Hand-crafted Features vs Residual Networks for Human Activities Recognition using Accelerometer

Anna Ferrari, Daniela Micucci, Marco Mobilio and Paolo Napoletano
Department of Informatics, Systems, and Communication
University of Milano - Bicocca, Milan, Italy
anna.ferrari@disco.unimib.it,{daniela.micucci,marco.mobilio,paolo.napoletano}@unimib.it

***Abstract*—Inertial sensors combined with supervised machine learning techniques are largely employed for automatic Human Activity Recognition (HAR). Machine learning scientists made available to the community a plenty of labeled datasets that permit, especially in the recent years, to develop sophisticated techniques, such the ones based on deep learning. These techniques have recently become very popular because they are highly accurate. Nevertheless, some researchers still use the combination of traditional classifiers, such as SVM and $k-$NN, with hand-crafted features or raw signals.**

The aim of this paper is to investigate the robustness of traditional classifiers combined with hand-crafted features compared with an end-to-end deep learning solution based on a Residual Network. Experiments on four public datasets are presented and discussed.

***Index Terms*—Inertial sensors; Machine learning; Deep Learning; Human Activity Recognition;**

I. Introduction and Background

Nowadays, commercial fitness bracelets, smartwatches, and smartphones are equipped with inertial sensors, such as accelerometer and gyroscope. Such sensors acquire signals that are exploited by machine learning methods for automatic Human Activity Recognition (HAR). HAR has many applications in several domains such as, for example, healthcare, sport, and entertainment.

During the last decade, a plenty of traditional machine learning as well as deep learning methods have been proposed in literature [1]–[5]. In the recent literature, deep learning methods are predominant [5]. Deep learning methods require a special hardware setup (Graphical Processing Units - GPUs) to speed up computation and a great amount of data to avoid overfitting during the training process. However, it is very rare to find consumer hardware equipped with GPUs, thus in most cases, deep learning methods run on cloud platform, such as, Google Cloud[1], Amazon AWS[2], and Microsoft Azure[3]. Large scale inertial datasets with millions of signals recorded by hundreds of human subjects are still not available, but several smaller datasets made of thousands of signals and dozens of human subjects are publicly available [6]. A recent platform to support long-term data collection of inertial signals have been proposed [7], [8] with the scope to make available to the scientific community a large dataset enriched with context information (e.g., characteristics of the subject, device position etc.). Moreover, since the public available datasets for HAR benchmarking are not consistent, both syntactically (e.g., different sampling frequency) and semantically (e.g., labels with different meanings), some researchers have proposed a platform for data integration [6] as well as methods for data homogenization [9].

Since to date, large scale inertial datasets are not available, it is therefore not obvious in this domain, which method between deep and traditional machine learning methods is the most appropriate, especially in those case where the hardware is low cost.

The aim of this paper is to investigate the robustness of traditional classifiers combined with hand-crafted features compared with an end-to-end deep learning solution based on a Residual Network that is one of the most performing network in the state of the art [10]. Experiments on four public datasets are presented and discussed.

[1]https://cloud.google.com
[2]https://aws.amazon.com/
[3]https://azure.microsoft.com

II. Material and Methods

A. Datasets

We experimented four datasets:

- *UCI HAR* [11], which includes tri-axial acceleration and gyroscope data of 6 ADLs (Activities of Daily Living) recorded with a Samsung Galaxy S II and performed by 30 volunteers.
- *MobiAct* [12], which includes tri-axial acceleration, gyroscope, and orientation data of 11 ADLs and 4 Falls recorded with a Samsung Galaxy S3 and performed by 67 volunteers.
- *Motion Sense* [13], which includes tri-axial acceleration and gyroscope data of 6 ADLs recorded with an iPhone 6s and performed by 30 volunteers.
- *UniMiB-SHAR* [14], which includes tri-axial acceleration data of 17 ADLs recorded with an Samsung Galaxy Nexus I9250 and performed by 24 volunteers.

Considering the acceleration only, each signal of the datasets is composed of three accelerometer components along the x, y, and z axis. Each signal component has been resampled at 50Hz and divided in segments of 2.56 seconds with an overlap between subsequent segments of 50% [4]. The resampling at 50Hz was necessary because the MobiAct dataset has been acquired at a frequency of about 87Hz. The resulting

978-1-7281-3570-0/19/$31.00 ©2019 IEEE

Authorized licensed use limited to: Arete. Downloaded on May 24,2020 at 18:56:02 UTC from IEEE Xplore. Restrictions apply.

