Vision-based Fall Detection using Body Geometry*

Beddiar Djamila Romaissa[1,2][0000−0002−1371−3881], Oussalah Mourad[1][0000−0002−4422−8723], Nini Brahim[2], and Bounab Yazid[1]

[1] Center for Machine Vision and Signal Analysis, University of Oulu, Finland
[2] Research Laboratory on Computer Science's Complex Systems, University Laarbi Ben M'hidi, Oum El Bouaghi, Algeria
ad_beddiar@esi.dz
Djamila.Beddiar@oulu.fi

Abstract. Falling is a major health problem that causes thousands of deaths every year, according to the World Health Organization. Fall detection and fall prediction are both important tasks that should be performed efficiently to enable accurate medical assistance to vulnerable population whenever required. This allows local authorities to predict daily health care resources and reduce fall damages accordingly. We present in this paper a fall detection approach that explores human body geometry available at different frames of the video sequence. Especially, the angular information and the distance between the vector formed by the head -centroid of the identified facial image- and the center hip of the body, and the vector aligned with the horizontal axis of the center hip, are then used to construct distinctive image features. A two-class SVM classifier is trained on the newly constructed feature images, while a Long Short-Term Memory (LSTM) network is trained on the calculated angle and distance sequences to classify falls and non-falls activities. We perform experiments on the Le2i fall detection dataset and the UR FD dataset. The results demonstrate the effectiveness and efficiency of the developed approach.

Keywords: Fall Detection, Elderly assistance, SVM classification, Deep Learning, LSTM, Pretrained models.

1 Introduction

Performing regular daily life activities by elderly population can cause fall. This is due to inherent factors such as age-related biological changes, neurological disorders, physiological health profile and environmental conditions. In general, fall can result because of sudden loss of balance, stability, dizziness or vertigo

* This work is partly supported by the Algerian Residential Training Program Abroad Outstanding National Program (PNE) that supported the first author stay at University of Oulu and the European YougRes project (# 823701), which are gratefully acknowledged.

during daily life movements. It can also be caused by chronic diseases, cognitive impairment, the use of walking aid or multiple medications, gait and visual deficit [21]. Falling is an abnormal human activity that occurs infrequently and unpredictably. It is defined in [1] as an event which results in a person coming to rest inadvertently on the ground or the floor or any other lower level.

It is acknowledged that fall is one of the major public health problems in the world that should be carefully addressed and appears to be the second leading cause of accidental or unintentional injury deaths [1]. Therefore, Fall detection, Fall classification and Fall prediction are recognized as important research directions in the study of falls and are among the hottest topics in health-care. Indeed, the availability of efficient methods to identify and, possibly, predict fall occurrence can have a huge public impact since it may significantly minimize damages, enable efficient medical assistance and provide daily health care for vulnerable population. Moreover, missing to identify falls can expose the individual to serious health and safety risks. It has an obvious effect on individual autonomy, independence and life quality. It is to note that experiencing fall many times, may lead to Basophobia, also called fear of falling [12]. This syndrome can cause many other disorders such as lack of mobility and independence, and/or social isolation. On the other hand, reducing the time interval between falling and rescuing is essential in order to minimize the negative consequences of falls, which raises the importance of fall prediction task.

Motivated by the importance of detecting falls and the observation that the vector formed by the head and the center hip of the body is aligned horizontally and in parallel to the ground during a falling posture, while it is perpendicular to the ground axis in a sitting or standing posture, we present in this paper a novel machine learning like approach for Fall Detection. Our approach relies on the calculation of the angle and the distance between the vector formed by the head and the center hip of the body to the vector formed on the horizontal axis of the center hip of the body. For each video sequence, we calculate, the above mentioned angle and distance among all the frames. The computed angles and distances form the new feature sets that characterize the video sequences. We train an LSTM network on these features to recognize fall and non-fall activities. Furthermore, we construct new images using these angle and distance sequences so that each video sequence is represented by one image of its corresponding angles and distances. Then, a two-class SVM is trained on these images to detect fall and non-fall activities. We use the Le2i dataset [5] and the UR FD dataset [13] to evaluate the performance of our method, different metrics have been employed to quantify the quality of the designed SVM classifier in detecting falls. The experimental results indicate that our approach is practical and achieves good accuracy in detecting falls. The main contributions of our proposal can be summarized into:

– A new method for downsampling the videos using optical flow in order to keep only frames with significant motion is put forward. This allows us to reduce the number of frames of the video to be executed in subsequent reasoning.

- A a new research dataset related to our manual annotation task containing the 2D coordinates of both center hip of the body and the head centroid (of facial representation) available from each frame of the video data is made available to research community.
- Calculating the angle and the distance between the head centroid and the center hip of the body.
- Tracking the variation of the above angular estimation across all frames of the re-sampled video.
- A new SVM-binary classification that distinguishes fall from non-fall scenarios using the sequence of angles and distances pertaining to each video has been devised.
- Contribution of other potential feature sets are explored and exploited for representing video sequences.
- A comparison between LSTM and SVM classification results has been carried out.

