Studies in Systems, Decision and Control 273

Hiram Ponce
Lourdes Martínez-Villaseñor
Jorge Brieva
Ernesto Moya-Albor *Editors*

Challenges and Trends in Multimodal Fall Detection for Healthcare

EXTRAS ONLINE

Springer

Studies in Systems, Decision and Control

Volume 273

Series Editor

Janusz Kacprzyk, Systems Research Institute, Polish Academy of Sciences,
Warsaw, Poland

The series "Studies in Systems, Decision and Control" (SSDC) covers both new developments and advances, as well as the state of the art, in the various areas of broadly perceived systems, decision making and control–quickly, up to date and with a high quality. The intent is to cover the theory, applications, and perspectives on the state of the art and future developments relevant to systems, decision making, control, complex processes and related areas, as embedded in the fields of engineering, computer science, physics, economics, social and life sciences, as well as the paradigms and methodologies behind them. The series contains monographs, textbooks, lecture notes and edited volumes in systems, decision making and control spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution and exposure which enable both a wide and rapid dissemination of research output.

** Indexing: The books of this series are submitted to ISI, SCOPUS, DBLP, Ulrichs, MathSciNet, Current Mathematical Publications, Mathematical Reviews, Zentralblatt Math: MetaPress and Springerlink.

More information about this series at http://www.springer.com/series/13304

Hiram Ponce · Lourdes Martínez-Villaseñor ·
Jorge Brieva · Ernesto Moya-Albor
Editors

Challenges and Trends in Multimodal Fall Detection for Healthcare

Springer

Editors
Hiram Ponce
Facultad de Ingeniería
Universidad Panamericana
Mexico City, Mexico

Lourdes Martínez-Villaseñor
Facultad de Ingeniería
Universidad Panamericana
Mexico City, Mexico

Jorge Brieva
Facultad de Ingeniería
Universidad Panamericana
Mexico City, Mexico

Ernesto Moya-Albor
Facultad de Ingeniería
Universidad Panamericana
Mexico City, Mexico

ISSN 2198-4182 ISSN 2198-4190 (electronic)
Studies in Systems, Decision and Control
ISBN 978-3-030-38747-1 ISBN 978-3-030-38748-8 (eBook)
https://doi.org/10.1007/978-3-030-38748-8

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This book presents challenging issues and current trends for designing human fall detection and classification systems, as well as healthcare technologies, using multimodal approaches. In healthcare, falls are frequent especially among elderly people and it is considered a major health problem worldwide. Recently, fall detection and classification systems have been proposed to address this problem, and to reduce the time, a person fallen receives assistance. For a comprehensive perspective of these healthcare technologies, the book is divided into two parts.

In the first part, human fall detection and classification systems are approached. A holistic view, from the design process to the implementation, is considered in the self-contained chapters presented. Moreover, these contributions mainly correspond to the challenges, methodologies adopted and results of the international competition, namely *Challenge UP—Multimodal Fall Detection* that was held during the International Joint Conference on Neural Networks (IJCNN) in 2019. Throughout this part, many of the chapters include open coding. This gives readers and practitioners the opportunity to be involved in the fall detection and classification problem by hands-on experience. First, chapter "Open Source Implementation for Fall Classification and Fall Detection Systems" presents a public multimodal dataset for fall detection and classification systems, namely UP-Fall detection. This dataset was part of the above-mentioned competition; thus, a concise tutorial on how to manipulate and analyze it, as well as how to train classification models and evaluate those using the dataset, for classification systems, is presented. In chapter "Detecting Human Activities Based on a Multimodal Sensor Data Set Using a Bidirectional Long Short-Term Memory Model: A Case Study," authors propose a deep learning model using bidirectional long short-term memory (Bi-LSTM) to detect five different types of falls using a dataset provided by the *Challenge UP* competition. The work corresponds to authors that won the third place. In contrast, chapter "Intelligent Real-Time Multimodal Fall Detection in Fog Infrastructure Using Ensemble Learning" presents a proposed methodology for conducting human fall detection near real time by reducing the processing latency. This approach considers distributing the fall detection chain over different levels of computing: cloud, fog, edge and mist. In addition, chapter "Wearable Sensors

Data-Fusion and Machine-Learning Method for Fall Detection and Activity Recognition" presents a method for fall detection and classification using the UP-Fall detection dataset. The authors present an interesting approach performing unsupervised similarity search in order to find the most similar users to the ones in test set, helping for parameter tuning. These authors won the first place in the *Challenge UP* competition. In contrast to the above sensor-based approaches, in chapter "Application of Convolutional Neural Networks for Fall Detection Using Multiple Cameras," authors present a fall detection system using a 2D convolutional neural network (CNN) evaluating independent information of two monocular cameras with different viewpoints, using the public UP-Fall detection dataset. The results obtained show that the proposed approach detects human falls with high accuracy, and it has comparable performance to a multimodal approach. Lastly, chapter "Approaching Fall Classification Using the UP-Fall Detection Dataset: Analysis and Results from an International Competition" presents the results of the competition and the lessons learned during this experience. In addition, it discusses trends and issues on human fall detection and classification systems.

On the other hand, the second part comprises a set of review and original contributions in the field of multimodal healthcare. These works present trends on ambient assisted living and health monitoring technologies considering the user-centered approach.

Chapter "Classification of Daily Life Activities for Human Fall Detection: A Systematic Review of the Techniques and Approaches" reviews the techniques and approaches employed to device systems to detect unintentional falls. The techniques are classified based on the approaches employed and the used sensors and noninvasive vision-based devices. In chapter "An Interpretable Machine Learning Model for Human Fall Detection Systems Using Hybrid Intelligent Models," authors propose a fall detection system based on intelligent techniques using feature selection techniques and fuzzy neural networks. The authors highlight the importance of feature selection techniques to improve the performance of hybrid models. The main goal was to extract knowledge through fuzzy rules to assist in the fall detection process. In chapter "Multi-sensor System, Gamification, and Artificial Intelligence for Benefit Elderly People," authors present a multi-sensory system into a smart home environment and gamification to improve the quality life of elderly people, i.e., avoiding social isolation and increasing physical activity. The proposal comprises a vision camera and a voice device, and artificial intelligence is used in the data fusion. Lastly, chapter "A Novel Approach for Human Fall Detection and Fall Risk Assessment" proposes a noninvasive fall detection system based on the height, velocity, statistical analysis, fall risk factors and position of the subject from depth information through cameras. The system is then adaptable to the physical conditions of the user.

We consider this book useful for anyone who is interested in developing human fall detection and classification systems and related healthcare technologies using multimodal approaches. Scientists, researchers, professionals and students will gain understanding on the challenges and trends on the field. Moreover, this book is also attractive to any person interested in solving signal recognition, vision and machine learning challenging problems given that the multimodal approach opens many experimental possibilities in those fields.

Lastly, the editors want to thank Universidad Panamericana for all the support given to this publication and the related research project that includes the organization of the international competition and the creation of the public dataset. The editors also want to thank Editor Thomas Ditzinger (Springer) for his valuable feedback and recognition to this work.

Mexico City, Mexico
November 2019

Hiram Ponce
Lourdes Martínez-Villaseñor
Jorge Brieva
Ernesto Moya-Albor

Acknowledgement This edited book has been funded by Universidad Panamericana through the grant "Fomento a la Investigación UP 2018," under project code UP-CI-2018-ING-MX-04.

Contents

Contributors

Vanessa S. Araujo Faculty UNA of Betim. Av. Gov. Valadares, Betim, MG, Brazil

Silvano Ramos de Assis Neto Universidade de Pernambuco, Pernambuco, Brazil

Lucas O. Batista Faculty UNA of Betim. Av. Gov. Valadares, Betim, MG, Brazil

Malika Bendechache Irish Institute of Digital Business, Dublin City University, Dublin, Ireland

Jorge Brieva Facultad de Ingeniería, Universidad Panamericana, Ciudad de México, México

V. Divya Department of Computer Science and Engineering, Thiagarajar College of Engineering, Madurai, India

Ricardo Espinosa Facultad de Ingeniería, Universidad Panamericana, Aguascalientes, Aguascalientes, Mexico

Matjaž Gams Jožef Stefan Institute, Ljubljana, Slovenia

Hristijan Gjoreski Faculty of Electrical Engineering and Information Technologies, Ss Cyril and Methodius University, Skopje, North Macedonia

Martin Gjoreski Jožef Stefan Institute, Ljubljana, Slovenia

Augusto J. Guimaraes Faculty UNA of Betim. Av. Gov. Valadares, Betim, MG, Brazil

Sebastián Gutiérrez Facultad de Ingeniería, Universidad Panamericana, Aguascalientes, Aguascalientes, Mexico

M. Mahadi Abdul Jamil Biomedical Engineering Modeling and Simulation (BIOMEMS) Research Group, Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, Johor, Malaysia

Ivana Kiprijanovska Faculty of Electrical Engineering and Information Technologies, Ss Cyril and Methodius University, Skopje, North Macedonia

R. Leena Sri Department of Computer Science and Engineering, Thiagarajar College of Engineering, Madurai, India

Mitja Luštrek Jožef Stefan Institute, Ljubljana, Slovenia

Theo Lynn Irish Institute of Digital Business, Dublin City University, Dublin, Ireland

Lourdes Martínez-Villaseñor Facultad de Ingeniería, Universidad Panamericana, Ciudad de México, México

Omar Mata School of Engineering and Sciences, Tecnológico de Monterrey, México City, Mexico

Alan Meier Lawrence Berkeley National Laboratory, University of California, Berkeley, CA, USA

Juana Isabel Méndez School of Engineering and Sciences, Tecnológico de Monterrey, México City, Mexico

Natasha Mladenovska Faculty of Electrical Engineering and Information Technologies, Ss Cyril and Methodius University, Skopje, North Macedonia

Arturo Molina School of Engineering and Sciences, Tecnológico de Monterrey, México City, Mexico

Ernesto Moya-Albor Facultad de Ingeniería, Universidad Panamericana, Ciudad de México, México

Anastasija Nikolovska Faculty of Electrical Engineering and Information Technologies, Ss Cyril and Methodius University, Skopje, North Macedonia

Yoosuf Nizam Department of Engineering, Faculty Engineering, Science and Technology, The Maldives National University (MNU), Malé, Maldives

José Núñez-Martínez Facultad de Ingeniería, Universidad Panamericana, Ciudad de México, México

Therese Peffer Institute for Energy and Environment, University of California, Berkeley, CA, USA

Hiram Ponce Facultad de Ingeniería, Universidad Panamericana, Ciudad de México, México

Pedro Ponce School of Engineering and Sciences, Tecnológico de Monterrey, México City, Mexico

Thiago S. Rezende Faculty UNA of Betim. Av. Gov. Valadares, Betim, MG, Brazil

Pierangelo Rosati Irish Institute of Digital Business, Dublin City University, Dublin, Ireland

Guto Leoni Santos Universidade Federal de Pernambuco, Pernambuco, Brazil

Elisson da Silva Rocha Universidade de Pernambuco, Pernambuco, Brazil

Paulo Vitor C. Souza Faculty UNA of Betim. Av. Gov. Valadares, Betim, MG, Brazil

Simon Stankoski Faculty of Electrical Engineering and Information Technologies, Ss Cyril and Methodius University, Skopje, North Macedonia

Patricia Takako Endo Universidade de Pernambuco, Pernambuco, Brazil

Marija Trajanoska Faculty of Electrical Engineering and Information Technologies, Ss Cyril and Methodius University, Skopje, North Macedonia

Bojana Velichkovska Faculty of Electrical Engineering and Information Technologies, Ss Cyril and Methodius University, Skopje, North Macedonia

Challenges and Solutions on Human Fall Detection and Classification

Throughout this part, human fall detection and classification systems are approached. A holistic view, from the design process to the implementation, is considered in the self-contained chapters presented. Moreover, these contributions mainly correspond to the challenges, methodologies adopted and results of the international competition namely *Challenge UP—Multimodal Fall Detection* that was held during the International Joint Conference on Neural Networks (IJCNN) in 2019.

Open Source Implementation for Fall Classification and Fall Detection Systems

Hiram Ponce, Lourdes Martínez-Villaseñor, José Núñez-Martínez, Ernesto Moya-Albor and Jorge Brieva

Abstract Distributed social coding has created many benefits for software developers. Open source code and publicly available datasets can leverage the development of fall detection and fall classification systems. These systems can help to improve the time in which a person receives help after a fall occurs. Many of the simulated falls datasets consider different types of fall however, very few fall detection systems actually identify and discriminate between each category of falls. In this chapter, we present an open source implementation for fall classification and detection systems using the public UP-Fall Detection dataset. This implementation comprises a set of open codes stored in a GitHub repository for full access and provides a tutorial for using the codes and a concise example for their application.

Keywords Human fall detection · Human activity recognition · Ambient assisted living · Machine learning · Open coding

---

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-38748-8_1) contains supplementary material, which is available to authorized users.

H. Ponce (✉) · L. Martínez-Villaseñor (✉) · J. Núñez-Martínez · E. Moya-Albor · J. Brieva
Facultad de Ingeniería, Universidad Panamericana,
Augusto Rodin 498, 03920 Ciudad de México, México
e-mail: hponce@up.edu.mx

L. Martínez-Villaseñor
e-mail: lmartine@up.edu.mx

J. Núñez-Martínez
e-mail: 0169723@up.edu.mx

E. Moya-Albor
e-mail: emoya@up.edu.mx

J. Brieva
e-mail: jbrieva@up.edu.mx

H. Ponce et al. (eds.), *Challenges and Trends in Multimodal Fall Detection for Healthcare*, Studies in Systems, Decision and Control 273,
https://doi.org/10.1007/978-3-030-38748-8_1

1 Introduction

Distributed social coding has gained popularity in recent years enabling software developers across the world to share their code and participate in different projects. GitHub is actually the most popular social coding platform and project hosting service [9]. GitHub has created many benefits for software developers. It has improved the way professionals work, it provides a traceable project repository, it is a meeting place for research communities with common interests, and it is also transforming the learning experience for software developer newcomers [21].

With the aim of benefit the fall detection system developers community, in this chapter, we present an open code project for fall detection and classification. There is a lack of reference frameworks and very few publicly available datasets for fall detection and classification. Open source code and publicly available datasets can leverage the development of fall detection and fall classification systems. These kinds of systems can reduce the time required for patient that suffer a fall to receive medical attention mitigating its consequence.

Thus, our goals are: (a) to promote sharing and reusing our UP-Fall Detection and classification dataset [12] and open source code; (b) with social collaboration, asses and improve the UP-Fall Detection dataset for multimodal fall detection and classification systems; and (c) contribute to software developers community providing a framework for different experimentation for multimodal fall detection and classification.

In this regard, it is possible to address different design issues in order to simplify a fall detection and classification system:

- Select which sensors or combination of sensors are to be used
- Determine the best placement of the sources of information
- Select the most suitable machine learning classification method for fall detection and classification and human activity recognition.

There are some examples of open source for fall detection published in GitHub Development Platform. We reviewed the ones with more influence based on stars given by the community. In [5], there is a GitHub repository available for real-time activities of daily living (ADL) and fall detection implemented in TensorFlow. The authors consider MobiFall dataset [18] for train and test which has 9 activities and 4 types of fall although all types are tagged as fall. The repository contains Python programs for data load, train a model with RNN, and several utilities which can be reused. Nevertheless, this code is poorly documented. The same authors also publish code for ADL recognition and fall detection using Convolutional Neural Networks (CNN) [6] using MobiFall [18] and SisFall [16] datasets for training and testing. Code for building CNN model is available, but documentation is also scarce. Other available projects also use Sisfall Dataset for model building like [3]. This project includes Jupiter Notebooks for data preparation and threshold based classification, code to implement three different machine learning classifiers (K-Nearest Neighbors, Support Vector Machine, Neural Networks) and implementing fall detection with Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) models.

An OpenCV project to detect a person fall in videos with Haarcascade is published in [15]. It provide a very simple code based on Haarcascade. A fall detection Android app based on accelerometer, gyroscope and magnetometer is published in [13]. With this app it is possible to send SMS to the mobile numbers stored a SQLite Database.

From this review, we determine that there are very few reliable open source projects for fall detection and none that also permit fall classification and human activity recognition. All of the reviewed projects only include one modality of data, mainly accelerometers, gyroscopes or cameras. Only one project enables implementation of three models based on machine learning classifiers.

Our proposal includes multiple modalities of source data collected in our own publicly available UP-Fall Detection dataset [12], detailed code and documentation for each phase of the processes of human activity recognition, fall detection and fall classification, and code to implement four different classification methods (i.e. Random Forest, Support Vector Machine, Multilayer Perceptron, K-nearest neighbors).

The rest of the chapter is organized as follows. Firstly, in Sect. 2, we overview the fall detection and fall classification process. In Sect. 3, we describe the UP-Fall Detection dataset. Then, in Sect. 4, we present our open code proposal for fall detection and fall classification. Section 5 shows a concise example on how to use our framework. Lastly, Sect. 6 presents some discussion on the presented work.

2 Fundamentals of Fall Detection and Classification

Falls are a major problem specially among the elderly causing fractures and serious injures [17]. The impact and consequences of fall can be reduced when the event is detected and medical attention is provided rapidly.

Monitoring elderly for abnormal events has gained interest in the assisted living community, so many fall detection and classification solutions are emerging. As discussed above, opportune fall detection is important, but it is also necessary to identify the type of fall. Different types of falls can provoke different types of injuries [1]. A better medical response or prevention treatment can be provided if the type of fall suffered is known. For instance, from the loss of consciousness it is more likely that a lateral fall occurs. A slip is more probably to cause serious injuries because the ease of breaking from a forward fall [17].

Likewise, fall prevention strategies can potentially be designed from analyzing the relation between the type of falls with the resulting injuries [14]. In this section, we briefly describe some fundamental concepts for better understanding fall detection and fall classification systems and processes.

Fig. 1 Activity recognition chain methodology

2.1 Fall Detection and Classification Processes

From the classification process point of view, fall detection is the task of discrimination between falls and activities of daily living (ADL). This means classifying a new observation predicting only two classes namely fall or non-fall activities. On the other hand, fall classification is the task of not only identifying a fall but also categorizing the type of fall for example falling backward, falling forward, falling while sitting, falling sideward, falling with knees, among others.

The processes of fall detection and fall classification are similar to human activity recognition. Figure 1 shows the activity recognition chain (ARC) approach proposed by [4]. This methodology is often adopted to develop the workflow of a fall detection and/or classification system. It consists mainly in five steps: (i) data acquisition, (ii) windowing, (iii) feature extraction, (iv) feature selection and (v) activity models and classification.

Data Acquisition

In the first step of fall detection and/or classification process, raw data is acquired from one or more different types of sources. Fall detection and classification systems can also differ depending mainly on the data acquisition system and algorithms used to detect the fall. Fall detection and classification approaches can be based on wearable sensors, and contextual sensors. The most commonly used wearable sensors are accelerometers, gyroscopes most recently embedded in smart phones or smart watches. Context-aware systems use sensors deployed in the environment like cameras, floor sensors, infrared sensors, thermal sensors, pressure sensors, radar and microphones among others [10]. According to Xu et al. [20] wearable sensors based in accelerometers and Kinect are the most recent trends for fall detection. Interest in multimodal approaches is rising in order to increase precision and robustness.

Windowing

A continuous sensor stream is obtained from data acquisition of the consecutive activities performed by a person. There are different ways of data segmentation of this data stream of time series. The most commonly used segmentation method for activity recognition is windowing. A segment or window of the time series of each attribute is mapped and label with a corresponding activity. A sliding window is moved over the time series data so this segment is used in the next steps of the process. The selection of the length of the window influences directly the performance of the classifier because it produces different number of samples. A large window size can include several activities and a very short window may not provide sufficient information.

Feature Extraction

The main motivation for extracting features from each time window is to obtain quantitative measures that allow signals to be compared and in further step of feature selection determine relevant information for the classification [11].

For each window, a feature extraction function is applied resulting in new features describing the activity. A wide range of features can be used depending on the modality of data acquisition. For signals for instance, statistical time domain and frequency domain feature extraction methods are used.

Raw signals can be used for both fall detection and classification with deep learning techniques avoiding feature extraction and feature selection steps. Wang et al. [19] point out some advantages of using deep learning for human activity recognition given that features can be learned automatically instead of manually designed. Deep generative models are able to exploit unlabeled data for model training.

Feature Selection

Many different features can be extracted from each time window. Nevertheless, the higher the dimensionality of the feature space, more training is needed and more computationally intensive becomes the classification [4]. Some features in the feature dataset might be irrelevant or redundant. The goal of automatic feature selection methods is to determine which features have more predictive power for classification in order to decrease the dimensionality of the feature space. A good introduction for feature selection and ranking is presented in [7].

Classification

Commonly used algorithms for fall detection are threshold-based, rule-based and shape-based, and machine learning techniques. Conventional approaches (i.e. threshold-based, rule-based and shape-based) are simpler to implement an less computationally expensive, but the rate of false positives is an issue [10]. Machine learning approaches obtain better performance results. The recent trend is to use machine learning methods for fall detection and classification [20].

The most frequently used evaluation metrics for fall detection and classification are accuracy,precision, sensitivity, specificity and F-measure.

2.2 Fall Detection and Classification Systems

Almost all of the simulated falls datasets consider more than one types of fall, however, very few fall detection systems actually identify and discriminate between each category of falls [1].

Hsieh et al. [8] presented a fall characteristics collection system for fall prevention strategies based on threshold methods. They analyzed data from a tri-axial accelerometer worn in the waist of six young adults. Their approach allows to classify eight different types of falls. Aziz and Robinovitch [2] proposed a system based for determining the causes of falls based on three-dimensional accelerometers worn in different body placements. Sixteen youn adults simulated falls due to slips, trips and other imbalance causes. In [1], Albert et al. presented a solution for fall classification based on accelerometers embedded in mobile phone. For their study 15 subjects simulated four different types of falls–left and right lateral, forward trips, and backward slips–while wearing mobile phones. Activities of daily living were recorded from nine subjects for ten days. They used Five different classification algorithms for detection and classification: Naïve Bayes, Decision trees, Support Vector Machines, Sparse Multinomial Logistic Regression, and k-nearest neighbors. Their results obtained 98% accuracy for fall detection and 99% accuracy for fall classification. Vavoulas et al. [18] developed a dataset based on inertial sensors from smartphones to enable the comparison of fall detection, fall classification and human activity recognition algorithms. They collected data from 24 volunteers who recorded four types of falls and nine activities of daily living.

3 Description of the UP-Fall Detection Dataset

As described above, this chapter presents the pipeline of multimodal fall classification and detection systems. To do so, this chapter uses the UP-Fall Detection dataset [12] for that purpose.

This is a public large dataset comprising a set of 44 raw sensor- and camera-based signals, of recordings of non-overlapping simple human daily activities and falls. These actions were performed by 17 healthy young subjects without any impairments (1.66 ± 0.05 m height and 66.8 ± 12.88 kg weight), 9 males and 8 females, ranging from 18 to 24 years old. A total of 11 activities/falls were recorded, during three attempts (trials). The dataset provides five types of falls (falling forward using hands, falling forward using knees, falling backward, falling sideward, and falling attempting to sit in an empty chair) and six daily activities (walking, standing, sitting, picking up an object, jumping, and laying), as summarized in Table 1. All falls were simulated by self-generation of the subjects. These falls were collected in the same direction (right-to-left).

The dataset was collected with three different modalities: wearable sensors, ambient sensors and cameras. Five inertial measurement units (IMUs) of three-axis accelerometer, three-axis gyroscope and one ambient light sensor, were placed in the body of the subjects, i.e. neck, waist, left wrist, right pocket and left ankle. Also, a brainwave sensor was located in the forehead. The ambient sensors comprised six pairs of infrared proximity devices, in grid formation, were placed around the central location of the action performances, such that they can detect the presence or absence of a person in the environment. Lastly, two cameras, one in lateral view and one in front view of the motion of the subjects, were located in the scene. After synchronization, cleaning and pre-processing, the dataset comprised 296,364 samples of raw

Table 1 Types of activities and falls in the dataset

Type	Description	Activity ID
Fall	Forward using hands	1
	Forward using knees	2
	Backward	3
	Sideward	4
	Attempting to sit in an empty chair	5
Daily activity	Walking	6
	Standing	7
	Sitting	8
	Picking up an object	9
	Jumping	10
	Laying	11

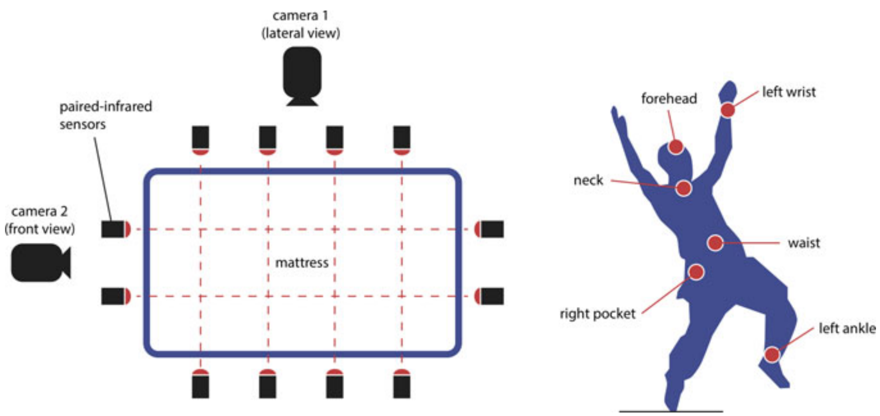


Fig. 2 Layout of the sensors and cameras used in the UP-Fall Detection dataset, adopted from [12]

sensor and camera signals, collected at 18 Hz. Figure 2 shows the layout of all the sensors and cameras used for creating the dataset.

The dataset is organized in two parts. The first part contains all the raw sensor signals and camera images. The second part contains a processed dataset of features. These features were extracted from the raw sensor signals in three different windowing sizes: 1-second, 2-second and 3-second time lengths. This windowing process was done with 50% of overlapping. The features correspond to twelve measurements in time: mean, standard deviation, root mean square, maximal amplitude, minimal amplitude, median, number of zero-crossing, skewness, kurtosis, first quartile, third quartile and median auto-correlation. In addition, six frequency features over the fast Fourier transform of the signals were also extracted: mean, median, entropy, energy, principal frequency and spectral centroid. Similarly, visual features were extracted from the raw images using a simple optical flow method, as reported in [12]. These visual features were extracted between two sequential images. After that, 1-second, 2-second and 3-second (with 50% of overlapping) windowing procedures were also computed.

For a complete description of the UP-Fall Detection dataset, it can be found in [12]. The dataset is publicly available at: http://sites.google.com/up.edu.mx/har-up/.

4 Open Code for Fall Detection and Classification

To simplify the process of developing a fall detection and classification system, we created a set of open codes in Python to download, handle and test our Fall-UP Detection dataset. These codes can be found in our GitHub repository: https://github.com/jpnm561/HAR-UP.

4.1 Setting up the Environment

Before starting, it is important to clone or download the repository. To do so, simply go to the GitHub repository https://github.com/jpnm561/HAR-UP and click on the *Clone or donwload* button. Copy all the files in your local computer and open Python.

After copying the files in a computer, the repository will be organized in seven folders, as follows:

- **Database download –** codes for accessing to the dataset repository.
- **Feature extraction –** codes for using our own feature dataset from sensor signals.
- **Camera OF (optical flow) –** codes for using the optical flow based visual features.
- **Binary features –** codes for converting the feature dataset useful to fall detection systems.
- **Feature selection –** codes for reducing the dimensions of the feature dataset.
- **Training –** codes for training machine learning models using a simple approach of splitting data into train and test sets.
- **K-cross validation –** codes for training machine learning models using the k-fold cross-validation approach.

We will work with different codes stored in these folders. We assume that readers are familiar with Python and the `scikit-learn` library.

4.2 Useful Definitions

We define some useful variables to manage and query the dataset. Throughout this chapter, the following variables are present:

`n_sub` – integer array indicating the first and last subject(s) when calling a function
`n_act` – integer array indicating the first and last activities when calling a function
`n_trl` – integer array indicating the first and last trial(s) when calling a function.

As seen later, features were extracted using a windowing method. In this case, the window length and the overlapping size are required to use the data. Thus, the following variable is considered:

`t_window` – string array indicating the window length and overlapping size in seconds

Three options can be used:

`'1&0.5'` – it refers to 1-second window length with 0.5 seconds on overlapping
`'2&1'` – it refers to 2-second window length with 1 second on overlapping
`'3&1.5'` – it refers to 3-second window length with 1.5 seconds on overlapping.

We integrate four different machine learning models for classification and detection. The variable associated to the type of model is:

`methods` – string array indicating the acronym of the model.

Four options can be used:

`'RF'` – random forest
`'SVM'` – support vector machines
`'MLP'` – multilayer perceptron
`'KNN'` – k-nearest neighbors.

Lastly, one of the goals of the UP-Fall Detection dataset is to provide different modalities while gathering data during the experiments. In that sense, it is important to retrieve data associated to one or a combination of more of these modalities. We use the following variable to set the modes:

`concept` – string array indicating the type of modality or combination of those.

Four options can be used:

`'IMU'` – inertial measurement units (all places in body)
`'Head'` – brainwave signal from helmet
`'Vision'` – cameras
`'IR'` – infrared sensors as ambient measures.

Combinations of concepts can be done using the underscore symbol (_) between concepts. For example, `'IMU_Head_IR'` means the combination of all IMU signals and the brainwave signal and the infrared sensors.

In addition, the IMU sensors can be retrieved separately from the location into the body, using the following string concepts:

`'IMU-Ankle'`	– IMU signals from left ankle location
`'IMU-Belt'`	– IMU signals from waist location
`'IMU-Neck'`	– IMU signals from neck location
`'IMU-RightPocket'`	– IMU signals from right pocket location
`'IMU-Wrist'`	– IMU signals from left wrist location.

4.3 Dataset Download Process

The UP-Fall Detection dataset is hosted on Google Drive. Thus, it is requiured to set up the Google Drive's API in Python. To do so, it needs the installation of `PyDrive` in the local Python environment (further information can be found in https://pythonhosted.org/PyDrive/). This can be done by running the following command in the Python terminal:

```
$ pip install PyDrive
```

After that, it is required to enable Google Drive's API in a personal Google account, make a project and get a client ID and client secret (i.e. these can be downloaded in a JSON file as **'client_secrets.json'**).[1]

To avoid errors, and constant authorization checks via browser, it is encouraged to make a YAML file called **'settings.yaml'** in the root directory, where **'client_secrets.json'** file should also be stored. This YAML file should have the following content:

```
client_config_backend: file
client_config:
    client_id: #your client id should be here
    client_secret: #your client secret should be here

save_credentials: True
save_credentials_backend: file
save_credentials_file: credentials.json

get_refresh_token: True
```

[1] For further instructions on how to enable the Google Drive's API, go to: https://pythonhosted.org/PyDrive/quickstart.html.

```
oauth_scope:
    - https://www.googleapis.com/auth/drive
    - https://www.googleapis.com/auth/drive.install
```

4.3.1 Downloading All Dataset

To download the whole dataset, including all IMU signals, brainwave signals, infrared sensor signals and videos from cameras, it can be done using the file **'DataBase-Download** » **Downloader_pydrive.py'**. The following instructions will download the dataset in the directory specified in the *ParentFolder* folder of `path`:

```
path = 'ParentFolder//'

dataBaseDownload(path)
```

After completion, all files will be located in `path` and the program will print the output files downloaded, as follows:

```
ParentFolder\
            \Subject#\
                     \Activity#\
                               \Trial#\
                                      \downloadedFile(1)
                                      ...
                                      \donwnloadedFile(i)
```

From the above, `dataBaseDownload()` downloads the raw dataset. But, it can be called the `featureDownload()` function to download the feature dataset (see Sect. 4.4).

4.3.2 Downloading the Dataset by Subject, Activity and Trial

By default, `dataBaseDownload()` retrieves the whole dataset. However, it is also possible to download specific chunks of data using the variables `n_sub`, `n_act` and `n_trl` for subjects, activities and trials, respectively. These variables are integer arrays representing the start and end indexes of any of these variables, like:

`n_sub = [start,end]` – subset from subject `start` to subject `end`
`n_act = [start,end]` – subset from activity `start` to activity `end`
`n_trl = [start,end]` – subset from trl `start` to trl `end`.

The possible integers to use in these variables depend on the number of subjects, activities and trials. Thus, the possible numbers of `n_sub` are `1-17`. The numbers of `n_act` are `1-5` for types of falls, `6-11` for daily activities and `20` for other activities. And, the numbers of `n_trl` are `1-3`.

For example, to get the data for subjects 1, 2, 3, 7, 10 and 11, it is possible to use the following commands:

```
dataBaseDownload(path, n_sub=[1,3])
dataBaseDownload(path, n_sub=[7,7])
dataBaseDownload(path, n_sub=[10,11])
```

To get the data for subject 1 with activities 4, 5, 6 and 7 for trials 2 and 3, then the following code will do it:

```
dataBaseDownload(path, n_sub=[1,1], n_act=[4,7], n_trl=[2,3])
```

4.3.3 Downloading the Dataset by Modality

The dataset is organized in raw signals acquired from sensors and those acquired from cameras. This information is containing in two formats as below:

- **CSV files –** these contain sensor data from five IMUs, one brainwave sensor, and six infrared sensors.
- **ZIP files –** these contain recorded videos from two cameras (lateral and front views).

By default, `dataBaseDownload()` downloads all CSV and ZIP files. However, this can be modified when calling the function.

The following code downloads sensor signals in CSV files:

```
dataBaseDownload(path, cameras = False)
```

The following code, in contrast, downloads recorded videos in ZIP files:

```
dataBaseDownload(path, csv_files = False)
```

The following code downloads the recorded videos associated to one camera. It uses the variable `n_cam` with `[1,1]` representing the lateral view (camera 1) and `[2,2]` representing the frontal view (camera 2):

```
dataBaseDownload(csv_files=False, n_cam=[1,1])
dataBaseDownload(csv_files=False, n_cam=[2,2])
```

4.4 Feature Extraction

The UP-Fall Detection dataset also contains a feature dataset. This information was previously calculated and then added to the dataset repository. It considers temporal and frequential features, as reported and implemented in [12]. This dataset contains the features extracted in windows with overlapping of 50%. Three different window lengths were set up: 1-second, 2-second and 3-second. For each window configuration, the data was synchronized.

The following section explains how to download this feature dataset and how to manipulate it accordingly to prepare the data for any convenience. However, if a different feature extraction is required, then this should be customized from the raw dataset downloaded in Sect. 4.3.

4.4.1 Downloading the Feature Dataset

The feature dataset is organized into the following files:

- **CSV files for sensors –** it contains the features extracted from IMUs, brainwave sensor and infrared sensors. These files can be found for the three window length configurations.
- **ZIP files for cameras –** it contains the optical flow extracted from two consecutive images in the video recordings. The optical flow is stored in two decomposition motions: horizontal (u) and vertical (v).
- **CSV files for compressed images –** it contains the components of optical flow values arranged by columns, using a 20×20 resized images; for both cameras.
- **CSV files for mean compressed images –** it contains the mean components of optical flow values, per window. These values come from the 20×20 resized images; for both cameras. These files can be found for the three window length configurations.

To download the feature dataset, it can be done using the file **'DataBaseDownload » Downloader_pydrive.py'**. The following instructions will download the dataset in the directory specified in the *ParentFolder* folder of `path`:

```
path = 'ParentFolder//'

featureDownload(path)
```

All files will be located in `path` and the program will print the output filenames, as follows:

```
ParentFolder\
            \Subject#\
                     \Activity$\
                               \Trial%\
                                      \Subject#Activity$Trial%Features1&0.5.csv
                                      \Subject#Activity$Trial%Features2&1.csv
                                      \Subject#Activity$Trial%Features3&1.5.csv
                                      \CameraFeaturesSubject#Activity$Trial%.csv
                                      \Subject#Activity$Trial%CameraFeatures1&0.5.csv
                                      \Subject#Activity$Trial%CameraFeatures2&1.csv
                                      \Subject#Activity$Trial%CameraFeatures3&1.5.csv
```

It is important to notice that the above instruction downloads all CSV files. ZIP files are avoided on purpose.

It is possible to specify the window length configuration for downloading the dataset. This can be done using the variable `t_window`, like:

```
featureDownload(path,t_window = ['1&0.5','2&1'])
```

The above instruction will download the feature dataset only for 1-second and 2-second window length configurations.

4.4.2 Other Options to Download the Feature Dataset

The following code downloads the feature dataset avoiding features taken from sensor data:

```
featureDownload(path, csv_files=False)
```

To avoid resized optical flow files from cameras, do:

```
featureDownload(path, cameras=False)
```

To avoid the mean taken from the resized optical flow files:

```
featureDownload(path, feat_cam_OF=False)
```

To allow downloading the ZIP files from cameras:

```
featureDownload(path, Complete_OF=True)
```

Also, it is possible to choose a specific files associated to one camera, as follows:

```
featureDownload(path, Complete_OF=True, n_cam =[1,1])
featureDownload(path, Complete_OF=True, n_cam =[2,2])
```

In the last two code lines, the first only downloads information from camera 1 (lateral view) and the second instruction only downloads data from camera 2 (front view).

4.4.3 Crafting the Feature Dataset

If the above feature dataset is not convenient, there is also possible to create a custom feature dataset. Thus, it is possible to manipulate the raw dataset (see Sect. 4.3) for preparing the features for further processing. This manipulation can be done in terms of the window length configuration, the types of sensors, features, subjects, activities and trials. The following explanation uses the file **'FeatureExtraction » featureExtraction.py'**.

The basic instruction for extracting features is `extraction()`, and it requires two specify both the path of the raw dataset and the path in which the features will be stored in the local computer:

```
d_base_path   = 'ParentFolder_raw//'
features_path = 'ParentFolder_features//'
extraction(d_base_path,features_path)
```

The parent folder of the raw dataset will remains unchangable, but a new parent folder will be created to storing the feature dataset, as shown in the output log:

```
ParentFolder_raw\
            \Subject#\
                     \Activity$\
                               \Trial%\
                                      \Subject#Activity$Trial%.csv
ParentFolder_features\
            \FeaturesTIMEWINDOW.csv
            \Subject#\
                     \Activity$\
                               \Trial%\
                                      \Subject#Activity$Trial%FeaturesTIMEWINDOW.csv
```

By default, features are calculated using the three window length configurations: `'1&0.5'`, `'2&1'` and `'3&1.5'`. However, it is possible to alter these calculations when calling the function `extraction()` to compute one or a combination of window length configurations. For example, the following code can be used in calculation of 1-second and 2-second window length configurations:

```
extraction(d_base_path,features_path,t_window = ['1&0.5','2&1'])
```

Moreover, it is possible to use the variable `t_window` to compute new window lengths using the instruction:

```
t_window = 'w&o'
```

Where, `w` refers to an integer value corresponding to the window length and `o` means a float value corresponding to the half of value of `w` with one decimal. For example, if it is required to calculate feature extraction in 4-second windows (and 50% of overlapping), then the following command will do:

```
extraction(d_base_path,features_path,t_window = ['4&2'])
```

Furthermore, the function `extraction()` extracts 18 features from 42 sensors (no cameras). But if required, it can be limited to calculate some of the features from some of the sensors. To do so, it should be modified the file **'FeatureExtraction » SFt_List.py'** which contains the whole list of sensors and features computed.

The extraction of features can also be limited to a subset of subjects, activities and/or trials. The following code exemplifies how to modify the `extraction()` function:

```
extraction(d_base_path,features_path, n_sub=[1,1], n_act=[4,7], single_f=False)
extraction(d_base_path,features_path, n_sub=[4,7], n_act=[4,7], single_f=False)
```

The code above will extract features from subjects 1, 4, 5, 6 and 7, only for activities 4, 5, 6 and 7 and the trials 1, 2 and 3. Notice that there is a flag value `single_f` that specifies that for each execution of `extraction()` the output files should be retained. In other words, if this that value is not set to `False`, then the output file from the first code line will be overwritten by the second output file.

4.5 Feature Selection

Feature selection is an optional step. However, it is recommended for minimizing the training data size and for selecting the most predictive features in the models.

A feature selection procedure is implemented in the GitHub repository, using the file **FeatureSelection** » **FeaturePreSelection.py**. It is one possibility to get reducing the number of features in the dataset. In this case, the procedure consists of doing three experiments that independently rank the features by the predictive power over a built model.

These experiments are: (i) the creation of an extra trees classifier with 250 estimators and no random state, (ii) a linear support vector machines model with L2-regularization as penalty, and (iii) a recursive feature elimination with random forest as classifier.

Then, an array of 100 features (or 30% of all features if there are less than a hundred) by each experiment are returned. These arrays are compared and, then, it shows the selected features and the frequency that they appeared in the arrays. At last, the most frequent features (i.e. those selected in two or more experiments) are selected. This can be simply done by calling the `preSelection()` function.

```
preSelection(concept)
```

The variable `concept` refers to a list of strings representing the types of modalities or specific sensors required for the experimentation (see Sect. 4.2). For example, the next instruction specifies that only the features corresponding to the IMUs from left ankle and right pocket are taken into account in the feature selection, in order to evaluate which features are better for each concept:

```
concept = ['IMU-Ankle','IMU-RightPocket']
preSelection(concept)
```

At each experiment, a model is built and the features are evaluated by adding one-by-one. To do so, the GitHub repository implements the random forest model using the file **FeatureSelection** » **RandomForest_Selection.py**. To measure the accuracy of the model, a score is calculated using the function `sel_Scores()`. By default, scores are computed as a binary classification. But, it can be modified for multiclass classification, as shown below:

```
sel_Scores(concept,binary=False)
```

The following example shows how to modify the file **RandomForest_Selection.py** for implementing a random forest using two windows: 1-second window, taken every 0.5 seconds; and 2-second window taken every second. This example uses a non-binary classification dataset and two experiments: selected IMU features (`'IMU'`) and selected right-pocket IMU features (`'IMU-RightPocket'`).

```
def main():
    concept = ['IMU','IMU-RightPocket']
    sel_RF(concept, t_window=['1&0.5','2&1'])
    sel_Scores(concept, t_window=['1&0.5','2&1'], binary=False)
if __name__ == "__main__":
    main()
```

It is important to say that this file uses the pre-selection features computed with `preSelection()`. Thus, after using this function, the output files of the pre-selected features should be located in the following path:

```
ParentFolder\
            \createFolder.py
            \RandomForest_Selection.py
            \IMU\
                \1&0.5\
                      \PreSelectedFTS_1&0.5_IMU.csv
                \2&1\
                    \PreSelectedFTS_2&1_IMU.csv
            \IMU-RightPocket\
                            \1&0.5\
```

```
                        \PreSelectedFTS_1&0.5_IMU-RightPocket.csv
                  \2&1\
                      \PreSelectedFTS_2&1_IMU-RightPocket.csv
```

At the end of execution of the file **RandomForest_Selection.py**, a set of output files will be created, as shown below:

```
ParentFolder\
            \IMU\
                \1&0.5\
                      \PreSelectionReport_1&0.5_IMU.csv
                      \PreSelectionReport_1&0.5_IMU.png
                      \PreSel_RF_outputs\
                                        \Output1.csv
                                        \Output2.csv
                                        ...
                                        \OutputN.csv
                \2&1\
                    \PreSelectionReport_2&1_IMU.csv
                    \PreSelectionReport_2&1_IMU.png
                    \PreSel_RF_outputs\
                                       \Output1.csv
                                       \Output2.csv
                                       ...
                                       \OutputN.csv
            \IMU-RightPocket\
                            \1&0.5\
                                  \PreSelectionReport_1&0.5_IMU-RightPocket.csv
                                  \PreSelectionReport_1&0.5_IMU-RightPocket.png
                                  \PreSel_RF_outputs\
                                                    \Output1.csv
                                                    \Output2.csv
                                                    ...
                                                    \OutputN.csv
                            \2&1\
                                \PreSelectionReport_2&1_IMU-RightPocket.csv
                                \PreSelectionReport_2&1_IMU-RightPocket.png
                                \PreSel_RF_outputs\
                                                 \Output1.csv
                                                 \Output2.csv
                                                 ...
                                                 \OutputN.csv
```

4.6 Training a Fall Detection System

A fall detection system consists of determining if a person falls or not during a period of time. From the computational point of view, this is a binary classification problem between *fall* or *no-fall*. In this regard, the training of a binary classifier model is required. This is done using the file **Training** » **BC_Training.py**. In this repository, the raw or feature dataset is split into 70% for training and 30% for testing.

The following example shows how to call the function for training the model using only Random Forest (RF) in two different windows: 1-second and 2-second window length configurations. Two experiments are set up using all features related to the IMU signals ('IMU') and only the selected right pocket IMU features ('IMU-RightPocket').

```
def main():
    concept = ['IMU','IMU-RightPocket']
    BC_Training(concept, t_window=['1&0.5','2&1'], methods=['RF'])
    BC_Scores(concept, t_window=['1&0.5','2&1'], methods=['RF'])
if __name__ == "__main__":
    main()
```

After the execution of the file **BC_Training.py**, this will create a set of files in the following paths:

```
ParentFolder\
            \IMU\
                \AvgConfusionMatrix_RF_IMU.jpg
                \Score_Mean_IMU.jpg
                \Score_StandardDeviation_IMU.jpg
                \Score_IMU_temp.csv
                \1&0.5\
                      \AvgConfusionMatrix_1&0.5_RF_IMU.jpg
                      \Score_1&0.5_RF_IMU.csv
                      \RF\
                         \Result_1&0.5_RF_1.csv
                         \Result_1&0.5_RF_2.csv
                         ...
                         \Result_1&0.5_RF_10.csv
                \2&1\
                    \AvgConfusionMatrix_2&1_RF_IMU.jpg
                    \Score_2&1_RF_IMU.csv
                    \RF\
                       \Result_2&1_RF_1.csv
                       ...
                       \Result_2&1_RF_10.csv
            \IMU-RightPocket\
                            \AvgConfusionMatrix_RF_IMU-RightPocket.jpg
                            \Score_Mean_IMU-RightPocket.jpg
                            \Score_StandardDeviation_IMU-RightPocket.jpg
                            \Score_IMU-RightPocket_temp.csv
                            \1&0.5\
                                  \AvgConfusionMatrix_1&0.5_RF_IMU-RightPocket.jpg
                                  \Score_1&0.5_RF_IMU-RightPocket.csv
                                  \RF\
                                     \Result_1&0.5_RF_1.csv
                                     ...
                                     \Result_1&0.5_RF_10.csv
                            \2&1\
                                \AvgConfusionMatrix_2&1_RF_IMU-RightPocket.jpg
                                \Score_2&1_RF_IMU-RightPocket.csv
                                \RF\
                                   \Result_2&1_RF_1.csv
                                   ...
                                   \Result_2&1_RF_10.csv
```

As shown, the resulting validation data set are stored (in CSV files), the scores calculated (in CSV files), the confusion matrix (in JPG files) and the bar graphs of all scores (in JPG files).

4.6.1 Data Augmentation in Binary Classes

In the case of fall detection system, classification models might fail because the number of windows representing falls are more less than the number of windows representing no-falls. In such that case, it is important to implement strategies to balance the data for training classification models.

One implementation is data augmentation. In the GitHub repository, this is done by doubling the number of falls (i.e. it repeats the windows representing falls) and taking away randomly two-thirds of the windows representing no-falls. This procedure enhances the data for training and creates CSV files for this dataset.

Open the file **BinaryFeatures** » **incrementFalls.py**. It is possible to call the function `imcrementFalls` for creating an augmented dataset of particular sensors and/or window length configurations. The following example shows the code for data augmentation in two different windows: 1-second and 2-second window length configurations. Two experiments are set up using all features related to the IMU signals ('IMU') and only the selected right pocket IMU features ('IMU-RightPocket').

```
def main():
    concept = ['IMU','IMU-RightPocket']
    incrementalFalls(concept, t_window=['1&0.5','2&1'])
if __name__=="__main__":
    main()
```

4.7 Training a Fall Classification System

A fall classification system consists of determining the type of fall a person suffers (or the activity done) during a period of time. From the computational point of view, this is a multi-class classification problem. In this regard, the training of a multi-class classifier model is required. This is done using the file **Training** » **MC_Training.py**. In this repository, the raw or feature dataset is split into 70% for training and 30% for testing.

The following example shows how to call the function for training the model using only Random Forest (RF) in two different windows: 1-second and 2-second window length configurations. Two experiments are set up using all features related to the IMU signals ('IMU') and only the selected right pocket IMU features ('IMU-RightPocket').

```
def main():
    concept = ['IMU','IMU-RightPocket']
    MC_Training(concept, t_window=['1&0.5','2&1'], methods=['RF'])
    MC_Scores(concept, t_window=['1&0.5','2&1'], methods=['RF'])
if __name__ == "__main__":
    main()
```

The output files are the same as in the fall detection system, extended for multi-classification.

4.8 K-Fold Cross-Validation

It is also possible to train and validate a classification model. This can be implemented through the k-fold cross-validation. This method splits the raw or feature dataset in k subsets. On one iteration, $k-1$ subsets are used for training while the remaining set is used for testing. This process repeats k times, until all subsets are leaved once for testing. The files on **K-CrossValidation** provide an easy way to implement this process.

The following example shows how to call the functions for splitting in $k = 20$ subsets and then training the model using only Random Forest (RF) in two different windows: 1-second and 2-second window length configurations. Two experiments are set up using all features related to the IMU signals ('IMU') and only the selected right pocket IMU features ('IMU-RightPocket').

First, modify the file **K-CrossValidation** » **k-crossvalidation.py** as follows:

```
def main():
     concept = ['IMU','IMU-RightPocket']
     k_crossFiles(concept, t_window=['1&0.5','2&1'], K=20)
if __name__=="__main__":
     main()
```

After that, a set of output files will be stored in the local computer. Then, modify and execute the file **K-CrossValidation** » **Training_function.py**:

```
def main():
    concept = ['IMU','IMU-RightPocket']
    training(concept, t_window=['1&0.5','2&1'], methods=['RF'], K=20)
if __name__=="__main__":
    main()
```

5 Example of a Fall Classification System

This example shows how to make the workflow for a fall classification system using the UP-Fall Detection dataset and the GitHub repository presented in this chapter. The example supposes the usage of the feature extraction already done in the dataset. Then, a feature selection is done and a set of experiments are specified. After that, several multi-class classification models are trained. Finally, the metrics and some graphs are presented to understand the process.

First, download the complete feature dataset (without the ZIP files of video recordings), over the three window length configurations, using the `featureDownload` function specifying a *ParentFolder* path. A set of output files will be stored in this path. Then, specify the types of sensors required in the experiments by using the `preSelection` function. In this example, we show how to define the combination of features from IMU sensors, the brainwave sensor and both cameras. After that, a feature selection is performed using the `sel_Scores` and `sel_RF` functions that will select the best features in the whole dataset based on the accuracy of a predefined random forest model. Finally, the training procedure is done over four machine learning models (i.e. RF, MLP, SVM, KNN) and the metrics are calculated.

The following code summarizes the whole process for fall classification using the UP-Fall Detection dataset. In addition, Fig. 3 shows the mean and standard deviation of the training models done 10 times. Table 2 summarizes the same scores. In addition, a sample of the averaged confusion matrices for each classification model in the 1-second window length configuration are shown in Figs. 4 and 5.

This simple example shows the whole workflow in the development of a fall classification system. As notice, the set of instructions using the GitHub repository is very concise, but it leads a powerful tool for feature extraction, feature selection, training classification models and evaluating the results. It is also possible to configure different experiments and compare them easily. As shown in this example, the MLP classification model is the best, in terms of the F_1-score metric, when using 1-second window length in the feature extraction. In terms of accuracy, RF model is preferable.

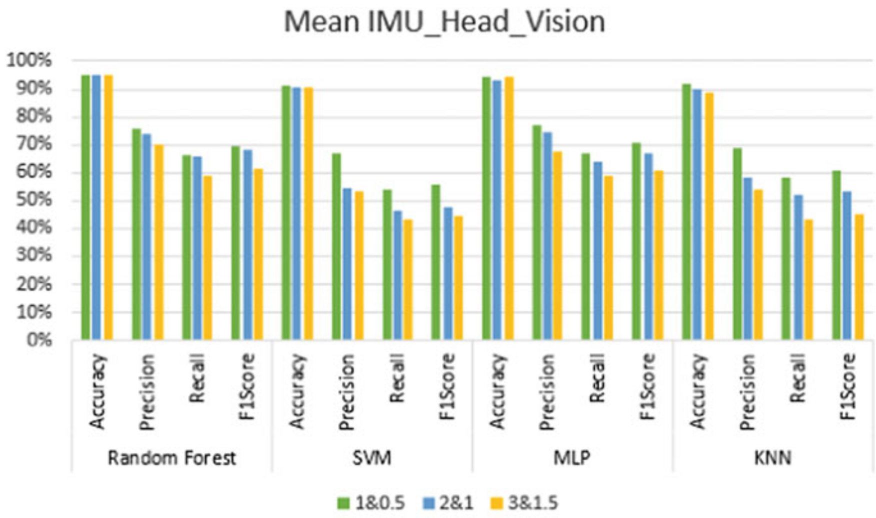
```
import DataBaseDownload.Downloader_pydrive as db
import FeatureSelection.FeaturePreSelection as fps
import FeatureSelection.RandomForest_Selection as fs
import Training.MC_Training as mc

# Download the feature dataset
path = 'FallClassificationSystem//'
db.featureDownload(path)

# Specify the combination of sensors
concept = ['IMU_Head_Vision']
fps.preSelection(concept)

# Selection of the most representative features
fs.sel_RF(concept)
fs.sel_Scores(concept, binary=False)

# Train the multi-class classifier models
methods = ['RF','SVM','MLP','KNN']
```



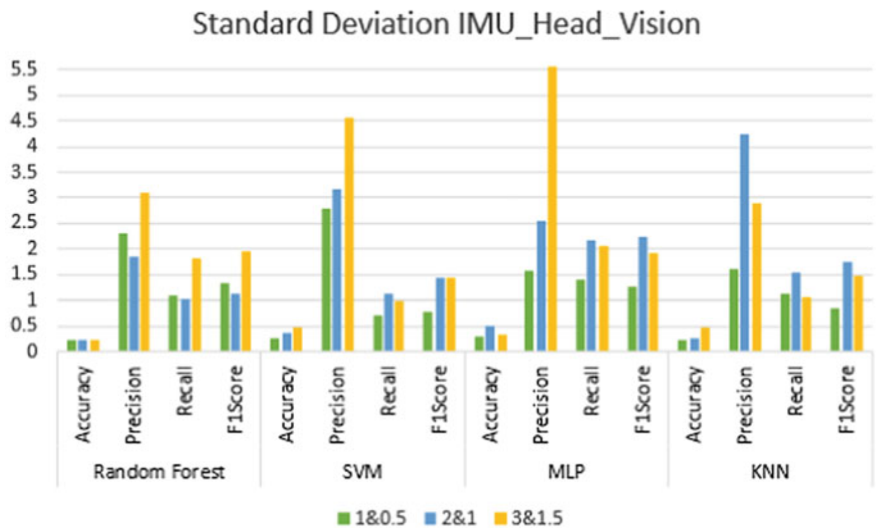


Fig. 3 Mean (top) and standard deviation (bottom) scores in the experiment using features from IMU and brainwave sensors, and cameras

Table 2 Scores of the testing models in the experiment. Values report the mean and standard deviation

Model	Accuracy (%)	Precision (%)	Recall (%)	F_1-score (%)
RF	95.09 ± 0.23	75.52 ± 2.31	66.23 ± 1.11	69.36 ± 1.35
SVM	91.16 ± 0.25	66.79 ± 2.79	53.82 ± 0.70	55.82 ± 0.77
MLP	94.32 ± 0.31	76.78 ± 1.59	67.29 ± 1.41	70.44 ± 1.25
KNN	92.06 ± 0.24	68.82 ± 1.61	58.49 ± 1.14	60.51 ± 0.85

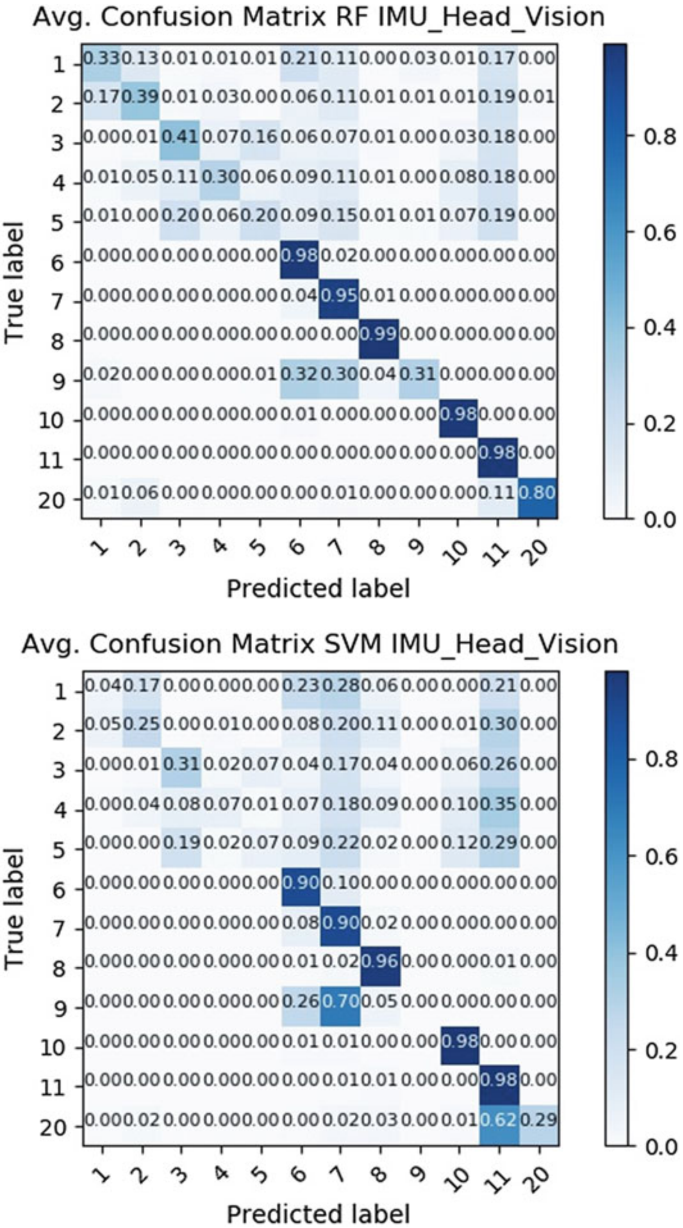


Fig. 4 Averaged confusion matrices of the classification models RF and SVM in the 1-second window length configuration

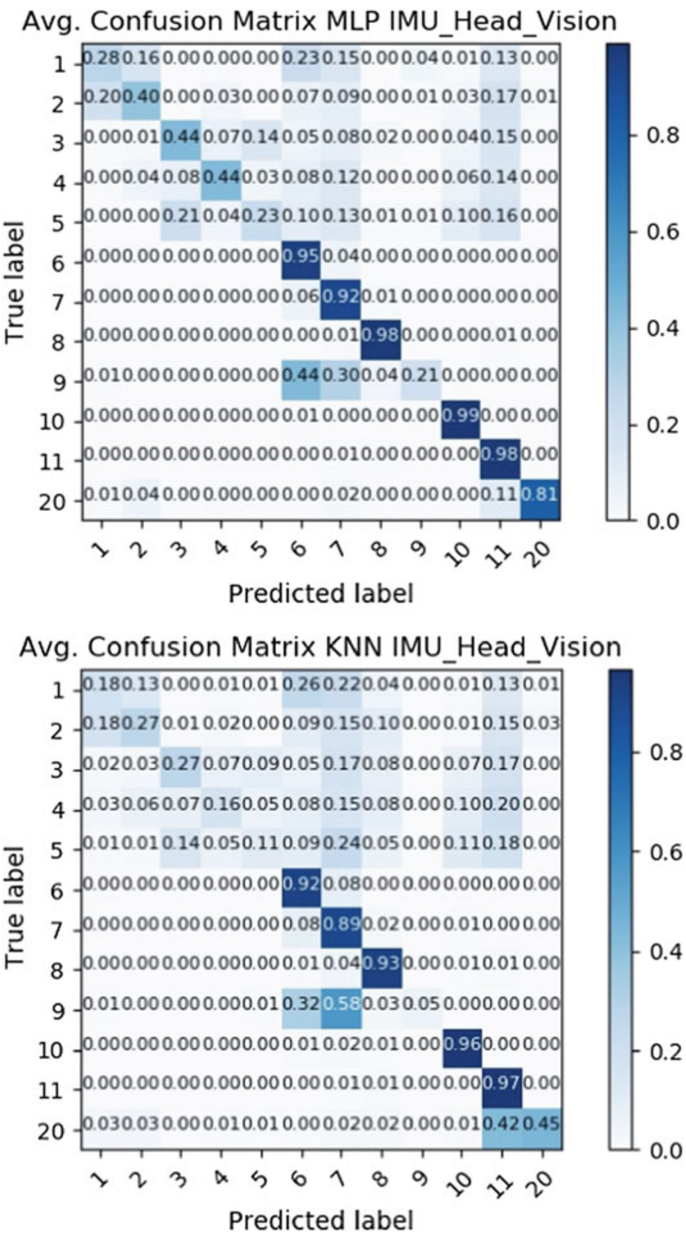


Fig. 5 Averaged confusion matrices of the classification models MLP and KNN in the 1-second window length configuration

```
mc.MC_Training(concept, methods)

# Calculate scores and generate graphs
mc.MC_Scores(concept, methods)
```

6 Discussion

This chapter proposed an open source implementation for fall classification and detection systems using the public UP-Fall Detection dataset. This implementation comprises a set of open codes stored in a GitHub repository for full access to anyone interested on developing these monitoring systems. Moreover, the chapter provides a tutorial for using the codes and a concise example for their application.

As described earlier, the proposed open source implementation is one of the most complete source coding found on the web that is specialized on human fall classification and detection systems. Moreover, the dataset used here provides sufficient intuition about different multimodal approaches (i.e. wearables, ambient sensors and vision-based sources) and multiple machine learning models.

The material presented in this chapter follows the typical workflow in fall classification and detection systems, from data acquisition through feature extraction, feature selection and training models, to model validation. In this regard, our open source implementation covers all these steps to fully understand the entire process.

Some benefits of this open source implementation can be highlighted. First, the implementation gives a global glimpse of the human activity recognition workflow typically used in health monitoring, sports, rehabilitation and ambient assisted living approaches when dealing with the discovery of human activities in the daily lives. Then, it provides sufficient utilities at each step for customizing downloading of dataset, crafting feature extraction, windowing or resampling, selecting features, determining different experiments for benchmarking, training different classification models, considering detection and classification performances, and evaluating with standard metrics. These utilities are potentially powerful for designing and developing real-time and real-world fall detection and classification systems. This implementation also uses many machine learning functions from the *scikit-learn* library, allowing the adoption of the framework easily. Also, the open source implementation exposes the necessary documentation to start coding easily, and it is complemented with the information presented in this chapter. However, it is important to remark that this open code implementation works directly with the public UP-Fall Detection dataset. However, other datasets cannot be implemented directly. If required, it is necessary to adapt the codes for that purpose.

Lastly, this open source implementation has released to promote sharing and reusing our UP-Fall Detection dataset; but also, to asses and improve this dataset. We also consider that it will contribute to software developers community in the field of human fall detection and classification systems.

Acknowledgements This research has been funded by Universidad Panamericana through the grant "Fomento a la Investigación UP 2018," under project code UP-CI-2018-ING-MX-04.

References

1. Albert, M.V., Kording, K., Herrmann, M., Jayaraman, A.: Fall classification by machine learning using mobile phones. PloS one **7**(5), e36556 (2012)
2. Aziz, O., Robinovitch, S.N.: An analysis of the accuracy of wearable sensors for classifying the causes of falls in humans. IEEE Trans. Neural Syst. Rehabil. Eng. **19**(6), 670–676 (2011)
3. Bello, Y.: BIOBOT: a fall detection system (FDS) using artificial intelligence (2019). https://github.com/SpaceMinds/FDS
4. Bulling, A., Blanke, U., Schiele, B.: A tutorial on human activity recognition using body-worn inertial sensors. ACM Comput. Surv. (CSUR) **46**(3), 1–33 (2014)
5. Feng, C.Z.Y.: Real-time ADLs and Fall Detection implement TensorFlow (2019). https://github.com/chizhanyuefeng/Realtime-Fall-Detection-for-RNN
6. Feng, C.Z.Y.: ADL and fall detection convolutional neural (2019). https://github.com/chizhanyuefeng/FD-CNN
7. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**(Mar), 1157–1182 (2003)
8. Hsieh, C.J., Chiang, C.Y., Huang, C.N., Chan, C.T., Hsu, S.J.: Development of fall characteristics collection system for fall prevention strategies. In: Proceedings of the 5th International Conference on Bioinformatics and Biomedical Engineering, Wuhan, China, pp. 10–12 (2011)
9. Hu, Y., Zhang, J., Bai, X., Yu, S., Yang, Z.: Influence analysis of github repositories. SpringerPlus **5**(1), 1268 (2016)
10. Igual, R., Medrano, C., Plaza, I.: Challenges, issues and trends in fall detection systems. Biomed. Eng. Online **12**(1), 66 (2013)
11. Lara, O.D., Labrador, M.A.: A survey on human activity recognition using wearable sensors. IEEE Commun. Surv. Tutor. **15**(3), 1192–1209 (2012)
12. Martínez-Villaseñor, L., Ponce, H., Brieva, J., Moya-Albor, E., Núñez-Martínez, J., Peñafort-Asturiano, C.: UP-fall detection dataset: a multimodal approach. Sensors **19**(9), 1988 (2019)
13. Padmanaban, V.: Detect a person fall in videos using OpenCv (2017). https://github.com/swift2891/Fall_Detection
14. Pannurat, N., Thiemjarus, S., Nantajeewarawat, E.: Automatic fall monitoring: a review. Sensors **14**(7), 12900–12936 (2014)
15. Santiago, E.: Detect a person fall in videos using OpenCv (2018). https://github.com/EikeSan/video-fall-detection
16. Sucerquia, A., López, J.D., Vargas-Bonilla, J.F.: Sisfall: a fall and movement dataset. Sensors **17**(1), 198 (2017)
17. Tinetti, M.E., Doucette, J., Claus, E., Marottoli, R.: Risk factors for serious injury during falls by older persons in the community. J. Am. Geriatr. Soc. **43**(11), 1214–1221 (1995)
18. Vavoulas, G., Pediaditis, M., Chatzaki, C., Spanakis, E.G., Tsiknakis, M.: The mobifall dataset: fall detection and classification with a smartphone. In: Artificial Intelligence: Concepts, Methodologies, Tools, and Applications, pp. 1218–1231. IGI Global (2017)
19. Wang, J., Chen, Y., Hao, S., Peng, X., Hu, L.: Deep learning for sensor-based activity recognition: a survey. Pattern Recognit. Lett. **119**, 3–11 (2019)
20. Xu, T., Zhou, Y., Zhu, J.: New advances and challenges of fall detection systems: a survey. Appl. Sci. **8**(3), 418 (2018)
21. Zagalsky, A., Feliciano, J., Storey, M.A., Zhao, Y., Wang, W.: The emergence of github as a collaborative platform for education. In: Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, pp. 1906–1917. ACM (2015)

Detecting Human Activities Based on a Multimodal Sensor Data Set Using a Bidirectional Long Short-Term Memory Model: A Case Study

Silvano Ramos de Assis Neto, Guto Leoni Santos, Elisson da Silva Rocha, Malika Bendechache, Pierangelo Rosati, Theo Lynn and Patricia Takako Endo

Abstract Human falls are one of the leading causes of fatal unintentional injuries worldwide. Falls result in a direct financial cost to health systems, and indirectly, to society's productivity. Unsurprisingly, human fall detection and prevention is a major focus of health research. In this chapter, we present and evaluate several bidirectional long short-term memory (Bi-LSTM) models using a data set provided by the Challenge UP competition. The main goal of this study is to detect 12 human daily activities (six daily human activities, five falls, and one post-fall activity) derived from multi-modal data sources - wearable sensors, ambient sensors, and vision devices. Our proposed Bi-LSTM model leverages data from accelerometer and gyroscope sensors located at the ankle, right pocket, belt, and neck of the subject. We

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-38748-8_2) contains supplementary material, which is available to authorized users.

S. R. de Assis Neto · E. da Silva Rocha · P. Takako Endo (✉)
Universidade de Pernambuco, Pernambuco, Brazil
e-mail: patricia.endo@upe.br

S. R. de Assis Neto
e-mail: silvano.neto@upe.br

E. da Silva Rocha
e-mail: esr2@ecomp.poli.br

G. L. Santos
Universidade Federal de Pernambuco, Pernambuco, Brazil
e-mail: gls4@cin.ufpe.br

M. Bendechache · P. Rosati · T. Lynn
Irish Institute of Digital Business, Dublin City University, Dublin, Ireland
e-mail: malika.bendechache@dcu.ie

P. Rosati
e-mail: pierangelo.rosati@dcu.ie

T. Lynn
e-mail: theo.lynn@dcu.ie

H. Ponce et al. (eds.), *Challenges and Trends in Multimodal Fall Detection for Healthcare*, Studies in Systems, Decision and Control 273,
https://doi.org/10.1007/978-3-030-38748-8_2

utilize a grid search technique to evaluate variations of the Bi-LSTM model and identify a configuration that presents the best results. The best Bi-LSTM model achieved good results for precision and f1-score, 43.30 and 38.50%, respectively.

Keywords Bi-LSTM · Human falls · Multimodal sensors · Human activities

1 Introduction

Falls are a major global public health problem. Research by the World Health Organisation (WHO) suggests that every year, approximately 37.3 million falls are severe enough to require medical attention and that falls are the second leading cause of fatal unintentional injuries (approx. 646,000 per annum), second only to road traffic injuries [1]. While older people have the highest risk of death or serious injury arising from a fall, children are also a high risk group for fall injury and death due to their stage of development associated characteristics and 'risk-taking' behaviors [1, 2]. Falls result in a significant direct financial cost to health systems, both in terms of in-patient and long term care costs, but also in indirect costs resulting from lost societal productivity of the focal person and caregivers [2]. To illustrate this impact, falls are estimated to be responsible for over 17 million lost disability-adjusted life years in productivity per annum [1]. Furthermore, fear of falling not only contributes to a higher risk of falling but can result in indirect negative health consequences including reduction or avoidance of physical activity and psychological issues, which can contribute to a lower quality of life [3].

Unsurprisingly, fall detection and prevention is a major focus of public health initiatives and research. Preventative initiatives include clinical interventions, environmental screening, fall risk assessment and modification, muscle strengthening and balance retraining, assistive devices, and education programs [1, 2]. Fall detection systems include non-wearable (sometimes referred to as context-aware systems) and wearable systems whose main objective is to alert when a fall event has occurred [4]. Research on fall detection systems suggests that these systems both reduce the fear of falling and actual falls as well as mitigating negative consequences of falls due to faster fall detection and intervention in the instance of a fall [5].

Advances in low-cost sensing devices and their integration into both mobile and so-called 'smart' environments have accelerated research into human activity recognition (HAR). Researchers are increasingly able to draw on a combination of wearable devices and fixed location data sources to inform HAR research efforts by providing different perspectives of a given event or human activity [6]. Making sense of this heterogeneous multi-modal data is not without challenges, not least those presented by the volume, variety, and velocity of such time-series data but also the specific human activity being explored and the efficacy of a given HAR technique [7–10].

In this chapter, we present a deep learning model to detect falls using multi-modal sensor data. We propose a bidirectional long short-term memory (Bi-LSTM) model that leverages data accelerometer and gyroscope sensors located at the ankle, right pocket, belt, and neck of the subject. We propose two model configurations, one identified empirically and a second identified using a grid search technique.

The rest of this chapter is organized as follows. In Sect. 2, we describe the basic concepts of LSTM and Bi-LSTM. We then present the methodology applied in this study in Sect. 3, both describing the data set and the evaluation metrics. Section 4 describes our Bi-LSTM model and Sect. 5 presents the results achieved by our models. Section 6 briefly presents related work. We conclude with a summary of our work and directions for further research in Sect. 7.

2 Long Short-Term Memory (LSTM)

Deep learning networks, such as Multilayer Perceptron (MLP), Convolutional Neural Networks (CNN), and Radial Basis Function Networks amongst others, assume that all inputs are independent of each other. As such, they are not appropriate for time-series data related to human activities. Recurrent Neural Networks (RNNs) are able to overcome this limitation by using a recurrent connection in every neuron [11]. The activation of a neuron is fed back to the neuron itself in order to provide a memory of past activations and to learn the temporal dynamics of time-series data [11]. However, RNNs have limitations when it comes to discovering patterns over long temporal intervals [12] as they are subject to both exploding and vanishing gradient problems [13, 14]. While the former is relatively easy to address using gradient clipping [12, 15], vanishing gradient problems are more challenging [16]. Long short-term memory (LSTM) is a variant of traditional RNN which overcomes both problems [16]. LSTM networks make use of recurrent neurons with memory blocks, working with the concept of gates [11, 17]. While they overcome vanishing and exploding gradient problems, each unit of an LSTM requires intensive calculations resulting in long training times [14]. Figure 1 presents a basic schema of an LSTM block.