TABLE I
NUMBER OF SEGMENTS AND CLASSES FOR EACH DATASET

Dataset	# train	# validation	# test	# classes
UCI HAR	7209	2060	1030	6
MobiAct	34070	9734	4867	15
Motion Sense	14945	4270	2135	6
UniMiB-SHAR	8240	2354	1177	17

segment for each axis contains 128 samples. The resampling and segmentation procedures were not applied to UniMiB-SHAR because such a dataset already contains overlapped segments of 151 samples. In fact, the dataset contains segments of 3 seconds sampled at 50Hz taken around a peak (higher than $1.5g$, with g being the gravitational acceleration) of the accelerometer signal. To be consistent with the other datasets, these segments were centrally windowed in order to obtain 128-dimensional segments.

Each dataset has been split in 70% training, 20% validation, and 10% test. Table I shows the total number of 128×3-dimensional segments available for the training (column *# train*), validation (column *# validation*), and test (column *# test*) sets. The last column *# classes* indicates the number of ADLs present in the dataset.

B. Hand-crafted features

For the experimentation of hand-crafted features, the k Nearest Neighbour (k-NN) and Support Vector Machines (SVM) classifiers have been used. The features used are:

- Raw data (denoted as *raw*): x,y, and z accelerometer segments (without any kind of processing) are concatenated and used as feature vectors [15];
- Magnitude of the segments (denoted as *magn*) [14];
- 21 features extracted from the magnitude of the segments (denoted as *hc magn*) [16]. Table II reports details about the 21 features.
- 21 features extracted from each of the three segments along the three axes x, y, and z (denoted as *hc raw*). The total number of features is 63.

In the case of SVM, the multi-class classifier has been implemented as multiple binary classifiers. We used the Statistics and Machine Learning Toolbox for k-NN and SVM. Optimum parameters of both classifiers have been found through cross-validation.

C. End-to-end deep learning solution

The Residual Network (ResNet) adopted for this study is based on the traditional architecture proposed by He *et al.* [17], which demonstrated to be very effective on the ILSVRC 2015 (ImageNet Large Scale Visual Recognition Challenge) validation set with a top 1- recognition accuracy of about 80%. Residual architectures are based on the idea that each layer of the network learns residual functions with reference to the layer inputs instead of learning unreferenced functions. He *et al.* [17] demonstrate that such architectures is easier to optimize and it gains accuracy also when the depth increase considerably.

TABLE II
HAND-CRAFTED FEATURES

Features	
Minimum	$min = \min_{j=1,...n}(x_j)$
Maximum	$max = \max_{j=1,...n}(x_j)$
Mean	$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$
Median	$Me = x_{0.5} : F(x_{0.5}) = 0.5$
Standard Deviation (SD)	$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$
Variance	$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$
Fourth central moment	$m_4 = \frac{1}{n}\sum_{j=1}^{n}(x - \bar{x})^4$
Fifth central moment	$m_5 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^5$
Skewness	$S = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{s^3}$
Kurtosis	$K = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4}{s^4}$
Root Mean Square (RMS)	$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$
Interquartile Difference	$ID = x_{0.75} - x_{0.25}$
Total Sum	$TS = \sum_{i=1}^{n} x_i$
Range	$R = max - min$
Entropy	$H(x) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)$
SD of the intervals between two successive peaks	$SDNN = \sqrt{\frac{1}{n-1}\sum_{j=1}^{n}(PP_j - \overline{PP})^2}$
RMS of the differences between two successive peaks	$RMSSD = \sqrt{\frac{1}{n-1}\sum_{j=1}^{n-1}(PP_{j+1} - PP_j)^2}$
Number of pairs of successive peaks intervals that differ by more than 50 ms	$pNN50 = p(\lvert PP_{j+1} - PP_j\rvert > 50)$
Sum of the spectral power components	$SP = \frac{1}{n}\sum_{j=1}^{f} \lvert FFT_j\rvert^2$
Mean of the spectral components	$\mu_f = \frac{1}{n}\sum_{j=1}^{n} FFT_j$
Median of the spectral components	$Me_f = FFT_{0.5} : F(f_{0.5}) = 0.5$

TABLE III
RESIDUAL NETWORK ARCHITECTURE

Layer name	shape
conv1	$\{1\times3\} \times n$
conv2_n	$\{1\times3 \times f_{maps}\} \times n$
conv3_n	$\{1\times3 \times 2f_{maps}\} \times n$
conv4_n	$\{1\times3 \times 4f_{maps}\} \times n$
avg_pool_x	1×32
fully conn.	$(1 \times 4f_{maps}) \times 15$
softmax	1×15

