

$$G(I_k, W) = \sum_{j=1}^{M} I_{k_j} * \frac{W_j}{\sum_{j=1}^{M} W_j} \tag{2}$$

The function G takes a sub-sequence $\{I_k\}$, and Gaussian weight vector W as input, and aggregates the information into a single frame. Here W_j represents the j^{th} element of Gaussian weight vector W, M denotes size of Gaussian weight vector. For example, if the size of the Gaussian weight vector M is 5 and the sub-sequence is $\{I_k\}$, which has five frames of the video. The vector W is given by W = [0.13, 0.6, 1, 0.6, 0.13]. A single frame is obtained by performing weighted summation of the five frames belonging to the sub-sequence $\{I_k\}$ as shown in (2). In other words, five frames are aggregated into a single frame using Gaussian weighing function. Similarly, the same process is repeated for subsequent five frames belonging to the next sub-sequence and so on. This sampling approach reduces the volume of data for training the deep learning model and also preserves the motion information in better way which helps to obtain better results in human activity recognition.

3 Spatio-temporal features extraction using deep learning models

In this section, initially we describe 2-D CNNs, and then we present a detailed discussion about the proposed 3-D CNN architecture, which learns the spatio-temporal features.

3.1 Convolutional neural networks

There are two major problems with Artificial Neural Networks (ANN) while dealing with real world data like images, videos, and any other high-dimensional data.

- ANNs do not maintain the local relationship among the neighboring pixels in an image.
- Since full connectivity is maintained throughout the network, the number of parameters are proportional to the input size.

To address these problems, Lecun et al. [30] introduced Convolutional Neural Networks (CNN), which are also called ConvNets.

Extensive amount of research is being carried out on images using CNN architectures to solve many problems in computer vision and machine learning. However, their application in video stream classification is comparatively a less explored area of research. In this paper, we performed 3D convolutions in the convolutional layers of proposed 3D CNN architecture to extract the spatial and temporal features. Net, we discuss the computational complexity analysis of 3D CNNs with respect to 2D CNNs.

3.2 Notations and computational complexity of 3D CNNs

In 2-D CNNs, features are computed by applying the convolutions spatially over images. Whereas in case of videos, we have to consider the temporal information along with spatial features. So, it is required to extract the motion information encoded in contiguous frames using 3D convolutions. The proposed 3-dimensional CNN architecture, shown in Fig. 2, uses 3D convolutions.

3.2.1 Notations

Let us consider an input feature map (or image) having I_h, I_w, and I_c as feature-map height, width, and number of channels, respectively. The 2D covolution operation is performed using a receptive field of dimension $F_h \times F_w$ by convolving the filter (receptive field) in both spatial and depth dimension. Whereas, 3D convolution operation is widely used while working with videos to capture the spatial and temporal features. A 3D convolution operation considers I_h, I_w, I_c, and I_d where I_d denotes the number of frames in the case of video input. The receptive field also has an additional dimension called filter depth F_d to

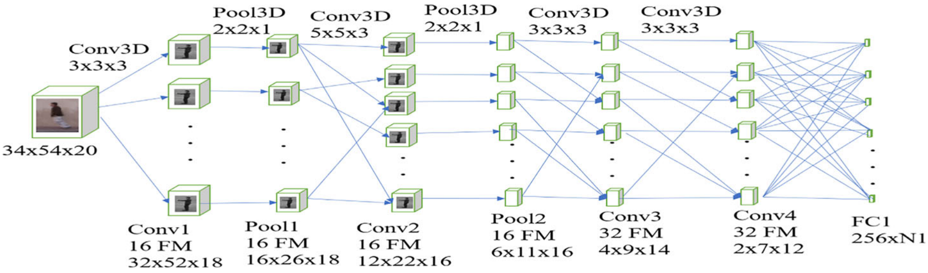

Fig. 2 Proposed 3-dimensional CNN for spatio-temporal feature construction (KTH dataset). The first two convolution layers $Conv1$ and $Conv2$, both have 16 feature maps of dimension $32 \times 52 \times 18$ and $12 \times 22 \times 16$, respectively. The $Pool1$ and $Pool2$ layers are followed by $Conv1$ and $Conv2$, to reduce the spatial dimension by half. $Conv3$ and $Conv4$ layers have 32 feature maps of dimension $4 \times 9 \times 14$ and $2 \times 7 \times 12$. Finally, a fully connected layer $FC1$ has 256 neurons

capture the motion information in temporal dimension. A typical 2D convolution operation accepts 3D volume of input data i.e., $I_h \times I_w \times I_c$ and generates a three dimensional output feature map of dimension $H_{out} \times W_{out} \times N$. Here, N indicates the number of filters used in this convolution layer. The height (H_{out}) and width (W_{out}) of the output feature map is computed as follows,

$$H_{out} = (I_h - F_h + 2P)/S + 1 \tag{3}$$

$$W_{out} = (I_w - F_w + 2P)/S + 1 \tag{4}$$

Next, we discuss the computational complexity of 2D, 3D convolutions in terms of Floating Point Operations (FLOPs).

