Fall Detection with LSTM and Attention Mechanism

Francesco Castro[1], Vincenzo Dentamaro[1] , Vincenzo Gattulli[1] and Donato Impedovo[1]

[1] *Department of Computer Science, University of Bari Aldo Moro, Via E. Orabona, 4, Bari, 70125, Italy*

Abstract

Falls in people are frequent and dangerous problems that can lead to death or infirmity, especially as people age. Therefore, fall detection and immediate intervention is essential to prevent more severe consequences. In this paper, two Fall Detection systems have been proposed. The proposed systems use 2D and 3D pose estimation of subjects respectively and a LSTM deep neural network architecture. An Attention Mechanism has been implemented in the neural network architecture to improve the performance of fall detection. The conducted experiments show that the model created with attention can generalize and does not suffer from scene bias. The 3D system using the Attention Mechanism has achieved an accuracy of 82% compared to the accuracy of 70% obtained without attention. With the 2D system using the neural network with Attention an accuracy of 72% has been obtained, instead, without attention, an accuracy of 52% has been obtained.

Keywords

Attention Mechanism, Deep Learning, Machine Learning, Fall Detection.

1. Introduction

Falling in the elderly is a public health problem as it can cause disabling fractures and, in addition, can also lead to psychological consequences that reduce a person's independence. Approximately 28-35% of people aged 65 years and older fall each year [1], a percentage that rises to 32-42% in those over the age of 70 [2], the frequency of falls increases with increasing age and frailty. Elderly people in nursing homes fall more frequently than those in the community. Approximately 30-50% of those hospitalized in long-term care fall each year, and 40% experience recurrent falls. Fall rates vary from nation to nation. For example, a study in the Southeast Asian region found that in China, 6 to 31 percent of elderly people fall each year, while in Japan, 20 percent fall. A study in the Americas region (Latin/Caribbean region) found that the percentage of elderly people who fall each year ranges from 21.6 percent in Barbados to 34 percent in Chile. However, many elderly people fall at home. In Italy, in 2002, it was estimated that 28.6% (26-31%) of people aged 65 and older fall within 12 months. Of these, 43% fall more than once. Sixty percent of falls occur at home. In the case of elderly people living in their own homes independently, about 50% of falls occur within their homes and immediate surroundings. They usually occur in used environments such as the bedroom, living room, kitchen, and bathroom. On the other hand, the rest of the falls occur in public settings or other people's homes [3].

In this paper, two Fall Detection systems are proposed. The first system uses 2D pose-estimation of subjects, and the second uses 3D pose-estimation of subjects. The neural network used for both systems mainly include LSTM and Attention Mechanism. Through this study, it is possible to show the efficiency of the Attention Mechanism in promptly detecting fall patterns from time series of features extracted from body joint coordinates.

The paper is structured as follows: the second section illustrates the state of the art of algorithms used by applications created for this purpose. Section three illustrates the Datasets used and the design of our experiment. Then, in Section four, experimental results will be described. Finally, Section five describes the conclusions and planned future work.

WAMWB'23: Workshop on Advances of Mobile and Wearable Biometrics, September 26, 2023, Athens, Greece

francesco.castro@unicampus.it (F. Castro); vincenzo.dentamaro@uniba.it (V. Dentamaro); vincenzo.gattulli@uniba.it (V. Gattulli); donato.impedovo@uniba.it (D. Impedovo)

0000-0002-8579-8941 (F. Castro); 0000-0003-1148-332X (V. Dentamaro); 0000-0001-9974-9414 (V. Gattulli); 0000-0002-9285-2555 (D. Impedovo)

2. Related works

In Fall Detection, the main techniques used can be divided into three categories:

1. Based on wearable devices;
2. Based on environmental sensors;
3. Based on image processing.

The architecture of systems based on wearable devices and environmental sensors is presented in Fig. 1:

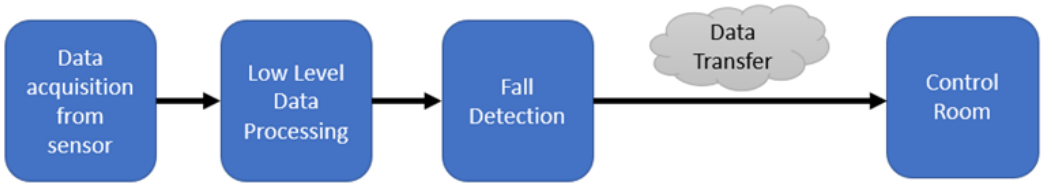

Figure 1: The architecture of systems based on wearable and environmental sensors.

In contrast, the architecture of vision-based systems i.e., using video or cameras is shown in Fig. 2 as discussed in [4]:

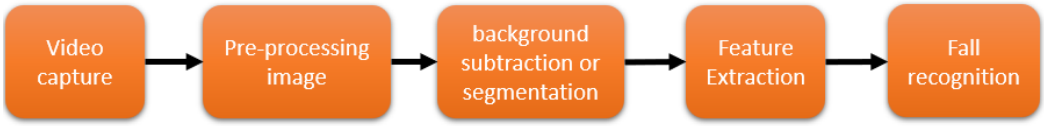

Figure 2: The architecture of vision-based fall detection systems.

