Journal of Visualized Experiments

Design and Analysis for Fall Detection System Simplification

--Manuscript Draft--

Article Type:	Invited Methods Article - JoVE Produced Video
Manuscript Number:	JoVE60361R4
Full Title:	Design and Analysis for Fall Detection System Simplification
Section/Category:	JoVE Engineering
Keywords:	Fall detection; human activity recognition; machine learning; healthcare; ambient assisted living; sensors; multimodal approaches; classification.
Corresponding Author:	Hiram Ponce Universidad Panamericana Mexico City, Mexico City MEXICO
Corresponding Author's Institution:	Universidad Panamericana
Corresponding Author E-Mail:	hponce@up.edu.mx
Order of Authors:	Lourdes Martinez-Villaseñor
	Hiram Ponce
Additional Information:	
Question	**Response**
Please indicate whether this article will be Standard Access or Open Access.	Open Access (US$4,200)
Please indicate the **city, state/province, and country** where this article will be **filmed**. Please do not use abbreviations.	Mexico City, Mexico

TITLE:

Design and Analysis for Fall Detection System Simplification

AUTHORS AND AFFILIATIONS:

Lourdes Martínez-Villaseñor, Hiram Ponce

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

Corresponding Author:

Lourdes Martínez-Villaseñor
lmartine@up.edu.mx

Email Addresses of co-authors:

Hiram Ponce (hponce@up.edu.mx)

KEYWORDS:

fall detection, human activity recognition, machine learning, healthcare, ambient assisted living, sensors, multimodal approaches, classification

SUMMARY:

We present a methodology based on multimodal sensors to configure a simple, comfortable and fast fall detection and human activity recognition system. The goal is to build a system for accurate fall detection that can be easily implemented and adopted.

ABSTRACT:

This paper presents a methodology based on multimodal sensors to configure a simple, comfortable and fast fall detection and human activity recognition system that can be easily implemented and adopted. The methodology is based on the configuration of specific types of sensors, machine-learning methods and procedures. The protocol is divided into four phases: (1) database creation (2) data analysis (3) system simplification and (4) evaluation. Using this methodology, we created a multimodal database for fall detection and human activity recognition, namely UP-Fall Detection. It comprises data samples from 17 subjects that perform 5 types of falls and 6 different simple activities, during 3 trials. All information was gathered using 5 wearable sensors (tri-axis accelerometer, gyroscope and light intensity), 1 electroencephalograph helmet, 6 infrared sensors as ambient sensors, and 2 cameras in lateral and front viewpoints. The proposed novel methodology adds some important stages to perform a deep analysis of the following design issues in order to simplify a fall detection system: a) select which sensors or combination of sensors are to be used in a simple fall detection system, b) determine the best placement of the sources of information, and c) select the most suitable machine learning classification method for fall and human activity detection and recognition. Even though some multimodal approaches reported in literature only focus on one or two of the above-mentioned issues, our methodology allows simultaneously solving these three design problems related to a human fall and activity detection and recognition system.

INTRODUCTION:

Since the world phenomenon of population aging[1], fall prevalence has increased and is actually considered a major health problem[2]. When a fall occurs, people require immediate attention in order to reduce negative consequences. Fall detection systems can reduce the amount of time in which a person receives medical attention sending an alert when a fall occurs.

There are various categorizations of fall detection systems[3]. Early works[4] classify fall detection systems by their method of detection, roughly analytical methods and machine learning methods. More recently, other authors[3,5,6] have considered data acquisition sensors as the main feature to classify fall detectors. Igual et al.[3] divides fall detection systems into context-aware systems, that include vision and ambient-sensor based approaches, and wearable device systems. Mubashir et al.[5] classifies fall detectors into three groups based on the devices used for data acquisition: wearable devices, ambience sensors, and vision-based devices. Perry et al.[6] considers methods for measuring acceleration, methods for measuring acceleration combined with other methods, and methods not measuring acceleration. From these surveys, we can determine that sensors and methods are the main elements to classify the general research strategy.

Each of the sensors has weaknesses and strengths discussed in Xu et al.[7]. Vision-based approaches mainly use normal cameras, depth sensor cameras, and/or motion capture systems. Normal web cameras are low cost and easy to use, but they are sensitive to environmental conditions (light variation, occlusion, etc.), can only be used in a reduced space, and have privacy issues. Depth cameras, such as the Kinect, provide full-body 3D motion[7] and are less affected by lighting conditions than normal cameras. However, approaches based on the Kinect are not as robust and reliable. Motion capture systems are more expensive and difficult to use.

Approaches based on accelerometer devices and smart phones/watches with built-in accelerometers are very commonly used for fall detection. The main drawback of these devices is that they have to be worn for long periods. Discomfort, obtrusiveness, body placement and orientation are design issues to be solved in these approaches. Although smartphones and smart watches are less obtrusive devices that sensors, older people often forget or do not always wear these devices. Nevertheless, the advantage of these sensors and devices is that they can be used in many rooms and/or outdoors.

Some systems use sensors placed around the environment to recognize falls/activities, so people do not have to wear the sensors. However, these sensors are also limited to the places where they are deployed[8] and are sometimes difficult to install. Recently, multimodal fall detection systems include different combinations of vision, wearable and ambient sensors in order to gain more precision and robustness. They can also overcome some of the single sensor limitations.

The methodology used for fall detection is closely related with human activity recognition chain (ARC) presented by Bulling et al.[9], which consists of stages for data acquisition, signal preprocessing and segmentation, feature extraction and selection, training and classification. Design issues must be solved for each of these stages. Different methods are used in each stage.

We present a methodology based on multimodal sensors to configure a simple, comfortable and fast human fall and human activity detection/recognition system. The goal is to build a system for accurate fall detection that can be easily implemented and adopted. The proposed novel methodology is based on ARC, but it adds some important phases to perform a deep analysis of the following issues in order to simplify the system: (a) select which sensors or combination of sensors are to be used in a simple fall detection system; (b) determine the best placement of the information sources ; and (c) select the most suitable machine learning classification method for fall detection and human activity recognition to create a simple system.

There are some related works in literature that address one or two of the above-mentioned design issues, but to our knowledge, there is no work that focuses on a methodology to overcome all of these problems.

Related works use multimodal approaches for fall detection and human activity recognition[10-12] in order to gain robustness and increase precision. Kwolek et al.[10] proposed the design and implementation of a fall detection system based on accelerometric data and depth maps. They designed an interesting methodology in which a three-axis accelerometer is implemented to detect a potential fall as well as the person’s motion. If the acceleration measure exceeds a threshold, the algorithm extracts a person differencing the depth map from the online updated depth reference map. An analysis of depth and accelerometer combinations was made using a support vector machine classifier.

Ofli et al.[11] presented a Multimodal Human Action Database (MHAD) in order to provide a testbed for new human activity recognition systems. The dataset is important since the actions were gathered simultaneously using 1 optical motion capture system, 4 multi-view cameras, 1 Kinect system, 4 microphones, and 6 wireless accelerometers. The authors presented results for each modality: the Kinect, the mocap, the accelerometer, and the audio.

