A Multimodal Deep Learning Network for Group Activity Recognition

Silvia Rossi
and Roberto Capasso
Department of Electrical Engineering
and Information Technologies
University of Naples Federico II
Naples, Italy
Email: silvia.rossi@unina.it

Giovanni Acampora
and Mariacarla Staffa
Department of Physics
University of Naples Federico II
Naples, Italy
Email: {giovanni.acampora,mariacarla.staffa}@unina.it

***Abstract*—Several studies focused on single human activity recognition, while the classification of group activities is still under-investigated. In this paper, we present an approach for classifying the activity performed by a group of people during daily life tasks at work. We address the problem in a hierarchical way by first examining individual person actions, reconstructed from data coming from wearable and ambient sensors. We then observe if common temporal/spatial dynamics exist at the level of group activity. We deployed a Multimodal Deep Learning Network, where the term multimodal is not intended to separately elaborate the considered different input modalities, but refers to the possibility of extracting activity-related features for each group member, and then merge them through shared levels. We evaluated the proposed approach in a laboratory environment, where the employees are monitored during their normal activities. The experimental results demonstrate the effectiveness of the proposed model with respect to an SVM benchmark.**

I. Introduction

Human activity/action recognition is an important yet challenging research area with many applications in health-care, smart environments, as well as in robotics [1]. The recognition of human activities has been approached in two different ways, namely using external (e.g., switches and cameras) and wearable sensors. In the former, the devices are fixed in predetermined points of interest, so the inference of activities entirely depends on the voluntary interaction of the users with the sensors. In these settings, computer vision-based techniques have been widely used for human activity tracking [2] and gesture recognition [3], [4], but they mostly require infrastructure support, for example, installation of cameras in the monitoring areas and human frontal poses and gestures with respect to the robot. Even additional communication channels are required in order to discriminate among activities [5]. In the latter, the devices are attached to the user (e.g., wristbands, wristwatches, armbands). Hence, the approach is to process the data from inertial measurement unit sensors worn on a user's body or built in a user's smartphone to track his/her motion [6]. In this work, we decided to rely on activity recognition from data obtained from smart-phone and iBeacon sensors, due to their advantage to be not-invasive as well as not requiring additional technicalities to solve issues which can arise when using external camera sensors.

Several studies on human activity recognition/prediction have been conducted [7], [8], thus, many classifiers are deployed in literature. Despite, such numerous approaches to activity recognition, the Group Activity Recognition (GAR) problem is still an open research area. Few approaches are present in the literature and they mainly rely on the use of external cameras [9]. Additionally, several approaches to GAR rely on the use of hand-crafted features for their predictive techniques [10] and, for this reason, they suffer from the limitations related to their representational abilities to model a complex group activity. Deep Learning (DL) techniques overcame this issue by providing the possibility of a hierarchical schema for learning and acting also as features extractors from raw data [11]. Results of DL approaches typically outperform state-of-the-art classification techniques.

Among DL approaches to address the GAR problem, Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNNs) are typically used. For example, in 2015, a CNN was proposed for GAR [12] from video data, where the activity of each user was recognized by a different CNN and, afterward, the results were used as input of a Multi-Layer Perceptron (MLP) that recognized the group activity. In this work, a neural network-based hierarchical approach has been leveraged to refine person action labels and to learn predicting the group activity simultaneously. Other approaches based on LSTM-based temporal models have also been proposed to learn discriminative information from time-varying activity data from videos [13]. Another example relying on the use of an LSTM-based network is represented by the work of Ramanathan et al. [9], where sports videos are analyzed with the aim of recognizing the principal actor of the scene, however, here the authors do not consider the aggregations of all people actions to determine the group activity. In general, the adoption of DL techniques yielded very good results mainly in video sequences analysis [14].

In the past, several ad-hoc models have been built to analyze the activities performed by a group that is intended as a single entity rather than as the composition of individuals [15], [16]. However, it has been shown that modeling a group activity with a hierarchical structure, where single individual

978-1-5090-6014-6/18/$31.00 ©2018 IEEE

components are first considered and then combined, achieves better results [17]. In [18], for example, the authors proposed the recognition of complex activities subdividing them into primitive activities and by creating a hierarchy.