The rest of this paper is organized as follows. First, we briefly provide background and previous research related to vision-based Fall detection (FD) in Section 2. Section 3 outlines our approach. Then, we describe and discuss in Section 4 the experimental results of our proposal on the publicly available datasets. Finally, we conclude our paper and set future directions for fall detection in Section 5.

2 Related Work

Fall detection techniques can be categorized into three major classes: ambient-based, wearable-based and vision-based systems [21]. Ambient-based systems use light, proximity, motion, and vibration sensors to collect daily life activities data and detect falls. Wearable-based systems rely on the sensors embedded in particular devices that the subject should wear in order to track his/her motion [12]. Additionally, vision-based systems use RGB or depth cameras to record the subject's activities, in indoor or outdoor environments [12]. The recorded images or videos are analyzed later to detect falls. Motivated by robustness, efficiency, ease of use and installation of the last methods, the approach that we present in this paper relates to vision-based FD. Thereby, we briefly report here some of the existing vision-based FD methods.

Roughly speaking, Vision-based FD approaches focus on meaningful fall related features extracted from the video frames such as silhouettes, body shape and skeleton information. These features are then used as input to some machine learning classifier such as SVM, KNN, Hidden Markov Models (HMM), among others, to train and later automatically detect fall and non-fall cases. For instance, [15] extracts distinctive features of human silhouettes to construct new action representations. The authors model the actions using a bag-of-words and conduct the classification using an extreme learning machine (ELM). Authors in [9] suggest robust features called History Triple Features using a generalization of the Radon Transform. Furthermore, SVM based approaches have proven their efficiency for fall detection tasks in many alternative works see, for instance, [7,8,11]. In [7], five distinct features are employed (aspect ratio, change in aspect ratio, fall angle, center speed and head speed). Authors in [8] use a normalized motion energy image to model the silhouette shape deformation features, while [11] proposes a novel descriptor, called Trajectory Snippet Histograms, to model the rapid motions change. They used Bag of Words to describe each video clip and train an SVM for unusual videos classification. In addition, shape and motion features are tracked to detect falls using a single camera based system in [18]. Likewise, [22] proposes to analyze dynamic appearance, shape and motion features of the target person and then characterize the human falls with simple velocity statistics of moving features. Authors in [4] suggest a vision-based fall detection system for elderly living alone. The system relies on the optical flow estimation to estimate the speed of motion and to deduce the fall activity accordingly, while comparing the last positions of the target.

On the other hand, many vision-based research is devoted to fall detection using Kinect sensors. This is because depth cameras can overcome some privacy issues related to traditional camera systems. For instance, [16] proposes a real-time fall detection system based on 3D Kinect depth maps. These depth maps are used to extract 3D silhouettes features. Similarly, [20] employs Kinect sensor to acquire point cloud images and extract energy fall features. Other researchers demonstrate that using Kinect sensor alone does not provide sufficient coverage and, therefore, cannot yield robust and efficient fall detection capabilities.

With the advance in Deep Learning (DL) approaches, many researchers put forward DL based approach for fall detection tasks. For instance, [6] proposes a real-time fall detection approach that allows the capture of RGB video streams, individual's position estimation and, thereby, fall detection likelihood, which then generates potential alert messages to caregivers with registered audio and video. In [19], the authors present a novel FD method based on Convolutional Neural Networks (CNN) using optical flow images. Moreover, transfer learning is widely used to take advantage of pre-trained models by reusing their network weights or fine-tuning the classification layers. For instance, [3] was able to efficiently detect falls using a CNN Alexnet architecture. In [10], the authors present a two-stream approach based on MobileVGG network. Similarly, the authors of [10] combine an improved lightweight VGG network and the motion characteristics of the human body. Likewise, a 3D CNN-based method combined with long short-term memory (LSTM) is also presented in [14]. The 3D CNN is used to extract motion and spatial features while the LSTM-based spatial visual attention scheme is incorporated to locate the fall in each frame. Authors in [2] present a fall detection system based on LSTM, using location features from the group of available joints in the human body. Inspired by the aforementioned work, we focus in this paper on vision-based fall detection using LSTM for classification of angle and distance features, that are extracted from video sequences. Transfer learning is performed to take advantage of the strong ability of the Resnet50 model in extracting significant features that were later fed to our two-class SVM classifier.

