Expert Systems With Applications 197 (2022) 116715

Contents lists available at ScienceDirect

Expert Systems With Applications

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

# RGB camera-based fallen person detection system embedded on a mobile platform

Sergio Lafuente-Arroyo, Pilar Martín-Martín *, Cristian Iglesias-Iglesias, Saturnino Maldonado-Bascón, Francisco Javier Acevedo-Rodríguez

*Alcalá University, Signal Theory and Communications Department, Alcalá de Henares, 28805 Madrid, Spain*

## ARTICLE INFO

*Keywords:*
Assistive robot
Fallen person detection
Face recognition
Object detector
Convolutional neural network
Support vector machine

## ABSTRACT

Most injuries in the elderly are due to falls. The response time, to attend to the critically injured in such fall cases, is crucial to their survival. This paper presents a low-cost, autonomous assistive patrol robot which additionally includes a fallen person detection module with facial recognition that allows identification of patients. Patrol robots could be beneficial for care centers, where there is a considerable number of patients that require care. In these conditions, falls can be generally detected by the robotic platform during the post-fall phase. This allows the system to work with no frame rate constraints, allowing other tasks to be run simultaneously. Based on the YOLO network, we propose two approaches for the fallen person detector. The first approach can differentiate between fallen persons and persons doing ordinary activity in a single stage, while the second is a two-staged approach. The network weights were obtained using a fine-tuning process by retraining with our own extended Fall Person Dataset (E-FPDS), which we release as a benchmark for other RGB vision-based approaches. Quantitative evaluations confirm that the detector performs robustly in detecting fallen persons in different situations. The results also show a recall of 98.97% in our test set.

## 1. Introduction

Recent developments in social science and technology have enabled using auxiliary systems with new tools to active aging and ease the burden of healthcare professionals in hospitals and daycare centers. Examples of such technologies that involve computer vision and robotics include a wide range of applications related to real-time assistance and humans' assistance, such as automatic fallen person detection. Fallen person detection systems (FPDS) have received a growing interest in recent years, and it is a crucial part of advanced assistive systems (Wang et al., 2020; Xu et al., 2018; Yusoff et al., 2021). Most injuries in the elderly are the result of falls; fractures of the hip, forearm, humerus, and pelvis usually result from the combined effect of falls and osteoporosis (Burns & Kakara, 2018). In such fall cases, the response time to attend to patients in a critical condition is crucial to their survival. Care center staff often do not see patients fall, especially when there are many patients. However, falls require immediate attention to reduce the risk of injury. Some authors have proposed patrol robots to help improve surveillance of elderly people and detect falls as soon as possible (Li et al., 2019; Menacho & Ordoñez, 2020; Tomoya et al., 2017).

Reliable systems can mitigate the negative consequences of falls and even save lives. Numerous efforts, presented in literature, have been made over the past years to detect fallen persons. Some examples can be found in Xu et al. (2018). However, in real applications, FPDS has always been a challenging task presenting challenges for understanding the scene using computer vision. The most important challenges are listed below:

- Complex dynamic environments: hospitals and care centers have complex indoor environment with varying illuminations, a variety of furniture, and people (patients, visitors, and personal staff).
- Perspective distortion: the projection of a fallen person on the image plane differs in terms of scale, orientation, and position. It depends on the camera motion and the possible motions of the fallen person, e.g., a person can be simply crawling, trying to find a phone, or trying get the system's attention.
- Activities: fall-lying pose is like other lying poses such as sleeping (common place in the elderly population). Contextual information is crucial for distinguishing between both activities in simple image processing.

* Corresponding author.
*E-mail addresses:* sergio.lafuente@uah.es (S. Lafuente-Arroyo), p.martin@uah.es (P. Martín-Martín), cristian.iglesias@uah.es (C. Iglesias-Iglesias), saturnino.maldonado@uah.es (S. Maldonado-Bascón), javier.acevedo@uah.es (F.J. Acevedo-Rodríguez).

https://doi.org/10.1016/j.eswa.2022.116715
Received 12 August 2020; Received in revised form 3 December 2021; Accepted 20 February 2022
Available online 28 February 2022
0957-4174/

- Occlusions: they are given when some elements of indoor structure, such as walls or pieces of furniture, occlude a fallen person partially. In this case, part of the body is missed in the image, making it more difficult.

It is hard to solve these problems with traditional image processing or machine learning methods. In recent years, deep learning techniques have been widely used in object detection. They provide an efficient way to extract features from images. Among these techniques, YOLO (You Only Look Once) (Redmon et al., 2016; Redmon & Farhadi, 2017) stands out for its simplicity and efficiency. YOLO can concurrently predict bounding box coordinates and associated class probabilities, and it achieves end-to-end target detection without a complex pipeline. Inspired by our previous fallen person detection algorithm (Maldonado-Bascón et al., 2019), in this work we propose two strategies based on YOLOv3 (Redmon & Farhadi, 2018) for fallen person detection. YOLOv3 algorithm performs well in object detection but does not consider the problem of a fallen person as a category in its training dataset (MS-COCO) (Lin et al., 2014).

Considering that the system must differentiate fallen persons from persons doing ordinary activities (standing up, sitting in a chair, lying on the sofa, walking, etc.), we tackle the problem with two different configurations. The two configurations differ on the number of stages (one or two) in the architecture. The two-stage configuration is composed of two blocks: (1) a Convolutional Neural Network (CNN) based on YOLO for detecting persons and (2) a Support Vector Machine (SVM) based stage that is capable of detecting fallen persons from among previously identified persons. To further enhance the extraction ability of CNN, the one-stage configuration tries to integrate both stages of the previous architecture into the YOLO network.

On the other hand, one of the main problems when comparing FPDS algorithms is the lack of large benchmark datasets as a standardized framework by the research community. Most of them are simulated in the laboratory to test the performances in the absence of real fall databases. In fact, in most studies dealing with fallen detection algorithms, authors create their own datasets with their particular characteristics: group of persons and predefined typologies of falls and activities. Unfortunately, these generated datasets are rarely published or made available. Therefore, the reproducibility of the tests and the cross-comparison with other algorithms becomes difficult. The datasets for fallen person detection can be separated mainly in four categories: (1) sensor-based databases which including wearable sensors (Casilari et al., 2017; Ferreira et al., 2020; Frank et al., 2010; Medrano et al., 2014; Sucerquia et al., 2017; Vavoulas et al., 2014; Vilarinho et al., 2015), (2) vision-based databases which including one or multiple RGB cameras (Auvinet, Rougier et al., 2010; Bosch-Jorge et al., 2014; Charfi et al., 2012; Mastorakis & Makris, 2014), (3) depth cameras such as Kinect (Ni et al., 2011; Wu et al., 2017; Zhang et al., 2014) and (4) multimodal databases which contain a combination of sensors and/or cameras (Chen et al., 2015; Kwolek & Kepski, 2014; Stone & Skubic, 2014). In vision-based datasets, images do not include critical situations for FPDS, such as lying/sleeping people, that can be easily confused with fallen persons. As many images are taken in a laboratory, there is a lack of variety in the image background, and many are taken from perspectives that are not suitable for robot patrolling. For all these reasons, in this paper, we release a large dataset, E-FPDS (Extended-Fallen People Dataset), which contains 6982 RGB annotated images with 5023 and 2275 fallen/non-fallen persons that try to solve the above mentioned problems. Each image is annotated with the rectangular regions of interest (ROIs) and the specific classes (persons and/or fallen persons).

Therefore, the main contributions of this study can be summarized by the following:

- A dataset of RGB images for fallen detection is collected and contributed by this paper. The primary intention to release this dataset is to compare different RGB vision-based algorithms, systems, and configurations fairly.
- The integration of an FPDS in a low-cost autonomous assistive robot to detect fallen persons in real-time and distinguish them from other daily activities.
- Two architectures based on YOLO are proposed for fallen person detection. In the first strategy, the features from the ROI detected by the YOLO network are provided as inputs to an SVM classifier, which decides if the person is in a fall or not. On the other hand, the second method uses a fine-tuning process by retraining YOLOv3 that detects a fallen person directly with a unique one-stage system. The results show that a suitable training of YOLOv3 can identify the fall-lying pose with very high recall and precision.
- As a complementary utility of the system, we introduce a face recognition stage to identify the fallen person. For this purpose, we explore how the face recognition algorithm from King (2017) needs to be adapted for lying postures under different viewing angles.

The paper is structured as follows: Section 2 focuses on a review of the related works. Detailed implementation of the framework is given in Section 3. The extended fallen person detection dataset, E-FPDS, is introduced in Section 4. Section 5 evaluates the performance of the proposed algorithms. Finally, we get conclusions and present future works in Section 6.

## 2. Related work

There is a lot of research on automatic fallen person detection algorithms to enable fast and proper assistance to the elderly. A complete monitoring system for the elderly suffering from Alzheimer's disease was proposed in Charlon et al. (2013). The most common methods consist of a combination of sensing and computing technologies to collect relevant data and develop algorithms that can detect falls based on those collected data; the paper (Mubashir et al., 2013) offers a survey of such systems. Depending on the nature of the employed sensors, fallen person detection systems can be classified into different groups. Igual et al. (2013) classify these systems into two generic groups: context-aware systems and wearable devices. Firstly, context-aware systems are based on sensors located in the environment around the user. Thus, these solutions typically integrate vision-based, and ambient-based systems (Mastorakis & Makris, 2014; Rimminen et al., 2010; Tzeng et al., 2010), including cameras, microphones, infrared, vibration sensors, and pressure sensors. The other type of fallen person detection system works with wearable sensors (Khan & Taati, 2017; Paoli et al., 2012). These systems employ accelerometers (and other mobility sensors) attached to the clothes or transported by the persons. Lately, sensors embedded in smartphones, smartwatches, and other portable devices have gained popularity due to their high affordability and global adoption. However, Mubashir et al. (2013) consider that vision-based and ambiance sensors should be in a different category. Zhang et al. (2015) make the classification based on the use of cameras so that the systems can be either non-vision-based or exclusively vision-based. Additionally, the vision-based devices can also be classified into three subcategories based on their type of camera: single RGB camera-based (Charfi et al., 2012; De Miguel et al., 2017; Hsu et al., 2015; Maldonado-Bascón et al., 2019), multiple RGB cameras-based (Auvinet, Multon et al., 2010; Cucchiara et al., 2007), and depth cameras-based (Lewandowski et al., 2017; Ni et al., 2011; Wu et al., 2017; Zhang et al., 2014).