3.2.2 Floating point operations

Normally, in the deep learning community Floating Point Operations (FLOPs) are used as a metric to measure efficiency of various models. The number of FLOPs correspond to 2D convolution layer L_i with filter dimension $F_h \times F_w$, number of filters to be N is given by,

$$FLOP_{2Dconv}(L_i) = F_h * F_w * I_c * H_{out} * W_{out} * N \tag{5}$$

While working with videos using 3D CNNs, we need to consider the number of frames (temporal) I_d as another dimension and the convolutional kernel also has an additional dimension to convolve in temporal (depth) F_d dimension. So the resultant FLOPs will become,

$$FLOP_{3Dconv}(L_i) = F_h * F_w * I_c * F_d * I_d * H_{out} * W_{out} * N \tag{6}$$

the depth of the input image I_c is 1 for gray-scale image, 3 for RGB image.

3.3 Proposed 3D CNN model: extracting Spatio-temporal features

Initially, the Gaussian Weighing function is used to aggregate the entire video into 20 frames (considered 100 frames from each video throughout our experiments). To reduce the memory overhead, person centered bounding boxes are retrieved as in [20, 21], which results in frames of spatial dimension 34 × 54 and 64 × 48 in case of KTH [46] and WEIZMANN [16] datasets, respectively.

In this paper, a 3D CNN model is proposed to extract spatio-temporal features, which is shown in Fig. 2. The proposed model considers the input of dimension 34 × 54 × 20, corresponding to 20 frames (encoded using GWF) of 34 × 54 pixels each. The proposed 3D CNN architecture has 5 learnable layers, viz., $Conv1$, $Conv2$, $Conv3$, $Conv4$, and $FC1$. $Pool1$ and $Pool2$ max pooling layers are applied after $Conv1$ and $Conv2$ to reduce the spatial dimension of the feature maps by half.

The abstract view of 3D convolutional operation is presented in Fig. 3. This illustration is for gray-scale video which has frame height, frame width, and number of frames and note that for RGB video there is another dimension, i.e,. frame depth. The $Conv1$ layer generates 16 feature maps of size 32 × 52 × 18 by convolving 16 3-D kernels of size 3 × 3 × 3. $Pool1$ layer down samples the feature maps by half, after applying sub-sampling operation with a receptive field of 2 × 2 × 1, which results in a 16 × 26 × 18 dimensional feature vector. The $Conv2$ layer results in a 12 × 22 × 16 dimensional feature map by convolving 16 filters of size 5 × 5 × 3 × 18. The $Pool2$ layer produces a 6 × 11 × 16 dimensional feature vector, by applying sub-sampling operation with a receptive field of 2 × 2 × 1. The 3^{rd} convolution layer ($Conv3$) produces 32 feature maps of dimension 4 × 9 × 14, which is obtained by

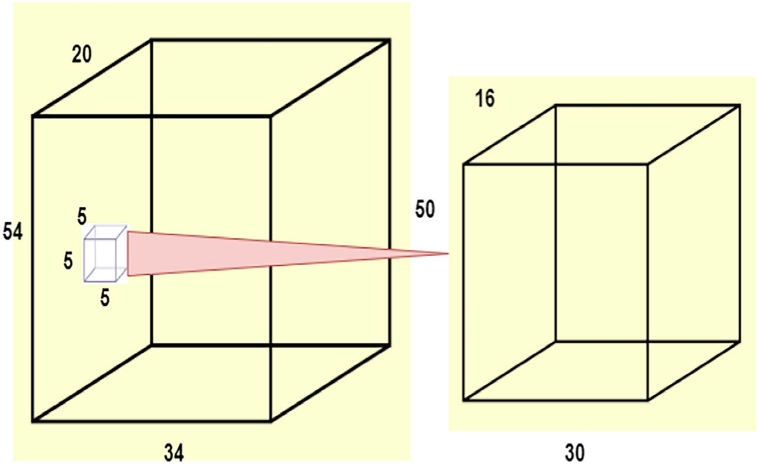

Fig. 3 Illustration of 3D convolution operation. A four dimensional filter (including image/frame depth) is convolved over a four dimensional input image/feature map. Here, we consider a gray-scale video as input (which has frame weight and frame width, and number of frames) to illustrate 3D convolution operation. A 3-D convolutional filter of dimension 5 × 5 × 5 is convolved over 54 × 34 × 20 that generates 50 × 30 × 16 dimensional feature map

convolving 32 kernels of dimension 3 × 3 × 3 × 16. The $Conv4$ layer generates 32 feature maps of dimension 2 × 7 × 12, which is obtained by convolving 32 filters of dimension 3 × 3 × 3 × 32. The feature maps produced by $Conv4$ layer are flattened into a single feature vector of dimension 5376 × 1, which is given as input to the 1^{st} fully connected layer ($FC1$). Finally, the $FC1$ layer produces 256 dimensional feature vector. The 3D CNN architecture proposed for spatio-temporal feature extraction, consists a total of 1,437,712 trainable parameters. The number of trainable parameters involved in the proposed 3D CNN are less comparable to the 3D CNNs proposed in [3, 21] for action recognition task (Fig. 4).