2.1. Wearable Device-Based Techniques

Fall detection techniques based on wearable devices (smartwatches, pendants, etc...) have built-in sensors that can detect the movement and position of the subject wearing them. One of the most widely used methods to measure physical activities using wearable devices is the use of the accelerometer, which involves measuring the acceleration of various body parts in time. However, a barometric pressure sensor was introduced in [5] to improve accelerometer-based fall detection techniques. This article discusses a sensor that collects acceleration and atmospheric pressure data via a wearable device attached to the subject's waist, and then the collected data is analyzed offline using a decision tree as a classifier. Some of the wearable devices immediately available in the market include the Apple Watch Series 4/5 and Sense4Care Angel4; these devices detect falls using the same combination of accelerometer and gyroscope. In Apple Watch, the wrist's trajectory and the impact's acceleration are measured to detect the fall. Because these devices measure values and compare them to a predetermined threshold, accuracy and sensitivity are quite low, as is the resulting increase in false positives [6].

Physiological responses such as changes in heart rate or blood pressure can result from physical activity and changes in body position, and these physiological responses can be used for fall detection. In the paper [5], the authors developed a wrist-worn prototype consisting of a two-axis accelerometer and a posture sensor that integrated with a health monitoring device with tele-signaling capabilities for emergency telemedicine, enabling fall detection, blood pressure detection, and single-channel electrocardiogram. The measured biological signals have limited fidelity because the pulse area has limited body contact; however, this shortcoming could be overcome with further development of the posture sensor [5]. Triaxial Accelerometry uses triaxial accelerometers designed to detect acceleration simultaneously in three axial directions. In the article [7], the authors developed a system for Fall Detection that combines several triaxial acceleration sensors distributed throughout the body and allows the detection of the most damaged body parts after a fall. Fall detection systems based on wearable sensors are not robust;

these often fail to detect falls, or there are many false positives. This is because these systems are triggered directly by sudden and significant changes in acceleration. However, in real life, various Activities of Daily Living (ADLs), such as standing up, sitting down, or moving from standing to lying down, have strong similarities with falls.

2.2. Environmental Sensor-Based Techniques

Environment-based Fall Detection systems can monitor human posture to detect falls. Systems based on wearable sensors, as opposed to environment-based systems, are not sensitive to environmental changes because they do not consider environmental factors. However, ecological systems provide a solution by collecting data from the user and examining the environment. These systems use external sensors installed around the area where the user's daily activities are carried out, such as a home or elderly care facility [6]. Current research on environmental systems for fall detection is based on the following:

1. Ultrasonic signals/radar: Ultrasonic sensor/radar is a motion detector that monitors and recognizes moving subjects based on the ultrasonic wave; this type of sensor is one of the most widely used in environmental fall detection systems. In the article [8], a deep neural network was used to predict the fall, and they achieved 95.64% accuracy, while in the paper [9], the KNN algorithm was used to predict the fall, and they achieved 95.5% accuracy [9];
2. Kinect Sensors: The Kinect sensor is between a motion sensor and a camera; this is composed of an RGB camera, infrared projector, and Structured light depth detectors that also calculate the time of flight (Time of light), a measurement needed to know the time it takes for an object to travel a distance through a medium. Like ultrasonic sensors, the Kinect is easy to install, and there are no major privacy issues; these sensors have been adopted in fall detection in articles [10];
3. Microphone: Regarding the microphone, the basic idea is to capture and analyze acoustic information to identify a fall; this device is cheap, small, and easy to purchase and install;
4. Pressure sensor: Pressure sensor is another popular feature in environmental systems; this type of sensor is usually installed under the floor to detect floor vibration and pressure to identify a fall. A device-less fall detection system based on a Raspberry Pi and three geophones is proposed in the article [11]. The fall mode is decomposed and characterized by time-dependent floor vibration characteristics and exploiting a Hidden Markov Model (HMM) achieves 95.74% accuracy. The disadvantage of pressure sensors is the high false positive it generates since they will consider a large object falling on the floor as a fall, moreover even if the sensor itself is inexpensive, it must be installed under the floor of the entire living area, which requires major home renovations and a complicated power supply for each sensor that ends up increasing the cost;
5. Infrared/Wi-Fi signals: Infrared/Wi-Fi signals for detection in the fall are usually used together with the other types of environmental sensors to increase accuracy. An infrared sensor combined with a pressure sensor is presented in the article [12], which achieved 96.7 percent specificity and 100 percent sensitivity. The infrared image is used to observe the entire environment while the pressure sensors analyze the action of the floor; this combination reduces the false alarm rate, as in the scenario where a large object falls to the floor or a slow fall occurs, there would be no false positives as the infrared sensor would identify it.

Overall, these environmental systems have the advantage of operating in a low-light region and not being limited by privacy issues compared to vision-based systems. These systems also provide a more comprehensive analysis of the user's posture by considering the environmental factors than wearable systems based on inertial sensors. The limitations of the ecological system are as follows:

- They are only suitable for indoor environments and cannot be installed or used outdoors;
- Since the sensors are always in fixed positions, it is challenging to implement such systems;

- Most environmental systems can only detect one person in the monitored area, which means no pets, no partners, and no friends;
- Although the sensors used are inexpensive, installing such a system requires a major renovation of the house because most of the sensors will be embedded under the floor or in the wall, which can be an expensive setup.