Wearable sensors offer several advantages over other sensors in terms of cost, size, weight, ease to use, and, most importantly, portability. However, the significant advance is the privacy and independence of them respect the particularities of the environment. On the other hand, these devices have limited acceptance by the users for three main reasons: the discomfort in wearing them during normal daily activities, easy to forget wearing them, and their false-positive alarms. Ambient sensor-based approaches sense the pressure of everything around the person and generate too many false alarms (Principi et al., 2016). As an alternative to wearables, vision-based methods have emerged. The main advantage of vision based systems is that the person does not need to wear any devices. Furthermore, cameras provide a very rich set of information about the person's behavior, and their use is becoming more and more prevalent in everyday life.

Analyzing the three types of vision-based systems, the biggest problem of RGB cameras is occlusion. This problem can be solved if the camera is placed higher in the wall or ceiling to have a larger field of view. However, there is no access to the body's vertical motion in those cases, which also provides useful information for some fallen person detection approaches (Lee & Mihailidis, 2005; Nait-Charif & McKenna, 2004). Apart from RGB cameras, the use of RGBD cameras, like Kinect, for fallen person detection has increased because it can additionally obtain 3D information by tracking a person. However, RGBD cameras cannot cover an entire room because the resolution decreases in the depth image, hindering detection. Finally, there are multi-camera systems which have two main drawbacks: synchronization and calibration to compute reliable 3-D information, making the systems more difficult to implement than monocular ones. Determining human action in video scenes is another method of vision-based systems, as Olivieri et al. (2012).

The purely vision-based approaches focus on the frames of videos or images to detect falls. Features such as silhouettes or bounding boxes are extracted from the frames/images to facilitate detection in those approaches. Some approaches use those features as input for a machine learning classifier (Zerrouki et al., 2016) to detect falls such as Hidden Markov Models (HMM) (Anderson et al., 2006; Töreyin et al., 2005), Gaussian Mixture Models (GMM) (Rougier et al., 2011; Vishwakarma et al., 2007), K-Nearest Neighbor (Liu et al., 2010) and SVM (Charfi et al., 2012; Ebrahimi et al., 2017; Maldonado-Bascón et al., 2019; Schüldt et al., 2004; Zhang et al., 2006). As mentioned previously, several methods exist to detect falls with good detection rates.

These works demonstrate the importance of integrating real-time fallen person detection solutions to further robotic solutions for aiding patients. The solution should be as simple as possible. Ideally, we should avoid installing complex sensor networks at centers or over the user's body. Furthermore, there is no doubt that cost-effectiveness must be considered since the provisioning of the robotic platform to the user usually results in an expense paid by families or health services. In the context of our application, which is an assistive patrol robot (APR) collecting interesting information as it goes, we formulate the problem of fall detection as a fall-lying pose recognition and not as a process of falling. The reason is the difficulty of capturing the person during the fall when the robot is on patrol. Since our detector system is triggered by each captured frame and the inter-frame information is not considered, a fallen person is generally detected in several consecutive frames. That is, we get redundant information to provide an alarm. Our APR has the following properties:

- The use of an RGB camera as a sensor for the detector instead of wearable sensors.
- It is robust enough to differentiate falls from other similar activities, such as lying down on a sofa or bed.
- It is capable of navigating in an environment.
- It is a simple and low-cost system, which will help increase accessibility to communities that require it.
- It provides face recognition as a complementary utility of the fallen person detection. However, face recognition can be difficult, depending on the pose of falling.
- It provides an alarm in case of detecting a fallen person. The generated information includes the localization of the robot, which is, in this case, close to the detected fallen person.

Therefore, according to the previous characteristics, this paper's main objective is to present a fallen person algorithm embedded on an autonomous APR to be used in elderly care centers. However, it could also be applied in other centers for children, youth, or adults, who may also benefit from FPDS, especially in persons with functional disabilities.

## 3. Assistive robot for fallen persons detection

This section presents the main parts of the proposed mobile assistive robot, including an embedded system with an application to fallen person detection. To make this application possible, it was necessary to use different types of hardware that facilitated the acquisition of data and software tools that analyzed the information sent by those devices. This way, mixing these technologies was possible to patrol around the place, detect falls, and provide an alarm with the person's identification if the face is visible.

In the beginning, the system architecture is outlined. Next, a fallen detection algorithm based on person detection with a fine-tuning YOLOv3 and fall classification using SVM is presented. Following, an alternative method based on fallen detection using a fine-tuning YOLOv3 with two classes is explained. Finally, we describe a complementary utility based on an adapted face recognition algorithm for identifying the fallen person.

### 3.1. Robotic platform

As a base for the fallen person detection approach, we use the last updated version of our autonomous assistance robot, "LOLA" (Fig. 1). LOLA has been designed entirely by our research team to monitor and help users. It is a differential wheeled robot, equipped with two motors and their corresponding encoders, all controlled with an open-source Arduino Mega board. We have performed all mechanical and electrical designs. Additionally, the outer shell, imitating a person wearing a tuxedo, was made entirely by 3D printing. Two batteries power the platform, and it includes an electronic driver interface to allow easy interconnection of the different parts of the system and all the power management. The complete platform measures approximately 800 mm, slightly higher than a table.

As for the sensors, the platform has the following: a LIDAR, a touch screen, and a frontal camera. The images are acquired using an RGB camera with a wide-angle and image size of 640x480 pixels. In order to integrate into the mobile robot all the high-level processing that cannot be embedded into the Arduino, the platform has a Jetson TX2 board from NVIDIA, including visual perception with the FPDS module and navigation. Fig. 2 shows the overview of the system architecture where all the different modules are separated for a more comprehensible structure.

### 3.2. Proposed architectures

In this work, we implement a vision-based fallen detector in an APR based on a YOLO network. The YOLOv3 model runs a deep learning CNN on an input image to produce network predictions. The CNN consists of 106 layers, including successive $3 \times 3$ and $1 \times 1$ convolutional layers, shortcut connections, up-sample layers, route layers, and detection layers. It is important to point out that even when the original YOLO is trained using the COCO dataset (Lin et al., 2014) with the "person" class, it does not have the capability of discriminating a fallen person against other human postures. Inspired by this idea, we propose two strategies to solve the problem:

- YOLO+SVM (called TDSVM-fall): this architecture (see Fig. 3) consists of two stages: (1) a CNN based on YOLO for detecting all persons in the image independently of their postures, and (2) an SVM for detecting fallen persons among the persons extracted previously. Here, we ran a fine-tuning process retrained YOLO for detecting people using PASCAL VOC (2007&2012) and our

