Polytechnica
https://doi.org/10.1007/s41050-021-00028-8

ORIGINAL ARTICLE

Human Activity Recognition from Accelerometer with Convolutional and Recurrent Neural Networks

M. K. Serrão[1] · G. de A. e Aquino[1] · M. G. F. Costa[1] · Cicero Ferreira Fernandes Costa Filho[1]

Received: 10 October 2020 / Revised: 14 December 2020 / Accepted: 16 January 2021

Abstract
Smartphones are present in most people's daily lives. Sensors embedded in these devices open the possibility of monitoring users' activities. The classification of the intricate data patterns collected through these sensors is a challenging task when considering hand-crafted features and pattern recognition algorithms. In this work, to face this challenge, we propose a convolutional neural network architecture along with two methods for transforming sensor data stream into images, and two recurrent neural networks, a long short time memory network and a gated recurrent unit network. The proposed model was evaluated using the UniMiB SHAR dataset. This dataset was acquired with accelerometers of mobile devices. The best macro average accuracy for classification of 17 types of activities, with 5-fold-cross-validation method, 95.49% was obtained with a gated recurrent unit network. The best macro average accuracy for classification with leave-one-subject-out method, 71.36%, was obtained with a convolutional neural network. These results are better than others previously published in the literature with the same dataset.

Keywords Human activity recognition · Convolutional neural networks · Accelerometer data

1 Introduction

Human Activity Recognition (HAR) is a widely explored field of study as it enables the development of applications in a great number of areas, such as medical (Subasi et al. 2018), security (Lisowska et al. 2018) and military (Park et al. 2016), among others. In medical area, patients with cognitive disorders can be monitored by applications that detect abnormal activities and assist in their safety and independence (Yin et al. 2008). Activity recognition is performed based on information received from different sensors (Yang et al. 2011). The monitoring of human activities has been favored by widespread use of smartphones and smartwatches, once these devices have inertial embedded sensors, such as accelerometers, magnetometers and gyroscopes.

Machine learning techniques, especially those known as deep learning, depend heavily on the availability of large training datasets in order to obtain the best results. The *University of Milano Bicocca Smartphone-based Human Activity Recognition* (UniMiB SHAR) dataset is a large human activity dataset that allows benchmark between different machine learning techniques (Micucci et al. 2017). This dataset was generated with acceleration samples from an Android smartphone and comprises 17 classes. Nine of them are activities of daily living (ADL) and 8, of fall types (FALL). Some works have been developed comparing the performance of several machine learning techniques in the UniMiB SHAR dataset [6–8].

Micucci et al. (2017) developed the UniMiB SHAR dataset and compared the performance of four different classifiers. For each classifier, for training and testing, the authors used 5-fold-cross-validation and leave-one-subject-out methods. The best result obtained by the authors for the 17 classes, using the cross-validation method, was a macro average accuracy of 82.86%. This result was obtained with the classifier k-Nearest Neighbor (KNN). The best result obtained with the leave-one-subject-out method was a macro average accuracy of 56.53%. This result was obtained with the Random Forest (RF) classifier.

Li et al. (2018) evaluated several classifiers and feature extraction methodologies, using OPPORTUNITY and

✉ Cicero Ferreira Fernandes Costa Filho
ccosta@ufam.edu.br

[1] Centro de P&D de Tecnologia Eletrônica e da Informação, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Amazonas, Manaus, Brazil

UniMiB SHAR datasets. For each proposed method, the authors used 5-fold-cross-validation and leave-one-subject-out methods. The best average accuracy obtained in the experiments carried out in the UniMiB SHAR dataset was 77.03%. This result was obtained using the Codebook method with Soft Assignment.

Falco et al. (2020) evaluated twenty-two different types of classifiers using the UniMiB SHAR dataset and the 5-fold-cross-validation method. The following classifiers were evaluated: Support Vector Machines with different kernels, KNN with different proximity measures and ensemble methods (boosted trees, bagging trees, subspace KNN, etc). The best result obtained for the 17 classes was a macro average accuracy of 86%, using an ensemble classifier based on subspace KNN.

Lv et al. (2020) proposed a hybrid deep network architecture. The proposed model, ConvLSTM, consists of fully connected convolutional layers. The idea explored by the authors is that a dense network ensures the maximum flow of information between the network layers, and a module for resource aggregating, that aggregates the extracted characteristics or each layer. The output of the aggregation module is the input of two Long Short Time Memory layers (LSTM layers). This model was evaluated in datasets OPPORTUNITY and UniMiB SHAR. For the UniMiBSHAR, the authors obtained a weighted F1-score of 78.4% and 97.3%, with leave-one-subject-out and 5-fold cross validation methods, respectively.

This work has the following objectives: 1) Propose a Convolutional Neural Network (CNN) architecture to classify the 17 classes of the UniMiB SHAR dataset; 2) Evaluate two different methods for converting an accelerometer data stream into an image, which will serve as input to a CNN; 3) Propose an LSTM architecture to classify the 17 classes of the UniMiB SHAR dataset; 4) Propose a Gated Recurrent Unit (GRU) architecture to classify the 17 classes of the UniMiB SHAR dataset; 5) Evaluate the CNN, LSTM and GRU performance with both leave-one-subject-out and 5-fold-cross-validation methods.

2 Materials and Methods

2.1 Dataset

The dataset used in this work was the UniMiB SHAR (Micucci et al. 2017). This dataset consists of samples from 30 different subjects, most of them female, aged between 18 and 60 years, height between 160 cm and 190 cm, and mass between 50 kg and 82 kg. The samples were acquired with the Samsung Galaxy Nexus I9250 smartphone with Android OS version 5.1.1 and equipped with the Bosh BMA220 acceleration sensor. The accelerometer signal consists of three components (x, y, z) that represent the acceleration in each of the three Cartesian axes. This signal was acquired with a sampling frequency of 50 Hz. In each sample, there are 151 values of each component, acquired in different timestamps. Therefore, for the three axes, there are 453 real values per sample.

The UniMiB SHAR dataset consists of 11,771 samples. The samples are divided into 17 classes, with 9 classes corresponding to Activities of Daily Living (ADL) and 8 classes corresponding to fall. Unfortunately, the bank is not balanced, with 7579 samples related to ADL and 4192 related to FALL. Table 1 shows the dataset classes divided in two groups, ADL and fall.

