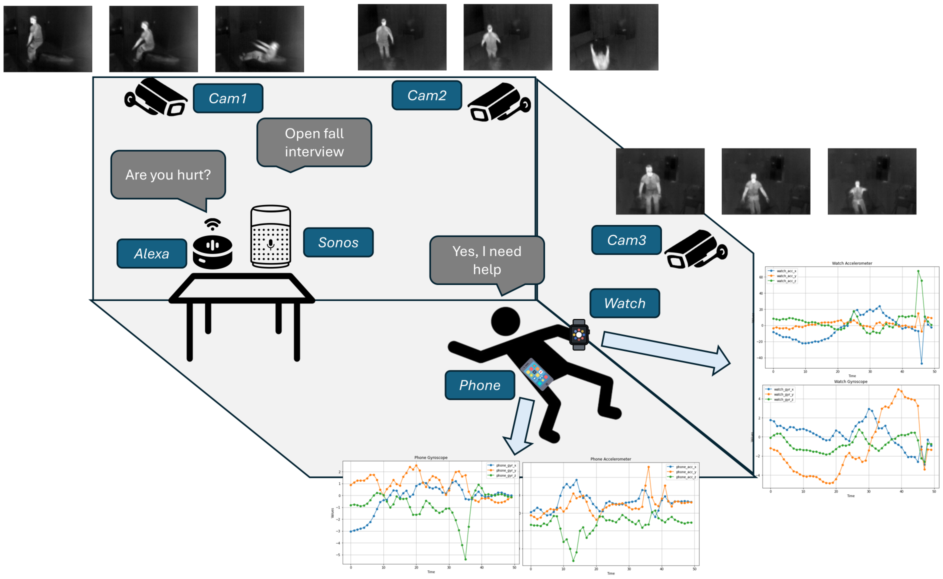
Graphical Abstract

Privacy-aware Fall Detection and Alert Management in Smart Environments using Multimodal Devices



Highlights

Privacy-aware Fall Detection and Alert Management in Smart Environments using Multimodal Devices

- Thermal imaging can be used to detect falls while preserving privacy.
- Multimodal detection is crucial for selecting the appropriate devices.
- Convolutional and recurrent neural networks are appropriate for detecting falls.
- A service with a voice assistant and a large language model reduce false positives.

Privacy-aware Fall Detection and Alert Management in Smart Environments using Multimodal Devices

Abstract

Falls are a leading cause of injury and mortality, especially among the elderly. While camera-based fall detection systems have shown success, they raise significant privacy concerns. Alternatives using wearable sensors or thermal cameras offer comparable accuracy but have yet to be combined for fall detection. Additionally, most research focuses on fall detection without addressing post-fall user condition or personalized alerts. This study aims to develop a privacy-aware fall detection and alert system leveraging both wearable sensors and thermal cameras. The system improves detection accuracy, addresses privacy concerns, and enhances alert management through personalized responses. We propose an Internet of Things (IoT)-based system integrating thermal cameras and wearable sensors. Edge-based computation enables real-time detection, with Internet connectivity used only for sending alerts in case of a fall. Various machine learning algorithms and sensor data are evaluated to determine their impact on detection accuracy. The system also includes voice interaction for user engagement. Experimental results show that fall detection using a convolutional neural network with thermal images from three viewpoints achieves an F1-score above 0.98. Similarly, traditional machine learning algorithms applied to wearable sensor data showed high performance. Post-processing techniques effectively reduce false positives, improving reliability. The proposed system ensures high accuracy while addressing privacy concerns. By integrating multimodal devices and edge-based computing, it offers a scalable, real-time solution for smart environments, ensuring timely responses through personalized alerts after falls.

Keywords: Fall detection, Multimodal devices, Alert system, Deep Learning, Thermal cameras

1. Introduction

Following road traffic accidents, falls represent the second most common cause of mortality globally [1]. The predominant proportion of falls is attributable to physiological and pathological alterations (morbidity, functional decline, inactivity, depression, and loss of autonomy [2]) associated with aging [3], a concern expected to escalate due to the increasing age of the population. The demographic of people over 60 years is projected to double in 2047 and triple in 2079 [4], presenting a formidable challenge to public services and the financial resources necessary to mitigate their impact [5, 6].

In this context, Ambient Assisted Living (AAL) systems can provide crucial support by enabling continuous monitoring and detection of abnormal situations, such as falls [7, 8], thus reducing hospitalizations and associated costs [9]. Current Fall Detection (FD) solutions generally rely on body-worn devices (e.g., sphygmomanometers) and cameras, which have demonstrated high accuracy in identifying falls [10]. However, deploying these systems in real-world environments remains a challenge due to high false negative rates, and privacy concerns have emerged as a significant barrier to wider adoption.

Thermal sensors have recently been proposed as an alternative that preserves user privacy while maintaining performance in low-visibility conditions [11]. Similarly, wearable sensors such as wristbands and smartphones offer a mobile solution for fall detection without spatial restrictions, though they typically exhibit lower precision compared to camera-based methods [12].

The integration of vision-based systems, ambient sensors, and wearable devices, termed "multimodal FD," has emerged as a promising approach, with the potential to better adapt to diverse environments and improve detection accuracy [12]. Although multimodal datasets combining cameras, wearable sensors, passive infrared sensors, and microphones exist [13, 14], thermal sensors have not yet been widely incorporated into these datasets. Moreover, there has been limited assessment of the strengths and weaknesses of different sensor modalities in identifying fall events across various contexts.

While the primary focus of many FD systems is maximizing detection accuracy, the critical task of delivering personalized alerts to caregivers or healthcare professionals has been largely overlooked. Most alert systems fail to account for the user's condition following a fall, such as whether the individual has recovered, sustained injuries, or requires immediate assistance [15]. Understanding the progression of the fall and accurately assessing the individual's post-fall state are essential for minimizing false alerts and ensuring timely, appropriate interventions.

This paper addresses these challenges by proposing a privacy-aware fall detection and alert management system using multimodal devices, including thermal cameras and wearable sensors. The goal is to achieve high detection accuracy while ensuring user privacy, alongside real-time alert management tailored to the condition of the individual post-fall. We present a comprehensive framework for fall detection that integrates thermal cameras and wearable devices, such as smartwatches and smartphones. Upon detecting a fall, the system automatically activates a voice assistant, which evaluates the user's level of consciousness and assesses the consequences of the fall through direct interaction with the individual. Based on the assessed risk level, the system can initiate an emergency call or send alerts to family members and healthcare professionals. Specifically, the key contributions of this study are as follows:

- A thorough comparison of thermal imaging and wearable-based devices for fall detection, incorporating various pre-processing techniques and Machine Learning (ML) models.

- The creation of a dataset, *UAL - Multimodal Fall Detection Dataset* (UAL-MFDD), which includes data from three sources: thermal images captured from three distinct perspectives, and gyroscope and accelerometer data collected from a smartwatch and a smartphone.

- The development of a voice assistant system that evaluates the user's condition post-fall, and either notifies relevant parties or contacts emergency services, depending on the severity of the situation.

- An explainable validation of the proposed neural network, using LIME [16], adapted to process sequences of images.

The remainder of the article is organized as follows. Section 2 analyzes some FD solutions and alert systems. Section 3 thoroughly describes the components of the proposed framework. Section 4 analyzes the performance of the framework and, finally, Section 5 highlights the main conclusions and future work.

2. Related works

2.1. Multimodal FD solutions

FD systems utilize wearable devices, vision-based sensors, and ambient sensors to detect falls [17]. Wearable devices, which are typically attached to the user, capture kinematic data using accelerometers and gyroscopes. Early systems relied on simple threshold-based techniques to trigger alerts [18], though these approaches often resulted in high false positive rates. To address this, ML [19] and Deep Learning (DL) algorithms [20] have been introduced. In this domain, several widely-cited datasets, including those from smartphones, smartwatches, IMU-based devices, and custom-built devices, have been instrumental in advancing research [21, 22, 23, 24, 25].

Ambient devices, such as passive infrared, acoustic, and infrared sensors, can detect falls with higher accuracy, although their effectiveness is contingent on environmental conditions and limited to areas where such devices are installed. Among these, sound sensors have demonstrated good performance in FD [26]. However, in most cases, ambient sensors are paired with cameras to improve fall detection accuracy [27, 28].