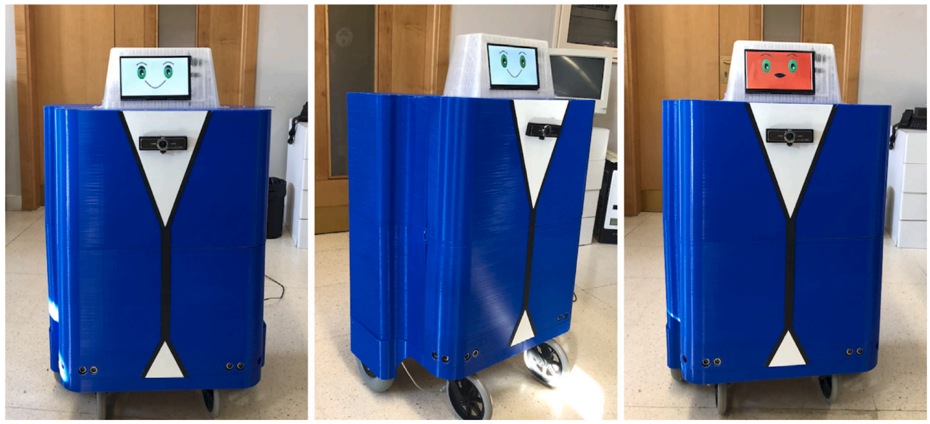


**Fig. 1.** Pictures of our low-cost autonomous robot "LOLA".

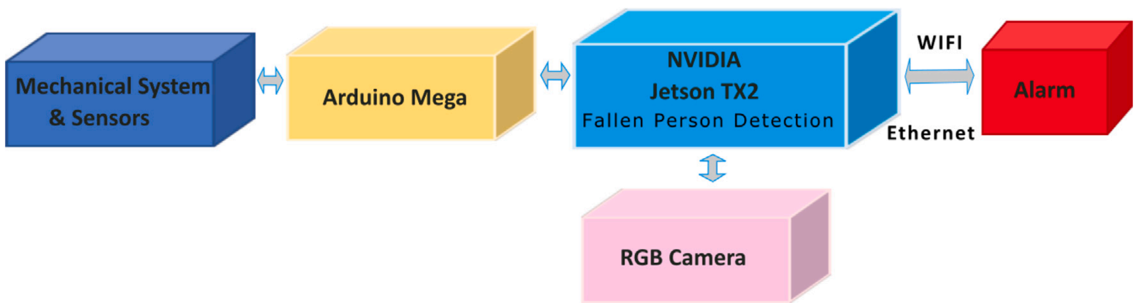



**Fig. 2.** Overview system architecture.

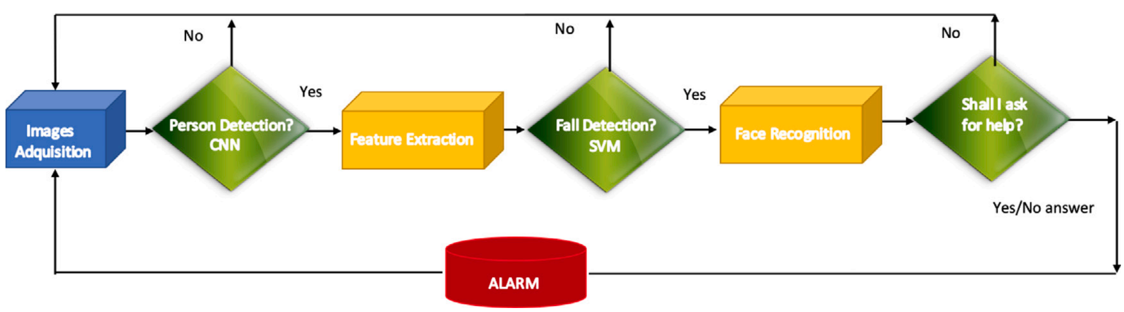



**Fig. 3.** Flowchart of the TDSVM-fall detection algorithm.

own dataset E-FPDS, which contains mainly fallen person images. Thus, the network worked with a unique class ("person" class) that considers all different human poses and activities. The main difference of this strategy concerning our previous work (Maldonado-Bascón et al., 2019) is that now we have re-trained the network to make it robust to changes of human poses, instead of correcting the orientation of the input images to the CNN in Maldonado-Bascón et al. (2019). In the previous work, cross-validation was performed on the training set to find the optimal parameters (the $C$ upperbound and kernel parameters) of a Radial Basis Function (RBF) kernel SVM. The SVM to be used in this approach is built upon that work.

- YOLO (called TD-fall method): this architecture (see Fig. 4) implements a two-class detector (person and fallen person classes) based on another fine-tuning process without a posterior need of an additional SVM classifier. The whole network can be end-to-end trained. The capacity of discrimination between persons and fallen persons is entirely transferred to the CNN, and the output of YOLO provides the detections when the robot is on patrol. The ROIs annotated as a non-fallen person in the datasets for the fine-tuning process include many different situations, such as standing, lying, or sitting.

As we can see in Figs. 3 and 4, an image is captured with the RGB camera and is given as input to the detection stage. If it detects a fallen person, the ROI is forwarded to the face recognition system, triggers an alarm, and generates a voice message to interact with the user asking if needs help. The algorithm provides information about the fallen person's localization and his/her identification when is possible.

### 3.3. Face recognition

In a patrol robot with an embedded FPDS, it is important to identify the subject in risk. In FPDS, the alignment of faces is the main problem to overcome because faces may appear downward, sideways, or occluded (if the person is on his/her back) in the plane image. Other difficulties to face are the small size of the face in the image when the camera is far from the target and the blurring caused by the possible motions of the fallen person.

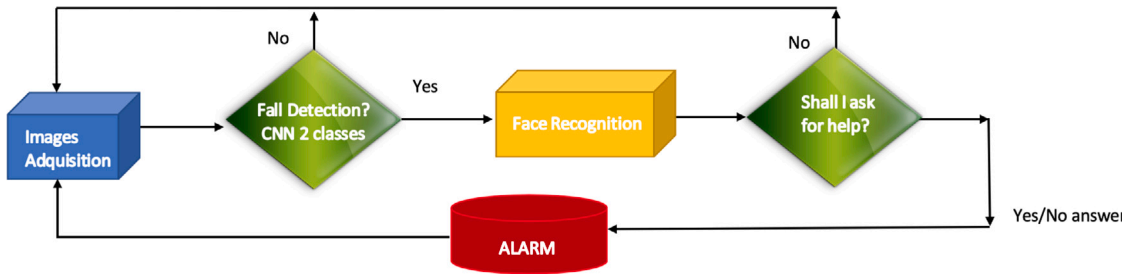



**Fig. 4.** Flowchart of the TD-fall detection algorithm.

CNNs have received increasing interest in face recognition, and several deep learning methods have been proposed. Guo and Zhang summarize them and make a concise overview of face recognition's main problems in Guo and Zhang (2019). Our face recognition module is based on the open-source *face_recognition package* (King, 2017), which was written in C++ and built using dlib's state-of-the-art face recognition. The pipeline consists of face detection, projecting faces, encoding faces, and finding the person from the encodings. The basic idea of encoding is to quantify the face with a vector of measurements. In this way, identifying someone over a registered database of people involves identifying the person with the closest measurements to the test image.

In our FPDS, the primary challenge is face detection robust enough to multiple rotations. In general, heads of fallen people on the floor appear with rotations closed to $\pm 90^{\circ}$ concerning conventional situations where a person is walking, standing up, or sitting. From our experiments with the mentioned library, we can confirm that the performance of the *face_recognition package* deteriorates with rotations higher than approximately 45°. As a solution to adapt the library *face_recognition* for lying postures, we feed the face recognition module with three inputs: the original ROI and the rotated versions corresponding to $\pm 90^{\circ}$. In this way, we apply facial recognition to the three versions of the original ROI to solve the rotation dependency. Although facial recognition is applied three times by each detected fallen person, it is important to consider the reduced computational load of the process because the facial recognition module is only triggered when a fallen person is detected.

Fig. 5 represents the flowchart of our proposal, where each one of the three ROIs (the original ROI detected and the both rotated versions) is processed independently. The faces localized by the detection block are fed to the face encoding stage, which returns a 128-length vector for each detection. Thus, vectors $\mathbf{f}_1$, $\mathbf{f}_2$ and $\mathbf{f}_3$ denote the encodings of the original and $\pm 90^{\circ}$ rotated versions, respectively. Then, the vectors of distances $\mathbf{d}_1$, $\mathbf{d}_2$ and $\mathbf{d}_3$ to the face prototypes of our dataset are calculated. Finally, the last stage of the face recognition algorithm identifies the person with the lowest distance whenever this distance is higher than a fixed threshold. In the example of Fig. 5, the system detects and recognizes the face only in the ROI clockwise rotated. However, some faces may be detected in more than one orientation even when the correct detection is given only by one of the three. For that reason, we include the block "combination of recognitions". It identifies the person as the one that corresponds to the index with the lowest distance in any of the three distance vectors.

Fig. 6 depicts three examples of our proposed algorithm in which the faces are detected: in the counterclockwise and clockwise rotations for the first two images, respectively, and in the original orientation for the last one. The name of each identified person is written above the corresponding ROI, which is represented with a green box.

## 4. A fallen people dataset: E-FPDS dataset

A fundamental problem in the analysis and comparison of the different fallen person detection algorithms is the lack of public datasets with a large number of people in lying-positions even in the large datasets (Antonello et al., 2017; Auvinet, Rougier et al., 2010; Charfi et al., 2013; Debard et al., 2016; Igual et al., 2015; Kwolek & Kepski, 2014; Martínez-Villaseñor et al., 2019). Additional limitations of these datasets are related to the lack of a wide range of situations such as several scenarios, fallen persons with many different poses and appearance. On the other hand, persons in resting positions (lying down on the sofa or in bed) are critical situations to discriminate against fallen persons.

To consider the above problems in the analysis of algorithms, we collected our own dataset to evaluate the performance of our proposed FPDS framework. We captured a significant number of images with different subjects simulating falls and other conventional activities to construct the dataset. Images were taken using an RBG camera mounted on our robotic platform and present a resolution of $640 \times 480$ size. In addition to capture images in different scenes and places, we also collected images in different illumination conditions (indoor and outdoor scenes). We will use the term fallen person instance to refer to a physical fallen person on the floor. Thus, each fallen person instance is captured in several images with different poses as the robot moves. The main features of this dataset are:

- Several scenarios with variable light conditions,
- Subjects with different ages and appearance,
- Images with one or several subjects,
- Images with fallen persons and resting persons.

E-FPDS dataset[1] includes images and their corresponding annotation files. Each $i$th ROI is characterized by its corresponding label $y_i$ ($y_i = 1$ for fallen person, $y_i = -1$ for non-fallen person) and its bounding box $b_i$ defined by the coordinates $\{x_i^{left}, x_i^{right}, y_i^{top}, y_i^{down}\}$.

We presented the first version of our FPDS dataset in Maldonado-Bascón et al. (2019). However, in this new work, we release an extended version of our dataset (called E-FPDS dataset) with a total of 6982 images with 5023 and 2275 annotated fallen and non-fallen persons, respectively grouped in 13 splits. The non-fallen persons are in different activities such as standing up, sitting in a chair, lying on the sofa, and walking. The dataset is composed of a training set, a validation set, and a test set. Tables 1–3 summarize the features of the E-FPDS database. The training dataset is used to fit the parameters (weights) of the network, whereas the validation set is used to tune the hyperparameters (i.e., the architecture) of the network. The test dataset provides an unbiased evaluation of a final model fit on the training dataset.

We built the training set with splits 1, 2, 3, 10, and 11, comprising 4808 images, including 3867 falls and 1005 non-falls. The test set includes splits from 4 to 8, with 391 falls and 830 non-falls in a total number of 973 images. Finally, the validation set comprises splits 12 and 13 with 765 falls and 440 non-falls in a total of 1201 images.

[1] E-FPDS dataset including the ground-truth bounding boxes annotations and the final optimized weights of ours trained models are public and available at https://gram.web.uah.es/data/datasets/fpds/index.html.