Dovgan et al.[12] proposed a prototype for detecting anomalous behavior, including falls, in the elderly. They designed tests for three sensor systems in order to find the most appropriate equipment for fall and unusual-behavior detection. The first experiment consists of data from a smart sensor system with 12 tags attached to the hips, knees, ankles, wrists, elbows and shoulders. They also created a test dataset using one Ubisense sensor system with four tags attached to the waist, chest and both ankles, and one Xsens accelerometer. In a third experiment, four subjects only use the Ubisense system while performing 4 types of falls, 4 health problems as anomalous behavior and different activity of daily living (ADL).

Other works in literature[13-15] address the problem of finding the best placement of sensors or devices for fall detection comparing the performance of various combinations of sensors with several classifiers. Santoyo et al.[13] presented a systematic assessment evaluating the importance of the location of 5 sensors for fall detection. They compared the performance of these sensor combinations using k-nearest neighbors (KNN), support vector machines (SVM), naïve Bayes (NB) and decision tree (DT) classifiers. They conclude that the location of the sensor on the subject has an important influence on the fall detector performance independent of the classifier used.

A comparison of wearable sensor placements on the body for fall detection was presented by Özdemir[14]. In order to determine sensor placement, the author analyzed 31 sensor combinations of the following positions: head, waist, chest, right wrist, right ankle and right thigh. Fourteen volunteers performed 20 simulated falls and 16 ADL. He found that the best performance was obtained when a single sensor is positioned on the waist from these exhaustive combination experiments. Another comparison was presented by Ntanasis[15] using Özdemir's dataset. The authors compared single positions on the head, chest, waist, wrist, ankle and thigh using the following classifiers: J48, KNN, RF, random committee (RC) and SVM.

Benchmarks of the performance of different computational methods for fall detection can also be found in literature[16-18]. Bagala et al.[16] presented a systematic comparison to benchmark the performance of thirteen fall detection methods tested on real falls. They only considered algorithms based on accelerometer measurements placed on the waist or trunk. Bourke et al.[17] evaluated the performance of five analytical algorithms for fall detection using a dataset of ADLs and falls based on accelerometer readings. Kerdegari[18] made also a comparison of the performance of different classification models for a set of recorded acceleration data. The algorithms used for fall detection were zeroR, oneR, NB, DT, multilayer perceptron and SVM.

A methodology for fall detection was proposed by Alazrai et al.[18] using motion pose geometric descriptor to construct an accumulated histogram-based representation of human activity. They evaluated the framework using a dataset collected with Kinect sensors.

In summary, we found multimodal fall detection related works[10-12] that compare the performance of different combinations of modalities. Some authors address the problem of finding the best placement of sensors[13-15], or combinations of sensors[13] with several classifiers[13,15,16] with multiple sensors of the same modality and accelerometers. No work was found in literature that address placement, multimodal combinations and classifier benchmark at the same time.

PROTOCOL:

All methods described here have been approved by the Research Committee of the School of Engineering of Universidad Panamericana.

NOTE: This methodology is based on the configuration of the specific types of sensors, machine-learning methods and procedures in order to configure a simple, fast and multimodal fall detection and human activity recognition system. Due to this, the following protocol is divided in phases: (1) database creation (2) data analysis (3) system simplification and (4) evaluation.

1. Database creation

1.1. Set up the data acquisition system. This will collect all data from subjects and store the information in a retrieval database.

1.1.1. Select the types of wearable sensors, ambient sensors and vision-based devices required as sources of information. Assign an ID for each source of information, the number of channels per source, the technical specifications and the sampling rate of each of them.

1.1.2. Connect all sources of information (i.e., wearables and ambient sensors, and vision-based devices) to a central computer or a distributed computer system:

1.1.2.1. Verify that wired-based devices are connected properly to one client computer. Verify that wireless-based devices are fully charged. Consider that low battery might impact wireless connections or sensor values. Moreover, intermittent or lost connections will increase loss of data.

1.1.3. Set up each of the devices to retrieve data.

1.1.4. Set up the data acquisition system for storing data on the cloud. Due to the large amount of data to be stored, cloud computing is considered in this protocol.

1.1.5. Validate that the data acquisition system fulfills data synchronization and data consistency[20] properties. This maintains the integrity of data storage from all the sources of information. It might require new approaches in data synchronization. For example, see Peñafort-Asturiano et al.[20].

1.1.5.1. Start collecting some data with the sources of information and store data in a preferred system. Include timestamps in all data.

1.1.5.2. Query the database and determine if all sources of information are collected at the same sample rates. If done properly, go to Step 1.1.6. Otherwise, perform up-sampling or down-sampling using criteria reported in Peñafort-Asturiano, et al.[20].

1.1.6. Set up the environment (or laboratory) by considering the conditions required and the restrictions imposed by the goal of the system. Set conditions for impact force attenuation in the simulated falls as compliant flooring systems suggested in Lachance, et al.[23] to ensure participants safety.

1.1.6.1. Use a mattress or any other compliant flooring system and place it at the center of the environment (or laboratory).

1.1.6.2. Keep all objects away from the mattress to give at least one meter of safe space all around. If required, prepare personal protective equipment for participants (e.g., gloves, cap, goggles, knee support, etc.).

NOTE: The protocol can be paused here.

1.2. Determine the human activities and falls that the system will detect after configuration. It is important to have in mind the purpose of the fall detection and human activity recognition system, as well as the target population.

1.2.1. Define the goal of the fall detection and human activity recognition system. Write it down in a planning sheet. For this case study, the goal is to classify the types of human falls and activities performed in an indoor daily basis of elderly people.

1.2.2. Define the target population of the experiment in accordance with the goal of the system. Write it down in the planning sheet. In the study, consider elderly people as the target population.

1.2.3. Determine the type of daily activities. Include some non-fall activities that look like falls in order to improve real fall detection. Assign an ID for all of them and describe them as detailed as possible. Set the time period for each activity to be executed. Write all this information down in the planning sheet.

1.2.4. Determine the type of human falls. Assign an ID for all of them and describe them as detailed as possible. Set the time period for each fall to be executed . Consider if the falls will be self-generated by the subjects or generated by others (e.g., pushing the subject). Write all this information down in the planning sheet.

1.2.5. In the planning sheet, write down the sequences of activities and falls that a subject will perform. Specify the period of time, the number of trials per activity/fall, the description to perform the activity/fall, and the activity/fall IDs.

NOTE: The protocol can be paused here.

1.3. Select the relevant subjects to the study that will execute the sequences of activities and falls. Falls are rare events to catch in real life and usually occur to old persons. Nevertheless, for safety reasons, do not include elderly and impaired people in fall simulation under medical advice. Stunts have been used to avoid injuries[22].

1.3.1. Determine the gender, age range, weight and height of the subjects. Define any impairment conditions required. Also, define the minimum number of subjects required for the experiment.

1.3.2. Randomly select the set of subjects required, following the conditions stated in the previous step. Use a call for volunteers to recruit them. Fulfill all ethical guidelines applicable from the institution and country, as well as any international regulation when experimenting with humans.

NOTE: The protocol can be paused here.

1.4. Retrieve and store data from subjects. This information will be useful for further experimental analysis. Complete the following steps under supervision by a clinical expert or a responsible researcher.

1.4.1. Start collecting data with the data acquisition system configured in Step 1.1.

1.4.2. Ask each of the subjects to perform the sequences of activities and falls declared in Step 1.2. Clearly save the timestamps of the start and end of each activity/fall. Verify that data from all sources of information is saved on the cloud.

