Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 52-11 (2019) 128–133

A Random Forest-based Approach for Hand Gesture Recognition with Wireless Wearable Motion Capture Sensors

Nicoló Bargellesi * **Mattia Carletti** * **Angelo Cenedese** ** **Gian Antonio Susto** ** **Matteo Terzi** **

* *Department of Information Engineering (DEI), University of Padova, Padova, Italy (e-mail: nicolo.bargellesi@studenti.unipd.it; mattia.carletti@unipd.it).*

** *Department of Information Engineering (DEI) and Human-Inspired Technology Center (HIT), University of Padova, Padova, Italy (e-mail: cenedese@dei.unipd.it; gianantonio.susto@dei.unipd.it; terzimat@dei.unipd.it)*

Abstract: Gesture Recognition has a prominent importance in smart environment and home automation. Thanks to the availability of Machine Learning approaches it is possible for users to define gestures that can be associated with commands for the smart environment. In this paper we propose a Random Forest-based approach for Gesture Recognition of hand movements starting from wireless wearable motion capture data. In the presented approach, we evaluate different feature extraction procedures to handle gestures and data with different duration. To enhance reproducibility of our results and to foster research in the Gesture Recognition area, we share the dataset that we have collected and exploited for the present work.

Keywords: Gesture Recognition, Machine Learning, Motion Capture, Random Forests

1. INTRODUCTION

Characterizing/sensing the human-in-the-loop has become a prominent topic in control system technologies, with applications in many fields: from gaming and spot (Pagliari and Pinto, 2015; Cenedese et al., 2016b) to health care (Sathiyanarayanan and Rajan, 2016), from home automation (Cenedese et al., 2015) to industry 4.0 and robotics (Lim et al., 2017; Liu and Wang, 2018). It is important in this context to differentiate between *activities* and *gestures*. Gestures are here considered as basis movements that compose an activity that is conversely completely characterized by one or more gestures: for example, in swimming, a stroke (gesture) completely characterize the style (activity) (Terzi et al., 2017). Two types of scenarios are typically envisioned in this context: (i) the wearable scenario (Cenedese et al., 2016a) where the human is equipped with sensors, like Inertial Measurement Units (IMUs); (ii) the smart environment (Várkonyi-Kóczy and Tusor, 2011) scenario, where the environment where human recognition takes place is equipped with sensors, without the human having to necessary carry sensors on his/her body.

This work is focused on *Gesture Recognition (GR)*, that is generally tackled as a classification task, where a set of predefined gestures has to be classified. GR is particularly relevant in smart environment applications (such as home automation (Belgioioso et al., 2014)), where gestures are associated with commands to be given to the smart environment. The contributions of this paper are: (i) we introduce a new GR dataset which, unlike many other datasets, is based on Motion Capture data, which comprise 3D world positions. We would like to stress that 3D positions trajectories allow to apply more interpretable data visualization techniques. Moreover, from raw positions, it is possible to retrieve accelerations and velocity data, commonly used in wearable applications. This is crucial in order to study the impact of loss of information using IMU data with respect to fully informative data; (ii) a pipeline for GR where: (i) we compare different approaches for feature extraction that are able to handle gestures of different duration, (ii) we employ Random Forest for classification.

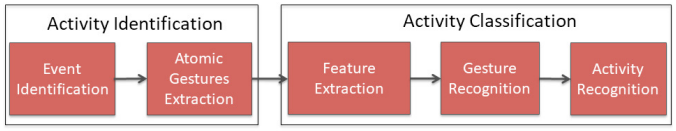

Fig. 1. Scheme of the classical ML approach to AR/GR problems.

The paper is structured as follows: Section 2 is dedicated to review recent works on GR, while in Section 3 the procedure to create the dataset is described. In Section 4 we introduce our approach for the feature extraction and GR phases and in Section 5 we present the results of our experiments. Finally, we draw the conclusions in Section 6.