2.3. Image Processing-Based Techniques

In addition to wearable devices and environmental sensors, vision-based systems using cameras can also be used for fall detection, which requires immediate assistance if a fall occurs. These systems usually use depth cameras or RGB cameras; depth cameras can calculate 3D information using a single camera and perform better in low light conditions [3]. On the other hand, the RGB camera is a standard camera that cannot capture 3D information and perform in common light conditions. However, these limitations can be overcome by using multiple RGB cameras with infrared sensors. In general, both RGB and depth cameras work well for fall detection. The main difference between the various systems based on image processing is in the Feature Extraction stage, there are four main methods for Feature Extraction, and they are as follows:

1. Shape change monitoring: The method involving Shape change monitoring approximates the subject using an estimated shape such as an ellipse or rectangle; this technique requires fewer processing resources and is easy to model, which is why most real-time vision systems adopt it.;
2. Postures figuring: The method of postures figuring is much more detailed than the method described above because it traces the subjects' joints or draws the body's contour to specify their posture, leading to greater accuracy. In the paper [13], after background subtraction, a Kalman filter with OpenCV is used to track the person by identifying a set of points in the areas of interest; the system is sensitive enough to notice small movements when the subject is stationary. Next, the KNN algorithm is used to predict a fall. The system described in the article [13] achieved 96.9% of accuracy;
3. Key point tracking: This method typically projects the subject's posture but only checks a few key points instead of all pixels. The article [14] proposes a robust fall detection system based on tracking human body parts using a depth camera. To do this, the system calculates the 3D trajectory of the head joint, and this data is given as input to the SVM which determines whether a falling motion has occurred. This system achieved an accuracy of 97.6%;
4. Inactivity detecting: Regarding inactivity detection, it is the fastest method since it requires almost no processing. However, this method is seldom used alone because of its high false alarm rates, plus it requires subjects to lie on the ground for a while, putting their lives at stake, to detect a fall. Therefore, this method is always combined with the three methods mentioned above.

Fall detection systems based on image processing are like environmental systems in that they share advantages in analyzing ecological factors, which wearable devices do not have; plus, both have the problem that they can only be used indoors and often have blind spots and huge expenses. However, image processing systems have unique advantages. Thanks to pattern recognition, image processing systems can identify, track, and monitor the target user even when multiple people or pets are in the monitored area. However, such video recording systems are always subject to privacy issues, plus by using complex real-time image processing algorithms, they require a huge amount of space and computing power [4].

3. Materials and methods

3.1. Dataset

Two datasets were used for the realization of the Fall Detection system. The first dataset is the UR Fall Detection Dataset (URFD). It contains:

- Total of 70 videos
- These videos have both images in RGB format and another format, but only those in RGB format were considered in our case.
- 30 videos contain falls and 40 do not.
- 30 videos containing falls do not directly contain the person falling, but rather, the person performing daily activities falls in different places. This way, an attempt is made to make the fall as realistic as possible.
- 40 videos that do not contain falls present people performing these daily activities in different places, such as lying down, sitting, etc.;
- There are also fall-like movements, such as squatting and picking up an object from the ground, to confuse future classification.

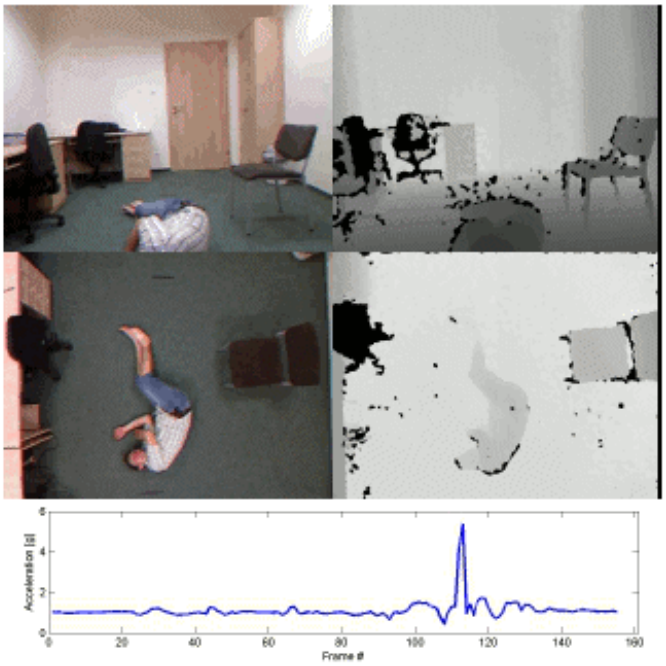

Figure 3: UR Fall Detection Dataset example.

The second dataset used Le2i is the one described in (Charfi et al., 2012); It contains:

- 250 videos, of which 192 contain falls and 57 contain everyday activities.
- The videos show the main difficulties an elderly person might encounter in an enclosed environment.
- In the case of videos with falls, as in the previous dataset, we have a person while performing daily activities fall; on the other hand, in the videos where falling is not present also here people perform various activities such as lying down, picking up an object from the ground, sweeping, going downstairs, etc...., in this way we try to simulate as many daily life scenarios as possible.
- In addition, the lighting is highly variable, as well as shadows and reflections that can be detected as moving objects.

The videos show people always dressed differently and in four main environments, which are as follows:

- Home;
- Office;
- Reading room;
- Coffee room.

Figure 4: Le2i dataset.

This way, the system's independence from the environment can be assessed. Of this dataset, however, not all the videos were used for the realization of the system since they are no longer available; in fact, a total of 201 videos were used; of these, 110 contain falls, and 91 do not contain falls.

