Vision-Based Fall Detection Systems

Subjects: Automation & Control Systems

Contributor: Jesús Gutiérrez

Vision-based fall detection systems have experienced fast development over the last years. To determine the course of its evolution and help new researchers, the main audience of this paper, a comprehensive revision of all published articles in the main scientific databases regarding this area during the last five years has been made. After a selection process, detailed in the Materials and Methods Section, eighty-one systems were thoroughly reviewed. Their characterization and classification techniques were analyzed and categorized. Their performance data were also studied, and comparisons were made to determine which classifying methods best work in this field. The evolution of artificial vision technology, very positively influenced by the incorporation of artificial neural networks, has allowed fall characterization to become more resistant to noise resultant from illumination phenomena or occlusion. The classification has also taken advantage of these networks, and the field starts using robots to make these systems mobile. However, datasets used to train them lack real-world data, raising doubts about their performances facing real elderly falls. In addition, there is no evidence of strong connections between the elderly and the communities of researchers.

Keywords: artificial vision ; neural networks ; fall detection ; fall characterization ; fall classification ; fall dataset

1. Introduction

In accordance with the UN report on the aging population [1], the global population aged over 60 doubled its number in 2017 compared to 1980. It is expected to double again by 2050 when they exceed the 2 billion mark. By this time, their number will be greater than the number of teenagers and youngsters aged 10 to 24.

The phenomenon of population aging is a global one, more advanced in the developed countries, but also present in the developing ones, where two-thirds of the worlds older people live, a number which is rising fast.

With this perspective, the amount of resources devoted to elderly health care is increasingly high and could, in the non-distant future, become one of the most relevant world economic sectors. Because of this, all elderly health-related areas have attracted great research attention over the last decades.

One of the areas immersed in this body of research has been human fall detection, as, for this community, over 30% of falls cause important injuries, ranging from hip fracture to brain concussion, and a good number of them end up causing death [2].

The number of technologies used to detect falls is wide, and a huge number of systems able to work with them have been developed by researchers. These systems, in broad terms, can be classified as wearable, ambient and camera-based ones [3].

The first block, the wearable systems, incorporate sensors carried by the surveilled individual. The technologies used by this group of systems are numerous, ranging from accelerometers to pressure sensors, including inclinometers, gyroscopes or microphones, among other sensors. R. Rucco et al. [4] thoroughly review these systems and study them in-depth. In this article, systems are classified in accordance with the number and type of sensors, their placement and the characteristics of the study made during the system evaluation phase concluding that most systems incorporate one or two accelerometric sensors attached to the trunk.

The second block includes systems whose sensors are placed around the monitored person and include pressure, acoustic, infra-red, and radio-frequency sensors. The last block, the object of this review, groups systems able to identify falls through artificial vision.

In parallel, over the last years, artificial vision has experienced fast development, mainly due to the use of artificial neural networks and their ability to recognize objects and actions.

This artificial vision development applied to human activity recognition in general, and human fall detection in particular, has given very fruitful outcomes in the last decade.

2. Discussion

2.1. Preprocessing

The final objective of this phase is either distortion and noise reduction or format adaptation, so downstream system blocks can extract characteristic features with classification purposes. Image complexity reduction could also be an objective during the preprocessing phase in some systems, so the computational cost can be reduced, or video streaming bandwidth use can be diminished.

The techniques grouped in this Section for decreasing noise are numerous and range from Gaussian smoothing used in [5] to the morphological operations executed in [6][5][7] or [8]. They are introduced in subsequent Section as a part of the foreground segmentation process.

Format adaptation processes are present in several of the studied systems, as is the case in [9], where images are converted to grayscale and have their histograms equalized before being transferred to the characterization process.

Image binarization, as in [10], is also introduced as a part of the systematic effort to reduce noise during the segmentation process, while some other systems, like the one presented in [11], pursue image complexity decreasing by transforming video signals from red, green and blue (RGB) to black and white and then applying a median filter, an algorithm which assigns new values to image pixels based on the median of the surrounding ones.

Image complexity reduction is a goal pursued by some systems, as the one proposed in [12], which introduces compressed sensing (CS), an algorithm first proposed by Donoho et al. [13] used in signal processing to acquire and reconstruct a signal. Through this technique, signals, sparse in some domain, are sampled at rates much lower than required by the Nyquist–Shannon sampling theorem. The system uses a three-layered approach to CS by applying it to video signals, which allows privacy preservation and bandwidth use reduction. This technique, however, introduces noise and over-smooths edges, especially those in low contrast regions, leading to information loss and image low-resolution. Therefore, image complexity reduction feature characterization often becomes a challenge.

2.2. Characterization

The second process step intends to express human pose and/or human motion as abstract features in a qualitative approach, to quantify their intensity in an ulterior quantity approach. These quantified features are then used with classifying purposes in the last step of the fall detection system.

These abstract pose/action descriptors can globally be classified into three main groups: global, local and depth.

Global descriptors analyze images as a block, segmenting foreground from background, extracting descriptors that define it and encoding them as a whole.

Local descriptors approach the abstraction problem from a different perspective and, instead of segmenting the block of interest, process the images as a collection of local descriptors.

Depth characterization is an alternative way to define descriptors from images containing depth information by either using depth maps or skeleton data extracted from a joint tracking process.

2.2.1. Global

Global descriptors try to extract abstract information from the foreground once it has been segmented from the background and encode it as a whole.

This kind of activity descriptors was very commonly used in artificial vision approaches to human activity recognition in general and to fall detection in particular. However, over time, they have been displaced by local descriptors or used in combination with them, as these ones are less sensitive to noise, occlusions and viewpoint changes.

Foreground segmentation is executed in a number of different ways. Some approaches to this concept establish a specific background and subtract it from the original image; some others locate regions of interest by identifying the silhouette edges or use the optical flow, generated as a consequence of body movements, as a descriptor. Some global characterization methods segment the human silhouette over time to form a space–time volume which characterizes the movement. Some other methods extract features from images in a direct way, as in the case of the system described in [9], where every three frames, the mean square error (MSE) is determined and used as an indicator of image similarity.

Silhouette Segmentation

Human shape segmentation can be executed through a number of techniques, but all of them require background identification and subtraction. This process, known as background extraction, is probably the most visually intuitive one, as its product is a human silhouette.

Background estimation is the most important step of the process, and it is addressed in different ways.

In [6][8][11][7], as the background is supposed constant, an image of it is taken during system initialization, and a direct comparison allows segmentation of any new object present in the video. This technique is easy and powerful; however, it is extremely sensitive to light changes. To mitigate this flaw, the system described in [5], where the background is also supposed stable, a median throughout time is calculated for every pixel position in every color channel. Then, it is directly subtracted from the observed image frame-by-frame.

Despite everything, the obtained product still contains a substantial amount of noise associated with shadows and illumination. To reduce it, morphological operators can be used as in [6][8][5][7]. Dilation and/or erosion operations are performed by probing the image at all possible places with a structuring element. In the dilation operation, this element works as a local maximum filter and, therefore, adds a layer of pixels to both inner and outer boundary areas. In erosion operations, the element works as a local minimum filter and, as a consequence, strips away a layer of pixels from both regions. Noise reduction after segmentation can also be performed through Kalman filtering, as in [14], where this filtering method is successfully used with this purpose.

An alternative option for background estimation and subtraction is the application of Gaussian mixture models (GMM), a technique used in [15][16][17][18][14], among others, that models the values associated with specific pixels as a mix of Gaussian distributions.

