Action Recognition from Motion Capture Data using Meta-cognitive RBF Network Classifier

Suraj Vantigodi
Video Analytics Lab, SERC
Indian Institute of Science,
Bangalore, India,
suraj.vantigodi@gmail.com

Venkatesh Babu Radhakrishnan
Video Analytics Lab, SERC
Indian Institute of Science,
Bangalore, India,
venky@serc.iisc.in

***Abstract*—Action recognition plays an important role in various applications, including smart homes and personal assistive robotics. In this paper, we propose an algorithm for recognizing human actions using motion capture action data. Motion capture data provides accurate three dimensional positions of joints which constitute the human skeleton. We model the movement of the skeletal joints temporally in order to classify the action. The skeleton in each frame of an action sequence is represented as a 129 dimensional vector, of which each component is a 3D angle made by each joint with a fixed point on the skeleton. Finally, the video is represented as a histogram over a codebook obtained from all action sequences. Along with this, the temporal variance of the skeletal joints is used as additional feature. The actions are classified using Meta-Cognitive Radial Basis Function Network (McRBFN) and its Projection Based Learning (PBL) algorithm. We achieve over 97% recognition accuracy on the widely used Berkeley Multimodal Human Action Database (MHAD).**

***Keywords*—Human action recognition, motion capture, Meta-Cognitive Radial Basis Function Network, Projection Based Learning.**

I. Introduction

Human action recognition is a challenging research problem in computer vision which has numerous applications in areas such as Human Computer Interface (HCI), Robotics and Automation, entertainment, etc. Recognizing of human actions may help us in identifying suspicious or abnormal activities. In gaming applications, the traditional controllers such as mouse and keyboard are less user-friendly. Using the body movements and hand gestures makes the game controlling more attractive and more user-friendly. Considering the vast variety of applications, action recognition has attracted a great deal of research over the past years. But most of the research has been concentrated on recognizing other action sequences using sequence of 2D images. Not untill recently numerous researchers have started using MoCap data for action recognition [1] and [2]. The newly arrived low cost depth sensor Microsoft Kinect and its real-time MoCap system [3], is extensively used in Human Computer Interaction applications like user interface for games, gait recognition [4], action recognition using depth [5] and [6].

Human action recognition has been of interest to researchers in computer vision [7] [8] [9] [10] [11] for many years. The problem can be defined as: given a sequence of images of a video of a human performing an action, the algorithm should predict the action being performed. One major hurdle in action recognition is the estimation of body parts in the action sequence. In the proposed approach, we do not address how to predict body parts and we assume that we already know them. We use the skeleton data provided by the motion capture system, KinectSDK or from a pose tracking system. Even if accurate 3D positions are available, the classification of MoCap sequences faces the difficulties of interclass similarity between different actions and intraclass variety of different performance instances of the same actions. For example, a punch action performed by different subjects can have variety of styles, and even the same subject does not perform the action exactly the same way each time. Different instance of the same action can also vary in the speed and the number of repetitions of the action.

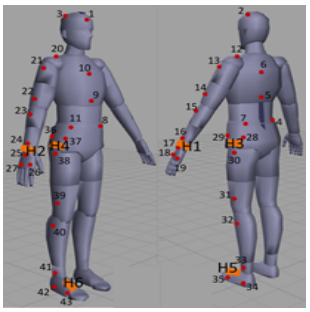

Fig. 1. *43 points forming the skeleton [12]*

Existing approaches for action recognition can be divided into two categories with respect to the data types that are used, 2D image sequences or video based, e.g. [7] [8] [9] [10] and those based on 3D motion streams on skeleton data, e.g. [12] [13] [14]. 3D approaches have many advantages over a 2D approachs as the dependence on viewpoint and illumination has been removed. Nonetheless, most algorithms use a 2D approach due to easy availability of video inputs and difficulty of recovering 3D information. Most state-of-the-art activity recognition methods which make use of 2D image stream, extract spatio-temporal interest points from monocular videos and learn their statistics, as manifested by work in [15] [16] and [17] . Ali and Shah [18] recognized human actions using kinematic features. Tran and Sorokin [19] used motion context and a learned nearest neighbor metric for classifying actions in YouTube videos of sporting events.