Table 1
Number of videos for each environment

Category	Number of videos	Fall videos	Non-fall videos
Home	60	38	22
Lecture room	27	18	9
Coffee room	70	60	10
Office	64	48	16

3.2. Design

This subchapter reviews all stages of creating the system using both 2D and 3D coordinates of the subjects' poses, starting with creating the dataset and arriving at the classification stage. The proposed system scheme is shown in Figure 5.

The videos from the first and second datasets previously described were processed in the following way. The following pre-processing steps were performed for each of these videos.

1. Extraction and storage of 2D/3D coordinates from the video frames;
2. Noise Elimination;
3. Extraction of Spatio-temporal features and sigma log normal;
4. Creation of the final dataset in CSV format.

3.2.1. Extraction and storage of 2D and 3D

Each video was divided into frames for the creation and storage of 2D and 3D pose coordinates, and for each frame, the pose of the person present in the frame was estimated. For the extraction of 2D coordinates of the person's body, the OpenPose library was used for the extraction of 3D coordinates, FrankMocap was used. OpenPose represents the first real-time multi-person system to jointly detect key points of the human body, hand, face, and foot on single images; then, given a video, OpenPose divides it into frames and for each one extracts the 2D coordinates of each body part and stores them in a file. In contrast, FrankMocap is a 3D motion capture system that,

given a video, transforms it into frames and, for each frame, provides an estimate of 3D poses for a person's body and hands. This system, therefore, makes it possible to obtain estimates of the 3D poses of people within the various frames without having a complex system of cameras capturing the subject from different perspectives.

OpenPose, as we have already mentioned, produces as output a file containing the extracted 2D coordinates for each body part, i.e., the key points. The key points extracted for the body parts of a person are 25 in total. However, not all key points were used to realize the fall detection system. Only 14 key points were used, which are the following: Nose, Neck, right shoulder, RightElbow, RigthWrist, LeftShoulder, LeftElbow, LeftWrist, RightHip, RightKnee RightAnckle, LeftHip, LeftKnee, LeftAnckle. All key points were not used because they were not relevant to the Fall Detection system realizations; in fact, the points that were not taken into account are related to, for example, the ears, heel, eyes, and toes, etc...

In the case where OpenPose could not extract the 2D coordinates of the various key points, since it extracts the coordinates only if they are visible in the frame, these coordinates were interpolated according to the value that the coordinate has in the other frames, in some cases (precisely in 15 videos) instead the videos were too dark and Open Pose was not able to extract any points. For 3D coordinate extraction, on the other hand, as has already been mentioned, FrankMocape was used, and the same key points used for 2D plus MidHip were extracted and used. With FrankMocap, you do not have the problem of null coordinates of key points since it already estimates all coordinates.

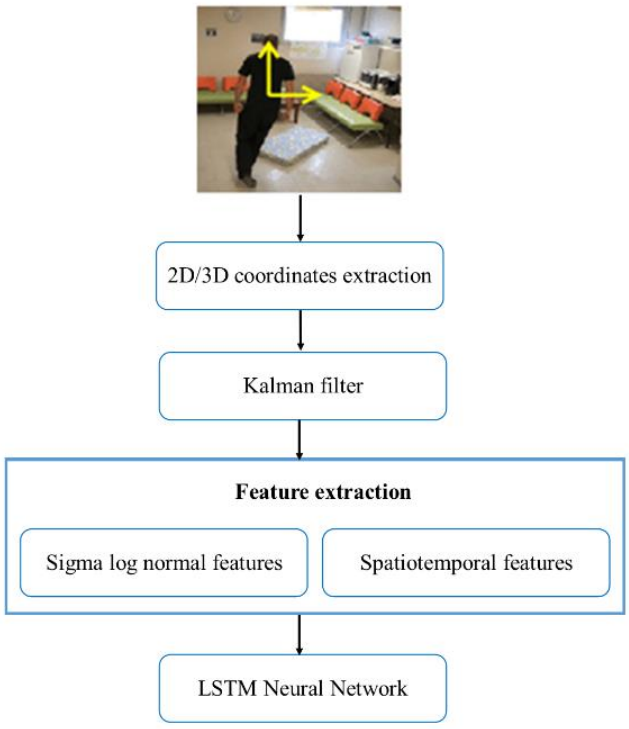

Figure 5: The scheme of the proposed system.

3.2.2 Noise eliminations

For noise elimination in the extracted 2D and 3D key points, the Kalman Filter was applied. The Kalman filter is an algorithm that, given observed measurements over time that contain noise, provides an estimate of values that tend to be more accurate than the actual measurements.

3.2.3 Extraction of spatiotemporal features and sigma log-normal

After applying the Kalman filter, 2D and 3D features were computed on the key points of the various body parts. For both 2D and 3D, two types of features were computed, which are as follows:

- Spatiotemporal features;
- The sigma log-normal features.

