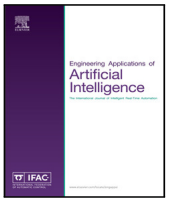
Engineering Applications of Artificial Intelligence 132 (2024) 107937

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Research paper

Transformer-based fall detection in videos

Adrián Núñez-Marcos [a,b,*], Ignacio Arganda-Carreras [c,d,e,f]

[a] HiTZ Center - Ixa, University of the Basque Country (UPV/EHU), Paseo Manuel Lardizabal 1, Donostia/San Sebastián, 20018, Spain
[b] Department of Computer Languages and Systems, University of the Basque Country (UPV/EHU), Paseo Rafael Moreno Pitxitxi 3, Bilbao, 48013, Spain
[c] Department of Computer Science and Artificial Intelligence, University of the Basque Country (UPV/EHU), Manuel Lardizabal 1, Donostia, 20008, Basque Country, Spain
[d] Donostia International Physics Center (DIPC), Manuel Lardizabal 4, Donostia, 20018, Basque Country, Spain
[e] IKERBASQUE, Basque Foundation for Science, Plaza Euskadi 5, Bilbao, 48009, Basque Country, Spain
[f] Biofisika Institute (CSIC-UPV/EHU), Barrio Sarriena, Leioa, 48940, Basque Country, Spain

ARTICLE INFO

MSC:
68T05
68T45

Keywords:
Fall detection
Computer vision
Transformer
Health

ABSTRACT

Falls pose a major threat for the elderly as they result in severe consequences for their physical and mental health or even death in the worst-case scenario. Nonetheless, the impact of falls can be alleviated with appropriate technological solutions. Fall detection is the task of recognising a fall, i.e. detecting when a person has fallen in a video. Such an algorithm can be implemented in lightweight devices which can then cater to the users' needs, e.g. alerting emergency services or caregivers. At the core of those systems, a model capable of promptly recognising falls is crucial for reducing the time until help comes. In this paper we propose a fall detection solution based on transformers, i.e. state-of-the-art neural networks for computer vision tasks. Our model takes a video clip and decides if a fall has occurred or not. In a video stream, it would be applied in a sliding-window fashion to trigger an alarm as soon as it detects a fall. We evaluate our fall detection backbone model on the large UP-Fall dataset, as well as on the UR fall dataset, and compare our results with existing literature using the former dataset.

1. Introduction

According to the Centers for Disease Control and Prevention,[1] falls represent a significant cause of injury and, in some cases, even fatalities over the age of 65 in the United States, where a fall occurs every second, every day, affecting one out of four elderly adults each year. In a society with an ever-ageing population, this issue not only presents health concerns but also creates economic challenges related to their treatment. The aftermath of falls often leads to a loss of independence, impacting elderly adults' daily live. Hence, preventing falls or alleviating their impact is of paramount importance for a healthy ageing. That is why research related to fall detection is crucial to develop technologies capable of aiding the elderly feel safer in their daily routines.

In this paper, we focus on vision-based approaches (those including a vision sensor) for fall detection due to the advantages they offer compared to their wearable sensor-based counterpart (wearable sensors like accelerometers, excluding wearable vision sensors). Vision-based approaches are less intrusive and eliminate compliance issues associated with wearing special garments, particularly for patients with cognitive issues such as dementia. Moreover, the widespread prevalence of cameras nowadays presents an opportunity to leverage their ubiquity, potentially allowing for the scalability of fall detection models beyond specific settings like smart homes to broader contexts such as public spaces. This holds especially true for 2D cameras, in contrast to 3D cameras (which are capable of capturing depth information). Additionally, 2D cameras provide a more cost-effective solution compared to 3D range sensors, which are often more expensive and may require additional hardware setup and calibration.

Thanks to the advent of deep learning for vision-based models, the performance of vision-based methods has significantly improved, closing the gap between sensor-based and vision-based models in terms of performance. In fact, the transformer technology introduced in Vaswani et al. (2017) has replaced Convolutional Neural Networks (CNNs) in many tasks. Consequently, in this paper, we propose the use of a transformer-based neural network for the detection of falls in videos.

Our objective is to extract features from raw RGB frames, without the need for additional computations such as optical flow (OF) images, skeletons/poses and so on. To the best of our knowledge, we are the

* Corresponding author at: Department of Computer Languages and Systems, University of the Basque Country (UPV/EHU), Paseo Rafael Moreno Pitxitxi 3, Bilbao, 48013, Spain.
E-mail address: adrian.nunez@ehu.eus (A. Núñez-Marcos).

[1] https://www.cdc.gov/injury/features/older-adult-falls/index.html

https://doi.org/10.1016/j.engappai.2024.107937
Received 28 February 2023; Received in revised form 16 January 2024; Accepted 17 January 2024
Available online 22 January 2024

first to directly apply transformers to address the fall detection task using only RGB images, without requiring to compute additional features. Given that fall detection models are usually deployed in lightweight devices for inference, it is imperative that the employed models have low latency and minimal dependencies. Additionally, since the timely detection of falls is critical due to their severe consequences, we adhere to the guidelines of the dataset we propose for evaluation, the UP-Fall dataset (Martínez-Villaseñor et al., 2019), by reporting detection results at 1-second intervals, or, equivalently, in a 16-frame video interval. The UP-Fall dataset comprises 11 activities, making it suitable for fall detection, as nearly half of the classes are related to falls.

We present two parallel evaluation strategies to facilitate a comprehensive comparison with the existing literature. The first strategy aligns with the approach taken by the original authors of the UP-Fall dataset and has also been used in the subsequent fall detection challenge they organised. The second strategy follows the approach of Espinosa et al. (2019), in which they compare their model in the binary classification (by grouping the 11 activities into two classes: fall and no fall) and multiclass classification settings.

Furthermore, we conducted experiments to assess the model's ability to learn from additional datasets and generalise effectively. To evaluate this, we selected the UR Fall dataset (Kwolek and Kepski, 2014) and a performed joint training using both the UP-Fall and UR Fall datasets. Subsequently, we evaluated the model's performance on each dataset separately. It is important to note that, while our model demonstrated the ability to learn from diverse data, we acknowledge the limitation that its real-world application would need a substantial dataset, which is currently unavailable for the fall detection task. Nonetheless, we believe that the model has the potential to adapt and further improve through additional data, as demonstrated by its performance on the UR Fall dataset.