Most of the 3D based approaches on the skeleton data are focused on modeling the movement of the skeletal joints. The

pixel information, such as color, is not available. The action sequence is represented by the set of 3D joint positions of the skeletal points. Kurillo et al.[20] propose the use of histogram of most informative joints. They try to model the action sequence as the histogram of occurrences of the top k important joints, which are found out by a hard threshold (most important joints). However, the authors fail to take temporal information into account. In [12], Kurillo et al. have represented each action sequence as a collection of 21-dimensional joint angle feature. Each action sequence is divided into 60 temporal windows and the variance of joint angle in each temporal window acts as the feature for each action sequence. They further make use of other modalities such as RGB data, depth data along with the motion capture data to obtain better classification accuracy.

When different subjects perform the same action, we observe that the same set of joints are activated. Even if the subjects perform same action differently, generating dissimilar joint trajectories, the joints used and the order in which they are used while performing the action remain the same. In our approach, we take advantage of this observation to capture temporal variances in human skeletal motion in a given action. We try to form a bag of words model for the action sequence by considering the variation in the angle made by the joints with respect to a fixed point on the skeleton as the feature.

The remainder of the paper is organized as follows. In section II we provide details of feature extraction and classification technique used. Various experimentation and results are discussed in section III, and finally, section IV summarizes and concludes the paper.

II. PROPOSED APPROACH

The movement of joints is unique for a given action. In this paper, we try to model the temporal movement of various points constituting the skeleton. We represent the given action sequence as a set of features, which are scale and view independent. The skeleton in each frame of an action sequence is represented as a 129 dimensional vector, of which each component is the 3D angle made by each joint with the hip joint. Finally, the video is represented as a histogram over a codebook obtained from all action sequences. Along with this, the temporal variance of the skeletal joints is also used as an additional feature.

These feature vectors are then used to classify the actions using a Projection Based Learning (PBL)algorithm of a Meta-cognitive Radial Basis Function Network (McRBFN). McRBFN emulates the Nelson and Narens model of human meta-cognition [21], and has 2 components, namely, a cognitive component and a meta-cognitive component. A radial basis function network with Gaussian activation function at the hidden layer is the cognitive component of McRBFN and a self-regulatory learning mechanism is its meta-cognitive component. McRBFN begins with zero hidden neurons, adds and prunes neurons until an optimum network structure is obtained. The self-regulatory learning mechanism of McRBFN self-regulated learning [22], [23], [24] to decide what-to-learn, when-to-learn and how-to-learn in a meta-cognitive framework.

A. Action Representation using BoW model

Bag of words is a standard approach used in document classification, were in a sparse histogram is formed by counting the occurrences of different words. The same approach can be applied to image features as well and is called the bag of visual words, which is a sparse vector of occurrence counts of a vocabulary of features as proposed in [25]. We represent the given action sequence as bag of words model by treating features of action sequence as the word.

In our approach, each frame of an action sequence is represented as a fixed length feature vector. We extract the features for all the frames of all the action sequences and do k-means clustering to obtain cluster centers or the code words. The number of cluster centers is fixed at 64. Once the codewords are obtained, each action is represented as a histogram over the codewords. This representation counts the occurrences of different representative poses (code words) in the action sequence. The BOW representation thus obtained will be representative of the action and is used to train the classifier.

B. Feature extraction

We model the given action sequence as a combination of features. First, the BoW model of the action sequence obtained with 3D angle made by each joint with the fixed point on the skeleton as the feature/word. The temporal variance of skeletal joints and the time weighted variance of the joints features along with the BoW representation help in better classification of the actions. The details about feature extraction are given below.

1) 3D Angle Representation: We represent each frame of the action sequence as a $129D$ length angle feature vector. $129D$ feature vector is obtained by computing $3D$ angle of each joint with the fixed point on the skeleton. In our experimentation, we chose the fixed point to be the midpoint of 9^{th} and 10^{th} joint. We compute the angles made by all the 43 joints in the $x-y$, $y-z$ and $z-x$ planes.