Figure 1 shows graphs of timestamps of x accelerometer values for the following 6 classes of a sample for a given subject: StandingUpFS (Standing up from sitting) in blue, StandingUpFL (Standing up from laying) in red, Walking in green, FallingBack (Generic falling backwards) in purple, HittingObstacle (Hitting an obstacle in the fall) in orange and Fall-ingWithPS (Falling forward with protection strategies) in cyan.

In this work, in order to compare the performances of the proposed neural network architectures, CNN, LSTM and GRU with the performances of other classifiers previously published in literature, we classified the 17 classes using two training-test methods, the leave-one-subject-out and 5-fold-cross-validation. In the former, the CNN, LSTM and GRU training was carried out with 29 individuals and tested with 1 individual. The CNN, LSTM and GRU were evaluated for all individuals. In the latter, the data were divided into 5 groups. Each model is trained with 4 groups and tested with

Table 1 UniMiB SHAR dataset classes

Action	Class name
ADL	Standing up from laying
	Lying down from standing
	Standing up from sitting
	Running
	Sitting down
	Going downstairs
	Going upstairs
	Walking
	Jumping
FALL	Falling backward while sitting on a chair
	Generic falling backward
	Falling forward with protection strategies
	Generic falling forward
	Falling rightward
	Falling leftward
	Hitting an obstacle in the fall
	Syncope

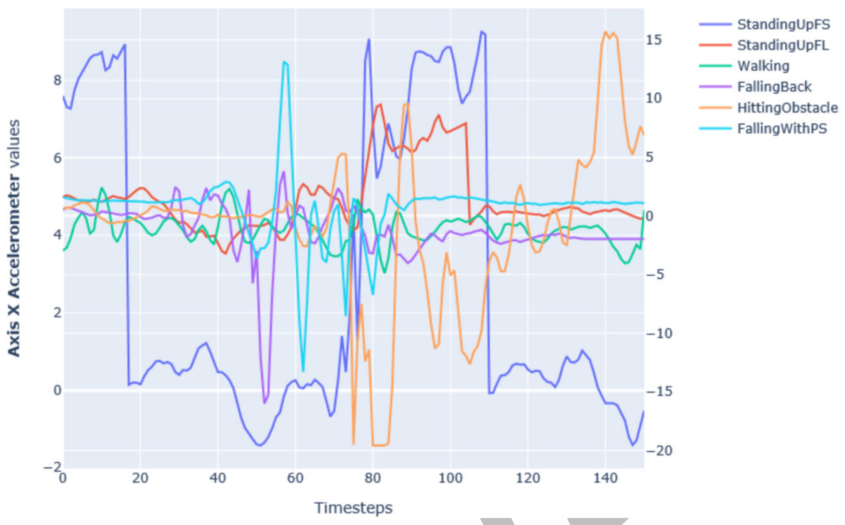


Fig. 1 Graphs of timestamps x accelerometer values for 6 classes of subject 1

1 group. The CNN, LSTM and GRU were evaluated for each fold.

2.2 Convolutional Neural Network

Convolutional Neural Networks can be used to deal with computer vision activities, performing the tasks of image classification and object segmentation. Much data such as text, spoken words, or financial and banking are not in the form of an image.

Although it is it is still possible to achieve competitive results with 1D-CNNs, by turning these data into a well-organized image, it is possible to use a 2D-CNN and obtain a higher performance (Sharma et al. 2019). For example, it is possible to create an image from the 2D distribution of one-dimensional audio data, or even, from the collection of temporal data from this audio, it is possible to build a spectrogram and use CNNs to classify such an image.

There are different methods to construct an image from a data stream. An image consists of spatially related pixels, and the positioning of these pixels can affect CNN's feature extraction and classification (Sharma et al. 2019).

When transforming a data sequence from an accelerometer into an image, in addition to the temporal sequence relationship generated by the continuous sampling of these data, it is possible to generate others data sensor relationships, because of the spatial relationship that appears when we distribute the values of samples of x, y and z in a two-dimensional matrix.

Each sample in the UniMiB SHAR dataset is a one-dimensional vector with 453 parameters, 151 for each of the axes (x, y and z). To use this data as a CNN input in Tensor Flow, they were transformed into a 4D tensor in the form (sample, row, column, channel). The sample coordinate of the 4D tensor is the order of the sample in the batch, and the channel coordinate is always equal to 1. The assembled matrix, with 151 rows and 3 columns, combining the sequential data from the 151 samples of x, y and z, corresponds to coordinates row and column of the 4D tensor. In one possible arrangement, shown in Fig. 2(a), the x samples fill the first rows, followed by the y samples, and, finally, by the z samples. We will refer to the tensor built that way as INPUT1.

In this work, we also carry out simulations using a different matrix arrangement of the x, y and z accelerometer samples. The assembled matrix, also with 151 rows and 3 columns, shown in Fig. 2(b), combines, in a same row data from sensors x, y and z. We will refer to the tensor built that way as INPUT2.

2.3 Long-Short Time Memory and Gated Recurrent Unit Networks

The LSTM and GRU architectures, two Recurrent Neural Networks (RNN), are used for sequential data processing. The ability to "remember" data from a sequence to better classify it, is the main characteristic of these architectures. They have reached the state of the art in applications such as voice recognition, text generation and others. Such networks have internal mechanisms that assure the long-term persistence of relevant information.