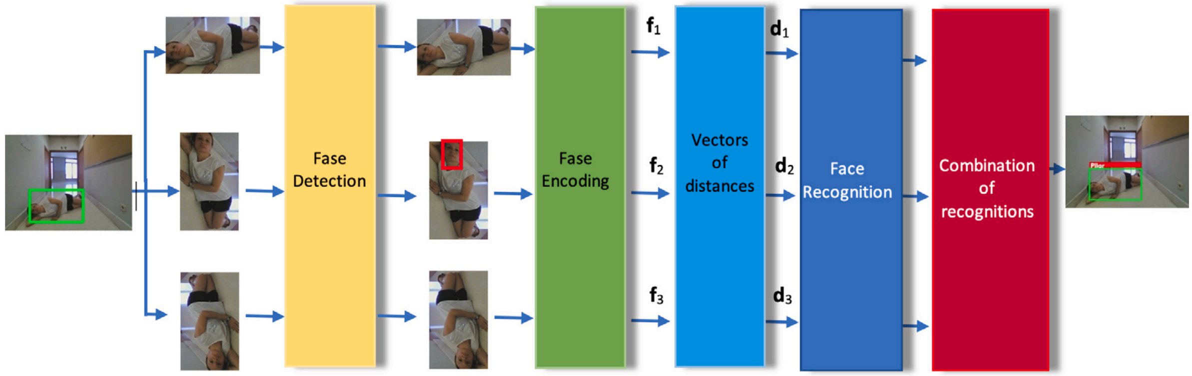



**Fig. 5.** Overall modified face recognition algorithm.

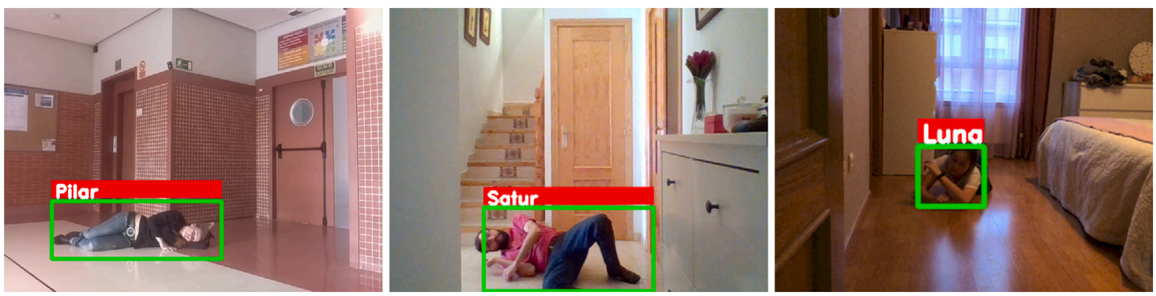



**Fig. 6.** Examples of output results from the proposed face recognition module.

**Table 1**
E-FPDS dataset — Training set.

| | Split 1 | Split 2 | Split 3 | Split 10 | Split 11 | Total |
|---|---|---|---|---|---|---|
| Number of images | 800 | 646 | 732 | 416 | 2214 | 4808 |
| Number of falls | 556 | 449 | 366 | 282 | 2214 | 3867 |
| Number of non-falls | 358 | 169 | 352 | 126 | 0 | 1005 |

**Table 2**
E-FPDS dataset — Validation set.

| | Split 12 | Split 13 | Total |
|---|---|---|---|
| Number of images | 614 | 587 | 1201 |
| Number of falls | 614 | 151 | 765 |
| Number of non-falls | 4 | 436 | 440 |

**Table 3**
E-FPDS dataset — Test set.

| | Split 4 | Split 5 | Split 6 | Split 7 | Split 8 | Total |
|---|---|---|---|---|---|---|
| Number of images | 117 | 553 | 42 | 51 | 210 | 973 |
| Number of falls | 104 | 49 | 42 | 15 | 181 | 391 |
| Number of non-falls | 3 | 704 | 0 | 39 | 84 | 830 |

Regarding the distribution of splits, the percentages of images from the whole dataset were 68.86% images for training, 13.93% images for testing, and 17.21% images for validation.

## 5. Experiment results

### 5.1. Metrics

To evaluate the performance of the proposed algorithms, we utilize precision and recall parameters (Charfi et al., 2013; Sokolova & Lapalme, 2009) as metrics. Both parameters are defined as:

$$Pr = \frac{TP}{TP + FP}; \tag{1}$$

$$Re = \frac{TP}{TP + FN}, \tag{2}$$

being $TP$ and $TN$ the true positives and negatives, respectively, and $FP$ and $FN$ the false positives and negatives, respectively. Precision ($Pr$) and recall ($Re$) work in different directions in the sense that improving precision typically reduces recall and vice versa. A robust FPDS requires a trade-off between a high recall (most fallen persons are detected) and a high precision (the number of false alarms is reduced).

To compare our system's outcomes with the ground-truth of annotated images, we adopted the Intersection over Union (IoU) criterion, and detections having a certain overlap with an annotated fallen person are considered as true positives. Unlike if the detection does not have enough overlap with any annotation, it is computed as a false positive. In order to set a threshold for IoU, $IoU_{th}$, we analyzed how this value affects the precision and recall parameters in paper (Maldonado-Bascón et al., 2019). It was observed that metrics values were almost independent of the selected threshold, setting in that case, that value to $IoU_{th} = 0.2$. We will keep this value for the proposed approaches.

In the first experiment, we evaluated the performance of the overall end-to-end fallen detection algorithms: TDSVM-fall and TD-fall, respectively. To further evaluate our proposed framework's performance, we conducted a set of comparative experiments on two other publicly released datasets. Finally, we tested the dependency of proposed methods with the camera's position to extend their usefulness to other possible applications.

### 5.2. Experiment 1: Comparison of proposed methods

In this first experiment, we aim to evaluate the performance of the proposed approaches: TDSVM-fall and TD-fall methods. The weights of the pre-trained convolutional layers are used as initial values. The new weights are obtained after a fine-tuning process by retraining the network with three datasets: our new E-FPDS in addition to PASCAL VOC 2007&2012.

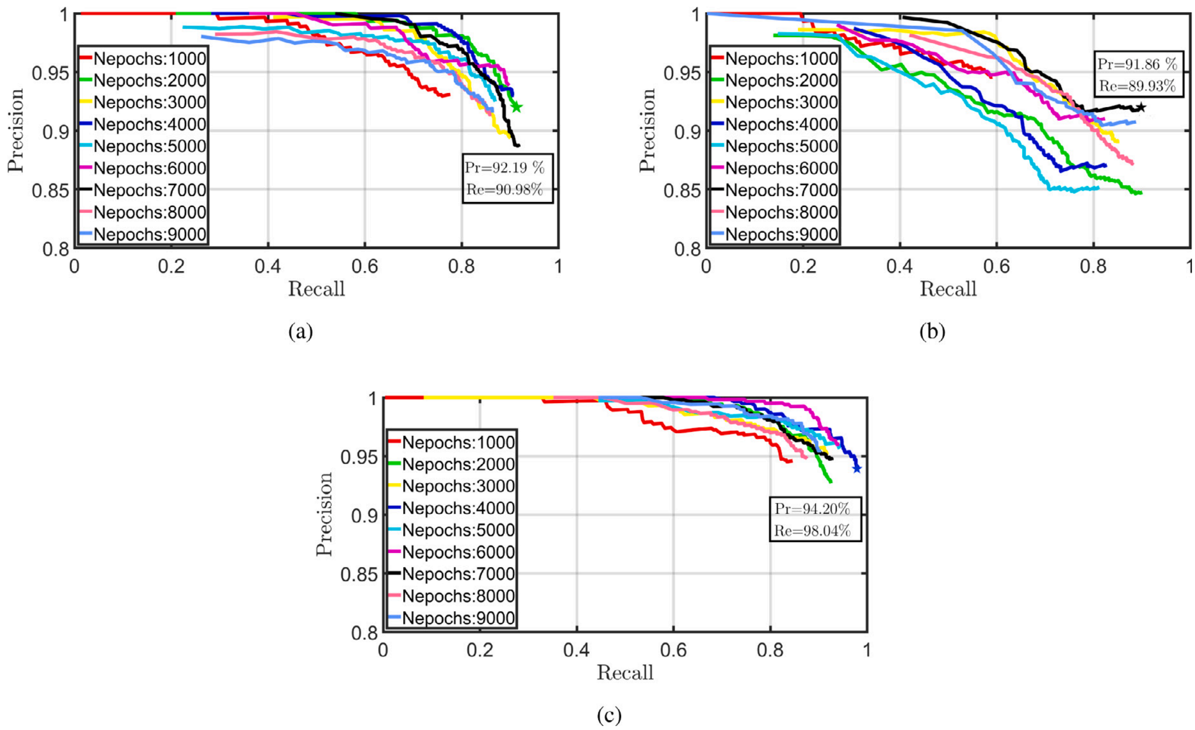



**Fig. 7.** Precision–recall curves over validation set for different number of epochs. (a) TDSVM-fall (20 classes). (b) TDSVM-fall (1 class). (c) TD-fall (2 classes).

Regarding the proposed methods, the number of classes to manage is one (person) and two (fallen person and non-fallen person) for TDSVM-fall and TD-fall, respectively. In the TDSVM-fall approach, YOLO was trained to detect only the "person" class, including many different situations from fallen and non-fallen persons (note that the SVM classifier is fed with the detections of YOLO for detecting fallen persons among the persons extracted previously). However, in order to analyze the ability of the network to transfer learning from a large source domain to a small target domain, we generated an additional model of a network in the TDSVM-fall with the 20 classes of PASCAL VOC (e.g., car, bus, TV, chair, etc.). The objective is to compare the behavior of the system when the number of classes to detect varies. To verify the effect of training, we select different training stages and visualize their impact. Here, we chose to train from 1000 to 9000 epochs with steps of 1000 for comparison.