Algorithm 1 ***: BoW***

Input: 129D angle feature of all the frames of action sequence $\{x_1, x_2 ... x_m\}$, Codewords $\{c_1, c_2 ... c_k\}$, where $c_i, x_t \in R^d$
Output: Histogram H
  for i = 1 to k **do**
    $H_i = 0$
  end for
  for t = 1 to m **do**
    $i = \underset{j \in \{1...K\}}{\text{argmin}} ||x_t - c_j||_2^2$
    $H_i = H_i + 1$
  end for
  $H = \frac{H}{M}$

Once we obtain the 129D features for each frame of the given action sequence, we obtain BoW representation for the action sequence by assigning each 129D feature vector to the nearest cluster center or the code word. The final BoW feature for the sequence is of dimension 64, Which represents the number of occurrences of the 64 codewords in the action sequence. For instance in the jumping jacks action sequence,

the 5^{th} pose in Figure 2 will have spike in the histogram bin corresponding to that pose.

The cluster centers obtained from the dictionary, represent the representative skeleton structures. The figure shows the representative skeletons extracted. We extract 64 clusters, which are nothing but 64 representative skeletons. The histogram represents the frequency of occurrence of these 64 representative skeleton configurations in the given action sequence.

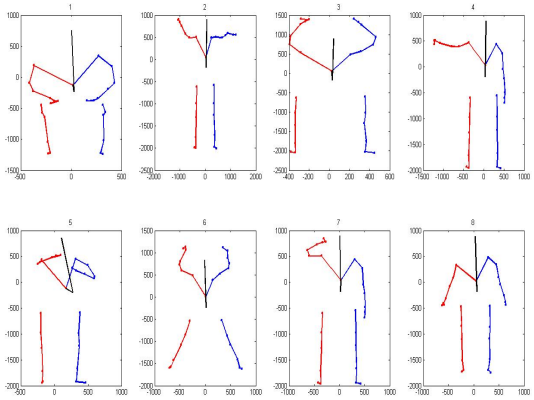

Fig. 2. *Representative sleleton structures corresponding to first eight clusters learnt from the dictionary*

2) Temporal variance of skeletal points: Different skeletal points get activated while performing different activities. Consider for example clapping and kicking actions. While performing a clapping action, the movement of hands is more than the legs. Similarly while kicking, the motion in legs more compared to the hands. So in a way, what all points are moving while performing an action gives some information about the action being performed. In order to find out which all skeletal points are moving and is the motion along x, y or z axis, we compute the temporal variance and the time weighted variance of the skeletal points, as proposed in [26]

We find the variance of the skeletal points in x, y and z directions and use this as a feature vector to classify the actions. This forms a 129 (43×3) length feature vector. The variance of the joints tell us about which all joints are engaged in performing the action. Variances along x, y and z indicate how much is the variance of the joint in that particular dimension. For instance, waving action, will have a high variance of points belonging to hand in x and y dimension but very less along z. Similarly, clap action will have high variance along x and z but less in y. Variance of j^{th} joint is represented as $(\sigma_x^j, \sigma_y^j, \sigma_z^j)$,where $j \in [1 \ldots 43]$. Thus we get the feature of size 129 (43×3).

Variance of skeletal joints alone is not sufficient for classification of all actions because it does not have any temporal information. Consider for example, actions sit down and stand up actions. These actions are same, the only difference being one is the reverse of the other. So, if we only use variance of joints for classifying these actions, there will be confusion. In order to solve this problem, we have to embed time information in the feature. In order to overcome this problem, time weighted variance is computed.

In time weighted variance, the frames are weighed linearly, before comping the variance as proposed in [26]. The first frame is given the weight 1 and the last frame is weighed 2. the weight is linearly increased from 1 to 2, equation 3 illustrates the weighing scheme. Fig.3 illustrates what additional information it adds, which would help us in classifying the actions. The sit down action has positive values for the variance of each joint, whereas the stand up action has negative values and the sit down and stand up action has both positive and negative variance. Time weighted variance along x dimension for joint j is computed as shown in equation 2. Similar computation is carried out for y and z dimensions.

$$v_x^j = var(x^j(t) \times w(t)) \tag{1}$$

where x^j is the x co-ordinate of j^{th} joint at time t and $w(t)$ is the weight given to each frame and it is computed as shown below in equation 3.

$$w(t) = 1 + \frac{t - t_0}{t_{end} - t_0} \tag{2}$$

Where,t is the current frame, t_0 denotes the first frame and t_{end} the last frame. The final feature vector for a given action sequence is thus represented as $F = [\sigma_x, \sigma_y, \sigma_z, v_x, v_y, v_z]$ which is of length 258 (43×6).