Inspired by this evidence, as well as by the success obtained in the field by the adoption of DL techniques, we address the GAR problem in a hierarchical way, by first examining individual person's actions. We aim at observing if common temporal/spatial dynamics exist at the level of group activity by exploiting a deep belief network (DBN) [19] model of DL-based networks. A multimodal DBN network has been designed for managing a multi-subject dataset. A multimodal network is typically used to elaborate the different input modalities separately and then to merge the results through the use of shared levels [20]. In this direction, we propose the elaboration of data coming from different subjects as different modalities to be merged at higher levels of the architecture. We evaluated the proposed approach in a laboratory environment, where the employees are monitored during their normal activities as single individuals or as a group of two people. The experimental results demonstrate the effectiveness of the proposed model with respect to an SVM baseline.

II. Material and Methods

Typically, a DBN is composed of cascading multiple Restricted Boltzmann Machines (RBMs) [21] that are energy-based unsupervised learning models. This deep architecture has been successfully used as a feature extractor for a different type of data (e.g., text, image, and sound data) and as a good initial training step for deep architectures.

RBMs are energy-based unsupervised learning models whose parameters are usually learned through the contrastive divergence (CD) [22] method. An RBM is a particular type of a Markov random field characterized by two layers: one visible n input units layer $v \in 0,1^n$, and one layer of m hidden stochastic units $h \in 0,1^m$, which are weighted and connected. No connection exists on the same layer. Every unit within an RBN has an activation energy E specified by the following energy function:

$$E(v,h) = -\sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_j h_j - \sum_{i=1}^{n}\sum_{j=1}^{m} v_i w_{ij} h_j \qquad (1)$$

where a_i and b_j represent, respectively, the bias of the visible i and of the hidden j unit, while $w_{i,j}$ is the weight between the units i and j of the two layers. A scalar energy function is associated with each configuration of the variables in the dataset. The learning process consists in shaping the energy function so that it assumes its desired form. The probability distribution of (v,h) is:

$$P(v,h) = \frac{1}{Z} e^{-E(v,h)} \qquad (2)$$

where Z is a normalizing factor to limit the probability in the range of $[0,1]$. Namely, $Z = \sum_{v,h} e^{-E(v,h)}$. The conditional distribution of the visible vector v given the hidden vector h can be derived from Equation 2 using $P(v) = (1/Z)\sum_h e^{-E(v,h)}$, where $P(v)$ represents the probability that the model we choose with the given weights will generate the visible units v. When we train the model, it is expected that we can get the maximum probability by changing the weights. Therefore, the final weights can be got through a stochastic descent of the gradient with the log-likelihood function of the training data:

$$\frac{\delta logp(v)}{\delta w_{ij}} = < v_i.h_j >_{data} - < v_i.h_j >_{model} \qquad (3)$$

where $< v_i.h_j >_{data}$ expresses the expectation of the observed data and the results of the weights in the training set, while $< v_i.h_j >_{model}$ is the same expectation under the distribution generated by the deployed model. Thus, we can describe the updating of the weights with the following equation:

$$\Delta w_{ij} = \varepsilon(< v_i.h_j >_{data} - < v_i.h_j >_{model}) \qquad (4)$$

where the parameter ε can be introduced in order to obtain the optimum performance. Specifically, ε represents the learning rate of the weights that decides the rate of the algorithm convergence.

According to other researchers [22], we adopted the most widely used learning method for RBMs consisting in the *Contrastive Divergence* (CD) algorithm, in order to overcome this computational hurdle. We expressed the Equation 4 as $\Delta w_{ij} = \varepsilon(< v_i.h_j >_{data} - < v_i.h_j >_{recon})$, where $< v_i.h_j >_{recon}$ is the reconstruction distribution of our model.

Then, we approximate the model distribution using Gibbs sampling with the probability distribution of input samples:

$$< v_i.h_j >_{data} = \frac{1}{n}\sum_{k=1}^{n} v_i^{(k)} P(v^{(k)}, h_j^{(k)}) \qquad (5)$$

$$< v_i.h_j >_{Gc} = \frac{1}{n}\sum_{k=1}^{n} \bar{v}_i^{(k)} P(\bar{v}^{(k)}, h_j^{(k)}) \qquad (6)$$

where, Gc denotes the approximated distribution of the $k-th$ iteration of Gibbs sampling and n is the number of samples, and $\bar{v}_i$ is a negative phase visible unit from Gibbs sampling.