3 Proposed Method

The starting point in our developed methodology consists in identifying relevant features that can genuinely distinguish fall from non-fall activities. In this respect, we noticed that when a person is sitting or standing, the head and the center hip form a vector which is perpendicular to the horizontal axis passing through the center hip, as illustrated in “Fig. 1 (a)” and “Fig. 1 (b)”. The horizontal axis is defined as a straight line parallel to the *X_axis* and passing through the center hip. In contrast, when a person is in a lying or falling posture, this vector is approximately aligned and in parallel to the horizontal axis of the center hip of the body. In addition, sitting slumped to one side leads to forming an angle of around 45° or 120° between the mentioned vector and the horizontal axis as shown in “Fig. 2”. The angle value depends on the degree of slump sitting. However, the posture is considered lying or falling when this value is close to 0° or 180° as shown in “Fig. 3”.

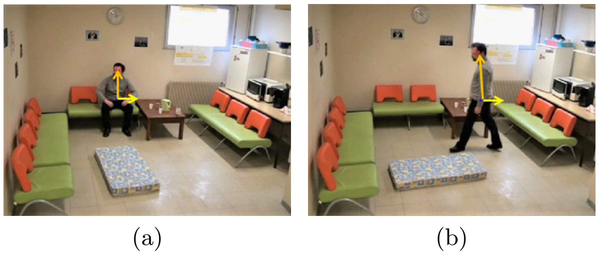

(a) (b)

Fig. 1. Samples from the Le2i fall detection dataset representing the angle α in (a) sitting and (b) standing postures. The value of α is around 90° in both postures.

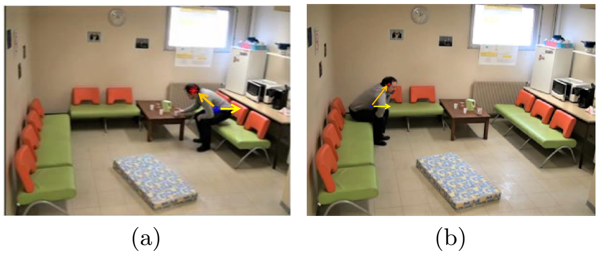

(a) (b)

Fig. 2. Samples from the Le2i fall detection dataset representing the angle α in (a) bending to the left posture and (b) bending to the right posture. The value of α is around 120° and 45° respectively.

To illustrate our approach mathematically, we refer to the head centroid by the point $H(x_h,y_h)$ and to the center hip of the body by the point $B(x_b,y_b)$. Let $\overrightarrow{U}$ be the vector from H to B. Similarly, let $\overrightarrow{V}$ be the vector joining the point B to the point $C(x_c,y_c)$. The point C is defined such that $x_c > x_b$ and $y_c = y_b$.

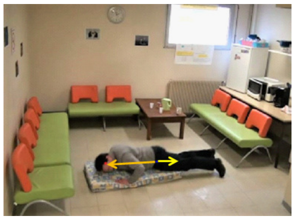


Fig. 3. Samples from the Le2i fall detection dataset representing the angle α in a falling posture. The value of α is around 180°.

Relying on this observation, we calculate for each video, the angle α formed between $\vec{U}$ and $\vec{V}$ and the distance γ between the head and the center hip of the body (i.e: the magnitude of the vector $\vec{U}$) for all its frames. These notations are used along the paper. Each video is therefore characterized by a feature vector containing the sequence of the computed angles and distances.