It can be observed from (Fig. 3(a), 3(c)) that the variance signal for the sit down and stand up actions look the same. So if we use only that feature, there will be confusion in classification. But when we take a look at time weighted variance (Fig. 3(b), 3(d)), the signals look different because of the temporal information introduced.

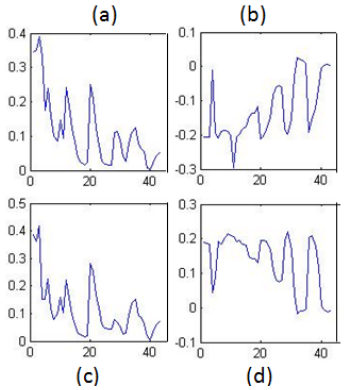

Fig. 3. *(a), (c) show the variance of 43 joints for sit down and stand up. (b) and (d) show time weighted variance for sit down and stand up actions. [26]*

From the histogram of the 3D angles, for a given action sequence we have a 64 dimensional vector. The variance of skeletal points and the time weighted variance of skeletal points are each of length 129. So the each action sequence is represented as a 322 length feature vector by concatenating all the 3 features. These set of features are used to classify the action using PBL-McRBFN classifier, the details of which is given in the next section.

C. PBL-McRBFN Classifier Overview

Consider a set of training samples $\mathbf{x}^t \in \Re^m$, where $\Re^m$ is the m- dimensional feature vector of the t^{th} training sample.

McRBFN is used to approximate the function $\mathbf{x}^t \in \Re^m \rightarrow \mathbf{y}^t \in \Re^n$, where n is the number of classes. Let $c^t \in (1, n)$ denote the class labels. The McRBFN formulation is adapted from [27] and has2 major components, namely the cognitive component and the meta-cognitive component.

1) Cognitive component of McRBFN: The cognitive component of McRBFN is a radial basis function network. For The hidden layer Gaussian activation function is used. For a given input $\mathbf{x}^t$, the predicted output $\widehat{y}_j^t$ is given in Equation 3.

$$\widehat{y}_j^t = \sum_{k=1}^{K} w_{kj} h_k^t, \qquad j = 1, \cdots, n \tag{3}$$

Where w_{kj} is the weight connecting the k^{th} hidden neuron to the j^{th} output neuron and h_k^t is the response of the k^{th} hidden neuron to the input $\mathbf{x}^t$ is given in Equation 4.

$$h_k^t \quad = \quad exp\left(-\frac{\|\mathbf{x}^t - \mu_k^l\|^2}{(\sigma_k^l)^2}\right) \tag{4}$$

Where $\mu_k^l \in \Re^m$ is the center and $\sigma_k^l \in \Re^+$ is the width of the k^{th} hidden neuron. l represents the corresponding class of the hidden neuron.

Projection based learning algorithm is used to estimate the optimal network output parameters. For t consecutive samples, the error function is given by Equation 5.

$$J(\mathbf{W}) = \frac{1}{2}\sum_{i=1}^{t}\sum_{j=1}^{n}\begin{cases} 0 & \text{if } \; y_j^i \widehat{y}_j^i > 1 \\ \left(y_j^i - \widehat{y}_j^i\right)^2 & \text{otherwise} \end{cases} \tag{5}$$

The resultant optimal output weights $\mathbf{W}^*$ corresponding to the error function $J(\mathbf{W}^*)$ is represented as in Equation 6.

$$\sum_{k=1}^{K} a_{kp} w_{kj} \quad = \quad b_{pj}, \tag{6}$$

which is a system of linear equation in the form of $\mathbf{W}^* = \mathbf{A}^{-1}\mathbf{B}$. Here, a_{kp} is the projection matrix and b_{pj} is the output matrix. The derivation of this equation can be found in [27].

