Exploiting the deep learning paradigm for recognizing human actions

Pasquale Foggia, Alessia Saggese, Nicola Strisciuglio and Mario Vento
Dept. of Computer Eng. and Electrical Eng. and and Applied Mathematics
University of Salerno
Via Giovanni Paolo II, 132, Fisciano (SA), Italy
{pfoggia, asaggese, nstrisciuglio, mvento}@unisa.it

Abstract

In this paper we propose a novel method for recognizing human actions by exploiting a multi-layer representation based on a deep learning based architecture. A first level feature vector is extracted and then a high level representation is obtained by taking advantage of a Deep Belief Network trained using a Restricted Boltzmann Machine. The classification is finally performed by a feed-forward neural network. The main advantage behind the proposed approach lies in the fact that the high level representation is automatically built by the system exploiting the regularities in the dataset; given a suitably large dataset, it can be expected that such a representation can outperform a hand-design description scheme. The proposed approach has been tested on two standard datasets and the achieved results, compared with state of the art algorithms, confirm its effectiveness.

1. Introduction

Human behavior analysis is becoming in the last decade a more and more attractive research topic; this interest is mainly due to the high number of application fields that may take advantage of such technology, ranging from ambient assisted living and video surveillance to business intelligence. Human behavior can be analyzed at different layers; although an universally accepted definition is not yet available, it is common to identify the following three layers: gestures, actions and activities. Gesture recognition focuses on small body parts (usually a single hand) and aims at enabling humans to communicate with a machine by naturally interacting without any mechanical devices. Actions are composed of multiple gestures temporally organized, such as drinking, eating or dancing. Finally, activity recognition involves interactions between humans or between humans and objects; a fighting between two persons or a person leaving a bag are two examples of activities [1, 6]. In this paper we will focus on action recognition. A number of surveys on this topic have been recently proposed [21, 24], which confirms the great interest of the scientific community in this field, although a definitive solution has not been found yet. Most proposed methods in the literature focus on the choice of a set of low-level features for representing the observed scene. For instance, in [9] the spatio-temporal volume is built by exploiting a 2D Gaussian filter and a 1D Gabor filter for dealing respectively with spatial and temporal dimensions. In [8] the *Radon* transform is employed, while in [25] its extended version, namely the $\mathfrak{R}$ transform, is exploited to guarantee scale and translation invariance.

One of the main shortcomings of the above mentioned approaches lies in the fact that the feature design is a very complex task. Indeed, it is necessary to find the right balance between a representation that preserves most of the details of the scene, with the cost of very large feature vectors that make difficult the training of a classifier, and a more abstract and compact representation, which entails the risk of loosing some discriminant ability. The solutions normally found (usually with a great effort) are often effective only for a narrow class of activities, and it is difficult to develop a system that can be easily extended to other kinds of actions [4]. A first attempt in this direction has been recently made by methods based on a *bag of visual words* approach [10, 17]: the main idea is to use the first-level feature vector to recognize small elements of an action called visual words; then the histogram of the occurrences of such visual words is used as a high-level feature vector to perform the classification of the action. The set of the visual words is defined by constructing a codebook using an unsupervised learning approach. In [5] the high-level representation is built by mapping visual words into a string and by using a kernel based approach for evaluating the similarity between actions. However, as shown in [4], the introduction of more than two layers of non linear operations (like in neural networks) for representing data, namely a *deep representation*, can potentially yield a significant improvement in the overall performance of the recognition systems. This is due to

the fact that architectures with insufficient depth are very limited: in fact, as proved in [3], the functions that can be represented by a k-depth architecture might require an exponential number of computational elements to be represented by a $(k-1)$-depth architecture. Considering that the number of computational elements strongly depends on the number of training examples available for tuning the network, an insufficiently deep architecture might bring to poor generalization capabilities. This is mainly why in the last years algorithms based on deep learning approaches achieved resounding success in the community of computer vision and in the field of action recognition: in [2] and [14], for instance, a 3D convolutional neural network has been used to learn spatio-temporal features directly from the raw images; in [15] and [13] Independent Subspace Analysis algorithm is combined with deep learning techniques such as stacking and convolution.