III. The proposed approach

To approach the GAR problem, we designed a Deep Belief Network (DBN) [23] composed of set three layers (see Figure 1):

1) the first is a Restricted Boltzmann Machines (RBM) [24], able to recognize key characteristics of the members of the group;
2) the second is a Restricted Boltzmann Machines (RBM) stack providing shared levels to recognize features of the whole group;
3) the third part is represented by a Multi-Layer Perceptron (MLP) [25], which is a feed-forward neural network artificial model mapping input datasets into a series of not linearly separable classes.

Starting from the low layer, the first classifier is a multimodal DBN (see Figure 1), and therefore it has a layer formed by n RBMs, one for each subject. The probability distribution

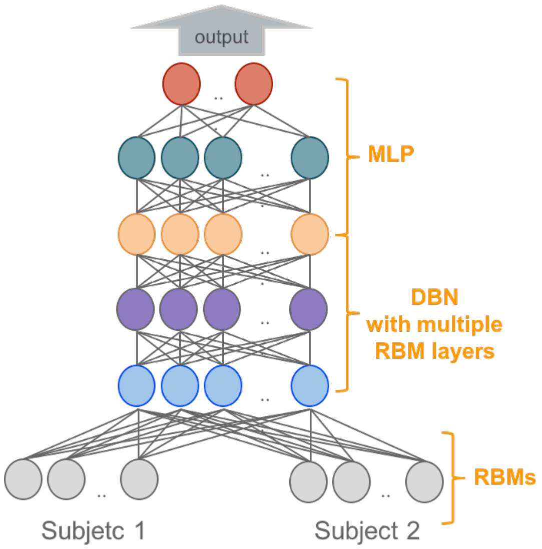

Fig. 1. Multimodal DBN

of each RBM is not affected by data received in input by the other RBM, so imposing a separation between data. The network is called multimodal since we took inspiration from the works of Srivastava et al. [26] and Ngiam et al. [27]. A system is typically called multimodal when it uses more than one modality at the same time, therefore in our approach, each activity is perceived in a different and isolated way before being elaborated together with the others. Therefore, an abstraction of the individual activities is not shared among the levels, and the two RBMs can be pre-trained in an unsupervised manner using the CD algorithm.

The RBMs can be used as blocks in a DBN to form the deep architecture. The first RBM of the stack must recognize data in the continuous, this variant of the RBM is called Continuous Restricted Boltzmann Machines (CRBM) [28] and accepts the continuous input through a Gaussian transformation on the visible layer in a binary input for the hidden level. A DBN constructed in this way is called Continuous Deep Belief Network (CDBN). The Gaussian transformations are the most used continuous distributions since different phenomena seem to follow, at least approximately, this distribution.

The outputs of the RBMs are concatenated into a standard DBN. This shared RBM is commonly called in the literature as Shared Level. To train the DBN a layer-by-layer approach is used. The aim is to determine the dependence of the units of a level from the units of the higher levels. The layer-by-layer training is based on the idea of using unit values of a level as data for training the next level. In this way, each layer attempts to model the distribution of its incoming data. This procedure is carried out by operating on the network, seeing it as a composition of simple RBM learning modules.

A greedy non-supervised learning paradigm [29] can be applied to the DBNs to train the RBMs stack as follows:

1) Train the first RBM with the training set vectors;
2) The hidden layer of the first trained RBM is used as an input vector for the second RBM of the stack;
3) The second RBM of the stack is trained with the data processed from the previous step;
4) Steps 2 and 3 are iterated for the desired number of RBMs (i.e., this number corresponds to the depth of the network);
5) The architecture parameters obtained from the previous steps are optimized by means of a likelihood measure or by a second training, this time supervised, using a mostly linear classifier.

This last supervised stage is implemented by using an MLP. The MLP is an oriented neural network with an input layer that takes an input vector from the output of visible layer of DBN. It propagates the vector to its hidden layers and, after, to the output layer for the final classification. The MLP structure is a supervised network that uses the stochastic gradient descent with mini-batches through the back-propagation algorithm.