As previously mentioned, camera-based devices are restricted to the monitored area, but the use of DL techniques [29] enables detailed scene analysis, generally achieving the highest accuracy in fall detection [12]. However, visible-spectrum cameras raise significant privacy concerns, which hinders their deployment in real-world environments. Common architectures such as convolutional, recurrent, and vision transformers are used to achieve state-of-the-art performance in fall detection [30]. Similar to wearable-based solutions, several video datasets are frequently employed to benchmark these approaches [31, 32].

To further improve performance, some researchers propose integrating multiple sensor types, a solution known as "multimodal FD" [13]. Xefteris et al. [12] conducted a comprehensive review of existing multimodal systems, evaluating them in terms of accuracy, response time, and power consumption, all of which significantly impact real-world deployments. The review concluded that sensor-based and camera-based solutions offer high accuracy and low power consumption, while wearable-based solutions are notable for their low response time and non-intrusiveness. Table 1 summarizes the various sensor types used in multimodal FD systems.

In an effort to aid the evaluation of FD systems using multimodal devices, the authors of [13] introduced the UP-Fall dataset, which incorporates mul-

Type of device	Privacy	Comfortable	Ambient dependent	Accuracy	Computational cost
Wearable	High	No	No	Medium	Low
Ambience	Medium	Yes	Yes	High	Medium
Vision	Low	Yes	Yes	High	High

Table 1: A sensors characterization used in multimodal FD systems.

tiple wearable devices attached to the user's wrists, knees, waist, and ankles, generating inertial data. The dataset also includes two cameras positioned at a height of 1.82 meters and six infrared binary sensors at a height of 0.4 meters. Additionally, users wore an electroencephalograph on their heads. Seventeen participants took part in the experiment, performing different types of simulated falls and Activities of Daily Living (ADL). This dataset has become a benchmark for various multimodal FD approaches [33, 34, 35]. Similarly, the MHAD dataset [14] incorporates several accelerometers and multi-view cameras, as well as microphones, although it does not include fall scenarios, focusing solely on ADL recorded from 12 subjects.

In [33], a multimodal FD system similar to ours was proposed, utilizing a Convolutional Neural Network (CNN) and an Long-Short Term Memory (LSTM) for multimodal data processing, with promising results. However, their test dataset deviated from a leave-one-subject-out validation approach, including data from all users. Additionally, the visible-spectrum images used in their study are richer in information compared to infrared images, which can potentially lead to better results, albeit at the cost of raising privacy concerns.

2.2. Thermal-sensors based solutions

Fall recognition using thermal images has not been extensively explored in the literature, largely due to challenges such as noise from temperature fluctuations and the low resolution of thermal sensors. Despite these obstacles, a few studies have utilized thermal cameras for fall detection. In [36], Optical Flow (OF) and Support Vector Machine (SVM) algorithms were employed to classify the state of the person. However, the results showed limited improvement, with an Area Under Curve (AUC) of only 64% in the Receiver Operating Characteristic (ROC) curve. More recent efforts have focused on DL techniques to autonomously extract relevant information from thermal images. In [37], a CNN with just two convolutional layers was proposed to recognize both ADL and fall activities. Using high-resolution thermal images, the system achieved an accuracy of 87%. The same high-resolution images were used in [38], where a Convolutional Long-Short Term Memory (ConvLSTM) auto-encoder was introduced for fall detection, yielding an accuracy of 83% through the inclusion of recurrent layers.

In contrast, [39] proposed an LSTM Neural Network (NN) that does not directly process raw thermal images, which were kept at a low resolution (32×24 pixels) to enhance privacy. In this approach, feature extraction was performed using OF, similar to the method applied in [36].

Notice that all reviewed works deal with the single-occupancy FD problem. However, there exist solutions that face the multi-occupancy problem. In [11], a scheme is proposed to detect people in a thermal image, generating single-occupancy images that are classified independently.

2.3. Main fall alert systems

While FD using multimodal sensors and cameras has been extensively researched, the processing and management of fall-related alerts remains an ongoing challenge.

In [40], a system is proposed that requires user interaction. When a fall is detected, the system prompts the user with a question. If the user does not respond, an Short Message Service (SMS) message is sent to the family, followed by an automatic call. The person receiving the call is responsible for assessing whether a fall has occurred and, if necessary, contacting emergency services.

The systems described in [41, 39] implement similar alarm functionalities using mobile alerts, SMS notifications, and automatic calls. In these cases, the user must confirm their well-being within the application. If no response is received, an automatic call is triggered and an SMS is sent. However, the key limitation of these systems is the requirement for the user to have a smartphone available at the time of the fall.

Some solutions address this limitation by removing the need for smartphone interaction. For instance, in [42], when a fall is detected, a Multimedia Messaging Service (MMS) is sent that includes a video clip of the fall. Similarly, in [43], the alert is sent via SMS. While these systems provide immediate notifications, they are prone to generating false positives (false alarms).

To reduce the need for user confirmation and decrease false alarms, email-based alert systems are proposed in [44, 45]. Upon detecting a fall, a sequence of images is sent via email to a designated caregiver, who then determines whether the fall is genuine or a false positive. Although this method incorporates video footage to assist decision-making, emails generally have slower delivery times compared to messaging applications like WhatsApp or Telegram, making them less ideal for real-time alerting.

3. Proposal

We present a FD framework capable of understanding and identifying the distinct stages of a fall (Falling, Fallen on the floor, and Recovering) using multimodal sensors. Once a fall is detected, an alarm management system evaluates the user's condition by prompting them with a series of questions. The user's responses, or the absence thereof, are recorded and transmitted to family members, healthcare professionals, or caregivers via an instant messaging application. Additionally, upon detecting a fall and based on user input, a smart assistant, supported by Alexa and a Large-Language Model (LLM) such as ChatGPT[1] or Ollama[2], provides the user with guidance on how to respond.

In Section 3.1, we describe the IoT system that collects data from the environment. Section 3.2 discusses the data preprocessing techniques, while Section 3.3 outlines the ML models used for fall identification. Finally, Section 3.4 details the alarm management system.

3.1. IoT system

The IoT system encompasses all devices and services integrated into the environment. It consists of two main components: sensors and devices that collect data from the user's interactions within the environment, and communication devices that facilitate interaction between the user and the FD system.

All devices are connected to the network via WiFi. A daemon running on a local computer manages the reception, merging, and storage of sensor data, while also executing the FD algorithms. Figure 1 illustrates the environment and the physical components of the IoT system.

[1] `https://chatgpt.com`

[2] `https://ollama.com`

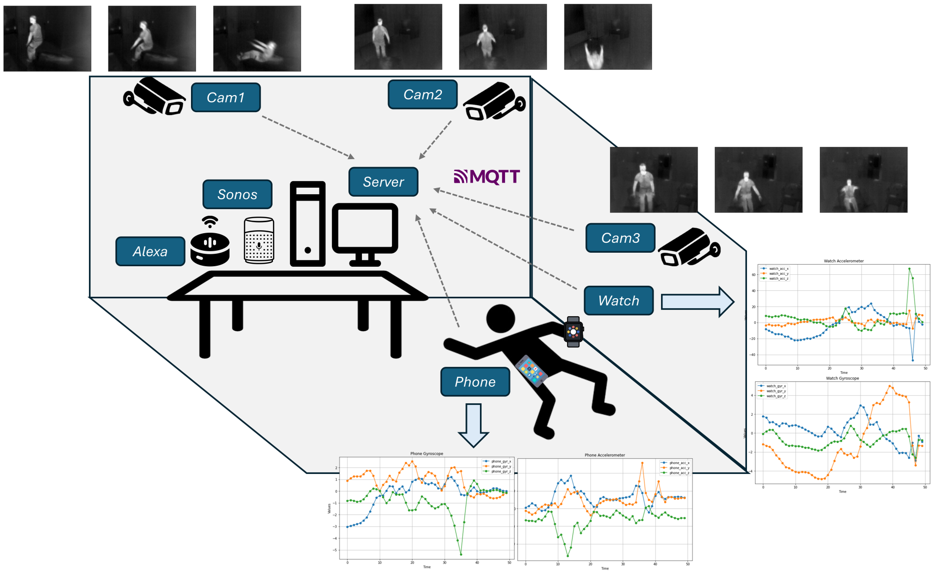


Figure 1: Devices of the IoT system. Three low-resolution thermal cameras simultaneously capture the same scene. Additionally, to determine its orientation and angular velocity, the user wears two gyroscopes (one in a wristwatch in the non-dominant wrist and other in a smartphone in a pocket of the trousers) and, to measures acceleration forces, two accelerometers (same configuration). All these signals (video, and inertial data) are sent to a local computer, which processes them and determines the appropriate action to take when a fall is detected.