A different approach is used in [19], where the Horprasert method [20] is applied for background subtraction. It uses a computational color model that separates the brightness from the chromaticity component. By doing it, it is possible to segment the foreground much more efficiently when light disturbances are present than with previous methods, diminishing this way light change sensitiveness. In this particular system, pixels are also clustered by similarity, so computational complexity can be reduced.

Some systems, like the one presented in [15], apply a filter to determine silhouette contours. In this particular case, a Sobel filter is used, which determines a two-dimensional gradient of every image pixel.

Other segmentation methods, like vibe [21], used in [22][23], store, associated with specific pixels, previous values of the pixel itself and its vicinity to determine whether its current value should be categorized as foreground or background. Then, the background model is adapted by randomly choosing which values should be substituted and which not, a clearly different perspective from other techniques, which give preference to new values. On top of that, pixel values declared as background are propagated into neighboring pixels part of the background model.

The system in [24] segments the foreground using the technique proposed in [25], where the optical flow (OF), which are presented in later Sections, is calculated to determine what objects are in motion in the image, feature used for foreground segmentation. In a subsequent step, to reduce noise, images are binarized and morphological operators are applied. Finally, the points marking the center of the head and the feet are linked by lines composing a triangle whose area/height ratio will be used as the characteristic classification feature.

Some algorithms, like the illumination change-resistant independent component analysis (ICA), proposed in [26], combine features of different segmentation techniques, like GMM and self-organizing maps, a well-known group of ANN able to classify into low dimensional classes very high dimensional vectors, to overcome the problems of silhouette segmentation associated with illumination phenomena. This algorithm is able to successfully tackle segmentation errors associated with sudden illumination changes due to any kind of light source, both in images taken with omnidirectional dioptric cameras and in plain ones.

ICA and vibe are compared in [23] by using a dataset specifically developed for that system with better results for the ICA algorithm.

In [27], foreground extraction is executed in accordance with the procedure described in [28]. This method integrates the region-based information on color and brightness in a codeword, and the collection of all codewords are grouped in an entity called codebook. Pixels are then checked in every single new frame and, when its color or brightness does not match the region codeword, which encodes area brightness and color bands, it is declared as foreground. Otherwise, the codeword is updated, and the pixel is declared as area background. Once pixels are tagged as foreground, they are clustered together, and codebooks are updated for each one of them. Finally, these regions are approximated by polygons.

Some systems, like the one in [27], use orthogonal cameras and fuse foreground maps by using homography. This way, noise associated with illumination variations and occlusion is greatly reduced. The system also calculates the observed polygon area/ground projected polygon area rate as the main feature to determine whether a fall event has taken place.

Self-organizing maps is a technique, well described in [29], used with segmentation purposes in [30]. When applied, initial background estimation is made based on the first frame at system startup. Every pixel of this initial image is associated with a neuron in an ANN through a weight. Those weights are constantly updated as new frames flow into the system and, therefore, the background model changes. Self-organizing maps have been successfully used to subtract foreground from background, and they have proved a good resilience to the light variation noise.

Binarization is a technique used for background subtraction, especially in infrared (IR) systems, as the one presented in [10], where the inputs IR signals pixels are assigned two potential values, 0 and 1. All pixels above a certain threshold value are assigned a value 1 (human body temperature dependent), and all others are given a value of 0. This way, images are expressed in binary format. However, the resulting image usually has a great amount of noise. To reduce it, the algorithm is able to detect contours through gradient determination. Pixels within closed contours whose dimensions are close to the ones of a person continue being assigned a value 1, while the rest are given a value 0.

Once the foreground has been segmented, it is time to characterize it through abstract descriptors that can be classified at a later step.

This way, after background subtraction, features used for characterization in [5] and [17] are silhouettes eccentricity, orientation and acceleration of the ellipse surrounding the human shape.

Characteristic dimensions of the bounding box surrounding the silhouette are also a common distinctive feature, as is the case in [18]. In [31], a silhouette's horizontal width is estimated at 10 vertically equally spaced points, and, in [7], five regions are defined in the bounding box, being its degree of occupancy by the silhouette is used as the classifying element.

Other features also used for characterization used in [15][32] include Hu moments, a group of six image moments in variables to translation, scale, rotation, and reflection, plus a seventh one, which changes sign for image reflection. These moments, assigned to a silhouette, do not change as a result of the point of view alterations associated with body displacements. However, they dramatically vary as a result of human body pose changes as the ones associated with a fall. This way, a certain resistance to noise due to the point of view change is obtained.

The Feret diameter, the distance from the two most distant points of a closed line when taking a specific reference orientation, is another used distinctive feature. The system described in [30] uses this distance, with a reference orientation of 90°, to characterize the segmented foreground.

Procrustes analysis is a statistical method that uses minimum square methods to determine the needed similarity transformations required to adjust two models. This way, they can be compared, and a Procrustes distance, which quantifies how similar the models are, can be inferred. This distance, employed in some of the studied systems as a characterization feature, is used to determine similarities between silhouettes in consecutive frames and, therefore, as a measure of its deformation as a result of pose variation.

The system introduced in [22], after identifying in each frame the torso section in the segmented silhouette, stores its position in the last 100 frames in a database and uses this trajectory as a feature for fall recognition.

To decrease sensitiveness to noise as a result of illumination noise and viewpoint changes, some systems combine RGB global descriptors and depth information.

This is the case of [33], where the system primarily uses depth information, but when it is not available, RGB information is used instead. In that case, images are converted to grayscale and pictures are formed by adding up the difference between consecutive frames. Then, features are extracted at three levels. At the pixel level, where gradients are calculated, at the patch level, where adaptive patches are determined, and at the global level, where a pyramid structure is used to combine patch features from the previous level. The technique is fully described in [34].

A different approach to the same idea is tried in [35], where depth information is derived from monocular images as presented in [36]. This algorithm uses monocular visual cues, such as texture variations, texture gradients, defocus and color/haze. It mixes all these features with range information derived from a laser range finder to generate, through a Markov random field (MRF) model, a depth map. This map is assembled by splitting the image into patches of similar visual cues and assigning them depth information that is related to the one associated with other image patches. Then, and to segment foreground from background, as the human silhouette has an almost constant depth, a particle swarm optimization (PSO) method is used to discover the optical window in which the variance of the image depth is minimum. This way, patches whose depth information is within the band previously defined are segmented as foreground.

This method, first introduced in [37], was designed to simulate collective behaviors like the ones observed in flocks of birds or swarms of insects. It is an iterative method where particles progressively seek optimal values. This way, in every iteration, depth values with the minimum variance associated with connected patches are approximated, increasing until an optimal value is reached.

Space–Time Methods

All previously presented descriptors abstract information linked to specific frames and, therefore, they should be considered as static data, which clustered along time, acquire a dynamic dimension.

Some methods, however, present visual information where the time component is already inserted and, therefore, dynamic descriptors could be inferred from them.

That is the case of the motion history image (MHI) process. Through this method, after silhouette segmentation, a 2-D representation of its movement, which can be used to estimate if the movement has been fast or slow, is built up. It was first introduced by Bobick et al. [38] and reflects motion information as a function of pixel brightness. This way, all pixels represent moving objects bright with an intensity function of how recent movement is. This technique is used in [39][6][14] to complement other static descriptors and introduce the time component.

Some systems, like the one introduced in [40], split the global MHI feature in sub-MHIs that are linked to the bounding boxes created to track people. This way, a global feature like MHI is actually divided into parts, and the information contained in each one of them is associated with the specific silhouette responsible for the movement. Through this procedure, the system is able to locally capture movement information and, therefore, able to handle several persons at the same time.

