IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 37, No. 1, February, 2020, pp. 37-44
Journal homepage: http://iieta.org/journals/ts

Human Action Recognition in Video Sequences Using Deep Belief Networks

Mehrez Abdellaoui[1,2*], Ali Douik[1]

[1] Networked Object Control and Communication Systems Laboratory, National Engineering School of Sousse, University of Sousse, Sousse 4054, Tunisia
[2] Higher Institute of Applied Sciences and Technology of Kairouan, University of Kairouan, Kairouan 3100, Tunisia

Corresponding Author Email: mehrez.abdellaoui@enim.rnu.tn

https://doi.org/10.18280/ts.370105

Received: 8 December 2019
Accepted: 20 January 2020

Keywords:
human action recognition, deep belief network, restricted Boltzmann machine, deep learning

ABSTRACT

For the last several decades, Human Activity Recognition (HAR) has been an intriguing topic in the domain of artificial intelligence research, since it has applications in many areas, such as image and signal processing. Generally, every recognition system can be either an end-to-end system or including two phases: feature extraction and classification. In order to create an optimal HAR system that offers a better quality of classification prediction, in this paper we propose a new approach within two-phase recognition system paradigm. Probabilistic generative models, known as Deep Belief Networks (DBNs), are introduced. These DBNs comprise a series of Restricted Boltzmann Machines (RBMs) and are responsible for data reconstruction, feature construction and classification. We tested our approach on the KTH and UIUC human action datasets. The results obtained are very promising, with the recognition accuracy outperforming the recent state-of-the-art.

1. INTRODUCTION

Recognition and understanding of human actions have become a subject of broad and current interest in the field of computer vision and image processing. It has found applications in many areas, such as: video surveillance [1], human-machine interaction [2] and video indexing [3]. The aim of a HAR system is to identify the simple actions of everyday life (e.g. running, walking, jumping, etc.) from videos. Each of these actions, carried out by a single person within a specified period of time, can be represented by a simple motion model.

Modeling action methods can be divided into two categories: spatio-temporal methods and sequential methods. Spatio-temporal approaches model the human action in the form of a 3D volume in a spatio-temporal dimension or as a set of characteristics extracted from the 3D volume. Resulting volumes from an image concatenation along the time axis are compared to measure their similarities. On the other hand, sequential approaches consider an action as a sequence of particular observations. More precisely, they represent a human action as a sequence of feature vectors extracted from the images and proceed to minimize the ones that are close to each other in terms of specific distance.

Indeed, such a video-based HAR system is composed of two elementary steps: The first one is the feature extraction from the video input frames. The second is the classification, which consists of categorizing the actions from video sequences using the vector of primitives extracted in the first step.

However, video-based HAR systems that use deep learning (DL) are receiving significant attention thanks to their ability to learn deep structures. Essentially, the DL technique is an extension of artificial neural networks that makes use of hierarchically organized layers for feature classification from non-linear data processing. Techniques based on DL outperform many conventional approaches to image processing and computer vision. DL methods show much promise in satisfying the requirements of HARs in two ways: First, there is the potential to discover features that are related to human body movement. This is convenient for manipulating complex human activities for the purpose of recognition. Second, performance can be improved beyond that of traditional recognition methods.

Deep Belief Networks (DBN) is a DL technique that has been proposed by Hinton [4]. DBN uses Restricted Boltzmann Machines in learning and classification. The reduced learning time allows DBNs to avoid the local minimum problem. Thus, in this paper, we present a new DBN-based HAR system which is able to extract features from video and classify that input data. DBNs have been employed in the literature for speech recognition [5], digit recognition [6], etc.

This paper is organized as follows: we briefly present, in section 2, some HAR techniques. In section 3, we describe our proposed approach and its details. Our experimental set up, performance analysis and discussions are detailed in section 4. Finally, a conclusion and future works are given is section 5.

2. RELATED WORKS

Many methods of HAR are proposed in the literature. Each one employs a specific technique for the two previously mentioned steps. In this context, we organize this section into two parts: in the first one, we present works that use handcrafted features. In their 2008 study, Laptev et al. [7] report a method for video classification that builds upon local space-time feature extraction and multichannel non-linear SVMs for classification. This method was applied to the standard KTH human action dataset and achieved a 91.8% accuracy rate. Klaser et al. [8] introduced a local descriptor for video sequences. The proposed descriptor is based upon histograms of oriented 3D spatio-temporal gradients. In order to compute 3D gradients for arbitrary scales, the authors first developed a memory efficient algorithm that relies upon integral videos. Then, they proposed a generic 3D orientation quantization which is based upon regular polyhedrons. Finally, they performed an in-depth evaluation of all descriptor parameters and optimized them for HAR. Their descriptor was applied to various human action datasets: ((KTH 91.4%), Weizmann (84.3%) and Hollywood (24.7%)).

Huang et al. [9] used a Histogram of Oriented Gradients feature to address HAR. This feature was applied to a Spatio-Temporal Interested Points, detected by Harris 3D on a Motion History Image. The classification step was developed using Artificial Neural Networks. The authors chose the KTH dataset for the training data and the MSR action dataset II for the testing data, obtaining 64.06% accuracy rate.