A. *DBN parameters tuning*

We implemented our DBN-based approach via Theano and Google Tensor-flow libraries endowed with additional genetic algorithms for improving the classification network convergence. This algorithm allows to test many different configurations of our proposed model of DBN, in terms of the number of layers and nodes in each layer, with the aim to converge to a sub-optimal solution, that is to individuate a configuration yielding to a high classification percentage. Moreover, we decided to keep the same number of nodes for each of the shared levels. The genetic algorithm starts from a set (called population) of possible solutions (called individuals). The process consists in evolving the population in the following way: at each iteration, it operates a selection of individuals of the current population, employing them to generate new elements, which will constitute the new population for the following iteration (called generation). The evolution is obtained through a partial recombination of the solutions. Each individual, in fact, transmits part of its genetic heritage to its descendants and random mutations introduce elements of disorder giving rise to individuals with characteristics different from those present in the original species. Once the evolution phase is over, the population is analyzed and only the solutions that best solve the problem are kept. The objective/fitness function used to evaluate the performance of the population is represented by the classification results computed among the classes the DBN should classify. The process is iterated until a suitable value for the objective function is obtained, meaning the so built Multimodal DBN structure yields results that at least out-performed some benchmark solutions.

IV. Experimental Setup and Evaluation

A. *The Dataset*

The considered dataset is the PRISCA lab DyadHAR dataset, where activities of daily living in lab are labeled by the use of a smart-phone. The PRISCA lab, a 250mq^2 laboratory, is divided into several such as: *Coffee area*, which is the area mainly used during coffee breaks; *Work area*, which is the

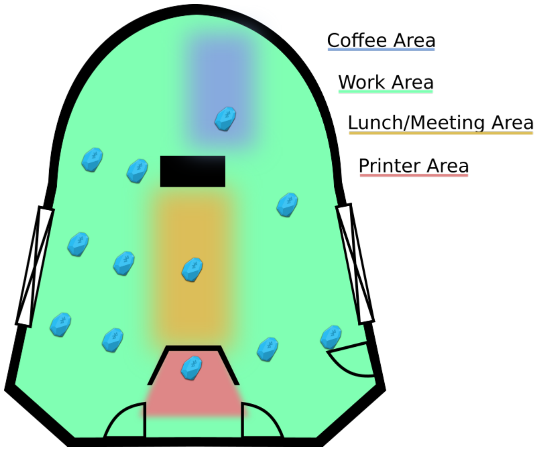

Fig. 2. PRISCA Lab map

largest area in the laboratory where there are working-stations. Each member of the laboratory has a position in this area that can change in different days; *Lunch/Meeting area*, which is the area that includes a table around which people meet when there are meetings or to eat; *Printer Area*, which is the area where there are printers.

The dataset contains 96 measurements for each sensor, i.e., 96x3 = 288 accelerometer data and 96x3 = 288 gyroscope data. Accelerometer and gyroscope data are extracted from a smart-phone at the height of the user belt. Additionally, 5x12 = 60 data represents the Received Signal Strength Indication (RSSI) values of the iBeacon placed in the lab (see Figure 3). For each device, the decimal value of the RSSI was saved, for those that were not seen by the scan activity a null value was inserted. Finally, the entry vector also contains timestamp (2 nodes) values representing the beginning of the activity. Hence, each of the input RBMs counts a total of 638 entry nodes.

For each entry of the DyadHAR dataset, there are 2 sequences, one relative to a subject and the other relative to a second subject as shown below:

1) **Exit** - 8781 entries;
2) **Meeting/Seminar** - 3478 entries;
3) **Coffee-break** - 2410 entries;
4) **Work** - 1114 entries;
5) **Lunch** - 3732 entries;
6) **No-Group activity** - 4419 entries;

The class No-Group activity considers two users not acting together.

B. DBN parameters setting

The genetic algorithm found that the optimal number of nodes was 7800 for each shared level with two shared level and 1000 nodes for the RBM layer that takes in input data of the two individual subjects.

The DBN training was performed in 2 steps:

- First, the RBMs were trained on the individual subjects using 15 non-supervised epochs with a learning rate of $\varepsilon = 5 \cdot 10^{-5}$ and 200 supervised epochs with a learning rate of $\varepsilon = 0.1$ and batch size equal to 25.
- Second, resulting data were fed to the DBN taking, in this way, 2000 input characteristics in input (1000 values for each RBM). For this step also, 15 unsupervised epochs were used, but with a variable learning rate of $\varepsilon = 3 \cdot 10^{-5}$ for the first shared level and 10^{-5} for the second shared level. The supervised epochs were 200 with a learning rate of $\varepsilon = 0.1$ and batch size of 25.

C. Results

In order to evaluate our approach, we compared the classification results obtained by the so designed DBN with those obtained through the adoption of a classic classification approach, such as Support Vector Machines (SVMs).