All the above mentioned approaches are autonomously able to learn features directly from the raw images: the main limitation lies in the fact that entire images need to be used for training the network, so determining a very high computational cost which makes often unfeasible the usage of such approaches in real applications. In this paper we propose a novel strategy based on deep learning: differently from other methods, a first level feature vector is extracted by using a set of features derived from depth images; from these vectors, a high level representation, is obtained by means of a Deep Belief Network trained using a Restricted Boltzmann Machine [23]. The experimental evaluation, carried out over two widely adopted datasets, confirms that the performance of the proposed approach is very promising, although working on a smaller amount of data if compared with traditional deep learning based approaches.

2. The proposed method

An overview of the proposed method is shown in Figure 1: the sequence of depth images is acquired by a Kinect sensor and is processed. Then, a set of well-known features (v) is extracted, in order to capture both spatial and temporal information. Finally, a higher level representation is obtained by means of a pyramidal l-layers Deep Belief Network (DBN), obtained by properly combining several Restricted Boltzmann Machines (RBM): the rationale behind this choice lies in the fact that the representational power of an RBM would be too limited and that more capacity could be achieved by having more hidden layers. Furthermore, the choice of starting from a feature vector instead of from the raw image data significantly reduces the number of hidden units required for properly representing the actions, and consequently the time needed for the elaboration. Note that the part of the image containing the person is composed by about 200×100 pixels, that would result in a first level vector made of 20,000 elements. The classification

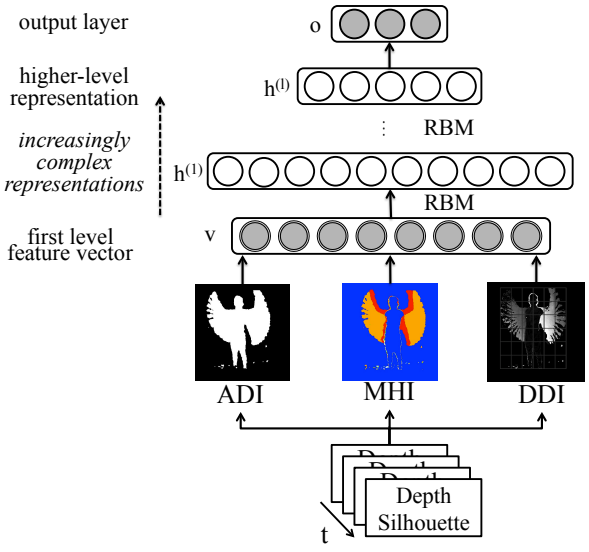

Figure 1: An overview of the proposed method. Depth images are analyzed and three derived images are extracted before computing the first level feature vector v. A more abstract representation is obtained into an unsupervised way by exploiting a DBN trained using a RBM. The classification is performed by a feed-forward neural network.

is performed by means of a feed-forward neural network, obtained by adding to the DBN a final output layer.

2.1. Low-Level data representation

Each action is encoded by a multilayer representation: at the first layer, spatio-temporal information are taken into account by processing the sequence of depth images obtained through a Kinect sensor. In particular, starting from raw data, we compute the *Average Depth Image* (*ADI*), the *Motion History Image* (*MHI*) and the *Depth Difference Image* (*DDI*), able to capture motion information respectively in the temporal and in the depth dimension [18]. Then, a 303-sized feature vector is extracted: Hu moments features encode the *MHI* and the *ADI* images, while the $\mathfrak{R}$ transform and Min-Max Depth Variations encode the *DDI*. As for the Min-Max Depth Variations, the main idea is to hierarchically partition the box containing the person into equal-sized cells and then to compute, for each cell, the maximum and the minimum values. As shown in [7] and [10], this combination is very promising since the considered features are complementary in their nature: Hu Moments and the Min-Max Depth Variations globally evaluate the overall distribution of the pixels, while the $\mathfrak{R}$ transform is able to capture local properties related to the alignment of subregions of the image.

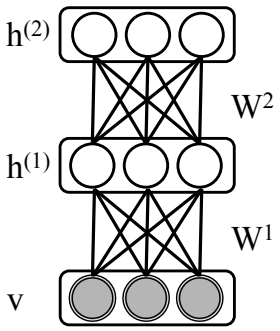

Figure 2: An example of DBN composed by three layers, a visible layer v and two hidden layers $h^{(1)}$ and $h^{(2)}$. W^1 and W^2 are the weight matrices, encoding the connections between the pair of layers.

