IET Image Processing

Research Article

Human activity recognition using 2D skeleton data and supervised machine learning

ISSN 1751-9659
Received on 8th January 2019
Revised 4th April 2019
Accepted on 27th August 2019
E-First on 11th October 2019
doi: 10.1049/iet-ipr.2019.0030
www.ietdl.org

Sumaira Ghazal[1] ✉, *Umar S. Khan*[1,2], *Muhammad Mubasher Saleem*[1,2], *Nasir Rashid*[1,2], *Javaid Iqbal*[1,2]

[1] *Department of Mechatronics Engineering, National University of Sciences and Technology, H-12, Islamabad, Pakistan*
[2] *National Centre of Robotics and Automation (NCRA), Rawalpindi, Pakistan*
✉ *E-mail: sumaira.ghazal86@mts.ceme.edu.pk*

Abstract: Vision-based human activity recognition (HAR) finds its application in many fields such as video surveillance, robot navigation, telecare and ambient intelligence. Most of the latest researches in the field of automated HAR based on skeleton data use depth devices such as Kinect to obtain three-dimensional (3D) skeleton information directly from the camera. Although these researches achieve high accuracy but are strictly device dependent and cannot be used for videos other than from specific cameras. Current work focuses on the use of only 2D skeletal data, extracted from videos obtained through any standard camera, for activity recognition. Appearance and motion features were extracted using 2D positions of human skeletal joints through OpenPose library. The approach was trained and tested on publically available datasets. Supervised machine learning was implemented for recognising four activity classes including sit, stand, walk and fall. Performance of five techniques including *K*-nearest neighbours (KNNs), support vector machine, Naive Bayes, linear discriminant and feed-forward back-propagation neural network was compared to find the best classifier for the proposed method. All techniques performed well with best results obtained through the KNN classifier.

1 Introduction

Vision-based human activity recognition (HAR) systems have gained much popularity in the current decade due to the wide range of applications related to this specific area of research. The goal of such a system is to examine the videos and extract useful spatial and temporal information which is helpful in human behaviour interpretation and scene understanding. Traditionally, task of activity recognition in videos is performed by human operators which require constant attention and vigilance because events of interest are not a common occurrence and rare as compared with the full length of the video. Hence, making manual systems tiresome and prone to error. Automated counterparts are more effective due to high speed, cost-effectiveness and accuracy. However, extracting useful information by processing video data is a challenging task due to the presence of noise in the background, high dimensionality of input video data, interrelated actions and interactions between humans and objects.

HAR in video sequences is composed of two parts: (a) feature extraction for activity representation and (b) activity recognition. Many types of features have been used in the literature for activity representation. Spatiotemporal features represent activities as two-dimensional (3D) space–time volumes. Approaches utilising these features extract optical flow, speed, direction, trajectories etc. [1]. Silhouette and skeleton-based features extract joint angles, distances, centroids and contours, e.g. in [2, 3]. Semantic features represent actions as descriptive models using some set of rules, e.g. in [4]. In some approaches, activities are represented using statistical models such as hidden Markov models as state sequences [5]. In some approaches, multi-modal features are used by combining different types of features such as audio data with motion data [6]. For recognition part, two types of approaches have been used in the literature. The first type is template matching-based approaches which use templates based on various features extracted from the images and recognise the activities by matching these templates with the template database [7]. The second type is machine learning approaches which can be subdivided into supervised and unsupervised methods. Supervised approaches use labelled data for predicting the labels of unknown data [8]. Unsupervised techniques are used for unlabelled data [9]. Applications of HAR systems are many including content-based video retrieval, human–computer interaction, ambient intelligence, visual surveillance, robotics and ambient assisted living.

Aim of this paper is to make use of human skeletal data to detect certain activities in a video sequence. In the literature, numerous techniques have been used for the extraction of suitable features for action recognition in videos. However, in most of the current researches, skeleton data is combined with the depth data obtained through some specific depth device which limits the scope of the research to particular type of cameras such as Kinect which provides depth information. The proposed research shows that promising results can be achieved by utilising skeleton data from 2D video streams without the requirement of any depth information. Comparison of the results with the state of the art shows that with some further work, this approach can be utilised in implementing the vision-based activity recognition in surveillance and monitoring applications using standard closed-circuit television cameras. A secondary objective in this paper is to compare the performances of some widely used supervised learning techniques to find the best performing classifier for activity recognition.

Rest of this paper is arranged as follows: Section 2 discusses the recent work in the area of activity recognition. Section 3 describes in detail the proposed methodology. Section 4 provides the implementation details. Section 5 discusses the results obtained, and finally, conclusions are drawn in Section 6.

2 Related work

Some of the prior work related to the present research has been discussed in this section. Hbali *et al.* computed spatial features in videos using Minkowski and cosine distances between 3D positions of skeletal joints. Positions of skeletal joints were obtained using Kinect sensor. Then, temporal features were computed by calculating difference between coordinates of a certain joint and the maximum and minimum values of the same joint in entire video sequence. For activity classification, they used random forest (RF) algorithm [10]. Jalal *et al.* obtained human skeleton and depth maps through Kinect. In the first step, pixel differentiation was used for removing noise in the background.