The paper makes two significant contributions. Firstly, we propose the first vision-based transformer specifically designed to learn solely from RGB data for fall detection. Secondly, we provide a comprehensive comparison of our results with the existing literature, specifically focusing on works that do not rely on additional features, thereby ensuring that the model directly learns from RGB frames using the UP-Fall dataset. Furthermore, we have made all the experimental code publicly available (see Section 3), enabling fellow researchers to easily verify and build upon our findings.

The remainder of the paper is organised as follows: Section 2 delves into the recent fall detection literature, Section 3 introduces our proposed transformer model and, in Section 4, we present the UP-Fall dataset, explain the evaluation strategy and compare our results with the existing literature. Finally, we give some concluding remarks on Section 5.

2. Related works

Fall detection (Alam et al., 2022) is the task of detecting when a person is falling so that an alarm can be raised and call, for example, an ambulance or warn someone. The types of approaches followed for this detection (depending on what is used to detect the fall) can be divided between sensor-based approaches (Nooruddin et al., 2021) and vision-based approaches (Gutiérrez et al., 2021). Vision-based methods are, in theory, very rich in information, but the computational capacity and the algorithms were not able to correctly exploit it until recently. Due to the increasing interest in deep learning networks, this research topic shifted its interest to the vision-based methods that will be explained in this literature review.

Fall detection cannot be approached as a regular video classification task. A potential fall needs to be detected as soon as possible (within a video stream) in order for a fall detection model to be useful in a real-life situation. That is why intermediate outputs need to be generated. The most common method, thus, is the use of a sliding window that takes a chunk of frames and decides whether a fall has occurred. For example, a pioneer work which introduced CNNs to solve the fall detection was Yu et al. (2017). The authors of that work extracted a binary silhouette of the person appearing in each frame and carried out a per-frame classification of the pose, and identified falls among their potential outputs. Instead of using a CNN to directly classify images, Wang et al. (2016) extracted several features from silhouette images, which also included CNN features among them. Both methods required to segment people from images, which may be prone to errors in some cases (e.g. multiple people, cluttered background, etc.). Instead, compared to those first works, we directly use the RGB frames to infer the fall.

Instead of binarising images, Núñez-Marcos et al. (2017) extracted OF images from videos to perform sliding-window-based fall classification, using 10 pairs of OF images to output a possible fall detection. The authors employed a VGG16 (Simonyan and Zisserman, 2014) network (with the feature extractor part frozen) and trained it to perform a binary classification task. Similarly, Espinosa et al. (2019) also extracted OF images but instead of directly stacking horizontal and vertical components, the magnitude of the flow was computed. Moreover, the authors combined those magnitudes from different cameras and resized them to a small resolution. Their model was a small CNN with a binary cross entropy loss. Similar to the first works introduced in this section, these also require the computation of additional features (in this case OF images), which can add more computational burden to the fall detection pipeline. In fact, depending on the lighting conditions, the generated OF images may not be really helpful since the OF algorithm does not correctly recognise the movement flow with not-controlled lighting conditions. Lu et al. (2018) trained a 3DCNN and an LSTM model in which the 3DCNN was pre-trained in the Sports-1M dataset (Karpathy et al., 2014) (not related to fall detection) and an LSTM was trained for fall detection making use of the already pre-trained feature extractor. We believe that the Transformer-based network we employ in this work is more interesting to model the temporal dynamics. Due to its self-attention component, the network can attend to all the tokens.

A multi-stream approach was proposed by Carneiro et al. (2019) with a VGG16 network as a backbone feature extractor. Each stream processed a different feature, namely: stacked OF, poses and RGB data. Chen et al. (2020) extracted the skeleton of the person of interest using OpenPose (Cao et al., 2019) and used a set of heuristics to decide whether the activity could be categorised as a (potential) fall. Moreover, the model incorporated the activation of an alarm which would be triggered if the subject could not stand up.

A mobile-device-oriented application for fall detection was designed by Han et al. (2020): a two-stream approach combining a motion-based feature extraction and a lightweight VGG architecture called mobileVGG. Khraief et al. (2020) presented a weighted neural multi-stream approach in which the input modalities were: (i) RGB (for colours and textures) and depth (for illumination), (ii) silhouette variations (in order to detect movement), (iii) amplitude and oriented flow and (iv) optical flow. The authors carried out experiments on early and late fusion and also on the weighting of each stream. Berlin and John (2021) employed a Siamese network trained by distance-metric-based learning. The network took pairs of different videos and measured their L1 distance before applying a sigmoid function to the result. If the videos are similar, their ground truth should be 1, or 0 otherwise. Gomes et al. (2022) used a YOLOv3 detection network (Redmon and Farhadi, 2018) to extract humans per-frame and the Kalman filter for the time-aware alignment of frame sequences (tracking each person in the scene). Each sequence was then classified into fall or not fall by a 3DCNN or a 2DCNN with an LSTM.

More recently, the authors of Yadav et al. (2022) evaluated their ARFDNet model with the same dataset we use, i.e. the UP-Fall dataset. ARFDNet is composed of (i) a skeleton extraction module, (ii) a CNN to extract spatial features and (iii) a Gated Recurrent Unit (Cho et al., 2014) module for the spatio-temporal features. The output of the latter

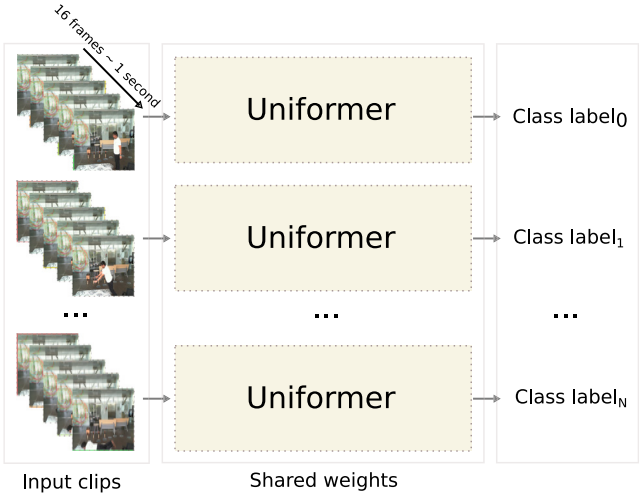


Fig. 1. Our proposed fall detection model. A sequence of input clips (of 1 s each) sampled from a video is passed through the Uniformer network to extract features. Besides, the Uniformer generates, for each clip, a probability distribution across the possible fall and no-fall classes of a given dataset. The highest probability is taken as the predicted class label of each clip.