3.1.1. Thermal cameras

In our study, three thermal cameras (FLIR Lepton 3.5) are employed, each mounted on a Raspberry Pi attached to the wall. All Raspberry Pi devices are connected to an Message Queuing Telemetry Transport (MQTT) broker hosted on the central computer. To synchronize image capture, a daemon on the computer transmits three messages per second to a specific topic in the broker. Upon receiving the message, each Raspberry Pi captures an image and sends it to the MQTT broker. The cameras are arranged in a triangular configuration, positioned at a height of 1.9 meters, ensuring multiple viewpoints of the user for comprehensive coverage and optimal perspective.

3.1.2. Wearable devices

In addition to thermal imaging, our approach integrates data from an Inertial Measurement Unit (IMU) sensor, capturing both accelerometer and gyroscope signals. The accelerometer records acceleration along the X, Y, and Z axes, while the gyroscope measures rotational motion on the same axes. These signals are collected using a smartwatch and a smartphone (refer to Figure 1).

The IMU data is sampled at a rate of 50 samples per second, utilizing an Android application capable of reading motion sensors[3]. To reduce battery consumption, the data is temporarily buffered in memory and transmitted to the local computer with a cadence of 2 seconds.

3.1.3. Speaker

Upon detecting a fall, communication with the user is established via a Sonos speaker, which is connected to the local network through WiFi. The speaker is controlled using an API written in Python called SoCo[4], which facilitates audio playback. To generate audio output, the Google Text-to-Speech (gTTs) library[5] converts a written message into an MP3 file, which is then played through the Sonos speaker using SoCo. In particular, the speaker activates an Alexa skill that prompts a series of questions to the user following a fall, helping to assess the aftermath and determine the necessary response.

[3] https://developer.android.com/develop/sensors-and-location/sensors/sensors_motion

[4] https://github.com/SoCo/SoCo

[5] https://gtts.readthedocs.io

3.1.4. Voice assistant

To engage with the user and ask follow-up questions (as outlined in the previous section), a custom Alexa skill is triggered using the Sonos speaker, which issues the command, “Alexa, open the fall interview.” For enhanced availability, security and privacy, the back-end of this Alexa skill is hosted locally on the computer, eliminating the need for an Internet connection. The connection between Alexa and the local computer is established via an *NGROK* tunnel[6]. The API supporting this interaction was developed using *Flask*[7].

3.2. Data conditioning

The data produced by the thermal cameras and motion detection devices must be adapted for use with ML algorithms. This section outlines the data preprocessing techniques applied in our approach.

3.2.1. Thermal images

The thermal cameras used generate images with a resolution of 160×120 pixels, with a depth of 11 bits per pixel, at a rate of 9 frames per second. However, for numerical efficiency, NNs require floating-point numerical representations that operate within the interval $[0, 1]$. Thus, the first preprocessing step is image normalization.

Since falls can occur in different regions of the scene, the model needs to recognize falls regardless of the user’s location. To achieve this, the position of the user in the scene is normalized, ensuring that the NN does not form a bias based on where the person is located. Without this step, the model could learn to associate the location of the person with fall classification, potentially leading to misclassifications. For example, if 80% of falls occur when the user is on the right side of the image, the model may become biased toward detecting falls in that region, resulting in incorrect classifications for falls on the left. To prevent such bias, we segmented, cropped, and centered the person in the image, setting the background to black. This processed image is then used as input for the fall detection system. For segmentation, we used the YOLOv5x model[8], a CNN that detects and segments objects in RGB images, providing a mask rather than just a bounding box. Despite working

[6] `https://ngrok.com/product/secure-tunnels`

[7] `https://flask.palletsprojects.com`

[8] `https://github.com/ultralytics/yolov5`

on gray-scale thermal images, YOLOv5x performed efficiently, running at 10 FPS on a typical computer with a NVIDIA GeForce GTX 1650 GPU.

Alternatively, we explored a different approach for segmenting the user based on background subtraction using OF. Comparing two consecutive images, the Farnebäck algorithm [46] calculates a field of motion vectors. By analyzing the displacement of objects, pixels with motion vectors below a certain threshold are classified as background. The Farnebäck estimator, even when running on a CPU, is significantly faster than YOLOv5x, achieving around 100 FPS on a typical Intel Core i7-10750H computer. Although less accurate than YOLOv5x, this approach offers the advantage of extracting motion fields as additional input for the fall detection system. In our implementation, the motion information is represented as 160 × 120 RGB pixels, where the luminance reflects the magnitude of the motion vector and the chroma encodes its direction.

To summarize, the preprocessing of thermal images supports three different fall detection configurations (see Figure 2) that are (1) using the original thermal images without further preprocessing, (2) segmenting the person by cropping and centering the image, with the background set to black, and (3) combining segmented images with OF data, using the OF to represent motion vectors.

3.2.2. IMU

The IMU data have been pre-processed using two different methods:

1. For ML algorithms, features such as the mean, maximum, minimum, range, and standard deviation were extracted from the accelerometer and gyroscope coordinates, following previous works [47, 48]. To refine these features, we applied SHAP (SHapley Additive exPlanations) [49] and Random Forest Importance [50] to determine the most important features. As a result, 11 features from the smartwatch and 12 from the smartphone were removed. For DL algorithms, which can process high-dimensional data directly, the use of aggregated features is not required.
2. A one-second window (50 samples) was used to capture the accelerometry signals. We then applied Min-Max normalization, where the maximum and minimum values for each feature were derived from the training dataset, and these values were subsequently used to normalize the features in the validation and test datasets.

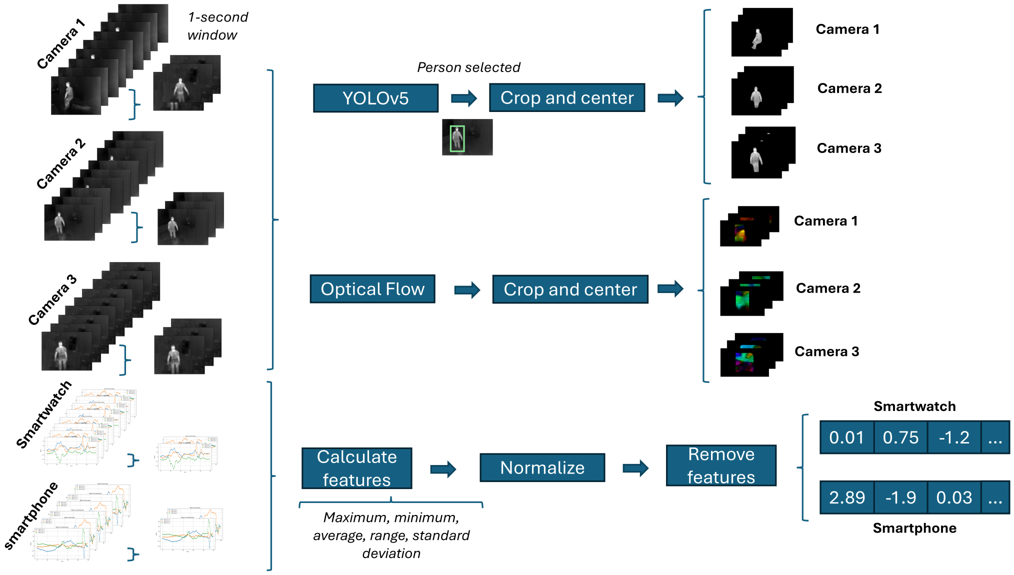


Figure 2: Data conditioning. When segmenting the person appearing in the scene, two different alternatives have been used: (1) an YOLOv5x model, and (2) the Farnebäck OF estimator to determine which pixels belong to the background (non-moving pixels) and which pixels belong to the foreground (pixels presumably belonging to the subject appearing in the scene). Regardless of the segmentation technique used, the OF has been considered to improve fall detection (see the Table 5). Finally, the inertial data are also pre-processed before being delivered to the FD.