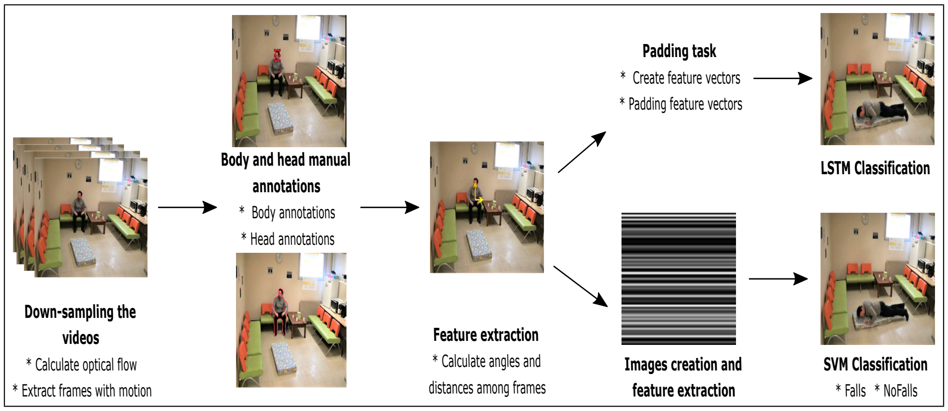


Fig. 4. The pipeline of our proposed fall detection approach

The first step of our approach consists in down-sampling the videos to reduce the number of their frames by keeping only frames which contain motion. This helps us to reduce both the computational and man-power burden effort in the next steps. Then, we use these frames to manually annotate the head and the body. We then calculate their centroid and center hip respectively. Next, we calculate the angle α and the distance γ and represent each video with a sequence of angles α_i and a sequence of distances γ_i, where i represents the frame's index. Once these sequences are created for each video, we distinguish two scenarios. In the first scenario we train an LSTM network on these features (angles,distances) and classify videos accordingly into falls and non-falls. For this purpose, we also devised a data augmentation strategy that handles the potential mismatch of input size to the LSTM network. For this purpose, we performed a simple padding task where the feature vector (angle and distance value) of the first frame is duplicated and concatenated in order to yield the same dimension as the largest sequence. In the second scenario, we create two sets of images that are fed later to a pretrained network in order to extract distinctive features. The extracted features are then trained with a two-class SVM classifier to detect falls from daily life activities. “Fig. 4” outlines the pipeline of our proposed fall detection approach. It is summarized in 4 steps as follows.

3.1 Down-sampling the videos

The length of the Le2i dataset video sequences vary from 30 seconds to 4 minutes with a frame rate of 25 frames per second. Indeed, the extraction of video frames can result into more than 1000 frames, which makes the process of manual head annotations very exhausting. Therefore, instead of using all the frames, we down-sample each video frames in order to reduce the intensive computational processing. Inspired by [17] where a video summarization technique based on motion analysis using optical flow computations to identify global extrema and local minimum between two maximums in the motion was presented, we also use the optical flow (OF) to down-sample our videos. For that, we exploit the Horn-Schunck method to estimate the motion in the video sequences. We observe that falls, in general, occur fast and can be characterized by a significant motion change among frames. Therefore, we keep those frames that represent meaningful motion change and construct a re-sampled video sequence using these frames, with a rate of 25 frames per second. It is to note that the new re-sampled videos contain less frames than the original videos. For example, a video of 1607 frames is reduced to 578 frames and another video of 1283 frames is downsampled to 583 frames. The downsampling rate is variable and depends on the mean values of the optical flow components of each video.

More specifically, to automate this downsampling process, we first estimate the optical flow among all the frames using the Horn-Schunck method that consists in resolving the constraint: $I_x.u + I_y.v + I_t = 0$. Where I_x, I_y, I_t are the spatiotemporal image brightness derivatives, while u and v correspond to the horizontal and the vertical optical flow components, respectively. Then, we calculate the mean of both horizontal *(Vx)* and vertical *(Vy)* components of the OF, which we call $meanVx$ and $meanVy$ respectively. Subsequently, the mean squared normalized error performance (MSE) is computed to estimate the similarity between the horizontal\vertical components of the OF of each frame and the mean value of horizontal and vertical components of the OF, respectively (*similarityVx* and *similarityVy*). “Equation (1)” demonstrates how to calculate such similarity using the MSE values.

$$similarityVz_i = \frac{1}{P}.\sum_{p-1}^{P}(Vz_i(p) - meanVz(p))^2 \tag{1}$$

Where z refers to either x or y component, P refers to the pixels of the frame, i refers to its index and p to a particular pixel of the frame i.

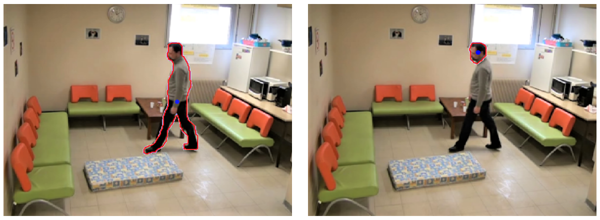
Frames that have a similarity $similarityVx_i$ (resp. $similarityVy_i$) above or equal their mean similarity $meanSimVx$ (resp. $meanSimVy$) are preserved while others are removed to construct the re-sampled video. Besides, the maintained frames should respect the conditions given by "(2)".

$$\left\{\begin{matrix} similarityVx_i >= meanSimVx \\ similarityVy_i >= meanSimVy \end{matrix}\right\} \tag{2}$$

Our videos are resampled and the number of frames of each video is reduced to almost the half. Our aim behind this is to facilitate the next step of our proposal which is done manually.

3.2 Body and head manual annotations

Once, the videos are re-sampled, and as our work focuses on the features extracted from the body geometry, we manually annotate the individual's head position in video frame and calculate its centroid to avoid subsequent problems originated in the tracking algorithm. More specifically, the annotation of each frame contains the frame's index, the localization of the head presented in terms of bounding box and the coordinates of the head centroid. In addition, we manually annotate the human body in video frames and estimate the center hip by calculating the centroid of the shape that surrounds the individual's body. Figure 5 illustrates the manual annotation of the body and the head, where the blue point corresponds to an estimation of the center hip of body in (a) and the centroid of the head in (b). The samples were taken from the Le2i dataset.



(a) (b)

Fig. 5. Samples from the Le2i fall detection dataset representing the manual annotation of a) the center hip of the body and b) the head.