was used for the classification of activities and falls. Similarly, Suarez et al. (2022) also fed their network with pose information. The former was composed of 1D CNN layers and a classifier on top. Inturi et al. (2023) used a CNN + LSTM combination with poses as input too. Mobsite et al. (2023) employed silhouettes as input to a ConvLSTM (Shi et al., 2015) model. And going even further, Galvão et al. (2022) completely segmented the person on each frame and trained a Generative Adversarial Network (Goodfellow et al., 2014) to classify activities of daily living. In this model, falls are considered anomalies and detected as such. All of these approaches require a preprocessing step of extracting poses, silhouettes or segment the person falling, which adds a computation overhead and can propagate errors to the next step.

Instead of using poses, other features were extracted in the work of Le et al. (2022) using wearable devices. These features, used as input for various traditional classification algorithms, allowed them to obtain very high F1 metric results (96.16 for falls and 99.90 for non-falls) on the UP-Fall dataset.

In contrast to most of these works, our model does not require additional features such as OF or depth images for the detection of falls. This alleviates the computational overhead of computing more features, which may be pivotal for lightweight devices with low computational resources (usually employed for inference).

3. Methodology

A fall detection model addresses the binary problem in which the model must decide, for a given input (e.g. a sequence of frames or data from a wearable device), whether a person is falling or not. For that purpose, our fall detection model's first objective was to exclusively use RGB frames. This means that additional features, e.g. OF or depth images, are not required, thus allowing for the development of computationally less intensive networks. This also reduces the latency, which is crucial for real-time fall detection applications. On the other hand, the second objective of our model was to process videos in a sliding-window fashion to produce intermediate outputs. With this, the model is able to detect falls shortly after processing a few frames, hence allowing the model to quickly respond to fall events.

More formally, consider an input video $X = \{x_1, x_2, \dots, x_N\}$ composed of N frames. We extract several chunks of size W (representing the number of frames within each chunk) and generate an output $P = \{p_1, p_2, \dots, p_{\lceil N/W \rceil}\}$, where each element $p_i = \{0, 1\}$ is the output result, indicating whether a fall has been detected in the ith chunk ($0 \leq i < \lceil N/W \rceil$). This high-level overview of the model is illustrated in Fig. 1. In a data stream, frames accumulate until W frames are available to create a chunk and a single output (indicating whether a fall has been detected) is generated. For the evaluation of our model, we will use a state-of-the-art fall detection dataset and, thus, we will consider sets of videos of varying sizes instead of a continuous stream of frames.

Our fall detection model takes each of the $clip_i$ ($0 \leq i < \lceil N/W \rceil$) chunks and passes them through a feature extraction network M. This network decides whether a fall has ocurred in the input video clip. Our chosen backbone network, M, is a Uniformer (Li et al., 2022), which is a vision transformer that, as highlighted by the authors, has a good balance between accuracy and computational efficiency. This is desirable for applications looking for a good performance but with a minimal latency. What the authors of Li et al. (2022) contribute in their paper is the Uniformer block, which is composed of three components: (i) the Dynamic Position Embedding (DPE), (ii) the Multi-Head Relation Aggregator (MHRA) and a feedforward network. Fig. 2 illustrates a Uniformer block with its three main components.

Concerning each of the components of the Uniformer block, the first one, the DPE, is a lightweight position encoding based on a depthwise convolution, adaptable to varying sequence lengths. The MHRA is a self-attention block designed to minimise redundancy; it works like a convolutional layer: it applies self-attention on a smaller neighbourhood of tokens instead of trying to apply attention over all tokens. This includes a token affinity matrix that expresses the relation between two tokens or positions. In shallow layers, token affinity is simply the relative distance between tokens. In deeper layers, token affinity is computed as the content similarity with the rest of the tokens within the neighbourhood. Having taken these three components into account to build a Uniformer block, the Uniformer network is built stacking local and global Uniformer blocks (i.e. stacking blocks that apply MHRA in shallow layers and blocks of deeper layers, respectively).

The Uniformer network is pretrained[2] on two human action classification datasets, Kinetics (Smaira et al., 2020) and Something-Something (Goyal et al., 2017), at a resolution of 224 × 224. Since the model is pretrained, W will be fixed to 16, i.e. 16 frames are taken to detect falls. Fig. 2 illustrates the structure of the model.

Each video clip of W frames is automatically labelled taking the majority vote of the per-frame ground-truth class labels. In other words, within a single chunk $clip_i = \{x_j, x_{j+1}, \dots, x_{j+W}\}$, each frame x_j will have its own label $y_j = \{0, 1, \dots, C\}$, where C represents the amount of classes in the dataset. The dataset comprises several classes, some of which are related to falls. Depending on the experiment, the number of classes can be reduced to 2 (binary classification) and, hence, each frame will be classified as negative or positive.

We trained the model on a per-clip basis, treating each clip of size W as a training sample. We employ cross entropy loss and the Adam optimiser for the training. After each epoch, an evaluation is conducted on the development set (that is, an evaluation dataset extracted from the training set and not used for training). Training is stopped when a chosen metric (F1 score in our experiments, see Section 4.2 for our evaluation metrics) computed on the development set does not improve after a predefined number of epochs. This number is referred to as patience and is shown in the experiment tables of Section 4.3. In what follows, the patience has been set to 10 epochs.