An LSTM network updates its block state according to gate activation. Thus, the input data provided to the LSTM network is fed into the gates that define which operation should be performed: write (input gate), read (output gate), or reset (forget gate). The mechanisms of these gates are based on component-wise multiplication of the input. The vectorial representation of each gate is as follows [11]:

$$i_t = \sigma_i(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{1}$$

$$f_t = \sigma_f(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{2}$$

$$c_t = f_t c_{t-1} + i_t\sigma_c(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{3}$$

$$o_t = \sigma_o(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{4}$$

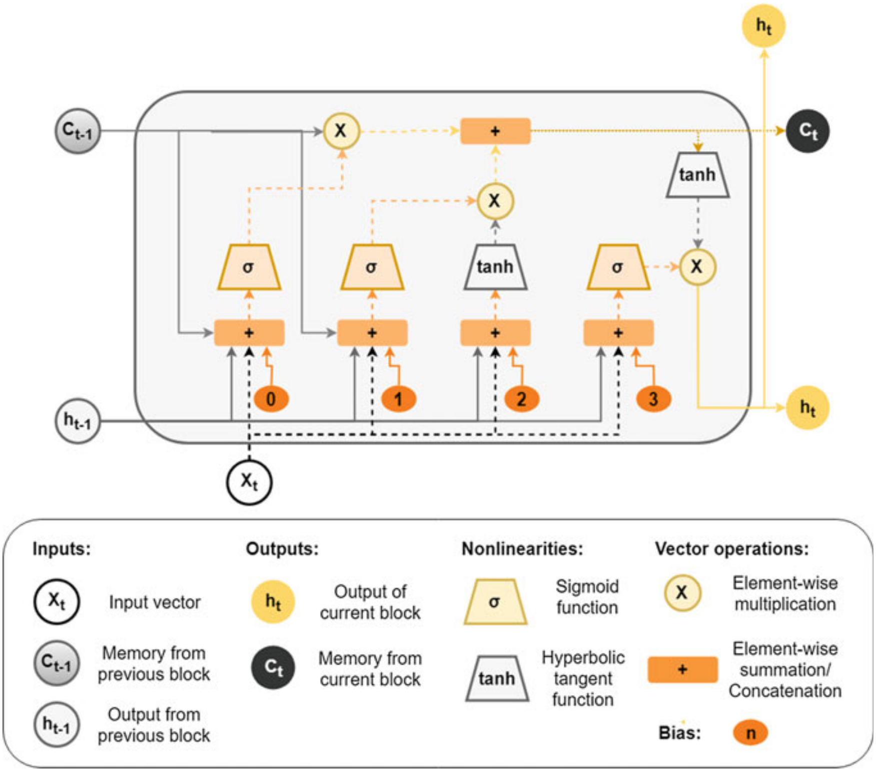


Fig. 1 Example of an LSTM block (adapted from [18])

$$h_t = o_t \sigma_h(c_t) \tag{5}$$

where i, f, o, and c represent the outputs of input gate, forget gate, output gate, and cell activation vectors, respectively; all of them have the same vector size h_t therefore defining the hidden value (i.e., the memory state of the block). σ_i, σ_f, and σ_o are, respectively, the non-linear functions of input, forget, and output gates. W_{xi}, W_{hi}, W_{ci}, W_{xf}, W_{hf}, W_{cf}, W_{xc}, W_{hc}, W_{xo}, W_{ho}, and W_{co} are weight matrices of the respective gates, where x and h are the input and the hidden value of LSTM block respectively. b_i, b_f, b_c, and b_o are the bias vectors of input gate, forget gate, cell, and output gate, respectively [11].

The main difference between an RNN and a feed-forward model is the ability of the RNN to consider past information at each specific time step [19]. However, in some use cases, a wider context must be taken into account. In speech recognition, for example, the correct classification and interpretation of a given sound depends on the proceeding phoneme [20]. Correct classification of other data types, such as text and time-series, also depends on both preceding and subsequent data.

Bidirectional RNNs (Bi-RNNs) are able to process both past and future information at each time step [21]. In order to do so, each hidden layer of a Bi-RNN is composed of two hidden layers i.e. one for processing the past time steps and another for processing future time steps. The outputs are then combined to compose a new output that is forwarded to the next hidden layers [19]. Therefore, the output of each time step includes more complete clues related to the wider context of each specific input data. For the study described in this chapter, we use Bi-LSTM, a type of Bi-RNN.

3 Methodology

3.1 The Data Set

The data set used for this study, UP-Fall Detection, was made available as part of the Challenge UP: Multi-modal Fall Detection competition [22, 23]. The data set includes five falls and six daily activities performed by 12 subjects (see Table 1). Subjects performed five different types of human falls (falling forward using hands, falling forward using knees, falling backwards, falling from a sitting position on an empty chair and falling sideward), six simple human daily activities (walking, standing, picking up an object, sitting, jumping, and lying down), and an additional activity labeled as "on knees" where a subject remained on their knees after falling.

Table 1 Description of activities [22].

Activity ID	Description
1	Falling forward using hands
2	Falling forward using knees
3	Falling backwards
4	Falling sideward
5	Falling from sitting in a chair
6	Walking
7	Standing
8	Sitting
9	Picking up an object
10	Jumping
11	Lying down
20	On knees

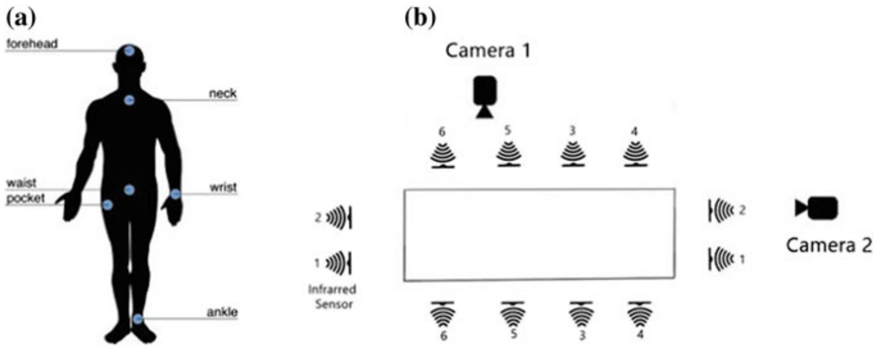


Fig. 2 Distribution of the sensors used to collect data: **a** Wearable sensors and EEG headset located on the human body, and **b** Layout of the context-aware sensors and camera views [22].

3.2 Data Collection

The data was collected using a multi-modal approach from wearable sensors, ambient sensors, and vision devices distributed as per Fig. 2. The experiments were conducted in a controlled laboratory environment in which light intensity did not vary; the ambient sensors and cameras remained in the same position during the data collection process.

For our study, we used data from five Mbientlab MetaSensor wearable sensors collecting raw data from a 3-axis accelerometer, a 3-axis gyroscope, and the ambient light value. These wearable sensors were located on the left wrist, under the neck, at the right trouser pocket, at the middle of the waist (on/in the belt), and at the left ankle. Also, data from one electroencephalograph (EEG) NeuroSky MindWave headset was used to measure the raw brainwave signal from a unique EEG channel sensor located at the forehead.

For context-aware sensors, six infrared sensors above the floor of the room measured the changes through interruption of the optical devices.

Lastly, two Microsoft LifeCam Cinema cameras were located above the floor, one for a lateral view and the other for a frontal view.

3.3 Evaluation Metrics

For the Challenge UP competition [22], the f1-score measure was used to evaluate proposed models, considering both precision and sensitivity (recall). The f1-score is calculated as shown in Eq. 6:

$$F1 = 2 \times \frac{Precision_\mu \times Sensitivity_\mu}{Precision_\mu + Sensitivity_\mu} \tag{6}$$

where $Precision_\mu$ is the average number of the number of true positives (TP) across all activities and falls divided by the sum of true positives (TP) and false positives (FP) (Eq. 7); and $Sensitivity_\mu$ is the average number of TP across all activities and falls divided by the sum of TP and false negatives (FN) (Eq. 8).

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Sensitivity = Recall = \frac{TP}{TP + FN} \tag{8}$$

In addition to the requirements of the Challenge UP competition outlined above, we also consider specificity and accuracy. While sensitivity is used to determine the proportion of actual positive cases predicted correctly and thus avoid false negatives, specificity is used to determine the proportion of actual negative cases predicted correctly i.e. the avoidance of false positives. Together, sensitivity and specificity provide a more informed decision on the efficacy of a given model. Specificity is calculated as the average number of true negatives (TN) divided by the sum of TN and FP (Eq. 9).

$$Specificity = \frac{TN}{TN + FP} \tag{9}$$

Accuracy is a metric widely used to compare machine and deep learning models because it evaluate generally how many samples of test data were labeled correctly. Accuracy can be calculated as the average number of TP and TN across all activities and falls divided by the total number of cases examined (Eq. 10).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

4 A Bi-LSTM Model for Human Activities Detection

We propose a Bi-LSTM model to identify human activities. The proposed empirical model (Fig. 3) is composed of three bidirectional layers containing 200 LSTM cells each (above this, the model started to obtain worse results), interspersed by dropout layers with a 25% probability rate in order to reduce the overfitting of the model. Hyperbolic tangent and sigmoid were set as functions of activation and recurrent activation of these cells, respectively. The output layer is a fully connected layer with 12 units (the data set contains 12 different activities to be classified; see Table 1) with softmax activation function. Figure 4 presents the code that implements our Bi-

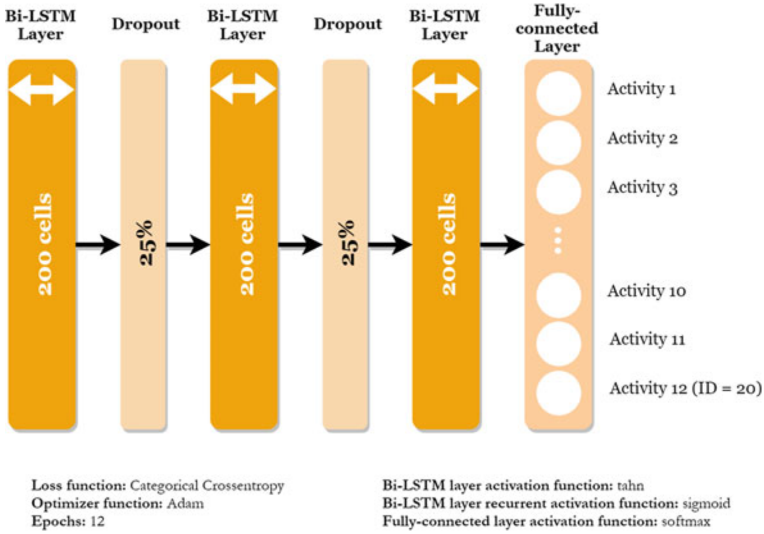


Fig. 3 Bidirectional LSTM network

```
model = Sequential()


activation_Dense='softmax'
units = 200

#LSTM MODEL

model.add(Bidirectional(CuDNNLSTM(units=units,
                                  return_sequences=True,
                                  input_shape=input_shape)))

model.add(Dropout(rate=0.25))

model.add(Bidirectional(CuDNNLSTM(units=units,
                                  return_sequences=True)))

model.add(Dropout(rate=0.25))

model.add(Bidirectional(CuDNNLSTM(units=units,
                                  return_sequences=True)))

model.add(Flatten())

model.add(Dense(12,activation=activation_Dense))


adam = optimizers.Adam(lr=0.001,
                       beta_1=0.9,
                       beta_2=0.999,
                       epsilon=None,
                       decay=0.0,
                       amsgrad=False)

model.compile(loss = 'categorical_crossentropy',
              optimizer = adam)
```

Fig. 4 Code that implements our Bi-LSTM model labelfig

LSTM model.[1] The model implementation was done using the Keras framework[2] with TensorFlow[3] as the backend.

As we are dealing with multi-label classification, we used categorical cross-entropy as a loss function [24]. Equation 11 illustrates the categorical cross-entropy function, where y is the array of real values, $\hat{y}$ is the array of predictions, N is the size of predictions, and M is the number of classes. The Adam algorithm as an optimizer [25].

$$L(y, \hat{y}) = -\sum_{j=0}^{M}\sum_{i=0}^{N}(y_{ij} * log(\hat{y}_{ij})) \tag{11}$$

The learning rate is equal to 0.001, β_1 and β_2 equal to 0.9 and 0.999, respectively. These parameters were defined empirically. For the training of this model, a pattern of at least 12 epochs was identified as the maximum reach of the network performance. Above 12 epochs, the performance tended to stabilize.

4.1 Data Pre-processing

In order to fit the data set to our model, we performed data pre-processing. Firstly, we sampled the data set on a per second basis (totaling 9,187 samples). The number of data points per second was not constant in our data set as each sensor may have collected data at different points in time. Thus, we used data padding in order to generate samples with a similar size. Thence, we considered the length of the greater sample (the second with more data points i.e. 22 points) and applied the padding; this was repeated until the sample comprised 22 points. In the end, the data set comprised 9187 complete samples.

Finally, we divided the data set in to two parts, allocating 80% (7,349 samples) for training and 20% (1,838 samples) for testing, an approach widely used in the literature [26, 27].

5 Results

5.1 Selecting the Best Bi-LSTM Model

We utilized a grid search technique to evaluate different Bi-LSTM architectures and then selected the best performing architecture for further evaluation.

[1]The entire code is available for download at https://github.com/GutoL/ChallengeUP.

[2]http://keras.io/.

[3]https://www.tensorflow.org/.

Table 2 Parameters and levels

Parameters	Levels
Number of layers	From 1 to 3, step 1
Number of nodes	From 100 to 250, step 25

Grid search identifies tuples from the combination of suggested values for two or more parameters, trains the model for each possible combination and compares the results of a predefined metric. Despite some limitations (see [28] for a more detailed discussion), grid search still represents the state of the art for hyper-parameter optimization and has been adopted in several machine learning studies (e.g., [29–32]).

As shown in Table 2, we defined different levels for different parameters of the model. The grid search was run 10 times and the average of all metrics was calculated in order to take into account the variation of results due to the stochastic nature of the optimization process [25].

Figures 5, 6, 7, 8, and 9 show the results for accuracy, precision, recall, specificity, and f1-score for all model configurations used in grid search, respectively.

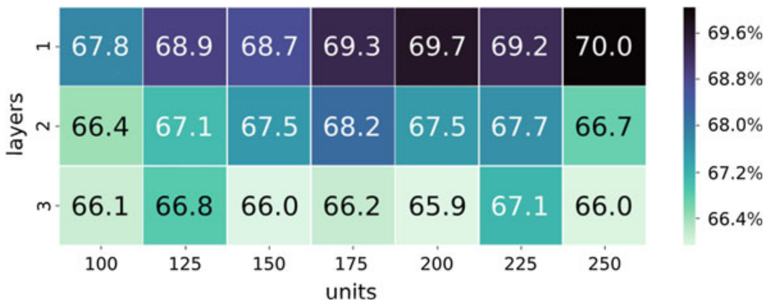


Fig. 5 Accuracy results for all model configurations used in the grid search approach

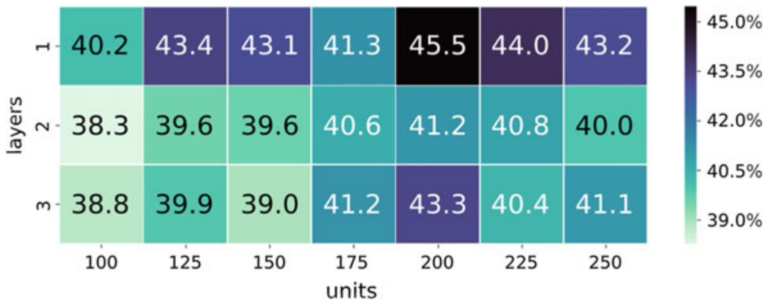


Fig. 6 Precision results for all model configurations used in the grid search approach

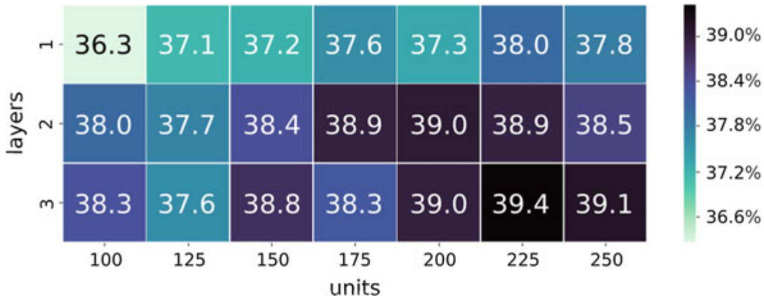


Fig. 7 Sensitivity results for all model configurations used in the grid search approach

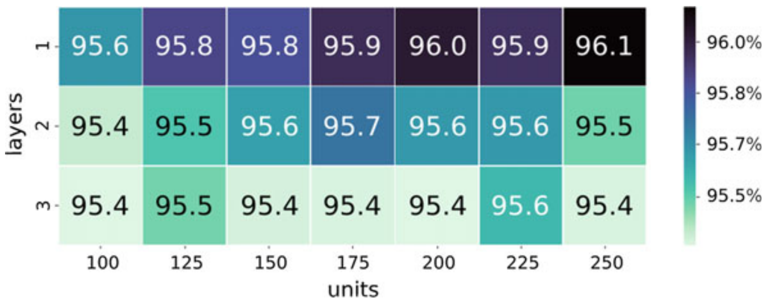


Fig. 8 Specificity results for all model configurations used in the grid search approach

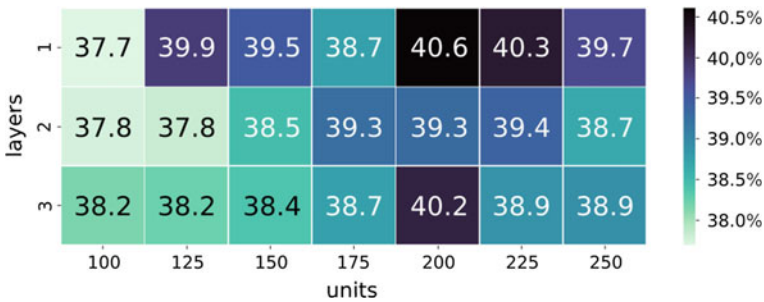


Fig. 9 F1-score results for all model configurations used in the grid search approach

Regarding **accuracy** (Fig. 5), the best model configuration uses 1 layer and 250 units, reaching, on average, 70%; while the model configuration with 3 layers and 200 units obtained the worst result, 65.9%, on average.

For **precision** (Fig. 6), the model configuration that gives, on average, the best result was 1 layer and 200 units (45.5%); while the worst average precision result (38.3%) was obtained by the model configuration with 2 layers and 100 units.

The **sensitivity** results (Fig. 7) suggest the configuration with 3 layers and 225 units presented the best recall result, achieving, on average, 39.4%; and the simplest model configuration, with 1 layer and 100 units presented the worst result, 36.3%, on average.

The **specificity** results, as shown in Fig. 8, suggest the model configuration that obtained the best result was composed of 1 layer and 250 units, achieving 96.10% specificity, on average. On the other hand, for this metric, different model configurations obtained the same worst level of specificity, on average, 95.40%. Such models were: 3 layers and 100, 150, 175, 200, and 250 units; and 2 layers and 100 units obtained.

Finally, considering the **f1-score**, as illustrated in Fig. 9, the model configuration that presented the best result was 1 layer and 200 units, with 40.60%, on average. The model with 1 layer and 225 units also achieved a good f1-score, 40.30%, and the model with 3 layers and 200 units found 40.20%. On the other hand, the model configuration that obtained the worst f1-score level was the simplest configuration, with 1 layer and 100 units, achieving 37.70%, on average; followed by the models with 2 layers and 100 and 125 units, both with 37.80%.

From the grid search results, one can note that there is no common behavior when analyzing the best performance per metric, meaning that for each metric, a different model can achieve the best result. The only exception was the model with 1 layer and 250 units, that found the best results for accuracy and specificity metrics.

The best model configuration in terms of accuracy (Fig. 5) uses 1 layer and 250 units, reaching, on average, 70%; while the model configuration with 3 layers and 200 units obtained the worst result with an average accuracy score of 65.9%. For precision (Fig. 6), the model configuration that gives the best performing model uses 1 layer and 200 units (average score 45.5%) while the worst model (average score 38.3%) uses 2 layers and 100 units. For sensitivity (Fig. 7), there seems to be a positive relationship between complexity and average sensitivity score with the simplest model configuration (1 layer and 100 units) showing the worst performance (average score 36.3%) with the second most complex model (3 layers and 225 units) providing the best results (average score 39.4%). In the case of specificity (Fig. 8), the model configuration that provides the best result uses 1 layer and 250 units (average score 96.10%) while a number of different configurations demonstrated poor performance (average score 95.40%). Finally, for the f1-score (Fig. 9), the model configuration using 1 layer and 200 units, the model using 1 layer and 225 units, and the model using 3 layers and 200 units, achieved similar results (average score 40.60, 40.30, and 40.20% respectively). On the other hand, the simplest model configuration (1 layer and 100 units) achieving the worst results (average score of 37.70%).

Interestingly, the results of the grid search suggest that there is no single model specification with consistently superior performance across different metrics. The model configurations achieving the best results according to one metric did not provide comparable results for any other metric. The only exception is the model with 1 layer and 250 units, that provides the best results for accuracy and specificity.

Another interesting observation relates to the number of layers in the model. In deep learning models, the concept of "depth" is related to the presence of several interconnected hidden layers with tens or hundreds of neurons. However, based on the results of our experiment, adding additional hidden layers does not always result in better model performance. For instance, the best result in terms of precision was obtained with 1 layer and 200 units (Fig. 6). Increasing the number of layers to 3, resulted in a 2.2% decrease in precision. A similar relationship appears across all other metrics with the only exception being sensitivity, where models with 3 layers tended to provide better results.

It is also worth highlighting that specificity is the metric with the highest average values, while sensitivity has the lowest. This suggests that our models are more able to predict true positives than true negatives.

Due to time constraints, we did not perform the grid search when initially designing the model submitted to the Challenge UP competition. Consequently, we present the results from two model configurations below, one identified empirically (Challenge UP results) and a second identified using grid search.

5.2 *Challenge up Results*

Figure 10 presents the confusion matrix regarding the test results using the Bi-LSTM model presented in Challenge UP. The model did not obtain good results in predicting falls (activities from 1 to 5). For example, all samples of activities 1 (falling forward using hands) and 3 (falling backwards) were misclassified by the model. This occurred because the data set has few samples of falls, and deep learning models perform better with data sets that contain larger sample sizes.

The Bi-LSTM model achieved the best results with classes that have more samples. Class 11 (lying down) achieved 638 correct predictions, followed by Class 7

Fig. 10 Confusion matrix of the challenge UP model

Table 3 Evaluation results (in %) for all activities using the Bi-LSTM model presented in Challenge UP competition

Activity	Precision	Sensitivity	Specificity	F1-score
1	0	0	98.6639	0
2	3.9216	11.7647	97.5089	5.8824
3	0	0	96.5795	0
4	18.7500	18.7500	99.3264	18.7500
5	5.2632	12.5000	97.2603	7.4074
6	91.7098	64.4809	97.9975	75.7219
7	70.5387	62.8186	89.5585	66.4552
8	58.6777	62.5000	86.2259	60.5286
9	0	0	99.6885	0
10	88.4848	52.3297	98.9403	65.7658
11	60.1887	72.5825	75.2347	65.8071
20	0	0	98.9691	0

(standing) with 419 hits. These classes obtained better results due to the simplicity of the activities captured by the sensors - the person remains still i.e. without making any movement. However, for those same classes (11 and 7), the model also misclassified a lot. For Class 11, the model misclassified 171 samples as Class 8 (sitting); and for Class 7, the model misclassified 223 samples as Class 11 (lying down).

Table 3 shows the evaluation results (in percentages) for precision, sensitivity, specificity, and f1-score for each activity presented in Table 1. Note that accuracy is not measured in this case because it is a global metric.

The model obtained good specificity results for all classes, achieving the best results for Class 9 (99.69%). Similarly, the model has a f1-score of 0% for Classes 1, 3, 9 and 20. However, for the other metrics, poor results were achieved. For example, for Classes 1, 3, 9, and 20, the value of the precision, sensitivity, and f1-score was 0%. In contrast, Classes 6, 7, 8, 10, and 11 achieved the best results for f1-score - 75.72%, 66.46%, 60.53%, 65.77%, and 65.81%, respectively. This is explained by the greater volume of samples for these classes in the data set.

One can see from Table 3 that the most critical metric for the Bi-LSTM model is sensitivity, which corresponds to the true positive rate. As the model misclassified several samples (Fig. 10), the overall sensitivity results are considered poor. Class 11 returned the highest sensitivity rate because it was the most correctly classified class in the data set. Table 4 presents the overall results for accuracy, precision, sensitivity, and f1-score for the model presented in Challenge UP. One can see that the model achieved 62.89% for accuracy, while other metrics achieved c. 32%. Since the data set used was unbalanced, the model classified the classes with more samples correctly (see Table 3).

Table 4 Overall metrics of the Bi-LSTM model presented in Challenge UP competition

Metrics	Results (%)
Overall accuracy	62.89
Mean global precision	33.13
Mean global sensitivity	32.52
Global f1-score	32.82

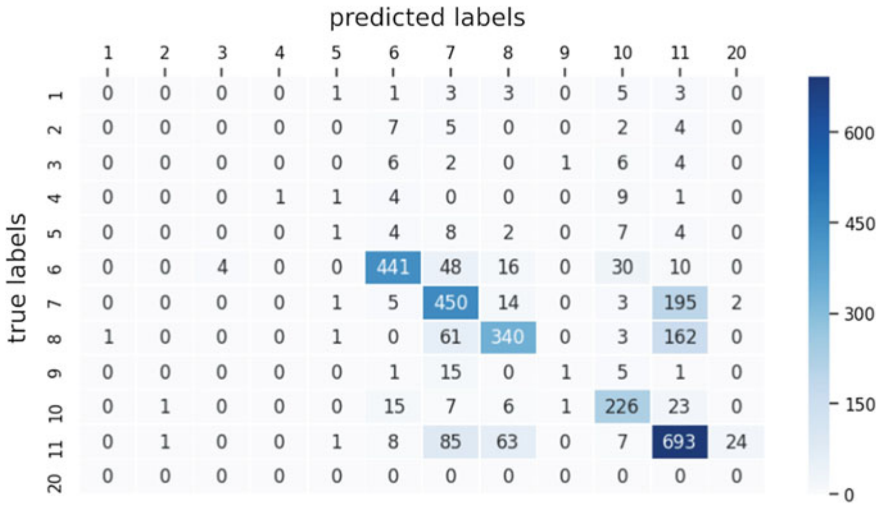


Fig. 11 Confusion matrix of the best model configuration found by the grid search

5.3 *Bi-LSTM Model Results (Grid Search)*

Figure 11 presents the confusion matrix for the best model configuration found by the grid search based on f1-score i.e. 1 layer and 200 units. The model did not obtain good results in predicting falls (activities from 1 to 5). In fact, activities 1 to 5 were largely misclassified. This is most likely related to the limited number of falls in the data set; deep learning models perform better with large samples. The Bi-LSTM model achieved the best results with classes that had more samples. For example, for Class 11 (lying down) and Class 7 (standing), the model generated 693 and 450 correct predictions respectively.

Table 5 presents the evaluation results (in %) for precision, sensitivity, specificity, and f1-score for each activity. Note that accuracy is not measured in this case because it is a global metric.

In, general, the model achieved very good specificity results for all classes (achieving 100% when considering Class 4). However, the model performed poorly for Classes 1, 2, 3, and 20 presenting a score of 0% for precision, sensitivity, and f1-score.

Table 5 Evaluation results (in %) for all activities using the best Bi-LSTM model found by the grid search

Activity	Precision	Sensitivity	Specificity	F1-score
1	0	0	99.9536	0
2	0	0	99.9072	0
3	0	0	99.8146	0
4	100	6.2500	100	11.7647
5	16.6667	3.8462	99.7682	6.2500
6	89.6341	80.3279	97.1072	84.7262
7	65.7895	67.1642	87.9195	66.4697
8	76.5766	59.8592	94.5749	67.1937
9	33.3333	4.3478	99.9071	7.6923
10	74.5875	81.0036	96.1577	77.6632
11	63	78.5714	78.2003	69.9294
20	0	0	98.8068	0

Table 6 Overall metrics for the best Bi-LSTM model configuration found by the grid search

Metrics	Results (%)
Overall accuracy	70.22
Mean overall precision	43.30
Mean overall sensitivity	34.67
Overall f1-score	38.50

These results illustrate some weaknesses of the proposed Bi-LSTM model configurations when working with an unbalanced data set. The classes that presented metrics equal to 0% were classes comprising relatively small samples.

Finally, Table 6 presents the overall results for accuracy, precision, sensitivity, and f1-score of the proposed model configuration. Similar to the results of the Challenge UP model, the model obtained the best result for accuracy (70.22%) when compared to the other metrics, reflecting the previously discussed uneven samples in the data set. The overall precision score was 43.30%; the sensitivity score was 34.67%, and the f1-score was 38.50%. One can note an improvement in all metrics from the initial model presented in Challenge UP to the revised model obtained using the grid search method.

6 Related Work

HAR can play an important role in people's daily lives due to its ability to learn important high-level knowledge about human activity from raw sensor inputs [33]. The increasing popularity of HAR is correlated with the diversity and popularity of wearable and on-body sensing devices such as accelerometers, gyroscopes, sound sensors, and image capture devices amongst others. HAR has drawn extensive attention in health and computer science research and is playing an increasingly important role in various research areas including home behaviour analysis [34], health monitoring [35], and gesture recognition [36].

There is a well established literature on HAR using machine learning. Historically, many studies focused on data with a single modality such as single sensor-based data [33, 37–39]. Single modality data is inherently limited for HAR studies in real-world settings due to high intra-class and low inter-class variations in the actions performed for a particular application [10]. Therefore, to exploit the benefits of machine learning techniques for a learning-based HAR, it is extremely important to have multi-modal data sets [33]. Multi-modal machine learning aims to build models that can process and relate information from multiple modalities [40].

More recently, there has been an increasing focus on the study of learning-based HAR using multi-modal data, and in particular, multi-modal time-series data. Existing methods can be divided into two categories: shallow learning-based HAR and deep learning-based HAR. The former relies on extracting a set of features from time-series sensor signals and mapping these handcrafted features to various human activities. Subsequently, a shallow supervised machine learning algorithm is applied to recognize activities. The most popular learning algorithms include decision trees [41, 42], K Nearest Neighbour (KNN) [43, 44], and Support Vector Machines (SVM) [45, 46]. For example, [46] extracts 561 features from an accelerometer and gyroscope, and applies a multi-class SVM to classify six different activities. The common characteristic of these methods is that they perform feature extraction manually which is task-dependent and requires human intervention, thereby impacting effectiveness. As a result, many researchers have turned their attention to deep learning approaches for automatic feature extraction. At the same time, implicit features can be learned by models that may not be possible using manual or handcrafted methods [47].

Many different deep learning models have been used to recognize human activities in a wide range of contexts including CNN, RNN, and particularly in the context of this chapter, LSTM networks. A very recent paper [48] proposed a baseball player behavior classification system using LSTM that accurately recognizes many baseball player behaviors. The classifier is trained on multi-modal data collected from multiple heterogeneous IoT sensors and cameras. [49] also used an LSTM network to detect daily human activities including eating and driving activity. The authors adopted a two-level ensemble model to combine class-probabilities of multiple sensor modalities, and demonstrated that a classifier-level sensor fusion technique for multi-modality can improve the classification performance compared to single modality data.

Authors in [50] used LSTM in a biometrics application to identify individual humans based on their motion patterns captured from smartphone features i.e. accelerometer, gyroscope and magnetometer data. The use of LSTM demonstrated that human movements convey necessary information about the person's identity and it is possible to achieve relatively good authentication results. The authors also demonstrated that the same LSTM algorithm can also be applied to other time-series data e.g. for gesture detection in a human conversation. In [51], inertial signals from a set of wearable sensors were used and fed as images into a CNN network to recognize human activities. Using both CNN and LSTM as a hybrid model, authors in [11] classified human activities. They used CNN to automatically extract spatial features from raw sensor signals, and LSTM to capture the temporal dynamics of the human movement.

Several surveys on recent advances on deep learning methods for multi-modal HAR have been completed and are worth reviewing for those interested in the domain [7–10].

While significant progress has been made, HAR remains a challenging task. This is partly due to the broad range of human activities as well as the rich variation in how a given activity can be performed. Deep learning shows great potential for a high-level abstraction of data. Therefore, more deep learning models need to be developed as self-configurable frameworks for HAR [47]. In this chapter, we propose a Bi-LSTM deep learning model to detect twelve types of human daily activities, and in particular, human falls. We use a multi-modal sensors data set generated from three different sources(i.e. wearable sensors, ambient sensors, and vision devices).

7 Conclusions

In this chapter, we propose a Bi-LSTM model to detect five different types of falls, six common daily human activities, and one post-fall activity. The data set was provided by the Challenge UP competition and was collected using a multi-modal approach generated from wearable sensors, ambient sensors, and vision devices.

Our Bi-LSTM model makes use of two wearable sensors (accelerometer and gyroscope) located at the ankle, right pocket, belt, and neck of the subject. In the training phase, the model was able to make good predictions of a selection of specific human activities (walking, standing, sitting, jumping and lying down). Our model is able to identify when a subject is lying down (when a fall has occurred) but it does not detect the type of fall (forward using hands, forward using knees, backwards, sideward, or falling after sitting on a chair). This result can be explained by the uneven samples of data by activity in the data set.

Future studies may explore other deep learning models, such as bidirectional gated recurrent units (Bi-GRU), a simplified version of LSTM layers, or CNNs, and compare the sensitivity, specificity, precision, and accuracy of a range of different models. Given the limitations of the data set used in this study and the impact on results, larger data sets with sufficiently large samples of each activity are required for wider use. Future research may involve creating actual or synthetic data sets to address these needs and leverage these and other multi-modal datasets (e.g. cameras and environmental sensors) for further study.

Acknowledgements This work is partly funded by the Irish Institute of Digital Business (dotLAB).

References

1. World Health Organization: Falls, Fact Sheet. http://www.who.int/news-room/fact-sheets/detail/falls (2018). Accessed 08 Oct 2008
2. Organization, W.H.: WHO Global Report on Falls: Prevention in Older Age. World Health Organization (2007)
3. Scheffer, A.C., Schuurmans, M.J., Van Dijk, N., Van Der Hooft, T., De Rooij, S.E.: Fear of falling: measurement strategy, prevalence, risk factors and consequences among older persons. Age Ageing **37**(1), 19 (2008)
4. Igual, R., Medrano, C., Plaza, I.: Challenges, issues and trends in fall detection systems. Biomed. Eng. Online **12**(1), 66 (2013)
5. Brownsell, S., Hawley, M.S.: Automatic fall detectors and the fear of falling. J. Telemed. Telecare **10**(5), 262 (2004)
6. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 689–696, (2011)
7. Rouast, P.V., Adam, M., Chiong, R.: Deep learning for human affect recognition: insights and new developments. IEEE Trans. Affect. Comput. (2019)
8. Baltrušaitis, T., Ahuja, C., Morency, L.P.: Multimodal machine learning: a survey and taxonomy. IEEE Trans. Pattern Anal. Mach. Intell. **41**(2), 423 (2019)
9. Nweke, H.F., Teh, Y.W., Mujtaba, G., Al-Garadi, M.A.: Data fusion and multiple classifier systems for human activity detection and health monitoring: review and open research directions. Inf. Fusion **46**, 147 (2019)
10. Chen, C., Jafari, R., Kehtarnavaz, N.: A survey of depth and inertial sensor fusion for human action recognition. Multimed. Tools Appl. **76**(3), 4405 (2017)
11. Ordóñez, F., Roggen, D.: Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors **16**(1), 115 (2016)
12. Bengio, Y., Simard, P., Frasconi, P., et al.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. **5**(2), 157 (1994)
13. Hochreiter, S.: The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int. J. Uncertain., Fuzziness Knowl.-Based Syst. **6**(02), 107 (1998)
14. Lin, H.Y., Hsueh, Y.L., Lie, W.N.: Abnormal event detection using microsoft kinect in a smart home. In: 2016 International Computer Symposium (ICS). IEEE, pp. 285–289 (2016)
15. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735 (1997)
16. Jozefowicz, R., Zaremba, W., Sutskever, I.: An empirical exploration of recurrent network architectures. In: International Conference on Machine Learning, pp. 2342–2350 (2015)
17. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J.: Lstm: a search space odyssey. IEEE Trans. Neural Netw. Learn. Syst. **28**(10), 2222 (2017)
18. Understanding lstm and its diagrams. https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714 (2016). Accessed Aug 2018
19. Zhao, R., Yan, R., Wang, J., Mao, K.: Learning to monitor machine health with convolutional bi-directional lstm networks. Sensors **17**(2), 273 (2017)
20. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: Deep learning, vol. 1. MIT press Cambridge (2016)
21. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673 (1997)
22. Challenge up website. Available at: https://sites.google.com/up.edu.mx/challenge-up-2019/overview?authuser=0 (2019). Last access Apr 2019
23. Martínez-Villaseñor, L., Ponce, H., Brieva, J., Moya-Albor, E., Núñez-Martínez, J., Peñafort-Asturiano, C.: Up-fall detection dataset: a multimodal approach. Sensors **19**(9), 1988 (2019)
24. Zhao, K., Chu, W.S., Zhang, H.: Deep region and multi-label learning for facial action unit detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3391–3399 (2016)

25. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014)
26. Berg, R.V.D., Kipf, T.N., Welling, M.: Graph convolutional matrix completion, arXiv preprint arXiv:1706.02263 (2017)
27. Mathis, A., Mamidanna, P., Cury, K.M., Abe, T., Murthy, V.N., Mathis, M.W., Bethge, M.: Deeplabcut: Markerless Pose Estimation of User-Defined Body Parts with Deep Learning. Nature Publishing Group, Tech. rep. (2018)
28. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13**(Feb), 281 (2012)
29. Li, J., Zhang, C., Li, Z.: Battlefield target identification based on improved grid-search svm classifier. In 2009 International Conference on Computational Intelligence and Software Engineering, pp. 1–4. IEEE (2009)
30. Hesterman, J.Y., Caucci, L., Kupinski, M.A., Barrett, H.H., Furenlid, L.R.: Maximum-likelihood estimation with a contracting-grid search algorithm. IEEE Trans. Nucl. Sci. **57**(3), 1077 (2010)
31. Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning, arXiv preprint arXiv:1611.01578 (2016)
32. Akay, M.F.: Support vector machines combined with feature selection for breast cancer diagnosis. Expert. Syst. Appl. **36**(2), 3240 (2009)
33. Wang, J., Chen, Y., Hao, S., Peng, X., Hu, L.: Deep learning for sensor-based activity recognition: a survey. Pattern Recognit. Lett. **119**, 3 (2019)
34. Vepakomma, P., De, D., Das, S.K., Bhansali, S.: A-wristocracy: deep learning on wrist-worn sensing for recognition of user complex activities. In: 2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN), pp. 1–6. IEEE (2015)
35. Ding, L., Fang, W., Luo, H., Love, P.E., Zhong, B., Ouyang, X.: A deep hybrid learning model to detect unsafe behavior: integrating convolution neural networks and long short-term memory. Autom. Constr. **86**, 118 (2018)
36. Núñez, J.C., Cabido, R., Pantrigo, J.J., Montemayor, A.S., Vélez, J.F.: Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. Pattern Recognit. **76**, 80 (2018)
37. Altun, K., Barshan, B.: Human activity recognition using inertial/magnetic sensor units. In: International Workshop on Human Behavior Understanding, pp. 38–51. Springer (2010)
38. Ermes, M., Pärkkä, J., Mäntyjärvi, J., Korhonen, I.: Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. IEEE Trans. Inf. Technol. Biomed. **12**(1), 20 (2008)
39. Lefebvre, G., Berlemont, S., Mamalet, F., Garcia, C.: Blstm-rnn based 3d gesture classification. In: International Conference on Artificial Neural Networks, pp. 381–388, Springer (2013
40. Jaimes, A., Sebe, N.: Multimodal human-computer interaction: a survey. Comput. Vis. Image Underst. **108**(1–2), 116 (2007)
41. Van Kasteren, T., Noulas, A., Englebienne, G., Kröse, B.: Accurate activity recognition in a home setting. In: Proceedings of the 10th International Conference on Ubiquitous Computing, pp. 1–9. ACM (2008)
42. Kwapisz, J.R., Weiss, G.M., Moore, S.A.: Activity recognition using cell phone accelerometers. ACM SigKDD Explor. Newsl. **12**(2), 74 (2011)
43. Hasan, S., Masnad, M., Mahmud, H., Hasan, M.: Human activity recognition using smartphone sensors with context filtering. In: Proceeding Ninth International Conference of Advances in Computer-Human Interactions, pp. 67–73 (2016)
44. Kunze, K., Lukowicz, P.: Dealing with sensor displacement in motion-based onbody activity recognition systems. In: Proceedings of the 10th International Conference on Ubiquitous Computing, pp. 20–29. ACM (2008)
45. Bulling, A., Roggen, D.: Recognition of visual memory recall processes using eye movement analysis. In: Proceedings of the 13th International Conference on Ubiquitous Computing, pp. 455–464. ACM (2011)

46. Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J.L.: A public domain dataset for human activity recognition using smartphones. In Esann (2013)
47. Kanjo, E., Younis, E.M., Ang, C.S.: Deep learning analysis of mobile physiological, environmental and location sensor data for emotion detection. Inf. Fusion **49**, 46 (2019)
48. Sun, S.W., Mou, T.C., Fang, C.C., Chang, P.C., Hua, K.L., Shih, H.C.: Baseball player behavior classification system using long short-term memory with multimodal features. Sensors **19**(6), 1425 (2019)
49. Chung, S., Lim, J., Noh, K.J., Kim, G., Jeong, H.: Sensor data acquisition and multimodal sensor fusion for human activity recognition using deep learning. Sensors **19**(7), 1716 (2019)
50. Neverova, N., Wolf, C., Lacey, G., Fridman, L., Chandra, D., Barbello, B., Taylor, G.: Learning human identity from motion patterns. IEEE Access **4**, 1810 (2016)
51. Yang, J., Nguyen, M.N., San, P.P., Li, X.L., Krishnaswamy, S.: Deep convolutional neural networks on multichannel time series for human activity recognition. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)

Intelligent Real-Time Multimodal Fall Detection in Fog Infrastructure Using Ensemble Learning

V. Divya and R. Leena Sri

Abstract As the data and computing power increases with surge in sensors and IoT devices to making things smart, the research trend is moving in the direction to help the elderly as a part of the smart home infrastructure. The elderly people of age over 60 years are at a high risk level due to adverse effects when fall occurs. There has been various researches in the field of automatic fall detection with the help of sensors, video surveillance, wearable devices etc. The detection and analysis has to be in near real-time to handle the fall efficiently. Decision making with the help of data from multiple sources tends to be more effective than decision from a single source. Hence, the use of multimodal fall detection has been in research which takes advantage of the data from multiple sources for optimal detection. Our proposed methodology helps in fall detection and relevant decision making in near real-time by reducing the processing latency. The low latency decision making is realized using the intermediate mist and fog layer for data filtering and immediate processing. The video data is analyzed at the edge and the fall is detected using edge processing. The use of the multimodal processing and detection increases the accuracy of prediction. Thus along with the video data, the data from the wearable sensors are also considered for detection. The detection is done using ensemble learning and compressed Deep neural networks for detection and image processing respectively. The blend of the fog infrastructure and the proposed algorithm helps improve the accuracy and reduce detection and decision making time.

Keywords Multimodal data · Mist and fog infrastructure · Ensemble learning · Compressed deep neural networks intelligent fall detection

V. Divya (✉) · R. Leena Sri
Department of Computer Science and Engineering,
Thiagarajar College of Engineering, Madurai, India
e-mail: vdivya.3793@gmail.com

R. Leena Sri
e-mail: rlsit@tce.edu

H. Ponce et al. (eds.), *Challenges and Trends in Multimodal Fall Detection for Healthcare*, Studies in Systems, Decision and Control 273,
https://doi.org/10.1007/978-3-030-38748-8_3

1 Introduction

Activity recognition has become the field of interest and research for many budding researchers in the field of virtual reality/ augmented reality and even in artificial intelligence. This field of research has led to the creation of pervasive intelligent environment which assists in creation of assistance and monitoring system in smart homes and other relevant environments [1]. This has led to the ease in the daily monitoring of human activities and assisting them in leading a healthy life. The activity monitoring system also helps in assessment of the daily activities such as walking, number of steps and calories spent, and also in monitoring the daily calorie intake and providing suggestions to healthy food intake. These monitoring systems helps in maintenance of the physical fitness of the individual and also in providing suggestions for further improvement. These researches in activity monitoring can also be used in case of adverse activities such as falling. Fall detection is an important aspect for elderly people health monitoring. Being the most abnormal and dangerous health hazard, detection and timely intervention of fall is an important part of health monitoring system [2, 3]. Fall may not always be intentional it may also be due to loss of balance during day-to-day activities [4]. The increase in the computational and the storage facilities have contributed a lot in this field of research. The improved vision and machine learning algorithms have been the greatest support in the development of the monitoring systems [5, 6]. In case of elderly people, they are unable to express their pain or agony due to their feeble disabilities. In most cases, the early detection of the fall could have helped or even avoid major health problems. Many times there are also cases where people are not treated for their fall due to insufficient and timely intervention. When left untreated or denied timely care, recurrent falls may occur hence leading to major threats. There has been reports from WHO [7] stating that the incidents of falls are very common and keeps increasing with increase in age. This also increases the fear of falling again and hence reducing the social interaction causing social isolation and in turn depression. Thus starting from a very small fall and the age factors included, it leads to major health problems.

This problem of timely fall detection has persisted for over years and the latest research and the development of the communication and the computational power has amplified the reach to the researchers and the provision for intervention. This has brought the physicians and the researchers closer and thereby making detection precise and reachable to the community. The major challenge lies in the collection of data which can be used in automated detection in smart environments. The data has to be extensive and precise so that it gives a highly accurate prediction [8]. The degree of fall, angle, body parameters and the time taken for recovery from fall all matters in proper decision making. These parameters may vary from one another and the normal factors may not be normal for others. It is not always possible to get the real-time data which is highly crucial for the automated detection. The data collected from laboratories in a controlled environment may be useful to a certain extent in creating a basic model for fall detection which includes the various sensor values from wearable sensors measuring the various parameters of the human experiencing a fall. These values tend to cause over-fitting and vary in various parameters when generalized to actual fall. Research from various medical agencies shows that when people are monitored for real-time fall data, the researchers often end up collecting a very large data of normal activities and a very small percentage of data indicating the actual parameters of a fall.

The long-term data collection has helped the researchers to collect data in the aspects of fall, but the skewed data pattern involving both the normal activities and the fall data stands a barrier in developing a general classifier for actual falls. The variety of data is in such a way that there may be a lot of unlabeled data and misleading data which makes the model building a challenging job. The major reason for this difficult model building may be due to the reliability on a single source of data collection from the available sensors [9]. The subjects put to test for the data collection do not always wear the sensors and sometimes it may happen in a way that during a fall the sensors may not be worn and the data collection is missed. Each data collected is unique in its own way and eases the model building. Various survey techniques on fall detection involves the analysis of data from various sensors and state of the art feature selection methodologies. Since the event of fall is a rare event, standard supervised learning methodologies may not prove to be efficient and the data from one person or situation may not suite for the other.

This chapter involves the survey on the various fall techniques available and our proposed methodology involving the introduction of intelligence to the detection framework. This intelligent framework learns from the data and does not rely on a source for generalization of the event. The detection and the reaction to the fall depends on the learning environment and the system progresses as the learning progresses. The decision is based on the analysis from various sensor values and also the most powerful vision sensor or the surveillance camera data. The numerical data combined with the image data creates a more intelligent database which improves the accuracy of detection.

2 Computing Paradigm Evolutions

Through the years the computing power and the platforms has evolved from parallel computing, distributed platforms, grid computing and finally cloud computing. The era of cloud has given the researchers the advantages of scalability, resource allocation on the go as the demand increases, flexibility, pay-as-you-go was introduced which caught the eye of the customers has led to the boom of the technology [10]. The service provisioning was realized using the initial three major models of Infrastructure as a service, Platform as a service and software as a service. The cloud services were provisioned at an affordable cost and only based on the consumption rate. These services were widely used by industries where the infrastructure and the platform cost can be put forth to the cloud provider and the industry takes care of the application development and the business intelligence of the customers. Though the cloud platform is still the widely used platform and many pioneers provide the finest service which has led to the era of “Service of Everything”, the technology has some limitations to be handled for better performance of the real-time applications [11].

As the products and research progresses towards Internet of Things (IoT) and more towards Internet of Everything (IoE), every connected devices gathers large amount of data each moment of time. Most of the time, the data collected is so enormous that the data is just stored remotely and is not analyzed for intelligence. With the use of cloud platform, these collected data has to be sent to a remote location for further processing and again the results are communicated back at the ground. This communication consumes larger bandwidth which can be utilized for effective computations. The world of automation requires latency free computations and near real-time decision making which is a drawback of the cloud platform. For example, the real-time applications such as the firefighting, health monitoring and immediate intervention are all latency sensitive applications which cannot afford the limitations of poor connectivity or processing and communication latency. This led to the raise of a new computing platform - Fog computing initially introduced by CISCO [12, 13]. The term of Fog is to bring the computation closer to the ground or closer to the end devices such that the bandwidth consumption and the communication latency reduces when compared to the cloud latency. The other major concern is the communication of the large data from the sensors or the end devices to the upper layer and handling a huge data. This is also handles by the edge nodes by basic filtering of the data and pushing only the needed and the important data for analytics.

The decisive property of the sensor data is that the data are time bound and when the time precedes, the data is of no use. For example the data from the temperature or the humidity sensors expire in later time and has to be analyzed and the actuation has to be sent within a time bound [14]. Thus the fog layer also handles this time sensitive data and provides temporary local storage which is discarded on analysis. Only the results of the analytics are moved on to the upper cloud layer for further intelligence on the application. Thus the fog layer acts an assist to the cloud layer in enhancing the performance of the application in terms of low latency low bandwidth consumption.

The roof computing is a very new paradigm under research. ROOF is used as a metaphor and an acronym (Real-Time Onsite Operations Facilitation) which lies between the cloud and the fog layer. Which is utmost one hop away from the end devices. This is sometimes called as Mist computing as well. In lieu with fog computing there has been researches in mist/roof computing that are in the budding stage of research. These terms vary in terms of the distance of computation from the end devices. The comparison of the computing paradigms is given in Table 1. These platforms help in the automatic detection of fall with least latency since the application is latency sensitive and has to be handled in near real-time. In order to realize the efficient fog architecture, the separation of the networking and computing is necessary to improve the performance of the platform. The use of the intelligent switch (SDN) is use to take care of the networking component of the fog infrastructure so that only the computing can be taken care by the application provider.

Table 1 Computing paradigm comparison

	Cloud	Fog	Edge	Mist
Architecture	Remote processing of the data to handle very large data over the internet and provide scalability on demand	Coined by CISCO extending computations to the edge of the network. Usually decentralized and the intelligence lies at the edge routers or gateways and integrated with the cloud	The edge processing is independent of the fog or the edge infrastructure. The inference is done on the edge device which reduces data transfer	A lightweight computing framework between the edge and the fog layer built using micro-controllers and microchips
Utilization	Ease of use and pay for what you use. No Infrastructure cost needed, all requirements are availed as service	Real time analytics with minimal latency. Privacy of data is ensured as the data is not sent to cloud unless needed	Extensive data filtering and helps save bandwidth. Low or no latency in decision making	Local decision making to support the edge and fog processing
Limitations	Latency, High bandwidth consumption. No privacy of data. No control over resource utilization and verification of the allotted resources	Initial infrastructure set-up cost though minimal	Less scalable and no resource pooling	

The advantages of the mist and the fog architectures are taken up by the proposed framework for detection

The SDN switch contains the intelligent controller to handle the networking and the data management in two separate planes with the help of the control and the data plane. The device level dependency of the network management is avoided to a greater extent with the help of SDN. This intelligent switch has been used to realize the Mist-Fog-Cloud architecture were in the fall detection data is analysed to provide low latency near real-time decision making. The architecture of SDN as proposed by Tomovic [15] is given in Fig. 1 and is further explained in Sect. 4.

This basic infrastructure of SDN as a part of fog architecture was incorporated to build a distributed fog infrastructure to help improve the speed and decision making in detection and decision making thereby delivering a low latency higher accurate application.

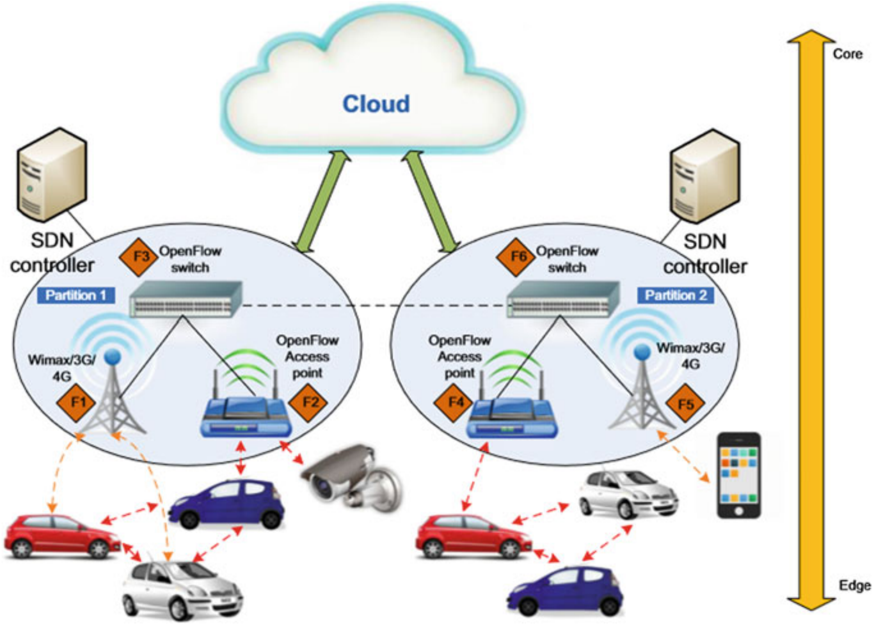


Fig. 1 SDN reference architecture

3 Related Works

The initial work on fall detection started with a very simple work [16] of finding the parameters of the fall during sleep, fall from sitting or standing from the data based on the wearable sensors. These parameters collected were a combination of data from the analysis of motion, posture, proximity and inactivity. Though fall is a rare event the author was able to collect a significant amount of data for analysis. There were also surveys by Perry et al. [17] which tries to analyze the real-time detection with the help od acceleration values. The survey concludes that the value of acceleration is the major parameter that is needed and can be the only parameter necessary for fall detection. The value of acceleration may tend to be deceiving when the sensor are not placed at the right position and the values may lead to a wrong prediction. Based on these surveys [18] presented a survey in an extremely different angle considering various kinds of sensors and approaches to detect fall. The survey viewed the changes in parameters based on the values from wearable sensors, vision based from the values of surveillance camera and also considered the influence of the ambiance of occurrence of fall. The data collected for the analysis were from every known source possible which includes accelerometer, posture, audio, video, spatio-temporal data. The survey concludes that though the wearable sensors can be cheap and large data can be collected at any point in time, the source is less reliable is highly influenced by the right placement which cannot be always ensured. The major reliable source reliable here s the video analytics that is highly robust.

In addition to surveys and discussion on data collection and reliability, there has been extensive study on effective use of the collected data. Machine learning being a reliable approach was used extensively starting from the feature extraction to detection [19]. The most common algorithms deployed for detection were the decision trees, KNN, Naive Bayes and SVM classifiers. The performance of theses classifiers varies based on the parameters chosen for classification. The major influencing parameters were the number of people in the scene in the video taken up for analytics, age, obtrusiveness and occlusion values. As the trend of analytics increased the parameter of energy consumption were also considered as a major parameter. The major detection and discussions were influenced by the synthetic data which amounts to almost 94% of the total collected data [20]. Thus the placement of the data and the data capture at the right point of time has a major influence on the results. An improved detection methodology was provided by Pannurat et al. [21] which involves the deep parameter consideration of the person which includes the height, weight, sensor type placed for detection. These parameters were helpful since the fall of each person is entirely different from the other.

The increase in the use of mobile phone has further influenced the data collection and fall detection based on pattern matching or threshold based techniques [22]. Even though data collection can be made easier with the help of smart phones and the sensors along with it, the threshold fixing is a tedious job and hence leads to missed alarms or fake alarms. The continuous monitoring system also leads to energy depletion and the monitoring cannot be done at a specified time also. These factors have to be considered and have to be seriously analyzed in order to present an efficient intelligent automatic fall detection system. Thus the following factors have to be considered to develop an efficient detection system.

- Develop a standard methodology of fall detection considering multimodal data in all possible perspectives for an efficient intelligent system.
- The threshold based analysis have to reduced as much as possible in decision making and made computationally effective since the threshold values may vary from person to person and makes it a tedious job in generalization.
- The dataset for training has to be considered from a publicly available benchmark dataset consisting of both sensor and video analytics for better accuracy.
- The detection and analysis has to done in two major perspectives as, data available for analysis and the analysis in case of least data available.

Considering these parameters, the data has to be analysed such that, in case of fewer data availability, sampling of data has to be done and the use of semi supervised learning can be used to make a complete use of the available data. The algorithms has to be cost efficient in case of both computation and power efficiency.

3.1 Handling Sensor Data

The era of cloud computing is on the bloom that any service needed be it computation, storage, decision making platform or even infrastructure is given at an affordable price available at any time. The era has become a notion of anything as a service. The cloud server is located at a remote place which provides service as needed by the client. Though the cloud platform is widely used by all almost all known applications, the limitations that arise due to the increase in the number of IoT devices and sudden urge of data from these devices have to be handled efficiently. The critical limitation of the platform is the connectivity between the remote cloud and the end devices. The data from the end devices such as the sensor data have to be analyzed immediately since these data validity is time bound. IoT data keeps changing and so after a certain time period when a new data arrives the older data is of no use to analyze. Thus the collected data has to be transferred and analysed as soon as possible. The issue of connectivity causes latency in the transfer of the data from the end devices to the remote cloud for analysis. In most of the applications such as fire alarms, autonomous or connected vehicles, the cloud server is distributed and the application runs as multiple instances in multiple cloudlets over a distributed area. This again causes more overhead due to the inter-cloud communication along with the transfer delay sometimes making the application expire before the completion of the analysis and decision making [14, 23].

For effective handling of these real-time data at a reduced latency, CISCO in collaboration with OpenFog consortium has come up with a new paradigm Fog Computing which offloads the time constrained decision making and computation to the edge thereby resolving the issue of expiration of the data importance. Described as "cloud closer to ground" by the consortium, the architecture also takes care of the fact that overloading of computation at the edge doesn't cause latency in computations. The task offloading to the edge has to be done keeping in mind the energy efficiency, computational latency and the capacity of the end devices to handle the computations. The computationally intensive tasks and the delay tolerant part of the applications happen at the cloud and the latency sensitive computations at the edge. Thus this new paradigm of edge/fog computing is not a replacement of the cloud platform but acts as an assisting layer or platform to enhance the performance of the application hosted. The sending and receiving of data to and from the cloud or fog layer consumes considerable amount of bandwidth. The major task resides in the reduction of the amount of data transfer to the upper layer and thereby reducing the bandwidth consumption. The research has moved on further closer to the end devices introducing the concept of mist computing or the mist layer where in the basic filtering of the data and the least computationally intensive task such as the rule based processing is done at the mist layer. The layer takes care of data aggregation, fusion, filtering and then sending the relevant data to the fog thereby reducing the bandwidth consumption and proper data processing with minimal error. The fog layer takes care of the data processing on the fly, anomaly detection, raising alarms and alerts in real-time. The fog layer is also responsible for the local temporary data storage, compression and further filtering hence reducing further load and amount

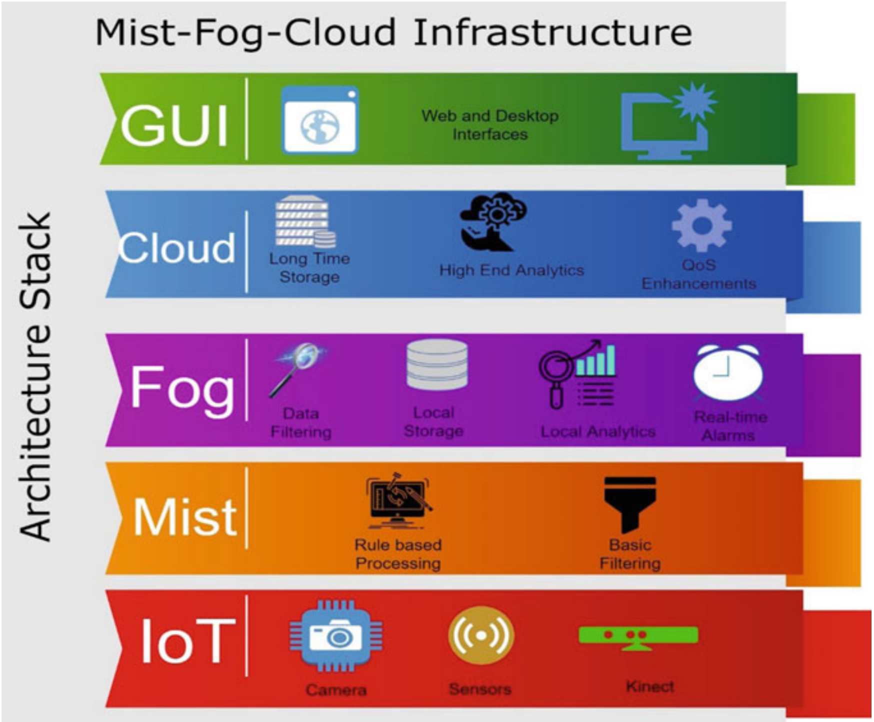


Fig. 2 Proposed architectural stack

of data transferred to the cloud. The combination of the mist-fog-cloud architecture helps in the improvement of the overall system performance, QoS of the application. Finally the cloud layer is responsible to handle the large amount of data and helps in the permanent data storage used for further intelligence of the application. The cloud also handles the highly computationally intensive computations and the large neural networks or the intelligent algorithms for decision making and automation. Proper entrustment of the load to the fog and the mist layers helps in the improvement of the overall performance of the system. Figure 2 gives the proposed architectural stack in automatic fall detection.

The lower most layer consists of the IoT devices which includes the smart camera, the wearable sensors such as the gyroscope, Accelerometer, Kinect etc to collect the data to detect the fall. This layer generates large amount of data monitored periodically. Sending the whole monitored data to the cloud leads to high energy and bandwidth consumption. Thus the processing of the data is spread among the mist, fog and the cloud layer which leads to the increase in the overall performance of the application.

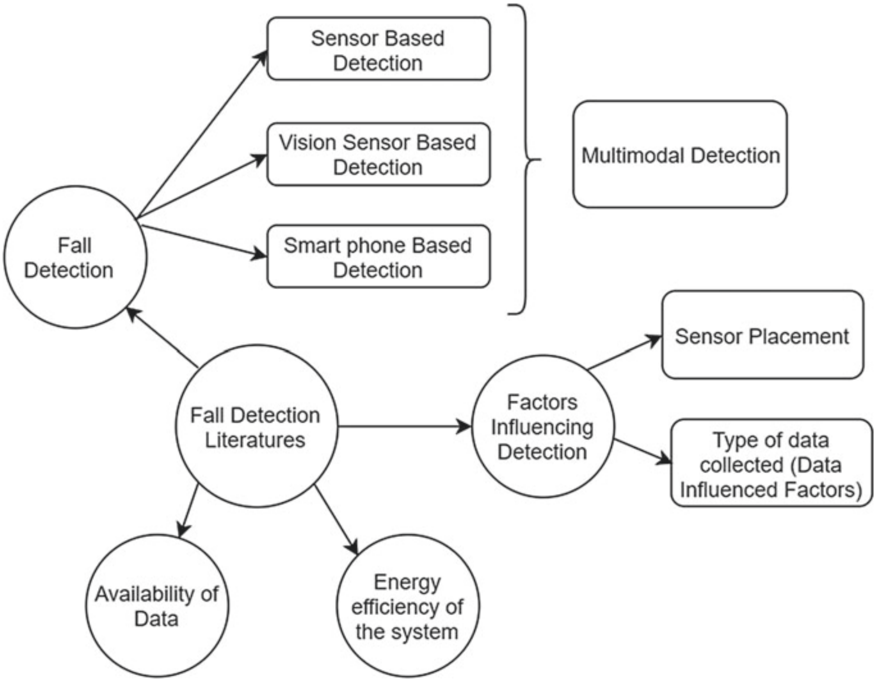


Fig. 3 Literature taxonomy

There has been review by Ren and Peng [24] which shows the factors and several literature that shows the advancements in fall detection. The overall view is given in Fig. 3.

There had been several deployment of algorithms for fall detection that ranges from threshold based detection to the use of machine learning algorithms. Table 2 shows the works on the deployed algorithms for fall detection.

A latest work done by Shahzad et al. [34] takes advantage of the accelerometer signals and tracking them using a smart-phone such that, real-time identification of fall can be differentiated from events such as lying in bed. This work also is one of the first to work on reducing false alarms on fall detection. The algorithm used here was support vector machine with multiple kernels that helped achieve high accuracy of as much as between 97.8 and 91.7%. The survey by Asadi-Aghbolaghi et al. [35] gives the various state of the art methodologies in the field of gesture and action recognition based on the image sequences based in deep learning approaches. Similarly, the latest survey by Zhang et al. [36] gives the advantages of deep learning in the 5G era for real-time data analytics and agile management of the network resources.

Any system has to be validated on a dataset and before choosing a dataset, a review on several datasets were taken as followed. There has been works on building a database from the data collected from wearable sensors. As the works reported by Vavoulas et al. [37], Sucerquia et al. [38], the database is all about sensors and is not

Table 2 Literature on algorithms

Work done	Place of experiment	Algorithm used	Accuracy/recall (%)
Bilgin et al. [25]	Laboratory	KNN	100/89.4
Yu et al. [26], Bashir et al. [27], Dai et al. [28]	Laboratory	Threshold based	81/84–86
Werner et al. [29]	Laboratory	Threshold based and SVM	90/NA
Sengto et al. [30]	Laboratory	Neural network	96.2/NA
Yu [31]	Laboratory	Directed acyclic support vector machine	NA/97.08
Ojetola et al. [32]	Laboratory	Decision tree	92/98–99
Shoaib et al. [33]	Realtime	Color matching and eclipse Matching	NA/96

multimodal. These are data collected from experiments mostly on elderly people. In addition to these sensor datasets, there are also databases that include data from vision sensors collected mainly from web cameras or RGB cameras. The work by Xu et al. [39] shows the use of the kinect imaging in human activity tracing and there has been datasets collected using this technology as included in the works [40, 41].

Among the various available datasets some being open source and some collected by the authors, the dataset used here for our work is one that is publicly available and is multimodal. The Dataset used in the work for fall detection is the UP-Fall Detection dataset [42] consisting of 11 activities which includes fall and regular activities collected with the help of wearable, ambient and vision sensors. The activities considered for detection includes, Falling forward using hands(A1), Falling forward using knees (A2), Falling backwards (A3), Falling sidewards (A4), Falling sitting in empty chair (A5), walking (A6), standing (A7), sitting (A8), picking up an object (A9), jumping (A10) and laying (A11). The data is collected by experiments from 17 subjects ranging from 18–24 years old and a mean weight of 66.8 kg collected over a period of 4 weeks. These sensor data is combined with the data from real-time surveillance which is detected using edge processing.

4 Environmental Setup

The traditional networking works in such a way that the signaling traffic, route population, configuration and network management, packet delivery and QoS are all taken care by the networking device deployed. With increase in the intelligent features of networking and the introduction of the intelligent networking with the help of Software Defined Networking, the packet delivery and the networking components are handled separately by the data plane and the control plane respectively. This also solves the problem of single point failure as the controllers are distributed in nature and the workload distribution and the distributed nature of deployment helps in maintaining the availability of the application. The deployment of the controllers helps in the visualization of the global view of the entire network. The distributed controllers deployed helps in the maintenance of a large independent network with a precise global view. Proper load balancing algorithms when employed can help realize enormous performance of the network and the application hosted. The communication between the controller and the data plane is ensured with the help of the underlying OpenFlow protocol. The salient features of the protocol helps statistical data collection of the network, intelligent traffic monitoring, and proactive decision making regarding the path of the packet flow. Though the intelligent switch or the OpenFlow architecture was primarily designed to set up a prominent Wide Area Network, it can also be extended to implement our own Edge architecture.

The Mist-Fog-Cloud (MFC) Architecture proposed here for our work uses the advantage of the intelligent switch to create a fog architecture to reduce the latency of the application processing. The switch takes care of the networking component so that the application processing and computation can be given more importance. The SDN controller acts as the fog controller which decides the data transfer to the upper cloud layer. The decision making and the alarm modules are deployed on the fog layer which gives immediate response to the user since it is closer to the user than the remote cloud. The latency is very much reduced when compared to the detection using a remote cloud platform. The communication delay is also reduced with the help of the intelligent switch were-in the user or the application provider need not program the routing or the switching mechanism from scratch.

Thus the created infrastructure helps in creating a large personalized network with a global view of the whole network. The number of smart cameras can be increased as per the area of coverage and the whole area is surveyed using the MFC infrastructure. The results of the smart detection is validated with the help of iFogSim simulator where in the infrastructure is simulated consisting of the smart cameras, mist and the fog nodes. The bandwidth consumption and the latency in the overall detection process is discussed in Fig. 4.

The initial unit proposed is the smart camera that is capable of both detection and communication of data. The camera is said to be smart since it has a neural network running for detection. The next block gives the trained neural network that has been compressed as described in 5. This trained model gives the ability of inference at the edge layer which is the smart camera. This forms the initial analysis and the next layer of analysis is done at the mist layer that helps in the initial filtering. The rule based filtering is a basic filtering that is threshold based filtering mechanism of the data. Here, the data at normal levels that do not contribute to fall is not sent to the upper layer for analytics. Only the data that contribute to fall is sent to inference of fall. This initial filtering helps in the reduction of bandwidth utility. Finally, the ensemble learning algorithm training at the cloud helps in conforming the fall with inference taken from both the vision and the sensor data. Since the decision making is

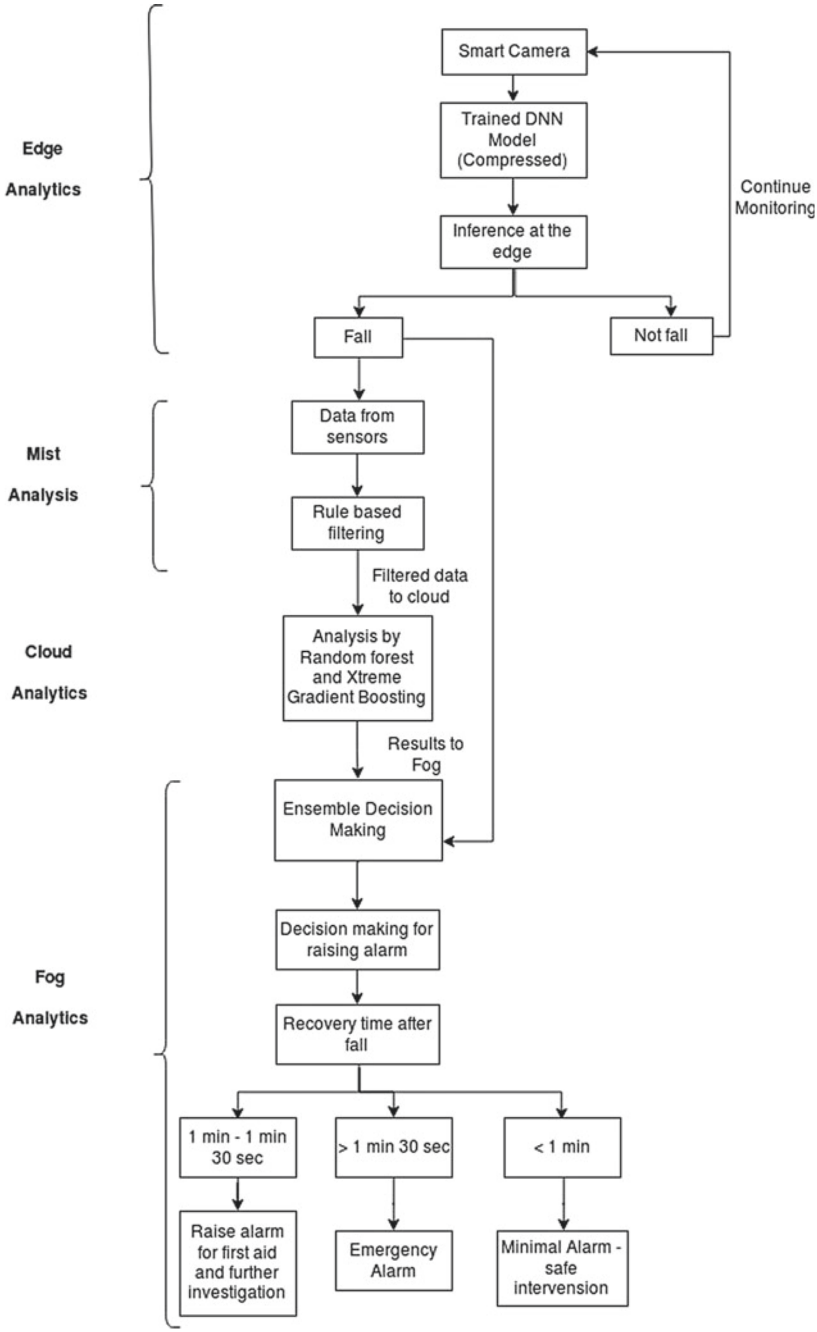


Fig. 4 Overall multimodal workflow

based on multimodal data, the accuracy of detection is better than a single algorithm or single data based detection. Finally, personalized alarm setting is provided that can be regulated based on the person under monitor.

5 Proposed Methodology

The automatic fall detection is layered among the three major layer of the Mist, fog and the cloud each handling the data in different volumes and intensities. The mist layer which is very near to the end device utmost one hop away or on the edge device is the mist computation. This layer is responsible for the basic data filtering and the rule based decision making. The compressed deep neural network model deployed on the smart camera acts as a mist device providing intelligence at the edge. A trained Deep Convolutional Neural Network (D-CNN) model, trained with the help of the surveillance camera images that are compressed to run at a low end device is deployed at the smart camera providing intelligence of detection at the edge. The Deep neural network compression technique used is described in Sect. 5.2.

The paper by the authors [43] have explained the use of neural networks and its advantages in the field of Artificial intelligence. Though the algorithm is of high computational complexity, the efficient use of the architecture helps in efficient energy aware computing with greater accuracy. With the basis of this claim, the DNN has been chosen for edge detection in the proposed work.

The compressed DNN maintains the accuracy of the model and also is trained much faster using transfer learning. The model training is done at the cloud and the end device running the model does only the inference thus the model is able to provide the anticipated performance and the accuracy to the detection. Since only the vision output cannot be considered as the fall may vary from person to person, the individual parameters of the person must also be considered, the data from the wearable sensors are also considered for detection and raising alarms. The deployment of the multimodal methodology increase the accuracy and the reliability of the system. Thus the edge detection output is sent to the upper fog layer only if the detection is a fall. When the detection is not a fall, the data is sent to the upper layer thereby conserving the bandwidth usage. Further, not all images captured by the smart camera is sent to the upper layer or to the cloud. This saves almost 5x power consumption [44] and considerable amount of bandwidth usage. Only when a fall is detected, the images are sent to the cloud storage creating a dataset for further intelligence. Since fall is a rare event this collection of the data to the cloud helps in building a good collection of training data for further intelligence of the application.

The other advantage of using smart edge detection is the minimized storage space. Since the detection is enabled at the edge layer, there is no need to store the entire surveillance images. The normal images are not stored and only the images that are detected to be outliers from both vision sensors and based on sensor data are taken up for storage and further analysis.

Now that the fall has been detected at the edge, in order to avoid false alarms and utilize the advantage of the multimodal detection system, the data from the sensors are also considered for decision making. The data from the sensors are collected in the mist layer where in the data are filtered using a rule based processing. The values above a certain threshold are taken up to the above layer for analysis. This rule based filtering of the data also helps in the reduction in the amount of data being sent to the above layer. This filtering also helps in the basic data pre-processing from the sensors. When not properly pre-processed, it leads to a huge amount of data transfer to the above layer and in turn leading to latency and abnormal bandwidth consumption. Thus the mist layer helps in the basic rule based processing of the data and minimal filtering operations which contributes to the overall performance enhancement of the application.