2.2. High-Level data representation

The obtained representation is not yet suitable for being directly fed to a classifier because, although much more compact than the original images, it is still too detailed to easily discriminate actions of interest. Thus, from this low-level representation, a hierarchical representation is built using a DBN. The main idea is that the feature vectors that may be useful to describe events of interest can be defined, in a hierarchical way, in terms of combination of more abstract concepts. Although increasing the computational effort required during the learning step, the main advantage of the introduction of a deep learning strategy lies in the fact that we do not need to spend a lot of time in designing a *good* feature set, but instead a robust representation can be obtained directly from the available samples. It is worth noting that the learning of such hierarchical representation is performed in an unsupervised way: the training data do not need to be labeled, so that it can be obtained with a relatively small effort.

Furthermore, such a representation is able to discover and disentangle all those underlying and *a priori* unknown factors of variations that the data may hold, so making the final representation much more reliable.

In particular, DBN is a deep-based neural network composed of multiple layers of latent variables, called hidden units. A DBN can be considered as the composition of simple learning modules, namely Restricted Boltzmann Machines (RBMs), as shown in Figure 2: in fact, the learning procedure is performed one layer at a time by treating the values of the hidden variables in one layer, directly or indirectly extracted from the data, as the input data for training the next layer, whose aim is to capture higher-order correlations in the data. The two layers are connected each other by means of the weight matrix W and there are no connections within a layer [11].

In a more formal way, given the observed input variables v and l hidden layers $h^{(k)}$, $k = \{1, ..., l\}$ composed by binary units $h_i^{(k)}$, a DBN models the joint distribution between v and $h^{(k)}$ as follows:

$$p(v, h^{(1)}, h^{(2)}, ..., h^{(l)}) = P(v|h^{(1)})P(h^{(1)}|h^{(2)})...P(h^{(l-1)}|h^{(l)})p(h^{(l-1)}, h^{(l)}), \quad (1)$$

being

$$P(h^{(k)}|h^{(k+1)}) = \prod_i P(h_i^{(k)}|h^{(k+1)}), \quad (2)$$

$$P(h^{(k)} = 1|h^{(k+1)}) = A\left(b_i^k + \sum_j W_{ij}^k h_j^{(k+1)}\right). \quad (3)$$

A is the sigmoid activation function of the unit, computed as follows:

$$A(x) = \frac{1}{1 + e^{-x}}. \quad (4)$$

Less formally, the above equations state that the network can be interpreted as a probabilistic model where the variables at each layer are conditionally independent given the values of the variables in the subsequent layer. Thus, the joint probability distribution is completely characterized when the joint probabilities between adjacent layers are known. These probabilities are expressed in terms of an energy function:

$$p(h^{(k-1)}, h^{(k)}) = \frac{1}{Z} e^{-E(h^{(k-1)}, h^{(k)})} \quad (5)$$

where E and Z respectively represent the energy computed between the layers $h^{(k)}$ and $h^{(k-1)}$ and a normalization constant. The energy is defined as:

$$E(h^{(k-1)}, h^{(k)}) = -\sum_{i \in h^{(k-1)}} a_i^{k-1} h_i^{(k-1)} - \sum_{j \in h^k} b_j^k h_j^{(k)} - \sum_{i,j} h_i^{(k-1)} h_j^{(k)} W_{ij}^k \quad (6)$$

Note that we assume $h^{(0)} = v$ in the above equation to simplify the notation; thus we refer to the input layer as layer 0. The training of this network is performed in an incremental way. Given a training set of input vectors $V = \{v_1, \ldots, v_n\}$, assumed to be independent, the coefficients linking layer 0 and layer 1 are computed by maximizing the probability of obtaining V (as previously remarked, for this step it is not necessary to know the class associated to each v). Once this step has been performed, the conditional probabilities of obtaining the variables $h^{(1)}$ given V are known, and can be used to sample a training set $H^{(1)}$ of

vectors of the layer 1. This training set is then used for computing the coefficients linking layer 1 and layer 2, and the process is repeated for all the layers of the DBN. It is the possibility of training the layers one at a time that differentiates this model from previous multilayer architectures, making it computationally feasible even for a large number of hidden layers.

In a sense, each layer of the DBN performs a decomposition of its input space, trying to express it as a combination of subspaces associated to its output nodes, that can thus be seen as features describing its input vectors. The combined effects of the layers make the structure of these subspaces and of the corresponding features increasingly more complex, and thus potentially more suitable to fit the structure of the problem at hand.