In Tables I and II classification results of a single folder for the DBN and SVN are shown. The SVN, in almost every folder, got better results for the Seminar and Lunch classes, but in all the other classes the DBN got better results. From the confusion matrix, notice that the SVM made a lot of mistakes with the single activity class. In fact, some single activities have been exchanged for group activities and vice versa.

TABLE I
DBN RESULTS ON A SINGLE FOLD FOR THE DYADHAR DATASET

Class	Exit	Meeting	Coffee-break	Working	Lunch	Single activity	Recall
Exit	1772	0	0	0	0	0	100%
Meeting	0	766	0	0	5	0	99.3%
Coffee-break	0	0	436	0	0	0	100%
Working	0	0	0	181	0	0	100%
Lunch	0	8	0	0	720	0	98.9%
Single activity	0	9	11	0	0	889	97.8%
Precision	100%	97.8%	97.5%	100%	99.3%	100%	

TABLE II
SVN RESULTS ON A SINGLE FOLD FOR THE DYADHAR DATASET

Class	Exit	Meeting	Coffee-break	Working	Lunch	Single activity	Recall
Exit	1769	0	0	0	0	3	99.8%
Meeting	0	771	0	0	0	0	100%
Coffee-break	0	0	432	0	0	4	99.0%
Working	0	0	0	181	0	0	100%
Lunch	0	0	0	0	728	0	100%
Single activity	20	17	13	4	6	849	93.3%
Precision	98.8%	97.8%	97.0%	97.8%	99.1%	99.1%	

The results obtained from the 5-fold cross-validation are shown in Table III. On average, the SVM achieved a precision of 98.6% and a recall of 98.8%. On the other hand, the DBN obtained an average precision of 99.1% and an average recall of 99.2% by improving the SVM results of 0.4%.

Finally, in Figure 4, a classes plot obtained by using the t-SNE algorithm is shown. The plot shows that there are

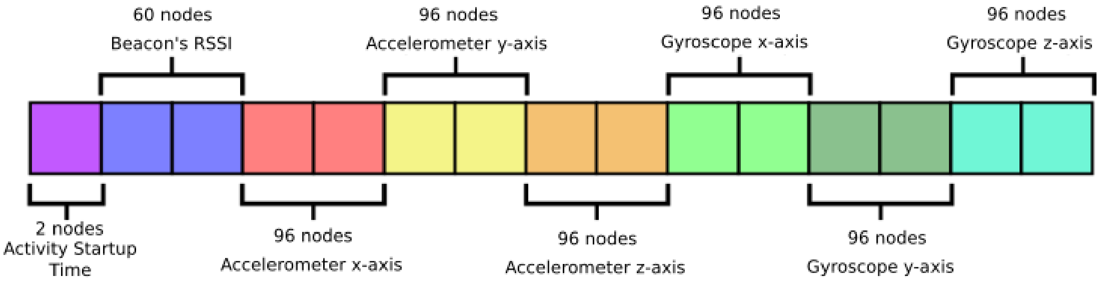

Fig. 3. SingleHAR input for the DBN

TABLE III
CLASSIFICATION RESULTS ON 5-FOLDS FOR THE DYADHAR DATASET

	SVM		DBN	
	Precision	Recall	Precision	Recall
Fold-1	98.3%	98.7%	99.1%	99.3%
Fold-2	98.6%	98.7%	98.4%	98.6%
Fold-3	98.6%	98.8%	99.4%	99.3%
Fold-4	98.6%	98.8%	99.2%	99.4%
Fold-5	99.0%	99.0%	99.5%	99.5%
Average	98.6	98.8	99.1	99.2

numerous areas of intersection between the various activities, so an algorithm that evaluates all the characteristics in a hierarchical way for the recognition becomes necessary. The deep methods, as seen above, stack processing layers, hence the primitive characteristics are combined to recognize medium-level features up to the more complex ones. The evidence is in the ability to recognize also non-group activities: distinguishing, for example, working and working in groups becomes difficult if the subjects are in the same area, these are indeed points on the boundaries of the areas. The intrinsic characteristics of the deep methods have allowed recognizing many limit cases, which the SVM has not recognized. For the same reasons the DBN has lost the comparison with activities such as Lunch and a Meeting, as they have been confused with each other. This happened because the positions in the laboratory and the gyroscope and accelerometer readings were similar, the only thing that changes, for which it is possible to discriminate among these activities, are the daytimes. In this case, the hierarchical evaluation has failed on some acknowledgments, while the search for the separation limit of the SVM has been more effective.