In the second part, we present works that employ deep learning technique. Ali and Wang [10] have proposed a human action modeling method based upon a two-dimensional wavelet and watermark embedding. The authors made use of DBN and the Discrete Cosine Transform technique for data learning and feature extraction. They employed the SVM classifier in the classification step. The authors tested their approach on the KTH dataset and obtained an accuracy rate exceeding 94.3%. Zhang et al. [11] presented a modified DBN model, which is composed of a conditional RBM to recognize human interactions in real-time. Conditional RBMs generate temporal information from human actions by determining joint positions. The authors demonstrated the robustness of their approach, as they achieved recognition accuracy on the MIT and MSR Action 3D datasets of 98.08% and 98.88%, respectively. Geng and Song [12] proposed a human recognition method based upon Convolutional Neural Networks (CNN), where a pre-training strategy making use of a convolutional auto-encoder has been introduced to reduce the high computational cost of training that has been provided by the CNN model. In the classification step, the authors used the SVM classifier to achieve a recognition rate of 92.49 % on the KTH dataset.

Recent studies have shown how CNN models could be applied to HAR. Ijjina and Chalavadi [13] introduced a genetic algorithm to provide an optimal initiation of CNN ponderations as the training step. A gradient descent algorithm was used to train the CNN classifier on the UCF50 dataset. The authors demonstrated the efficiency of their approach with recognition accuracy equal to 99.98%. Ji et al. [14] proposed a 3D CNN architecture to create a HAR system. The 3D CNN model detects a set of features from both spatial and temporal dimensions by carrying out 3D convolutions, thereby capturing and encoding the motion information in multiple adjacent frames. The authors developed a model which generates multiple channels of information from the input frames, and the final feature representation is achieved by combining information from all channels.

Ke et al. [15] have also proposed deep neural network architecture. The authors introduce a new approach for 3D HAR with skeleton sequences derived from 3D trajectories of human skeleton joints. They have suggested a method which comprises several stages. First, each skeleton sequence is transformed into three video clips. Each clip consists of several frames for spatial and temporal feature learning employing CNN. Specifically, for each channel of the 3D coordinates, the authors transformed the sequence into a video clip using grayscale images, which show spatial and structural information of the articulations. Those frames are provided for deep CNN to learn high-level features. Next, the CNN features of the three clips at the same time-step are concatenated into a feature vector. Each one represents the temporal information of the entire skeleton sequence and the particular spatial relationship of the joints. In the last stage, the authors use a Multi-Task Learning Network to jointly process the feature vectors of all time-steps in parallel for action recognition.

Finally, Uddin and Zia [16] have proposed a DBN-based HAR model using the 3D Body Joint Motion Features where recognition of human actions depends upon the magnitude and direction of body joints extracted from depth videos. A human body silhouette is determined from the coordinates of each articulation introduced in the input frames. This proposed HAR method demonstrated superior performance with the MSRC-12 dataset (97.93%), the MSRDailyActivity3D dataset (91.56%), and a specific dataset containing six human actions (96.97%).

3. PROPOSED APPROACH

The proposed approach aims to improve the accuracy of human activity classification by employing a new DBN-based HAR method. As a first step, we segment the video sequences from the human activity dataset into frames. Next, we convert the result into binary frames and we carry out a set of morphological filtering operations on the new frames in order to enhance their quality. Following this, we transform the new frames into a binary vector in order to create an input matrix that contains the training data and the testing data, as well as their labels. This matrix represents the input data for our DBN architecture, as shown in Figure 1. In the final step, we train the DBN classifier with the training data matrix and extract the classification result.

For the binarization step, we utilized two techniques, depending on how the background lighting compared to the object in the frame. The thresholding algorithm was employed where the object was clearer compared to the background, or vice versa. On the other hand, the background detection algorithm was used for the frames that are characterized by similar degrees of illumination for the object and the background. Then, we applied some morphological filters such as erosion, dilation, etc. for eliminating binary frame noise as well as finishing. From each binary frame, a binary vector was created such that the number of vector columns is the product of the number of columns and the number of lines (i.e. the original frame size). As each binary vector takes a line in the matrix, this yields a matrix in which each line represents a binary frame (see Figure 1).