2.3. Classification

The classification is performed by using a feed-forward (FF) neural network: starting from the DBN, the FF is set up by adding an output layer whose size is equal to the number of the classes. The value of each node of this layer is obtained from the values of the last hidden layer $h^{(l)}$ using a weighted linear combination and the sigmoid function $A(.)$:

$$out_i = A(a_i^{out} + \sum_j h_j^{(l)} \cdot W_{ij}^{out}) \quad (7)$$

The weights for the output layer are obtained by a supervised training using the well known Back Propagation algorithm; during this phase, Back Propagation is also used to refine the weights of the hidden layers. It is important to note that this is the only part of the training that requires labeled training data. This property is useful for applications where obtaining raw data is simple, while the labeling is a lengthy and costly process, as is the case of human actions recognition: by exploiting also the unlabeled data, it is possible to build a more complex network than the one obtainable from the available labeled data only.

3. Experimental Results

In this section we will discuss the results achieved by the proposed method and compare with other ones at the state of the art. In particular, three different approaches have been considered: the first uses an LVQ classifier, while the other two adopt a high level feature representation based on a bag of words approach. Note that both [7] and [10] use the same global descriptors adopted in the proposed approach. Finally, in [9] local features are combined with a bag of words based approach. A leave one out strategy has been used for testing all the considered methodologies.

The experimentation has been carried out over two standard and only recently proposed datasets, namely the Berkeley Multimodal Human Action Detection (hereinafter MHAD) dataset and the MIVIA dataset. The main justification behind this choice lies in the following two aspects: first, the datasets provide reference background and depth images, needed to extract the considered first level features. Furthermore, both the datasets have been widely used and then a comparison with state of the art approaches is possible.

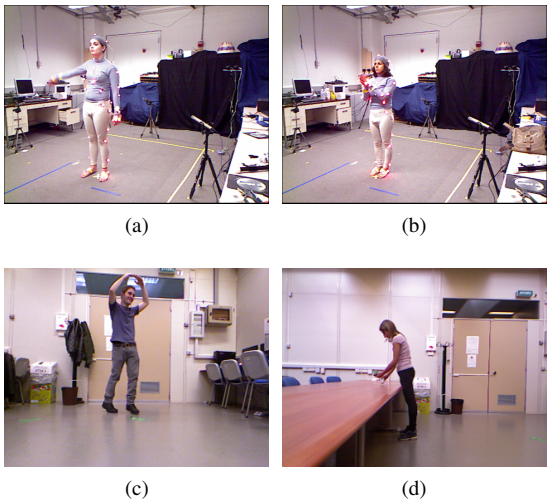

Figure 3: Images extracted from the MHAD (a,b) and the MIVIA (c, d) datasets.

The MHAD dataset [20] contains 11 actions, performed by 12 different subjects, namely 7 males and 5 females. Each subject performs 5 repetitions, so resulting in 660 sequences corresponding to about 82 minutes of recording time. The MIVIA dataset [10] contains 7 actions, performed by 14 different subjects, namely 7 males and 7 females. Each subject performs 3 repetitions of each action. The actions of both the datasets are listed in Table 1, while a few example images are shown in Figure 3.

One of the main disadvantage in the use of a deep learning based approach lies in the fact that no rules are defined for setting up the network parameters, and only the experience of the designer may help in this hard task. In this paper, we followed the advices given in [12]: the biases a and b are initialized to 0, while the weights W are initialized to random values drawn from a normal distribution $\mathcal{N}(0, 0.001)$. The momentum, able to increase the speed of learning, has been initialized to 0.5, in order to make the learning step more stable than no momentum at all. Furthermore, the size of the mini-batches is set up to a multiple (30 for both MHAD and MIVIA datasets) of the number of classes, by randomly selecting the samples of each batch. As for the number of hidden states, this parameter has a strong influence on the distribution that the RBM and then

the DBN is able to learn, since the representational power may be corrupted by a wrong setting of such parameter.

Several theoretical demonstrations have been provided about the optimal number of hidden units able to represent any data distributions [16][19][12]. Only recently, in [22], it has been experimentally proved that while the number of units is small, adding new units to the RBM has a strong impact on the performance during the classification step. On the other side, as we increase the number of hidden units, the performance improvement is not sufficient to justify the increasing number of hidden units, and then the higher effort required in terms of computational time.