2) Meta-cognitive component of McRBFN: The meta-cognitive component models the dynamics of the cognitive component. During the learning process, the meta-cognitive component monitors the cognitive component and updates its dynamic model of the cognitive component. The meta-cognitive component uses predicted class label ($\widehat{c}^t$), maximum hinge loss (E^t), confidence of classifier ($\hat{p}(c^t|\mathbf{x}^t)$) and class-wise significance (ψ_c) as the measure of knowledge in the new training sample. The meta-cognitive component builds two sample-based and two neuron-based learning strategies using the knowledge measures and the self-regulated thresholds. One of these strategies is chosen for the new training sample such that the cognitive component learns them accurately and achieves better generalization performance.

The meta-cognitive component knowledge measures are defined as shown below:

Predicted class label ($\widehat{c}^t$): $\widehat{c}^t = \arg\max_{j \in 1, \cdots, n} \widehat{y}_j^t$

Maximum hinge loss (E^t): The hinge loss $\left(\mathbf{e^t} = \left[e_1^t, \cdots, e_j^t, \cdots, e_n^t\right]^T\right) \in \Re^n$ is

$$e_j^t = \begin{cases} 0 & \text{if } \; y_j^t \widehat{y}_j^t > 1 \\ y_j^t - \widehat{y}_j^t & \text{otherwise} \end{cases} \qquad j = 1, \cdots, n \tag{7}$$

The maximum absolute hinge loss (E^t) is given by

$$E^t = \max_{j \in 1,2,\cdots,n} \left|e_j^t\right| \tag{8}$$

Confidence of Classifier ($\hat{p}(c^t|\mathbf{x}^t)$) is given as

$$\hat{p}(j|\mathbf{x}^t) = \frac{min(1, max(-1, \hat{y}_j^t)) + 1}{2}, \quad j = c^t \tag{9}$$

Class-wise Significance (ψ_c): The class-wise distribution portrays as the key impact factor in calculating the performance of the classifier. Let K^c be the number of neurons associated with the class c, then class-wise spherical potential or class-wise significance (ψ_c) is defined as

$$\psi_c = \frac{1}{K^c}\sum_{k=1}^{K^c} h\left(\mathbf{x}^t, \mu_k^c\right) \tag{10}$$

The ψ_c is the knowledge contained in the sample, a smaller value of ψ_c (close to 0) indicates that the sample is new.

3) Learning strategies: The meta-cognitive component has four learning strategies, which directly addresses the basic principles of self-regulated human learning.
Sample delete strategy is given by

$$\widehat{c}^t \; == \; c^t \;\; \textbf{AND} \;\; \hat{p}(c^t|\mathbf{x}^t) \; \geq \; \beta_d \tag{11}$$

Where β_d is the deletion threshold. If β_d is close to 1 then all samples participates in the learning which may result in over-training. Please refer [27] for further understanding on deletion thresholds.
Neuron growth strategy is given by

$$\left(\widehat{c}^t \neq c^t \;\textbf{OR}\; E^t \; \geq \; \beta_a\right) \;\textbf{AND}\; \psi_c(\mathbf{x}^t) \; \leq \; \beta_c \tag{12}$$

Where β_c is the knowledge measurement threshold, the value of which lies between 0 and 1. If β_c is closer to 0,then very few neurons will be added to the network and on the other hand, if β_c is closer to 1, the resultant network may contain very poor generalization ability. Similar to [27], based on the nearest neuron distances, the new hidden neuron center and width parameters are determined for the different overlapping/no-overlapping conditions.