The code for these experiments can be accessed on GitHub.[3]

[2] https://huggingface.co/Sense-X/uniformer_video
[3] https://github.com/AdrianNunez/transformer-based-fall-detection

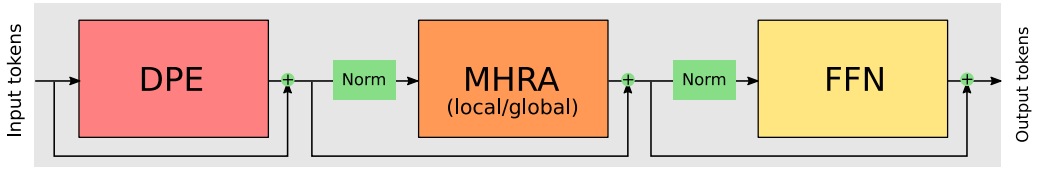


Fig. 2. Uniformer block that is stacked to build the Uniformer network. It is composed of three main components: the Dynamic Position Embedding (DPE), the Multi-Head Relation Aggregator (MHRA) and a feedforward network (FFN). The purpose of the MHRA is to minimise redundancy. We refer readers to the original Uniformer publication (Li et al., 2022) for further details about its architecture.

Fig. 3. UP-Fall Detection Dataset sample video frames. Example of a sequence of frames showing a fall, corresponding to Subject 1, Activity 1, Trial 1.

Table 1
UP-Fall dataset's activities or classes. Classes 1–5 are fall-related classes.

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

4. Evaluation

4.1. Datasets

The UP-Fall dataset (introduced by Martínez-Villaseñor et al. 2019) is a large fall detection dataset composed of 11 activities (see Table 1), each with 3 trials, and recorded using 17 young adults without impairments. The dataset contains data from wearable sensors, ambient sensors and vision devices (although, in this paper, only the latter will be used). Concerning the vision devices, two cameras are available, each providing a distinct viewpoint of the falls. For our experiments, we only employed the data from camera 1 since the data obtained from camera 2 was considered to be too noisy. A sample sequence (from camera 1) of the dataset is shown in Fig. 3.

The dataset can be binarised by merging classes 1 to 5 into a single class, which we call "Class 1", while the rest are merged into another one which we will refer to as "Class 0". Depending on the evaluation strategy employed, the binary setting or the multiclass setting will be used.

The UR Fall dataset (Kwolek and Kepski, 2014) is another fall detection dataset comprising 70 videos, where 30 of them contain a fall event (see Fig. 4 for an example). Since fall detection datasets are inherently unbalanced in terms of classes (since there are many more non-fall samples), we restricted the dataset to these 30 videos and did include the remaining 40 videos without falls.

The dataset has been annotated frame by frame with three possible labels: "fall has not occurred", "falling" and "on the floor" (after the fall). We binarise the dataset so that any frame not labelled as "falling" is considered a "not fall" frame. Moreover, the dataset also contains data from accelerometers and another camera view. The former will not be employed in this work since we are exclusively interested in vision-based approaches. The additional camera view provides a top-down perspective, which is not usual in fall detection datasets. It would be interesting to cover it in another work as a top-view approach, but we have deemed it out of the scope of this work.

4.2. Evaluation methodology

In order to compare our work with the state of the art, we adopted two evaluation strategies. We will simply refer to them as the first and the second evaluation strategies.

In the first evaluation strategy we will adopt in this work, which was originally proposed in the paper of the dataset (Martínez-Villaseñor et al., 2019) and has been described in Section 4.1, a multiclass classification problem is addressed. The authors also proposed a public fall detection challenge, which was presented in Ponce and Martínez-Villaseñor (2020). This is precisely the first evaluation strategy we will adopt in this work. We split the data into three sets: training, development and test. The training set is used to tune the network's weights; the development set is used to evaluate the model iteratively and stop the training; and the test set is used for the final evaluation. The following subjects' data is used for training: 1, 3, 4, 7 and 10–14, in total they comprise 70% of the dataset. The trial 3 of subjects 1, 3 and 4 were chosen by us for the development set, as the original challenge does not specify how to create a development set. For the testing or evaluation set, the challenge proposes the data from subjects 15–17. The detection results to be evaluated must be given using windows of 1 s of duration, without overlapping. The label of a given window is considered to be the most frequent one among the labels of individual frames within the window, as described in Section 3.

The second evaluation strategy we employed is the one originally presented by Espinosa et al. (2019) in which the classes are binarised, i.e. any fall class is considered class 1 while the rest of activities are grouped in class 0. For the sake of comparison with the literature, we also obtained results for the multiclass setting. All trial 3 data is used for the test set while the remaining trials' data is used for the training set. Just like in the previous strategy, we created a development set taking trial 2 data of subjects 1, 3 and 4.

The metrics proposed for the evaluation are the accuracy and the F1 score (using the implementation of Pedregosa et al. 2011). The former one is usually given in the state of the art, although it is not very useful in fall detection datasets as they tend to be skewed, i.e. there are many more negative samples than positive samples, making the accuracy not reliable. In fact, in tasks such as fall detection, in which

Fig. 4. UR Fall Detection Dataset sample video frames (camera 0, first sample with fall).

Table 2
Summary of the experiments performed with the first evaluation strategy as proposed by Dodge et al. (2019).

Computing infrastructure	Nvidia A100	
Best validation accuracy	98.21	
Best validation F1-score	91.89	
Training duration	5.12 h	
Model implementation	https://github.com/AdrianNunez/transformer-based-fall-detection	
Hyperparameter	**Search space**	**Best assignment**
Number of epochs	{10, 50}	50
Learning rate	{10e−4, 5e−5 10e−5, 5e−6}	1e−5
Batch size	16	16
Weight decay	0.00001	0.00001
Early stopping patience (in epochs)	10	10
Oversample classes	{No, Yes}	Yes
Model variation	{Small400, Baseline400, Small600, Baseline600}	Small400

Table 3
Results for the first evaluation strategy with the UP-Fall dataset.

ID	Accuracy	F1-score
Martínez-Villaseñor et al. (2019)	94.32 (±0.31)	70.44 (±1.25)
Challenge 1st position (Ponce and Martínez-Villaseñor, 2020)	–	82.47
Challenge 2nd position (Ponce and Martínez-Villaseñor, 2020)	–	34.04
Challenge 3rd position (Ponce and Martínez-Villaseñor, 2020)	–	31.37
Challenge honorific mention (Ponce and Martínez-Villaseñor, 2020)	–	60.40
Ours	96.67	82.24

not detecting a fall can lead to serious consequences, it is crucial to avoid false negatives. Given the small amount of positive samples in fall detection datasets, the accuracy metric can misleading, as a high accuracy can also come with a relatively high number of false negatives. Alternatively, the F1 score is proposed in the UP-Fall challenge and is recommended as an alternative to the accuracy as it takes into account the unbalanced nature of fall datasets. For our experiments, we computed the unweighted mean of F1 scores across classes.