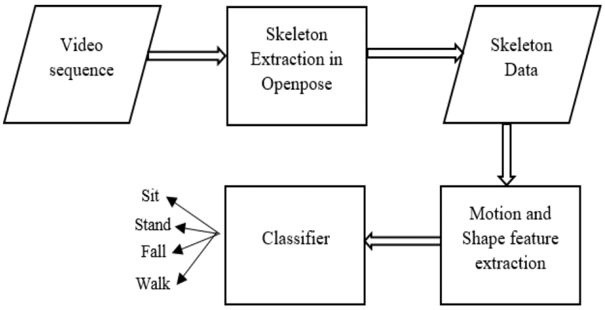

Video sequence
Skeleton Extraction in Openpose
Skeleton Data
Motion and Shape feature extraction
Classifier
Sit
Stand
Fall
Walk

Fig. 1 *System framework*

Then, human silhouettes were extracted by using connected component labelling method. Using these silhouettes they extract features including joint angles and distances, the frame-wise difference between joint positions, pixel intensities and orientation of gradients. In their research, hidden Markov models (HMMs) were used for training and testing [11]. Tamou *et al.* also used 3D joint positions extracted through Kinect for action recognition. They calculated difference in joints position in all frames and difference in joints positions between current frame and initial frame. Then, they concatenated both types of features and calculated mean vector for representation. This feature vector was then used as an input to the RF classifier for action classification [12]. The works given in [10–12] also use skeletal joints' features for activity recognition as are used in the proposed method; however, the positions of skeletal joints are obtained by using a depth camera such as Kinect in each case. In our work, we extract skeletal joints' positions directly from the video sequence without using any specific depth camera or device.

Albawendi *et al.* have used a combination of motion and shape features for detecting falls. They obtained human silhouettes in a video frame by background segmentation, and then they fit an ellipse around the silhouette to find change in shape in case of fall. They have used motion history image to find motion features. The features used for fall detection include orientation of the ellipse and rate of change of motion of the person [13]. Núñez-Marcos *et al.* have used optical flow to represent motion between two frames. They input optical flow images to convolutional neural network (CNN) for feature learning. A fully connected neural network is then used to generate fall/no fall signal [14]. Poonsri and Chiracharit have used Gaussian mixture models, and principal component analysis (PCA) to extract features such as orientation of human silhouettes in a video and aspect ratio for detecting falls [15]. Zerrouki *et al.* extracted silhouette-based features. In the first step, they use background subtraction to obtain human silhouettes. Then, they divide silhouette area into five occupancy zones occupied by head, arms and legs. The extracted features were composed of the ratio of each area with the total silhouette area. They used AdaBoost classifier for action classification [16].

Wang *et al.* used Kinect to make their dataset. The human silhouette was extracted using background subtraction and connected components technique. Then, centre of gravity of human silhouette was extracted. Features including ratio of upper and lower body and distance between centre and edge contour were extracted. At the end, learning vector quantisation (LVQ) neural network was used for action recognition [17]. Manzi *et al.* used unsupervised classification for extracting key poses from a video frame. 3D joints' positions were extracted using a depth camera and skeleton tracker. After that, *K* means clustering algorithm was used to find key poses in activity in the form of ordered pairs of cluster centres. Multiclass support vector machine (SVM) was used for classification [18]. Le *et al.* used Kinect to obtain 3D joints positions in space. They calculated joint angles and performed various experiments by varying the number of joints used for calculating the angles. In their research, four postures using SVM classifier were identified [19]. Wachs *et al.* used template matching technique for detecting pedestrian postures for intelligent vehicles. They used annotated silhouettes of human body to create a dictionary of body patches. Four filters including *x* and *y* derivatives, Gaussian and delta function were convolved with images. Cross-correlation of the filtered images was performed with the dictionary patches to find the location of the patch in the image. They used AdaBoost algorithm to combine weak learners to produce a strong classifier for their method [20].

Kushwaha and Srivastava used background subtraction for obtaining human silhouette. They calculated the distance between contour of the silhouette and its centre of mass and called it the distance signal feature. An SVM classifier was used for activity classification [21]. Wang *et al.* performed action recognition on untrimmed videos. A 150 frames temporal sliding window was used to segment videos into short video clips. Action recognition was performed separately on these clips, and the results of individual clips were combined to obtain the result of the complete video. They obtained appearance features using CNNs and extracted four motion features including histogram of oriented gradients, histogram of optical flow, horizontal motion boundary histogram and vertical motion boundary histogram. Combined motion and appearance features were then inputted to the multiclass SVM for final recognition [22].

In the presented paper, shape and motion features were extracted from the video frames to recognise four types of activities, i.e. sit, stand, walk and fall. These features were extracted using human skeleton data which was directly obtained from a video sequence.

3 Methodology

The proposed methodology consisted of three main steps including skeleton extraction, spatial and temporal features extraction and activity recognition. The complete framework for the approach is presented in Fig. 1.

3.1 Skeleton extraction

Skeleton data was extracted using OpenPose library.

3.1.1 OpenPose [23]: The library is based on the works of Cao *et al.* [24] that extracts the locations of skeletal joints for all the persons in an image or a video frame. The key points are extracted at a rate of 3.7 fps.

The output of OpenPose is stored in the form of a 3D array providing information about the person number, joint number and *x*, *y* coordinates of the respective joint in the frames. This information obtained from OpenPose was used to extract spatiotemporal features for action classification.