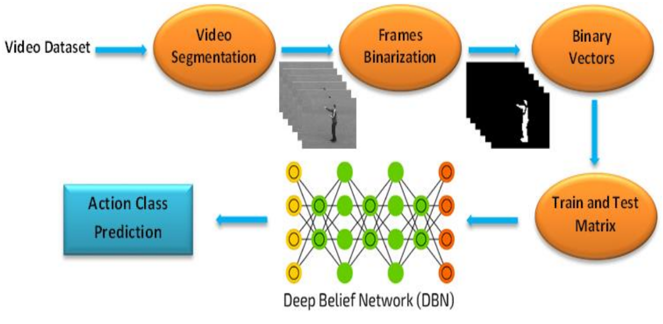


Figure 1. Framework of the proposed method

3.1 Deep belief networks

Recently, a large amount of research has been carried out on the subject of deep learning. The aim of this research is to determine the means by which unsupervised learning can be used to represent input data in more abstract ways. Input data represented in such ways can be applied to numerous tasks, including classification and regression. Standard neural networks were intended to learn those deep representations. However, deep neural networks (i.e. neural networks that have many hidden layers) are difficult to train using gradient descent [17]. The DBN solved this problem by adding an unsupervised pre-training phase to a greedy layer. This unsupervised pre-training builds a representation from which it is possible to do successful supervised learning by fine-tuning the resulting weights using gradient descent learning [5, 17]. Stated differently, the unsupervised stage sets the network weights closer to a good solution than random initialization, avoiding local minima when we use a supervised gradient descent. The DBN is an ANN architecture that is used to create generative graphics models, which was first introduced by Hinton [1]. The DBN is a type of deep neural network composed of multiple layers of stochastic and latent variables which are connected to each other, but not between units within each layer (see Figure 2). The DBN model can generate probabilistic, binary and Gaussian data. The first of the DBN's two parts includes reconstruction layers, which are responsible for converting the input data into an abstract representation. The second part is formed by layers that transform this abstract representation into classification labels for the purpose of class prediction.

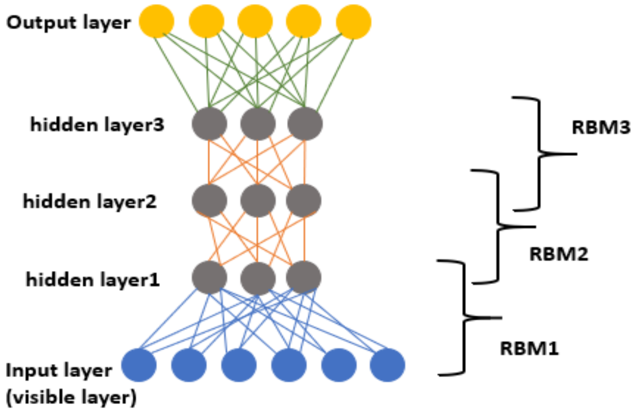


Figure 2. Schematic representation of a DBN model

The DBN model can be considered, from Hinton's perspective [4], as an amalgam of simple learning modules, each of which is an RBM. The RBM contains a layer of visible units that represents the data input and a layer of hidden units representing features that capture higher-order correlations in the data. This also leads to a fast training procedure, an unsupervised layer-by-layer, where contrastive divergence is applied to each sub-network in turn, starting from the lowest pair of layers (the lowest visible layer being a training set). The most important property of DBNs is the layer-by-layer technique for learning the top-down, generative weights that determines how the variables in a single layer depend on the variables in the higher layer. The crucial idea behind DBNs is that their weights (*W*), learned by a RBM, define the prior distribution over visible vectors and ponderations *p(v|h,W)* as well as the prior distribution over hidden vectors *p(h|W)*. The probability of generating a visible vector can be written as:

$$p(v) = \sum_h p(h|W) \cdot p(v|h,W) \qquad (1)$$

After determining *W*, where *W* is the matrix of symmetrically weighted connections between the visible layer and the hidden layer, we preserve *p(v|h, W)* and replace *p(h|W)* with a better model of the aggregated posterior distribution over hidden vectors (i.e. the non-factorial distribution produced by averaging the factorial posterior distributions produced by the individual data vectors). A better model is obtained by treating the hidden activity vectors produced from the training data as the training data for the next learning module. The values of the latent variables in each layer, after learning, can be deduced by a single bottom-to-top pass that begins with a data vector observed in the lower layer and uses the generative weights in the reverse direction. DBNs are competitive with respect to five essential points:

- They can be fine-tuned as neural networks.
- They have diverse non-linear hidden layers.
- They are generatively pre-trained.
- They can act as a non-linear dimensionality reduction method for input features vectors.
- The network teacher is another sensory input.

DBNs exist in two possible forms: the auto-encoder DBN and the classifier DBN.

An auto-encoder DBN is a simple three-layer neural network where the output units are directly connected to the input units. Generally, the number of hidden units is much less than the number of visible units. The auto-encoding process is divided into two steps: the encoding (compression) of an input vector to adjust in a smaller representation, and the reconstruction (decoding). The training task consists of minimizing an error or a reconstruction (i.e. finding the most efficient compact representation for the input data). A DBN auto-encoder [18] is a model comprising auto-encoder RBMs that permit creation of a generative model for extracting features from the encrypted data. Usually, the data vector is saved in the last hidden layer. In addition, the auto-encoders are a general class of algorithms used to compress representations of input data.

A classifier DBN, which we are interested in this work, is used in learning and supervised classification. The recognition process takes advantage of the architecture of the latter to give exact classification results, such that the first layer of the DBN, which is the visible layer, represents the input data vector, the hidden layers show the primitive detectors, or reconstructors, from the visible layer data, and the last layer of the DBN is the SoftMax layer, containing the classification labels. As such, the classifier DBN architecture requires that the last RBM be discriminative to ensure that the output data is labeled correctly.