1.4.3. If the activities were not properly done or there were issues with devices (e.g., lost connection, low battery, intermittent connection), discard the samples and repeat Step 1.4.1 until no device issues are found. Repeat Step 1.4.2 for each trial, per subject, declared in the sequence of Step 1.2.

NOTE: The protocol can be paused here.

1.5. Pre-process all data acquired. Apply up-sampling and down-sampling for each of the sources of information. See details about pre-processing data for fall detection and human activity recognition in Martínez-Villaseñor et al.[21].

NOTE: The protocol can be paused here.

2. Data Analysis

2.1. Select the mode of data treatment. Select **Raw Data** if the data stored in the database will be used outright (i.e., using deep learning for automatic feature extraction) and go to Step 2.2. Select **Feature Data** if feature extraction will be used for further analysis and go to Step 2.3.

2.2. For **Raw Data**, no extra steps are required so go to Step 2.5.

2.3. For **Feature Data**, extract features from the raw data.

2.3.1. Segment raw data in time windows. Determine and fix the time window length (e.g., frames of one-second size). In addition, determine if these time windows will be overlapping or not. A good practice is to choose 50% overlapping.

2.3.2. Extract features from each segment of data. Determine the set of temporal and frequential features to be extracted from the segments. See Martínez-Villaseñor et al.[21] for common feature extraction.

2.3.3. Save the feature extraction data set on the cloud, in an independent database.

2.3.4. If different time windows will be selected, repeat Steps 2.3.1 to 2.3.3, and save each feature data set in independent databases.

NOTE: The protocol can be paused here.

2.4. Select the most important features extracted and reduce the feature data set. Apply some commonly used feature selection methods (e.g., univariate selection, principal components analysis, recursive feature elimination, feature importance, correlation matrix, etc.).

2.4.1. Select a feature selection method. Here, we used feature importance.

2.4.2. Use each feature to train a given model (we employed RF) and measure the accuracy (see **Equation 1**).

2.4.3. Rank the features by sorting in order of the accuracy.

2.4.4. Select the most important features. Here, we used the best ranked first ten features.

NOTE: The protocol can be paused here.

2.5. Select a machine learning classification method and train a model. There are well-known machine learning methods[16-18,21], such as: support vector machines (SVM), random forest (RF), multilayer perceptron (MLP) and k-nearest neighbors (KNN), among many others.

2.5.1. Optionally, if a deep learning approach is selected, then consider[21]: convolutional neural networks (CNN), long short-term memory neural networks (LSTM), among others.

2.5.2. Select a set of machine learning methods. Here, we used the following methods: SVM, RF, MLP and KNN.

2.5.3. Fix the parameters of each of the machine learning methods, as suggested in literature[21].

2.5.4. Create a combined feature data set (or raw data set) using the independent feature data sets (or raw data sets), to combine types of sources of information. For example, if a combination of one wearable sensor and one camera is required, then combine the feature data sets from each of these sources.

2.5.5. Split the feature data set (or raw data set) in training and testing sets. A good choice is to randomly divide 70% for training and 30% for testing.

2.5.6. Run a k-fold cross-validation[21] using the feature data set (or raw data set), for each machine learning method. Use a common metric of evaluation, like accuracy (see **Equation 1**) to select the best model trained per method. Leave-one subject-out (LOSO) experiments[3] are also recommend.

2.5.6.1. Open the training feature data set (or raw data set) in the preferred programming language software. Python is recommended. For this step, use pandas library to read a CSV file as follows:
training_set = pandas.csv(<filename.csv>).

2.5.6.2. Split the feature data set (or raw data set) in pairs of inputs-outputs. For example, use Python to declare the x-values (inputs) and the y-values (outputs):
training_set_X = training_set.drop('tag',axis=1), training_set_Y = training_set.tag
where tag represents the column of the feature data set that includes the target values.

2.5.6.3. Select one machine learning method and set the parameters. For example, use SVM in Python with the library sklearn like the following command:
classifier = sklearn.SVC(kernel = 'poly')
in which the kernel function is selected as polynomial.

2.5.6.4. Train the machine learning model. For example, use the above classifier in Python to train the SVM model:
classifier.fit(training_set_X,training_set_Y).

2.5.6.5. Compute the estimates values of the model using the testing feature data set (or the raw data set). For example, use the estimate function in Python as follows: **estimates = classifier.predict(testing_set_X)** where testing_set_X represents the x-values of the testing set.

2.5.6.6. Repeat Steps 2.5.6.1 to 2.5.6.5, the number of times k specified in the k-fold cross validation (or the number of times required for the LOSO approach).

2.5.6.7. Repeat Steps 2.5.6.1 to 2.5.6.6 for each machine learning model selected.

NOTE: The protocol can be paused here.

2.5.7. Compare the machine learning methods by testing the selected models with the testing data set. Other metrics of evaluation can be used: accuracy (**Equation 1**), precision (**Equation 2**), sensitivity (**Equation 3**), specificity (**Equation 4**) or F1-score (**Equation 5**), where *TP* are the true positives, *TN* are the true negatives, *FP* are the false positives and *FN* are the false negatives.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$precision = \frac{TP}{TP+FP} \quad (2)$$

$$sensitivity = \frac{TP}{TP+FN} \quad (3)$$

$$specificity = \frac{TN}{TN+FP} \tag{4}$$

$$F_1 score = 2 \cdot \frac{precision \times sensitivity}{precision + sensitivity} \tag{5}$$

2.5.8. Use other beneficial performance metrics such as the confusion matrix[9] to evaluate the classification task of the machine learning models, or a decision-independent precision-recall[9] (PR) or receiver operating characteristic[9] (ROC) curves. In this methodology, recall and sensitivity are considered equivalent.

2.5.9. Use qualitative features of the machine learning models to compare among them, such as: ease of machine learning interpretation; real-time performance; limited resources of time, memory and processing computing; and ease of machine learning deployment in edge devices or embedded systems.

2.5.10. Select the best machine learning model using the information from: The quality metrics (**Equations 1–5**), the performance metrics and the qualitative features of machine learning feasibility of Steps 2.5.6, 2.5.7 and 2.5.8.

NOTE: The protocol can be paused here.

3. System simplification

3.1. Select the suitable placements of sources of information. Sometimes, it is necessary to determine the best placement of sources of information (e.g., which location of a wearable sensor is better).

3.1.1. Determine the subset of sources of information that will be analyzed. For example, if there are five wearable sensors in the body and just one has to be selected as the best sensor placed, each of these sensors will be part of the subset.

3.1.2. For each source of information in this subset, create a separate data set and store it separately. Keep in mind that this data set could be either the previous feature data set or the raw data set.

NOTE: The protocol can be paused here.

3.2. Select a machine learning classification method and train a model for one source of information placement. Complete Steps from 2.5.1 to 2.5.6 using each of the data sets created in Step 3.1.2. Detect the most suitable source of information placement by ranking. For this case study, we use the following methods: SVM, RF, MLP and KNN.

Note: The protocol can be paused here.

3.3. Select the suitable placements in a multimodal approach if a combination of two or more sources of information are required for the system (e.g., combination of one wearable sensor and one camera). In this case study, use waist-wearable sensor and camera 1 (lateral view) as the modalities.

3.3.1. Select the best source of information of each modality in the system and create a combined feature data set (or raw data set) using the independent data sets of these sources of information.