Spatiotemporal features make it possible to analyze the movement of people in space and understand when a person is falling. The features calculated for 2D were as follows: Joint Displacement, Displacement in x, Displacement in y, Joint Velocity, Velocity in x, Velocity in y, Joint Acceleration, Acceleration in x, Acceleration in y, and Tangent Angle to Trajectory. For the displacement calculation for each video, the coordinates of the 2D key points in pairs of frames were taken. In contrast, the difference between the time of the first frame and the time of the second frame was taken as the reference time to calculate the various velocities and accelerations. For the 3D key points, on the other hand, the spatiotemporal features calculated were as follows: Joint Displacement, Displacement in x, Displacement in y, Displacement in z, Joint Velocity, Velocity in x, Velocity in y, Velocity in z, Joint Acceleration, Acceleration in x, Acceleration in y, Acceleration in z, Angle tangent to the trajectory. The calculation of these features was performed in the same way as the 2D features, however, using the coordinates of the 3D key points. Normal sigma log features are usually studied in the field of digital signatures: in fact, they are an essential tool of signature recognition as they allow us to learn of the variations of Acceleration and pressure during motion. These features have been used because they allow for the analysis of variations in Acceleration, a fundamental aspect of the knowledge of falls. These types of features are computed for both 3D and 2D for each body part within each frame of the video and associated with spatiotemporal features based on time, while the number of normal sigma logs extracted for each frame of the video is repeated for all body parts in the frame.

3.2.4 Creation of the dataset in CSV format

After extracting the 2D features for each frame of the videos in the two datasets, they were labeled with one of the videos falling otherwise with 0 and were stored in a CSV file, one for each video. Then two CSV files were created representing the two final datasets for 2D; the first CSV file is the union of the previously created CSV files related to the videos belonging to the first dataset; also, the second CSV file is the union of all the CSV files created previously however associated with the videos belonging to the second dataset. For the creation of the final datasets with 3D coordinates, the same procedure described above was carried out as for the 2D coordinates.

3.2.5 Classifier

A Neural Network with the following architecture was used as the classifier. Its architecture is shown in Fig. 6. The 2D and 3D datasets were given as input to the neural network in the form of a vector of sequences, where each sequence is a video, i.e., the set of appropriately labeled frames that make up the video itself. Padding was performed to ensure that all the sequences had the same size by adding zeros to the beginning of the sequences. As can be seen from the figure above showing the network architecture used, we have a masking layer, which is used to disregard the various zeros inserted in the sequences to make sure that the various sequences have the same length; then we have an LSTM layer with some neurons equal to 64, a Dropout layer to reduce overfitting of the model with a rate of 0. 2, the Attention layer with 128 neurons, a dense layer with 34 neurons a Normal initializer kernel and the L2 regularize that further decreases the overfitting, finally we have a Dens layer with two neurons, as we have two classes dropped and not dropped, with SoftMax activation function.

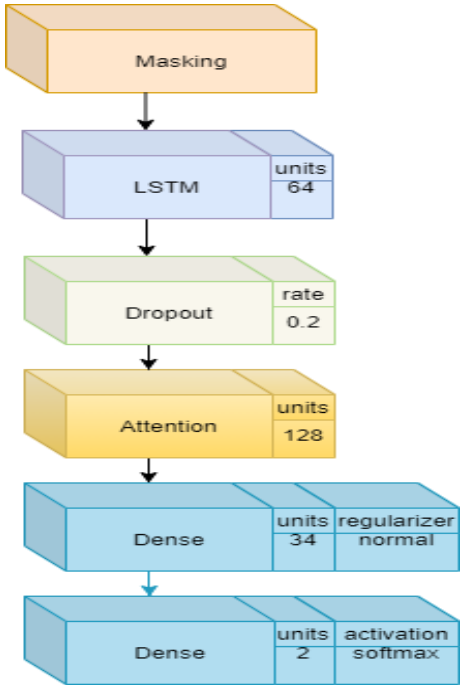

Figure 6: The architecture of the neural network used.

4. Experiments and results

This chapter presents the conducted experiments and the results obtained using training validation, testing strategies, and the databases created from 2D and 3D coordinates. The experiments evaluate the system performance with and without Attention Mechanism in term of Precision, Sensitivity, Specificity, F1-measure, Accuracy and AUC_ROC.

4.1 Training validation and test

The two previously created datasets for both 2D and 3D were used for the Training Validation and Test phase. Two strategies were used for this phase:

The first strategy is to merge the two datasets into one dataset and then divide them into three parts for each phase (Training, Validation, and Test). The split made was as follows (75% of the initial dataset for the Training phase, 12% of the initial dataset for the Validation phase, and 13% of the initial dataset for the Test phase). On the other hand, the second strategy uses the first dataset for the Training and Validation phase, precisely 77% for Training and 23% for Validation, and the second dataset for the Testing phase.

With both the first and second strategies during the training phase, the model was trained on 150 epochs on the training set; at each epoch, Validation was carried out with the validation set and stored the model with the highest accuracy among the epochs. At the end of Training and Validation, testing was carried out with the test set, using the best model obtained among the epochs. Two strategies were used for this phase, performing both an intra-dataset and inter-dataset test to see the model's behavior on completely different videos than those used for the training and validation phase. For the testing phase, the metrics used were as follows: Precision, Accuracy, Sensitivity, Specificity, F1-measure, and AUC_ROC.

4.2 Results obtained with the First Strategy

The first strategy, described in the previous chapter, of combining the two datasets into a single dataset and then dividing it into three parts one for each phase (Training, Validation, and Testing) was repeated 20 times first using the neural network with Attention and then using the

neural network without Attention. The average results obtained over the 20 repetitions with the dataset created from the 2D coordinates are as follows (Table 2).