While adding a neuron to McRBFN, the output weights are initialized as:
The size of matrix $\mathbf{A}$ is increased from $K \times K$ to $(K+1) \times (K+1)$

$$\mathbf{A}^t = \left[\begin{array}{c|c} \mathbf{A}_{K\times K}^{t-1} + \left(\mathbf{h}^t\right)^T \mathbf{h}^t & \mathbf{a}_{K+1}^T \\ \hline \mathbf{a}_{K+1} & a_{K+1,K+1} \end{array}\right] \tag{13}$$

The size of $\mathbf{B}$ is increased from $K \times n$ to $(K+1) \times n$

$$\mathbf{B}^{t}_{(K+1)\times n} = \begin{bmatrix} \mathbf{B}^{t-1}_{K\times n} + (\mathbf{h}^{t})^{T} (\mathbf{y}^{t})^{T} \\ \mathbf{b}_{K+1} \end{bmatrix} \tag{14}$$

and $\mathbf{b}_{K+1} \in \Re^{1\times n}$ is a row vector assigned as

$$b_{K+1,j} = \sum_{i=1}^{K+1} h^{i}_{K+1} \tilde{y}^{i}_{j},\ j = 1, \cdots, n \tag{15}$$

After neglecting $\mathbf{h}^{t}$ vector in Eqs. (13) and (15) the output weights are estimated finally as

$$\begin{bmatrix} \mathbf{W}^{t}_{K} \\ \mathbf{w}^{t}_{K+1} \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{A}^{t-1}_{K\times K} & \mathbf{a}^{T}_{K+1} \\ \hline \mathbf{a}_{K+1} & a_{K+1,K+1} \end{array}\right]^{-1} \begin{bmatrix} \mathbf{B}^{t-1}_{K\times n} \\ \mathbf{b}_{K+1} \end{bmatrix} \tag{16}$$

Where $\mathbf{W}^{t}_{K}$ is the output weight matrix for K hidden neurons, and $\mathbf{w}^{t}_{K+1}$ is the vector of output weights for new hidden neuron after learning from t^{th} sample. Please refer to the neuron growth architecture in [27] to further understand the output weights.

Parameters update strategy: The present (t^{th}) training sample helps to update the output weights of the cognitive component ($\mathbf{W}_{K} = [\mathbf{w}_{1}, \mathbf{w}_{2}, \cdots, \mathbf{w}_{K}]^{T}$) if the following condition is satisfied.

$$c^{t} == \widehat{c}^{t} \ \ \textbf{AND} \ \ E^{t} \geq \beta_{u} \tag{17}$$

Where β_{u} is the self-adaptive update threshold. If β_{u} is closer to 50% of E^{t}, then fewer samples will be used and most of the samples will be dragged to the end of the training sequence. The resultant network will not approximate the function accurately. If a lower value is chosen, then all samples will be used in updating without any change in the training sequence.

The $\mathbf{A} \in \Re^{K\times K}$ and $\mathbf{B} \in \Re^{K\times n}$ matrices updated as

$$\mathbf{A}^{t} = \mathbf{A}^{t-1} + \left(\mathbf{h}^{t}\right)^{T} \mathbf{h}^{t};\ \ \mathbf{B}^{t} = \mathbf{B}^{t-1} + \left(\mathbf{h}^{t}\right)^{T} \left(\mathbf{y}^{t}\right)^{T} \tag{18}$$

Finally the output weights are updated as

$$\mathbf{W}^{t}_{K} = \mathbf{W}^{t-1}_{K} + \left(\mathbf{A}^{t}\right)^{-1} \left(\mathbf{h}^{t}\right)^{T} \left(\mathbf{e}^{t}\right)^{T} \tag{19}$$

Where $\mathbf{e}^{t}$ is hinge loss for t^{th} sample as in Eq. (7).

Sample reserve strategy: If the new training sample does not satisfy either the sample deletion or the neuron growth or the cognitive component parameters update criterion, then the current sample is pushed to the rear end of data stream. Since McRBFN modifies the strategies based on the knowledge in the current sample, these samples may be used in later stage. The above process is repeated for every incoming samples. Next, we shall evaluate the action recognition ability of PBL-McRBFN on MHAD dataset.

III. EXPERIMENTS AND RESULTS

The proposed approach is evaluated on the widely used Multimodal Human Action Detection dataset. The dataset has various modalities like, motion capture, kinect depth map, RGB, and Accelerormeter. For our experimentation we use only the motion capture data.

A. Dataset Description

The Multi modal Human Action Dataset (MHAD) [12] is a publicly available dataset. The database has 11 actions performed by 12 subjects. All subjects perform 5 repetitions of each action yielding about 660 ($12 \times 11 \times 5$) action sequences. The actions are jump, jumping jacks, bend punch, arms above head, hand wave, clap, throw, sit down, stand up, sit down/stand up. Figure 4 illustrates the actions.