3.2 Spatiotemporal features extraction

Features were extracted using the positions of eight body joints including right hip, right knee, right ankle, right shoulder, left hip, left knee, left ankle and left shoulder. Out of the total 18 joints positions provided by OpenPose, these eight were selected because these joints are presumably more relevant to the types of activities which we wanted to recognise. Shape features included:

(a) angles between left/right hip and knee joints;
(b) angles between left/right knee and ankle joints;
(c) angles between left/right hip and ankle joints; and
(d) ratio of distances between left/right hip, knee and left/right knee and ankle joints.

Motion features included:

(a) displacement of left/right shoulder between consecutive frames;
(b) displacement of left/right hip between consecutive frames;
(c) displacement of left/right knee between consecutive frames; and
(d) displacement of left/right ankle between consecutive frames.

These 16 features were extracted for all the frames in a video.

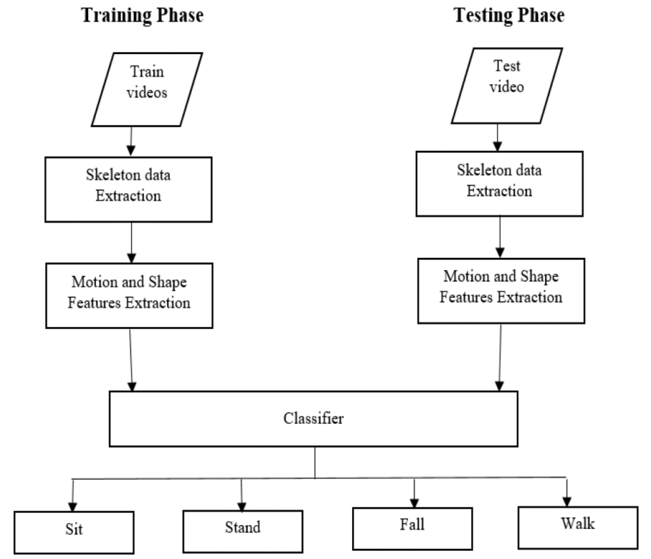


Fig. 2 *Flowchart for activity recognition process*

3.3 Post-processing

The feature vector obtained in the previous step could not be directly used in classification step since there was a need to remove any noise in case of wrongly detected key points. Hence, a ten frames window was used for averaging the output of feature vector, thus reducing the dimension of final feature vector with a factor of 10, i.e. (total number of frames in a video divided by 10) × 16.

3.4 Activity recognition

Supervised machine learning was used for activity recognition. Since the final performance of any algorithm is greatly affected by the performance of the selected classifier, choice of best classification technique is of prime importance for any method. In current research, five widely known supervised learning techniques were implemented and the results were compared to find the best performing classifier for the proposed method. The supervised learning techniques used in this research include:

(a) *Feed-forward back-propagation neural networks (BPNNs)*: Powerful parametric non-linear model for pattern recognition and prediction problems.
(b) *K-nearest neighbours (KNNs) classifier*: Simple and robust with only a few parameters for tuning (*k* and distance metric).
(c) *SVM classifier*: Widely used in the literature, margin maximisation in SVM allows robustness. Supports kernels for data which is not linearly separable. Regularisation allows for avoiding overfitting.
(d) *Linear discriminant classifier*: A widely used and analytically simple classification technique for multiclass classification.
(e) *Naive Bayes classifier*: An easy and fast algorithm for multiclass classification problems.

Activity recognition step included two phases:

(a) *Training phase*: Feature vectors for all the training videos were obtained and combined to form a train vector. Corresponding labels for the train videos were also stored separately for the testing phase.
(b) *Testing phase*: In this phase, the feature vector for the test video was calculated. This feature vector along with the train vector and the labels was then inputted to the classifier for final labelling. Fig. 2 gives a complete flowchart for the activity recognition process.

4 Implementation

The algorithm was developed using MATLAB Statistics and Machine Learning Toolbox. This section gives details for the network architectures of the classifiers and the datasets used for training and testing purposes.

4.1 Network architecture

Architecture for all the classifiers is given as follows.

4.1.1 Feed-forward BPNN: Also known as multilayer perceptron, feed-forward BPNN is a useful tool for classification problems. These have three types of layers including the input layer which receives the input data, the output layer at which the output is obtained and the hidden layer/s which lie between the other two types of layers. In this paper, the neural network was implemented using one hidden layer consisting of 14 neurones with sigmoid activation function. The output layer had four neurones with softmax function. About 50% of the data was used for training and 35% for testing. About 15% of the data was used for validation purpose.

4.1.2 KNNs classifier: It works on the principle of learning the class of new data samples by finding similar samples from the training data. Class of new data is found by majority voting of its KNNs. A distance function is used to find similarity between the neighbours. In the current research, KNN classifier was implemented using Mahalanobis distance as the distance metrics to find neighbours. The value of *k* was found by experimentally adjusting until the classifier achieved the lowest error. The empirically calculated value of *k* was 4.

4.1.3 SVM classifier: Out of many possible hyperplanes, SVM classifier finds the most optimal hyperplane which maximises the separation between classes. In this paper, MATLAB ECOC classifier was used which is the multiclass model for SVMs. Multiclass learning in this model is achieved by reducing multiclass problem to binary classification using one versus one coding design. A linear kernel function was used in the current research.