Fig. 7 shows precision–recall curves on the validation set for the two approaches to determine the optimum values of the hyperparameters (confidence threshold, $C_{th}$, and the number of epochs $N_{epochs}$ in training). The colors of the curves depict the results using the different weights. Extreme points of curves with the highest recall correspond to low values of $C_{th}$, whereas the lowest recall points correspond to the highest values. Since non-detection of a fallen person has more serious implications than a false alarm, we do not give the same importance to precision and recall for selecting the best combinations of hyperparameters. Thus, we determine the optimum values of hyperparameters that provide a high recall even when we try to maintain a high precision as possible. From a detailed analysis of precision–recall curves, we selected as best results the obtained for $N_{epochs} = 2000$ (TDSVM-fall 20 classes), $N_{epochs} = 7000$ (TDSVM-fall 1 class) and $N_{epochs} = 4000$ (TD-fall 2 classes) with $C_{th} = 0.2$ in all cases. Values of precision and recall of selected optimum points (marked with a star symbol) are annotated in Figs. 7(a), 7(b) and 7(c). The weights of our optimized models are public and available at the same site URL as the E-FPDS dataset.

Using the optimum hyperparameters, we evaluate the performance of the proposed approaches on the test dataset and compare our results with the previous ones that we obtained in paper (Maldonado-Bascón et al., 2019), where input images to CNNs were corrected previously to eliminate the possible changes of the pose. Table 4 summarizes the results, and it is observed that the TDSVM-fall approach outperformed the previous approach (Maldonado-Bascón et al., 2019) in terms of precision. The precision rate increased by 4.51% and 3.44%, using TDSVM-fall (20 classes) and (1 class), respectively. However, in the recall case, the TDSVM-fall (1 class) increases the recall by 2.56% while it decays 1.03% in the TDSVM-fall (20 classes). Our interpretation of these results in the TDSVM-fall (20 classes) approach is based on the overall loss function optimization because optimization considers the 20 classes and not only the "person" class. However, since our primary goal is getting a high value of recall, the TD-fall method achieves a very high recall rate of 98.97% with only 4 FN and a precision rate of 93.93%. Using that criterion, we conclude that the TD-fall method gets the best performance. Fig. 8 shows the precision–recall curve of the TD-fall approach over the test set by varying the $C_{th}$ with the selected number of epochs $N_{epochs} = 4000$.

**Table 4**
Comparison of methods on test set.

| | TP | FN | FP | Pr (%) | Re (%) |
|---|---|---|---|---|---|
| Fall detection with pose correction (Maldonado-Bascón et al., 2019) | 360 | 31 | 17 | 95.49 | 92.07 |
| TDSVM-fall (20 Classes) | 356 | 35 | 0 | 100 | 91.04 |
| TDSVM-fall (1 Class) | 370 | 21 | 4 | 98.93 | 94.63 |
| TD-fall | 387 | 4 | 25 | 93.93 | 98.97 |

Results over the test set (Table 4) confirm that the TD-fall approach outperforms the TDSVM-fall approach in the recall parameter. However, it shows a lower precision rate. To analyze that behavior, we have explored the 25 FPs in the test images. We found out that the FPs are caused mainly for three situations:

- 10 FPs (see Fig. 9(a)): a leg of a chair was detected as a fallen person in several images.

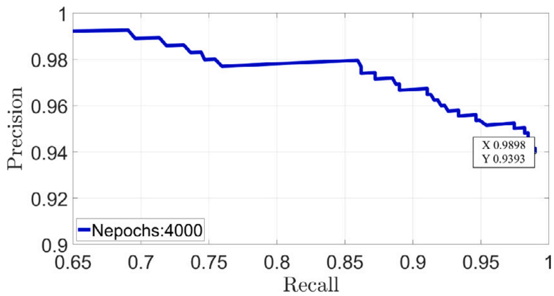



**Fig. 8.** Precision–Recall curve for TD-fall over the test set for *nepochs* = 4000.

- 8 FPs (see Fig. 9(b)): persons in the process of falling were annotated as non-fallen persons (we consider a fallen person only when the person is lying on the floor).
- 7 FPs (see Fig. 9(c)): persons that appear in the scene are surrounded by objects of similar color to clothes.

The optimized TD-fall method was adopted as a solution, and it was integrated into our assistive patrol robot for fallen person detection. A demo video is provided on the site.[2] The runtime of this approach in the NVIDIA® Jetson with 8 GB RAM was approximately on average 0.27s. This time allows detection in real-time, given an update rate of 3 frames per second.

### 5.2.1. Analysis of the confidence score

In order to analyze the influence of the confidence score $C_s$ at the output of YOLO in the final decision, we obtained the histograms of $C_s$ for the two approaches on the validation set (see Fig. 10). We have divided the range of values into 20 uniform intervals from 0 to 1. Histograms are built by computing the frequency normalized of TPs and FPs, being the frequency in each $i$th interval the number of TPs and FPs with a confidence score that falls in the range of the considered interval. As we expected, almost all the TPs were detected with a high confidence score close to 1. However, in FPs, their confidence distribution does not follow a definite shape indicating that FPs were detected with a wide range of values, except in the TDSVM-fall approach with a unique class. In this last case, there exists a considerable concentration of FPs with high values. The trade-off between a high number of TPs and a low number of FPs with the priority of no missing TPs led us to select a confidence threshold of $C_{th} = 0.2$.

Figs. 11 and 12 represent examples of detections with the TD-fall approach corresponding to extreme values of confidence scores. Thus, Fig. 11 includes both TPs corresponding to the lowest and the highest scores, with values of $C_s = 0.24$ (fallen person partially occluded) and $C_s = 1.0$ (a fallen person easy to detect), respectively. In addition, examples of Fig. 12 correspond to the FPs with the lowest and the highest scores, with values of $C_s = 0.25$ (shoes on the floor that the system confuses with the target) and $C_s = 0.94$ (fallen person lying on a bed). Note that bounding boxes of TPs and FPs are depicted with green and red colors, respectively, in the outcome images.

### 5.2.2. Analysis of non-max suppression

YOLO applies Non-Maximum Suppression (NMS) technique to avoid the problem of multiple detections per object. It joins the ROIs with high IoUs and selects the ROI with the higher confidence score. Therefore, the IoU parameter of YOLO needs to be optimized. Fig. 13 shows the number of FPs and FNs for different values of the IoU threshold, $IoU_{nms}$. As we can observe, the number of FNs is almost not affected by this threshold, whereas the number of FPs shows a strong dependency on the IoU threshold, $IoU_{nms}$. That is, the number of FPs increases notably as the value of $IoU_{nms}$ grows. These results have been obtained on the validation set, and we have fixed $IoU_{nms} = 0.3$ for all methods because higher values provide a relevant increase in the number of FPs. Note that the meaning of $IoU_{nms}$ of the non-max suppression technique is different from the $IoU_{th}$ for the metric of evaluation.

[2] https://gram.web.uah.es/data/datasets/fpds/index.html.

**Table 5**
Results on the IASLAB-RGBD and URFD datasets.

| | | TP | FN | FP | Pr (%) | Re (%) |
|---|---|---|---|---|---|---|
| IASLAB-RGBD | Fall detection with pose correction | 271 | 92 | 53 | 83.69 | 74.72 |
| | TDSVM-fall (20 classes) | 257 | 104 | 108 | 70.41 | 71.19 |
| | TDSVM-fall (1 class) | 295 | 66 | 36 | 89.12 | 81.71 |
| | TD-fall (2 classes) | 301 | 60 | 3 | **99.01** | **83.37** |
| URFD | Fall detection with pose correction | 884 | 19 | 149 | 85.57 | 97.89 |
| | TDSVM-fall (20 classes) | 884 | 19 | 305 | 74.34 | 97.89 |
| | TDSVM-fall (1 class) | 847 | 56 | 258 | 76.65 | 93.79 |
| | TD-fall (2 classes) | 885 | 18 | 0 | **100** | **98.00** |

## 5.3. Experiment 2: Other datasets

In this experiment, we evaluated the performance of the two proposed algorithms with other datasets. This experiment supposes proof of the capacity to generalize our approaches to other scenarios not seen previously. Specifically, we decided to use two public datasets: the Intelligent Autonomous Systems Laboratory Fallen Person Dataset (IASLAB-RGBD) ([Dataset] Department of Information Engineering-University of Padua, 2017) and the UR-Fall Detection Dataset (URFD) ([Dataset] Interdisciplinary Centre for Computational Modelling-University of Rzeszow, 2014):

- IASLAB-RGBD: This dataset was presented in Antonello et al. (2017) and generated using a Microsoft Kinect One V2 camera mounted on a mobile robot. We used a dataset with 374 images, including 361 falls and 133 non-falls.
- URFD: This dataset was presented in Kwolek and Kepski (2014) and contains 70 sequences (30 falls + 40 activities of daily living). Fall events were recorded with 2 Microsoft Kinect cameras. For this experiment, we used only the 30 fall sequences with 2103 images, including 903 falls and 1199 non-falls.

Both sets are close enough to our dataset due to the height of the camera at which the original images were taken (see examples of images in Fig. 14). We created the needed ground-truth bounding boxes because neither of both datasets includes available annotations of ROIs. Results are shown in Table 5 and we compare with our previous contribution (Maldonado-Bascón et al., 2019). As we can observe, the TD-fall method exhibits superior performance in new scenarios. It can achieve precision rates of 99.01% and 100% and recall 83.37% and 98% in IASLAB-RGBD and URFD datasets, respectively. Worth noting, the FNs, resulting in using this approach, are the lowers ones, an essential issue for our proposed methods.