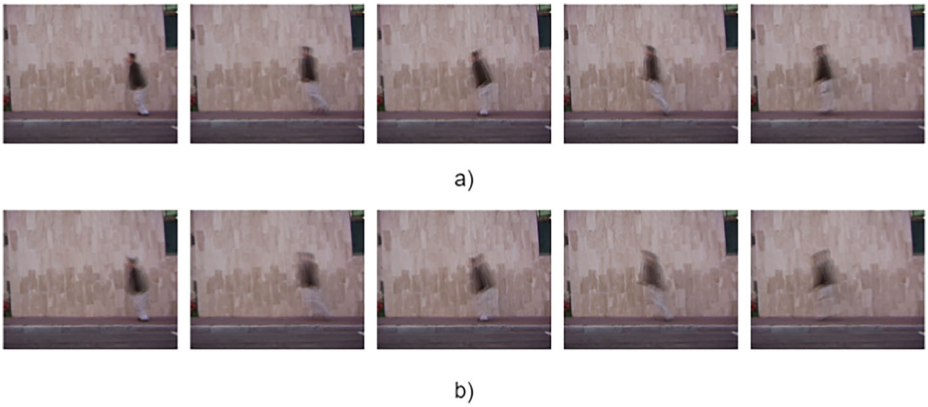

Fig. 4 The aggregated video frames obtained using a) the proposed Gaussian Weighting Function (GWF) and b) Average video sampling. GWF better represents the motion information compared with taking the average of 5 consecutive video frames

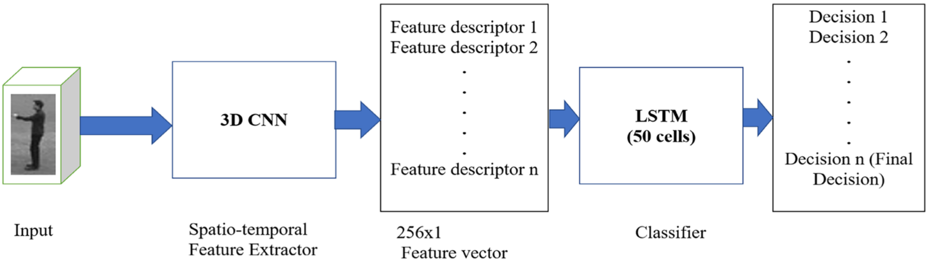

Fig. 5 The proposed two-steps deep neural network approach. Encoded frames are given as input to the 3D CNN model to extract spatio-temporal features as discussed in Secton 3.3 . The proposed 3D CNN model generates 256 × 1 dimensional feature vector, which is given as input the LSTM model to classify human actions. The LSTM has one hidden layer with 50 cells, that accumulates the individual decisions corresponding to small temporal neighborhood (4 frames) of the video

For WEIZMANN dataset, we used same architecture with necessary modifications. However, throughout the architecture same hyper-parameters (number of filters, filter size) are maintained as in the case of KTH dataset. The 3D CNN model proposed for WEIZMANN dataset takes input of dimension 64 × 48 × 20. This model has four Conv layers ($Conv1$, $Conv2$, $Conv3$, and $Conv4$) and two max-Pooling layers ($Pool1$, $Pool2$) layers, and towards the end one fully connected layer ($FC1$). The $Conv1$ layer results in 16 feature maps of dimension 62 × 46 × 18, which is obtained by convolving 16 kernels of size 3 × 3 × 3. The $Pool1$ layer generates reduce the spatial dimension by half, after applying sub-sampling with a receptive field of 2 × 2 × 1, which generates 31 × 23 × 18 dimensional feature vector. The $Conv2$ layer generates 16 feature maps of dimension 27 × 19 × 16, this is obtained by applying 16 filters of size 5 × 5 × 3 × 16. The $Pool2$ layer generates a 13 × 9 × 16 dimensional feature vector by sub-sampling with a receptive field of 2 × 2 × 1. The $Pool2$ layer does not consider the right and bottom border feature values to avoid the dimension mismatch between input and filter size. The $Conv3$ layer results in a 11 × 7 × 14 dimensional feature vector, which is obtained by convolving 32 filters of size 3 × 3 × 3 × 16. The $Conv4$ layer results in 32 feature maps of dimension 9 × 5 × 12, which is obtained by convolving 32 filters of dimension 3 × 3 × 3 × 32. The output of $Conv4$ layer is rolled into a single column vector of dimension 17280 × 1. At the end of the architecture, $FC1$ layer has 256 neurons, which results in a 256 dimensional feature vector. The proposed 3D CNN architecture for WEIZMANN human action dataset consists of 4,485,136 number of learnable parameters. The learned spatio-temporal features are given as input to LSTM model to learn the label of the entire sequence. We resize the spatial dimension of the frames of CASIA-B dataset [59] from 352 × 240 to 64 × 48 so that the same 3D CNN which is used for WEIZMANN can be used for CASIA-B Human Gait Recognition (HGR).

3.4 Classification using long short-term memory (LSTM)

Once the 3D-CNN architecture is trained, it learns the spatio-temporal features automatically. The learned features are provided as input to an LSTM architecture (a Recurrent Neural Networks (RNN)) for classification. RNNs are widely used deep learning models to accumulate the individual decisions related to small temporal neighborhood of the video. RNNs make use of recurrent connections to analyze the temporal data. However, RNNs able to learn the information which are about short duration. To learn the class label of the entire sequence, Long Short-Term Memory (LSTM) [14] is employed, which accumulates the individual decisions corresponds to each small temporal neighborhood. To obtain a sequence, we have considered every 4 frames as a temporal neighborhood. To classify human actions, we employ an RNN model having a hidden layer of LSTM cells. Figure 5 shows the overview of the proposed two-steps learning process. The input to this RNN architecture is 256 $FC1$ features per time step. These 256 dimensional input features are fully connected with LSTM cells. The number of LSTM cells considered are 50 as in [3]. The training details of the proposed 3D CNN architecture is presented in Section 4.4.1.