4.1.4 Discriminant analysis classifier: The type of discriminant used in this research was linear. Linear discriminant classifier searches for linear transformations to achieve best data separation by taking into account interclass and intraclass scatterers. Mean and covariance parameters for each class were found using fitcdiscr function which first calculates the sample mean for each class and then calculates sample covariance by subtracting sample mean from the class observations for each class.

4.1.5 Naive Bayes classifier: It uses probability theory and Bayes theorem for predicting the class of unknown data. This classifier was implemented such that Gaussian distribution was used for estimating the model parameters, i.e. mean and standard deviation for each class.

4.2 Dataset collection

Two publically available datasets were used for the performance evaluation of the proposed method.

4.2.1 Microsoft research (MSR) daily activity3d dataset [25]: This dataset is composed of the videos of ten people performing 16 different types of activities. For this research, a total of 68 videos with three labels, i.e. sit, stand and walk were used. The resolution of the videos is 640×480 px^2. Fig. 3 shows some samples from this dataset for three classes.

4.2.2 Le2i fall detection dataset [26]: This dataset contains videos captured at four different locations from a single camera. The actors perform various day-to-day activities. There are two categories in this dataset, i.e. videos with fall occurring and videos without fall. A total of 30 videos from different locations were used in this research with the label fall. The resolution of videos for this

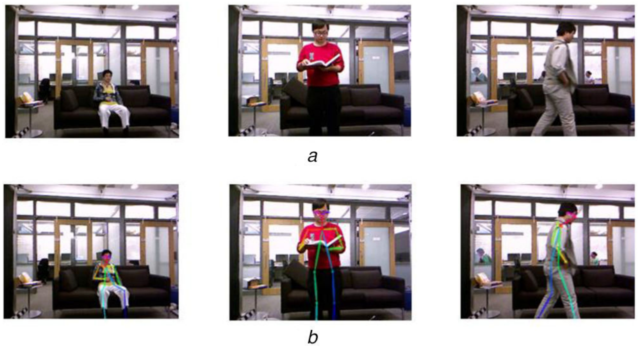


Fig. 3 *Activity samples for sit, stand and walk classes from MSR daily activity dataset*
(a) Before the extraction of the human skeleton, ***(b)*** After the extraction of human skeleton

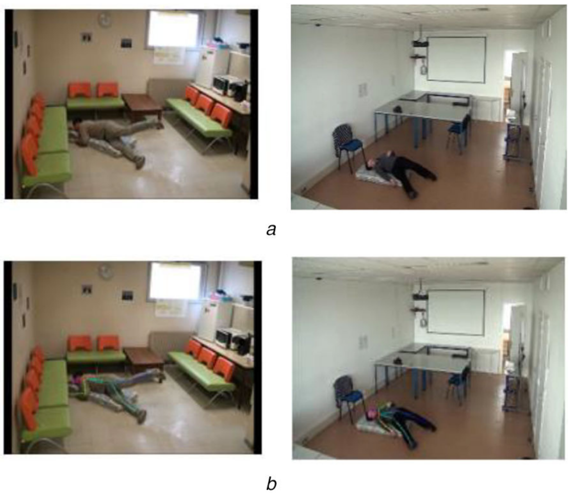


Fig. 4 *Activity samples for the class fall from Le2i fall detection dataset*
(a) Before the extraction of the human skeleton, ***(b)*** After the extraction of human skeleton

Table 1 Distribution of frames for training and testing

Dataset	Total frames	Train frames	Test frames
MSR	12,540	6070	6470
Le2i	3980	2250	1730
total	16,520	8320	8200

dataset is 320 × 240 px^2. Fig. 4 shows some samples of fall from Le2i dataset.

4.3 Dataset distribution

A total of 98 videos were used for evaluation of the proposed method after combining both the datasets. The total number of frames was 16,520. Table 1 gives the distribution of datasets for training and testing purposes.

5 Results and discussion

The results were obtained by performing two sets of experiments. First, without performing any normalisation to the train and test data, and second, after the application of normalisation. Results for both the types of experiments are presented in this section.

5.1 Performance parameters

Performance of the classifiers was evaluated based on three parameters including recall, precision and accuracy. These parameters were used to compare the performance of the proposed method with the state of the art.

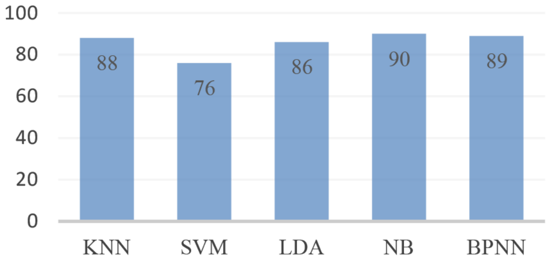


Fig. 5 *Comparison of accuracy without normalisation*

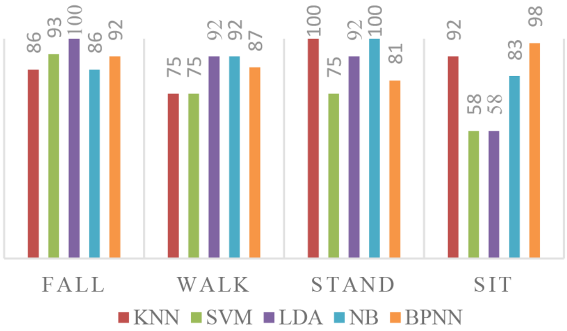


Fig. 6 *Comparison of recall without normalisation*

5.1.1 Recall: The recall is the number of true instances retrieved out of the total instances of a class. It is also known as sensitivity. The value is found using the equation below:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{1}$$

5.1.2 Precision: It is a measure of the number of true instances out of the total instances retrieved as positive of a class. It is also known as a positive prediction value. The value is found using the equation below:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

5.1.3 Accuracy: It is a ratio of correct predictions out of the total instances. The value is found using the equation below:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}} \tag{3}$$

5.2 Results without data normalisation

It was observed that the best overall results for first set of experiments were achieved using Naive Bayes classifier with an overall accuracy of 90% (Fig. 5). Lowest accuracy was observed with SVM classifier with only 76% samples correctly classified.