2. LITERATURE REVIEW

In recent years, GR/Activity Recognition (AR) problems have been tackled using Machine Learning (ML) tech-

10.1016/j.ifacol.2019.09.129

niques under the main area of time series classification, possibly supervised (Terzi et al., 2017; Cenedese et al., 2016b). Generally, one of the main challenges one encounter when applying such approach is to translate the raw time series into a format that can be used by ML classifiers; typically, the general scheme is given in Fig. 1: the pipeline contains two blocks, atomic gestures extraction and feature extraction, aiming to translate the informative content in the classical form $X \in \mathbb{R}^{N \times p}$, where N and p are, respectively, the number of observations (executions of atomic gestures in this context) and the number of features extracted from each atomic gesture execution.
This is also the approach used in this paper, where the atomic gestures extraction phase is performed on time series containing multiple executions of the same gesture and the classifiers employed are the well-known Random Forests (RFs) (Breiman, 2001), which guarantee low bias and low variance estimates. In addition to feature-based approaches, there exist other similar approaches such as distance-based ones. For a complete review on time series classification, see (Susto et al., 2018).
In the last five years, Deep Learning (DL) proved to obtain state-of-the-art results for many tasks, like Computer Vision and it has been successfully applied also to GR problems (Yang et al., 2015; Hammerla et al., 2016; Ordóñez and Roggen, 2016). The common DL pipeline follows Fig.1 where the features extraction phase can be embedded with the Neural Network architecture, allowing a plug-and-play approach to modeling. Although DL approaches are able to achieve high accuracy and to overcome a dedicated feature extraction phase, this is obtained at the cost of lack of robustness and interpretability, and the need of a large number of training data. When the dataset cardinality is not big (a typical situation in GR and many other real-world problems), RFs represent a valuable alternative for classification tasks, and in particular to GR as shown by (Negin et al., 2013; Camgöz et al., 2014).

3. DATASET CREATION

In this Section we describe in detail the apparatus, the data acquisition and elaboration procedures exploited to build the GR dataset.

3.1 Experimental setup

Data acquisition sessions took place in the Multi AGent & Motion Analysis and Gait Intelligent Control (MAG^2IC) laboratory of the Department of Information Engineering at the University of Padova. The laboratory is equipped with a BTS Smart-D Motion Capture system (MAGIC, 2019), composed of 12 infrared video cameras which collect three-dimensional position data of 8 photo-reflexive markers. At a given sampling instant, for each marker the Motion Capture system acquires the x-component, the y-component and the z-component of the marker position. As a result, at each sampling instant we have a total of 24 acquired values. The cameras sampling rate is 340 Hz .
The 8 markers have been placed in areas of the body that allow a better characterization of the gesture, with the future goal of making it possible data collection in everyday applications, using consumer wearable devices. Specifically, as shown in Fig. 2, the markers locations are: on left and right wrists (smartwatch location); on left and right arms (running band location); on left and right legs (smartphone location, inside the pocket); on left and right ankles (other running sensors location).
In this work we analyzed gestures performed statically (i.e. standing still in the same point of the space) and through the use of the right arm only (on the frontal plane), therefore measurements from markers other than those placed on the right arm and wrist are not particularly informative. However, we decided to include in the dataset also such measures, in the eventuality that they could be useful elements of comparison, in classification phase, for possible future work on the recognition of dynamically executed gestures (walking or running) or using both arms.

3.2 Data acquisition

The data acquisition process was divided into several sessions, each of which containing the measurements relating to multiple executions of gestures belonging to a single category. The categories of gestures for which data were collected are 10: circle shape ('◯'), eight shape ('8'), square shape ('□'), triangle shape ('△'), M shape ('M'), S shape ('S'), U shape ('U'), V shape ('V'), vertical movement ('⇂') and horizontal movement ('⥋'). With the purpose of providing greater intra-class variability, for each gesture category we considered 11 different modes of execution: clockwise (CK), counterclockwise (CCK), small amplitude (SA), medium amplitude (MA), big amplitude (BA), low velocity (LV), medium velocity (MV), high velocity (HV), vertical deformation (VD), horizontal deformation (HD) and diagonal deformation (DD). Notice that there are some incompatible gesture category-mode pairs: for example, for the M shape gesture the CW and CCW modes have no sense.
In order to create an automated procedure for the identification of markers, during the Motion Capture system calibration phase we positioned the coordinate reference system at the center of the room, oriented towards the direction in which the gestures were performed. The calibration thus obtained guarantees that, once the subject

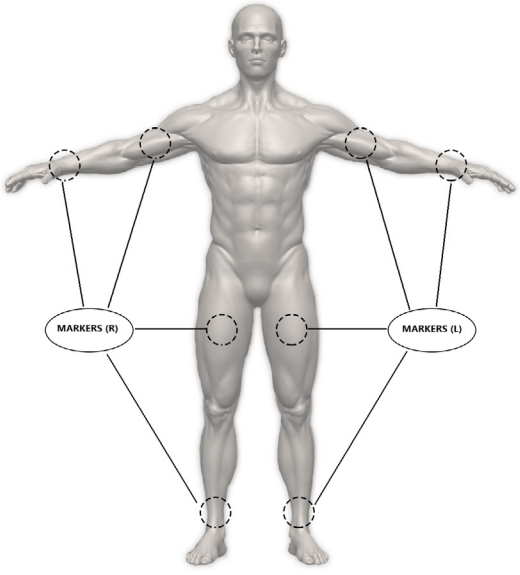

Fig. 2. Markers locations.