4.3. Results

The results of our experiments are compared with the state of the art if the comparison is fair, i.e. the results are compared under the same evaluation strategy, data split and so on. We divided the experiments into two sets: those experiments using the first evaluation strategy and those using the second one. In the latter, we also divided the experiments between those using a binary classification approach and those following a multiclass classification setting.

Among the works that are left out of this comparison, we have Ramirez et al. (2021, 2022), in which the authors extracted skeleton poses from RGB frames. Ramirez et al. (2021) only used individual frames, but Ramirez et al. (2022) employed 1-second windows of poses (poses of every frame) to classify instances between fall and not fall. However, their data split was randomly selected and, hence, it is not directly comparable with any of the two strategies presented here. Their best results were obtained with a Random Forest classifier, obtaining a 99.81% of accuracy and a 99.56 of F1. Afterwards, the same authors extended this work with Ramirez et al. (2023). Since in their first work they did not obtain good results using an LSTM model, in this new work they used a BERT model (Devlin et al., 2018), whose inputs were pose sequences. They initially obtained an accuracy of 81.14% and an F1 score of 80.95, but they argued that the lower results are a consequence of the class imbalance. To alleviate this, they artificially augmented the dataset using a GAN network called TABGAN (Ashrapov, 2020). With this new data taken into consideration, the accuracy and F1 score increased to 99.50% and 87.20, respectively.

Following with the use of poses, Taufeeque et al. (2021) obtained poses with a multi-camera and multi-person approach. Their approach also employed an LSTM network and obtained an F1 score of 92.5. Meanwhile, Galvão et al. (2021b) employed a spatio-temporal graph neural network (pretrained on a large activity recognition dataset) as a feature extractor. An autoencoder tried to reconstruct the input and, in case the error was higher than a predefined threshold, an anomaly (a fall) was detected. Their proposed method led them to an accuracy of 98.62% and an F1 score of 93. All the works mentioned here detect falls in a binary setting (not multiclass), but they do not share the data splits of the first and second evaluation strategies and, therefore, cannot be directly compared with our experiments. Nonetheless, they also obtained remarkable results, compared with the results obtained by our model.

4.3.1. Results under the first evaluation strategy

With the first evaluation strategy, we made the hyperparameter search detailed in Table 2 following the guideline to present machine learning results published by Dodge et al. (2019). Four variations of the Uniformer were used, namely, the small and baseline versions pretrained on Kinetics-400 and on Kinetics-600.

The results of the experiment with this evaluation strategy are shown in Table 3 alongside other approaches in the literature that follow the same evaluation strategy. Martínez-Villaseñor et al. (2019) presented the UP-Fall dataset and some baseline experiments using that dataset with traditional machine learning algorithms, i.e. no deep

Table 4
Second evaluation strategy's search space and best assignments.

Computing infrastructure	Nvidia A100	
Best validation accuracy	99.02 (binary), 97.37 (multiclass)	
Best validation F1-score	93.83 (binary), 97.20 (multiclass)	
Training duration	1,67 h (binary), 12.76 h (multiclass)	
Model implementation	https://github.com/AdrianNunez/transformer-based-fall-detection	
Hyperparameter	Search space	Best assignment
Number of epochs	{10, 50}	50
Learning rate	{10e−4, 10e−5}	1e−4
Batch size	16	16
Weight decay	0.00001	0.00001
Early stopping patience (in epochs)	10	10
Class weight for falls	{1, 2}	1
Oversample classes	{No, Yes}	Yes
Window size	{8, 16}	16 (binary), 8 (multiclass)
Model variation	{Small400, Baseline400, Small600, Baseline600}	Small400

Table 5
Results for evaluation strategy 2 with UP-Fall dataset (with multiclass classification).

ID	Accuracy	F1-score
Espinosa et al. (2019)	82.26	72.94
Ours	93.17	93.39

Table 6
Results for evaluation strategy 2 with UP-Fall dataset (with binary classification).

ID	Accuracy	F1-score
Espinosa et al. (2019)	95.64	97.43
Ours	99.17	94.14

learning algorithm was used. The models they applied were Random Forests, Support Vector Machines, k-Nearest Neighbours and Multilayer Perceptrons. They also explored various data types and their combinations: (i) infrared sensor data, (ii) wearable IMU data, (iii) all wearable IMU data and the EEG headset data, (iv) all infrared sensors, all wearable IMU data and the EEG headset data, (v) camera data, (vi) all infrared sensors and camera data and (vii) all wearable IMU, EEG headset and camera data. Their best result in terms of accuracy and F1-score, shown in Table 3, was obtained with a Multilayer Perceptron and a window size of 1 s, using all wearable IMU, EEG headset and camera data as input.

After the aforementioned work, the team launched the challenge presented in Ponce and Martínez-Villaseñor (2020). They presented the winners of the challenge and one honorific mention. The results obtained by these four participants are shown in Table 3. The winner employed a Random Forest and sensor data, the second place used a 1-layer CNN and sensor data, the third place made use of a bi-LSTM (the data used is not mentioned) and the honorific mention did not send a short paper and, thus, it is unknown how they obtained their result.

With the first evaluation strategy, we obtained a result similar to the first position of the challenge presented in Ponce and Martínez-Villaseñor (2020) only relying on vision data, without the need of the sensor data they employed. Besides, compared with the best baseline model proposed in Martínez-Villaseñor et al. (2019), we have an improvement of more than 10 points in the F1 score.

4.3.2. Results under the second evaluation strategy

With the second evaluation strategy, we also made a hyperparameter search. The details have been written down in Table 4. Once again, four variations of the Uniformer were explored.