Table 2
First 2D dataset strategy results on URFD dataset

Metrics	With Attention	Without Attention
Precision	**0.897 +/- 0.088**	0.772 +/- 0.12
Sensitivity	**0.91 +/- 0.09**	0.807 +/- 0.109
Specificity	**0.883 +/- 0.103**	0.73 +/- 0.167
F1-measure	**0.898 +/- 0.06**	0.778 +/- 0.065
Accuracy	**0.897 +/- 0.061**	0.768 +/- 0.077
AUC_ROC	**0.897 +/- 0.061**	0.768 +/- 0.077

The results obtained with the dataset created from the 3D coordinates are as follows (Table 3).

Table 3
First 3D dataset strategy results on URFD dataset

Metrics	With Attention	Without Attention
Precision	**0.894 +/- 0.082**	0.699 +/- 0.066
Sensitivity	**0.802 +/- 0.127**	0.706 +/- 0.084
Specificity	**0.892 +/- 0.099**	0.689 +/- 0.126
F1-measure	**0.834 +/- 0.07**	0.696 +/- 0.031
Accuracy	**0.847 +/- 0.055**	0.696 +/- 0.037
AUC_ROC	**0.847 +/- 0.055**	0.697 +/- 0.036

4.3 Results obtained with the Second Strategy

The results obtained using the second strategy for the training validation and testing phase, the one described in the previous paragraph, which is to use the two initial datasets separately one for the training and validation phase and the other for the testing phase, were as follows: Using the network with Attention and the network without attention as a classifier with the 2D dataset were as follows (Table 4).

Table 4
Second 2D dataset strategy results on LE2I dataset

Metrics	With Attention	Without Attention
Precision	**0.633**	0.0
Sensitivity	**0.95**	0.0
Specificity	**0.542**	0.958
F1-measure	**0.76**	0.0
Accuracy	**0.727**	0.523
AUC_ROC	**0.746**	0.479

In contrast, the results obtained with the 3D dataset are as follows (Table 5).

Table 5
Second 3D dataset strategy results on LE2I dataset

Metrics	With Attention	Without Attention
Precision	**0.9**	0.625
Sensitivity	**0.474**	0.263

Specificity	**0.974**	0.923
F1-measure	**0.621**	0.37
Accuracy	**0.81**	0.707
AUC_ROC	**0.724**	0.593

As can be seen, in the second strategy it is impossible to have average results because a separate dataset was used as the test set, consequently, the examples in the test set would not change if more iterations were performed. We summarize the results obtained in the following graphs to get a more immediate view of the results obtained, as shown in Figures 7, 8, 9 and 10.

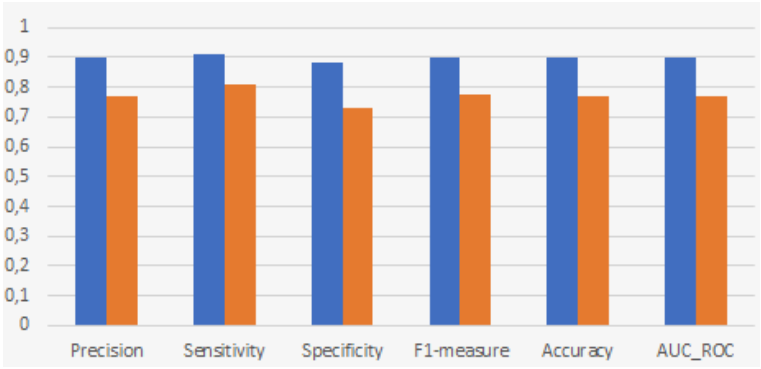

Figure 7: Result obtained with the first strategy on 2D dataset (Blue: With Attention, Orange: Without Attention).

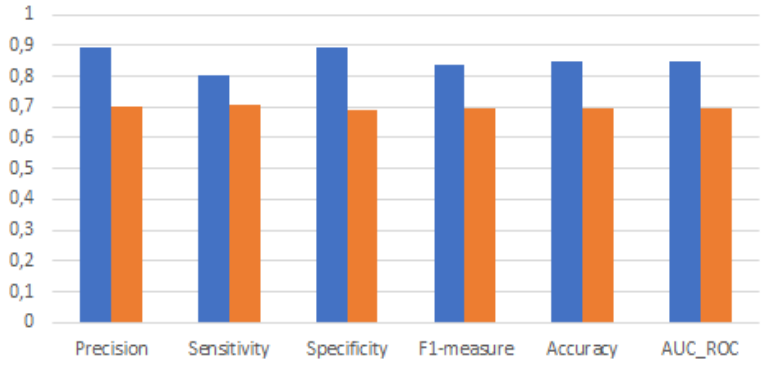

Figure 8: Result obtained with the first strategy on 3D dataset (Blue: With Attention, Orange: Without Attention).

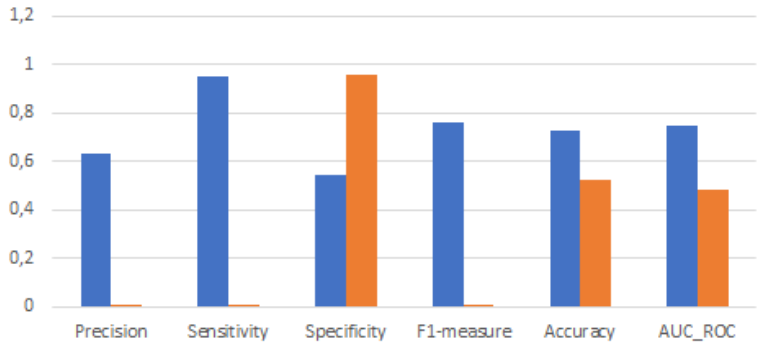

Figure 9: Result obtained with the second strategy on 2D dataset (Blue: With Attention, Orange: Without Attention).