2.2.2. Local

Local descriptors approach the problem of pose and movement abstraction in a different way. Instead of segmenting the foreground and extracting characteristic features from it, encoding them as a block, they focus on area patches from which relevant local features, characteristic of human movement or human pose, can be derived.

Over time, local descriptors have substituted or complemented global ones, as they have proofed to be much more resistant to noise or partial occlusion.

Characterization feature techniques focused on fall detection, pay attention to head motion, body shape changes and absence of motion [41]. The system introduced in [42] uses the two first groups of features. It models body shape changes and head motion by using the extended CORE9 framework [43]. This framework uses minimum bounding rectangles to abstract body movements. The system slaves bounding boxes to legs, hands and head, which is taken as the reference element. Then, directional, topological, and distance relations are established between the reference element and the other ones. All this information is finally used for classification purposes.

The vast majority of studied systems that implement local descriptors do it through the use of ANNs. ANNs are a major research area at the moment, and their application to the artificial vision and human activity recognition is a hot topic. These networks, which simulate biological neural networks, were first introduced by Rosenblatt [44] through the definition of the perceptron in 1958.

There are two main families of ANNs with application in artificial vision, human pose estimation and human fall detection, which have been identified in this research. These two families are convolutional neural networks (CNN) and recurrent neural networks (RNN).

ANNs are able to extract feature maps out of input images. These maps are local descriptors able to characterize the different local patches that integrate an image.

RNNs are connectionist architectures able to grasp the dynamics of a sequence due to cycles in its structure. Introduced by Hopfield [45], they retain information from previous states and, therefore, they are especially suitable to work with sequential data when its flow is relevant. This effect of information retention through time is obtained by implementing recurrent connections that transfer information from previous time steps to either other nodes or to the originating node itself.

Among RNNs architectures, long short-term memory (LSTM) ones are especially useful in the field of fall detection. Introduced by Hochreiter [46], LSTMs most characteristic feature is the implementation of a hidden layer composed of an aggregation of nodes, called memory cells. These items contain nodes with a self-linked recurrent connection, which guarantees information will be passed along time with no vanishing. Unlike other RNNs, whose long-term memory materializes through weights given to inputs, which change slowly during training, and whose short-term memory is implemented through ephemeral activations, passed from a node to the successive one, LSTMs introduce an intermediate memory step in the memory cells. These elements internally retain information through their self-linked recurrent connections, which include a forget gate. Forget gates allow the ANN to learn how to forget the contents of previous time steps.

LSTM topologies, like the one implemented in [47], allow the system to recall distinctive features from previous frames, incorporating, this way, the time component to the image descriptors. In this particular case, an RNN is built by placing two LSTM layers between batch normalization layers, whose purpose is to make the ANN faster. Finally, a last layer of the network, responsible for classification, implements a Softmax algorithm.

Some LSTMs architectures, like the one described in [48], are used to determine characteristic foreground features. This ANN is able to establish a silhouette center and establish angular speed, which will be used as a reference to determine whether a fall event has taken place.

The system proposed in [49] includes several LSTM layers. This encoding-decoding architecture integrates an encoding block, which encodes the input data, coming from a CNN block used to identify joints and estimate body pose, to a vector of fixed dimensionality, and a decoding block, composed of a layer able to output predictions on future body poses. This architecture is based on the seq2seq model proposed in [50] and has been successfully used in this system with prediction purposes, substantially reducing fall detection time as it is assessment is made on a prediction, not on observation.

A specific LSTM design is the bidirectional one (Bi-LSTM). This architecture integrates two layers of hidden nodes connected to inputs and outputs. Both layers implement the idea of information retention through time in a different way. While the first layer has recurrent connections, in the second one, connections are flipped and passed backward through the activation function signal. This topology is incorporated in [51], where Bi-LSTM layers are stacked over CNN layers used to segment incoming images.

CNNs were inspired by the neural structure of the mammal visual system, very especially by the patterns proposed by Hubel et al. [52]. The first neural network model with visual pattern recognition capability was proposed by Fukushima [53], and, based on it, LeCun and some collaborators developed CNNs with excellent results in pattern recognition, as shown in [54][55].

This family of ANNs is assembled by integrating three main types of layers; convolutional, pooling and fully connected, each one of them playing a different role. Every layer of the CNN receives an input, transforms it and delivers an output. This way, the initial layers, which are convolutional ones, deliver feature maps out of the input images, whose complexity is reduced by the pooling layers. Eventually, these maps are led to the fully connected layers, where the feature maps are converted into vectors used for classification.

A typical CNN architecture is shown in Figure 2.

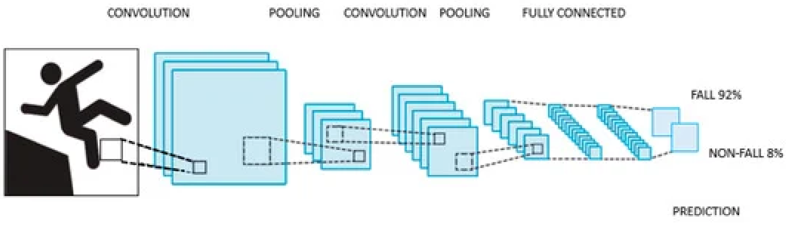

Figure 2. Typical convolutional neural network (CNN) architecture.

Some systems, like the one in [56], where a YoLOv3 CNN is used, take the input image and modify its scale to get several feature maps out of the same image. In this case, the CNN is used to generate three different sets of feature maps, based on three image scales, which eventually, after going through the fully connected layers, will be used for classification.

A similar approach is used in [57], where a YoLOv3 CNN identifies people. Identified people are tracked, and a CNN ANN extracts characteristic features from each person in the image. The feature vectors are passed to an LSTM ANN whose main task is to retain features over time so the temporal dimension can be added to the spatial features obtained by the CNN. The final feature vectors, coming out of the LSTM layers, are sent to a fully connected layer, which implements a Softmax algorithm used for event classification.

In [58], the object detection task, performed by a YoLO CNN, is combined with object tracking, a task developed by DeepSORT [59], a CNN architecture able to track multiple objects after they have been detected.

The approach made in [60] to detect a fallen person uses a YoLOv3 CNN to detect fallen bodies on the ground plane. It maximizes the sensitivity by turning 90 and 270 degrees all images and compare the bounding boxes found in the same image. Then, features are extracted from the bounding box, which will be used as classification features.

In [18][61], a wide residual network, which is a type of CNN, takes as input an OF and derives feature maps out of it. These maps are delivered to the fully connected layers, which, in turn, will pass vectors for movement classification to the last layers of the ANN.

A similar procedure is followed by the system in [10], whose ANN mixes layers of CNN, which deliver features maps from the incoming binarized video signal, with layers of radial basis function neural networks (RBFNN), which will be used as a classifier.

Another interesting type of CNN is the hourglass convolutional auto-encoder (HCAE), introduced in [62]. This kind of architecture piles convolutional and pooling layers over fully connected ones to get a feature vector, and then it follows the inverse process to reconstruct the input images. The HCAE compares the error value between the encoded-decoded frames and the original frames, applying back-propagation for self-tuning. Ten consecutive frames are inputted into the system to guarantee it captures both image and action features.