V. DISCUSSION AND CONCLUSIONS

Despite the huge diffusion of low cost sensors, such as accelerometer and gyroscope, able to track key features for activity recognition, the recognition of complex as well as group activities that relies on such type of sensors is still immature due to the lack of precision of the currently existing recognition methods and the scarcity of labeled data for the training phase. Most sensor-based activity recognition systems use supervised machine learning algorithms such as Support Vector Machines and Decision Trees. A quite new approach is presented by Deep Learning techniques that introduce numerous advantages in terms of system performance and flexibility regarding the problem of detecting humans group activities. First of all, its hierarchical structure intrinsically provides an effective tool for extracting hierarchical characteristics from high-dimensional data. This permits to lower the computational costs. Second, models such as DBN can use unlabeled activity samples for an unsupervised pre-training phase, which may be particularly useful due to the lack of labeled datasets related to this topic and to a large amount of unlabeled data. Additionally, these techniques are more robust against the problem of over-fitting with respect to the discriminatory models [30].

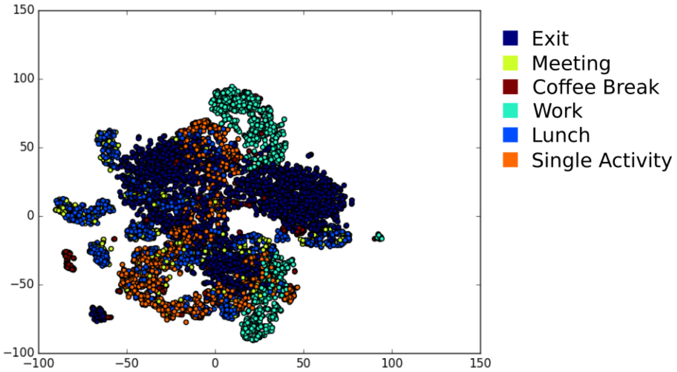

Fig. 4. t-SNE plot of the DyadHAR dataset

In this paper, we presented a DBN-based solution to the problem of Group Activity Recognition. We adopted a hierarchical multimodal structure, where the first meta-layer is deployed to recognize single users action and the second one to observe if common temporal/spatial dynamics exist yielding to a group activity. We evaluated the proposed approach in a laboratory environment, where the participants labeled their daily activities through the use of an app on a mobile phone. Additionally, we also considered ambient sensors in order to combine this information with those coming from the mobile phone for improving the activity detection performance. The experimental results on the considered datasets demonstrated that the proposed model outperforms state-of-the-art SVN-based action classification techniques.

ACKNOWLEDGMENT

This study was has been partially supported by MIUR (Italian Ministry of Education, Universities, and Research) within the PRIN2015 research project UPA4SAR - "User-centered Profiling and Adaptation for Socially Assistive Robotics" (grant n. 2015KB-L78T).

References

[1] E. Burattini, A. Finzi, S. Rossi, and M. Staffa, "Attentional human-robot interaction in simple manipulation tasks," in *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*, March 2012, pp. 129–130.

[2] M. Staffa, M. D. Gregorio, M. Giordano, and S. Rossi, "Can you follow that guy?" in *ESANN*, 2014.

[3] V. Magnanimo, M. Saveriano, S. Rossi, and D. Lee, "A bayesian approach for task recognition and future human activity prediction," in *the 23rd IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2014, pp. 726–731.

[4] S. Iengo, S. Rossi, M. Staffa, and A. Finzi, "Continuous gesture recognition for flexible human-robot interaction," in *IEEE International Conference on Robotics and Automation, ICRA Hong Kong, China, May 31 - June 7*, 2014, pp. 4863–4868.

[5] R. Caccavale, E. Leone, L. Lucignano, S. Rossi, M. Staffa, and A. Finzi, "Attentional regulations in a situated human-robot dialogue." in *RO-MAN*. IEEE, 2014, pp. 844–849.

[6] G. Li, C. Zhu, J. Du, Q. Cheng, W. Sheng, and H. Chen, "Robot semantic mapping through wearable sensor-based human activity recognition," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 5228–5233.