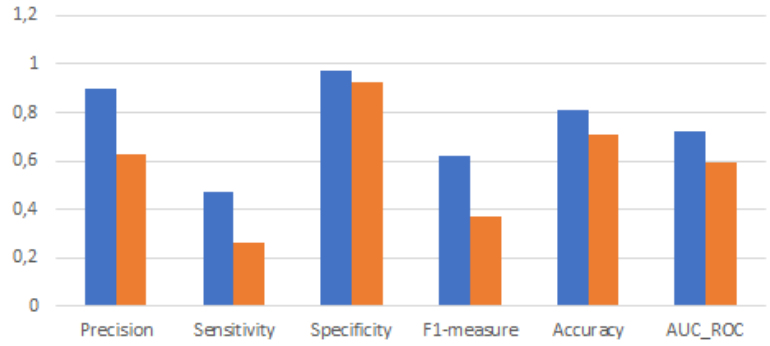

Figure 10: Result obtained with the second strategy on 3D dataset (Blue: With Attention, Orange: Without Attention).

The neural network with attention performs significantly better than the neural network without attention, using both the first and second strategies for training validation and testing, as shown in Figures 7, 8, 9 and 10.

Table 6 and Table 7 present state-of-the-art comparisons of the same datasets. Unfortunately, it is not easy to compare accuracies because of the different settings of the tests: some works used 80% of the dataset instances for training and 20% for testing, while in our work we used a 10-Fold cross-validation setting. Looking at the results on the URFD dataset it is possible to observe that our solution achieved comparable results concerning other solutions. On the Le2I dataset, instead, our solution both in 2D and 3D achieved inferior results. This can be mainly imputed to the decided experimental setting which was selected to decrease the chance of bias and create inaccurate predictions.

Table 6
Comparison with other works on URFD dataset

Work in	Accuracy
Vishnu et al., 2021	82.1%
Killian et al., 2021	88.8%
Yao et al., 2020	90.53%
This Work 2D Solution with attention	89.7%

Table 7
Comparison with other works on LE2I dataset

Work in	Accuracy
Vishnu et al., 2021	86.8%
Chhetri et al., 2021	91.4%
Asif et al., 2020	89.9%
This Work 3D Solution with attention	81%

5. Conclusions

In this paper two Fall Detection systems have been proposed. The first system uses 2D pose-estimating subjects, and the second uses 3D pose-estimating subjects by a neural network with attention and without attention. The neural network with attention achieves significantly better results than the neural network without attention, using both the first and second strategies for training validation and testing (Fig. 6-7). Note especially that in the second strategy, where an entire dataset is used as a test set, there are also good results here with both 2D and 3D. In fact, with the 2D dataset, have an Accuracy and Precision, using the neural network with attention of 72% and 63% respectively, while not using attention, an accuracy of 52% and a Precision of 0% have been obtained. In contrast, with the 3D dataset, using the neural network with attention, we

achieve an Accuracy of 81% and a Precision of 90%. In comparison, without attention, we achieve an Accuracy of 70% and a Precision of 62%. This shows us that the model with attention can be generalized because it does not bind to Training data. In addition, the model created with the attention mechanism is also able to generalize because it does not suffer from scene bias since, although there are many different scenarios in the videos, the attention mechanism manages to put "attention" on the subject present in the video while neglecting the context/scenario. It is also possible to observe that applying the Attention mechanism to perform time series modeling of body joint coordinates results in a sensible jump in accuracies, showing that temporal pattern weighting is a winning strategy when it comes to fall detection and, in general, activity recognition. In the future, other state-of-the-art technologies such as the Multi-Speed Transformer, which uses the multiresolution intuition to model time series classification with a fast and a slow branch, will be used in the same conditions to improve the fall detection performance.

Acknowledgements

Francesco Castro is a PhD student enrolled in the National PhD in Artificial Intelligence, XXXVIII cycle, course on Health and life sciences, organized by Università Campus Bio-Medico di Roma. The research of Dott. Vincenzo Gattulli was funded by PON Ricerca e Innovazione 2014-2020 FSE REACT-EU, Azione IV.4 "Dottorati e contratti di ricerca su tematiche dell'innovazione" CUP H99J21010060001.

This project is supported by the AMICA project.

References

[1] A. J. Blake et al., “Falls by elderly people at home: prevalence and associated factors.,” Age Ageing, vol. 17, no. 6, pp. 365–372, Nov. 1988, doi: 10.1093/AGEING/17.6.365.

[2] M. E. Tinetti, M. Speechley, and S. F. Ginter, “Risk factors for falls among elderly persons living in the community,” N Engl J Med, vol. 319, no. 26, pp. 1701–1707, Dec. 1988, doi: 10.1056/NEJM198812293192604.

[3] J. Mehta, G. Czanner, S. Harding, D. Newsham, and J. Robinson, “Visual risk factors for falls in older adults: a case-control study,” BMC Geriatr, vol. 22, no. 1, pp. 1–9, Dec. 2022, doi: 10.1186/S12877-022-02784-3/TABLES/3.