The data filtered from the mist layer is now pushed to the fog layer for further processing and intelligence. The fog layer is responsible for temporary local storage of the data and minimal analysis to provide intelligence to the data analytics. The filtered data from the lower layer are sent to the cloud and considered for analysis using random forest and xtreme Gradient boosting. The results of these algorithms saying if the values collected is a fall or not is sent to the fog layer. This result is then aggregated with the results from the video analytics and with the help of ensemble method of voting an alarm is raised. The raising of alarm and the minimal decision making is done at the fog layer so that it is not computationally intensive and is closer to the ground.

The overall work-flow of the automatic detection and alarm raising is given in Fig. 4. The combined advantage of the MistFogCloud infrastructure helps in a better detection environment with the least possible latency and near realtime decision making.

5.1 Detection at the Edge

In the era of cloud computing, there has been intensive research with the rise in IoT devices to handle the large streaming data from the devices. The following are the most recent protuberant researches in fog computing and DNN compression.

Bonomi et al. [13] have proposed the need for the new paradigm for handling the IoT data with the help of fog layer. They have outlined the applications that can be enhanced with the new paradigm of fog computing. The research challenges faced during their survey are resource management, fault-tolerant service provisioning. Since the nodes that are used as fog nodes have low computing and networking capabilities, Bilal et al. [45] have surveyed the IoT applications in focused on energy consumption, latency and efficient integration with the cloud environment. The challenge faced was on energy consumption reduction.

In the toll of setting up the fog environment, various devices like a switch, router, mobile phones, raspberry pi etc. were used as fog devices or nodes. Dastjerdi et al. [46] have used Cisco IOx device management for the services in Fog environment.

The paper also explores the various applications that can be enhanced with the help of fog computing. To help integrate with the real world applications, various hierarchical architectures for fog computing with integration with cloud were proposed. Cao et al. [47] have used the novel fog architecture for healthcare application in fall detection in stroke patients. They have improved the efficiency of the real-world system with the introduction of the new intermediate layer. The challenge faced was the ability to handle faulty devices and providing availability in case of device failure in the fog layer. Aazam et al. [48] have designed a smart gateway architecture which decides the need for application offloading based on the residual energy of the devices running the application. The use of machine learning algorithm has helped in the decision making of choosing the low energy level node for offloading.

With the wide use of the Fog architecture in almost all applications like the smart cities, hospitals, smart home etc., there was a rising issue of data privacy and security. Lu et al. [49] have proposed the use of the Chinese remainder theorem to produce and use the hash functions to secure the data being processed from the streaming IoT devices and sensors. Thus Fog computing layer can be used in applications wherein there is a need for privacy preservation and the data has to be processed locally than sending them to the cloud for data analytics. Now that the applications are low latency providing real-time decision making, there has been high availability in case of critical applications. Wang et al. [50] have proposed a self-adaptive module which performs Directed Diffusion and Limited Flooding to enhance the reliability of the data being transmitted. This does not consider the failure of the controller or the fog device and there is no option for load balancing of the network. This framework has been implemented in the healthcare sector in the transmission of the ICU data. With the high evolution of decision making and analytics in fog architecture, there were research on deep learning networks being able to run on mobile devices and smaller end devices providing analytics at the edge. Cheng et al. [51] have surveyed the various model compression for deep neural networks. The paper deals with the various methods like the knowledge distillation and Low-rank factorization that are application specific and the sparcity of the network cannot be controlled by the developer.

Building and training a deep neural network from the scratch and training a huge amount of data or images on the fly is highly time-consuming. Yosinski et al. [52] have listed the importance of the transfer learning and the retraining of the model. The paper states transfer learning for training a DNN at a reduced time. It helps in time reduction by providing a reduced search space during training. Figure 5 represents the time reduction by the reduced search space used in transfer learning.

5.2 Smart Edge Device with Compressed DNN

The initial subtask is running a part of the task in the end device. To do so, the end device has to be made smart to take the necessary decisions at the edge. The end devices are equipped with DNN for decision making. The decision making

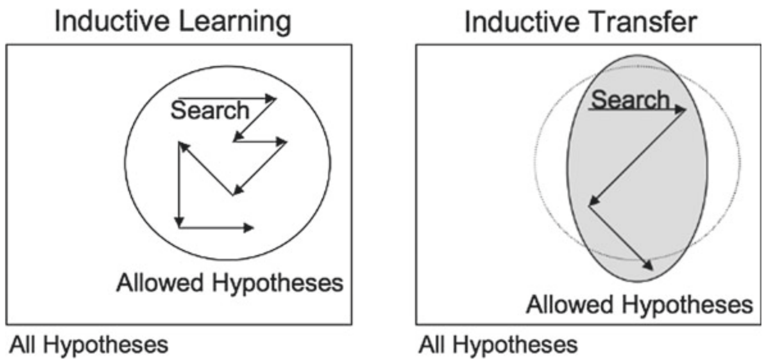


Fig. 5 Limited search space in transfer learning

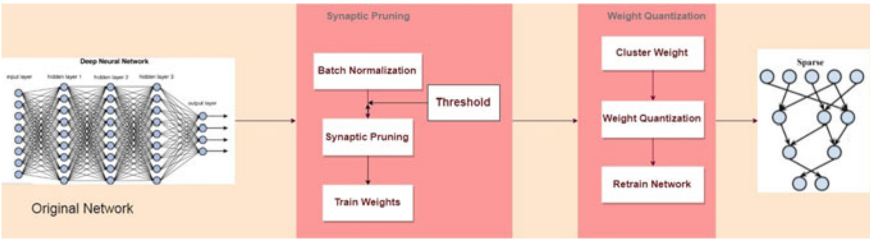


Fig. 6 Deep neural network compression

should be competent for the fog devices that have low memory and low computing capabilities. DNN is highly dense and occupy large space to run and needs high computing capability to run these heavy models. In order to run the DNN in the low-end edge devices, the models have to be compressed and made capable enough to run in these low capability devices like the surveillance camera, Raspberry Pi, low-end computers etc. Figure 6. Shows the overall process of DNN compression.

The initial step of compression is the synaptic pruning. The synaptic values show the amount of information the connection contributes to the final result. This information is extracted by normalizing the weight and the input data. Consider the model with C channels with n filters used in the convolutional layer thus generating n output features with the activation function a and the output features are given by Eq. 1

$$\sum_{c=1}^{C} x_c^{ip} \times \eta_{n,c} + b \tag{1}$$

where b is the bias added to the layer, $x_c^i p$ is the input feature in the cth layer and $\eta_{(n,c)}$ represents the kernal in the nth filter. As proposed by Sergey et al. [53] Batch Normalization is used here for weight normalization which gives faster convergence and generalization. The normalization is performed in mini batches as given in Eq. 2 and 3,

$$Normalization = \frac{x_{ip} - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{2}$$

where μ is the mean of the values, σ is the standard deviation of the values in the input layer x_{ip} and ϵ is the factor to denote the mini-batch normalization

$$x_{out} = \gamma_c \times Normalization\,(x) + \beta \tag{3}$$

where γ is the scaling factor and β is the shifting factor.

The Synaptic strength is defined by Eq. 4

$$S_{n,c} = \gamma_c \times r_{n,c} \tag{4}$$

where r is the norm of kernel n. The network is pruned by removing the synapses below a threshold t. The threshold is chosen by the desired sparcity. The value of p% synaptic strength in the network will be set as the threshold for pruning. In our work, the percentage of pruning is set to 85–90%.

The Weight quantization of the pruned network is done by clustering the similar weights and grouping them together and represent them as a sparse matrix using their average. Figure 7 shows the sample reduction of the weight and representation of the weights as a sparse matrix.

The neural network is now put in place to run on the edge smart camera. The decision making of fall or not is captured by the smart camera and only when it is detected as a fall, the data is sent to the upper layer thereby reducing the content transfer and bandwidth utilization.

Since the edge devices are low end devices, running a very large model is not possible which may even crash the system. The deep neural network is trained with the input images of fall and not fall images, the model is compressed and then deployed onto the edge device, here in this case is a smart camera. There is no training or testing at the edge where as only the inference happens at the edge via the deplyed compressed model. The compression leads to a significant decrease in the

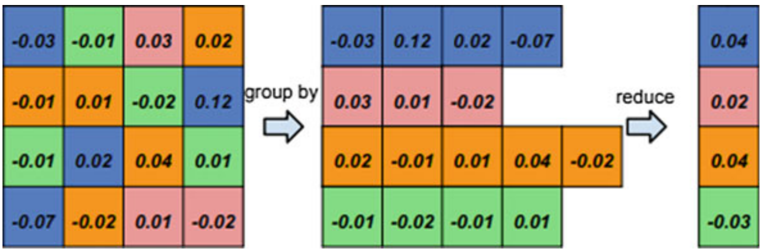


Fig. 7 Sparse matrix representation

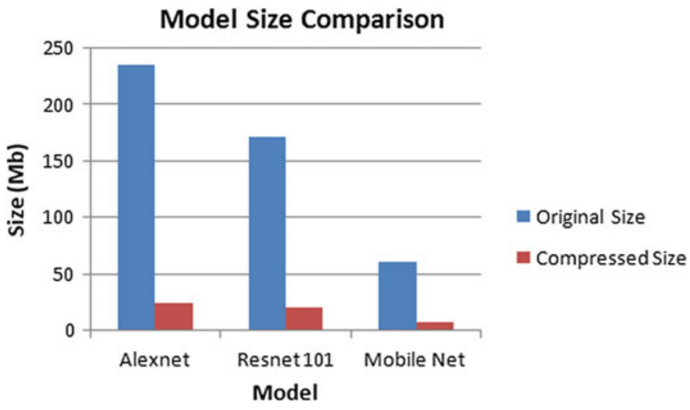


Fig. 8 Model size comparison

size of the model thus making it possible to run at the low end edge devices. Figure 8 shows the reduction in size of the models that were tested to be deployed at the edge devices.

The model after compression of the layers using synaptic pruning and sparse matrix representation considering the parameters such as Accuracy, Storage, Latency, Energy Consumption are employed in mobile devices and Raspberry Pi for testing. The models are of reduced size and can be used to run in the edge devices.

Table 3 shows the devices and the configuration of the devices wherein the compressed model was successfully employed for edge processing. The created compressed model was also trained on extra images that were not in the existing model too. This gives way for customization of the model for any needed application and not rely on only the existing set of trained images. The models were tested on mobile devices with minimum RAM and memory and on a raspberry pi with the bare minimum memory.

Table 3 Model deployment at the edge

Device	Name of the device	RAM	ROM	Power	Models trained
Mobile phone	Redmi 5A	2 GB	16 GB	3000 mAh	MobileNet, ResNet 101
	Redmi 6A	2 GB	32 GB	3000 mAh	MobileNet, AlexNet
	Lenovo A6600 Plus	2 GB	16 GB	2300 mAh	ResNet 101
Raspberry Pi	Pi Zero	512 MB	–	–	ResNet 101
	Raspberry Pi 3	1 GB MB	–	–	ResNet 101

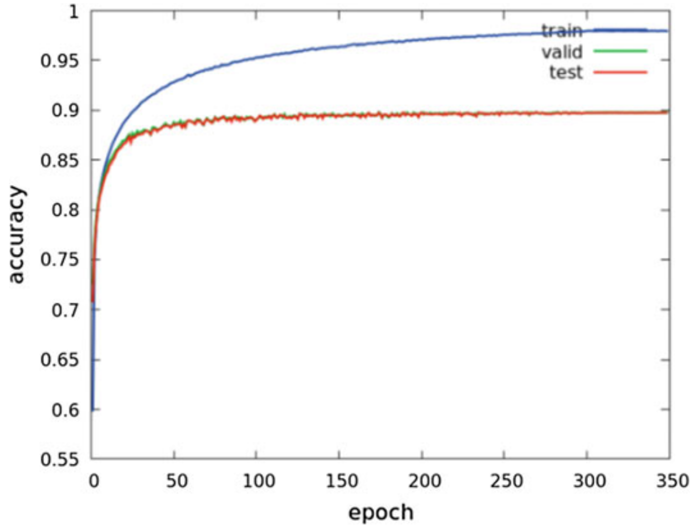
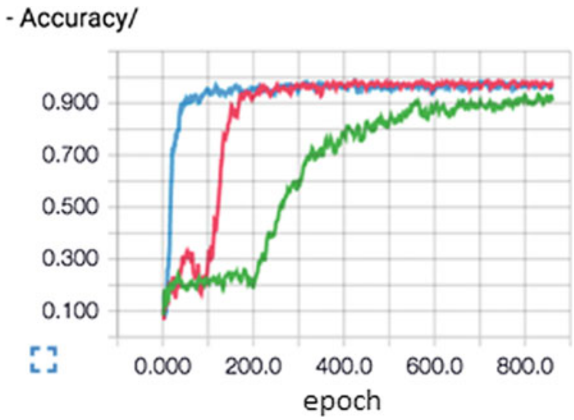


Fig. 9 Mobile net accuracy

Fig. 10 AlexNet accuracy



The models were trained and tested in the tensorflow platforms and the tensor board gives the training, validation and the testing statistics of the model. Figures 9 and 10 shows the sample model accuracy taken from Tensor Board for the MobileNet and the AlexNet model respectively. The graph is generated for Training, Validation and testing accuracy. The types of a graph generated is shown below for different epochs and models. The accuracy of the models do not deteriorate to a large scale even when the size of the DNN is reduced to a large extent.

5.3 Ensemble Detection

Once the data from the smart camera is classified to be a fall, the data from the sensors are also collected and pushed to the mist layer which is one hop away from the end devices. The mist layer device helps on the basic rule based filtering of the data where the data above the safe threshold level is sent to the cloud for further high level machine learning processing. The basic filtering in the mist layer helps in the reduction of the bandwidth usage and helps data communication to upper layer only when necessary. The high level and computationally intensive analytics are performed at the cloud which here in the case is a private OpenStack cloud. The random forest and the Xtreme Gradient Boosting algorithms are employed for analytics and the results of the smart camera and the two machine learning algorithms are considered for the detection. The results from the smart camera are given higher weight and an ensemble decision making using voting helps in more accurate detection. The results of the ensemble learning is shown in Table 4. The overall accuracy of detection by the ensemble learner is 98.13%.

The dataset was split for training (70%) and testing (30%) on the proposed infrastructure using the proposed ensemble algorithm. Table 4 shows The values of accuracy, Precision, Recall and the F1 score of the proposed algorithm.

The parameters that are considered to increase the accuracy of detection by various researchers is described in the Table 5. These parameters help generalize the detection process based on the personalized parameters of each person. The detection varies based on the additional parameters of each person since the vital parameters normal for one may not be normal for other.

Table 4 Algorithm estimation

Class	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
1	99.73	93.00	99.00	95.00
2	99.78	94.00	98.00	96.00
3	99.6	90.00	97.00	93.00
4	99.67	91.00	98.00	94.00
5	99.53	86.00	98.00	92.00
6	99.27	100.00	96.00	98.00
7	99.51	100.00	98.00	99.00
8	99.93	100.00	100.00	100.00
9	99.66	94.00	94.00	94.00
10	99.6	98.00	98.00	98.00
11	99.98	100.00	100.00	100.00

Table 5 Fall detection parameters

Participants	Additional parameters	Devices used
Older adults, patients suffering from palsy, geriatric patients identified with the risk of falling, older adults with the risk of falling	Age, gender, weight, BMI, use of walking aid	Accelerometer, infrared sensor along with an alarm, gyroscope, magnetometer, camera, doppler radar, kinect

6 Proposed MFC Architecture Versus Cloud Architecture

As discussed in the previous sections, the detection accuracy is acceptable in the range of 98.13%. The use of the proposed MFC infrastructure as its advantage of reduced latency, low bandwidth use over the use of just cloud infrastructure. The end to end latency of the edge detection to the decision making of raising alarm in the fog layer is analysed which is much lesser than the cloud infrastructure. The end to end latency of the proposed and the cloud infrastructure is given in Fig. 11.

The number of cameras are increased as the area of surveillance is increased. This is simulated using the iFogSim simulator where the real-time parameters are simulated with increase in the number of cameras. The latency is measured for each increase in the number of cameras. The Latency includes the time of edge detection, Communication delay from the edge to the mist, Data filtering at the mist, Computation at the cloud, and decision at the fog for alarms. The fog latency includes

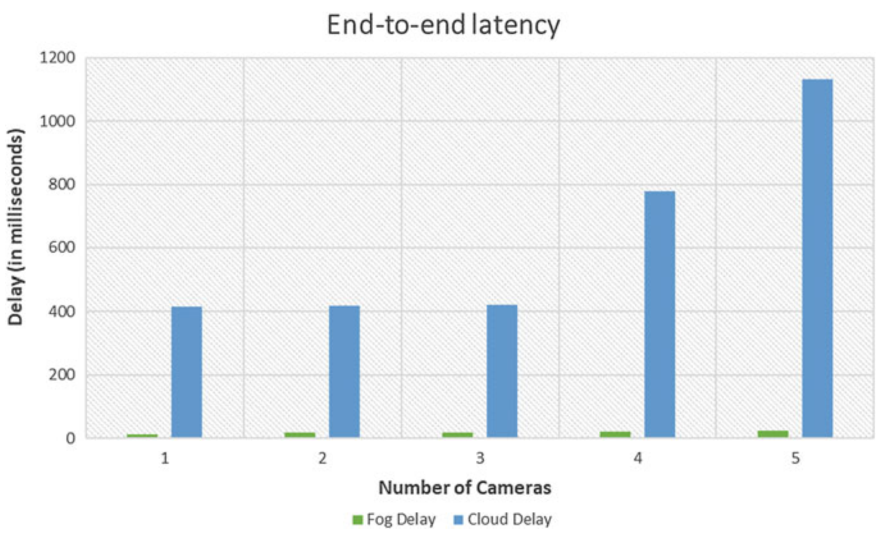


Fig. 11 End to end latency comparison

the processing at the edge, mist and the fog layer. The cloud latency involves the process of filtering, computation and the decision making at the cloud.

Large data is collected by the sensors and the smart camera. Sending all these data to the cloud each and every time in the absence of the intermediate Fog layer induces a very high bandwidth usage. The introduction of the mist and the fog layer helps in the reduction of the amount of data being sent to the upper layer. The initial filtering happens at the edge device which is the smart camera, where in only if a fall is detected, the decision is sent to the upper layer for further decision making. The next level of filtering is done at the mist layer, where the is basic filtering of the sensor data where only the values above the threshold is pushed to the upper layer for decision making. The final decision of raising alarm is done at the fog layer which is closer to the ground in order to provide near real-time alarms. Figure 12 shows the comparison of the network utilization between the fog and cloud infrastructure. The fog layer provides an efficient bandwidth utilization as compared to the cloud only infrastructure.

Thus from the above experiments, the proposed architecture for detection proves to be better than the cloud in terms of latency, network utilization and accuracy. The introduction of the edge computing and the intermediary filtering layers helps improve the overall performance of the hosted application. The decision on raising alarm is also an important feature to be taken care of. When the system has detected a fall, it is very important to raise an alarm based on the intensity of the fall. When a person has fallen and the fall is not severe, raising an alarm would cause chaos to the person and his/her relatives. Thus our proposed system also analyses the time taken to getup back to the normal position and raises an alarm accordingly. The time based analysis is given in Table 6. The timing of the alarm can be changed based on the age and other medical history of the person being monitored in order to have a more precise alarm system.

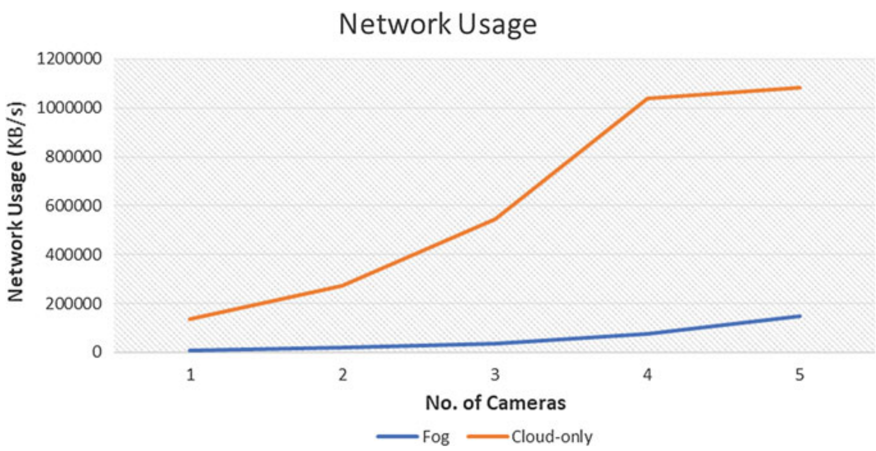


Fig. 12 Network utilization comparison

Table 6 Decisions on alarms

S. No	Time	Type of alarm
1	Less than a minute	Minimal alarm for safe intervention
2	1 min to 1 min 30 sec	Alarm for first aid and further investigation
3	Greater than a minute and 30 sec	Emergency alarm

7 Conclusion and Future Work

The Mist-Fog-Cloud architecture to detect the fall with low latency high accuracy has been proposed in assistance to the cloud infrastructure. The fog layer has been implemented using the Software Defined Networking switch which provides a smart networking component to the architecture there by leaving the pressure of setting up and managing the network off the developer. The fog controller provides an overall view of the network and the details of the nodes thus helping in the creating a fault tolerant distributed fog infrastructure. The first level of fall detection is done at the edge running a compressed DNN. The next level of detection consists of the basic threshold based filtering and ensemble learning of the sensor values which increases the detection accuracy to 98.13%. Finally the decision of alarm is done at the fog layer which reduces the latency to raise the alarm.

The future work of the proposed algorithm involves the use of other vital parameters inorder to improve the accuracy still more and provide better generalization of fall detection. The ensemble algorithm may also include reinforcement learning which can still be able to assess the values and predict the fall still more reducing the time of detection at a much higher accuracy.

References

1. Xue, Y., Jin, L.: A naturalistic 3d acceleration-based activity dataset and benchmark evaluations. In: 2010 IEEE International Conference on Systems, Man and Cybernetics, pp. 4081–4085. IEEE (2010)
2. Mubashir, M., Shao, L., Seed, L.: A survey on fall detection: principles and approaches. Neurocomputing **100**, 144–152 (2013)
3. Igual, R., Medrano, C., Plaza, I.: Challenges, issues and trends in fall detection systems. Biomed. Eng. Online **12**(1), 66 (2013)
4. Robinovitch, S.N., Brumer, R., Maurer, J.: Effect of the squat protective response on impact velocity during backward falls. J. Biomech. **37**(9), 1329–1337 (2004)
5. Ke, S.-R., Thuc, H.L.U., Lee, Y.-J., Hwang, J.-N., Yoo, J.-H., Choi, K.-H.: A review on video-based human activity recognition. Computers **2**(2), 88–131 (2013)
6. Incel, O.D., Kose, M., Ersoy, C.: A review and taxonomy of activity recognition on mobile phones. BioNanoScience **3**(2), 145–171 (2013)

7. World Health Organization: Who Global Report on Falls Prevention in Older Age 2007. World Health Organization, pp. 1–7 (2015)
8. Tran, T.-H., Le, T.-L., Hoang, V.-N., Vu, H.: Continuous detection of human fall using multimodal features from kinect sensors in scalable environment. Comput. Methods Programs Biomed. **146**, 151–165 (2017)
9. Makhlouf, A., Nedjai, I., Saadia, N., Ramdane-Cherif, A.: Multimodal system for fall detection and location of person in an intelligent habitat. Procedia Comput. Sci. **109**, 969–974 (2017)
10. Zhang, Q., Cheng, L., Boutaba, R.: Cloud computing: state-of-the-art and research challenges. J. Internet Serv. Appl. **1**(1), 7–18 (2010)
11. Massonet, P., Naqvi, S., Ponsard, C., Latanicki, J., Rochwerger, B., Villari, M.: A monitoring and audit logging architecture for data location compliance in federated cloud infrastructures. In: IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum, vol. **2011**, pp. 1510–1517. IEEE (2011)
12. OpenFog Consortium Architecture Working Group: Openfog Reference Architecture for Fog Computing. OPFRA001 **20817**, 162 (2017)
13. Bonomi, F., Milito, R., Zhu, J., Addepalli, S.: Fog computing and its role in the internet of things. In: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing. ACM, pp. 13–16 (2012)
14. Pop, D., Iuhasz, G., Craciun, C., Panica, S.: Support services for applications execution in multi-clouds environments. In: 2016 IEEE International Conference on Autonomic Computing (ICAC), pp. 343–348. IEEE (2016)
15. Tomovic, S., Yoshigoe, K., Maljevic, I., Radusinovic, I.: Software-defined fog network architecture for iot. Wirel. Pers. Commun. **92**(1), 181–196 (2017)
16. Yu, X.: Approaches and principles of fall detection for elderly and patient. In: HealthCom 2008-10th International Conference on e-health Networking, Applications and Services, pp. 42–47. IEEE (2008)
17. Perry, J.T., Kellog, S., Vaidya, S.M., Youn, J.-H., Ali, H., Sharif, H.: Survey and evaluation of real-time fall detection approaches. In: 6th International Symposium on High Capacity Optical Networks and Enabling Technologies (HONET), vol. **2009**, pp. 158–164. IEEE (2009)
18. Hijaz, F., Afzal, N., Ahmad, T., Hasan, O.: Survey of fall detection and daily activity monitoring techniques. In: 2010 International Conference on Information and Emerging Technologies, pp. 1–6. IEEE (2010)
19. Delahoz, Y.S., Labrador, M.A.: Survey on fall detection and fall prevention using wearable and external sensors. Sensors **14**(10), 19806–19842 (2014)
20. Schwickert, L., Becker, C., Lindemann, U., Maréchal, C., Bourke, A., Chiari, L., Helbostad, J., Zijlstra, W., Aminian, K., Todd, C., et al.: Fall detection with body-worn sensors. Z. für Gerontol. Und Geriatr. **46**(8), 706–719 (2013)
21. Pannurat, N., Thiemjarus, S., Nantajeewarawat, E.: Automatic fall monitoring: a review. Sensors **14**(7), 12900–12936 (2014)
22. Luque, R., Casilari, E., Morón, M.-J., Redondo, G.: Comparison and characterization of android-based fall detection systems. Sensors **14**(10), 18543–18574 (2014)
23. Di Martino, B.: Applications portability and services interoperability among multiple clouds. IEEE Cloud Comput. **1**(1), 74–77 (2014)
24. Ren, L., Peng, Y.: Research of fall detection and fall prevention technologies: a systematic review. IEEE Access **7** (2019)
25. Erdogan, S.Z., Bilgin, T.T.: A data mining approach for fall detection by using k-nearest neighbour algorithm on wireless sensor network data. Iet Commun. **6**(18), 3281–3287 (2012)
26. Yu, M., Yu, Y., Rhuma, A., Naqvi, S.M.R., Wang, L., Chambers, J.A.: An online one class support vector machine-based person-specific fall detection system for monitoring an elderly individual in a room environment. IEEE J. Biomed. Health Inform. **17**(6), 1002–1014 (2013)
27. Baek, W.-S., Kim, D.-M., Bashir, F., Pyun, J.-Y.: Real life applicable fall detection system based on wireless body area network. In: IEEE 10th Consumer Communications and Networking Conference (CCNC), vol. **2013**, pp. 62–67. IEEE (2013)

28. Dai, J., Bai, X., Yang, Z., Shen, Z., Xuan, D.: Perfalld: A pervasive fall detection system using mobile phones. In: 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp. 292–297. IEEE (2010)
29. Werner, F., Diermaier, J., Schmid, S., Panek, P.: Fall detection with distributed floor-mounted accelerometers: an overview of the development and evaluation of a fall detection system within the project ehome. In: 2011 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops, pp. 354–361. IEEE (2011)
30. Sengto, A., Leauhatong, T.: Human falling detection algorithm using back propagation neural network. In The 5th: Biomedical Engineering International Conference, vol. **2012**, pp. 1–5. IEEE (2012)
31. Yu, M., Rhuma, A., Naqvi, S.M., Wang, L., Chambers, J.: A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment. IEEE Trans. Inf. Technol. Biomed. **16**(6), 1274–1286 (2012)
32. Ojetola, O., Gaura, E.I., Brusey, J.: Fall detection with wearable sensors–safe (smart fall detection). In: 2011 Seventh International Conference on Intelligent Environments, pp. 318–321. IEEE (2011)
33. Shoaib, M., Dragon, R., Ostermann, J.: View-invariant fall detection for elderly in real home environment. In: Fourth Pacific-Rim Symposium on Image and Video Technology. IEEE **2010**, 52–57 (2010)
34. Shahzad, A., Kim, K.: Falldroid: an automated smart-phone-based fall detection system using multiple kernel learning. IEEE Trans. Ind. Inform. **15**(1), 35–44 (2018)
35. Asadi-Aghbolaghi, M., Clapes, A., Bellantonio, M., Escalante, H.J., Ponce-López, V., Baró, X., Guyon, I., Kasaei, S., Escalera, S.: A survey on deep learning based approaches for action and gesture recognition in image sequences. In, : 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), vol. **2017**, pp. 476–483. IEEE (2017)
36. Zhang, C., Patras, P., Haddadi, H.: Deep learning in mobile and wireless networking: a survey. IEEE Communications Surveys & Tutorials (2019)
37. Vavoulas, G., Pediaditis, M., Chatzaki, C., Spanakis, E., Tsiknakis, M.: Artificial intelligence: Concepts, methodologies, tools, and applications (2017)
38. Sucerquia, A., López, J., Vargas-Bonilla, J.: Sisfall: a fall and movement dataset. Sensors **17**(1), 198 (2017)
39. Xu, T., Zhou, Y., Zhu, J.: New advances and challenges of fall detection systems: a survey. Appl. Sci. **8**(3), 418 (2018)
40. Ma, X., Wang, H., Xue, B., Zhou, M., Ji, B., Li, Y.: Depth-based human fall detection via shape features and improved extreme learning machine. IEEE J. Biomed. Health Inform. **18**(6), 1915–1922 (2014)
41. Zhang, Z., Conly, C., Athitsos, V.: Evaluating depth-based computer vision methods for fall detection under occlusions. In: International Symposium on Visual Computing, pp. 196–207. Springer (2014)
42. Martínez-Villaseñor, L., Ponce, H., Brieva, J., Moya-Albor, E., Núñez-Martínez, J., Peñafort-Asturiano, C.: Up-fall detection dataset: a multimodal approach. Sensors **19**(9), 1988 (2019)
43. Sze, V., Chen, Y.-H., Yang, T.-J., Emer, J.S.: Efficient processing of deep neural networks: a tutorial and survey. Proc. IEEE **105**(12), 2295–2329 (2017)
44. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149 (2015)
45. Bilal, K., Khalid, O., Erbad, A., Khan, S.U.: Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers. Comput. Netw. **130**, 94–120 (2018)
46. Dastjerdi, A.V., Buyya, R.: Fog computing: helping the internet of things realize its potential. Computer **49**(8), 112–116 (2016)
47. Cao, Y., Chen, S., Hou, P., Brown, D.: Fast: a fog computing assisted distributed analytics system to monitor fall for stroke mitigation. In: 2015 IEEE International Conference on Networking, Architecture and Storage (NAS), pp. 2–11. IEEE (2015)

48. Aazam, M., Zeadally, S., Harras, K.A.: Offloading in fog computing for iot: review, enabling technologies, and research opportunities. Futur. Gener. Comput. Syst. **87**, 278–289 (2018)
49. Lu, R., Heung, K., Lashkari, A.H., Ghorbani, A.A.: A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced iot. IEEE Access **5**, 3302–3312 (2017)
50. Wang, K., Shao, Y., Xie, L., Wu, J., Guo, S.: Adaptive and fault-tolerant data processing in healthcare iot based on fog computing. IEEE Trans. Netw. Sci. Eng. (2018)
51. Cheng, Y., Wang, D., Zhou, P., Zhang, T.: A survey of model compression and acceleration for deep neural networks. arXiv preprint arXiv:1710.09282 (2017)
52. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks?' In: Advances in Neural Information Processing Systems, pp. 3320–3328 (2014)
53. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift arXiv preprint arXiv:1502.03167 (2015)

Wearable Sensors Data-Fusion and Machine-Learning Method for Fall Detection and Activity Recognition

Hristijan Gjoreski, Simon Stankoski, Ivana Kiprijanovska, Anastasija Nikolovska, Natasha Mladenovska, Marija Trajanoska, Bojana Velichkovska, Martin Gjoreski, Mitja Luštrek and Matjaž Gams

Abstract Human falls are common source of injury among the elderly, because often the elderly person is injured and cannot call for help. In the literature this is addressed by various fall-detection systems, of which most common are the ones that use wearable sensors. This paper describes the winning method developed for the Challenge Up: Multimodal Fall Detection competition. It is a multi-sensor data-fusion machine-learning method that recognizes human activities and falls using 5 wearable inertial sensors: accelerometers and gyroscopes. The method was evaluated on a dataset collected by 12 subjects of which 3 were used as a test-data for the challenge. In order to optimally adapt the method to the 3 test subjects, we performed an unsupervised similarity search—that finds the three most similar users to the three users in the test data. This helped us to tune the method and its parameters to the 3 most similar users as the ones used for the test. The internal evaluation on the 9 users showed that with this optimized configuration the method achieves 98% accuracy. During the final evaluation for the challenge, our method achieved the highest results (82.5% F1-score, and 98% accuracy) and won the competition.

Keywords Activity recognition · Fall detection · Accelerometers · Machine learning · Wearable sensors

Electronic Supplementary Material The online version of this chapter (https://doi.org/10.1007/978-3-030-38748-8_4) contains supplementary material, which is available to authorized users.

H. Gjoreski (✉) · S. Stankoski · I. Kiprijanovska · A. Nikolovska · N. Mladenovska · M. Trajanoska · B. Velichkovska
Faculty of Electrical Engineering and Information Technologies, Ss Cyril and Methodius University, Skopje, North Macedonia
e-mail: hristijang@feit.ukim.edu.mk

M. Gjoreski · M. Luštrek · M. Gams
Jožef Stefan Institute, Ljubljana, Slovenia

H. Ponce et al. (eds.), *Challenges and Trends in Multimodal Fall Detection for Healthcare*, Studies in Systems, Decision and Control 273,
https://doi.org/10.1007/978-3-030-38748-8_4

1 Introduction

Human falls are critical health-related problems for the elderly [14], and the statistics show that approximately 30% of people over the age of 65 fall each year, and this proportion increases to 40% in those aged more than 70 (Gillespie et al. [10]. According to Worth Health Organization [32] about 20% of the elderly who fall require medical attention. Furthermore, falls and the fear of falling are important reasons for nursing-home admission [28]. Falls are particularly critical when the elderly person is injured and cannot call for help. These reasons, combined with the increasing accessibility and miniaturization of sensors and microprocessors, are driving the development of fall-detection (FD) systems.

Fall detection has received significant attention in recent years, however it still represents a challenging task [13]. The Challenge UP: Multimodal Fall Detection competition[1] presents a great opportunity the activity-recognition community to test and compare their approaches. The goal of the challenge is to recognize, as accurately as possible, 12 activities, including 5 falls.

This paper describes the method that we developed for the competition.[2] It is a multi-sensor data-fusion machine-learning method that recognizes human activities and falls using 5 accelerometers and 5 gyroscopes. It includes several steps: data preprocessing, data segmentation, sensor orientation correction, feature extraction, feature selection, hyperparameter optimization, and training a machine learning model.

The evaluation was performed on a dataset provided by the organizers of the competition. It consists of wearable sensors data collected by 12 subjects of which 3 were used as a test data for the challenge. Our method was ranked first, achieving highest recognition performance: 82.5% F1-score, and 98% accuracy.

The rest of the paper is organized as follows. The dataset is explained in Sect. 2, whereas section three is dedicated to explaining the methodology of our system. In the description of the methodology we discuss the preprocessing applied to our data, the sensor orientation correction, as well as the feature extraction and feature selection procedures. In Sect. 4 we focus on the evaluation methods for the pipeline, and in Sect. 5 we conclude the paper.

2 Related Work

Activity recognition (AR) and fall detection (FD) approaches can be divided into those that use wearable and non-wearable sensors, respectively. The most common

---

[1] The Challenge Up Multimodal Competition, available at: https://sites.google.com/up.edu.mx/challenge-up-2019/overview.

[2] The code developed for the challenge is available at: https://github.com/challengeupwinner/challengeupcode.

non-wearable approach is based on cameras [34]. Video-based human activity recognition is a hot research area in computer vision to help people with special needs. Miguel et al. [23] developed a computer-vision based system to recognize abnormal activity in daily life in a supportive home environment. The system tracked activity of subjects and summarized frequent active regions to learn a model of normal activity. It detected falling events as abnormal activity, which is very important in-patient monitoring systems. Although this approach is physically less intrusive for the user compared to one based on wearable sensors, it suffers from problems such as target occlusion, time-consuming processing and privacy concerns.

The most mature approach to both AR and FD is probably using wearable accelerometers, [4, 15, 17, 27]. The most common accelerometer-based AR approach uses machine learning. Typically, a sliding window passes over the stream of sensor data, and data in each window are classified with one of the known classification methods, such as decision trees (DTs) and support vector machines (SVM). The most frequent AR task is classifying activities in relation to movement, e.g., walking, running, standing still and cycling [17, 27].

An alternative approach to accelerometer-based AR is based on manually created rules [20]. These rules are usually based on features that are calculated from sensor orientations and accelerations. Bourbia et al. [4] presented an approach in which decision rules are used to recognize activities. Another implementation of such rules was presented by Lai et al. [21], who used six accelerometers, placed on the neck, waist, both wrists and both thighs and reported accuracy of 99.5%.

Fall detection has also been addressed in related studies [22]. Some of the first studies include Williams et al. [31] and Doughty et al. [7]. In this approaches the fall is detected by detecting a change in body orientation from upright to lying immediately after a large negative acceleration. Later, this algorithm was upgraded and fine-tuned by Aziz et al. [2] and Putra et al. [24, 25].

Degen et al. [6] presented a fall detector worn on the wrist that incorporates a multi-stage fall detection algorithm. The first condition is the detection of a high velocity towards the ground. Next, an impact needs to be detected within 3 s. After impact, the activity is observed for 60 s, and if at least 40 s of inactivity are recorded, an alarm is activated. The results show no false alarms, but large percentage of backwards and sideways falls were not detected.

The most common approaches to FD are rules that use thresholds applied to accelerations and features derived from them. Ren et al. [26] proposed personalized and adaptive threshold model and showed that accuracy increases for 1–3% compared to other threshold models. Wu et al. [33] developed a fall detection system based on a single, triaxial, accelerometer, which results showed lower sensitivity and specificity compared to multi-sensor approaches. Hardjianto et al. [18] used an accelerometer on smartphone, with six variations of the placement of the device. The method used for fall detection is threshold method applying only one parameter, the value of resultant acceleration. It resulted in 98.1% of accuracy, 96.9% of sensitivity, and 100 specificity.

In recent years there are also approaches that use machine learning instead of threshold-based algorithms for FD. Putra et al. [24, 25] proposed an event-triggered machine learning approach that aligns each fall stage so that the characteristic features of the fall stages are more easily recognized. The F1-score reached by the chest-worn sensor is 98% and 92% for the waist-worn sensor.

3 Dataset

The dataset provided for the competition includes 12 activities, performed by 12 subjects. The data from 9 of the subjects were released for training the models, the data from the remaining 3 subjects were used for final evaluation of the competitors. The subjects performed 7 simple human daily activities (walking, standing, sitting, picking up an object, jumping, laying, on knees) and 5 types of falls (falling forward using hands, falling forward using knees, falling backwards, falling sideward, falling sitting in empty chair). The distribution of the data according to the activities is shown in Fig. 1.

The dataset was recorded using multiple types of sensors, i.e. wearable sensors, ambient sensors and vision devices. The wearable sensors were located in the left wrist, under the neck, at right pocket of pants, at the middle of waist (in the belt), and in the left ankle. Each of these sensors contains 3-axis accelerometer, 3-axis gyroscope and a sensor for ambient light. Also, one electroencephalograph, located at the forehead, was used for measuring the brainwave signals. The ambient sensors include six infrared sensors placed above the floor of the room, and all of them reported changes in interruption of the optical devices. Lastly, the dataset was enriched with images from two cameras, which captured the subjects while doing the activities. The sampling rate of the sensors used in the dataset is 20 Hz.

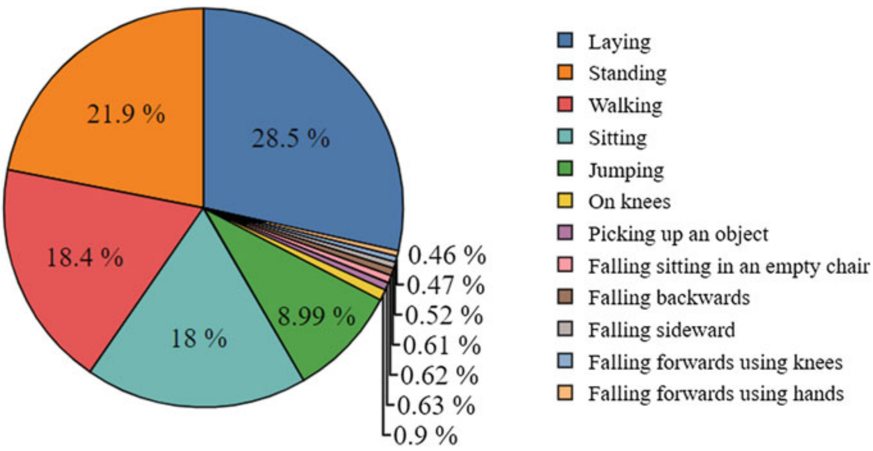


Fig. 1 The distribution of the data according to the activities

4 Method

The method that we developed for this study is shown in Fig. 2. It includes data preprocessing (filters, data segmentation—sliding window), feature engineering and extraction, feature selection, and finally a classification model to recognize the activity. Each of the steps are described in the following subsections.

4.1 Data Preprocessing

Signal segmentation is very important step in the activity recognition process. Therefore, we segmented the sensor signals using a sliding window size of 0.5 s with a 0.25 s overlap. This way the model recognizes activity every 0.25 s. The window size and the sliding factor are important in data processing and have to be tuned correctly for the task at hand. Longer windows naturally contain more data and are expected to enable greater classifying accuracy, especially for more complex activities. Shorter windows, on the other hand, make it possible to detect activity changes faster. Considering the fact that we aim to achieve accurate fall detection and falls last shorter than one second on average in this dataset, the window size had to be chosen so that it is small enough in relation to average fall duration. The optimal window size in our experiments was determined empirically.

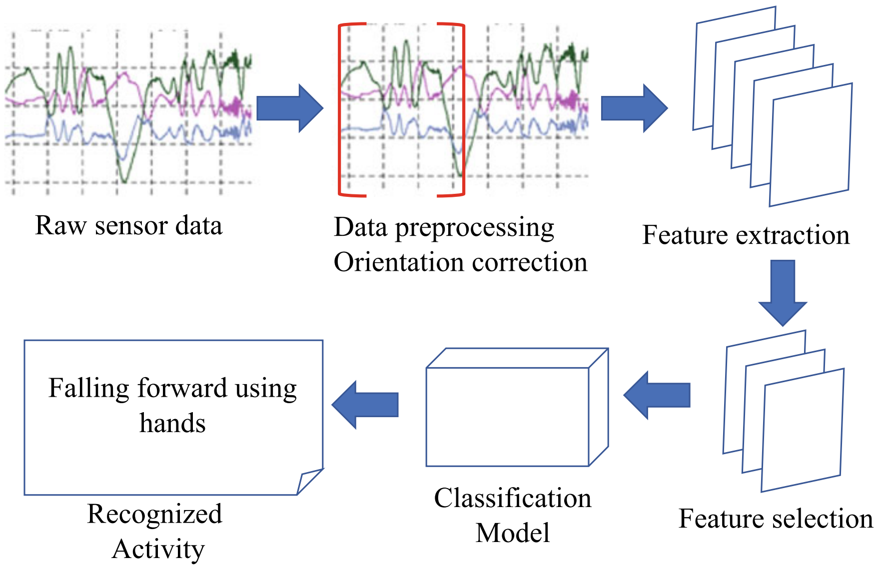


Fig. 2 Activity recognition and fall detection pipeline

Beside the raw sensor signals (x, y and z axis) we additionally extracted the magnitude of the acceleration vector. It was calculated for the accelerometer, as well as the gyroscope and is shown in (1).

$$m = \sqrt{x^2 + y^2 + z^2} \tag{1}$$

4.2 Orientation Correction

After analyzing the data, we noted that the orientation of the sensors varies between users, and even more between different trials. Therefore, we have developed a method that corrects the orientation of the sensors, i.e., it uses rotation matrices to correct (rotate) the accelerometers data. The method corrects accelerometer axes orientation by applying a rotation transformation to the device's raw data [16]. To calculate the angle between the actual acceleration (e.g. the Earth's gravity (g) for static activities) and some of the axis (e.g., x-axis) we used the formula shown in (2)—where the values a_x, a_y and a_z represent the actual acceleration vector.

$$\varphi_x = \arccos\left(\frac{a_x}{\sqrt{a_x^2 + a_y^2 + a_z^2}}\right) \tag{2}$$

The coordinate system is rotated using trigonometry and rotational matrices, in such a way that it corrects the data. In order to do this, one should calculate the difference between the expected angle (φ_x) and the rotated angle (φ_{xr}). This difference gives the angle by which the coordinate system should be rotated in order to correct the orientation of the sensor. The rotation is performed by a rotation matrix, which describes a rotation of a coordinate system with respect to another orientation. An acceleration vector in the initial reference frame can be transformed into a vector in a rotated frame by multiplication of the initial vector with the rotation matrix [9]. In three dimensions, rotations are possible around the three principal axes (x, y and z).

To achieve this, we used the *standing* activity as a reference in order to compute the current angle φ_{xr}. This kind of reference angles (orientation vectors) are defined for all accelerometers (neck, wrist, belt, right pocket and ankle). As a reference angle (φ_x) we used the angle under which the sensors of our referent subject are placed. The method then calculates the rotation angle for every other subject in the dataset with respect to the referent subject. Once it is calculated, all raw accelerometer data thereafter are multiplied by the rotation matrix to achieve the corrected orientation.

4.3 Feature Extraction and Selection

In order to extract as much features as possible, we used the TSFRESH library. It performs time series feature extraction and selection, which we exploited in generating approximately 12,000 features. In the next step we performed feature selection in order to reduce the number of features and keep only the most relevant ones. We focused on removing those features which did not contribute to the accuracy of our model and/or increased the odds for overfitting [19].

In the first step, we discarded the features containing missing and Not-a-Number (NaN) values, which resulted in leaving 7700 features. Then, we estimated the mutual information between each feature and the label (activity). We sorted the features in descending order according to this value, as the higher the mutual information, the higher the dependency of the label from the corresponding feature. In the next step, we divided the features in groups of 100. We began with the first group of features, where we calculated the Pearson correlation coefficient for every pair of features. If the correlation between a pair exceeded a threshold of 0.8, out of the two we removed the feature with the lower mutual information. To the remaining features of the group we appended the following group of 100 features. The process was repeated until all the initial groups of 100 were iterated.

Finally, we selected the definite set of features using a wrapper feature selection algorithm. Here, the first step was to utilize the best scoring feature in regard to the value of its mutual information with the label and train a classification model to estimate the macro F1-score. Then, in every following step, the next feature of the uncorrelated features was added to the previously kept features. Once the feature was added, the model was retrained and a new F1 score obtained. If, at each step, the score decreased not more than 1%, the newly added feature was kept. Otherwise, the feature was dismissed, making the feature list before and after said step unchanged. This measure, repeated for all the remaining features, allowed us to take into consideration every wearable sensor and at the same time prevented us from overfitting our model. The final feature selection resulted in 152 relevant, uncorrelated, class-defining features.

4.4 Classification

We compared three machine learning algorithms: *Decision Tree*, *XGBoost*, and *Random Forest*. After thorough evaluation and comparison (see the results in Sect. 5.3), we have chosen the best performing one, i.e., *Random Forest*. This algorithm showed more robust performance when tested in different scenarios and different users. In the following paragraphs each of the algorithms is described in relation to our activity recognition task.

Decision Tree [29] is an algorithm that learns a model in a form of a tree structure. In particular, it divides the dataset into smaller subsets while at the same time the decision tree is incrementally learned. The final result is a tree with decision nodes with two or more branches, each representing values for the feature tested, and leaf nodes which represent a decision on the activity. In our case, all of the features are numeric (this means the same feature can be used multiple times), which resulted in very large trees.

XGBoost [5] is efficient implementation of the gradient boosted trees algorithm. It is a supervised learning algorithm, which predicts the activity by combining the estimates of a set of simpler, weaker models—in our case decision trees models. It uses a gradient descent algorithm to minimize the loss when adding new models. This way, it minimizes an objective function that combines a convex loss function and a penalty term for model complexity. The training proceeds iteratively, adding new trees that predict the errors of prior trees that are then combined with previous trees to make the final prediction of the activity.

Random Forest [11] is ensemble of decision tree classifiers. During training, the Random Forest algorithm creates multiple decision trees each trained on a bootstrapped sample of the original training data and searches only across a randomly selected subset of the input variables to determine a split (for each node). For classification, each tree in the Random Forest predicts the activity, and the final output of the classifier is determined by a majority vote of the trees. This way, the activity that is predicted by most of the decision trees will be chosen as final.

4.5 Hyperparameter Optimization

In the final step, we performed a hyperparameter optimization for each of the 3 algorithms explained in the previous subsection. Hyperparameter optimization is a process of finding a set of optimal hyperparameters for a learning algorithm, where a hyperparameter is a parameter whose value is used to control the learning process. Finally, this optimization finds a tuple of hyperparameters that yields an optimal model which minimizes the error function (maximizes the accuracy) given a dataset.

There are different methods for optimizing hyperparameters: Grid Search; Random Search, Bayesian optimization, Gradient-based optimization, etc. We chose Random Search method as it is one of the most commonly used methods for hyperparameter optimization in time-series data and showed to be more robust compared to the other techniques [3]. Random Search replaces the exhaustive enumeration of all combinations by selecting them randomly. This can be simply applied to the discrete setting, but also generalizes to continuous and mixed spaces. It usually outperforms Grid search, especially when only a small number of hyperparameters affects the final performance of the machine learning algorithm—which was the case in our study. Additionally, Random search is more efficient compared to Grid search, which spends too much time evaluating unpromising regions of the hyperparameter search space because it has to evaluate every single combination in the grid. Random search in contrast, does a better job of exploring the search space and therefore can usually find a good combination of hyperparameters in far fewer iterations [3].

The following hyperparameters were optimized:

- Decision Tree: Maximum number of levels in tree; Minimum number of samples required to split a node; Minimum number of samples required at each leaf node;
- Random Forest: Number of trees in random forest; Number of features to consider at every split; Maximum number of levels in tree; Minimum number of samples required to split a node; Minimum number of samples required at each leaf node;
- XGBoost: The learning rate; Minimum child weight; Number of estimators; Minimum number of samples required to split a node; Minimum number of samples required at each leaf node; Maximum depth.

5 Evaluation

5.1 Dataset Split

In order to optimally adapt the method to the 3 test users, we have performed an unsupervised similarity search—that finds the three most similar users to the three users in the test data. This helped us to tune the method and its parameters to the 3 most similar users as the ones used for the test.

The method uses each user's data individually and performs a K-means clustering, where K is the number of classes/activities, i.e., we used 6 (all the falls are similar and therefore we grouped them). After performing the clustering, then we calculated the centroid for each cluster, which resulted in 6 centroids per user. Then, we calculated a distance matrix that contained the distances between the 6 clusters of the train user, and the 6 centroids from the test user. We calculated this matrix for each pair of users, i.e., we calculated 27 distance matrices (the 9 users in train vs the 3 users in test). For each matrix we have calculated the *distance between the pair of users*, i.e., we calculated the minimum sum of the distances that covers all the 6 clusters. This way we were able to find the 3 most similar (minimal distance) users to the ones used for the test.

The most similar subjects to the test users: 15, 16 and 17, are: 4, 3 and 13 respectively.

5.2 Evaluation Metrics

Accuracy is the most commonly used metric that can be calculated from a confusion matrix. Its main drawback is that it hides information on the specific nature of errors (the proportions of FP and FN) [30]. It is calculated as following:

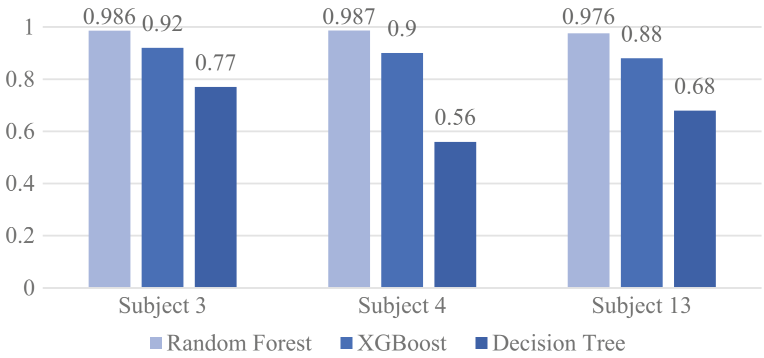


Fig. 3 Accuracy comparison

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \tag{3}$$

We assessed the performance of the model by not only using the accuracy, but also macro F1-score (F1-macro). The F1-macro is the unweighted mean of the F1-scores for the different labels [1]. It can be calculated as harmonic mean between precision and recall, where the average is calculated per label and then averaged across all labels. If Pi and Ri are the precision and recall for each label, then the F1-macro is calculated as in (4):

$$F1\text{-}macro = \frac{1}{Q}\sum_{i=1}^{Q}\frac{2 * Pi * Ri}{Pi + Ri} \tag{4}$$

5.3 Algorithm Comparison

A summary of the results is shown in Figs. 3 and 4, which shows that the system successfully recognized the activities using optimized Random Forest classifier, with high accuracy (97–99%), and F1 macro score (84–90%). The results using other classifiers are significantly worse.

5.4 Confusion Matrices

The following 4 confusion matrices show the performance achieved for the 3 users summarized (Table 2), and each of the users individually (Tables 3, 4 and 5). The

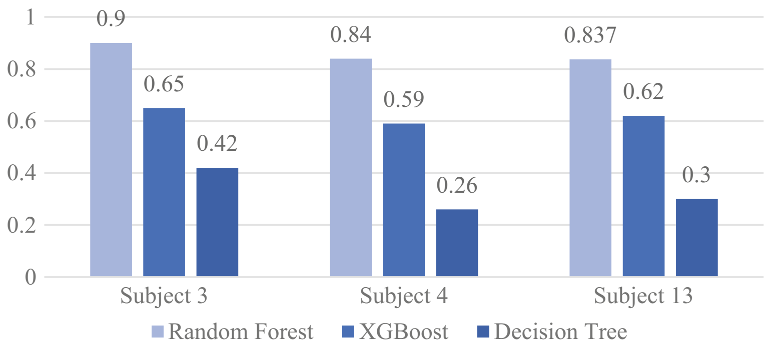


Fig. 4 Comparison macro F1-score

Table 1 Activities and their corresponding IDs

Activity ID	Description
1	Falling forward using hands
2	Falling forward using knees
3	Falling backwards
4	Falling sideward
5	Falling sitting in empty chair
6	Walking
7	Standing
8	Sitting
9	Picking up an object
10	Jumping
11	Laying
12	On knees

IDs of the activities correspond to Table 1. Note that in the training data the *on knees* activity is missing and therefore it is omitted in the results.

The results show that activities 6, 7, 8, 9, 10, 11 (walking, standing, sitting, picking up an object, jumping and lying) are correctly recognized most of the time. Some problem occurs with falling activities, but most likely this is due to the small number of instances and the impossibility of the model to be enough trained on them.

5.5 Challenge UP Competition Final Results

The confusion matrix obtained by the final evaluation during the competition is presented in Table 6. The overall results show that our method achieved 82.5% F1-macro, and 98% accuracy. Although the resulting matrix looks generally satisfying, it

Table 2 Confusion matrix for the 3 users (Subject 3, Subject 4, Subject 13)

Activity	1	2	3	4	5	6	7	8	9	10	11
1	23	1	1	1	0	2	3	0	1	0	0
2	1	23	1	3	0	3	0	0	1	1	1
3	0	1	33	3	3	1	2	0	0	1	3
4	0	0	4	42	0	0	3	0	0	0	1
5	2	0	3	3	27	0	2	0	0	0	0
6	0	0	0	0	0	1909	3	0	7	0	0
7	0	0	0	1	0	34	2357	0	3	2	0
8	0	0	0	0	0	0	0	1902	4	0	29
9	0	0	0	0	2	0	1	0	56	0	3
10	1	0	0	0	0	8	7	0	0	943	0
11	0	0	0	2	1	0	0	0	0	0	1021

Table 3 Confusion matrix for User 3

Activity	1	2	3	4	5	6	7	8	9	10	11
1	9	0	0	0	0	0	2	0	1	0	0
2	1	14	0	2	0	2	0	0	0	1	1
3	0	0	15	0	0	1	2	0	0	0	3
4	0	0	0	15	0	0	3	0	0	0	1
5	0	0	3	0	20	0	0	0	0	0	0
6	0	0	0	0	0	657	0	0	0	0	0
7	0	0	0	1	0	7	759	0	0	2	0
8	0	0	0	0	0	0	0	642	4	0	0
9	0	0	0	0	0	0	0	0	23	0	0
10	0	0	0	0	0	2	4	0	0	309	0
11	0	0	0	2	1	0	0	0	0	0	1021

is noticeable that the biggest issue is the second activity—*falling forward using knees*. Almost half of the instances that belong to this activity are classified as *standing* by our model. We speculate that the reason for this is the lack of instances which represent this activity. Another issue is the imperfection of data labeling. The activity *falling forward using knees* consists of two parts: first standing and then kneeling. It is possible that much of the standing may be labeled as falling due to too little available time.

The rest of the activities are recognized with much higher accuracy. The activities *jumping* and *sitting* are recognized with 100%, which is due to the dissimilarity to any other activity. The other three activities recognized with 100% are: *standing*, *falling backwards* and *falling sideward*, probably because of the orientation correction procedure. We speculate that due to the orientation correction our model was able to

Table 4 Confusion matrix for User 4

Activity	1	2	3	4	5	6	7	8	9	10	11
1	6	0	0	0	0	1	1	0	0	0	0
2	0	4	0	1	0	0	0	0	1	0	0
3	0	0	9	0	2	0	0	0	0	1	0
4	0	0	0	8	0	0	0	0	0	0	0
5	2	0	0	2	4	0	2	0	0	0	0
6	0	0	0	0	0	645	0	0	7	0	0
7	0	0	0	0	0	6	772	0	3	0	0
8	0	0	0	0	0	0	0	640	0	0	0
9	0	0	0	0	1	0	0	0	23	0	0
10	1	0	0	0	0	2	1	0	0	330	0
11	0	1	0	2	0	0	0	0	1	7	1037

Table 5 Confusion matrix for User 13

Activity	1	2	3	4	5	6	7	8	9	10	11
1	8	1	1	1	0	1	0	0	0	0	0
2	0	5	1	0	0	1	0	0	0	0	0
3	0	1	9	3	1	0	0	0	0	0	0
4	0	0	4	19	0	0	0	0	0	0	0
5	0	0	0	1	3	0	0	0	0	0	0
6	0	0	0	0	0	607	3	0	0	0	0
7	0	0	0	0	0	21	826	0	0	0	0
8	0	0	0	0	0	0	0	620	0	0	29
9	0	0	0	0	1	0	1	0	10	0	3
10	0	0	0	0	0	4	2	0	0	304	0
11	1	1	0	1	0	0	0	0	0	0	971

successfully distinguish different falls based on the correct acceleration direction. The final activity *in knees* was poorly recognized, probably due to the short duration (few seconds) and the lack of this activity in the training data.

6 Conclusion

The paper presented the winning ML method of the Challenge Up: Multimodal Fall Detection competition. The method is tuned for robustness and real-time performance by combining multiple wearable inertial sensors: accelerometer and gyroscope, in

Table 6 Confusion matrix—challenge up: multimodal fall detection final results

Activity	1	2	3	4	5	6	7	8	9	10	11	12	Recall
1	10	0	0	0	0	0	0	0	0	0	1	0	0.91
2	0	8	0	0	0	0	6	0	0	0	0	0	0.57
3	0	0	16	0	0	0	0	0	0	0	0	0	1.00
4	0	0	0	15	0	0	0	0	0	0	0	0	1.00
5	0	0	0	0	18	0	0	1	0	0	0	0	0.95
6	0	0	0	0	0	549	0	0	0	0	0	0	1.00
7	2	0	2	1	3	0	659	0	2	0	0	0	0.99
8	0	0	0	0	0	0	0	547	0	0	0	0	1.00
9	0	0	0	0	0	0	2	0	21	0	0	0	0.91
10	0	0	0	0	0	0	0	0	0	279	0	0	1.00
11	4	3	0	0	3	0	1	20	0	0	877	0	0.97
12	0	6	0	0	0	0	0	0	0	0	3	0	0.00
Precision	0.63	0.47	0.89	0.94	0.75	1.00	0.99	0.96	0.91	1.00	1.00	NaN	0.98

Accuracy = 98.03% Precision = 85.77% Recall = 79.42% F1-score = 82.47%

order to recognize activities and detect falls. It includes several steps: data preprocessing, data segmentation, sensor orientation correction, feature extraction, feature selection, hyperparameter optimization, and training a machine learning model.

During the development of the method we have noted that the orientation of the sensors varies between users, and even more between different trials. Therefore, we have developed a method that corrects the orientation of the sensors, i.e., it uses rotation matrices to correct (rotate) the accelerometers data.

We applied extensive feature extraction and selection procedure. It is a three-step procedure that selects an optimal subset of features (152 features) from the 12,000 features initially calculated from the raw sensor data.

Finally, to optimally adapt the method to the 3 test users, we have performed an unsupervised similarity search—that finds the three most similar users to the three users in the test data. This helped us to tune the method and its parameters to the 3 most similar users as the ones used for the test.

The internal evaluation on the 9 users showed that with this optimized configuration the method achieves 98% accuracy. All these steps allowed us to develop accurate fall detection and activity recognition algorithm, that achieved the highest results (82.5% F1-score, and 98% accuracy) at the competition and received the first award.

The method has several limitations. First, it uses 5 wearable sensors, which is impractical for everyday usage by an elderly person. For the future work, we plan to focus more on the practical implementation of the method into a commercial fall detection system. First, we intend to reduce the number of sensors but without losing accuracy. This way the system will be less intrusive and more user-friendly. Another improvement in this direction can be achieved by introducing specially designed clothes, which will include pockets for the sensors. Additionally, the interaction between the user and the system should be introduced by using smartphone, smartwatch, tablet or PC as a medium for showing system's notifications (fall detected, system malfunction, etc.), similar to Gjoreski et al. [12].

Acknowledgements We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. The authors declare that they have no conflict of interest.

References

1. Asch, V.V.: Macro- and micro-averaged evaluation measures. Comput. Sci. (2013)
2. Aziz, O., Musungi, M., Park, E.J., Mori, G., Robinovitch, S.N.: A comparison of accuracy of fall detection algorithms (threshold-based vs. machine learning) using waist-mounted tri-axial accelerometer signals from a comprehensive set of falls and non-fall trials. J. Med. Biol. Eng. Comput. **55**(1), 45–55 (2017)
3. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J Mach. Learn. Res. **13**, 281–305 (2012)
4. Bourbia, A.L., Son, H., Shin, B., Kim, T., Lee, D., Hyun, S.J.: Temporal dependency rule learning based group activity recognition in smart spaces (2016)
5. Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. In: ACM SIGKDD International Conference, vol. 22, pp. 785–794 (2016)
6. Degen, T., Jaeckel, H., Rufer, M., Wyss, S.: SPEEDY: A fall detector in a wrist watch. In: Proceeding Seventh IEEE International Symposium on Wearable Computing, pp. 184–187 (2003)
7. Doughty, K., Lewis, R., McIntosh, A.: The design of a practical and reliable fall detector for community and institutional telecare. J. Telemed. Telecare **6**, S150–S154 (2000)
8. Friedman, S.M., Munoz, B., West, S.K., Rubin, G.S., Fried, L.P.: Falls and fear of falling: Which comes first? A longitudinal prediction model suggests strategies for primary and secondary prevention. J. Am. Geriatr. Soc. 1329–1335 (2000)
9. Friedman, A., Chehade, N.H., Chien, C., Pottie, G.: Estimation of accelerometer orientation for activity recognition, pp. 2076–2079. Engineering in Medicine and Biology Society (EMBC) (2012)
10. Gillespie, L.D., Robertson, M.C., Gillespie, W.J., Lamb, S.E., Gates, S., Cumming, R.G., Rowe, B.H.: Interventions for preventing falls in older people living in the community (review). The Cochrane Library, 4 (2009)
11. Gislason, P.O., Benediktsson, J.A., Sveinsson, J.R.: Random forests for land cover classification. Pattern Recogn. Lett. **27**(4), 294–300 (2006)
12. Gjoreski, H., Bizjak, J., Gams, M.: Using smartwatch as telecare and fall detection device. In: 12th International Conference on Intelligent Environments (IE), pp. 242–245 (2016a)
13. Gjoreski, M., Gjoreski, H., Luštrek, M., Gams, M.: How accurately can your wrist device recognize daily activities and detect falls? Sensors **16**(6), 800 (2016b)
14. Gjoreskic, H., Gams, M., Luštrek, M.: Context-based fall detection and activity recognition using inertial and location sensors. J. Ambient Intell. Smart Env. **6**(4), 419–433 (2014)
15. Gjoreski, H., Lustrek, M., Gams, M.: Accelerometer placement for posture recognition and fall detection. In: Seventh International Conference on Intelligent Environments, pp. 47–54 (2011)
16. Gjoreski, H., Gams, M., Lutrek, M.: Human activity recognition: from controlled lab experiments to competitive live evaluation. In: IEEE International Conference on Data Mining Workshop (ICDMW), pp. 139–145 (2015)

17. Hammerla, N.Y., Halloran, S., Plotz, T.: Deep, convolutional, and recurrent models for human activity recognition using wearables (2016)
18. Hardjianto, M., Istiyanto, J.E., Putra, A.E.: Fall detection on humans using threshold method based on smartphone accelerometer data (2017)
19. Janko, V., Gjoreski, M., Slapničar, G., Mlakar, M., Reščič, N., Bizjak, J., Drobnič, V., Marinko, M., Mlakar, N., Gams, M., Luštrek, M.: Winning the Sussex-Huawei locomotion-transportation recognition challenge. In: Kawaguchi, N., Nishio, N., Roggen, D., Inoue, S., Pirttikangas, S., Van Laerhoven, K. (eds.) Human Activity Sensing. Springer Series in Adaptive Environments. Springer, Cham (2019)
20. Kozina, S., Gjoreski, H., Gams, M., Luštrek, M.: Three-layer activity recognition combining domain knowledge and meta- classification. JMBE **33**(4), 406–414 (2013)
21. Lai, C., Huang, Y.M., Park, J.H., Chao, H.C.: Adaptive body posture analysis for elderly-falling detection with multisensors. IEEE Intell. Syst. **25**, 2–11 (2010)
22. Lustrek, M., Gjoreski, H., Vega, N.G., Kozina, S., Cvetkovic, B., Mirchevska, V., Gams, M.: Fall detection using location sensors and accelerometers. IEEE Pervas. Comput. **14**(4), 72–79 (2015)
23. Miguel, K., Brunete, A., Hernando, M., Gambao, E.: Home camera-based fall detection system for the elderly. Sensors (2017)
24. Putra, I., Brusey, J., Gaura, E., Vesilo, R.: An event-triggered machine learning approach for accelerometer-based fall detection (2017a)
25. Putra, P., Brusey, J., Gaura, E., Vesilo, R.: An event-triggered machine learning approach for accelerometer-based fall detection (2017b)
26. Ren, L., Shi, W.: Chameleon: Personalised and adaptive fall detection of elderly people in home-based environments (2016)
27. Ronao, C.A., Cho, S.-B.: Human activity recognition with smartphone sensors using deep learning neural networks. Expert Syst. Appl. **59**, 235–244 (2016)
28. Tinetti, M.E., Williams, C.S.: Falls, injuries due to falls, and the risk of admission to a nursing home. The New England J Medicine **337**, 1279–1284 (1997)
29. Wang, Y., Witten, I.H.: Induction of model trees for predicting continuous classes. Eur. Conf. Mach. Learn. **9**, 128–137 (1996)
30. Ward, J.A., Lukowicz, P., Gellersen, H.W.: Performance metrics for activity recognition. ACM transactions on intelligent systems and technology (2011)
31. Williams, G., Doughty, K., Cameron, K., Bradley D.A: A smart fall and activity monitor for telecare applications. In: Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (1998)
32. World Health Organization (WHO): Global brief for World Health Day, Good health adds life to years. (2012) http://whqlibdoc.who.int/hq/2012/WHO_DCO_WHD_2012.2_eng.pdf
33. Wu, F., Zhao, H., Zhao, Y., Zhong, H.: Development of a wearable-sensor-based fall detection system. Int. J. Telemed. Appl. (2015)
34. Zhang, S., Wei, Z., Nie, J., Huang, L., Wang, S., Li1, Z.: A review on human activity recognition using vision-based method. J Healthcare Eng (2017)

Application of Convolutional Neural Networks for Fall Detection Using Multiple Cameras

Ricardo Espinosa, Hiram Ponce, Sebastián Gutiérrez, Lourdes Martínez-Villaseñor, Jorge Brieva and Ernesto Moya-Albor

Abstract Currently one of the most important research issue for artificial intelligence and computer vision tasks is the recognition of human falls. Due to the current exponential increase in the use of cameras is it common to use vision-based approach for fall detection and classification systems. On another hand deep learning algorithms have transformed the way that we see vision-based problems. The Convolutional Neural Network (CNN) as deep learning technique offers more reliable and robust solutions on detection and classification problems. Focusing only on a vision-based approach, for this work we used images from a new public multimodal data set for fall detection (UP-Fall Detection dataset) published by our research team. In this chapter we present fall detection system using a 2D CNN analyzing multiple camera information. This method analyzes images in fixed time window frames extracting features using an optical flow method that obtains information of relative motion between two consecutive images. For experimental results, we tested this approach in UP-Fall Detection dataset. Results showed that our proposed multi-vision-based approach detects human falls achieving 95.64% in accuracy with a simple CNN network architecture compared with other state-of-the-art methods.

Keywords Deep learning · Fall detection system · CNN · Multiple cameras

1 Introduction

Human Activity Recognition (HAR) has been popularizing on the research community, particularly detecting human falls on elderly people. Falls can produce injuries, body damages, fractures, etc. Actually, falls are the second leading cause of accidental

R. Espinosa (✉) · S. Gutiérrez
Facultad de Ingeniería, Universidad Panamericana, Josemaría Escrivá de Balaguer 101, 20290 Aguascalientes, Aguascalientes, Mexico
e-mail: respinosa@up.edu.mx

H. Ponce · L. Martínez-Villaseñor · J. Brieva · E. Moya-Albor
Facultad de Ingeniería, Universidad Panamericana, Augusto Rodin 498, 03920 Ciudad de México, Mexico

H. Ponce et al. (eds.), *Challenges and Trends in Multimodal Fall Detection for Healthcare*, Studies in Systems, Decision and Control 273,
https://doi.org/10.1007/978-3-030-38748-8_5

injury and injury-related deaths on people 65 years old and older [1]. "Approximately 28–35% of people aged 65 and over fall each year increasing to 32–42% for those over 70 years of age" [2]. Falls usually cause functional dependencies on elderly people. Many of these related deaths are caused by a long-laying as a long period of time where the victim stays immobile on the floor.

Oneill et al. [3] classify human falls in 3 categories depending on the direction: forward, backward and to-the-side. The most common falls are the forward falls with 38% in men under 65 years old and 62% in men older than 65. In the same way, in women forward falls take place in 62% in women under 65, and 60% in women older than 65.

In 1987 the Kellogg International Working Group [4] on fall prevention on elderly people defined a fall as unintentionally coming to the ground or some lower level and other than as a consequence of sustaining a violent blow, loss of consciousness, sudden onset of paralysis as in stroke or an epileptic seizure. A human fall usually starts with a short free fall period. This causes the acceleration's amplitude to drop significantly under the 1G mark. This represents the time when the fall happens. The fall must stop and it causes acceleration and a peak in the graph. When the amplitude crosses a maximum limit, suggests a fall [5].

It has been proved that the medical consequences of a fall are highly dependant on the response and rescue time. Hypothetically, fall detection systems can improve the response time to medical attention and reduce the medical consequences of a fall.

Thanks to extraordinary advances and research in embedded sensor systems, mobile devices and microelectronics, Internet of things (IoT) systems allows people to interact with technology. This implies access to large amounts of data about people's daily actions to be able to perform fall detection systems and to be able for make possible faster and better assistance to elderly people.

Some of the approaches for fall detection systems are sensors-based, vision-based and multimodal-based strategies. Sensors-based approaches use ambient, smart devices and wearable sensors to provide important information as acceleration, absense / presence, etc. Otherwise, vision-based strategy uses images, as main input, like: 3D reconstructions of an environment, simple 2D RGB video sequences with one or multiple cameras, or depth images from 3D depth sensors. Multimodal-based approaches collect as much information as possible from cameras, microphones, wearable sensors, ambient sensors, smart devices, and many others and combine all this information to improve fall detection and classification in a workable way.

There are two principal approaches for detecting activities and falls, analytical and machine learning methods [6]. Analytical methods can solve fall detection using threshold algorithms. For example, in a fall, a person usually hits the ground or an obstacle. This "impact-shock" causes an intense reverse of the acceleration, mainly on the direction of the trajectory. This change, can be detected by a threshold value. In these methods, the most difficult task is to adapt detections to different types of falls and different people, because thresholds are different by person and/or by type of fall [6]. To attack this problem, there are different strategies like pattern normalization [7] or correlation-based algorithms [7]. Currently, recent investigations report to choose the threshold using optimization algorithms [8].

Otherwise, machine learning methods have been gaining more popularity due to the flexibility of the algorithms to different subject and types of falls [9]. Some of the best known supervised learning techniques used for fall detection systems are: Multi-Layer Perceptron (MLP) [10], Support Vector Machines (SVM) [11], Hidden Markov Models (HMM), decision trees, random forest, k-Nearest Neighbors (KNN) [12], and Convolutional Neural Networks (CNN) [13]; CNN as a deep learning method.