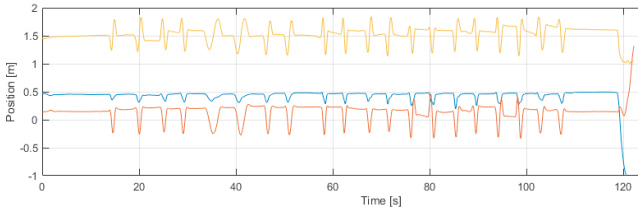

Fig. 3. Time series associated with the x-component (blue), the y-component (yellow) and the z-component (orange) of the right wrist marker position when performing the circle shape (medium amplitude).

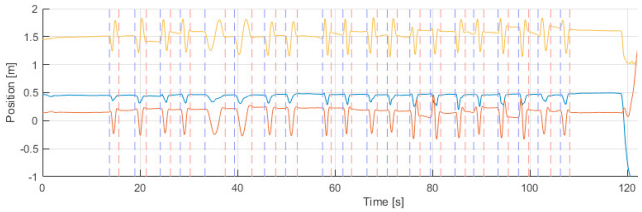

Fig. 4. Extraction of single executions by detecting start points (blue dashed vertical lines) and stop points (red dashed vertical lines).

has assumed the correct position in the preparation phase, each time series captured by the system can be uniquely associated with the corresponding marker by the simple analysis of the marker position at the start time. Indeed, at the start time markers positioned on opposite sides (right or left) differ at least in the z-component which is negative or positive, respectively, while markers positioned at different locations on the same side differ at least in the y-component (i.e. their height).
As mentioned above, each session includes multiple executions of the same gesture and, thus, we had to organize the data acquisition procedure in a way that facilitates the subsequent segmentation of the measurements into the single executions. Therefore, for each feasible gesture category-mode pair the data acquisition procedure is based on the following steps:

(1) Start the acquisition of the Motion Capture system.
(2) Quiet phase (10-12 seconds) for the identification of the *start* and the markers.
(3) Execution of individual gestures (separated by pauses of 2-3 seconds).
(4) Quiet phase (10-12 seconds) for the identification of the *stop*.
(5) Interruption of the acquisition.

An example of acquisition is given in Fig. 3, where are reported the time series associated with the x-component (blue), the y-component (yellow) and the z-component (orange) of the right wrist marker position when performing the circle shape (medium amplitude).

3.3 Atomic gesture extraction

Given the synchronous nature of the collected data, in order to identify the periods of activity it is sufficient to analyze the most informative time series, i.e. those associated with the marker positioned on the right wrist (with which the gestures were performed). The hypothesis that allows the identification of the temporal interval in which a gesture is performed is the presence of a period of immobility before and after each execution (indicatively set to 2-3 seconds, as anticipated in Section 3.2). In order to identify such activity intervals, we considered overlapping windows on the time series. By evaluating the standard deviation within each window, we can detect for each single execution the start point and the stop point, as can be seen in Fig. 4 (blue and red dashed vertical lines, respectively). Notice that time series describing single executions may have different lengths, even if we consider the same gesture category and mode. This aspect must be given particular attention in model design phase as not all classification algorithms can deal natively with inputs of variable size. In Tab. 1 the number of single executions obtained for each gesture category-mode pair is reported. The total number of observations is 1537. The dataset is available at `https://gitlab.dei.unipd.it/dl_dei/gesture_recognition`.

4. PROPOSED APPROACH

In this Section we describe the three feature extraction methods that we used and compared in our experiments and we recall the basic functioning of Random Forests, the classifiers used in the GR phase.