3.3.2. Select a machine learning classification method and train a model for these combined sources of information. Complete Steps 2.5.1 to 2.5.6 using the combined feature data set (or raw data set). In this study, use the following methods: SVM, RF, MLP and KNN.

NOTE: The protocol can be paused here.

4. Evaluation

4.1. Prepare a new data set with users in more realistic conditions. Use only the sources of information selected in the previous step. Preferable, implement the system in the target group (e.g., elderly people). Collect data in longer periods of time.

4.1.1. Optionally if the target group is used only, create a selection group protocol including the terms of exclusion (e.g., any physical or psychological impairment) and stop criteria prevention (e.g., detect any physical injury during the trials; suffering nausea, dizziness and/or vomiting; fainting). Consider also ethical concerns and data privacy issues.

4.2. Evaluate the performance of the fall detection and human activity recognition system developed so far. Use Equations 1–5 to determine the accuracy and predictive power of the system, or any other performance metrics.

4.3. Discuss about the findings on the experimental results.

REPRESENTATIVE RESULTS:

Creation of a Database

We created a multimodal dataset for fall detection and human activity recognition, namely UP-Fall Detection[21]. The data were collected over a four-week period at the School of Engineering at Universidad Panamericana (Mexico City, Mexico). The test scenario was selected considering the following requirements: (a) a space in which subjects could comfortably and securely perform falls and activities, and (b) an indoor environment with natural and artificial light that is well suited for multimodal sensors settings.

There are data samples from 17 subjects that performed 5 types of falls and 6 different simple activities, during 3 trials. All information was gathered using an in-house data acquisition system with 5 wearable sensors (tri-axis accelerometer, gyroscope and light intensity), 1 electroencephalograph helmet, 6 infrared sensors as ambient sensors, and 2 cameras at lateral and front viewpoints. **Figure 1** shows the layout of the sensor placement in the environment and on the body. The sampling rate of the whole dataset is 18 Hz. The database contains two data sets: the consolidated raw data set (812 GB), and a feature data set (171 GB). All the databases ware stored in the cloud for public access: https://sites.google.com/up.edu.mx/har-up/. More details on data acquisition, pre-processing, consolidating and storing of this database as well as details on synchronization and data consistency can be found in Martínez-Villaseñor et al.[21].

For this database, all subjects were healthy young volunteers (9 males and 8 females) without any impairment, ranging on 18 to 24 years old, with mean height of 1.66 m and mean weight of 66.8 kg. During data collection, the technical responsible researcher was supervising that all the activities were performed by the subjects correctly. Subjects performed five types of falls, each one for 10 seconds, as falling: forward using hands (1), forward using knees (2), backwards (3), sitting in an empty chair (4) and sideward (5). They also conducted six daily activities for 60 s each except for jumping (30 s): walking (6), standing (7), picking up an object (8), sitting (9), jumping (10) and laying (11). Although simulated falls cannot reproduce all types of real-life falls, it is important at least to include representative types of falls enabling the creation of better fall detection models. It is also relevant to use ADLs and, in particular, activities that can usually be mistaken with falls such as picking up an object. The types of fall and ADLs were selected after a review of related fall detection systems[21]. As an example, **Figure 2** shows a sequence of images of one trial when a subject falls sideward.

We extracted 12 temporal (mean, standard deviation, maximal amplitude, minimal amplitude, root mean square, median, zero-crossing number, skewness, kurtosis, first quartile, third quartile and autocorrelation) and 6 frequential (mean, median, entropy, energy, principal frequency and spectral centroid) features[21] from each channel of the wearable and ambient sensors comprising 756 features in total. We also computed 400 visual features[21] for each camera about the relative motion of pixels between two adjacent images in the videos.

Data Analysis between Unimodal and Multimodal Approaches

From the UP-Fall Detection database, we analyzed the data for comparison purposes between unimodal and multimodal approaches. In that sense, we compared seven different combinations of sources of information: infrared sensors only (IR); wearable sensors only (IMU); wearable sensors and helmet (IMU+EEG); infrared and wearable sensors and helmet (IR+IMU+EEG); cameras only (CAM); infrared sensors and cameras (IR+CAM); and wearable sensors, helmet and cameras (IMU+EEG+CAM). In addition, we compared three different time window sizes with 50% overlapping: one second, two seconds and three seconds. At each segment, we selected the most useful features applying feature selection and ranking. Using this strategy, we employed only 10 features per modality, except in the IR modality using 40 features. Moreover, the comparison was done over four well-known machine learning classifiers: RF, SVM, MLP and KNN. We employed 10-fold cross-validation, with datasets of 70% train and 30% test, to train the machine learning models. **Table 1** shows the results of this benchmark, reporting the best performance obtained for each modality depending on the machine learning model and the best window length configuration. The evaluation metrics report accuracy, precision, sensitivity, specificity and F1-score. **Figure 3** shows these results in a graphical representation, in terms of F1-score.

From **Table 1**, multimodal approaches (infrared and wearable sensors and helmet, IR+IMU+EEG; and wearable sensors and helmet and cameras, IMU+EEG+CAM) obtained the best F1-score values, in comparison with unimodal approaches (infrared only, IR; and cameras only, CAM). We also noticed that wearable sensors only (IMU) obtained similar performance than a multimodal approach. In this case, we opted for a multimodal approach because different sources of information can handle the limitations from others. For example, obtrusiveness in cameras can be handled by wearable sensors, and not using all wearable sensors can be complemented with cameras or ambient sensors.

In terms of the benchmark of the data-driven models, experiments in **Table 1** shown that RF presents the best results in almost all the experiment; while MLP and SVM were not very consistent in performance (e.g., standard deviation in these techniques shows more variability than in RF). About the window sizes, these did not represent any significant improvement among them. It is important to notice that these experiments were done for fall and human activity classification.

Sensor Placement and Best Multimodal Combination

On the other hand, we aimed to determine the best combination of multimodal devices for fall detection. For this analysis, we restricted the sources of information to the five wearable sensors and the two cameras. These devices are the most comfortable ones for the approach. In addition, we considered two classes: fall (any type of fall) or no-fall (any other activity). All the machine learning models, and window sizes remain the same as in the previous analysis.

For each wearable sensor, we built an independent classifier model for each window length. We trained the model using 10-fold cross-validation with 70% training and 30% testing data sets. **Table 2** summarizes the results for the ranking of the wearable sensors per performance classifier, based on the F1-score. These results were sorted in descending order. As seen in **Table 2**, the best performance is obtained when using a single sensor at the waist, neck or tight right pocket (shadowed region). In addition, ankle and left wrist wearable sensors performed the worst. **Table 3** shows the window length preference per wearable sensor in order to get the best performance in each classifier. From the results, waist, neck and tight right pocket sensors with RF classifier and 3 s window size with 50% overlapping are the most suitable wearable sensors for fall detection.

We conducted a similar analysis for each camera in the system. We built an independent classifier model for each window size. For training, we did 10-fold cross-validation with 70% training and 30% testing data sets. **Table 4** shows the ranking of the best camera viewpoint per classifier, based on the F1-score. As observed, the lateral view (camera 1) performed the best fall detection. In addition, RF outperformed in comparison with the other classifiers. Also, **Table 5** shows the window length preference per camera viewpoint. From the results, we found that the best location of a camera is in lateral viewpoint using RF in 3 s window size and 50% overlapping.