In addition to normalization, feature selection, and outlier removal, data augmentation is an important step. By generating synthetic data that introduces variations in the original dataset, model performance can often be improved. In our case, we employed the SMOTE (Synthetic Minority Oversampling Technique) method [51] to enhance the dataset.

3.3. Machine learning and neural network models

Various ML and DL methods have been proposed in the literature to detect falls in multimodal setups. However, no single algorithm or architecture stands out as the best across all scenarios. In this work, multiple models are built and tested to determine the most suitable model for each device and preprocessing method.

3.3.1. Images models

Batches of three images (captured in a one-second window) are processed from three different perspectives. The input captures both spatial information (e.g., the person, the background, and location) and temporal relations (e.g., the movement of the person across images).

One model tested is the CNN, which is particularly adept at capturing spatial relationships in images, thanks to its convolutional layers that can automatically extract features. When processing images from multiple perspectives, we found it more effective to handle each image separately in different convolutional branches (see Figure 3) [52]. Thus, in one configuration, the NN input consists of three separate image batches.

Since the input images also contain temporal information, we process image sequences (three images) to detect falls. Recurrent NNs, particularly those incorporating LSTM units [53], are well-suited for temporal data processing. However, these networks cannot directly handle raw images, so we propose combining them with convolutional layers that first extract spatial features.

Additionally, OF information can be incorporated into these systems. Each OF field is processed independently using the same convolutional architecture applied to the thermal images. Finally, before inputting the data into the ML model (CNN or ConvLSTM), the feature maps generated by the (block of) convolutional layers applied to OF fields are multiplied by the feature maps from the convolutional layers applied to the thermal images. This process merges the data from different branches (thermal images, OF,

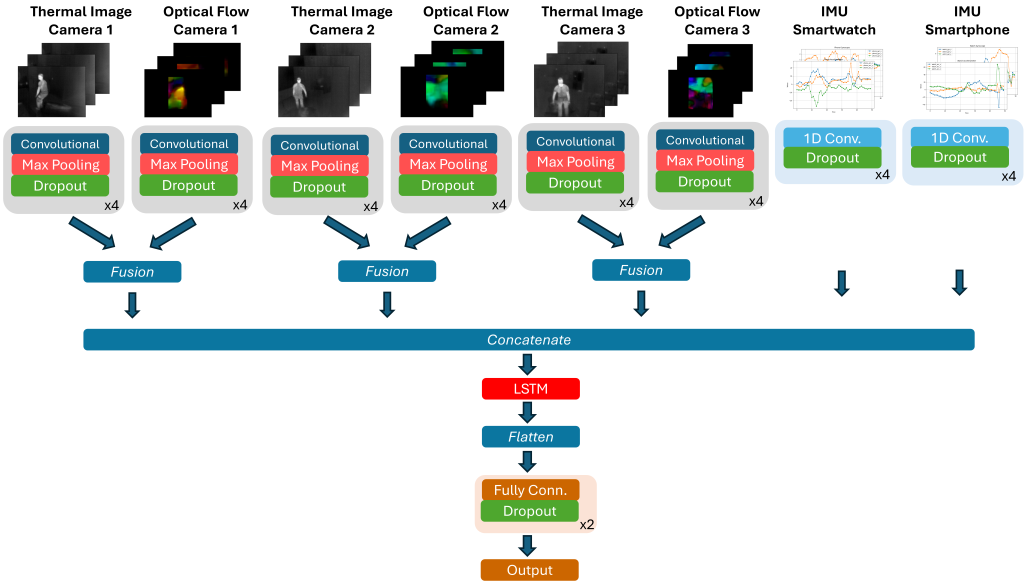


Figure 3: Architecture of the proposed model. The input data (the thermal images, the OF fields, and the IMU signals) are depicted at the top, and the output (the classification of the scene) is at the bottom of the figure. This is the complete model that shows all the possible operations/configurations that we have implemented (notice that some modules have been described in a compacted form). The Table 5 shows the performance of some ablation configurations of this model.

and IMU signals) in a “Fusion” module (see Figure 3), which is then passed to the classifier’s final layers.

3.3.2. IMU models

For processing IMU data, we propose using a CNN with 1D convolutions, rather than 2D, to independently extract patterns from each sensor’s coordinates, searching for common patterns across different axes. Additionally, recurrent NNs, particularly those with LSTM units, are a good fit for this type of temporal data, though fully connected layers can also be used at the cost of higher memory consumption.

We also propose combining the IMU data with the architectures used for processing thermal images (and OF). The IMU data serve as an additional input, and the extracted information is fused with the network branches processing the images.

All of the proposed architectures are illustrated in Figure 3.

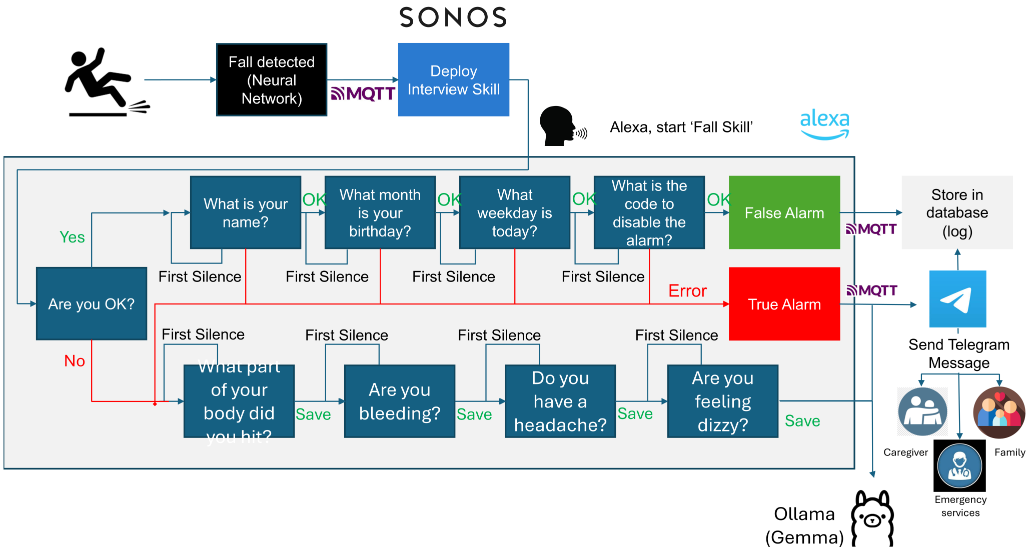


Figure 4: Alert System. After the detection of a fall, the system establishes a conversation with the users to determine the severity of the fall. If the user is OK and can recover by him/herself, the alarm system aborts sending an alarm message. Otherwise, the alarm system sends a message to the person in charge.

3.4. Alert system

The proposed system not only focuses on detecting the stages (Normal, Falling, Fallen, or Recovering) using signals from thermal cameras and wearable devices but also includes an alert mechanism that assesses the user's condition if abnormal behavior is detected. When a sequence of time windows (containing merged signals) is classified as abnormal (Falling, Fallen, or Recovering), the alert system is automatically triggered. It begins by assessing the user's state through questions delivered via an Alexa skill. If the user is conscious and has recovered, no alarm is sent. However, if there is no response or the responses are incoherent, an alarm is activated. In such cases, the system offers the user the option to interact with an LLM, which provides valuable advice on managing or recovering from the fall. A data flow diagram of the alert system is shown in Figure 4.

The main component of the alert system is the Alexa skill, with the backend distributed between Amazon's infrastructure and a local computer. The interaction between Alexa and the user is split into two parts: (1) an initial interview with predefined static questions to assess the user's consciousness, and (2) a conversation with an LLM-based language model, such as GPT

(Generative Pre-trained Transformer) or the Ollama framework.