An alternate approach is the one presented in [63], where a CNN identifies objects (including people) and associate vectors to them. These vectors, which measure features, characterize both the human shape itself and its spatial relations with surrounding objects. This way, events are classified not only as a function of geometrical features of the silhouette but also as a function of its spatial relations with other objects present in the image. This approach has proven very useful to detect incomplete falls where pieces of furniture are involved.

A good number of approaches, as in [64], use 3D CNNs to extract spatiotemporal features out of 2D images, like the ones used in this system. This way, ANNs are used not only to extract spatial features associated with pose recognition but also to capture the temporal relation established among successive poses leading to a fall. The system in [65] uses this approach, creating a dynamic image by fusing in a single image all the frames belonging to a time window and passing this image to the ANN as the input from where extracting features.

Certain convolutional architectures, like the ones integrated into OpenPose and used in [58][66], can identify human body key points through convolutional pose machines (CPM), as shown in Figure 3, a CNN able to identify those features. These key points are used to build a vector model of the human body in a bottom-up approach.

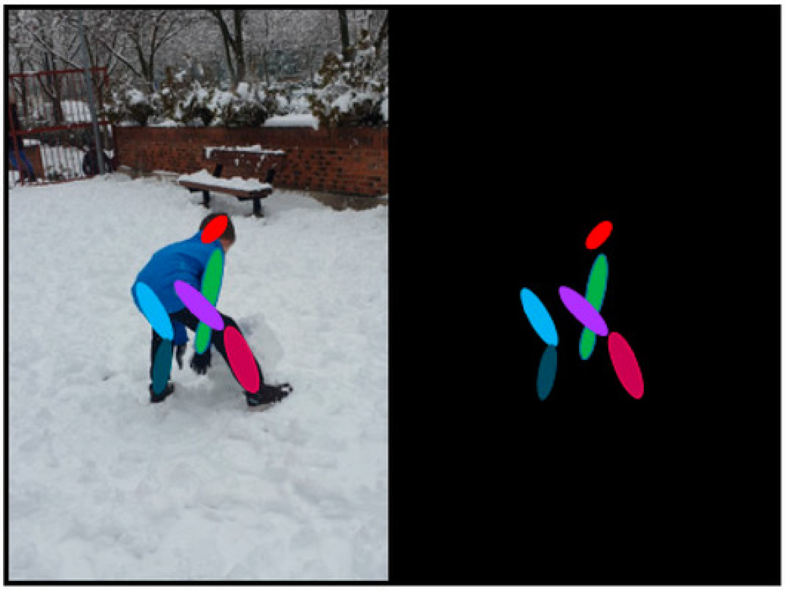

Figure 3. Convolutional pose machine presentation.

To correct possible mistakes, this approximation is complemented in [66] by a top-down approach through single shot multibox detector-MobileNet (SSD-MobileNet), another convolutional architecture able to identify multiple objects, human bodies in this case. SSD-MobileNet, lighter and requires less computational power than typical SSDs, is used to remove all key points identified by OpenPose not being part of a human body, correcting this way, inappropriate body vector constructions.

A similar approach is used in [67], where a CNN is used to generate an inverted pendulum based on five human key points, knees, the center of the hip line, neck and head. The motion history of these joints is recorded, and a subsequent module calculates the pendulum rotation energy and its generalized force sequences. These features are then codified in a vector and used for classification purposes.

The system in [68] uses several ANNs and selects the most suitable one as a function of the environment and the characteristics of the tracked people. In addition, it uploads wrongly categorized images which are used to retrain the used models.

2.2.3. Depth

Descriptors based on depth information have gained ground thanks to the development of low-cost depth sensors, such as Microsoft Kinect®. This affordable system counts with a software development kit (SDK) and applications able to detect and track joints and construct human body vector models. These elements, together with the depth information from stereoscopic scene observation, have raised great interest among the artificial vision research community in general and the human fall detection system developers in particular.

A good number of the studied systems use depth information, solely or together with RGB one, as the data source in the abstraction process leading to image descriptor construction. These systems have proved to be able to segment foreground, greatly diminishing interference due to illumination interferences up to the distance where stereoscopic vision procedures are able to infer depth data. Fall detection systems use this information either as depth maps or skeleton vector models.

Depth Map Representation

Depth maps, unlike RGB video signals, contain direct three-dimensional information on objects in the image. Therefore, depth map video signals integrate raw 3D information, so three-dimensional characterization features can be directly extracted from them.

This way, the system described in [69] identifies 16 regions of the human body marked with red tape and position them in space through stereoscopic techniques. Taking that information as a base, the system builds the body vector (aligned with spine orientation) and identifies its center of gravity (CG). Acceleration of CG and body vector angle on a vertical axis will be used as features for classification.

Foreground segmentation of human silhouette is made by these systems through depth information, by comparing depth data from images and a reference established at system startup. This way, pixels appearing in an image at a distance different from the one stored for that particular pixel in the reference are declared as foreground. This is the process followed by [70] to segment the human silhouette. In an ulterior step, descriptors based on bounding box, centroid, area and orientation of the silhouette are extracted.

Other systems, like the one in [71], extract background by using the same process and the silhouette is determined as the major connected body in the resulting image. Then, an ellipse is established around it, and classification will be made as a function of its aspect ratio and centroid position. A similar process is followed in [72], where, after background subtraction, an ellipse is established around the silhouette, and its centroid elevation and velocity, as well as its aspect ratio, are used as classification features.

The system in [73] uses depth maps to segment silhouettes as well and creates a bounding box around them. Box top coordinates are used to determine the head velocity profile during a fall event, and its Hausdorff distance to head trajectories recorded during real fall events is used to determine whether a fall has taken place. The Hausdorff distance quantifies how far two subsets of a metric space are from each other. The novelty of this system, leaving aside the introduction of the Hausdorff distance as described in [74], is the use of a moving capture (MoCap) technique to drive a human model using software to simulate its motion (OpenSim), so profiles of head vertical velocities can be captured in ADLs, and a database can be built. This database is used, by the introduction of the Hausdorff distance, to assess falls.

The system in [75], after foreground extraction by using depth information as in the previous systems, transforms the image to a black and white format and, after de-noising it through filtering, calculates the HOG. To do it, the system determines the gradient vector and its direction for each image pixel. Then, a histogram is constructed, which integrates all pixels' information. This is the feature used for classification purposes.

In [76], silhouettes are tracked by using a proportional-integral-differential (PID) controller. A bounding box is created around the silhouette, and features are extracted in accordance with [77]. A fall will be called if thresholds established for features are exceeded. Faces are searched, and when identified, the tracking will be biased towards them.

Some other systems, like the one in [78], subtracts background by direct use of depth information contained in sequential images, so the difference between consecutive depth frames is used for segmentation. Then, the head is tracked, so the head vertical position/person height ratio can be determined, which, together with CG velocity, is used as a classification feature.

In [79], all background is set to a fixed depth distance. Then, a group of 2000 body pixels is randomly chosen, and for each of them, a vector of 2000 values, calculated as a function of the depth difference between pairs of points, is created. These pairs are determined by establishing 2000 pixel offset sets. The obtained 2000-value vector is used as a characteristic feature for pose classification.

The system introduced in [16], after the human silhouette is segmented by using depth information through a GMM process, calculates its curvature scale space (CSS) features by using the procedures described in [36]. CSS calculation method convolutes a parametric representation of a planar curve, silhouette edge in this case, with a Gaussian function. This way, a representation of the arc length vs. curvature is obtained. Then, silhouettes features are encoded, together with the Gaussian mixture model used in the aforementioned CSS process, in a single Fisher vector, which will be used, after being normalized, for classification purposes.