Let us begin by comparing our multiclass result (see Table 5) with the one obtained by Espinosa et al. (2019). We were able to obtain a 20 point difference in the F1 score with respect to them. For the binary classification case, shown in Table 6, we are 3 points below in the F1 score, although both results are very high. Nonetheless, our purpose was to create a model that only takes RGB frames, without any additional computation and, in contrast, Espinosa et al. (2019) used OF images. In fact, the task may get easier using OF images due to the erased background clutter. Our model, in contrast, seems to generalise better to more classes, maybe due to the usage of RGB frames and the suppression of appearance-related features.

Even though the comparison it not fair, the works presented in the introduction of Section 4.3, i.e. Ramirez et al. (2021, 2022), Ashrapov (2020), Taufeeque et al. (2021), Galvão et al. (2021b), also presented results of a binary fall detection task. We were able to perform better than most of them even though we did not compute skeletons.

4.3.3. Joint fine-tuning with UP-Fall and UR Fall datasets

To assess the adaptability of our approach to other datasets, we conducted an additional experiment by combining two datasets: UP-Fall and UR Fall (both introduced in Section 4.1). Using the pretrained network (on UP-Fall) without fine-tuning on the new dataset (UR Fall) the results were unsatisfactory, as shown in the first row of Table 7. The accuracy was only 43.48% and the F1 score was 30.30. This outcome is attributed to the fact that the original benchmark-trained model lacks the ability to generalise to any fall event, as it has not been trained with sufficient data from diverse sources. However, collecting a massive amount of data for fall detection is currently not possible (to the best of our knowledge). To address this limitation, we propose a fine-tuning approach (i.e. re-training the pretrained Uniformer from scratch) in which we train the network with both datasets together (mixed in the same training procedure) to observe how the model adapts when provided with more data.

The training procedure for this experiment followed the same approach as in our previous experiments (using the second evaluation strategy with binary classes). We used a combined development dataset (including samples of both classses, equally represented) to guide the training. In order to identify the optimal fine-tuning learning rate, we explored three different learning rates: $1e^{-4}$, $5e^{-4}$ and $5e^{-5}$ (the best result was obtained with $5e^{-5}$). Additionally, we experimented with the use of a class weight of 2 for the fall class to address any class imbalance issues that may arise during training and we saw that the use of this weight improved the results. Furthermore, to ensure a fair representation of the fall class in the UR Fall dataset, we performed oversampling. The fall class was oversampled to match the number of samples in the negative class within the same dataset. This oversampling technique allowed us to mitigate potential biases and improve the model's ability to learn from both classes effectively.

The results can be found in Table 7. Although the training is performed with both datasets at the same time, the evaluation is divided as seen in Table 7 to assess the results on both datasets separately. A slight drop in performance is observed on the UP-Fall dataset, likely attributed to the model having to learn the appearance of another dataset. Nevertheless, even with this drop, the performance on both datasets remains remarkably high in terms of F1 score. This outcome is encouraging and suggests that the model has the potential to generalise well to different fall scenarios.

Table 7
Results for evaluation strategy 2 with the UP-Fall and UR Fall datasets (with binary classification) mixed together. The first result for UR Fall has been computed using the best model fine-tuned with UP-Fall in previous experiments.

Dataset	Accuracy	F1-score
UR Fall (not fine-tuned)	43.48	30.30
UP-Fall (jointly fine-tuned)	99.03	92.35
UR Fall (jointly fine-tuned, w/o oversampling)	91.30	89.73
UR Fall (jointly fine-tuned, w/ oversampling)	95.45	94.76

4.3.4. Comparison with wearable-based fall detection

Throughout this paper, we have focused on vision-based approaches, specifically those using 2D cameras. However, it is essential to acknowledge that wearable-sensor-based solutions have their own set of advantages and disadvantages, depending on the specific scenario. In terms of performance, wearable sensors often provide more discriminative data for the detection of falls, which can lead to a higher accuracy in this task compared to vision-based methods. As a result, wearable-sensor-based solutions tend to achieve better results in fall detection tasks. In this section, we present a comparative analysis, contrasting the results obtained from our vision-based approach with those of wearable-sensor-based solutions. By understanding the trade-offs and strengths of each approach, we aim to provide insights into the relative merits of vision-based and wearable-sensor-based fall detection models.

Table 8 presents a summary of the recent results from the literature for the UP-Fall dataset, specifically focusing on studies using wearable-sensor information or a combination of sensor and RGB data. Our results in this table are based on the second evaluation strategy, as we conducted experiments in both binary and multiclass settings.

It is important to note that a direct comparison between the approaches listed in Table 8 and the model proposed in this paper may not be fair, as they may not share the same train/evaluation splits, compute metrics differently and have different clip lengths for generating outputs. Moreover, some works adopt a binary configuration (i.e., fall or not fall), while others consider all possible classes of the dataset. However, this comparison allows us to observe that our vision-based transformer approach achieves results that are close to the state-of-the-art solutions in the sensor-based fall detection task. This finding further reinforces the promise and potential of vision-based methods for fall detection and highlights the effectiveness of our proposed approach in capturing relevant information from RGB data to identify fall events accurately.

It is worth mentioning that the goal of this comparison is not to establish superiority over other approaches but rather to put in context the performance of our method in relation to the existing body of literature. We believe that the diverse range of fall detection techniques showcased in Table 8 contributes to a comprehensive understanding of the advancements in this field and emphasises the significance of our contributions within the vision-based fall detection domain.

Table 8
Results of the literature of fall detection using the UP-Fall dataset for the evaluation and sensor data or skeleton information as input. For our results, we used the second evaluation strategy.