The head centroid and the center hip of the body are used later to calculate their associated distance γ and the angle α between the vector $\overrightarrow{U}$ and the vector formed by the horizontal axis corresponding to the x coordinate of the center hip called $\overrightarrow{V}$.

For the angle calculus, we can calculate its cosine value and deduce the corresponding angle. The cosine is calculated using the law of cosines and the Euclidean norm is used to calculate the magnitude of vectors. "Equation (3)" illustrates how we calculate the cosine of the angle α. $\overrightarrow{HC}$ refers to the vector between the head centroid and the axis point C and $\left\|\overrightarrow{X}\right\|$ is the Euclidean norm of the vector $\overrightarrow{X}$.

$$cos(\alpha) = \frac{-\left\|\overrightarrow{HC}\right\|^2 + \left\|\overrightarrow{U}\right\|^2 + \left\|\overrightarrow{V}\right\|^2}{2.\left\|\overrightarrow{U}\right\|.\left\|\overrightarrow{V}\right\|} \tag{3}$$

We therefore calculate the distance between the head and the center hip of the body among all the video frames using the Euclidean norm.

3.3 Feature extraction

As mentioned above, we discern two scenarios. In the first one, we construct our feature vectors using angles and distances. The angles and the distances are calculated between the vectors $\overrightarrow{U}$ and $\overrightarrow{V}$ among all frames of the re-sampled videos. Therefore, each video is characterized either by the feature vector $V = \{\alpha_1, \alpha_2, \alpha_3 ... \alpha_i\}$ or $V = \{[1, \alpha_1,\gamma_1], [2,\alpha_2,\gamma_2], [3,\alpha_3,\gamma_3] ... [i,\alpha_i,\gamma_i]\}$ where i is the index of the video frame. Since the video sequences do not contain the same number of frames, these feature vectors are of different lengths and could not be fed directly to the classifier which require the input size to be fixed. For that, we perform a padding strategy that allows to keep all the vectors of the same dataset with the same size. The new length is calculated to be the maximum value of the vectors' lengths. So, each vector is extended to the new length by adding the new values to its beginning. In order to not influence the feature vector with random new values, we fill them out using the first value of the vector: angle and distance of the first frame. Feature vectors are then fed to a classifier: SVM or LSTM for fall detection. For example, for a video V_1 characterized by $V_1 = \{[1,\alpha_1,\gamma_1], [2,\alpha_2,\gamma_2], [3,\alpha_3,\gamma_3] ... [k,\alpha_k,\gamma_k]\}$ where K represents the number of its frames. Let us refer to the maximum value of all video lengths with Max, where $K <= Max$. We add (Max-K) elements of value $[\alpha_1,\gamma_1]$ at the beginning of V_1, so V_1 becomes $V_1 = \{[1,\alpha_1,\gamma_1], \{[2,\alpha_1,\gamma_1],...,\{[Max\text{-}K,\alpha_1,\gamma_1],\{[1+Max\text{-}K,\alpha_1,\gamma_1] [2+Max\text{-}K,\alpha_2,\gamma_2], [3+Max\text{-}K,\alpha_3,\gamma_3] ... [k,\alpha_k,\gamma_k]\}$

In the second scenario, we use only the angles calculated above to construct the first set of images (gray level images). Hence, the feature vector $V = \{\alpha_1, \alpha_2, \alpha_3 ... \alpha_i\}$ is employed to construct the newly created gray level image for video V. However, we concatenate angles and distances to construct the second set of images (RGB images). The newly created RGB image for the video V from the second set is made using the feature vector $V = \{[1, \alpha_1,\gamma_1], [2,\alpha_2,\gamma_2], [3,\alpha_3,\gamma_3] ... [i,\alpha_i,\gamma_i]\}$ where values of i build the first channel, values α_i build the second channel and values γ_i build the third channel respectively. Each video is characterized by an image from the first set and an image from the second set. This way, these images encode the angle sequences and the distance sequences taking into account the temporal aspect of the video illustrated by the first channel (the video frames). We give examples of created gray-level and RGB images of falls in “Fig. 6 (a)”, “Fig. 6 (b)” and non-fall activities in “Fig. 6 (c)” and “Fig. 6 (d)”.

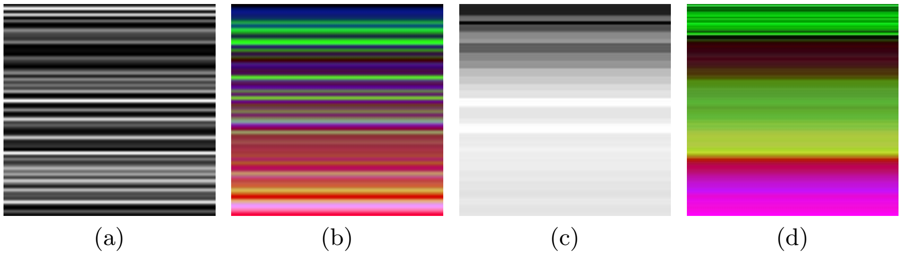


Fig. 6. Samples of created images from only angles (a and c) and created images from angles and distances (b and d). (a) and (b) represent falls while (c) and (d) represent non-falls.