Recently, deep learning methods have changed and improved the way on how to board computer vision problems. Regarding CNN, its main characteristic considers to automatically learn features from training data, making a practical automatic feature extraction for images. CNN has been extensively used on several image processing problems like in [14] in which authors use deep learning to detect accidents using optical flow as a feature extraction method and then testing on real videos. In [15], a single CNN was trained using images directly to classify skin injuries and cancer with AUC (Area under the curve) of 0.96%. Also, CNN has been used in fall detection systems with a sensor-based approach with 92.3% accuracy [16] and with a wearable-based approach with 0.75 AUC.

In vision-based approaches for fall recognition systems, deep learning has been applied as a successful model. For example, Núñes-Marcos et al. [17] implemented a CNN to avoid manual feature engineering , letting the convolutional layers of their system to extract the most important features of the images getting sensitivity and specificity of 94%. A vision-based approach with CNN has also been implemented in [18] in which authors use a 3D CNN with videos of people's kinematics as inputs achieving 100% accuracy evaluated on different data sets.

Lately, our research group released a public multimodal data set for fall detection, called UP-Fall Detection Dataset [19]. The data was collected from different sources of information: wearable sensors, ambient sensors and cameras. Up to now, we have studied this data set with a multimodal approach [19]. However, the different techniques and skills required for building and setting a multimodal all detection system makes it difficult to implement in the real world. Moreover, wearable and ambient sensors are conditioned by the subject and environment making its portability difficult. In that sense, we are interested in creating a vision based fall detection system using its data set and the video recording from the multi-cameras.

Additionally, fall detection systems based on single RGB cameras are often viewpoint-dependant, according to [20]. This raises the need of new data sets when a camera is moved to different viewpoints and, especially, different heights. To deal this issue, different camera viewpoints in a data set collection can help to identify when a given method has viewpoint-independant properties or not. For that end, a fall detection system must be reliable regardless of the position of the subject when falling, respect to the camera.

From the above, this research presents a fall detection system based on a 2D CNN inference method and multiple cameras. As we'll describe later, this approach analyzes images on fixed time window frames extracting features using an optical flow method that obtains information from relative motion between two consecutive images from video recordings acquired from cameras in different viewpoints. For experimentation, we tested this approach in our public UP-Fall Detection data set. The results showed that our proposed multi-vision-based approach detects human falls using a sample CNN network architecture, achieving competitive performance compared with other methods in the state-of-the-art. Also, it is comparable with the performance from a multimodal approach.

Even though CNN has been used in fall detection systems with good performance using a particular data set, Casilari et al. [21] concluded that these systems should be trained and tested with different data sets due to the different number of samples, types of falls and different time series performing any fall. In that sense, the implementation of CNN in multi-cameras vision-based approach, specifically for the UP-Fall Detection data set, might increase the state-of-the-art of fall detection systems.

The main contributions of this work considers: (i) the use of multiple cameras with CNN for fall detection and classification, (ii) the implementation of this approach in the UP-Fall Detection Database, and (iii) the competitive performance comparable with other well-known supervised learning methods. Of what we know, there is just one work [22] that combines CNN with a multiple camera vision-based approach to recognize human falls. In contrast with our proposal, authors in [22] use a voting strategy of the results from independent cameras; while ours use the information from all cameras in the same machine learning model.

The remaining of the chapter is organized as follows. First, we review and analyze different approaches for fall detection systems, focusing on vision-based solutions. Then we present a description of the UP-Fall Detection data set, and we present the proposal in detail. Later on, we explain the experimentation and include the results and discussions from this proposal. Finally we conclude our results.

2 Fall Detection Systems

HAR and fall detection systems are hard tasks, and there are several ways to achiev them due to the many different approaches proposed in literature. For instance, Lara et al. [23] and Noury [6] divide Har taxonomy into three general approaches depending on the source of the information: external, wearable or video sensing. From them, there are case studies related to sensor-based [24], vision-based [25], smartphone-based [26], and multimodal-based [27] approaches to tackle human fall recognition, as described below.

2.1 Sensors-Based Fall Detection Systems

With the increasing technology and accessibility of mobile sensors, fall detection systems have been designed for real-world purposes. Human activity can be tracked, monitored and labeled as data coming from different types of sensors on several locations in the environment and in the human body. An important application of a sensor-based approach is detecting abnormal activities from wearable sensors in determined areas [24]. Then, abnormal activity detection methods can be applied to constantly track each individual's movement to check if the person are out of normal [24]. In [28] using triaxial sensors and SVM as an inference method, the authors achieved 98.33% accuracy. Or in [29] using acceleration and Euler angles, we achieved 100% Accuracy, Sensitivity and Specificity. However some disadvantages of heterogeneous sensor networks come from the fact that human activities usually involves different parts. Moreover, many physiological and bio-mechanical studies have shown that mist of human activities performing day-to-day are inherently multimodal [30]. Thus, different types of sensors are required to gather different data type.

2.2 Wearable-Based Fall Detection Systems

Wearable-based approaches are common solutions for fall detection, taking advantage from wearable technologies due to low cost, live tracking capabilities and small sizes. In [31] a Shimmer device was used for acquisition and transition data. The wearable device was placed on the chest scoring 98.8% accuracy using different machine learning models. In [8], authors used wearable band placed on the wrist scoring 0.95 Specificity and 0.83 Sensitivity using threshold-based peak recognition with SVM; for classification, in which they optimized the best threshold value for different data sets.

In wearables and smartphone devices, energy storage is a problem to solve because they require to be on to be able to track information from the subjects. The lifetime of wearables and smartphone devices is limited due to the capacity of the battery, and constant charging is necessary preventing the constant tracking of the patient's activities [31].

2.3 Smartphone-Based Fall Detection Systems

Nowadays, smartphones have multiple integrated sensors and too much processing capacity that grow over the years. Smartphones can measure user movements offside of a controller in a non-intrusive way. Smartphone-based fall detection systems us smartphone sensors, like gyroscope, tri-axial accelerometer, or altimeter, to achieve them in a long period of time. For example, a case study using this approach can be found in [32], in which authors use a smartphone-based tri-axial accelerometer with statittical time-domain features. Then, they applied main component analysis methods for feature selection, and finally they inferred outputs with MLP scoring 92% accuracy. Another example is the work of Vilarinho [33] who combined smartphone and smartwatch sensors, using threshold-based techniques and pattern recognition algorithms for recognizing falls with 63% accuracy and daily activities with 78% accuracy.

2.4 Multimodal-Based Fall Detection Systems

Data gathering is an important task in fall detection and classification systems, mainly about ambient sensors, wearables, cameras, microphones, RFID tags, among many others that could be used for the recognition task. Using wearables sensors is not able to distinguish a large number of explicit and/or complex human activities, difficulty similar in ambient sensors for context aware. In this regard, multimodal-based approaches can combine more than one source of data to get more information about the environment and the user. These approaches make such fall detection and classification feasible by leveraging selective different modes of sensing in the wide range of sources [27].

Because of multi-modal approaches comprise many different sources of data from the subjects and environments, there are some weaknesses as reported in [34]: (i) many information requires to apply more robust feature extraction and feature selection techniques, as well as taking on account in machine learning approaches for different types of input data, making the fall detection system, computationally expensive and hard task and (ii) multiple sensors with complex placement on the body (and the environment) could cause higher costs, practical deployment difficulties, and obtrusiveness, especially for elderly people.

2.5 Vision-Based Fall Detection Systems.

Traditionally, fall detection systems have been tackled using computer vision and image processing techniques in window frames of images to classify activities. With recent progress, in-depth imaging non-invasive sensors produce high-quality deep images. This information is also analyzed for human tracking, monitoring and user recognition systems [35–37], and also for monitoring and recognizing daily activities of the subjects in indoor environments [38].

The majority of vision-based approaches have been working with simple RGB cameras, web cameras, motion camera systems, or even Kinect [39]. For instance, the usage of Kinect for fall detection has increasd given that it can obtain 3D information, like the human pose or even track limbs [39].

Classical vision-based fall detection and classification strategies consist of five phases [3], as follows: (1) data acquisition from video sequences, (2) feature extraction from images, (3) feature selection and (4) learning and inference. There are multple machine learning techniques used in literature, like SVM [40] or random forest [41]. Zerrouki et al. [38], proposed a fall detection system based on human silhouette shape variation in vision monitoring and SVM to identify postures. Then, they used HMM to classify data into fall and non-fall events. Rougier et al. [42] tracked the person's silhouette along with the video sequences.Using shape analyzing methods through the silhouettes was quantified the shape deformation. Finally, falls were detected from daily activities using Gaussian Mixture Models (GMM).

Vision-based systems can be addressed on two categories: monocular systems and multi-camera systems. In monocular-based fall detection systems, depends on a single camera. Moving a camera to different viewpoints would require collecting new training data for that specific viewpoint and calibrate the camera. However, these systems can fail because of occluding objects between the target and the camera. Zhang et al. [43] proposed a model using multiple Kinect devices to achieve that problem using their won OCCU dataset that was created with occluded and non occluded falls. Kwolek et al. [28] extracted depth maps about the environment and the person's silhouette in combination with 3-axis accelerometers and SVM as a machine learning technique. In terms of multi-camera fall detection systems, Thome and Miguet [44] proposed to use a HMM to distinguish falls from a metric image rectification in each view. Anderson et al. [45] analyzed the states of 3D objects, called the voxel of a person, obtained from two cameras. All these works are able to construct 3D models with multiple cameras in order to reconstruct the environment. This task is particularly hard because the cameras need to be calibrated to compute properly 3D information. It also presents issues on synchronization of video sequences of each camera, making it mode difficult do implement than a monocular-based approach.

So, from the point of view of these systems deployment, 2D multiple camera are a better option, mainly for the low cost and ease of implementation. It is also important to highlight that cameras are already installed in many public places, such as airports, shops, elderly care centers, that can be used for fall detection systems as well.

There are many works using CNN on monocular vision-based fall detection systems with excellent results [16, 18, 46, 47]. Moreover, there are many works using a multi-camera approach with different classical machine learning models or other algorithms [48–52] and there is only one work using multi-camera an do CNN in fall detection systems [22].

2.6 Vision-Based Fall Detection Systems Using CNN

Recent investigations about fall recognition systems have been taking advantage from the deep learning success on recognition and classification tasks using regular images, deep images, infrared images, etc. Deep learning CNN works searching relevant features in images, avoiding the feature engineering task and providing a versatile automatic feature extraction depending on its architecture of convolutional and inference layers [53].

For instance, some or the recent works about fall detection systems reported in literature consider [54] in which authors use a rule-based filters before an input convolutional layer combining the convolutional layer output with optical flow features to choose a better input for the inference phase of its 3D CNN architecture, scoring 92.67% accuracy. In [55], authors use infrared (IR) images and a 3D CNN to find features on three color channels on real-home situations, taking into account a spatio-temporal image information, scoring 85% accuracy on test video sequences.

There are many works using CNN on monocular vision-based fall detection systems with excellent results [16, 18, 46, 47]. Moreover there are several works using multi-camera approaches with different classical machine learning models or other algorithms [48–52] and there is only one work using multi-camera and CNN in fall detection systems [22].

3 Data Set Description

In this work, we used a public dataset called UP-Fall Detection dataset [19]. This data set was collected with information of 17 young healthy subjects with no impariments (9 males and 8 females) ranging from 18–24 years old, the mean height of 1.66 m and mean weight of 66.8 kg performing 11 acivities and 3 trials per activity, six simple human daily activities and five different types of human fall using a multimodal approach, with wearable sensor, ambient sensors, and vision devices. The dataset as well as the feature dataset are publicly available.

The activities and falls stored in this data set are summarized in Table 1. All the data was colleced using 14 devices: 5 Mbientlab MetaSensor wearable sensors collecting raw data from a 3-axis accelerometer, 3-axis gyrscope, and ambient light sensors; 1 electroencephalograph (EEG) NeuroSky MindWave headset was used to measure the raw brainwave signal from its unique EEG channel located at the forehead; as context-aware sensors, we insalled 6 infrared sensors as a grid 0.4 m above the floor of the room, to measure the changes in interruption of the optical devices; and lastly, 2 Microsoft LifeCam Cinema cameras at 1.82 m above the floor, for a lateral view and a frontal view in relation to the subject. All these devices were located as shown in Fig. 1. For more information about the UP-Fall Detection data set, see reference [19].

In this work, we only used information from the two cameras in the data set, taking advantage of the multiple cameras distributions. So, we aim to implement a fall detection and classification system using two cameras, and to compare its performance when only using a monocular-based approach. To this end, CNN will be the classification model.

4 Description of the Proposal

In this work, we adopted the traditional workflow for fall detection systems [23] that consists of the following steps: (i) data collection, (ii) windowing, (iii) feature extraction, and (iv) learning and inference. It is shown in Fig. 2. All this steps were implemented with Python 3.7.3.

Table 1 Activities performed by subjects, adapted from [19]

Activity ID	Description	Duration (s)	Abbreviation
1	Falling forward using hands	10	FH
2	Falling forward using knees	10	FF
3	Falling backwards	10	FB
4	Falling sideward	10	FS
5	Falling sitting in an empty chair	10	FE
6	Walking	60	W
7	Standing	60	S
8	Sitting	60	ST
9	Picking up an object	10	P
10	Jumping	30	J
11	Laying	60	L

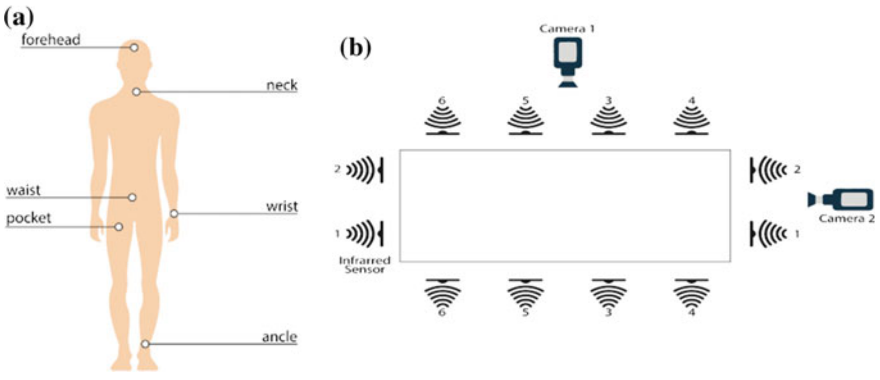


Fig. 1 Distribution of the sensors. **a** Wearable sensors and EEG helmet in the human body. **b** Layout of ambient sensors and multiple cameras. Adapted from [19]

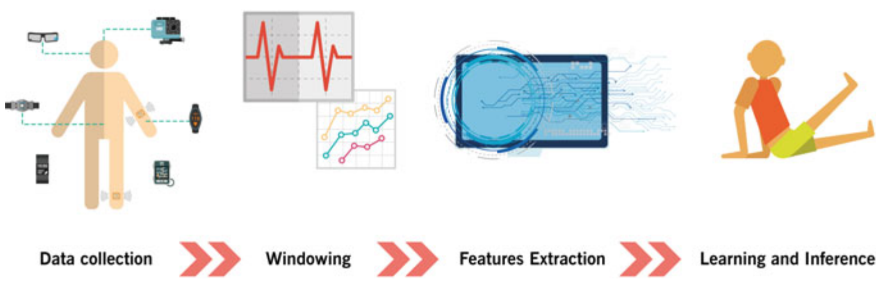


Fig. 2 Traditional workflow for fall detection systems

4.1 Data Collection

One of the most challenging phases in the traditional workflow for fall detection systems and in general, machine learning problems is the data collection task. Nowadays, deep learning techniques require large amounts of data to be trained and tested correctly and other factors as the number of individuals, their physical characteristics, their divers characteristics in terms of gender, age, height, weight, and health conditions [23].

As we explained in the Sect. 3, we use the UP-Fall detection dataset to achieve data collection. To summarize, this data contains information of 17 young subjects performing 11 different activities, 5 falls and 6 activities. For this, we used the information from two RGB cameras in different viewpoints, taking images from the subjects [19].

The images are available in http://sites.google.com/up.edu.mx/har-up/ that contains images from 17 subjects with 3 trials per activity performing 11 activities and different kind of falls. Each package contains multiple images of its respective activity and trial with 2 csv files, U and V matrices that were extracted from rgb images using optical flow algorithm obtaining U (Horizontal apparent movement) and V (Vertical apparent movement) as shown in Fig. 3.

4.2 Windowing

Windowing approaches in fall detection systems are commonly used to segment time series of performed falls. The segmentation is the process of diving sensor signals into smaller data segments. This process has been performed in different ways in the activity recognition field and fall detection systems. Segmentation techniques can be categorized into three groups, namely activity-defined windows, event-defined windows and sliding windows [56].

We adopted a sliding windows windowing approach to capture temporal dependencies between samples. In this case, we split all data into fixed length time windows, for each activity and fall. Our implementation uses one-seconds windows with 0.5

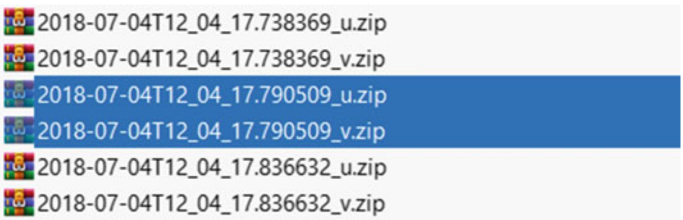


Fig. 3 Downloading data structure

seconds of overlapping. This way we can analyze the fall on each window extracted, simpler than analyzing every capture. The result of this step are multiple 1 second window length series of images to be processed in the next step.

We segmented the data into 1-second windows using a Python Data Analysis Library called pandas to help us compress the data. We created 0.5-second windows, then we added 2 consecutive window to get our 1-second window. To this, we divided its values by the number of images we have on this 1-second window.

Once extracted information from packages described in data collection section we used the same arrays extracted from zip packages *U* and *V* as arrays given to store all the file names for our images. Then we used *d* as our temporal variable to organize these file names into the corresponding window, we called this variable `Master`. This variable contains an array of windows, each of this window contains the files that have to go in a specific window. The implementation code in python 3.7 is shown in Fig. 4.

Then we created a variable to store data in each 0.5 second window called `CSVMaster`. We read each array from `Master` and saved the sum of all the tables. The implementation code in python 3.7 is shown in Fig. 5.

Finally we added each consecutive segment to add to 1 second, dividing the result by the number of images in both segments to record the average in each window. The implementation code in python 3.7 is shown in Fig. 6.

```
for d in U:
    if(Separator > datetime.datetime.strptime(d[:-6],'%Y-%m-%dT%H_%M_%S.%f')
    and Separator2 < datetime.datetime.strptime(d[:-6],'%Y-%m-%dT%H_%M_%S.%f')):
        Temp.append(d)
    else:
        Master.append(Temp)
        Temp = []
        Temp.append(d)
        Separator = Separator + datetime.timedelta(microseconds = 500000)
        Separator2 = Separator2 + datetime.timedelta(microseconds = 500000)
Master.append(Temp)
```

Fig. 4 Organizing data

```
CSVMaster = []
for files in Master:
    a = 0
    for file in files:
        a = a + pandas.read_csv(Fullpath + "\\" + file, header = None)
    CSVMaster.append(a)
```

Fig. 5 Creating 0.5 window images

```
for i in range(len(CSVMaster)-1):
    save = (CSVMaster[i] + CSVMaster[i+1])/(len(Master[i]) + len(Master[i+1]))
    save.to_csv(Savepath + "\\" + Separator2.strftime('%Y-%m-%d %H-%M-%S.%f') +
    "_" + (Separator).strftime('%Y-%m-%d %H-%M-%S.%f') + "_U.csv",header = False, index = False)
    Separator = Separator + datetime.timedelta(microseconds = 500000)
    Separator2 = Separator2 + datetime.timedelta(microseconds = 500000)
```

Fig. 6 Mixing data to create 1-Second windows

4.3 Feature Extraction

Feature extraction is a general method in which you develop a transformation of the input space into a lower dimensional space that preserves most of the relevant information in order to improve data analysis [57]. In fall detection systems, exist multiple techniques depending on the data type to extract information, in vision-based approaches optical flow algorithms provide very important information about the aparent movements in images and has been applied with success in combination of CNN in multiple works as [54] and [46].

For feature extraction, each window frame is pre-processed to get information that could provide enough information to describe the activity. In this study case, the optical flow algorithm [19, 58] was used as visual features extracted from each camera. This algorithm obtains the displacements between two window frames, which allow us to distinguish movements and directions without taking into account the static features in the image. The obtained features are the horizontal and vertical relative movements of the pixels on the images, U and V [58]. The resultant, D, corresponds to the magnitude of the relative movement, as in Eq. 1, where the resultant matrix images are the same size as the original window images. 640×480 in our case.

$$Di, j = \sqrt{U^2 i, j + V^2 i, j} \tag{1}$$

To this phase we have all U and V images collected with 1 second window with 0.5 second overlapping. The next step that we applied in order to extract relevant features from each windowed image was calculate the euclidean distance from U (horizontal features) and V (Vertical features) obtaining one image gray scale or with apparent movement horizontally and vertically in each image as shown in Fig. 7.

Once we had all the resultant images from euclidean distances between U and V matrices in order to reduce the computational effort before training and testing phase we reduce the size of the resultant images using a opencv python extension with *resize* function with 80% respect original resultant images as shown in Fig. 8.

4.4 Learning and Inference

In literature [6], there are many ways to achieve this phase, two of them are machine learning and deep learning algorithms. This step looks to train and test the output from

```
def dist_euclidiana(v1, v2):
    soma = math.pow(v1 - v2, 2)
    return math.sqrt(soma)

image1 = pandas.read_csv(Path+"\\"+file+"\\"+file2, header = None)
image2 = pandas.read_csv(Path+"\\"+file+"\\"+file2.replace('U','V'), header = None)
i1=np.array(image1.values)
i2=np.array(image2.values)
ir=i1
for idx,i in enumerate(i1):
    for idy,j in enumerate(i2):
        ir[idx][idy] = dist_euclidiana(i2[idx][idy],i1[idx][idy])
np.savetxt(Path+"\\"+file+"\\"+file2.replace('U','R'), ir, delimiter=",")
print(Path+"\\"+file+"\\"+file2.replace('U','R'))
```

Fig. 7 Euclidean distance function

```
import cv2

Path = "E:\\DatosFinal"
for file in os.listdir(Path):
    if(os.path.isdir(Path+"\\"+file)):
    src = cv2.imread(Path+"\\"+file, cv2.IMREAD_UNCHANGED)
    scale_percent =80

    width = int(src.shape[1] * scale_percent / 100)
    height = int(src.shape[0] * scale_percent / 100)
    dsize = (width, height)
    output = cv2.resize(src, dsize)
```

Fig. 8 Resizing function

feature engineering to classify the performed fall with inputs from the environment sensors, wearable sensors and in this particular work, from multi-camera vision-based approach.

In deep learning, CNN has revolutionized the way computer vision problems are treated due to the discovery of structure representation in big data sets. This method has improved drastically the state-of-the-art in image processing [53].

But finding a suitable architecture of the CNN is a hard task [53]. Literature has reported multiple types of network architectures, depending on the problem to solve. For example, nowadays, many network structures for image recognition and classification problems have been reported like: AlexNet [59], ClarifaiNet [60], GoogLeNet [61] and VGGNet [62]. All these networks have proved to be efficient in their own problem domains; also, they can be used as pre-trained models so that users can reduce the time to re-train them. However, these complex architectures, might be improved.

In this work, we designed a CNN with three convolutional layers and three 2D max-pooling layers for feature extraction and three fully connected layers for fall detection. Then we fixed the size of all the images to 38×51 pixels, the fixed size constraint comes only from fully connected layers, which exist deeper into the stage of the network [63]. The CNN receives the magnitudes, D, calculated from U and

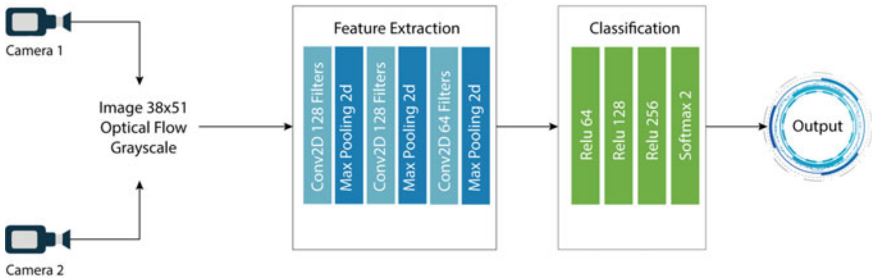


Fig. 9 Our proposal of CNN architecture to multi-camera vision-based fall detection system

V, which were converted to gray-scale images size 38 × 51, representing the optical flow features extracted. Then, these images come to the input layer that comprises of 64 filters of convolution with a kernel size of 3 × 3. The second layer has 128 filters and the same kernel size, and the third, 256. After each convolutional layer, 2D max-pooling layers are employed to synthetize output convolutions. Then, the results are inputted to three fully connected layers, i.e. 64 rectified linear units (ReLU) in the first layer, 128 ReLU in the second and 2D softmax layer with a single output. The 2D softmax layer is employed to perform fall detection classes. Figure 9 explains the representation of the proposed CNN (Table 2).

As described, the UP-Fall Detection data set is integrated from 11 different activities performed by 17 subjects, three trials per activity. In order to train the CNN, we split the data taking trials 1 and 2 for each activity and subject as training data (67%), and trial 3 as testing set (33%). the training data set containing 42,000 gray-scale images of size 38 × 51 with optical flow as pre-processing; while the testing data set containing 21,000 gray-scale images with the same pre-processed optical flow. For training purposes, we trained during 50 epochs, using the Adam optimizer and binary cross-entropy loss function, as defined in Eq. 2 where p is the prediction of the network and t is the ground truth.

$$loss(p, t) = -(t * \log p + (1 - t) * \log 1 - p) \tag{2}$$

5 Experimentation

To analyze our proposal, the experiments were carried out in three branches: (i) experiments to test our CNN model and compare it with classical machine learning methods SVM (Support Vector Machines), RF (Random Forest), MPL (Multi-Layer Perceptron), KNN (K-Nearest Neighbors), (ii) experiments to compare monocular with multi-camera vision-based fall detection system approaches, and (iii) test our proposal not only for detection, but also in classifying activities and falls using the multi-camera vision-based approach

Table 2 Cross-validation for convolutional architecture layers

CNN Architecture	Accuracy (%)	Precision (%)	Sensitivity (%)	Specificity (%)	F1-score (%)
64 64 64	95.40	86.28	83.76	97.54	95.40
64 64 128	95.27	88.26	80.34	98.03	84.11
64 64 256	94.90	83.67	83.57	96.99	83.62
64 128 64	94.62	82.36	83.27	96.71	82.81
64 128 128	94.66	86.49	77.83	97.76	81.94
64 128 256	95.15	85.74	82.35	97.46	84.14
64 256 64	94.32	85.86	76.00	97.69	80.63
64 256 128	94.92	86.45	79.91	97.69	83.05
64 256 256	94.90	91.18	74.48	98.67	81.98
128 64 64	95.17	86.21	82.11	97.58	84.11
128 64 128	94.80	96.02	97.89	78.02	96.95
128 64 256	94.79	97.07	96.74	84.18	96.91
128 128 64	**95.64**	**96.91**	**97.95**	**83.08**	**97.43**
128 128 128	95.44	96.19	98.49	78.87	97.33
128 128 256	95.05	97.88	96.22	88.70	97.04
128 256 64	94.28	96.32	96.92	79.91	96.62
128 256 128	94.51	97.00	96.47	83.82	96.74
128 256 256	95.19	96.84	97.48	82.78	97.16
256 64 64	94.81	96.16	97.76	78.81	96.95
256 64 128	94.26	96.25	96.97	79.54	96.61
256 64 256	94.38	96.34	97.03	80.03	96.68
256 128 64	94.75	96.19	97.64	79.05	96.91
256 128 128	94.72	97.63	96.08	87.36	96.85
256 128 256	94.40	96.66	96.71	81.86	96.68
256 256 64	94.10	96.31	96.71	79.91	96.51
256 256 128	94.57	96.36	97.24	80.09	96.80
256 256 256	94.09	96.19	96.83	79.18	96.51

Bold indiactes the best combination of layers in our CNN architecture cross-validation

In this experiment, we used training and testing data sets with information from two cameras. We decided to use the data from one camera per model, and then, use both cameras at the same time [23]. For windowing, 1-second fixed time lengths with 0.5 seconds of overlapping were used. Images are treated as gray-scale and optical flow implementation as feature extraction. We resized images to 38 × 51, and we used a benchmark between classical machine learning methods (SVM, MLP, RF and KNN) and the CNN depicted in Fig. 9.

These experiments' purpose is to explore and compare the performance between monocular vision-based and multi-camera vision-based fall detection systems, and also to make a benchmark of classical machine learning methods and CNN for fall detection using the latter approach.

To measure the performance of our model, we used accuracy, sensitivity, specificity, precision and F1-score as metrics. As shown in Eqs. (3)–(7) where TP refers to True Positives, TN to True Negatives, FP to False Positives and FN to False Negatives.

$$accuracy = \frac{(TP + TN)}{2TP + TN + FP + FN} \tag{3}$$

$$precision = \frac{TP}{TN + FP} \tag{4}$$

$$sensitivity = \frac{TP)}{TP + FP} \tag{5}$$

$$specificity = \frac{TN)}{TN + FP} \tag{6}$$

$$F1 - score = 2 * \frac{precision * sensitivity}{precision + sensitivity} \tag{7}$$

All these experiments were implemented in Python 3.7.3 using the sklearn3 framework for classical machine learning techniques and keras4 for its GPUs management [64].

5.1 Results and Discussion

The experimental results are described in this section. Then a discussion from the analysis is described bellow.

Fall Detection Using Conventional Machine Learning Models. We conducted an experiment using the optical flow-based features from both cameras at the same time. We trained four conventional machine learning models: SVM, RF, MLP and KNN. Table 3 shows the meta-parameters setting for these models. We build the models using 67% training data and the remaining 33% for testing data. Table 4 summarizes the results using the visual features extracted on 1-second windows length with 0.5 seconds of overlapping.

Conventional machine learning models can't predict human falls with good performance on accuracy, precision, sensitivity, specifity or F1-score as we can see on Table 4. KNN seems to be the best performer based on the F1-score (15.27%). SVM performs the best on acurracy with 32.40%. In the end, these machine learning mod-

Table 3 Parameter settings used for training in the classification models

Classifier	Parameters
SVM	kernel = "radial basis function" kernel coefficient =1 c = 1 shrinking = 1 tolerance = 0.001
RF	minimum samples split = 2 minimum samples leaf = 1 estimators = 2 bootstrap = 1
MLP	activation function = "reLU" hidden layers = 100 penalty parameter = 0.0001 batch size = $min(200, num_samples)$ shuffle = 1 initial learning rate = 0.001 tolerance = 0.0001 exponential decay(first moment) = 0.9 exponential decay(second moment) = 0.999 regularization coefficient = 0.000000001 solver = "stochastic gradient" maximum epochs = 10
KNN	neighbors = 5 leaf size = 30 distance metric = "euclidean"

Table 4 Performance obtained by the classical ML model

Model	Accuracy (%)	Precision (%)	Sensitivity (%)	Specificity (%)	F1-score (%)
SVM	32.40	14.03	14.10	90.03	14.0649
RF	29.30	14.45	14.30	91.26	14.3746
MPL	30.08	9.05	11.03	93.65	9.9423
KNN	**27.30**	**16.32**	**14.35**	**90.96**	**15.2717**

els achieved an average accuracy of 29.77%. From the results, we might assume that conventional machine learning methods using windowing and the sklearn3 library can be found at: extraction, as explained above, are not robust enough. In order to improve this performance, we considered to implement CNN as described later.

Fall Detection Using CNN In this experiment, we trained three different CNN models: (i) a CNN model using visual features from the lateral view, (ii) a CNN model using visual features only from the front view, and (iii) a CNN model using the visual features from both cameras.

Table 5 Performance of the CNN models using lateral view, front view and both views

Data	Method	Accuracy (%)	Precision (%)	Sensitivity (%)	Specificity (%)	F1-score (%)
(Cam1) Lateral view	Proposed CNN	95.24	95.24	97.72	81.58	97.20
(Cam2) Frontal view	Proposed CNN	94.78	96.30	97.57	79.67	96.93
(Cam1 & Cam 2)	Proposed CNN	95.64	96.91	97.95	83.08	97.43
(Cam1 & Cam 2)	VGG-16 CNN	84.44	84.44	100	0	91.56

The results are included in Table 5. We can see that the performance is very similar in any of the combinations. In comparison, the lateral view is slightly better than the frontal one [42]. However, the lateral view shows less specificity (79.67%) than the frontal one (81.58%), which it could lead in misclassification. The combination of both views maintains the output performance of the lateral view. This is important because if occlusion happens in a camera, it will be feasible to detect falls using only one camera, as supported on literature [42]. On the other hand, we made an experiment using the VGG-16 CNN architecture using UP-Fall.

With this, we compared other multi-camera vision-based fall detection systems [36, 40, 46] with out proposed method, considering that were implemented using machine learning methods.

To compare, we used the multi-camera vision-based database, Multicam [65]. This data set consists of 24 performances in which 22 trials have at least one human fall and the remaining 2 containing confused events. Each performance, recorded from 8 different views. The same setup is used for all the videos, with some furniture reallocation [65]. For training purposes of our proposal, we selected two viewpoints (lateral and frontal), from this dataset, splitting training (67%) and testing (33%) sets. Table 6 summarizes the performance results in sensitivity and specificity [40, 46, 66].

As shown in Table 6, it is seen that our proposed method can be competitive in terms of the state-of-the-art, mainly about sensitivity. In addition, our method can handle fall detection using two cameras, in contrast to the eight cameras utilized in the other approaches. Moreover, the network architecture of our proposal (Fig. 9) is very simple compared to other works. For example, Núñez-Marcos in [46] used a VGG-16 architecture modified to receive inputs, authors in [66] occupied PCA to extract features and SVM for classification, and in [40] authors presented a multivariate exponentially weighted moving average (MEWMA) and SVM with 2 steps for classification (see Table 4). In that sense, our system has good performance, taking into consideration its much smaller time for training and the simplicity of its architecture.

Table 6 Comparing between our proposal and other multi-cameras vision-based fall detection systems reported in the state-of-the-art, using the Multicam data set

Proposal	Method	Sensitivity (%)	Specificity (%)	Cameras (%)
Wang et al. [66]	SVM	89.20	90.30	8
Wang et al. [40]	SVM	93.70	92.00	8
Núñez et al. [46]	VGG-16 CNN	99.00	96.00	8
Ours (Combined)	CNN	97.95	83.08	2

Table 7 Comparing between our proposal and other multi-cameras vision-based fall detection systems reported in the state-of-the-art, using the Multicam data set

Data	Accuracy (%)	Precision (%)	Sensitivity (%)	Specificity (%)	F1-Score (%)
Ours	82.26	74.25	71.67	77.48	72.94
Martínez-Villaseñor, et al. [19]	95.00	77.70	69.90	99.50	72.80

Daily Activities and Fall Classification Using CNN Lastly, we conducted an experiment for daily activities and fall classification using our proposal. In this case, everything recorded in the UP-Fall Detection data set was taken into account to convert the CNN into a multi-class classifier, as in Table 1.

We applied our proposal using both cameras, and the results, unlike the performance obtained in [19] using the same data set, are represented in Table 7. As shown, our proposal is slightly worse than the multimodal-based approach results presented by Martínez-Villaseñor et al. [19]. This was expected since a multimodal approach (wearable sensors, EEG helmet and camera) is better than a single modality like ours. It is also important to notice that the F1-score in both approaches are similar, 72.94% for our proposal, and 72.80% for the multimodal approach. From the results presented in Table 5, the performance obtained by our proposal can be considered competitive (similar F1-score), its easier to implement (due to the number of sensors) and less intrusive (wearable sensors), in comparison with the multimodal-based approach in [19].

6 Discussion

The proposed multi-camera vision-based all detection and classification systems offers a comparable solution to the state-of-the-art methods. These results support the evidence about: the predictable power of our proposed fall detection system using two viewpoints (97.43% of the F1-score), the out-performance of conventional machine learning methods (SVM, RF, MLP and KNN) using optical flow based features, the usage of less cameras, with acceptable performance, than other reported in the state-of-the-art (97% sensitivity and 80% specificity), and a similar performance (70.8°% of F1-score) comparable to a multimodal approach (72.80% of F1-score).

As seen previously, the advantages of our proposal can be pointed as follows. The multi-camera approach offers robust solutions recognizing falls although when a occlusion happens in a viewpoint, as long as the camera focuses on the subject. This can be observed in our proposal in Table 5 that reports similar results when using one camera or both. Our proposal offers a simple CNN architecture (Fig. 3) and low computational cost of implementation. Due to the vision-based nature of our approach, an important point to discuss, is the invasion of privacy by the constant video surveillance. We take off the relevant information about the fall in the images using the optical flow calculated from the video sequence, therefore the privacy of the person is not affected because this fall data, do not contain any personal information.

It is important to consider some limitations of our proposal while using it. A vision-based approach is subjected to the quality of the image, the position of the camera, and the presence of the object. In addition, privacy issues should be addressed before the implementation: unless this is a limitation, the original images taken from the cameras, shouldn't be stored, after all they only have to be used for extracting the optical flow features. However, it remains as an important drawback since cameras are always taking videos of the subjects. In addition, computational complexity in terms of memory and time processing is important. This in fact hinders a real-time fall detection system to be scalable [23].

From the samples from the UP-Fall dataset, 42,958 training sampled arranged in 1-second windows were analysed and 21,038 testing samples also in 1-second windows were employed in our experiments. The results were competitive respect if the state-of-the-art in both detection (Tables 5 and 6) and classification (Table 7) tasks. It is important to remember the age of the subjects who performed falls and activities to build the UP-Fall Detection dataset used in this work. This dataset was made with information of 17 young healthy subjects without any impairment (9 males and 8 females) from 18–24 years old. Nevertheless in [67], it demonstrates that using a dataset built just by young people does not have significant differences like testing in elderly people. With this in mind, we consider that out approach can be applied on real situations considered as future work.

To this end, the results demonstrated that our proposal is competitive compared to the state-of-the-art in multi-camera vision-based approaches for detection systems, and also it is competitive as a fall classification model (Table 6), even contrasting to a multimodal-based approach as reported in [19].

7 Conclusions

In this paper, we presented a multi-camera vision-based fall detection and classification system taking advantage of CNN. In addition, we combined the CNN models with visual features extracted from sequences of images using the optical flow estimation. In this work, we used the UP-Fall Detection data set as a case study. We conducted different experiments for: benchmarking our proposal with conventional machine learning models, analyzing the performance of our proposal in single and multi-cameras vision-based approaches, and extending our model for fall classification as well.

From the experimental results, we concluded that our proposed multi-camera vision-based fall detection and classification system outperforms conventional machine learning methods, saves computation due to the simple CNN architecture, and it is competitive with the state-of-the-art and multimodal-based approaches.

Lastly, future works considers to implement this approach in a real-world ambient assisted living system, and to analyze and propose improvements to issues on privacy, pervasiveness, changes in environmental conditions and oclusion. In addition, we will consider to test our system in a real situation.

8 Data Availability

If you require to see the codes of this research, please contact the corresponding author.

Conflict of interest The authors declare that there are no conflicts of interest regarding the publication of this article.

Funding Statement The authors declare that this work was performed as part of their employment in Universidad Panamericana (Mexico).

References

1. Department of Health and Human Services: Fatalities and injuries from falls among older adults - United States, 1993–2003 and 2001–2005. pp. 1221–1224, November 2006. Morbidity and Mortality Weekly Report
2. Schneider, M.: Introduction to Public Health. Jones and Bartlett, Sudbury, MA (2011)
3. Oneill, T.W., Varlow, J., Silman, A.J., Reeve, J., Reid, D.M., Todd, C., Woolf, A.D.: Age and sex influences on fall characteristics. Ann. Rheum. Dis. **53**(11), 773–775 (1994). https://doi.org/10.1136/ard.53.11.773
4. Lord, S.R., Sherrington, C., Menz, H.B., Close, J.C. (n.d.): Strategies for prevention. Falls in Older People, 173–176. https://doi.org/10.1017/cbo9780511722233.011
5. Bourke, A., Lyons, G.: A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. Med. Eng. Phys. **30**(1), 84–90 (2008). https://doi.org/10.1016/j.medengphy.2006.12.001
6. Noury, N., Fleury, A., Rumeau, P., Bourke, A., Laighin, G. O., Rialle, V., Lundy, J. (2007). Fall detection - Principles and Methods. 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. https://doi.org/10.1109/iembs.2007.4352627
7. Wu, Y., Su, Y., Hu, Y., Yu, N., Feng, R.: A multi-sensor fall detection system based on multivariate statistical process analysis. J. Med. Biol. Eng. **39**(3), 336–351 (2019). https://doi.org/10.1007/s40846-018-0404-z

8. Khojasteh, S., Villar, J., Chira, C., González, V., Cal, E.D.: Improving fall detection using an on-wrist wearable accelerometer. Sensors **18**(5), 1350 (2018). https://doi.org/10.3390/s18051350
9. Mubashir, M., Shao, L., Seed, L.: A survey on fall detection: principles and approaches. Neurocomputing **100**, 144–152 (2013). https://doi.org/10.1016/j.neucom.2011.09.037
10. Jalal, A., Kamal, S., Kim, D.: A depth video sensor-based life-logging human activity recognition system for elderly care in smart indoor environments. Sensors **14**(7), 11735–11759 (2014). https://doi.org/10.3390/s140711735
11. Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J.L.: Training computationally efficient smartphone-based human activity recognition models. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8131 LNCS, pp. 426–433 (2013)
12. Jatoba, L.C., Grossmann, U., Kunze, C., Ottenbacher, J., Stork, W.: Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity. In: 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 5250–5253 (2008)
13. Münzner, S., Schmidt, P., Reiss, A., Hanselmann, M., Stiefelhagen, R., Dürucheb, R.: CNN-based sensor fusion techniques for multimodal human activity recognition. In: Proceedings of the 2017 ACM International Symposium on Wearable Computers - ISWC 17. (2017). https://doi.org/10.1145/3123021.3123046
14. Bortnikov, M., Khan, A., Khattak, A.M., Ahmad, M.: Accident Recognition via 3D CNNs for Automated Traffic Monitoring in Smart Cities. Adv. Intell. Syst. Comput. Adv. Comput. Vis. **256–264** (2019). https://doi.org/10.1007/978-3-030-17798-0_22
15. Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M., Thrun, S.: Dermatologist-level classification of skin cancer with deep neural networks. Nature **542**(7639), 115–118 (2017). https://doi.org/10.1038/nature21056
16. Fakhrulddin, A.H., Fei, X., Li, H.: Convolutional neural networks (cnn) based human fall detection on body sensor networks (bsn) sensor data. In 2017 4th ICSAI (Nov 2017)
17. Núñez-Marcos, A., Azkune, G., Arganda-Carreras, I.: Vision-based fall detection with convolutional neural networks. Wirel. Commun. Mob. Comput. **2017**, (2017). https://doi.org/10.1155/2017/9474806
18. Lu, N., Wu, Y., Feng, L., Song, J.: Deep learning for fall detection: three-dimensional CNN combined with LSTM on video kinematic data. IEEE J. Biomed. Health Inform. **23**(1), 314–323 (2019). https://doi.org/10.1109/jbhi.2018.2808281
19. Martínez-Villaseñor, L., Ponce, H., Brieva, J., Moya-Albor, E., Núñez-Martínez, J., Peñafort-Asturiano, C.: UP-fall detection dataset: a multimodal approach. Sensors **19**(9), 1988 (2019). https://doi.org/10.3390/s19091988
20. Zhang, Z., Conly, C., Athitsos, V.: A survey on vision-based fall detection. Proceedings of the 8th ACM international conference on Pervasive technologies related to assistive environments. ACM (2015). http://dx.doi.org/10.1145/2769493.2769540
21. Casilari, E., Santoyo-Ramón, J., Cano-García, J.: Analysis of public datasets for wearable fall detection systems. Sensors **17**(7), 1513 (2017). https://doi.org/10.3390/s17071513
22. Kong, Y., Huang, J., Huang, S., Wei, Z., Wang, S.: Learning spatiotemporal representations for human fall detection in surveillance video. J. Vis. Commun. Image Represent. **59**, 215–230 (2019). https://doi.org/10.1016/j.jvcir.2019.01.024
23. Lara, O.D., Labrador, M.A.: A survey on human activity recognition using wearable sensors. IEEE Commun. Surv. Tutor. **15**, 1192–1209 (2013)
24. Yin, J., Yang, Q., Pan, J.: Sensor-based abnormal human-activity detection. IEEE Trans. Knowl. Data Eng. **20**(8), 1082–1090 (2008). https://doi.org/10.1109/tkde.2007.1042
25. Xu, X., Tang, J., Zhang, X., Liu, X., Zhang, H., Qiu, Y.: Exploring techniques for vision based human activity recognition: methods, systems, and evaluation. Sensors **13**(2), 1635–1650 (2013). https://doi.org/10.3390/s130201635
26. Dungkaew, T., Suksawatchon, J., Suksawatchon, U.: Impersonal smartphone-based activity recognition using the accelerometer sensory data. In: 2017 2nd International Conference on Information Technology (INCIT) (2017). https://doi.org/10.1109/incit.2017.8257856

27. Bharti, P.: Complex activity recognition with multi-modal multi-positional body sensing. J. Biom. Biostat. **08**(05) (2017). https://doi.org/10.4172/2155-6180-c1-005
28. Kwolek, B., Kepski, M.: Human fall detection on embedded platform using depth maps and wireless accelerometer. Comput. Methods Programs Biomed. **117**(3), 489–501 (2014)
29. Mao, A., Ma, X., He, Y., & Luo, J.: Highly portable, sensor-based system for human fall monitoring. Sensors (Switzerland), **17**(9). https://doi.org/10.3390/s17092096 (2017)
30. Chetty, G., White, M., Singh, M., Mishra, A.: Multimodal activity recognition based on automatic feature discovery. in: 2014 International Conference on Computing for Sustainable Global Development (INDIACom) (2014). https://doi.org/10.1109/indiacom.2014.6828039
31. Kerdjidj, O., Ramzan, N., Ghanem, K., Amira, A., Chouireb, F.: Fall detection and human activity classification using wearable sensors and compressed sensing. J. Ambient. Intell. Humanized Comput. (2019). https://doi.org/10.1007/s12652-019-01214-4
32. Torres-Huitzil, C., Nuno-Maganda, M.: Robust smartphone-based human activity recognition using a tri-axial accelerometer. In: 2015 IEEE 6th Latin American Symposium on Circuits & Systems (LASCAS) (2015). https://doi.org/10.1109/lascas.2015.7250435
33. Vilarinho, T., Farshchian, B., Bajer, D.G., Dahl, O.H., Egge, I., Hegdal, S.S., Lønes, A., Slettevold, J.N., Weggersen, S.M.: A combined smartphone and smartwatch fall detection system. In: Proceedings of the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, pp. 1443–1448 Liverpool, UK, 26–28 October 2015
34. Koshmak, G., Loutfi, A., Linden, M.: Challenges and issues in multisensor fusion approach for fall detection: review paper. J. Sens. **2016**, 1–12 (2016). https://doi.org/10.1155/2016/6931789
35. Turaga, P., Chellappa, R., Subrahmanian, V.S., Udrea, O.: Machine recognition of human activities: a survey. IEEE Trans. Circuits Syst. Video Technol. **18**, 1473–1488 (2008)
36. Raty, T.D.: Survey on contemporary remote surveillance systems for public safety. IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. **40**, 493–515 (2010)
37. Albanese, M., Chellappa, R., Moscato, V., Picariello, A., Subrahmanian, V.S., Turaga, P., Udrea, O.: A constrained probabilistic petri net framework for human activity detection in video. IEEE Trans. Multimed. **10**, 1429–1443 (2008)
38. Zerrouki, N., Houacine, A.: Combined curvelets and hidden Markov models for human fall detection. Multimed. Tools Appl. **77**(5), 6405–6424 (2017). https://doi.org/10.1007/s11042-017-4549-5
39. Auvinet, E., Multon, F., Saint-Arnaud, A., Rousseau, J., Meunier, J.: Fall detection with multiple cameras: an occlusion resistant method based on 3-D silhouette vertical distribution. IEEE Trans. Inf. Technol. Biomed. **15**(2), 290–300 (2011)
40. Charfi, I., Miteran, J., Dubois, J., Atri, M., Tourki, R.: Definition and performance evaluation of a robust SVM based fall detection solution. SITIS **12**, 218–224 (2012)
41. Kozina, S., Gjoreski, H., Gams, M., Luštrek, M.: Efficient activity recognition and fall detection using accelerometers. In: Communications in Computer and Information Science Evaluating AAL Systems Through Competitive Benchmarking pp. 13–23 (2013). https://doi.org/10.1007/978-3-642-41043-7_2
42. Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J.: Robust video surveillance for fall detection based on human shape deformation. IEEE Trans. Circuits Syst. Video Technol. **21**(5), 611–622 (2011). https://doi.org/10.1109/tcsvt.2011.2129370
43. Zhang, Z., Conly, C., & Athitsos, V. (2014). Evaluating depth-based computer vision methods for fall detection under occlusions . pp. 196–207. https://doi.org/10.1007/978-3-319-14364-4_19
44. Thome, N., Miguet, S., Ambellouis, S.: A real-time, multiview fall detection system: a LHMM-based approach. IEEE Trans. Circuits Syst. Video Technol. **18**(11), 1522–1532 (2008). https://doi.org/10.1109/tcsvt.2008.2005606
45. Anderson, D., Luke, R.H., Keller, J.M., Skubic, M., Rantz, M., Aud, M.: Linguistic summarization of video for fall detection using voxel person and fuzzy logic. Comput. Vis. Image Underst. **113**(1), 80–89 (2009). https://doi.org/10.1016/j.cviu.2008.07.006

46. Wang, K., Cao, G., Meng, D. Chen, W., Cao, W.: Automatic fall detection of human in video using combination of features. In: Proceedings of the 2016 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2016, pp. 1228–1233, China (December 2016)
47. Nait Aicha, A., Englebienne, G., van Schooten, K.S., Pijnappels, M., Kröse, B.: Deep learning to predict falls in older adults based on daily-life trunk accelerometry. Sensors (Basel, Switzerland) **18**(5), 1–14 (2018). https://doi.org/10.3390/s18051654
48. Shieh, W., Huang, J.: Falling-incident detection and throughput enhancement in a multi-camera video-surveillance system. Med. Eng. Phys. **34**(7), 954–963 (2012). https://doi.org/10.1016/j.medengphy.2011.10.016
49. Mousse, M.A., Motamed, C., Ezin, E.C.: Percentage of human-occupied areas for fall detection from two views. Vis. Comput. **33**(12), 1529–1540 (2016). https://doi.org/10.1007/s00371-016-1296-y
50. Zhang, S., Li, Z., Wei, Z., Wang, S.: An automatic human fall detection approach using RGBD cameras. in: 2016 5th International Conference on Computer Science and Network Technology (ICCSNT) (2016).https://doi.org/10.1109/iccsnt.2016.8070265
51. Hekmat, M., Mousavi, Z., Aghajan, H.: Multi-view Feature fusion for activity classification. In: Proceedings of the 10th International Conference on Distributed Smart Camera - ICDSC 16 (2016). https://doi.org/10.1145/2967413.2967434
52. Su, S., Wu, S., Chen, S., Duh, D., Li, S.: Multi-view fall detection based on spatio-temporal interest points. Multimed. Tools Appl. **75**(14), 8469–8492 (2015). https://doi.org/10.1007/s11042-015-2766-3
53. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
54. Hsieh, Y.-Z., Jeng, Y.-L.: Development of home intelligent fall detection IoT system based on feedback optical flow convolutional neural network. IEEE Access **6**, 6048–6057 (2018). https://doi.org/10.1109/access.2017.2771389
55. Akula, N.V.A., Shah, A.K., Ghosh, R.: A spatio-temporal deep learning approach for human action recognition in infrared videos. Opt. Photonics Inf. Process. XII (2018). https://doi.org/10.1117/12.2502993
56. Banos, O., Galvez, J.-M., Damas, M., Pomares, H., Rojas, I.: Window size impact in human activity recognition. Sensors **14**(4), 6474–6499 (2014). https://doi.org/10.3390/s140406474
57. Khalid, S., Khalil, T., Nasreen, S.: A survey of feature selection and feature extraction techniques in machine learning. In: Proceedings of the Science and Information Conference (SAI), London, UK, 27–29 August 2014
58. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artif. Intell. **17**(1–3), 185–203 (1981)
59. Krizhevsky, A., Sutskever, I. Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS ’12), pp. 1097–1105, Lake Tahoe, Nev, USA (December 2012)
60. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: ECCV, pp. 818–833 (2014)
61. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556 (2014). 1, 2, 3
62. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. CoRR, abs/1409.4842 (2014) 1, 2
63. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: Proceeding 13th European Conference Computer Vision, pp. 346–361 (2014)
64. Francois, C., et al.: Keras (2015) https://github.com/fchollet/keras
65. Auvinet, E., Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J.: Multiple cameras fall dataset. DIRO-Université de Montréal, Tech. Rep. 1350 (2010)
66. Wang, S., Chen, L., Zhou, Z., Sun, X., Dong, J.: Human fall detection in surveillance video based on PCANet. Multimed. Tools Appl. **75**(19), 11603–11613 (2015). https://doi.org/10.1007/s11042-015-2698-y
67. Sucerquia, A., López, J.D., Vargas-Bonilla, F.: Real-Life/Real-Time Elderly Fall Detection with a Triaxial Accelerometer (2018). https://doi.org/10.20944/preprints201711.0087.v3

Approaching Fall Classification Using the UP-Fall Detection Dataset: Analysis and Results from an International Competition

Hiram Ponce and Lourdes Martínez-Villaseñor

Abstract This chapter presents the results of the Challenge UP – Multimodal Fall Detection competition that was held during the 2019 International Joint Conference on Neural Networks (IJCNN 2019). This competition lies on the fall classification problem, and it aims to classify eleven human activities (i.e. five types of falls and six simple daily activities) using the joint information from different wearables, ambient sensors and video recordings, stored in a given dataset. After five months of competition, three winners and one honorific mention were awarded during the conference event. The machine learning model from the first place scored 82.47% in F_1-score, outperforming the baseline of 70.44%. After analyzing the implementations from the participants, we summarized the insights and trends of fall classification.

Keywords Ambient assisted living · Machine learning · Competition · Human fall detection · Abnormal behavioral analysis

1 Introduction

Falls are frequent especially among old people and it is a major health problem according to World Health Organization [2]. Fall detectors can alleviate this problem and can reduce the time in which a person who suffered a fall receives assistance. Recently, there has been an increase in fall detection system development based mainly in sensor and/or context approaches. An important challenge reported in literature [3] is the lack of publicly available datasets that enable comparison between techniques. In that sense, we provide this dataset in the benefit of researchers in the

H. Ponce (✉) · L. Martínez-Villaseñor (✉)
Facultad de Ingeniería, Universidad Panamericana,
Augusto Rodin 498, 03920 Ciudad de México, Mexico
e-mail: hponce@up.edu.mx

L. Martínez-Villaseñor
e-mail: lmartine@up.edu.mx

H. Ponce et al. (eds.), *Challenges and Trends in Multimodal Fall Detection for Healthcare*, Studies in Systems, Decision and Control 273,
https://doi.org/10.1007/978-3-030-38748-8_6

fields of wearable computing, ambient intelligence, and vision. In addition, new machine learning algorithms can be proven with this dataset.

In this competition, participants can to do experiments considering different combination of multimodal sensors in order to determine the best combination of sensors with the aim of improving the reliability and precision of fall detection systems. It is also important for the human activity recognition and machine learning research communities to be able to fairly compare their fall detection solutions.

This competition can be interesting in particular to the growing research community of human activity recognition and fall detection. Moreover, it is also attractive to any person interested in solving signal recognition, vision, and machine learning challenging problems given that the multimodal dataset provided opens many experimental possibilities.

2 Description of the Competition

The *Challenge UP – Multimodal Fall Detection* competition, or simply the competition, was co-located during the 2019 International Joint Conference on Neural Networks (IJCNN 2019). The awarding ceremony of the competition was held on July 15th, 2019 in Budapest, Hungary. However, it was opened from December 3rd, 2018 to April 26th, 2019. The details about this competition are described following.

2.1 Aims and Scope

The competition aimed to classify eleven human activities (i.e. 5 types of falls and 6 simple daily activities) using the joint information from different wearables, ambient sensors and video recordings, stored in a given dataset. This classification was restricted to be done by any, possibly hybrid, machine learning models.

To do so, the competition was scheduled in several steps, mainly for training the model with labeled data and for testing the model with unlabeled data. For convenience, participants were able to use as much information as they wanted. In that sense, the competition dealt with different engineering and computational skills from the participants, through the sensor to image signal processing, the fusion of them, and the abilities to design and deploy different intelligent systems to reach the goal.

2.2 Data

For this competition, we used the UP-Fall Detection dataset [1]. This is a public and large dataset mainly for fall detection and classification that includes 12 activities

Table 1 Statistics of the subjects, adopted from [1]

Subject ID	Age	Height (m)	Weight (kg)	Gender
1	18	1.70	99	Male
2	20	1.70	58	Male
3	19	1.57	54	Female
4	20	1.62	71	Female
5	21	1.71	69	Male
6	22	1.62	68	Male
7	24	1.74	70	Male
8	23	1.75	88	Male
9	23	1.68	70	Female
10	19	1.69	63	Male
11	20	1.65	73	Female
12	19	1.60	53	Female
13	20	1.64	55	Male
14	19	1.70	73	Female
15	21	1.57	56	Female
16	20	1.70	62	Male
17	20	1.66	54	Female

and three trials per activity. Subjects performed 6 simple human daily activities as well as 5 different types of human falls. These data were collected over 17 subjects (see Table 1) using a multimodal approach, i.e. wearable sensors, ambient sensors and vision devices. The consolidated dataset (812 GB), as well as, the feature dataset (171 GB) is publicly available in: http://sites.google.com/up.edu.mx/har-up/. At the time of the competition, the dataset remained private and until April 27th, 2019.

The data was collected over a period of four weeks, in the Faculty of Engineering, Universidad Panamericana in Mexico City, Mexico. During data collection, 17 subjects (9 males and 8 females) of 18–24 years old (i.e. mean height of 1.66 m and mean weight of 66.8 kg), were invited to perform 11 different activities, as shown in Table 2. Falls and daily activities are not overlapped. So, each trial contains information of one of these activities. All the sequences of data was labeled manually. In addition, an *unknown/other* activity was labeled for other unrecognizable activities different from the previous ones [1].

This dataset comprises five Mbientlab MetaSensor wearable sensors collecting raw data from the 3-axis accelerometer, the 3-axis gyroscope and the ambient light value. These wearables were placed in the left wrist, under the neck, at right pocket of the pants, at the middle of waist (in the belt), and in the left ankle. Also, one electroencephalograph (EEG) NeuroSky MindWave helmet was included to measure the raw brainwave signal from one EEG channel sensor located at the forehead. For ambient sensors, the dataset retrieved information from six infrared sensors placed, as a grid, 0.40 m above the floor of the room, to measure the changes in the interruption

Table 2 Types of activities and falls in the dataset

Type	Description	Activity ID
Fall	Forward using hands	1
	Forward using knees	2
	Backward	3
	Sideward	4
	Attempting to sit in an empty chair	5
Daily activity	Walking	6
	Standing	7
	Sitting	8
	Picking up an object	9
	Jumping	10
	Laying	11
Other	Unknown	20

of these devices. Lastly, two Microsoft LifeCam Cinema cameras were located at 1.82 m above the floor, one for lateral view and the other for frontal view, related to the motion of the activities. Table 3 summarizes all the sensors installed for data collection. The dataset was down-sampled to 18 Hz for data synchronization and coherence purposes [1]. Lastly, Fig. 1 shows the placements of wearables, ambient sensors and cameras while collecting the dataset [1]. For further details about the UP-Fall Detection dataset, see [1].

2.2.1 Training Data

For the training data, we exposed the raw dataset from 9 subjects with IDs: 1, 3, 4, 7, 10–14; with all three trials per activity. These data also contained all class labels (activity IDs). The training data set represented 70% of all data considered for this competition. No missing values were presented in the training set.

2.2.2 Testing Data

For the testing data, we exposed the raw dataset from 3 subjects with IDs: 15–17; with all three trials per activity. In this case, data did not contained the class labels. This obeys to the goal of the competition, and the labels of this portion of data remained privately for the participants. In the evaluation step, these labels were used for evaluating the performance of the classification models developed by the participants. No missing values were presented in the testing set.

Table 3 List of devices for measurements, adopted from [1]

Device ID	Device name	Channel name	Units	Signal ID
1	Wearable ankle	X-axis accelerometer	g	1
		Y-axis accelerometer	g	2
		Z-axis accelerometer	g	3
		Roll gyroscope	deg/s	4
		Pitch gyroscope	deg/s	5
		Yaw gyroscope	deg/s	6
		Luminosity	Lux	7
2	Wearable pocket	X-axis accelerometer	g	8
		Y-axis accelerometer	g	9
		Z-axis accelerometer	g	10
		Roll gyroscope	deg/s	11
		Pitch gyroscope	deg/s	12
		Yaw gyroscope	deg/s	13
		Luminosity	Lux	14
3	Wearable waist	X-axis accelerometer	g	15
		Y-axis accelerometer	g	16
		Z-axis accelerometer	g	17
		Roll gyroscope	deg/s	18
		Pitch gyroscope	deg/s	19
		Yaw gyroscope	deg/s	20
		Luminosity	Lux	21
4	Wearable neck	X-axis accelerometer	g	22
		Y-axis accelerometer	g	23
		Z-axis accelerometer	g	24
		Roll gyroscope	deg/s	25
		Pitch gyroscope	deg/s	26
		Yaw gyroscope	deg/s	27
		Luminosity	Lux	28
5	Wearable wrist	X-axis accelerometer	g	29
		Y-axis accelerometer	g	30
		Z-axis accelerometer	g	31
		Roll gyroscope	deg/s	32
		Pitch gyroscope	deg/s	33
		Yaw gyroscope	deg/s	34
		Luminosity	Lux	35
6	EEG headset	Raw brainwave signal	μV	36
7	Infrared 1	No interruption	False(0)/true(1)	37
8	Infrared 2	No interruption	False(0)/true(1)	38
9	Infrared 3	No interruption	False(0)/true(1)	39
10	Infrared 4	No interruption	False(0)/true(1)	40
11	Infrared 5	No interruption	False(0)/true(1)	41
12	Infrared 6	No interruption	False(0)/true(1)	42
13	Camera 1	Lateral view	640×480 px	43
14	Camera 2	Frontal view	640×480 px	44

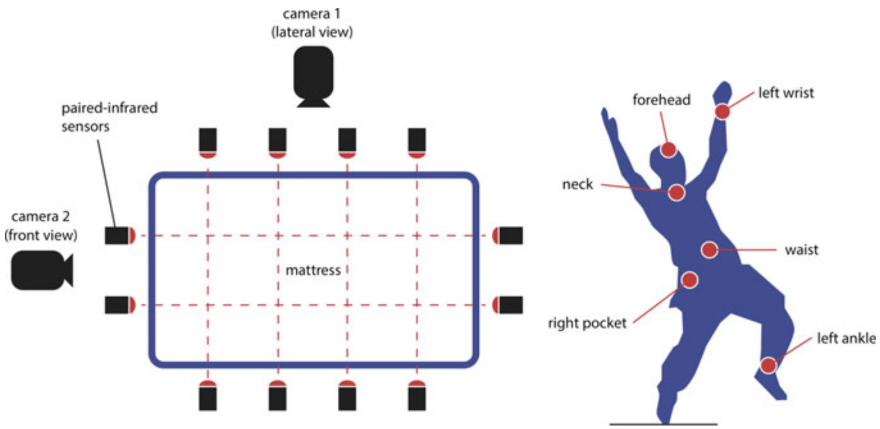


Fig. 1 Layout of the sensors and cameras used in the UP-Fall detection dataset, adopted from [1]

2.3 Classification Task

The main task of the competition is to classify the falls and activities of 3 subjects (testing data set). This is a challenging task since there are diverse of subjects (see Table 1) and they performed activities in different ways. Moreover, the best combination of sensors, feature selection and feature extraction procedures is a challenging task in human activity recognition.

2.4 Metrics and Evaluation

The F_1-score metric was used in the evaluation of the competition. F_1-score considers the average $precision_\mu$ and average $recall_\mu$ of the test as shown in (1), where average $precision_\mu$ computes in average, of all activities and falls, of the number of true positives over the sum of true and false positives; and average $recall_\mu$ computes in average, of all activities and falls, of the number of true positives over the sum of true positives and false negatives. The greater and close to 1, the better the metric.

$$F_1 score = 2 \times \frac{precision_\mu \times recall_\mu}{precision_\mu + recall_\mu} \tag{1}$$

For evaluation, we asked the participants to send the class estimations of the 3 subjects of the testing set. However, these estimations are done in 1-second time window frames. In that sense, the estimated classes were calculated as the most frequent class in 1-second. Similarly, the labels that we retained were also condensed in the most frequent class per 1-second windows without overlapping.

2.5 *Competition Policies*

The following conditions of participation were required during the competition. These policies applied for winning the competition, and the event was divided into several steps through five months of competition; as described below. Participation required complying with the rules of the challenge, published in the official website of the competition (https://sites.google.com/up.edu.mx/challenge-up-2019/).

2.5.1 Conditions of Participation

Prize eligibility was restricted by US government export regulations and the host country laws (Budapest, Hungary). The organizers, sponsors, their students, close family members (parents, sibling, spouse or children) and household members, as well as any person having had access to the truth values or to any information about the data or the challenge design giving him (or her) an unfair advantage, were excluded from participation. However, a disqualified person might submit one or several entries in the challenge and request to have them evaluated, provided that they notify the organizers of their conflict of interest. If a disqualified person submitted an entry, this entry was not be part of the final ranking and did not qualify for prizes.

The participants were aware that organizers reserve the right to evaluate for scientific purposes any entry made in the challenge, whether or not it qualifies for prizes. For participation, the participants registered through the Registration Form displayed in the official website. Teams or solo participants were allowed for entering to the competition.

2.5.2 Awards

The three top ranking participants qualified for awards (travel award, prize and award certificate). To compete for awards, the participants were asked for sending a short paper briefly describing their methods and the codes used for getting the results. There was no other publication requirement. However, this edited book intended to publish the main results of the competition, from the point of view of the participants and the organizers.

2.5.3 Timeline

The competition opened from December 3rd, 2018 until April 26th, 2019. During the five months period, the competition was divided into several steps as shown in Fig. 2. These dates comprised the registration opening (December 3rd, 2018); the training set release (January 14th, 2019) for analyzing and training models by participants; the testing set release (March 25th, 2019) for testing the trained models;

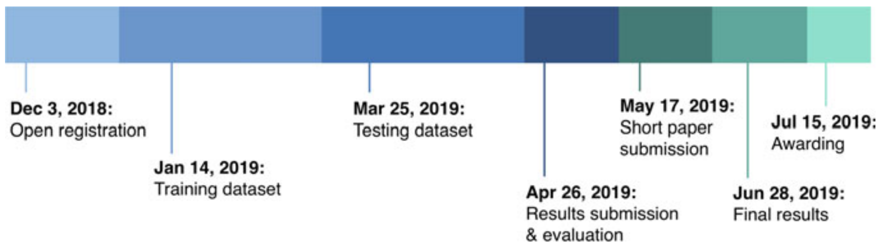


Fig. 2 Timeline of the competition

the submission deadline (April 26th, 2019) for submitting the testing results; the short paper submission deadline (May 17th, 2019) for submitting the complimentary paper describing the way to achieve the challenge; the final decision (June 28th, 2019) for presenting the shortlisted participants; and lastly, the awarding ceremony (July 15th, 2019) for presenting the winners of the competition during the conference IJCNN 2019.

3 Results from the Competition

For this competition, 22 registrations were done (11 as individuals and 11 as teams). Participants were from 14 different countries: Australia, Brazil, China, Estonia, France, Germany, India, Iran, Ireland, Macedonia, Saudi Arabia, Taiwan, Togo and United States of America.

After the results and short paper submission, we announced the three winners of the competition based on the F_1-score metric:

- *First place*: Hristijan Gjoreski (and team) [82.47%]
- *Second place*: Egemen Sahin [34.04%]
- *Third place*: Patricia Endo (and team) [31.37%]
- *Honorific mention*: Vuko Jovicic [60.40%].

The *First place* team used the sensor signals from the wearables. They firstly corrected the orientation of the sensor signals due to the fact that wearables were placed without any particular orientation. After that, they trained three machine learning models, but random forest was the best model that performed 82.47% in F_1-score. Figure 3 shows the confusion matrix of the testing results.

The *Second place* individual tackled the challenge using firstly a standardization of the sensors data (i.e. wearables, ambient sensors and brainwave helmet). Then, he trained 1-dimensional convolutional neural network. After this process, the model performed 34.04% in F_1-score. Figure 4 shows the confusion matrix of the testing results.

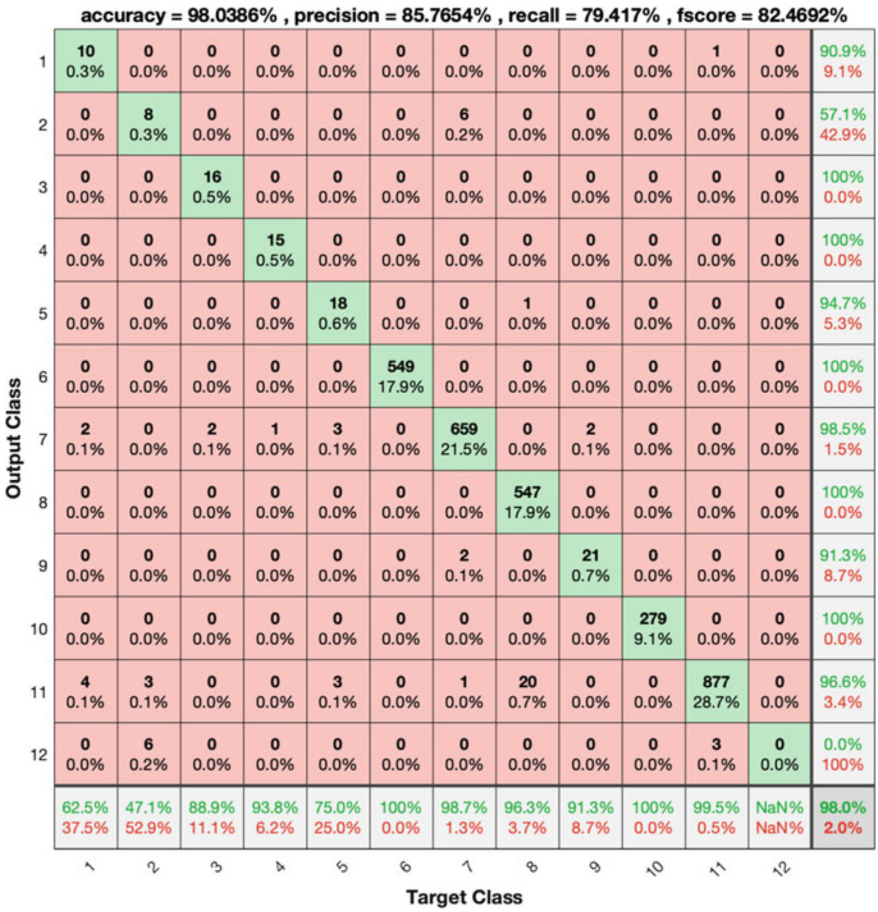


Fig. 3 Confusion matrix of the testing results from *First place*

The *Third place* team employed a bidirectional long short-term memory networks model to achieve the fall classification problem. In this regard, they performed 31.37% in F_1-score. Figure 5 shows the confusion matrix of the testing results.

Lastly, the *Honorific mention* individual obtained a great result in terms of the F_1-score; but, he did not submit the short paper. In this regard, we did not know how he achieved the performance of his model. For that reason, this individual could not be one of the winners. Figure 6 shows the confusion matrix of the testing results.

Although we did not provided a baseline for the participants, we tested four conventional machine learning models: support vector machines (SVM), random forest (RF), multilayer percepron (MLP) and k-nearest neighbors (KNN). This benchmark was published in [1]. We reproduce the baseline in Table 4. As shown, the result from the *First place* is the only one that outperforms the baseline, while the result from *Honorifc mention* is equivalent to the KNN performance.