The choice of DBN architecture is an important factor in ensuring robust HAR systems. As such, our DBN-based HAR method utilises an architecture that comprises a classifier DBN, which, in turn, is composed of three RBMs, two generative RBMs for the training stage and feature extraction as well as a discriminative RBM to classify the data input vector. The last layer is a SoftMax regression which is used for obtaining the output from the hidden layer. The structure of the DBN is shown in Figure 3. The entire process of DBN-based HAR consists of three major steps: training, fine-tuning and classification. For the DBN training, a bottom-up layer-wise unsupervised learning method is applied for feature learning, such as the raw data stream flows from the visible layer to the H3 hidden layer in layer-by-layer process. Then, a coarse network is built. Following this, a top-down supervised learning method, a contrastive version of Wake-Sleep, is utilised to fine-tune weights for the whole network. Finally, we use the SoftMax regression to classify an input normalized trajectory image and output the result. SoftMax regression is a generalization of logistic regression that is used for multi-class classification. In contrast, logistic regression can only be employed in binary classification.

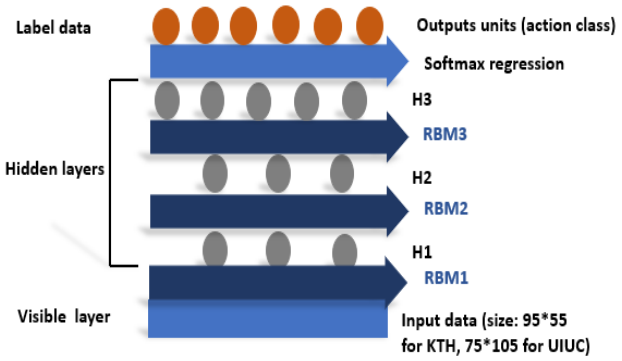


Figure 3. Schematic representation of a DBN model

3.2 Restricted Boltzmann machine

The RBMs are stochastic and probabilistic neural networks that can be used in unsupervised modes. According to Hinton [19], they are useful algorithms for size reduction, classification, regression, collaborative filtering and feature-based learning. The RBMs are elementary sub-networks comprising the DBM which are composed of two layers: a visible layer and a hidden layer with binary units. The two layers are connected and there are no connections within a layer. RBMs can be used to find the inherent relation between the binary data, whose energy function is defined as follows:

$$E(v,h;\theta) = -\sum_{i=1}^{I}\sum_{j=1}^{J}\left(w_{ij}v_i h_j\right) - \sum_{i=1}^{I}\left(a_i v_i\right) - \sum_{i=1}^{I}\left(b_j h_j\right) \qquad (2)$$

where, ω_{ij} is the connection weight between hidden unit h_j and visible unit v_i, a_i and b_j are the bias terms. I and J are the numbers of visible and hidden units, respectively. θ is the model parameter. The joint probability distributions over visible and hidden units are defined in terms of the energy function:

$$p(v,h;\theta) = \frac{exp\left(-E\left(v,h;\theta\right)\right)}{Z} \qquad (3)$$

where, $Z = Z\left(\theta\right) = \sum_{h,v} exp\left(E\left(v,h;\theta\right)\right)$ is a partition function. The marginal probability of a visible unit of Booleans is the sum over all possible hidden layer configurations:

$$p(v;\theta) = \sum_{h}\frac{exp\left(-E\left(v,h;\theta\right)\right)}{Z} \qquad (4)$$

Since the RBM has the shape of a bipartite graph. The visible units and the hidden units are mutually independent, and there are no connections between the same layer units. Hence, the conditional probability can be written as:

$$p(h_j = 1 \mid v;\theta) = \delta\left(\sum_{i=1}^{I}\omega_{ij}v_i + b_j\right) \qquad (5)$$

$$p(v_i = 1 \mid h;\theta) = \delta\left(\sum_{j=1}^{J}\omega_{ij}h_j + a_i\right) \qquad (6)$$

where, $\delta(x) = 1\big/\left(1 + exp\left(-x\right)\right)$. The weights update required to perform gradient descent in the log-likelihood can be obtained as follows:

$$\Delta\omega_{ij} = \varepsilon\left(\left(v_i h_j\right)_{data} - \left(v_i h_j\right)_{mod\ el}\right) \qquad (7)$$

where, ε is the learning rate in both training set and model. The learning rule for polarization parameters is given by:

$$\Delta a_i = \varepsilon\left(\left(v_i\right)_{data} - \left(v_i\right)_{mod\ el}\right) \qquad (8)$$

$$\Delta b_j = \varepsilon\left(\left(h_j\right)_{data} - \left(h_j\right)_{mod\ el}\right) \qquad (9)$$

4. EXPERIMENTS

To evaluate the proposed approach properly, we applied it on two different datasets. Different classification and evaluation criteria were used. And the results obtained were compared and discussed taking into account the recent introduced techniques.