3.4 Classification

We train a Long short-term memory (LSTM) network using the sequence of both angles and distances in the first scenario to detect falls and non-falls. Besides, to make the learning faster, we construct our LSTM model using a biLSTM layer that allows us to access data in both forward and reverse directions.

To detect falls in the second scenario, distinctive features are extracted from our two sets of feature images using a pretrained model. The first set consists in images constructed from angles only while the second set consists in images constructed using frames, angles and distance sequences. In our approach, we use activations of the Resnet50 and the AlexNet networks as our features. Then, we feed them to a two-class SVM classifier to distinguish between falls and daily life activities.

4 Experimental Results and Discussion

Fall is a kind of unpredictable action that occurs infrequently. Due to rarity of occurrence of falls, most existing fall detection datasets are set up by simulated fall data. The lack of such benchmark datasets and real fall data make the evaluation process of fall detection systems hard and less convincing. We evaluate our approach on the publicly available Le2i FD [5] and UR Fall Detection [13] datasets. In the subsequent section, we present these two datasets as well as the evaluation metrics employed to evaluate the performance of our proposal and, finally, the experimental results we obtained.

4.1 Experimental setup

We evaluate firstly the results obtained from our SVM and LSTM models that are trained on two configurations of features. The first feature set is composed of angles only while the second set is composed of angles and distances. Next, we evaluate the features extracted from our constructed images using Resnet50 model to those extracted using AlexNet model. We compare as well the results obtained from images constructed from (1) angles and; (2) angles and distances.

Fall detection is a binary classification problem where the classifier should specify the existence or absence of a fall in a video sequence. Sensitivity and specificity are most accurate to evaluate the performance of such system. For the purpose of experiment evaluation, we calculate Accuracy, Precision, Recall, F1_measure for each testing case.

We apply, in the same way as [19], a k-fold cross validation to our LSTM and SVM models with k=5 where the UR fall detection and the Le2i datasets were randomly split into k equal size subsets. At each iteration of the 5 iterations, we compose the training and validation sets with 4 subsets and one single set, respectively. We use a data augmentation process to transform training and test sets with an optional pre-processing stage such as resizing which helped us to resize the images of the data-store in order to make them compatible with the input size of the pretrained model. Therefore, at each epoch, the training set is modified slightly to get better results and to avoid overfitting. The results are computed across the combination of all the iterations.

4.2 Datasets

The Le2i fall detection dataset contains 221 videos of 131 falls and 90 daily life activities (ADL). The different activities are recorded by a single fixed camera with a frame rate of 25 frames/s and a resolution of 320x240 pixels. All the activities are simulated by several actors and are gathered at four different locations: Home, Office, Coffee room and Lecture room. The dataset illustrates many difficulties of realistic video sequences of an elderly home or office such as variable illumination and occlusion. The manual annotations of 191 videos was given, with extra information representing the ground-truth of the fall position and the localization of the body in the image sequence.

The UR fall detection dataset contains 70 (30 falls + 40 activities of daily living) sequences [13]. Two Microsoft Kinect cameras were used to record fall events from two different perspectives where ADL were recorded with only one camera. This results into 60 fall sequences and 40 non-fall activities.

4.3 Experiment results

For the Le2i dataset, we achieve a sensitivity of 100% for the first set composed of angles only and the second set of features composed of angles and distances, both sets trained on the LSTM network. We achieve a best sensitivity of 100% for SVM trained on features extracted with Resnet50 from images built using angles. It is clear from Table 1 that the results obtained from LSTM trained on angle and distance features are higher than the results obtained from angle features only in terms of accuracy, precision and F_score.

Table 1. Performance results for our FD approach (feature images) on the Le2i subset using an AlexNet and a Resnet50 models for feature extraction

Features	**Accuracy**	**Precision**	**Recall**	**F_score**
Features + SVM				
Angle	0.692	0.875	0.700	0.778
Angle+Distance	0.731	0.842	0.800	0.821
Features + LSTM				
Angle	0.731	0.731	**1.000**	0.845
Angle+Distance	0.769	0.769	**1.000**	0.870
Feature images + AlexNet + SVM				
Angle	0.769	0.818	0.900	0.857
Angle+Distance	0.885	0.952	0.909	0.931
Feature images + Resnet50 + SVM				
Angle	**0.962**	0.952	**1.000**	**0.975**
Angle+Distance	**0.962**	**1.000**	0.950	0.974