Lastly, we chose two possible placements of wearable sensors (i.e., waist and tight right pocket) to be combined with the camera of lateral viewpoint. After the same training procedure, we obtained the results from **Table 6**. As shown, the RF model classifier got the best performance in accuracy and F1-score in both multimodalities. Also, the combination between waist and camera 1 ranked in the first position obtaining 98.72% in accuracy and 95.77% in F1-score.

FIGURE AND TABLE LEGENDS:

Figure 1: **Layout of the wearable (left) and ambient (right) sensors in the UP-Fall Detection database.** The wearable sensors are placed in the forehead, the left wrist, the neck, the waist, the right pocket of the pants and the left ankle. The ambient sensors are six paired infrared sensors to detect the presence of subjects and two cameras. Cameras are located at the lateral view and at the front view, both with respect to the human fall.

Figure 2: **Example of a video recording extracted from the UP-Fall Detection database**. At the top, there is a sequence of images of a subject falling sideward. At the bottom, there is a sequence of images representing the vision features extracted. These features are the relative motion of pixels between two adjacent images. White pixels represent faster motion, while black pixels represent slower (or near zero) motion. This sequence is sorted from left to right, chronologically.

Figure 3: **Comparative results reporting the best F1-score of each modality with respect to the machine learning model and the best window length.** Bars represent the mean values of F1-score. Text in data points represent mean and standard deviation in parenthesis.

Table 1: **Comparative results reporting the best performance of each modality with respect to the machine learning model and the best window length (in parenthesis).** All values in performance represent the mean and the standard deviation.

Table 2: **Ranking of the best wearable sensor per classifier, sorted by the F1-score (in parenthesis).** The regions in shadow represent the top three classifiers for fall detection.

Table 3: **Preferred time window length in the wearable sensors per classifier.**

Table 4: **Ranking of the best camera viewpoint per classifier, sorted by the F1-score (in parenthesis).** The regions in shadow represent the top classifier for fall detection.

Table 5: **Preferred time window length in the camera viewpoints per classifier.**

Table 6: **Comparative results of the combined wearable sensor and camera viewpoint using 3-second window length.** All values represent the mean and standard deviation.

DISCUSSION:

It is common to encounter challenges due to synchronization, organization and data inconsistency problems[20] when a dataset is created.

Synchronization

In the acquisition of data, synchronization problems arise given that multiple sensors commonly work at different sampling rates. Sensors with higher frequencies collect more data than those with lower frequencies. Thus, data from different sources will not be paired correctly. Even if sensors run at the same sampling rates, it is possible that data will not be aligned. In this regard, the following recommendations might help to handle these synchronization problems[20]: (i) register timestamp, subject, activity and trial in each data sample obtained from the sensors; (ii) the most consistent and less frequent source of information has to be used as reference signal for synchronization; and (iii) use automatic or semi-automatic procedures to synchronize video recordings that manual inspection would be impractical.

Data pre-processing

Data pre-processing must also be done, and critical decisions influence this process: (a) determine the methods for data storage and data representation of multiple and heterogeneous sources (b) decide the ways to store data in the local host or on the cloud (c) select the organization of data, including the file names and folders (d) handle missing values of data as well as redundancies found in the sensors, among others. In addition, for the data cloud, local buffering is recommended when possible to mitigate loss of data at the uploading time.

Data inconsistency

Data inconsistency is common between trials finding variations in data sample sizes. These issues are related to data acquisition in wearable sensors. Brief interruptions of data acquisition and data collision from multiple sensors leads to data inconsistencies. In these cases, inconsistency detection algorithms are important to handle online failure in sensors. It is important to highlight that wireless-based devices should be monitored frequently throughout the experiment. Low battery might impact connectivity and result in loss of data.

Ethical

Consent to participate and ethical approval are mandatory in every type of experimentation where people are involved.

Regarding the limitations of this methodology, it is important to notice that it is designed for approaches that consider different modalities for data collection. The systems can include wearable, ambient and/or vision sensors. It is suggested to consider the power consumption of devices and the lifetime of batteries in wireless-based sensors, due to the issues such as loss of data collection, diminishing connectivity and power consumption in the whole system. Moreover, this methodology is intended for systems that use machine learning methods. An analysis of the selection of these machine learning models should be done beforehand. Some of these models could be accurate, but highly time and energy consuming. A trade-off between accurate estimation and limited resource availability for computing in machine learning models must be taken into consideration. It is also important to observe that, in data collection of the system, the activities were conducted in the same order; also, trials were performed in the same sequence. For safety reasons, a protective mattress was used for subjects to fall onto. In addition, the falls were self-initiated. This is an important difference between simulated and real falls, which generally occur towards hard materials. In that sense, this dataset recorded falls with an intuitive reaction trying not to fall. Moreover, there are some differences between real falls in elderly or impaired people and the simulation falls; and these must be taken into account when designing a new fall detection system. This study was focused on young people without any impairments, but it is remarkable to say that the selection of subjects should be aligned to the goal of the system and the target population who will use it.

From the related works described above[10-18], we can observe that there are authors that use multimodal approaches focusing in obtaining robust fall detectors or focus on placement or performance of the classifier. Hence, they only address one or two of the design issues for fall detection. Our methodology allows solving simultaneously three of the main design problems of a fall detection system.

For future work, we suggest designing and implementing a simple multimodal fall detection system based on the findings obtained following this methodology. For real-world adoption, transfer learning, hierarchical classification and deep learning approaches should be used for developing more robust systems. Our implementation did not consider qualitative metrics of the machine learning models, but real-time and limited computing resources have to be taken into account for further development of human fall and activity detection/recognition systems. Lastly, in order to improve our dataset, tripping or almost falling activities and real-time monitoring of volunteers during their daily life can be considered.

ACKNOWLEDGMENTS:

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.

DISCLOSURES:

The authors have nothing to disclose.

REFERENCES:

1. United Nations, World Population Prospects: The 2017 Revision, Key Findings and Advance Tables. ESA/P/WP/248. *Department of Economic and Social Affairs, Population Division* (2017).

2. World Health Organization. Ageing, and Life Course Unit. *WHO Global Report on Falls Prevention in Older Age* (2008).

3. Igual, R., Medrano, C., Plaza, I. Challenges, Issues and Trends in Fall Detection Systems. *Biomedical Engineering Online*. **12** (1), 66 (2013).

4. Noury, N. et al. Fall Detection-Principles and Methods. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 1663–1666 (2007).

5. Mubashir, M., Shao, L., Seed, L. A Survey on Fall Detection: Principles and Approaches. *Neurocomputing*. **100**, 144–152 (2002).

6. Perry, J.T. et al. Survey and Evaluation of Real-Time Fall Detection Approaches. In *Proceedings of the 6th International Symposium High-Capacity Optical Networks and Enabling Technologies*. 158–164 (2009).

7. Xu, T., Zhou, Y., Zhu, J. New Advances and Challenges of Fall Detection Systems: A Survey. *Applied Sciences*. **8** (3), 418 (2018).

8. Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J. Robust Video Surveillance for Fall Detection Based on Human Shape Deformation. *IEEE Transactions on Circuit Systems for Video Technologies*. **21**, 611–622 (2011).

9. Bulling, A., Blanke, U., Schiele, B. A Tutorial on Human Activity Recognition Using Body-Worn Inertial Sensors. *ACM Computing Surveys*. **46** (3), 33 (2014).