4 Experiments, results and discussions

As the proposed method aims to classify human actions in a video, where the videos are captured at a distance from the performer, we trained and evaluated the proposed 3D CNN model on KTH, WEIZMANN, CASIA-B datasets. Also we experimented with transfer learning techniques, where proposed method is trained with KTH and then tested on WEIZMANN dataset, and vice versa. Throughout our experiments, we consider validation accuracy as the evaluation metric.

4.1 KTH dataset

KTH dataset [46] is one among the popular datasets in human action recognition. This dataset consists of six actions, viz., walking, jogging, running, boxing, hand-waving, and hand-clapping which were carried out by 25 persons and the videos were recorded in four different scenarios (outdoor, variations in scale, variations in cloths, and indoor). A few samples from KTH dataset [46] are presented in Fig. 6. The spatial dimension of each frame is 160 × 120 pixels and the rate of frames per second (fps) is 25. This dataset has 600 videos. All the videos were captured from a distance from the performer. As a result, the

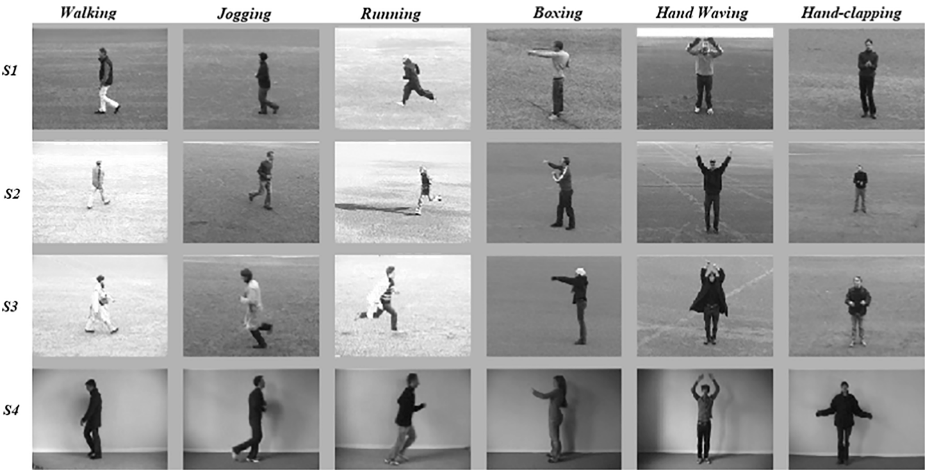

Fig. 6 A few sample of actions from KTH dataset [46]. Six different actions are shown column-wise. The videos were recorded in four scenarios, outdoors $s1$, outdoor with scale-variation $s2$, outdoors with different cloths $s3$, and indoors $s4$, which is shown row-wise in the figure

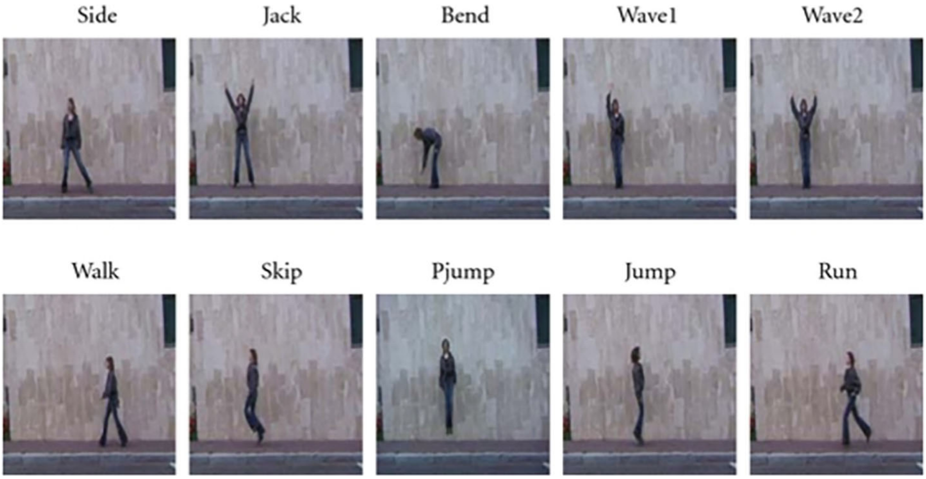

Fig. 7 An illustration of actions from WEIZMANN dataset [16]. Action labels are specified above the corresponding frames.

area covered by the person is less than 10% of the whole frame. We split the entire dataset randomly into training (8+8 people) and validation (9 people) as in [3, 46].

4.2 WEIZMANN dataset

The WEIZMANN human activity recognition dataset [16] consists of 90 videos that correspond to ten actions, which were performed by nine different people. The ten actions are gallop sideways (Side), jumping-back (jack), bending, one-hand-waving (Wave1), two-hands-waving (Wave2), walking, skipping, jumping in place (Pjump), jumping-forward (jump), and running. The spatial dimension of each frame is 180 × 144, and is at 25 frames per second (fps). A few sample of frames and corresponding action labels of WEIZMANN dataset are depicted in Fig. 7. The area covered by the person is less than 12% of the entire frame, due to the reason that videos were captured from a distance from the performer. We consider 50% of videos for training and remaining 50% videos are used for testing the performance of the proposed model as in [24].

4.3 CASIA-B Human Gait Recognition (HGR) dataset

CASIA-B [59] is a widely used dataset for HGR. The videos are recorded in indoor environment and recorded with many variations like different view-angles, wearing different cloths, and carrying things. The FPS rate is 25. The frame resolution is 352 × 240. We utilize the video frames in the ratio of 70:30 such that 70% of video frames are used for training and 30% of the video frames are used to validate the performance of the proposed model.