The SVM classifier caused most errors in the TDSVM-fall. It should be noted the high number of FPs in contrast to the TD-fall approach. However, from the inspection of the images, we observe that most of the FPs are due to the same fact. In both scenarios, there is an object that causes many FPs: a sofa in the IASLAB-RGBD dataset (see Figs. 15(a) and 15(b)) and a backpack in the URFD dataset (see Figs. 15(c) and 15(d)). The backpack's appearance is similar to a fallen person, and the confusion of the sofa can be due to the training images with persons lying on the sofa. In the TDSVM-fall approach (1 class), 21 FPs of the 36 FPs (58,33%) and 258 FPs from 258 FPs (100%) are due to the confusion of the mentioned objects with fallen persons. However, using TDSVM-fall (20 classes), which exhibits worse behavior, the method has 104 FPs of the 108 FPs (94.75%) and 289 of the 305 FPs (94.75%) due to the presence of these two objects. As reflected by Table 5, we can conclude that the TD-fall approach,

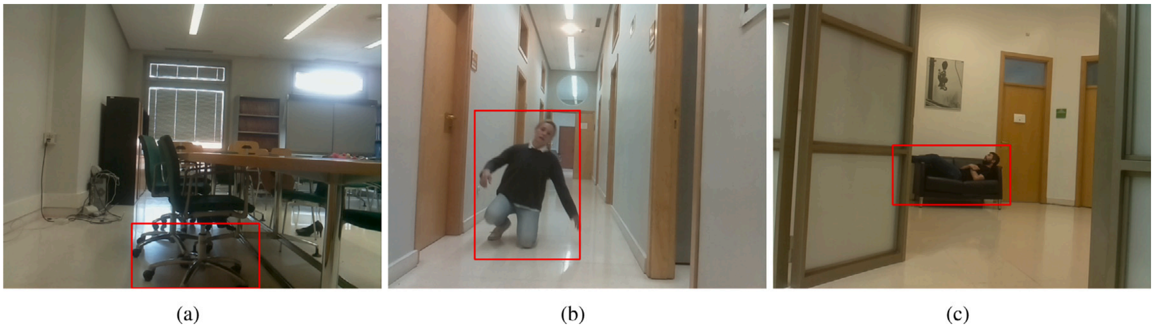



**Fig. 9.** Examples of FP with the TD-fall method. (a) Wrong identification. (b) Middle fall situation. (c) Similar color objects.

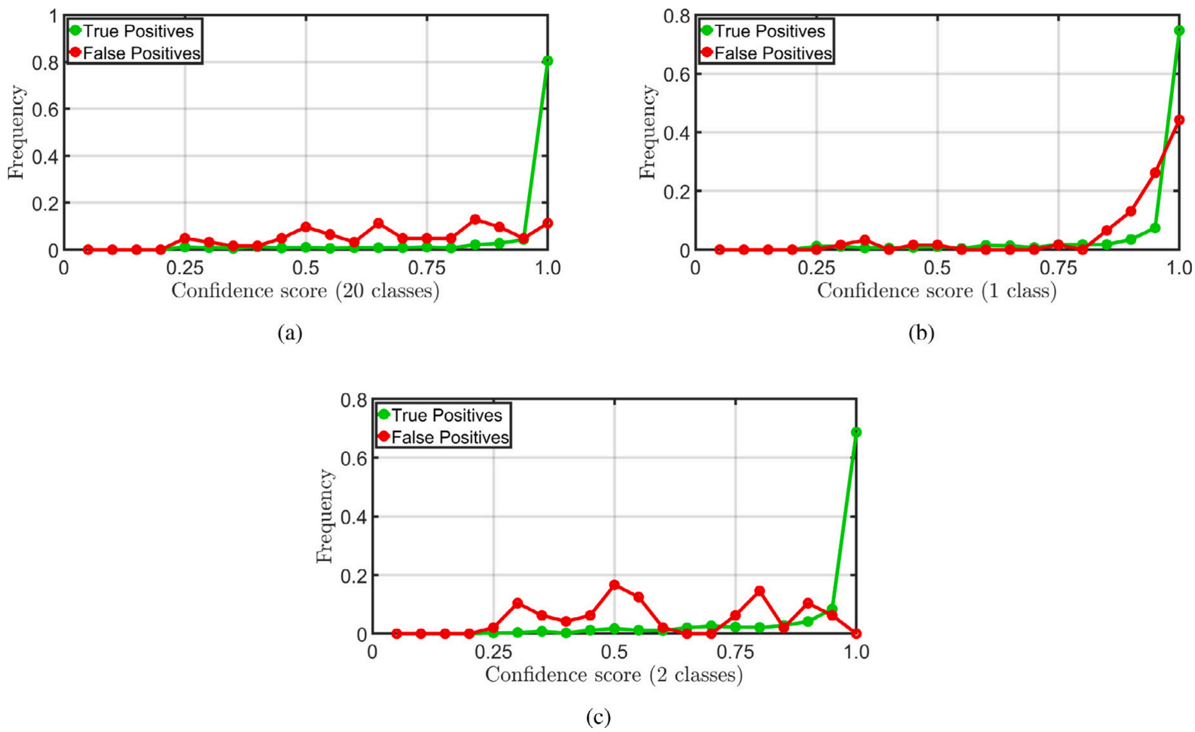



**Fig. 10.** Histogram of the confidence score $C_s$ on the validation set. (a) TDSVM-fall (20 classes). (b) TDSVM-fall (1 class). (c) TD-fall (2 classes).

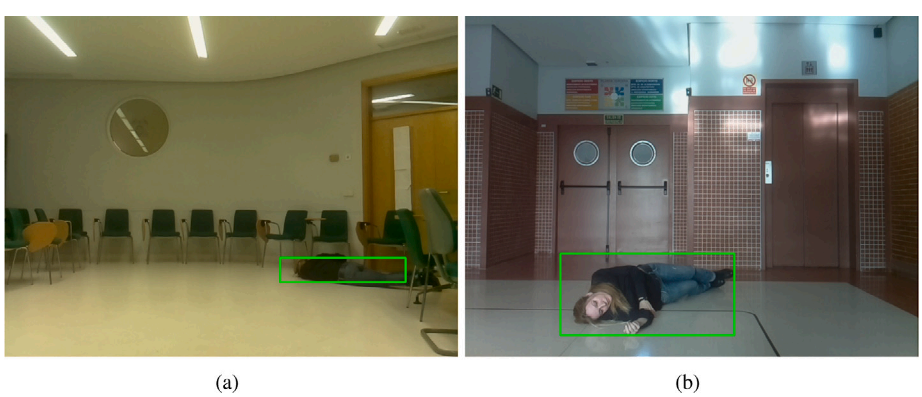
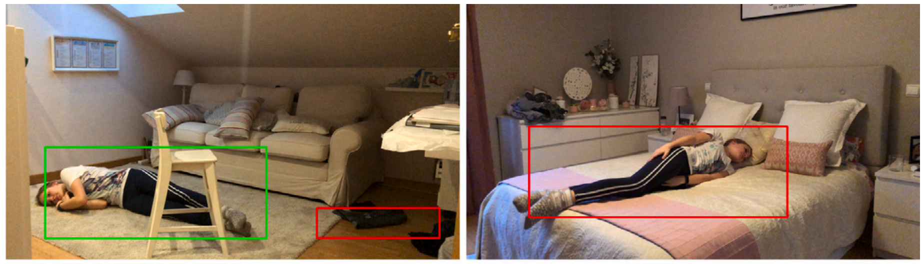



**Fig. 11.** Confidence score of some TPs examples in the TD-fall method. (a) $C_s$ = 0.24. (b) $C_s$ = 1.0.

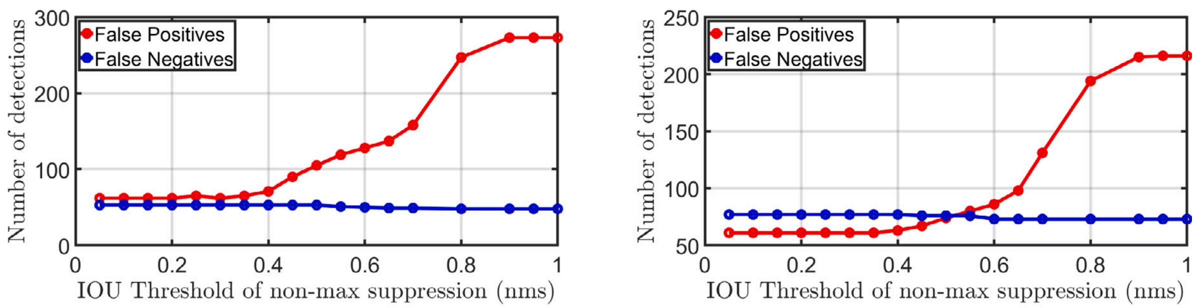
which discriminates fallen persons against other human poses, shows a superior generalization concerning the remaining approaches.

As a third set to evaluate the performance of our proposal, we created a complementary dataset (Elderly set) with volunteers over 65 years old in different indoor scenarios, which is available, too. Since the elderly collective is one of the most vulnerable when it comes to accidents, our motivation is to integrate the proposed methods in our patrol robot for detecting fallen elderly in care centers. Our elderly set comprises 413 annotated images in different situations (conventional activities and fallen persons) of the same size as images of the E-FPDS

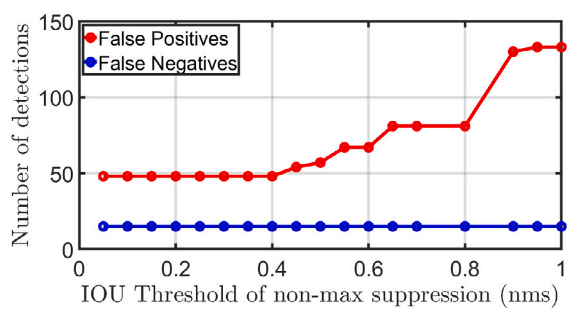


(a) (b)

**Fig. 12.** Confidence score of some FPs examples with the TD-fall method (TP in green and FP in red). (a) $C_s$ = 0.25. (b) $C_s$ = 0.94.