Based on the above considerations, several experiments have been conducted. Note that, because of the large number of parameters as well as the high number of possible configurations, a grid search approach is not feasible for deep architectures. For this reason, we decided to focus on a hierarchical network: this choice is due to the fact that the idea is to increase the number of regions in which to partition the feature space in the lower layers, for better characterizing the different actions. The dimensionality of the feature vector, and then the number of regions, are hierarchically reduced by finding a kind of similarity between the regions. The main advantage deriving from this choice lies in the fact that regions joining different actions can be merged because of their low discriminative power; second, the dimensionality of the final feature vectors will be significantly reduced, so improving the overall performance of the final layer, which is the one that need to be trained using labeled data. For such reasons, we considered a pyramidal DBN, composed by 6 layers whose size is respectively 2048, 1024, 512, 256, 128 and 64.

The achieved results are summarized in Table 1, where the accuracy rate for every class is shown. As we can see, the introduction of multiple layers influences the performance of the proposed approach, which confirms its effectiveness over both the considered datasets. Furthermore, it is evident that the performance is strongly affected by the quantity of available data, which intrinsically depends on the length per recording. Indeed, the classification of the action in the classes providing more input data, such as *sit down then stand up* or *bending - hands up all the way down* in the MHAD dataset or *random movements* and *stopping* in the MIVIA dataset strongly outperforms the one of the classes corresponding to shorter actions. It is mainly due to the fact that the representation is obtained in an unsupervised way, and then the partition of the space is polarized towards those classes providing more samples. Of course, as expected, increasing the amount of unlabeled data provided as input to the deep network would significantly increase the performance of the proposed approach, making it especially suitable for its use in real applications, where a lot of unlabeled data may be easily provided before the setup of the system.

MHAD					
Action	**Length**	**Prop**	**[10]**	**[7]**	**[9]**
Jumping in place	5	**78.0**	75.0	74.9	44.9
Jumping jacks	7	92.8	**96.4**	58.9	63.7
Bending - hands up all the way down	12	**90.8**	60.7	47.5	73.9
Punching (boxing)	10	83.2	**92.9**	67.9	66.9
Waving - two hands	7	**94.9**	58.9	70.2	72.2
Waving - one hand (right)	7	**88.5**	75	35.5	75.0
Clapping hands	5	**84.6**	60.7	59.9	63.3
Throwing a ball	3	47.4	**75.0**	48.9	32.5
Sit down then stand up	15	**95.5**	75.0	57.3	77.0
Sit down	2	32.7	76.8	**79.5**	10.8
Stand up	2	28.7	**82.1**	64.9	14.0
Average Accuracy		**85.8**	72.9	57.4	66.2

(a)

MIVIA					
Action	**Length**	**Prop**	**[10]**	**[7]**	**[9]**
Opening a jar	2	**73.3**	67.9	**73.3**	49.6
Drinking	3	70.4	53.6	**73.9**	53.4
Sleeping	3	76.8	**98.2**	65.7	84.6
Random Movements	11	96.8	75.0	**99.0**	79.6
Stopping	7	87.3	**100**	62.9	60.0
Interacting with a table	3	85.3	**100**	94.0	84.6
Sitting	3	72.7	82.1	80.9	**89.9**
Average Accuracy		**84.7**	83.0	82.5	74.6

(b)

Table 1: Average Length per recording (in seconds) and accuracy rate of the single actions tested on the MHAD (a) and the MIVIA (b) datasets. The results obtained by the proposed approach (*Prop* in the tables) are compared with [10], [7] and [9].

4. Conclusions

In this paper we proposed a method for recognizing human actions by a deep learning based approach. A set of global descriptors is extracted from depth images and a high level representation is obtained by employing a Deep Belief Network trained using a Restricted Boltzmann Machine. The classification is performed by a feed-forward

neural network. The main advantage of the proposed approach lies in the fact that the high level representation is autonomously extracted by the data into an unsupervised way, without requiring the feature engineering, which is a very labour intensive task, usually needed for this kind of applications. The method has been tested over two standard datasets and the achieved results, compared with state of the art approaches, confirm its robustness and its effectiveness.

References

[1] G. Acampora, P. Foggia, A. Saggese, and M. Vento. Combining neural networks and fuzzy systems for human behavior understanding. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 88–93, Sept 2012.