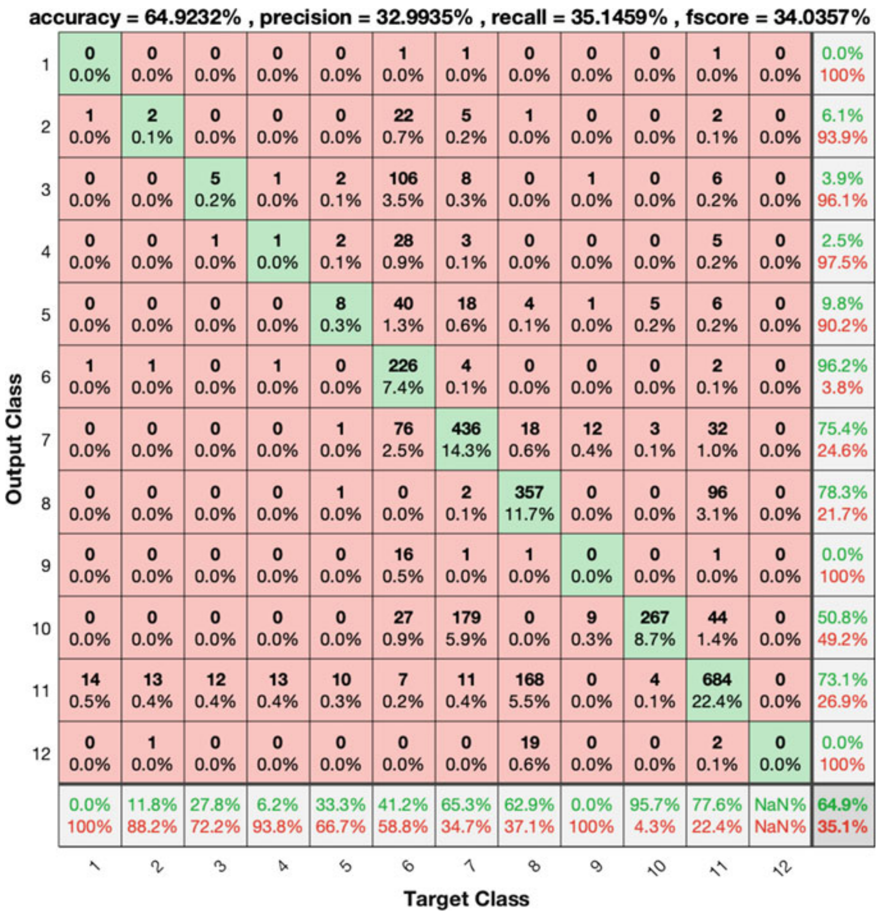


Fig. 4 Confusion matrix of the testing results from *Second place*

Table 4 Baseline using four conventional machine learning models. Values reported are the corresponding F_1-score evaluation, in terms of mean and standard deviation

Model	F_1-score (%)
RF	69.36 ± 1.35
SVM	55.82 ± 0.77
MLP	70.44 ± 1.25
KNN	60.51 ± 0.85

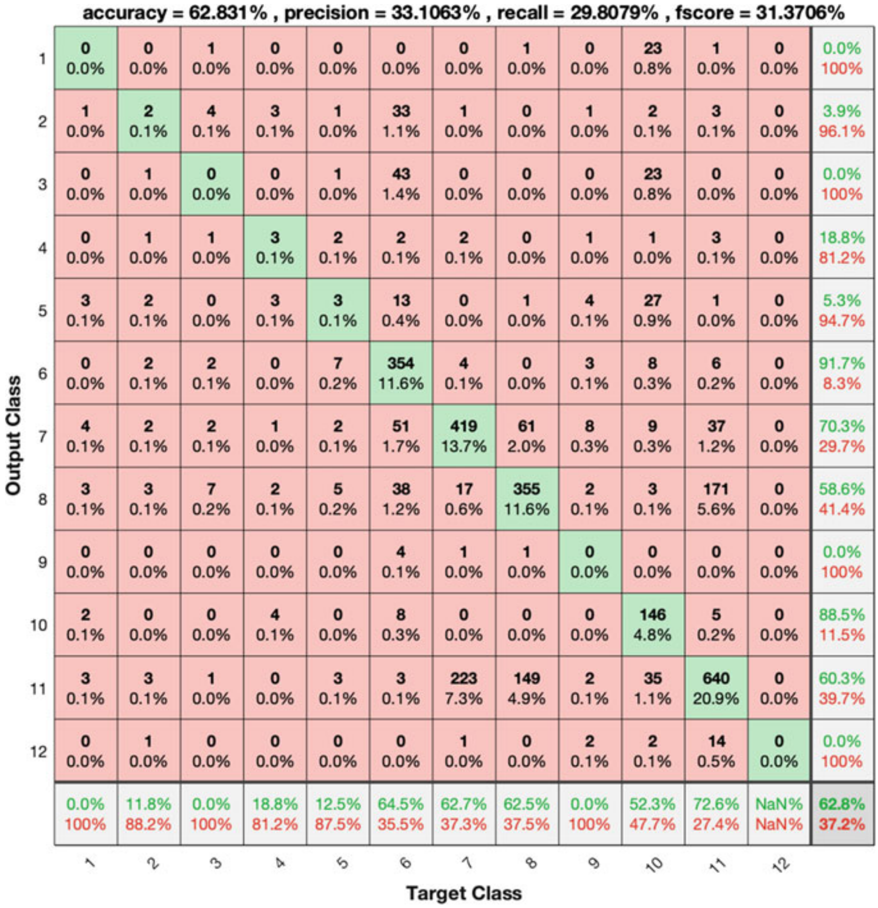


Fig. 5 Confusion matrix of the testing results from *Third place*

4 Concluding Remarks

This competition aimed to propose a multi-class classification model for the problem of human fall classification. In addition, the competition was proposed for challenging participants to apply their computational and machine learning skills in a public, large and multimodal dataset. After the competition ends, we can conclude the following remarks.

In terms of the machine learning models used, it can be seen that conventional machine learning models were employed (e.g. RF, decision trees and KNN). But also, more recent models like convolutional neural networks or bidirectional long short-term memory networks were implemented. Moreover, in terms of the data modality, wearable-based approaches are the most frequent used (i.e. in this competition in

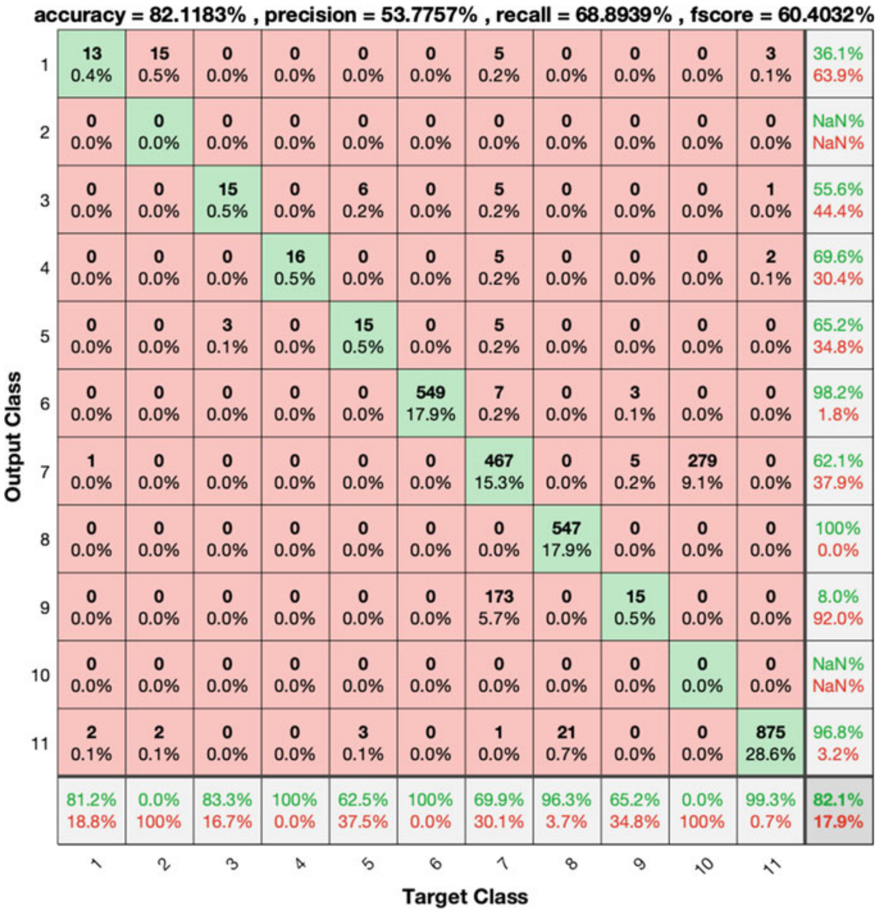


Fig. 6 Confusion matrix of the testing results from *Honorific mention*

all the cases). Ambient sensors were selected in just one attempt. But, cameras were not used by any of the participants. The latter can be associated to the fact that video processing considers complexity and different skills that many of the practitioners do not have. Also, a multimodal approach was not done by any of the participants. It is worth noting that multimodal offers better performance, but it is complex to approach and computationally expensive. In terms of the workflow in data manipulation, participants considers a similar pipeline mainly consisting on: data pre-processing, (temporal) segmentation, feature engineering and training machine learning models. To this end, selection of the best machine learning models and pipelines have to be studied further. Right now, quantitative metrics leads the decision-making process; but this should not be the only criteria for selecting machine learning models and/or strategies to approach fall classification.

On the other hand, the UP-Fall Detection dataset fulfilled the expectations of practitioners in the field of human activity recognition and fall classification. In this regard, this dataset masks the data acquisition problem by giving clean and coherent sensor and camera signals. It can also be used for benchmark machine learning models, as well as different modalities approaches. It is important to highlight that this dataset is publicly available, so practitioners in the field can access and use it as required. Lastly, this dataset provides an important test-bed of machine learning models that can improve the skills of users to develop other applications like in robotics, human-machine interaction, ambient assisted living, among many others.

Finally, fall classification is still an open problem in computer sciences and healthcare, and different open issues have to be faced. For instance, subjects do not perform actions in the same way; but, underlying patterns can be extracted for further analysis. There is some limitation in data since target population is difficult to recruit (e.g. population size, age, type of impairments, etc.). Also, there is highly unbalanced data sets (*falls* vs. *no-falls*). In terms of the sources of information, detection of the best placement of sensors/cameras (and features) is still an issue. Moreover, limitations in resources like computations, memory or budget are constant obstacles in the deployment of these fall classification systems. Of course, there is a need for real-time implementations that should be studied and enhanced. Furthermore, data privacy is still an open concern of fall classification mainly because sensors and cameras are intrusive in daily lives.

Acknowledgements This research has been funded by Universidad Panamericana through the grant "Fomento a la Investigación UP 2018," under project code UP-CI-2018-ING-MX-04.

References

1. Martínez-Villaseñor, L., Ponce, H., Brieva, J., Moya-Albor, E., Núñez-Martínez, J., Peñafort-Asturiano, C.: UP-fall detection dataset: a multimodal approach. Sensors **19**(9), 1988 (2019)
2. Organization, W.H., et al.: Who Global Report on Falls Prevention in Older Age 2007. World Health Organization, pp. 1–7 (2015)
3. Zhang, Z., Conly, C., Athitsos, V.: A survey on vision-based fall detection. In: Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments, p. 46. ACM (2015)

Reviews and Trends on Multimodal Healthcare

This part comprises a set of review and original contributions in the field of multimodal healthcare, taking into account human fall detection and classification. Also, these works present trends on ambient assisted living and health monitoring technologies considering the user-centered approach.

Classification of Daily Life Activities for Human Fall Detection: A Systematic Review of the Techniques and Approaches

Yoosuf Nizam and M. Mahadi Abdul Jamil

Abstract Human fall detection systems are an important part of assistive technology, since daily living assistance are very often required for many people in today's aging population. Human fall detection systems play an important role in our daily life, because falls are the main obstacle for elderly people to live independently and it is also a major health concern due to aging population. There has been several researches conducted using variety of sensors to develop systems to accurately classify unintentional human fall from other activities of daily life. The three basic approaches used to develop human fall detection systems include some sort of wearable devices, ambient based devices or non-invasive vision based devices using live cameras. This study reviewed the techniques and approaches employed to device systems to detect unintentional falls and classified them based on the approaches employed and sensors used.

Keywords Human Fall · Daily life activities · Assistive technology · Elderly care

1 Introduction

Assistive technology is an emerging research area since daily living assistance are very often required for many people in today's aging populations including disabled, overweight, obese and elderly people. The main purpose of assistive technology is to provide better living and health care to those in need, especially elderly people who live alone. It is mainly aimed at allowing them to live independently in their

Y. Nizam (✉)
Department of Engineering, Faculty Engineering, Science and Technology, The Maldives National University (MNU), Malé, Maldives
e-mail: yoosuf.nizam@mnu.edu.mv

M. M. A. Jamil
Biomedical Engineering Modeling and Simulation (BIOMEMS) Research Group, Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia
e-mail: mahadi@uthm.edu.my

H. Ponce et al. (eds.), *Challenges and Trends in Multimodal Fall Detection for Healthcare*, Studies in Systems, Decision and Control 273,
https://doi.org/10.1007/978-3-030-38748-8_7

own home as long as possible, without having to change their life style. In order, to provide better living for them, it is important to have continuous monitoring systems in their home to inform the health care representatives of any emergency attendance. Among such monitoring systems, fall detection systems are increasing in interest since statistics [1, 2] has shown that fall is the main cause of injury related death for seniors aged 79 [3, 4] or above and it is the second common source of injury related (unintentional) death for all ages [5, 6]. Furthermore, fall is the biggest threat among all other incidents to elderly and those people who are in need of support [3, 7–16]. Accordingly, fall can have severe consequences for elderly people, especially if not attended in a short period of time [17]. Similarly, unintentional human fall represents the main source of morbidity and mortality among elderly [18]. Therefore, accurate human fall detection systems are very important to support independent living. Since it had been proved that the medical consequences of a fall are highly dependent on the response and rescue time of the medical staff [19, 20], which is, in fact, only possible with an accurate and reliable fall detection systems that can provide fall alerts. Such systems are also vitally important, since there may be a case where someone losses consciousness or are unable to call for help after a fall event. Additionally, the likelihood of recovering from a fall event has also shown to be reduced if the person remain longer unattended [21]. Therefore, highly accurate fall detection systems can significantly improve the living of elderly people and enhance the general health care services too.

There has been plenty of researches conducted in this area to develop systems and algorithms for enhancing the functional ability of the elderly and patients [19]. This in fact, led to the improvement in the technologies used to develop such systems and thus enhanced the detection ratio to make such systems adaptable and acceptable. Recent researches conducted on human fall monitoring approaches for elderly people were categorized into wearable sensor based, wireless based, ambiance sensor based, vision and floor sensor/electric field sensor-based approach to distinguish the different fall detection methods employed [22]. This categorization of fall detection methods, reflects the characteristics of the movement that leads to fall. Therefore, it is also important to recognize those characteristics of movement in order to understand the existing algorithms used to detect falls and to device new algorithms to enhance the performance of such systems.

The various methods that has been used to detect human fall such as using a camera to identify a human fall posture or using various sensors to detect fall, shares some common features. From the analysis, the different fall detection methods based on various sensors were divided into three main approaches [23]. These three approaches are further divided into different sub-categories depending on the sensor and algorithm used to distinguish the detection methods. In this regard, the three basic approaches are wearable based device, camera based systems and ambience based devices [19, 23, 24] as shown in Fig. 1.

As shown in Fig. 1, wearable based device is further divided into two sub-categories based on the fall detection methods used. They are inactivity (motion based) and posture based approaches. Similarly, ambient/fusion based devices are divided into two types; those that used floor sensors or electric field and those that

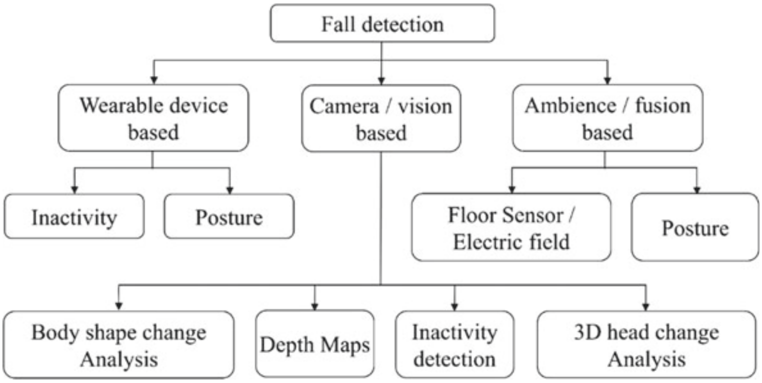


Fig. 1 Hierarchy of fall detection methods

used posture based sensors to detect fall. Camera or vision based approach is divided into four different sub-categories.

Wearable based devices use accelerometers and gyroscopes which are embedded into garments or any wearing gadgets such as belts, wrist watches, necklace or jackets. The basic concept used to classify human fall is either identifying the posture or through activity/inactivity detection. Ambient/fusion based devices are a type of non-invasive and non-vision based approach. It either use the concept of posture identification through various sensors or uses floor vibration or electric sensors to detect the subject hitting the sensor. On the other hand, vision based (non-invasive) approach uses live cameras or multiples of such cameras to accurately detect human falls through utilizing the analytical and machine learning methods based on a computer vision model. They utilized various approaches to classify human fall including the changes in body shape between frames, activity/inactivity detection of the subject, three-dimensional (3D) analysis of the subject using more than one camera and generating a depth map of the scene with the help of depth sensors. Except the depth sensor based method, the other types in vision based approaches use RGB (Red Green Blue) cameras and therefore they are subject to rejection from users due to privacy concerns. They are also rejected due to the high cost of the systems, installation and camera calibration issues. A depth image-based approach could solve the privacy issues arising from video based systems. Studies representing this approach can also be divided into three types based on the approaches used. The first category represents those works that employed joint measurements or used human joint movements from depth information to detect fall. The second category includes the works that depended only on depth data with any supervised and unsupervised machine learning to classify human fall from other activities of daily life and the third category includes studies that make use of wearable devices along with the depth sensor. This study presents a review of these fall detection techniques. From the classification of daily life activities to the different approaches used to identify unintentional human fall are reviewed in the following sections.

2 Common Daily Life Activities and Their Associated Falls

Classification of human activities is an important research topic in the field of computer vision and rehabilitation. It is also increasingly in use for many applications including intelligent surveillance, quality of life (such as health monitoring) devices for elderly people, content-based video retrieval and human-computer interaction [25–28]. There has been plenty of researches conducted to automatically recognize human activities, yet it remains a challenging problem. Many approaches are used to identify how human moves in the scene. Techniques employed includes tracking of movements, body posture estimation, space-time shape templates and overall pattern of appearance [29–34].

Identification of characteristics of daily life activities is important in order to classify them, especially for fall detection. The characteristics of the activities can help to identify any uniqueness or dissimilarities between activities which in turn will support in distinguishing them. The characteristics of activities are basically derived in terms of height change pattern, rate of change of velocity, deviation of the height and position of the subject during and after the movement. This identification of characteristics of daily activities together with the pattern and the rate of change is important to classify them, especially unintentional human fall. Pattern of change is referred to the variation of subject's height with respective to floor during any of the activity and the rate of change is the changes in velocity of the subject during that period. Deviation of height is the similar pattern of change of height except that this is the statistical standard deviation of that changes in height pattern. The changes in the pattern itself refers the height of the subject in different frames. In some cases, acceleration can help to identify the difference, where the changes in velocity does not clearly show the variation of speed during the movement.

Generally a human fall is an unintentional or involuntarily event that causes a person to rest on the ground or any other lower level object [35]. Basically, falls have varying meaning which may be caused from either intrinsic or extrinsic factors. Intrinsic are those that had caused from any physiological reasons and extrinsic are from the environment or other hazards. Tenetti et al. defined a fall from non-hospitalized geriatric population as "an event which results in a person coming to rest unintentionally on the ground or lower level, not as a result of a major intrinsic event (such as stroke) or overwhelming hazard." [36]. By adapting this definition, fall for inpatient, acute, and long-term are described as "unintentionally coming to rest on the ground, floor, or other lower level, but not as a result of syncope or overwhelming external force." [37]. Therefore, this study interchangeably uses human fall or unintentional fall to refer to any such unintended descend to the floor or on any object with or without injury which may had cause from any factors (intrinsic or extrinsic).

Some of the activities of daily life that are closely related to human fall are shown in Table 1, along with the characteristics considered in the classification of the events. The characteristics of each of the activity is divided into static and dynamic components. Dynamic component describes the change in subject's height and rate

Table 1 Characteristics of some of the daily life activities and falls

Activities	Characteristics	
	Static	Dynamic
Sitting on chair	Hip center and head vertical, angle formed from hip center to head and hip center to knee <90°	Height drop slowly straight to down Velocity increase slightly to straight down
Stand up from sitting on chair	Posture while standing Key joint will be straight (head to torso then hip to the joints of the two limbs)	Height will increase slowly at the beginning and gradually the key joint of the subject will be straight Velocity increase to up (at the beginning the speed will be slower and will increase as the body is lifted)
Sitting on floor	Hip center and head vertical, hip center, left and right knee on floor	Height drop slowly down to floor from left, right or in front of the subject Velocity increase slowly down to floor from left, right or in front of the subject
Stand up from sitting on floor	Posture while standing Key joint will be straight (head to torso then hip to the joints of the two limbs)	Height will increase slowly at the beginning and gradually the key joint of the subject will be straight Velocity increase to up (at the beginning the speed will be slower and will increase as the body is lifted)
Lying on floor	Hip center and head horizontal, all joints below or equal previous knee height	Height drop slowly down to floor from left, right or in front of the subject Velocity increase slowly down to floor from left, right or in front of the subject
Stand up from lying of floor	Posture while standing Key joint will be straight (head to torso then hip to the joints of the two limbs)	At the beginning, height will increase vertically up to any direction until the body is balanced to the hip or using the arms and the legs. After that the height will increase straight up The instant velocity will increase slowing at the beginning and will be at rest for a while and then increase rapidly

(continued)

Table 1 (continued)

Activities	Characteristics	
	Static	Dynamic
Lying on bed	Hip center and head horizontal, all joints above or equal to previous knee height	Height drop slowly down to left, right, back or in front of the subject Velocity increase slowly down onto left, right, back or in front of the subject
Fall from standing	All joints below or equal previous knee height	Height drop rapidly down to left, right, back or in front of the subject Velocity increase rapidly down onto left, right, back or in front of the subject
Walking		Height will fluctuate slowing due to errors Instant velocity will fluctuate depending on the walking speed to the direction of walking
Fall from chair	All joints below or equal previous knee height	Height drop rapidly down to left, right, back or in front of the subject Velocity increase rapidly down onto left, right, back or in front of the subject The duration for both height drops, and the velocity is smaller than fall from standing
Fall while trying to sit on chair	All joints below or equal previous knee height	Height drops slowly and all of a sudden it drops rapidly Velocity increases slowly and then very fast until the body rests on floor
Fall while trying to sit on floor	All joints below or equal previous knee height	Height drops slowly, then it subsides and drops rapidly Velocity increases slowly and comes to normal, then very fast until the body rests on floor

(continued)

Table 1 (continued)

Activities	Characteristics	
	Static	Dynamic
Fall from bed	All joints below or equal previous knee height (some joints may not be detected)	Height drop rapidly down to left, right, back or in front of the subject Velocity increase rapidly down onto left, right, back or in front of the subject The duration for both height drops, and the velocity is smaller
Fall while sitting on floor	All joints below or equal previous knee height	Height drop rapidly down to left, right, back or in front of the subject Velocity increase rapidly down onto left, right, back or in front of the subject The duration for both height drops, and the velocity is smaller

of change of velocity during the activity. Static component will describe the position of the subject at the end of the activity, in terms of joint angles and joint position with respect to other joints.

Falls can be divided into many types from the characteristics of movements that causes falls. The work presented by Yu, divided falls into four types [24]. They are fall from sleeping (bed), fall from sitting (chair, on floor or any object), fall from walking or standing on the floor and fall from standing on a support such as ladder or any such tools. All the four types of falls do share some common characteristics even though it has significant differences [19]. This study is concerned with the first three types of fall, because the last type of fall is not common amongst elderly people since they normally occur among workers or people doing household jobs. The existence of fall like characteristics in normal daily life activities such as similarities in crouch and lying on floor with fall is the main challenge that the researchers are facing in developing systems to classify falls from human activities.

The different approaches and methods of fall detection mentioned do share a similar general framework [24]. Some of the fall detection methods used only one sensor indicator with a threshold while others used complicated algorithms and image processing to detect falls. The only distinguishing factor among them lies on the sensors used, number of sensors and their detection algorithms. Such as data acquisition can vary from single sensor to multiple sensors to sense one indicator and different cameras working together to collect data [24]. The framework followed in the three basic types of fall detection also differ from the methods of architectural design and communication between inter components.

3 Overview of Fall Detection Approaches

Most of the previous academic researches on fall detection had based their work on some sort of wearable devices with embedded sensors to identify posture and motion of the body. The sensors were either placed on garments or are in the form of wearable devices. Different types of sensors were used to make such devices and hence there are variety of methods to detect falls.

Among them accelerometer is the most extensively used sensor to realize human fall. They make use of the measure of the acceleration of the body to identify any potential fall activity. Apart from that there were some works conducted on determining falls using the human posture movements. Body orientation as posture was used to detect fall using either posture sensors or multiple of accelerometers [38, 39]. Combination of accelerometer and gyroscopic data also proved to determine falls more accurately than using a single sensor. Accelerometer can provide kinetic force while gyroscope can help to estimate the current posture [40]. The combination of two sensors can also help to identify any false measurement from anyone of the sensor.

Wearable based devices, basically employs the unique pattern of motion, the fall possess to distinguish it from other activities of daily life. Therefore such devices are prone to give false alarm by triggering fall alarm from any irregular motions of a daily life activity [24, 41]. The sensor position and the fusion of data techniques used also greatly affects the preformation of such systems [42, 43]. Similarly, it is also very likely that the user may forgot to wear the device during their daily life activities [44, 45]. Especially elderly people, because they are likely to have weaker memory. Moreover it is often rejected by the elderly due to the difficulties of the wearing such devices or garments. Regardless of these issues, they are popular for the advantages of been cheap, readily available, and easy to setup and operate.

Ambient based methods shown in Fig. 1, basically used non-invasive sensors for developing fall detection systems. This approach usually used array of sensors to identify falls through pressure sensing, vibrational data, IR sensor and single far-field microphone [46–50]. Similarly like wearable based approach, they do possess several disadvantages. Since it is mostly based on pressure sensor which is very prone to measure weight of all objects and thus generates high false alarms. Unlike wearable devices, it is less disturbance to the users and some of the devices like floor vibration based devices are fully non-invasive. Similarly like wearable devices it is very cost effective and does not require high installation costs [46, 48].

In overall, most of the wearable sensor based devices cannot distinguish fall from sitting especially if it is accelerometer based. Similarly, it is very often rejected by the wearer due to the difficulty of wearing such belts or garments and discomforts associated within it. Ambient based devices are also prone to generate high false alarms mainly from the pressure sensors which measure the pressure of everything.

On the other hand, vision based devices are fully non-invasive and are based on computer vision to identify human from the scene and detect falls. They used image processing to segment human subject from the scene and employ mathematical models to classify human activities. As a result, they are very accurate in identifying human falls than the other two approaches and they are fully non-invasive to the user. On the other hand, such systems are rejected by the users due to privacy concerns. Consequently, such systems were subject to the physiological effects on the users due to the recordings of their daily life activities. A brief review of the previous studies from the three basic approaches used for developing human fall detection systems were discussed in the following sub-sections.

3.1 Wearable Based Techniques

The basic approaches used to develop fall detection using wearable sensors are discussed in the following sub-sections.

3.1.1 Accelerometer Based Devices

Among the wearable devices, body posture and motion recognition using accelerometer is the most extensively used method to realize a fall. They used the measure of the acceleration of the body with respect to the position where the sensor is placed. This accelerometric data was then used to detect any potential fall activity. The acceleration data from triaxial accelerometers worn on the waist, wrist, and head were used in many studies to monitor for fall events [51–54]. Some of the studies employed machine learning such as Hidden Markov Model (HMM) [55] and Support Vector Machine (SVM) [56, 57] on the accelerometer data for fall detection. One study used discrete wavelet transformation (DWT) to process the collected data from the accelerometer placed in shoulder position of a jacket [58]. While another study used a personal server for controlling the data acquired from multiple biomedical sensors to detect fall event [59]. Smartphones are also used to take benefits of the integrated sensors for fall detection [60, 61]. Kangas et al., presented a study to determine simple threshold for an accelerometry based parameters for fall detection [62].

Some studies used vertical velocity thresholding [63] or vertical acceleration from piezo electric accelerators [64, 65] to detect human fall events. Acceleration and angular velocities were also used to differentiate real fall events from other normal daily life activities [66, 67]. Another study presented a wireless sensor node with triaxial accelerometers to identify the body acceleration for monitoring daily activities [68]. A sensor network based fall detection was also used, to monitor the user with complete privacy and security [69]. Chen et al. created a wireless low power sensor network with small, non-invasive low power motes (sensor nodes) for fall detection [70].

The changes in body motion and body position were also used for fall detection [71, 72]. Rapid impact and body orientation based fall detection proposed by Zheng et al. used a two stage fall detection algorithm which can locate the wearer and send alarm [73]. The changes in body orientation was also used to identify negative acceleration for fall detection [74]. Another study described the use of IoT on a sensor node with an accelerometer the data is transmitted to fog layer [75]. The IoT are also increasing in use in recent days with the development of the new techniques and smart sensors [20].

3.1.2 Posture Sensor Based Devices

There are some works conducted on determining fall using the posture of the subject or posture movements. Body orientation as posture is used to detect fall using either posture sensors or multiple accelerometers. Combination of tri-axial accelerometers and gyroscopes were also used to identify human behavior and posture of the subject such as the study proposed by Baek et al. which used the two sensors as a necklace sensor node for fall detection [76]. Two studies employed set of sensors to identify the movement and posture of the subject for fall detection [38, 77]. Kaluza et al. presented a posture reconstruction ideology for fall detection algorithm by locating the wireless tags which were placed on body parts (sewn on clothes) such as hips, ankles knees, wrists elbows and shoulders. This fall detection algorithm used acceleration thresholds along with velocity profiles. Acceleration was derived from the movements of the tags [39].

Another body posture based fall detector proposed by Sudarchan et al. used a trixial accelerometer placed on the lumbar region to study the tilt angle. They used the changes in acceleration in three axes to find the body posture [78]. While Kangas et al. used a waist worn tri-axial accelerometer, transceiver and microcontroller to develop a new fall detector prototype based on fall associated impact and end posture [79]. Another study used a multi-level data fusion framework on multiple sensor nodes [80].

3.1.3 Accelerometer and Gyroscope Based Devices

Combination of accelerometer and gyroscope data has also proved to determine fall more accurately than any one of the sensor alone. As discussed, accelerometer can provide kinetic force while gyroscope can help to estimate the current posture. The combination of two sensors can also help to identify any false measurement from anyone sensor. Hwang et al. used this concept to propose real time monitoring ambulatory system for human fall detection [81]. Nyan et al. also used 3D accelerometer and 2D gyroscope worn on thigh which was based on Body Area Network (BAN) to prevent from fall related injuries by inflatable airbag for hip protection before the impact. The system was based on the concept that thigh segments will not exceed a certain threshold angle to side and forward in normal daily activities except during a fall event [40]. Another study presented a novel fall detection system using same two sensors to recognize four kinds of static postures: standing, bending, sitting and lying. Motion between these static postures were considered as dynamic transitions and if the transition before lying posture is not intentional, a fall is detected [82].

Using the same two sensors, a physical activity monitoring system was presented by Dinh et al. The wearable device detects the physical activities using 3-axial accelerometer, a 2-axial gyroscope and a heartbeat detection circuit. [83]. On the other hand, Bourke et al., used a bi-axial gyroscope sensor mounted on the truck to differentiate fall from normal daily activities. Fall was determined from measurement of pitch and roll angular velocities with a threshold-based algorithm [84].

In the studies discussed above, a fall is distinguished from other normal daily life activities using the unique pattern of motion the fall possess. Therefore, it is prone to generate false alarm by triggering a fall event from a motion of any other similar movement such as lying on floor from standing. More ever it is often rejected by the elderly due to the difficulty of wearing the devices or garments. Irrespective of this, it does have the advantages of been cheap, easy to setup and operate.

3.2 Ambient Based Techniques

This approach usually used array of sensors to identify falls through pressure sensing, vibrational data, IR sensor and single far-field microphone. They make use of the ambient noise including vision, audio and floor vibration caused from any potential fall activity [85].

One study used array of vibration sensors on the floor to identify fall by analyzing location data [46]. The methods used was based on the perception that a human fall will always cause a vibration pattern on the floor and implies that the vibration generated from fall is significantly different from normal daily activities and at the same it will be different from the vibration generated by objects falling on the floor. Livak et al. based their method on the floor vibration and acoustic sensing for fall detection. The detection was based on the detection of vibration and sound signal from an accelerometer and a microphone with advanced processing techniques. The proposed system could detect falls with a high accuracy for distance up to 5 m and system is adaptive that it could be calibrated to any kind of floor and room acoustics. It was also free from ambient noise because the algorithm had to detect a vibration event in the first stage. Results from the testing database showed a sensitivity of 95% and a specificity of 95% [86].

A pressure sensor was also used to design a bed exit detection apparatus by using bladder or other fluid carrying devices in fluid communication. This is particularly a patient presence detection system that enabled the care giver to get alerted about both the presence and absence of the patient on patient carrying surface. Especially whether the patient was sitting upright or were leaning, falling forward or falling sideways [87].

The other non-invasive sensor based approach was the use of IR sensor to locate and track a thermal target in the sensor's field of view such as the study proposed by

Sixsmith et al. which was based on pyroelectric IR sensors placed on wall for detecting fall [47, 48]. A wireless sensor network with array of sensors and event detection and modalities and distributed processing for smart home monitoring application was also used for fall detection [49].

Audio signal from a single far-field microphone was also used to detect human falls in the home environment, distinguishing them from competing noise, by using audio. A study modeled each fall or noise segment using a GMM super-vector to distinguish them from background noise and classify the audio segments into falls and other types of noise using SVM built on GMM super-vector kernel [50]. One study used three radar signal variables to detect falls along with an unsupervised multilinear feature extraction method [88]. The use of WiFi Channel state information is also an emerging non-invasive approach for fall detection. It is does has the advantage of being ubiquitous and low cost compared to the previous radar based approach [89].

Ambient approach too possesses several disadvantages like wearable devices. Since it is mostly based on pressure sensor which is very prone to measure weight of all objects, thus generates high false alarms. Unlike wearable based devices, it is less disturbance to the users. Similar like wearable based devices it is very cost effective and does not require high installation costs.

3.3 Vision Based Techniques

Vision or camera based devices are increasingly in use due to its multiple advantages over wearable and other non-invasive sensor based devices. Some of the reasons includes the capabilities of these cameras to detect multiple events simultaneously and relief from the wearing difficulties of wearable devices such as garments for fall detection. Most importantly the recorded video from camera can be used for verification after a fall event. Although vision based approach do possess the disadvantages of not preserving the users' privacy, it is very commonly employed in many research works. Selected previous works on normal color camera or vision based devices are briefly reviewed in the following sub-sections.

3.3.1 Inactivity Approach

With this approach, a fall is detected based on the inactivity period on the floor. Camera or motion detector tracks the person to obtain motion traces and based on it, a fall is determined. The orientation change of the body was used to detect inactivity and if inactivity occurs in certain context, a fall is detected [90].

One study, demonstrated the usefulness of unusual inactivity detection as an indication for fall detection. The proposed method enables inactivity outside the usual zones of inactivity such as chairs or beds to be identified. The combination of this with body pose and motion information provided important information for fall detection [91].

3.3.2 Shape Change Approach

The main perception with this approach is that the shape of a person will change from standing to lying if a fall occurs. One study used a Hidden Markov Model (HMM) based fall detection where HMM used video features to differentiate fall from walking. A second HMM based approach used audio features to differentiate falling sound from talking [92]. Another HMM-based algorithm used multiple features extracted from silhouette: height of bounding box, magnitude of motion vector, determinant of covariance matrix and ratio of width to height of bounding box of person [93]. Thome and Miguet proposed robust Hierarchical Hidden Markov Model (HHMM) based algorithm to detect fall, where HHMM is used to model the motion. Many improvements are possible including automating the rectification processes using Hough transfer for detecting sets of parallel lines and computing the orthogonal vanishing points [94].

Another method, to detect falls on the floor using multiple cameras presented by Auvinet et al. evaluated the variation of the person's height for fall detection. The advantages of proposed system with multiple camera were larger common field of view and detection of an occlusion in one camera could be dealt from the other camera [95]. An alternative method used, 3D shape of body extracted from multiple cameras to detect fall. This multi-camera vision system for detecting and tracking people employs warping people's silhouette technique to exchange visual information between overlapped cameras when camera handover occurs [96].

A single wide-angle camera based method used the angle between the projected gravity vector and the line from feet to head of the human and normalized size of the upper body for fall detection [97]. Another study used a rule based algorithm with an Omni-camera and used context information for fall detection. Fall was determined based on the ratio of width to height of the bounding box of body in the image [98, 99]. One study demonstrated a machine learning framework for fall detection and daily activity classification using acceleration and angular velocity data from two public datasets. They tested the performance of artificial neural network (ANN), K-nearest neighbors (KNN), quadratic support vector machine (QSVM) and ensemble bagged tree (EBT) in recognizing the activities such as falling, walking, walking upstairs, walking downstairs, sitting, standing lying using acceleration and velocity data [100].

3.3.3 3D Head Motion Analysis

The principle behind this approach is that during a human fall, vertical motion is faster than the horizontal motion. Rouger et al. developed fall detector using monocular 3D head tracking. The method presented was based on Motion History Image (MHI) and changes in human shape. The detection method was based on the fact that the motion is large when a fall occurs. Therefore, the system first detects a large motion and if a motion is detected then the shape of the person in the video sequence are analyzed. In this stage the concept used is that during a fall, the human shape changes and at the end of the fall the person is usually on the floor for few second and with less body movements [101].

Vision based approach was the most reliable technique for fall detection compared to the other approaches [96, 99, 101]. If individual different sub-categories are compared, inactivity detection was simple in terms of processing and hence they are less reliable. Shape detection algorithms was more reliable because body shape detection could give more accurate information about fall than head detection. The 3D body shape detection used more camera and required complex computing. The recent trend in fall detection is the use of depth sensor for human detection and movement recognition. The three techniques discussed in vision based approach namely inactivity detection, shape change and 3D head motion analysis could be implemented with a single depth sensor.

3.4 Review of Depth Sensor Based Approach

The use of depth sensors for fall detection, can be further broaden into three subcategories depending on the approach employed to device the systems, as illustrated in Fig. 2. The differences between them is only the fall detection algorithm employed and the method used to generate potential fall alert which starts the fall confirmation process. The first category represents those works that employed joint measurements or used human joint movements from depth information to detect fall. The second category includes those that depended only on depth data with any supervised and unsupervised machine learning to classify human fall from other activities of daily life. The works that make use of an inertial devices (wearable devices) along with the depth sensor and employs either machine learning or joint data for the classification of human activities is discussed under the third category in Sect. 3.4.3.

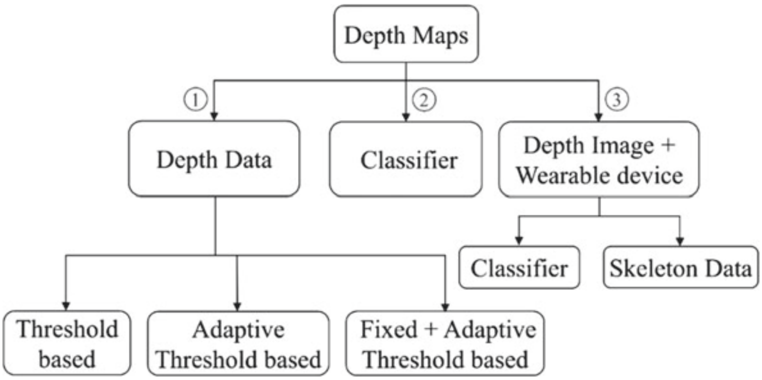


Fig. 2 Hierarchy of depth map based methods

The first and the second approach (shown in Fig. 2) uses only depth maps generated from the sensor while the third approach makes use of wearable device along with the depth data to improve the accuracy and cover the areas not visible to the depth sensor. From the study, it was found that the only option to fully utilize the benefits of being non-invasive and to achieve the confidence of the users is to use only depth images. Since the use of wearable devices along with depth sensor violates the concept of non-invasive. Thus, makes it also subject to rejection from users due to the wearing difficulties and other drawbacks of wearing devices.

The techniques employed in those that used only depth images (the first and the second approach in Fig. 2) to classify human activities are based on joint measurement and movement for fall detection. The first sub-category of the first approach (depth data) signifies the works that used fixed thresholding on the skeleton data or on extracted human joints to detect fall. The second and third sub-category in this approach represents those works that used an adaptive threshold and a combination of fixed and adaptive thresholds respectively.

The third approach shown in Fig. 2, which used wearable device along with the depth images is also classified into two sub-categories. The first category represents those that used joint measurement methods on the skeleton data or extracted human joints and a classifier on the accelerometer data. The second category includes those that used machine learning either on the accelerometer data and or on the depth images for the classification of human fall.

3.4.1 Joint Measurement and Movements Based Approaches

This section discusses on the works that had based their systems only on depth information. Including those that used skeleton data from depth sensor and those that used mathematical models and algorithms to detect fall by either extracting human subject or tracking the movements. Most of the studies used distance of key joints and velocity for fall detection. Rougier et al. used human centroid height relative to the ground and body velocity. They have also dealt with occlusions, which was a weakness of previous works and claimed to have a good detection results with an overall success rate of 98.7% [102]. While another study used distance and velocity as two algorithms [103]. The distance and y-coordinate of joints were also used as two algorithms to detect falls, where a fall is detected by thresholding the distance between these joint to the floor, if the floor is visible or detected [104]. If the floor plane is not detected, then a second algorithm is used which depends on the skeleton coordinate system. The second algorithm detects a fall if the y-coordinates of the joints mentioned are less than a given threshold.

Another study presented a distance and velocity based algorithm with four main steps, where fall detection is based on the vertical speed and the distance from the ground to head and the centroid [105]. Distance and angle based method was proposed by Yang et al. The characteristics of the shape of the moving objects is illustrated by an ellipse which is computed using a set of movement functions. The parameters calculated are the centroids of ellipses, the angles between the ellipses and the floor plane. Fall is detected when the distance and angle between the ellipses and floor plane are lower than some threshold [106].

A combination of the skeleton and RGB data from Kinect sensor was also used in a way that if skeleton data is available, the algorithm uses vertical velocity and the height to the floor plane of the human center. Otherwise, the motion map from the gray scale images which is represented by an improved kernel descriptor is input to a linear Support Vector Machine (SVM). This fall detection flowchart is based on skeleton and RGB features (which is switchable) depending on the availability [107]. On the other hand, Zhang S. et al. used three velocity features and a head to hip height difference with an adaptive threshold [108].

The velocity of the joint hitting the floor was also used to distinguish the fall accident from sitting or lying down on the floor. An abnormal action is detected if the distance between any of the joints to the floor is smaller than an adaptive threshold and any of the vertical velocities with respect to floor is larger than a velocity threshold at the same time. Fall is detected if the distance between the head and floor is lower than a recover threshold for certain period of time which is also adaptive to the height of the person [109].

Only one study used distance parameter for fall detection. They used distance from head to floor plane which is calculated in every following frame and a fall is indicated if an adaptive threshold has reached. The centroid height of the human subject is used as the second decision to confirm fall event. Fall event is confirm only when the distance from head to floor (head height) and the distance between silhouette centroid to floor (centroid height) is lower than the thresholds at the same time [110].

Similarly, one study based their fall detection mainly on velocity components. They also proposed a two-stage based fall detection where in the first stage, the vertical state of the subject from each depth image frame is characterized and then segments the ground events from vertical state time series obtained by tracking the subject over time. The vertical state of the subject is characterized using three features. The second stage confirms that a fall has occurred by using an ensemble of decision trees and a set of features extracted from on-ground event. For the computation of fall event, five features from each on-ground event is extracted [111].

The orientation of the person's major axis and the height of the spine from floor was also used to detect fall event. For the computation of the orientation, the head, shoulder center, spine, hip and the mean position of the knees were considered. The 3D ground floor and the spine distance to floor is also calculated. Fall is detected, if the major orientation of the person is parallel to the floor and the height of the spine is near the floor [112].

Four studies employed bounding box based approach for fall detection. In the first study a fall is detected using velocity and inactivity calculations. Where velocity measurement is based on the contraction or expansion of the width, height and depth of the 3D bounding box [113]. The second study used the contraction and expansion speed of the width, height and depth of the 3D human bounding box with its position in the space [114]. The third study introduced an adaptive directional bounding boxes technique to detect falls. They used four main features to analyze fall events. They are: Directional Height (DH)/Directional Width (DW) ratio, center of gravity ratio, diagonal ratio and Bounding Box (BB)-Height ratio. They also used dynamic state machine to encompass both forward and backward tracking [115]. The fourth study used skeleton tracking along with bounding box analysis [116].

A privacy-preserving fall detection method proposed by Gasparrini et al., used raw data directly from the sensor. The data were analyzed, and the system extracts the elements to classify all the blobs in the scene through the implemented solutions. A fall is detected if the depth blob associated to a person is near to the floor [117].

A motion based fall detection approach used extracted skeleton data of human using Kinect sensor to detect and monitored the person, especially the changes in motion is examined. Fall is detected using the changes in either Y or Z coordinate of the key frame. A fall from sitting or standing is confirmed if the body motion gets involved in Y or Z coordinate [118]. Another similar approach employed the torso angle, the centroid height and their motion characteristics for human fall representation. These parameters were used to create a human torso motion model (HTMM) which is a threshold based approach for fall detection. The 3D position of the hip center and the shoulder center joints in depth images is used to build proposed model. The subject's torso angle and the centroid height is the key features in the HTMM. They used four thresholds, first threshold is for torso angle which starts detection, the second is the threshold for the changing rate of torso angle, the third is the threshold for velocity of centroid height and the fourth is the threshold for tracking time after the torso angle exceeds the first threshold. When a person is detected, the position of the joints is extracted, and the torso angle is calculated. When this angle is greater than a given threshold, the rate of changes of the torso angle and the centroid height are recorded frame by frame for a given period of time and a fall is detected when this changing rate reaches the thresholds [119].

3.4.2 Classifier Based Approaches

Studies representing this category employed some sort of machine learning classification either directly on the depth data or on the 3D skeleton joint position from Kinect sensor. Alazrai et al. presented a new approach for fall detection using 3D skeleton joint position from Kinect sensor. This data is used to build a view-invariant descriptor for human activity representation or motion-pose geometric descriptor (MPGD). MPGD comprises of a motion and pose profile which allows the capturing of semantic context of the human movements from the video sequences. Fall detection is first expressed as a posterior-maximization problem where the posterior probability is estimated using a multi-class SVM classifier [120].

A combination of fall characterization using shape based approach and a machine learning classifier to identify human fall from other activities was proposed by Ma et al. At first human silhouette is extracted from depth images. Adaptive Gaussian Mixture Model (GMM) is used for human segmentation from background. The second step involves finding of the features of the detected subject [121]. On the other hand, another study used a deep learning classifier based human fall detection using infra-red depth sensor measurements with feature selection and Non-Linear Principal Component Analysis (NPCA) [122]. While an acoustic based fall detection system using Kinect sensor with Minimum Variance Distortion less Response (MVDR) adaptive beamforming reduced the false alarm ratio by 80% as compared to the case when no depth sensor is used [123].

Another study, accomplished fall detection using only depth images with a classifier trained on features representing the extracted person both in depth images and in point cloud [124]. The features extracted are the same as that of the study in previous section [125].

The only study that had based fall detection on statistical method employed how human moved during the last few frames for the classification. They used statistical decision making as opposed to hard-coded decision making used in related works. Duration of fall in frames, total head drop-change of head height, maximum speed (largest drop in head height), smallest head height and fraction of frames where head-height dropped is considered for fall detection [126].

3.4.3 Wearable Device and Depth Map Based Approaches

This section is dedicated for discussing the works that make use of a wearable device along with depth sensor for fall detection. The wearable devices are sometimes used to generate potential fall alert and/or are used to cover the areas not visible to the depth sensor. Either machine learning or changes in human joints from depth information are used to confirm the fall event. Two studies used joint data for fall detection along with a wearable device. The first study used the distance of the centroid of the segmented person to the ground plane with a threshold for fall detection [127]. The second study that employed joint data for fall detection presented three algorithms. The first algorithm for fall detection uses acceleration data from the wearable device on the wrist and skeleton data from Kinect sensor. A fall is detected, if three conditions is fulfilled. The second algorithm uses the same parameters and concept except that the accelerometer is placed on the waist. The third algorithm uses variation of the skeleton joints, distance of spine_base joint from floor and magnitude of waist accelerometer. Fall is detected, if the acceleration peak of greater than 3 g is observed with 2 s after the distance of the joints reaches a threshold value of 20 cm [128].

There were four studies that used classifiers for fall detection along with a wearable device. One study used a wearable accelerometer based device to indicate any potential fall activity and whether the wearer is in motion. The authentication of fall event after a potential fall indication from the accelerometer is accomplished from depth images using SVM classifier on the features extracted. The features extracted to confirm fall are the ratio of width to height of the person's bounding box (h/w), a ratio of the height of the person's surrounding box in current frame to the physical height of the person (h/h_{max}), the distance between centroid to the floor (D) and the standard deviation from the centroid for the abscissa and the applicate [125]. Similarly, other studies used a k-NN classifier for lying pose detection after a potential fall alert from a wearable device [129, 130].

While another study made the use of accelerometer as optional to indicate a potential fall and a Support Vector Machine (SVM) based person finder is used to confirm the presence of the tracked person and the head location. A cascaded classifier consisting of lying pose detector and dynamic transition detector is also executed [131]. Another approach made the use of accelerometer and video based approach switchable for different situations. Since in cases such as during changing clothes or while washing, it might not be comfortable to use the wearable sensor. In such situation the system relies on Kinect camera only [132].

A previous work [128] was extended by Cippitelli et al. which presented a fall risk estimation and fall detection tool using a wearable and vision based sensor. This work was aimed to propose an integrated system to gain both the fall risk assessment and fall detection in indoor home environment. They also provide a fall risk assessment tool with the Kinect sensor and an accelerometer placed at the chest in the same setup which can be switched when required. The test is divided into five phases, sit-to-stand, walk, turn, walk, turn-to-sit. During sit-to-stand phase, the person stands from chair to start walking. The parameters evaluated are maximum inclination of the torso angle and the time required to stand up. During the walk phase, steps of the person are extracted from both the accelerometer and the Kinect. The turn phase, is when the subject turns to walk back to the chair. The parameter extracted in this phase is the time required to perform the action (turn 180°). The parameter evaluated in the walk back phase is the cadence, because the person is not facing the Kinect sensor and therefore the skeleton data is very noisy. In the last phase, turn-to-sit (the subject turns and sit on chair) which is the time required for the movement. The time required for the entire fall risk test is also computed [133].

4 Evaluation of Fall Detection Algorithms

The related works discussed in last section, used different fall detection algorithms on depth images, which were either based on a fixed threshold value or an adaptive threshold to detect fall. While others used, threshold based wearable device to generate fall alert and then used some predefined classification on the images such as Support Vector Machine (SVM) to confirm the fall event. The following sub-sections evaluates different fall detection algorithms employed with the thresholding's used to classify human falls.

4.1 Fixed Thresholding Based Techniques

For fall detection systems that is based only on depth images, the thresholds are basically the height of the subject from ground plane or centroid height and velocity or speed. The combination of these parameters in different order are used to detect fall. The height thresholds used in the related works varies from 0.1 to 0.6 m, via as the average height or thickness of the subjected detected on floor by the sensor is approximately 0.4 m. The velocity deviation considered to bias the decision varies from −1 to 2 m per second. These algorithms do not simply depend on the threshold to predict fall event, rather reliable detection of the joints and pattern of different activities are also considered. This helps to improve the detection rate and reduce the false alarm ratio.

A study that is based on skeleton data extracted from Kinect sensor used two algorithms for fall detection. The first algorithm used only skeleton position data and determines fall based on a single frame. The distance between each joint to floor is computed and fall is detected if the maximum distance is lower than a threshold value. The second algorithm calculates the vertical velocity of each joint to floor plane over many frames. The velocities of all the joints are averaged and if the average velocity (the negative downward velocity) is lower than a threshold of −1 meter per second, a fall is detected [103].

Another study used the human centroid height from floor plane to detect falls that are not fully occluded. If the activity is completely occluded, the body velocity prior to occlusion is analyzed to detect fall. Centroid height is the distance from the 3D centroid to the ground plane and the body velocity is the centroid displacement over a one second period. The threshold for the centroid height and velocity was determined through a training data which consists of daily activities (with some occluded activities) like walking, sitting and crouching down. The centroid height (D_{train}) and the body velocity (V_{train}) computed from the data recorded with Kinect was used to determine the two threshold from the mean value and standard deviation with 97.5% confidence interval [102]. The minimum centroid height ($T_{D_{\min}}$) and the maximum body velocity ($T_{V_{\max}}$) thresholds computed from the Eqs. (1) and (2) are 35.8 cm and 0.63 m/s respectively.

$$T_{D_{\min}} = \left(\overline{D}_{train}\right) - \left(1.96\sigma_{D_{train}}\right) \tag{1}$$

$$T_{V_{\max}} = \left(\overline{V}_{train}\right) + \left(1.96\sigma_{D_{train}}\right) \tag{2}$$

Fall detection algorithm in another study, used extracted raw depth data from the sensor for preprocessing and segmentation to prepare the data for the next steps. The algorithm then identifies, splits and classifies the different clusters of pixel in the frames and identifies human subject. Once the human subject is identified, it is then tracked and height information is evaluated to detect fall if the distance to floor goes below 0.4 m [117].

The proposed fall detection algorithm in [105] first detects the head position and computes its vertical speed. Fall is detected if the head speed and the head (height) satisfies the falling condition or otherwise if the centroid speed and the centroid (height) satisfies the falling condition. The fixed thresholds for the head to satisfy the falling condition is its vertical velocity higher than 2 m/s and distance from head to ground (height) less than 0.5 m. For the second condition, the threshold for centroid speed is 1 m/s and centroid height is 0.5 m.

In another algorithm, the silhouette of the moving individuals in each depth image is obtained by background subtraction and floor plane was estimated by v-disparity map [106]. The characteristics of the shape of the moving object was described by an ellipse which is computed from a set of moment functions. The centroid of ellipse and the angles between the ellipses and floor is computed and it is then converted to real world coordinates. Fall is detected when the distance from the centroid of the human body to floor and the angles between the ellipses and the floor are below some threshold. The distance threshold is 0.5 m and the angle threshold are 45°.

A 3D bounding box based fall detection algorithm used the width, height and depth as the box parameters. These parameters are estimated as the differences of the maximum and minimum points on x, y and z dimensions. The concept used is that during a fall, the height of the box will be decreasing, and the width and the depth will be increasing for lateral fall or vice versa for forward or backward fall. The parameters used for fall detection are the first derivative (speed) of the height (V_h) and the width-depth composition (V_{WD}). The other parameter is the real-world y-coordinate of the top left vertex of the bounding box or simply the y-coordinate of head centroid (V_y). At the beginning of the algorithm, the speed values (V_y and V_{WD}) are checked whether it is greater than their thresholds for a time (T_h and T_{WD}) interval of n frames. Here n is the selected number of frames in the sampling window (SW). The parameter (V_y) is also checked if it is less than a threshold (T_y). Then the algorithm will start tracking V_y to identify any inactivity of the subject. If the speed of V_y is less than the T_h threshold for 10 frames, fall is confirmed. For experimental, the following thresholds are used. The thresholds for T_h is 1.1, T_y is between 1.4 to 1.7, T_y is 0.5 and the number of frames in SW is between 5 to 8.

4.2 Adaptive Thresholding Based Techniques

The adaptive threshold used in one study was calculated by multiplying the first computed head height with 0.25, this made the method adaptive to the person with different heights [110]. If the head height is lower than the threshold value, then the centroid height is used as second judgement or else the threshold value, the centroid height and centroid position is not considered. A fall is detected if the head height and the centroid height are lower than the threshold value at the same time.

Another study refers an abnormal action if any one of the distances from hip center, head or neck to the floor plane is lower than a fall threshold (DT_{fall}) and any of their vertical velocities with respect to floor are larger than a velocity threshold (*VT)* at the same time [109]. The alert is triggered, if the distance between the head and the floor are lower than the threshold ($DT_{recover}$) for a period of time. Here the threshold DT_{fall} and $DT_{recover}$ are adaptive to the height of the subject. The fall threshold DT_{fall} is 1.5 times the distance between head and neck (height from head to neck) as shown in Eq. (3).

$$DT_{fall} = 1.5 \times (\text{Distance from head to neck}) \tag{3}$$

The recover threshold ($DT_{recover}$) is three time of DT_{fall} as shown in Eq. (4).

$$DT_{recover} = 3 \times (DT_{fall}) \tag{4}$$

The vertical velocity with respect to floor is defined in Eq. (5).

$$V_n = \frac{(D_n(joint, floor) - D_{n-1}(joint, floor))}{T} \tag{5}$$

Here, n and $n - 1$ is the nth and (n – 1)th frame and T is the frame period.

Another study used three velocity features and a head to hip height difference for fall detection, where the velocity features used adaptive thresholds. An adaptive head velocity threshold was used as a first step to detect abnormal head movement [108]. The threshold is adaptive to the person's height (h) as follows.

$$\text{Head\_threshold} = 0.6 \times \left(\frac{\sqrt{2h}}{2}\right) \tag{6}$$

The other two adaptive thresholds are hip horizontal velocity (V_hip_h) and hip vertical velocity (V_hip_v) thresholds which are also adaptive to the height of the subject. The two thresholds are:

$$\text{V\_hip\_h} = 0.55 \times \left(\frac{h}{4}\right) \tag{7}$$

$$\text{V\_hip\_v} = 0.7 \times \left(\frac{h}{4}\right) \tag{8}$$

4.3 Fixed and Adaptive Thresholding Based Techniques

A combination of fixed and adaptive threshold is used in one study, where a stationery threshold of 0.3 m is used as a threshold to detect fall if the floor plane is detected [104]. If floor plane is not visible another algorithm checks if the y-coordinates of the joints are below a given threshold to detect the fall event. This algorithm uses a kind of adaptive threshold which is simply 0.7 times the difference of y-coordinate of head and right-ankle (which depends on the height of the user). A fall is detected if the difference of absolute value of previous distance from y-coordinate of head to y-coordinate of right-ankle (at the time adaptive threshold is calculated) with any new distance between the same two joints are greater than the adaptive threshold calculated and if the y-coordinate of other joints are also below a given threshold. The joints considered in this study are head, shoulder-center, hip-center, right-angle and left-ankle.

4.4 Fusion of Thresholding and Classifier Based Techniques

This section discusses on the fall detection algorithms employed in systems that used only depth images with classifiers and those that used an initial device to identify any potential fall. Fall detection systems based on some sort of classifier, either runs the algorithm on some features representing human in depth image or directly on the depth image.

An unobtrusive fall detection proposed in a study used only depth images and person was detected on the basis of the depth reference image [124]. A low computational method is demonstrated for updating the depth reference image. The ground plane is extracted using v-disparity images, Hough transform and RANSAC algorithm. Fall is detected using a classifier trained on features of the human subject from depth images and point clouds.

A deep learning classifier for fall detection based on the infrared sensor measurement mainly focused on statistical properties as generalization. The proposed deep learning classifier consists of 5 hidden neurons as oppose to 15 hidden neurons in neural network. The structure of the proposed deep learning includes a feature selection based on Gram-Schmidt orthogonalization and NPCA block for transforming the raw data into a non-linear manifold [122].

The automated fall detection approach using only depth data presented in [121] is based on shape features and improved machine learning. The algorithm consists of shape-based fall characterization and a learning based classifier to identify human fall from other daily activities.

The rest of the depth map based studies, basically depends on an alert from the wearable device to start fall confirmation process from depth images. The wearable devices contain an accelerometer, and/or a gyroscope and the threshold used varies from 2.5 to 3 g. After the threshold is reached from the wearable device data then either a classifier is used to confirm if the subject is lying on floor or combination of height and velocity or height with a threshold is used to detect human fall.

5 Challenges and Issues in Existing Fall Detection Systems

This section discusses the main challenges that researchers are facing in developing reliable fall detection systems due to the limitation of the technologies. The major issues and pitfalls of the existing fall detection systems from all the approaches are addressed in this section.

5.1 Issue of Synchronization Between Devices

A recent trend in fall detection systems incorporates a wearable fall detector with vision based systems to improve the accuracy and to cover the areas which is not visible to the vision based device. The wearable device is sometimes configured to generate potential fall alert to the vision based mechanism that will confirm the fall event. The wearable devices are usually wirelessly connected to the main system using Bluetooth which has higher sampling rate than that of the RGB or depth cameras. In any case, the lack of synchronization between the wearable and the vision based device can cause the system to miss important timing to identify potential fall activity.

5.2 Issues with Data Fusion from Multiple Sensors

There are several issues arising in systems due to data fusion from multi-sensors for fall detection. Data fusion techniques combines data from more than one sensor and related information from associated databases to achieve maximum accuracies than using a single sensor [134, 135]. It includes issues from reliable data measurement, data communication to reliable data analysis [136].

Generally, data fusion has several advantages including improved data authenticity but there are number of issues that made it challenging [137]. The challenges in data fusion for fall detection that should be analyzed and considered before developing a fall detection frameworks includes data correlation, conflicting data, processing framework, computational power and increase in overall cost of the system [138]. Conflicting data, is meant for the issues of interpreting similar activities differently by the unrelated sensors during monitoring process. The data fusion strategy can differ from the approaches and sensors involved. Most of the current systems analyze data for each sensor component separately and apply the fusion as a final step [136]. In some cases, the raw sensor data without any preprocessing are send to a common framework for processing. The data are also processed in parallel from different sensors and it is fused with another unrelated sensor. A careful decision is required depending on the sensors used to avoid unnecessary complication in the fusion algorithm and to reduce the computation time. The computational cost will also increase due to the additional amount of data collected by the sensor.

5.3 Lack of Accuracy Due to Off-Line Training

Those studies that employed machine learning to classify human fall with off-line training data are subject to misinterpret from the differences in background color and surrounding objects. Since the background and any surrounding lighting condition of the off-line database may be completely different from the actual users' environment. This can degrade the classification accuracy with most of the machine learning algorithms. It is likely that a learning algorithm will be biased for an input data, if it is trained from different training datasets. Thus, a learning algorithm can have a high variance for any particular input data, in case if it had shown different output values when trained on another dataset. Generally, this is about the prediction error of the classifier which is associated with the amount of bias and variance of the learning algorithm which often requires tradeoff [139]. The number of the true data and their complexities together with the dimensionality of the input space can further complicate the learning process. In addition, the difference in between the two-setups including the placement of the camera can have significant impact of the performance.

5.4 Image Extraction Timing for Fall Confirmation

Some of the fall detection algorithms that is based on machine learning to authenticate the fall event uses depth image extracted at the time when a potential fall is detected. The timing when the depth image is extracted and fed to the machine learning classifier has a significant impact on the accuracy. Those classifiers simply check for a lying posture on the floor which highly depends on a clear appearance of the subject. Therefore, the potential fall activity alert that starts the fall authentication should take care of the exact time when the subject completely rests on floor. Other than that, any occlusions present in the depth image which is fed into the classifier can degrade the performance. As a result, the performance of such algorithms highly depends on the accuracy of the potential fall alert mechanism and robustness of the classifier.

5.5 Privacy Concerns

This is in-fact the main reason for the rejection of non-invasive vision based devices that used RGB camera and utilize live video recording. Generally, users would not like to be watched or their private living be recorded for any purpose. This could be the main challenge if developing a non-invasive human fall detection system using RGB cameras.

5.6 Hardware Limitations

5.6.1 Battery Life of Wireless Devices

Stand-alone devices with wireless communications powered by batteries are prone to battery life issues. The device itself can worse the issue, if not properly designed. Power-saving strategies are required at software and circuit level to minimize the power consumption. Standby capabilities and limiting the radio transmissions are the main approaches to implement power-saving. Implementation of power saving techniques can arise other issues such as wakeup delays from standby mode and time required to start the defined process can lead to loss of critical information. The data from wireless devices build with accelerometers for fall detection or prediction also need to be toughly analyzed in-order to fine-tune the circuitry and the software behind the system to limit the radio activity. The limitation of the currently available battery technology is also one of the technological barrier for remote monitoring systems relying on wearable sensors [140].

Investigations showed that accelerometer based wearable devices, often use orientation of the sensor for fall prediction which requires a gyroscope for accurate detection of the sensor's current orientation. Since the acceleration data alone is not enough to reliably evaluate the actual orientation of the device [141], gyroscopes are often used along with accelerometers. The use of gyroscope improves the performance of the system at the cost of an additional device, consuming extra power.

5.6.2 Limitation of Smartphones

Smartphones based fall detectors are prone to generate problems, simply because they are not intended for fall detection [142]. The functionality of the in-build accelerometer, the features of the operating system and the sensing architecture of the smartphones are not initially intended for such application, especially for a real-time fall detection [138]. Therefore, it is very likely that such fall detectors may behave differently on different phones depending on the smartphone architecture. Such fall detectors will also be subject to the processing capabilities and battery powers of the smartphones [61]. Furthermore, the accuracy of fall detection may dramatically degrade depending on where the user places the phone. For an example, if the system is designed for users to place the phone on waist and if the user mistakenly places it in his/her pocket, then the accuracy may be very low.

5.6.3 Limitation in Wireless Transmission

Wearable based fall detectors often use wireless communication including Bluetooth and ZigBee for communicating between the wearable gadget and the main system.

The technologies behind these are subject to many restrictions inheriting from the limitations in wireless communication including interferences. The common issues in using Bluetooth wireless communication accounts for the capacity limitation, coverage limitation and power management. Many studies [54, 56, 143–145] had chosen Zigbee technology due to its advantages over Bluetooth technology. Zigbee is low cost and low powered, but the limited coverage and the replacement cost of the components are the main disadvantages. In addition, any failure in communication between the device and main system may often require technical assistance, thus making the entire system unstable and unreliable. This issue can be common for systems with components from different vendors and systems that are poorly designed.

5.7 Performance Degradation Due to Simulated Activities

Most of the real-time based fall detection system and in-fact almost all the related works are tested and validated with simulated activities. Simulated activities are far different from real life activities, especially when real life activities of elderly are simulated by healthy volunteers. As a result, the evaluation of such systems is limited, and it cannot achieve the proposed detection rate from simulated falls in real world. This could have a higher impact for velocity based algorithm [103], especially if the actual falls have shorter duration than the simulated falls.

5.8 Response Time

Response time of the sensor or camera and the device can play an important role in degrading the performance of the systems. Apart from this, the effectiveness of the fall detection algorithm (including the thresholds employed, flow and any external triggers used) can also affect the response time of the system and thus degrade the performance.

5.9 Disturbance in the Environment

The motion from the visitors [146] is also an issue for non-invasive sensor based systems, apart from the disturbance of obstacles in the scene, especially radar based approaches. To delineate such noises, additional measurements are very often required which will increase the computational cost.

6 Recent Developments and Future Directions

The three basic approaches used to develop human fall detection systems are wearable based devices, ambient based devices and vision based devices. In this work, depth map based devices are categorized under vision based approach as shown in Fig. 1. The studies representing the three basic approaches are basically structured to solve the drawbacks of one another. For an example, ambient based devices used to solve the issues in wearable based devices and wearable devices also solves some of the problems that ambient sensors failed to handle. Before the advent of cheap depth sensors in the market, vision based devices using RGB camera used to take care of the main issue in wearable and ambient based devices with even higher accuracy at the cost of expensive systems and setup. Even with RGB cameras, the concerns arising regarding the acceptability and reliability of the fall detection systems are not limited rather added its own drawbacks such as capturing and recording of color videos leading to privacy concerns. Additionally, the cost of the systems, camera calibrations, requirement of adequate lightening and setup are common issues.

The advent of cheap Red-Green-Blue-Depth (RGBD) cameras, has paved way to the development of novel systems to overcome the limitations of these previous works [125, 147]. The cheap depth sensors such as Microsoft Kinect sensor, can extract depth information of the objects in the scene even with low lighting condition. The auto calibration capability and other features of the sensor can negligibly reduce the issues concerning with RGB cameras. One of the main advantage of the Kinect sensor is that it can be place in certain places according to user requirements [132], unlike the complex installation procedure of some RBG fall detection systems. It is also worth noting that by using only the depth images it can preserve the privacy of users [132].

The first sub-categories in depth sensor based approach used joint position or measurements and its movement with thresholding for fall detection. The second sub-category used fusion of wearable and depth maps with machine learning or joint position for fall detection. The third sub-category employs machine learning or other classifier only on depth images. The research studies based on fusion of an initial devices and depth sensor is not very relevant, since their system design and performance are all subject to the drawback of wearable devices which is regarded as main causes of rejection of fall detection systems. Wearable devices are mainly rejected due to the inconvenience in carrying them during daily life activities [138]. In addition, they used the wearable device to generate any potential fall movement and to start the depth image based classifier to confirm fall. In such cases the capabilities of the depth sensor are not fully exploited and thus the actual accuracies that could be achieved are not realized. Generally, the overall performance of such systems solely depends on the effectiveness of the wearable device to identify potential fall movements.

In some of the studies, detection does not always depend on the wearable device because in certain situations like while changing cloths it is not possible to wear the device [125]. Therefore, in such cases the systems depend on the depth sensor.

This requires a proper time synchronization between wireless initial device sampling rate and depth sensor frame rate. It is impossible to access and control the Kinect embedded clock [148]. But it is important to synchronize the Kinect sensor and main system and the wireless initial device in order to properly integrate such fall detection systems which depend on one another and requires switching of fall authentication process between devices.

The other two approaches in depth map based hierarchy depends only on the depth image generated from the depth sensor as illustrated in Fig. 2. From the review of literature, it was also found that exploring the depth information alone can minimize the issues faced by the previous work for fall detection. The only issue that cannot be dealt is the limitation of the Kinect sensor's viewing spectrum, which can be solved using more than one sensor depending on the coverage requirement.

Previous works had used different techniques on the depth image to classify human fall from other activities of daily life. Some of the studies used extracted human joint measurement and movement over time to identify human fall. While others used machine learning or classifiers either on the depth image or extracted human features to detect human fall. The use of machine learning classification can have many problems apart from the computational cost and have complex implementation when compared with threshold based approaches [149]. As discussed, off-line training can degrade the accuracy and the time when the image is extracted can play an important role in identifying the lying posture. They used different algorithms to segment human subject from the depth image. Some works, developed their own preprocessing directly on the raw data, while it is not possible to achieve the established auto calibration of the Kinect sensor with manual preprocessing, even though it was not primarily developed for fall detection. If any developed preprocessing cannot auto calibrate when a subject enters and exit into the view of the sensor, then fall detection algorithm working on top of the preprocessing will not receive adequate information to make an accurate decision.

The approaches that is based only on the depth images and used joint measurements instead of classifiers basically use the distance of human joints from floor plane and their vertical velocities to classify and authenticate human fall from other activities of daily life. Various joints such as head joint height from floor, centroid height and their respective velocities are fed into an algorithm with some thresholding to identify any falling action. With this approach, some fall detection algorithms cannot work in case of occlusion, because it cannot calculate the distance to the ground [105]. The use of joint height in fall detection shows good performance for falls ending on the floor but it has failed if the end of the fall is occluded behind furniture [102]. It is also found that it could be solved using velocity just before the occlusion.

Basically, most of the related works used selected skeleton joints or features of the subject with a predefined algorithm for fall detection. The algorithms either uses a fixed or adaptive threshold within a flow to make the decision. One of the study used [126], a statistical approach with features using a Bayesian framework. It is very clear from the literature that, a lot more effort still remain, to device algorithms to follow the changes in the selected key features of the subject to fully utilize the capabilities of the depth sensor. It was found that a low-computational algorithm with statistical analysis of the key features of the subject can significantly minimize the issue of obstacles blocking the view of the subject. This could on the other hand, make the system more stable especially by avoiding the computation hungry machine learners and other classifiers.

It is also worth to be noted that none of the related works so far had utilized any fall risk level estimation during fall detection. Studies conducted on fall risk assessments were primarily aimed to identify potential fall risk patients for nursing homes or hospitals. This was achieved either through questionnaires or using sensors to detect likely physical weakness of the patients that may cause falls. This was then used to categorize patients with high fall risks or low fall risks to provide better healthcare and avoid fall injuries. Incorporating a robust fall risk level estimation protocol within the fall detection algorithm to adapt appropriate parameters depending on the movement of the user can improve the fall detection accuracy greatly. Since fall risk factors can help to adapt the fall detection process depending on the risk level of the users. This is because the nature of fall and characteristics of other activities of daily life differ with fall risk levels. For users with high risk of falls, the fall detection algorithm could switch to intensive detection process and for users with low fall risks, the algorithm could track changes after a gap, thus reducing computational costs. Since fall risk level of the user also changes overtime, such an incorporation could make any fall detection system to outperform over other available systems.

7 Performance of the Related Works on Fall Detection

This section demonstrates the performance of the related works using confusion matrix performance measures. Apart from the three basic confusion matrix measures, the other parameters considered in this evaluation are precision, F-score, Mathews correlation coefficient (MCC), error rate and miss rate.

The collected original performance measures and confusion matrix data for the related works are illustrated in Table 2. Out of the fourteen studies only six studies provided confusion matrix data. Even though, some of the performance measures does not give fair values for comparison between imbalanced data sets, parameters such as accuracy and MCC are affected only in extreme cases [150]. The following performance measures are used for the comparison.

Precision or Positive Predictive Value (PPV) is a ratio of correctly predicted positive observations to the total predicted positive observations. The best precision is a value of 1.0 (one) and the worst is a zero value. Precision is calculated by dividing the correct positive prediction and the total number of positive predictions as shown in Eq. (9).

$$PPV = \frac{TP}{TP + FP} \tag{9}$$

Table 2 Collected performance measures from selected previous studies

No.	Study/approach			No. of				Accuracy	Sensitivity	Specificity	Precision	F-score	MCC	Error rate	Miss rate
				Simulated falls	Detected falls	Simulated non-falls	Detected non-falls								
1.	[104]	Distance	2 m	4	2	–	–	50							
			2.5 m	4	3	–	–	75							
			3 m	4	4	–	–	100							
			3.5 m	4	4	–	–	100							
2.	[112]		Image coordinate	40	31	32	32	87.5	77.5	100	1	0.87	0.78	0.125	0.225
			World coordinate	40	37	32	32	95.8	92.5	100	1	0.96	0.91	0.042	0.075
3.	[102]			25	24	54	54	98.73	96	100	1	0.98	0.97	0.013	0.04
4.	[105]			30	29	31	31	98.4	96.67	100	1	0.98	0.97	0.016	0.03
5.	[114]			–	–	–	–	76–87	68–83	89–95	–	–			0.32–0.17
6.	[126]			26	26	61	59	97.7	100	96.72	0.93	0.96	0.95	0.023	0
7.	[107]			–	–	–	–	91.49	90	92.59	0.93	0.91			0.1
8.	[119]			100	98	100	97	97.5	98	97	0.97	0.98	0.95	0.025	0.02
9.	[120]			–	–	–	–	71.4	75	–	0.74	0.74			0.25
10.	[133]			–	–	–	–	99	99	100					0.01
11.	[132]			–	–	–	–	77	91	92					0.09
12.	[131]		SVM	–	–	–	–	98.29	99.41	97.24	0.97	0.98			0.01
			K-NN	–	–	–	–	99.15	99.41	98.90	99	0.99			0.01
13.	[125]		SVM	–	–	–	–	90	100	80	0.83	0.91			0
			Accelerometer + SVM on depth	–	–	–	–	98.33	100	90	0.91	0.95			0
14	[108]			120	109	120	118	94.58	90.83	98.3	0.98	0.94	0.89	0.054	0.85

F-score is a measure of accuracy in terms of precision and recall. It is simply the harmonic means of the precision and recall. It is calculated using the formula given in Eq. (10). The best value of F-score is when it approaches one and it is worst at zero.

$$F\text{-}score = \left(\frac{2TP}{2TP + FP + FN}\right) or \left(2 \times \frac{Precision \times Recall}{Precision + Recall}\right) \tag{10}$$

The MCC is a measure of quality of binary two-class classification used in machine learning [151]. It is a balanced measure which can be used for irregular class sizes and it is a correlation coefficient between the observed and the predicted observations. The best prediction is a coefficient of $+1$ and -1 represents the worst prediction. MCC is calculated using the following formula.