4.4 Experimental Results

To validate the performance of the proposed 3D CNN model, throughout our experiments, we have considered videos up to 4 seconds length (100 frames) and aggregated them into 20 frames using Gaussian Weighing Function as discussed in Section 2. To reduce the memory consumption, we have used the person-centered bounding boxes as in [20, 21]. Apart from these simple preprocessing steps we have not performed any other complex preprocessing like optical flow, gradients, etc.

4.4.1 Training Setup

To train the proposed 3-D CNN architectures, ReLU [27] is used as the activation function after every $Conv$ and FC layers (except output FC layer). We have experimented with the 3D CNNs in which all layers have filters of dimension 3x3, 5x5, and 7x7. From these experiments, we choose the better performing filter dimension which is 3x3 in our case. Later, we try to find the better performing layer-specific filter dimension. More concretely, our initial set of experiments are aimed at finding the optimal filter dimension at architecture level, later, it is constrained to layer-level. Through these experiments, we have considered the best performing network hyperparameters. Initially learning rate is considered as 1×10^{-4}. The value of the learning rate reduced with a factor of $\sqrt[2]{0.1}$ after every 100 epochs. The developed models are trained for 300 epochs using Adam optimizer [25] with $\beta_1 = 0.9$, $\beta_2 = 0.99$, and decay = 1×10^{-6}. The 80% of entire data is used to train the 3D CNN model and remaining data is utilized to test the performance of the model. After employing Gaussian Weighting function, we obtained 20 frames corresponding to an entire video. To reduce the amount of over-fitting, we generated 1800 and 270 videos (of length 20 frames) for KTH and WEIZMANN datasets, respectively, using data-augmentation techniques like vertical flip, horizontal flip, rotation by 30°. We also employed dropout [49] (after each $Conv$, FC layers except final FC layer with a rate of 0.4, after $ReLU$ is applied) along with data augmentation to reduce the amount of over-fitting.

4.4.2 Results and Discussions

The obtained results are compared with the state-of-the-art methods as shown in Tables 1, 2, and 3 on KTH, WEIZMANN, and CASIA-B datasets, respectively. Baccouche et al. [3] reported 94.39% accuracy over KTH dataset using a 3D CNN architecture having five trainable layers. However, they have not evaluated their model on WEIZMANN dataset, we obtained 94.58% accuracy through our experiment (input dimension is 64x48x9) using the same architecture (same w.r.t number of features, filter size, number of neurons in FC layers) as in [3]. After employing the proposed scheme of generating aggregated video to the 3D CNN model proposed in [3], we observed that the model outperforming the original model. However, the Dynamic Image Network proposed by [4] results in high amount of over-fitting due to which it produces only 85.2%, 86.8% accuracies for KTH, WEIZMANN datasets. We achieve 95.04%, 95.01%, 94.22%, 98.017%, 96.34%, and 96.05% accuracies on walking, jogging, running, boxing, hand-waving, and hand-clapping human actions, respectively.

The proposed 3D CNN model produces 95.78%, 95.27% accuracies on KTH and WEIZMANN datasets, respectively, when the size of Gaussian weight vector is 5. From Tables 1 and 2, we can observe that the proposed 3D CNN model outperforming other deep learning based models on both the datasets. However, the HGR results reported over CASIA-B are obtained using the finetuning the pre-trained CNNs mentioned in the Table 3. It is evident from Table 3 that, employing the proposed video sampling method as a pre-processing step increases the HGR performance. For example, fine-tuning the pre-trained DenseNet-121 using CASIA-B results in 94.7% validation accuracy, whereas, this performance is increased by 0.87 after employing Gaussian Weighing Function (GWF) as a pre-processing

Table 1 A performance comparison of state-of-the-art methods on KTH dataset with proposed 3D CNN model using 5-folds cross validation test

S.No.	Method	Features	Classification Accuracy
1	Baccouche et al. [3]	3D CNN features	94.58
2	Baccouche et al. [3] + GWF (Ours)	3D CNN features	94.9
3	orelick et al. [16]	Space-time saliency, Action dynamics	97.83
4	Fathi et al. [12]	Mid-level motion features	100
5	Bilen et al. [4]	2D CNN features	85.2
5	Jaouedi et al. [19]	hand-crafted + 2D CNN features	96.3
5	Ramya et al. [43]	distance transform and entropy features	91.4
5	Liu et al. [31]	3D CNN features	91.93
6	Proposed 3D CNN	3D CNN features	94.14
7	**Proposed 3D CNN + GWF (Ours)**	**3D CNN features**	**95.78 ± 0.58**
8	**Proposed method applying Transfer Learning***	**3D CNN features**	**96.53 ± 0.07**

The best performing deep learning based human activity recognition methods are highlighted in bold

*Fine-tuning the last two FC layers of pre-trained model, which is trained on WEIZMANN dataset

Table 2 Comparing the state-of-the-art human action recognition approaches on WEIZMANN dataset with the proposed 3D CNN model using 5-folds cross validation test