4.1 Datasets

4.1.1 KTH dataset

KTH dataset [20] is the most commonly used dataset in HAR systems. It contains six human actions: boxing, handclapping, handwaving, jogging, running and walking. These actions are performed by 25 people under four different scenarios (outside, outside with variation of scale, outside with different clothes and inside). As such, in total, the KTH database encompasses 600 video sequences shot on homogeneous backgrounds with a static camera. These sequences are stored in AVI format and they have been reduced to a spatial resolution of 160×120 pixels for each frame. Figure 4 shows an example of images detected from the KTH dataset. Our proposed method makes use of binary data for the DBN architecture. Preprocessing consists of converting the input data into binary data. We start by segmenting each video sequence into grayscale frames. Then, we transform each frame into a binary one. Following this, we apply a morphological filter to each frame in order to enhance its quality. The size of all frames is then standardized to 95×55 pixels. In this way, we change each frame into a binary vector so as to create an input matrix that contains the training data and the testing data, as well as the labels.

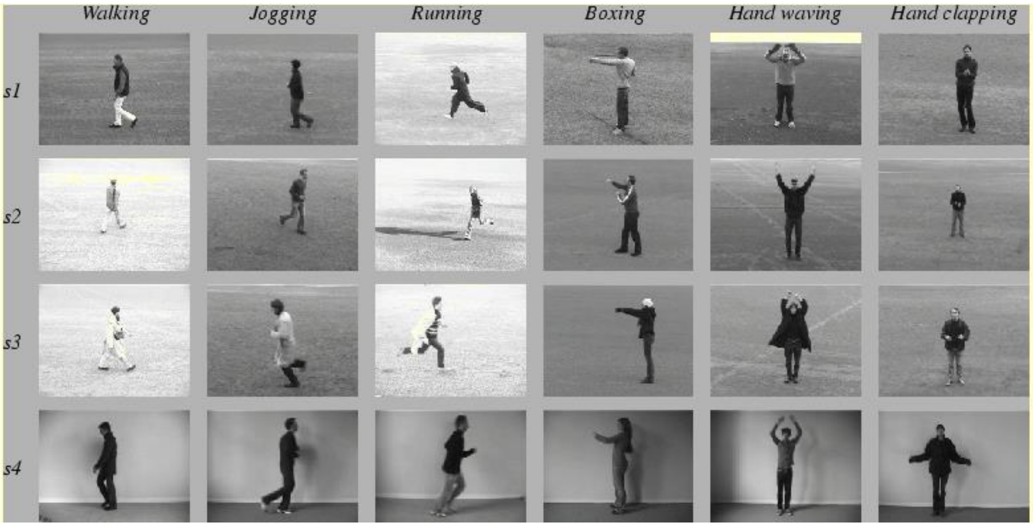


Figure 4. Image samples from KTH dataset

4.1.2 UIUC dataset

UIUC dataset [21] contains fourteen human actions: walking, running, waving, raise-one-hand, jumping, jumping jacks, clapping, jump-from-situp, stretching out, turning, sitting-to-standing, crawling, standing-to-sitting and pushing up. These actions are distributed across three sections: the first contains 271 image sequences, the second consists of 191 image sequences and the third includes 70 image sequences. Figure 5 shows image samples from the UIUC dataset.

Figure 5. Image samples from UIUC dataset

This dataset allows us to avoid the first preprocessing steps, such as segmentation of video sequences and binarization, because the foreground masks have already been prepared. To improve the quality of the foreground masks, we apply a morphological filter and fix the size of all images at 75×105 pixels.

4.2 Tests and results

We tested our proposed method on Matlab R2016b and used the DeeBNET Toolbox [18]. All experiments were run on a 64-bit Windows 10 PC with an Intel Core i7- 6500 CPU, 8 GB of RAM and an NVIDIA GeForce 920MX (2GB) graphics card. For the KTH dataset, we used 18 video sequences from different scenarios for the training data and 12 video sequences from different scenarios for the testing data. However, for the UIUC dataset, we tested the DBN architecture with 10 human actions (running, waving, walking, clapping, jumping, jumping jacks, stretching out, turning, raise-one-hand and sitting-to-standing). Hence, we used 30 image sequences for the training data and 20 image sequences for the testing data.

4.2.1 Training parameters

For each dataset, a DBN model was built. In Table 1, the training parameters for each dataset are shown. We choose the 5% most relevant frames from the training set. 7409 frames for the KTH dataset and 2228 for UIUC dataset.

For the test step, we randomly chose 10% from the test data corresponding to 70180 frames for the KTH dataset and 26400 for UIUC dataset. In fact, for the KTH dataset we tested 7018 frames and 2640 for the UIUC dataset. These tests were conducted 5 times randomly to cross-correlate the results obtained, leading to an average accuracy varying with ± 0.03%.