Finally, a block of systems creates volumes based on normal distributions constructed around point clouds. These distributions, called voxels, are grouped together, and descriptors are extracted out of voxel clusters to determine, first, whether they represent a human body and then to assess if it is in a fallen state.

This way, the system presented in [80] first estimates the ground plane by assuming that most of the pixels belonging to every horizontal line are part of the ground plane. The ground can then be estimated, line per line, attending to the pixel depth values as explained in the procedure described in [81]. To clean up the pictures, all pixels below the ground plane are discarded. Then, normal distributions transform (NDT) maps are created as a cloud of points surrounded by normal distributions with the physical appearance of an ellipsoid. These distributions, created around a minimum number of points, are called voxels and, in this system, are given fixed dimensions. Then, features that describe the local curvature and shape of the local neighborhood are extracted from the distributions. These features, known as IRON [82], allow voxel classification as being part of a human body or not and, this way, voxels tagged as human are clustered together. IRON features are then calculated for the cluster representing a human body, and the Mahalanobis distance between that vector and the distribution associated with fallen bodies is calculated. If the distance is below a threshold, the fall state is declared.

A similar process is used in [83], where, after the point cloud is truncated by removing all points not contained in the area in between the ground plane and a parallel one 0.7 m over it by applying the RANSAC procedure [84], NDTs are created and then segmented in patches of equal dimensions. A support vector machine (SVM) classifier determines which ones of those patches belong to a human body as a function of their geometric characteristics. Close patches tagged as humans are clustered, and a bounding box is created around. A second SVM determines whether clusters should be declared as a fallen person. This classification is refined, taking data from a database of obstacles of the area, so if the cluster is declared as a fallen person, but it is contained in the obstacle database, the declaration is skipped.

2.3. Classification

Once pose/movement abstract descriptors have been extracted from video images, the next step of the fall detection process is classification. In broad terms, during this phase, the system classifies movement and or pose as a fall or a fallen state through an algorithm that is part of one of these two categories; generative or discriminative models.

Discriminative models are able to determine boundaries between classes, either by explicitly being given those boundaries or by setting them themselves using sets of pre-classified descriptors.

Generative models approach the classification problem in a totally different way, as they explicitly model the distribution of each class and then use the Bayes theorem to link descriptors to the most likely class, which, in this case, can only be a fall or a not fall state.

2.3.1. Discriminative Models

The final goal of any classifier is assigning a class to a given set of descriptors. The discriminative models are able to establish the boundaries separating classes, so the probability of a descriptor belonging to a specific class can be given. In other terms, given α as a class, and [A] as the matrix of descriptor values associated with a pose or movement, this family of classifiers is able to determine the probability **P** (α|[A]).

Feature-Threshold-Based

Feature-threshold-based classification models are broadly used in the studied systems. This approach is easy and intuitive, as the researcher establishes threshold values for the descriptors, so their associated events can be assigned to a specific class in case those thresholds are exceeded.

This is the case of the system proposed in [5]. It classifies the action as a fall or a non-fall in accordance with a double rationale. On one hand, it establishes thresholds of ellipse features to estimate whether the pose fits a fallen state; on the other, an MHI feature exceeding a certain value indicates a fast movement and, therefore, a potential fall. The system proposed in [17] adds acceleration to the former features and, in [85], head speed over a certain threshold and CG position out of the segment defined by ankles are indicatives of a fall.

Similar approaches, where threshold values are determined by system developers based on previous experimentation, are implemented in a good number of the studied systems, as they are simple, intuitive and computationally inexpensive.

Multivariate Exponentially Weighted Moving Average

Multivariate exponentially weighted moving average (MEWMA) is a statistical process control to monitor variables that use the entire history of values of a set of variables. This technique allows designers to give a weighting value to all recorded variable outputs, so the most recent ones are given higher weight values, and the older ones are weighted lighter. This way, the last value is weighted λ (being λ a number between 0 and 1) and previous β values are weighted λ^β. Limits to the value of that weighted output are established, taking as a basis the expected mean and standard deviation of the process. Certain systems, like [86], use this technique for classification purposes. However, as it is unable to distinguish between falling events and other similar ones, events tagged as fall by the MEWMA classifier need to go through an ulterior support vector machine classifier.

Support Vector Machines

Support vector machines (SVM) are a set of supervised learning algorithms.

SVM's are used for regression and classification problems. They create hyperplanes in high dimension spaces that separate classes nonlinearly. To fulfill this task, SVMs, similar to artificial neural networks, use kernel functions of different types.

K-Nearest Neighbor

K-nearest neighbor (KNN) is an algorithm able to model the conditional probability of a sample belonging to a specific class.

KNNs assume that classification can be successfully made based on the class of the nearest neighbors. This way, if for a specific feature, all x closest sample neighbors are part of a determined class, the probability of the sample being part of that class will be assessed as very high. This study is repeated for every feature contained in the descriptor, so a final assessment based on all features can be made. The algorithm usually gives different weights to the neighbors, and heavier weights are assigned to the closest ones. On top of that, it also assigns different weights to every feature. This way, the ones assessed as most relevant get heavier weights.

Decision Tree

Decision trees (DT) are algorithms used both in regression and classification. It is an intuitive tool to make decisions and explicitly represents decision-making. Classification DTs use categorical variables associated with classes. Trees are built by using leaves, which represent class labels, and branches, which represent characteristic features of those classes. DTs built process is iterative, with a selection of features correctly ordered to determine the split points that minimize a cost function that measures the computational requirements of the algorithm.

Random forest (RF) is an aggregation technique of DT whose main objective is avoiding overfitting. To accomplish this task, the training dataset is divided into subgroups, and therefore, a final number of DTs, equal to the number of dataset subgroups, is obtained. All of them are used in the process, so the final classification decision is actually a combination of the classification of all DTs.

Boost Classifier

Boost classifier algorithms are a family of classifier building techniques that create strong classifiers by grouping weak ones. It is done by adding up models created from the training data until the system is perfectly predicted or a maximum number of models is reached.

This is done by building a model from the training data. Then, a second model is created to correct the errors from the first one. Models are added until the training set is well predicted or a maximum number of them is added. During the boosting process, the first model is trained on the entire database while the rest are fitted to the residuals of the previous ones.

Deep Learning Models

Some ANN implement a Softmax function, a generalization of the logistic function used for multinomial logistic regression, in its last layers. This function is used as the activation function of the nodes of the last layer of a neural network, so its output is normalized to a probability distribution over the different output classes.

Multilayer perceptron (MLP) is a type of multilayered ANN with hidden layers between the entrance and the exit ones able to sort out classes non linearly separable. Each node of this network is a neuron that uses a nonlinear activation function, and it is used for classification purposes in some systems.

Radial basis function neural networks (RBFNN) are used in the last layer of a number of systems to classify the feature vectors coming from previous CNN layers. This ANN is characterized by using radial basis functions as activation functions and yields better generalization capabilities than other architectures, such as Softmax, as it is trained via minimizing the generalized error estimated by a localized-generalization error model (L-GEM).

Often, the last layers of ANN architectures are fully connected ones, where all nodes of a layer are connected to all nodes in the next one. In these structures, the input layer is used to flatten outputs from previous layers and transform them into a single vector, while subsequent layers apply weights to determine a proper tagging and, therefore, successfully classify events.