The LSTM network has the ability of classifying temporal series of unknown interval and duration. Concerning a daily activity, it does not matter if it occurs over a long- or short-time interval. What matters is the activity pattern. Hidden cell states storage the long-term memory of the network. The network output is called the network state and storage the short-term network memory. The inputs of a LSTM network are the network output state in a previous timestamp and the input sequence current value. The data input takes place using three gates: the forget gate, that removes information from the hidden state that are no longer useful; the input gate, that adds useful information and the output gate, that extracts useful

Fig. 2 (**a**) data matrix of INPUT1 tensor and (**b**) data matrix of INPUT2 tensor. Both tensors are used as input of CNN

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ \cdots & \cdots & \cdots \\ x_{151} & y_1 & y_2 \\ y_3 & y_4 & y_5 \\ y_6 & y_7 & y_8 \\ \cdots & \cdots & \cdots \\ y_{150} & y_{151} & z_1 \\ z_2 & z_3 & z_4 \\ \cdots & \cdots & \cdots \\ z_{149} & z_{150} & z_{151} \end{bmatrix} \qquad \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ x_{80} & y_{80} & z_{80} \\ x_{81} & y_{81} & z_{81} \\ x_{82} & y_{82} & z_{82} \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ x_{150} & y_{150} & z_{150} \\ x_{151} & y_{151} & z_{151} \end{bmatrix}$$

(a) (b)

The size of this figure is too big. I believe it could be reduce for only one column

information of the current hidden state, to generate the network state.

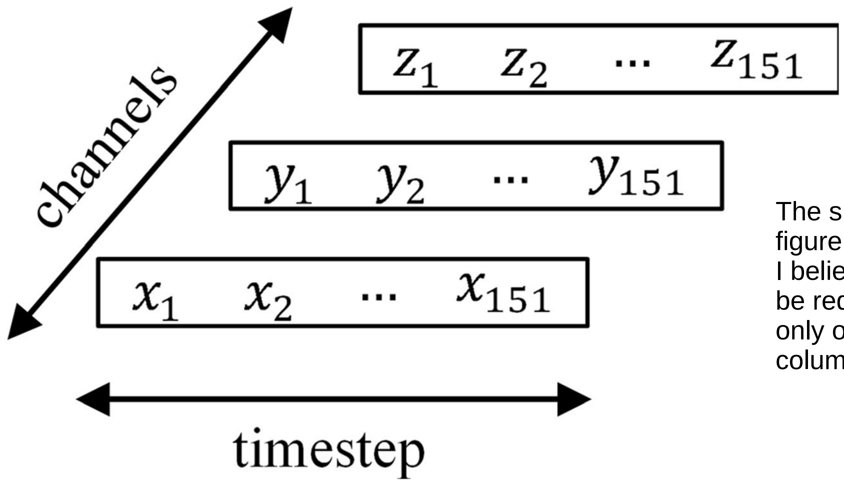
The GRU network has an architecture like that of LSTM network. The main differences are the following: they have only two input gates, the update gate and the forget gate, that have a role like those of input and output gates, in LSTM architecture. Other difference is that there are no hidden states anymore. The network output is the only network state. The GRU network is considered an evolution of the LSTM network.

In this work, to implement the LSTM and GRU networks in Tensor Flow, the samples were arranged in a 3D tensor in the form (sample, row, column). The sample coordinate of the 3D tensor is the order of the sample in the batch ach dataset. The assembled matrix, with 151 rows and 3 columns, shown in Fig. 3, combining the sequential data from the 151 samples, corresponds to row and column coordinates of the 3D tensor. We will refer to the tensor built that way as INPUT3. The data arrangement in this tensor is like INPUT1 data distribution. The only difference in the tensor dimension. INPUT3 has three dimensions, while INPUT1 has four dimensions.

2.4 Architectures and Training Parameters

The CNN architecture chosen for the development of the method has three subsampling steps and one classification

Fig. 3 Data matrix of INPUT3 tensor used as input of RNN architectures



The size of this figure is too big I believe it could be reduced to only one column.

step. Each subsampling step consists of a convolutional layer (Conv2D) followed by a max-pooling layer (MaxPooling2D). The classification step consists of a Flatten layer that vectorizes the output from the previous step, a Dense and a Softmax layer to carry out the classification.

The first Conv2D layer is made up of 24 kernel-size filters (6 × 3), the second Conv2D layer is made up of 44 kernel-size filters (3 × 3), and the Third Conv2D layer is made up of 36 kernel-size filters (3 × 3). MaxPooling2D beds have a pooling region equal to (2 × 1). The first Dense layer has 200 neurons and the second Dense layer has 17 neurons, to assign the input to one of the 17 classes. All the convolution operations used a stride of 1. The padding was done to obtain an output size equal to input. Figure 4 shows the CNN architecture proposed in this work.

To propose the LSTM architecture, we experimented with different number of layers and number of units in each layer. The one with the best results was the one with 5 stacked layers. The sizes of each layer are the following: 512 units, 256 units, 128 units, 64 units and 32 units. The last layer is a dense layer, with 17 units (one for each class) and the softmax activation function, that returns the probability of an input patter belongs to a given class. The GRU architecture is like the LSTM architecture. Figure 5 shows the architecture of both LSTM and GRU networks.

The early stop method, with a patience of 5, was used to avoid overfitting.