By the comparison of the results given in Fig. 6, it was observed that the class fall was best classified using linear discriminant analysis (LDA). LDA and Naïve Bayes (NB) both showed better performances for class walk. For stand, 100% recall was observed for both NB and KNN. Best sit recall was achieved using BPNN.

Also, it was observed from Fig. 7 that even though recognition rate of the class sit was relatively less as compared with the other classes for all the classifiers, the precision remained highest showing that very few samples from other classes were misidentified as sit. Moreover, precision for fall class remained lowest in all the classifiers.

The individual confusion matrices for all the classifiers are given in Fig. 8.

5.3 Results with data normalisation

The second set of experiments were performed after the application of data normalisation to convert angle values in the range [−1, 1]

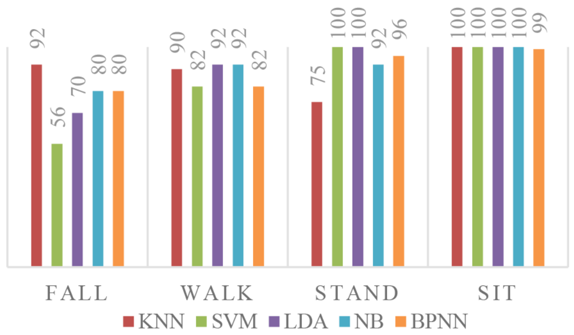


Fig. 7 *Comparison of precision without normalisation*

a

	Fall	Walk	Stand	Sit
Fall	86	7	7	0
Walk	8	75	17	0
Stand	0	0	100	0
Sit	0	0	8	92

b

	Fall	Walk	Stand	Sit
Fall	93	7	0	0
Walk	25	75	0	0
Stand	17	8	75	0
Sit	42	0	0	58

c

	Fall	Walk	Stand	Sit
Fall	100	0	0	0
Walk	8	92	0	0
Stand	0	8	92	0
Sit	42	0	0	58

d

	Fall	Walk	Stand	Sit
Fall	86	8	8	0
Walk	8	92	0	0
Stand	0	0	100	0
Sit	17	0	0	83

e

	Fall	Walk	Stand	Sit
Fall	92	5	1	2
Walk	10	87	3	0
Stand	7	12	81	0
Sit	1	1	0	98

Fig. 8 *Confusion matrices without normalisation for*
(a) KNN, ***(b)*** SVM, ***(c)*** LDA, ***(d)*** NB, ***(e)*** BPNN

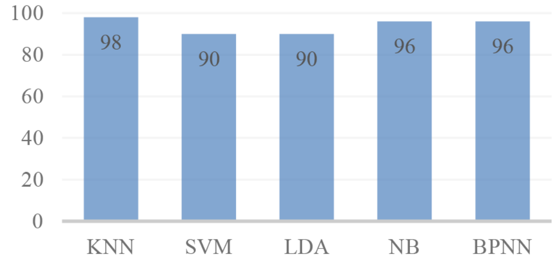


Fig. 9 *Comparison of accuracy after normalisation*

using (4) and displacement values and ratios in the range [0, 1] using (5). Since the datasets had different resolutions, it was postulated that better results could be achieved if the data was converted to a similar range by the application of normalisation

$$(x - x_{\min})/(x_{\max} - x_{\min}) \tag{4}$$

$$2 \times (x - x_{\min})/(x_{\max} - x_{\min}) - 1 \tag{5}$$

It was observed that accuracy for all the classifiers increased significantly (Fig. 9) with the highest increase in SVM, i.e. from 76 to 90%. Best accuracy was achieved in KNN with 98% of the videos classified correctly. A remarkable increase in recall and precision was also observed. KNN, SVM, LDA and NB classifiers gave 100% recall for the classes fall, walk and stand (Fig. 10). BPNN also gave good results for these three classes, and best results were achieved for the class sit with 100% samples correctly classified.