10. Kwolek, B., Kepski, M. Human Fall Detection on Embedded Platform Using Depth Maps and Wireless Accelerometer. *Computational Methods and Programs in Biomedicine*. **117**, 489–501 (2014).

11. Ofli, F., Chaudhry, R., Kurillo, G., Vidal, R., Bajcsy, R. Berkeley MHAD: A Comprehensive Multimodal Human Action Database. In *Proceedings of the 2013 IEEE Workshop on Applications of Computer Vision*. 53–60 (2013).

12. Dovgan, E. et al. Intelligent Elderly-Care Prototype for Fall and Disease Detection. *Slovenian Medical Journal*. **80**, 824–831 (2011).

13. Santoyo-Ramón, J., Casilari, E., Cano-García, J. Analysis of a Smartphone-Based Architecture With Multiple Mobility Sensors for Fall Detection With Supervised Learning. *Sensors*. **18** (4), 1155 (2018).

14. Özdemir, A. An Analysis on Sensor Locations of the Human Body for Wearable Fall Detection Devices: Principles and Practice. *Sensors*. **16** (8), 1161 (2016).

15. Ntanasis, P., Pippa, E., Özdemir, A.T., Barshan, B., Megalooikonomou, V. Investigation of Sensor Placement for Accurate Fall Detection. In *International Conference on Wireless Mobile Communication and Healthcare*. 225–232 (2016).

16. Bagala, F., et al. Evaluation of Accelerometer-Based Fall Detection Algorithms on Real-World Falls. *PLoS One*. **7**, e37062 (2012).

17. Bourke, A.K. et al. Assessment of Waist-Worn Tri-Axial Accelerometer Based Fall-detection Algorithms Using Continuous Unsupervised Activities. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 2782–2785 (2010).

18. Kerdegari, H., Samsudin, K., Ramli, A. R., Mokaram, S. Evaluation of Fall Detection Classification Approaches. In *4th International Conference on Intelligent and Advanced Systems*. 131–136 (2012).

19. Alazrai, R., Mowafi, Y., Hamad, E. A Fall Prediction Methodology for Elderly Based on a Depth Camera. In *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 4990–4993 (2015).

20. Peñafort-Asturiano, C.J., Santiago, N., Núñez-Martínez, J.P., Ponce, H., Martínez-Villaseñor, L. Challenges in Data Acquisition Systems: Lessons Learned from Fall Detection to Nanosensors. In *2018 Nanotechnology for Instrumentation and Measurement*. 1–8 (2018).

21. Martínez-Villaseñor, L. et al. UP-Fall Detection Dataset: A Multimodal Approach. *Sensors*. **19** (9), 1988 (2019).

22. Rantz, M. et al. Falls, Technology, and Stunt Actors: New approaches to Fall Detection and Fall Risk Assessment. *Journal of Nursing Care Quality*. **23** (3), 195–201 (2008).

23. Lachance, C., Jurkowski, M., Dymarz, A., Mackey, D. Compliant Flooring to Prevent Fall-Related Injuries: A Scoping Review Protocol. *BMJ Open*. **6** (8), e011757 (2016).

Figure 1

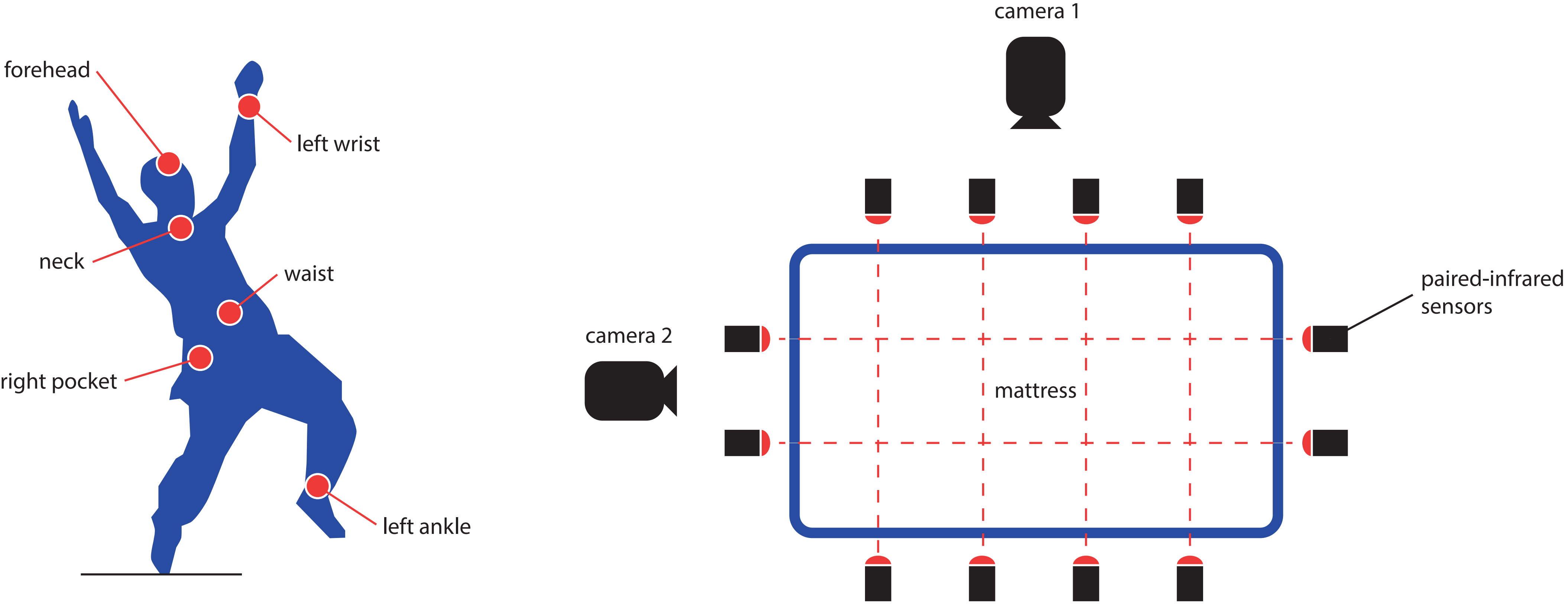
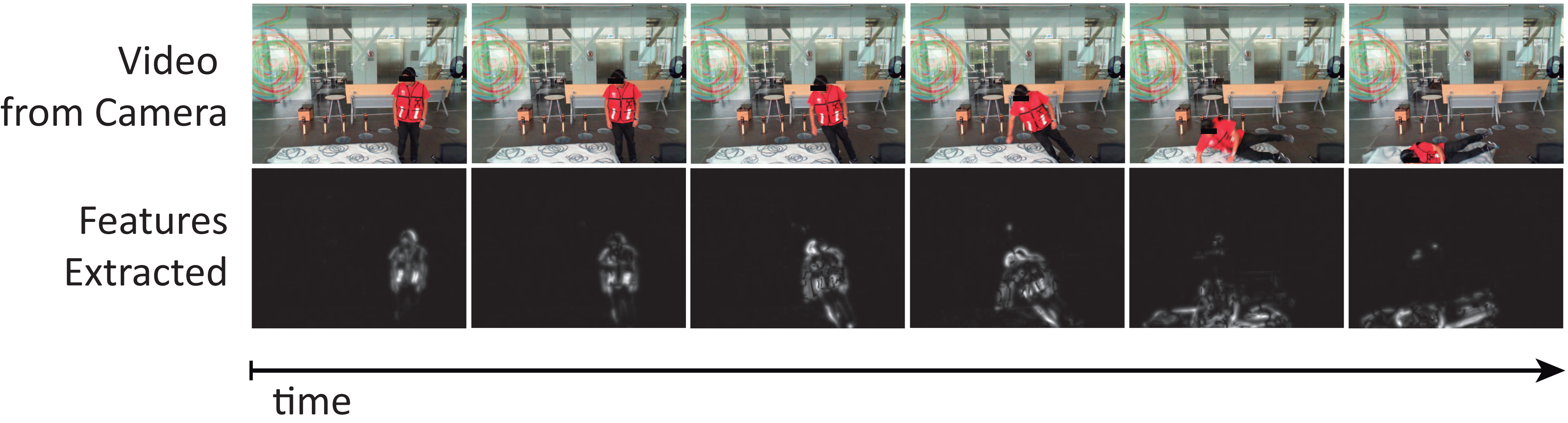

forehead
left wrist
neck
waist
right pocket
left ankle
camera 1
camera 2
mattress
paired-infrared
sensors