When a fall is detected over several consecutive time windows, the Sonos speaker triggers the Alexa skill by saying, “Alexa, open the fall interview”. Alexa then begins with: “Hello, I have detected that you have fallen. Are you okay?” If the user responds affirmatively, Alexa proceeds with questions to verify their consciousness, such as: “What is the name of the person who lives in this house?” or “What month is your birthday?”

If the user correctly answers these questions, the alarm is not raised. Otherwise, a help alert is sent via Telegram, and a new interview is conducted with questions aimed at assessing the severity of the fall. These questions include: “What part of your body did you hit?” “Are you bleeding?” “Do you have a headache?” “Are you feeling dizzy?”

At this stage, the user can request advice from the LLM. The model is designed to provide concise responses (less than 50 words). To tailor the advice, the system integrates context from the user’s responses to the Alexa interview. For example, if the user says, “I have fallen and hit my head on the floor” or “I see blood on my head but have no headache” the LLM incorporates these details into its response. Furthermore, the model is instructed to behave like a healthcare professional and includes recommendations extracted from literature and medical guides [54, 55], such as: “Remain calm and assess the situation to determine if there are any serious injuries.”

4. Experimental analysis

In this section, the experimental setup is described first, providing the reader with a complete understanding of the evaluation phase. Then, an ablation study analyzing the impact of each component in the system in the final performance is included. Finally, an analysis of the fall detection results is carried out.

4.1. Setup

This section defines the experimental setup in order to provide the details of the experiments and enable their reproducibility. First, the description of the dataset is assessed, as well as the time-window defined to process the data in real time. Then, the dataset splitting technique and the metrics used to measure the results are described. After this, the hyperparameters of the training phase has been indicated. Finally, the evaluation of the LLM models is carried out.

4.1.1. Dataset description

One of the main contributions of this work is the UAL-MFDD dataset, recorded at the Smart Home of the University of Almería. It contains data from 17 actors (12 men and 5 women), each performing a sequence of movements, including 20 simulated falls from three different perspectives. The falls consist of five different types: fall to the knees, front fall, left lateral fall, right lateral fall, and backward fall. In total, 340 falls were recorded, with a combined recording time of approximately 12.2 hours.

4.1.2. Latency

In order to perform the classification of the stages of the falls, a time window is necessary for analyzing the input data. Larger time windows generally provide more reliable results but increase the system's latency. All the results in this section are based on a 1-second time window, which offers a good compromise between reliability and latency.

4.1.3. Training configuration and validation metrics

The validation of the proposed system was conducted using a leave-one-group-out (LOGO) approach [56]. This involves creating separate groups of actors for training and validation. The dataset was split into three subsets: training, validation, and test. The training and validation datasets contain 13 actors, while the test dataset includes 4 actors. This approach allows for evaluating the system's generalization to unseen users. For the training and validation datasets, the data was shuffled, with 80% used for training and 20% for validation.

The dataset splits exhibit class imbalance, with the Normal state being overrepresented and the Fall state underrepresented. Table 2 details the number and percentage of samples for each class across the different dataset parts.

To assess the performance of various models, the F1-Score was selected as the primary evaluation metric, as it computes the harmonic mean between precision and recall, making it well-suited for imbalanced datasets [57]. The F1-Score was calculated using the test dataset, which consists of actors not included in the training or validation sets.

4.1.4. Training parameters

The DL models were trained with a maximum of 500 epochs, applying early stopping after 50 epochs if no improvement was detected. The batch

Dataset	Class				Total samples
	Normal	Falling	Fallen	Recovering	
Training	50.11	2.30	40.46	7.13	26804
Validation	50.72	2.46	39.79	7.03	6700
Test	52.10	1.63	40.29	5.98	10402
Duration (s)	-	[1, 2]	[15, 20]	[5, 8]	43906

Table 2: Class distribution in the training, validation and testing datasets. The minimum and maximum duration of each state has been also provided (except for the Normal class). There is a fall every $[20, 30]$ seconds.

size was set to 64, enabling faster training while ensuring each batch contained samples from all classes. The Adam optimizer was used to adjust the weights, with default parameters as specified in the TensorFlow framework.

4.1.5. Evaluation of LLM models

An LLM has been integrated into the alert system to provide assistance after detecting a fall. One of the primary challenges faced during the implementation of this feature was the model's response latency. To address this issue, we evaluated the average latency of the models available in the Ollama tool by sending 10 requests to a virtual machine configured with 16 processors, 32 GB of RAM, and 160 GB of disk space. The prompt used for each request was, "I have fallen and hurt myself, can you give me some advice?" The evaluation was conducted in two sets: in the first, the default configuration of each model was used to generate the response; in the second, a customized configuration was applied, where the model was instructed to act as a fall care assistant. This customized setup utilized 16 threads, limited responses to 150 words, and retained a history of the last 10 questions and answers. Table 3 summarizes the size of the models, their latency for both default and customized configurations, and the number of characters generated in their responses.

The results show that the Gemma model achieved the lowest latency (2.82 seconds) in the customized configuration, while still generating responses with fewer than 150 characters. Gemma [58] is a family of lightweight, state-of-the-art models developed by Google, based on the same research and technology as the Gemini models [59].

Model (parameters)	Size GB	Latency (s) default/custom	Num. characters default/custom
Llama-3 (70B)	40	Doesn't respond	- / -
Llama-2 (70B)	39	Doesn't respond	- / -
Llama-2 (13B)	7.3	119.10 / 16.24	1467 / 156
Solar (10.7B)	6.1	116.16 / 23.02	1783 / 360
Gemma (7B)	4.8	74.20 / 6.57	1483 / 116
Llama-3 (8B)	4.7	88.10 / 17.56	1970 / 432
Mistral (7B)	4.1	65.47 / 9.77	1425 / 259
Neural Chat (7B)	4.1	75.11 / 9.37	1694 / 211
Starling (7B)	4.1	98.64 / 41.94	2203 / 887
Code Llama (7B)	3.8	77.77 / 23.15	1491 / 758
Llama-2 (7B)	3.8	31.39 / 11.57	685 / 343
LLaVA (7B)	4.5	52.56 / 22.10	1188 / 778
Orca Mini (3B)	1.9	20.40 / 9.08	830 / 396
Phi-2 (2.7B)	1.7	48.47 / 4.15	2215 / 259
Dolphin Phi (2.7B)	1.6	21.52 / 9.21	1079 / 454
Gemma (2B)	1.4	25.68 / **2.82**	1460 / 116

Table 3: Comparison of delay between customized configuration and default configuration in different Ollama models.

4.2. Ablation study

This section explores some key questions: Are all the devices necessary for fall detection? Which combination of devices is the most effective? While CNNs excel with image data, LSTM networks should also perform well with sequences. So, which neural network architecture is best suited for the FD problem? What image and IMU-signal conditioning techniques are optimal? To answer these questions and evaluate the system's performance, an ablation study was conducted to quantify the impact of various model components, information sources, and techniques.

4.2.1. Impact of IMU signals

In this study, raw IMU signals were directly input into the model without any preprocessing. The ablation results, shown in Table 4, reveal that when LSTM is not utilized, using IMU signals from both the smartwatch and smartphone yields better results. However, when the LSTM module is included, using only the smartphone data performs better. This may be because, during a fall, the movement of the trunk (monitored by the smartphone) exhibits less variation than the movement of the arms (monitored by the smartwatch).

Model	Device	Test F1-Score
CNN	Smartwatch	0.846
	Smartphone	0.858
	Both	**0.864**
LSTM	Smartwatch	0.655
	Smartphone	**0.780**
	Both	0.609

Table 4: Impact of the IMU signals on the detection performance.

4.2.2. Impact of the visual inputs

The visual inputs consist of thermal images and OF fields. The following configurations were evaluated (as shown in Figure 3):

- **Raw**: The raw (unprocessed) images from individual cameras (Raw/C1, Raw/C2, Raw/C3), combinations of two cameras (Raw/C1C2, Raw/C1C3, Raw/C2C3), and all three cameras (Raw/C1C2C3).
- **Only-OF**: Only OF fields are used as input.
- **Raw+OF**: Both thermal images and motion detected through OF fields are considered.
- **PCC (Person Cropped and Centered)**: Similar to Raw, but the person is cropped and centered against a black background.
- **PCC+OF**: The same as PCC, but with OF fields included.