S.No.	Method	Features	Classification Accuracy
1	Nazir et al. [39]	Bag of Expressions (BoE)	99.51
2	Abdul et al. [1]	Bag of Visual Words	97.20
3	Ji et al. [21]	3D CNN features	90.20
4	Wang et al. [55]	Dense Trajectories and motion boundary descriptor	95.00
5	Gilbert et al. [15]	Mined Hierarchical compound features	94.50
6	Yang et al. [57]	Multi-scale oriented neighborhood features	96.50
7	Kovashka et al. [26]	Hierarchical Space time neighborhood features	94.53
5	Liu et al. [31]	3D CNN features	95.75
5	Ramya et al. [43]	distance transform and entropy features	92.5
8	Khan et al. [24]	fusion of hand-crafted and deep learning features	99.4
8	Bilen et al. [4]	2D CNN features	86.8
9	Baccouche et al. [3]	3D CNN features	94.39
10	Baccouche et al. [3] + GWF (Ours)#	3D CNN features	94.78 ± 0.11
11	Proposed 3D CNN	3D CNN features	95.03 ± 0.08
12	**Proposed 3D CNN + GWF (Ours)**	**3D CNN features**	**95.27** ± **0.45**
13	**Proposed Method applying Transfer learning**$	**3D CNN features**	**95.86** ± **0.3**

The best performing deep learning based human activity recognition methods are highlighted in bold

#Encoded frames are given as input to the 3D CNN model proposed in [3]. (the size of Gaussian vector is 5)

$Fine-tuning the last two FC layers of pre-trained model, which is trained on KTH dataset

Table 3 The performance comparison of state-of-the-art methods on CASIA-B dataset with proposed 3D CNN model using 3-folds cross-validation

S.No.	Method	Features	Classification Accuracy
1	Baccouche et al. [3][#]	3D CNN features	89.52
1	Baccouche et al. [3] + GWF (Ours)*	3D CNN features	91.43
2	Mehmood et al. [35] (DenseNet-201)	3D CNN features	94.2
3	Zhang et al. [60]	3D CNN features	91.2
4	Alotaibi et al. [2]	2D CNN features	90.43
5	Proposed 3D CNN	3D CNN features	91.72 ± 0.03
6	Proposed 3D CNN + GWF (Ours)	3D CNN features	92.8 ± 0.2
7	VGG-16	3D CNN features	93.8
8	**VGG-16 + GWF (Ours)**	**3D CNN features**	$\mathbf{94.1 \pm 0.08}$
9	DenseNet-121	3D CNN features	94.7
10	**DenseNet-121 + GWF (Ours)**	**3D CNN features**	$\mathbf{95.27 \pm 0.64}$

[#] The human gait recognition are reproduced using the 3D CNN proposed in [3]

[*] Encoded frames are given as input to the 3D CNN model proposed in [3]. (the size of Gaussian vector is 5)

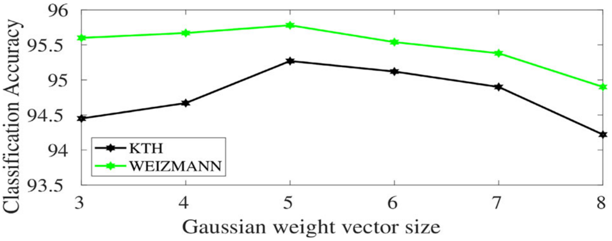

Fig. 8 A performance comparison of proposed 3D CNN model by varying the size of Gaussian weight vector. The size of the Gaussian weight vector is considered as 3, 4, 5, 6, 7, and 8 in our experiments

step. Please note that we have not employed any additional pre-processing such as removing carried objects other than aggregating the multiple frames into a single frame.

When compared with human action recognition methods involving hand-crafted features, our method produces competitive results with state-of-the-art on both KTH and WEIZMANN datasets. We also experimented the performance of our model by varying the size of Gaussian weight vector W in the range from 3 to 8. The performance variation of proposed model is shown in Fig. 8 by varying the size of Gaussian weight vector W'. We observe that the proposed 3D CNN architecture is showing the best accuracy, when the size of Gaussian weight vector $W = 5$. Based on the results depicted in Tables 1 and 2, we can conclude that, our 3D CNN architecture outperforms the state-of-the-art deep learning architectures.

However, due to the small size of the available dataset of such kind, the proposed deep learning based method could not outperform the hand-crafted feature based methods (although showing a comparable result).

Basha et al. [47] shown the necessity of the fully connected layers based on the depth of the CNN. Motivated by their work, experiments are conducted by varying the number of trainable layers in the proposed 3D CNN architecture. The amount of over-fitting increases in the context of both the datasets after inclusion of more FC layers.

The performance of the proposed 3D CNN architecture with varying number of trainable layers is depicted in Fig. 9.

4.4.3 Fine-tuning the pre-trained 3D CNNs

A common practice in deep learning community (especially to deal with small datasets) is that, using the pre-trained models to reduce the training time and obtaining competitive results by training the models for a fewer number of epochs. Generally, these pre-trained

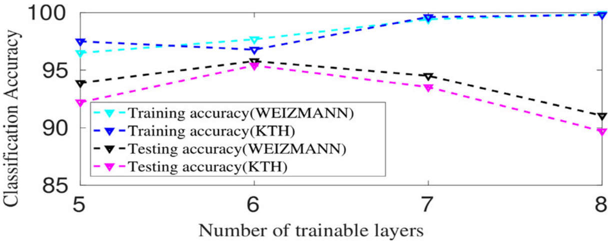

Fig. 9 Comparing the Training and Testing accuracies of both the datasets by varying the number of trainable layers (5, 6, 7, and 8) in the proposed 3D CNN architecture

models work as feature extractors. With this motivation, we utilized the pre-trained model of KTH dataset to fine-tune over WEIZMANN dataset and vice-versa. Note that the dimensions of the input are resized to fit the frames as input to the 3D CNN. The last two layers ($Conv4$, $FC1$) of the proposed 3D CNN model are fine-tuned in both the cases. Results of the above experiments are reported in the last rows of the Tables 1 and 2, respectively. We can observe a little increase in the classification accuracy for both the datasets, after applying the above scheme.