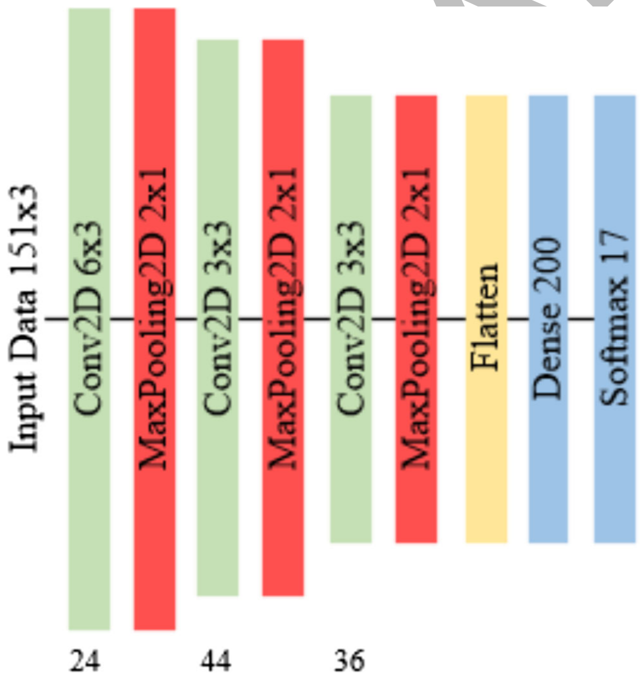


Fig. 4 CNN architecture for human activity recognition

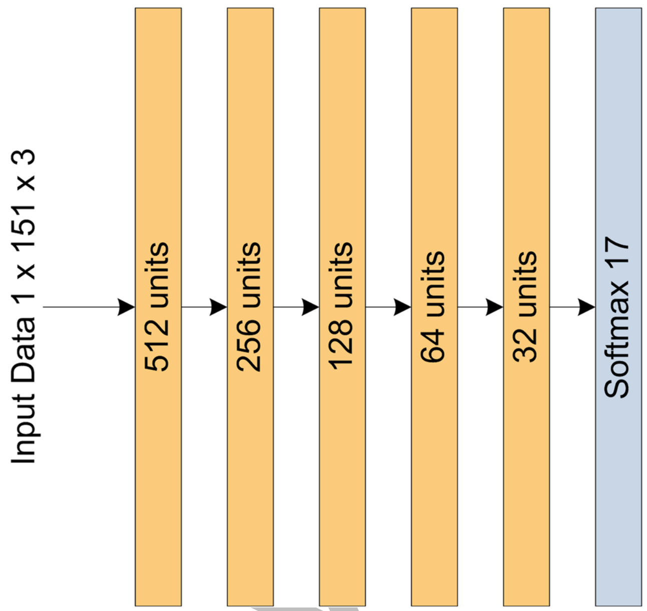


Fig. 5 Architecture of both LSTM and GRU networks for human activity recognition

2.5 Metrics

As stated before, two training-testing methodologies were used: leave-one-subject-out and 5-fold-cross-validation. The metrics used to evaluate the models performance were the average of the accuracies, average F1-score for the 30 subjects and 5 folds and the macro-average accuracy (MAA), which is the arithmetic average of the accuracies obtained for each class and is computed using the Eq. (1), where: N is the total of classes in the dataset, TP_i is the number of items of class i correctly classified and NP_i is the number of times that class i occurs in database.

$$MAA = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{NP_i} \qquad (1)$$

3 Results and Discussion

3.1 CNN Results

The CNN network was evaluated with two different tensors in input: INPUT1 and INPUT2. The results with INPUT1 and INPUT 2, using the leave-one-subject-out method, are shown in Tables 2 and 3, respectively. These Tables show the accuracy, the F1-score and the MAA for each individual and the average values for 30 individuals. The results of training with INPUT1 and INPUT2, using 5-fold-cross-validation, are shown in Tables 4 and 5, respectively. These Tables show

Table 2 Results with INPUT1 tensor and one-leave-subject-out method

Subject	MAA (%)	Accuracy (%)	F1-score (%)	Subject	MAA (%)	Accuracy (%)	F1-score (%)
1	60.42	72.39	71.39	16	71.76	80.34	79.20
2	68.42	77.36	75.01	17	73.68	80.94	79.71
3	76.92	86.18	85.56	18	74.16	81.03	81.35
4	44.01	50.48	51.63	19	77.32	84.21	83.48
5	76.85	80.87	78.79	20	61.73	78.74	78.24
6	83.70	87.86	87.71	21	67.62	71.42	71.72
7	73.08	81.82	81.86	22	62.78	69.01	67.33
8	76.93	88.92	89.26	23	75.00	81.29	79.05
9	65.03	75.36	72.47	24	59.03	77.58	76.04
10	84.66	84.02	83.42	25	66.19	76.05	76.03
11	69.13	78.06	78.20	26	87.61	88.99	89.00
12	69.64	85.00	84.51	27	77.25	86.79	86.87
13	74.07	81.17	80.60	28	86.50	90.51	89.04
14	67.54	79.84	78.23	29	67.28	75.53	73.96
15	69.31	81.77	81.32	30	73.32	79.90	79.52
Average MAA (%)						71.36	
Average Accuracy (%)						79.78	
Average F1-score (%)						79.02	

the accuracy, the F1-score and the MAA value for each fold and the average value.

We note that, in the leave-one-subject-out method, the accuracy for each subject differs greater from each other, due to the vitality and the unique way which individuals carry out activities. Also, classes in subjects are not balanced and some of them do not have data for all 17 classes, which makes training and validation difficult. Differently, in the 5-fold-cross-validation, the accuracies of each fold are close to each other, because, in each fold, the classes are balanced.

With the leave-one-subject-out method, the MAA obtained with INPUT1 tensor is 5.16% greater than the one obtained

Table 3 Results with INPUT2 tensor and one-leave-subject-out method

Subject	MAA (%)	Accuracy (%)	F1-score (%)	Subject	MAA (%)	Accuracy (%)	F1-score (%)
1	53.71	62.76	61.24	16	62.31	75.64	75.42
2	60.00	69.98	68.46	17	69.65	80.08	79.42
3	76.13	86.84	86.04	18	60.69	69.31	68.54
4	31.92	39.22	42.63	19	72.24	80.82	79.06
5	68.83	76.17	75.89	20	68.40	78.30	77.17
6	79.25	84.46	84.22	21	70.46	71.78	71.82
7	71.84	80.00	80.72	22	47.29	55.16	54.04
8	80.12	91.43	91.67	23	72.07	80.06	80.35
9	67.84	74.15	73.28	24	52.57	70.54	68.91
10	81.48	83.70	83.17	25	61.30	71.35	71.13
11	71.13	72.19	72.25	26	85.30	89.28	89.35
12	40.90	50.20	52.55	27	78.48	88.48	87.94
13	68.84	67.90	68.28	28	87.33	90.82	89.84
14	63.56	75.80	75.78	29	66.43	71.45	70.97
15	64.71	78.73	78.18	30	67.99	80.14	78.79
Average MAA (%)						66.74	
Average Accuracy (%)						74.89	
Average F1-score (%)						74.57	

Table 4 Results with INPUT1 tensor and 5-fold-cross-validation

Folders	MAA (%)	Accuracy (%)	F1-score (%)
1	90.52	93.58	93.53
2	90.07	92.99	92.95
3	90.80	94.09	94.06
4	90.42	93.41	93.37
5	90.39	93.50	93.46
Average (%)	90.44	93.51	93.48

with INPUT2 tensor. With 5-fold-cros-validation, this difference is 1.67%. Therefore, the input matrix shape has in influence in the CNN classification result.