[7] S. Rossi, F. Ferland, and A. Tapus, "User profiling and behavioral adaptation for hri: A survey," *Pattern Recognition Letters*, vol. 99, no. Supplement C, pp. 3 – 12, 2017, user Profiling and Behavior Adaptation for Human-Robot Interaction.

[8] X. Hong, C. D. Nugent, M. D. Mulvenna, S. I. McClean, B. W. Scotney, and S. Devlin, "Evidential fusion of sensor data for activity recognition in smart homes." *Pervasive and Mobile Computing*, vol. 5, no. 3, pp. 236–252, 2009.

[9] V. Ramanathan, J. Huang, S. Abu-El-Haija, A. N. Gorban, K. Murphy, and L. Fei-Fei, "Detecting events and key actors in multi-person videos." in *CVPR*. IEEE Computer Society, 2016, pp. 3043–3053.

[10] T. Lan, Y. Wang, W. Yang, S. N. Robinovitch, and G. Mori, "Discriminative latent models for recognizing contextual group activities." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 8, pp. 1549–1562, 2012.

[11] G. Ercolano, D. Riccio, and S. Rossi, "Two deep approaches for adl recognition: A multi-scale lstm and a cnn-lstm with a 3d matrix skeleton representation," in *26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Aug 2017, pp. 877–882.

[12] Z. Deng, M. Zhai, L. Chen, Y. Liu, S. Muralidharan, M. J. Roshtkhari, and G. Mori, "Deep structured models for group activity recognition," *arXiv preprint arXiv:1506.04191*, 2015.

[13] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and F.-F. Li, "Every moment counts: Dense detailed labeling of actions in complex videos." *CoRR*, vol. abs/1507.05738, 2015. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1507.html#YeungRJAML15

[14] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos." in *NIPS*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., 2014, pp. 568–576. [Online]. Available: http://dblp.uni-trier.de/db/conf/nips/nips2014.html#SimonyanZ14

[15] D. Gordon, J.-H. Hanne, M. Berchtold, A. A. N. Shirehjini, and M. Beigl, "Towards collaborative group activity recognition using mobile devices," *Mobile Networks and Applications*, vol. 18, no. 3, pp. 326–340, 2013.

[16] S. M. Khan and M. Shah, "Detecting group activities using rigidity of formation." in *ACM Multimedia*, H. Zhang, T.-S. Chua, R. Steinmetz, M. S. Kankanhalli, and L. Wilcox, Eds. ACM, 2005, pp. 403–406. [Online]. Available: http://dblp.uni-trier.de/db/conf/mm/mm2005.html#KhanS05

[17] T. Lan, L. Sigal, and G. Mori, "Social roles in hierarchical models for human activity recognition." in *CVPR*. IEEE Computer Society, 2012, pp. 1354–1361. [Online]. Available: http://dblp.uni-trier.de/db/conf/cvpr/cvpr2012.html#LanSM12

[18] B. Bruno, F. Mastrogiovanni, A. Sgorbissa, T. Vernazza, and R. Zaccaria, "Analysis of human behavior recognition algorithms based on acceleration data," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1602–1607.

[19] N. L. Roux and Y. Bengio, "Deep belief networks are compact universal approximators." *Neural Computation*, vol. 22, no. 8, pp. 2192–2207, 2010.

[20] G. Fortino, S. Galzarano, R. Gravina, and W. Li, "A framework for collaborative computing and multi-sensor data fusion in body sensor networks." *Information Fusion*, vol. 22, pp. 50–70, 2015.

[21] H. Larochelle and Y. Bengio, "Classification using discriminative restricted boltzmann machines." in *ICML*, ser. ACM International Conference Proceeding Series, W. W. Cohen, A. McCallum, and S. T. Roweis, Eds., vol. 307. ACM, 2008, pp. 536–543.

[22] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, pp. 1771–1800, 2002.

[23] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[24] L. Arnold, S. Rebecchi, S. Chevallier, and H. Paugam-Moisy, "An introduction to deep learning." in *ESANN*, 2011.

[25] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[26] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," in *Advances in neural information processing systems*, 2012, pp. 2222–2230.

[27] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.

[28] H. Chen and A. F. Murray, "Continuous restricted boltzmann machine with an implementable training algorithm," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 150, no. 3, pp. 153–158, 2003.

[29] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, *et al.*, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, vol. 19, p. 153, 2007.

[30] A.-R. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.