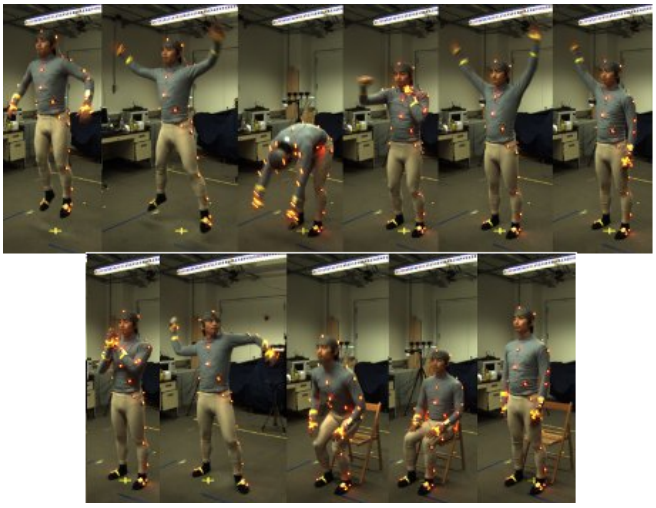

Fig. 4. *Different actions in MHAD database [12] starting from top left, jump, jumping jacks, bend, punch, arms above head, hand wave, clap, throw, sit down, stand up, sit down/stand up*

The motion capture data which is captured using the optical motion capture system Impulse. This captures active LED markers. Tight-fitting suit with 43 LED markers was used to get 3D trajectories of 43 points which represent the skeleton of the subject. We use leave one out approach for classification, wherein we use all the subjects except one for training. We achieve better results than the state of the art algorithms, which is reflected in the confusion matrix in Figure 5. The implementation of the proposed algorithm is done in MATLAB, on a 32-bit Intel (R) Core i3 CPU, $3.06GHz$ machine, with 4.0 GB RAM.

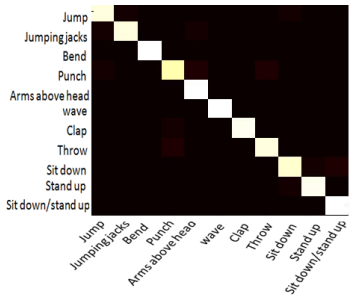

Fig. 5. *Confusion Matrix showing the Classification results using PBL classifier*

In [12], the authors have presented the results using all the 5 modalities available in the MHAD dataset. And by using the MoCap modality alone, the they have achieved an overall accuracy of 79.93%. Its reported that recognition rate of 93.8 % is achieved using Kinect and MoCap modalities. Also Ferda

ofli et al.[20] have achieved an accuracy of 94.91% using motion capture data by using 7 subjects for training and 5 for testing. Babu et al.[26], have achieved an accuracy of 96.06% by using the leave one out approach and with a feature vector of length 258. With the proposed approach we have achieved a better result than the state of the art is achieved. We report an accuracy 97.58% using the MoCap modality alone. We have used the feature vector of length 322, which are easy to compute and handle. Table 1 shows the comparision of results of the proposed approach with [16], [26] and [17].

Algorithm	Recognition accuracy
HMIJ [20]	94.91%
Kurillo et al.[12]	79.93%
Babu et al.[26]	96.06%
Proposed	**97.58%**

Table 1: Comparison of results on MHAD database

IV. CONCLUSION

In this paper, we have proposed a novel approach for human action recognition system for MoCap data, using the BoW model representation of the actions sequence along with the variance of the skeletal joints. The feature vectors are independent of the duration of the action and the starting point of the action in the given action sequence. The proposed algorithm is evaluated on the MoCap data of the Multi-modal Human Action Database using the PBL-McRBFN classifier and found to perform better than the state of the art approaches.

REFERENCES

[1] Michalis Raptis, Darko Kirovski, and Hugues Hoppe, "Real-time classification of dance gestures from skeleton animation," in *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2011, pp. 147–156.

[2] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1290–1297.