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{11}$$

Misclassification rate or error rate is a measure of how often it is wrong. It is equivalent to one minus the accuracy. It is calculated by dividing all the incorrect prediction over the total number of data as shown in Eq. (12). The best error rate is a value of zero while the worst is a value of one.

$$Error\, rate = \frac{FP + FN}{TP + TN + FP + FN} \tag{12}$$

Miss rate or False Negative Rate (FNR) is a ratio of positive class which gave negative outcomes over the given total positive class or simply one minus sensitivity. It is calculated by dividing the positive test results predicted as negative by the total positive observations as shown in the following equation.

$$Miss\, rate = \frac{FN}{FN + TP} \tag{13}$$

Table 2, shows the results collected and some self-computed measures from the performance information available from previous studies. Some of the performance measures are directly given in the respective study while the others are computed from the available information. All the studies were reviewed in previous sections.

The first study [104] of Table 2, which is based on height of the subject, presented the accuracy of the algorithm at different distances (how far the subject is to the sensor). The average accuracy for the given four distances is 81.25%. This study does not give any other information except the number of simulated fall events and the number of detected fall events. Therefore, no other performance measure can be computed for this work.

The third [102] and the fourth [105] study which was based height and velocity showed a higher MCC with lowest error rate. The accuracy of the third, the fourth and the proposed approach is 98.73, 98.4 and 98.3 respectively. But both of them failed to identify one fall event which generated miss rates.

The study (in 8th row) which is based on height and angle of torso also generated higher error rates and showed some miss rates as well. On the other hand, the study in 6th row showed zero percentage of miss rates. It is the only study that was based on statistical approach, but they were focused on a Bayesian framework.

The second study [112] shown in the same table is based on the height and performance measures are given for two approaches as 3D sensor using image coordinates and using world coordinates. This study classified all non-fall events correctly and achieved 100% specificity.

The study presented in 5th row [114] of Table 2, was a bounding box based approach which is counted among the studies that possess highest miss rates. The study demonstrated in 7th row [107] which is based on height, velocity and SVM also showed slower performances in all the available measures. Similarly, the SVM based approach in 9th row [120] of the same table showed even lower performances in all the measures available. The approaches in the 10th [133] and the 11th row [132] employs wearable device. The study in 10th row achieved the highest accuracy, but it is to be noted that it is a wearable based method has the drawbacks of wearing devices. But the study in 11th row showed a slower performance in all the measures. The last two approaches presented in the 12th [131] and the 13th [125] row also employs wearable devices for fall detection along with the depth sensor. The study in 12th row, showed results for both SVM and K-NN classification where all of them possessed some miss rates. The study in 13th row, showed performance measures for SVM only on depth maps and SVM on depth maps with accelerometer.

The study [108] in the 14th row of the same table, showed lower performances in all the measures except for specificity and precision. Because this study failed to identify only two non-fall events from a larger sample size than the proposed approach. This study failed to identify two non-fall events from a total of 120 simulations. But this study failed to identify 11 fall events out of 120 simulated events. Identification of fall events are more important than misclassification of non-fall events. Because fall detection system should not miss any fall event and the first goal is to correctly classify all the fall events. A false alarm from misclassification of non-fall event is acceptable but not a misclassification of a fall event. Simply meaning that all the fall events should be correctly identified, and alarm generated even though it includes few non-fall events. This study failed to identify more fall events than non-fall events leading to high error rate.

As far as the f-score is concerned, all the approaches in row 3, 4, 8 and 12 that achieved a higher f-score had a miss rate of minimum 0.01. Similarly, all the studies in row 12, 10, 4 and 3 that claimed a higher accuracy also possess miss rates. The studies that showed higher precision in row 2, 3, 4, 8 and 12, all failed to identify at least one fall event.

Some graphical plots can help to better understand the performance measures described above. Such as the Receiver Operator Characteristics (ROC) curves, which shows the trade-off between false positive and true positives rates. ROC curves are

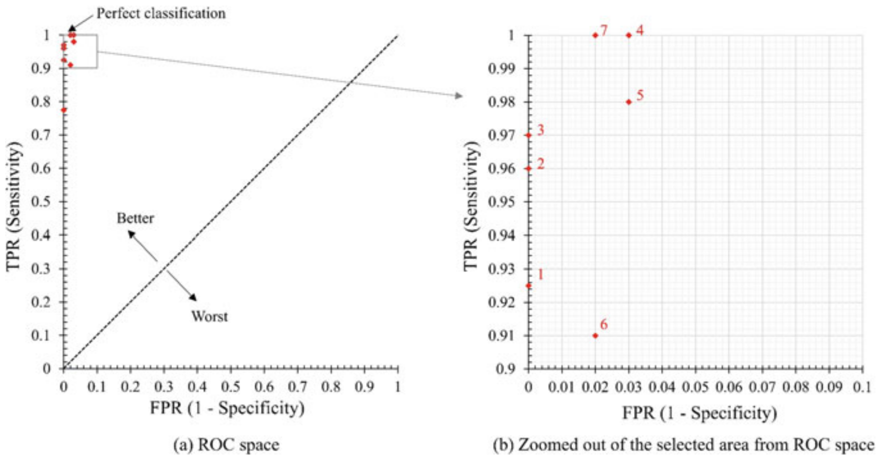


Fig. 3 ROC space of all the balance studies

another way of evaluating performance measures of classifiers, besides confusion matrix.

Due to the lack of performance data from related works the following shows only a ROC space and a miss rate vs error rate space to graphically illustrate some performance measures described in Table 2 after balancing. As mentioned, ROC space shows the relative trade-offs between the true positives (benefits) and false positives (costs). It is plotted by defining false positive rate (FPR) which is equivalent to (1-specificity) on x-axis and true positive rate (TPR) which is equivalent to sensitivity on y-axis. In other words, ROC space is simply a performance graphing of sensitivity versus (1-specificity) for different classifiers, where each classifier has one pair (1-specificity, sensitivity) corresponding to a single point in the plot. In this space, the perfect classifier will be at point (0, 1) as shown in Fig. 3. Since a perfect classifier will have no false positive errors with all true positives. This also indicates that any point in ROC space is better than the other if it is to the northwest of the other.

Figure 3 part (a), illustrates the ROC space plot of the performance of the balanced related works in Table 2. The part (b) of the same figure shows a zoomed out of the space representing the most perfect classification, because except one, all the points representing the other studies are appearing in that region. For easy visual inspection of the best performer that area is zoomed out in part (b) of Fig. 3.

The numbering besides each of the points in the part (b) of Fig. 3, is showing the corresponding balanced studies in Table 2. Similarly, Fig. 4, demonstrates the miss rate vs error rate space of the same performance data in Table 2. In this space the best performance is at point (0, 0), whichever is approaching the origin point possesses lower rates. Figure 4, is plotted by taking the error rates on x-axis and miss rates on y-axis. This figure simply illustrates the same numerical measures (miss rates and error rates) in Table 2 (after balancing).

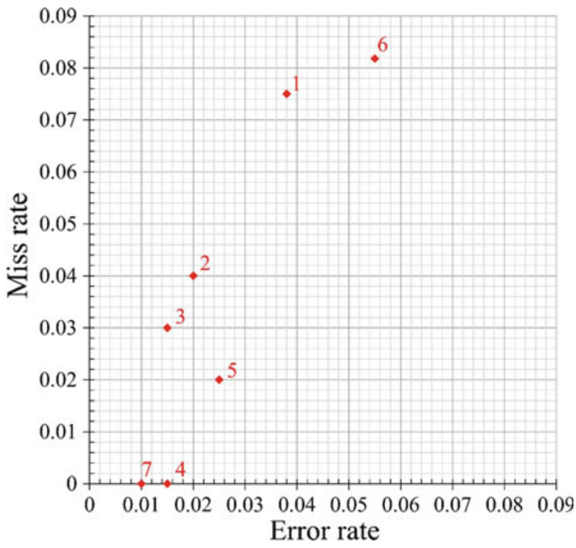


Fig. 4 Miss rate versus error rate space

8 Conclusion

This study presented a review of different approaches used for human fall detection including the conventional methods to the recent developments. Studies representing the three basic approaches (wearable, ambient and vision based approaches) for fall detection were briefly reviewed including the studies based on depth sensor. The underlining features and methods along with the algorithms used for fall detections were described. The merits, reliability and pitfalls in the existing approaches were also briefly highlighted in different sections. A brief discussion on the depth sensor based approaches were presented and then a research gap study was demonstrated. Finally a performance comparison of selected previous studies were presented. There are many future works, which can be done to improve such fall detection system's ability to accurately classify human fall from other activities of daily life. Most importantly further experiments are required to see if increasing the gap for velocity calculation can improve the performance of the system in sensing velocity changes. Additional work is required to derive a better method to classify human fall and lying on floor from standing, since these are the two activities that are very similar to one another in terms of the classification algorithms used.

References

1. Griffiths, C., Rooney, C., Brock, A.: Leading causes of death in England and Wales–how should we group causes. Health Stat. Quarterly **28**(9) (2005)
2. Baker, S.P., Harvey, A.: Fall injuries in the elderly. Clin. Geriatr. Med. **1**(3), 501–512 (1985)

3. Stevens, J.A., Corso, P.S., Finkelstein, E.A., Miller, T.R.: The costs of fatal and non-fatal falls among older adults. Injury Prevent. **12**(5), 290–295 (2006)
4. Kannus, P., Sievänen, H., Palvanen, M., Järvinen, T., Parkkari, J.: Prevention of falls and consequent injuries in elderly people. Lancet **366**(9500), 1885–1893 (2005)
5. Bourke, A., O'brien, J., Lyons, G.: Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. Gait Posture **26**(2), 194–199 (2007)
6. Kangas, M., Konttila, A., Lindgren, P., Winblad, I., Jämsä, T.: Comparison of low-complexity fall detection algorithms for body attached accelerometers. Gait Posture **28**(2), 285–291 (2008)
7. Gostynski, M.: Prevalence, circumstances and consequences of falls in institutionalized elderly; a pilot study. Sozial-und Praventivmedizin **36**(6), 341–345 (1990)
8. Gurley, R.J., Lum, N., Sande, M., Lo, B., Katz, M.H.: Persons found in their homes helpless or dead. N. Engl. J. Med. **334**(26), 1710–1716 (1996)
9. Lin, L.-J., Chiou, F.-T., Cohen, H.H.: Slip and fall accident prevention: a review of research, practice, and regulations. J. Saf. Res. **26**(4), 203–212 (1996)
10. Lord, C.J., Colvin, D.P.: Falls in the elderly: detection and assessment. In: Proceedings of the Annual International Conference of the IEEE 1991, vol. 13, pp. 1938–1939. IEEE, Engineering in Medicine and Biology Society (1991)
11. Lord, S.R., Sherrington, C., Menz, H.B., Close, J.C.: Falls in older people: risk factors and strategies for prevention. Cambridge University Press (2007)
12. Sadigh, S., Reimers, A., Andersson, R., Laflamme, L.: Falls and fall-related injuries among the elderly: a survey of residential-care facilities in a Swedish municipality. J. Commun. Health **29**(2), 129–140 (2004)
13. Salvà, A., Bolíbar, I., Pera, G., Arias, C.: Incidence and consequences of falls among elderly people living in the community. Med. Clin. **122**(5), 172–176 (2004)
14. Teasell, R., McRae, M., Foley, N., Bhardwaj, A.: The incidence and consequences of falls in stroke patients during inpatient rehabilitation: factors associated with high risk. Arch. Phys. Med. Rehabil. **83**(3), 329–333 (2002)
15. Tinetti, M.E., Williams, C.S.: Falls, injuries due to falls, and the risk of admission to a nursing home. N. Engl. J. Med. **337**(18), 1279–1284 (1997)
16. Almeida, O., Zhang, M., Liu, J.-C.: Dynamic fall detection and pace measurement in walking sticks. In: High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability. HCMDSS-MDPnP 2007, pp. 204–206. IEEE 2007
17. Shany, T., Redmond, S.J., Narayanan, M.R., Lovell, N.H.: Sensors-based wearable systems for monitoring of human movement and falls. Sens. J. IEEE **12**(3), 658–670 (2012)
18. Harrington, L., Luquire, R., Vish, N., Winter, M., Wilder, C., Houser, B., Pitcher, E., Qin, H.: Meta-analysis of fall-risk tools in hospitalized adults. J. Nurs. Adm. **40**(11), 483–488 (2010)
19. Mubashir, M., Shao, L., Seed, L.: A survey on fall detection: Principles and approaches. Neurocomputing **100**, 144–152 (2013)
20. Xu, T., Zhou, Y., Zhu, J.J.A.S.: New advances and challenges of fall detection systems: a survey. **8**(3), 418 (2018)
21. Coahran, M., Hillier, L.M., Van Bussel, L., Black, E., Churchyard, R., Gutmanis, I., Ioannou, Y., Michael, K., Ross, T., Mihailidis, A.: Automated fall detection technology in inpatient geriatric psychiatry: nurses' perceptions and lessons learned. **37**(3), 245–260 (2018)
22. Arshad, A., Khan, S., Alam, A.Z., Ahmad, F.I., Tasnim, R.: A study on health monitoring system: recent advancements. IIUM Eng. J. **15**(2), 87–99 (2014)
23. Luo, S., Hu, Q.: A dynamic motion pattern analysis approach to fall detection. In: 2004 IEEE International Workshop on Biomedical Circuits and Systems, pp. 1–5–8a. IEEE (2004)
24. Yu, X.: Approaches and principles of fall detection for elderly and patient. In: 10th International Conference on 2008 e-health Networking, Applications and Services, HealthCom 2008, pp. 42–47. IEEE (2008)
25. Kovashka, A., Grauman, K.: Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2046–2053. IEEE (2010)

26. Liu, H., Tang, H., Xiao, W., Guo, Z., Tian, L., Gao, Y.: Sequential bag-of-words model for human action classification. CAAI Trans. Intell. Technol. **1**(2), 125–136 (2016)
27. Ryoo, M.S.: Human activity prediction: early recognition of ongoing activities from streaming videos. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 1036–1043. IEEE (2011)
28. Ryoo, M., Chen, C.-C., Aggarwal, J., Roy-Chowdhury, A.: An overview of contest on semantic description of human activities (SDHA) 2010. In: Recognizing Patterns in Signals, Speech, Images and Videos. pp. 270–285. Springer (2010)
29. Bouchard, G., Triggs, B.: Hierarchical part-based visual object categorization. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, pp. 710–715. IEEE (2005)
30. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: Tenth IEEE International Conference on Computer Vision, ICCV 2005, pp. 1395–1402. IEEE (2005)
31. Parameswaran, V., Chellappa, R.: Human action-recognition using mutual invariants. Comput. Vis. Image Underst. **98**(2), 294–324 (2005)
32. Ramanan, D., Forsyth, D.A.: Automatic annotation of everyday movements. In: Advances in Neural Information Processing Systems, pp. 1547–1554 (2004)
33. Rao, C., Shah, M.: View-invariance in action recognition. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, pp. II-II. IEEE (2001)
34. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, pp. 32–36. IEEE (2004)
35. Gibson, M.J.: The prevention of falls in later life: a report of the Kellogg International Work Group on the prevention of falls by the elderly. Dan. Med. Bull. **34**(4), 1–24 (1987)
36. Tinetti, M.E., Speechley, M., Ginter, S.F.: Risk factors for falls among elderly persons living in the community. N. Engl. J. Med. **319**(26), 1701–1707 (1988)
37. Agostini, J.V., Baker, D.I., Bogardus R.S.T., Jr,: Prevention of falls in hospitalized and institutionalized older people. making health care safer: a critical analysis of patient safety practices **43**, 281–299 (2001)
38. Kang, J.M., Yoo, T., Kim, H.C.: A wrist-worn integrated health monitoring instrument with a tele-reporting device for telemedicine and telecare. IEEE Trans. Instrum. Measur. **55**(5), 1655–1661 (2006)
39. Kaluža, B., Luštrek, M.: Fall detection and activity recognition methods for the confidence project: a survey. Population **2020**(2030), 2040 (2010)
40. Nyan, M., Tay, F.E., Murugasu, E.: A wearable system for pre-impact fall detection. J. Biomech. **41**(16), 3475–3481 (2008)
41. Noury, N., Fleury, A., Rumeau, P., Bourke, A., Laighin, G., Rialle, V., Lundy, J.: Fall detection-principles and methods. In: 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2007, pp. 1663–1666. IEEE (2007)
42. Tsinganos, P., Skodras, A.J.S.: On the comparison of wearable sensor data fusion to a single sensor machine learning technique in fall detection. **18**(2), 592 (2018)
43. Sucerquia, A., López, J., Vargas-Bonilla, J.J.S.: Real-life/real-time elderly fall detection with a triaxial accelerometer. **18**(4), 1101 (2018)
44. Khojasteh, S., Villar, J., Chira, C., González, V., De La Cal, E.J.S.: Improving fall detection using an on-wrist wearable accelerometer. **18**(5), 1350 (2018)
45. Broadley, R., Klenk, J., Thies, S., Kenney, L., Granat, M.J.S.: Methods for the real-world evaluation of fall detection technology: a scoping review. **18**(7), 2060 (2018)
46. Alwan, M., Rajendran, P.J., Kell, S., Mack, D., Dalal, S., Wolfe, M., Felder, R.: A smart and passive floor-vibration based fall detector for elderly. In: Information and Communication Technologies, ICTTA'06, 2nd 2006, pp. 1003–1007. IEEE (2006)
47. Sixsmith, A., Johnson, N.: A smart sensor to detect the falls of the elderly. Pervas. Comput. IEEE **3**(2), 42–47 (2004)

48. Sixsmith, A., Johnson, N., Whatmore, R.: Pyroelectric IR sensor arrays for fall detection in the older population. In: Journal de Physique IV (Proceedings), pp. 153–160. EDP Sciences (2005)
49. Tabar, A.M., Keshavarz, A., Aghajan, H.: Smart home care network using sensor fusion and distributed vision-based reasoning. In: Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks, pp. 145–154. ACM (2006)
50. Zhuang, X., Huang, J., Potamianos, G., Hasegawa-Johnson, M.: Acoustic fall detection using Gaussian mixture models and GMM supervectors. In: IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP 2009, pp. 69–72. IEEE (2009)
51. Narayanan, M.R., Lord, S.R., Budge, M.M., Celler, B.G., Lovell, N.H.: Falls management: detection and prevention, using a waist-mounted triaxial accelerometer. In: 29th Annual International Conference of the IEEE 2007 Engineering in Medicine and Biology Society, EMBS 2007, pp. 4037–4040. IEEE (2007)
52. Clifford, M.A., Borras, R.L., Gomez, L.: System and method for human body fall detection. In: Google Patents (2007)
53. Wang, C.-C., Chiang, C.-Y., Lin, P.-Y., Chou, Y.-C., Kuo, I., Huang, C.-N., Chan, C.-T.: Development of a fall detecting system for the elderly residents. In: The 2nd International Conference on Bioinformatics and Biomedical Engineering, ICBBE 2008, pp. 1359–1362. IEEE (2008)
54. Benocci, M., Tacconi, C., Farella, E., Benini, L., Chiari, L., Vanzago, L.: Accelerometer-based fall detection using optimized ZigBee data streaming. Microelectron. J. **41**(11), 703–710 (2010)
55. Lim, D., Park, C., Kim, N.H., Kim, S.-H., Yu, Y.S.: Fall-detection algorithm using 3-axis acceleration: combination with simple threshold and hidden Markov model. J. Appl. Math. **2014**, 1–8 (2014)
56. Doukas, C., Maglogiannis, I., Tragas, P., Liapis, D., Yovanof, G.: Patient fall detection using support vector machines. In: IFIP International Conference on Artificial Intelligence Applications and Innovations, pp. 147–156. Springer (2007)
57. Yuwono, M., Moulton, B.D., Su, S.W., Celler, B.G., Nguyen, H.T.: Unsupervised machine-learning method for improving the performance of ambulatory fall-detection systems. Biomed. Eng. Online **11**(1), 9–19 (2012)
58. Nyan, M., Tay, F.E., Manimaran, M., Seah, K.: Garment-based detection of falls and activities of daily living using 3-axis MEMS accelerometer. In: J. Phys.: Conf. Ser. **1**, 1059–1067. IOP Publishing (2006)
59. Estudillo-Valderrama, M.Á., Roa, L.M., Reina-Tosina, J., Naranjo-Hernández, D.: Design and implementation of a distributed fall detection system—personal server. IEEE Trans. Inf. Technol. Biomed. **13**(6), 874–881 (2009)
60. Lee, Y., Yeh, H., Kim, K.-H., Choi, O.: A real-time fall detection system based on the acceleration sensor of smartphone. **10**, 1847979017750669 (2018)
61. Santoyo-Ramón, J., Casilari, E., Cano-García, J.J.S.: Analysis of a smartphone-based architecture with multiple mobility sensors for fall detection with supervised learning. **18**(4), 1155 (2018)
62. Kangas, M., Konttila, A., Winblad, I., Jamsa, T.: Determination of simple thresholds for accelerometry-based parameters for fall detection. In: 29th Annual International Conference of the IEEE 2007 Engineering in Medicine and Biology Society, EMBS 2007, pp. 1367–1370. IEEE (2007)
63. Bourke, A.K., O'Donovan, K.J., Nelson, J., OLaighin, G.M.: Fall-detection through vertical velocity thresholding using a tri-axial accelerometer characterized using an optical motion-capture system. In: 30th Annual International Conference of the IEEE 2008 Engineering in Medicine and Biology Society, EMBS 2008, pp. 2832–2835. IEEE (2008)
64. Noury, N., Hervé, T., Rialle, V., Virone, G., Mercier, E., Morey, G., Moro, A., Porcheron, T.: Monitoring behavior in home using a smart fall sensor and position sensors. In: 1st Annual International, Conference On Microtechnologies in Medicine and Biology, pp. 607–610. IEEE (2000)

65. Noury, N., Barralon, P., Virone, G., Boissy, P., Hamel, M., Rumeau, P.: A smart sensor based on rules and its evaluation in daily routines. In: Proceedings of the 25th Annual International Conference of the IEEE 2003 Engineering in Medicine and Biology Society, pp. 3286–3289. IEEE (2003)
66. Tamura, T., Yoshimura, T., Sekine, M., Uchida, M., Tanaka, O.: A wearable airbag to prevent fall injuries. IEEE Trans. Inf. Technol. Biomed. **13**(6), 910–914 (2009)
67. Bianchi, F., Redmond, S.J., Narayanan, M.R., Cerutti, S., Lovell, N.H.: Barometric pressure and triaxial accelerometry-based falls event detection. IEEE Trans. Neural Syst. Rehabil. Eng. **18**(6), 619–627 (2010)
68. Purwar, A., Jeong, D.U., Chung, W.Y.: Activity monitoring from real-time triaxial accelerometer data using sensor network. In: International Conference on Control, Automation and Systems, ICCAS'07, pp. 2402–2406. IEEE (2007)
69. Brown, G.: An accelerometer based fall detector: development, experimentation, and analysis. In: EECS/SUPERB, pp. 1–9 (2005)
70. Chen, J., Kwong, K., Chang, D., Luk, J., Bajcsy, R.: Wearable sensors for reliable fall detection. In: 27th Annual International Conference of the 2006 Engineering in Medicine and Biology Society, IEEE-EMBS 2005, pp. 3551–3554. IEEE (2005)
71. Petelenz, T.J., Peterson, S.C., Jacobsen, S.C.: Elderly fall monitoring method and device. In: Google Patents (2002)
72. Patel, S.V., Deepa, A.M.: Human fall detection using MEMS accelerometer. Int. J. Emerging Technol. Adv. Eng. **4**(5), 370–373 (2014)
73. Zheng, J., Zhang, G., Wu, T.: Design of automatic fall detector for elderly based on triaxial accelerometer. In: 3rd International Conference on Bioinformatics and Biomedical Engineering, ICBBE 2009, pp. 1–4. IEEE (2009)
74. Mathie, M.J., Coster, A.C., Lovell, N.H., Celler, B.G.: Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement. Physiol. Meas. **25**(2), R1–R20 (2004)
75. Tcarenko, I., Gia, T.N., Rahmani, A.M., Westerlund, T., Liljeberg, P., Tenhunen, H.: Energy-efficient iot-enabled fall detection system with messenger-based notification. In: International Conference on Wireless Mobile Communication and Healthcare, pp. 19–26. Springer (2016)
76. Baek, W.-S., Kim, D.-M., Bashir, F., Pyun, J.-Y.: Real life applicable fall detection system based on wireless body area network. In: Consumer Communications and Networking Conference (CCNC), 2013 IEEE, pp. 62–67. IEEE (2013)
77. Le, T.M., Pan, R.: Accelerometer-based sensor network for fall detection. In: Biomedical Circuits and Systems Conference, BioCAS 2009. IEEE 2009, pp. 265–268. IEEE (2009)
78. Sudarshan, B., Hegde, R., Satyanarayana, B.: Design and development of fall detector using fall acceleration. In: IJRET (2013)
79. Kangas, M., Vikman, I., Wiklander, J., Lindgren, P., Nyberg, L., Jämsä, T.: Sensitivity and specificity of fall detection in people aged 40 years and over. Gait Posture **29**(4), 571–574 (2009)
80. Shi, H., Zhao, H., Liu, Y., Gao, W., Dou, S.-C.J.S.: Systematic analysis of a military wearable device based on a multi-level fusion framework: research directions. **19**(12), 2651 (2019)
81. Hwang, J.-Y., Kang, J., Jang, Y.W., Kim, H.-C.: Development of novel algorithm and real-time monitoring ambulatory system using Bluetooth module for fall detection in the elderly. In: 26th Annual International Conference of the IEEE 2004 Engineering in Medicine and Biology Society, IEMBS'04, pp. 2204–2207. IEEE (2004)
82. Li, Q., Stankovic, J., Hanson, M., Barth, A.T., Lach, J., Zhou, G.: Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. In: Sixth International Workshop on 2009 Wearable and Implantable Body Sensor Networks, BSN 2009, pp. 138–143. IEEE (2009)
83. Dinh, A., Teng, D., Chen, L., Shi, Y., McCrosky, C., Basran, J., Del Bello-Hass, V.: Implementation of a physical activity monitoring system for the elderly people with built-in vital sign and fall detection. In: Sixth International Conference on 2009 Information Technology: New Generations, ITNG'09, pp. 1226–1231. IEEE (2009)

84. Bourke, A.K., Lyons, G.M.: A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. Med. Eng. Phys. **30**(1), 84–90 (2008)
85. Wang, Y., Wu, K., Ni, L.M.: Wifall: device-free fall detection by wireless networks. IEEE Trans. Mob. Comput. **16**(2), 581–594 (2017)
86. Litvak, D., Zigel, Y., Gannot, I.: Fall detection of elderly through floor vibrations and sound. In: 30th Annual International Conference of the IEEE 2008 Engineering in Medicine and Biology Society, EMBS 2008, pp. 4632–4635. IEEE (2008)
87. Scott, T.E.: Bed exit detection apparatus. In; Google Patents (2000)
88. Erol, B., Amin, M.G.: Radar data cube analysis for fall detection. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2018, pp. 2446–2450. IEEE (2018)
89. Palipana, S., Rojas, D., Agrawal, P., Pesch, D.: Mobile, wearable, technologies, U.: FallDeFi: ubiquitous fall detection using commodity Wi-Fi devices. **1**(4), 155 (2018)
90. Jansen, B., Deklerck, R.: Context aware inactivity recognition for visual fall detection. In: Pervasive Health Conference and Workshops, pp. 1–4. IEEE (2006)
91. Nait-Charif, H., McKenna, S.J.: Activity summarisation and fall detection in a supportive home environment. In: Proceedings of the 17th International Conference on 2004 Pattern Recognition, ICPR 2004, pp. 323–326. IEEE (2004)
92. Töreyin, B.U., Dedeoğlu, Y., Çetin, A.E.: HMM based falling person detection using both audio and video. In: Computer Vision in Human-Computer Interaction, pp. 211–220. Springer (2005)
93. Anderson, D., Keller, J.M., Skubic, M., Chen, X., He, Z.: Recognizing falls from silhouettes. In: 28th Annual International Conference of the IEEE 2006 Engineering in Medicine and Biology Society, EMBS'06, pp. 6388–6391. IEEE (2006)
94. Thome, N., Miguet, S.: A HHMM-based approach for robust fall detection. In: 9th International Conference on 2006 Control, Automation, Robotics and Vision, ICARCV'06., pp. 1–8. IEEE (2006)
95. Auvinet, E., Reveret, L., St-Arnaud, A., Rousseau, J., Meunier, J.: Fall detection using multiple cameras. In: 30th Annual International Conference of the IEEE 2008 Engineering in Medicine and Biology Society, EMBS 2008, pp. 2554–2557. IEEE (2008)
96. Cucchiara, R., Prati, A., Vezzani, R.: A multi-camera vision system for fall detection and alarm generation. Expert Syst. **24**(5), 334–345 (2007)
97. Bosch-Jorge, M., Sánchez-Salmerón, A.-J., Valera, Á., Ricolfe-Viala, C.: Fall detection based on the gravity vector using a wide-angle camera. Expert Syst. Appl. **41**(17), 7980–7986 (2014)
98. Miaou, S.-G., Shih, F.-C., Huang, C.-Y.: A smart vision-based human fall detection system for telehealth applications. In: Proceedings of Third ISATED International Conference on Telehealth 2007, pp. 7–12 (2007)
99. Miaou, S.-G., Sung, P.-H., Huang, C.-Y.: A customized human fall detection system using omni-camera images and personal information. In: 1st Transdisciplinary Conference on 2006 Distributed Diagnosis and Home Healthcare, D2H2, pp. 39–42. IEEE (2006)
100. Chelli, A., Pätzold, M.J.I.A.: A machine learning approach for fall detection and daily living activity recognition. **7**, 38670–38687 (2019)
101. Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J.: Fall detection from human shape and motion history using video surveillance. In: 21st International Conference on 2007 Advanced Information Networking and Applications Workshops, AINAW'07, pp. 875–880. IEEE (2007)
102. Rougier, C., Auvinet, E., Rousseau, J., Mignotte, M., Meunier, J.: Fall detection from depth map video sequences. In: Toward Useful Services for Elderly and People with Disabilities, pp. 121–128 Springer (2011)
103. Kawatsu, C., Li, J., Chung, C.: Development of a fall detection system with Microsoft Kinect. In: Robot Intelligence Technology and Applications 2012, pp. 623–630. Springer (2013)
104. Mundher, Z.A., Zhong, J.: A real-time fall detection system in elderly care using mobile robot and kinect sensor. Int. J. Mater. Mech. Manufact. **2**(2), 133–138 (2014)
105. Nghiem, A.T., Auvinet, E., Meunier, J.: Head detection using kinect camera and its application to fall detection. In: 2012 11th International Conference on 2012 Information Science, Signal Processing and their Applications (ISSPA), pp. 164–169. IEEE (2012)

106. Yang, L., Ren, Y., Zhang, W.: 3D depth image analysis for indoor fall detection of elderly people. Digital Commun. Netw. **2**(1), 24–34 (2016)
107. Tran, T.-H., Le, T.-L., Hoang, V.-N., Vu, H.: Continuous detection of human fall using multimodal features from Kinect sensors in scalable environment. Comput. Methods Programs Biomed. **146**, 151–165 (2017)
108. Zhang, S., Li, Z., Wei, Z., Wang, S.: An automatic human fall detection approach using RGBD cameras. In: 2016 5th International Conference on 2016 Computer Science and Network Technology (ICCSNT), pp. 781–784. IEEE (2016)
109. Bian, Z.-P., Chau, L.-P., Magnenat-Thalmann, N.: A depth video approach for fall detection based on human joints height and falling velocity. In: International Conference on Computer Animation and Social Agents, pp. 1–4 (2012)
110. Yang, L., Ren, Y., Hu, H., Tian, B.: New fast fall detection method based on spatio-temporal context tracking of head by using depth images. Sensors **15**(9), 23004–23019 (2015)
111. Stone, E.E., Skubic, M.: Fall detection in homes of older adults using the Microsoft Kinect. IEEE J. Biomed. Health Inform. **19**(1), 290–301 (2015)
112. Planinc, R., Kampel, M.: Introducing the use of depth data for fall detection. Pers. Ubiquit. Comput. **17**(6), 1063–1072 (2013)
113. Mastorakis, G., Makris, D.: Fall detection system using Kinect's infrared sensor. J. Real-Time Image Proc. **9**(4), 635–646 (2014)
114. Bevilacqua, V., Nuzzolese, N., Barone, D., Pantaleo, M., Suma, M., D'Ambruoso, D., Volpe, A., Loconsole, C., Stroppa, F.: Fall detection in indoor environment with kinect sensor. In: Proceedings, 2014 IEEE International Symposium on 2014 Innovations in Intelligent Systems and Applications (INISTA), pp. 319–324. IEEE (2014)
115. Yajai, A., Rasamequan, S.: Adaptive directional bounding box from RGB-D information for improving fall detection. J. Vis. Commun. Image Represent. **49**, 257–273 (2017)
116. Bansal, D., Alsadoon, A., Prasad, P., Paul, M., Elchouemi, A.: Elderly people fall detection system using skeleton tracking and recognition. **15**(9), 423–431 (2018)
117. Gasparrini, S., Cippitelli, E., Spinsante, S., Gambi, E.: A depth-based fall detection system using a Kinect® sensor. Sensors **14**(2), 2756–2775 (2014)
118. Praveen Kumar, M., Seyezhai, M.S. R.: Kinect sensor based human fall detection system using skeleton detection algorithm. In: Paper presented at the International Conference on Engineering Innovations and Solutions (ICEIS) (2016)
119. Yao, L., Min, W., Lu, K.: A new approach to fall detection based on the human Torso motion model. Appl. Sci. **7**(10), 993–1009 (2017)
120. Alazrai, R., Momani, M., Daoud, M.I.: Fall detection for elderly from partially observed depth-map video sequences based on view-invariant human activity representation. Appl. Sci. **7**(4), 316–334 (2017)
121. Ma, X., Wang, H., Xue, B., Zhou, M., Ji, B., Li, Y.: Depth-based human fall detection via shape features and improved extreme learning machine. IEEE J. Biomed. Health Inform. **18**(6), 1915–1922 (2014)
122. Jankowski, S., Szymański, Z., Dziomin, U., Mazurek, P., Wagner, J.: Deep learning classifier for fall detection based on IR distance sensor data. In: 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), pp. 723–727. IEEE (2015)
123. Li, Y., Banerjee, T., Popescu, M., Skubic, M.: Improvement of acoustic fall detection using Kinect depth sensing. In: 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 6736–6739. IEEE (2013)
124. Kepski, M., Kwolek, B.: Unobtrusive fall detection at home using kinect sensor. In: International Conference on Computer Analysis of Images and Patterns, pp. 457–464. Springer (2013)
125. Kwolek, B., Kepski, M.: Human fall detection on embedded platform using depth maps and wireless accelerometer. Comput. Methods Programs Biomed. **117**(3), 489–501 (2014)
126. Zhang, Z., Liu, W., Metsis, V., Athitsos, V.: A viewpoint-independent statistical method for fall detection. In: 21st International Conference on 2012 Pattern Recognition (ICPR), pp. 3626–3630. IEEE (2012)

127. Kepski, M., Kwolek, B.: Human fall detection using kinect sensor. In: Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013 pp. 743–752. Springer (2013)
128. Gasparrini, S., Cippitelli, E., Gambi, E., Spinsante, S., Wåhslén, J., Orhan, I., Lindh, T.: Proposal and experimental evaluation of fall detection solution based on wearable and depth data fusion. In: ICT Innovations 2015. pp. 99–108. Springer (2016)
129. Kepski, M., Kwolek, B.: Fall detection using ceiling-mounted 3d depth camera. In: 2014 International Conference on 2014 Computer Vision Theory and Applications (VISAPP), pp. 640–647. IEEE (2014)
130. Kwolek, B., Kepski, M.: Improving fall detection by the use of depth sensor and accelerometer. Neurocomputing **168**, 637–645 (2015)
131. Kepski, M., Kwolek, B.: Detecting human falls with 3-axis accelerometer and depth sensor. In: 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society 2014, pp. 770–773. IEEE (2014)
132. Kepski, M., Kwolek, B., Austvoll, I.: Fuzzy inference-based reliable fall detection using Kinect and accelerometer. In: International Conference on Artificial Intelligence and Soft Computing 2012, pp. 266–273. Springer (2012)
133. Cippitelli, E., Gasparrini, S., Gambi, E., Spinsante, S.: An integrated approach to fall detection and fall risk estimation based on RGB-depth and inertial sensors. In: Proceedings of the 7th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion 2016, pp. 246–253. ACM (2016)
134. Waltz, E., Llinas, J.: Multisensor Data Fusion, vol. 685. Artech House Boston (1990)
135. Walts, E.: Data fusion for C3I. Command, Control, Communications Intelligence (C3I) Handbook, pp. 217–226
136. Lee, H., Park, K., Lee, B., Choi, J., Elmasri, R.: Issues in data fusion for healthcare monitoring. In: Proceedings of the 1st International Conference on PErvasive Technologies Related to Assistive Environments, pp. 3–10. ACM (2008)
137. Khaleghi, B., Khamis, A., Karray, F.O., Razavi, S.N.: Multisensor data fusion: a review of the state-of-the-art. Inf. Fusion **14**(1), 28–44 (2013)
138. Koshmak, G., Loutfi, A., Linden, M.: Challenges and issues in multisensor fusion approach for fall detection: review paper. J. Sens. **2016**, 1–12 (2015)
139. James, G.M.: Variance and bias for general loss functions. Mach. Learn. **51**(2), 115–135 (2003)
140. Patel, S., Park, H., Bonato, P., Chan, L., Rodgers, M.: A review of wearable sensors and systems with application in rehabilitation. J. Neuroeng. Rehabil. **9**(1), 21–37 (2012)
141. Bianchi, V., Grossi, F., Matrella, G., De Munari, I., Ciampolini, P.: Fall detection and gait analysis in a smart home environment. Gerontechnology **7**(2), 73–77 (2008)
142. Sposaro, F., Tyson, G.: iFall: an Android application for fall monitoring and response. In: Annual International Conference of the IEEE 2009 Engineering in Medicine and Biology Society, EMBC 2009. pp. 6119–6122. IEEE (2009)
143. Aquino, K.A.M., Magno, J.L.S., Tuason, G.A.C.: Human Fall Detection Using Three-Axis Accelerometer and ZigBee Technology. Mapúa Institute of Technology (2012)
144. Doukas, C., Maglogiannis, I.: Advanced patient or elder fall detection based on movement and sound data. In: Second International Conference on 2008 Pervasive Computing Technologies for Healthcare, Pervasive Health 2008, pp. 103–107. IEEE (2008)
145. Wang, J., Zhang, Z., Li, B., Lee, S., Sherratt, R.: An enhanced fall detection system for elderly person monitoring using consumer home networks. IEEE Trans. Consum. Electron. **60**(1), 23–29 (2014)
146. Liu, L., Popescu, M., Skubic, M., Rantz, M.: An automatic fall detection framework using data fusion of Doppler radar and motion sensor network. In: 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 5940–5943. IEEE (2014)
147. Zhang, C., Tian, Y., Capezuti, E.: Privacy preserving automatic fall detection for elderly using RGBD cameras. In: Computers Helping People with Special Needs, pp. 625–633 (2012)

148. Cippitelli, E., Gasparrini, S., Gambi, E., Spinsante, S., Wåhsléný, J., Orhany, I., Lindhy, T.: Time synchronization and data fusion for RGB-depth cameras and inertial sensors in AAL applications. In: 2015 IEEE International Conference on Communication Workshop (ICCW), pp. 265–270. IEEE (2015)
149. Vallabh, P., Malekian, R., Computing, H.: Fall detection monitoring systems: a comprehensive review. **9**(6), 1809–1833 (2018)
150. Vihinen, M.: How to evaluate performance of prediction methods? Measures and their interpretation in variation effect analysis. BMC Genom. **13**(4), S2 (2012)
151. Matthews, B.W.: Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochimica et Biophysica Acta (BBA)-Protein Struct. **405**(2), 442–451 (1975)

An Interpretable Machine Learning Model for Human Fall Detection Systems Using Hybrid Intelligent Models

Paulo Vitor C. Souza, Augusto J. Guimaraes, Vanessa S. Araujo, Lucas O. Batista and Thiago S. Rezende

Abstract This chapter presents an assessment of falls and everyday situations in people by sensors dataset collected in fall simulation. This evaluation was performed through the use of intelligent techniques and models based on feature selection techniques and fuzzy neural networks. Therefore, this work can be seen as an auxiliary approach of presenting a vision of knowledge extraction for the construction of actions, prevention, and training to functional that will work in areas correlated to health impacts of people who may have difficulties or injuries due to the impact suffered in a fall. The results obtained were compared with state of the art for the theme and the version of the hybrid model that acts on the most relevant dataset dimensions identifying falls obtained results that surpassed the other models submitted to the test. They were successful in extracting various information from a highly sophisticated and incredibly dimensional dataset to help professionals from various areas expand their investigations in the field of falling people.

Keywords Hybrid models · Cluster · Feature selection · Falls

1 Introduction

Studies dating from the early 1990s already provided robust evidence on the aging of the world population and its impacts on various economic sectors [36]. The health of the elderly concerns many researchers today, especially with the significant aging

P. V. C. Souza (✉) · A. J. Guimaraes · V. S. Araujo · L. O. Batista · T. S. Rezende
Faculty UNA of Betim. Av. Gov. Valadares, 640 Centro, Betim, MG 32510-010, Brazil
e-mail: v.souzaaraujo@yahoo.com.br

A. J. Guimaraes
e-mail: augustojunioguimaraes@gmail.com

L. O. Batista
e-mail: lo.batista@outlook.com

T. S. Rezende
e-mail: silvarezendethiago@hotmail.com

H. Ponce et al. (eds.), *Challenges and Trends in Multimodal Fall Detection for Healthcare*, Studies in Systems, Decision and Control 273,
https://doi.org/10.1007/978-3-030-38748-8_8

of the population and the emergence of new situations that can cause problems in the lives of human beings [21, 49]. The study of falls is improving today to meet these demands of today's society through cutting-edge research to identify cognitive, motor, and brain aspects [53] that can lead to complex health problems in older people [6, 9]. Thus, studying characteristics, impacts on falling [41], brain responsiveness [48], and other elements can help in treating people who have experienced such problems or bruises in times of falling.

Recent research by Martinez-Villasenor et al. [44] it sought to address relevant aspects of actions taken by a group of controlled people to create a data set on diverse reactions of the human body during common everyday activities and distinct types of falls. The comparative approach to the identification of existing patterns in falls approached evaluations of data, images, and hybrid characteristics selection contexts. Therefore this database can inspire advanced research applied to other areas.

To complement the aspects raised in the research by Martinez-Villasenor et al. [44], this chapter proposes to complement the analyzes previously performed with the database to other views and intelligent techniques, with main highlights for categorized characteristics selection techniques and the application of multi-class problem classification through a synergistic hybrid model between the concepts of neural networks and fuzzy systems.

A large number of collected data, coupled with a considerable number of dimensions, can make it difficult to understand and extract existing factors within the existing patterns in each type of fall. Therefore, the use of intelligent techniques, especially grouping and feature selection, can determine fundamental characteristics to identify falls within a context of injuries or situations that eventually led to this type of situation. Hybrid fuzzy neural network techniques can work with positive responses to determine falls while extracting knowledge from the data through fuzzy rules. This knowledge can facilitate the construction of expert systems and provide qualified training for professionals who will analyze causes and possible consequences of a fall, especially in older people and children, who generally have great difficulty explaining the context in which they suffered the kind of accident.

The main objective of this chapter is to highlight the importance of feature selection techniques in improving the ability to classify hybrid models by choosing the most relevant dimensions to find the answers about falling behavior in people. Besides, the objective is to extract knowledge through fuzzy rules to assist in the expert systems construction in the subject.

The main highlight of this chapter is working in the same time with the extraction of knowledge through the use of intelligent techniques based on traditional feature extraction to identify correlation factors between extracted terms and finally, use of a hybrid model for fuzzy rule extraction and possible construction of expert systems. These factors corroborate intelligent techniques that are better accepted by other areas of science, primarily related to health. The big problem with smart models is that they are seen as uninterpretable models because their results have a hard time understanding how they were obtained [1]. This work seeks to explore the interpretability of results through fuzzy techniques.

The remainder of the article is organized as follows. Section 2 presents the theoretical concepts involved in this research, like related work on falls, intelligent problem assessments, topics, and techniques covered in the chapter. Section 3 presents the approaches to be used in the chapter and the way they use to extract knowledge from a database. In Sect. 4, the hybrid model used for fuzzy rule extraction is presented to the reader. In Sect. 5, tests, model configurations, and results are presented to the reader. Finally, Sect. 6 provides conclusions.

2 Related Works

The recognition of falls and the various resources used to identify motor [63], health [59] and cognitive dynamics [43] involved in this phenomenon have become a significant part of studies in science, mainly due to the impacts on people due to the injuries and disorders caused in this kind of situation.

Common elements of day and day, as well as types of falls that can occur with people of all ages, can reflect on injuries (mainly if falls occur with the elderly). Therefore techniques for data collection [44, 45], body reflexes [17, 33], and results from these behaviors [51, 60] become fundamental for predictive and corrective treatments for daily life. Thus new smart techniques can work to predict factors that can lead to less damage to people by falling. In the literature, many authors work with the theme, mainly in the falls' prediction and data collection of their behaviors.

The performance of artificial intelligence is present in the works proposed by Cameron et al. [10], a data mining method applied by Peng et al. [47], the model proposed by Ma et al. who used an Extreme Learning Machine allied with a shape features process [40] or, finally, the work of Albert et al. which uses the combination of machine learning and mobile concepts [2].

Work on structure and processing data on falls has been the subject of study since the late 2000s. Early studies sought to explore fall using video images collected from [4], just as it was addressed in the Charfi database et al. [14] in 2013. The work of [37] has already addressed Kinnect's fall data collection as well as the work of Ma et al. [40], which adds fall identification to extreme learning machine. Other works address falls through audio, and video [22], video and support vector machine [55], and another approach that works on video identification [29] in the home environment. New datasets to deal with the problem were also proposed in the works of Riquelme et al. [52] and in the paper by Maldonado et al. [42] which addresses Assistive Robot-assisted fall detection data and Deep Learning concepts. More specifically, with the database that will be the target of the experiments of this paper, recent works have been developed focused on fault detection through concepts of computer vision and the use of convolutional neural networks [20].

Therefore, the scope and relevance of this area of study for the well-being of modern society are verified.

3 Intelligent Techniques for Knowledge Extraction

The search for knowledge through data has become a central area in contemporary research. Like the internet, computing, and new technology trends created a large volume of data, and human decisions became more complex. Therefore, any tool that assists the extraction of relevant elements in a data feature extraction becomes a great ally in the routine of people and companies, especially in business and medicine. This can occur with the extraction of patterns in large quantities of data that can become a knowledge base within the business, procedures, and routine of human beings. This section will present techniques and concepts linked to the concept of knowledge extraction through data.

3.1 Explainable AI

Explainable AI (XAI) is a field of broad current interest in machine learning that aims to discuss how decisions in intelligent AI models are produced, explaining them to the audience who uses them, and attempting to bring meaning to existing concepts. This area inspects and endeavors to understand the steps and models involved in decision making by intelligent models. In general, this thinking attempts to produce an understanding of the central claims of intelligent models, mainly linked to the complex operation of their methods (often referred to as black-box methods) [25].

However, there are two main issues associated with XAI. First, correctly defining the concept of XAI proves to be considerably challenging. The second problematic factor is the trade-off assessment on some tasks between performance and explainability. It is necessary to control and normalize specific assignments or industries and force them to look for integrated transparency AI solutions. Thus new, more comprehensive techniques can emerge to disseminate quick, practical actions [15].

3.2 Feature Extraction

Feature extraction methods determine subspace θ dimensionality space from a η dimensionality space, where ($\theta \leq \eta$) [27].

This technique that discriminates and distinguishes objects of different classes is defined as pattern recognition. The set of similarities between these characteristics is termed as standard. The combination of both concepts is intended to analyze a given data set and organize it according to patterns, from which the description of a recurring situation and its solution can be reused several times in different situations through the use of patterns, be they objects or data. The recognition of elements or objects requires the establishment of quantifiable parameters that are dependent on position, dimensions, texture, color, and others. Thus knowledge of a database can be obtained through the similarity of concepts, behaviors, and trends of an evaluated dataset [27].

These techniques can define problem dimensions that most correlate with expected responses. Similarly, they can also identify characteristics that are not directly related to problem assessment. This technique, therefore, can be used to improve the classification of patterns by classifiers acting as preprocessing elements by selecting subgroups of best features that represent the problem.

3.3 Fuzzy Neural Networks

Hybrid models produced in science seek to act synergistically in solving complex problems so that they use the best provided by the techniques involved in their architecture [26].

Fuzzy neural networks (FNN) fall into this context by being able to unite concepts of artificial neural networks and the concepts of fuzzy systems. In general, these models use training techniques commonly employed in neural networks and can extract knowledge from the database by constructing fuzzy rules. Therefore in some views, FNNs are seen as fuzzy inferential systems aggregated to neural networks. These networks operate in various areas of knowledge, from robotics [7, 19] to elements focused on health [11, 12, 23, 24, 57]. Its full application has the main advantage of problem-solving and at the same time, the extraction of knowledge that can result in the construction of expert systems [46].

4 Fuzzy Neural Network Applied to Fault Detection

In this chapter, the fuzzy neural network used was proposed by Campos Souza [56] and widely modified to address different problem contexts with different natures. This model is a multilayer neural network with a fuzzification layer, a fuzzy rule-building layer, and the last layer is composed of an artificial neural network of a single neuron. This network has a wide range in binary problem solving, and in this chapter, it will be used to handle multi-class problems.

The fuzzy neural network used in this chapter has an architecture similar to a Multilayer Perceptron network. However, in its first layer, to build the fuzzy neurons, we use the fuzzification process from the ANFIS technique [32] that partitions the data by dividing them through a partition in the grid derived from the use of equally spaced Gaussian membership functions. First layer neurons are formed by the projection of the Gaussian centers and their respective sigma. In the second layer, fuzzy logical neurons are responsible for aggregating the neurons of the first layer, transforming them through fuzzy operators (t-norm and s-norm). Thus the set of the first two layers of the model is responsible for transforming the data into fuzzy rules, which consequently generate IF/THEN relations to represent the problem data

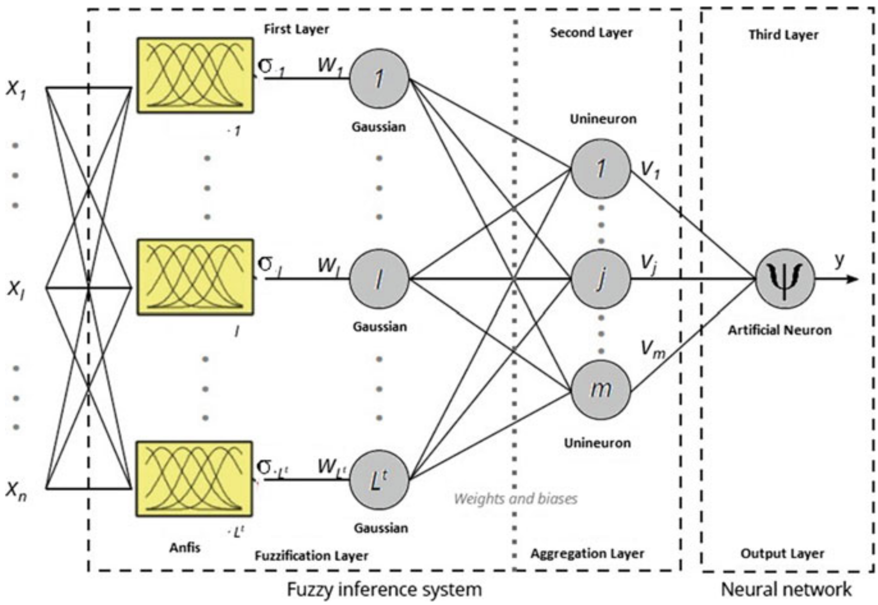


Fig. 1 Fuzzy neural network architecture

logically through linguistic resources. Considering the fuzzification technique is an exponential relationship between problem dimensions and the number of membership functions chosen and consequently, for the creation of the first and second layer neurons of the model, a resampling regularization technique is applied to select a set of neurons in the second layer that is most related to the problem. Finally, the training model is responsible for finding the weights that make up the response of the inference system and act as synaptic weights of the neuron present in the neural network of aggregation of the third layer of the model. This single neuron (also called the Singleton) is given a transforming function (Ψ) in the approximations performed by the model to obtain multiple fall classifications. The whole scheme of the FNN is presented graphically in Fig. 1.

Thus, some specific characteristics of the hybrid model will be exhibited below.

4.1 First Layer

In the primary layer of the model, a fuzzification procedure that employs equalized Jang [32] presented partitioning of the problem variable space and is applied to generate regularly M spaced membership functions of the Gaussian type. Consequently, the neurons of the first layer describe the fuzzification method of the problem data supporting analysis and can be recognized as the first step in discovering problem interpretability through the use of fuzzy logic. Therefore, these neurons, have whose activation functions are membership functions of fuzzy sets that granulate the data space, to form a fuzzy distribution like in De Campos Souza et al. [16].

The Adaptive Network-Based Fuzzy Inference System (ANFIS) [32] consists of five succeeding layers able to building the membership functions of the problem decision space. It acts similarly to fuzzy inference systems, and their adaptive capabilities make them applicable to a wide range of subject areas. A property of the ANFIS model is that the parameter set can be decomposed to use a more efficient hybrid learning rule than the traditional mechanisms found in the literature. Layer 1 and 4 nodes are adaptive, and their values are the parameters of the antecedent and consequent parts of the rule, respectively. The first layer of ANFIS is qualified for determining fuzzy membership functions through adaptive parameters. This layer can be expressed by [32]:

$$\Phi_i^1 = \mu_{A_i}(x), \qquad i = 1, 2 \ldots, n \tag{1}$$

where x is input value, Φ_i^1 is membership value of fuzzy variable A_i. a_i, b_i, c_i are the adaptive parameters commonly referred as premise parameters [32].

Another relevant factor for understanding the model is already in the second layer of the ANFIS, where every node in this layer is a fixed node which acts as a product operation as in Sugeno fuzzy model [32]:

$$\Phi_i^2 = W_i = \mu_{A_i}(x) \times \mu_{B_i}(y), \qquad i = 1, 2 \ldots, n \tag{2}$$

These categories of groups, through membership functions with the equivalent features, can support a detailed analysis of the problem. This method performs a non-linear mapping of its data space to the output area. This mapping is conducted by diverse fuzzy IF/THEN rules, where individually one relates the bounded execution of the mapping. The antecedents in the rules of a fuzzy inference system deliver a multidimensional neural division in the grid. In this work, the method operates with a definition of 200 neurons in the first layer to bypass the problem of high dimensionality. This decision is adequate for many preceding experiments that confirmed that the ability to create fuzzy rules made the issue much more complicated than was essential. For each input variable x_{ij}, L neurons are defined A_{lj}, $l = 1, \ldots L$. Therefore, the expected result in the first layer of the model is the degrees of association related to the inputs submitted to the model [16]:

$$a_{jl} = \mu_{A_l}, \; j = 1 \ldots \; n, \; \; l = 1 \ldots \; L \tag{3}$$

where for each ANFIS input result, the number of problem inputs and the number of fuzzy sets is respectively n and L [16]. This layer reaches out in the chapter because it is useful for extracting primary knowledge from the problem evaluated. So it is reasonable to identify initial relations between dimensions of the problem and membership functions.

Equation. 4 defines the Gaussian functions ($\cap$) used in the construction of the first layer Gaussian fuzzy neurons:

$$\cap\,(x, c, \sigma) = \mu_i^j = e^{\left(-\frac{1}{2}\frac{x_i - c_i^j}{\sigma_i^j}\right)} \tag{4}$$

where i is the number of variables and j is the number of rules, x_i is the input variable and c, and σ are the adjustable parameters of the membership functions, and are called antecedent parameters, which are nonlinear coefficients and correspond to the center and the variability of the membership function.

4.2 Second Layer

The neurons used in the second layer are a particular type of neuron (Unineurons) [38], capable of simultaneously using the concepts of fuzzy arithmetic operators in the second layer of the FNN. They apply the concepts of a fuzzy operator uninorm [62] to implement also simplified operations according to the activation function of the fuzzy neurons. Its composition supports the unineuron to practice both concepts of a neuron *and*, or a neuron *or*. It can be recognized as a mapping that stretches triangular norms by providing the identity element to be a value in the unity interval that has the characteristics commutativity, associativity, and monotonicity, as well as its identity element. The second layer's FNN implements the aggregation of the L_c neurons from the first layer. In this work is represented as follows:

$$U(x, y) = \begin{cases} g\ T(\frac{x}{g}, \frac{y}{g}), if\ y\ \in [0, g] \\ g + (1-g)\ S\ (\frac{x-g}{1-g}, \frac{y-g}{1-g}),\ if\ y\ \in\ (g, 1] \\ \varphi(x, y),\ otherwise \end{cases} \tag{5}$$

$$\varphi(x, y) = \begin{cases} max(x, y)\ , \text{if } g \in [0, 0.5] \\ \min(x, y)\ , \text{if } g \in (0.5, 1] \end{cases}, \tag{6}$$

where T is a *t-norm* (algebraic product), S is a *s-norm* (probabilistic sum) and g is the identity element. It can be affirmed, when $g = 0$ the uninorm is the orneuron type [30], and when $g = 1$ the uninorm is the andneuron type [30]. This ease of change of fuzzy operators allows the model to be more adaptable to the nature of the problem evaluated.

The Unineuron proposed in Lemos et al. [38] performs aggregation operations to unify existing fuzzy values and make the model more manageable. This process can be seen as a preliminary process for turning two values into one and is defined by the following two steps:

1 each pair $(a_i, w_i) = b_i = \mathbf{p}\ (a_i, w_i)$;
2 unified aggregation = $\mathbf{U}\ (b_1, b_2 \ldots b_n)$, where n is the number of inputs.

The function p (relevancy transformation) is responsible for converting the inputs and respective weights into individual transformed values. This role succeeds in the four condition:

- *monotonicity in value*,
- *zero importance elements should have no effect*,
- *normality of importance of one*,
- *consistency of effect of the weight*.

In order for a p function to meet all four necessary requirements of the relevancy transformation operator, Yager proposed the formulation [61]:

$$p(w, a, g) = w.a + \bar{w}.g \tag{7}$$

where $\bar{w}$ represents the complement of w. Using the weighted aggregation reported above the unineuron can be written as [38]:

$$\mathbf{z} = UNI(w; a; g) = U_{i=1}^{n} p(w_i, a_i, g) \tag{8}$$

where T is a *t-norms* (product), s is a *s-norms* (probabilistic sum).

4.3 Third Layer

A Singleton represents the third layer of the FNN model, that is, a single neuron competent of working as a classification [39]. Therefore, the answers obtained can predict it linked to falls. Consequently, this neuron present in the third layer can be seen from the following equation [13, 16]:

$$y = \sum_{j=0}^{l} f(z_l, v_l) \tag{9}$$

where $z_0 = 1$, v_0 is the bias, and z_j and v_j, $j = 1, \ldots, l$ are the output of each fuzzy neuron of the second layer and their corresponding weight, respectively. Finally, f is the neuron activation function (linear function).

4.4 Training Model

The model training algorithm is based on concerning the neural network synaptic weights (and, consequently, the quality of the fuzzy rules of the inference system) through the concepts of applying Moore Penrose's pseudo-inverse [54]. Therefore there is no necessity to update the parameters recursively, and at the same time, there is a reduction in the influence of randomly defined parameters on the model structure. This technique allows weights to be set and in one step. To solve possible overfitting problems in FNN training, a resampling regularization technique [5] is used in the objective function of training to define the neurons (or fuzzy rules) that contribute most efficiently to the model.

The values of the neural network weights of the hybrid model are qualified for the operationalization of the neuron nominations to implement the function approximation. When synaptic weights are defined analytically, the efficiency of the model is evidenced and favors the obtaining of fast and accurate solutions. In this chapter, this vector is considered by the Moore-Penrose pseudo Inverse [16]:

$$\mathbf{v} = \mathbf{Z}^{+}\mathbf{y} \tag{10}$$

where $y = [y_1, y_9, \ldots, y_n]^T$ is the desired output vector and $\mathbf{Z}^{+}$ is pseudo-inverse of Moore-Penrose [54] of $\mathbf{z}$. The l+1 dimensional input space ($\mathbf{z}$), generating a n x l + 1 feature matrix is presented as [16]:

$$\mathbf{z} = \begin{bmatrix} z_0 & z_1(x_1) & z_2(x_2) & \cdots & z_1(x_1) \\ z_0 & z_1(x_1) & z_2(x_2) & \cdots & z_1(x_1) \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ z_0 & z_1(x_n) & z_2(x_n) & \cdots & z_1(x_n) \end{bmatrix} \tag{11}$$

4.5 Regularization Method

In the model composition, $\mathbf{z}$ is the least norm of the least-squares resolution for the weights of the output layer. Fundamentally, the purpose of learning is presented to find the $\mathbf{v}$ parameter that minimizes the error between the network output and the expected output for all training data:

$$\sum_{i=1}^{n} \|\mathrm{z}(x_i)\mathrm{v} - y_i\| \tag{12}$$

Since the number of second layer neurons may contain unnecessary information, it is essential to use analytical techniques to determine the relevance of the collected neurons to the problem analyzed. Statistical evaluations that attempt to discover the correlations between two vectors are performed by regularization techniques, such as Least-Angle Regression (LARS) proposed by Efron et al. [18].

The model proposed by Bach [5], called Bolasso (Model Consistent Lasso Estimation through the Bootstrap), uses this methodology to find a combination of neurons that meet a previously defined criterion in a set of bootstrap replications.

This resampling approach is used to increase the stability of the model selection algorithm. Bolasso uses the LARS algorithm to operate on several bootstrap replications of the neurons in the second layer to perform model selection. For each repetition, a distinct subset of the regressors is selected. The neurons (z_Γ) to be included in the final model are defined according to the frequency with which each of them is chosen through different tests. A consensus threshold is determined, say $\gamma = 70\%$, and thus a regressor is included, if selected in at least 70% of the assays.

Therefore, if a substantial number of bootstrap replications are chosen, there is a considerable probability that a fuzzy neuron will appear in the list of the most relevant ones. Thus, the more relaxed selection criteria may determine a large number of candidate neurons at the end of the tests, just as the high γ value may allow the bootstrap model to be rigid and select a small set of regressors. The fuzzy inference system acknowledgments can extract knowledge from a model-evaluated database, so the rules obtained can be an element for developing expert systems where a system with two inputs and two membership functions for each input is presented. Thus, an example of a fuzzy rule set can be presented as follows [16]:

$$
\begin{aligned}
Rule_1:\ If\ x_{i1}\ is\ A_1^1\ with\ certainty\ w_{11}\ldots\\
and/or\ x_{i2}\ is\ A_1^2\ with\ certainty\ w_{21}\ldots\\
Then\ y_1\ is\ v_1\\
\\
Rule_2:\ If\ x_{i1}\ is\ A_2^1\ with\ certainty\ w_{12}\ldots\\
and/or\ x_{i2}\ is\ A_2^2\ with\ certainty\ w_{22}\ldots\\
Then\ y_2\ is\ v_2\\
\\
Rule_3:\ If\ x_{i1}\ is\ A_3^1\ with\ certainty\ w_{13}\ldots\\
and/or\ x_{i2}\ is\ A_3^2\ with\ certainty\ w_{23}\ldots\\
Then\ y_3\ is\ v_3\\
\\
Rule_4:\ If\ x_{i1}\ is\ A_4^1\ with\ certainty\ w_{14}\ldots\\
and/or\ x_{i2}\ is\ A_4^2\ with\ certainty\ w_{24}...\\
Then\ y_4\ is\ v_4
\end{aligned}
\tag{13}
$$

Therefore it can be concluded that the set of possibly generated fuzzy rules is linked to a proportional relationship between the number of dimensions of the evaluated database and the number of pertinence functions chosen by the user.

4.6 FNN for Multiple Class Problems

Concerning the adequacy of the model, the multi-class responses the artificial neuron in the third layer will receive a transformation function in the approximations obtained in its results in the model output, represented by the Eq. 9. Thus the equation that represents the output of the model is now seen by:

$$y = \sum_{j=0}^{l} \Psi f(z_l, v_l) \tag{14}$$

and

$$\Psi = \begin{cases} y_l & , \vartheta < y_l \\ \text{round}(\vartheta) & , y_l \leq \vartheta \leq y_u \\ y_u & , \vartheta > y_u \end{cases} \tag{15}$$

where Ψ is a rounding purpose that reflects the nearest integer value within the range of the lowest and the highest required value class, as expressed in Eq. (15), where ϑ represents the output of the FNN model as described in Eq. (9), y_l and y_u are the lower and higher expected classes.

Therefore, the model used in this article deals with aspects of using a hybrid model that combines neural network techniques and fuzzy systems to solve problems of falling people. For this, it is necessary to define the number of membership functions, the number of bootstrap replications, and the decision consensus of the model pruning/regularization model.

5 Fall Detection Test Using Intelligent Models

In this chapter, we will reproduce the experiments following the second use case reported in [44].[1] Therefore, this work is exclusively data-based, thus focusing its results on fuzzy rules that are drawn from research on the most diverse types of falls and everyday situations.

5.1 General Characteristics of the Database

Data were collected over a four-week period in July 2018 on the dependencies of the Engineering College of a Mexican university located in the capital of Mexico [44]. Eight females performed the activities to be evaluated, and nine males have randomly

[1] Fall detection using data only from wearable IMUs. This experiment was replicated due to the fuzzy neural network constraint working with numerical data only.

Table 1 Characteristics of test-takers

Subject ID	Age	Height (m)	Weight (kg)	Gender
1	18	1.70	99	Male
2	20	1.70	58	Male
3	19	1.57	54	Female
4	20	1.62	71	Female
5	21	1.71	69	Male
6	22	1.62	68	Male
7	24	1.74	70	Male
8	23	1.75	88	Male
9	23	1.68	70	Female
10	19	1.69	63	Male
11	20	1.65	73	Female
12	19	1.60	53	Female
13	20	1.64	55	Male
14	19	1.70	73	Female
15	21	1.57	56	Female
16	20	1.70	62	Male
17	20	1.66	54	Female

chosen and with distinct characteristics (See Table 1). The script for data collection was to evaluate a set of activities performed by humans during various stages of life, such as walking, taking an object, sitting, jumping, lying down, and standing. On the other hand, we also sought to collect data on five types of human falls (falling forward using hands, falling forward using knees, falling backward, sitting in an empty chair, and falling sideways). The tests were standardized to obtain data from this research following the following assumptions [44]:

– Daily activities were performed for 60 s with exceptions to the jump (30 s) and the collection of an object (10 s).
– To evaluate the falls, the default time was 10 s.[2]

In the experiments, a laboratory room with light intensity control (not to affect the sensor's performance) was used with cameras and fixed position environment sensors. Thus there is a lower probability of sensor failure and their respective data transmission. The sensors used (five wearable Mbientlab MetaSensor sensors) sought to collect raw data in the experiment: 3-axis accelerometer, 3-axis gyroscope, and ambient light value. These devices were installed on the left wrist, under the neck, in the right pocket of the pants, in the middle of the waist (in the belt), and in the left ankle in each of the participants. To complement the collection of fundamental elements

[2] An extra activity has been labeled "kneeling" (20) when a subject remains on his knees after falling [44].

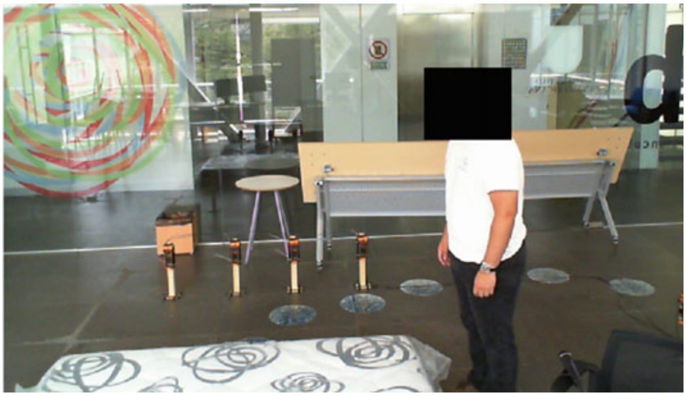


Fig. 2 Example scenario for data collection [44]

in this experiment, we used a headset (NeuroSky MindWave Electroencephalogram (EEG)) to measure the raw brainwave signal from its unique EEG channel sensor located on the forehead. Thus, each participant has the same number of sensors, seeking to identify behaviors of parts relevant to fall processing.

Already in the environmental assessment, six infrared sensors were used as a grid 0.40 m above the floor of the room, to measure changes in the interruption of optical devices, where 0 means interruption and 1 without interruption. The other sensors used in the experiments are not used in this evaluated database [44]. The configurations of the equipment used can be observed in figure.

Figure 2 presents a real example of conducting experiments during the period for which data were collected. This representation demonstrates the location of the falls, the sensors placed in the environment, and the volunteer's body.

The arrangement of equipment in the test volunteers and the types of equipment used are explained in Fig. 3.

Figure 4 shows the relationship of Brain Activity by activity performed by the test participants. You can identify patterns in this context.

Another factor to be highlighted is the behavior of brain waves in the execution of certain tasks. You can see this trend in the Fig. 5.

Finally, it should be noted that some devices did not adequately transmit the results of subject 8 in activity 11 (Trials 2 and 3 for subject 8 in activity 11 are unavailable). Because they are so many collected records, we consider that such absence would not affect the evaluation of results, nor the extraction of knowledge from the dataset.

5.2 *Models Used in the Experiments*

The models used in the tests are commonly used in problems of such complex nature. The classifiers used in this experiment were developed in java (WEKA platform) [28]

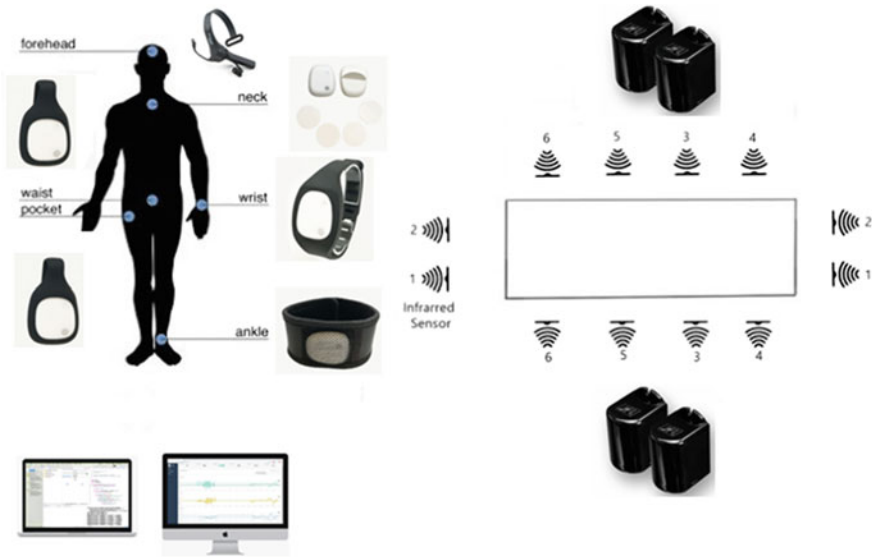


Fig. 3 Equipment used in the tests and their positions [44]

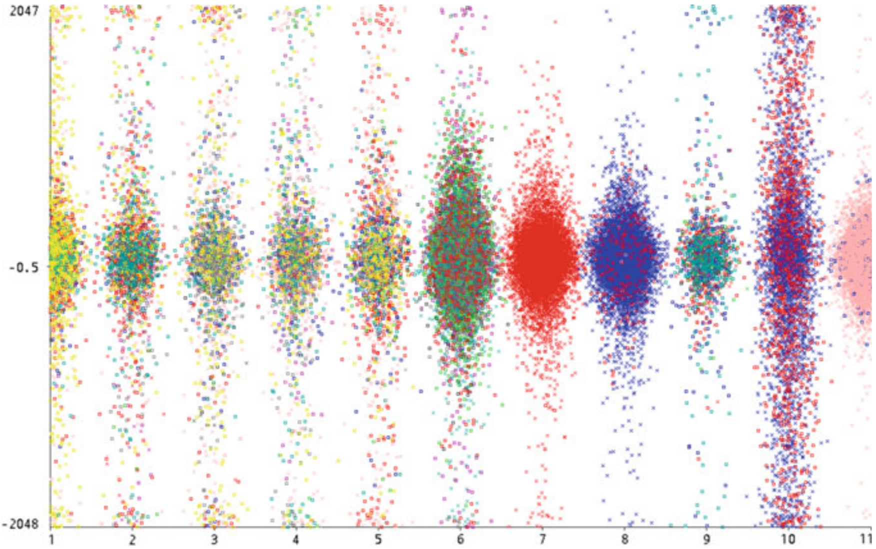


Fig. 4 Brain activity x clusters

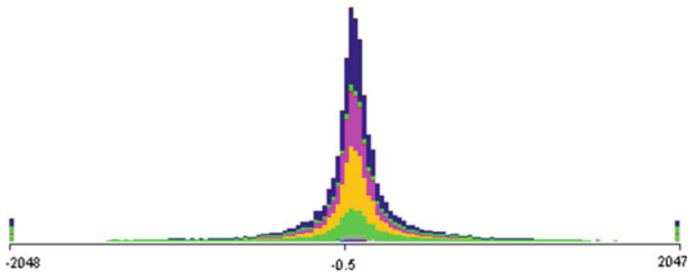


Fig. 5 Brain activity x class

and Matlab. The fuzzy neural network (FNN)[3] explained in the Sect. 3.3 was used to classify falls. To make a proper comparison, models in WEKA [28] were used for the tests, such as Naive Bayes (NB) [34],[4] BayesNet (BN)[5] [58], Random Tree (RT) [3],[6] J48 [50],[7] and a multilayer network that uses the concepts of Extreme Learning Machine (ELM) [31].[8]

As in the article by Martínez-Villaseñor et al. [44], let us look at the database using clustering techniques and feature selection techniques. Therefore, for the hybrid model proposed in this chapter, it is expected an evaluation with all available dimensions and the evaluation model with the most relevant features for the problem. This analysis seeks to evaluate if the most relevant attributes to the problem can improve the classification capacity of the fuzzy neural network model. Therefore the FNN-FS[9] model will bring the model responses by evaluating the most relevant features identified through cluster techniques and feature selection. For the definition of the most relevant dimensions to the problem, the answers of the most relevant classes will be evaluated and together with a committee of characteristics selection techniques.

[3]M=3, bt=8, $\gamma = 70\%$. For a preliminary 10-k-fold test for M = [3, 4, 5], bt = [4, 8, 16] and $\gamma = [50\%, 60\%, 70\%]$.

[4]useKernelEstimator=false, debug=false, displayModelInOldFormat=false, doNotCheckCapabilities=false, useSupervisedDiscretization=false.

[5]estimator=SimpleEstimator, debug=false, searchAlgorithm=F2, doNotCheckCapabilities=false, useADTree=false.