[2] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In A. Salah and B. Lepri, editors, *Human Behavior Understanding*, volume 7065 of *Lecture Notes in Computer Science*, pages 29–39. Springer Berlin Heidelberg, 2011.

[3] Y. Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127, Jan. 2009.

[4] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE T Pattern Anal.*, 35(8):1798–1828, Aug 2013.

[5] L. Brun, G. Percannella, A. Saggese, and M. Vento. Recognition of human actions by kernels of visual strings. In *Advanced Video and Signal-Based Surveillance (AVSS), 2014 IEEE International Conference on*, 2014.

[6] L. Brun, A. Saggese, and M. Vento. Dynamic scene understanding for behavior analysis based on string kernels. *Circuits and Systems for Video Technology, IEEE Transactions on*, 2014.

[7] V. Carletti, P. Foggia, G. Percannella, A. Saggese, and M. Vento. Recognition of human actions from rgb-d videos using a reject option. In A. Petrosino, L. Maddalena, and P. Pala, editors, *New Trends in Image Analysis and Processing ICIAP 2013*, volume 8158 of *Lecture Notes in Computer Science*, pages 436–445. Springer Berlin Heidelberg, 2013.

[8] Y. Chen, Q. Wu, and X. He. Human action recognition based on radon transform. In W. Lin, D. Tao, J. Kacprzyk, Z. Li, E. Izquierdo, and H. Wang, editors, *Multimedia Analysis, Processing and Communications*, volume 346 of *Studies in Computational Intelligence*, pages 369–389. Springer Berlin Heidelberg, 2011.

[9] P. Dollar, V. Rabaud, G. Cottrell, and S. 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*, pages 65–72, Oct 2005.

[10] P. Foggia, G. Percannella, A. Saggese, and M. Vento. Recognizing human actions by a bag of visual words. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pages 2910–2915, Oct 2013.

[11] G. E. Hinton. Deep belief networks. *Scholarpedia*, 4(5):5947, 2009.

[12] G. E. Hinton. A practical guide to training restricted boltzmann machines. In *Neural Networks: Tricks of the Trade (2nd ed.)*, pages 599–619. 2012.

[13] J. Jang, Y. Park, and I. Suh. Empirical evaluation on deep learning of depth feature for human activity recognition. In M. Lee, A. Hirose, Z.-G. Hou, and R. Kil, editors, *Neural Information Processing*, volume 8228 of *Lecture Notes in Computer Science*, pages 576–583. Springer Berlin Heidelberg, 2013.

[14] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE T Pattern Anal.*, 35(1):221–231, Jan 2013.

[15] Q. Le, W. Zou, S. Yeung, and A. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, pages 3361–3368, June 2011.

[16] N. Le Roux and Y. Bengio. Representational power of restricted boltzmann machines and deep belief networks. *Neural Comput.*, 20(6):1631–1649, June 2008.

[17] W. Li, Q. Yu, H. Sawhney, and N. Vasconcelos. Recognizing activities via bag of words for attribute dynamics. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2587–2594, June 2013.

[18] V. Megavannan, B. Agarwal, and R. Babu. Human action recognition using depth maps. In *Signal Processing and Communications (SPCOM), 2012 International Conference on*, pages 1 –5, july 2012.

[19] G. Montufar and N. Ay. Refinements of universal approximation results for deep belief networks and restricted boltzmann machines. *Neural Comput.*, 23(5):1306–1319, May 2011.

[20] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Berkeley mhad: A comprehensive multimodal human action database. In *IEEE Workshop on Applications on Computer Vision (WACV)*. IEEE, 2013.

[21] R. Poppe. A survey on vision-based human action recognition. *Image Vision Comput.*, 28(6):976 – 990, 2010.

[22] A. Romero and C. Gatta. Do we really need all these neurons? In *Pattern Recognition and Image Analysis*, volume 7887 of *Lecture Notes in Computer Science*, pages 460–467. Springer Berlin Heidelberg, 2013.

[23] R. Salakhutdinov and G. E. Hinton. Deep boltzmann machines. In *AISTATS*, pages 448–455, 2009.

[24] S. Vishwakarma and A. Agrawal. A survey on activity recognition and behavior understanding in video surveillance. *Visual Comput.*, 29(10):983–1009, 2013.

[25] C. Yuan, X. Li, W. Hu, H. Ling, and S. Maybank. 3d r transform on spatio-temporal interest points for action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 724–730, June 2013.