Table III details the network architecture proposed for this study. The input size of the network is $1 \times 128 \times 3$, that corresponds to 3 segments along the three axes x, y, and z. The network architecture is made of an initial convolutional block, 3 residual stages, each containing a variable number n of residual blocks, average pooling layer, fully connected layer, and softmax layer. A convolutional block is made of three layers: convolutional, batch normalization, and ReLu. A residual block is made of 2 subsequent convolutional blocks and an addition operator that sums the input of the residual block with the output of the residual block itself. Each convolutional layer is $1 \times 3 \times f_{maps}$, where f_{maps} is the number of feature maps of the filter. For each dataset, the best values for n and f_{maps} have been found by following a grid search approach: n ranged between 3 and 21, while f_{maps} ranged between 10 and 200.

Figure 1 shows the best network for UCI-HAR, obtained with $n = 1$ and $f_{maps} = 90$. For all the datasets, the networks have been optimized through the Stochastic Gradient Descent

Authorized licensed use limited to: Arete. Downloaded on May 24,2020 at 18:56:02 UTC from IEEE Xplore. Restrictions apply.

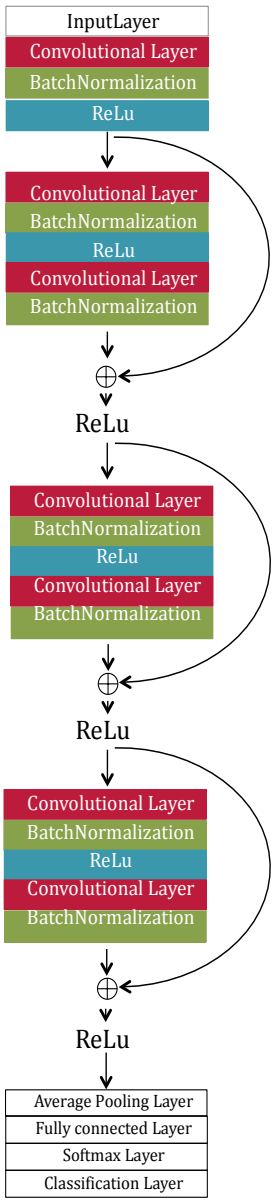

Fig. 1. Residual Neural Network architecture of the UCI-HAR dataset.

with Momentum (SGDM), using a piecewise learning update strategy with an initial value of 0.1 and a drop factor of 0.1. The batch size was 128, the total number of epochs was 80 and the early stopping has been used to avoid overfitting. We used the Matlab Deep Learning Toolbox for training and testing the Residual

III. Experiments

Tables IV and V show results achieved by all the methods considered in terms of macro average accuracy (i.e., the average of each class accuracy). The accuracy of each class is computed as ratio between the number of segments correctly classified and the total number of segments of that class. ResNet achieves better performance than traditional methods in all datasets apart from MobiAct. Most important, the standard deviation of the ResNet method is close to zero. Accuracy of k-NN and SVM is quite similar, while among hand-crafted features the most performing is the *hc raw*.

Overall, ResNet is the best performing with an average accuracy across datasets of 92.94%, the second best across classifiers and datasets are the *hc raw* features with an average accuracy of 79.18%. The third best are the *raw* features with an average accuracy of 76.04%. The worst are the *magnitude* and *magnitude raw* features with an average accuracy of 62.57% and 61.81% respectively.

In summary, the average gap between hand-crafted features combined with traditional classifiers and deep learning is about 15%. This experimentation actually confirms that deep learning outperforms traditional machine learning approaches.

IV. Conclusions

This paper presented a comparison between traditional classifiers combined with hand-crafted features and an end-to-end deep learning solution based on a Residual Network. Experiments on four public datasets demonstrated that overall deep learning solutions overcome the state of the art, thus suggesting that, even when the large scale datasets are not available, this techniques on average perform better than traditional machine learning approaches. However, hand-crafted features may be preferable in those cases where the hardware is low cost and thus does not permit deep learning solutions to run in a short time.

As future work, it would be interesting to evaluate the robustness of deep learning techniques across position of the wearable device, age, height, and gender of the human subject.

References

[1] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[2] F. J. Ordonez, G. Englebienne, P. De Toledo, T. Van Kasteren, A. Sanchis, and B. Kröse, "In-home activity recognition: Bayesian inference for hidden markov models," *IEEE Pervasive Computing*, vol. 13, no. 3, pp. 67–75, 2014.

[3] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 15)*, 2015.

[4] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert systems with applications*, vol. 59, pp. 235–244, 2016.