These differences are due to the CNN architecture and to the data grouping in different matrices, as shown in Fig. 2. To understand it, imagine the application of the kernel filter of the first convolution layer (6 × 3) on the matrices on tensors INPUT1 and INPUT2, with the most left and highest point of the kernel overlapping the element x_1 of both matrices, as shown in Eqs. (2) and (3).

$$\begin{aligned} Char_{INPUT1} = {} & w_{11}x_1 + w_{12}x_2 + w_{13}x_3 + w_{21}x_4 + w_{22}x_5 \\ & + w_{23}x_6 + w_{31}x_7 + w_{32}x_8 + w_{33}x_9 \\ & + w_{41}x_{10} + w_{42}x_{11} + w_{43}x_{12} + w_{51}x_{13} \\ & + w_{52}x_{14} + w_{53}x_{15} + w_{61}x_{16} + w_{62}x_{17} \\ & + w_{63}x_{18} \end{aligned} \tag{2}$$

$$\begin{aligned} Char_{INPUT2} = {} & w_{11}x_1 + w_{12}y_1 + w_{13}z_1 + w_{21}x_2 + w_{22}y_2 \\ & + w_{23}z_2 + w_{31}x_3 + w_{32}y_3 + w_{33}z_3 \\ & + w_{41}x_{10} + w_{42}x_{11} + w_{43}x_{12} + w_{51}x_{13} \\ & + w_{52}x_{14} + w_{53}x_{15} + w_{61}x_{16} + w_{62}x_{17} \\ & + w_{63}x_{18} \end{aligned} \tag{3}$$

Table 5 Results with INPUT2 tensor and 5-fold-cross-validation method

Folders	MAA (%)	Accuracy (%)	F1-score (%)
1	85.95	90.65	90.62
2	86.95	91.07	91.02
3	89.86	92.77	92.76
4	88.40	91.92	91.86
5	87.36	91.29	91.33
Average (%)	87.71	91.54	91.52

where: $w_{ij,} 1 \leq i \leq 6,\ 1 \leq j \leq 3$ are *coefficients of kernel mask of the first convolutional layer.*

In Eq. (2), to generate $Char_{INPUT1}$, an activity temporal window of 18 timestamps and data of only one sensor are used: x_1 *to* x_{18}. Conversely, in Eq. (3), to generate $Char_{INPUT2}$, an activity temporal window of 4 timestamps, and data of three sensors are used: x_1 *to* x_4, y_1 *to* y_4 and z_1 *to* z_4. As the results obtained with the matrix of tensor INPUT1 outperform the results obtained with matrix of tensor INPUT2, we conclude that generating characteristics with samples of a great number of timestamps on the first convolutional layer is more important than generating characteristics with samples of more sensors. In other words, to identify an activity, a small temporal window is less suited than a large temporal window.

Figure 6 shows the confusion matrix obtained for subject 6, with tensor INPUT1 and leave-one-subject-out method. As shown in Table 2, the obtained MAA for subject 6 was 83.70%. Analyzing the confusion matrix, we notice that the network made correct classifications for most classes. Only for the FallingBackSC (Falling backwards while sitting on a chair) and Syncope classes the CNN made more misclassifications. Figure 7 shows the confusion matrix for the same subject 6, with tensor INPUT2 and leave-one-subject-out method. As shown in Table 3, MAA for subject 6 was 79.25%. We notice that, with tensor INPUT2, the CNN made more misclassifications than with tensor INPUT1.