4.1 Feature extraction

As anticipated in Section 3.3, the time series describing single executions obtained through the atomic gesture extraction procedure may contain a variable number of samples, hence we need to perform a feature extraction phase before training a classification model. The goal of the feature extraction procedure is to describe each single execution with a feature vector x_i of fixed dimension p, so that a design matrix X with N rows and p columns can be created (where N is the number of single executions). Below we describe the three feature extraction methods that we used and compared in our experiments.
- *Time series cropping* (CROP). We denote with L_{min} the minimum time series length, which coincides with the number of samples contained in the shortest time series of the dataset. This method consists in truncating time series whose length exceeds the minimum length L_{min}. As a result, each single execution is described with a feature vector of dimension $p = 3 \cdot L_{min}$ for each marker (i.e. a vector containing the first L_{min} samples of the x-component, y-component and z-component, respectively). An example with $L_{min} = 100$ is given in Fig. 5 (middle), where blue samples are used as features, while orange ones are discarded.
- *Time series resampling* (RESAMP). Differently from the previous method, in which the same sample density is preserved at the price of not finishing the slowest executions, the resampling procedure consists in preserving the overall shape of the time series and discarding some intermediate samples. In particular, the resampling period depends on the length of the time series at hand: the longer the time series, the bigger the resampling period. Similarly to the previous method, the resampling procedure guarantees that all time series contain the same number of samples, equal to L_{min}, and each single execution is described with a feature vector of dimension $p = 3 \cdot L_{min}$ for each

Table 1. Number of single executions for each gesture category-mode pair.

	CW	CCW	SA	MA	BA	LV	MV	HV	VD	HD	DD	**TOT**
◯	18	20	20	20	20	20	20	20	18	18	20	**214**
8	20	20	18	20	18	18	18	18	18	16	16	**200**
△	16	16	14	14	14	-	-	-	16	16	-	**106**
□	15	16	14	14	14	14	14	14	16	16	-	**147**
M	-	-	14	14	14	14	14	14	10	12	24	**130**
S	-	-	14	14	14	14	14	14	10	12	24	**130**
U	-	-	14	16	14	14	14	14	12	12	24	**134**
V	-	-	14	14	14	12	14	14	12	12	24	**130**
⇋	-	-	20	22	20	22	22	20	-	-	50	**176**
⇅	-	-	20	22	20	22	18	20	-	-	48	**170**

marker. An example is given in Fig. 5 (bottom), where blue small vertical lines indicate the samples selected by the resampling method.

- *Summary statistics* (STAT). This method consists in summarizing the information contained in each time series through a predetermined number of functions of the data (usually referred to as *statistics*). In this work we decided to represent each time series with 5 statistics: minimum value, maximum value, sample mean, sample variance and energy. As a result, each single execution is described with a feature vector of dimension $p = 3{\cdot}5 = 15$ for each marker.

4.2 Random Forest

RFs are high precision statistical modeling approaches which have been shown to be able to outperform many algorithms for both regression and classification. RF is an ensemble method that deploys in parallel B Decision Trees (DTs), a classical approach to statistical learning that, in its naive form, is generally prone to overfitting and poor prediction capabilities with high-dimensional modelling problems; RF corrects for decision trees aforementioned problems. We denote with $X \in \mathbb{R}^{N\times p}$ the design matrix

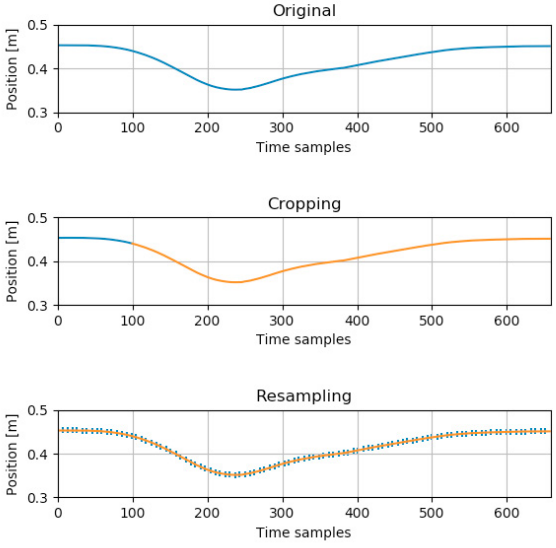

Fig. 5. Time series cropping (middle) and resampling (bottom) of the original x-component (top) of the right wrist marker position when performing the circle shape (medium amplitude). Blue sample are used as features, orange samples are discarded.