Finally, another ANN structure useful for classification is the autoencoder one. Autoencoders are ANNs trained to generate outputs equal to inputs. Its internal structure includes a hidden layer where all neurons are connected to every input and output node. This way, autoencoders get high dimensional vectors and encode their features. Then, these features are decoded back. As the number of dimensions of the output vector may be reduced, this kind of ANNs can be used for classification purposes by reducing the number of output dimensions to the number of final expected classes.

2.3.2. Generative Models

The approach of generative models to the classification problem is completely different from the one followed by the discriminative ones.

Generative models explicitly model the distribution of each class. This way, given α as a class, and [A] as the matrix of descriptor values associated with a pose or movement, if both P ([A]|α) and P (α) can be determined, it will be possible, by direct application of the Bayes theorem, to obtain P (α|[A]), which will solve the classification problem.

Hidden Markov Model

Classification using the hidden Markov model (HMM) algorithm is one of the three typical problems that can be solved through this procedure. It was first proposed with this purpose by Rabiner et al. [87] to solve the speech recognition problem, and it is used in [88] to classify the feature vectors associated with a silhouette.

HMMs are stochastic models used to represent systems whose state variables change randomly over time. Unlike other statistical procedures, like Markov chains, which deal with fully observable systems, HMMs tackle partially observable systems. This way, the final objective of the HMM classifying problem resolution will be decided, on the basis of the observable data (feature vector), whether a fall has occurred (hidden system state).

The system proposed in [88] determines, using an HMM as a classifier, on the basis of silhouette surface, centroid position and bounding box aspect ratio, whether a fall takes place or not. To do it, and to take as a reference recorded falls, a probability is assigned to the two possible system states (fall/not fall) based on value and variation along the event timeframe period of the feature vector. This classifying technique is used with success in this system, though in [89], a brief summary of the numerous limitations of this basic HMM approach is presented, and several more efficient extensions of the algorithm, such as variable transition HMM or the hidden semi-Markov model, are introduced. These algorithm variations are developed as the basic HMM process is considered ill-suited for modeling systems where interacting elements are represented through a vector of single state variables.

A similar classification approach using an HMM classifier is used in [90], where future states predicted by an autoregressive-moving-average (ARMA) algorithm are classified as fall or not-fall events. ARMA models are able to predict future states of a system based on a previous time-series. The model integrates two modules, an autoregressive one, which uses a linear combination of weighted previous system state values, and a moving average one, which linearly combines weighted previous errors between system state real values and predicted ones. In the model, errors are assumed to be random values that fit a Gaussian distribution of mean 0 and variance σ^2.

References

1. United Nations. World Population Ageing 2017: Highlights; Department of Economic and Social Affairs, United Nations: New York, NY, USA, 2017.
2. Sterling, D.A.; O’connor, J.A.; Bonadies, J. Geriatric falls: Injury severity is high and disproportionate to mechanism. J. Trauma Inj. Infect. Crit. Care 2001, 50, 116–119.
3. Vallabh, P.; Malekian, R. Fall detection monitoring systems: A comprehensive review. J. Ambient. Intell. Humaniz. Com put. 2018, 9, 1809–1833.
4. Rucco, R.; Sorriso, A.; Liparoti, M.; Ferraioli, G.; Sorrentino, P.; Ambrosanio, M.; Baselice, F. Type and Location of Wear able Sensors for Monitoring Falls during Static and Dynamic Tasks in Healthy Elderly: A Review. Sensors 2018, 18, 161 3.
5. Basavaraj, G.M.; Kusagur, A. Vision based surveillance system for detection of human fall. In Proceedings of the 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEIC T), Bangalore, India, 19–20 May 2017; pp. 1516–1520.
6. Bhavya, K.R.; Park, J.; Park, H.; Kim, H.; Paik, J. Fall detection using motion estimation and accumulated image map. I n Proceedings of the 2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Seoul, Korea, 2 6–28 October 2016; pp. 1–2.
7. Harrou, F.; Zerrouki, N.; Sun, Y.; Houacine, A. An Integrated Vision-Based Approach for Efficient Human Fall Detection i n a Home Environment. IEEE Access 2019, 7, 114966–114974.

8. Alaoui, A.Y.; El Hassouny, A.; Thami, R.O.H.; Tairi, H. Video based human fall detection using von Mises distribution of motion vectors. In Proceedings of the 2017 Intelligent Systems and Computer Vision (ISCV), Fez, Morocco, 17–19 April 2017; pp. 1–5.

9. Galvao, Y.M.; Albuquerque, V.A.; Fernandes, B.J.T.; Valenca, M.J.S. Anomaly detection in smart houses: Monitoring elderly daily behavior for fall detecting. In Proceedings of the 2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI), Arequipa, Peru, 8–10 November 2017; pp. 1–6.

10. Zhong, C.; Ng, W.W.Y.; Zhang, S.; Nugent, C.; Shewell, C.; Medina-Quero, J. Multi-occupancy Fall Detection using Non-Invasive Thermal Vision Sensor. IEEE Sens. J. 2020, 21, 1.

11. Dai, B.; Yang, D.; Ai, L.; Zhang, P. A Novel Video-Surveillance-Based Algorithm of Fall Detection. In Proceedings of the 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Beijing, China, 13–15 October 2018; pp. 1–6.

12. Liu, J.-X.; Tan, R.; Sun, N.; Han, G.; Li, X.-F. Fall Detection under Privacy Protection Using Multi-layer Compressed Sensing. In Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, 28–31 May 2020; pp. 247–251.

13. Donoho, D.L. Compressed sensing. IEEE Trans. Inf. Theory 2006, 52, 1289–1306.

14. Thummala, J.; Pumrin, S. Fall Detection using Motion History Image and Shape Deformation. In Proceedings of the 2020 8th International Electrical Engineering Congress (iEECON), Chiang Mai, Thailand, 4–6 March 2020; pp. 1–4.

15. Rajabi, H.; Nahvi, M. An intelligent video surveillance system for fall and anesthesia detection for elderly and patients. In Proceedings of the 2015 2nd International Conference on Pattern Recognition and Image Analysis (IPRIA), Rasht, Iran, 11–12 March 2015; pp. 1–6.

16. Aslan, M.; Sengur, A.; Xiao, Y.; Wang, H.; Ince, M.C.; Ma, X. Shape feature encoding via Fisher Vector for efficient fall detection in depth-videos. Appl. Soft Comput. 2015, 37, 1023–1028.

17. Lin, C.; Wang, S.-M.; Hong, J.-W.; Kang, L.-W.; Huang, C.-L. Vision-Based Fall Detection through Shape Features. In Proceedings of the 2016 IEEE Second International Conference on Multimedia Big Data (BigMM), Taipei, Taiwan, 20–22 April 2016; pp. 237–240.

18. Soni, P.K.; Choudhary, A. Automated Fall Detection From a Camera Using Support Vector Machine. In Proceedings of the 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), Gangtok, India, 25–28 February 2019; pp. 1–6.

19. Chong, C.-J.; Tan, W.-H.; Chang, Y.C.; Batcha, M.F.N.; Karuppiah, E. Visual based fall detection with reduced complexity horprasert segmentation using superpixel. In Proceedings of the 2015 IEEE 12th International Conference on Networking, Sensing and Control, Taipei, Taiwan, 9–11 April 2015; pp. 462–467.

20. Horprasert, T.; Harwood, D.; Davis, L.S. A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection. In Proceedings of the IEEE ICCV'99 FRAME-RATE Workshop, Kerkyra, Greece, 20 September 1999.