[3] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[4] M. S. Naresh Kumar and R. Venkatesh Babu, "Human gait recognition using depth camera: A covariance based approach," in *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing*, 2012, ICVGIP '12, pp. 20:1–20:6.

[5] Vennila Megavannan, Bhuvnesh Agarwal, and R. Venkatesh Babu, "Human action recognition using depth maps," in *International Conference on Signal Processing and Communications*, 2012, pp. 1–5.

[6] R. Venkatesh Babu, R. Savitha, S. Suresh, and Bhuvanesh Agarwal, "Subject independent human action recognition using spatio-depth information and meta-cognitive RBF network," *Engineering Applications of Artificial Intelligence*, vol. 26, pp. 2010–2021, 2013.

[7] Lee W Campbell and Aaron F Bobick, "Recognition of human body motion using phase space constraints," in *Fifth International Conference on Computer Vision*, 1995, pp. 624–630.

[8] James W Davis and Aaron F Bobick, "The representation and recognition of human movement using temporal templates," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 928–934.

[9] Vasu Parameswaran and Rama Chellappa, "View invariance for human action recognition," *International Journal of Computer Vision*, vol. 66, no. 1, pp. 83–101, 2006.

[10] Cen Rao, Alper Yilmaz, and Mubarak Shah, "View-invariant representation and recognition of actions," *International Journal of Computer Vision*, vol. 50, no. 2, pp. 203–226, 2002.

[11] Antonios Oikonomopoulos, Ioannis Patras, and Maja Pantic, "Spatiotemporal saliency for human action recognition," in *IEEE International Conference on Multimedia and Expo, 2005*, 2005, pp. 4–pp.

[12] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy, "Berkeley mhad: A comprehensive multimodal human action database," in *IEEE Workshop on Applications of Computer Vision (WACV)*, 2013, pp. 53–60.

[13] Xi Chen and Markus Koskela, "Classification of rgb-d and motion capture sequences using extreme learning machine," in *Image Analysis*, pp. 640–651. 2013.

[14] Saehoon Yi and Vladimir Pavlovic, "Sparse granger causality graphs for human action classification," in *International Conference on Pattern Recognition (ICPR)*, 2012, pp. 3374–3377.

[15] Piotr Dollár, Vincent Rabaud, Garrison Cottrell, and Serge Belongie, "Behavior recognition via sparse spatio-temporal features," in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*. IEEE, 2005, pp. 65–72.

[16] Ivan Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[17] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld, "Learning realistic human actions from movies," in *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR*, 2008, pp. 1–8.

[18] Saad Ali and Mubarak Shah, "Human action recognition in videos using kinematic features and multiple instance learning," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, pp. 288–303, 2010.

[19] Du Tran and Alexander Sorokin, "Human activity recognition with metric learning," in *European Conference on Computer Vision–ECCV*, pp. 548–561. 2008.

[20] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy, "Sequence of the most informative joints (smij): A new representation for human skeletal action recognition," *Journal of Visual Communication and Image Representation*, 2013.

[21] L. Narens T. O. Nelson, "Metamemory: A theoretical framework and new findings," 1992.

[22] A. L. Wnden, "Metacognitive knowledge and language learning, applied linguistics," pp. 515–537. 1998.

[23] W. P. Rivers, "Autonomy at all costs: An ethnography of metacognitive self-assessment and self-management among experiences language learner," in *The Modern language journal*, pp. 279–290. 2001.

[24] F. Fujita R. Issacson, "Metacognitive knowledge monitering and selffrequlated learning: Academic success and reflection on leaning," in *Journal of the Scholarship of Teaching and Learning 6(1)*, pp. 39–55. 2006.

[25] Josef Sivic and Andrew Zisserman, "Video google: A text retrieval approach to object matching in videos," in *International Conference on Computer Vision*, 2003, pp. 1470–1477.

[26] Suraj Vantigodi and R. Venkatesh Babu, "Human action recognition using motion capture data," in *National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*. 2013.

[27] G Sateesh Babu, S Suresh, K Uma Sangumathi, and HJ Kim, "A projection based learning meta-cognitive rbf network classifier for effective diagnosis of parkinsons disease," in *Advances in Neural Networks–ISNN*, pp. 611–620. 2012.