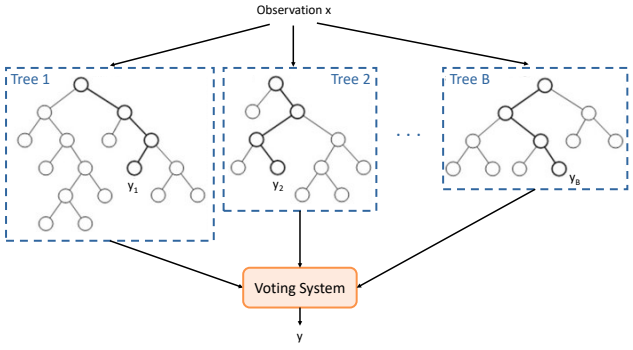

Fig. 6. Random Forest (RF) scheme with B decision trees. Figure adapted from Susto (2017).

and with $Z \in \mathbb{R}^N$ the vector containing the target variables that we want to predict. In the case of a classification task Z contains the classes associated to each observation, while in the case of a regression task Z typically contains continuous values.

A scheme of a RF is depicted in Fig. 6; for a new observation, whose input data are contained in the vector x, the k-th tree assigns a class y_k (in the case of a classification RF) or a value y_k (in the case of a regression RF). The RF output is decided as follows: in the case of a regression RF, the output value y is the average over $\mathcal{Y} = \{y_1, \dots, y_B\}$, while in the case of a classification RF, the output y is chosen as the mode of the set $\mathcal{Y}$. The B trees are constructed by exploiting a procedure called Bootstrap Aggregating or Bagging (Dietterich, 2000). With the bagging procedure, B random samples with replacement of the training data are considered and a DT is fit to each sample; for $b = 1, \dots, B$

(1) a random sample $[X_b, Z_b]$ with replacement of the training data is considered;
(2) a DT $f_b(\cdot)$ is computed on $[X_b, Z_b]$.

In the case of a classification problem, for a new observation x the corresponding RF output (i.e. the predicted class) is then computed as the mode of $\mathcal{Y} = \{y_1, \dots, y_B\}$, where $y_b = f_b(x)$ for $b = 1, \dots, B$.

5. EXPERIMENTAL RESULTS

In this Section we describe the two experiments carried out on the GR dataset described in Section 3 along with some considerations on the obtained results. In order to reduce the computational cost, we only exploited measurements associated to the most informative markers, i.e. those located on the right arm and right wrist, respectively.

Table 2. Experiment #1: accuracy and computational time average (standard deviation).

	CROP	RESAMP	STAT
Accuracy	0.93 ± 0.02	$\mathbf{0.97 \pm 0.01}$	0.96 ± 0.02
Comp. time	0.25 ± 0.01	0.22 ± 0.01	$\mathbf{0.07 \pm 0.004}$

Table 3. Cumulative confusion matrix for experiment #1 with CROP method.

		PRED ○	PRED 8
TRUE	○	6054	356
	8	502	5588

Table 4. Cumulative confusion matrix for experiment #1 with RESAMP method.

		PRED ○	PRED 8
TRUE	○	6270	140
	8	177	5913

For both experiments we performed 100 random iterations of the RF algorithm (with 100 DTs) for each feature extraction method. For the evaluation of the models we adopted a 70%-30% train-test split and computed classification accuracy (averaged over the 100 iterations) and cumulative confusion matrix on test set.

5.1 Experiment #1

For the first experiment we considered a reduced version of the GR dataset, which comprises only executions of circle shape and eight shape with a total number of available observations equal to 414. The shortest time series has length $L_{min} = 317$, therefore CROP and RESAMP methods represent each single execution with a feature vector of dimension $p = L_{min} \cdot n_{comp} \cdot n_{marker} = 317 \cdot 3 \cdot 2 = 1902$, where n_{comp} denotes the number of components of each measurement and n_{marker} the number of markers considered. The STAT method, instead, represents each execution with a feature vector of dimension $p = 5 \cdot n_{comp} \cdot n_{marker} = 5 \cdot 3 \cdot 2 = 30$ and it can be noticed that the value of p, in this case, is independent of the shortest time series length. As can be seen in Table 2, the more compact representation provided by STAT method allows to reduce the training time if compared to CROP and RESAMP. However, the best classification accuracy is achieved with the RESAMP method, suggesting that the representation given by STAT can be enriched by including more informative additional statistics. The CROP method gives the worst performance and this may be due to the fact that relevant information from the original signal is discarded, especially when processing slow gestures. By inspecting the cumulative confusion matrices in Table 3, Table 4 and Table 5, it can be noticed that the classification accuracy for the circle shape is slightly higher than for the eight shape for all feature extraction methods. This difference is further emphasised for the CROP method, i.e. the method with the worst performance.