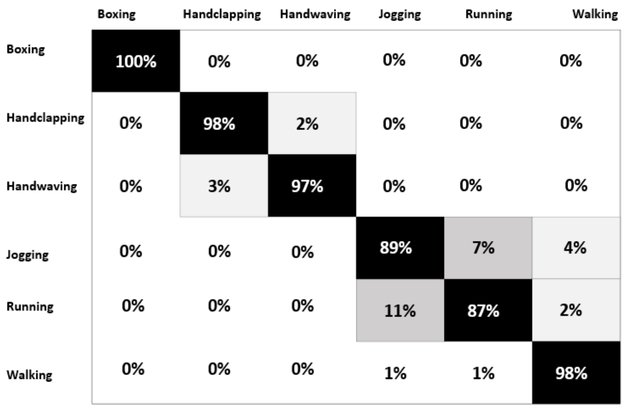
Table 1. Training parameters of the DBN model for each dataset

Training parameter	KTH	UIUC
Number of supervised epochs	30	30
Number of visible layers	1	1
Number of units in visible layer	5225	7875
Number of hidden layers	3	3
Output units	6	10
Supervised learning rate	0.01	0.1
Number of iterations	2200	2500

4.2.2 Classification results

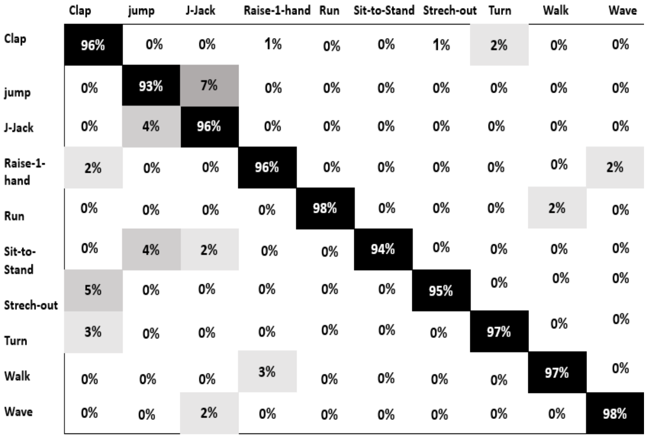
The confusion matrices in Figures 6 and 7 provide the classification results of the proposed method for the KTH and UIUC datasets respectively. The diagonal values depict the correctly predicted samples and the off-diagonal values represent the miss-classified samples. After applying the back-propagation algorithm, we found that our approach resulted in fewer errors and offered higher accuracy for all of the human action classes in both datasets. On average, our DBN-based HAR method reached a recognition rate of 94.83% for KTH and 96% for UIUC.

Analysis of the confusion matrices revealed that the classification rates are high for all of the classes, except for jogging and running in KTH dataset. These two human actions exhibit similarity when examining single frames from the video. To better differentiate between them, we would have to take motion into consideration, which cannot be considered by the proposed DBN-based HAR. For each class, we computed the evaluation metrics True positive (*Tp*), True negative (*Tn*), False positive (*Fp*) and False negative (*Fn*).



	Boxing	Handclapping	Handwaving	Jogging	Running	Walking
Boxing	100%	0%	0%	0%	0%	0%
Handclapping	0%	98%	2%	0%	0%	0%
Handwaving	0%	3%	97%	0%	0%	0%
Jogging	0%	0%	0%	89%	7%	4%
Running	0%	0%	0%	11%	87%	2%
Walking	0%	0%	0%	1%	1%	98%

Figure 6. Confusion matrix of the proposed method on the KTH dataset



	Clap	jump	J-Jack	Raise-1-hand	Run	Sit-to-Stand	Strech-out	Turn	Walk	Wave
Clap	96%	0%	0%	1%	0%	0%	1%	2%	0%	0%
jump	0%	93%	7%	0%	0%	0%	0%	0%	0%	0%
J-Jack	0%	4%	96%	0%	0%	0%	0%	0%	0%	0%
Raise-1-hand	2%	0%	0%	96%	0%	0%	0%	0%	0%	2%
Run	0%	0%	0%	0%	98%	0%	0%	0%	2%	0%
Sit-to-Stand	0%	4%	2%	0%	0%	94%	0%	0%	0%	0%
Strech-out	5%	0%	0%	0%	0%	0%	95%	0%	0%	0%
Turn	3%	0%	0%	0%	0%	0%	0%	97%	0%	0%
Walk	0%	0%	0%	3%	0%	0%	0%	0%	97%	0%
Wave	0%	0%	2%	0%	0%	0%	0%	0%	0%	98%

Figure 7. Confusion matrix of the proposed method on the UIUC dataset

The performance of the HAR system has been measured using the following evaluation metrics: Accuracy (*Acc*), Precision (*Pr*), Sensitivity (*Sn*) and Specificity (*Sp*) rates are deducted according to the following equations:

$$Acc = \frac{Tp + Tn}{Tp + Tn + Fp + Fn} \tag{10}$$

$$Pr = \frac{Tp}{Tp + Fp} \tag{11}$$

$$Sn = \frac{Tp}{Tp + Fn} \tag{12}$$

$$Sp = \frac{Tn}{Tn + Fp} \tag{13}$$

Table 2. Classification results for KTH dataset

Class	TP	TN	FP	FN	Pr (%)	Sn (%)	Sp (%)	Acc (%)
Boxing	1	4.69	0	0	100	100	100	100
Handclapping	0.98	4.71	0.03	0.02	97.03	98	99.37	99.13
Handwaving	0.97	4.72	0.02	0.03	97.97	97	99.58	99.13
Jogging	0.89	4.8	0.12	0.11	88.11	89	97.56	96.11
Running	0.87	4.82	0.08	0.13	91.57	87	98.38	96.44
Walking	0.98	4.71	0.06	0.02	94.23	98	98.74	98.61
Total	**5.69**	**5.69**	**0.31**	**0.31**	**94.83**	**94.83**	**94.83**	**94.83**