The results, shown in Table 5, demonstrate:

1. Multiple perspectives improve fall classification, but the gain is small (only a 0.2 increase in the F1-Score).
2. The combination of accelerometer and gyroscope data yields results similar to using only cameras.
3. LSTM is effective when processing data from a single camera (C2).
4. Incorporating OF boosts performance, producing the best results in five cases (CNN/C3, CNN/C1C2, CNN/C2C3, CNN/C1C2C3, and CNN+LSTM/IMU+C1C2C3). However, OF alone is insufficient.
5. PCC is not critical (only superior in the PCC+OF/C1C3 case), suggesting that system inference time can be reduced by not using YOLO.

		F1-Score							
Model	Pre-proces.	C1	C2	C3	C1C2	C1C3	C2C3	C1C2C3	IMU + C1C2C3
CNN	Raw	**0.923**	0.849	0.929	0.925	0.927	0.930	0.921	0.914
	Only-OF	0.790	0.806	0.837	0.868	0.870	0.885	0.877	0.896
	Raw+OF	0.904	0.864	**0.930**	**0.932**	0.921	**0.932**	**0.950**	0.924
	PCC	0.891	0.851	0.883	0.918	0.930	0.889	0.904	0.925
	PCC+OF	0.897	0.861	0.899	0.923	**0.936**	0.923	0.922	0.895
CNN + LSTM	Raw	0.896	**0.873**	0.918	0.930	0.926	0.920	0.921	0.917
	Only-OF	0.821	0.804	0.830	0.846	0.231	0.865	0.832	0.879
	Raw+OF	0.916	0.859	0.904	0.799	0.910	0.824	0.813	**0.947**
	PCC	0.885	0.839	0.900	0.921	0.920	0.901	0.938	0.921
	PCC+OF	0.824	**0.873**	0.877	0.860	0.678	0.858	0.739	0.916

Table 5: Performance of the different configurations of pre-processing techniques cameras plus IMU-signals.

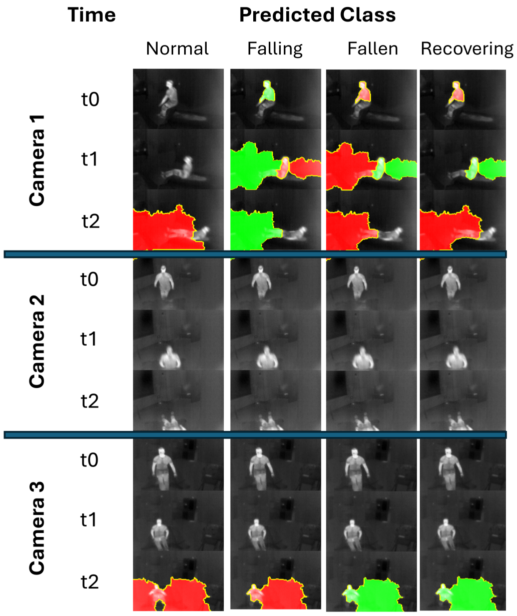


Figure 5: Explainability of the inference of the CNN blocks when a fall is recorded by the cameras. `t0`, `t1` and `t2` represent three diferent instant of time, being `t0` the first (oldest) one. The pixels that recognise the correct estate at each column are coloured in green and the pixels that recognize an incorrect estate are in red.

ML Algorithm	Device	SMOTE	F1-Score
kNN	Smartwatch	No	0.896
	Smartphone	No	0.871
	Both	No	**0.900**
RandomForest	Smartwatch	No	0.925
	Smartphone	No	0.875
	Both	No	**0.931**
XGBoosting	Smartwatch	No	0.926
	Smartphone	Yes	0.889
	Both	No	**0.933**
GradientBoosting	Smartwatch	No	0.904
	Smartphone	No	0.881
	Both	Yes	**0.916**

Table 6: Performance of IMU-based models.

A more detailed analysis of the visual inputs can be conducted using model explainability techniques [16], which shows how the inference is carried out. As seen in Figure 5 (the code for reproducing this experiment is available at GitHub[9]), the second camera (C2) provides the least information about the fall. Among cameras 1 (C1) and 3 (C3), C1 captures the fall more clearly due to its lateral perspective, whereas C3 is frontal. Additionally, the state recorded at `t0` is key for recognizing the class Falling using C1.

4.2.3. Impact of the LSTM network

The model depicted in Figure 3 integrates an LSTM network to capture temporal correlations in the input signals (cameras and IMU). However, as shown in Table 5, the LSTM network has a marginal effect on overall performance. Despite this, it demonstrates the ability to effectively combine input signals, delivering near-optimal performance in the Raw+OF configuration.

4.3. IMU-based FD using ML algorithms

Some traditional ML algorithms can effectively classify data with low-dimensional inputs, such as those generated by wearable devices. This makes it feasible to apply techniques like k-Nearest Neighbors (kNN) [60], Random Forest [61], XGBoost [62], and Gradient Boosting [63] to detect falls using IMU data. The results of these classifiers are presented in Table 6. As shown,

[9] `https://github.com/marcoslupion/lime-multi-input.git` where the LIME library has been modified to work with several input images at the same time.

Class	Precission	Recall	F1-Score
Normal	0.997	0.972	0.985
Falling	0.401	0.858	0.546
Fallen	0.980	0.919	0.949
Recovering	0.638	0.841	0.725

Table 7: Performance of the classification.

Class	Precission	Recall	F1-Score
Normal	0.997	0.972	0.985
Fall	0.971	0.997	0.984

Table 8: Performance classification when only to different classes are considered.

the performance of these ML models is slightly lower compared to the results obtained using visual information (as in Table 5). Nonetheless, the results confirm that wearable devices like smartwatches and smartphones are highly suitable for fall detection.

4.4. Class analysis

Table 7 presents the precision, recall, and F1-Score metrics for the different fall stages (as detailed in Table 2). It can be observed that the most underrepresented classes, such as Falling and Recovering, are more challenging to identify. This is because these transition classes are shorter in duration (between 1 and 2 seconds) compared to the Normal and Fallen states, which last at least 5 seconds. Notably, the transition classes (Falling and Recovering) have recall values of 0.858 and 0.841, respectively. This suggests that the model can recognize these states in around 85% of cases, but there are a significant number of false positives, as indicated by the lower precision values. In the literature, false positives are a well-known issue in FD systems, and in some cases, they hinder the adoption of such systems in real-world environments.

To address this issue, we combined the Falling, Fallen, and Recovering classes into a new class labeled “Fall”. The updated results are shown in Table 8. As seen in the table, the recall for the positive class (Fall) is now 0.997, indicating that the system can recognize almost all instances where the user is either falling, on the ground, or recovering. This ensures a robust detection of these states, enabling the system to trigger alerts without missing critical falls.

Furthermore, the system’s performance can be maximized (i.e., detect all falls without false positives) by only generating an alarm when a sequence of N consecutive 2-second fall detections is recognized. In our experiments, we found that setting $N > 2$ ensures 100% accuracy in fall detection.

To clarify, the test dataset included 80 fall sequences, each lasting between 20 and 30 seconds. In these sequences, the Falling state lasted between 1 and 2 seconds, the Fallen state approximately between 15 and 20 seconds, and the Recovering state between 5 and 8 seconds. Using binary classification, the system was able to identify all falls, achieving 100% accuracy. There were also 81 normal state sequences, all of which were correctly classified. In only 4 instances, fall states were detected in more than 10% of the time during normal sequences, but these predictions occurred at non-consecutive intervals, thus preventing the triggering of false positive alarms.

5. Conclusions and future work

This work analyzed the performance of an smart fall detection and alarm generation system capable of interacting with the user. Multiple sources of visual information (infrared cameras) and motion data (accelerometers and gyroscopes) were utilized. To process this information, a deep learning model was proposed to analyze these sources directly, along with data provided by an optical flow estimator. The results of the system's evaluation using a new public dataset—comprising 17 actors and more than 340 falls recorded at the University of Almería—showed that:

1. The use of infrared information is effective for fall detection, yielding results comparable to those obtained using visible-spectrum data, preserving the privacy as much as possible.
2. The use of motion data generated by wearable devices also provides satisfactory results, though slightly lower in performance compared to visual information.
3. Both sources of information (infrared and inertial) can be combined using deep learning techniques to create more robust systems.
4. The incorporation of the optical flow as an additional input enhances performance.
5. The fall detection system can be integrated with an alarm system using common IoT devices that the user is likely to already have at home.