Table 2. Performance results for our FD approach (feature images) on the UR fall detection dataset using an AlexNet and a Resnet50 models for feature extraction

Features	**Accuracy**	**Precision**	**Recall**	**F_score**
Features + SVM				
Angle	0.700	0.727	0.727	0.727
Angle+Distance	0.850	0.818	0.900	0.857
Features + LSTM				
Angle	0.600	0.579	**1.000**	0.734
Angle+Distance	0.850	0.800	**1.000**	0.889
Feature images + AlexNet + SVM				
Angle	0.950	**1.000**	0.917	**0.957**
Angle+Distance	0.920	0.923	0.923	0.923
Feature images + Resnet50 + SVM				
Angle	**0.960**	**1.000**	0.833	0.907
Angle+Distance	**0.960**	**1.000**	0.900	0.947

Table 1 illustrates the results obtained for both sets of images (constructed from angles versus angles + distances) using the activations of the AlexNet and the Resnet50 models as well as results of training an LSTM straight on the features for the Le2i dataset. We can see from this table that the results obtained from the images constructed from the angles and the distances (RGB images) are higher than the results obtained from images constructed from angles only when using Resnet50. This can lead to conclude that Resnet50 performs well on RGB images and gives more significant features. However, we notice the opposite in the results obtained for both sets of images using the activations of the AlexNet model.

Similarly for the UR fall detection dataset, a sensitivity of 100% for training the LSTM on the first set composed of angles only and the second set of features composed of angles and distances. The sensitivity obtained for SVM classification

Table 3. Comparison between performance results of our FD approach with other existing approaches on the Le2i dataset

Approaches	**Accuracy**	**Precision**	**Recall**	**F_score**
Combined curvlets + HMM [23]	**97.02%**	-	98.00%	-
OF + CNN [19]	97.00%	-	93.60%	-
ours: **Angle + Distance + Resnet50 + SVM**	96.20%	**100%**	95.00%	**97.40%**
ours: **Angle + AlexNet + SVM**	76.90%	81.80%	90.00%	85.70%
ours: **Angle + Distance + LSTM**	76.90%	76.90%	**100%**	87.00%

Table 4. Comparison between performance results of our FD approach with other existing approaches on the UR Fall detection dataset

Approaches	**Accuracy**	**Precision**	**Recall**	**F_score**
Combined curvlets + HMM [23]	**96.88%**	-	-	-
OF + CNN [19]	95.00%	-	**100%**	-
ours: **Angle + Distance + Resnet50 + SVM**	96.00%	**100%**	90.00%	94.70%
ours: **Angle + AlexNet + SVM**	95.00%	**100%**	91.70%	**95.70%**
ours: **Angle + Distance + LSTM**	85.00%	80.00%	**100%**	88.90%

of images extracted from our newly created images using the Alexnet activations as features is better than the one obtained from the Resnet50 activations for both sets of features(angles, angles+distances). However, the accuracy and the precision values are higher while using the Resnet50 activations.

We can observe from Table 1 and Table 2 as well that using the new images gave us better results than directly feeding the features vectors: angles and distances to our LSTM and SVM models. The results were by far improved by creating these images and extracting significant features from them using pretrained models.

We compare our results with [19] and [23] since we used the same protocol evaluation and same metrics, although we acknowledge the difficulty in performing reliable comparison with other state-of-art works. Table 3 and table 4 illustrate our results versus the results obtained by [19] and [23] on the Le2i dataset and UR FD dataset respectively, where we present different variants of our approach. We can see from these tables that we outperformed the aforementioned state-of-the-art results. This can be justified by the fact that our approch is not dependent on the background and illumination changes unlike [19] who are using optical flow and [23] who are combining SVM with hidden markov models. Both models depends on the RGB videos and can be influenced by illumination or occlusion.

5 Conclusion and Future Directions

We present in this paper an effective vision-based approach for fall detection based on angles calculation. Our approach allows us to construct gray-scale images of calculated angles between the head, the center hip of the target subjects and the horizontal axis passing through the center hip. Another set of images is constructed using angles and distances between the head and the center hip of the body as well. These constructed sets of images constitute our distinctive features for fall detection task. Next, an SVM classifier is used along with a pretrained model to classify the constructed images into falls and daily life activities. We compare in this paper the features extracted using both the Resnet50 and the Alexnet models. We use the Le2i dataset and the UR fall detection dataset to evaluate the performance of our approach using the accuracy, precision, recall and F_score evaluation metrics. Experimental results show that the performances of our proposed approach are comparable to that of the state-of-the-art fall detection methods and outperform [19] and [23]. However, some limitations are also noticed. For instance, it will be desirable to improve the approach to distinguish between lying and falling postures. Besides, in the future, we would also like to automatically annotate the head and the body center hip positioning of individuals from video sequences. On the other hand, there is a room for improvement in the training pipeline through a better selection of training samples inputted to our SVM and LSTM classifiers, better optimization of LSTM parameters and through pursuing a cross-dataset based approach. We have, for instance, noticed the prospect of performing a cross-view evaluation to investigate the performance of the approach when different perspectives are studied.