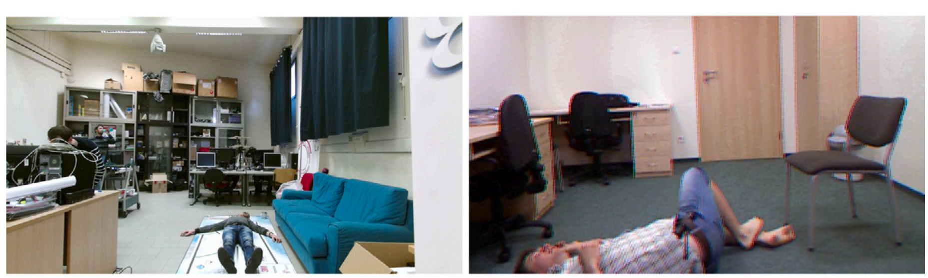



(a) (b)




(c)

**Fig. 13.** Non-Max Suppression technique. (a) TDSVM-fall (20 classes). (b) TDSVM-fall (1 class). (c) TD-fall (2 classes).

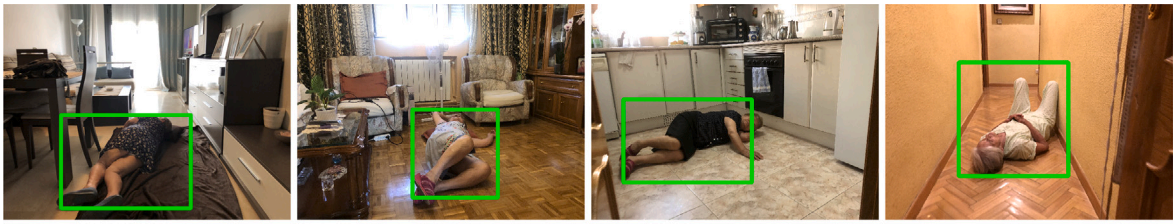


(a) (b)

**Fig. 14.** Examples of images from other datasets. (a) IASLAB-RGBD dataset. (b) URFD dataset.

(a) (b) (c) (d)

**Fig. 15.** Examples of false positives on other datasets (TP in green and FP in red). (a, b) IASLAB-RGBD dataset. (c,d) URFD dataset.



(a) (b) (c) (d)

**Fig. 16.** Exemplar results on elderly dataset (TP in green).

(a) (b) (c) (d)

**Fig. 17.** Images taken at different heights: (a) 0.75 m, (b) 1.0 m, (c) 1.5 m and (d) 2.0 m.

**Table 6**
Performance over the different camera's height.

| Number of falls | Camera's height | TDSVM-fall 20 class | | | TDSVM-fall 1 classes | | | TD-fall 2 classes | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | TP | FP | FN | TP | FP | FN | TP | FP | FN |
| 99 | 0.75 m | 99 | 0 | 0 | 99 | 0 | 0 | 99 | 0 | 0 |
| 158 | 1 m | 158 | 0 | 0 | 158 | 0 | 0 | 157 | 0 | 1 |
| 179 | 1.5 m | 179 | 0 | 0 | 179 | 0 | 0 | 179 | 0 | 0 |
| 143 | 2 m | 143 | 0 | 0 | 142 | 0 | 1 | 142 | 0 | 1 |

set. We tested the performance on this set with the TD-fall method and achieved precision and recall rates of 98.86% and 95.97%, respectively with $TP = 262$, $FP = 3$ and $FN = 11$. Some example images are shown in Fig. 16. These results allow us to conclude that the detector system is hardly affected by the age or appearance of the persons in the images.

### 5.4. Experiment 3: Performance for different camera's heights

Finally, we have conducted a complementary experiment to analyze the capacity of the system to detect fallen persons in images taken from different camera heights. For this purpose, we have captured additional sets of images (non-used before) taken at 0.75 m, 1 m, 1.5 m, and 2 m above the floor. All of the images have been taken with a distance from the image plane to the nearest seen floor of 2.5 m (see Fig. 17). Results from Table 6 show excellent performance in all cases and demonstrate the robustness of proposed strategies to changes in the camera's height.

## 6. Conclusions and future work

This paper shows that real-time fallen person detection can be accomplished with remarkable accuracy by using a single RGB camera without the need for additional sensors or 3D reconstruction of human posture. We have introduced two approaches based on the YOLO detector. The first approach, TDSVM-fall, is based on a two-stage algorithm composed of a YOLO person detector and a RBF SVM fall classifier. The generated region proposals by YOLO are further fed into the SVM classifier, which takes charge of discriminate fallen persons against non-fallen persons. In the second approach, namely TD-fall, we have demonstrated that it is possible to fuse detection and classification steps using an end-to-end convolutional network. We have examined our proposed framework on our own test set (E-FPDS) that can be used as a standard reference for other fallen person detector RGB vision-based algorithms. It contains 6982 annotated images covering different scenarios and human poses distributed in three subsets: training, validation, and test. Quantitative evaluations showed that the TD-fall method with a unique stage achieved precision and recall rates of 93.93% and 98.97%. The main advantage of our approach is the ability to differentiate between fallen persons lying on the floor and other different lying positions. Additionally, we examined our proposed methods on images from other public datasets. The TD-fall approach exhibited high performance in entirely new scenarios and confirmed that the proposed framework was feasible. Since we are mainly concerned with assistive patrol robots for care centers, we formulated the problem of fallen detection as lying posture detection and not as a process of falling due to the lack of capturing the person during the fall. Future research will incorporate complementary functionalities to the assistive robot to improve the interaction with persons. Additionally, improvements could come from the integration of other complementary sensors.

## CRediT authorship contribution statement

**Sergio Lafuente-Arroyo:** Conceptualization, Software, Methodology. **Pilar Martín-Martín:** Validation, Writing – original draft, Data curation. **Cristian Iglesias-Iglesias:** Software, Validation. **Saturnino Maldonado-Bascón:** Conceptualization, Methodology, Supervision. **Francisco Javier Acevedo-Rodríguez:** Conceptualization, Data curation, Software.

## 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.

## Acknowledgments

## Funding

This work was supported by the Spain's Ministry of Science and Innovation in the project AIRPLANE [PID2019-104323RB-C31] and Community of Madrid-University of Alcalá, Spain [CM/JIN/2019-022].

## References

Anderson, D., Keller, J. M., Skubic, M., Chen, X., & He, Z. (2006). Recognizing falls from silhouettes. In *2006 international conference of the ieee engineering in medicine and biology society* (pp. 6388–6391). IEEE.

Antonello, M., Carraro, M., Pierobon, M., & Menegatti, E. (2017). Fast and robust detection of fallen people from a mobile robot. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 4159–4166). IEEE.

Auvinet, E., Multon, F., Saint-Arnaud, A., Rousseau, J., & Meunier, J. (2010). Fall detection with multiple cameras: An occlusion-resistant method based on 3-d silhouette vertical distribution. *IEEE Transactions on Information Technology in Biomedicine*, *15*(2), 290–300.

Auvinet, E., Rougier, C., Meunier, J., St-Arnaud, A., & Rousseau, J. (2010). *Multiple cameras fall dataset*: *Tech. rep. 1350*, DIRO-Université de Montréal.

Bosch-Jorge, M., Sánchez-Salmerón, A.-J., Valera, A., & Ricolfe-Viala, C. (2014). Fall detection based on the gravity vector using a wide-angle camera. *Expert Systems with Applications*, *41*(17), 7980–7986.

Burns, E., & Kakara, R. (2018). Deaths from falls among persons aged ≥ 65 years - United States, 2007–2016. *Morbidity and Mortality Weekly Report*, *67*(18), 509–514.

Casilari, E., Santoyo-Ramón, J. A., & Cano-García, J. M. (2017). Umafall: A multisensor dataset for the research on automatic fall detection. *Procedia Computer Science*, *110*, 32–39.

Charfi, I., Miteran, J., Dubois, J., Atri, M., & Tourki, R. (2012). Definition and performance evaluation of a robust SVM based fall detection solution. In *2012 eighth international conference on signal image technology and internet based systems* (pp. 218–224). IEEE.

Charfi, I., Miteran, J., Dubois, J., Atri, M., & Tourki, R. (2013). Optimized spatio-temporal descriptors for real-time fall detection: comparison of support vector machine and adaboost-based classification. *Journal of Electronic Imaging*, *22*(4), Article 041106.

Charlon, Y., Fourty, N., Bourennane, W., & Campo, E. (2013). Design and evaluation of a device worn for fall detection and localization: Application for the continuous monitoring of risks incurred by dependents in an Alzheimer's care unit. *Expert Systems with Applications*, *40*(18), 7316–7330.

Chen, C., Jafari, R., & Kehtarnavaz, N. (2015). Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *2015 ieee international conference on image processing (ICIP)* (pp. 168–172). IEEE.

Cucchiara, R., Prati, A., & Vezzani, R. (2007). A multi-camera vision system for fall detection and alarm generation. *Expert Systems*, *24*(5), 334–345.

[Dataset] Department of Information Engineering- University of Padua (2017). IASLAB-RGBD Fallen person dataset. http://robotics.dei.unipd.it/117-fall (Online; accessed 30 November 2021).