Figure 2



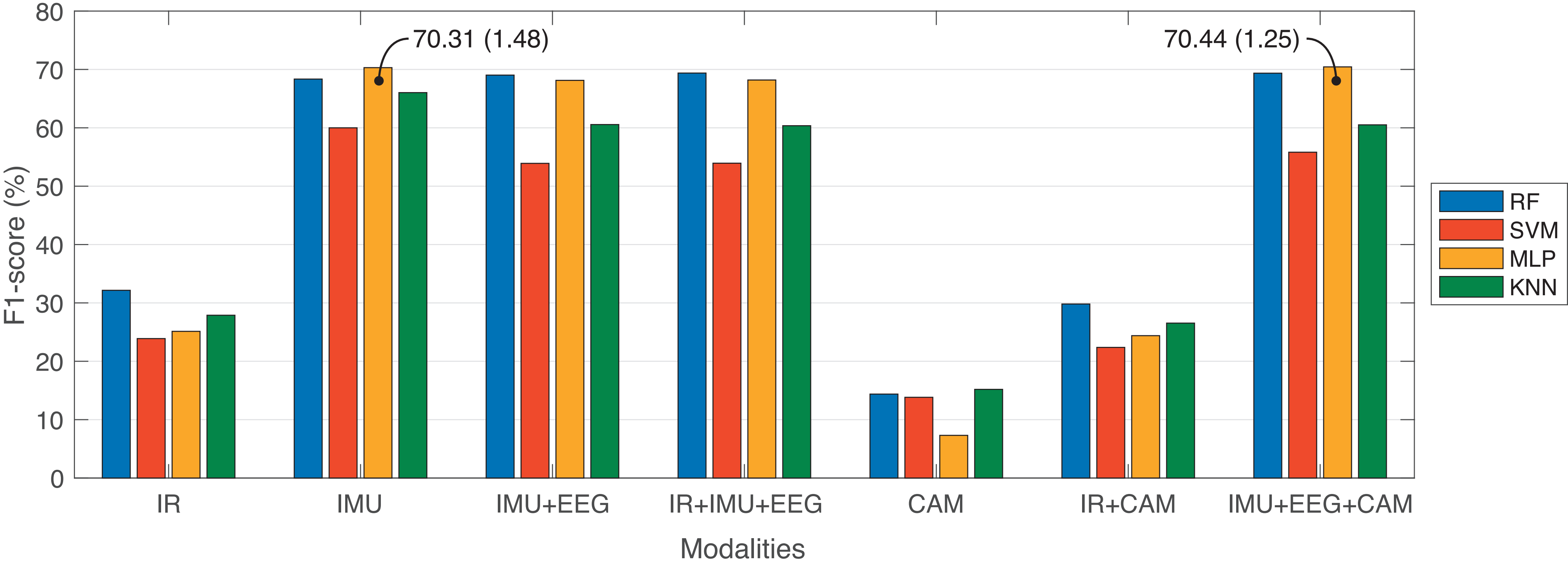

80
70
60
50
40
30
20
10
0
F1-score (%)
70.31 (1.48)
70.44 (1.25)
IR
IMU
IMU+EEG
IR+IMU+EEG
CAM
IR+CAM
IMU+EEG+CAM
Modalities
RF
SVM
MLP
KNN

Table 1

Modality	Model	Accuracy (%)
IR	RF (3 sec)	67.38 ± 0.65
	SVM (3 sec)	65.16 ± 0.90
	MLP (3 sec)	65.69 ± 0.89
	kNN (3 sec)	61.79 ± 1.47
IMU	RF (1 sec)	95.76 ± 0.18
	SVM (1 sec)	93.32 ± 0.23
	MLP (1 sec)	95.48 ± 0.25
	kNN (1 sec)	94.90 ± 0.18
IMU+EEG	RF (1 sec)	95.92 ± 0.29
	SVM (1 sec)	90.77 ± 0.36
	MLP (1 sec)	93.33 ± 0.55
	kNN (1 sec)	92.12 ± 0.31
IR+IMU+EEG	RF (2 sec)	95.12 ± 0.36
	SVM (1 sec)	90.59 ± 0.27
	MLP (1 sec)	93.26 ± 0.69
	kNN (1 sec)	92.24 ± 0.25
CAM	RF (3 sec)	32.33 ± 0.90
	SVM (2 sec)	34.40 ± 0.67
	MLP (3 sec)	27.08 ± 2.03
	kNN (3 sec)	34.03 ± 1.11
IR+CAM	RF (3 sec)	65.00 ± 0.65
	SVM (3 sec)	64.07 ± 0.79
	MLP (3 sec)	65.05 ± 0.66
	kNN (3 sec)	60.75 ± 1.29
	RF (1 sec)	95.09 ± 0.23

IMU+EEG+CAM	SVM (1 sec)	91.16 ± 0.25
	MLP (1 sec)	94.32 ± 0.31
	kNN (1 sec)	92.06 ± 0.24

Precision (%)	Sensitivity (%)	Specificity (%)
36.45 ± 2.46	31.26 ± 0.89	96.63 ± 0.07
26.77 ± 0.58	25.16 ± 0.29	96.31 ± 0.09
28.19 ± 3.56	26.40 ± 0.71	96.41 ± 0.08
30.04 ± 1.44	27.55 ± 0.97	96.05 ± 0.16
70.78 ± 1.53	66.91 ± 1.28	99.59 ± 0.02
66.16 ± 3.33	58.82 ± 1.53	99.32 ± 0.02
73.04 ± 1.89	69.39 ± 1.47	99.56 ± 0.02
69.05 ± 1.63	64.28 ± 1.57	99.50 ± 0.02
74.14 ± 1.29	66.29 ± 1.66	99.59 ± 0.03
62.51 ± 3.34	52.46 ± 1.19	99.03 ± 0.03
74.10 ± 1.61	65.32 ± 1.15	99.32 ± 0.05
66.86 ± 1.32	58.30 ± 1.20	98.89 ± 0.05
74.63 ± 1.65	66.71 ± 1.98	99.51 ± 0.03
64.75 ± 3.89	52.63 ± 1.42	99.01 ± 0.02
73.51 ± 1.59	66.05 ± 1.11	99.31 ± 0.07
67.33 ± 1.94	58.11 ± 1.61	99.21 ± 0.02
14.45 ± 1.07	14.48 ± 0.82	92.91 ± 0.09
13.81 ± 0.22	14.30 ± 0.31	92.97 ± 0.06
8.59 ± 1.69	10.59 ± 0.38	92.21 ± 0.09
15.32 ± 0.73	15.54 ± 0.57	93.09 ± 0.11
33.93 ± 2.81	29.02 ± 0.89	96.34 ± 0.07
24.10 ± 0.98	24.18 ± 0.17	96.17 ± 0.07
28.25 ± 3.20	25.40 ± 0.51	96.29 ± 0.06
29.91 ± 3.95	26.25 ± 0.90	95.95 ± 0.11
75.52 ± 2.31	66.23 ± 1.11	99.50 ± 0.02