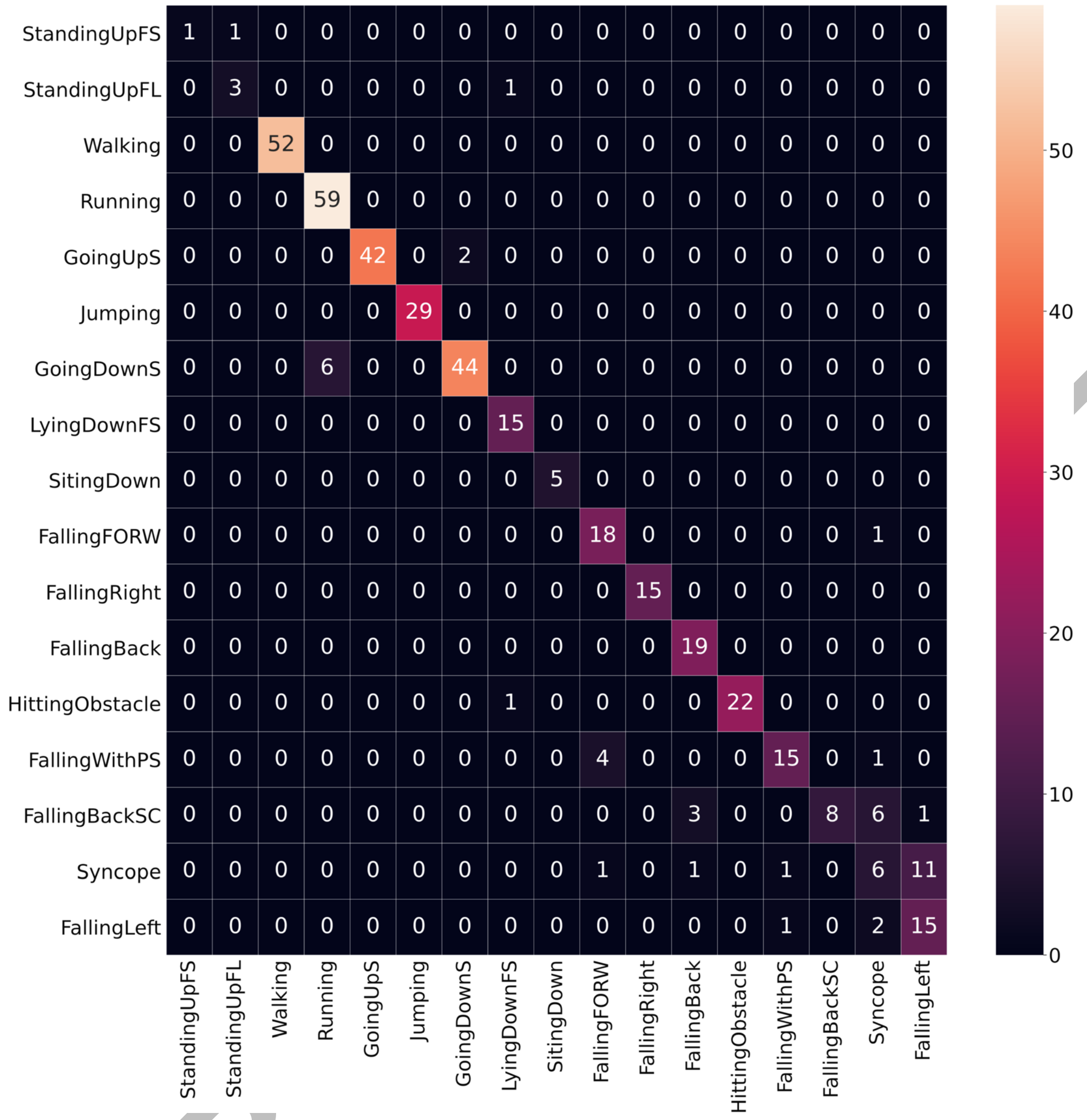
3.2 LSTM and GRU Results

The LSTM and GRU networks were trained and evaluated using the same input data, tensor INPUT3. The results for LSTM and GRU networks, using the leave-one-subject-out method, are shown in Tables 6 and 7, respectively. These tables show the accuracy, F1-score and MAA for the 30 subjects. The results of LSTM and GRU networks using the 5-fold cross validation method are shown in Tables 8 and 9, respectively.

As shown, the performance of LSTM and GRU architectures with the leave-one-subject-out method are similar. For the accuracy, F1-score and MAA metrics, the differences are less than 1%. With the 5-cross-validation method, nevertheless, the GRU architecture present a better performance than the LSTM one. For the accuracy and F1-score metrics, the difference is higher than 5%, while for the MAA metric, the difference is higher than 8%.

3.3 Discussion

Training and validation using the leave-one-subject-out method proved to be complicated, due to the vigor with which each person performs the activities. By the way, the dataset has



	StandingUpFS	StandingUpFL	Walking	Running	GoingUpS	Jumping	GoingDownS	LyingDownFS	SitingDown	FallingFORW	FallingRight	FallingBack	HittingObstacle	FallingWithPS	FallingBackSC	Syncope	FallingLeft
StandingUpFS	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
StandingUpFL	0	3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Walking	0	0	52	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Running	0	0	0	59	0	0	0	0	0	0	0	0	0	0	0	0	0
GoingUpS	0	0	0	0	42	0	2	0	0	0	0	0	0	0	0	0	0
Jumping	0	0	0	0	0	29	0	0	0	0	0	0	0	0	0	0	0
GoingDownS	0	0	0	6	0	0	44	0	0	0	0	0	0	0	0	0	0
LyingDownFS	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0
SitingDown	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
FallingFORW	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0	1	0
FallingRight	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0
FallingBack	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0
HittingObstacle	0	0	0	0	0	0	0	1	0	0	0	0	22	0	0	0	0
FallingWithPS	0	0	0	0	0	0	0	0	0	4	0	0	0	15	0	1	0
FallingBackSC	0	0	0	0	0	0	0	0	0	0	0	3	0	0	8	6	1
Syncope	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	6	11
FallingLeft	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	15

Fig. 6 Confusion matrix for subject 6, with INPUT1 matrix and leave-one-subject-out method

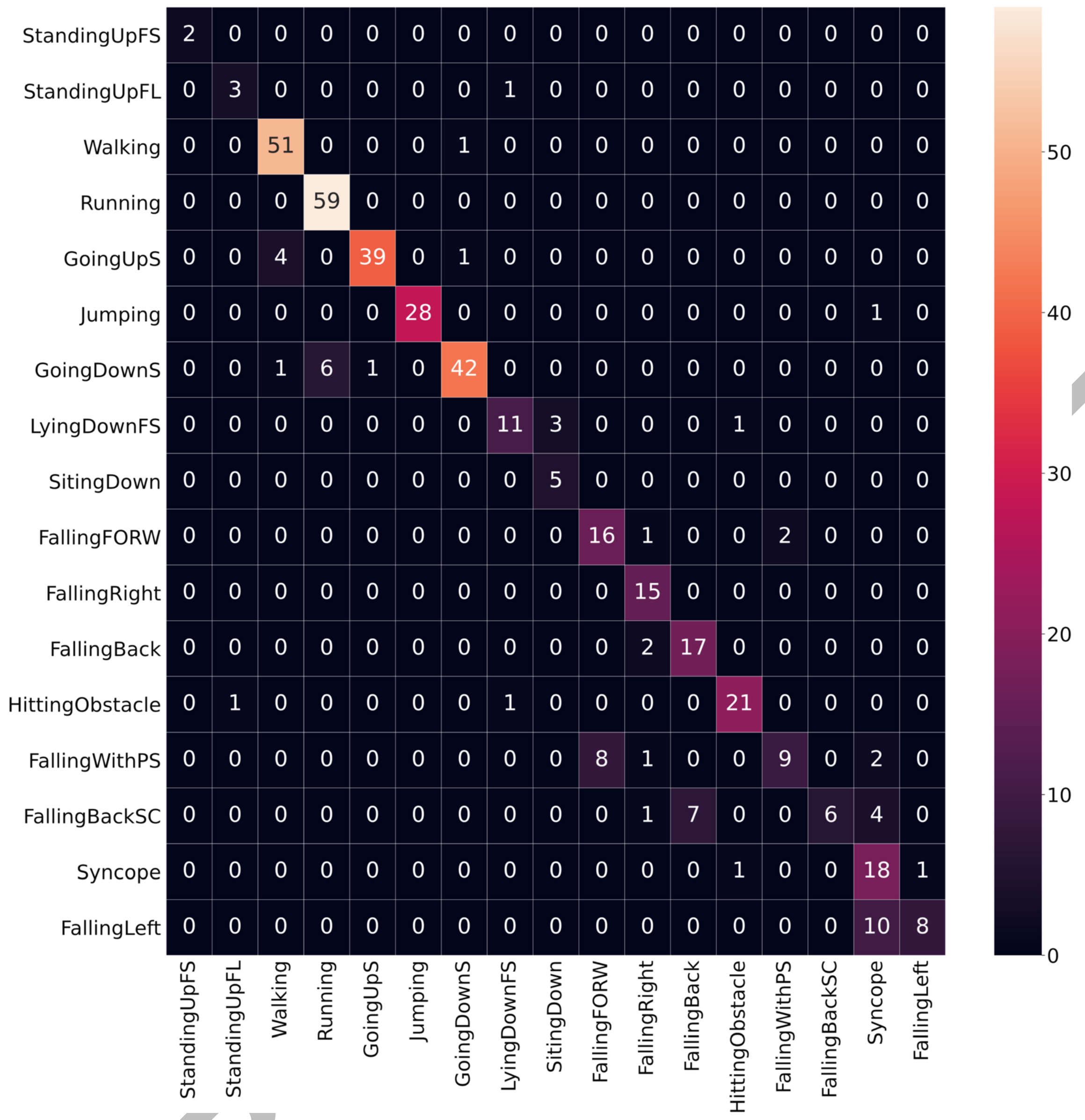
individuals between 18 and 60 years old. This makes ADL samples quite different from one subject to other.