Table 3. Classification results for UIUC dataset

Class	TP	TN	FP	FN	Pr (%)	Sn (%)	Sp (%)	Acc (%)
Clapping	0.96	8.64	0.01	0.04	90.56	96	98.85	98.56
Jump	0.93	8.67	0.08	0.07	92.08	93	99.08	98.64
Jump-Jack	0.96	8.64	0.09	0.04	91.42	96	98.97	98.66
Raise-1-Hand	0.96	8.64	0.03	0.04	96.96	96	99.65	99.27
Running	0.98	8.62	0.03	0.02	97.03	98	99.65	99.48
Sitting-to-standing	0.94	8.66	0	0.06	100	94	100	99.37
Streching-out	0.95	8.65	0.01	0.05	98.99	95	99.88	99.37
Turning	0.97	8.63	0.02	0.03	97.97	97	99.77	99.48
Walking	0.97	8.63	0.02	0.03	97.97	97	99.77	99.48
Waving	0.98	8.62	0.02	0.02	98	98	99.77	99.58
Total	**9.6**	**9.6**	**0.4**	**0.4**	**96**	**96**	**96**	**96**

Tables 2 and 3 above show the obtained performances for KTH and UIUC datasets respectively.

4.3 Comparison with existing deep learning methods

To evaluate our proposed method, we compared the obtained classification results with several methods of the state-of-the-art using deep learning methods and using the same datasets. Considering the KTH, our method with an average accuracy equal to 94.83% outperforms Ali and Wang method [10] (94.3%), Geng and Song method [12] (92.49%), Baccouche et al. method [22] (91.04%) and Ji et al. method [14] (90.2%). For UIUC dataset, with an average accuracy equal 96%, we obtained also better results compared to Chalamala and Kumar method [23] (80%).

Table 4 summarizes the classification accuracy of our method and the recent methods from the state-of-art using deep learning.

Table 4. Performance comparison of the proposed method with recent deep learning methods

Method		Acc (%)	
		KTH	UIUC
Proposed method		**94.83**	**96**
Ali and Wang	[10]	94.30	—
Geng and Song	[12]	92.49	—
Ji et al.	[14]	91.04	—
Baccouche et al.	[22]	90.20	—
Chalamala and Kumar	[23]	—	80

4.4 Computational complexity

The creation of a robust and efficient recognition system takes two main considerations into account. The first is the quality of classification and identification (i.e. the correct prediction rate). The second is the execution time, with the aim of carrying out the classification in real-time. The method introduced in this paper uses a DBN classifier with a back-propagation algorithm, which can be parallelized. We employed a Deep Learn Toolbox GPU-based DBN implementation to train our DBN classifier, which led to a reduction in training time. We executed various tests with different numbers of iterations and different numbers of epochs. We observed optimal results at 30 epochs for the KTH dataset and 20 epochs for the UIUC dataset. The DBN classifiers rapid training period reduces its computational complexity and, thus, allows it to be used in HAR.

5. CONCLUSION AND FUTURE WORKS

In this paper, we proposed a new method for human action recognition using a deep learning technique inspired from a DBN classifier model. The proposed DBN architecture is composed of three RBMs, two of them are generative and are used for the training stage and feature extraction. The third one is a discriminative RBM to classify the data input vector. The last layer is a SoftMax regression which is used to obtain the output from the hidden layer.

To assess the proposed method, we performed experiments for human action recognition on two popular and challenging datasets: KTH and UIUC. Indeed, the training of the network was performed on 5% of the most relevant frames from each dataset. While for the test step, we randomly chose 10% from the test data. The experimental results prove that the proposed method outperforms the recent state-of-the-art HAR methods. In fact, the obtained accuracy reached around 95% and 96% for KTH and UIUC datasets respectively.

Future works will focus on two important points: The implementation of a DBN model with unsupervised classification data, and the inclusion of the motion capture data to highlight the temporal information in HAR systems by considering the movement tracking task.

REFERENCES

[1] Ullah, A., Muhammad, K., Haq, I.U., Baik, S.W. (2019). Action recognition using optimized deep autoencoder and CNN for surveillance data streams of non-stationary environments. Future Generation Computer Systems, 96: 386-397. http://dx.doi.org/10.1016/j.future.2019.01.029

[2] Perez, M., Liu, J., Kot, A.C. (2019). Interaction recognition through body parts relation reasoning. Springer Asian Conference on Pattern Recognition (ACPR), Auckland, pp. 1-12.