5 Conclusion

We introduced an information-rich sampling technique using Gaussian weighing function as a pre-processing step before giving it as input to any deep learning model, for better classification of human actions from videos. The proposed scheme aggregates consecutive k frames into a single frame by applying a Gaussian weighted summation of the k frames. We further proposed a 3D CNN model that learns and extracts spatio-temporal features by performing 3D convolutions. The classification of the human actions are performed using LSTM. Experimental results on both KTH and WEIZMANN datasets show that proposed model produces comparable results, among the state-of-the art. Whereas, the proposed 3D CNN model outperforms the state-of-the-art deep CNN models. In future, we aim at employing the proposed video sampling method for applications such as human driver behavior recognition for autonomous driving, social distancing detection for helping prevent COVID-19, and many more.

Acknowledgments We acknowledge the support of NVIDIA with the donation of the GeForce Titan XP GPU used for this research.

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Abdulmunem A, Lai YK, Sun X (2016) Saliency guided local and global descriptors for effective action recognition. Computational Visual Media 2(1):97–106
2. Alotaibi M, Mahmood A (2017) Improved gait recognition based on specialized deep convolutional neural network. Comput Vis Image Underst 164:103–110
3. Baccouche M, Mamalet F, Wolf C, Garcia C, Baskurt A (2011) Sequential deep learning for human action recognition. In: International workshop on human behavior understanding, pp 29–39. Springer
4. Bilen H, Fernando B, Gavves E, Vedaldi A, Gould S (2016) Dynamic image networks for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3034–3042
5. Buddubariki V, Tulluri SG, Mukherjee S (2016) Event recognition in egocentric videos using a novel trajectory based feature. In: Proceedings of the tenth indian conference on computer vision graphics and image processing, pp 76 ACM
6. Chaudhry R, Ravichandran A, Hager G, Vidal R (2009) Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In: 2009 IEEE Conference on computer vision and pattern recognition, pp 1932–1939, IEEE
7. Chen M, Hauptmann A (2009) Mosift: Recognizing human actions in surveillance videos
8. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv:1406.1078

9. Das Dawn D, Shaikh SH (2016) A comprehensive survey of human action recognition with spatio-temporal interest point (stip) detector. Vis Comput 32(3):289–306
10. Di H, Li J, Zeng Z, Yuan X, Li W (2018) Regframe: fast recognition of simple human actions on a stand-alone mobile device. Neural Comput Applic 30(9):2787–2793
11. Dollár P, Rabaud V, Cottrell G, Belongie S (2005) Behavior recognition via sparse spatio-temporal features. In: Visual surveillance and performance evaluation of tracking and surveillance, 2005. 2nd Joint IEEE International Workshop on, pp 65–72. IEEE
12. Fathi A, Mori G, Action recognition by learning mid-level motion features (2008) Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8. IEEE
13. Gao Z, Zhang H, Liu AA, Guangping X, Xue Y (2016) Human action recognition on depth dataset. Neural Comput & Applic 27(7):2047–2054
14. Gers FA, Schraudolph NN, Schmidhuber J (2002) Learning precise timing with lstm recurrent networks. Journal of machine learning research 3(Aug):115–143
15. Gilbert A, Illingworth J, Bowden R (2011) Action recognition using mined hierarchical compound features. IEEE Trans Pattern Anal Mach Intell 33(5):883–897
16. Gorelick L, Blank M, Shechtman E, Irani M, Basri R (2007) Actions as space-time shapes. IEEE transactions on pattern analysis and machine intelligence 29(12):2247–2253
17. Harris C, Stephens M (1988) A combined corner and edge detector. In: Alvey vision conference, vol 15, pp 10–5244. Citeseer
18. Herath S, Harandi M, Porikli F (2017) Going deeper into action recognition: a survey. Image and vision computing 60:4–21
19. Jaouedi N, Boujnah N, Bouhlel MS (2020) A new hybrid deep learning model for human action recognition. J King Saud Univ- Comput Inf Sci 32(4):447–453
20. Jhuang H, Serre T, Wolf L, Poggio T (2007) A biologically inspired system for action recognition. In: Computer Vision, 2007. ICCV 2007 IEEE 11th International Conference on, pp 1–8, Ieee
21. Ji S, Wei X, Yang M, Yu K (2013) 3d convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence 35(1):221–231
22. Kar A, Rai N, Sikka K, Sharma G (2017) Adascan: Adaptive scan pooling in deep convolutional neural networks for human action recognition in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3376–3385
23. Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Li FF (2014) Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp 1725–1732
24. Khan MA, Sharif M, Akram T, Raza M, Saba T, Rehman A (2020) Hand-crafted and deep convolutional neural network features fusion and selection strategy: an application to intelligent human action recognition. Appl Soft Comput 87:105986
25. Kingma DP, Ba J (2014) Adam:, A method for stochastic optimization. arXiv:1412.6980
26. Kovashka A, Grauman K (2010) Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In: Computer Vision and Pattern Recognition (CVPR) IEEE Conference on, pp 2046–2053, IEEE
27. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
28. Laptev I (2005) On space-time interest points. Int J Comput Vis 64(2-3):107–123
29. Laptev I, Pérez P (2007) Retrieving actions in movies. In: Computer Vision, 2007. ICCV IEEE 11th International Conference on, pp 1–8, IEEE, p 2007
30. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
31. Liu X, Qi DY, Xiao HB (2020) Construction and evaluation of the human behavior recognition model in kinematics under deep learning. Journal of Ambient Intelligence and Humanized Computing, pp 1–9
32. Lu X, Wang W, Danelljan M, Zhou T, Shen J, Van Gool L (2020) Video object segmentation with episodic graph memory networks. arXiv:2007.07020
33. Lu X, Wang W, Shen J, Crandall D, Luo J (2020) Zero-shot video object segmentation with co-attention siamese networks. IEEE Transactions on Pattern Analysis and Machine Intelligence
34. Malgireddy MR, Nwogu I, Govindaraju V (2013) Language-motivated approaches to action recognition. J Mach Learn Res 14(1):2189–2212
35. Mehmood A, Khan MA, Sharif M, Khan SA, Shaheen M, Saba T, Riaz N, Ashraf I (2020) Prosperous human gait recognition: an end-to-end system based on pre-trained cnn features selection. Multimedia Tools and Applications, pp 1–21
36. Mukherjee S (2015) Human action recognition using dominant pose duplet. In: International conference on computer vision systems, pp 488–497. Springer