Table 5. Cumulative confusion matrix for experiment #1 with STAT method.

		PRED ○	PRED 8
TRUE	○	6151	259
	8	292	5798

Table 6. Experiment #2: accuracy and computational time average (standard deviation).

	CROP	RESAMP	STAT
Accuracy	0.79 ± 0.02	$\mathbf{0.94 \pm 0.01}$	0.84 ± 0.02
Comp. time	1.40 ± 0.08	1.25 ± 0.06	$\mathbf{0.31 \pm 0.03}$

Table 7. Cumulative confusion matrix for experiment #2 with CROP method.

		PRED ○	8	△	□	M	S	U	V	⇋	↑↓
TRUE	○	5515	397	7	77	43	121	179	3	62	52
	8	368	5328	0	0	0	59	37	24	74	95
	△	10	0	2740	377	1	59	0	0	33	11
	□	109	0	310	3472	388	18	0	0	25	30
	M	67	0	5	290	3292	86	0	0	39	112
	S	427	71	29	37	140	2464	211	26	288	248
	U	223	4	0	0	0	139	2631	689	173	203
	V	115	37	0	0	0	89	798	2172	239	414
	⇋	110	17	119	86	18	58	28	204	4591	46
	↑↓	2	68	0	0	53	187	121	431	80	4199

Table 8. Cumulative confusion matrix for experiment #2 with RESAMP method.

		PRED ○	8	△	□	M	S	U	V	⇋	↑↓
TRUE	○	6231	100	0	23	5	13	60	2	22	0
	8	102	5793	0	0	0	0	0	1	14	75
	△	0	0	2983	226	13	0	0	0	9	0
	□	28	0	87	4173	10	30	0	0	23	1
	M	0	0	39	23	3829	0	0	0	0	0
	S	47	56	0	3	0	3523	1	0	114	197
	U	20	0	0	0	0	9	3566	465	2	0
	V	0	0	0	0	0	0	451	3392	21	0
	⇋	1	0	16	0	20	44	1	27	5112	56
	↑↓	0	0	0	0	0	118	0	0	36	4987

Table 9. Cumulative confusion matrix for experiment #2 with STAT method.

		PRED ○	8	△	□	M	S	U	V	⇋	↑↓
TRUE	○	5950	144	19	83	54	61	131	2	0	12
	8	191	5587	31	1	0	79	38	18	0	40
	△	50	26	2867	200	2	25	36	10	15	0
	□	281	22	144	3648	214	30	13	0	0	0
	M	330	2	16	55	3278	57	136	17	0	0
	S	432	574	209	221	159	1778	226	175	3	164
	U	61	281	76	56	145	85	2887	354	28	89
	V	38	227	30	0	8	58	321	2770	226	186
	⇋	2	5	3	3	11	22	3	52	5176	0
	↑↓	0	2	0	33	0	68	89	59	10	4880

5.2 Experiment #2

For the second experiment we considered the whole GR dataset, which consists of 1537 observations. The shortest time series has length $L_{min} = 167$, therefore CROP and RESAMP methods represent each single execution with a feature vector of dimension $p = L_{min} \cdot n_{comp} \cdot n_{marker} = 167 \cdot 3 \cdot 2 = 1002$. The analysis of the results in Table 6 leads to considerations analogous to those given in Section 5.1 for experiment #1.

When considering the whole dataset it is worth highlighting the value of the information contained in the cumulative confusion matrices given in Table 7, 8 and 9. Indeed,

in the context of smart environments each gesture corresponds to a specific command and this correspondence is typically designed according to two principles:

- most used commands are associated with the most ergonomic gestures;
- critical commands (i.e. those that cannot be failed) are associated with gestures with the lowest false positive/false negative rates.

It is evident that the overall classification accuracy does not provide enough information for this purpose and the knowledge about the distribution of misclassified gestures is of fundamental importance for the design of a trustworthy and robust system. For example, if we consider the cumulative confusion matrix given in Table 8 (associated with the best performing feature extraction method, RESAMP) we would associate a critical command to shape M (low false negative and false positive rates) rather than shape U (high false negative and false positive rates).