[Dataset] Interdisciplinary Centre for Computational Modelling-University of Rzeszow (2014). UR Fall detection dataset (URFD). http://fenix.univ.rzeszow.pl/mkepski/ds/uf.html (Online; accessed 30 November 2021).

De Miguel, K., Brunete, A., Hernando, M., & Gambao, E. (2017). Home camera-based fall detection system for the elderly. *Sensors*, *17*(12), 2864.

Debard, G., Mertens, M., Deschodt, M., Vlaeyen, E., Devriendt, E., Dejaeger, E., Milisen, K., Tournoy, J., Croonenborghs, T., Goedemé, T., Tuytelaars, T., & Vanrumste, B. (2016). Camera-based fall detection using real-world versus simulated data: How far are we from the solution? *Journal of Ambient Intelligence and Smart Environments*, *8*(2), 149–168.

Ebrahimi, M., Khoshtaghaza, M., Minaei, S., & Jamshidi, B. (2017). Vision-based pest detection based on SVM classification method. *Computers and Electronics in Agriculture*, *137*, 52–58.

Ferreira, J. M., Pires, I. M., Marques, G., Garcia, N. M., Zdravevski, E., Lameski, P., Flórez-Revuelta, F., Spinsante, S., & Xu, L. (2020). Activities of daily living and environment recognition using mobile devices: A comparative study. *Electronics*, *9*(1), 180.

Frank, K., Vera Nadales, M. J., Robertson, P., & Pfeifer, T. (2010). Bayesian recognition of motion related activities with inertial sensors. In *Proceedings of the 12th ACM international conference adjunct papers on ubiquitous computing-adjunct* (pp. 445–446).

Guo, G., & Zhang, N. (2019). A survey on deep learning based face recognition. *Computer Vision and Image Understanding*, *189*, Article 102805.

Hsu, Y.-W., Perng, J.-W., & Liu, H.-L. (2015). Development of a vision based pedestrian fall detection system with back propagation neural network. In *2015 IEEE/SICE international symposium on system integration (SII)* (pp. 433–437). IEEE.

Igual, R., Medrano, C., & Plaza, I. (2013). Challenges, issues and trends in fall detection systems. *Biomedical Engineering Online*, *12*(1), 66.

Igual, R., Medrano, C., & Plaza, I. (2015). A comparison of public datasets for acceleration-based fall detection. *Medical Engineering & Physics*, *37*(9), 870–878.

Khan, S. S., & Taati, B. (2017). Detecting unseen falls from wearable devices using channel-wise ensemble of autoencoders. *Expert Systems with Applications*, *87*, 280–290.

King, D. (2017). Lib c++ library. https://face-recognition.readthedocs.io/en/latest/readme.html (Online; accessed 30 November 2021).

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

Lee, T., & Mihailidis, A. (2005). An intelligent emergency response system: preliminary development and testing of automated fall detection. *Journal of Telemedicine and Telecare*, *11*(4), 194–198.

Lewandowski, B., Wengefeld, T., Schmiedel, T., & Gross, H.-M. (2017). I see you lying on the ground—Can I help you? Fast fallen person detection in 3D with a mobile robot. In *2017 26th IEEE international symposium on robot and human interactive communication (RO-MAN)* (pp. 74–80). IEEE.

Li, L., Zhang, X., Mu, X., & Xu, H. (2019). Research on elderly fall prediction for walking posture of elderly-assistant robot. In *Proceedings of the 2019 2nd international conference on control and robot technology* (pp. 30–34).

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740–755). Cham: Springer.

Liu, C.-L., Lee, C.-H., & Lin, P.-M. (2010). A fall detection system using k-nearest neighbor classifier. *Expert Systems with Applications*, *37*(10), 7174–7181.

Maldonado-Bascón, S., Iglesias-Iglesias, C., Martín-Martín, P., & Lafuente-Arroyo, S. (2019). Fallen people detection capabilities using assistive robot. *Electronics*, *8*(9), 915.

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.

Mastorakis, G., & Makris, D. (2014). Fall detection system using Kinect's infrared sensor. *Journal of Real-Time Image Processing*, *9*(4), 635–646.

Medrano, C., Igual, R., Plaza, I., & Castro, M. (2014). Detecting falls as novelties in acceleration patterns acquired with smartphones. *PLoS One*, *9*(4), Article e94811.

Menacho, C., & Ordoñez, J. (2020). Fall detection based on CNN models implemented on a mobile robot. In *2020 17th international conference on ubiquitous robots (UR)* (pp. 284–289). IEEE.

Mubashir, M., Shao, L., & Seed, L. (2013). A survey on fall detection: Principles and approaches. *Neurocomputing*, *100*, 144–152.

Nait-Charif, H., & McKenna, S. J. (2004). Activity summarisation and fall detection in a supportive home environment. In *Proceedings of the 17th international conference on pattern recognition, 2004. ICPR 2004* (pp. 323–326). IEEE.

Ni, B., Wang, G., & Moulin, P. (2011). Rgbd-hudaact: A color-depth video database for human daily activity recognition. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)* (pp. 1147–1153). IEEE.

Olivieri, D. N., Conde, I. G., & Sobrino, X. A. V. (2012). Eigenspace-based fall detection and activity recognition from motion templates and machine learning. *Expert Systems with Applications*, *39*(5), 5935–5945.

Paoli, R., Fernández-Luque, F. J., Doménech, G., Martínez, F., Zapata, J., & Ruiz, R. (2012). A system for ubiquitous fall monitoring at home via a wireless sensor network and a wearable mote. *Expert Systems with Applications*, *39*(5), 5566–5575.

Principi, E., Droghini, D., Squartini, S., Olivetti, P., & Piazza, F. (2016). Acoustic cues from the floor: a new approach for fall classification. *Expert Systems with Applications*, *60*, 51–61.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779–788).

Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7263–7271).

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

Rimminen, H., Lindström, J., Linnavuo, M., & Sepponen, R. (2010). Detection of falls among the elderly by a floor sensor using the electric near field. *IEEE Transactions on Information Technology in Biomedicine*, *14*(6), 1475–1476.

Rougier, C., Meunier, J., St-Arnaud, A., & Rousseau, J. (2011). Robust video surveillance for fall detection based on human shape deformation. *IEEE Transactions on Circuits and Systems for Video Technology*, *21*(5), 611–622.

Schüldt, C., Laptev, I., & Caputo, B. (2004). Recognizing human actions: a local SVM approach. In *Proceedings of the 17th international conference on pattern recognition, 2004. ICPR 2004, Vol. 3* (pp. 32–36). IEEE.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437.

Stone, E. E., & Skubic, M. (2014). Fall detection in homes of older adults using the microsoft kinect. *IEEE Journal of Biomedical and Health Informatics*, *19*(1), 290–301.

Sucerquia, A., López, J. D., & Vargas-Bonilla, J. F. (2017). Sisfall: A fall and movement dataset. *Sensors*, *17*(1), 198.

Tomoya, A., Nakayama, S., Hoshina, A., & Sugaya, M. (2017). A mobile robot for following, watching and detecting falls for elderly care. *Procedia Computer Science*, *112*, 1994–2003.

Töreyin, B. U., Dedeoğlu, Y., & Çetin, A. E. (2005). Hmm based falling person detection using both audio and video. In *International workshop on human-computer interaction* (pp. 211–220). Springer, Berlin, Heidelberg.

Tzeng, H.-W., Chen, M.-Y., & Chen, J.-Y. (2010). Design of fall detection system with floor pressure and infrared image. In *2010 international conference on system science and engineering* (pp. 131–135). IEEE.

Vavoulas, G., Pediaditis, M., Chatzaki, C., Spanakis, E. G., & Tsiknakis, M. (2014). The mobifall dataset: Fall detection and classification with a smartphone. *International Journal of Monitoring and Surveillance Technologies Research (IJMSTR)*, *2*(1), 44–56.

Vilarinho, T., Farshchian, B., Bajer, D. G., Dahl, O. H., Egge, I., Hegdal, S. S., Lønes, A., Slettevold, J. N., & Weggersen, S. M. (2015). 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* (pp. 1443–1448). IEEE.

Vishwakarma, V., Mandal, C., & Sural, S. (2007). Automatic detection of human fall in video. In *International conference on pattern recognition and machine intelligence* (pp. 616–623). Springer, Berlin, Heidelberg.

Wang, X., Ellul, J., & Azzopardi, G. (2020). Elderly fall detection systems: A literature survey. *Frontiers in Robotics and AI*, *7*, 71.

Wu, H., Ma, X., Zhang, Z., Wang, H., & Li, Y. (2017). Collecting public RGB-D datasets for human daily activity recognition. *International Journal of Advanced Robotic Systems*, *14*(4), Article 1729881417709079.

Xu, T., Zhou, Y., & Zhu, J. (2018). New advances and challenges of fall detection systems: A survey. *Applied Sciences*, *8*(3), 418.

Yusoff, A., Salleh, S., Zaman, I., & Siswanto, W. (2021). Classification of fall detection system for elderly: Systematic review. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(3), 1769–1780.

Zerrouki, N., Harrou, F., Houacine, A., & Sun, Y. (2016). Fall detection using supervised machine learning algorithms: A comparative study. In *2016 8th international conference on modelling, identification and control (ICMIC)* (pp. 665–670). IEEE.

Zhang, H., Berg, A. C., Maire, M., & Malik, J. (2006). Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)* (pp. 2126–2136). IEEE.

Zhang, Z., Conly, C., & Athitsos, V. (2014). Evaluating depth-based computer vision methods for fall detection under occlusions. In *International symposium on visual computing* (pp. 196–207). Cham: Springer.

Zhang, Z., Conly, C., & Athitsos, V. (2015). A survey on vision-based fall detection. In *Proceedings of the 8th ACM international conference on pervasive technologies related to assistive environments* (pp. 1–7).