The imbalance of the dataset also contributes to a lower performance. The dataset is also not uniform in the way the samples were obtained. Sometimes, repeated samples or subjects who do not have labels for all classes are found.

As shown in Fig. 1, we observe that data has a DC component, derived from the orientation in which the cell phone is located. Removing this component is not feasible if you want to classify the activities. However, when looking at the detection of more specific activities, such as fall and non-fall, this would be indispensable.

Concerning the one-leave-subject-out method, the CNN presents better results than the RNN. The way the patterns are presented to the networks influences their performance. With CNN, for example, the best results were obtained with tensor INPUT1. Therefore, we conclude that, in CNN input, it is more important to group data from a large time window than from different sensors.

Although we have small architectures, to avoid overfitting, we apply the early stop method. This contributed to improving the generalization of the networks. During the training, we stored the network configuration with better performance in the test set. Not necessarily the



	StandingUpFS	StandingUpFL	Walking	Running	GoingUpS	Jumping	GoingDownS	LyingDownFS	SitingDown	FallingFORW	FallingRight	FallingBack	HittingObstacle	FallingWithPS	FallingBackSC	Syncope	FallingLeft
StandingUpFS	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
StandingUpFL	0	3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Walking	0	0	51	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Running	0	0	0	59	0	0	0	0	0	0	0	0	0	0	0	0	0
GoingUpS	0	0	4	0	39	0	1	0	0	0	0	0	0	0	0	0	0
Jumping	0	0	0	0	0	28	0	0	0	0	0	0	0	0	0	1	0
GoingDownS	0	0	1	6	1	0	42	0	0	0	0	0	0	0	0	0	0
LyingDownFS	0	0	0	0	0	0	0	11	3	0	0	0	1	0	0	0	0
SitingDown	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
FallingFORW	0	0	0	0	0	0	0	0	0	16	1	0	0	2	0	0	0
FallingRight	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0
FallingBack	0	0	0	0	0	0	0	0	0	0	2	17	0	0	0	0	0
HittingObstacle	0	1	0	0	0	0	0	1	0	0	0	0	21	0	0	0	0
FallingWithPS	0	0	0	0	0	0	0	0	0	8	1	0	0	9	0	2	0
FallingBackSC	0	0	0	0	0	0	0	0	0	0	1	7	0	0	6	4	0
Syncope	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	18	1
FallingLeft	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	8

Fig. 7 Confusion matrix for subject 6, with INPUT2 matrix and leave-one-subject-out method

better performance takes place at the end of a training section.

The performance of the CNN architecture was better than the performance of the RNN architectures with leave-one-subject-out method. Nevertheless, the performance of the RNN architecture was better than the performance of the CNN architecture with the 5-fold-cross-validation method.

The performance of GRU and LSTM networks were similar with the leave-one-subject-out method. Nevertheless, the former has a better performance with the 5-fold-cross-validation method.

With the leave-one-subject-out method, the CNN with tensor INPUT1 presents the following metric values: MAA of 71.36%, accuracy of 79.78% and F1-score of 79.02%. These results are better than the ones obtained (Li et al. 2018; Lv et al. 2020; Micucci et al. 2017). Li et al. (2018) obtained an accuracy pf 77.03% and F1-score of 75.93%. Lv et al. (2020) obtained an F1-score of 78.4%. Micucci et al. (2017) obtained an MAA of 56.53%.

With the 5-fold-cross-validation method, we obtained the following values with GRU architecture and tensor INPUT3: MAA of 95.49%, accuracy of 97.83% and F1-

Table 6 Results of LSTM model and one-leave-subject-out method

Subject	MAA (%)	Accuracy (%)	F1-score (%)	Subject	MAA (%)	Accuracy (%)	F1-score (%)
1	51.38	65.36	63.71	16	63.75	78.84	77.99
2	60.88	72.04	71.61	17	64.82	79.23	78.06
3	67.09	83.22	82.41	18	57.68	68.27	67.22
4	28.98	34.08	40.09	19	75.97	82.71	82.49
5	64.86	72.82	72.07	20	65.5	85.03	84.86
6	79.34	86.41	85.29	21	62.77	67.5	66.61
7	66.34	77.88	77.3	22	50.1	55.99	53.39
8	76.19	88.66	89.21	23	48.43	58.89	57.79
9	55.22	68.59	67.85	24	53.4	74.96	74.24
10	78.14	81.47	81.51	25	68.87	76.54	75.56
11	61.54	68.88	68.08	26	72.18	85.71	85.36
12	63.65	51.6	46.22	27	83.85	88.48	88
13	56.23	72.67	70.93	28	79.82	87.02	87.11
14	63.17	76.88	76.03	29	66.62	73.39	72.07
15	63	77.97	76.44	30	63.78	81.12	80.31
Average MAA (%)						63.78	
Average Accuracy (%)						74.07	
Average F1-score (%)						73.33	

score of 97.55%. These results are better than the results obtained (Falco et al. 2020; Micucci et al. 2017) and comparable to the results obtained by Lv et al. (2020). Falco et al. (2020) obtained an MAA of 86%. Micucci et al. (2017) obtained an MAA of 82.86%. Lv et al. (2020) obtained an F1-score of 97.3% and an MAA of 95.3%.