6. CONCLUSION

In this paper we presented an algorithm for Gesture Recognition of hand movements based on Random Forests: we showed this classifier combined with a proper feature extractor can be an effective approach to smart environment applications. In the problem at hand, the RESAMP method experimentally proved to be an effective choice for extracting features in term of accuracy at small price of time complexity with respect to the STAT method. Moreover, we published the dataset we used as a benchmark in order to foster research in the GR area.

In future works, we plan to augment the GR dataset with IMU data and analyze the corresponding loss of information with respect to Motion Capture data. We also plan to extend the approach proposed in this paper to the recognition of dynamically executed gestures (e.g. walking, running) and gestures executed with both arms.

REFERENCES

Belgioioso, G., Cenedese, A., Cirillo, G.I., Fraccaroli, F., and Susto, G.A. (2014). A machine learning based approach for gesture recognition from inertial measurements. In *53rd IEEE Conference on Decision and Control*, 4899–4904. IEEE.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5–32.

Camgöz, N.C., Kindiroglu, A.A., and Akarun, L. (2014). Gesture recognition using template based random forest classifiers. In *ECCV Workshops*.

Cenedese, A., Minetto, L., Susto, G.A., and Terzi, M. (2016a). Human activity recognition with wearable devices: A symbolic approach. *PsychNology Journal*, 14(2).

Cenedese, A., Susto, G.A., Belgioioso, G., Cirillo, G.I., and Fraccaroli, F. (2015). Home automation oriented gesture classification from inertial measurements. *IEEE Transactions on automation science and engineering*, 12(4), 1200–1210.

Cenedese, A., Susto, G.A., and Terzi, M. (2016b). A parsimonious approach for activity recognition with wearable devices: an application to cross-country skiing. In *2016 European Control Conference (ECC)*, 2541–2546. IEEE.

Dietterich, T.G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, 40(2), 139–157.

Hammerla, N.Y., Halloran, S., and Plötz, T. (2016). Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv preprint arXiv:1604.08880*.

Lim, G.H., Pedrosa, E., Amaral, F., Lau, N., Pereira, A., Dias, P., Azevedo, J.L., Cunha, B., and Reis, L.P. (2017). Rich and robust human-robot interaction on gesture recognition for assembly tasks. In *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 159–164. IEEE.

Liu, H. and Wang, L. (2018). Gesture recognition for human-robot collaboration: A review. *International Journal of Industrial Ergonomics*, 68, 355–367.

MAGIC (2019). Multi Agent Motion Analysis and Gait Intelligent Control. `https://sparcs.dei.unipd.it/index.php/laboratories/magic/`. [Online; accessed 29-April-2019].

Negin, F., zdemir, F., Yksel, K.A., Akgl, C.B., and Eril, A. (2013). A decision forest based feature selection framework for action recognition from rgb-depth cameras. In *2013 21st Signal Processing and Communications Applications Conference (SIU)*, 1–4. doi:10.1109/SIU.2013.6531398.

Ordóñez, F. and Roggen, D. (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1), 115.

Pagliari, D. and Pinto, L. (2015). Calibration of kinect for xbox one and comparison between the two generations of microsoft sensors. *Sensors*, 15(11), 27569–27589.

Sathiyanarayanan, M. and Rajan, S. (2016). Myo armband for physiotherapy healthcare: A case study using gesture recognition application. In *2016 8th International Conference on Communication Systems and Networks (COMSNETS)*, 1–6. IEEE.

Susto, G.A. (2017). A dynamic sampling strategy based on confidence level of virtual metrology predictions. In *2017 28th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, 78–83. IEEE.

Susto, G.A., Cenedese, A., and Terzi, M. (2018). Time-series classification methods: review and applications to power systems data. In *Big data application in power systems*, 179–220. Elsevier.

Terzi, M., Cenedese, A., and Susto, G. (2017). A multivariate symbolic approach to activity recognition for wearable applications. *IFAC-PapersOnLine*, 50(1), 15865–15870.

Várkonyi-Kóczy, A.R. and Tusor, B. (2011). Human–computer interaction for smart environment applications using fuzzy hand posture and gesture models. *IEEE Transactions on Instrumentation and Measurement*, 60(5), 1505–1514.

Yang, J., Nguyen, M.N., San, P.P., Li, X.L., and Krishnaswamy, S. (2015). Deep convolutional neural networks on multichannel time series for human activity recognition. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.