[5] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[6] P. Siirtola, H. Koskimäki, and J. Röning, "Openhar: A matlab toolbox for easy access to publicly open human activity data sets," in *Proceedings of the ACM International Joint Conference and International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers (UbiComp18)*, 2018.

[7] A. Ferrari, D. Micucci, M. Marco, and P. Napoletano, "A framework for long-term data collection to support automatic human activity recognition," in *Proceedings of Intelligent Environments: Workshop on Reliable Intelligent Environment (IE 19)*, 2019.

[8] D. Ginelli, D. Micucci, M. Mobilio, and P. Napoletano, "UniMiB AAL: An Android Sensor Data Acquisition and Labeling Suite," *Applied Sciences*, vol. 8, no. 8, 2018.

Authorized licensed use limited to: Arete. Downloaded on May 24,2020 at 18:56:02 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
EXPERIMENTAL RESULTS - MEAN CLASS ACCURACY(STANDARD DEVIATION CLASS ACCURACY): SVM VS RESNET

	SVM				**ResNet**
Dataset	raw	magn	hc raw	hc magn	
UCI-HAR	79.51 (± 17.40)	53.10 (± 25.48)	79.47 (± 20.59)	48.45 (± 22.12)	**90.73 (± 10.92)**
MobiAct	77.93 (± 22.71)	63.63 (± 24.13)	76.73 (± 26.11)	59.95 (± 23.94)	**92.98 (± 8.65)**
MotionSense	90.04 (± 14.36)	78.22 (± 29.59)	96.39 (± 3.79)	83.45 (± 21.13)	**99.47 (± 0.87)**
UniMiB-SHAR	58.26 (± 16.85)	52.27 (± 18.10)	58.08 (± 16.70)	50.81 (± 15.49)	**88.59 (± 8.52)**

TABLE V
EXPERIMENTAL RESULTS - MEAN CLASS ACCURACY(STANDARD DEVIATION CLASS ACCURACY): k-NN VS RESNET

	k-NN				**ResNet**
Dataset	raw	magn	hc raw	hc magn	
UCI-HAR	73.71 (± 26.78)	46.92 (± 29.89)	69.35 (± 17.04)	37.75 (± 13.39)	**90.73 (± 10.92)**
MobiAct	87.69 (± 9.07)	77.81 (± 13.60)	91.86 (± 6.72)	80.50 (± 10.74)	**92.98 (± 8.65)**
MotionSense	79.19 (± 31.83)	73.51 (± 25.16)	95.82 (± 5.61)	81.34 (± 20.30)	**99.47 (± 0.87)**
UniMiB-SHAR	61.97 (± 11.83)	55.13 (± 14.29)	65.74 (± 12.99)	52.22 (± 11.70)	**88.59 (± 8.52)**

[9] A. Ferrari, D. Micucci, M. Marco, and P. Napoletano, "On the homogenization of heterogeneous inertial-based databases for human activity recognition," in *Proceedings of IEEE SERVICES Workshop on Big Data for Public Health Policy Making*, 2019.

[10] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018.

[11] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones." in *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN13)*, 2013.

[12] G. Vavoulas, C. Chatzaki, T. Malliotakis, M. Pediaditis, and M. Tsiknakis, "The mobiact dataset: Recognition of activities of daily living using smartphones." in *Proceedings of Information and Communication Technologies for Ageing Well and e-Health (ICT4AgeingWell16)*, 2016.

[13] M. Malekzadeh, R. G. Clegg, A. Cavallaro, and H. Haddadi, "Protecting sensory data against sensitive inferences," in *Proceedings of the Workshop on Privacy by Design in Distributed Systems (W-P2DS18)*, 2018.

[14] D. Micucci, M. Mobilio, and P. Napoletano, "Unimib shar: A dataset for human activity recognition using acceleration data from smartphones," *Applied Sciences*, vol. 7, no. 10, p. 1101, 2017.

[15] D. Micucci, M. Mobilio, P. Napoletano, and F. Tisato, "Falls as anomalies? an experimental evaluation using smartphone accelerometer data," *Journal of Ambient Intelligence and Humanized Computing*, vol. 8, no. 1, pp. 87–99, 2017.

[16] S. Bianco, P. Napoletano, and R. Schettini, "Multimodal car driver stress recognition," in *Proceedings of the EAI International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth19)*, 2019.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)16)*, 2016, pp. 770–778.

Authorized licensed use limited to: Arete. Downloaded on May 24,2020 at 18:56:02 UTC from IEEE Xplore. Restrictions apply.