	Type	Binary?	Accuracy	F1-score
Ponce et al. (2020)	Sensor+RGB	✓	98.72	95.77
Waheed et al. (2021)	Sensor	✓	97.21	97.43[a]
Galvão et al. (2021a)	RGB+Sensor	✓	99.99	–
Al Nahian et al. (2021a)	Sensor	✓	96.00	97.00[a]
Al Nahian et al. (2021b)	Sensor	✓	100.00	–
Ashrapov (2020)	Skeleton	✓	99.50	87.20
Taufeeque et al. (2021)	Skeleton	✓	–	92.5
Galvão et al. (2021b)	Skeleton	✓	98.62	93
Ramirez et al. (2021)	Skeleton	✓	99.34	98.52
Ramirez et al. (2022)	Skeleton	✓	99.81	99.56
Ramirez et al. (2023)	Skeleton	✓	81.14	80.95
Ours	RGB	✓	99.17	94.14
	Type	Binary?	Accuracy	F1-score
Martínez-Villaseñor et al. (2019)	Sensor	✗	95.49	70.31
Chahyati and Hawari (2020)	Sensor	✗	–	81.40
Chahyati and Hawari (2020)	RGB+Sensor	✗	–	95.44
Ramirez et al. (2021)	Skeleton	✗	99.45	92.34
Le et al. (2022)	Sensor	✗	–	99.60
Mohan Gowda et al. (2022)	RGB+Sensor	✗	99.2	98.4
Islam et al. (2023)	RGB+Sensor	✗	97.90	97.88
Yan et al. (2023)	Skeleton+Sensor	✗	98.05	88.30
Ours	RGB	✗	93.17	93.39

[a] Manually computed based on Recall and Precision.

5. Conclusions

In this paper, we introduced a transformer-based fall detection model, leveraging the Uniformer architecture. Our RGB-only approach, aligned with UP-Fall dataset guidelines, achieved competitive or improved results compared to existing methods without relying on additional features or wearable-sensor data. Our fall detection model demonstrates the capability to promptly emit an alarm upon detecting a fall event.

Future research avenues include exploring anticipation capabilities, inspired by recent works such as Li and Song (2023). Collaborating with healthcare professionals is also crucial for refining our model's real-world application. Their insights will guide adjustments to meet end-user needs effectively.

Furthermore, to improve the robustness and generalisability of our model, a larger, diverse fall detection dataset is essential. This expansion will facilitate training a more adaptable and reliable neural network.

In conclusion, our work lays a solid foundation for vision-based fall detection models and presents a promising direction for future research. By exploring proactive fall detection, collaborating with healthcare professionals, and collecting more comprehensive datasets, we aspire to continue advancing the field of fall detection and contribute to improving the safety and well-being of individuals at risk of falls.

CRediT authorship contribution statement

Adrián Núñez-Marcos: Conceptualization, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Ignacio Arganda-Carreras:** Conceptualization, Writing – review & editing.

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

The code used has been publicly shared.

Acknowledgements

This work is supported by grant PID2021-126701OB-I00 funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe", and by grant GIU19/027 funded by the University of the Basque Country UPV/EHU.

References

Al Nahian, M.J., Ghosh, T., Al Banna, M.H., Aseeri, M.A., Uddin, M.N., Ahmed, M.R., Mahmud, M., Kaiser, M.S., 2021a. Towards an accelerometer-based elderly fall detection system using cross-disciplinary time series features. IEEE Access 9, 39413–39431.

Al Nahian, M.J., Ghosh, T., Al Banna, M.H., Uddin, M.N., Islam, M.M., Taher, K.A., Kaiser, M.S., 2021b. Social group optimized machine-learning based elderly fall detection approach using interdisciplinary time-series features. In: 2021 International Conference on Information and Communication Technology for Sustainable Development. Icict4sd, IEEE, pp. 321–325.

Alam, E., Sufian, A., Dutta, P., Leo, M., 2022. Vision-based human fall detection systems using deep learning: A review. Comput. Biol. Med. 105626.

Ashrapov, I., 2020. Tabular GANs for uneven distribution. arXiv preprint arXiv:2010.00638.

Berlin, S.J., John, M., 2021. Vision based human fall detection with siamese convolutional neural networks. J. Ambient Intell. Humaniz. Comput. 1–12.

Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., Sheikh, Y.A., 2019. OpenPose: Realtime multi-person 2D pose estimation using part affinity fields. IEEE Trans. Pattern Anal. Mach. Intell.

Carneiro, S.A., da Silva, G.P., Leite, G.V., Moreno, R., Guimaraes, S.J.F., Pedrini, H., 2019. Multi-stream deep convolutional network using high-level features applied to fall detection in video sequences. In: 2019 International Conference on Systems, Signals and Image Processing. IWSSIP, IEEE, pp. 293–298.

Chahyati, D., Hawari, R., 2020. Fall detection on multimodal dataset using convolutional neural netwok and long short term memory. In: 2020 International Conference on Advanced Computer Science and Information Systems. ICACSIS, IEEE, pp. 371–376.

Chen, W., Jiang, Z., Guo, H., Ni, X., 2020. Fall detection based on key points of human-skeleton using openpose. Symmetry 12 (5), 744.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Dodge, J., Gururangan, S., Card, D., Schwartz, R., Smith, N.A., 2019. Show your work: Improved reporting of experimental results. arXiv preprint arXiv:1909.03004.

Espinosa, R., Ponce, H., Gutiérrez, S., Martínez-Villaseñor, L., Brieva, J., Moya-Albor, E., 2019. A vision-based approach for fall detection using multiple cameras and convolutional neural networks: A case study using the UP-fall detection dataset. Comput. Biol. Med. 115, 103520.

Galvão, Y.M., Ferreira, J., Albuquerque, V.A., Barros, P., Fernandes, B.J., 2021a. A multimodal approach using deep learning for fall detection. Expert Syst. Appl. 168, 114226.

Galvão, Y.M., Portela, L., Barros, P., de Araújo Fagundes, R.A., Fernandes, B.J., 2022. OneFall-GAN: A one-class GAN framework applied to fall detection. Eng. Sci. Technol. Int. J. 35, 101227.

Galvão, Y.M., Portela, L., Ferreira, J., Barros, P., Fagundes, O.A.D.A., Fernandes, B.J., 2021b. A framework for anomaly identification applied on fall detection. IEEE Access 9, 77264–77274.

Gomes, M.E.N., Macêdo, D., Zanchettin, C., de Mattos-Neto, P.S.G., Oliveira, A., 2022. Multi-human fall detection and localization in videos. Comput. Vis. Image Underst. 220, 103442.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial networks (2014), CoRR. 1406, arXiv preprint arXiv:1406.2661.

Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al., 2017. The "something something" video database for learning and evaluating visual common sense. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5842–5850.

Gutiérrez, J., Rodríguez, V., Martin, S., 2021. Comprehensive review of vision-based fall detection systems. Sensors 21 (3), 947.