In future work, the focus will be on fall prediction, which will require recording datasets in controlled environments. Additionally, all software and models will be packaged and deployed as a cloud solution, making them accessible to researchers.

CRediT authorship contribution statement.

Declaration of competing interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability. Data will be made available on request.

Acknowledgments.

References

[1] World Health Organization, Falls, https://www.who.int/news-room/fact-sheets/detail/falls, online; accessed 07 march 2022 (April 21 2021).

[2] M. Terroso, N. Rosa, A. Torres Marques, R. Simoes, Physical consequences of falls in the elderly: a literature review from 1995 to 2010, European Review of Aging and Physical Activity 11 (1) (2014) 51–59. doi:10.1007/s11556-013-0134-8.

[3] A. F. Ambrose, G. Paul, J. M. Hausdorff, Risk factors for falls among older adults: A review of the literature, Maturitas 75 (1) (2013) 51–61. doi:10.1016/j.maturitas.2013.02.009.

[4] United Nations, World population prospects 2022, https://population.un.org/wpp/Download/SpecialAggregates/EconomicTrading/, online; accessed 21 october 2022 (2022).

[5] L. N. Saftari, O.-S. Kwon, Ageing vision and falls: a review, Journal of physiological anthropology 37 (1) (2018) 1–14. doi:10.1186/s40101-018-0170-1.

[6] K. A. Hartholt, E. F. van Beeck, S. Polinder, N. van der Velde, E. M. van Lieshout, M. J. Panneman, T. J. van der Cammen, P. Patka, Societal consequences of falls in the older population: injuries, healthcare costs, and long-term reduced quality of life, Journal of Trauma and Acute Care Surgery 71 (3) (2011) 748–753. doi:10.1097/TA.0b013e3181f6f5e5.

[7] M. S. Khan, M. Yu, P. Feng, L. Wang, J. Chambers, An unsupervised acoustic fall detection system using source separation for sound interference suppression, Signal processing 110 (2015) 199–210. doi:10.1016/j.sigpro.2014.08.021.

[8] K. Ozcan, S. Velipasalar, Wearable camera-and accelerometer-based fall detection on portable devices, IEEE Embedded Systems Letters 8 (1) (2015) 6–9. doi:10.1109/LES.2015.2487241.

[9] M. Bernstein, "low-tech" personal emergency response systems reduce costs and improve outcomes, Managed care quarterly 8 (1) (2000) 38–43.

[10] K. K. Peetoom, M. A. Lexis, M. Joore, C. D. Dirksen, L. P. De Witte, Literature review on monitoring technologies and their outcomes in independently living elderly people, Disability and Rehabilitation: Assistive Technology 10 (4) (2015) 271–294. doi:10.3109/17483107.2014.961179.

[11] C. Zhong, W. W. Y. Ng, S. Zhang, C. D. Nugent, C. Shewell, J. Medina-Quero, Multi-occupancy fall detection using non-invasive thermal vision sensor, IEEE Sensors Journal 21 (4) (2021) 5377–5388. doi:10.1109/JSEN.2020.3032728.

[12] V.-R. Xefteris, A. Tsanousa, G. Meditskos, S. Vrochidis, I. Kompatsiaris, Performance, challenges, and limitations in multimodal fall detection systems: A review, IEEE Sensors Journal 21 (17) (2021) 18398–18409. doi:10.1109/JSEN.2021.3090454.

[13] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez, C. Peñafort-Asturiano, Up-fall detection dataset: A multimodal approach, Sensors 19 (9) (2019) 1988. doi:10.3390/s19091988.

[14] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, R. Bajcsy, Berkeley mhad: A comprehensive multimodal human action database, in: 2013 IEEE workshop on applications of computer vision (WACV), IEEE, 2013, pp. 53–60. doi:10.1109/WACV.2013.6474999.

[15] M. Lupión, V. González-Ruiz, J. F. Sanjuan, J. Medina-Quero, P. M. Ortigosa, Detection of unconsciousness in falls using thermal vision sensors, in: K. Daimi, A. Al Sadoon (Eds.), Proceedings of the

ICR'22 International Conference on Innovations in Computing Research, Springer International Publishing, Cham, 2022, pp. 3–12. doi:10.1007/978-3-031-14054-9_1.

[16] M. T. Ribeiro, S. Singh, C. Guestrin, "why should i trust you?": Explaining the predictions of any classifier (2016). doi:10.48550/arXiv.1602.04938.

[17] T. Xu, Y. Zhou, J. Zhu, New advances and challenges of fall detection systems: A survey, Applied Sciences 8 (3) (2018) 418. doi:10.3390/app8030418.

[18] H. W. Guo, Y. T. Hsieh, Y. S. Huang, J. C. Chien, K. Haraikawa, J. S. Shieh, A threshold-based algorithm of fall detection using a wearable device with tri-axial accelerometer and gyroscope, in: 2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), IEEE, 2015, pp. 54–57. doi:10.1109/ICIIBMS.2015.7439470.

[19] P. Tsinganos, A. Skodras, On the comparison of wearable sensor data fusion to a single sensor machine learning technique in fall detection, Sensors 18 (2) (2018) 592. doi:10.3390/s18020592.

[20] A. Jefiza, E. Pramunanto, H. Boedinoegroho, M. H. Purnomo, Fall detection based on accelerometer and gyroscope using back propagation, in: 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), IEEE, 2017, pp. 1–6. doi:10.11591/eecsi.v4.1079.

[21] G. Vavoulas, M. Pediaditis, C. Chatzaki, E. G. Spanakis, M. Tsiknakis, The mobifall dataset: Fall detection and classification with a smartphone, International Journal of Monitoring and Surveillance Technologies Research (IJMSTR) 2 (1) (2014) 44–56. doi:10.4018/ijmstr.2014010103.

[22] T. Vilarinho, B. Farshchian, D. G. Bajer, O. H. Dahl, I. Egge, S. S. Hegdal, A. Lønes, J. N. Slettevold, S. M. Weggersen, A combined smartphone and smartwatch fall detection system, in: 2015 IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing, IEEE, 2015, pp. 1443–1448. doi:10.1109/CIT/IUCC/DASC/PICOM.2015.216.

[23] E. Casilari, J. A. Santoyo-Ramón, J. M. Cano-García, Umafall: A multisensor dataset for the research on automatic fall detection, Procedia Computer Science 110 (2017) 32–39. doi:10.1016/j.procs.2017.06.110.

[24] C. Medrano, R. Igual, I. Plaza, M. Castro, Detecting falls as novelties in acceleration patterns acquired with smartphones, PloS one 9 (4) (2014) e94811. doi:10.1371/journal.pone.0094811.

[25] A. Sucerquia, J. D. López, J. F. Vargas-Bonilla, Sisfall: A fall and movement dataset, Sensors 17 (1) (2017) 198. doi:10.3390/s17010198.

[26] Y. Zigel, D. Litvak, I. Gannot, A method for automatic fall detection of elderly people using floor vibrations and sound—proof of concept on human mimicking doll falls, IEEE transactions on biomedical engineering 56 (12) (2009) 2858–2867. doi:10.1109/TBME.2009.2030171.

[27] E. E. Geertsema, G. H. Visser, M. A. Viergever, S. N. Kalitzin, Automated remote fall detection using impact features from video and audio, Journal of biomechanics 88 (2019) 25–32. doi:10.1016/j.jbiomech.2019.03.007.

[28] S. Spinsante, M. Pepa, S. Pirani, E. Gambi, F. Fioranelli, Micro doppler radar and depth sensor fusion for human activity monitoring in aal, in: Sensors: Proceedings of the Fourth National Conference on Sensors, February 21-23, 2018, Catania, Italy 4, Springer, 2019, pp. 519–528. doi:10.1007/978-3-030-04324-7_62.