[4] C. Jariyavajee, A. Faphatanchai, W. Saeheng, C. Tuntithawatchaikul, B. Sirinaovakul, and J. Polvichai, “An Improvement in Fall Detection System by Voting Strategy,” 34th International Technical Conference on Circuits/Systems, Computers and Communications, ITC-CSCC 2019, Jun. 2019, doi: 10.1109/ITC-CSCC.2019.8793440.

[5] J. M. Kang, T. Yoo, and H. C. Kim, “A wrist-worn integrated health monitoring instrument with a tele-reporting device for telemedicine and telecare,” IEEE Trans Instrum Meas, vol. 55, no. 5, pp. 1655–1661, Oct. 2006, doi: 10.1109/TIM.2006.881035.

[6] Z. Wang, V. Ramamoorthy, U. Gal, and A. Guez, “Possible Life Saver: A Review on Human Fall Detection Technology,” Robotics 2020, Vol. 9, Page 55, vol. 9, no. 3, p. 55, Jul. 2020, doi: 10.3390/ROBOTICS9030055.

[7] C. F. Lai, S. Y. Chang, H. C. Chao, and Y. M. Huang, “Detection of cognitive injured body region using multiple triaxial accelerometers for elderly falling,” IEEE Sens J, vol. 11, no. 3, pp. 763–770, 2011, doi: 10.1109/JSEN.2010.2062501.

[8] H. Sadreazami, M. Bolic, and S. Rajan, “TL-FALL: Contactless Indoor Fall Detection Using Transfer Learning from a Pretrained Model,” Medical Measurements and Applications, MeMeA 2019 - Symposium Proceedings, Jun. 2019, doi: 10.1109/MEMEA.2019.8802154.

[9] Y. T. Chang and T. K. Shih, “Human fall detection based on event pattern matching with ultrasonic array sensors,” Ubi-Media 2017 - Proceedings of the 10th International Conference on Ubi-Media Computing and Workshops with the 4th International Workshop

on Advanced E-Learning and the 1st International Workshop on Multimedia and IoT: Networks, Systems and Applications, Oct. 2017, doi: 10.1109/UMEDIA.2017.8074149.
[10] J. Barabas, T. Bednar, and M. Vychlopen, "Kinect-based platform for movement monitoring and fall-detection of elderly people," 2019 Proceedings of the 12th International Conference on Measurement, MEASUREMENT 2019, pp. 199–202, May 2019, doi: 10.23919/MEASUREMENT47340.2019.8780004.
[11] Y. Huang, W. Chen, H. Chen, L. Wang, and K. Wu, "G-Fall: Device-free and Training-free Fall Detection with Geophones," Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks workshops, vol. 2019-June, Jun. 2019, doi: 10.1109/SAHCN.2019.8824827.
[12] H. W. Tzeng, M. Y. Chen, and J. Y. Chen, "Design of fall detection system with floor pressure and infrared image," 2010 International Conference on System Science and Engineering, ICSSE 2010, pp. 131–135, 2010, doi: 10.1109/ICSSE.2010.5551751.
[13] K. de Miguel, A. Brunete, M. Hernando, and E. Gambao, "Home Camera-Based Fall Detection System for the Elderly," Sensors 2017, Vol. 17, Page 2864, vol. 17, no. 12, p. 2864, Dec. 2017, doi: 10.3390/S17122864.
[14] Z. P. Bian, J. Hou, L. P. Chau, and N. Magnenat-Thalmann, "Fall detection based on body part tracking using a depth camera," IEEE J Biomed Health Inform, vol. 19, no. 2, pp. 430–439, Mar. 2015, doi: 10.1109/JBHI.2014.2319372.
[15] I. Charfi, J. Miteran, J. Dubois, M. Atri, and R. Tourki, "Definition and performance evaluation of a robust SVM based fall detection solution," 8th International Conference on Signal Image Technology and Internet Based Systems, SITIS 2012r, pp. 218–224, 2012, doi: 10.1109/SITIS.2012.155.
[16] C. Vishnu, R. Datla, D. Roy, S. Babu, and C. K. Mohan, "Human Fall Detection in Surveillance Videos Using Fall Motion Vector Modeling," IEEE Sens J, vol. 21, no. 15, pp. 17162–17170, Aug. 2021, doi: 10.1109/JSEN.2021.3082180.
[17] L. Killian et al., "Fall prevention and detection in smart homes using monocular cameras and an interactive social robot," GoodIT 2021 - Proceedings of the 2021 Conference on Information Technology for Social Good, pp. 7–12, Sep. 2021, doi: 10.1145/3462203.3475892.
[18] C. Yao, J. Hu, W. Min, Z. Deng, S. Zou, and W. Min, "A novel real-time fall detection method based on head segmentation and convolutional neural network," J Real Time Image Process, vol. 17, no. 6, pp. 1939–1949, Dec. 2020, doi: 10.1007/S11554-020-00982-Z/TABLES/4.
[19] S. Chhetri, A. Alsadoon, T. A. D. in, P. W. C. Prasad, T. A. Rashid, and A. Maag, "Deep Learning for Vision-Based Fall Detection System: Enhanced Optical Dynamic Flow," Comput Intell, vol. 37, no. 1, pp. 578–595, Mar. 2021, doi: 10.1111/coin.12428.
[20] U. Asif, S. von Cavallar, J. Tang, and S. Harrer, "SSHFD: Single Shot Human Fall Detection with Occluded Joints Resilience," Frontiers in Artificial Intelligence and Applications, vol. 325, pp. 2656–2663, Apr. 2020, doi: 10.48550/arxiv.2004.00797.