37. Mukherjee S, Biswas SK, Mukherjee DP (2011) Recognizing human action at a distance in video by key poses. IEEE Trans Circuits Syst Video Technol 21(9):1228–1241
38. Mukherjee S, Biswas SK, Mukherjee DP (2014) Recognizing interactions between human performers by 'dominating pose doublet'. Mach Vis Appl 25(4):1033–1052
39. Nazir S, Yousaf MH, Nebel JC, Velastin SA (2018) A bag of expression framework for improved human action recognition Pattern Recognition Letters
40. Ning X, Duan P, Li W, Zhang S (2020) Real-time 3d face alignment using an encoder-decoder network with an efficient deconvolution layer. IEEE Signal Process Lett 27:1944–1948
41. Ning X, Ke G, Li W, Zhang L (2020) Jwsaa: Joint weak saliency and attention aware for person re-identification Neurocomputing
42. Ning X, Ke G, Li W, Zhang L, Bai X, Tian S (2020) Feature refinement and filter network for person re-identification. IEEE Transactions on Circuits and Systems for Video Technology
43. Ramya P, Rajeswari R (2021) Human action recognition using distance transform and entropy based features. Multimed Tools Appl 80(6):8147–8173
44. Sarfraz S, Murray N, Vivek S, Diba A, Van Gool L, Stiefelhagen R (2021) Temporally-weighted hierarchical clustering for unsupervised action segmentation. arXiv:2103.11264
45. Schindler K, Van Gool L (2008) Action snippets: How many frames does human action recognition require? In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp 1–8. IEEE
46. Schuldt C, Laptev I, Caputo B (2004) Recognizing human actions: a local svm approach. In: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol 3, pp 32–36. IEEE
47. Shabbeer Basha SH, Dubey SR, Pulabaigari V, Mukherjee S (2020) Impact of fully connected layers on performance of convolutional neural networks for image classification. Neurocomputing 378:112–119
48. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems, pp 568–576
49. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
50. Srivastava N, Mansimov E, Salakhudinov R (2015) Unsupervised learning of video representations using lstms. In: International conference on machine learning, pp 843–852
51. Stauffer C, Grimson WEL (1999) Adaptive background mixture models for real-time tracking. In: 1999 Proceedings IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149), vol 2, pp 246–252. IEEE
52. Sun L, Jia K, Chen K, Yeung DY, Shi BE, Savarese S (2017) Lattice long short-term memory for human action recognition. In: The IEEE International Conference on Computer Vision (ICCV)
53. Taylor GW, Fergus R, LeCun Y, Bregler C (2010) Convolutional learning of spatio-temporal features. In: European conference on computer vision, pp 140–153. Springer
54. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3d convolutional networks. In: Computer Vision (ICCV), 2015 IEEE International Conference on, pages 4489–4497. IEEE
55. Wang H, Kläser A, Schmid C, Liu CL (2013) Dense trajectories and motion boundary descriptors for action recognition. Int J Comput Vis 103(1):60–79
56. Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Van Gool L (2016) Temporal segment networks: Towards good practices for deep action recognition. In: European conference on computer vision, pp 20–36. Springer
57. Yang J, Ma Z, Xie M (2015) Action recognition based on multi-scale oriented neighborhood features. Int J Signal Process, Image Process Pattern Recognit 8(1):241–254
58. Yu J, Kim DY, Yoon Y, Jeon M (2019) Action matching network: open-set action recognition using spatio-temporal representation matching. The Visual Computer, pp 1–15
59. Yu S, Tan D, Tan T (2006) A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition. In: 18th International Conference on Pattern Recognition (ICPR'06), vol 4, pp 441–444. IEEE
60. Zhang Y, Huang Y, Wang L, Yu S (2019) A comprehensive study on gait biometrics using a joint cnn-based method. Pattern Recogn 93:228–236
61. Ziaeefard M, Bergevin R (2015) Semantic human activity recognition: a literature review. Pattern Recogn 48(8):2329–2345

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.