Table 7 Results of GRU model and one-leave-subject-out method

Subject	MAA (%)	Accuracy (%)	F1-score (%)	Subject	MAA (%)	Accuracy (%)	F1-score (%)
1	57.72	62.5	58.19	16	63.29	74.78	74.16
2	56.35	71.35	70.79	17	66.35	76.44	74.67
3	60.93	80.92	81.25	18	57.52	65.52	64.77
4	31.49	34.08	36.53	19	74.27	82.33	81
5	61.13	73.48	71.14	20	57.12	76.14	75.92
6	82.35	88.83	88.38	21	51.39	56.07	54.55
7	65.84	76.36	74.39	22	44.77	52.48	52.79
8	73.89	85.89	85.08	23	66.12	73.01	72.17
9	52.67	70.77	69	24	55.48	73.49	72.16
10	79.89	80.83	80.17	25	62.75	69.63	68.73
11	62.92	72.7	72.76	26	77.25	87.79	87.72
12	61.79	45.8	41.88	27	79.56	90.45	89.01
13	61.6	72.15	71.36	28	86.83	88.61	87.81
14	60.47	74.19	73.52	29	56.73	71.65	70.99
15	49.8	72.15	70.09	30	64.59	77.45	75.86
Average MAA (%)						62.76	
Average Accuracy (%)						72.59	
Average F1-score (%)						71.56	

Table 8 Results with LSTM model and 5-fold-cross-validation method

Folders	MAA (%)	Accuracy (%)	F1-score (%)
1	80.81	89.34	89.17
2	89.16	93.97	93.96
3	90.14	94.48	94.48
4	80.77	89.51	89.29
5	91.88	95.28	95.28
Average (%)	86.55	92.51	92.43

4 Conclusions

In this paper, we seek to compare the recognition of human activities with convolutional neural network and recurrent neural networks, using the UniMiB SHAR dataset. We proposed CNN, LSTM e GRU architectures and converted an accelerometer data stream into an image, to serves as input to CNN. Two different methods of converting the data stream into an image were evaluated. We conclude that, in the CNN input, it is more important to group data from a large time window than from different sensors.

The performance of the CNN architecture was better than the performance of the RNN architectures with leave-one-subject-out method. Nevertheless, the performance of the RNN architectures was better than the performance of the CNN architecture with the 5-fold-cross-validation method.

For classification of 17 human activities with the leave-one-subject-out method, the CNN architecture, with tensor INPUT1, presents an MAA of 71.36%, a mean accuracy pf 79.78% and F1-score of 79.02%. These results are better than others previously published in the literature with the same dataset. With the 5-fold-cross-validation method, the GRU architecture, with tensor INPUT3, presents an MAA of 95.49%, an accuracy of 97.83% and F1-score of 97.55%. These results are better or comparable to other results previously published in the literature with the same dataset.

Table 9 Results with GRU model and 5-fold-cross-validation method

Folders	MAA (%)	Accuracy (%)	F1-score (%)
1	95.75	97.62	97.62
2	94.24	97.37	97.35
3	95.20	97.45	97.44
4	96.42	97.83	97.82
5	95.82	97.54	97.53
Average (%)	95.49	97.83	97.55

In future works, other activity recognition datasets could be explored to assess whether the data grouping methods proposed in this work have the same performance.

Acknowledgments This research, carried out within the scope of Samsung-UFAM Project for Education and Research (SUPER), according to Article 48 of Decree n° 6.008/2006(SUFRAMA), was funded by Samsung Electronics of Amazonia Ltda., under the terms of Federal Law n° 8.387/1991, through agreement 001/2020, signed with Federal University of Amazonas and FAEPI, Brazil.

Declarations

Conflict of Interest The authors had declared no interest conflict.

References

Falco ID, Pietro GD, Sannino G (2020) Evaluation of artificial intelligence techniques for the classification of different activities of daily living and falls. Neural Comput & Applic 32:747–758. https://doi.org/10.1007/s00521-018-03973-1

Li F, Shirahama K, Nisar MA, Köping L, Grzegorzek M (2018) Comparison of feature learning methods for human activity recognition using wearable sensors. Sensors 18:1–22. https://doi.org/10.3390/s18020679

Lisowska A, O'Neil A, Poole I (2018) Cross-cohort evaluation of machine learning approaches to fall detection from accelerometer data, 11th international joint conference on biomedical engineering systems and technologies

Lv T, Wang X, Jin L, Xiao Y, Song M (2020) A hybrid network based on dense connection and weighted feature aggregation for human activity recognition. IEEE Access 8:68320–68332. https://doi.org/10.1109/ACCESS.2020.2986246

Micucci D, Mobilio M, Napoletano P (2017) UniMiB SHAR: a dataset for human activity recognition using acceleration data from smartphones. Appl Sci 7:1–19. https://doi.org/10.3390/app7101101

Park SY, Ju H, Park CG (2016) Stance phase detection of multiple actions for military drill using foot-mounted IMU, International Conference on Indoor Positioning and Indoor Navigation

Sharma A, Vans E, Shigemizu D, Boroevich KA, Tsunoda T (2019) Deepinsight: a methodology to transform a non-image data to an image for convolution neural network architecture. Sci Rep 9:1–7. https://doi.org/10.1038/s41598-019-47765-6

Subasi A, Radhwan M, Kurdi R, Khateeb K (2018) IoT based mobile healthcare system for human activity recognition, 15th learning and technology conference

Yang J, Lee J, Choi J (2011) Activity recognition based on RFID object usage for smart Mobile devices. J Comput Sci Technol 26:239–246. https://doi.org/10.1007/s11390-011-9430-9

Yin J, Yang Q, Pan JJ (2008) Sensor-based abnormal human-activity detection. IEEE Trans Knowl Data Eng 20:1082–1090. https://doi.org/10.1109/TKDE.2007.1042