Han, Q., Zhao, H., Min, W., Cui, H., Zhou, X., Zuo, K., Liu, R., 2020. A two-stream approach to fall detection with MobileVGG. IEEE Access 8, 17556–17566.

Inturi, A.R., Manikandan, V., Garrapally, V., 2023. A novel vision-based fall detection scheme using keypoints of human skeleton with long short-term memory network. Arab. J. Sci. Eng. 48 (2), 1143–1155.

Islam, M.M., Nooruddin, S., Karray, F., Muhammad, G., 2023. Multi-level feature fusion for multimodal human activity recognition in internet of healthcare things. Inf. Fusion 94, 17–31.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L., 2014. Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1725–1732.

Khraief, C., Benzarti, F., Amiri, H., 2020. Elderly fall detection based on multi-stream deep convolutional networks. Multimedia Tools Appl. 79 (27), 19537–19560.

Kwolek, B., Kepski, M., 2014. Human fall detection on embedded platform using depth maps and wireless accelerometer. Comput. Methods Programs Biomed. 117 (3), 489–501.

Le, H.-L., Nguyen, D.-N., Nguyen, T.-H., Nguyen, H.-N., 2022. A novel feature set extraction based on accelerometer sensor data for improving the fall detection system. Electronics 11 (7), 1030.

Li, S., Song, X., 2023. Future frame prediction network for human fall detection in surveillance videos. IEEE Sens. J.

Li, K., Wang, Y., Zhang, J., Gao, P., Song, G., Liu, Y., Li, H., Qiao, Y., 2022. Uniformer: Unifying convolution and self-attention for visual recognition. arXiv preprint arXiv:2201.09450.

Lu, N., Wu, Y., Feng, L., Song, J., 2018. Deep learning for fall detection: Three-dimensional CNN combined with LSTM on video kinematic data. IEEE J. Biomed. Health Inf. 23 (1), 314–323.

Martínez-Villaseñor, L., Ponce, H., Brieva, J., Moya-Albor, E., Núñez-Martínez, J., Peñafort-Asturiano, C., 2019. UP-fall detection dataset: A multimodal approach. Sensors 19 (9), 1988.

Mobsite, S., Alaoui, N., Boulmalf, M., Ghogho, M., 2023. Semantic segmentation-based system for fall detection and post-fall posture classification. Eng. Appl. Artif. Intell. 117, 105616.

Mohan Gowda, V., Arakeri, M.P., Raghu Ram Prasad, V., et al., 2022. Multimodal classification technique for fall detection of alzheimer's patients by integration of a novel piezoelectric crystal accelerometer and aluminum gyroscope with vision data. Adv. Mater. Sci. Eng. 2022.

Nooruddin, S., Islam, M., Sharna, F.A., Alhetari, H., Kabir, M.N., et al., 2021. Sensor-based fall detection systems: a review. J. Ambient Intell. Humaniz. Comput. 1–17.

Núñez-Marcos, A., Azkune, G., Arganda-Carreras, I., 2017. Vision-based fall detection with convolutional neural networks. Wirel. Commun. Mobile Comput. 2017.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in python. J. Mach. Learn. Res. 12, 2825–2830.

Ponce, H., Martínez-Villaseñor, L., 2020. Approaching fall classification using the up-fall detection dataset: Analysis and results from an international competition. In: Challenges and Trends in Multimodal Fall Detection for Healthcare. Springer, pp. 121–133.

Ponce, H., Martínez-Villaseñor, L., Nunez-Martinez, J., 2020. Sensor location analysis and minimal deployment for fall detection system. IEEE Access 8, 166678–166691.

Ramirez, H., Velastin, S.A., Aguayo, P., Fabregas, E., Farias, G., 2022. Human activity recognition by sequences of skeleton features. Sensors 22 (11), 3991.

Ramirez, H., Velastin, S.A., Cuellar, S., Fabregas, E., Farias, G., 2023. BERT for activity recognition using sequences of skeleton features and data augmentation with GAN. Sensors 23 (3), 1400.

Ramirez, H., Velastin, S.A., Meza, I., Fabregas, E., Makris, D., Farias, G., 2021. Fall detection and activity recognition using human skeleton features. IEEE Access 9, 33532–33542.

Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., Woo, W.-c., 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Adv. Neural Inf. Process. Syst. 28.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Smaira, L., Carreira, J., Noland, E., Clancy, E., Wu, A., Zisserman, A., 2020. A short note on the kinetics-700-2020 human action dataset. arXiv preprint arXiv:2010.10864.

Suarez, J.J.P., Orillaza, N., Naval, P., 2022. AFAR: a real-time vision-based activity monitoring and fall detection framework using 1D convolutional neural networks. In: 2022 14th International Conference on Machine Learning and Computing. ICMLC, pp. 555–559.

Taufeeque, M., Koita, S., Spicher, N., Deserno, T.M., 2021. Multi-camera, multi-person, and real-time fall detection using long short term memory. In: Medical Imaging 2021: Imaging Informatics for Healthcare, Research, and Applications, Vol. 11601. SPIE, pp. 35–42.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.

Waheed, M., Afzal, H., Mehmood, K., 2021. NT-FDS—A noise tolerant fall detection system using deep learning on wearable devices. Sensors 21 (6), 2006.

Wang, K., Cao, G., Meng, D., Chen, W., Cao, W., 2016. Automatic fall detection of human in video using combination of features. In: 2016 IEEE International Conference on Bioinformatics and Biomedicine. BIBM, IEEE, pp. 1228–1233.

Yadav, S.K., Luthra, A., Tiwari, K., Pandey, H.M., Akbar, S.A., 2022. ARFDNet: An efficient activity recognition & fall detection system using latent feature pooling. Knowl.-Based Syst. 239, 107948.

Yan, J., Wang, X., Shi, J., Hu, S., 2023. Skeleton-based fall detection with multiple inertial sensors using spatial-temporal graph convolutional networks. Sensors 23 (4), 2153.

Yu, M., Gong, L., Kollias, S., 2017. Computer vision based fall detection by a convolutional neural network. In: Proceedings of the 19th ACM International Conference on Multimodal Interaction. pp. 416–420.