[6]seed=1, allowUnclassifiedInstances=false, debug=false, minNum=1.0, numFolds=0, doNotCheckCapabilities=false, maxDepth=0, minVarianceProp=0.001, KValue=0.

[7]seed=1, unpruned=false, confidenceFactor=0.25, numFolds=3, reducedErrorPruning=false, useLaplace=false, doNotMakeSplitPointActualValue=false, debug=false, subtreeRaising=true, saveInstanceData=false, binarySplits=false, doNotCheckCapabilities=false, minNumObj=2, useMDLcorrection=true.

[8]Gaussian Kernel activation Function, 10 hidden neurons.

[9]The model will have the same parameter setting used in the FNN.

5.3 Test Evaluation Criteria

The experiments were used through the cross-validation technique, defining 70% of the samples to train the models and the remainder (30%) to test the model's identification ability to identify falls. All samples were randomly selected for 30 repetitions to avoid the tendency in the classifier models.

The strategy of one against all involves training a single classifier class, with samples of this class as positive samples and all other samples as negative. This strategy requires base classifiers to produce a confidence score with real value for their decision, rather than just a class label. Typical metrics used in multi-class are used in the case of binary classification. The metric is calculated for each class, treating it as a binary classification problem after grouping all other classes as belonging to the second class. Then the binary average is measured across all classes to obtain an average macro (treat all similar classes).

AUC: represents degree or measure of separability- Eq. 20.

Sensitivity: means the symmetry of actual positives that are precisely classified as such- Eq. 18.

Specificity:measures the proportion of exact negatives that are correctly identified- Eq. 19.

Precision: proportion of positive identifications was actually correct- Eq. 17.

$$ACC = \frac{TP + TN}{TP + FN + TN + FP} \tag{16}$$

$$PRE = \frac{TP}{TP + FP} \tag{17}$$

$$SEN = \frac{TP}{TP + FN} \tag{18}$$

$$SPE = \frac{TN}{TN + TP} \tag{19}$$

$$AUC = \frac{1}{2}(sensitivity + specificity) \tag{20}$$

where TP and TN are the true positives and true negatives, and FP and FN are the false positives and false negatives and finally, the time are measured in seconds.

The evaluation criteria will be the same as those used in the article by Martìnez-Villaseñor et al. [44], seeking to verify if the model maintained the assertive capacity of the other state-of-the-art tests on the subject.

5.4 Feature Selection

For the execution of these tests, we kept the initial WEKA settings [28], and the test settings followed the 10-k-fold criterion and seed = 1. For table normalization, only the eight best results of each technique were collected. The first feature selection technique uses the Pearson correlation concept [8] and the results are shown in Table 2.

The second technique evaluates the gain ratio [35] between the dimension and the evaluated attribute (Table 3).

In the evaluation that determines which dimensions have more significant information gain, Table 4 presents the results of the experiments.

After the tests are operated, we determine the most relevant set of classes that will be part of assessing and extracting responses about falling people. Therefore, for the FNN-FS model, the following dimensions will be part of the experiment:

– AnkleAccelerometerX-axis
– AnkleLuminosity
– RightPocketAccelerometerX-axis
– RightPocketAccelerometerZ-axis

Table 2 Feature selection- correlationAttributeEval

Average merit	Average rank	Attribute
0.332 ± 0	1 ± 0	39 Infra-red 3
0.327 ± 0	2 ± 0	1 AnkleAccelerometerX-axis
0.324 ± 0	3 ± 0	24 NeckAccelerometerZ-axis
0.312 ± 0	4 ± 0	43 Activity
0.27 ± 0.001	5 ± 0	10 RightPocketAccelerometerZ-axis
0.247 ± 0	6 ± 0	17 BeltAccelerometerZ-axis
0.213 ± 0.001	7 ± 0	41 Infra-red 5
0.211 ± 0.001	8 ± 0	42 Infra-red 6

Table 3 Feature selection- GainRatioAttributeEval

Average merit	Average rank	Attribute
0.754 ± 0	1 ± 0	43 Activity
0.503 ± 0.001	2 ± 0	39 Infra-Red 3
0.271 ± 0.001	3 ± 0	41 Infra-Red 5
0.267 ± 0	4 ± 0	21 BeltLuminosity
0.26 ± 0	5.1 ± 0.3	35 WristLuminosity
0.259 ± 0.001	6.4 ± 0.66	38 Infra-Red 2
0.259 ± 0	6.5 ± 0.5	28 NecktLuminosity
0.251 ± 0.002	8.7 ± 0.78	16 BeltAccelerometerY-axis

Table 4 Feature Selection- InfoGainAttributeEval

Average merit	Average rank	Attribute
2.446 ± 0	1 ± 0	28 NeckLuminosity
2.41 ± 0.001	2 ± 0	35 WristLuminosity
2.278 ± 0.001	3 ± 0	43 Activity
2.113 ± 0.001	4 ± 0	21 BeltLuminosity
1.77 ± 0.002	5 ± 0	7 AnkleLuminosity
1.568 ± 0.003	6 ± 0	16 BeltAccelerometerY-axis
1.431 ± 0.003	7 ± 0	8 RightPocketAccelerometerX-axis
1.391 ± 0.001	8 ± 0	10 RightPocketAccelerometerZ-axis

– BeltAccelerometerY-axis
– BeltAccelerometerZ-axis
– BeltLuminosity
– NeckAccelerometerZ-axis
– NeckLuminosity
– WristLuminosity
– Infra-Red 2
– Infra-Red 3
– Infra-Red 5
– Infra-red 6
– Activity.

5.5 *Test Results*

Table 5 presents the results of the experiments performed to identify falls.

Fuzzy neural network results were no better than Bayesian algorithms. However, in the comparison between the fuzzy neural network models and the model that used

Table 5 Test results in output response of models on fall classification problems

Dataset	FNN	FNN-FS	NB	BN	RT	J48	ELM
ACC	85.42 (4.92)	94.12 (2.54)	86.73 (0.09)	**96.35 (0.06)**	93.11(0.33)	95.24 (0.06)	87.98 (7.34)
SPE	0.83 (0.03)	0.95 (0.03)	0.99 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.79 (0.76)
SEN	0.37 (0.01)	0.62 (0.01)	0.24 (0.02)	0.97 (0.01)	0.38 (0.03)	0.47 (0.03)	0.34 (0.02)
AUC	0.6000 (0.06)	0.7850 (0.27)	0.615 (0.01)	0.985 (0.01)	0.690 (0.03)	0.735 (0.02)	0.5650 (0.25)
PRE	0.20 (0.01)	0.56 (0.01)	0.16 (0.01)	0.52 (0.01)	0.39 (0.03)	0.49 (0.02)	0.18 (0.02)
TIME	9810.53 (12.62)	2954.91 (21.51)	110.53 (2.62)	886.03 (152.18)	307.53 (27.41)	7190.17 (269.16)	267.67 (1.21)

Bold represents the better results of the test

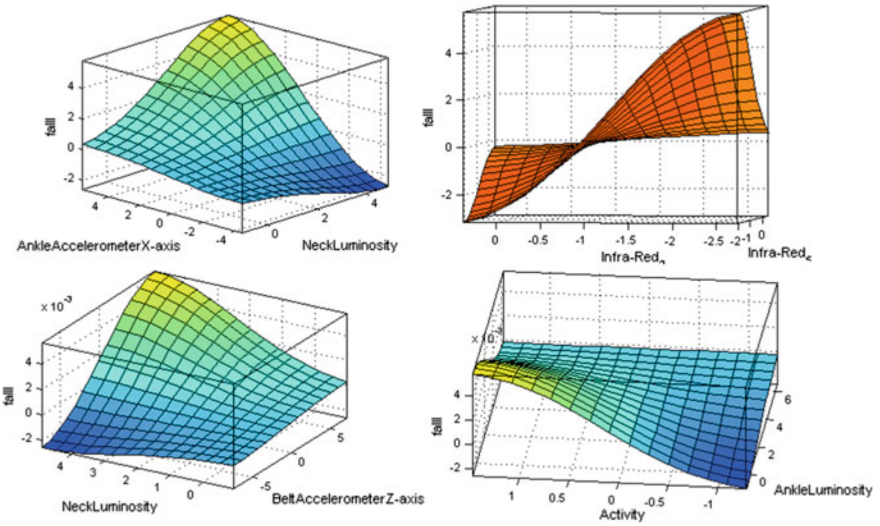


Fig. 6 Decision surface generated by the FNN model

the database with the most relevant dimensions, there was a substantial increase in accuracy and a significant decrease in the execution time of the algorithm, which proves that the selection of characteristics can effectively improve the ability to identify falls in people.

Figure 6 presents the decision space results of the hybrid algorithm, demonstrating how the decision space is realized to achieve the proper results presented in Table 5.

The following fuzzy rules exemplify the extraction of knowledge from the database.

1. If (AnkleAccelerometerX-axis is Low) and/or (AnkleLuminosity is High) and/or (RightPocketAccelerometerX-axis is Low) and/or (RightPocketAccelerometerZ-axis is Low) and/or (BeltAccelerometerY-axis is Low) and/or (BeltAccelerometer Z-axis is Low) and/or (BeltLuminosity is Low) and/or (NeckAccelerometerZ-axis is Low) and/or (NeckLuminosity is Low) and/or (WristLuminosity is Low) and/or (Infra-Red2 is true) and (Infra-Red3 is true) and/or (Infra-Red5 is false) and/or (Infra-Red6 is true) and/or (Activity is CommonActivities) then (v_1 is -4.60)
2. If (AnkleAccelerometerX-axis is Low) and (AnkleLuminosity is High) and (RightPocketAccelerometerX-axis is Low) and (RightPocketAccelerometerZ-axis is High) and (BeltAccelerometerY-axis is Low) and (BeltAccelerometerZ-axis is High) and (BeltLuminosity is Low) and (NeckAccelerometerZ-axis is High) and (NeckLuminosity is Low) and (WristLuminosity is Low) and (Infra-Red2 is true) and (Infra-Red3 is false) and (Infra-Red5 is false) and (Infra-Red6 is false) and (Activity is Falling) then (v_2 is -7.48)
3. If (AnkleAccelerometerX-axis is High) and (AnkleLuminosity is Low) and (RightPocketAccelerometerX-axis is High) and (RightPocketAccelerometerZ-

axis is High) and (BeltAccelerometerY-axis is High) and (BeltAccelerometerZ-axis is Low) and (BeltLuminosity is Low) and (NeckAccelerometerZ-axis is High) and (NeckLuminosity is Low) and (WristLuminosity is High) and (Infra-Red2 is false) and (Infra-Red3 is true) and (Infra-Red5 is false) and (Infra-Red6 is false) and (Activity is CommonActivities) then (v_{13} is 4.10)

In this case, the weight v, according to Eq. 13 represents the weight of the fuzzy rules that consequently participate in the defuzzification process. The v weights found by the fuzzy inference system determine the degree of participation of the fuzzy rule in the definition of falls. So the more the ratio found to be useful in finding the falls, the higher its value. The same relationship happens when you have a rule that does not contribute so much to identifying falls. These weights are linearly combined in the artificial neural network to find aspects of falls. That value defines what kind of fall the data represents.

The great advantage of using hybrid models is the ability to merge the most helpful of two techniques, overcoming each other's difficulties through the synergistic union of complementary concepts. This happens when fuzzy neural networks use the interpretability and uncertainty handling capabilities of fuzzy systems in association with the training and approximation capabilities of artificial neural networks. Consequently, the model can extract knowledge from a database in the form of fuzzy rules and obtain assertive answers through neural network training techniques. Therefore, the set of rules extracted from a database can foster several new expert systems, facilitating the dissemination of intelligent model results in areas that are not specialists in these concepts. Its ability to adapt to the diverse contexts of science allows expert systems based on fuzzy rules to be built to facilitate the routine of professionals in different areas.

6 Conclusion

We can conclude that intelligent techniques can consistently address knowledge extraction in a fall dataset linked to accident prevention contexts in people. To this end, this knowledge can be extracted and transformed into expert systems, training methodologies, and ways of monitoring people who are prone to injury from falls.

Thus the use of classifying models can be seen as examples of a significant gain in the correctness of actions, mainly focused on Bayesian techniques, thus allowing the treatment of certain groups with elements in common. Finally, the feature selection part showed that even on an extremely complex basis, it is possible to identify elements that have a better explanation of a given problem.

Future work may act on obtaining new classification techniques, applying other classifying models, using other feature extraction methodologies.

Acknowledgements The acknowledgments of this work go to Faculdade Una de Betim, the Federal Center for Technological Education of Minas Gerais and Professor Hiram Ponce, who trusted our work and invited us to contribute his book.

References

1. Adadi, A., Berrada, M.: Peeking inside the black-box: a survey on explainable artificial intelligence (xai). IEEE Access **6**, 52138–52160 (2018)
2. Albert, M.V., Kording, K., Herrmann, M., Jayaraman, A.: Fall classification by machine learning using mobile phones. PloS one **7**(5), e36556 (2012)
3. Aldous, D.: The continuum random tree. i. Ann. Probab. 1–28 (1991)
4. Auvinet, E., Multon, F., Saint-Arnaud, A., Rousseau, J., Meunier, J.: Fall detection with multiple cameras: an occlusion-resistant method based on 3-d silhouette vertical distribution. IEEE Trans. Inf. Technol. Biomed. **15**(2), 290–300 (2010)
5. Bach, F.R.: Bolasso: model consistent lasso estimation through the bootstrap. In: Proceedings of the 25th International Conference on Machine Learning, pp. 33–40. ACM (2008)
6. Baleanu, D., Jajarmi, A., Bonyah, E., Hajipour, M.: New aspects of poor nutrition in the life cycle within the fractional calculus. Adv. Differ. Equations **2018**(1), 230 (2018)
7. Bencherif, A., Chouireb, F.: A recurrent tsk interval type-2 fuzzy neural networks control with online structure and parameter learning for mobile robot trajectory tracking. Appl. Intell. 1–13 (2019)
8. Benesty, J., Chen, J., Huang, Y., Cohen, I.: Pearson correlation coefficient. In: Noise reduction in speech processing, pp. 1–4. Springer (2009)
9. Bone, A.E., Gomes, B., Etkind, S.N., Verne, J., Murtagh, F.E., Evans, C.J., Higginson, I.J.: What is the impact of population ageing on the future provision of end-of-life care? population-based projections of place of death. Palliat. Med. **32**(2), 329–336 (2018)
10. Cameron, R., Zuo, Z., Sexton, G., Yang, L.: A fall detection/recognition system and an empirical study of gradient-based feature extraction approaches. In: UK Workshop on Computational Intelligence, pp. 276–289. Springer (2017)
11. de Campos Souza, P.V., Guimaraes, A.J.: Using fuzzy neural networks for improving the prediction of children with autism through mobile devices. In: 2018 IEEE Symposium on Computers and Communications (ISCC), pp. 01086–01089. IEEE (2018)
12. de Campos Souza, P.V., Guimaraes, A.J., Araujo, V.S., Rezende, T.S., Araujo, V.J.S.: Using fuzzy neural networks regularized to support software for predicting autism in adolescents on mobile devices, pp. 115–133. Springer Singapore, Singapore (2019). https://doi.org/10.1007/978-981-13-8614-5_7
13. de Campos Souza, P.V., Torres, L.C.B., Guimaraes, A.J., Araujo, V.S., Araujo, V.J.S., Rezende, T.S.: Data density-based clustering for regularized fuzzy neural networks based on nullneurons and robust activation function. Soft Computing **23**(23), 12475–12489 (2019). https://doi.org/10.1007/s00500-019-03792-z
14. Charfi, I., Miteran, J., Dubois, J., Atri, M., Tourki, R.: Optimized spatio-temporal descriptors for real-time fall detection: comparison of support vector machine and adaboost-based classification. J. Electron. Imaging **22**(4), 041106 (2013)
15. Core, M.G., Lane, H.C., Van Lent, M., Gomboc, D., Solomon, S., Rosenberg, M.: Building explainable artificial intelligence systems. In: AAAI, pp. 1766–1773 (2006)
16. de Campos Souza, P.V., Silva, G.R.L., Torres, L.C.B.: Uninorm based regularized fuzzy neural networks. In: 2018 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), pp. 1–8 (2018). https://doi.org/10.1109/EAIS.2018.8397176
17. Diraco, G., Leone, A., Siciliano, P.: An active vision system for fall detection and posture recognition in elderly healthcare. In: Proceedings of the Conference on Design, Automation and Test in Europe, pp. 1536–1541. European Design and Automation Association (2010)
18. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al.: Least angle regression. Ann. Stat. **32**(2), 407–499 (2004)

19. Er, M.J., Gao, Y.: Robust adaptive control of robot manipulators using generalized fuzzy neural networks. IEEE Trans. Ind. Electron. **50**(3), 620–628 (2003)
20. Espinosa, R., Ponce, H., Gutiérrez, S., Martínez-Villaseñor, L., Brieva, J., Moya-Albor, E.: A vision-based approach for fall detection using multiple cameras and convolutional neural networks: a case study using the up-fall detection dataset. Computers in Biology and Medicine p. 103520 (2019)
21. Feng, Q., Yeung, W.J.J., Wang, Z., Zeng, Y.: Age of retirement and human capital in an aging china, 2015–2050. Eur. J. Popul. **35**(1), 29–62 (2019)
22. Geertsema, E.E., Visser, G.H., Viergever, M.A., Kalitzin, S.N.: Automated remote fall detection using impact features from video and audio. J. Biomech. **88**, 25–32 (2019)
23. Guimarães, A.J., Araujo, V.J.S., Araujo, V.S., Batista, L.O., de Campos Souza, P.V.: A hybrid model based on fuzzy rules to act on the diagnosed of autism in adults. In: J. MacIntyre, I. Maglogiannis, L. Iliadis, E. Pimenidis (eds.) Artificial Intelligence Applications and Innovations, pp. 401–412. Springer International Publishing, Cham (2019)
24. Guimarães, A.J., Araujo, V.J.S., de Campos Souza, P.V., Araujo, V.S., Rezende, T.S.: Using fuzzy neural networks to the prediction of improvement in expert systems for treatment of immunotherapy. In: Ibero-American Conference on Artificial Intelligence, pp. 229–240. Springer (2018)
25. Gunning, D.: Explainable artificial intelligence (xai). Defense Advanced Research Projects Agency (DARPA), nd Web **2** (2017)
26. Gupta, M., Rao, D.: On the principles of fuzzy neural networks. Fuzzy Sets Syst. **61**(1), 1–18 (1994)
27. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.A.: Feature extraction: foundations and applications, vol. 207. Springer (2008)
28. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD Explor. Newsl. **11**(1), 10–18 (2009)
29. Harrou, F., Zerrouki, N., Sun, Y., Houacine, A.: An integrated vision-based approach for efficient human fall detection in a home environment. IEEE Access **7**, 114966–114974 (2019)
30. Hirota, K., Pedrycz, W.: Or/and neuron in modeling fuzzy set connectives. IEEE Trans. Fuzzy Syst. **2**(2), 151–161 (1994)
31. Huang, G.B., Zhou, H., Ding, X., Zhang, R.: Extreme learning machine for regression and multiclass classification. IEEE Trans. Syst., Man, Cybern., Part B (Cybernetics) **42**(2), 513–529 (2011)
32. Jang, J.S.: Anfis: adaptive-network-based fuzzy inference system. IEEE Trans. Syst., Man, Cybern. **23**(3), 665–685 (1993)
33. Jansen, B., Temmermans, F., Deklerck, R.: 3d human pose recognition for home monitoring of elderly. In: 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 4049–4051. IEEE (2007)
34. John, G.H., Langley, P.: Estimating continuous distributions in bayesian classifiers. In: Proceedings of the Eleventh conference on Uncertainty in artificial intelligence, pp. 338–345. Morgan Kaufmann Publishers Inc. (1995)
35. Karegowda, A.G., Manjunath, A., Jayaram, M.: Comparative study of attribute selection using gain ratio and correlation based feature selection. Int. J. Inf. Technol. Knowl. Manag. **2**(2), 271–277 (2010)
36. Keyfitz, N., Flieger, W., et al.: World population growth and aging. University of Chicago Press Economics Books (1991)
37. Kwolek, B., Kepski, M.: Human fall detection on embedded platform using depth maps and wireless accelerometer. Comput. Methods Programs Biomed. **117**(3), 489–501 (2014)
38. Lemos, A., Caminhas, W., Gomide, F.: New uninorm-based neuron model and fuzzy neural networks. In: 2010 Annual Meeting of the North American Fuzzy Information Processing Society, pp. 1–6 (2010). https://doi.org/10.1109/NAFIPS.2010.5548195
39. Lemos, A., Kreinovich, V., Caminhas, W., Gomide, F.: Universal approximation with uninorm-based fuzzy neural networks. In: 2011 Annual Meeting of the North American Fuzzy Information Processing Society, pp. 1–6 (2011). https://doi.org/10.1109/NAFIPS.2011.5752000

40. Ma, X., Wang, H., Xue, B., Zhou, M., Ji, B., Li, Y.: Depth-based human fall detection via shape features and improved extreme learning machine. IEEE J. Biomed. Health Inform. **18**(6), 1915–1922 (2014)
41. Makino, K., Makizako, H., Doi, T., Tsutsumimoto, K., Hotta, R., Nakakubo, S., Suzuki, T., Shimada, H.: Impact of fear of falling and fall history on disability incidence among older adults: P rospective cohort study. Int. J. Geriatr. Psychiatry **33**(4), 658–662 (2018)
42. Maldonado-Bascón, S., Iglesias-Iglesias, C., Martín-Martín, P., Lafuente-Arroyo, S.: Fallen people detection capabilities using assistive robot. Electronics **8**(9), 915 (2019)
43. Martin, K., Thomson, R., Blizzard, L., Wood, A., Garry, M., Srikanth, V.: Visuospatial ability and memory are associated with falls risk in older people. Dement. Geriatr. Cogn. Disord. **27**(5), 451–457 (2009)
44. Martínez-Villaseñor, L., Ponce, H., Brieva, J., Moya-Albor, E., Núñez-Martínez, J., Peñafort-Asturiano, C.: Up-fall detection dataset: a multimodal approach. Sensors **19**(9), 1988 (2019)
45. Parra-Dominguez, G.S., Snoek, J., Taati, B., Mihailidis, A.: Lower body motion analysis to detect falls and near falls on stairs. Biomed. Eng. Lett. **5**(2), 98–108 (2015)
46. Pedrycz, W.: Fuzzy neural networks and neurocomputations. Fuzzy Sets Syst. **56**(1), 1–28 (1993)
47. Peng, Y., Zhang, T., Sun, L., Chen, J.: A novel data mining method on falling detection and daily activities recognition. In: 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 675–681. IEEE (2015)
48. Perera, S., Mody, S.H., Woodman, R.C., Studenski, S.A.: Meaningful change and responsiveness in common physical performance measures in older adults. J. Am. Geriatr. Soc. **54**(5), 743–749 (2006)
49. Plouffe, L., Kalache, A.: Towards global age-friendly cities: determining urban features that promote active aging. J. Urban Health **87**(5), 733–739 (2010)
50. Quinlan, J.R.: C4. 5: Programs for Machine Learning. Elsevier (2014)
51. Rasheed, M.B., Javaid, N., Alghamdi, T.A., Mukhtar, S., Qasim, U., Khan, Z.A., Raja, M.H.B.: Evaluation of human activity recognition and fall detection using android phone. In: 2015 IEEE 29th International Conference on Advanced Information Networking and Applications, pp. 163–170. IEEE (2015)
52. Riquelme, F., Espinoza, C., Rodenas, T., Minonzio, J.G., Taramasco, C.: ehomeseniors dataset: an infrared thermal sensor dataset for automatic fall detection research. Sensors **19**(20), 4565 (2019)
53. Segal, D.L., Qualls, S.H., Smyer, M.A.: Aging and mental health. Wiley (2018)
54. Söderström, T., Stewart, G.: On the numerical properties of an iterative method for computing the moore-penrose generalized inverse. SIAM J. Numer. Anal. **11**(1), 61–74 (1974)
55. Soni, P.K., Choudhary, A.: Automated fall detection from a camera using support vector machine. In: 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), pp. 1–6. IEEE (2019)
56. Souza, P.V.C.: Regularized fuzzy neural networks for pattern classification problems. Int. J. Appl. Eng. Res. **13**(5), 2985–2991 (2018)
57. Souza, P.V.C., dos Reis, A.G., Marques, G.R.R., Guimaraes, A.J., Araujo, V.J.S., Araujo, V.S., Rezende, T.S., Batista, L.O., da Silva, G.A.: Using hybrid systems in the construction of expert systems in the identification of cognitive and motor problems in children and young people. In: 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1–6 (2019). https://doi.org/10.1109/FUZZ-IEEE.2019.8858906
58. Thepade, S.D., Kalbhor, M.M.: Extended performance appraise of bayes, function, lazy, rule, tree data mining classifier in novel transformed fractional content based image classification. In: 2015 International Conference on Pervasive Computing (ICPC), pp. 1–6. IEEE (2015)
59. Tilling, L.M., Darawil, K., Britton, M.: Falls as a complication of diabetes mellitus in older people. J. Diabetes Its Complicat. **20**(3), 158–162 (2006)
60. Wickramasinghe, A., Torres, R.L.S., Ranasinghe, D.C.: Recognition of falls using dense sensing in an ambient assisted living environment. Pervasive Mob. Comput. **34**, 14–24 (2017)
61. Yager, R.R.: Uninorms in fuzzy systems modeling. Fuzzy Sets Syst. **122**(1), 167–175 (2001)

62. Yager, R.R., Rybalov, A.: Uninorm aggregation operators. Fuzzy Sets Syst. **80**(1), 111–120 (1996)
63. Zigel, Y., Litvak, D., Gannot, I.: A method for automatic fall detection of elderly people using floor vibrations and sound–proof of concept on human mimicking doll falls. IEEE Trans. Biomed. Eng. **56**(12), 2858–2867 (2009)

Multi-sensor System, Gamification, and Artificial Intelligence for Benefit Elderly People

Juana Isabel Méndez, Omar Mata, Pedro Ponce, Alan Meier, Therese Peffer and Arturo Molina

Abstract Over the years, the elderly people population will become more than children population; besides since 2018 people over 65 years old outnumbered the population under 5 years old. Growing up involves biological, physical, social and psychological changes that may lead to social isolation and loss of loved ones, or even to the sense of loss of value, purpose or confidence. Moreover, as people are aging, they usually spend more time at home. As a solution, social inclusion through mobile devices and smart home seem to be ideal to avoid that lack of purpose in life, confidence or value. Smart homes collect and analyze data from household appliances and devices to promote independence, prevent emergencies and increase the quality of life in elderly people. In that regard, the multi-sensor system allows the expert to know more about the elderly people needs to propose actions that improve the elderly people's quality of life, as they can read and analyze through sensors their facial expressions, voice, among others. Gamification in older people may motivate elderly people to socialize with their peers through social interaction and by doing activities as exercising. Thus, this chapter proposes to use an adaptive neural network fuzzy inference to evaluate camera and voice devices that come from the multi-sensor

J. I. Méndez · O. Mata · P. Ponce (✉) · A. Molina
School of Engineering and Sciences, Tecnológico de Monterrey, México City, Mexico
e-mail: pedro.ponce@tec.mx

J. I. Méndez
e-mail: A01165549@itesm.mx

O. Mata
e-mail: omar.mata@tec.mx

A. Molina
e-mail: armolina@tec.mx

A. Meier
Lawrence Berkeley National Laboratory, University of California, Berkeley, CA 94720, USA
e-mail: akmeier@ucdavis.edu

T. Peffer
Institute for Energy and Environment, University of California, Berkeley, CA 94720, USA
e-mail: tpeffer@berkeley.edu

H. Ponce et al. (eds.), *Challenges and Trends in Multimodal Fall Detection for Healthcare*, Studies in Systems, Decision and Control 273,
https://doi.org/10.1007/978-3-030-38748-8_9

system to propose an interactive and tailored human-machine interface for the elderly people in the home.

Keywords Multi-sensor system · Smart home · ANFIS · Gamification · HMI · Voice detection · Face detection · Social inclusion · Physical activity

1 Introduction

This chapter proposes the use of a multisensory system into a smart home environment; this multisensory system allows the connectivity of all the sensors that could be installed in a smart home in order to know more about the requirements of elderly people, specifically for avoiding social isolation and incrementing physical activity to improve their healthy conditions. Those problems are considered extremely important thus must be solved since elderly population is vulnerable. In addition, this multi-sensor structure is focused on the decision fusion stage that is based on the information that is coming from all the sensors and evaluated for an adaptive neural network fuzzy inference (ANFIS). This proposal also describes how the elderly people can be engaged to improve their quality of life by a gamification strategy and a fuzzy logic system, which are running on a human machine interface (HMI). Figure 1 describes the general proposal for data fusion on HMI at smart homes.

1.1 Elderly People

By 2018, for the first time, the population of people over 65 years old outnumbered children under 5 years old. Additionally, the United Nations expects that by 2050, there will be 1.5 billion people aged 65 and will outnumber the adolescents and youth population from 15 to 24 years old (1.3 billion). Thus, the proportion of elderly people in the world is projected to reach from 9% to 16% in 2050 [1].

Figure 2 shows a graphic of the estimated and projected global population by broad age group from 1950 to 2100, where population over 65 will increase while population under 24 will decrease.

Elderly people can be seen as *individuals with a wealth of life experience, with interests and aspirations in their later life, as limitations and losses* [2]. Aging involves biological changes, such as cognitive deterioration, physical strength diminished or languishing sensory perception and detection; and social changes, like social isolation and loss of loved ones, or as a sense of loss of value, purpose or confidence [3, 4]. Furthermore, one of the challenging changes in aging is the loss of autonomy in daily life, this causes a modification of the living environment. Thus, social inclusion and social exclusion allow access to social engagement and participation through social relationships, civic activities, local services and financial resources [5]. Moreover, social relationships are related to better health and well-being in late

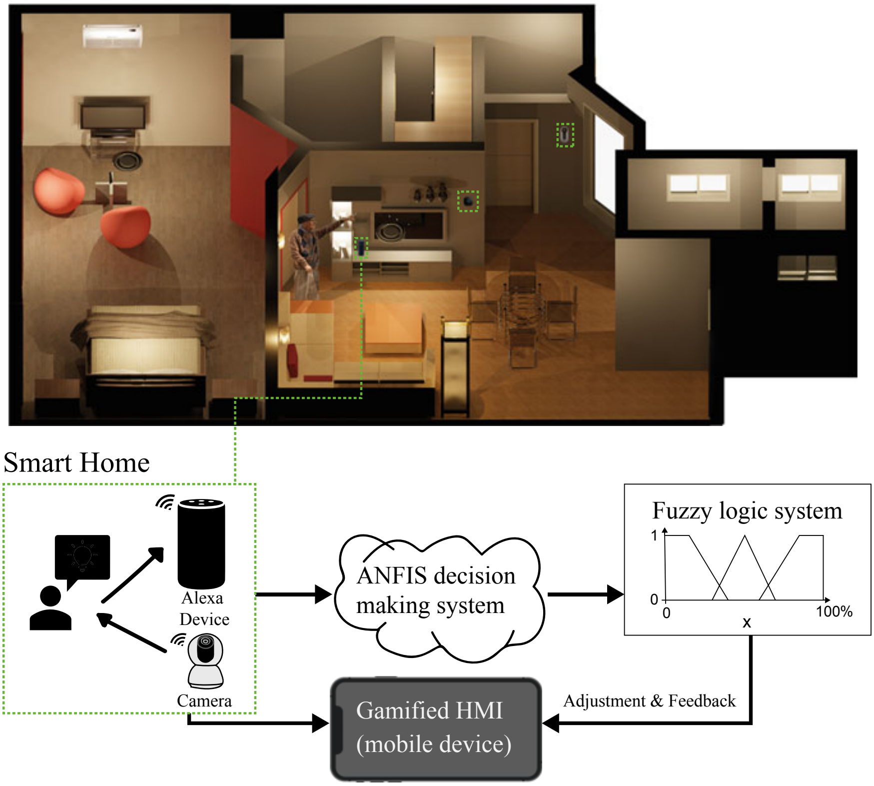


Fig. 1 General proposal for data fusion on HMI at smart homes

life. Elderly people use technology to shape their social contexts; however, its use depends on the positive or negative evaluation of a device. Table 1 shows the theoretical and practical implications of the conceptual model of technology use and social context in late life [6].

In that regard, since 1988, gerontechnology emerged as a task to solve problems and challenges found by aging people. Ambient Assisted Living (AAL) technologies support elderly people to maintain and continue their daily life more independently [7]. Moreover, the arising of technology allows connecting devices and systems to exchange communication with individuals and collect the data derived from that interaction [8].

There is an increase of life expectancy in the world, due to technological and medical improvements; it is relevant to develop and implement new strategies and technologies for the elderly that improves health care, independence, social inclusion, among others. In that regard, smart homes may allow elderly people to stay comfortable by monitoring their health and social inclusion with unobtrusive and non-invasive remote devices [7].

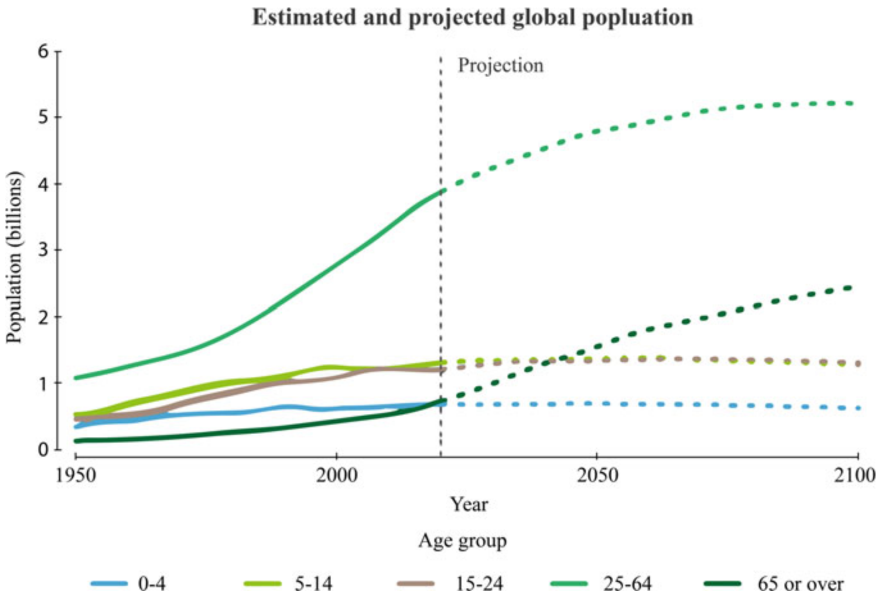


Fig. 2 Estimated and projected global population by broad age group from 1950 to 2100 [1]

Table 1 Theoretical and practical implications of the conceptual model of technology use and social context in later life from [6]

Potential of social contexts for enhancing technology use	Potentials of technology for enhanced beneficial effects of social contexts	Outcomes of relationship regulation
Demands: Challenges in social contexts that motivate technology use	**Selection**: Supporting the proactive shaping of social contexts	**Goal**: Maximization of positive experiences in social contexts
Resources: Opportunities in social contexts that motivate technology use	**Optimization**: Enhancing investment and means for improving social contexts	**Outcomes**: Enhanced relationship quality and perceived closeness
Interplay processes: Influence of demands and resources depends on the person	**Compensation**: Providing means to compensate for loss and burdens in social contexts	

1.2 Smart Homes

In 1984 the American Association of House Builders introduced the concept of smart homes in terms of “wired homes” [9]. However, the term is often defined based on technological aspects and usage. For instance, the construction sector defined a smart home as *a residence equipped with computing and information technology, which anticipates and responds to the needs of the occupants, working to promote* *their comfort, convenience, security and entertainment through the management of technology within the home and connections to the world beyond* [9], or as a *living environment that has the technology to allow devices and systems to be controlled automatically* [10]. The health care sector define it as a *residence equipped with technology that facilitates monitoring of residents' health status and/or that promotes independence, prevents emergencies, and increases their quality of life* or as an *assisted interactive dwelling house* [10]. This type of home collects and analyzes data, gives information to the habitants and manage different domestic appliances [11]. Figure 3 shows some common household appliances used in a smart home as Smart TV, electric stove, coffee maker, interior lighting, washing machine, refrigeration, and connected thermostats. Furthermore, smart household appliances could be accepted if [10]:

- Elderly know that those type of appliances exist.
- The appliances and devices can be quickly and cheaply obtained.
- Those products demonstrate they reduce or eliminate physical demands for their operation.

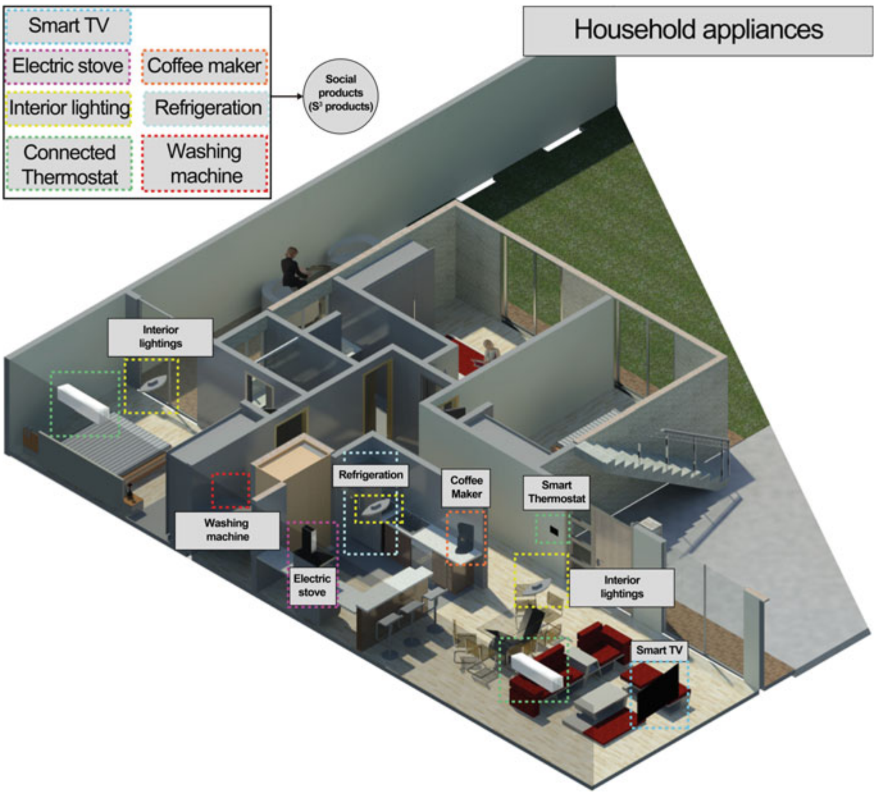


Fig. 3 Household appliances

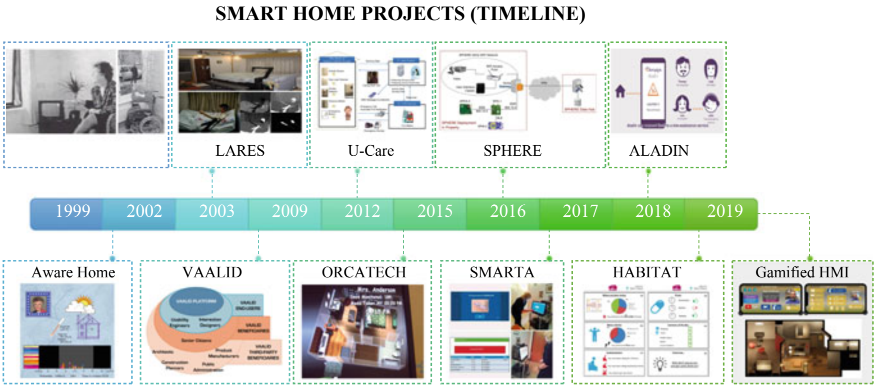


Fig. 4 Timeline of Smart Home projects for the elderly people

- The usability of the product considers characteristics as maneuverability.

Since the adoption of technology, the development of the smart home for the elderly has been researched. Figure 4 shows the timeline of the smart house development for the elderly. In 1998, the AID project (Assisted Interactive Dwelling) proposed to use homes that consider equally the non-disabled, elderly or disabled tenants through the concept of barrier-free design by taking advantage of the smart technology at that time. They used a demo that considered a motor-driven window, door and curtain control, enhanced heating controls, 'keyless' door locks, a video-entry system, an enhanced security system, and infra-red bathroom controls. The idea was to raise awareness amongst caring agency, housing providers, architects, product manufacturers and end-users to promote the use of technology in homes [12]. In 2002, the Aware Home project proposed to use a Digital Family Portrait to support awareness of the long-term health, activity, and social well-being of senior adults living by themselves. This portrait tracked the daily activity of the users and showed the last six actions performed by them; it also included the weather, indoor and outdoor temperature and the number of room-to-room transitions in 15-min increments [13]. The next year, in 2003 the LARES project [14] consisted of an intelligent human-friendly residential system that implemented an intelligent bed robot for the elderly and the handicapped where an arm was attached to the bed for transporting objects; the human-oriented interface informed the users intention to the bed robot; and the home network was equipped to transmit and share information between each device. In 2009, the VAALID project (Accessibility and Usability Framework for AAL Interaction Design Process) began to develop new tools and methods to make easier and more dynamic the creation, design, construction, implementation and evaluation processes for technological solutions within ALL to ensure the accessibility and usability of the environment for the elderly [15]. In 2012 the U-Care project applied the Korean government method where they use the activity data alone to effectively analyze the status of the solitary elderly people; the home was equipped with a gas leak detector, gateway, absence button, smoke detector and activity sensor to analyze the user's activity level, which are monitored with five-second intervals, and the activity data calculated every hour [16]. Since 2015 the ORCATECH (Oregon Center for Aging and Technology) project of aging developed an advanced platform for assessing the health of people living within their homes using passive sensing technologies to enable accurate assessment of cognitive and physical health [17, 18]. In 2016 the SPHERE project (Sensor Platform for Healthcare in a Residential Environment) aimed to develop a smart home platform of non-medical network sensors, capable of gathering and integrating multiple types of data about the home environment and the behaviors of its residents to understand a range of healthcare needs [19]. In 2017 the SMARTA project consisted of developing and testing a personal health system that integrated standard sensors as well as innovative wearables and environmental sensors to allow home telemonitoring of vital parameters and detection of anomalies in daily activities, thus supporting active aging through remote healthcare [20]. In 2018, the European Commission funded the ALADIN project (Smart Home-Care solutions for the Elderly) to create smart furniture solutions for the elderly and the nursing homes that take care of the elderly; ALADIN is a homecare assistant which increases the independence of elderly people living alone [21, 22]. Recently, the HABITAT project (Home Assistance Based on the Internet of Things for the Autonomy of Everybody) developed an IoT-based platform for assistive and reconfigurable spaces that integrates RFID, wearable electronics, wireless sensor networks, and artificial intelligence. This project has the purpose of assisting needy people in their homes in safe conditions, helping them to conduct autonomously most of the activities tied to the satisfaction for their primary needs, sustaining actions focused on hospitalization and home care [7]. Finally, this year, the Tecnologico de Monterrey and UC Berkeley are developing the Gamified HMI project. This first stage takes advantages of the camera and Alexa to train an ANFIS model to propose a tailored gamified HMI that teaches, engages and motivates elderly people to keep in touch with their peers, caregivers, doctors, and family members to promote social inclusion and happiness and avoid social isolation and depression; the following steps will consider the smart household appliances to use them as social products that will interact with the end-user and the devices and between devices [23].

2 Multi Sensors: Data Fusion

When multiple sensors are collecting information, which later is combined, they can make accurately inferences than could not be achieving by a single sensor; moreover, if these sensors are placed under a reference framework that is able to map the value of the property or attribute to a quantitative measurement in a consistent and predictable manner, this could be extremely attractive as a framework because it also includes functions which can be described in terms of compensation, information processing, communication and integration. This framework is called multisensor data fusion [24–26]. The multisensor data fusion concept is based on fundamental tasks done

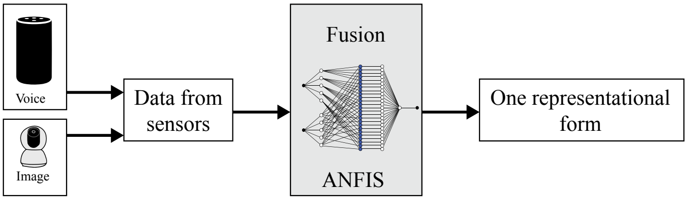


Fig. 5 Data fusion legend

by animals and humans since they use multiple sensors to improve their ability to survive when they have to identify potential threats and the have to perform some functions; for instance, the five human senses used in daily tasks are hear, touch, sight, smell and taste.

The primarily functions are:

- **Compensation** when sensors respond to environmental changes by self-diagnostic tests, self-calibration and adaption, a compensation process is taking place.
- **Information processing** is the stage linked to signal conditioning, data reduction, event detection and decision-making.
- **Communications** this stage is based on implementing a standardized inference protocol for connecting the sensor and the outside world.
- **Integration** is generated when the sensing and computation processes are linked under the same silicon chip or system.

Besides, a **decision making stage** is a primary stage in the multisensory system because it could be considered as the central stage in which the information from sensors are processed in order to create a decision according with the sensed data.

Thus, the concept of data fusion is not new, but the evolution of innovative sensors, digital systems, machine-learning algorithms, advanced processing techniques and real time processing devices allow creating a high-performance data fusion system. Figure 5 represents that concept; data from Alexa and camera are collected to put them on the ANFIS system to analyze the data and represent the information in one form. For instance, the output could be the required features to propose a tailored interface.

2.1 Multisensor Configuration

The configuration of sensor is focused on two main detections that are presented below.

1. Detection of facial expression and body posture for detecting the mood of an elderly person as well as physical limitations.
2. Detection of nonconventional behavior through a voice survey; for getting this information an Alexa system was used in which a survey proposed by Yesavage

[27] was deployed in order to rate depression in the elderly. This scale is known as Geriatric Depression Scale (GDS); from 0 to 9 is a not depressed user, from 10 to 19 a mild depressive individual and from 20 to 30 a severe depressive person. This questionnaire is usually completed in 10 min or less and was initially validated with depressed patients and with elderly people without any history of mental disorder [4].

2.2 Multisensor: Decision Fusion

The representation of knowledge could be achieved by an inference system, which could be normally generated through an inference system [28]. When it was proposed to use in this multisensory system into the decision making a fuzzy logic system, it was described as an extension from binary values (0 and 1). However, an ANFIS could be trained and generate automatically the linguistic rules so a knowledge base is crated [24].

2.3 ANFIS: Adaptive Neuro-Fuzzy Inference Systems

Sometimes, conventional mathematical modeling algorithms do not deal with vague or uncertain information. Thus, Fuzzy systems using linguistic rules (IF-THEN) have the strength and ability to reason as humans, without employing precise and complete information. However, a problem arises, how to transfer human knowledge to a fuzzy system. Several proposals have been made, such as the combination of artificial neural networks with fuzzy systems. Artificial neural networks have the ability to learn and adapt from experience, thus complementing fuzzy systems. Among the most important techniques is the ANFIS, an adaptive neuro-fuzzy inference system proposed by Jang [24] in 1993, which generates fuzzy IF-THEN rule bases and fuzzy membership functions automatically. ANFIS is based on adaptive networks which is a super set of feedforward artificial neural networks with supervised learning capabilities as stated by Jang in [24, 29]. It is a topology of nodes directionally connected, almost all the nodes depend on parameters that are changed according certain learning rules that will minimize an error criteria. The most used learning rule is the gradient descent method; however, Jang proposed a hybrid learning rule that incorporates least square estimation.

2.4 Multisensor Topology Proposed: Detection of Nonconventional Behavior

The multisensory system is based on two elements; the first one is an Alexa voice smart home controller [30] and a face detection for human emotions.

Alexa applies a survey to the end-user in order to know more about his mood, a classification between 0 and 1 according with the responses is done using the survey presented in Table 2; this survey was presented in [4]. The basic internal structure of Alexa is shown in Fig. 6.

3 Emotions Classification: Detection of Facial Expression

An online detection of emotions based on facial expressions is achieve using a webcam and a PC running Python with the OpenCV library. The PC has an Intel Core i7-7500U dual core processor with 8 GB of RAM and an integrated camera HP wide vision HD of 0.92 MP and resolution of 1280 × 720 px.

The process for the emotion detection consists mainly in the use of two separate artificial neural networks. The first one is a deep neural network that extracts the faces from the image. For the second part a convolutional neural network is used to classify the extracted face within seven different levels of emotions: angry, disgust, fear, happy, sad, surprised, and neutral.

The overall process is listed next:

1. Get one frame from the webcam and resize to 300 × 300 pixels
2. Extract the biggest face detected using a DNN
3. Resize the face to 48 × 48 pixels in grayscale
4. Use a CNN to detect the emotions
5. Plot the level of each emotion detected.

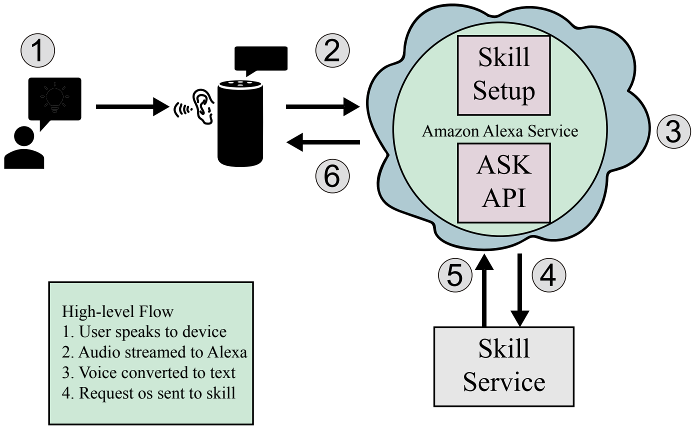
3.1 Face Detection

The idea for the deep networks is that they can extract low, middle and high-level features in a multi-layer architecture, hence the common trend to stack layers going "deeper" in the net. But deep networks are hard to train because when they start to converge a degradation problem shows, as the gradient is back-propagated it gets smaller because of the repetitive multiplications. So, as the network goes deeper it gets saturated. As a solution to this, the residual networks were introduced in which an identity shortcut connection is used to skip one or more layers to perform an identity mapping and their outputs are added to the outputs of the stacked layers as shown in Fig. 7.

Table 2 Survey for elderly people in order to detect social isolation problems (depression)

No.	Question	Yes/no
1	Are you basically satisfied with your life?	Yes(1) ǀ no (0)
2	Have you dropped many of your activities and interests?	Yes(1) ǀ no (0)
3	Do you feel that your life is empty?	Yes(1) ǀ no (0)
4	Do you often get bored?	Yes(1) ǀ no (0)
5	Are you hopeful about the future?	Yes(1) ǀ no (0)
6	Are you bothered by thoughts you can't get out of your head?	Yes(1) ǀ no (0)
7	Are you in good spirits most of the time?	Yes(1) ǀ no (0)
8	Are you afraid that something bad is going to happen to you?	Yes(1) ǀ no (0)
9	Do you feel happy most of the time?	Yes(1) ǀ no (0)
10	Do you often feel helpless?	Yes(1) ǀ no (0)
11	Do you often get restless and fidgety?	Yes(1) ǀ no (0)
12	Do you prefer to stay at home rather than going out and doing new things?	Yes(1) ǀ no (0)
13	Do you frequently worry about the future?	Yes(1) ǀ no (0)
14	Do you feel you have more problems with memory than most?	Yes(1) ǀ no (0)
15	Do you think it is wonderful to be alive now?	Yes(1) ǀ no (0)
16	Do you often feel downhearted and blue?	Yes(1) ǀ no (0)
17	Do you feel pretty worthless the way you are now?	Yes(1) ǀ no (0)
18	Do you worry a lot about the past?	Yes(1) ǀ no (0)
19	Do you find life very exciting?	Yes(1) ǀ no (0)
20	Is it hard for you to get started on new projects?	Yes(1) ǀ no (0)
21	Do you feel full of energy?	Yes(1) ǀ no (0)
22	Do you feel that your situation is hopeless?	Yes(1) ǀ no (0)
23	Do you think that most people are better off than you are?	Yes(1) ǀ no (0)
24	Do you frequently get upset over little things?	Yes(1) ǀ no (0)
25	Do you frequently feel like crying?	Yes(1) ǀ no (0)
26	Do you have trouble concentrating?	Yes(1) ǀ no (0)
27	Do you enjoy getting up in the morning?	Yes(1) ǀ no (0)
28	Do you prefer to avoid social gatherings?	Yes(1) ǀ no (0)
29	Is it easy for you to make decisions?	Yes(1) ǀ no (0)
30	Is your mind as clear as it used to be?	Yes(1) ǀ no (0)

For the face detection part, a model included in the OpenCV library was used based on a single-shot-multibox detector and a ResNet-10 architecture as backbone. This model was already trained with images available on the model zoo of the Caffe framework [31]. The results of this network are shown in the Fig. 8.



https://developer.amazon.com/es-MX/alexa/alexa-skills-kit/?gclid=EAIaIQobChMI2s3EruSz5QIVTvDACh2kMAoGEAAYASAAEgKeTfD_BwE

Fig. 6 Alexa internal structure

Fig. 7 Residual block

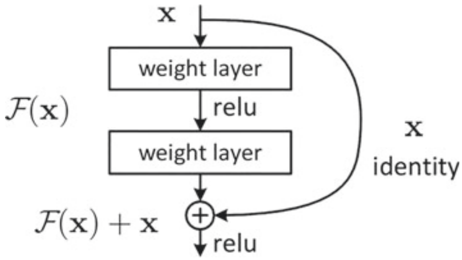


Fig. 8 Detection and extraction of detected face

3.2 Emotion Detection

Convolutional neural networks (CNN) is a deep learning algorithm which from an input image, it assigns importance to various features by decomposing the image and

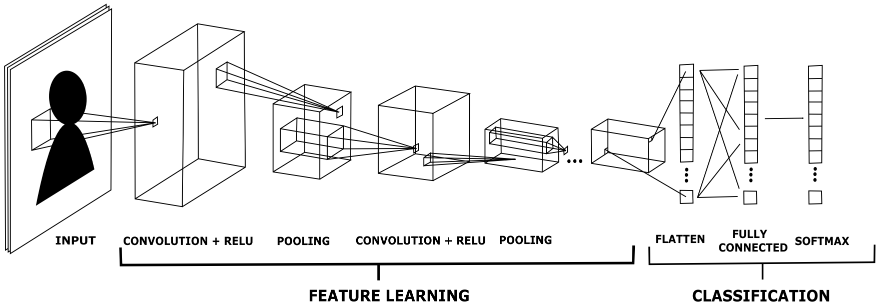


Fig. 9 Convolutional neural networks (CNN) architecture

compressing it into simple features hence it can differentiate one from another [32]. The CNN architecture use layers that convolve the inputs with filters and compresses them, the objective of the convolution operation is to extract the low-level features from the input. Then a pooling layer responsible for reducing the spatial size of the convolved features to decrease the computational power required to process the data. Lastly a fully connected (FC) layer with a softmax classification technique for learning non-linear combinations of the extracted features. This architecture is shown in the Fig. 9.

For the emotion detection part of this work a VGG-like network is used. A VGG net is a deep convolutional network developed by Oxford's Visual Geometry Group [33] which is publicly available online. The specific architecture used is shown in Table 3.

The training of the network was made with two merged public databases. The first one is the FER2013 dataset from the Kaggle competition [34], it contains seven facial expressions in 35,887 images. The second is the KDEF database [35] (Karolinska Directed Emotional Faces) which contains 4900 pictures on facial emotional expressions for 70 different models. The KDEF database images had to be adjusted to have the same format of the FER2013 database, that meant to extract the face in B&W and resize it to 48×48 pixels. Figure 10 shows the training graph through 1134 epochs.

The results obtained with this network are shown in the Fig. 11.

The proposed configuration effectively gives a grade for each of the facial expressions detected and works with no noticeable delay on the live camera recording. Therefore, the grade of happiness or sadness detected can be used to train the ANFIS system. Therefore, this configuration helps as a social connector that can promote social interaction between their relatives and friends; besides, this emotion classification may detect daily activities or moods to address any unusual activity and addressing those rare activities in early stages [36].

Table 3 VGG-like network for emotion classification

Layer type	Output size	Filter size/Stride	Block
Input image	48 × 48 × 1	3 × 3, k = 32	1
CONV (Relu, BN)	48 × 48 × 3	3 × 3, k = 32	
CONV (Relu, BN)	48 × 48 × 3	3 × 3, k = 32	
POOL	24 × 24 × 32	2 × 2	
Dropout	24 × 24 × 32		
CONV (Relu, BN)	24 × 24 × 64	3 × 3, k = 64	2
CONV (Relu, BN)	24 × 24 × 64	3 × 3, k = 64	
POOL	12 × 12 × 64	2 × 2	
Dropout	12 × 12 × 64		
CONV (Relu, BN)	12 × 12 × 128	3 × 3, k = 128	3
CONV (Relu, BN)	12 × 12 × 128	3 × 3, k = 128	
POOL	6 × 6 × 128	2 × 2	
Dropout	6 × 6 × 128		
FC (Relu, BN)	64		4
Dropout			
FC (Relu, BN)	64		5
Dropout			
Softmax	7		6

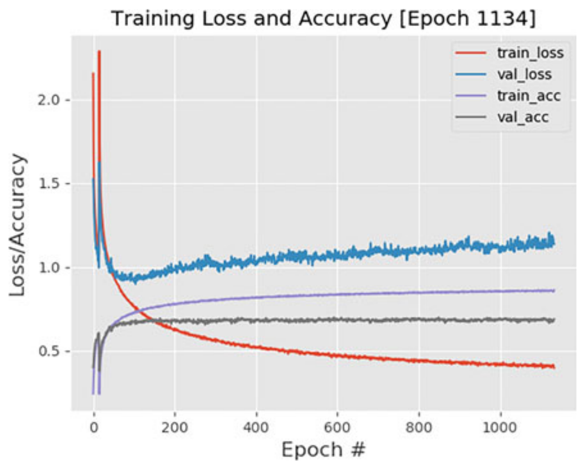


Fig. 10 Training graph for VGG-like network

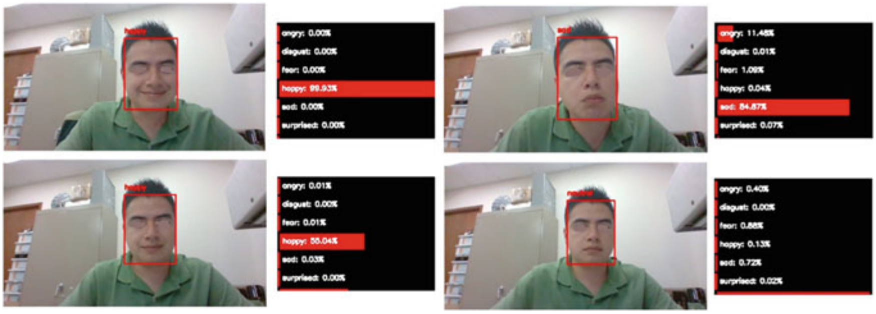


Fig. 11 Emotion classification results

4 Gamification

Since the 1980s decade, there are references of 'gamifying' applications; however, since the 2000s decade, several definitions of Gamification terms have been defined. In 2003, Pelling used gamification to create game-like interfaces for electronic devices [37]. In 2008, Terill defined the concept as *taking game mechanics and applying to other web properties to increase engagement* [38]. In 2011, Huotari and Hamari defined it as *a process of enhancing a service with affordances for gameful experiences in order to support the user's overall value creation* [39]. In 2011, Deterding et al. defined it as *the use of game elements and game-design techniques in non-game contexts* [40]. In 2015, Chou defined it as *the craft of deriving fun and engaging elements found typically in games and thoughtfully applying them to real-world or productive activities* [41]. Therefore, the main goal of gamification is to increase the motivation of users by using game-like techniques and applying them effectively in the real world to influence user's behavior and improve user's skills, competencies and creativity [42].

Besides, the use of gamification within a device can improve enjoyment, health care and promote social interaction in elderly people [43]. Ponce et al. [23] proposed including social factors in the design process by implementing a gamification strategy to send stimuli to change consumer behavior. Mendez et al. [44] developed a three-step framework to propose a tailored HMI using a fuzzy logic system in connected thermostats to teach, engage and motivate users to save energy. Moreover, that framework was adapted to take advantage of that connected thermostat platform to promote social interaction and physical activity for the elderly people [45].

The Internet maintains and enhances social relationships through email, instant messaging, social networks, discussion forums and blogs have positive adoptions among elderly people. Besides, technology can change daily life in three ways [8]:

1. It should shape social contexts regarding age-specific needs.
2. It should improve the contact's quality and create meaningful relations.
3. It should compensate for losses and burdens that individuals may face throughout adulthood.

4.1 Kaleidoscope Framework

Kappen proposed the kaleidoscope framework that has intrinsic and extrinsic motivation to promote physical activity in the elderly people over 50 years old [46].

- Intrinsic Motivation
  - Autonomy: Customization, purpose, independence. Related to the improvement of fitness performance, comfortable routines, incremental progression, reinforcing success, internalizing rewards and responsibility. Freedom of modulating fitness routines, structured routines, modularity, choice of changing goals, reinforcing profession through visual and verbal feedback.
  - Competence: Engagement based, achievement-based, performance-based. It is related to the complexity of activity routines, challenges with the repetitions of the activity, focus on remembering activity steps, and ease of understanding.
  - Relatedness: Relationships, sharing, preferences. This element is related to fostering social connections. Sharing achievements and experiences, setting an example for peers within the fitness activity domain, exchanging feedback with peers and trainers, and being validated for performance by the trainer and doctors.
- Extrinsic Motivation: Encompass factors of external regulation, identification, and integration. Moreover, these motivations are not as valued by older adults, and tangible rewards are mostly related to food.
  - Rewards
  - Incentives
  - Leaderboards
  - Points
  - Badges.

4.2 Octalysis Framework

Chou [41] proposed a complete framework based on extrinsic, intrinsic, positive, and negative motivation. Figure 12 displays the framework proposed by him.

- Extrinsic motivation: People are motivated because of external recognition or economic rewards.
- Intrinsic motivation: People are motivated due to inner motivation; the activity itself is rewarding on its own without tangible goals to achieve.
- Positive motivation: The activity is entertaining because people feel successful, happy, and powerful.
- Negative motivation: People engages in the activities because the user fears lose something.

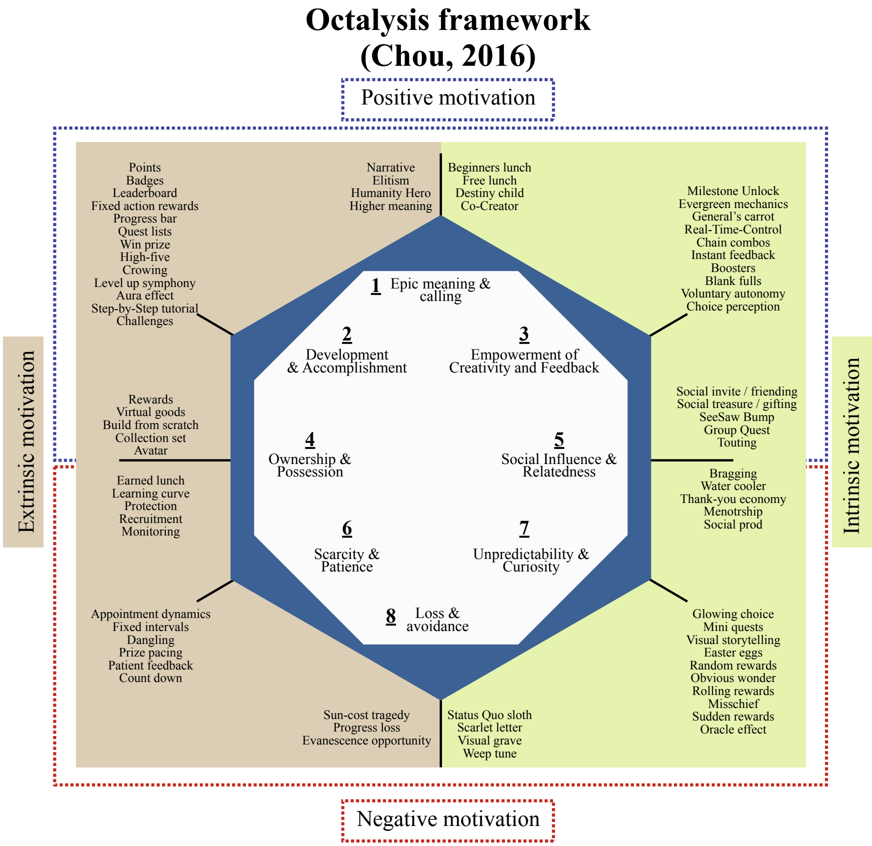


Fig. 12 Octalysis framework proposed by Chou [41]

- Core 1: Epic meaning and calling: People believe that what they do is more significant than themselves.
- Core 2: Development and accomplishment: People believe they are succeeding, progressing, developing skills, achieving mastery, among others.
- Core 3: Empowerment of creativity and feedback: People realize a creative process by trying several combinations to achieve goals.
- Core 4: Ownership and possession: This core is known as the desire core. People believe and feel they are in control of something.
- Core 5: social influence and relatedness: People are motivated due to social elements.
- Core 6: Scarcity and impatience: People want something because it is challenging to have it.
- Core 7: Unpredictability and curiosity: People are engaged due to the uncertainty of what is going to happen next. This core is behind the gambling addiction.
- Core 8: Loss and avoidance: People try to prevent something terrible to happen.

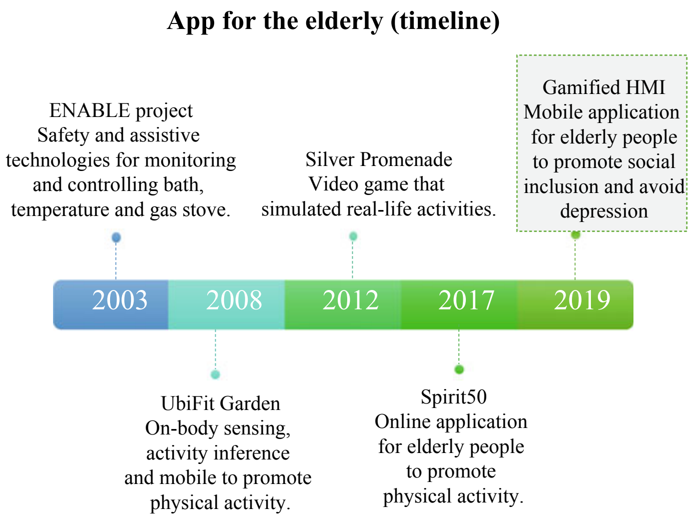


Fig. 13 Timeline of daily activity projects that considered game elements

Figure 13 shows a brief timeline of four projects for the elderly that included game elements within the platform.

5 Proposal

Figure 14 shows the trained ANFIS system for the detection of facial expression and nonconventional behavior through voice survey to propose a gamification strategy to engages the elderly people to promote social inclusion and avoid depression or

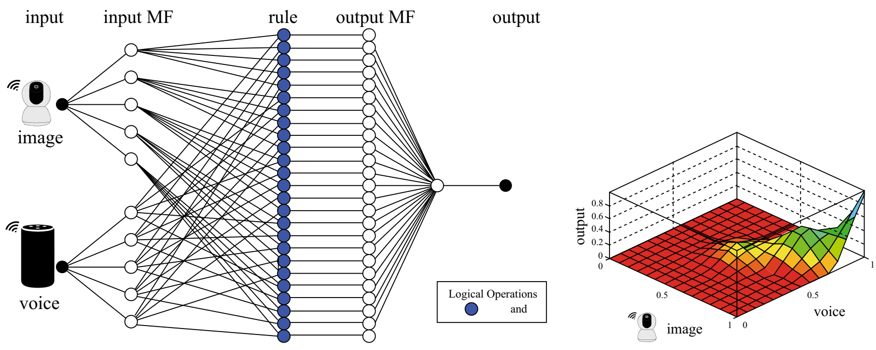


Fig. 14 Trained ANFIS using image and voice as input elements and the output the gamified elements required for the HMI

sadness in those individuals. The premise is that by taking advantage of household appliances, it is possible to promote social interaction in elderly people and avoid activities that cause sadness in the elderly.

Figure 15 shows that the input values of the ANFIS system are the camera and the Alexa device; the input membership function for both devices are ranged from the sad face (from 0 to 0.25), a bit sad (from 0 to 0.5), neutral (from 0.25 to 0.75), a bit happy (from 0.5 to 1) to a happy face (from 0.75 to 1). Besides the sad face uses a Z-shape membership function, the faces from a bit sad to a bit happy uses the triangular membership function and the happy faces uses the S-Shape membership function. The expected output membership functions ranged from extrinsic, negative, positive to intrinsic motivations. Furthermore, the output values are more biased to the negative side of the lower values because it is considered that the elderly people tend to feel isolated [47]; thus our proposal looks for social inclusion and physical activity. Therefore, the tailored gamified HMI will be more focused on showing extrinsic motivation through messages, videos, rewards and tips about physical activity that will depend on how active or inactive, and sad or happy the elderly people are.

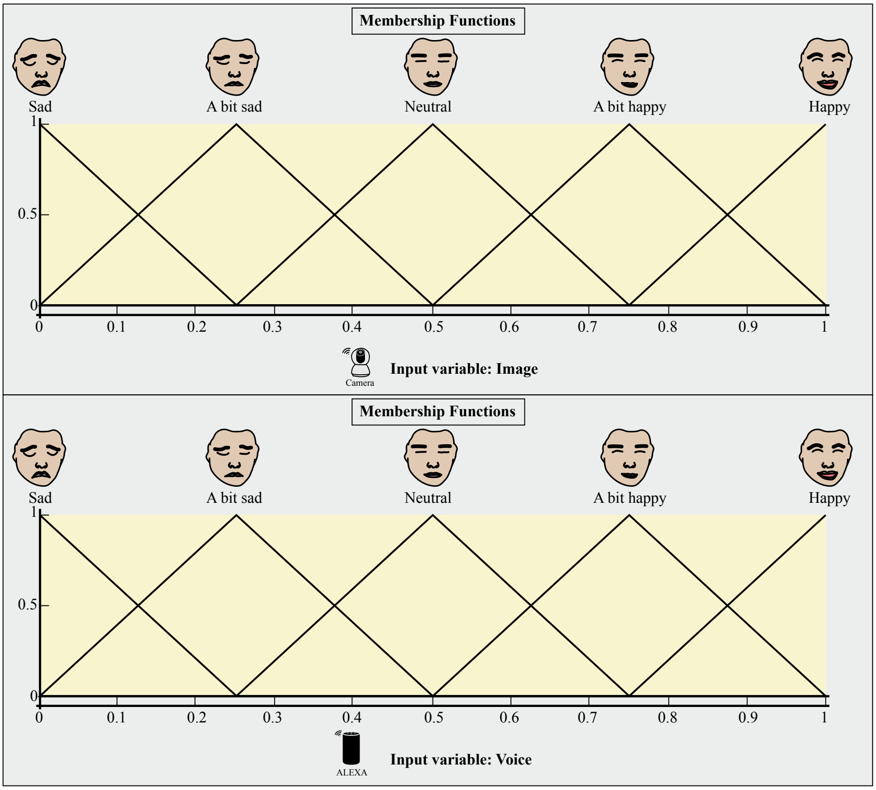


Fig. 15 Input membership functions of the trained ANFIS

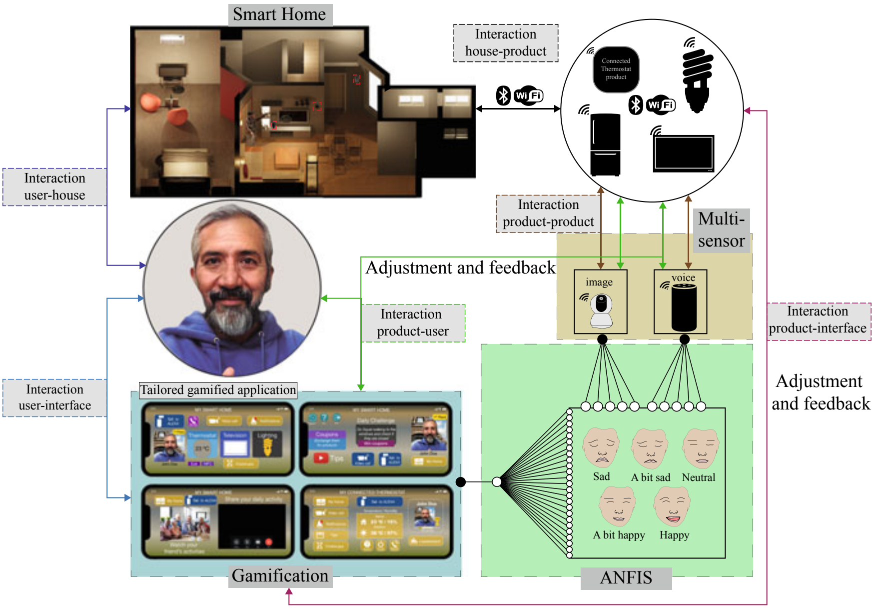


Fig. 16 Proposal diagram

Consequently, Fig. 16 shows how the proposal works. Six types of interactions are used to avoid elderly isolation: *user-house, house-product, product-product, product-user, product-interface,* and *user-interface*. As the elderly user spends most of the time at home, this user is continuously in touch with the household elements, such as doors, windows, furniture, household appliances, among others (*interaction user-house*). Then, the interaction between *house-product* determines which household appliances are mostly used. Moreover, when the elderly consumer interacts with the products, it is profiled the user's routines and activities, which are also monitored by the HMI (*interaction product-interface*), while Alexa and the camera take decisions for the user to get more comfortable the life inside the house (*interaction product-user*); for instance, Alexa can diminish the light intensity, or turn off the TV (*interaction product-product*). Through, the multi-sensor system, Alexa and the camera analyzes the elderly facial expressions and communicate by voice with the user to monitor any change in the elderly mood. Finally, throughout the ANFIS system (see Table 4), it is shown which gamification elements should be displayed in the tailored interface to promote social inclusion and physical activity in the elderly (*interaction user-interface*). The adjustment and feedback between the tailored HMI, the household appliances, and the camera and Alexa are required to continue engaging and motivating the user to be in touch with friends and family members, to improve habits, to exercise. Therefore, with this proposal, the elderly quality of life may improve.