[3] Majd, M., Safabakhsh, R. (2019). Correlational Convolutional LSTM for human action recognition. Neurocomputing. https://doi.org/10.1016/j.neucom.2018.10.095

[4] Hinton, G.E. (2009). Deep belief networks. Scholarpedia, 4(5): 5947. http://dx.doi.org/10.4249/scholarpedia.5947

[5] Huang, Y., Tian, K., Wu, A., Zhang, G. (2019). Feature fusion methods research based on deep belief networks for speech emotion recognition under noise condition. Journal of Ambient Intelligence and Humanized Computing, 10(5): 1787-1798. http://dx.doi.org/10.1007/s12652-017-0644-8

[6] Hinton, G.E., Osindero, S., Teh, Y.W. (2006). A fast learning algorithm for deep belief nets. Neural Computation, 18(7): 1527-1554. http://dx.doi.org/10.1162/neco.2006.18.7.1527

[7] Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B. (2008). Learning realistic human actions from movies. IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, pp. 1-8. http://dx.doi.org/10.1109/CVPR.2008.4587756

[8] Klaser, A., Marszalek M., Schmid, C. (2008). A spatio-temporal descriptor based on 3D-gradients. in M. Everingham, C. Needham and R. Fraile (eds.), BMVC 2008 - 19th British Machine Vision Conference, Leeds, United Kingdom, pp. 1-10. http://dx.doi.org/10.5244/C.22.99

[9] Huang, Y., Yang, H., Huang, P. (2012). Action recognition using hog feature in different resolution video sequences. International Conference on Computer Distributed Control and Intelligent Environmental Monitoring, Zhangjiajie, Hunan, China, pp. 85-88. http://dx.doi.org/10.1109/CDCIEM.2012.27

[10] Ali, K.H., Wang, T. (2014). Learning features for action recognition and identity with deep belief networks. International Conference on Audio, Language and Image Processing, Shanghai, China, pp. 129-132. http://dx.doi.org/10.1109/ICALIP.2014.7009771

[11] Zhang, H., Zhou, F., Zhang, W., Yuan, X., Chen, Z. (2014). Real-time action recognition based on a modified deep belief network model. IEEE International Conference on Information and Automation, Tianjin, China, pp. 225-228. http://dx.doi.org/10.1109/ICInfA.2014.6932657

[12] Geng, C., Song, J. (2016). Human action recognition based on convolutional neural networks with a convolutional auto-encoder. 5th International Conference on Computer Sciences and Automation Engineering, Sanya, China, pp. 933-938. http://dx.doi.org/10.2991/iccsae-15.2016.173

[13] Ijjina, E.P., Chalavadi, K.M. (2016). Human action recognition using genetic algorithms and convolutional neural networks. Pattern Recognition, 59: 199-212. http://dx.doi.org/10.1016/j.patcog.2016.01.012

[14] Ji, S., Xu, W., 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. http://dx.doi.org/10.1109/TPAMI.2012.59

[15] Ke, Q., Bennamoun, M., An, S., Sohel, F., Boussaid, F. (2017). A new representation of skeleton sequences for 3d action recognition. IEEE conference on computer vision and pattern recognition, Hawaï, pp. 3288-3297. http://dx.doi.org/10.1109/CVPR.2017.486

[16] Uddin, M., Kim, J. (2017). A robust approach for human activity recognition using 3-D body joint motion features with deep belief network. KSII Transactions on Internet & Information Systems, 11(2): 1118-1133. https://doi.org/10.3837/tiis.2017.02.028

[17] Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H. (2007). Greedy layer-wise training of deep networks. Neural Information Processing Systems, Vancouver, pp. 153-160.

[18] Keyvanrad M.A., Homayounpour, M.M. (2016). Deebnet, a new object oriented MATLAB toolbox for deep belief networks. http://ceit.aut.ac.ir/~keyvanrad/DeeBNet%20Toolbox.html, accessed on Feb. 20th, 2020.

[19] Hinton, G.E. (2007). Boltzmann machine. Scholarpedia 2(5): 1668. http://dx.doi.org/10.4249/scholarpedia.1668

[20] Schuldt, C., Laptev, I., Caputo, B. (2004). Recognizing human actions: a local SVM approach. 17th International Conference on Pattern Recognition, Cambridge, UK, 3: 32-36. http://dx.doi.org/10.1109/ICPR.2004.1334462

[21] Tran, D., Sorokin, A. (2008). Human activity recognition with metric learning. European Conference on Computer Vision, Marseille, pp. 548-561. http://dx.doi.org/10.1007/978-3-540-88682-2_42

[22] Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., Baskurt, A. (2011). Sequential deep learning for human action recognition. International Workshop on Human

Behavior Understanding, Berlin, Heidelberg, pp. 29-39. http://dx.doi.org/10.1007/978-3-642-25446-8_4

[23] Chalamala, S.R., Kumar, P. (2016). A probabilistic approach for human action recognition using motion trajectories. 7th International Conference on Intelligent Systems, Modelling and Simulation, Bangkok, Thailand, pp. 185-190. http://dx.doi.org/10.1109/ISMS.2016.39