[29] N. Lu, Y. Wu, L. Feng, J. Song, Deep learning for fall detection: Three-dimensional cnn combined with lstm on video kinematic data, IEEE Journal of Biomedical and Health Informatics 23 (1) (2019) 314–323. doi:10.1109/JBHI.2018.2808281.

[30] E. Alam, A. Sufian, P. Dutta, M. Leo, Vision-based human fall detection systems using deep learning: A review, Computers in biology and medicine 146 (2022) 105626. doi:10.1016/j.compbiomed.2022.105626.

[31] X. Ma, H. Wang, B. Xue, M. Zhou, B. Ji, Y. Li, Depth-based human fall detection via shape features and improved extreme learning machine,

IEEE Journal of Biomedical and Health Informatics 18 (6) (2014) 1915–1922. doi:10.1109/JBHI.2014.2304357.

[32] Z. Zhang, C. Conly, V. Athitsos, Evaluating depth-based computer vision methods for fall detection under occlusions, in: Advances in Visual Computing, Springer International Publishing, Cham, 2014, pp. 196–207. doi:10.1007/978-3-319-14364-4_19.

[33] Y. M. Galvão, J. Ferreira, V. A. Albuquerque, P. Barros, B. J. Fernandes, A multimodal approach using deep learning for fall detection, Expert Systems with Applications 168 (2021) 114226. doi:10.1016/j.eswa.2020.114226.

[34] R. Espinosa, H. Ponce, S. Gutiérrez, L. Martínez-Villaseñor, J. Brieva, E. Moya-Albor, A vision-based approach for fall detection using multiple cameras and convolutional neural networks: A case study using the up-fall detection dataset, Computers in biology and medicine 115 (2019) 103520. doi:10.1016/j.compbiomed.2019.103520.

[35] H. Ramirez, S. A. Velastin, I. Meza, E. Fabregas, D. Makris, G. Farias, Fall detection and activity recognition using human skeleton features, IEEE Access 9 (2021) 33532–33542. doi:10.1109/ACCESS.2021.3061626.

[36] S. Vadivelu, S. Ganesan, O. V. R. Murthy, A. Dhall, Thermal imaging based elderly fall detection, Springer International Publishing, Cham, 2017, pp. 541–553. doi:10.1007/978-3-319-54526-4_40.

[37] A. Akula, A. K. Shah, R. Ghosh, Deep learning approach for human action recognition in infrared images, Cognitive Systems Research 50 (2018) 146–154. doi:10.1016/j.cogsys.2018.04.002.

[38] J. Nogas, S. S. Khan, A. Mihailidis, Fall detection from thermal camera using convolutional lstm autoencoder, in: Proceedings of the 2nd workshop on aging, rehabilitation and independent assisted living, IJCAI workshop, 2018. doi:10.29007/XT7R.

[39] A. Naser, A. Lotfi, M. D. Mwanje, J. Zhong, Privacy-preserving, thermal vision with human in the loop fall detection alert system, IEEE Transactions on Human-Machine Systems (99) (2022) 1–12. doi:10.1109/THMS.2022.3203021.

[40] F. Sposaro, G. Tyson, ifall: An android application for fall monitoring and response, in: 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2009, pp. 6119–6122. doi:10.1109/IEMBS.2009.5334912.

[41] A. Z. Rakhman, Kurnianingsih, L. E. Nugroho, Widyawan, u-fast: Ubiquitous fall detection and alert system for elderly people in smart home environment, in: 2014 Makassar International Conference on Electrical Engineering and Informatics (MICEEI), 2014, pp. 136–140. doi:10.1109/MICEEI.2014.7067326.

[42] X. Yu, X. Wang, P. Kittipanya-Ngam, H. L. Eng, L.-F. Cheong, Fall detection and alert for ageing-at-home of elderly, in: M. Mokhtari, I. Khalil, J. Bauchet, D. Zhang, C. Nugent (Eds.), Ambient Assistive Health and Wellness Management in the Heart of the City, Springer, Berlin, Heidelberg, 2009, pp. 209–216. doi:10.1007/978-3-642-02868-7_26.

[43] M. Safarzadeh, Y. Alborzi, A. N. Ardekany, Real -time fall detection and alert system using pose estimation, in: 2019 7th International Conference on Robotics and Mechatronics (ICRoM), 2019, pp. 508–511. doi:10.1109/ICRoM48714.2019.9071856.

[44] E. E. Stone, M. Skubic, Testing real-time in-home fall alerts with embedded depth video hyperlink, in: C. Bodine, S. Helal, T. Gu, M. Mokhtari (Eds.), Smart Homes and Health Telematics, Springer International Publishing, Cham, 2015, pp. 41–48. doi:10.1093/geront/gnv044.

[45] N. B. Joshi, S. Nalbalwar, A fall detection and alert system for an elderly using computer vision and internet of things, in: 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology (RTEICT), 2017, pp. 1276–1281. doi:10.1109/RTEICT.2017.8256804.

[46] G. Farnebäck, Two-Frame Motion Estimation Based on Polynomial Expansion, in: Scandinavian conference on Image analysis, Springer, 2003, pp. 363–370. doi:10.1007/3-540-45103-X_50.

[47] M. Lupion, F. Cruciani, I. Cleland, C. Nugent, P. M. Ortigosa, Data augmentation for human activity recognition with generative adversarial

networks, IEEE Journal of Biomedical and Health Informatics (2024). doi:10.1109/JBHI.2024.3364910.

[48] F. Cruciani, C. Sun, S. Zhang, C. Nugent, C. Li, S. Song, C. Cheng, I. Cleland, P. Mccullagh, A public domain dataset for human activity recognition in free-living conditions, in: 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), IEEE, 2019, pp. 166–171. doi:10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00071.

[49] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Advances in neural information processing systems 30 (2017). doi:10.48550/arXiv.1705.07874.

[50] G. Louppe, L. Wehenkel, A. Sutera, P. Geurts, Understanding variable importances in forests of randomized trees, Advances in neural information processing systems 26 (2013).

[51] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, Journal of artificial intelligence research 16 (2002) 321–357. doi:10.1613/jair.953.

[52] M. Lupión, A. Polo-Rodríguez, J. Medina-Quero, J. F. Sanjuan, P. M. Ortigosa, 3d human pose estimation from multi-view thermal vision sensors, Information Fusion 104 (2024) 102154. doi:10.1016/j.inffus.2023.102154.

[53] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780. doi:10.1162/neco.1997.9.8.1735.

[54] S. A. de Medicina Familiar y Comunitaria, Cómo actuar ante las caídas en ancianos - SAMFyC, https://www.samfyc.es/revista/como-actuar-ante-las-caidas-en-ancianos/, accessed on 05/13/2024 (05 2012).

[55] A. de Madrid, Los accidentes en las personas mayores iii-cómo actuar ante un accidente - página de salud pública

del ayuntamiento de Madrid, https://madridsalud.es/los-accidentes-en-las-personas-mayores-iii-como-actuar-ante-un-accidente/, accessed on 05/13/2024.

[56] A. Adin, E. T. Krainski, A. Lenzi, Z. Liu, J. Martínez-Minaya, H. Rue, Automatic cross-validation in structured models: Is it time to leave out leave-one-out?, Spatial Statistics 62 (2024) 100843. doi:10.1016/j.spasta.2024.100843.

[57] D. M. W. Powers, Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation (2020). arXiv:2010.16061, doi:10.48550/arXiv.2010.16061.

[58] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love, et al., Gemma: Open models based on gemini research and technology, arXiv preprint arXiv:2403.08295 (2024). doi:10.48550/arXiv.2403.08295.

[59] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, et al., Gemini: a family of highly capable multimodal models, arXiv preprint arXiv:2312.11805 (2023). doi:10.48550/arXiv.2312.11805.

[60] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 (1) (1967) 21–27. doi:10.1109/TIT.1967.1053964.

[61] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32. doi:10.1023/A:1010933404324.

[62] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 785–794. doi:10.1145/2939672.2939785.

[63] J. H. Friedman, Greedy function approximation: a gradient boosting machine, Annals of Statistics 29 (5) (2001) 1189–1232. doi:10.1214/aos/1013203451.