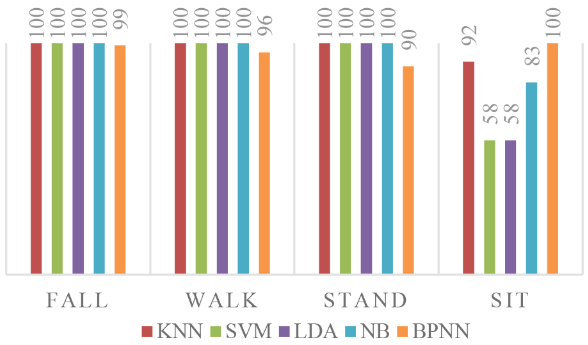


Fig. 10 *Comparison of recall after normalisation*

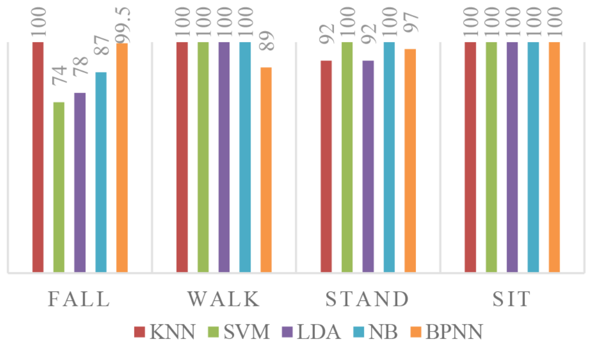


Fig. 11 *Comparison of precision after normalisation*

a

	Fall	Walk	Stand	Sit
Fall	100	0	0	0
Walk	0	100	0	0
Stand	0	0	100	0
Sit	0	0	8	92

b

	Fall	Walk	Stand	Sit
Fall	100	0	0	0
Walk	0	100	0	0
Stand	0	0	100	0
Sit	42	0	0	58

c

	Fall	Walk	Stand	Sit
Fall	100	0	0	0
Walk	0	100	0	0
Stand	0	0	100	0
Sit	34	0	8	58

d

	Fall	Walk	Stand	Sit
Fall	100	0	0	0
Walk	0	100	0	0
Stand	0	0	100	0
Sit	17	0	0	83

e

	Fall	Walk	Stand	Sit
Fall	98	1	1	0
Walk	1	96	3	0
Stand	0	10	90	0
Sit	0	0	0	100

Fig. 12 *Confusion matrices after data normalisation for*
(a) KNN, ***(b)*** SVM, ***(c)*** LDA, ***(d)*** NB, ***(e)*** BPNN

There was no improvement in the results of the class sit after normalisation for rest of the classifiers.

Another observation was that though precision for fall was good for both BPNN and KNN, it was quite less as compared with other classes for LDA, SVM and NB which meant that for these classifiers, most of the samples from other classes were misclassified as fall henceforth decreasing the precision for the same. Highest precision was found for the class sit with all the classifiers giving 100% precision (Fig. 11).

The individual confusion matrices for all the classifiers are given in Fig. 12.

5.4 Comparison to state of the art

Table 2 gives a comparison of the individual results of all the classes with state-of-the-art methods. Poonsri and Chiracharit used

Table 2 Comparison with state of the art

	Technique	Fall, %	Walk, %	Sit, %
proposed method with normalised skeleton data	KNN	100	100	92
	SVM	100	100	58
	LDA	100	100	58
	NB	100	100	83
	BPNN	99	96	100
Poonsri and Chiracharit	MoG, PCA	93	—	—
Núñez-Marcos *et al.*	optical flow, CNN	99	—	—
Hbali *et al.*	3D skeleton and depth features, RF	—	88	75
Tamou *et al.*	3D skeleton and depth features, RF	—	95	90
Jalal *et al.*	3D skeleton and depth features, HMMs	—	96	87

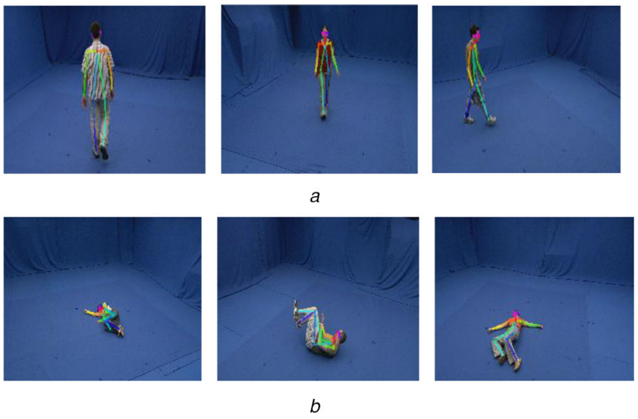


Fig. 13 *Different views of activity samples from i3D post multi-view dataset after skeleton extraction*
(a) Walk, ***(b)*** Fall

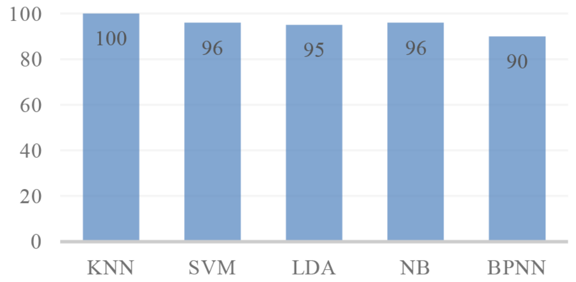


Fig. 14 *Comparison of accuracy for i3d multi-view dataset*

58 videos from Le2i dataset for fall detection. Núñez-Marcos *et al.* report their results for fall and no fall classes on Le2i dataset. The proposed method outperforms both the techniques for detection of class fall. Hbali *et al.*, Tamou *et al.* and Jalal *et al.* report their results on MSR daily activity dataset. Their results for classes walk and sit have been compared with the proposed method.