66.79 ± 2.79	53.82 ± 0.70	99.07 ± 0.02
76.78 ± 1.59	67.29 ± 1.41	99.42 ± 0.03
68.82 ± 1.61	58.49 ± 1.14	99.19 ± 0.02

F1-score (%)
32.16 ± 0.99
23.89 ± 0.41
25.13 ± 1.09
27.89 ± 1.13
68.35 ± 1.25
60.00 ± 1.34
70.31 ± 1.48
66.03 ± 1.52
69.03 ± 1.48
53.91 ± 1.16
68.13 ± 1.16
60.56 ± 1.02
69.38 ± 1.72
53.94 ± 1.47
68.19 ± 1.02
60.36 ± 1.71
14.38 ± 0.89
13.83 ± 0.27
7.31 ± 0.82
15.19 ± 0.52
29.81 ± 1.16
22.38 ± 0.23
24.39 ± 0.88
26.54 ± 1.42
69.36 ± 1.35

55.82 ± 0.77
70.44 ± 1.25
60.51 ± 0.85

Table 2

#	IMU	
	RF	SVM
1	(98.36) Waist	(83.30) Right Pocket
2	(95.77) Neck	(83.22) Waist
3	(95.35) Right Pocket	(83.11) Neck
4	(95.06) Ankle	(82.96) Ankle
5	(94.66) Left Wrist	(82.82) Left Wrist

type	
MLP	**KNN**
(57.67) Right Pocket	(73.19) Right Pocket
(44.93) Neck	(68.73) Waist
(39.54) Waist	(65.06) Neck
(39.06) Left Wrist	(58.26) Ankle
(37.56) Ankle	(51.63) Left Wrist

Table 3

IMU type	Window	
	RF	SVM
Left Ankle	2-sec	3-sec
Waist	3-sec	1-sec
Neck	3-sec	3-sec
Right Pocket	3-sec	3-sec
Left Wrist	2-sec	2-sec

Length	
MLP	**KNN**
1-sec	3-sec
1-sec	2-sec
2-sec	2-sec
2-sec	2-sec
2-sec	2-sec

Table 4

#	Camer	
	RF	SVM
1	(62.27) Lateral View	(24.25) Lateral View
2	(55.71) Front View	(0.20) Front View

a view	
MLP	**KNN**
(13.78) Front View	(41.52) Lateral View
(5.51) Lateral View	(28.13) Front View

Table 5

Camera	Window	
	RF	SVM
Lateral View	3-sec	3-sec
Front View	2-sec	2-sec

Length	
MLP	**KNN**
2-sec	3-sec
3-sec	2-sec

Table 6

Multimodal	Classifier	Accuracy (%)
Waist + Lateral View	RF	98.72 ± 0.35
	SVM	95.59 ± 0.40
	MLP	77.67 ± 11.04
	KNN	91.71 ± 0.61
Right Pocket + Lateral View	RF	98.41 ± 0.49
	SVM	95.79 ± 0.58
	MLP	84.92 ± 2.98
	KNN	91.71 ± 0.58

Precision (%)	Sensitivity (%)	F1-score (%)
94.01 ± 1.51	97.63 ± 1.56	95.77 ± 1.15
100	70.26 ± 2.71	82.51 ± 1.85
33.73 ± 11.69	37.11 ± 26.74	29.81 ± 12.81
77.90 ± 3.33	61.64 ± 3.68	68.73 ± 2.58
93.64 ± 1.46	95.79 ± 2.65	94.69 ± 1.67
100	71.58 ± 3.91	83.38 ± 2.64
55.70 ± 11.36	48.29 ± 25.11	45.21 ± 14.19
73.63 ± 3.19	68.95 ± 2.73	71.13 ± 1.69

Table of Materials

Name of Material/ Equipment	Company	Catalog Number
Inertial measurement wearable sensor	Mbientlab	MTH-MetaTracker
Electroencephalograph brain sensor helmet MindWave	NeuroSky	80027-007
LifeCam Cinema video camera	Microsoft	H5D-00002
Infrared sensor	Alean	ABT-60
Bluetooth dongle	Mbientlab	BLE
Raspberry Pi	Raspberry	Version 3 Model B
Personal computer	Dell	

Comments/Description
Tri-axial accelerometer, tri-axial gyroscope and light intensity wearable sensor.
Raw brainwave signal with one forehand sensor.
2D RGB camera with USB cable interface.
Proximity sensor with normally closed relay.
Dongle for Bluetooth connection between the wearable sensors and a computer.
Microcontroller for infrared sensor acquisition and computer interface.
Intel Xeon E5-2630 v4 @2.20 GHz, RAM 32GB

COVER LETTER

October 16th, 2019.

Prof. Dr. Bing Wu
Advisory Board
Journal of Visualized Experiments (JoVE)

Dear Editor,

We are pleased to submit our revised manuscript entitled “Design and Analysis for Fall Detection System Simplification” for possible inclusion in Journal of Visualized Experiments (JoVE).

In this revised version, we enhanced the manuscript considering the editorial comments. Thank you for the valuable feedback.

Below, you will find the point-to-point revision of the comments.

Editorial:

Editorial comments	**Answers to the editorial**
1. Step 1.2.1-1.2.5: Do these steps also involve coding algorithms? As you can imagine, filming someone coding would not be visually interesting and helpful to readers. Please either provide some details for each step to show the readers unhighlight these steps.	These steps do not involve coding algorithms. They are planning steps, so we provided more details in the protocol. As a suggestion for the script, these steps can be filming while a person (e.g. scientist) is writing these plans in a whiteboard, setting the goal of the study, the target population, the types of falls and types of activities to be done during the experiment. This is a crucial phase in our protocol, so we would really like to see it on the video. Furthermore, we decided to highlight the steps that are more willing to be filming; trying to skip any coding algorithms or routines in the computer. The only exception is step 2.5.6 in which we detailed as much as we can about the coding settings. This is another big phase in our protocol and it is important to have a coherent story in the video.
2. For all computational steps, please specify all actions so that a script can be generated for filming the protocol. In your current protocol, a lot of steps are generally described. For example, in step 2.5.7: Compare the machine learning methods by testing the selected models with the testing data set. However, you did not explain	We included more details in all the steps highlighted for filming. As mentioned above, we decided to highlight the steps more important in the protocol, such that 2.5.7 is not highlighted anymore.

how to test the model. To generate a script, all actions involved in testing the sample need to be included. Please ensure that all details are included in your protocol.	

All the changes were highlighted in yellow at the Word file.

We are looking forward to hearing from you soon.

Best regards,

Lourdes Martínez-Villaseñor

Hiram Ponce