21. Barnich, O.; Van Droogenbroeck, M. ViBe: A Universal Background Subtraction Algorithm for Video Sequences. IEEE Trans. Image Process. 2011, 20, 1709–1724.

22. Wang, X.; Liu, H.; Liu, M. A novel multi-cue integration system for efficient human fall detection. In Proceedings of the 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO), Uttarakhand, India, 3–4 December 2016; pp. 1319–1324.

23. Kottari, K.N.; Delibasis, K.; Maglogiannis, I. Real-Time Fall Detection Using Uncalibrated Fisheye Cameras. IEEE Trans. Cogn. Dev. Syst. 2019, 12, 588–600.

24. Juang, L.H.; Wu, M.N. Fall Down Detection Under Smart Home System. J. Med. Syst. 2015, 39, 107–113.

25. Mittal, A.; Paragios, N. Motion-based background subtraction using adaptive kernel density estimation. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, Washington, DC, USA, 27 June–2 July 2004.

26. Delibasis, K.; Goudas, T.; Maglogiannis, I. A novel robust approach for handling illumination changes in video segmentation. Eng. Appl. Artif. Intell. 2016, 49, 43–60.

27. Mousse, M.A.; Motamed, C.; Ezin, E.C. Video-Based People Fall Detection via Homography Mapping of Foreground Polygons from Overlapping Cameras. In Proceedings of the 2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Bangkok, Thailand, 23–27 November 2015; pp. 164–169.

28. Mousse, M.A.; Motamed, C.; Ezin, E.C. Fast Moving Object Detection from Overlapping Cameras. In Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics, Colmar, France, 21–23 July 2015; pp. 296–303.

29. Mario, I.; Chacon, M.; Sergio, G.D.; Javier, V.P. Simplified SOM-neural model for video segmentation of moving objects. In Proceedings of the 2009 International Joint Conference on Neural Networks, Atlanta, GA, USA, 14–19 June 2009; pp. 474–480.

30. Sehairi, K.; Chouireb, F.; Meunier, J. Elderly fall detection system based on multiple shape features and motion analysis. In Proceedings of the 2018 International Conference on Intelligent Systems and Computer Vision (ISCV), Fez, Morocco, 2–4 April 2018; pp. 1–8.

31. Shanshan, X.; Xi, C. Fall detection method based on semi-contour distances. In Proceedings of the 2018 14th IEEE International Conference on Signal Processing (ICSP), Beijing, China, 12–16 August 2018; pp. 785–788.

32. Joshi, N.B.; Nalbalwar, S. A fall detection and alert system for an elderly using computer vision and Internet of Things. In Proceedings of the 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India, 19–20 May 2017; pp. 1276–1281.

33. Tran, T.-H.; Le, T.-L.; Hoang, V.-N.; Vu, H. Continuous detection of human fall using multimodal features from Kinect sensors in scalable environment. Comput. Methods Programs Biomed. 2017, 146, 151–165.

34. Nguyen, V.-T.; Le, T.-L.; Tran, T.-H.; Mullot, R.; Courboulay, V.; Van-Toi, N. A new hand representation based on kernels for hand posture recognition. In Proceedings of the 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), Ljubljana, Slovenia, 4–8 May 2015; Volume 1, pp. 1–6.

35. Ko, M.; Kim, S.; Kim, M.-G.; Kim, K. A Novel Approach for Outdoor Fall Detection Using Multidimensional Features from A Single Camera. Appl. Sci. 2018, 8, 984.

36. Ma, X.; Wang, H.; Xue, B.; Zhou, M.; Ji, B.; Li, Y. Depth-Based Human Fall Detection via Shape Features and Improved Extreme Learning Machine. IEEE J. Biomed. Health Inform. 2014, 18, 1915–1922.

37. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN’95-International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.

38. Bobick, A.F.; Davis, J.W. The recognition of human movement using temporal templates. IEEE Trans. Pattern Anal. Mach. Intell. 2001, 23, 257–267.

39. Gunale, K.G.; Mukherji, P. Fall detection using k-nearest neighbor classification for patient monitoring. In Proceedings of the 2015 International Conference on Information Processing (ICIP), Pune, India, 16–19 December 2015; pp. 520–524.

40. Feng, Q.; Gao, C.; Wang, L.; Zhang, M.; Du, L.; Qin, S. Fall detection based on motion history image and histogram of oriented gradient feature. In Proceedings of the 2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Xiamen, China, 6–9 November 2017; pp. 341–346.

41. Nizam, Y.; Haji Mohd, M.N.; Abdul Jamil, M.M. A Study on Human Fall Detection Systems: Daily Activity Classification and Sensing Techniques. Int. J. Integr. Eng. 2016, 8, 35–43.

42. Kalita, S.; Karmakar, A.; Hazarika, S.M. Human Fall Detection during Activities of Daily Living using Extended CORE9. In Proceedings of the 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), Gangtok, India, 25–28 February 2019; pp. 1–6.

43. Kalita, S.; Karmakar, A.; Hazarika, S.M. Efficient extraction of spatial relations for extended objects vis-à-vis human activity recognition in video. Appl. Intell. 2018, 48, 204–219.

44. Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. Psychol. Rev. 1958, 65, 386–408.

45. Hopfield, J.J. Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. USA 1982, 79, 2554–2558.

46. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. Neural Comput. 1997, 9, 1735–1780.

47. Hasan, M.; Islam, S.; Abdullah, S. Robust Pose-Based Human Fall Detection Using Recurrent Neural Network. In Proceedings of the 2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON), Dhaka, Bangladesh, 29 November–1 December 2019; pp. 48–51.

48. Kumar, D.; Ravikumar, A.K.; Dharmalingam, V.; Kafle, V.P. Elderly Health Monitoring System with Fall Detection Using Multi-Feature Based Person Tracking. In Proceedings of the 2019 ITU Kaleidoscope: ICT for Health: Networks, Standards and Innovation (ITU K), Atlanta, GA, USA, 4–6 December 2019.

49. Hua, M.; Nan, Y.; Lian, S. Falls Prediction Based on Body Keypoints and Seq2Seq Architecture. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea, 27 October–3 November 2019; pp. 1251–1259.

50. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. arXiv 2014, arXiv:1409.3215.

51. Chen, Y.; Li, W.; Wang, L.; Hu, J.; Ye, M. Vision-Based Fall Event Detection in Complex Background Using Attention Guided Bi-Directional LSTM. IEEE Access 2020, 8, 161337–161348.

52. Hubel, D.H.; Wiesel, T.N. Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. J. Physiol. 1962, 160, 106–154.

53. Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern. 1980, 36, 193–202.

54. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. Proc. IEEE 1998, 86, 2278–2324.

55. Tygert, M.; Bruna, J.; Chintala, S.; LeCun, Y.; Piantino, S.; Szlam, A. A Mathematical Motivation for Complex-Valued Convolutional Networks. Neural Comput. 2016, 28, 815–825.

56. Wang, X.; Jia, K. Human Fall Detection Algorithm Based on YOLOv3. In Proceedings of the 2020 IEEE 5th International Conference on Image, Vision and Computing (ICIVC), Qingdao, China, 23–25 July 2020; pp. 50–54.

57. Feng, Q.; Gao, C.; Wang, L.; Zhao, Y.; Song, T.; Li, Q. Spatio-temporal fall event detection in complex scenes using attention guided LSTM. Pattern Recognit. Lett. 2020, 130, 242–249.

58. Wang, B.-H.; Yu, J.; Wang, K.; Bao, X.-Y.; Mao, K.-M. Fall Detection Based on Dual-Channel Feature Integration. IEEE Access 2020, 8, 103443–103453.

59. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3645–3649.

60. Maldonado-Bascón, S.; Iglesias-Iglesias, C.; Martín-Martín, P.; Lafuente-Arroyo, S. Fallen People Detection Capabilities Using Assistive Robot. Electron. 2019, 8, 915.

61. Carlier, A.; Peyramaure, P.; Favre, K.; Pressigout, M. Fall Detector Adapted to Nursing Home Needs through an Optical-Flow based CNN. In Proceedings of the 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, 20–24 July 2020; Volume 2020, pp. 5741–5744.

62. Cai, X.; Li, S.; Liu, X.; Han, G. Vision-Based Fall Detection With Multi-Task Hourglass Convolutional Auto-Encoder. IEEE Access 2020, 8, 44493–44502.

63. Min, W.; Cui, H.; Rao, H.; Li, Z.; Yao, L. Detection of Human Falls on Furniture Using Scene Analysis Based on Deep Learning and Activity Characteristics. IEEE Access 2018, 6, 9324–9335.

64. Ma, C.; Shimada, A.; Uchiyama, H.; Nagahara, H.; Taniguchi, R.-I. Fall detection using optical level anonymous image sensing system. Opt. Laser Technol. 2019, 110, 44–61.

65. Fan, Y.; Levine, M.D.; Wen, G.; Qiu, S. A deep neural network for real-time detection of falling humans in naturally occurring scenes. Neurocomputing 2017, 260, 43–58.

66. Sun, G.; Wang, Z. Fall detection algorithm for the elderly based on human posture estimation. In Proceedings of the 2020 Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), Busan, Korea, 13–16 October 2020; pp. 172–176.

67. Zhang, J.; Wu, C.; Wang, Y. Human Fall Detection Based on Body Posture Spatio-Temporal Evolution. Sensors 2020, 20, 946.

68. Chen, Y.; Kong, X.; Meng, L.; Tomiyama, H. An Edge Computing Based Fall Detection System for Elderly Persons. Procedia Comput. Sci. 2020, 174, 9–14.

69. Pattamaset, S.; Charoenpong, T.; Charoenpong, P.; Chianrabutra, C. Human fall detection by using the body vector. In Proceedings of the 2017 9th International Conference on Knowledge and Smart Technology (KST), Chonburi, Thailand, 1–4 February 2017; pp. 162–165.

70. Kasturi, S.; Jo, K.-H. Human fall classification system for ceiling-mounted kinect depth images. In Proceedings of the 2017, 17th International Conference on Control, Automation and Systems (ICCAS), Jeju, Korea, 18–21 October 2017; pp. 1346–1349.

71. Kalinga, T.; Sirithunge, C.; Buddhika, A.; Jayasekara, P.; Perera, I. A Fall Detection and Emergency Notification System for Elderly. In Proceedings of the 2020 6th International Conference on Control, Automation and Robotics (ICCAR), Singapore, 20–23 April 2020; pp. 706–712.

72. Panahi, L.; Ghods, V. Human fall detection using machine vision techniques on RGB–D images. Biomed. Signal Process. Control. 2018, 44, 146–153.

73. Mastorakis, G.; Ellis, T.; Makris, D. Fall detection without people: A simulation approach tackling video data scarcity. Expert Syst. Appl. 2018, 112, 125–137.

74. Junejo, I.N.; Foroosh, H. Euclidean path modeling for video surveillance. Image Vis. Comput. 2008, 26, 512–528.

75. Kong, X.; Meng, Z.; Nojiri, N.; Iwahori, Y.; Meng, L.; Tomiyama, H. A HOG-SVM Based Fall Detection IoT System for El derly Persons Using Deep Sensor. Procedia Comput. Sci. 2019, 147, 276–282.

76. Hernandez-Mendez, S.; Maldonado-Mendez, C.; Marin-Hernandez, A.; Rios-Figueroa, H.V. Detecting falling people by autonomous service robots: A ROS module integration approach. In Proceedings of the 2017 International Conference on Electronics, Communications and Computers (CONIELECOMP), Cholula, Mexico, 22–24 February 2017; pp. 1–7.

77. Maldonado, C.; Rios-Figueroa, H.V.; Mezura-Montes, E.; Marin, A.; Marin-Hernandez, A. Feature selection to detect fall en pose using depth images. In Proceedings of the 2016 International Conference on Electronics, Communications an d Computers (CONIELECOMP), Cholula, Mexico, 24–26 February 2016; pp. 94–100.

78. Merrouche, F.; Baha, N. Depth camera based fall detection using human shape and movement. In Proceedings of the 2016 IEEE International Conference on Signal and Image Processing (ICSIP), Beijing, China, 13–15 August 2016; pp. 586–590.

79. Abobakr, A.; Hossny, M.; Nahavandi, S. A Skeleton-Free Fall Detection System From Depth Images Using Random De cision Forest. IEEE Syst. J. 2017, 12, 2994–3005.

80. Lewandowski, B.; Wengefeld, T.; Schmiedel, T.; Gross, H.-M. I see you lying on the ground–Can I help you? Fast fallen person detection in 3D with a mobile robot. In Proceedings of the 2017 26th IEEE International Symposium on Robot a nd Human Interactive Communication (RO-MAN), Lisbon, Portugal, 28 August–1 September 2017; pp. 74–80.

81. Labayrade, R.; Aubert, D.; Tarel, J.-P. Real time obstacle detection in stereovision on non flat road geometry through "v -disparity" representation. In Proceedings of the Intelligent Vehicle Symposium, Versailles, France, 17–21 June 2002.

82. Schmiedel, T.; Einhorn, E.; Gross, H.-M. IRON: A fast interest point descriptor for robust NDT-map matching and its app lication to robot localization. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 3144–3151.

83. Antonello, M.; Carraro, M.; Pierobon, M.; Menegatti, E. Fast and robust detection of fallen people from a mobile robot. I n Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 4159–4166.

84. Fischler, M.A.; Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analy sis and Automated Cartography. Read. Comput. Vis. 1987, 24, 726–740.

85. Otanasap, N.; Boonbrahm, P. Pre-impact fall detection approach using dynamic threshold based and center of gravity i n multiple Kinect viewpoints. In Proceedings of the 2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE), NakhonSiThammarat, Thailand, 12–14 July 2017; pp. 1–6.

86. Harrou, F.; Zerrouki, N.; Sun, Y.; Houacine, A. Vision-based fall detection system for improving safety of elderly people. IEEE Instrum. Meas. Mag. 2017, 20, 49–55.

87. Rabiner, L.; Juang, B. An introduction to hidden Markov models. IEEE ASSP Mag. 1986, 3, 4–16.

88. Htun, S.N.; Zin, T.T.; Tin, P. Image Processing Technique and Hidden Markov Model for an Elderly Care Monitoring Syst em. J. Imaging 2020, 6, 49.

89. Natarajan, P.; Nevatia, R. Online, Real-time Tracking and Recognition of Human Actions. In Proceedings of the 2008 IE EE Workshop on Motion and video Computing, Copper Mountain, CO, USA, 8–9 January 2008; pp. 1–8.

90. Taghvaei, S.; Jahanandish, M.H.; Kosuge, K. Auto Regressive Moving Average Hidden Markov Model for Vision-based Fall Prediction-An Application for Walker Robot. Assist. Technol. 2016, 29, 19–27.

Retrieved from https://encyclopedia.pub/entry/history/show/16916