5.5 Experimental results on multi-view dataset

To check the robustness of the proposed method in a multi-view scenario, another set of experiments was performed on i3D post multi-view human action dataset [27]. This dataset consists of image sequences of 8 actors performing 13 actions in 8 different camera views. The resolution of the videos is 1920 × 1080 px^2. For this experiment, a total of 128 videos with labels ‘walk’ and ‘fall’ were used. The total number of frames used was 1344 out of which half were used for training and half for testing purpose. Fig. 13 shows some samples from this dataset.

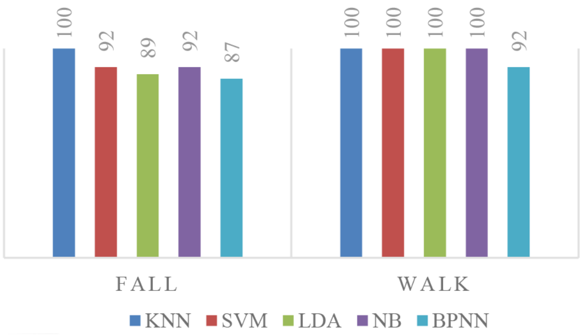


Fig. 15 *Comparison of recall for i3d multi-view dataset*

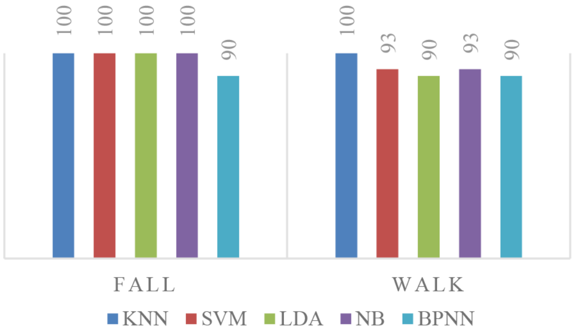
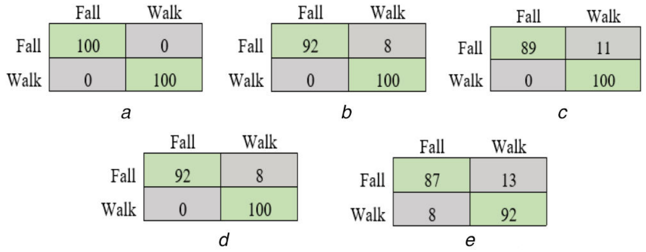


Fig. 16 *Comparison of precision for i3d multi-view dataset*



a

	Fall	Walk
Fall	100	0
Walk	0	100

b

	Fall	Walk
Fall	92	8
Walk	0	100

c

	Fall	Walk
Fall	89	11
Walk	0	100

d

	Fall	Walk
Fall	92	8
Walk	0	100

e

	Fall	Walk
Fall	87	13
Walk	8	92

Fig. 17 *Confusion matrices for i3D multi-view dataset for*
(a) KNN, ***(b)*** SVM, ***(c)*** LDA, ***(d)*** NB, ***(e)*** BPNN

The results were collected and compared for all the five classifiers. The parameters and network architectures of all the classifiers were kept the same as described in Section 4.1. It was observed that the KNN classifier showed best performance with all the videos correctly classified for both the classes (Fig. 14).

From the results shown in Fig. 15, it was observed that the class walk was recognised with a 100% recall using KNN, SVM, LDA and Naive Bayes classifiers while BPNN gave slightly less recall of 92%. For the recognition of ‘fall’, KNN outperformed all the other classifiers with 100% result.

From Fig. 16, it can also be noted that even though recall of walk was higher for all the classifers, the precision was 100% for KNN only and relatively lesser for the other classifiers. It was because some fall videos were identified as walk, thus reducing the precision for the later. However, precision for fall remained 100% for KNN, SVM, Naive Bayes and LDA, The individual confusion matrices for all the classifiers are given in Fig. 17.

6 Conclusions

In this paper, four activity classes were recognised using 2D skeletal data and supervised machine learning. It was found that compatible results can be achieved using 2D skeleton data for activity recognition as compared with the techniques employing 3D skeletal data. Five widely known classifiers were used for activity recognition, and results were compared. It was observed that the best results were achieved with KNN classifier with an overall accuracy of 98%. It was also observed that better results were obtained after applying normalisation to train and test data. Fall, walk and stand were best recognised with an accuracy of 100% with KNN, SVM, LDA and Naive Bayes classifiers. Sit class was best recognised with an accuracy of 100% with BPNN. Fall had relatively low precision as compared with other classes for SVM, LDA and Naive Bayes classifiers while sit had highest precision for all the classifiers. The robustness of the proposed method was also tested in a multi-camera view scenario for two classes. It was observed that KNN outperformed rest of the classifier by giving a 100% result. In future, this work can be extended further by adding more activity classes for better human behaviour understanding in videos.