References

1. World health organization, who global report on falls prevention in older age. Tech. rep. (2007)
2. Adhikari, K., Bouchachia, H., Nait-Charif, H.: Long short-term memory networks based fall detection using unified pose estimation. In: Twelfth International Conference on Machine Vision (ICMV 2019). vol. 11433, p. 114330H. International Society for Optics and Photonics (2020)
3. Anishchenko, L.: Machine learning in video surveillance for fall detection. In: 2018 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT). pp. 99–102. IEEE (2018)
4. Bhandari, S., Babar, N., Gupta, P., Shah, N., Pujari, S.: A novel approach for fall detection in home environment. In: 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE). pp. 1–5. IEEE (2017)
5. Charfi, I., Miteran, J., Dubois, J., Atri, M., Tourki, R.: Optimized spatio-temporal descriptors for real-time fall detection: comparison of support vector machine and adaboost-based classification. Journal of Electronic Imaging **22**(4), 041106 (2013)
6. Ciabattoni, L., Foresi, G., Monteriù, A., Pagnotta, D.P., Tomaiuolo, L.: Fall detection system by using ambient intelligence and mobile robots. In: 2018 Zooming Innovation in Consumer Technologies Conference (ZINC). pp. 130–131. IEEE (2018)
7. Debard, G., Mertens, M., Deschodt, M., Vlaeyen, E., Devriendt, E., Dejaeger, E., Milisen, K., Tournoy, J., Croonenborghs, T., Goedemé, T., et al.: 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 (2016)

8. Feng, W., Liu, R., Zhu, M.: Fall detection for elderly person care in a vision-based home surveillance environment using a monocular camera. signal, image and video processing **8**(6), 1129–1138 (2014)
9. Goudelis, G., Tsatiris, G., Karpouzis, K., Kollias, S.: Fall detection using history triple features. In: Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments. pp. 1–7 (2015)
10. Han, Q., Zhao, H., Min, W., Cui, H., Zhou, X., Zuo, K., Liu, R.: A two-stream approach to fall detection with mobilevgg. IEEE Access **8**, 17556–17566 (2020)
11. Iscen, A., Armagan, A., Duygulu, P.: What is usual in unusual videos? trajectory snippet histograms for discovering unusualness. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 794–799 (2014)
12. Khan, S.S., Hoey, J.: Review of fall detection techniques: A data availability perspective. Medical engineering & physics **39**, 12–22 (2017)
13. Kwolek, B., Kepski, M.: Human fall detection on embedded platform using depth maps and wireless accelerometer. Computer methods and programs in biomedicine **117**(3), 489–501 (2014)
14. Lu, N., Wu, Y., Feng, L., Song, J.: Deep learning for fall detection: Three-dimensional cnn combined with lstm on video kinematic data. IEEE journal of biomedical and health informatics **23**(1), 314–323 (2018)
15. 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 journal of biomedical and health informatics **18**(6), 1915–1922 (2014)
16. Mastorakis, G., Makris, D.: Fall detection system using kinect's infrared sensor. Journal of Real-Time Image Processing **9**(4), 635–646 (2014)
17. Mendi, E., Clemente, H.B., Bayrak, C.: Sports video summarization based on motion analysis. Computers & Electrical Engineering **39**(3), 790–796 (2013)
18. Nguyen, V.A., Le, T.H., Nguyen, T.T.: Single camera based fall detection using motion and human shape features. In: Proceedings of the Seventh Symposium on Information and Communication Technology. pp. 339–344 (2016)
19. Nunez-Marcos, A., Azkune, G., Arganda-Carreras, I.: Vision-based fall detection with convolutional neural networks. Wireless communications and mobile computing **2017** (2017)
20. Peng, Y., Peng, J., Li, J., Yan, P., Hu, B.: Design and development of the fall detection system based on point cloud. Procedia computer science **147**, 271–275 (2019)
21. Ramachandran, A., Karuppiah, A.: A survey on recent advances in wearable fall detection systems. BioMed Research International **2020** (2020)
22. Yun, Y., Gu, I.Y.H.: Human fall detection in videos via boosting and fusing statistical features of appearance, shape and motion dynamics on riemannian manifolds with applications to assisted living. Computer Vision and Image Understanding **148**, 111–122 (2016)
23. Zerrouki, N., Houacine, A.: Combined curvelets and hidden markov models for human fall detection. Multimedia Tools and Applications **77**(5), 6405–6424 (2018)