Table 4 25 rules from the fuzzy system of the trained ANFIS

Rule	IF	AND	THEN
	IMAGE	VOICE	Gamified motivations
1	Sad	Sad	Extrinsic
2	Sad	A bit sad	Extrinsic
3	Sad	Neutral	Extrinsic
4	Sad	A bit happy	Extrinsic
5	Sad	Happy	Extrinsic
6	A bit sad	Sad	Extrinsic
7	A bit sad	A bit sad	Extrinsic
8	A bit sad	Neutral	Extrinsic
9	A bit sad	A bit happy	Extrinsic
10	A bit sad	Happy	Extrinsic
11	Neutral	Sad	Extrinsic
12	Neutral	A bit sad	Extrinsic
13	Neutral	Neutral	Negative
14	Neutral	A bit happy	Extrinsic
15	Neutral	Happy	Extrinsic
16	A bit happy	Sad	Extrinsic
17	A bit happy	A bit sad	Extrinsic
18	A bit happy	Neutral	Extrinsic
19	A bit happy	A bit happy	Positive
20	A bit happy	Happy	Extrinsic
21	Happy	Sad	Extrinsic
22	Happy	A bit sad	Extrinsic
23	Happy	Neutral	Extrinsic
24	Happy	A bit happy	Extrinsic
25	Happy	Happy	Intrinsic

Therefore, the tailored gamified HMI must consider mainly the extrinsic motivation to promote social inclusion and promote happiness in elderly people. The gamification elements required in the interface are: points, badges, rewards, leaderboard, progress bar, higher meaning, narrative, elitism, virtual goods, build from scratch, avatar, monitoring through social connectors as video call [41, 46]. If the elderly people are neutral either in voice and image, the gamification elements can also include progress loss, visual storytelling, random rewards, status quo, mentorship that helps the user become happier and social included. If the elderly user is a bit happy, then the interface can include real-time control, instant feedback, voluntary autonomy, and choice perception, social invite, group quest. Finally, if the user is totally happy, then the elements should include visual storytelling, random rewards, status quo, mentorship, real-time control, instant feedback, voluntary autonomy, choice perception, social invite, group quest.

6 Results

The literature review indicates that the most common gamification elements for the elderly to promote social interaction and physical activity are: feedback, social sharing, challenges, leaderboard, rewards, social connector, monitoring, and a profile [36, 41, 42, 45, 46, 48, 49]. Table 5 displays the gamification elements in a mobile device for the elderly. Those elements consider the extrinsic, intrinsic, positive, and negative motivation based on the Octalysis and Kaleidoscope framework. Figure 17 proposes a typical layout for the elderly user without considering any customization. This dashboard is a proposal for a connected thermostat, it shows the most common gamification elements that may appear in an interface whose primary purpose is to promote social interaction and physical activity. The right side of the screen has the purpose that elderly user can track his/her progress, compare with friends, check their weekly and monthly challenges, and see how rewards are available to be achieved. In the right upper side of the dashboard, it is displayed the daily challenge to promote physical activity in the elderly. The *Tips* button advises the elderly user on how to improve and learn more about physical activity and its benefits as wells as the kind of household appliances are in his/her home. The *Be a HERO* button works as an interaction button where the elderly user gives or receives advises of his friends. Finally, the video conference screen works as a social connector because the elderly user can call his/her family member or friends. With this video call screen, it is possible to determine through the face detection if the elderly user is sad or happy.

Based on the ANFIS proposal, in Fig. 18, it is displayed a tailored HMI based on an elderly user that is a bit happy. The first image is the interface for the smart home, which initially has connected the thermostat, television and lighting in the house. The second and third image displays an interface for a bit happy user as it includes real-time control, instant feedback, social invite, the points, badges, leaderboard and the monitoring through social connector (Alexa, and video call). Finally, the last image displays the connected thermostat interface, where it also includes positive and extrinsic motivation.

Moreover, if the elderly user is sad or depressed, the gamified interface will display elements that promote social inclusion as video calls or social sharing like feedback their peers through challenges. If the elderly user is neutral, the interface will display a sober layout to determine and recognize the user's interest; for instance, Fig. 19 shows the proposal of layout for this type of user; the left image shows the general display for the smart home, this type of user prefers a more sober layout where it is easy to have a call with friends or with Alexa.

Thus, the gamified application monitors the elderly mood in order to propose a tailored interface. If the user does not engage, then the interface re-adapts to propose

Table 5 Gamification elements to be used for the tailored HMI

CORE DRIVE	Motivation	Gamification	Sad	A bit sad	Neutral	A bit happy	Happy
1. Epic meaning and calling	Extrinsic	Narrative			×	×	×
		Humanity hero		×	×	×	×
		Higher meaning	×	×	×	×	×
2. Development and accomplishment	Extrinsic	Points	×	×	×	×	×
		Badges	×	×	×	×	×
		Leaderboard	×	×	×	×	×
		Progress bar	×	×	×	×	×
		Challenges	×	×	×	×	×
3. Empowerment of creativity and feedback	Intrinsic	Choice perception				×	×
		Real-time control		×	×	×	×
		Instant feedback	×	×	×	×	×
4. Ownership and possession	Extrinsic	Rewards	×	×	×	×	×
		Build from scratch					×
		Avatar				×	×
		Monitoring	×	×	×	×	×
5. Social influence and relatedness	Intrinsic	Social invite				×	×
		Mentorship				×	×
		Touting					×
		Social sharing	×	×	×	×	×
		Social connector	×	×	×	×	×
6. Scarcity and patience	Extrinsic	Patient feedback					×
7. Unpredictability and curiosity	Intrinsic	visual storytelling				×	×
		Random rewards					×
8. Loss and avoidance	Extrinsic	Progress loss	×	×	×		
		Status quo	×	×	×		

Fig. 17 General dashboard for a connected thermostat layout without any personalization

Fig. 18 Tailored HMI proposal for *a bit happy* elderly user

Fig. 19 Tailored HMI proposal for *a neutral* user

other elements. The Octalysis and Kaleidoscope frameworks help as a guideline to promote the gamification elements based on the motivations (intrinsic, extrinsic, positive, and negative), Table 5 displays those gamification elements. Besides, this proposal is just considering the gamified elements to be displayed on the interfaces; however, the design and distribution of this HMI may be improved by applying the ten heuristics proposed by Nielsen [50].

For the last three decades, it is proposed the inclusion of elderly people in the smart home by tracking their daily activities, detecting falls, remembering to them activities. However, to the best of the author's knowledge, any of these proposals have considered a tailored interface based on the elderly people mood to promote social inclusion and avoid depression by using devices as Alexa that allows having a conversation with the elderly or through video calls to promote social relatedness. This proposal aims to teach, engage and motivate users to feel included in the society by promoting interaction with Alexa, their peers and family.

7 Discussion

This chapter proposes the inclusion of Alexa and cameras to track the elderly people and check their daily status, their mood in order to improve their quality of life by promoting social inclusion and physical exercise. The multi-sensor system is used within a smart home environment to identify the physical characteristics of elderly people. Thus, the voice and face detection are evaluated on an ANFIS system to propose the personalized gamified elements that run in an HMI needed each type of user.

The emotions detected are ranged from a 5 type-scale: sad (the lowest membership function value), a bit sad, neutral, a bit happy, and happy (the highest membership function value). Those emotions are measured with the survey for elderly people presented in Table 2 [4]; these questions are asked to the elderly people through Alexa. Thus, the initial interaction begins during this questionnaire. During the initial tests that face detection was done using the webcam from the PC, and the facial expression concorded with the results. Therefore, the face detection can be used from the mobile or tablet camera the elderly individual is using.

Then, based on those ranges the ANFIS system, that is biased to the lower values to propose a gamified interface focused more on the extrinsic motivation, as this type of motivation is more accessible to measure through metrics, of how many times the individual is using the application or even the daily physical activities the individual is doing and how much time the user spends exercising [41, 46]. Besides, the application displays the gamification elements from Octalysis and Kaleidoscope framework that engage the elderly people to improve their social skills and physical activity. This initial approach takes in consideration that one family member is continuously in contact with the elderly user, thus if any urgency occurred, the application would contact the family member immediately to check if their elderly familiar is fine.

However, a clear disadvantage of this proposal is that the elderly user and the family members require to accept that the house will be permanently monitored. Moreover, the type of mobile phone or tablet that the elderly user has may be inconvenient, and the socioeconomic level where this type of product can be used. Besides, it is required that the elderly accept Alexa as an initial manner to interact. Another possible failure is the face detection; the face may display another feeling rather than what the program is detecting; however, the face detection algorithm can be updated with an artificial neural network that can train the user's faces and detect more accurately the user feelings. As it was mentioned on the Results section, the gamified interface can be improved by applying the ten Nielsen heuristics [50]: *visibility of system status, match between system and the real world, user control and freedom, consistency and standards, error prevention, recognition rather than recall, flexibility and efficiency of use, aesthetic and minimalist design, help users recognize, diagnose, and recover from errors, and help and documentation.*

This proposal intends to change the way of how the products are used, and to take advantage of the household appliances to make them more social, it means, to produce a social interaction between the user and the product. A product that can take into account the needs of the elderly to promote in and the unconscious way social interaction and therefore, improve their physical activity. With the multisensor system, it is possible to take advantage of sensors, or household appliances that use sensors to analyze the elderly pattern and propose, for instance, a customized application that best fit to the user.

8 Conclusion

In this chapter, a Multi-sensor system for helping elderly people by using gamification and artificial intelligence within a tailored HMI have been proposed. Elderly people will continue increasing, whereas younger people will decrement, since last year, the elderly population became more than the younger population. Therefore, the elderly people require to be considered, and as some of their problems have led to social isolation, it is needed a strategy that improves their quality of life. This strategy considers the interaction of the elderly people and the products; with the interaction between household appliances and Alexa, and gamification features within an application it is possible to engage and motivate the user to be socially included and improve their quality of life, too. Nowadays, it is not enough to have connected products that are not able to profile the elderly user and take actions based on the user tasks that benefit them without making intrusive decisions. With this proposal, it is intended to profile and know better the user in order to propose an accurate application that improves their social and physical skills without affecting their personality or without affecting their freedom. This HMI set the opportunity of creating an atmosphere where the connected products can interact with the elderly and propose social and physical activities that will help the user to feel included.

Acknowledgements Research Project supported by Tecnologico de Monterrey and UC Berkeley under the collaboration ITESM-CITRIS Smart thermostat, deep learning and gamification (https://citris-uc.org/2019-itesm-seed-funding/).

References

1. United Nations: World Population Prospects Highlights, 2019 revision Highlights (2019)
2. Sayago, S. (ed.): Perspectives on Human-Computer Interaction Research with Older People. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-06076-3
3. Kostopoulos, P., et al.: Enhance daily live and health of elderly people. Proc. Comput. Sci. **130**, 967–972 (2018). https://doi.org/10.1016/j.procs.2018.04.097
4. Meiselman, H.L. (ed.): Emotion Measurement. Elsevier; Woodhead Publishing, Amsterdam (2016)
5. Victor, C.R., et al.: The social world of older people: understanding loneliness and social isolation in later life. Open Univ. Press, Maidenhead (2009)
6. Kamin, S.T., et al.: Social contexts of technology use in old age. In: Kwon, S. (ed.) Gerontechnology. Springer Publishing Company, New York, NY (2016). https://doi.org/10.1891/9780826128898.0003
7. Borelli, E., et al.: HABITAT an IoT solution for independent elderly. Sensors **19**(5), 1258 (2019). https://doi.org/10.3390/s19051258
8. Kwon, S.: Gerontechnology: Research, Practice, and Principles in the Field of Technology and Aging. Springer Publishing Company, New York, NY (2017)
9. Harper, R. (ed.): Inside the Smart Home. Springer, London ; New York (2003)
10. Gutman, G.M., et al.: Smart home technologies supporting aging in place. In: Kwon, S. (ed.) Gerontechnology. Springer Publishing Company, New York, NY (2016). https://doi.org/10.1891/9780826128898.0011
11. Hargreaves, T., Wilson, C.: Smart Homes and Their Users. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-68018-7
12. Bühler, C., Knops, H.: Assistive Technology on the Threshold of the New Millennium. IOS Press (1999)
13. Abowd, G.D., et al.: The Aware Home: A Living Laboratory for Technologies for Successful Aging, 8
14. Bien, Z.Z., et al.: LARES: An Intelligent Sweet Home for Assisting the Elderly and the Handicapped, 8
15. VAALID. http://www.sabien.upv.es/en/project/vaalid/. Last accessed 22 Oct 2019
16. Kwon, O., et al.: Single activity sensor-based ensemble analysis for health monitoring of solitary elderly people. Expert Syst. Appl. **39**(5), 5774–5783 (2012). https://doi.org/10.1016/j.eswa.2011.11.090
17. Jacobs, P.G., Kaye, J.A.: Ubiquitous real-world sensing and audiology-based health informatics. J. Am. Acad. Audiol. **26**(9), 777–783 (2015). https://doi.org/10.3766/jaaa.15010
18. Elder Care. ORCATECH | OHSU. https://www.ohsu.edu/oregon-center-for-aging-and-technology/elder-care. Last accessed 22 Oct 2019
19. Woznowski, P., et al.: SPHERE: a sensor platform for healthcare in a residential environment. In: Angelakis, V., et al. (eds.) Designing, Developing, and Facilitating Smart Cities, pp. 315–333. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-44924-1_14
20. Pigini, L., et al.: Pilot test of a new personal health system integrating environmental and wearable sensors for telemonitoring and care of elderly people at home (SMARTA project). Gerontology **63**(3), 281–286 (2017)

21. Aladin: A Smart Home-Care Solution for the Elderly. Aladin Project | H2020 | CORDIS | European Commission. https://cordis.europa.eu/project/rcn/216735/factsheet/en. Last accessed 22 Oct 2019
22. Domalys Improve Vulnerable People's Life. https://www.domalys.com/en/. Last accessed 22 Oct 2019
23. Ponce, P., et al.: The Next generation of social products based on sensing, smart and sustainable (S3) features: a smart thermostat as case study. In: 9th IFAC Conference on Manufacturing Modelling, Management and Control, p. 6 (2019)
24. Jang, J.-R.: ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans. Syst. Man Cybern. **23**(3), 665–685 (1993). https://doi.org/10.1109/21.256541
25. Llinas, J., Waltz, E.: Multisensor Data Fusion. Artech House, Boston (1990)
26. Waltz, E.: Data fusion for C3I: a tutorial. In: Command, Control, Communications Intelligence (C3I) Handbook. EW Communications, Palo Alto, CA (1986)
27. Yesavage, J.A., et al.: Development and validation of a geriatric depression screening scale: a preliminary report. J. Psychiatr. Res. **17**(1), 37–49 (1982). https://doi.org/10.1016/0022-3956(82)90033-4
28. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**(3), 338–353 (1965). https://doi.org/10.1016/S0019-9958(65)90241-X
29. Ponce Cruz, P., Ramírez-Figueroa, F.D.: Intelligent control systems with LabVIEW. Springer, London; New York (2010)
30. Amazon Developer Services. https://developer.amazon.com/. Last accessed 22 Oct 2019
31. Jia, Y., et al.: Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia—MM '14, pp. 675–678. ACM Press, Orlando, FL, USA (2014). https://doi.org/10.1145/2647868.2654889
32. Balderas, D., et al.: Convolutional long short term memory deep neural networks for image sequence prediction. Expert Syst. Appl. **122**, 152–162 (2019). https://doi.org/10.1016/j.eswa.2018.12.055
33. Cao, Q., et al.: VGGFace2: a dataset for recognising faces across pose and age. In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), pp. 67–74. IEEE, Xi'an (2018). https://doi.org/10.1109/FG.2018.00020
34. Challenges in Representation Learning: Facial Expression Recognition Challenge. https://kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge. Last accessed 22 Oct 2019
35. Goeleven, E., et al.: The Karolinska directed emotional faces: a validation study. Cogn. Emot. **22**(6), 1094–1118 (2008). https://doi.org/10.1080/02699930701626582
36. Muñoz, D., et al.: Social Connector: A Ubiquitous System to Ease the Social Interaction Among Family Community Members. 30, 1, 13 (2015)
37. Werbach, K., Hunter, D.: For the win: how game thinking can revolutionize your business. Wharton Digital Press, Philadelphia (2012)
38. My Coverage of Lobby of the Social Gaming Summit. http://www.bretterrill.com/2008/06/my-coverage-of-lobby-of-social-gaming.html. Last accessed 30 Sept 2019
39. Huotari, K., Hamari, J.: Defining gamification: a service marketing perspective. In: Proceeding of the 16th International Academic MindTrek Conference on - MindTrek '12, p. 17. ACM Press, Tampere, Finland (2012). https://doi.org/10.1145/2393132.2393137
40. Deterding, S., et al.: From Game Design Elements to Gamefulness: Defining "Gamification", 7 (2011)
41. Chou, Y.: Actionable Gamification Beyond Points, Badges, and Leaderboards. CreateSpace Independent Publishing Platform (2015)
42. Stieglitz, S., et al.: Gamification. Springer, Heidelberg, New York, NY (2016)
43. Malwade, S., et al.: Mobile and wearable technologies in healthcare for the ageing population. Comput. Methods Programs Biomed. **161**, 233–237 (2018). https://doi.org/10.1016/j.cmpb.2018.04.026
44. Méndez, J.I., et al.: S4 Product Design Framework: A Gamification Strategy based on Type 1 and 2 Fuzzy Logic, 15

45. Mendez, J.I., et al.: Framework for Promoting Social Interaction and Physical Activity in Elderly People Using Gamification and Fuzzy Logic Strategy, 5 (2019)
46. Kappen, D.L.: Adaptive Engagement of Older Adults' Fitness through Gamification. ACM Press (2015)
47. Mohammad, S.A., et al.: Inclusion of social realm within elderly facilities to promote their wellbeing. Proc. Soc. Behav. Sci. **234**, 114–124 (2016). https://doi.org/10.1016/j.sbspro.2016.10.226
48. Bock, B.C., et al.: Exercise videogames, physical activity, and health: Wii heart fitness: a randomized clinical trial. Am. J. Prev. Med. **56**(4), 501–511 (2019). https://doi.org/10.1016/j.amepre.2018.11.026
49. Sardi, L., et al.: A systematic review of gamification in e-Health. J. Biomed. Inform. **71**, 31–48 (2017). https://doi.org/10.1016/j.jbi.2017.05.011
50. Heuristics for User Interface Design: Article by Jakob Nielsen, https://www.nngroup.com/articles/ten-usability-heuristics/. Last accessed 23 Oct 2019

A Novel Approach for Human Fall Detection and Fall Risk Assessment

Yoosuf Nizam and M. Mahadi Abdul Jamil

Abstract There are several studies concerned in providing quality health care services to all people. Among them, human fall detection systems play an important role, because fall is the main obstacle for elderly people to live independently and it is also a major health concern due to aging population. The three basic approaches used to develop fall detection systems include some sort of wearable, ambient or non-invasive based devices. Most of such systems are very often rejected by users due to the high false alarm and difficulties in carrying them during their daily life activities. Thus, this study proposes a non-invasive fall detection system based on the height, velocity, statistical analysis, fall risk factors and position of the subject from depth information. The proposed algorithm also utilizes fall risk factors of the user, during fall detection process to classify the subject with chances of fall such as high fall risk or low fall risk level. Thus, making the system adaptable to the physical condition of the user. The proposed system also performs a fall injury assessment after the fall event to alert for appropriate assistance and includes a fall risk assessment tool which can work independently to predict falls or to classify people with their fall risk levels. From the experimental results, the proposed system was able to achieve an average accuracy of 98.3% with sensitivity of 100% and specificity of 97.7%.

Keywords Depth sensor · Fall detection · Fall risk assessment · Daily activity classification

Y. Nizam (✉)
Department of Engineering, Faculty Engineering, Science and Technology, The Maldives National University (MNU), Malé, Maldives
e-mail: yoosuf.nizam@mnu.edu.mv

M. M. A. Jamil
Biomedical Engineering Modeling and Simulation (BIOMEMS) Research Group, Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia
e-mail: mahadi@uthm.edu.my

H. Ponce et al. (eds.), *Challenges and Trends in Multimodal Fall Detection for Healthcare*, Studies in Systems, Decision and Control 273,
https://doi.org/10.1007/978-3-030-38748-8_10

1 Introduction

Adaptive technology is an emerging research area since daily living assistance are very often needed for many people in today's aging populations including disabled, overweight, obese and elderly people. The main purpose of assistive technology is to provide better living and health care to those in need, especially elderly people who live alone. In order, to provide better living for them, it is important to have continuous human monitoring systems in their home to inform the health care representatives of any emergency assistance. Among such monitoring systems, fall detection systems are increasingly in interest since statistics [1, 2] has shown that fall is the main cause of injury related death for seniors aged 79 [3, 4] or above and it is the second common source of injury related (unintentional) death for all ages [5, 6]. Furthermore, fall is the biggest threat among all other incidents to elderly and those people who are in need of support [3, 7–16]. Accordingly, fall can have severe consequences for elderly people, especially if not attended in a short period of time [17]. Similarly, unintentional human fall represents the main source of morbidity and mortality among elderly [18]. Hence, accurate and autonomous human fall detection systems are very important to allow the elderly people to live independently without having to change their life style.

The three basic approaches used to develop human fall detection systems are wearable based devices, ambient based devices and vision based devices. The studies representing the three approaches are basically structured to solve the drawbacks of one another. For an example, ambient based devices used to solve the issues in wearable based devices and wearable devices also solves some of the problems that ambient sensors failed to handle. Before the advent of cheap depth sensors in the market, vision based devices using RGB camera used to take care of the main issue in wearable and ambient based devices with even higher accuracy at the cost of expensive systems and setup. Even with RGB cameras, the concerns arising regarding the acceptability and reliability of the fall detection systems are not limited rather added its own drawbacks such as capturing and recording of color videos leading to privacy issues. Additionally, the cost of the systems, camera calibrations, requirement of adequate lightening and setup are common issues.

The advent of cheap Red-Green-Blue-Depth (RGBD) cameras, has paved way to the development of novel systems to overcome the limitations of these previous works [19, 20]. The cheap depth sensors such as Microsoft Kinect sensor, can extract depth information of the objects in the scene even with very low lighting condition. The auto calibration capability and other features of the sensor can negligibly reduce the issues concerning with RGB cameras. One of the main advantage of the Kinect sensor is that it can be place in certain places according to user requirements [21], unlike the complex installation procedure of some RBG fall detection system. It is also worth noting that by using only the depth images it can preserve the privacy of users [21]. Since the proposed algorithm is based only on Kinect sensor, the three categories of studies using depth sensor is analyzed below to validate the chosen approach and the derived methodology.

The first sub-categories in depth sensor based approach used joint position or measurements and its movement with some thresholds for fall detection. The second sub-category used fusion of wearable and depth maps with machine learning or joint position for fall detection. The third sub-category employs machine learning or other classifier only on depth images. The research studies based on fusion of an initial devices and depth sensor is not very relevant to compare with the proposed algorithm, since their system design and performance are all subject to the drawback of wearable devices which is regarded as main causes of rejection of fall detection systems. Wearable devices are mainly rejected due to the inconvenience in carrying them during daily life activities [22]. In addition, they used the wearable device to generate any potential fall movement and to start the depth image based classifier to confirm fall. In such cases the capabilities of the depth sensor are not fully exploited and thus the actual accuracies that could be achieved are not realized. Generally, the overall performance of such systems solely depends on the effectiveness of the wearable device to identify potential fall movements.

In some of the studies, detection does not always depend on the wearable device because in certain situations like while changing cloths it is not possible to wear the device [20]. Therefore, in such cases the systems depend on the depth sensor. This requires a proper time synchronization between wireless initial device sampling rate and depth sensor frame rate. It is also impossible to access and control the Kinect embedded clock [23]. But it is important to synchronize the Kinect sensor and main system and the wireless initial device in order to properly integrate such fall detection systems which depend on one another and requires switching of fall authentication process between devices.

The other two approaches in depth map based approaches depends only on the depth image generated from the depth sensor. From the review of literature, it was also found that exploring the depth information alone can minimize the issues faced by the previous work for fall detection. The only issue that cannot be dealt is the limitation of the Kinect sensor's viewing spectrum, which can be solved using more than one sensor depending on the coverage requirement.

Previous works has used different techniques on the depth image to classify human fall from other activities of daily life. Some of the studies used extracted human joint measurement and movement over time to identify human fall. While others used machine learning or classifiers either on the depth image or extracted human features to detect human fall. The use of machine learning classification can have many problems apart from the computational cost. Additionally, off-line training can degrade the accuracy and the time when the image is extracted can play an important role in identifying the lying posture. They used different algorithms to segment human subject from the depth image. Some works, developed their own preprocessing directly on the raw data, while it is not possible to achieve the established auto calibration of the Kinect sensor with manual preprocessing, even though it was not primarily developed for fall detection. If any developed preprocessing cannot auto calibrate when a subject enters and exit into the view of the sensor, then fall detection algorithm working on top of the preprocessing will not receive adequate information to make an accurate decision.

The approaches that is based only on the depth images and used joint measurements instead of classifiers basically use the distance of human joints from floor plane and their vertical velocities to classify and authenticate human fall from other activities of daily life. Various joints such as head joint height from floor, centroid height and their respective velocities are fed into an algorithm with some thresholding to identify any falling action. With this approach, some fall detection algorithms cannot work in case of occlusion, because it cannot calculate the distance to the ground [24]. The use of joint height in fall detection shows good performance for falls ending on the floor but it has failed if the end of the fall is occluded behind furniture [25]. It also showed that it could be solved using velocity just before the occlusion.

The closely related works to the proposed algorithm employed joint measurements and its movement over time for fall detection. Basically they all, used selected skeleton joints or features of the subject with a predefined algorithm for fall detection. The algorithms either uses a fixed or adaptive threshold within a flow to make the decision. One of the study used [26], a statistical approach with features using a Bayesian framework. It is very clear from the literature that, a lot more effort still remain, to device algorithms to follow the changes in the selected key features of the subject to fully utilize the capabilities of the depth sensor. It was found that a low-computational algorithm with statistical analysis of the key features of the subject can significantly minimize the issue of obstacles blocking the view of the subject. This could on the other hand, make the system more stable especially by avoiding the computation hungry machine learners and other classifiers.

It is also worth to be noted that none of the related works so far had utilized any fall risk level estimation during fall detection. Studies conducted on fall risk assessments were primarily aimed to identify potential fall risk patients for nursing homes or hospitals. This was achieved either through questionnaires or using sensors to identify likely physical weakness of the patients that may cause falls. This was then used to categorize patients with high fall risks or low fall risks to provide better healthcare and avoid fall injuries.

This study incorporates a robust fall risk level estimation protocol within the fall detection algorithm to adapt appropriate parameters depending on the movement of the user. The inclusion of fall risk factors in the proposed approach during fall detection process is never implemented to our knowledge and it is expected that the fall risk identification could assist in improving fall detection process and also reduce computation costs. Since fall risk factors can help to adapt the fall detection process depending on the risk level of the users. This is because the nature of fall and characteristics of other activities of daily life differ with fall risk levels. For users with high risk of falls, the fall detection algorithm could switch to intensive detection process and for users with low fall risks, the algorithm could track changes after a gap, thus reducing computational costs. Since fall risk level of the user also changes overtime, such an incorporation could make the proposed fall detection to outperform over the available systems. The assessment of the consequences of fall or fall injury is also included in the proposed system to evaluate the significance of injury and inform for appropriate healthcare, which is also a contribution of this work. The proposed is also incorporated a fall risk assessment tool which can work independently to predict falls or to classify people with their fall risk levels.

2 Fall Detection

2.1 Overview of the Method

The proposed methodology for fall detection and identification of fall risk level uses the depth information generated from Kinect sensor. The parameters required for the proposed fall detection algorithm and the fall risk assessment procedure are computed from the generated depth information.

As per the characteristics of daily activities of human life, some of them can be classified using the height change pattern alone. For activities that are similar like fall events, velocity for the duration of the movement is also required. This is because such activities are found to have similar height change pattern as falls. Like intentionally lying on floor and fall from standing is very similar in nature in terms of height change pattern. The only major distinguishing factor is the changes of height over time. Therefore, the basic components for fall detections are the changes in velocity and height which was applied in different order depending on the fall risk levels of the subject. Fall risk levels of the subject is a measure of physical weakness or any difficulties the subject is facing during their daily life activities that may have high chances of falls than other normal elderly person.

The following Fig. 1, shows the skeleton generated from the sensor labeled with joints considered in the proposed algorithm. Table 1, shows the combinations of joints considered for the computation of different parameters used in the proposed algorithm for the classification of human fall from other activities of daily life. The joint data utilized by proposed fall injury estimation are also listed in the following Table 1. The joints marked in red (in Fig. 1), are key joint considered in the proposed methodology. If any of them are not visible (available) and/or are not accurately detected, then the joints marked in pink color in Fig. 1 and Table 1 (left shoulder, right shoulder, left hip and right hip) are used instead. For an example, if torso joint is not visible, then the average of left and right hip is used instead.

Combinations of labeled joints in Fig. 1 are used for the calculations of subject's height, speed, velocity, acceleration, position of joint, activity, fall risk factors and fall injury estimations as described in Table 1. Speed is calculated for the computation of velocity, since velocity consists of magnitude and direction components. Rate of change of velocity is considered in this proposed works, because it can clearly show the difference. Acceleration is considered where the changes in velocity is not clear enough to distinguish the changes in pattern or variation. Fall risk factors are calculated from foot, arm and trunk to identify any difficulty the subject is facing during their normal daily life activities which may cause them to fall easily. Activity is detected from the movements of head and arms. Fall injury estimation simply uses

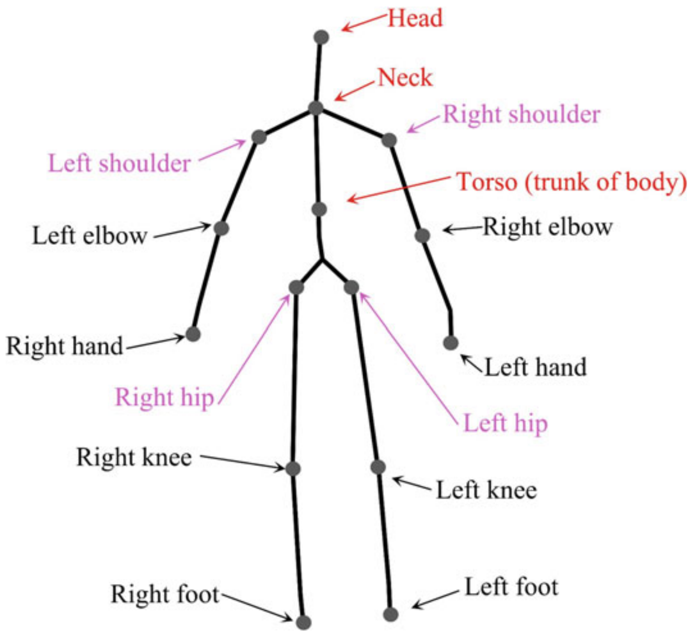


Fig. 1 Illustration of joints considered in the proposed algorithm

Table 1 Joint coordinates used for the different parameters

Parameters	Head	Neck	Shoulders	Torso	Hips	Elbows	Hands	Knees	Foots
Height	✓	-	-	✓	-	-	-	✓	-
Speed	✓	✓	✓	✓	✓	-	-	-	-
Velocity	✓	-	✓	✓	✓	-	-	-	-
Acceleration	✓	-	✓	✓	✓	-	-	-	-
Position	✓	✓	✓	✓	-	✓	✓	-	-
Fall risk	-	-	-	✓	✓	✓	-	-	✓
Activity	✓	-	-	-	-	-	✓	-	-
Fall injury	✓	-	✓	✓	✓	✓	✓	✓	✓

all the major joints, because it will require all the joint data to assess how the subject stood up and any changes that may increase the chances of unintentional fall in future or simply the fall risk level.

Fall risk assessment in this study simply refers to the analysis of the physical capabilities of the user, which was primarily used in hospitals and nursing homes to identify the patients with their fall risk level. The proposed fall risk assessment is integrated with the proposed fall detection system and it can work independently to measure the level of fall risks (chances of fall) as a clinical tool for patients and elderly.

Some of the features of the proposed fall risk assessment are also used in the proposed fall detection algorithm. These features of the fall risk assessment were used during the fall detection (fall risk factors) process and after a fall event (fall injury estimation). Some of these features of fall risk assessment were used in fall detection process, because nature of falls and other activities of human life differ depending on physical strength of the subject or the user. Therefore, common fall risk factors were used during fall detection process to adapt the algorithm for people different fall risk levels. Fall injury estimation after a fall event was used for alerting appropriate assistance. This was simply aimed to assess the ability of the subject to return to normal activities and to measure the level of injuries for low impact falls only.

The use of some of the features of fall risk assessments during fall detection process, indeed furnished the proposed fall detection algorithm by enhancing the detection ratio and improving capabilities. Furthermore, the added fall injury estimation after a fall event, also increased the capabilities of the fall detection system. The overall proposed system was also enhanced through the addition of this proposed independent fall risk assessment tool together with the fall detection mechanism.

2.2 *Activity Classification*

Classification of human activities is an important research topic in the field of computer vision and rehabilitation. It is also increasingly in use for many applications including intelligent surveillance, quality of life (such as health monitoring) devices for elderly people, content-based video retrieval and human-computer interaction [27–30]. There has been plenty of researches conducted to automatically recognize human activities, yet it remains a challenging problem. Many approaches are used to identify how human moves in the scene. Techniques employed includes tracking of movements, body posture estimation, space-time shape templates and overall pattern of appearance [31–36].

Identification of characteristics of daily life activities is important in order to classify them. The characteristics of the activities can help to identify any uniqueness or dissimilarities between activities which in turn will support in distinguishing them. The characteristics of activities were derived in terms of height change pattern, rate of change of velocity, deviation of the height and position of the subject during and after the movement.

This identification of characteristics of daily activities together with the pattern and the rate of change is important to classify them, especially unintentional human fall. The pattern of change is referred to the variation of subject's height with respective to floor during any of the activities and the rate of change is the changes in velocity of the subject during that period. Deviation of height is the similar pattern of change of height except that this is the statistical standard deviation of the changes in height pattern. The changes in the pattern itself refers the height of the subject in different frames. In some cases, acceleration can help to identify the difference, where the changes in velocity does not clearly show the variation of speed during the movement.

The daily activity classification procedure used in the proposed algorithm along with the parameters including height, velocity, speed, statistical analysis and position of joints are discussed in this section. Height is typically referred to the distance from head to foot (height of the subject), but in this work height is calculated by taking the distance difference between head and floor at any given time even if the subject is sitting or standing using the following Eq. 1.

$$\text{Height (H)} = \frac{|\text{Ax} + \text{By} + \text{Cz} + \text{D}|}{\sqrt{\left(\text{A}^2 + \text{B}^2 + \text{C}^2\right)}} \tag{1}$$

where: x, y and z are the coordinates of the joint. A, B, C and D are the floor plane x-coordinates, y-coordinates, z-coordinates and w-coordinates respectively.

Equation 1, is a general equation to compute height of any joint from floor plane or distance of that joint with respective to floor. The proposed algorithm used the height of head, torso, and knee joints. The height of head from floor is used to identify any movement that leads to a drop of head to the direction of floor from any side. Since for any kind of unintentional fall the head will drop down until it hits the floor or an object. The pattern of height drops for unintentional fall supposed to be downward to the direction of floor from any side. A similar height changes is observed for intentional lying on floor, except the duration and amount of fluctuations. The higher variation (height drop) observed in unintentional movement is due to the speed of action from the gravitational force of earth and the body weight. It was also found that sometimes this variation does not give any significant information to be used in activity classification. Therefore, velocity is used to differentiate such activities. Similarly, there are also activities that possesses similar velocity pattern. Without significant differences in these parameters, it would be difficult to implement a classifier either using machine learning or threshold approach. Therefore, the proposed algorithm is very carefully designed to deal such situations, with a unique flow of height, velocity, acceleration and motion detection.

Speed is the rate of change of position of the subject, calculated from the change in the position of torso and head center over time. Velocity is the change of position per unit time in a particular direction. This velocity is used to classify activities in terms of rate of change of head and torso joint. Velocity is calculated using the key joint position after considering the direction of the movement, by dividing it with the time taken for the movement. Acceleration is calculated by taking the difference of two velocities over the time taken between them.

Unintentional human fall is classified from other activities of daily life by considering the height drop pattern and rate of change of velocity of the movements for simple cases. Simple cases, is referred for those unintentional falls that are easily distinguishable from other activities either due to the nature of the fall event or no obstacles blocking the view. For an example, a high impact falls where the velocity is very high.

The decision on the classification procedure is made from fall risk levels. For high fall risk cases, fall detection will check velocities twice and than height. With these two parameters, the fall detection procedure will also employ statistical analysis or joint positions before a fall alert is triggered. Statistical analysis will be run, if the velocities are not showing any significant values. Statistical analysis, evaluates the standard deviation of subject's height for the duration of action which can easily identify the drop of the height. Statistical analysis will also measure the total drop of height during that period of time and the number of frames where the height had shown a smaller value than the previous frame. Thus, this analysis will give very strong information to classify human fall. Standard deviation (σ) is computed using the following formula in Eq. 2.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{2}$$

where $\sum$ means "sum of", μ is mean of the data sets, x is a value in data set, N is the number of data elements and i is the i-th data value.

This analysis is basically used only if the fall risk level is high or the position of joints fail to classify the activity. For low fall risk cases, the classification procedure will be based on the velocity and height change combinations. This will be the simple and low computational, human fall classification approach where changes in velocity of the body joint and height of joints between frames are the basic components. The most significant parameter here is the velocity, because for any unintentional action the immediate distinguishing characteristics will be the speed and of course, the direction. The direction component of velocity and the height are all considered to ground level (vertically to floor plane) in this study. If the changes in velocity are not noticeable then the changes in height was considered and in some cases the acceleration are also used, to identify fall events. This is because acceleration is calculated from velocities and therefore it could show the fluctuations that is not visible with velocity. To make sure all unintentional fall events are detected, activity detection was also used in some cases where changes in velocity and height does not show significant variations.

The proposed algorithm, uses position of joints to confirm fall events. The basic idea behind fall confirmation is to see if the key joint positions of the subject at the time of a potential fall alert is near or on the floor. This process is executed after the proposed fall detection algorithm senses a potential fall event, therefore the subject's joints are supposed to be on the floor level. Especially the upper body parts, hence the key features of the subject in this process is the joint position representing head, torso, left shoulder, right shoulder, left elbow, right elbow, left hand, right hand and the neck. The positions of these joints are simply considered to be the height of the joints which is calculated from the joint coordinates (x, y and z values).

Other variables employed in the proposed algorithm to find fall risk levels are step_symmetry, trunk_sway, and spread_arm. The step_symmetry, is an estimate of the step inequality which can be realized by measuring the left and the right step lengths. Step length is the distance between left and right step, which can be measured using x-axis or z-axis coordinates depending on the direction of the movement. If the direction of the movement is on x-axis then the following Eq. 3, was used to compute the Step_symmetry and if the direction of the movement is on z-axis then z-values were used instead of x-values in the Eq. 3.

$$\text{Step\_symmetry} = (R\_foot_x - L\_foot_x)_{PF} - (R\_foot_x - L\_foot_x)_{CF} \tag{3}$$

where, *R_foot* is the right foot, *L_foot* is the left foot, *x* is the x-value or x-axis coordinate value, *PF* is the previous frame and *CP* is current frame.

Trunk_sway is a measure of how far the subject bends, side to side from trunk and it was calculated by taking the changes of torso position with respect to the hip position. The amount of bend or the Trunk_sway value is simply an average of the difference of torso and hip position between frames. This variation can be calculated by taking x-axis values, if the direction of the movement is on z-axis as shown in the following Eq. 4 and using z-axis values instead of x-values if the direction of the movement is on x-axis.

$$\text{Trunk\_sway} = \frac{\left(Torso_x - \left(\frac{L\_hip_x + R\_hip_x}{2}\right)\right)_{PF} + \left(Torso_x - \left(\frac{L\_hip_x + R\_hip_x}{2}\right)\right)_{CF}}{2} \tag{4}$$

where, *L_hip* is the left hip position and *R_hip* is the right hip position.

The last parameter of fall risk factors (Spread_arm), is a measure of how much the two arms are spread. This parameter gives important information about the fall risk level and to predict falls. Since during a loss of control of the body due slip or any other fall like event, it is normal to spread the arms to balance the body, especially common among those who are afraid of falls. This parameter was computed by taking the difference of torso position and the two (left and right) arms (hands). Similarly, like Trunk_sway, spread_arm is also calculated from x-axis if the direction of the movement is on z-axis using the formula in the Eq. 5 and using z-axis values instead of x-values if the direction is on the x-axis. The average of the distance of the two arms to the torso were threshold between the frames to identify any action where the subject is spreading the arms to balance the body or trying to hold something to control the body.

$$\text{Spread\_arm} = (\text{Torso\_hand\_diff})_{CP} - (\text{Torso\_hand\_diff})_{PF} \tag{5}$$

Here, Torso_hand_diff is the difference of distance between torso and hand which is calculated using the following Eq. 6.

$$\text{Torso\_hand\_difference} = \left(\frac{\left(Torso_x - R_{hand_x}\right) + \left(Torso_x - L_{hand_x}\right)}{2} \right) \tag{6}$$

where, *R_hand* is right elbow joint and *L_hand* is the left elbow joint.

2.3 Proposed Algorithm

The proposed algorithm for human fall detection consists of six processes which is divided into two groups (2 stages). The first group is responsible for the identification of any potential fall event and the second group is responsible for the confirmation or verification of the fall event with the parameters described in previous section. The six processes are data acquisition and generation of skeleton data, computation of fall detection parameters, fall detection with fall risk factors, normal fall detection, fall confirmation, and statistical analysis and verification respectively. The first four processes belongs to the first stage (or first group) of fall detection, and are basically responsible for the identification any potential fall event using changes in velocity, height and fall risk levels. The last two processes belonging to the second stage of fall detection, which will confirm or verify the fall event depending on the fall type and availability of joint information. The following Fig. 2, illustrates the overall proposed fall detection algorithm with boxes for each of the six processes and the fall injury estimation (third stage of the proposed algorithm) after a fall event.

The proposed fall detection starts from Process 1, which acquires data from the sensor and generates skeleton data. The generated data are then feed to Process 2, a sub-preprocessing block for Process 6 and to the processing in third stage. The data collected will be stored in Buffer 1, which is available to all other processes. Process 2, computes the required fall detection parameters and stores in Buffer 2, which is also available to all other processes. The computed, fall risk factors and velocity from this process is used to decide which process to be executed next. If the fall risk factor is flagged as high, than Process 3 is executed to detect fall with risk factors. In case, if the fall risk factors are normal or low, then the computed velocity from Process 2, are used to either start Process 4, for normal fall detection or start Process 5, for immediate fall confirmation.

The Process 5, is dedicated for fall confirmation if the velocity from Process 2 or Process 3, is flagged as high and no activity or high acceleration is flagged from Process 4. In case, if the Process 5, couldn't confirm a fall, then the Process 6, will be executed which plays the role of fall verification. The Process 6, are also executed from Process 3 and Process 4. It is primarily designed to detect fall using statistical

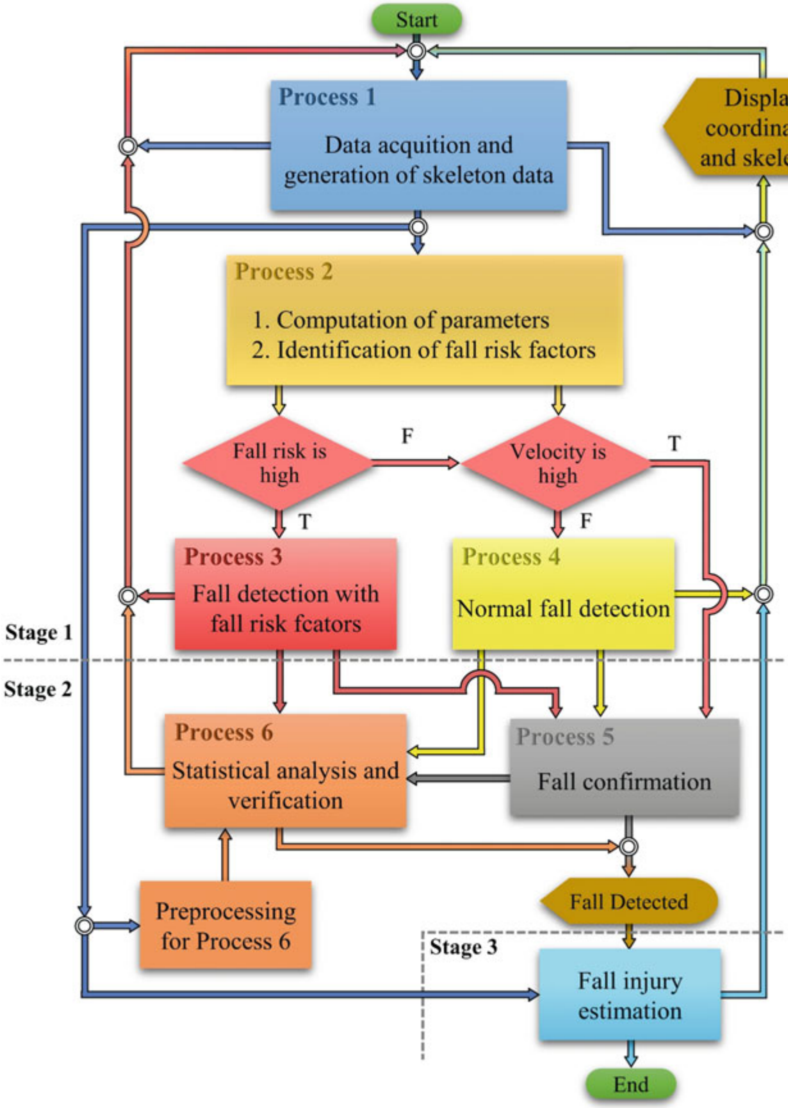


Fig. 2 Process flow of the proposed algorithm

analysis. The small block just below the Process 6, block is dedicated to do all preprocessing required for fall detection using statistical analysis. With the proposed algorithm in Fig. 2, a fall is normally confirmed by Process 5, or verified through Process 6, or directly detected from Process 6. Once a fall event is confirmed, the third stage of the proposed algorithm which is responsible for fall injury estimation are executed.

It is to be noted that the proposed algorithm in Fig. 2, immediately executes fall confirmation process (Process 5) after it notices an abnormal velocity change. But an abnormal increase in velocity can be caused from many activities where the pattern and the changes in direction of height are different, such as increase in walking speed (increased step frequency or step length), sitting on chair or floor, lying down on floor or bed and running. For an example while walking and running, the height changes are supposed to give a straight fluctuation on to any direction and for other activities mentioned above the direction will be straight down or most probably diagonally down to y-axis of the image. Even though the algorithm considers the changes in height, only if it could not sense an increase in velocity, the algorithm was able to differentiate such activities. Since there are many possible flows that a fall event could be detected from the proposed algorithm, almost all possible fall event in any settings will be captured by the algorithm.

Fall injury estimation, after a fall event is one of the novelty of the proposed fall detection algorithm. This is primarily aimed to identify the consequences of the fall event and to generate the required fall alert depending on the injury levels. At the beginning of this stage, fall event will be already confirmed and the procedure in this stage simply identifies the level of injury. Fall injury estimation in this study refers to the analysis of the physical injuries after a fall event for alerting appropriate assistance. It is incorporated with the fall detection system, because the aim of the proposed system was meant to allow elderly and people with special assistance to live independently on their own. Since there may be cases where the subject can recover from the fall event without any assistance or the injury levels are minor that it can be handled by the subject on their own. This fall injury estimation is simply aimed to assess the ability of the subject to return to normal activities and to measure the level of injuries for low impact falls only. Low impact falls is any unintentional fall where the subject can recover from fall event without external assistance.

Assessment of the consequences of the fall or the impact on the subject after a fall event is as important as the detection of fall. It includes the assessments of how the subject stood up, walking style, duration of placement of the two legs, movement of the feet and changes in walking speed. This assessment can play an important role for the medical staffs to decide on whether to provide assistance to the user or to ignore the fall alert as normal. Therefore, with an accurate fall injury estimation, the proposed fall detection system will be further enhanced to generate fall alert only when assistance is needed for the user. This would also help to improve the overall ambulatory care systems by allowing them to know the level of injury or the impact of the fall and the suitable assistance required.

Injury levels are estimated using some of the common fall risk assessment parameters including the fall risk factors and few other fall risk assessment variables. The parameters used for fall injury estimation are the changes in head height, angle and distance between head and right foot, angle and distance between head and left foot, arm_spread, step_symmetry, trunk_sway, walking speed (gait speed) and step_continuity. The first two parameters are used identify if the fallen subject has stood up. The last four parameters ('step_symmtry', 'trunk_sway', 'step_continuity' and ‘walking speed’) are used to decide the injury level. The variable ‘walking speed’ is the gait speed or the speed calculated from the step length over the time taken.

At the beginning of this stage, movements will be identified first and then it will check if the fallen subject has stoop up as shown in Fig. 3. This will be accomplished using the height change pattern of the head and changes of the angles between head and the two legs. How much the arms are spread during the period when the head height is increasing are also used to identify if the subject has stoop up. If the procedure couldn’t find any action similar like standing up from lying on floor or on flat surface, then it will end the fall detection algorithm as shown in Fig. 3. If the fallen subject has stood up, then the mentioned fall injury parameters will be computed. In case, if none of the parameters has reached its thresholds then the generated ‘fall event’ will be marked as ‘normal’ and the proposed fall detection algorithm will be restarted. This is because the fallen subject recovered from the fall event and there was no sign of any injury. In case, if the any one of the four parameters had reached

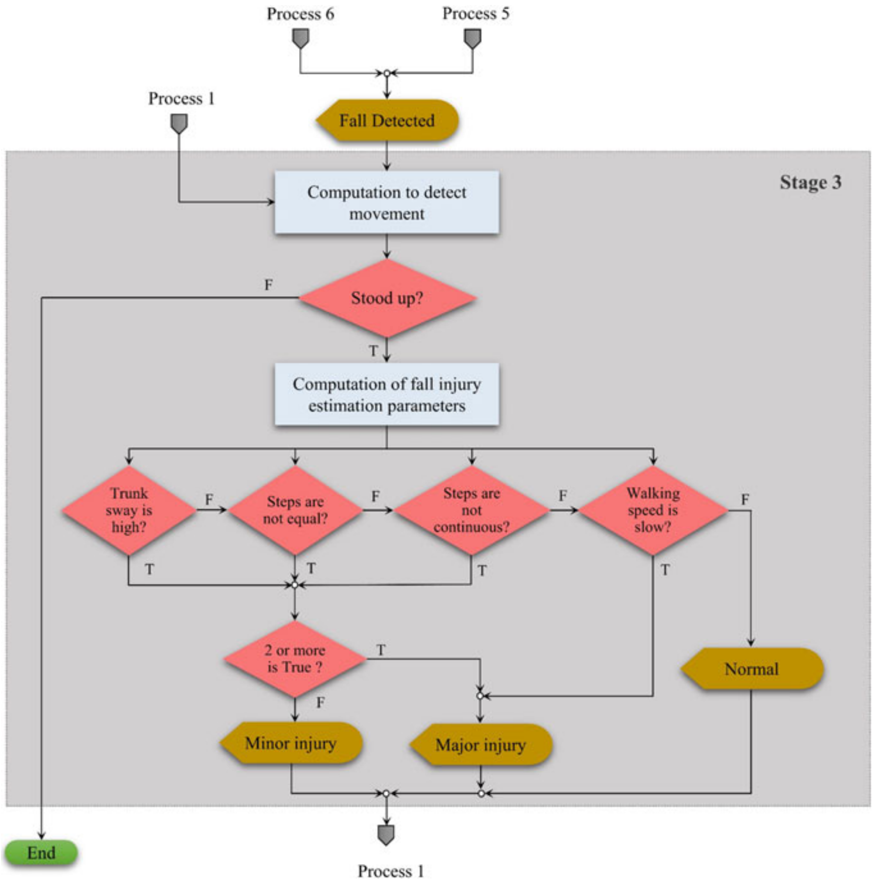


Fig. 3 The processing of the third stage of the fall detection algorithm

its thresholds, then the parameters are compared to decide if the injury is 'Major' or 'Minor'. If the parameter 'walking speed' or any two of the other parameters had reached their thresholds then the detected 'fall event' is marked as 'Major injury', here 'walking speed' has given a higher priority. In any other case, the detected 'fall event' is marked as 'Minor injury'. In any of this case (if the subject had stood up after the fall event), the proposed fall detection algorithm will be restarted, because it is very likely that a second fall event may occur.

3 Fall Risk Assessments

Fall risk assessment is a measure of the physical strength or the ability of the subject to withstand from unintentional falls due to aging, weaker body, balancing problems, unsteadiness during walking, fears of fall or any other issues that may often lead to falls. In this study, injury estimation is simply referred to the analysis of injuries caused and fall risk assessment of the subject after a fall event. This includes the changes in walking style, stride duration and other fall risk assessment parameters to identify the changes in physical condition of the subject which can measure the level of injury caused. All these parameters are based on the determinants of gait (DoG). Thus, a gait analysis on the identification of the deviations of normal gait can predict when a subject is at a greater risk of falling. The same parameters are used for Fall Risk Assessment and identification of fall risk factors for fall detection. Injury Estimation uses some of the parameters of Fall Risk Assessment and other variables determining the position of subjects.

The factors screened for fall risk assessments in the proposed algorithm are derived from different standardized fall risk assessment tools such as TINETTI Balance assessment tool and STRATIFY. The proposed fall risk assessment tool includes a questionnaire based section and device based assessment. The questionnaire based section, is used to find the history of falls, physical and mental strength of the user. They include three main questions, to find information about the number of falls in past six months, steadiness while standing or walking and fear about falls. If the user had experienced a fall event in the past six months, question will also be asked to know if it had caused any injury. Once this section of the fall risk assessment is filled, the remaining need to be generated from the developed system using the depth sensor. To generate the device based assessment, the user has to perform a predefined actions (user to sit on a chair at the beginning of the assessments and stand up from sitting on chair, walk for 3 meters at normal speed, turn around and walk back to the chair, turn and sit down on chair). The data extracted from the user actions are assessed based on the parameters discussed in previous section and analysis is conducted to identify fall risk levels.

In total, six assessments are conducted where, the first assessment is based on questionnaires and the remaining five using the variables discussed in previous section. Each assessment contains one or more parameters which contribute a score to the final assessment. For each of the variables, scores are given to any one of the three risk levels. The higher the risk level, the higher is the score. If the risk level of a variable is normal, then the score given is zero point and one points if the risk level is moderate. For higher risk levels than moderate, the highest score given is two points.

The questionnaires based section account for up to 26% of the total assessment and includes three scores for the three questions. The maximum score for this section will be 5 points. For the first question, if there were no falls in past six months, then the score given should be zero point and if there is only one single fall event without any injury then the score given should be one point. For more than one fall events or if the user was injured in the any one of the fall event, then the score given is two points. The second question also includes three levels of scores, zero point if user does not experience steadiness, one point if the user experiences steadiness and two points if the user experiences high steadiness during standing or walking. The third question had two levels of scores, zero point if the user had no worries about fall and one point if the user worries about fall.

The remaining five assessments, analyses the gait stability using temporal, spatial and positional variables. All the scores in this section are generated by the system, based on the data extracted from the actions performed by the user. The second section of the assessment includes temporal parameters such as gait speed, step duration, stride time variability and swing time variability. The last two parameters are to be calculated from the changes in consecutive stride duration and swing duration. The third section, includes parameters that make use of step width, step length and height of the joints. They include parameters that checks whether the swinging limb clears the floor (raises up) and it pass the stance foot (weight bearing limb). If the swinging limb does not pass the stance limb or in worst case if the swinging limb does not clear the floor, indicates that the gait is weak. This is because in such case, the user will be mostly walking very slowly and/or by dragging the legs. If the limb clears the floor and pass the stance foot, then the score given will be zero else one point and this will be computed for the two legs.

The last three sections, will check for step symmetry, step continuity and trunk sway for the duration of the fall risk assessment protocol. Step symmetry performs step equality check and gives one point if on average they appear unequal else zero point. Step continuity checks if the steps (left and right) are placed continuously on average, if true, zero points will be given else one point. The last section, will check if the user's upper body moves side to side while walking. This section contains three level of risks. If the upper body of the user moves from side to side, the score given will be two points, because it indicates that the user has weak balance. If there was no trunk sway but the knee flexes while walking or the arm are spread to balance the body then the score given will be one point and zero point if there is no trunk sway, no flexion of knee and arms.

As the user performs the actions, the system will generate the parameters and allocate scores for each of them based on the ranges and limits. At the end, all the scores will be summed up including questionnaire based scores to produce a final score and then it will be matched with any one of the fall risk levels in Table 2.

Table 2 Fall risk level indicators

Score	Fall risk level
Score ≤ 8	Normal or low risk level
$9 \leq$ Score ≤ 12	Moderate risk level
Score ≥ 13	High risk level

4 Results and Discussion

The proposed system had been tested with different activities of daily life such as walking, running, sitting on chair, sitting on floor, lying on floor and falling from standing. The experimental results showed that the proposed system can classify human fall from other activities of daily life using height changes and velocity of the subject together with the position of the subject after fall. The extraction of joint position after a potential fall activity showed good performance in differentiating human activities and confirming human fall. The order of velocity and height of the subject used in the proposed algorithm greatly helped in eliminating human activities that are closely analogous to falls, before going to the final stage for fall confirmation. Thus, the algorithm proved to reduce the error rate since those activities that are mostly misinterpreted as fall (such as lying on floor) is classified out here before fall confirmation.

As a comparison to some of the related works, the proposed system showed good sensitivity. Most of the related works failed to identify at least one fall event, which can make such systems unreliable. Some of the studies showed better accuracy but lacked in sensitivity which is essential for fall detection system, because for a medical staff or care giver it is acceptable to attend a non-fall event than missing a real fall event. Generally, the proposed system lacked in specificity while the related works lacked in sensitivity. Lacks in specificity simply means a false alarm from non-fall activity and lacks in sensitivity is simply missing a real fall alarm. The reduction in sensitivity of the related works might be caused from their scope to reduce false alarm of the system, by adjusting the thresholds such as velocity threshold by assuming that a fall event that requires assistance will have a high impact (falling velocity is high and/or with injuries). In doing so they have sacrificed any possible detection of low impact fall event (falling velocity is low with no or very minor injuries). The proposed system is designed to detect all the fall events, even if the event is of low impact or simply a slip while lying down on floor. This is because, the first criteria for a fall detection system is to reduce false alarm of a fall event and reducing false alarm from non-fall activities comes second.

Since the proposed algorithm uses instant speed and the changes in height as the basis for activity classification, the system was evaluated for the accuracy of detection of movements and the identification of the differences of the observed parameters for different activities of daily life. The instant speed for the activities changes overtime and therefore the speed needs to be extracted at the right time to get the real value to classify the activity. Thus, the proposed algorithm extracts speeds at different process and switches to other process depending on the observed values

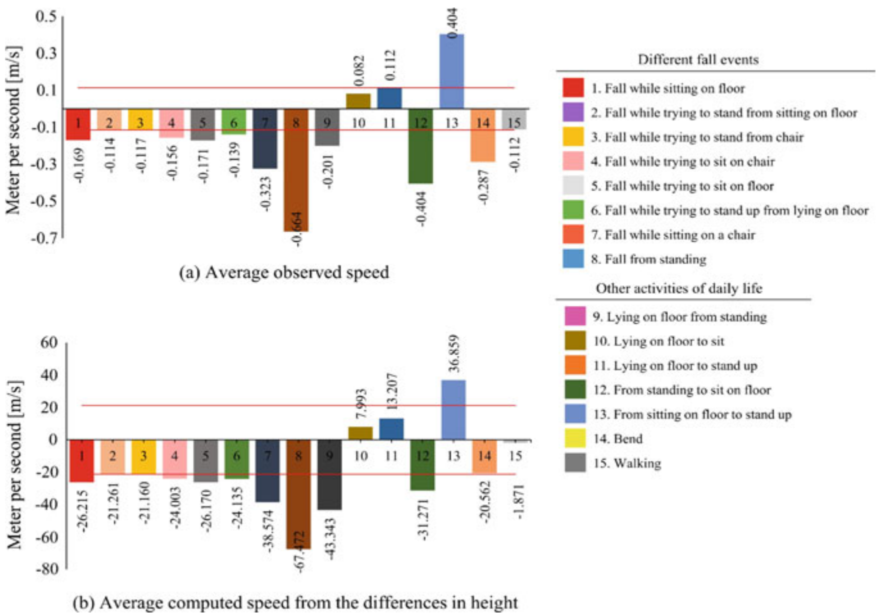


Fig. 4 Average observed and computed speed (in m/s) for the activities

to avoid any misinterpretation. Similarly, the average speed for the activities did not give very strong information about nature of the activity and it was also found not very relevant to use in fall detection. Since average speed can be calculated at the end of the event and the fall detection algorithm depends on it, then the system will be subject to unreliability and response time issues. Since, the average speed at the end of the event does not show high distinguishing differences as shown in Fig. 4. This figure illustrates the average observed speed and the average computed speed from the changes in height from the beginning of the event to the end of the event (the distance difference of head position from the beginning to the end of the event or the distance travelled by head). The part (a), of the Fig. 4, shows the average observed speed by the system from the instant speed generated throughout the duration of the event and part (b) shows the average speed computed from the distance travelled by the head for the same activities.

In Fig. 4, the red line indicates the minimum average speeds observed from the fall events of the respective graph. Even though, all the fall events reach the minimum indicated levels, there were some other activities that also reached this level. The non-fall events that crossed the limit are the event 9, 12, 13 and 14 which is standing to lying on floor, standing to sit on floor, sitting on floor to stand up and bend over respectively. It can be concluded that the average speeds for the activities can be applied for fall detection along with other procedures to eliminate those activities that gives higher average speed.

Table 3 Confusion matrix data for the available system (AS) and the proposed algorithm (PA)

Total = 20		Predicted		Total actual
		Falls	None-falls	
Actual	Falls	True Positive (TP) AS = 8, PS = 10	False Negative (FN) AS = 2, PS = 0	10
	None-falls	False Positive (FP) AS = 0, PS = 1	True Negative (TN) AS = 10, PS = 9	10
Total predicted		AS = 8, PS = 11	AS = 12, PS 9	20

Since the signs of the values denote the direction component of the velocity, the average speed for the activities in number 10, 11 and 13 are with positive values. Since they represent event that are going up (height increasing) and for the other activities, the height was dropping downward and therefore the signs of those values were negative.

The proposed system was also benchmarked against an available software [37] from (http://www.robofest.net/FDR), representing a study that used height and velocity for fall detection [38]. This software was developed using Microsoft SDK for Kinect and thus, requires Windows 7 operating system with a minimum of 2 core 2.66 MHz processor. Once Kinect SDK drivers (v1.0.0) are installed, this software can be setup by installing the application. At the end of the installation, the setup creates a shortcut at the desktop which can be used to launch the application.

This system was used to benchmark against the proposed algorithm by simulating the same activities in the same environment for this software and for the proposed system. Ten fall events and ten non-fall activities were performed by one persons at the same location (distance from the sensor to the location of the subject) for this available system (AS) and the proposed algorithm (PA). The following Table 3, shows the number of detected activities for the simulations conducted for the available system [37] and the proposed algorithm respectively.

The results in Table 3, for the available system shows an accuracy of 90% with a sensitivity of 80% and a specificity of 100%. Whereas the results of the simulations conducted in this benchmarking with the proposed algorithm shows an accuracy of 95% with a sensitivity of 100% and a specificity of 90% as per the data in Table 3. The available system identified all the non-fall activities but failed to identify two real fall events and the proposed algorithm identified all the fall events but failed to identify a non-fall event. It is also worth to note that the available system is subject to the draw backs of not preserving the privacy of users, because it is using color streams of the sensor. At the same time the available system also uses height and velocity thresholds from skeleton data. Thus, the performance is solely dependent on accurate detection of the joints and other issues such as obstacles.

From the results, it can be concluded that the available system is very specific, and the proposed algorithm is very sensitive. The available system is very specific in the sense that it makes sure not to generate any false alarm. The threshold used for the height and velocity were lowered to detect only high impact falls or falls that possess high velocity with rapid changes in height. Whereas the proposed algorithm makes sure to detect any possible fall events and in doing so it is subject to detect some fall like activities as well. The only activity that the proposed algorithm misclassified is lying on floor which is very similar to fall events. The proposed algorithm is actually detecting those lying on floor that are faster than usual as fall events. The proposed algorithm is purposefully designed like this to identify any low impact falls or falls that are caused while lying on floor. As discussed, it is acceptable to have false alarm from non-fall event (where assistance is not required) than missing a fall event that requires assistance. It is very clear from the results that by lowering the thresholds, the system becomes very specific (reduces false alarm) but it is very often likely to miss a fall event that requires assistance.

The main argument here is about the sensitivity and specificity or simple how sensitive the system is in classifying a fall event from other activities of daily life and how specific the system is to reduce any false alarm in order to reduce unnecessary attendance of medical staffs to assist the user. Thus, a tradeoff is often required while deriving an algorithm for fall detection, either to make it very sensitive or specific. A sensitive system will detect all the fall events often with some few false classifications of other similar activities as fall events. Whereas a specific system will make sure to detect only fall events even by ignoring some fall events that are similar to other activities of daily life such as lying on floor. Therefore, the aim of the proposed algorithm was to detect all the falls (not to miss any fall event that would require assistance) even with false alarms from other daily life activities to ensure quality healthcare.

5 Conclusion

This study proposed a human fall detection system based on human height, velocity, statistical analysis, fall risk level of the user and position of body computed from depth images generated by the Microsoft Kinect sensor. From the investigation conducted on the limitations and major drawbacks of the depth map based fall detection approaches, it was found that the use of statistical analysis and fall risk level during fall detection can reduce these issues to a greater degree. In order, to adapt the algorithm for people with difference fall risk level, the determinants of gait were also studied. The deviation in gait were used to predict (fall risk level) falls during fall detection and some of the gait constraints were used to identify the fall injury after such an event which accounts for the major contributions of the proposed algorithm. These gait constraints were also used to develop a fall risk assessment tool which can work independently of the proposed fall detection in classifying people with their fall risk level. Apart from that, the characteristics of the daily life activities especially those that are closely similar to unintentional fall events were studied to derive the proposed fall detection algorithm. The experimental results showed that the algorithm used on the system can accurately distinguish fall movements from other daily activities of daily life with an average accuracy of 98.3%. The system was also able to gain a sensitivity of 100% with a specificity of 97.7%. The proposed system accurately distinguished all the fall events from other activities of daily life, even though it failed to identify some lying down on floor from fall. The included fall risk assessment tool also showed promising results in generating fall risk levels from the simulated walking. Irrespective of this, further improvements are required on the computation of the angles for lower body posture identification. The proposed fall detection system could also be further improved by considering additional joints especially in the final stage for fall confirmation. Additional work is required to derive a better method to classify human fall and lying on floor from standing, since these are the two activities that are very similar to one another in terms of height change pattern. A special attention is also required to see how the system performs in classifying unintentional falls of people with different heights. This is because the duration of fall event for a shorter person will be less than for a taller person. This results in rapid height change pattern for shorter people and therefore it is often likely to miss important information in velocity computations. Since the algorithm is designed to adapt to the differences in physical conditions of the people (fall risk level) and the algorithm is so far tested only with simulated activities by healthy volunteers, it has to be extensively tested and verified with people from different settings (those living in community, hospitals, nursing home and any such elderly care center) in their real-life activities. Since the sensor can segment two more people accurately, the system needs to be upgraded to detect human fall of more than one person. This needs proper segmentation and tracking of the subjects within the view of the sensor. The issue of the obstacle also requires special attention, because with the issue of the viewing spectrum, the arrangement of the furniture and other equipment in the user's location can make the system to generate fatal errors. This indeed could be solved by considering having two more sensors configured to cover all the critical areas.

References

1. Griffiths, C., Rooney, C., Brock, A.: Leading causes of death in England and Wales–how should we group causes. Health Statistics Quarterly **28**(9) (2005)
2. Baker, S.P., Harvey, A.: Fall injuries in the elderly. Clin. Geriatr. Med. **1**(3), 501–512 (1985)
3. Stevens, J.A., Corso, P.S., Finkelstein, E.A., Miller, T.R.: The costs of fatal and non-fatal falls among older adults. Injury Prev. **12**(5), 290–295 (2006)
4. Kannus, P., Sievänen, H., Palvanen, M., Järvinen, T., Parkkari, J.: Prevention of falls and consequent injuries in elderly people. The Lancet **366**(9500), 1885–1893 (2005)
5. Bourke, A., O'brien, J., Lyons, G.: Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. Gait & Posture **26**(2), 194–199 (2007)
6. Kangas, M., Konttila, A., Lindgren, P., Winblad, I., Jämsä, T.: Comparison of low-complexity fall detection algorithms for body attached accelerometers. Gait & Posture **28**(2), 285–291 (2008)
7. Gostynski, M.: Prevalence, circumstances and consequences of falls in institutionalized elderly; a pilot study. Sozial-und Praventivmedizin **36**(6), 341–345 (1990)
8. Gurley, R.J., Lum, N., Sande, M., Lo, B., Katz, M.H.: Persons found in their homes helpless or dead. N. Engl. J. Med. **334**(26), 1710–1716 (1996)

9. Lin, L.-J., Chiou, F.-T., Cohen, H.H.: Slip and fall accident prevention: a review of research, practice, and regulations. J. Safety Res. **26**(4), 203–212 (1996)
10. Lord, C.J., Colvin, D.P.: Falls in the elderly: detection and assessment. In: Engineering in Medicine and Biology Society, 1991. Proceedings of the Annual International Conference of the IEEE 1991, vol. 13, pp. 1938–1939 (1991)
11. Lord, S.R., Sherrington, C., Menz, H.B., Close, J.C.: Falls in Older People: Risk Factors and Strategies for Prevention. Cambridge University Press (2007)
12. Sadigh, S., Reimers, A., Andersson, R., Laflamme, L.: Falls and fall-related injuries among the elderly: a survey of residential-care facilities in a Swedish municipality. J. Commun. Health **29**(2), 129–140 (2004)
13. Salvà, A., Bolíbar, I., Pera, G., Arias, C.: Incidence and consequences of falls among elderly people living in the community. Med. Clin. **122**(5), 172–176 (2004)
14. Teasell, R., McRae, M., Foley, N., Bhardwaj, A.: The incidence and consequences of falls in stroke patients during inpatient rehabilitation: factors associated with high risk. Arch. Phys. Med. Rehabil. **83**(3), 329–333 (2002)
15. Tinetti, M.E., Williams, C.S.: Falls, injuries due to falls, and the risk of admission to a nursing home. N. Engl. J. Med. **337**(18), 1279–1284 (1997)
16. Almeida, O., Zhang, M., Liu, J.-C.: Dynamic fall detection and pace measurement in walking sticks. In: High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability, 2007. HCMDSS-MDPnP 2007, pp. 204–206. IEEE (2007)
17. Shany, T., Redmond, S.J., Narayanan, M.R., Lovell, N.H.: Sensors-based wearable systems for monitoring of human movement and falls. Sens. J. IEEE **12**(3), 658–670 (2012)
18. Harrington, L., Luquire, R., Vish, N., Winter, M., Wilder, C., Houser, B., Pitcher, E., Qin, H.: Meta-analysis of fall-risk tools in hospitalized adults. J. Nurs. Adm. **40**(11), 483–488 (2010)
19. Zhang, C., Tian, Y., Capezuti, E.: Privacy preserving automatic fall detection for elderly using RGBD cameras. In: Computers Helping People with Special Needs, pp. 625–633 (2012)
20. Kwolek, B., Kepski, M.: Human fall detection on embedded platform using depth maps and wireless accelerometer. Comput. Methods Programs Biomed. **117**(3), 489–501 (2014)
21. Kepski, M., Kwolek, B., Austvoll, I.: Fuzzy inference-based reliable fall detection using Kinect and accelerometer. In: International Conference on Artificial Intelligence and Soft Computing, pp. 266–273. Springer, Heidelberg (2012)
22. Koshmak, G., Loutfi, A., Linden, M.: Challenges and issues in multisensor fusion approach for fall detection: review paper. J. Sens. **2016**, 1–12 (2015)
23. Cippitelli, E., Gasparrini, S., Gambi, E., Spinsante, S., Wåhslény, J., Orhany, I., Lindhy, T.: Time synchronization and data fusion for RGB-depth cameras and inertial sensors in AAL applications. In: IEEE International Conference on Communication Workshop (ICCW), pp. 265–270. IEEE (2015)
24. Nghiem, A.T., Auvinet, E., Meunier, J.: Head detection using kinect camera and its application to fall detection. In: 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA), pp. 164–169. IEEE (2012)
25. Rougier, C., Auvinet, E., Rousseau, J., Mignotte, M., Meunier, J.: Fall detection from depth map video sequences. In: Toward Useful Services for Elderly and People with Disabilities, pp. 121–128. Springer, Heidelberg (2011)
26. Zhang, Z., Liu, W., Metsis, V., Athitsos, V.: A viewpoint-independent statistical method for fall detection. In: 21st International Conference on Pattern Recognition (ICPR), pp. 3626–3630. IEEE (2012)
27. Kovashka, A., Grauman, K.: Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2046–2053. IEEE (2010)
28. Liu, H., Tang, H., Xiao, W., Guo, Z., Tian, L., Gao, Y.: Sequential Bag-of-Words model for human action classification. CAAI Trans. Intell. Technol. **1**(2), 125–136 (2016)
29. Ryoo, M.S.: Human activity prediction: early recognition of ongoing activities from streaming videos. In: IEEE International Conference on Computer Vision (ICCV), pp. 1036–1043. IEEE (2011)

30. Ryoo, M., Chen, C.-C., Aggarwal, J., Roy-Chowdhury, A.: An overview of contest on semantic description of human activities (SDHA) 2010. In: Recognizing Patterns in Signals, Speech, Images and Videos, pp. 270–285. Springer, Heidelberg (2010)
31. Bouchard, G., Triggs, B.: Hierarchical part-based visual object categorization. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, pp. 710–715. IEEE (2005)
32. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: Tenth IEEE International Conference on Computer Vision. ICCV 2005, pp. 1395–1402. IEEE (2005)
33. Parameswaran, V., Chellappa, R.: Human action-recognition using mutual invariants. Comput. Vis. Image Underst. **98**(2), 294–324 (2005)
34. Ramanan, D., Forsyth, D.A.: Automatic annotation of everyday movements. In: Advances in Neural Information Processing Systems, pp. 1547–1554 (2004)
35. Rao, C., Shah, M.: View-invariance in action recognition. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, vol. 2, pp. II–II. IEEE (2001)
36. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: Proceedings of the 17th International Conference on Pattern Recognition. ICPR 2004, vol. 3, pp. 32–36. IEEE (2004)
37. Kawatsu, C., Li, J., Chung, C.: FDR (Fall Detection & Reporting System) V1.0 (2013). https://www.robofest.net/FDR/. Accessed 13 March 2018
38. Kawatsu, C., Li, J., Chung, C.: Development of a fall detection system with Microsoft Kinect. In: Robot Intelligence Technology and Applications 2012, pp. 623–630. Springer, Heidelberg (2013)