7 Acknowledgments

We thank the authors of OpenPose library whose work laid the foundation for this research

8 References

[1] Wrzalik, M., Krechel, D.: 'Human action recognition using optical flow and convolutional neural networks'. 2017 16th IEEE Int. Conf. Machine Learning and Applications (ICMLA), Cancun, Mexico, December 2017

[2] Zhu, G., Zhang, L., Shen, P., *et al.*: 'An online continuous human action recognition algorithm based on the Kinect sensor', *Sensors*, 2016, **16**, (2), p. 161

[3] Goudelis, G., Tsatiris, G., Karpouzis, K.: 'Fall detection using history triple features'. Proc. Eighth ACM Int. Conf. Pervasive Technologies Related to Assistive Environments, no. 81, Corfu, Greece, July 2015

[4] Zhang, Z., Wang, C., Xiao, B., *et al.*: 'Robust relative attributes for human action recognition', *Pattern Anal. Appl.*, 2015, **18**, (1), pp. 157–171

[5] Song, Y., Morency, L.P., Davis, R.: 'Action recognition by hierarchical sequence summarization'. Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition, Portland, OR, USA, 2013, pp. 3562–3569

[6] Wu, Q., Wang, Z., Deng, F., *et al.*: 'Realistic human action recognition with multimodal feature selection and fusion', *IEEE Trans. Syst. Man Cybern. Syst.*, 2013, **43**, pp. 875–885

[7] Li, C., Hua, T.: 'Human action recognition based on template matching', *Procedia Eng.*, 2011, **15**, pp. 2824–2830

[8] Zerrouki, N., Houacine, A.: 'Automatic classification of human body postures based on the truncated SVD', *J. Adv. Comput. Netw.*, 2014, **2**, (1), pp. 58–62

[9] Yang, Y., Saleemi, I., Shah, M.: 'Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013, **35**, pp. 1635–1648

[10] Hbali, Y., Hbali, S., Lahoucine, B., *et al.*: 'Skeleton-based human activity recognition for elderly monitoring systems', *IET Comput. Vis.*, 2017, **12**, (1), pp. 16–26

[11] Jalal, A., Kamal, S., Kim, D.: 'A depth video-based human detection and activity recognition using multi-features and embedded hidden Markov models for health care monitoring systems', *Int. J. Inter. Multimed. Artif. Intell.*, 2017, **4**, (4), pp. 54–62

[12] Tamou, A., Ballihi, L., Aboutajdine, D.: 'Automatic learning of articulated skeletons based on mean of 3D joints for efficient action recognition', *Int. J. Pattern Recognit. Artif. Intell.*, 2016, **31**, (4), pp. 1–17

[13] Albawendi, S., Lotfi, A., Powell, H., *et al.*: 'Video based fall detection using features of motion, shape and histogram'. Proc. 11th Pervasive Technologies Related to Assistive Environments Conf., Corfu, Greece, June 2018

[14] Núñez-Marcos, A., Azkune, G., Arganda-Carreras, I.: 'Vision-based fall detection with convolutional neural networks', *Wirel. Commun. Mob. Comput.*, 2017, **2017**, (1), pp. 1–16

[15] Poonsri, A., Chiracharit, W.: 'Fall detection using Gaussian mixture model and principal component analysis'. Ninth Int. Conf. Information Technology and Electrical Engineering (ICITEE), Phuket, Thailand, 2017

[16] Zerrouki, N., Harrou, F., Sun, Y., *et al.*: 'Vision-based human action classification using adaptive boosting algorithm', *IEEE Sens. J.*, 2018, **18**, (12), pp. 5115–5121

[17] Wang, W., Chang, J., Haung, S., *et al.*: 'Human posture recognition based on images captured by the Kinect sensor', *Int. J. Adv. Robot. Syst.*, 2016, **13**, (2), p. 1

[18] Manzi, A., Dario, P., Cavallo, F.: 'A human activity recognition system based on dynamic clustering of Skeleton data', *Sensors (Basel)*, 2017, **17**, (5), p. 1100

[19] Le, T., Nguyen, M., Nguyen, T.: 'Human posture recognition using human skeleton provided by Kinect'. 2013 Int. Conf. Computing, Management and Telecommunications, Ho Chi Ming City, Vietnam, January 2013

[20] Wachs, J.P., Kölsch, M., Goshorn, D.: 'Human posture recognition for intelligent vehicles', *J. Real-Time Image Process.*, 2010, **5**, (4), pp. 231–244

[21] Kushwaha, A.K.S., Srivastava, M.R.: 'A framework for human activity recognition using pose feature for video surveillance system', Int. J. Comput. Appl., (0975 - 8887) Next Generation Technologies for e-Business, e-Education and e-Society (NGTBES-2016), Ghaziabad, India, March 2016

[22] Wang, L., Qiao, Y., Tang, X.: 'Action recognition and detection by combining motion and appearance features', THUMOS14 Action Recognition Challenge, 2014, pp. 1–6

[23] OpenPose library. Available at https://github.com/CMU-Perceptual-Computing-Lab/openpose, (accessed 27th September 2017)

[24] Cao, Z., Simon, T., Wei, S., *et al.*: 'Real-time multi-person 2D pose estimation using part affinity fields'. Conf. Computer Vision and Pattern Recognition, Honolulu, Hawaii, 2017

[25] Li, W.: 'MSR daily activity 3D dataset', 2012. Available at https://www.uow.edu.au/~wanqing/#Datasets, (accessed 4th April 2018)

[26] Le2i Fall Detection Dataset, 2013. Available at http://le2i.cnrs.fr/Fall-detection-Dataset?lang=fr (accessed 17th July 2018)

[27] Gkalelis, N., Kim, H., Hilton, A., *et al.*: 'The i3DPost multi-view and 3D human action/interaction', Proc. Conf. Visual Media Production, London, UK, 2009, pp. 159–168. Available at http://kahlan.eps.surrey.ac.uk/i3dpost_action/ (accessed 17th July 2018)